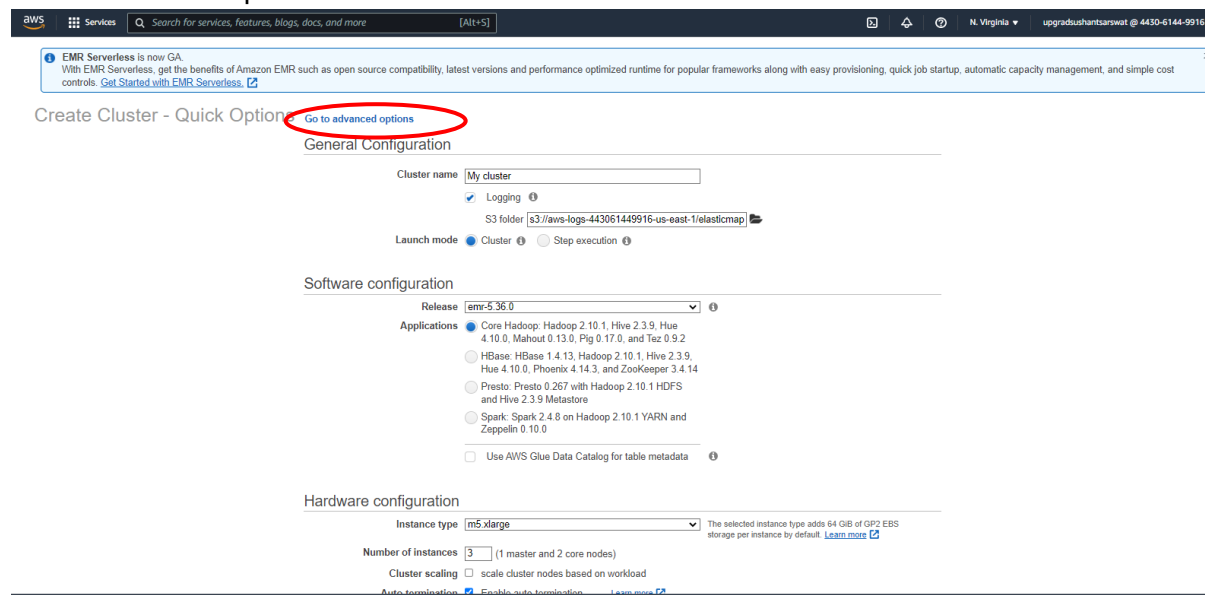


CAPSTONE PROJECT INSTANT HEALTH ALERT SYSTEM

SUBMITTED BY : Sushant Sarswat (C-38)

1. Setting up EMR cluster for Hive and other software. Select EMR in AWS services and Select 'Go to advanced options'



The screenshot shows the AWS EMR console 'Create Cluster - Quick Options' page. A red circle highlights the 'Go to advanced options' link. The page includes sections for General Configuration, Software configuration, and Hardware configuration.

General Configuration

- Cluster name: My cluster
- Logging: ☒ Logging
- S3 folder: s3://aws-logs-44306144916-us-east-1/elasticmap
- Launch mode: ☒ Cluster ☐ Step execution

Software configuration

Release: emr-5.36.0

Applications:

- ☒ Core Hadoop: Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2
- ☐ HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.9, Hue 4.10.0, Phoenix 4.14.3, and ZooKeeper 3.4.14
- ☐ Presto: Presto 0.267 with Hadoop 2.10.1 HDFS and Hive 2.3.9 Metastore
- ☐ Spark: Spark 2.4.8 on Hadoop 2.10.1 YARN and Zeppelin 0.10.0
- ☐ Use AWS Glue Data Catalog for table metadata

