Capstone: Instant Health Alert System – Mid Submission

A script to build an external hive table for the threshold data and view threshold data

An HBase table named **threshold_ref** has been created in HBase with 3 column families: attribute, limit, alert. A hive table has been created on top of this HBase table.

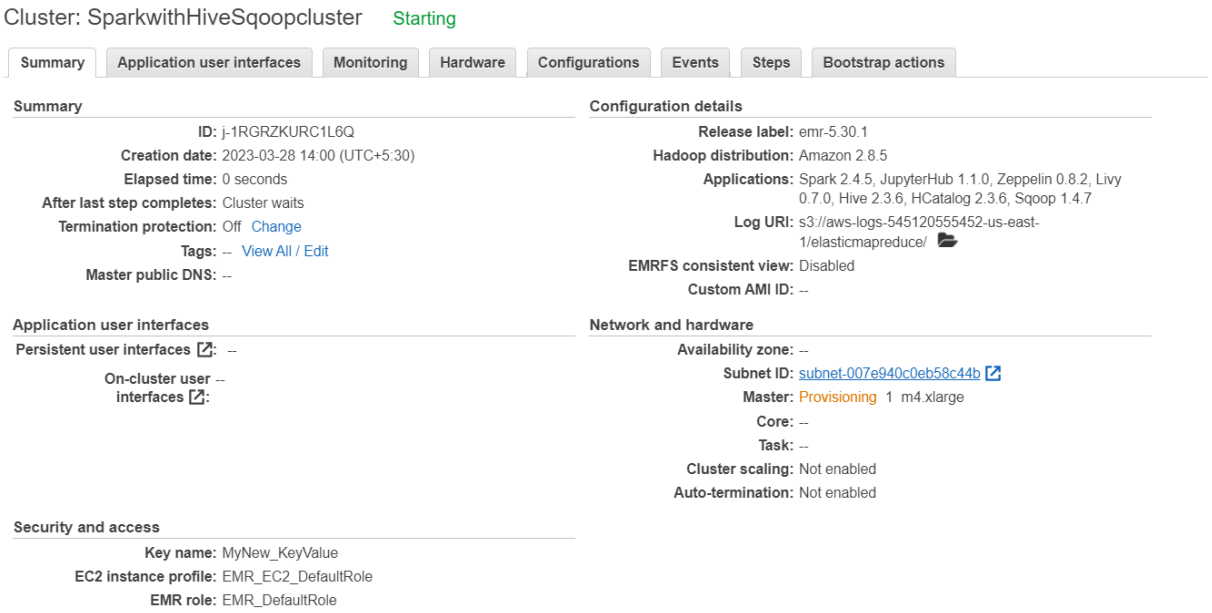
Threshold_Reference_Table in Hive

```
CREATE EXTERNAL TABLE Threshold_Reference_Table (
  key int,
  Attribute string,
  low_age_limit int,
  high_age_limit int,
  Low_Range_Value int,
  High_Range_Value int,
  Alert_Flag int,
  Alert_Message string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
  'hbase.columns.mapping' = ':key, attribute:attribute, limit:low_age_limit, limit:high_age_limit, limit:low_value,
limit:high_value, alert:alert_flag, alert:alert_message',
  'hbase.table.name' = 'threshold_ref'
)
TBLPROPERTIES ('hbase.mapred.output.outputtable' = 'threshold_ref');
```

