



Ricardo Ferreira

Principal Developer Advocate @ Elastic

Down to the Rabbit Hole with Pulsar I/O

Wed, Jun 16, 12:15 pm PT

WHAT IS PULSAR I/O?

APACHE PULSAR



IT'S THE PLUMBING PART



"EVERYTHING IS FINE
IN THE BACKEND"

IT'S FINE





RICARDO FERREIRA

DEVELOPER ADVOCATE 

- ❑ ELASTIC COMMUNITY TEAM 
- ❑ HASHICORP AMBASSADOR 
- ❑ LEARNED ABOUT DATA I/O AT:
CONFLUENT, ORACLE, RED HAT
- ❑ DISTRIBUTED SYSTEMS, O11Y,
STREAMING SYSTEMS, DATABASES
- ❑ RIFERREI@ELASTIC.CO
- ❑ RIFERREI@RIFERREI.COM

AGENDA

- UNDERSTANDING THE PULSAR I/O ARCHITECTURE
- INSTALLING AND MANAGING PULSAR CONNECTORS
- TROUBLESHOOTING AND DEBUGGING TECHNIQUES

UNDERSTANDING THE PULSAR I/O ARCHITECTURE

THE ARCHITECTURE IS LIKE A LASAGNA 🤪

PULSAR
CONNECTORS



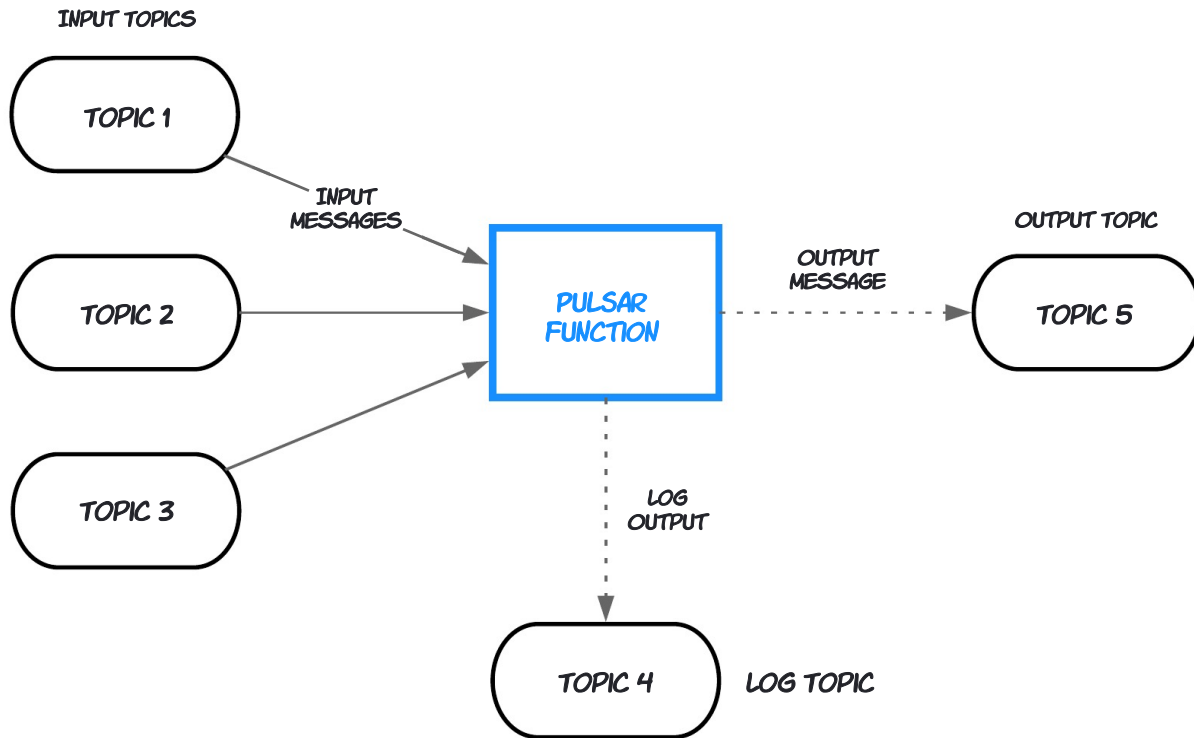
PULSAR
FUNCTIONS



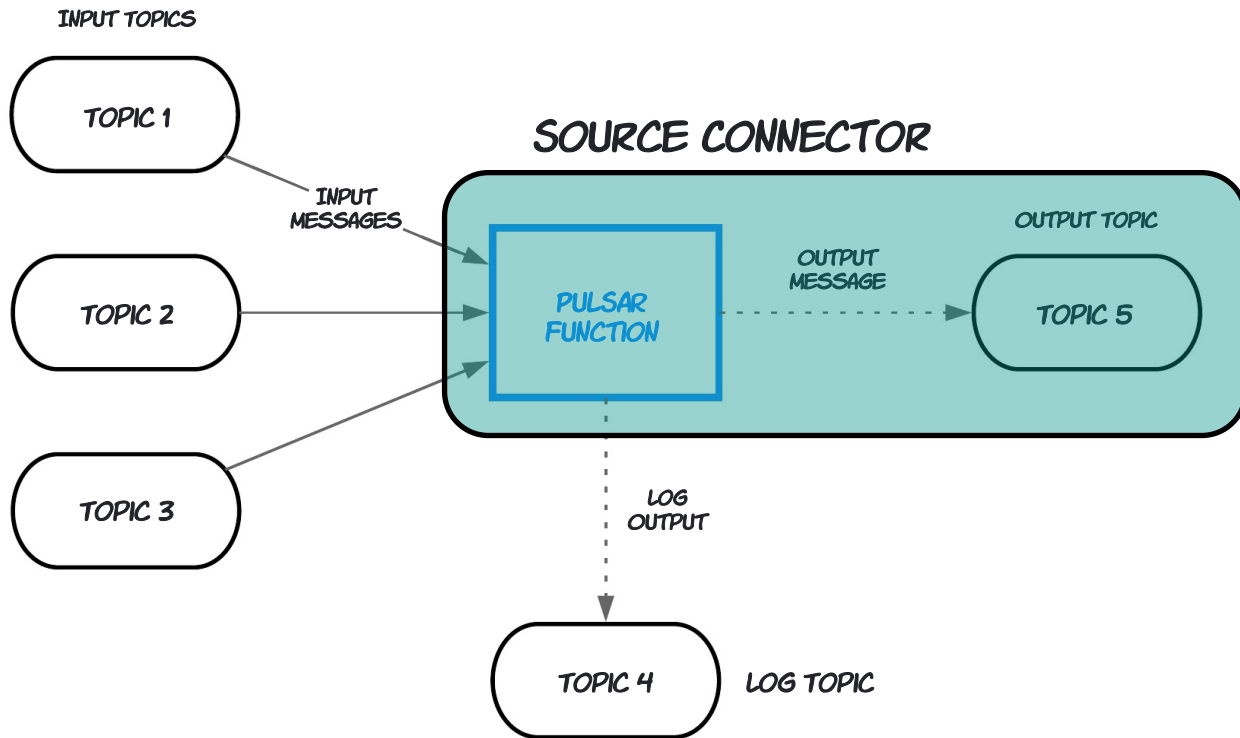
FUNCTIONS
WORKER



PULSAR FUNCTIONS PROGRAMMING MODEL



ANATOMY OF A SOURCE CONNECTOR



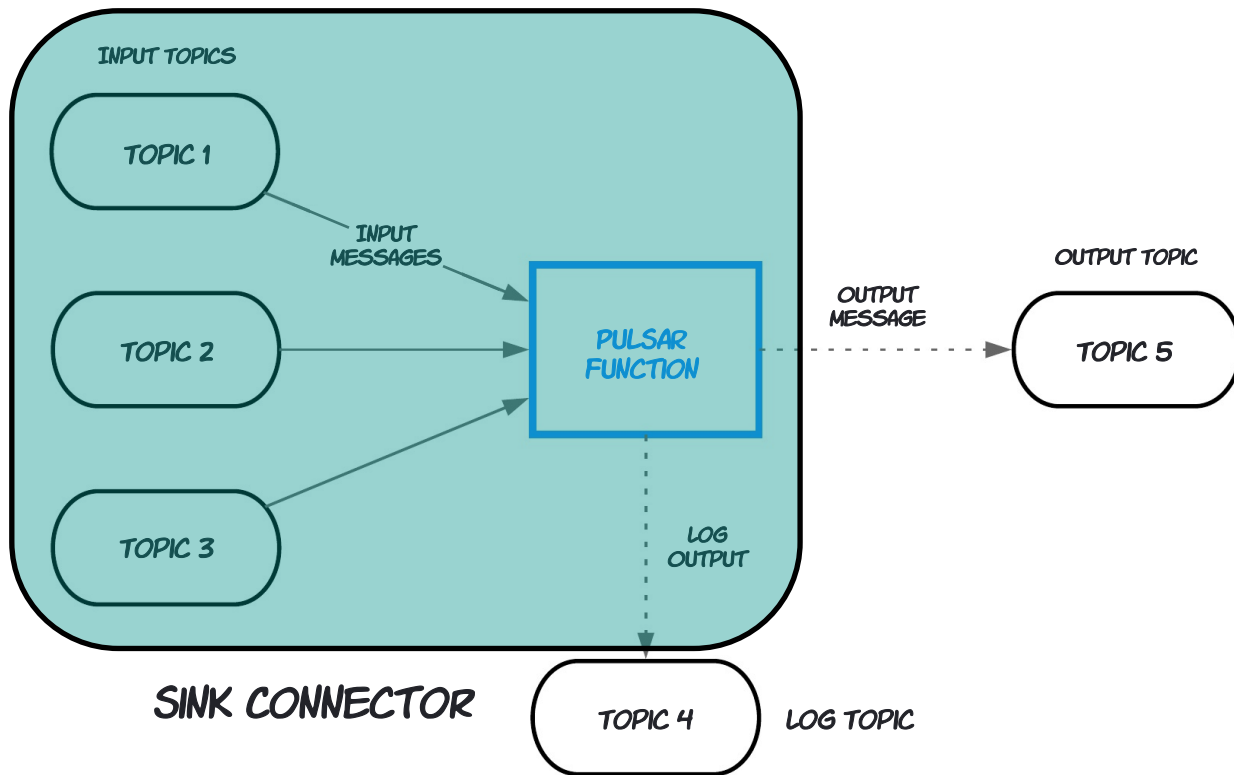
ANATOMY OF A SOURCE CONNECTOR

```
@InterfaceAudience.Public
@InterfaceStability.Stable
public interface Source<T> extends AutoCloseable {

    void open(final Map<String, Object> config,
              SourceContext sourceContext) throws Exception;

    Record<T> read() throws Exception;
}
```

ANATOMY OF A SINK CONNECTOR



ANATOMY OF A SINK CONNECTOR

```
@InterfaceAudience.Public
@InterfaceStability.Stable
public interface Sink<T> extends AutoCloseable {

    void open(final Map<String, Object> config,
              SinkContext sinkContext) throws Exception;

    void write(Record<T> record) throws Exception;

}
```

RECORDS ARE YOUR UNIT-OF-WORK

```
@InterfaceAudience.Public
@InterfaceStability.Stable
public interface Record<T> {

    default Optional<String> getTopicName() {
        return Optional.empty();
    }

    default Optional<String> getKey() {
        return Optional.empty();
    }

    default Schema<T> getSchema() {
        return null;
    }

    T getValue();

    default Optional<Long> getEventTime() {
        return Optional.empty();
    }

    default Optional<String> getPartitionId() {
        return Optional.empty();
    }
}
```

