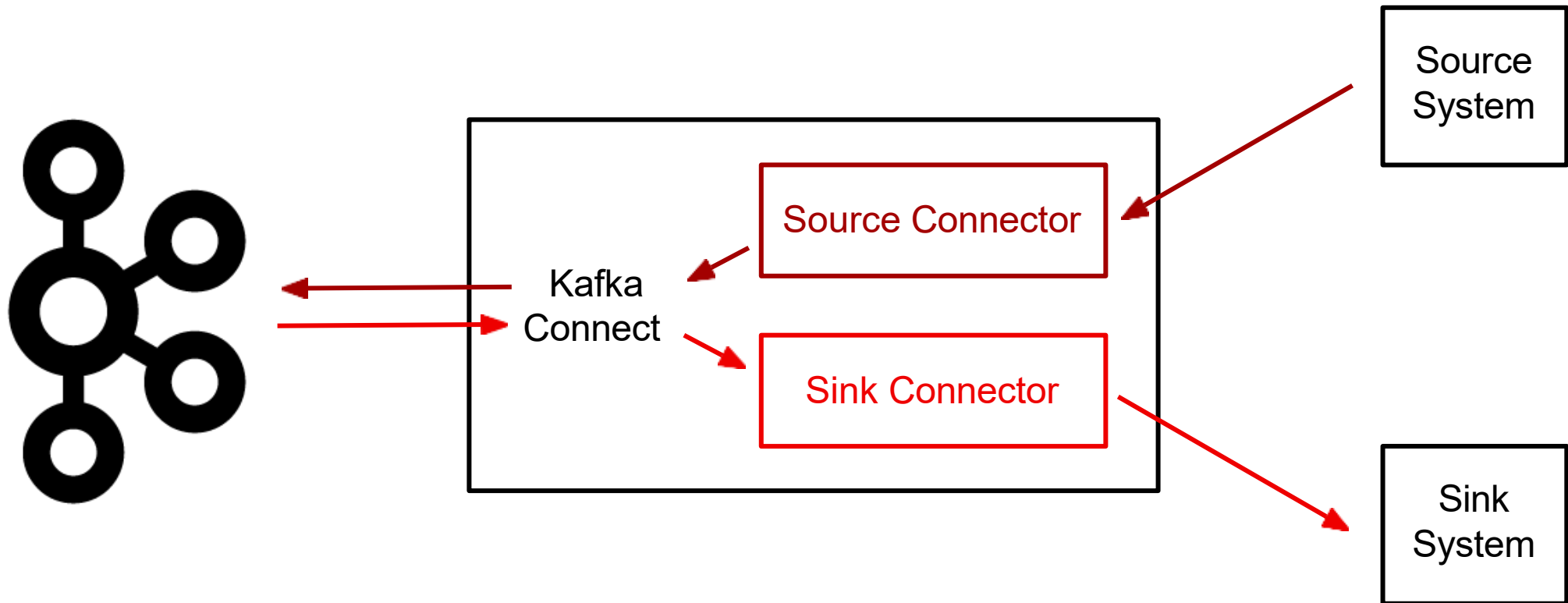


# Getting Started with Kafka Connect

Seshagiri Sriram



# What is Kafka Connect?

# An API for implementing connectors

Package org.apache.kafka.connect.source

## Class SourceConnector

java.lang.Object<sup>Ⓔ</sup>  
org.apache.kafka.connect.connector.Connector  
org.apache.kafka.connect.source.SourceConnector

All Implemented Interfaces:

Versioned

```
public abstract class SourceConnector  
extends Connector
```

SourceConnectors implement the connector interface to pull data from another system a

### Field Summary

Fields inherited from class org.apache.kafka.connect.connector.Conn

context

Package org.apache.kafka.connect.sink

## Class SinkConnector

java.lang.Object<sup>Ⓔ</sup>  
org.apache.kafka.connect.connector.Connector  
org.apache.kafka.connect.sink.SinkConnector

All Implemented Interfaces:

Versioned

```
public abstract class SinkConnector  
extends Connector
```

SinkConnectors implement the Connector interface to send Kafka data to another system.

### Field Summary

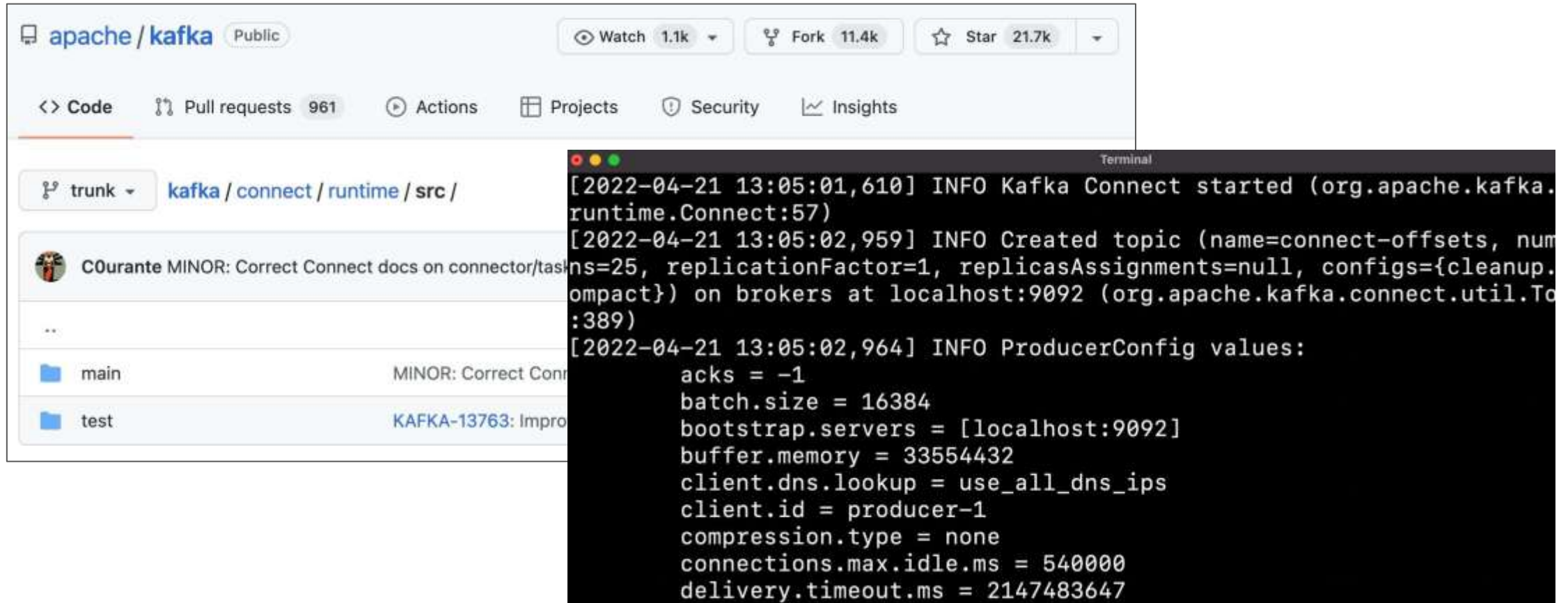
#### Fields

Modifier and Type	Field	Description
static final String <sup>Ⓔ</sup>	TOPICS_CONFIG	Configuration key for the list of input topics for this connector.

