Final Assignment (Part 2) - Creating Streaming Data Pipelines using Kafka

Scenario

You are a data engineer at a data analytics consulting company. You have been assigned to a project that aims to de-congest the national highways by analyzing the road traffic data from different toll plazas. As a vehicle passes a toll plaza, the vehicle's data like vehicle_id,vehicle_type,toll_plaza_id and timestamp are streamed to Kafka. Your job is to create a data pipe line that collects the streaming data and loads it into a database.

Objectives

In this assignment you will create a streaming data pipe by performing these steps:

- Start a MySQL Database server.
- Create a table to hold the toll data.
- Start the Kafka server.
- Install the Kafka python driver.
- Install the MySQL python driver.
- Create a topic named toll in kafka.
- Download streaming data generator program.
- Customize the generator program to steam to toll topic.
- Download and customise streaming data consumer.
- Customize the consumer program to write into a MySQL database table.
- Verify that streamed data is being collected in the database table.

Exercise 1 - Prepare the lab environment

Step 1: Download Kafka.

- wget https://archive.apache.org/dist/kafka/2.8.0/kafka_2.12-2.8.0.tgz

Step 2: Extract Kafka.

- tar -xzf kafka_2.12-2.8.0.tgz

Step 3: Start MySQL server.

- start_mysql

Step 4: Connect to the mysql server, using the command below. Make sure you use the password given to you when the MySQL server starts. Please make a note or record of the password because you will need it later.

- mysql --host=127.0.0.1 --port=3306 --user=root --password=Mjk0NDQtcnNhbm5h

Step 5: Create a database named tolldata.

mysql > create database tolldata;

Step 6: Create a table named livetolldata with the schema to store the data generated by the traffic simulator.

Run the following command to create the table:

mysql > use tolldata;

mysql > create table livetolldata(timestamp datetime,vehicle_id int,vehicle_type char(15),toll_plaza_id smallint);

Step 7: Disconnect from MySQL server.

mysql > Exit

Step 8: Install the python module kafka-python using the pip command. This python module will help you to communicate with kafka server. It can used to send and receive messages from kafka.

- python3 -m pip install kafka-python

Step 9: Install the python module mysql-connector-python using the pip command. This python module will help you to interact with mysql server.

- python3 -m pip install mysql-connector-python==8.0.31

Exercise 2 - Start Kafka

Task 2.1 - Start Zookeeper

Start zookeeper server.

Task 2.2 - Start Kafka server

Start Kafka server

Task 2.3 - Create a topic named toll

Create a Kakfa topic named toll

Task 2.4 - Download the Toll Traffic Simulator

Download the toll_traffic_generator.py from the url given below using 'wget'.

- wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Final%20Assignment/toll_traffic_generator.py

Open the code using the theia editor using the "Menu –> File –>Open" option.

Task 2.5 - Configure the Toll Traffic Simulator

Open the toll_traffic_generator.py and set the topic to toll.

Task 2.6 - Run the Toll Traffic Simulator

Run the toll_traffic_generator.py.

Hint : python3 <pythonfilename> runs a python program on the theia lab.

Task 2.7 - Configure streaming_data_reader.py

Download the streaming_data_reader.py from the url below using 'wget'.

- wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DB0250EN-SkillsNetwork/labs/Final%20Assignment/streaming_data_reader.py

Open the streaming_data_reader.py and modify the following details so that the program can connect to your mysql server.

- TOPIC
- DATABASE
- USERNAME
- PASSWORD

Task 2.8 - Run streaming_data_reader.py

Run the streaming_data_reader.py

python3 streaming_data_reader.py

Task 2.9 - Health check of the streaming data pipeline.

If you have done all the steps till here correctly, the streaming toll data would get stored in the table livetolldata.