Hardware configuration

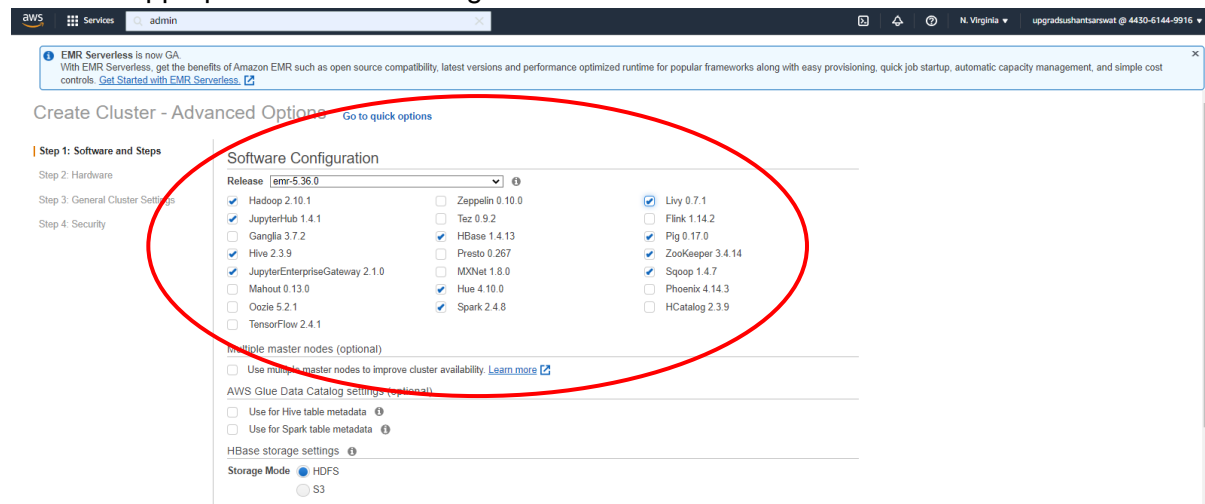
Instance type: m5.xlarge

Number of instances: 3 (1 master and 2 core nodes)

Cluster scaling: ☐ scale cluster nodes based on workload

Auto-termination: ☒ Enable auto-termination

2. Select appropriate software configuration



The screenshot shows the AWS EMR console 'Create Cluster - Advanced Options' page. A red circle highlights the 'Software Configuration' section. The page includes a sidebar with steps and a main configuration area.

Step 1: Software and Steps

Software Configuration

Release: emr-5.36.0

Applications:

- ☒ Hadoop 2.10.1
- ☒ JupyterHub 1.4.1
- ☐ Ganglia 3.7.2
- ☒ Hive 2.3.9
- ☒ JupyterEnterpriseGateway 2.1.0
- ☐ Mahout 0.13.0
- ☐ Oozie 5.2.1
- ☐ TensorFlow 2.4.1
- ☐ Zeppelin 0.10.0
- ☐ Tez 0.9.2
- ☒ HBase 1.4.13
- ☐ Presto 0.267
- ☐ MXNet 1.8.0
- ☒ Hue 4.10.0
- ☒ Spark 2.4.8
- ☒ Livy 0.7.1
- ☐ Flink 1.14.2
- ☐ Pig 0.17.0
- ☒ ZooKeeper 3.4.14
- ☒ Sqoop 1.4.7
- ☐ Phoenix 4.14.3
- ☐ HCatalog 2.3.9

Multiple master nodes (optional)

☐ Use multiple master nodes to improve cluster availability. [Learn more](#)

AWS Glue Data Catalog settings (optional)

☐ Use for Hive table metadata

☐ Use for Spark table metadata

HBase storage settings

Storage Mode: ☒ HDFS ☐ S3

3. Machine Selection

Instance types

<input type="radio"/> m3.2xlarge	8	30	160 SSD
<input type="radio"/> m4.large	2	8	EBS only
<input checked="" type="radio"/> m4.xlarge	4	16	EBS only
<input type="radio"/> m4.2xlarge	8	32	EBS only
<input type="radio"/> m4.4xlarge	16	64	EBS only
<input type="radio"/> m4.10xlarge	40	160	EBS only
<input type="radio"/> m4.16xlarge	64	256	EBS only
<input type="radio"/> m5.xlarge	4	16	EBS only
<input type="radio"/> m5.2xlarge	8	32	EBS only
<input type="radio"/> m5.4xlarge	16	64	EBS only
<input type="radio"/> m5.8xlarge	32	128	EBS only