Fields inherited from class org.apache.kafka.connect.connector.Connector

context

## A runtime for executing connector processes



The image shows a screenshot of the Apache Kafka GitHub repository and a terminal window. The repository page displays the 'kafka' repository with 1.1k watches, 11.4k forks, and 21.7k stars. The 'Code' tab is selected, showing the file structure with 'main' and 'test' folders. The terminal window shows the following logs:

```
[2022-04-21 13:05:01,610] INFO Kafka Connect started (org.apache.kafka.runtime.Connect:57)
[2022-04-21 13:05:02,959] INFO Created topic (name=connect-offsets, num
ns=25, replicationFactor=1, replicasAssignments=null, configs={cleanup.
ompact}) on brokers at localhost:9092 (org.apache.kafka.connect.util.To
:389)
[2022-04-21 13:05:02,964] INFO ProducerConfig values:
acks = -1
batch.size = 16384
bootstrap.servers = [localhost:9092]
buffer.memory = 33554432
client.dns.lookup = use_all_dns_ips
client.id = producer-1
compression.type = none
connections.max.idle.ms = 540000
delivery.timeout.ms = 2147483647
```

# Connector

```
79 public class MongoDBConnector extends SourceConnector {  
  
102     @Override  
103     public void start(Map<String, String> props) {  
104         // Validate the configuration ...  
105         final Configuration config = Configuration.from(props);  
106         if (!config.validateAndRecord(MongoDbConnectorConfig.ALL_FIELDS,  
107             throw new ConnectException("Error configuring an instance of  
108         })  
109  
110         this.config = config;  
111  
112         // Set up the replication context ...  
113         taskContext = new MongoDbTaskContext(config);  
114         this.connectionContext = taskContext.getConnectionContext();  
}
```

# Connector

```
79 public class MongoDBConnector extends SourceConnector {  
102     @Override  
103     public void start(Map<String, String> props) {  
104         // Validate the configuration ...  
105         final Configuration config = Configuration.from(props);  
106         if (!config.validateAndRecord(MongoDbConnectorConfig.ALL_FIELDS,  
107             throw new ConnectException("Error configuring an instance of  
108         }  
109  
110         this.config = config;  
111  
112         // Set up the replication context ...  
113         taskContext = new MongoDbTaskContext(config);  
114         this.connectionContext = taskContext.getConnectionContext();  
}
```



**debezium-connector-mongodb-1.9.0.Final.jar**

NB: Sample Code is provided in the Source Code

# Connector

```
79 public class MongoDBConnector extends SourceConnector {  
  
102 @Override  
103 public void start(Map<String, String> props) {  
104     // Validate the configuration ...  
105     final Configuration config = Configuration.from(props);  
106     if (!config.validateAndRecord(MongoDbConnectorConfig.ALL_FIELDS,  
107         throw new ConnectException("Error configuring an instance of  
108     }  
109  
110     this.config = config;  
111  
112     // Set up the replication context ...  
113     taskContext = new MongoDbTaskContext(config);  
114     this.connectionContext = taskContext.getConnectionContext();  
}
```

**debezium-connector-mongodb-1.9.0.Final.jar**

Connect

**plugin.path**

**debezium-connector-mongodb-1.9.0.Final.jar**



# Connector

```
79 public class MongoDBConnector extends SourceConnector {  
  
102     @Override  
103     public void start(Map<String, String> props) {  
104         // Validate the configuration ...  
105         final Configuration config = Configuration.from(props);  
106         if (!config.validateAndRecord(MongoDbConnectorConfig.ALL_FIELDS,  
107             throw new ConnectException("Error configuring an instance of  
108         }  
109  
110         this.config = config;  
111  
112         // Set up the replication context ...  
113         taskContext = new MongoDbTaskContext(config);  
114         this.connectionContext = taskContext.getConnectionContext();  
}
```

**debezium-connector-mongodb-1.9.0.Final.jar**

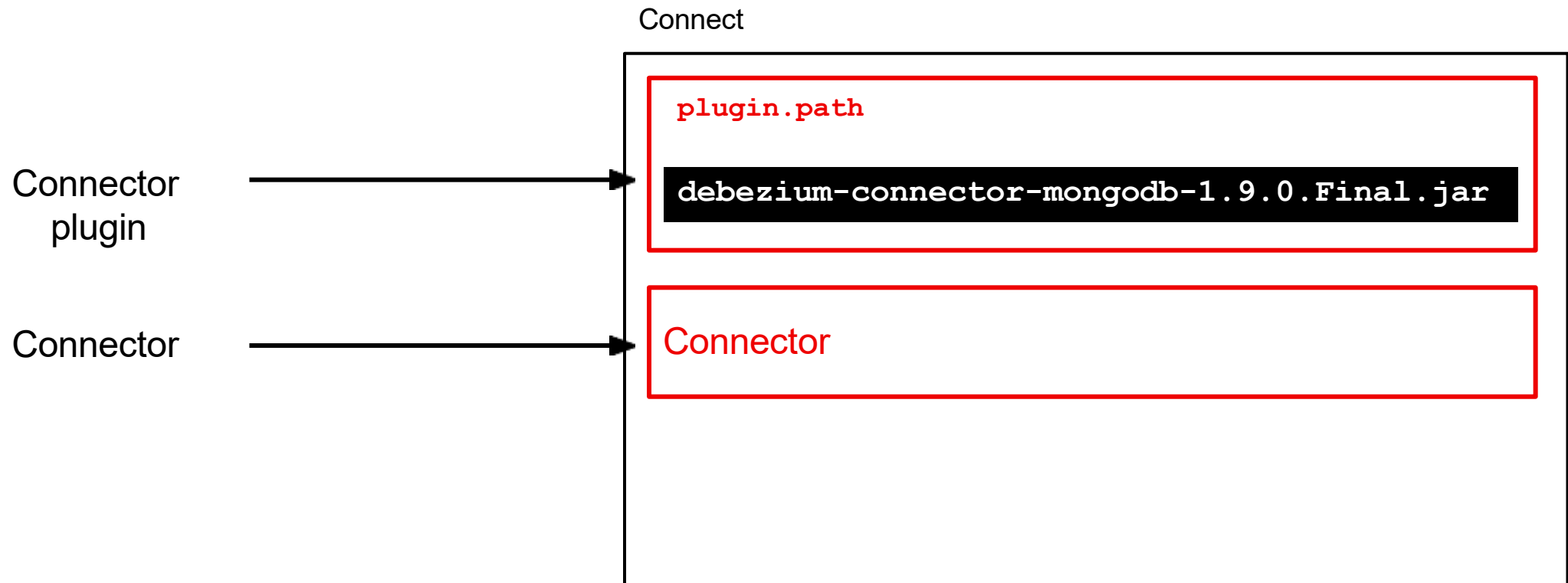
## Connect

plugin.path

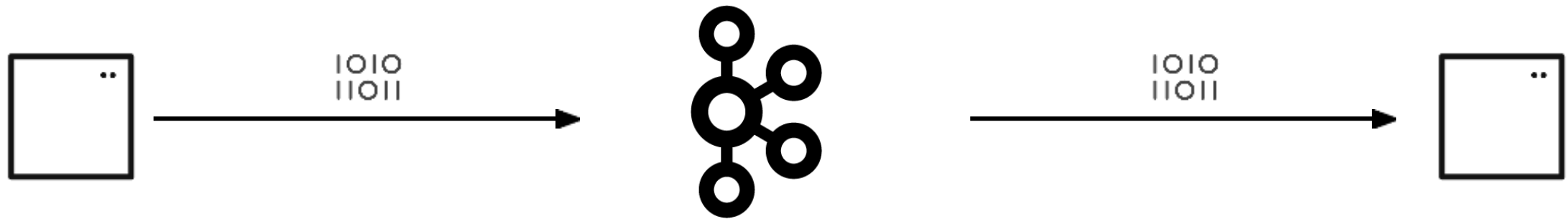
**debezium-connector-mongodb-1.9.0.Final.jar**

