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

- 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 customize 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

Instructions to set up lab environment

Download Kafka

```
wget https://archive.apache.org/dist/kafka/3.7.0/kafka_2.12-3.7.0.tgz
```

Extract Kafka from the zip file

```
tar -xzf kafka_2.12-3.7.0.tgz
```

```
theia@theiadocker-vaishnavis26:/home/project$ wget https://archive.apache.org/dist/kafka/3.7.0/kafka_2.12-3.7.0.tgz
--2025-06-05 07:16:09--  https://archive.apache.org/dist/kafka/3.7.0/kafka_2.12-3.7.0.tgz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 65.108.204.189
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 119203328 (114M) [application/x-gzip]
Saving to: 'kafka_2.12-3.7.0.tgz'

kafka_2.12-3.7.0.tgz     100%[=============================>] 113.68M  26.5MB/s    in 5.1s

2025-06-05 07:16:14 (22.3 MB/s) - 'kafka_2.12-3.7.0.tgz' saved [119203328/119203328]

theia@theiadocker-vaishnavis26:/home/project$ ls
kafka_2.12-3.7.0.tgz
theia@theiadocker-vaishnavis26:/home/project$ tar -xzf kafka_2.12-3.7.0.tgz
theia@theiadocker-vaishnavis26:/home/project$ ls
kafka_2.12-3.7.0  kafka_2.12-3.7.0.tgz
theia@theiadocker-vaishnavis26:/home/project$
```

Configure KRaft and start server

Change to the kafka_2.12-3.7.0 directory.

```
cd kafka_2.12-3.7.0
```

Generate a cluster UUID that will uniquely identify the Kafka cluster.

```
KAFKA_CLUSTER_ID="$(bin/kafka-storage.sh random-uuid)"
```

KRaft requires the log directories to be configured. Run the following command to configure the log directories passing the cluster id.

```
bin/kafka-storage.sh format -t $KAFKA_CLUSTER_ID -c config/kraft/server.properties
```

Now that KRaft is configured, you can start the Kafka server by running the following command.

```
bin/kafka-server-start.sh config/kraft/server.properties
```

```
theia@theiadocker-vaishnavis26:/home/project$ cd kafka_2.12-3.7.0
theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$
theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$ KAFKA_CLUSTER_ID="$(bin/kafka-stor
age.sh random-uuid)"
theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$ ls
LICENSE  NOTICE  bin  config  libs  licenses  site-docs
theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$
theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$ bin/kafka-storage.sh format -t $KA
FKA_CLUSTER_ID -c config/kraft/server.properties
metaPropertiesEnsemble=MetaPropertiesEnsemble(metadataLogDir=Optional.empty, dirs={/tmp/kraft-com
bined-logs: EMPTY})
Formatting /tmp/kraft-combined-logs with metadata.version 3.7-IV4.
theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$

theia@theiadocker-vaishnavis26:/home/project/kafka_2.12-3.7.0$ bin/kafka-server-start.sh config/k
raft/server.properties
[2025-06-05 07:20:43,436] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log
4jControllerRegistration$)
[2025-06-05 07:20:43,828] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disa
ble client-initiated TLS renegotiation (org.apache.zookeeper.common.X509Util)
[2025-06-05 07:20:44,061] INFO Registered signal handlers for TERM, INT, HUP (org.apache.kafka.co
mmon.utils.LoggingSignalHandler)
[2025-06-05 07:20:44,065] INFO [ControllerServer id=1] Starting controller (kafka.server.Controll
erServer)
to be started (kafka.server.BrokerServer)
[2025-06-05 07:20:46,724] INFO [BrokerServer id=1] Finished waiting for all of the SocketServer A
cceptors to be started (kafka.server.BrokerServer)
[2025-06-05 07:20:46,724] INFO [BrokerServer id=1] Transition from STARTING to STARTED (kafka.ser
ver.BrokerServer)
[2025-06-05 07:20:46,724] INFO Kafka version: 3.7.0 (org.apache.kafka.common.utils.AppInfoParser)
[2025-06-05 07:20:46,725] INFO Kafka commitId: 2ae524ed625438c5 (org.apache.kafka.common.utils.Ap
pInfoParser)
[2025-06-05 07:20:46,725] INFO Kafka startTimeMs: 1749122446724 (org.apache.kafka.common.utils.Ap
pInfoParser)
[2025-06-05 07:20:46,726] INFO [KafkaRaftServer nodeId=1] Kafka Server started (kafka.server.Kafk
aRaftServer)
```

Start MySQL server and setup the database

Open New terminal.