Cancel

Save

4. Cluster Nodes and instances

Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. [Learn more about instance purchasing options](#)

Console options for automatic scaling have changed. [Learn more](#)

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m4.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB Add configuration settings	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Core Core - 2	m4.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB Add configuration settings	2 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	0 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps

Step 2: Hardware

Step 3: General Cluster Settings

Step 4: Security

General Options

Cluster name upgrad_capstone

☒ Logging

S3 folder s3://aws-logs-443061449916-us-east-1/elasticmap

☐ Log encryption

☒ Debugging

☐ Termination protection

5. Security Options

Create Cluster - Advanced Options

[Go to quick options](#)

[Step 1: Software and Steps](#)

[Step 2: Hardware](#)

[Step 3: General Cluster Settings](#)

Step 4: Security

Security Options

EC2 key pair **RHEL1** ⓘ

☒ Cluster visible to all IAM users in account ⓘ

Permissions ⓘ

☒ Default ☐ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role [EMR_DefaultRole](#) ☐ Use EMR_DefaultRole_V2 ⓘ

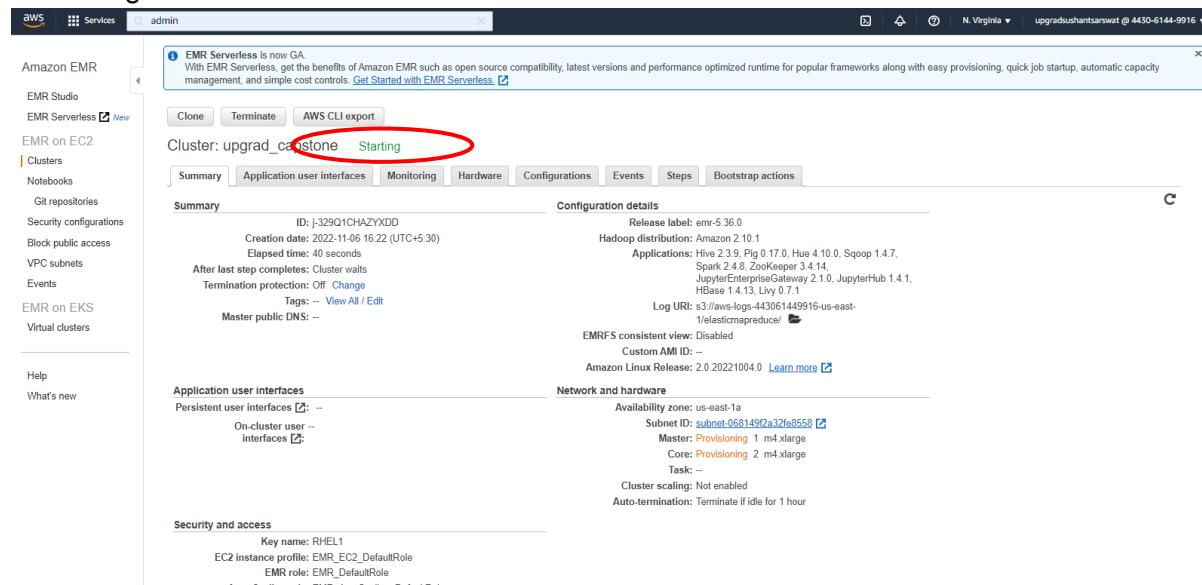
EC2 instance profile [EMR_EC2_DefaultRole](#) ⓘ