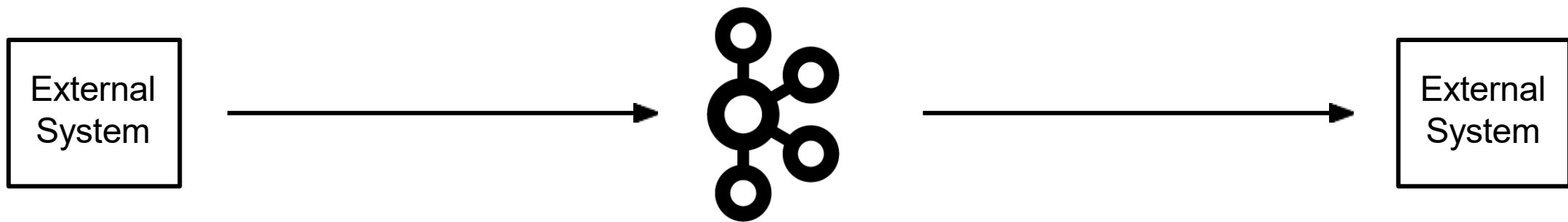
Connector

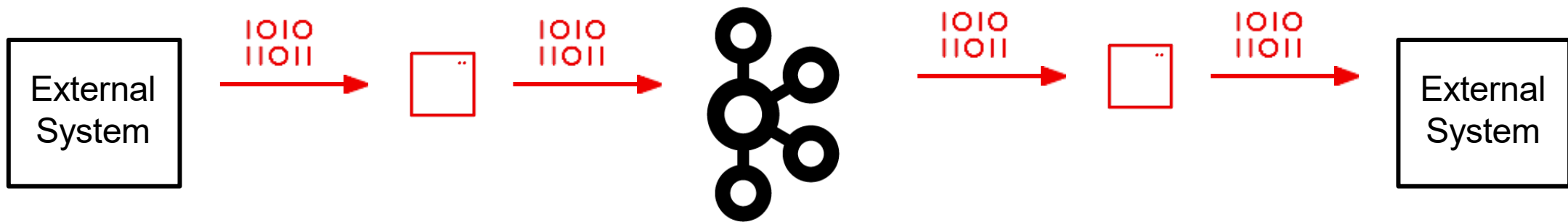
# Connector

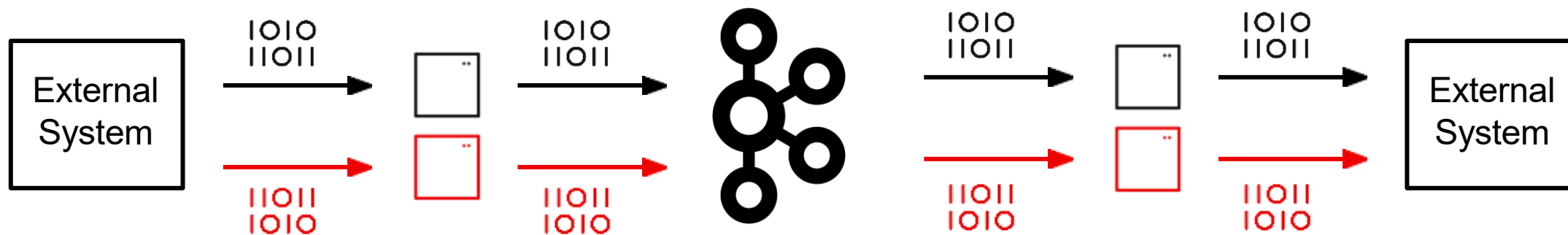


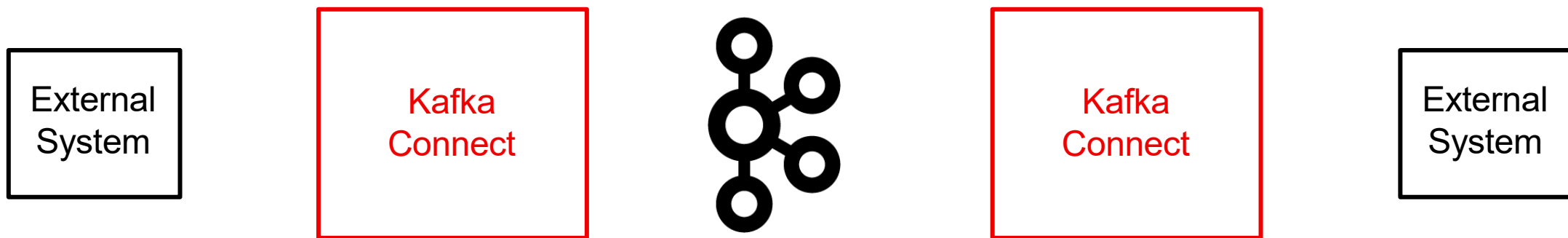
# Why use Kafka Connect?