SOURCE CONNECTOR

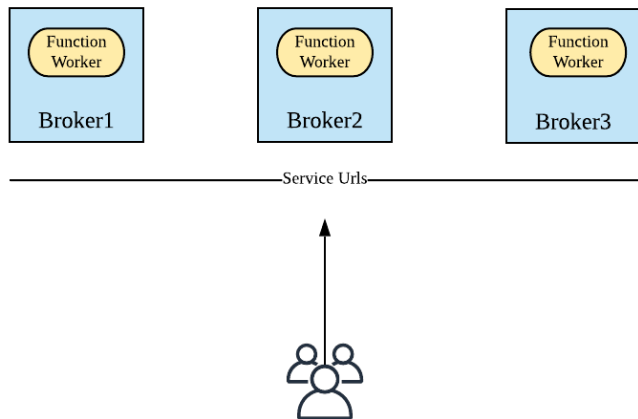
```
Record<T> read() throws Exception;
```

SINK CONNECTOR

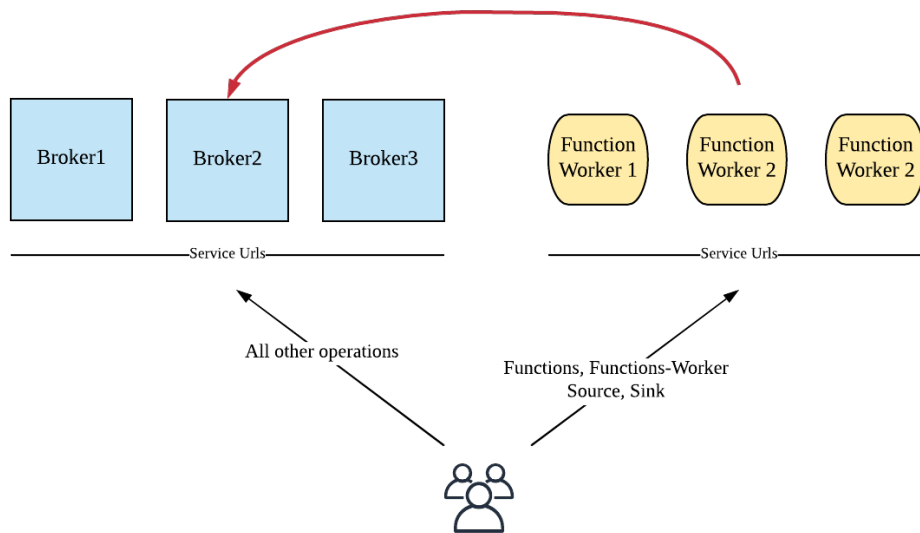
```
void write(Record<T> record) throws Exception;
```

FUNCTIONS WORKER IS HOW YOU DEPLOY

RUNNING ALONG
WITH BROKERS



RUNNING IN THEIR
OWN CLUSTER



FUNCTIONS WORKER IS HOW YOU DEPLOY

- RUNNING ALONG WITH BROKERS
 - LESS CLUSTERS TO MANAGE. BETTER OPERATIONAL SIMPLICITY.
 - NO RESOURCES ISOLATION. CPU, MEMORY, AND NETWORK IS SHARED.
- RUNNING IN THEIR OWN CLUSTER
 - RIGHT-SIZED DEPLOYMENT AS RESOURCES ARE EXCLUSIVE.
 - MORE CLUSTERS TO MANAGE. HARD TO OPERATE AT SCALE.

RUNNING ALONG WITH BROKERS

1. CONF/BROKER.CONF

```
functionsWorkerEnabled=true
```

2. CONF/FUNCTIONS_WORKER.YML

```
numFunctionPackageReplicas: 1
pulsarFunctionsNamespace: public/functions
pulsarFunctionsCluster: standalone
functionMetadataTopicName: metadata

functionRuntimeFactoryConfigs:
|   logDirectory: logs/

connectorsDirectory: ./connectors
functionsDirectory: ./functions
```

RUNNING ALONG WITH BROKERS

CHECKING IF WORKER ON BROKER IS CORRECT:

```
curl <broker-ip>:8080/admin/v2/worker/cluster
```

RESULT SHOULD BE:

```
[
  {
    "workerId": "<worker-id>",
    "workerHostname": "<worker-hostname>",
    "port": 8080
  }
]
```

RUNNING IN THEIR OWN CLUSTER

1. CONF/BROKER.CONF

```
functionsWorkerEnabled=false
```

2. CONF/FUNCTIONS_WORKER.YML

```
workerId: standalone  
workerHostname: localhost  
workerPort: 6750  
workerPortTls: 6751  
  
numFunctionPackageReplicas: 1  
pulsarServiceUrl: pulsar://localhost:6650  
pulsarWebServiceUrl: http://localhost:8080  
pulsarFunctionsCluster: standalone
```