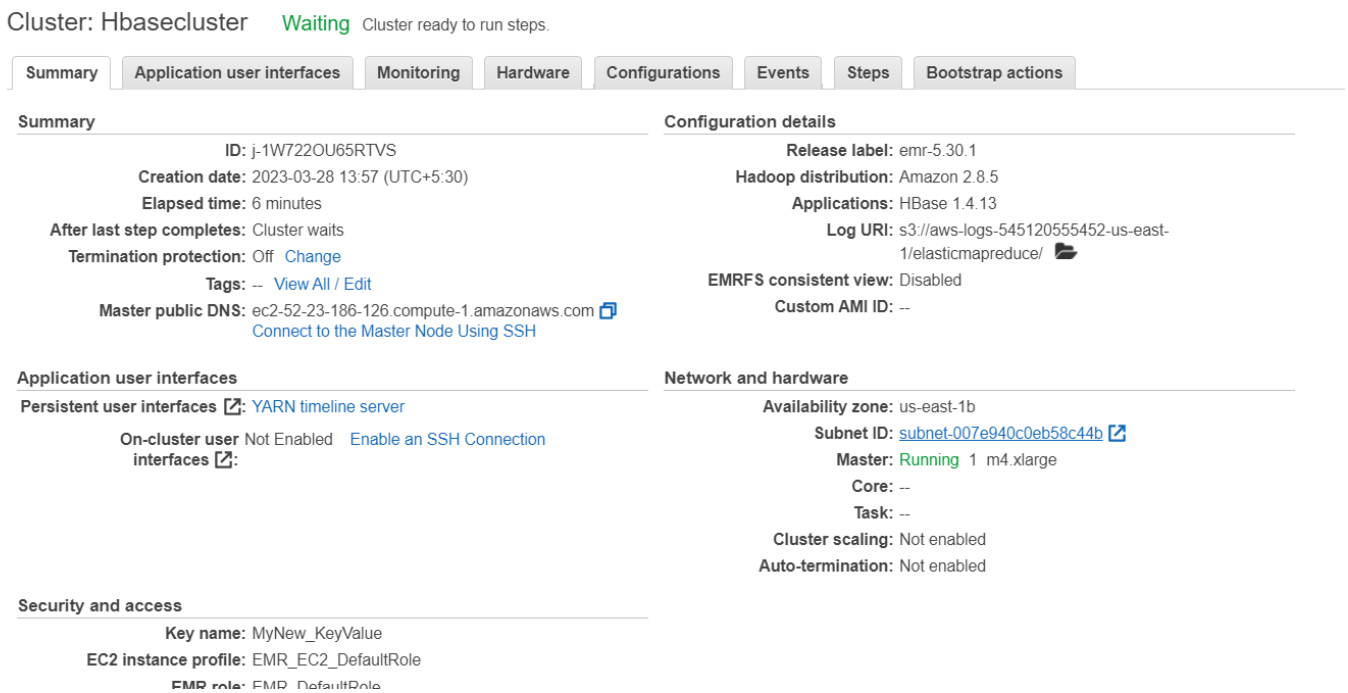
Set up for the Hive and HBase integration

1. Set up Hive and HBase on two separate clusters.

Screenshot of Hive Cluster



Screenshot of HBase Cluster



2. For the Hive-HBase integration on different clusters, few inbound rules were added to the security group for HBase master node and Hive master node.

Screenshot of HBase cluster's master node (Security Group – master rules)

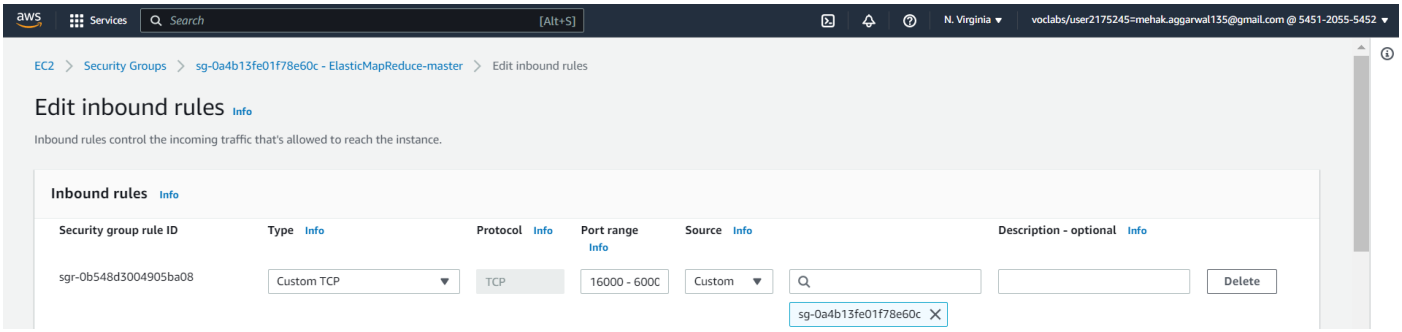
Following rule is added:

Type: "Custom TCP Rule"

Protocol: "TCP"

Port Range: "16000 - 60000"

Source: "Custom" and enter the Master Security Group for Hive cluster's master node.



Screenshot of Hive cluster's master node (Security Group – master rules)

Following rule is added:

Type: "Custom TCP Rule"

Protocol: "TCP"

Port: "10000"

Source: "Custom" and enter the Master Security Group for HBase cluster's master node.