List the top 10 rows in the table livetolldata.

```
theia@theiadocker-delacruzdan1:/home/project$ cd kafka_2.12-2.8.0
theia@theiadocker-delacruzdan1:/home/project/kafka_2.12-2.8.0$ bin/zookeeper-server-start.sh config/zookeeper.properties
JVMJ9VM007W Command-line option unrecognised: -Xlog:gc*:file=/home/project/kafka_2.12-2.8.0/bin/../logs/zookeeper-gc.log:time,tags:filecount=10,file
size=100M
[2023-02-24 17:32:04,181] INFO Reading configuration from: config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-02-24 17:32:04,183] WARN config/zookeeper.properties is relative. Prepend ./ to indicate that you're sure! (org.apache.zookeeper.server.quorum
.QuorumPeerConfig)
[2023-02-24 17:32:04,198] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-02-24 17:32:04,199] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-02-24 17:32:04,207] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2023-02-24 17:32:04,207] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
```

```
theia@theiadocker-delacruzdan1:/home/project$ cd kafka_2.12-2.8.0
theia@theiadocker-delacruzdan1:/home/project/kafka_2.12-2.8.0$ bin/kafka-server-start.sh config/server.properties
JVMJ9VM007W Command-line option unrecognised: -Xlog:gc*:file=/home/project/kafka_2.12-2.8.0/bin/../logs/kafkaServer-gc.log:time,tags:filecount=10,fi
lesize=100M
[2023-02-24 17:33:49,556] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2023-02-24 17:33:49,943] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation (org.apache.
```

```
theia@theiadocker-delacruzdan1:/home/project$ cd kafka_2.12-2.8.0
theia@theiadocker-delacruzdan1:/home/project/kafka_2.12-2.8.0$ bin/kafka-topics.sh --create --topic toll --bootstrap-server localhost:9092
Created topic toll.
```

toll_traffic_generator.py

```python
"""
Top Traffic Simulator
"""
from time import sleep, time, ctime
from random import random, randint, choice
from kafka import KafkaProducer
producer = KafkaProducer(bootstrap_servers='localhost:9092')

TOPIC = 'toll'

VEHICLE_TYPES = ("car", "car", "car", "car", "car", "car", "car", "car",
                 "car", "car", "car", "truck", "truck", "truck",
                 "truck", "van", "van")
for _ in range(100000):
    vehicle_id = randint(10000, 10000000)
    vehicle_type = choice(VEHICLE_TYPES)
    now = ctime(time())
    plaza_id = randint(4000, 4010)
    message = f"{now},{vehicle_id},{vehicle_type},{plaza_id}"
    message = bytearray(message.encode("utf-8"))
    print(f"A {vehicle_type} has passed by the toll plaza {plaza_id} at {now}.")
    producer.send(TOPIC, message)
    sleep(random() * 2)

```

```
theia@theiadocker-delacruzdan1:/home/project$ python3 toll_traffic_generator.py
A truck has passed by the toll plaza 4001 at Fri Feb 24 17:40:02 2023.
A van has passed by the toll plaza 4008 at Fri Feb 24 17:40:04 2023.
A truck has passed by the toll plaza 4008 at Fri Feb 24 17:40:05 2023.
A van has passed by the toll plaza 4000 at Fri Feb 24 17:40:07 2023.
A car has passed by the toll plaza 4007 at Fri Feb 24 17:40:08 2023.
A car has passed by the toll plaza 4008 at Fri Feb 24 17:40:09 2023.
A car has passed by the toll plaza 4005 at Fri Feb 24 17:40:11 2023.
A truck has passed by the toll plaza 4000 at Fri Feb 24 17:40:12 2023.
A car has passed by the toll plaza 4010 at Fri Feb 24 17:40:14 2023.
A truck has passed by the toll plaza 4007 at Fri Feb 24 17:40:14 2023.
A van has passed by the toll plaza 4006 at Fri Feb 24 17:40:16 2023.
A truck has passed by the toll plaza 4007 at Fri Feb 24 17:40:17 2023.
A truck has passed by the toll plaza 4010 at Fri Feb 24 17:40:18 2023.
A car has passed by the toll plaza 4001 at Fri Feb 24 17:40:20 2023.
A truck has passed by the toll plaza 4009 at Fri Feb 24 17:40:21 2023.
A car has passed by the toll plaza 4008 at Fri Feb 24 17:40:21 2023.
A truck has passed by the toll plaza 4001 at Fri Feb 24 17:40:22 2023.
A car has passed by the toll plaza 4006 at Fri Feb 24 17:40:23 2023.
A car has passed by the toll plaza 4008 at Fri Feb 24 17:40:23 2023.
A car has passed by the toll plaza 4001 at Fri Feb 24 17:40:24 2023.
```

