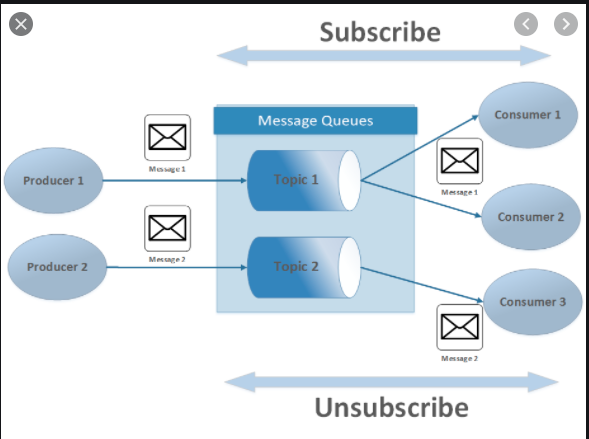
**Kafka**

Kafka is a open source distributed message streaming platform that uses publish and subscribe mechanism to stream the records.

**Publish and Subscribe mechanism in kafka**



**Run kafka and zookeeper server**

nohup /data/confluent/bin/zookeeper-server-start /data/confluent/etc/kafka/zookeeper.properties &

nohup /data/confluent/bin/kafka-server-start /data/confluent/etc/kafka/server.properties &

**Activate Virtual Environment**

source /data/airflow/airflow\_venv/bin/activate

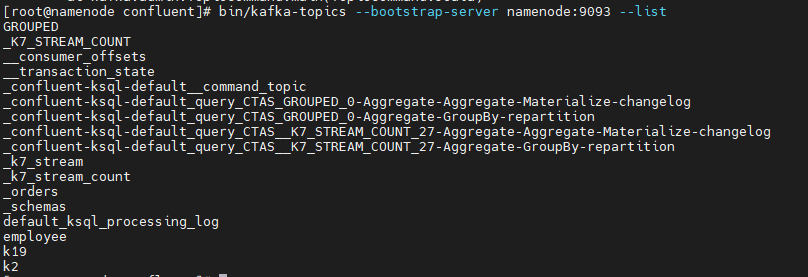
**Create topic**

bin/kafka-topics --bootstrap-server namenode:9093 --create --topic employee



**List topics**

bin/kafka-topics --bootstrap-server namenode:9093 --list



**Delete topic**

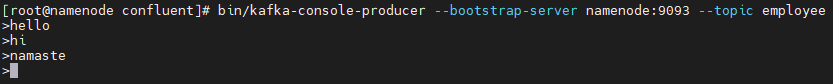
bin/kafka-topics --bootstrap-server namenode:9093 --delete --topic k19



**Producers and consumers**

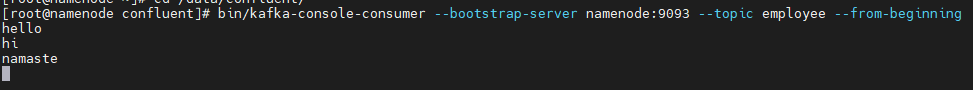
**Create producer**

bin/kafka-console-producer --bootstrap-server namenode:9093 --topic employee



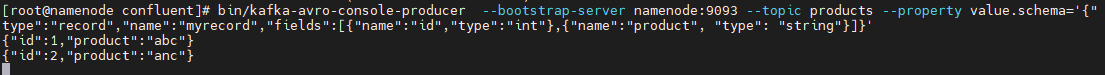
**Create consumer**

bin/kafka-console-consumer --bootstrap-server namenode:9093 --topic employee --from-beginning

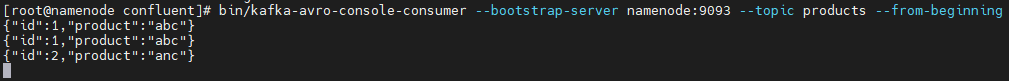


**Avro Producer and Consumer**

bin/kafka-avro-console-producer --bootstrap-server namenode:9093 --topic products --property value.schema='{"type":"record","name":"myrecord","fields":[{"name":"id","type":"int"},{"name":"product", "type": "string"}]}'



bin/kafka-avro-console-consumer --bootstrap-server namenode:9093 --topic products --from-beginning