RUNNING IN THEIR OWN CLUSTER

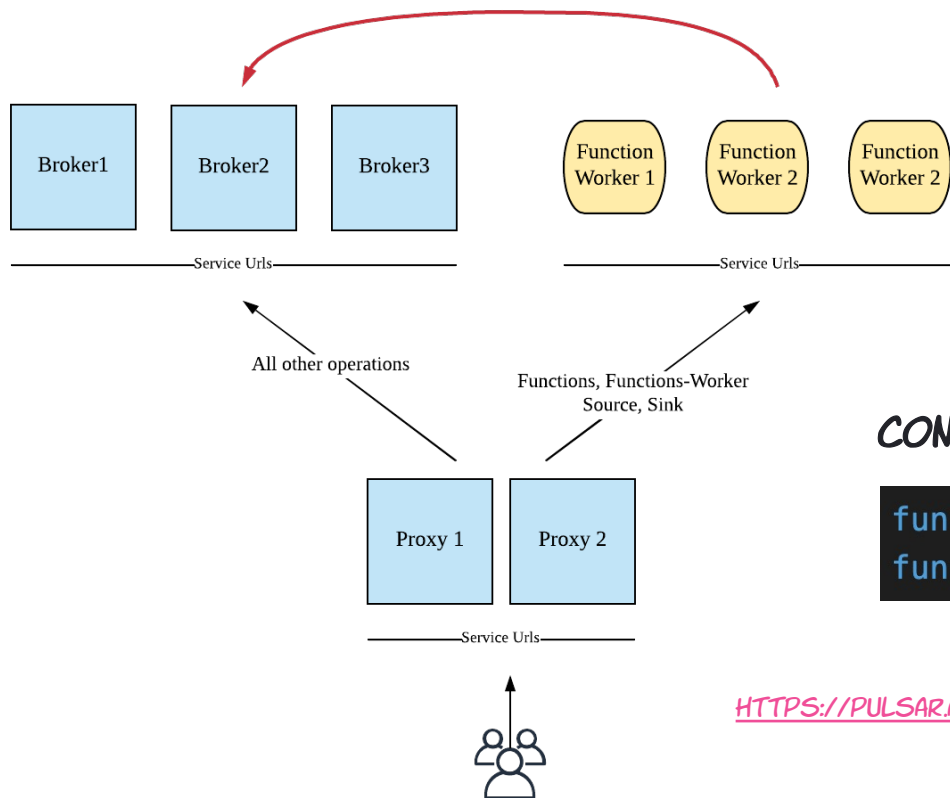
CHECKING IF FUNCTIONS WORKER IS CORRECT:

```
curl <function-worker-ip>:8080/admin/v2/worker/cluster
```

RESULT SHOULD BE:

```
[
  {
    "workerId": "<worker-id>",
    "workerHostname": "<worker-hostname>",
    "port": 8080
  }
]
```

FIXING THE ADMIN REST REQUESTS



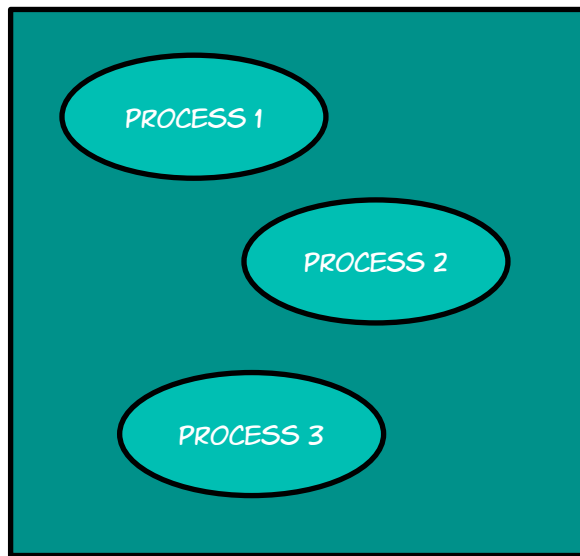
CONF/PROXY.CONF

```
functionWorkerWebServiceURL=  
functionWorkerWebServiceURLTLS=
```

[HTTPS://PULSAR.APACHE.ORG/DOCS/EN/ADMINISTRATION-PROXY](https://pulsar.apache.org/docs/en/administration-proxy)

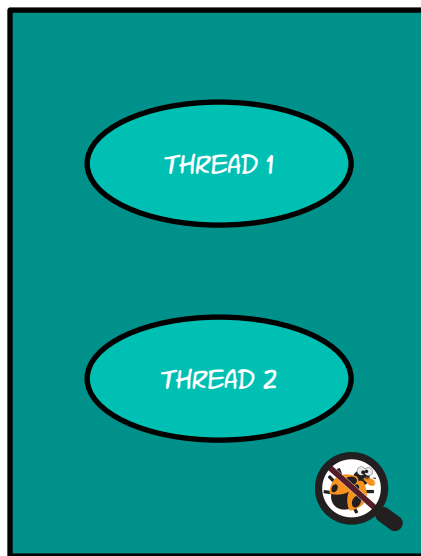
FUNCTIONS RUNTIME CONFIGURATION

PROCESS RUNTIME (DEFAULT)



HOST MACHINE

THREAD RUNTIME



JVM

KUBERNETES RUNTIME



K8S CLUSTER

FUNCTIONS RUNTIME CONFIGURATION

```
pulsar-2.7.2 → bin/pulsar-admin sinks create \  
--archive pulsar-io-elastic-search-2.7.2.nar \  
--tenant public --namespace default \  
--name pulsar_summit -sink-config-file es.yml \  
--inputs pulsar-summit \  
--parallelism 4 --ram 8589934592
```



RESOURCE	SPECIFIED AS	RUNTIME
CPU	NUMBER OF CORES	DOCKER, K8S
RAM	NUMBER OF BYTES	DOCKER, K8S
DISK	NUMBER OF BYTES	DOCKER, K8S