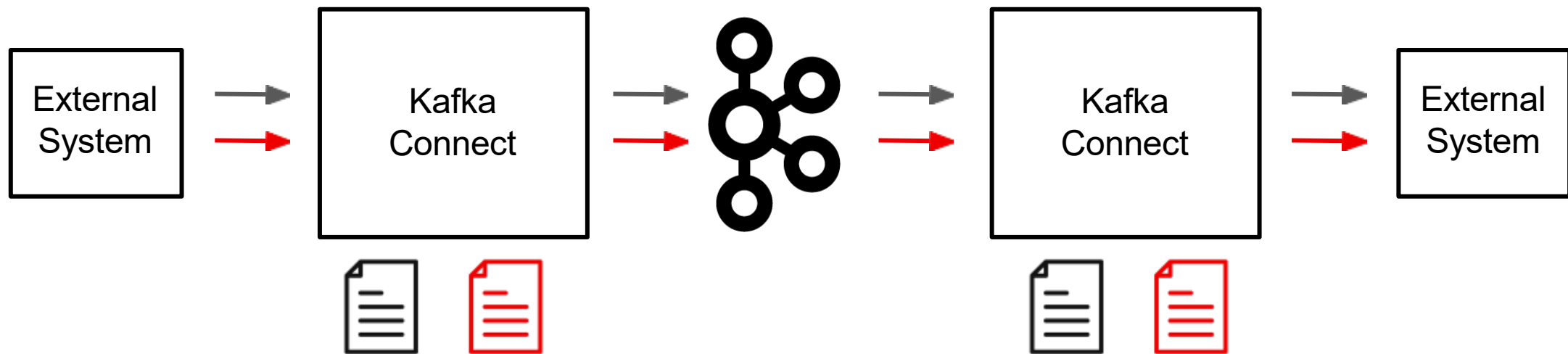


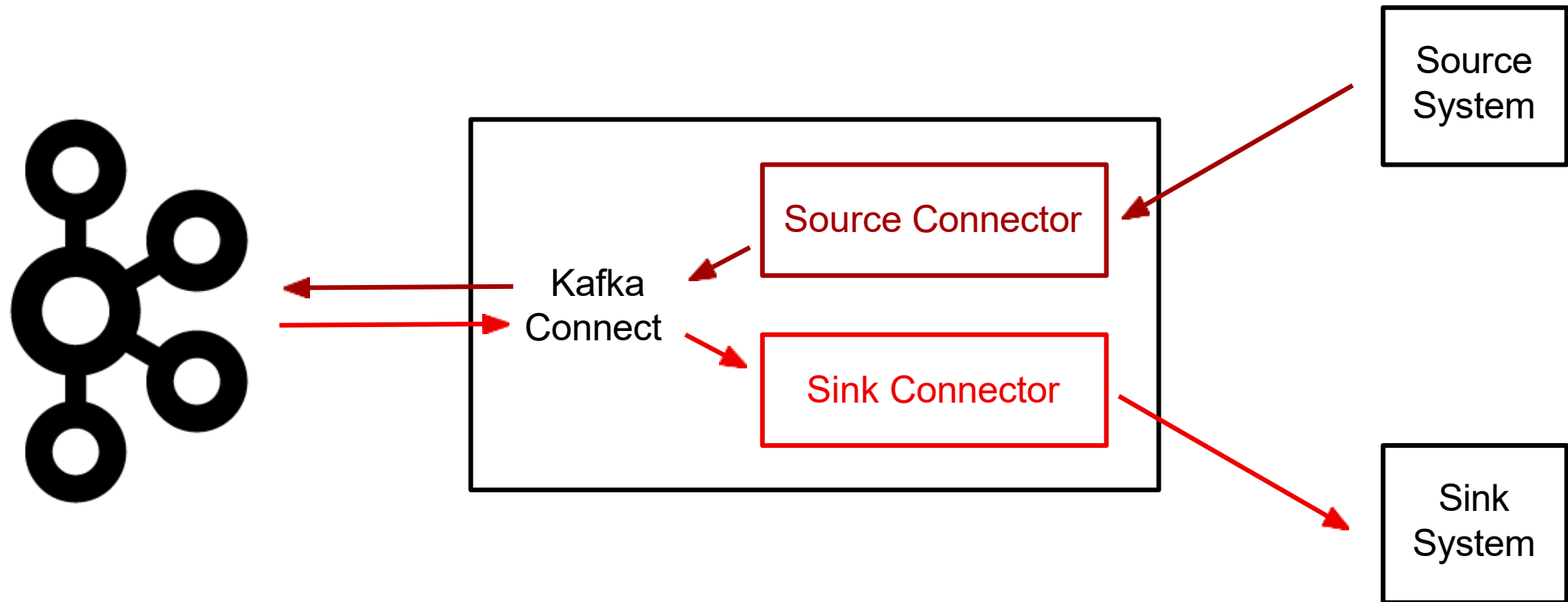


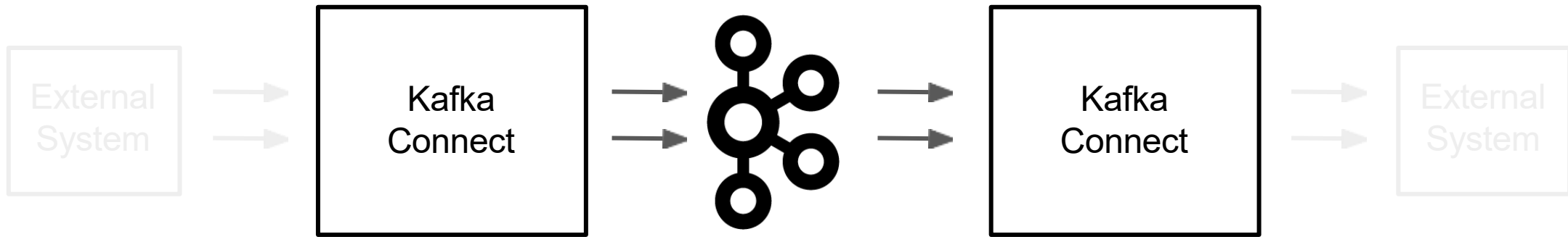




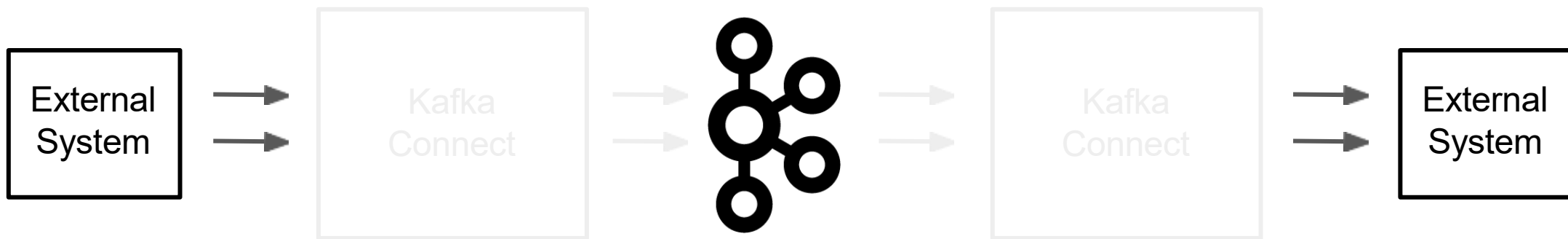








## Why use Kafka Connect?



## Kafka Connect use cases



Data pipelines



Interoperability



Extending existing  
systems

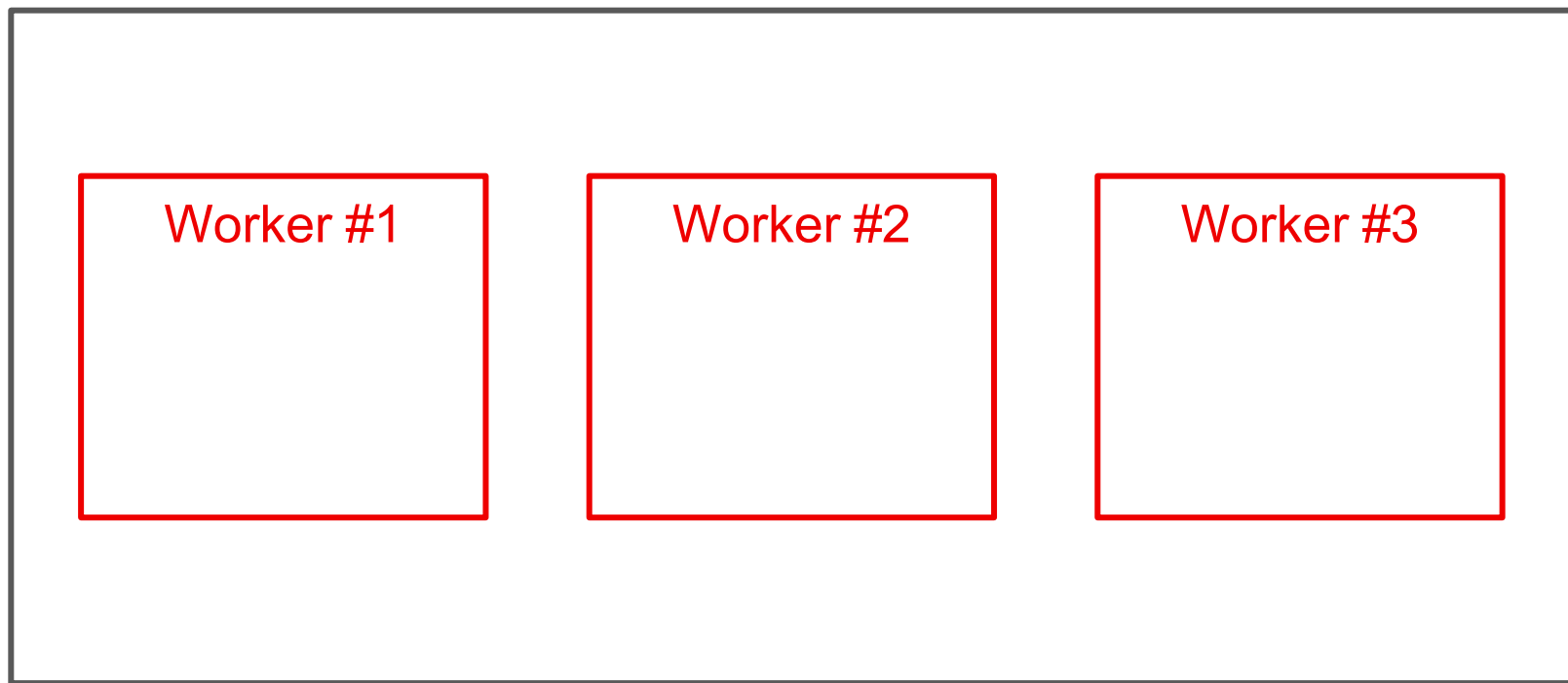