Auto Scaling role [EMR_AutoScaling_DefaultRole](#) ⓘ

▶ Security Configuration

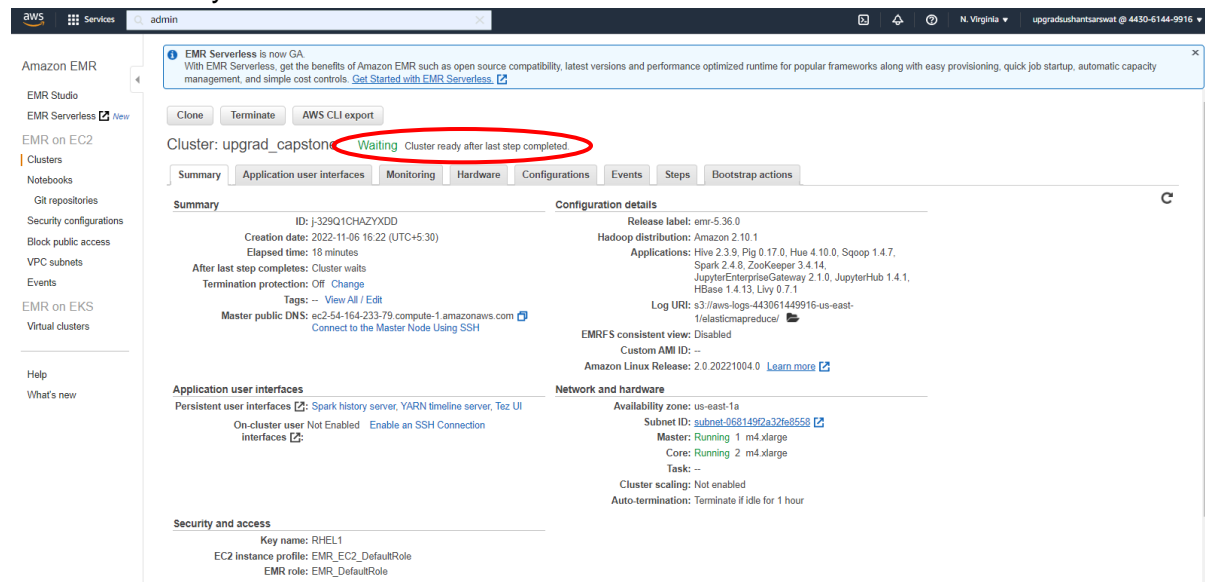
▶ EC2 security groups

6. Starting the cluster



The screenshot shows the AWS Management Console interface for an Amazon EMR cluster. The cluster name is 'upgrad_capstone' and its status is 'Starting', which is circled in red. The console displays various tabs for the cluster, including Summary, Application user interfaces, Monitoring, Hardware, Configurations, Events, Steps, and Bootstrap actions. The Summary tab is selected, showing details such as the cluster ID (j-329Q1CHAZYXDD), creation date (2022-11-06 16:22 UTC+5:30), and elapsed time (40 seconds). The Configuration details section shows the release label (emr-5.36.0), Hadoop distribution (Amazon 2.10.1), and applications (Hive 2.3.9, Pig 0.17.0, Hue 4.10.0, Sqoop 1.4.7, Spark 2.4.8, ZooKeeper 3.4.14, JupyterEnterpriseGateway 2.1.0, JupyterHub 1.4.1, HBase 1.4.13, Livy 0.7.1). The Network and hardware section shows the availability zone (us-east-1a), subnet ID (subnet-0681492a32f68558), and master node configuration (Provisioning 1 m4.xlarge, Core: Provisioning 2 m4.xlarge). The Security and access section shows the key name (RHEL1), EC2 instance profile (EMR_EC2_DefaultRole), and EMR role (EMR_DefaultRole).

7. Cluster Ready



8. STATEMENT TO BUILD AN EXTERNAL HIVE TABLE FOR THRESHOLD DATA AND VIEW THRESHOLD DATA

(a) Create a text file in HDFS to load threshold values data by logging into Hadoop and opening a text file by using `vi threshold.txt` command

```
[hadoop@ip-172-31-46-73 ~]$ vi threshold.txt
```