INSTALLING AND MANAGING PULSAR CONNECTORS

THE BAG OF GOLD FOR PULSAR CONNECTORS

STREAMNATIVE
HUB

PULSAR
CONNECTORS

PULSAR
CLI

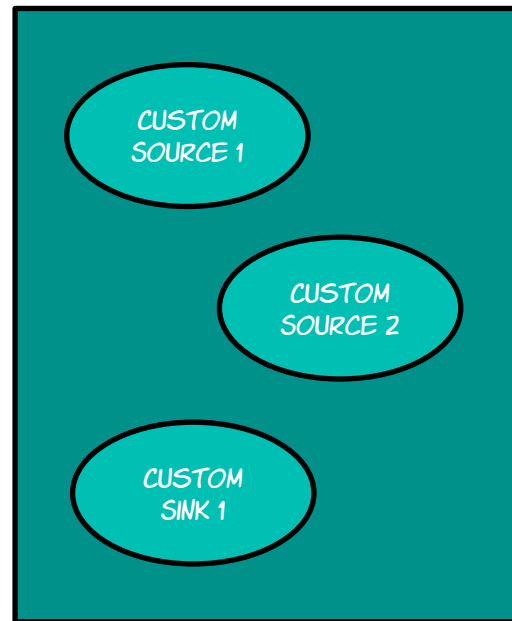
GITHUB

TWO TYPES OF CONNECTORS

1. BUILT-IN CONNECTORS



2. CUSTOM CONNECTORS



STREAMNATIVE HUB: HOME OF CONNECTORS

StreamNative Hub

Discover Apache Pulsar Ecosystem

FILTERS

Submit a Plugin

Connectors

Source

Sink

Offloaders

Protocol handlers

Data processing

Logging

Monitoring



ActiveMQ Sink

The ActiveMQ sink connector pulls messages from Pulsar topics and persist messages to ActiveMQ clusters.



StreamNative



ActiveMQ Source

The ActiveMQ source connector receives messages from ActiveMQ clusters and writes messages to Pulsar topics.



StreamNative



Aerospike Sink

The Aerospike sink connector pulls messages from Pulsar topics to Aerospike clusters



Apache Software Foundation



AMQP1_0 sink

The AMQP1_0 sink connector pulls messages from Pulsar topics and persists messages to AMQP brokers.



StreamNative



AMQP1_0 source

The AMQP1_0 source connector receives messages from AMQP service and writes messages to Pulsar topics.



StreamNative



Canal Source

The Canal source connector pulls messages from MySQL to Pulsar topics.



Apache Software Foundation



Cassandra Sink

The Cassandra sink connector pulls messages from Pulsar topics to Cassandra clusters



Apache Software Foundation



Cloud Storage Sink

The Cloud Storage sink connector pulls messages from Pulsar topics and persists messages to Cloud Storage.



StreamNative



Debezium MongoDB Source

The Debezium MongoDB source connector pulls messages from MongoDB and persists the messages to Pulsar topics



Apache Software Foundation



Debezium MySQL Source

The Debezium source connector pulls messages from MySQL and persists the messages to Pulsar topics.



Apache Software Foundation



Debezium PostgreSQL Source

The Debezium source connector pulls messages from PostgreSQL and persists the messages to Pulsar topics.



Apache Software Foundation



ElasticSearch Sink

The ElasticSearch sink connector pulls messages from Pulsar topics and persists the messages to indexes



Apache Software Foundation

[HTTPS://HUB.STREAMNATIVE.IO](https://hub.streamnative.io)

STREAMNATIVE HUB: CODE AND EXAMPLES



VERSION
2.5.1

Download

Features

- Use ElasticSearch sin...

Tags

Pulsar IO
ElasticSearch Sink

Enterprise Support

StreamNative supported

License

Apache License 2.0

Author

["ASF"]

Source

Source Code

< All hubs ...

ElasticSearch Sink

The ElasticSearch sink connector pulls messages from Pulsar topics and persists the messages to indexes.

Configuration

The configuration of the ElasticSearch sink connector has the following properties.

Property

Name	Type	Required	Default	Description
<code>elasticSearchUrl</code>	String	true	" " (empty string)	The URL of elastic search cluster to which the connector connects.
<code>indexName</code>	String	true	" " (empty string)	The index name to which the connector writes messages.
<code>typeName</code>	String	false	"_doc"	The type name to which the connector writes messages to. The value should be set explicitly to a valid type name other than "_doc" for Elasticsearch version before 6.2, and left to default otherwise.
<code>indexNumberOfShards</code>	int	false	1	The number of shards of the index.
<code>indexNumberOfReplicas</code>	int	false	1	The number of replicas of the index.
<code>username</code>	String	false	" " (empty string)	The username used by the connector to connect to the elastic search cluster. If <code>username</code> is set, then <code>password</code> should also be provided.
<code>password</code>	String	false	" " (empty string)	The password used by the connector to connect to the elastic search cluster. If <code>username</code> is set, then <code>password</code> should also be provided.

Example

Before using the ElasticSearch sink connector, you need to create a configuration file through one of the following methods.

Configuration

Elasticsearch after 6.2

- ISNN

[Edit this page](#)
[Request change](#)

Configuration
Property

Example

Configuration
Elasticsearch after
6.2
Elasticsearch
before 6.2

Usage

PULSAR WEBSITE: CODE AND EXAMPLES

 **PULSAR** 2.7.2

Docs Download Clients REST APIs ▾ Cll ▾ Blog Community ▾ Apache ▾ English

ElasticSearch sink connector

EDIT

ConfigurationPropertyExampleConfiguration

The ElasticSearch sink connector pulls messages from Pulsar topics and persists the messages to indexes.

Configuration

The configuration of the ElasticSearch sink connector has the following properties.

Property

Name	Type	Required	Default	Description
<code>elasticSearchUrl</code>	String	true	" " (empty string)	The URL of elastic search cluster
<code>indexName</code>	String	true	" " (empty string)	The index name to which the connector writes
<code>typeName</code>	String	false	"_doc"	The type name to which the connector writes The value should be set explicitly for Elasticsearch version before 6.2, otherwise it will be ignored.
<code>indexNumberOfShards</code>	int	false	1	The number of shards of the index
<code>indexNumberOfReplicas</code>	int	false	1	The number of replicas of the index
<code>username</code>	String	false	" " (empty string)	The username used by the connector to connect to the elastic search cluster If <code>username</code> is set, then <code>password</code> should also be provided.
<code>password</code>	String	false	" " (empty string)	The password used by the connector to connect to the elastic search cluster If <code>username</code> is set, then <code>password</code> should also be provided.

