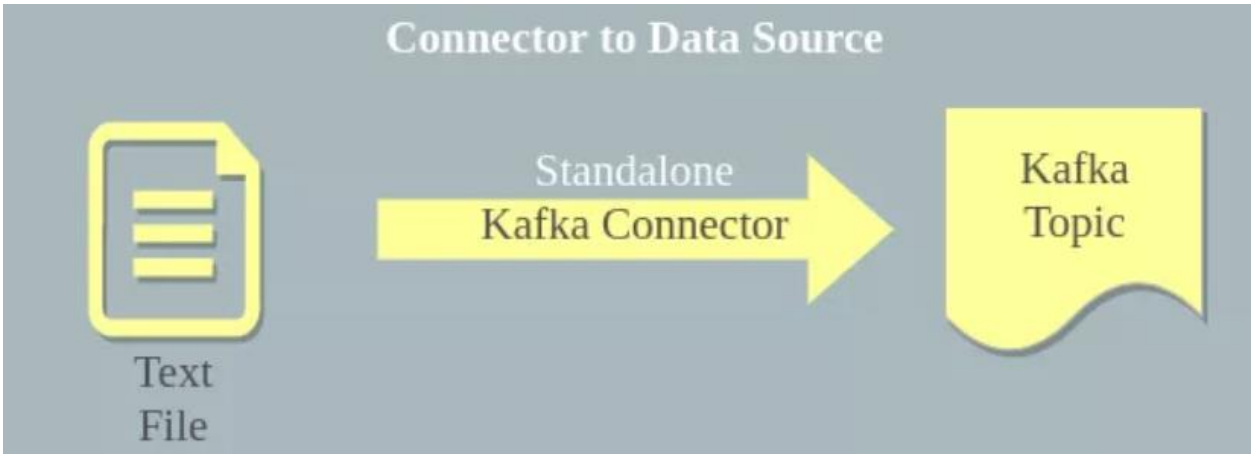
Exercise 1: Apache Kafka Connector Example – Import Data from text file into Kafka

1. Use Case: Setup a standalone connector to listen on a text file and import data from the text file. once the connector is setup, data in text file is imported to a Kafka Topic as messages. And any further data appended to the text file creates an event. These events are being listened by the Connector. The change in data is written to the Kafka Topic.



1. Navigate to the location where confluent is installed.
   cd /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0

   $ bin/kafka-topics\
   --create \
   --zookeeper localhost:2181 \
   --replication-factor 1 \
   --partitions 1 \
   --topic my-connect-test

```
[cloudera@quickstart confluent-4.0.0]$ bin/kafka-topics --create --zookeeper loc
alhost:2181   --replication-factor 1   --partitions 1   --topic my-connect-test
Created topic "my-connect-test".
[cloudera@quickstart confluent-4.0.0]$
```

2. Next is to create a text file, test.txt next to bin folder. Create a folder config and create connect-file-source.properties inside the folder.

   Paste the content in the properties file as

```
#my-file-source.properties config file
```

```
name=local-file-source
connector.class=FileStreamSource
tasks.max=1
file=test.txt
topic=my-connect-test
```

The test.txt file has to be created with some content like

```
Hi, Today is the last day of the training.
We are learning kafka.
```

```
[cloudera@quickstart confluent-4.0.0]$ ls
bin  etc  logs  README  share  src  test.db  test.txt  tmp  untitled folder
```

3. Now, Processes that execute Kafka Connect connectors and tasks are called workers. Since we are reading data from a single machine and publishing to Kafka.
   let us add connect-standalone.properties.

   The content for the properties file is

```
#bootstrap kafka servers
bootstrap.servers=localhost:9092

# specify input data format
key.converter=org.apache.kafka.connect.storage.StringConverter
value.converter=org.apache.kafka.connect.storage.StringConverter

# The internal converter used for offsets, most will always want to use the built-in default
internal.key.converter=org.apache.kafka.connect.json.JsonConverter
internal.value.converter=org.apache.kafka.connect.json.JsonConverter
internal.key.converter.schemas.enable=false
internal.value.converter.schemas.enable=false

# local file storing offsets and config data
offset.storage.file.filename=connect.offsets
```