Connect to the MySQL server using the command below in the terminal. Replace pwd according the server password

```
mysql --host=mysql --port=3306 --user=root --password='C0M30Oo1IvgHgZUYKKG3u144'
```

create the database

```
create database tolldata;
```

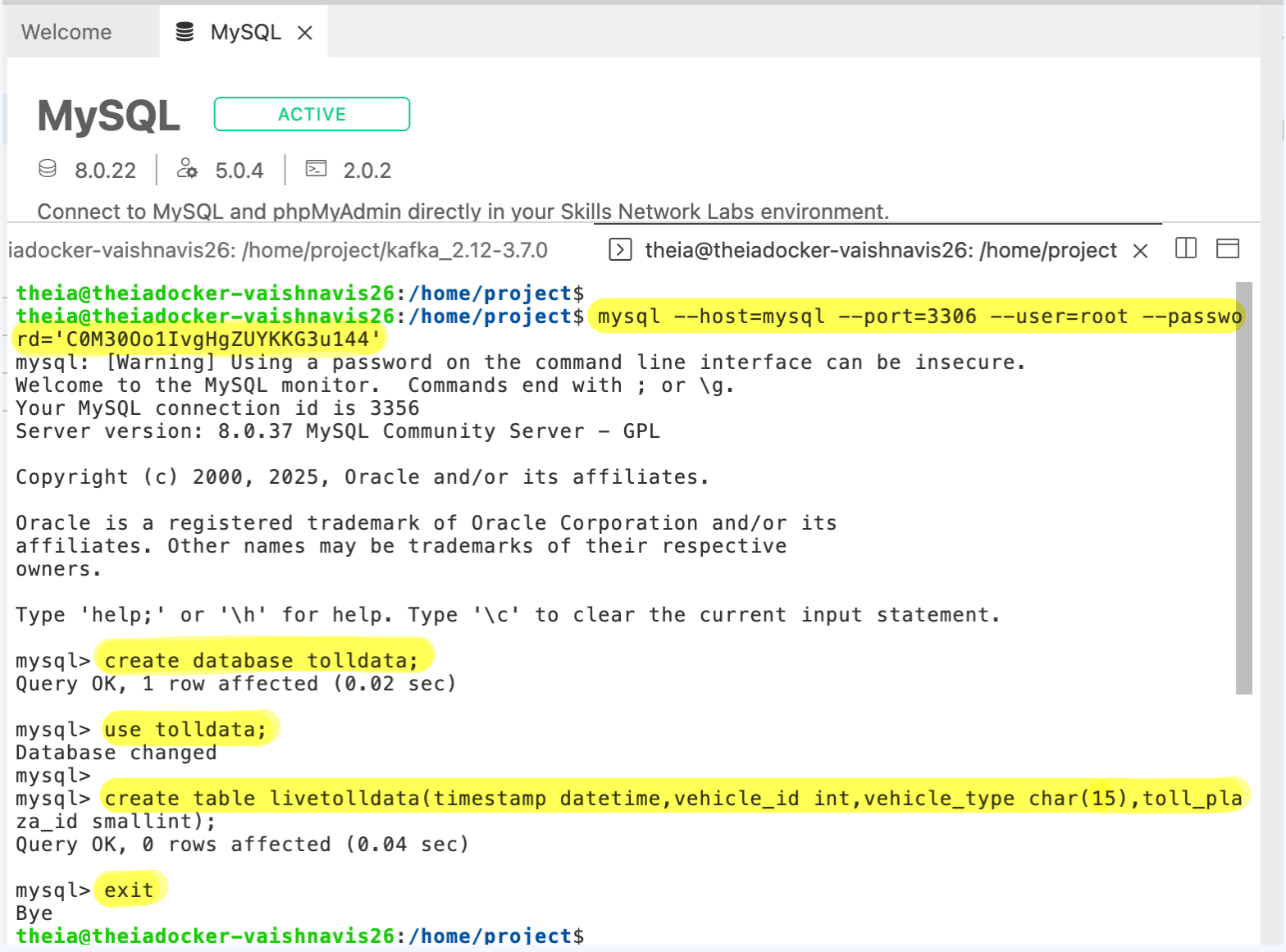
Create a table named livetolldata with the schema to store the data generated by the traffic simulator.

```
use tolldata;

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

Disconnect from the MySQL server.