Example

Before using the ElasticSearch sink connector, you need to create a configuration file through one of the following methods.

Configuration

For Elasticsearch After 6.2

- JSON

```
{
  "elasticSearchUrl": "http://localhost:9200",
  "indexName": "my_index",
  "username": "scooby",
  "password": "doobie"
}
```

- YAML

```
configs:
  elasticSearchUrl: "http://localhost:9200"
  indexName: "my_index"
  username: "scooby"
  password: "doobie"
```

GETTING STARTED WITH PULSAR I/O

- FOR DEVELOPMENT

- USE THE "PULSAR-ALL" DOCKER IMAGE. INCLUDES ALL CONNECTORS.
- RUN AS A THREAD WITH THE "LOCALRUN" OPTION FROM THE ADMIN CLI.

- FOR PRODUCTION

- INSTALL THE CONNECTORS ON ALL BROKERS/FUNCTION WORKERS.
- CONNECTORS WILL BE A LIST OF .NAR FILES ON THE ./CONNECTORS.

VERIFYING YOUR PULSAR I/O SETUP

CHECKING WHICH SOURCE CONNECTORS ARE AVAILABLE

```
pulsar-2.7.2 → bin/pulsar-admin sources available-sources
```

RESULT SHOULD BE:

```
kinesis
Kinesis connectors
-----

mongo
MongoDB source and sink connector
-----

netty
Netty Tcp or Udp Source Connector
-----
```

VERIFYING YOUR PULSAR I/O SETUP

CHECKING WHICH SINK CONNECTORS ARE AVAILABLE

```
pulsar-2.7.2 → bin/pulsar-admin sinks available-sinks
```

RESULT SHOULD BE:

```
-----  
elastic_search  
Writes data into Elastic Search  
-----
```

```
flume  
flume source and sink connector  
-----
```

```
hbase  
Writes data into hbase table  
-----
```

TESTING CONNECTORS WITH LOCALRUN:

```
pulsar-2.7.2 → bin/pulsar-admin sinks localrun \  
  --archive pulsar-io-elastic-search-2.7.2.nar \  
  --tenant public --namespace default \  
  --name myLocalTest --sink-config-file es.yml \  
  --inputs myInputTopic
```

TROUBLESHOOTING AND DEBUGGING TECHNIQUES

HOW TO INVESTIGATE PULSAR I/O ISSUES

SOURCE CODE

METRICS

LOGS

TRACES

LOCALRUN

BREAKPOINTS

PROXY

STATS

PROBLEM: MY CONNECTOR IS NOT RUNNING

1. CHECK THE CONNECTOR CONFIGURATION

```
pulsar-admin sinks get \  
  --tenant public \  
  --namespace default \  
  --name pulsar_summit
```

```
{  
  "tenant": "public",  
  "namespace": "default",  
  "name": "pulsar_summit",  
  "className": "org.apache.pulsar.io.elasticsearch.ElasticSearchSink",  
  "inputSpecs": {  
    "pulsar-summit": {  
      "isRegexPattern": false,  
      "schemaProperties": {},  
      "consumerProperties": {}  
    }  
  },  
  "configs": {  
    "elasticsearchUrl": "http://localhost:9200",  
    "indexName": "pulsar-summit"  
  },  
  "parallelism": 1,  
  "processingGuarantees": "ATLEAST_ONCE",  
  "retainOrdering": false,  
  "autoAck": true,  
  "archive": "builtin://elastic_search"  
}
```

PROBLEM: MY CONNECTOR IS NOT RUNNING

2. CHECK THE CURRENT CONNECTOR STATUS

```
pulsar-admin sinks status \  
  --tenant public \  
  --namespace default \  
  --name pulsar_summit
```

```
{  
  "numInstances" : 1,  
  "numRunning" : 0,  
  "instances" : [ {  
    "instanceId" : 0,  
    "status" : {  
      "running" : false,  
      "error" : "Connection refused",  
      "numRestarts" : 41,  
      "numReadFromPulsar" : 0,  
      "numSystemExceptions" : 0,  
      "latestSystemExceptions" : [ ],  
      "numSinkExceptions" : 0,  
      "latestSinkExceptions" : [ ],  
      "numWrittenToSink" : 0,  
      "lastReceivedTime" : 0,  
      "workerId" : "c-standalone-fw-localhost-8080"  
    }  
  } ]  
}
```

PROBLEM: MY CONNECTOR IS NOT RUNNING

3. CHECK THE STATUS FROM THE TOPIC

```
pulsar-admin topics stats pulsar-summit
```

```
"msgRateIn" : 0.0,  
"msgThroughputIn" : 0.0,  
"msgRateOut" : 0.08333366928607659,  
"msgThroughputOut" : 11.500046361478569,  
"bytesInCounter" : 690,  
"msgInCounter" : 5,  
"bytesOutCounter" : 690,  
"msgOutCounter" : 5,  
"averageMsgSize" : 0.0,  
"msgChunkPublished" : false,  
"storageSize" : 690,  
"backlogSize" : 0,  
"offloadedStorageSize" : 0,  
"publishers" : [ ],
```

```
"consumers" : [ {  
  "msgRateOut" : 0.08333366928607659,  
  "msgThroughputOut" : 11.500046361478569,  
  "bytesOutCounter" : 690,  
  "msgOutCounter" : 5,  
  "msgRateRedeliver" : 0.0,  
  "chuckedMessageRate" : 0.0,  
  "consumerName" : "fb4c9",  
  "availablePermits" : 995,  
  "unackedMessages" : 0,  
  "avgMessagesPerEntry" : 591,  
  "blockedConsumerOnUnackedMsgs" : false,  
  "lastAkedTimestamp" : 1623272244416,  
  "lastConsumedTimestamp" : 1623272244350,
```

