# Capstone: Instant Health Alert System – Mid Submission

***Create Python script to create producer application to read from RDS and push the message into the topic in below format and list the messages in the topic***

***kafka_produce_patient_vitals.py***

- Import the necessary dependencies

```
# Import required dependencies
from kafka import KafkaProducer
import mysql.connector
import time
import json
```

- Collect the whole patient vital data from RDS by providing host, database, and user credentials.

```
# Collect whole data from rds
connection = mysql.connector.connect(host='upgraddetest.cyaielc9bmnf.us-east-1.r
ds.amazonaws.com', database='testdatabase',user='student', password='STUDENT123'
)
cursor = connection.cursor()
query = "SELECT customerId, heartBeat,bp FROM patients_vital_info"

# Execute the above query
cursor.execute(query)
patient_vital_records = cursor.fetchall()

cursor.close()
connection.close()
```

- Setting up producer configurations by providing bootstrap server and serializer.

```
# Setting up the producer configurations
topicName = 'patient_vital_topic'
producer = KafkaProducer(bootstrap_servers = ['ec2-44-196-94-216.compute-1.amazo
naws.com:9092'], api_version=(0,11,5),value_serializer=lambda x:json.dumps(x).en
code('utf-8'))
```

- Pushing the message from producer every second to the Kakfa topic ***patient_vital_topic***

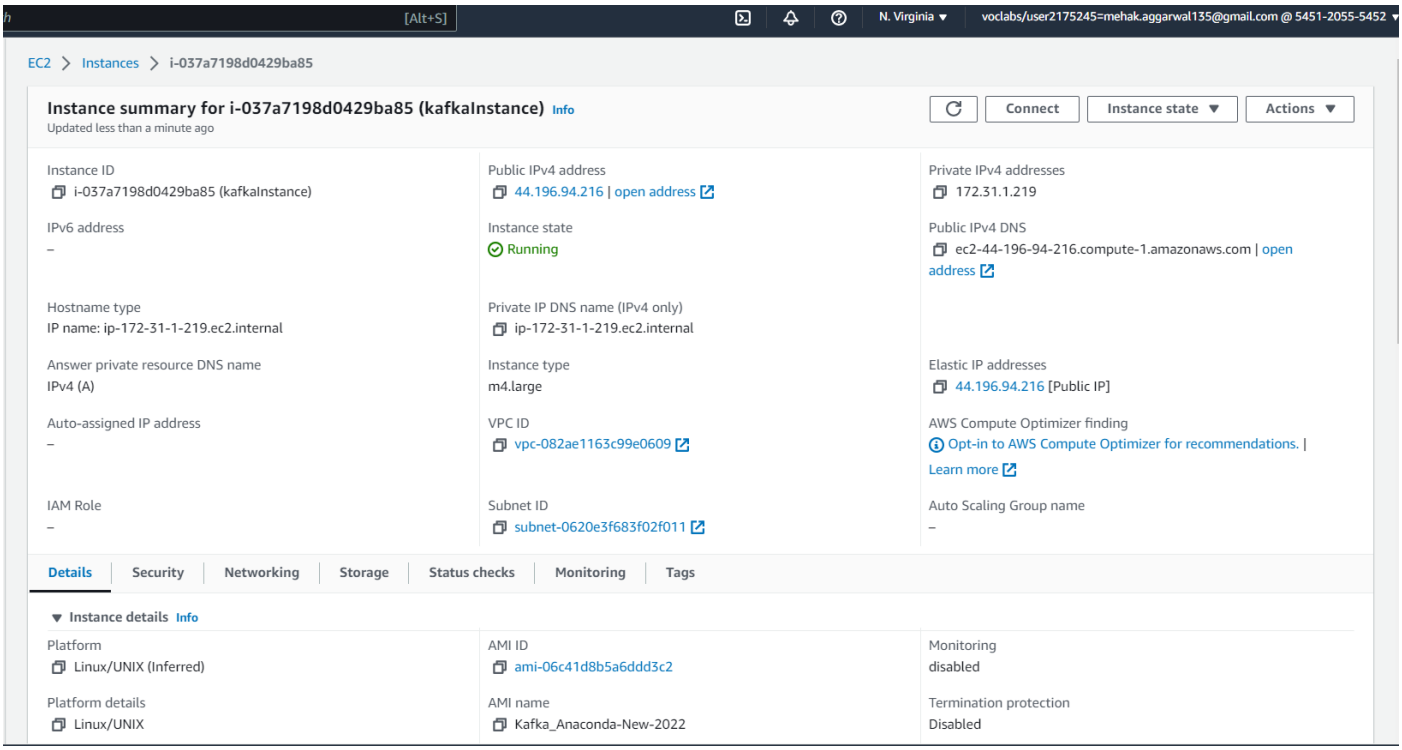
```
# Sending the message from the producer every second
for customerId,heartBeat,bp in patient_vital_records:

    # create a message to send in JSON format
    msg = {"customerId":customerId,"heartBeat":heartBeat,"bp":bp}

    # send the message to the Kafka topic
    producer.send(topicName,msg)
    print(msg)

    # wait for one second before sending the next message
    time.sleep(1)
-- INSERT --                                          34,18         Bot
```

## *Kafka EC2 instance Configuration*




## *Start the Zookeeper server*

```
cd downloads/kafka_2.12-2.3.0
bin/zookeeper-server-start.sh config/zookeeper.properties
```

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ bin/zookeeper-server-start.sh confi
g/zookeeper.properties
```