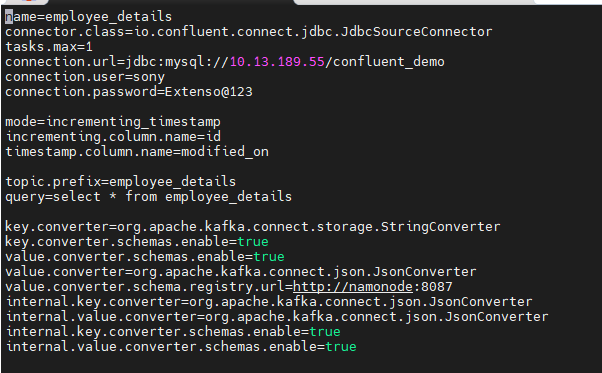


**Source Connetor**

One of the application of source connector is to to transfer data from Database to Kafka topic.

1. Set up properties file for mysql source connector

vi /data/confluent/etc/kafka/connect-mysql.properties



1. Execute the script from command line

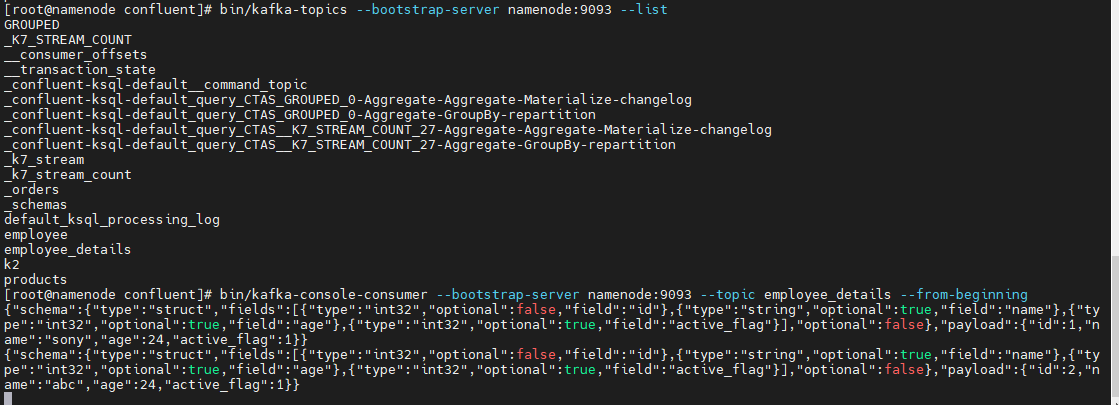
*CLASSPATH=/data/confluent/share/kafka/plugins/lib/mysql-connector-java-8.0.22.jar bin/connect-standalone etc/kafka/connect-standalone.properties etc/kafka/connect-mysql.properties*

Now each time we insert data into table, rows inserted will be reflected in topic.

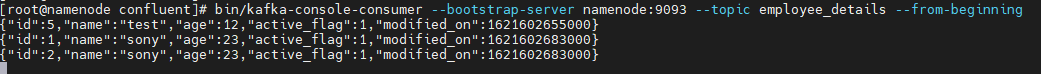
New topic will be created in case topic was not present earlier.

*bin/kafka-console-consumer --bootstrap-server namenode:9093 --topic employee\_details --from-beginning*

Output when schema enable is false:



Output when schema enable is true:

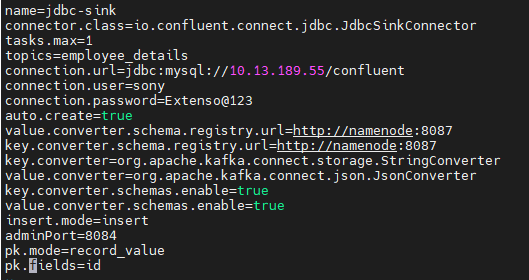


**Sink Connector**

One of the application of Sink connector is used to transfer data from kafka topic to Database.

1. Set up properties file for sink connector

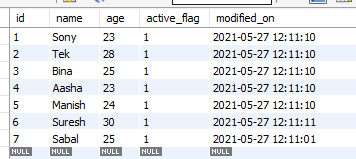
*vi etc/kafka/connect-mysql-sink.properties*



1. Execute following command in command line to run sink connector

*CLASSPATH=/data/confluent/share/kafka/plugins/lib/mysql-connector-java-8.0.22.jar bin/connect-standalone etc/kafka/connect-standalone-sink.properties etc/kafka/connect-mysql-sink.properties*

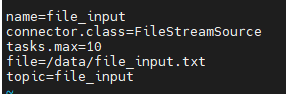
As soon as sink connector starts executing, records in stream will be transmitted to destination table in destination database.



