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 salesitems

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

--bootstrap-server localhost:9092 \

--topic salesitems

3) Create Kafka sink topics named processedsales and processedsales2

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

--bootstrap-server localhost:9092 --topic processedsales

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

--bootstrap-server localhost:9092 --topic processedsales2

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

Run the following from the same directory as the tumbling\_windows.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 sales\_producer.py program

python sales\_producer.py --bootstrap-server localhost:9092 --topic salesitems

6) In another terminal / cmd prompt (with the Python Virtual Environment activated) run the Flink program tumbling\_windows.py

To run the Table API version:

python tumbling\_windows.py

To run the SQL Interface version:

python tumbling\_windows\_sql.py

7) Verify that the processedsales and processedsales2 topic has data using the kafka-console-consumer

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

--bootstrap-server localhost:9092 \

--topic processedsales

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

--bootstrap-server localhost:9092 \

--topic processedsales2

8) 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