**We all connect to kafka03 node: ssh student@35.246.196.7**

**We will enable Kafka Connect on this node.**

Kafka Connect workers package ships with Apache Kafka, so there is no need to install it separately. Please note that for production it’s recommended to run Connect on separate servers then the brokers.

curl localhost:8083 // check status of connect - if nothing/could not connect returned then we need to start Kafka Connect, if shows something like this ({"version":"2.6.2","commit":"da65af02e5856e34","kafka\_cluster\_id":"U13Ik5t0Qle4niSnbDCofA"}

) then you need to just start the connectors.

**Starting Connect:** we start connect on a worker by calling a script and a properties file - if started you don’t to run below command:

/opt/kafka/bin/connect-distributed.sh /opt/kafka/config/connect-distributed.properties &

Note: the most important setting in connect-distributed.properties is group-id. All workers in the same Connect cluster need to have the same groupid.

group.id is one of the most important configurations in this file. Worker groups are created according to group id. So if we start multiple worker with same group id, they will be in the same worker cluster.

**Let’s check our Connect cluster is running, by checking the REST API:**

curl localhost:8083

**Let’s see what connector plugins are available on this worker:**

curl localhost:8083/connector-plugins |jq

At least the below 2 plug-ins will appear.

[

{

"class": "org.apache.kafka.connect.file.FileStreamSinkConnector",

"type": "sink",

"version": "2.3.0"

},

{

"class": "org.apache.kafka.connect.file.FileStreamSourceConnector",

"type": "source",

"version": "2.3.0"

}

]

Looks like we have the File Source/Sink connector (Apache Kafka has these 2 connectors included by default). Let’s see now the active connectors on this worker:

curl localhost:8083/connectors

* **so let’s activate a new file stream one:**

**To start a connector we need to send a POST call to http://localhost:8083/connectors endpoint with the configuration of the Connector that we want to run.**

**We need a configuration file in JSON format in file: file-src.json (you can find this in /tmp and make a copy out of it - modify the name of the connector, the input file name and the topic name)**

{

"name":"file-source-vali",

"config":{"connector.class":"FileStreamSource","file":"/tmp/test-input.csv","topic":"vali-file"}

}

And then we load the connector:

curl -d @"/tmp/file-src.json" -H "Content-Type: application/json" -X POST <http://localhost:8083/connectors>

**Note: please pay attention and personalize the connector name and the config file name**

**Let’s see the connectors running:**

curl localhost:8083/connectors // file-source-vali will appear

curl localhost:8083/connectors/file-source-vali |jq // to see the configurations of the connector

Now, let’s create a tmp/test-input.csv and place some data inside

Cat > /tmp/test-input.csv

**And check using the consumer console the content of vali-file topic:**

/opt/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic test-file --from-beginning

Now, let’s add more data:

Cat >> /tmp/test-input.csv

And then run the consumer console again:

/opt/kafka/bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic vali-file --from-beginning

**Note: you can see that the topic contains quite many fields, something like:**

{"schema":{"type":"string","optional":false},"payload":"1"}

{"schema":{"type":"string","optional":false},"payload":"2"}

{"schema":{"type":"string","optional":false},"payload":"3"}

{"schema":{"type":"string","optional":false},"payload":"4"}

{"schema":{"type":"string","optional":false},"payload":"5"}

Since we haven’t specified any converter for the data - by default JSON is chosen as format and schema is also output.

**Now using the File Sink connector let’s save a topic content into a file:**

We need a config file: file-sink.json

{

"name":"file-sink-vali",

"config":{"connector.class":"FileStreamSink","file":"/tmp/test-output.csv","topics":"vali-file"}

}

**Load the connector:**

curl -d @"/tmp/file-sink.json" -H "Content-Type: application/json" -X POST <http://localhost:8083/connectors>

And now look for test2-output.csv

If we type in more data in test2.csv we should have more data in test2-output.csv

Cat >> tmp/test2.csv

**Note: to Delete a connector run:**

curl -X DELETE localhost:8083/connectors/file-source-vali