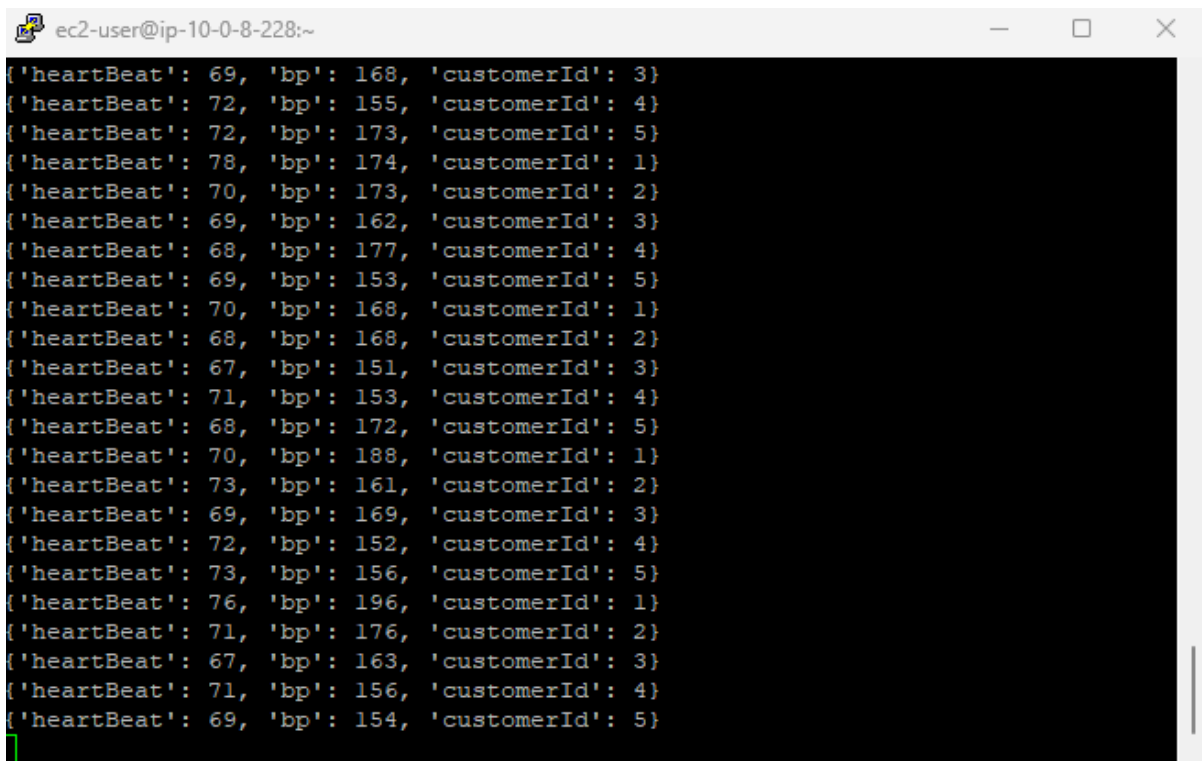# Code Logic – Instant Health Alert System

In this project, we will push the patient vital information to Kafka topic, through spark streaming we will get the data compare with threshold values. If patient vital information is not within lower and upper limits then email notification is sent to the doctor.