PROBLEM: MY CONNECTOR IS NOT RUNNING

4. CHECK THE CONNECTOR LOGS

`λ riferrei pulsar-2.7.2 → tail -100 logs/functions/public/default/pulsar_summit/pulsar_summit-0.log`



```
at org.apache.http.impl.nio.client.CloseableHttpAsyncClientBase$1.run(CloseableHttpAsyncClientBase$1.java:1) ... 1 more
16:42:24,971 ERROR [public/default/pulsar_summit-0] [instance: 0] JavaInstanceRunnable - [public/default/pulsar_summit-0] java.net.ConnectException: Connection refused
at org.elasticsearch.client.RestClient.extractAndWrapCause(RestClient.java:849) ~[?:?]
at org.elasticsearch.client.RestClient.performRequest(RestClient.java:259) ~[?:?]
at org.elasticsearch.client.RestClient.performRequest(RestClient.java:246) ~[?:?]
at org.elasticsearch.client.RestHighLevelClient.internalPerformRequest(RestHighLevelClient.java:1583) ~[?:?]
at org.elasticsearch.client.RestHighLevelClient.performRequest(RestHighLevelClient.java:1583) ~[?:?]
at org.elasticsearch.client.IndicesClient.exists(IndicesClient.java:1016) ~[?:?]
at org.apache.pulsar.io.elasticsearch.ElasticSearchSink.createIndexIfNeeded(ElasticSearchSink.java:70) ~[?:?]
at org.apache.pulsar.io.elasticsearch.ElasticSearchSink.open(ElasticSearchSink.java:70) ~[?:?]
at org.apache.pulsar.functions.instance.JavaInstanceRunnable.setupOutput(JavaInstanceRunnable.java:2) ~[?:?]
at org.apache.pulsar.functions.instance.JavaInstanceRunnable.setup(JavaInstanceRunnable.java:2) ~[?:?]
at org.apache.pulsar.functions.instance.JavaInstanceRunnable.run(JavaInstanceRunnable.java:2) ~[?:?]
at java.lang.Thread.run(Thread.java:834) [?:?]
```

PROBLEM: MY CONNECTOR IS NOT RUNNING

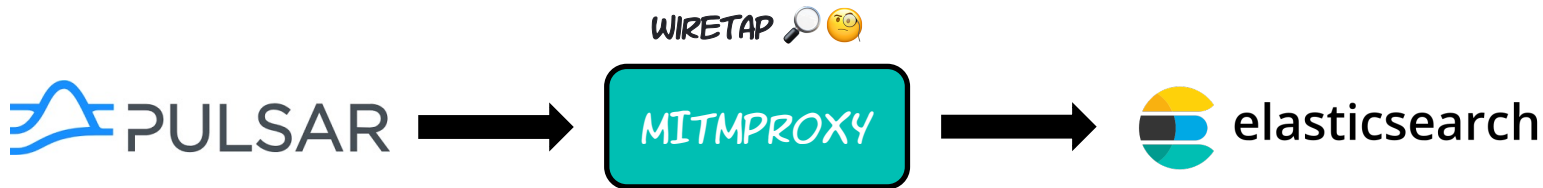
5. DEBUG WITH LOCALRUN

```
pulsar-2.7.2 → bin/pulsar-admin sinks localrun \  
--archive pulsar-io-elastic-search-2.7.2.nar \  
--tenant public --namespace default \  
--name myLocalTest --sink-config-file es.yml \  
--inputs myInputTopic --parallelism 1
```