```
[2023-03-27 07:36:02,631] INFO Server environment:os.version=4.14.193-149.317.am
zn2.x86_64 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-03-27 07:36:02,631] INFO Server environment:user.name=ec2-user (org.apache
.zookeeper.server.ZooKeeperServer)
[2023-03-27 07:36:02,631] INFO Server environment:user.home=/home/ec2-user (org.
apache.zookeeper.server.ZooKeeperServer)
[2023-03-27 07:36:02,631] INFO Server environment:user.dir=/home/ec2-user/downlo
ads/kafka_2.12-2.3.0 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-03-27 07:36:02,767] INFO tickTime set to 3000 (org.apache.zookeeper.server
.ZooKeeperServer)
[2023-03-27 07:36:02,767] INFO minSessionTimeout set to -1 (org.apache.zookeeper
.server.ZooKeeperServer)
[2023-03-27 07:36:02,767] INFO maxSessionTimeout set to -1 (org.apache.zookeeper
.server.ZooKeeperServer)
[2023-03-27 07:36:02,860] INFO Using org.apache.zookeeper.server.NIOServerCnxnFa
ctory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactor
y)
[2023-03-27 07:36:02,912] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.
zookeeper.server.NIOServerCnxnFactory)
```

## *Start the Kafka server*

```
cd downloads/kafka_2.12-2.3.0
bin/kafka-server-start.sh config/server.properties
```

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ bin/kafka-server-start.sh config/se
rver.properties
```

```
[2023-03-27 07:39:07,453] INFO [GroupMetadataManager brokerId=0] Finished loadin
g offsets and group metadata from __consumer_offsets-42 in 0 milliseconds. (kafk
a.coordinator.group.GroupMetadataManager)
[2023-03-27 07:39:07,453] INFO [GroupMetadataManager brokerId=0] Finished loadin
g offsets and group metadata from __consumer_offsets-45 in 0 milliseconds. (kafk
a.coordinator.group.GroupMetadataManager)
[2023-03-27 07:39:07,454] INFO [GroupMetadataManager brokerId=0] Finished loadin
g offsets and group metadata from __consumer_offsets-48 in 1 milliseconds. (kafk
a.coordinator.group.GroupMetadataManager)
[2023-03-27 07:39:17,421] INFO [GroupCoordinator 0]: Member consumer-1-c0563008-
a3f4-404a-b164-3308ce9844a5 in group spark-kafka-source-a58d042d-622a-465e-b549-
83750e6d4bc8-502956407-driver-0 has failed, removing it from the group (kafka.co
ordinator.group.GroupCoordinator)
[2023-03-27 07:39:17,425] INFO [GroupCoordinator 0]: Preparing to rebalance grou
p spark-kafka-source-a58d042d-622a-465e-b549-83750e6d4bc8-502956407-driver-0 in
state PreparingRebalance with old generation 1 (__consumer_offsets-34) (reason:
removing member consumer-1-c0563008-a3f4-404a-b164-3308ce9844a5 on heartbeat exp
iration) (kafka.coordinator.group.GroupCoordinator)
[2023-03-27 07:39:17,428] INFO [GroupCoordinator 0]: Group spark-kafka-source-a5
8d042d-622a-465e-b549-83750e6d4bc8-502956407-driver-0 with generation 2 is now e
mpty (__consumer_offsets-34) (kafka.coordinator.group.GroupCoordinator)
```

### *Create the Kafka topic (patient_vital_topic)*

```
cd downloads/kafka_2.12-2.3.0
bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 -
-topic patient_vital_topic
```

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --create --boot
strap-server localhost:9092 --replication-factor 1 --partitions 1 --topic patien
t_vital_topic
```

### *View the list of Kafka topics*

```
bin/kafka-topics.sh --list --bootstrap-server ec2-44-196-94-216.compute-1.amazonaws.com:9092
```

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ bin/kafka-topics.sh --list --bootst
rap-server ec2-44-196-94-216.compute-1.amazonaws.com:9092
__consumer_offsets
patient_vital_topic
```

Use WINSCP to transfer file to downloads/kafka_2.12-2.3.0 folder.

```
ec2-user@ip-172-31-1-219:~/downloads/kafka_2.12-2.3.0
login as: ec2-user
Authenticating with public key "imported-openssh-key"
Last login: Mon Mar 27 07:32:47 2023 from 49.43.96.190

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-1-219 ~]$ cd downloads/kafka_2.12-2.3.0
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ ls
bin     kafka_produce_patient_vitals.py  LICENSE  NOTICE
config  libs                             logs     site-docs
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$
```

Install the required package dependencies

```
pip install kafka-python
pip install mysql-connector-python
pip install mysql-connector-repackaged
```

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ pip install mysql-connector-python
```

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ pip install mysql-connector-repacka
ged
```

### *Run the python producer script*

This python script reads data from RDS per second and simulates IoT devices to push patient vital data in the JSON format into the Kafka Queue.

```
python kafka_produce_patient_vitals.py
```

**Screenshot of JSON vital information written to the console (per second)**

```
[ec2-user@ip-172-31-1-219 kafka_2.12-2.3.0]$ python kafka_produce_patient_vitals
.py
{'heartBeat': 74, 'bp': 202, 'customerId': 1}
{'heartBeat': 68, 'bp': 173, 'customerId': 2}
{'heartBeat': 71, 'bp': 152, 'customerId': 3}
{'heartBeat': 72, 'bp': 166, 'customerId': 4}
{'heartBeat': 68, 'bp': 171, 'customerId': 5}
{'heartBeat': 70, 'bp': 189, 'customerId': 1}
{'heartBeat': 72, 'bp': 173, 'customerId': 2}
{'heartBeat': 68, 'bp': 178, 'customerId': 3}
```