

CODE LOGIC

Script 1 is the producer application -- Fetches data from RDS and pushes patient's vital details into

Kafka Queue 1 (**PatientVitalInfo**)

Script 2 is used to store the incoming data from the kafka queue onto HDFS (for storing into Hive)

Script 3 is used to compare patient vital details with threshold & fetch related contact information and push anomalous data into Kafka Queue 2 (**DoctorQueue**)

Script 4 is python application to send SNS alert to subscribe email ID.

Commands to start the scripts [In the mentioned Order]:

Script 1) kafka_produce_patient_vitals.py [M1] : `python kafka_produce_patient_vitals.py`

Script 2) kafka_spark_patient_vitals.py [M2] :

`export SPARK_KAFKA_VERSION=0.10`

`spark2-submit --jars spark-sql-kafka-0-10_2.11-2.3.0.jar kafka_spark_patient_vitals.py`

Script 3) kafka_spark_generate_alerts.py [M2] :

`export SPARK_KAFKA_VERSION=0.10`

`spark2-submit --jars spark-sql-kafka-0-10_2.11-2.3.0.jar kafka_spark_generate_alerts.py`

Script 4) kafka_consume_alerts.py [M1]:

`python kafka_consume_alerts.py`

Here,

M1 = Machine 1 --> Running Kafka brokers for both topics --> PatientVitalInfo & DoctorQueue

M2 = Machine 2 [Cloudera Box] --> Running apache-spark --> To perform manipulation of streaming

data and for storing data on hive,hbase

```
#!/bin/bash
```

```
# Start zookeeper & kafka-server
```

```
bin/zookeeper-server-start.sh config/zookeeper.properties &
```

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

```
# Script for Kafka Topic Creation
```

```
# CREATE PatientVitalInfo TOPIC
```

```
# Create topic [ single partition with replication factor 1 ]
```

```
bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1  
--topic PatientVitalInfo
```

```
# Verify Topic was successfully created
```

```
bin/kafka-topics.sh --list --bootstrap-server localhost:9092
```

```
# Make sure messages are going into topic while running the producer application
```

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic PatientVitalInfo --  
frombeginning
```

```
# CREATE DoctorQueue TOPIC
```

```
# Create topic [ single partition with replication factor 1 ]  
bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1  
--topic DoctorQueue  
# Verify Topic was successfully created  
bin/kafka-topics.sh --list --bootstrap-server localhost:9092  
# Make sure messages are going into topic while running the producer application  
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 --topic DoctorQueue --  
frombeginning
```