# Some best practice tips

---

# Run in Distributed Mode

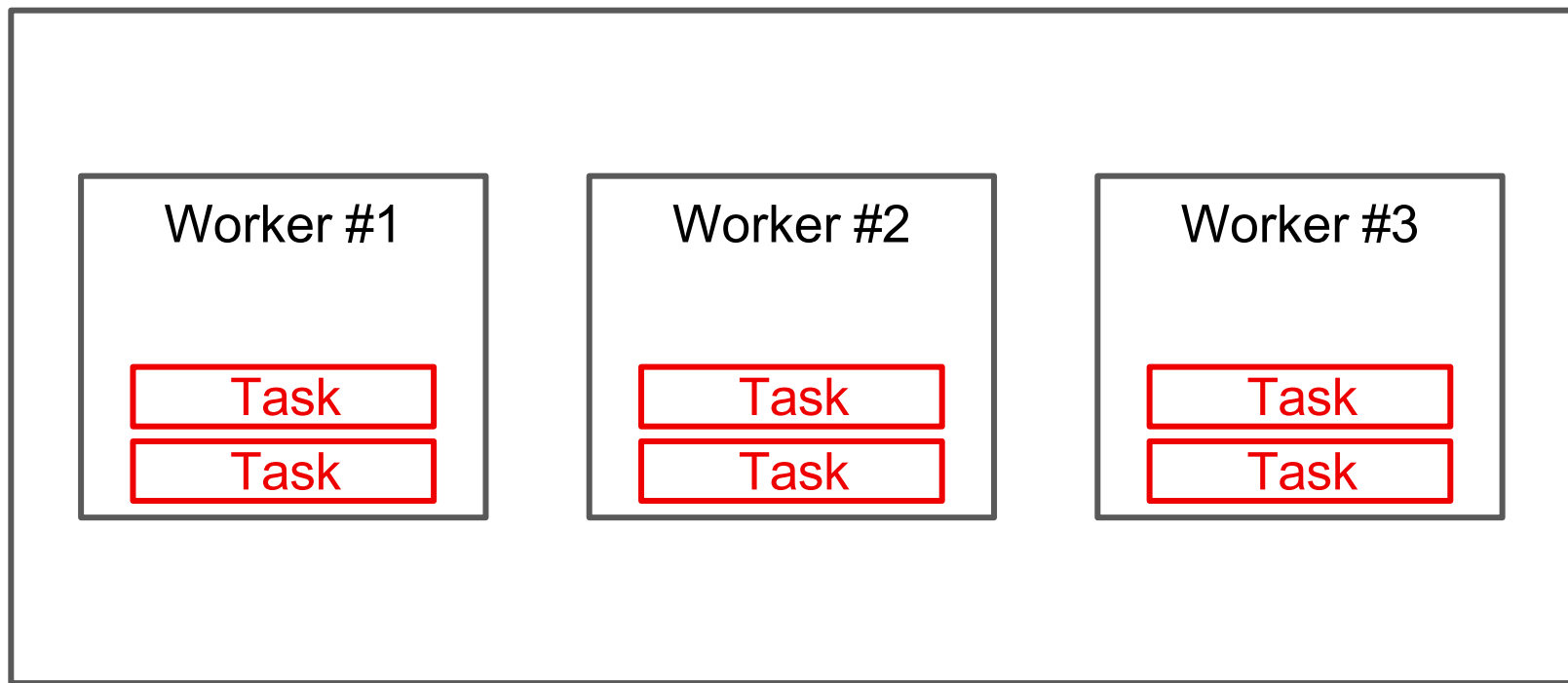


Run in Distributed mode

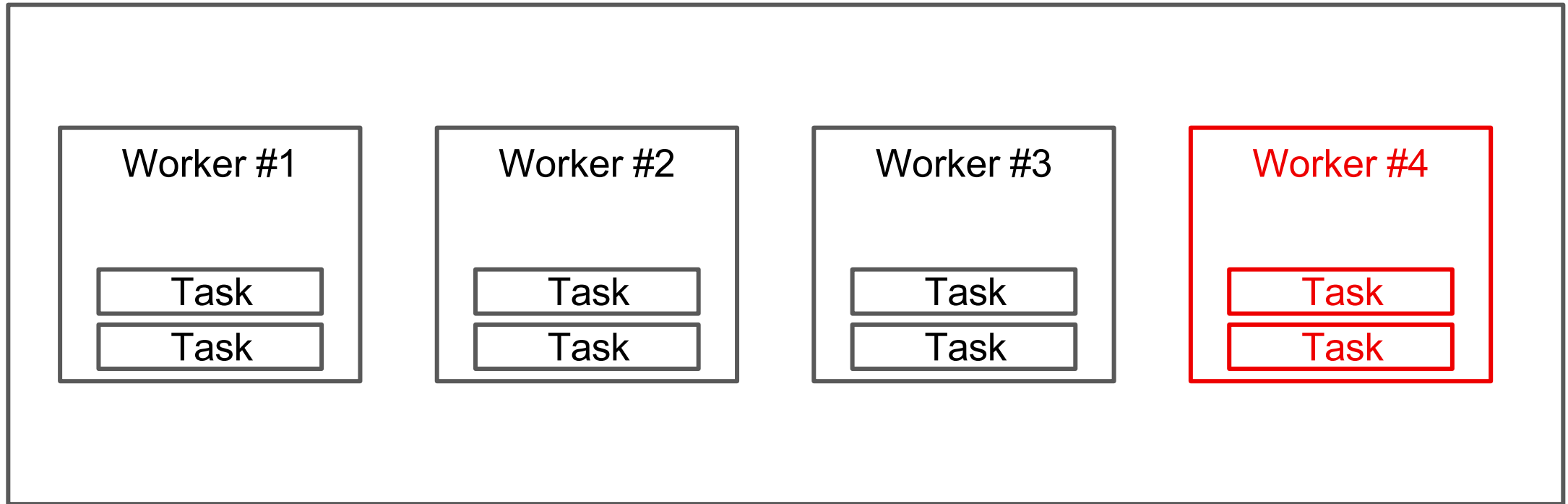


Connect cluster



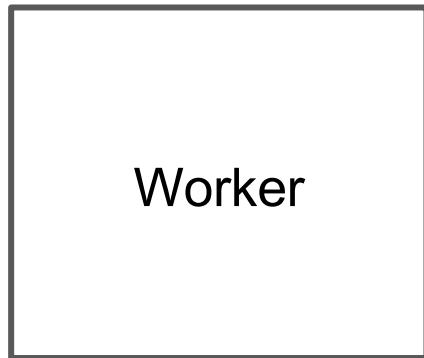


Connect cluster



Connect cluster

# Configuration



- `group.id`
- `offset.storage.topic`
- `config.storage.topic`
- `status.storage.topic`