**Create kafka topic for File as an input**

1. Add a sample file in some location say /data/file\_input.txt
2. Make required changes in config file

vi etc/kafka/connect-file-source.properties

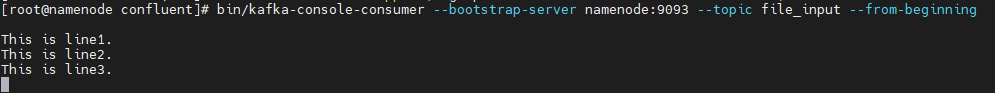


1. Execute File Stream Source connector

*bin/connect-standalone etc/kafka/connect-standalone.properties etc/kafka/connect-file-source.properties*

1. Create a consumer that consumes data from topic created

*bin/kafka-console-consumer --bootstrap-server namenode:9093 --topic file\_input --from-beginning*



**Kafka Python**

**Installation**

pip install kafka-python

**Writing python producer Code**

**A] Producer to produce hello message into a topic**

1. Create a file named producer\_hello.py
2. Write a piece of code that will write “hello” message to topic named hello

from time import sleep

from json import dumps

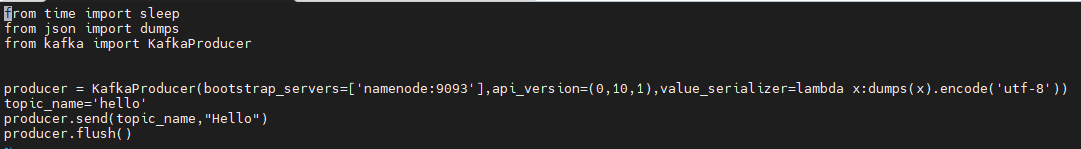
from kafka import KafkaProducer

producer = KafkaProducer(bootstrap\_servers=['namenode:9093'],api\_version=(0,10,1),value\_serializer=lambda x:dumps(x).encode('utf-8'))

topic\_name='hello'

producer.send(topic\_name,"Hello")

producer.flush()



1. Run python code as python producer\_hello.py and message will be sent by producer to respective topic.



**B] Producer to produce message that contains number from 0-10 into a topic named numtest**

from time import sleep

from json import dumps

from kafka import KafkaProducer

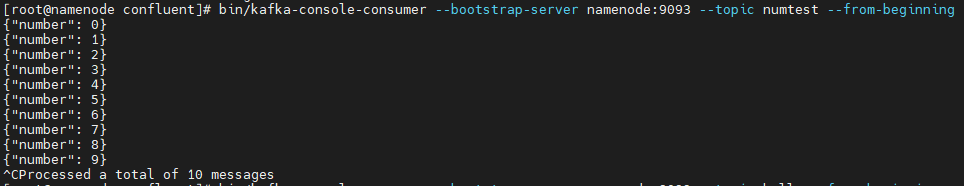
producer = KafkaProducer(bootstrap\_servers=['namenode:9093'],value\_serializer=lambda x:dumps(x).encode('utf-8'))

for e in range(10):

data = {'number' : e}

producer.send('numtest', value=data)

sleep(5)



**C] Producer to send message to topic from Mysql**

*from sqlalchemy import create\_engine*

*from time import sleep*

*from json import dumps*

*from kafka import KafkaProducer*

*producer = KafkaProducer(bootstrap\_servers=['namenode:9093'],value\_serializer=lambda x:dumps(x).encode('utf-8'))*

*engine = create\_engine("mysql://sony:Extenso@123@10.13.189.55/confluent")*

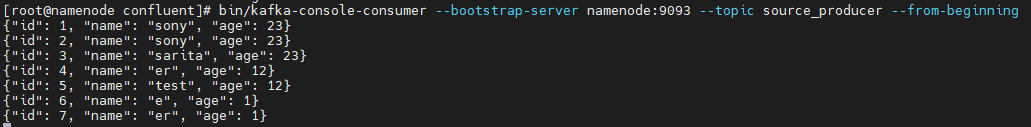
*result = engine.execute("select id,name,age from employee\_details")*

