1) Created an EMR cluster version (5.30.1)with 1 master(General purpose with 40 GB) memory and 2 slave nodes with Hadoop,sqoop,spark,Hive,Hbase,Hcatalog,Hue,zeppelin
2) To create a Hbase table that contains threshold values , login into EMR cluster and type hbase shell and enter.

3) Then execute all the commands given in **hbase.pdf**

4) To create the topics which acts as a streaming execute **kafka.pdf** (mid-submission)commands in local host
5) To execute the producer application use below command.

6) spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 kafka_spark_patient_vitals.py

7) To create all 2 hive tables (patient vitals information, Reference_threshold) , login into EMR cluster and type hive to enter hive commandline console.

8) Execute the commands given in **hive1.pdf**(mid-submission),**hive2.pdf**

9) Before using Sqoop to connect to RDS and import data, configure mysql connector to do the import operation , exceute below statments
   - wget https://de-mysql-connector.s3.amazonaws.com/mysql-connector-java-8.0.25.tar.gz
   - tar -xvf mysql-connector-java-8.0.25.tar.gz
   - cd mysql-connector-java-8.0.25/
   - sudo cp mysql-connector-java-8.0.25.jar /usr/lib/sqoop/lib/

10) Now configure the correct connection database connection parameters such as port, database , table to import data into hive table , execute **Sqoop.pdf**

11) To execute the spark streaming application that analyses and alerts patient use below command in emr cluster.Before that create a kafka topic **"doctors-queue"** in local host.

   - **bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic doctors-queue**
   - **export SPARK_KAFKA_VERSION=0.10**
   - **spark-submit --jars /opt/cloudera/parcels/CDH-5.15.1-1.cdh5.15.p0.4/lib/hive/hive-hbase-handler.jar --packages org.apache.hbase:hbase0.92.1 --packages org.apache.spark:spark-sql-kafka-0-10_2.11:2.4.5 kafka_spark_generate_alerts.py**

12) To execute the kafka consumer application , first configure SNS using amazon console and test by publishing the message into topic "Doctorqueue" take a note of the ARN number to use in consumer application.
13) Use boto3 library to publish messages into SNS queue as push notification
14) Execute "**python kafka_consume_alerts.py"** to **start the consumer application to send alerts as sns push notifications**