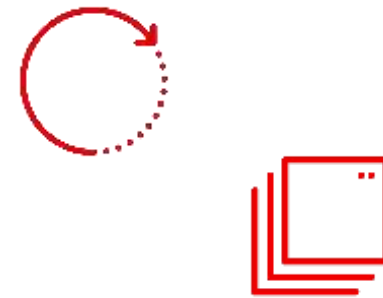
---

# Use automation for managing clusters



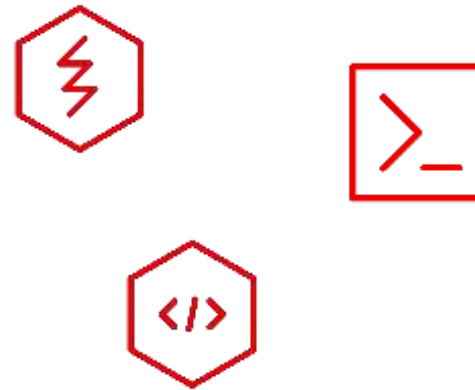
# Responsibilities

- Restarting failed connectors and tasks
- Scaling up and down
- Rolling out upgrades
- Adding new connector plugins



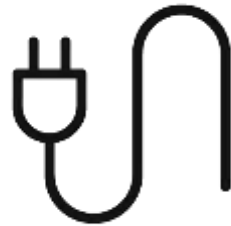
# Solutions

- Operators
- Pre-built images
- Tools



---

# Know the connector plugins



# Know the connector plugins

Converters



Transformations

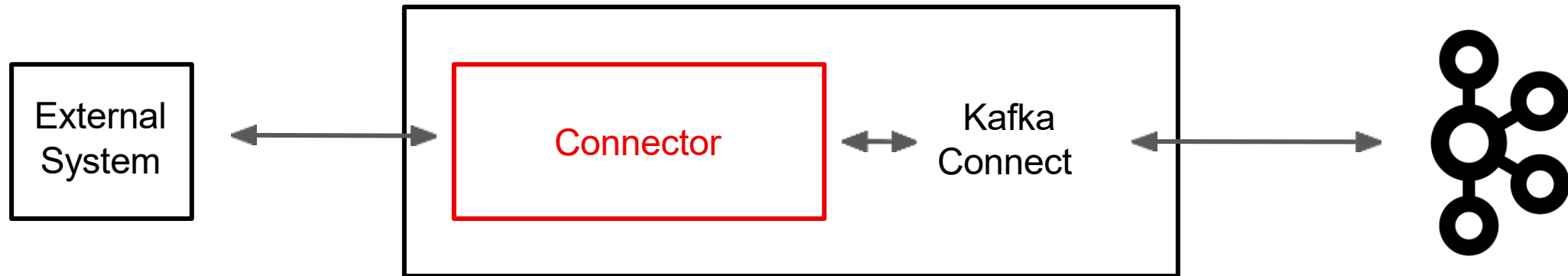


Predicates



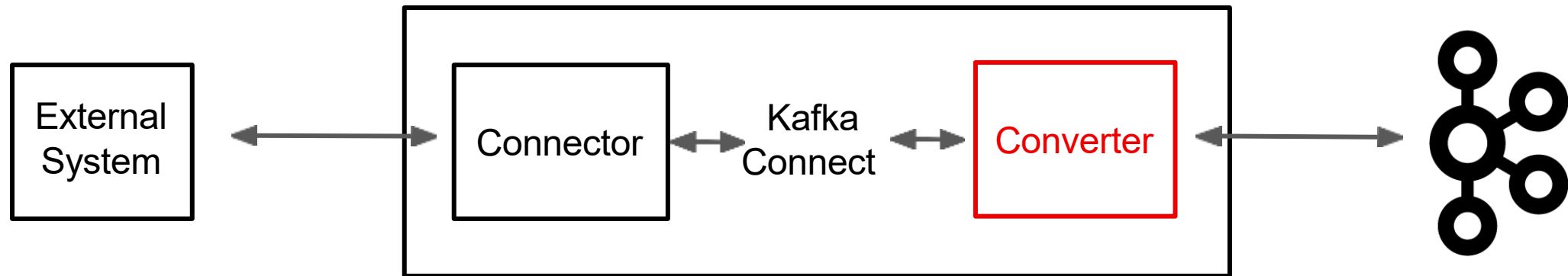


# Connector plugins



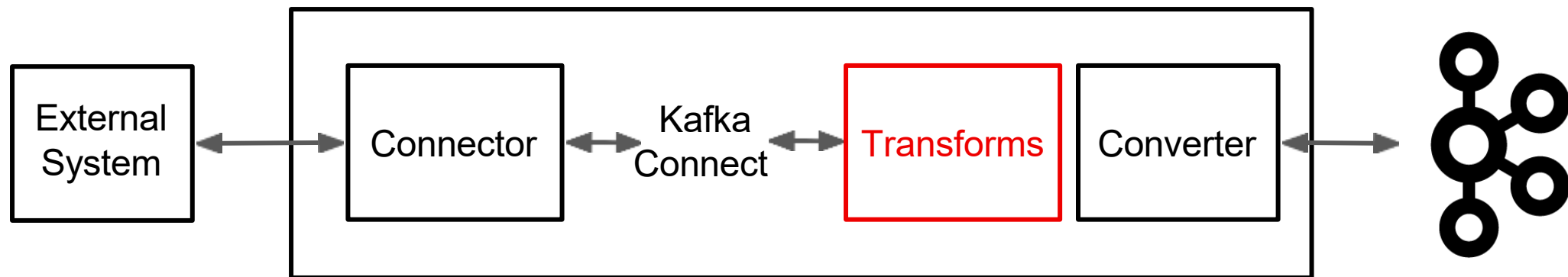
# Converters + Header Converters

```
key.converter=org.apache.kafka.connect.storage.StringConverter  
value.converter=org.apache.kafka.connect.storage.StringConverter  
header.converter=org.apache.kafka.connect.storage.SimpleHeaderConverter
```



# Transformations

```
{  
  "transforms": "filterTombstones",  
  "transforms.filterTombstones.type": "o.a.k.connect.transforms.Filter",  
}
```



# Predicates

```
{  
  "transforms": "filterTombstones",  
  "transforms.filterTombstones.type": "o.a.k.connect.transforms.Filter",  
  "transforms.filterTombstones.predicate": "isTombstone",  
  
  "predicates": "isTombstone",  
  "predicates.isTombstone.type": "o.a.k...predicates.RecordIsTombstone"  
}
```

# Predicates

```
{  
  "transforms": "filterTombstones",  
  "transforms.filterTombstones.type": "o.a.k.connect.transforms.Filter",  
  "transforms.filterTombstones.predicate": "isTombstone",  
  "predicates": "isTombstone",  
  "predicates.isTombstone.type": "o.a.k...predicates.RecordIsTombstone"  
}
```