PLAY WITH THE NUMBER
OF CONNECTOR THREADS

PROBLEM: SINK IS NOT RECEIVING ANY DATA



-DHTTP.PROXY = MITMPROXYHOST

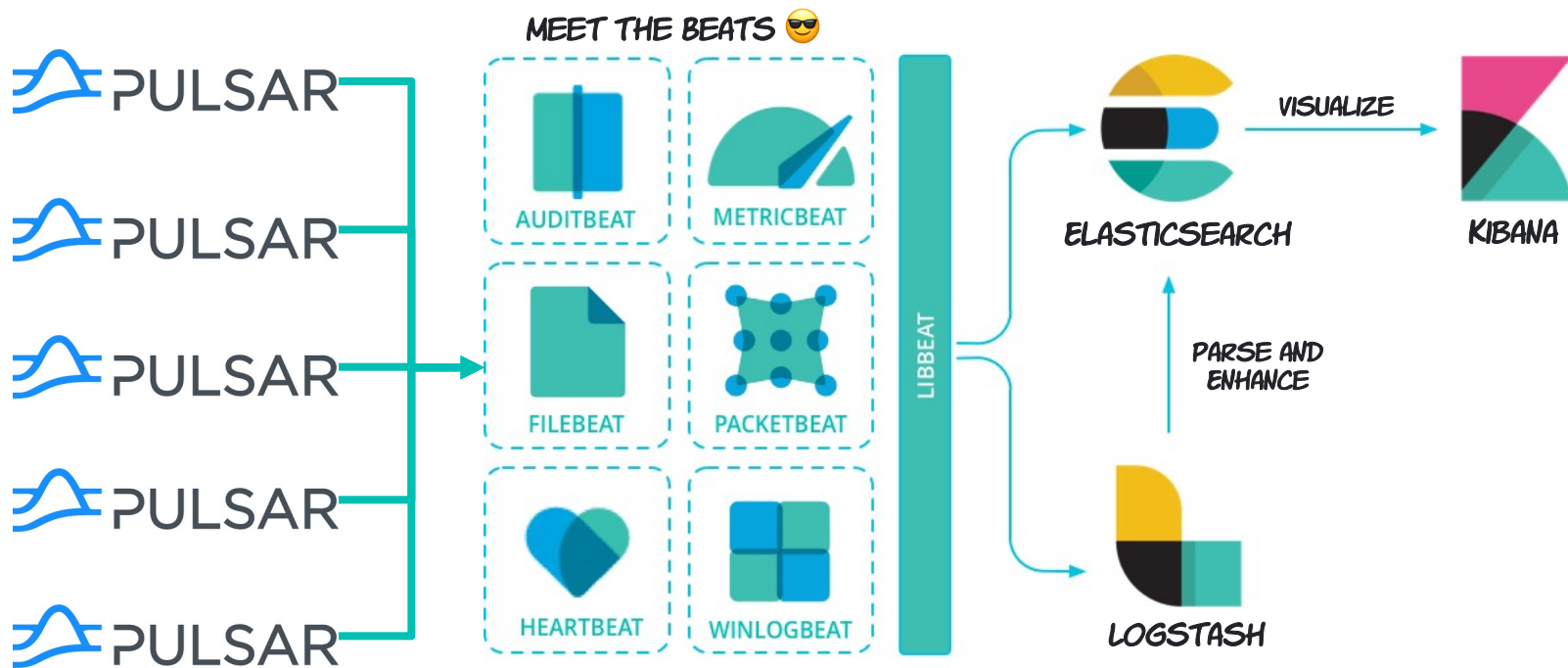
-DHTTP.PORT = 8080



```
Flows
GET https://www.google.com/
  → 200 text/html 64.52k 487ms
GET https://www.google.com/logos/doodles/2018/doodle-snow-games-day-12-6870619765473280-s.png
  → 200 image/png 2.63k 184ms
GET https://www.google.com/logos/2018/snowgames_skijump/cta.png
  → 200 image/png 13.4k 229ms
>> GET https://www.gstatic.com/external_hosted/createsjs/createsjs-2015.11.26.min.js
  → 200 text/javascript 48.51k 475ms
GET https://ssl.gstatic.com/gb/images/l2_2ec824b0.png
  → 200 image/png 231.64k 235ms
GET https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_variation_0.pb
  → 200 application/octet-stream 67.92k 356ms
GET https://ssl.gstatic.com/safebrowsing/csd/client_model_v5_ext_variation_0.pb
  → 200 application/octet-stream 67.92k 412ms
GET https://www.google.com/logos/2018/snowgames_skijump/snowgames_skijump18.js
  → 200 text/javascript 258.16k 900ms
POST https://www.google.com/gen_204?s=webaft&atyp=csi&ei=vCGLW6uMsKk8gTys6yIAw&rt=wsrt.2615,aft.1379,prt.1379
  → 204 text/html [no content] 379ms
GET https://www.gstatic.com/og/_/js/k=og.og2.en_US.ulHn0gNl161.0/rt=j/m=def/exm=in,fof/d=1/ed=1/rs=AA2YrTuv0KajN...
  → 200 text/javascript 46.4k 265ms
GET https://www.google.com/xjs/_/js/k=xjs.s.en.zjivxe8FVgY.0/m=sx,sb,cdos,cr,elog,hsm,jso,r,d,csi/am=wCL0oME3yP8-
  → 200 text/javascript 144.26k 368ms
GET https://www.google.com/xjs/_/js/k=xjs.s.en.zjivxe8FVgY.0/m=aa,abd,asyncl,dvl,foot,fpe,ipv6,lu,m,mu,sf,sonic,s-
  → 200 text/javascript 30.54k 195ms
GET https://www.google.com/logos/2018/snowgames_skijump/main-sprite.png
  → 200 image/png 1.1k 100ms
[14/36]
: replay_client [Flow]
```

[HTTPS://MITMPROXY.ORG](https://mitmproxy.org)

PROBLEM: MULTIPLE CLUSTERS AND LOGS



WHAT ABOUT...

WHEN YOU HAVE NO
IDEA ABOUT WHAT
IS GOING ON?



DEBUG THE CONNECTOR'S CODE

1. ENABLE THE JDWP PROTOCOL ON PULSAR

```
PULSAR_EXTRA_OPTS=${PULSAR_EXTRA_OPTS:-} -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:7777
```

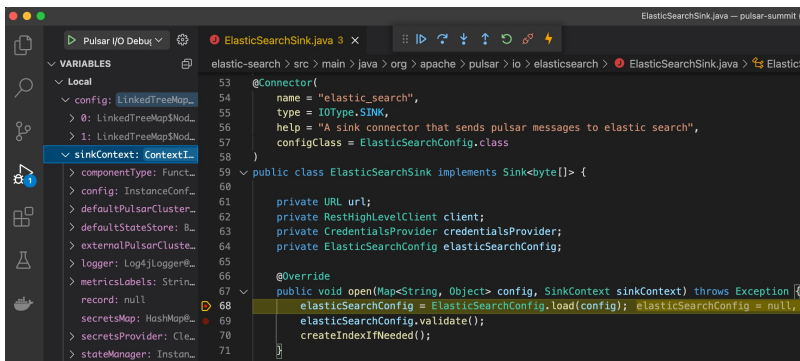
2. CONFIGURE THE FUNCTION RUNTIME TO THREAD

```
functionRuntimeFactoryClassName: org.apache.pulsar.functions.runtime.thread.ThreadRuntimeFactory
functionRuntimeFactoryConfigs:
  threadGroupName: "Thread Function Container Group"
```

3. ATTACH YOUR IDE TO THE JVM

```
"configurations": [
  {
    "type": "java",
    "request": "attach",
    "projectName": "pulsar-io-elastic-search",
    "name": "Pulsar I/O Debug",
    "hostName": "localhost",
    "port": 7777
  }
]
```

4. SET BREAKPOINTS AND DEBUG







Ricardo Ferreira

Principal Developer Advocate @ Elastic

Down to the Rabbit Hole with Pulsar I/O

THANK YOU 