4. Try starting the connector

```
bin/confluent start schema-registry
```

5. Optional: View the available predefined connectors with this command:

```
bin/confluent list connectors
```

6. Now, let us Start Kafka Standalone Connector for which we need following two configuration files.
   connect-standalone.properties
   connect-file-source.properties

```
bin/connect-standalone config/my-standalone.properties config/connect-file-source.properties
```

```
[cloudera@quickstart confluent-4.0.0]$ bin/connect-standalone config/my-standalone.properties config/connect-file-source.properties
[2018-03-13 13:14:28,146] INFO Kafka Connect standalone worker initializing ... (org.apache.kafka.connect.cli.ConnectStandalone:65)
[2018-03-13 13:14:28,171] INFO WorkerInfo values:
        jvm.args = -Xmx256M, -XX:+UseG1GC, -XX:MaxGCPauseMillis=20, -XX:InitiatingHeapOccupancyPercent=35, -XX:+ExplicitGCInvokesConcurrent, -Djava.awt.headless=true, -Dcom.sun.manag
icate=false, -Dcom.sun.management.jmxremote.ssl=false, -Dkafka.logs.dir=/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/bin/../logs, -Dlog4j.configuration=file:b
        jvm.spec = Oracle Corporation, Java HotSpot(TM) 64-Bit Server VM, 1.7.0_67, 24.65-b04
        jvm.classpath = /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jackson-mapper-asl-1.9.13.jar:/home/cloudera/training/confluent-oss-4.0.
ule-jaxb-annotations-2.9.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/kafka-log4j-appender-1.0.0-cp1.jar:/home/cloudera/training/conflu
/support-metrics-client-4.0.0.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/hk2-utils-2.5.0-b32.jar:/home/cloudera/training/confluent-oss-
-transforms-1.0.0-cp1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/osgi-resource-locator-1.0.1.jar:/home/cloudera/training/confluent-oss-
.11-1.0.0-cp1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/xz-1.5.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.
loudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jetty-security-9.2.22.v20170606.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.
era/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javax.servlet-api-3.1.0.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java
oudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jersey-server-2.25.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java
nfluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javassist-3.21.0-GA.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jersey-conta
nt-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/log4j-1.2.17.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/plexus-utils-3.0.24.jar:
nfluent-4.0.0/share/java/kafka/kafka_2.11-1.0.0-cp1-scaladoc.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javax.inject-2.5.0-b32.jar:/hom
ent-4.0.0/share/java/kafka/commons-codec-1.9.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/avro-1.8.2.jar:/home/cloudera/training/confluen
ommons-logging-1.2.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/validation-api-1.1.0.Final.jar:/home/cloudera/training/confluent-oss-4.0.
```

Now, let us start the consumer to fetch the record in the topic

```
bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic my-connect-test --from-beginning
```

```
[cloudera@quickstart confluent-4.0.0]$ bin/kafka-console-consumer --bootstrap-server localhost:9092 --topic my-connect-test --from-beginning
This is the last day of the training.
The training is about kafka and other api to connect to kafka.
```

Exercise 2: Apache Kafka Connector Example – Import Data from MySQL into Kafka

Pre-requisites to work with Kafka connect

1. Confluent should be installed
2. Java version 1.7 should be installed
3. Mysql database should be installed
4. Kafka and Schema Registry are running locally on the default ports.

Steps to study the functionality of jdbc-connector using kafka connect

2. Connect to MySQL with user root and password cloudera

```
[cloudera@quickstart ~]$ mysql -u root -p
```

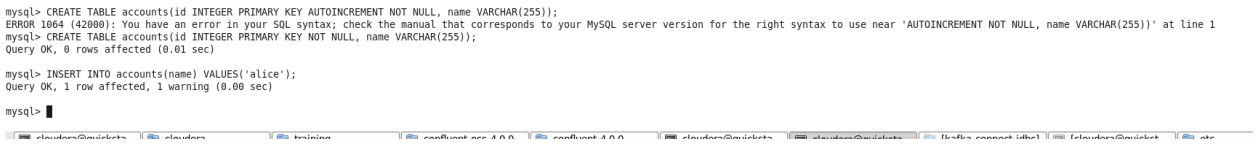
```
[cloudera@quickstart ~]$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

3. Next create a user, database, and a table:

```
mysql> create database training;
Query OK, 1 row affected (0.00 sec)

mysql> use training;
Database changed

mysql> CREATE TABLE accounts
(id integer primary key NOT NULL,
name varchar(255)
);
```