In the Hive Shell

- Connect the HBase client on your Hive cluster to the HBase cluster that contains your data.

set hbase.zookeeper.quorum= <public DNS name of the master node of the HBase cluster>;

```
set hbase.zookeeper.quorum=ec2-52-23-186-126.compute-1.amazonaws.com;
```

```
^C[hadoop@ip-172-31-82-68 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: true
hive> set hbase.zookeeper.quorum=ec2-52-23-186-126.compute-1.amazonaws.com;
```

- Create a database ***patient_health_care***

```
create database if not exists patient_health_care;
```

```
hive> create database if not exists patient_health_care;
OK
Time taken: 0.855 seconds
```

- Use database patient_health_care

```
use patient_health_care;
```

```
hive> use patient_health_care;
OK
Time taken: 0.046 seconds
```

- Create external table named ***Threshold_Reference_Table***

```
CREATE EXTERNAL TABLE Threshold_Reference_Table (
  key int,
  Attribute string,
  low_age_limit int,
  high_age_limit int,
  Low_Range_Value int,
  High_Range_Value int,
  Alert_Flag int,
  Alert_Message string
)
STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
WITH SERDEPROPERTIES (
  'hbase.columns.mapping' = ':key, attribute:attribute, limit:low_age_limit, limit:high_age_limit,
limit:low_value, limit:high_value, alert:alert_flag, alert:alert_message',
  'hbase.table.name' = 'threshold_ref'
)
TBLPROPERTIES ('hbase.mapred.output.outputtable' = 'threshold_ref');
```

```
hive> CREATE EXTERNAL TABLE Threshold_Reference_Table (
    >    key int,
    >    Attribute string,
    >    low_age_limit int,
    >    high_age_limit int,
    >    Low_Range_Value int,
    >    High_Range_Value int,
    >    Alert_Flag int,
    >    Alert_Message string
    > )
    > STORED BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler'
    > WITH SERDEPROPERTIES (
    >    'hbase.columns.mapping' = ':key, attribute:attribute, limit:low_age_lim
it, limit:high_age_limit, limit:low_value, limit:high_value, alert:alert_flag, a
lert:alert_message',
    >    'hbase.table.name' = 'threshold_ref'
    > )
    > TBLPROPERTIES ('hbase.mapred.output.outputtable' = 'threshold_ref');
OK
Time taken: 2.31 seconds
```

- View the contents of Threshold_Reference_Table

```
set hive.cli.print.header = true;
SELECT * FROM Threshold_Reference_Table order by key;
```

```
hive> set hive.cli.print.header = true;
hive> select * from Threshold_Reference_Table order by key;
Query ID = hadoop_20230328084446_4e04390b-9e15-4799-a8f7-ef03baf510dd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1679992808558
_0001)

Map 1: -/-      Reducer 2: 0/1
Map 1: 0/1      Reducer 2: 0/1
Map 1: 0/1      Reducer 2: 0/1
Map 1: 0(+1)/1  Reducer 2: 0/1
Map 1: 1/1      Reducer 2: 0(+1)/1
Map 1: 1/1      Reducer 2: 1/1
```

Screenshot of Threshold_Reference_Table records:

```
OK
threshold_reference_table.key   threshold_reference_table.attribute     threshol
d_reference_table.low_age_limit threshold_reference_table.high_age_limit       t
hreshold_reference_table.low_range_value        threshold_reference_table.high_r
ange_value      threshold_reference_table.alert_flag    threshold_reference_tabl
e.alert_message
1       heartBeat       0       40      0       69      1       Low Heart Rate t
han Normal
2       heartBeat       0       40      70      78      0       Normal
3       heartBeat       0       40      79      9999    1       Higher Heart Rat
e than Normal
4       bp      0       40      0       160     1       Low BP than Normal
5       bp      0       40      161     220     0       Normal
6       bp      0       40      221     9999    1       Higer BP than Normal
7       heartBeat       41      100     0       65      1       Low Heart Rate t
han Normal
8       heartBeat       41      100     66      73      0       Normal
9       heartBeat       41      100     74      9999    1       Higher Heart Rat
e than Normal
10      bp      41      100     0       150     1       Low BP than Normal
11      bp      41      100     151     180     0       Normal
12      bp      41      100     181     9999    1       Higher BP than Normal
Time taken: 15.749 seconds, Fetched: 12 row(s)
```