# Predicates

```
{  
  "transforms": "filterTombstones",  
  "transforms.filterTombstones.type": "o.a.k.connect.transforms.Filter",  
  "transforms.filterTombstones.predicate": "isTombstone",  
  
  "predicates": "isTombstone",  
  "predicates.isTombstone.type": "o.a.k...predicates.RecordIsTombstone"  
}
```

---

# Consider using a Schema Registry



# My Events

```
{  
  "title": "Getting up to Speed with Kafka Connect"  
  "speaker": "Mickael and Kate",  
  "room": "Arora West"  
}
```



# Schema

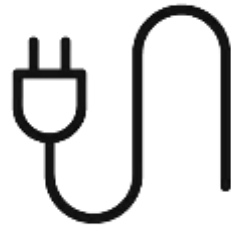
```
{
  "type": "object",
  "properties": {
    "title": {"type": "string"},
    "speaker": {"type": "string"},
    "room": {
      "type": "string",
      "enum": ["Arora West", "Arcadia"]
    }
  },
  "required": ["title", "room"]
}
```

# Using Avro Schemas with Apicurio

```
key.converter=io.apicurio.registry.utils.converter.AvroConverter  
key.converter.apicurio.registry.url=http://registry:8080/api  
...
```

---

# Know the worker plugins



# Configuration Providers

```
{  
  "file": "myfile.txt",  
  "topic": "source-topic",  
  "ssl.truststore.password": "secret"  
}
```

# Configuration Providers

```
{  
  "file": "myfile.txt",  
  "topic": "source-topic",  
}
```

# Configuration Providers

```
config.providers=file  
config.providers.file.class=o.a.k.c.config.provider.FileConfigProvider
```

```
{  
  "file": "myfile.txt",  
  "topic": "source-topic",  
}
```

# Configuration Providers

```
{  
  "file": "myfile.txt",  
  "topic": "source-topic",  
  "truststore.password": "secret"  
}
```

```
{  
  "file": "myfile.txt",  
  "topic": "source-topic",  
  "truststore.password": "${file:/tmp/ssl.properties:password}"  
}
```

# REST Extensions

```
rest.extension.classes=o.a.k.c...extension.BasicAuthSecurityRestExtension
```

- Authentication
- Authorization
- Validation
- JAX-RS plugins



---

# Rely on Connect Metrics



# Rely on Connect Metrics



- Tasks/Connectors status

`kafka.connect:type=connect-worker-metrics`

- Connection to Kafka Brokers

`kafka.connect:type=connect-node-metrics,client-id=<>,node-id=<>`

- Connector metrics

---

# Master the REST API



# Manage logger level

```
GET /admin/loggers
```

```
PUT /admin/loggers/org.apache.kafka.connect -d '{"level": "DEBUG"}
```

# Restart connectors and tasks

Connector

```
POST /connectors/file-sink/restart
```

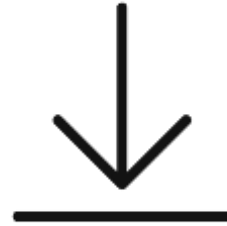
Task

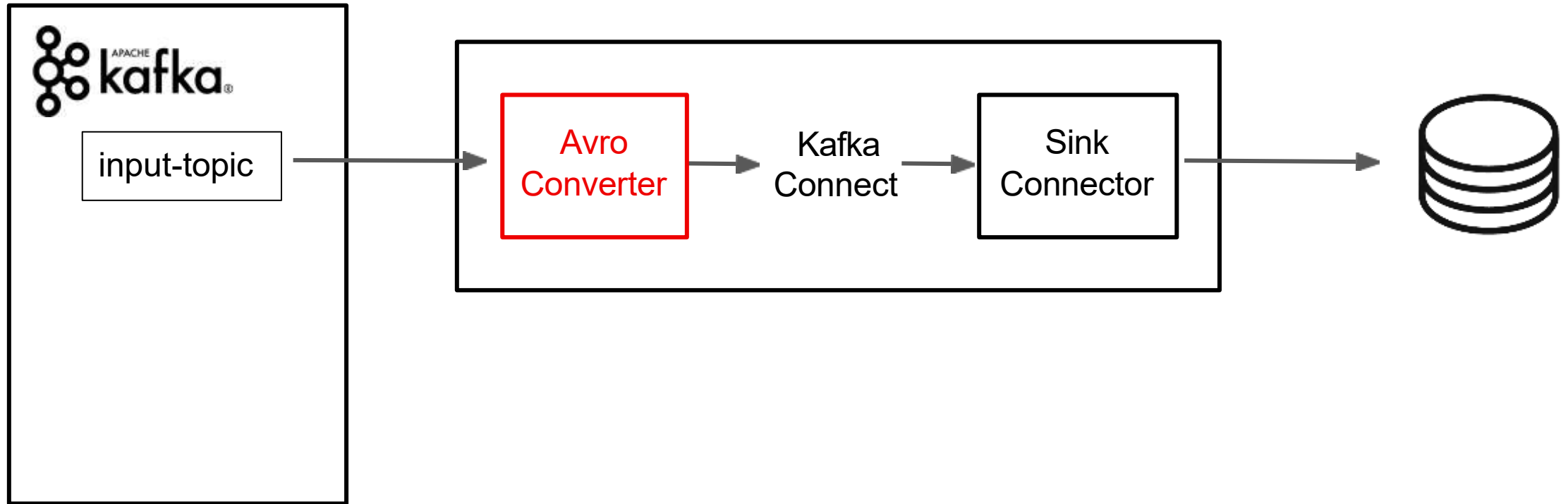
```
POST /connectors/file-sink/tasks/0/restart
```

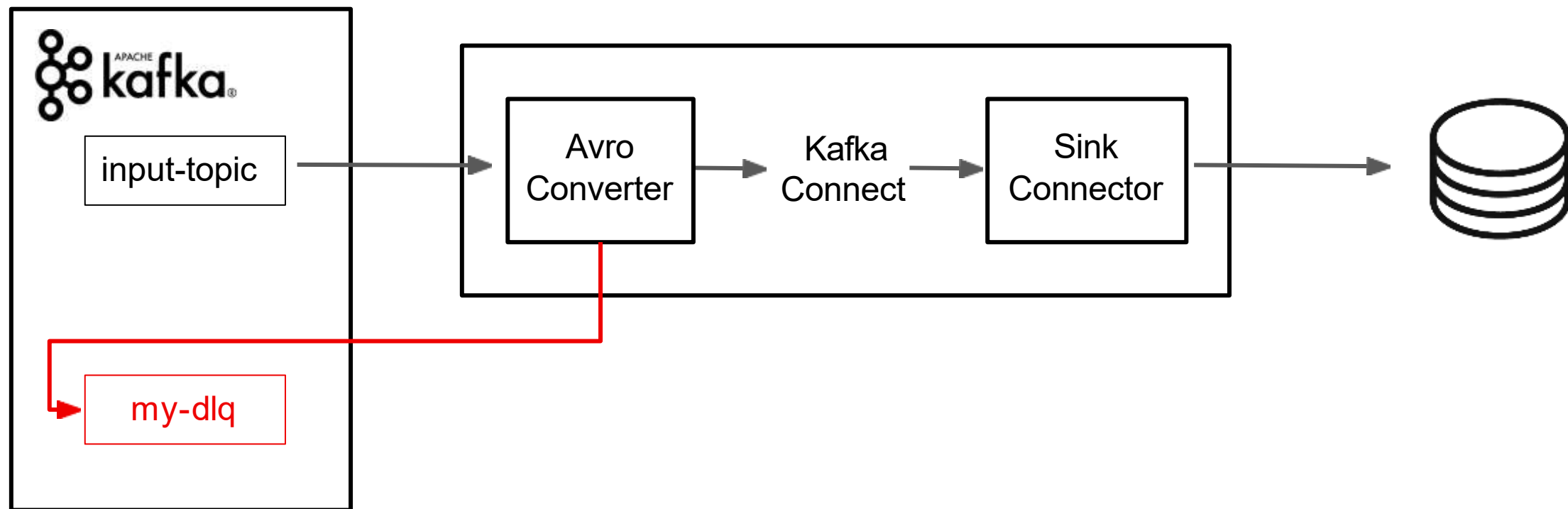
```
POST /connectors/file-sink/restart?includeTasks=true&onlyFailed=true
```