```
mysql> show tables;
+--------------------+
| Tables_in_training |
+--------------------+
| accounts           |
+--------------------+
1 row in set (0.00 sec)

mysql>
```

```
INSERT INTO accounts(name) VALUES('alice');
```



4. Loading the jdbc connector
   a) Checking the predefined connectors in confluent using the below command:
   b) Navigate to the location where confluent is installed.
      cd /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0

bin/confluent list connectors

```
bash: export: command not found
[cloudera@quickstart confluent-4.0.0]$ bin/confluent list connectors
connect is [DOWN]
Bundled Predefined Connectors (edit configuration under etc/):
  elasticsearch-sink
  file-source
  file-sink
  jdbc-source
  jdbc-sink
  hdfs-sink
  s3-sink
[cloudera@quickstart confluent-4.0.0]$
```

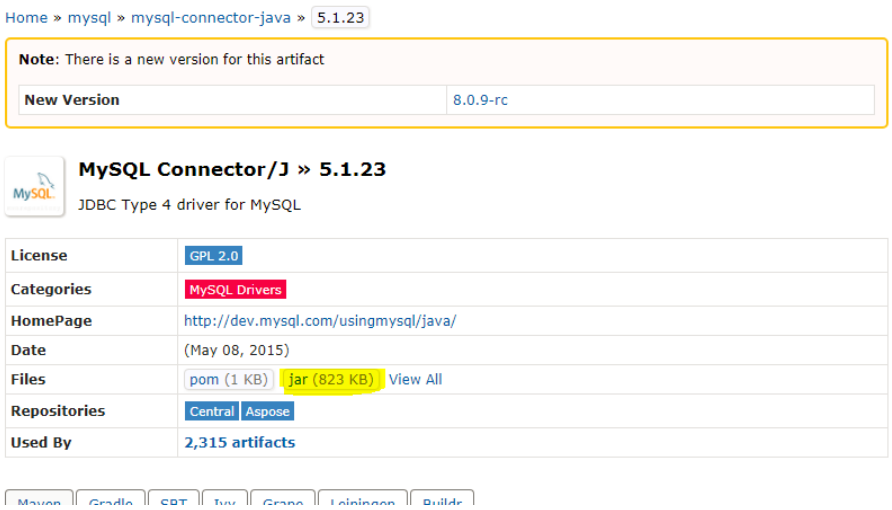
   c) Load the jdbc-source connector with the command:

bin/confluent load jdbc-source from the location /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0 (navigate to the location)

5. Download the mysql connector java jar from below location :

Download the jar from https://mvnrepository.com/artifact/mysql/mysql-connector-java/5.1.23

Click on the highlighted link “jar” which enables downloading the jar(yellow)



6. Copy the above MySQL Connector Jar in the following location :

```
/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka-connect-jdbc
```

7. Create a quickstart-mysql.properties in the same location with the below content:

```
name=test-mysql-jdbc-autoincrement

connector.class=io.confluent.connect.jdbc.JdbcSourceConnector

tasks.max=1

connection.url=jdbc:mysql://localhost:3306/training?user=root&password=cloudera

mode=incrementing

incrementing.column.name=id

table.whitelist=accounts

topic.prefix=test-mysql-jdbc-
```

Notes:

connection.url : In the connector configuration there are no security parameters. This is because SSL is not part of the JDBC standard and will depend on the JDBC driver in use. In general, you will need to configure SSL via the connection.url parameter which this jdbc driver connection url for mysql.

table.whitelist : is the configuration for setting the table which we wat to copy in kafka

if autoincrementing column is set in a table , the column has to be specified.

topic-prefix : Prefix to prepend to table names to generate the name of the Kafka topic to publish data to.

8. Start Zookeeper, Kafka and Schema Registry using the command :