toll_traffic_generator.py | streaming_data_reader.py

streaming_data_reader.py > ...

```python
"""
Streaming data consumer
"""
from datetime import datetime
from kafka import KafkaConsumer
import mysql.connector

TOPIC='toll'
DATABASE = 'mysql'
USERNAME = 'delacruzdan1'
PASSWORD = 'MTA1NzgtZGVsYWNy'

print("Connecting to the database")
try:
    connection = mysql.connector.connect(host='localhost', database=DATABASE, user=USERNAME, password=PASSWORD)
except Exception:
    print("Could not connect to database. Please check credentials")
else:
    print("Connected to database")
cursor = connection.cursor()

print("Connecting to Kafka")
consumer = KafkaConsumer(TOPIC)
print("Connected to Kafka")
print(f"Reading messages from the topic {TOPIC}")
for msg in consumer:

    # Extract information from kafka

    message = msg.value.decode("utf-8")

    # Transform the date format to suit the database schema
    (timestamp, vehcile_id, vehicle_type, plaza_id) = message.split(",")

    dateobj = datetime.strptime(timestamp, '%a %b %d %H:%M:%S %Y')
    timestamp = dateobj.strftime("%Y-%m-%d %H:%M:%S")

    # Loading data into the database table

    sql = "insert into livetolldata values(%s,%s,%s,%s)"
    result = cursor.execute(sql, (timestamp, vehcile_id, vehicle_type, plaza_id))
    print(f"A {vehicle_type} was inserted into the database")
    connection.commit()
connection.close()
```

```
theia@theiadocker-delacruzdan1:/home/project$ python3 streaming_data_reader.py
Connecting to the database
Connected to database
Connecting to Kafka
Connected to Kafka
Reading messages from the topic toll
A truck was inserted into the database
A car was inserted into the database
A car was inserted into the database
A car was inserted into the database
A van was inserted into the database
A truck was inserted into the database
A car was inserted into the database
A truck was inserted into the database
A car was inserted into the database
A car was inserted into the database
A car was inserted into the database
A truck was inserted into the database
A truck was inserted into the database
A car was inserted into the database
A van was inserted into the database
A car was inserted into the database
A car was inserted into the database
A van was inserted into the database
```

```
mysql> SELECT * FROM livetolldata LIMIT 10;
+---------------------+------------+--------------+---------------+
| timestamp           | vehicle_id | vehicle_type | toll_plaza_id |
+---------------------+------------+--------------+---------------+
| 2023-02-24 17:52:01 |    5117515 | truck        |          4005 |
| 2023-02-24 17:52:02 |    3986772 | car          |          4009 |
| 2023-02-24 17:52:03 |    3252296 | car          |          4007 |
| 2023-02-24 17:52:05 |     611507 | car          |          4002 |
| 2023-02-24 17:52:06 |    1320647 | van          |          4010 |
| 2023-02-24 17:52:07 |    8055309 | truck        |          4010 |
| 2023-02-24 17:52:08 |    1439727 | car          |          4002 |
| 2023-02-24 17:52:08 |    9523750 | truck        |          4004 |
| 2023-02-24 17:52:10 |    9543838 | car          |          4005 |
| 2023-02-24 17:52:12 |    3795626 | car          |          4003 |
+---------------------+------------+--------------+---------------+
10 rows in set (0.00 sec)
```