Create Confluent Kafka development cluster and load data into Kafak

1) Launch Confluent Kafka Docker Environment

In same directory as docker-compose.yaml file from course materials run the following.

docker-compose up -d

2) Create Kafka source topic named productsales

docker exec -it broker kafka-topics --create \

--bootstrap-server localhost:9092 \

--topic productsales

3) Create Kafka sink topic named productsales2

docker exec -it broker kafka-topics --create \

--bootstrap-server localhost:9092 --topic productsales2

4) Download the Kafka Connector Jar using HTTPie HTTP Shell Client

Run the following from the same directory as the kafka\_sink.py file downloaded from course materials

http --download https://repo.maven.apache.org/maven2/org/apache/flink/flink-sql-connector-kafka\_2.11/1.13.0/flink-sql-connector-kafka\_2.11-1.13.0.jar

5) Run kafka\_sink.py program

python kakfa\_sink.py

6) In another terminal / cmd prompt start a kafka console producer to write data to productsales topic

docker exec -it broker kafka-console-producer \

--bootstrap-server localhost:9092 \

--topic productsales

7) Paste the following into the console producer then CTRL + C to exit

{"seller\_id": "LNK", "product": "Toothbrush", "quantity": 22, "product\_price": 3.99, "sales\_date": "2021-07-01"}

{"seller\_id": "LNK", "product": "Dental Floss", "quantity": 17, "product\_price": 1.99, "sales\_date": "2021-07-01"}

{"seller\_id": "LNK", "product": "Toothpaste", "quantity": 8, "product\_price": 4.99, "sales\_date": "2021-07-01"}

{"seller\_id": "OMA", "product": "Toothbrush", "quantity": 29, "product\_price": 3.99, "sales\_date": "2021-07-01"}

{"seller\_id": "OMA", "product": "Toothpaste", "quantity": 9, "product\_price": 4.99, "sales\_date": "2021-07-01"}

{"seller\_id": "OMA", "product": "Dental Floss", "quantity": 23, "product\_price": 1.99, "sales\_date": "2021-07-01"}

{"seller\_id": "LNK", "product": "Toothbrush", "quantity": 25, "product\_price": 3.99, "sales\_date": "2021-07-02"}

{"seller\_id": "LNK", "product": "Dental Floss", "quantity": 16, "product\_price": 1.99, "sales\_date": "2021-07-02"}

{"seller\_id": "LNK", "product": "Toothpaste", "quantity": 9, "product\_price": 4.99, "sales\_date": "2021-07-02"}

{"seller\_id": "OMA", "product": "Toothbrush", "quantity": 32, "product\_price": 3.99, "sales\_date": "2021-07-02"}

{"seller\_id": "OMA", "product": "Toothpaste", "quantity": 13, "product\_price": 4.99, "sales\_date": "2021-07-02"}

{"seller\_id": "OMA", "product": "Dental Floss", "quantity": 18, "product\_price": 1.99, "sales\_date": "2021-07-02"}

{"seller\_id": "LNK", "product": "Toothbrush", "quantity": 20, "product\_price": 3.99, "sales\_date": "2021-07-03"}

{"seller\_id": "LNK", "product": "Dental Floss", "quantity": 15, "product\_price": 1.99, "sales\_date": "2021-07-03"}

{"seller\_id": "LNK", "product": "Toothpaste", "quantity": 11, "product\_price": 4.99, "sales\_date": "2021-07-03"}

{"seller\_id": "OMA", "product": "Toothbrush", "quantity": 31, "product\_price": 3.99, "sales\_date": "2021-07-03"}

{"seller\_id": "OMA", "product": "Toothpaste", "quantity": 10, "product\_price": 4.99, "sales\_date": "2021-07-03"}

{"seller\_id": "OMA", "product": "Dental Floss", "quantity": 21, "product\_price": 1.99, "sales\_date": "2021-07-03"}

8) Run kafka-console-consumer to be sure data was loaded into source topic productsales

docker exec -it broker kafka-console-consumer --from-beginning \

--bootstrap-server localhost:9092 \

--topic productsales

9) Verify that the productsales2 topic has data using the kafka-console-consumer

docker exec -it broker kafka-console-consumer --from-beginning \

--bootstrap-server localhost:9092 \

--topic productsales2

10) Clean up docker based Kafka environment

Run the following from the same directory as docker-compose.yaml file downloaded from course materials

docker-compose down -v