```
bin/confluent start schema-registry
```

```
[2018-03-13 08:50:28,099] INFO Started o.e.j.s.ServletContextHandler@7b621ab4{/,null,AVAILABLE} (org.eclipse.jetty.server.handler.ContextHandler:744)
[2018-03-13 08:50:28,107] INFO Started ServerConnector@1c4db8d1{HTTP/1.1}{0.0.0.0:8083} (org.eclipse.jetty.server.ServerConnector:266)
[2018-03-13 08:50:28,107] INFO Started @23524ms (org.eclipse.jetty.server.Server:379)
[2018-03-13 08:50:28,108] INFO REST server listening at http://127.0.0.1:8083/, advertising URL http://127.0.0.1:8083/ (org.apache.kafka.connect.runtime.rest.RestServer:150)
[2018-03-13 08:50:28,108] INFO Kafka Connect started (org.apache.kafka.connect.runtime.Connect:55)
[2018-03-13 08:50:28,317] INFO ConnectorConfig values:
        connector.class = io.confluent.connect.jdbc.JdbcSourceConnector
        key.converter = null
        name = test-mysql-jdbc-autoincrement
        tasks.max = 1
        transforms = null
        value.converter = null
 (org.apache.kafka.connect.runtime.ConnectorConfig:223)
[2018-03-13 08:50:28,318] INFO EnrichedConnectorConfig values:
        connector.class = io.confluent.connect.jdbc.JdbcSourceConnector
        key.converter = null
        name = test-mysql-jdbc-autoincrement
        tasks.max = 1
        transforms = null
        value.converter = null
```

It starts zookeeper , kafka and schema registry in one go.

9. Now start the standalone jdbc connector mysql source using the below command
   Navigate to the location /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0.

```
bin/connect-standalone   etc/schema-registry/connect-avro-standalone.properties
/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka-connect-jdbc
/quickstart-mysql.properties
```

```
[cloudera@quickstart confluent-4.0.0]$ bin/connect-standalone etc/schema-registry/connect-avro-standalone.properties /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka-connect-jdbc/quickstart-mysql.properties
[2018-03-13 12:40:44,210] INFO Kafka Connect standalone worker initializing ... (org.apache.kafka.connect.cli.ConnectStandalone:65)
[2018-03-13 12:40:44,274] INFO WorkerInfo values:
        jvm.args = -Xmx256M, -XX:+UseG1GC, -XX:MaxGCPauseMillis=20, -XX:InitiatingHeapOccupancyPercent=35, -XX:+ExplicitGCInvokesConcurrent, -Djava.awt.headless=true, -Dcom.sun.management.jmxremote, -Dcom.sun.management.jmxremote.authenticate=false, -Dcom.sun.management.jmxremote.ssl=false, -Dkafka.logs.dir=/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/bin/../logs, -Dlog4j.configuration=file:bin/../etc/kafka/connect-log4j.properties
        jvm.spec = Oracle Corporation, Java HotSpot(TM) 64-Bit Server VM, 1.7.0_67, 24.65-b04
        jvm.classpath = /home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jackson-mapper-asl-1.9.13.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jackson-module-jaxb-annotations-2.9.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/kafka-log4j-appender-1.0.0-cp1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/support-metrics-client-4.0.0.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/hk2-utils-2.5.0-b32.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/connect-transforms-1.0.0-cp1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/osgi-resource-locator-1.0.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/kafka_2.11-1.0.0-cp1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/xz-1.5.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javax.annotation-api-1.2.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jetty-security-9.2.22.v20170606.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/slf4j-log4j12-1.7.25.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javax.servlet-api-3.1.0.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jetty-continuation-9.2.22.v20170606.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jersey-server-2.25.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jopt-simple-5.0.4.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javassist-3.21.0-GA.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jersey-container-servlet-2.25.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/log4j-1.2.17.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/plexus-utils-3.0.24.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/kafka_2.11-1.0.0-cp1-scaladoc.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/javax.inject-2.5.0-b32.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/commons-codec-1.9.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/avro-1.8.2.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/commons-logging-1.2.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/validation-api-1.1.0.Final.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/kafka-streams-1.0.0-cp1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jackson-jaxrs-json-provider-2.9.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/zookeeper-3.4.10.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/jackson-jaxrs-base-2.9.1.jar:/home/cloudera/training/confluent-oss-4.0.0-2.11.tar/confluent-4.0.0/share/java/kafka/commons-validator-1.4.1.j
```

10. To check the data in kafka , run the Connect Avro console consumer

```
bin/kafka-avro-console-consumer --new-consumer --bootstrap-server
localhost:9092 --topic test-mysql-jdbc-accounts --from-beginning
```

```
[cloudera@quickstart confluent-4.0.0]$ bin/kafka-avro-console-consumer --new-con
sumer --bootstrap-server localhost:9092 --topic test-mysql-jdbc-accounts --from-
beginning
The --new-consumer option is deprecated and will be removed in a future major re
lease.The new consumer is used by default if the --bootstrap-server option is pr
ovided.
{"id":0,"name":{"string":"alice"}}
```