*for row in result:*

*data={"id": row['id'],"name":row['name'],"age":row['age']}*

*producer.send('source\_test', value=data)*

*sleep(5)*



**Writing Python Consumer Code**

A] Python to consume message from a topic and print it in console

*import pandas as pd*

*from kafka import KafkaConsumer*

*from json import loads*

*from sqlalchemy import create\_engine*

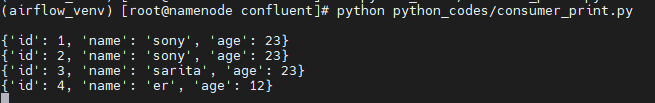
*consumer = KafkaConsumer('emp\_topic',bootstrap\_servers=['namenode:9093'],auto\_offset\_reset='earliest',enable\_auto\_commit=True,group\_id='my\_group')*

*engine = create\_engine("mysql://sony:Extenso@123@10.13.189.55/confluent")*

*for message in consumer:*

*message = loads(message.value)*

*print(message)*



**B] Python to consume message from a topic and insert it into a table in some database**

*import pandas as pd*

*from kafka import KafkaConsumer*

*from json import loads*

*from sqlalchemy import create\_engine*

*consumer = KafkaConsumer('emp\_topic',bootstrap\_servers=['namenode:9093'],auto\_offset\_reset='earliest',enable\_auto\_commit=True,group\_id='roup')*

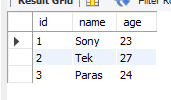
*engine = create\_engine("mysql://sony:Extenso@123@10.13.189.55/confluent")*

*for message in consumer:*

*message = loads(message.value)*

*message\_df=pd.DataFrame([message])*

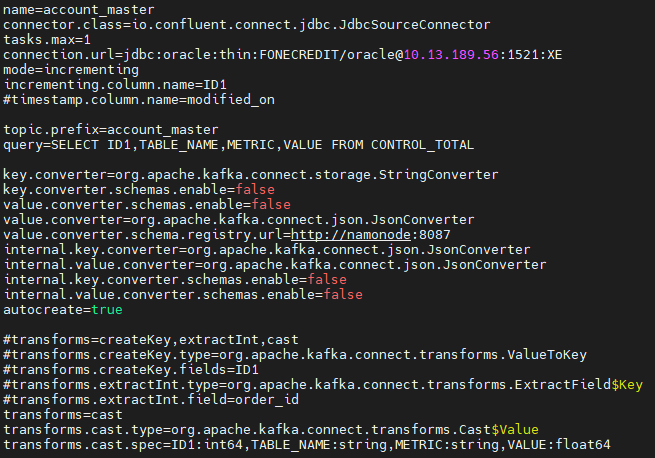
*message\_df.to\_sql('users',con=engine, if\_exists='append',index=False)*



**Oracle as Source Connector and MySQL as Sink Connector**

1. Create a connect-oracle-source.properties file for source connector

*vi /data/confluent/etc/kafka/connect-oracle-source.properties*



*name=account\_master*

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

*tasks.max=1*

*connection.url=jdbc:oracle:thin:FONECREDIT/oracle@10.13.189.56:1521:XE*

*mode=incrementing*

*incrementing.column.name=ID1*

*topic.prefix=account\_master*

*query=SELECT ID1,TABLE\_NAME,METRIC,VALUE FROM CONTROL\_TOTAL*

*key.converter=org.apache.kafka.connect.storage.StringConverter*

*key.converter.schemas.enable=false*

*value.converter.schemas.enable=false*

*value.converter=org.apache.kafka.connect.json.JsonConverter*

*value.converter.schema.registry.url=http://namonode:8087*

*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*

*autocreate=true*

*transforms=cast*

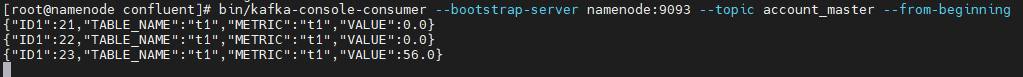
*transforms.cast.type=org.apache.kafka.connect.transforms.Cast$Value*

*transforms.cast.spec=ID1:int64,TABLE\_NAME:string,METRIC:string,VALUE:float64*

1. Execute this oracle source connetor to make oracle source connector work

*CLASSPATH=/data/confluent/share/kafka/plugins/lib/ojdbc7.jar bin/connect-standalone etc/kafka/connect-standalone.properties etc/kafka/connect-oracle-source.properties*