Code logic in kafka_produce_patient_vitals.py file

1. Using mysql.connector module of python patient vital information is fetched from RDS (Relational Database Service) stored in patients_vital_info table.
2. The fetched data from step 1 is converted to json format.
3. The json data from second step is sent to Kafka topic "patient-vital-info".
4. The json data is sent to kafka topic with 1 second gap between the data.


```
ec2-user@ip-10-0-8-228:~
{'heartBeat': 69, 'bp': 168, 'customerId': 3}
{'heartBeat': 72, 'bp': 155, 'customerId': 4}
{'heartBeat': 72, 'bp': 173, 'customerId': 5}
{'heartBeat': 78, 'bp': 174, 'customerId': 1}
{'heartBeat': 70, 'bp': 173, 'customerId': 2}
{'heartBeat': 69, 'bp': 162, 'customerId': 3}
{'heartBeat': 68, 'bp': 177, 'customerId': 4}
{'heartBeat': 69, 'bp': 153, 'customerId': 5}
{'heartBeat': 70, 'bp': 168, 'customerId': 1}
{'heartBeat': 68, 'bp': 168, 'customerId': 2}
{'heartBeat': 67, 'bp': 151, 'customerId': 3}
{'heartBeat': 71, 'bp': 153, 'customerId': 4}
{'heartBeat': 68, 'bp': 172, 'customerId': 5}
{'heartBeat': 70, 'bp': 188, 'customerId': 1}
{'heartBeat': 73, 'bp': 161, 'customerId': 2}
{'heartBeat': 69, 'bp': 169, 'customerId': 3}
{'heartBeat': 72, 'bp': 152, 'customerId': 4}
{'heartBeat': 73, 'bp': 156, 'customerId': 5}
{'heartBeat': 76, 'bp': 196, 'customerId': 1}
{'heartBeat': 71, 'bp': 176, 'customerId': 2}
{'heartBeat': 67, 'bp': 163, 'customerId': 3}
{'heartBeat': 71, 'bp': 156, 'customerId': 4}
{'heartBeat': 69, 'bp': 154, 'customerId': 5}
```

Fig 1: Screen shot showing the json data which is sent into kafka topic.

Code logic in kafka_spark_patient_vitals.py file

1. Spark Code to readStream from kafka using ip address and port number of kafka server and subscribing to "patient-vital-info" topic.

2. Schema is defined to read data from kafka topic

3. The data is collected in batches as and when kafka topic receives the data.

4. For each collected records a message time field is added which holds the current time stamp.

5. The collected data is written to console using writeStream function.

6. The collected data is written to hdfs file system using parquet format.

```
hadoop@ip-10-0-10-63:~                                    —    □    ×

|customerId|heartBeat|bp |message_time         |
+----------+---------+---+---------------------+
|3         |69       |168|2023-05-20 07:21:40.003|
|4         |71       |160|2023-05-20 07:21:40.003|
|5         |68       |170|2023-05-20 07:21:40.003|
|1         |74       |165|2023-05-20 07:21:40.003|
|2         |72       |161|2023-05-20 07:21:40.003|
|3         |69       |156|2023-05-20 07:21:40.003|
|4         |69       |170|2023-05-20 07:21:40.003|
|5         |69       |165|2023-05-20 07:21:40.003|
|1         |74       |192|2023-05-20 07:21:40.003|
|2         |68       |169|2023-05-20 07:21:40.003|
+----------+---------+---+---------------------+


-------------------------------------------
Batch: 6
-------------------------------------------

+----------+---------+---+---------------------+
|customerId|heartBeat|bp |message_time         |
+----------+---------+---+---------------------+
|3         |67       |177|2023-05-20 07:21:50.003|
|4         |67       |160|2023-05-20 07:21:50.003|
|5         |69       |178|2023-05-20 07:21:50.003|
|1         |72       |187|2023-05-20 07:21:50.003|
|2         |73       |163|2023-05-20 07:21:50.003|
|3         |72       |173|2023-05-20 07:21:50.003|
|4         |72       |163|2023-05-20 07:21:50.003|
|5         |72       |155|2023-05-20 07:21:50.003|
|1         |70       |211|2023-05-20 07:21:50.003|
|2         |72       |163|2023-05-20 07:21:50.003|
+----------+---------+---+---------------------+


-------------------------------------------
Batch: 7
-------------------------------------------

+----------+---------+---+---------------------+
|customerId|heartBeat|bp |message_time         |
+----------+---------+---+---------------------+
|3         |56       |171|2023-05-20 07:22:00.003|
|4         |66       |157|2023-05-20 07:22:00.003|
|5         |73       |166|2023-05-20 07:22:00.003|
|1         |71       |210|2023-05-20 07:22:00.003|
|2         |67       |161|2023-05-20 07:22:00.003|
|3         |71       |160|2023-05-20 07:22:00.003|
|4         |71       |157|2023-05-20 07:22:00.003|
|5         |67       |171|2023-05-20 07:22:00.003|
|1         |77       |183|2023-05-20 07:22:00.003|
|2         |69       |151|2023-05-20 07:22:00.003|
+----------+---------+---+---------------------+
```

Fig 2: Screen shot showing spark streaming in batches

Fig 3: Screen shot showing parquet file list