---

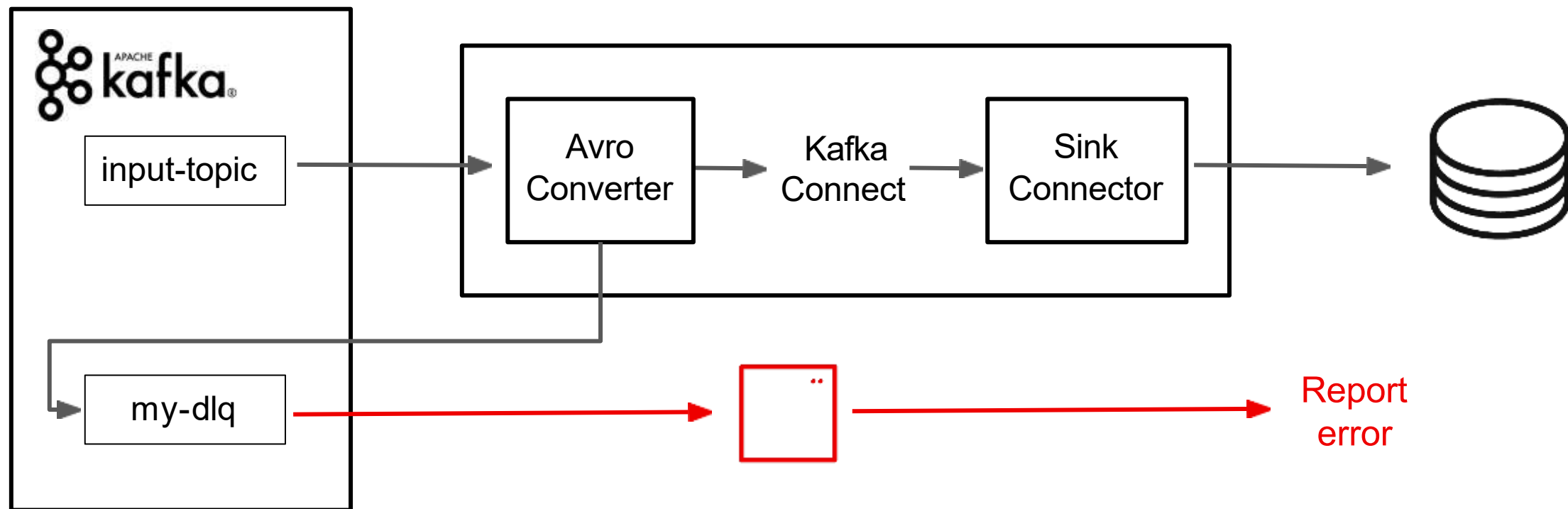
# Use Dead Letter Queues












---


# Try kcctl




main kcctl / README.md Go to file ...

 **jvenant** #167 allow multiple connector names and regexp ✓

Latest commit 2e8980a on 4 Mar History

12 contributors 


204 lines (142 sloc) 6.82 KB <> File Raw Blame Icons Edit Delete



# kcctl – Your Cuddly CLI for Apache Kafka Connect

*It's Casey. Casey Cuddle.*

This project is a command-line client for [Kafka Connect](#). Relying on the idioms and semantics of *kubectl*, it allows you to register and examine connectors, delete them, restart them, etc. You can see what *kcctl* is about in this short video:



```
gertling@development:~$ kcctl get plugins
NAME      CLASS
kcctl     dev.werling.tech.kafka.source.DistributedSourceConnector 1.0
kcctl     io.deltalake.connector.deltalake.DeltaSinkConnector      1.2.0-SNAPSHOT
kcctl     org.apache.kafka.connect.file.FileStreamSinkConnector   1.2.0
kcctl     org.apache.kafka.connect.file.FileStreamSourceConnector 1
kcctl     org.apache.kafka.connect.rest.RestSinkConnector         1
kcctl     org.apache.kafka.connect.rest.RestSourceConnector       1
kcctl     org.apache.kafka.connect.rest.RestSinkConnector         1.2.0

gertling@development:~$ kcctl get connectors
NAME      TYPE      STATE      TARGET
first-test source     RUNNING    0 / 0
second-test sink       STOPPED    0 / 0

gertling@development:~$ kcctl register --name test
```



---

# Don't forget to Upgrade



# What coming next in Connect?

## KIP-769:

APIs to list all connector  
plugins and retrieve  
their configuration  
definitions

Kafka 3.2

## Existing endpoints

```
GET /connector-plugins
```

```
PUT /connector-plugins/{connector plugin}/config/validate
```

## New endpoint

```
GET /connector-plugins
```

```
PUT /connector-plugins/{connector plugin}/config/validate
```

```
GET /connector-plugins/{connector plugin}/config
```



```
$ curl http://localhost:8083/connector-plugins/FileStreamSinkConnector/config
[
  {
    "name": "file",
    "type": "STRING",
    "required": false,
    "default_value": null,
    "importance": "HIGH",
    "documentation": "Destination filename. If not specified, the standard output
will be used",
    "group": null,
    "width": "NONE",
    "display_name": "file",
    "dependents": [],
    "order": -1
  }
]
```

KIP-618:

# Exactly-Once Support for Source Connectors

Voted

- Source connectors use a **transactional producers** for writing records and offsets
- Connectors can **declare** whether they support exactly once
- Workers have exactly once **disabled by default**

## KIP-802:

# Validation Support for Kafka Connect SMT and Converter Options

Discussion in  
progress

## New capability with existing endpoint


```
PUT /connector-plugins/{connector plugin}/config/validate
```

- Transformations
- Predicates
- Converters

# KIP-821: Connect Transforms support for nested structures

Discussion in  
progress


```
{  
  "room": "Arcadia",  
  "speaker": {  
    "name": {  
      "speaker-name": "Kate"  
    }  
  }  
}
```



```
{  
  "room": "Arcadia",  
  "speaker": {  
    "name": {  
      "speaker-name": "Kate"  
    }  
  }  
}
```



```
{  
  "room": "Arcadia",  
  "speaker": {  
    "name": {  
      "speaker-name": "Kate"  
    }  
  }  
}
```



```
{  
  "transforms": "smt1",  
  "transforms.smt1.type": "org.apache.kafka.connect.transforms.ValueToKey",  
  "transforms.smt1.field.style": "nested",  
  "transforms.smt1.fields": "speaker.name"  
}
```

```
{  
  "room": "Arcadia",  
  "speaker-name": "Kate"  
}
```