(b) Insert the following lines in the threshold.txt file

```
1|heartBeat|0|40|0|69|1|LowerHeartRatethanNormal|
2|heartBeat|0|40|70|78|0|Normal|
3|heartBeat|0|40|79|9999|1|HigherHeartRatethanNormal|
4|bp|0|40|0|160|1|LowerBPthanNormal|
5|bp|0|40|161|220|0|Normal|
6|bp|0|40|221|9999|1|HigherBPthanNormal|
7|heartBeat|41|100|0|65|1|LowHeartRatethanNormal|
8|heartBeat|41|100|66|73|0|Normal|
9|heartBeat|41|100|74|9999|1|HigherHeartRatethanNormal|
10|bp|41|100|0|150|1|LowBPthanNormal|
11|bp|41|100|151|180|0|Normal|
12|bp|41|100|181|9999|1|HigherBPthanNormal|
```

Save the file by clicking **escape** key and save the file using `:wq!`

```
1|heartBeat|0|40|0|69|1|LowerHeartRatethanNormal
2|heartBeat|0|40|70|78|0|Normal
3|heartBeat|0|40|79|9999|1|HigherHeartRatethanNormal
4|bp|0|40|0|160|1|LowBPthanNormal
5|bp|0|40|161|220|0|Normal
6|bp|0|40|221|9999|1|HigherBPthanNormal
7|heartBeat|41|100|0|65|1|LowHeartRatethanNormal
8|heartBeat|41|100|66|73|0|Normal
9|heartBeat|41|100|74|9999|1|HigherHeartRatethanNormal
10|bp|41|100|0|150|1|LowBPthanNormal
11|bp|41|100|151|180|0|Normal
12|bp|41|100|181|9999|1|HigherBPthanNormal

:wq!
```

(c) Log into hive

```
[hadoop@ip-172-31-46-73 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: false
hive>
```

(d) Create database using command **create database if not exists Patient_db;**

```
hive> Create database if not exists Patient_db;  
OK  
Time taken: 0.845 seconds
```

(e) Use database by using command `use Patient_db;`

```
hive> Use Patient_db;
OK
Time taken: 0.078 seconds
```

(f) Create external hive table for threshold values using the following command

```
create external table if not exists ThresholdValueTable(
Key int,
Attribute varchar(20),
Low_Age_Limit int,
High_Age_Limit int,
Low_Range_Value int,
High_Range_Value int,
```

Alert_Flag int,
Alert_Message varchar(255))
row format delimited fields terminated by '|'
lines terminated by '\n' stored as textfile;

```
hive> create external table if not exists ThresholdValueTable(
  > Key int,
  > Attribute varchar(20),
  > Low_Age_Limit int,
  > High_Age_Limit int,
  > Low_Range_Value int,
  > High_Range_Value int,
  > Alert_Flag int,
  > Alert_Message varchar(255))
  > row format delimited fields terminated by '|'
  > lines terminated by '\n' stored as textfile;
OK
Time taken: 0.783 seconds
```

(g) Load data from threshold.txt into the table ThresholdValueTable using the following command

load data local inpath 'threshold.txt' into table ThresholdValueTable;

```
hive> load data local inpath 'threshold.txt' into table ThresholdValueTable;
Loading data to table patient_db.thresholdvaluetable
OK
Time taken: 1.195 seconds
```

(h) Verify that the values have been loaded properly into the table by using the following command

select * from thresholdvaluetable;

```
hive> select * from thresholdvaluetable;
OK
1      heartBeat      0      40      0      69      1      LowerHeartRatethanNormal
2      heartBeat      0      40      70      78      0      Normal
3      heartBeat      0      40      79      9999     1      HigherHeartRatethanNormal
4      bp      0      40      0      160     1      LowBPthanNormal
5      bp      0      40      161     220     0      Normal
6      bp      0      40      221     9999     1      HigherBPthanNormal
7      heartBeat      41     100     0      65      1      LowHeartRatethanNormal
8      heartBeat      41     100     66      73      0      Normal
9      heartBeat      41     100     74      9999     1      HigherHeartRatethanNormal
10     bp      41     100     0      150     1      LowBPthanNormal
11     bp      41     100     151     180     0      Normal
12     bp      41     100     181     9999     1      HigherBPthanNormal
Time taken: 0.216 seconds, Fetched: 12 row(s)
```