1. A topic named account\_master will be created which can be verified by consuming it using a consumer



1. Write a python program that reads data from topic and insert it into table in MySQL

*import pandas as pd*

*from kafka import KafkaConsumer*

*from json import loads*

*from sqlalchemy import create\_engine*

*consumer = KafkaConsumer('account\_master',bootstrap\_servers=['namenode:9093'],auto\_offset\_reset='earliest',enable\_auto\_commit=True,group\_id='group')*

*engine = create\_engine("mysql://sony:Extenso@123@10.13.189.55/confluent")*

*for message in consumer:*

*message = loads(message.value)*

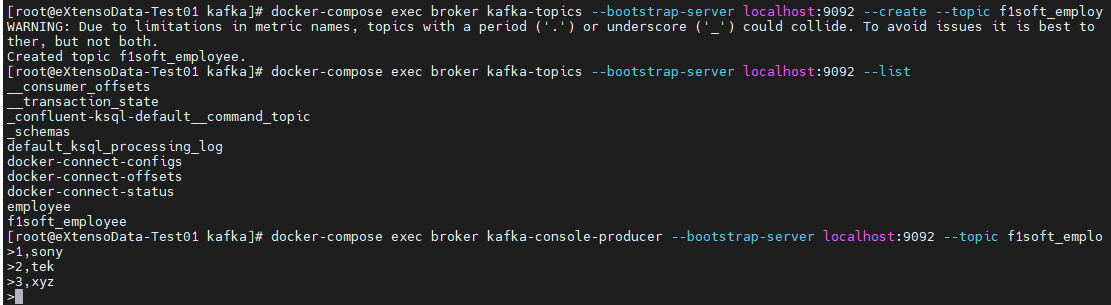
*message\_df=pd.DataFrame([message])*

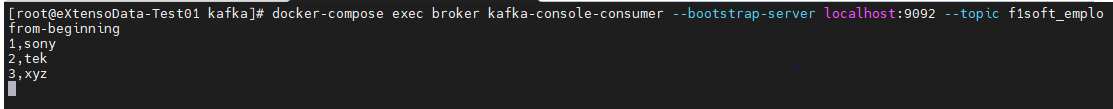
message\_df.to\_sql('account\_master',con=engine, if\_exists='append',index=False)

1. Execute python consumer code and we will be able to view data from oracle view to MySQL table through topic.

**Kafka using docker**

**Producer and Consumer**





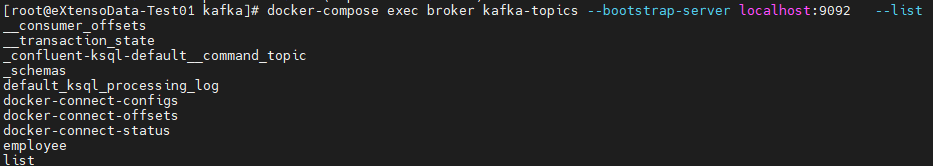
**Create topics**

docker-compose exec broker kafka-topics --bootstrap-server localhost:9092 --create --topic employee



**List topics**

docker-compose exec broker kafka-topics --bootstrap-server localhost:9092 --list



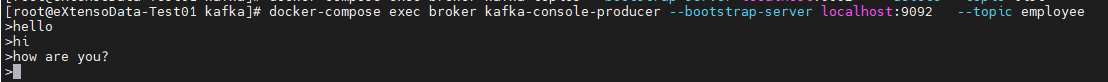
**Delete topics**

docker-compose exec broker kafka-topics --bootstrap-server localhost:9092 --delete --topic employee



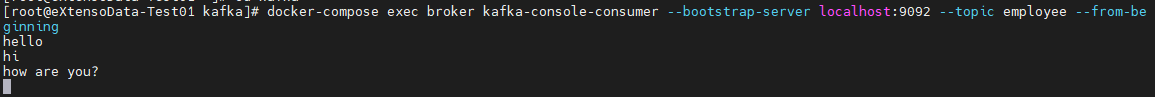
**Create producer**

docker-compose exec broker kafka-console-producer --bootstrap-server localhost:9092 --topic employee



**Create consumer**

docker-compose exec broker kafka-console-consumer --bootstrap-server localhost:9092 --topic employee from-beginning



**Ksqldb**

**Run ksql**

docker-compose exec ksqldb-cli ksql <http://ksqldb-server:8088>

docker-compose -it image\_id exec broker kafka-topics --bootstrap-server localhost:9092 --list