```
exit
```



Install the Python packages

Install the Python module kafka-python

- This Python module will help you to communicate with kafka server. It can used to send and receive messages from Kafka.

```
pip3 install kafka-python
```

Install the Python module mysql-connector-python using the pip command.

```
pip3 install mysql-connector-python==8.0.31
```

```
theia@theiadocker-vaishnavis26:/home/project$ pip3 install kafka-python
Defaulting to user installation because normal site-packages is not writeable
Collecting kafka-python
  Downloading kafka_python-2.2.10-py2.py3-none-any.whl (309 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 309.3/309.3 KB 7.2 MB/s eta 0:00:00
Installing collected packages: kafka-python
Successfully installed kafka-python-2.2.10
theia@theiadocker-vaishnavis26:/home/project$ pip3 install mysql-connector-python==8.0.31
Defaulting to user installation because normal site-packages is not writeable
Collecting mysql-connector-python==8.0.31
  Downloading mysql_connector_python-8.0.31-cp310-cp310-manylinux1_x86_64.whl (23.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.5/23.5 MB 69.5 MB/s eta 0:00:00
Collecting protobuf<=3.20.1,>=3.11.0
  Downloading protobuf-3.20.1-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 71.7 MB/s eta 0:00:00
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.31 protobuf-3.20.1
theia@theiadocker-vaishnavis26:/home/project$
```

toll_traffic_generator.py

Producer side code to Generate Data

```
"""
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-vaishnavis26:/home/project$ python3 toll_traffic_generator.py
A van has passed by the toll plaza 4010 at Thu Jun  5 07:54:20 2025.
A car has passed by the toll plaza 4007 at Thu Jun  5 07:54:22 2025.
A car has passed by the toll plaza 4007 at Thu Jun  5 07:54:23 2025.
A van has passed by the toll plaza 4001 at Thu Jun  5 07:54:25 2025.
A truck has passed by the toll plaza 4002 at Thu Jun  5 07:54:27 2025.
A truck has passed by the toll plaza 4005 at Thu Jun  5 07:54:28 2025.
A car has passed by the toll plaza 4000 at Thu Jun  5 07:54:30 2025.
```

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

TOPIC='toll'
DATABASE = 'tolldata'
USERNAME = 'root'
PASSWORD = 'C0M30Oo1IvgHgZUYKKG3u144'

print("Connecting to the database")
try:
    connection = mysql.connector.connect(host='mysql', 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-vaishnavis26:/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
```

```
mysql> select * from livetolldata limit 10;
+---------------------+------------+--------------+---------------+
| timestamp           | vehicle_id | vehicle_type | toll_plaza_id |
+---------------------+------------+--------------+---------------+
| 2025-06-05 07:56:27 |    1110057 | truck        |          4008 |
| 2025-06-05 07:56:27 |    5097529 | car          |          4001 |
| 2025-06-05 07:56:29 |    4097430 | car          |          4002 |
| 2025-06-05 07:56:29 |    8872895 | car          |          4001 |
| 2025-06-05 07:56:31 |    4879280 | truck        |          4008 |
| 2025-06-05 07:56:32 |    1171545 | car          |          4009 |
| 2025-06-05 07:56:34 |    7519766 | truck        |          4009 |
| 2025-06-05 07:56:34 |    9470552 | truck        |          4004 |
| 2025-06-05 07:56:35 |    1970009 | car          |          4001 |
| 2025-06-05 07:56:36 |    4930854 | truck        |          4010 |
+---------------------+------------+--------------+---------------+
10 rows in set (0.00 sec)
```