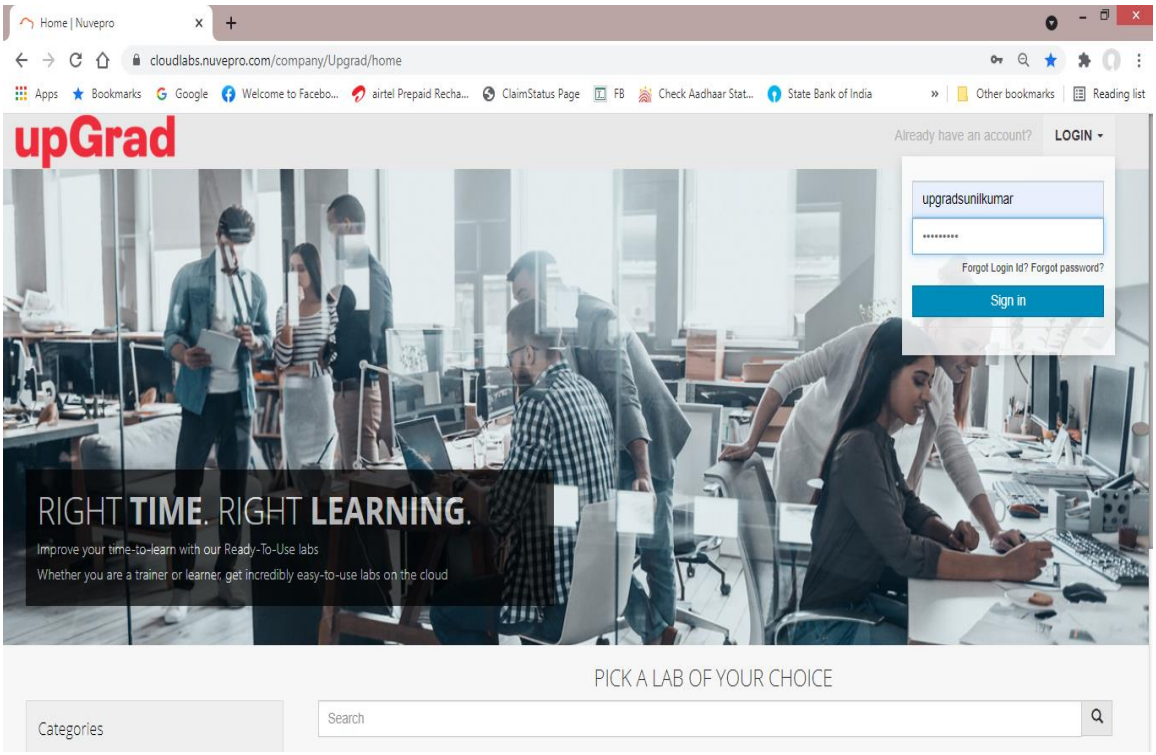
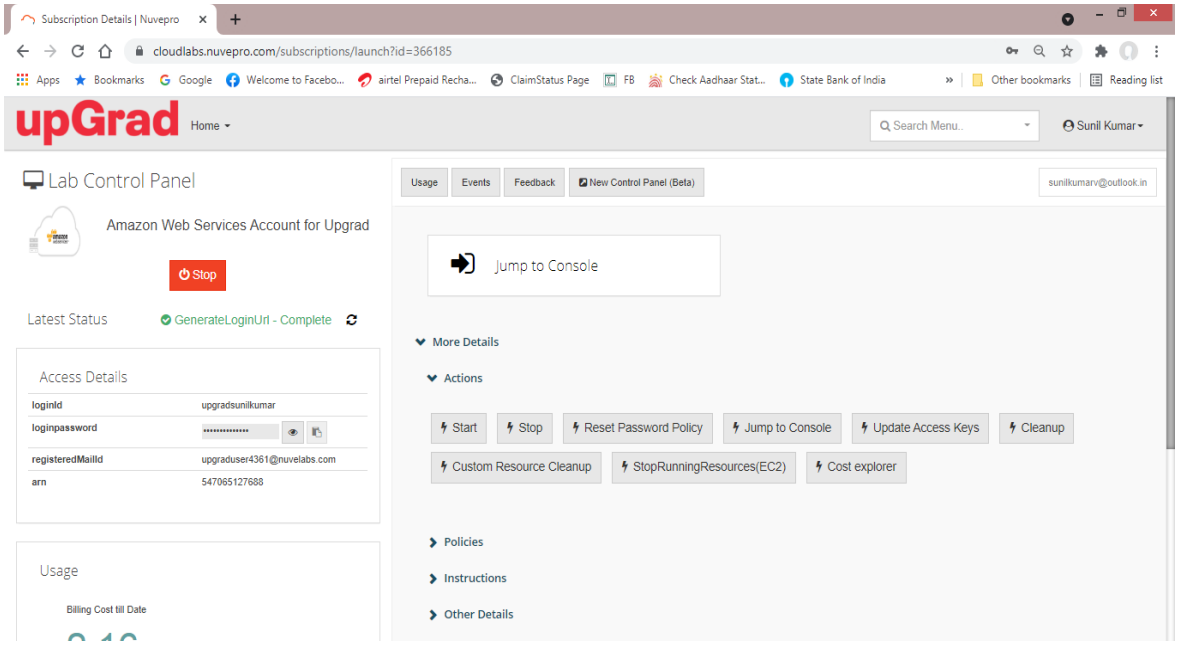
Hive Case Study:

Logged into Amazon Web Services using the following link

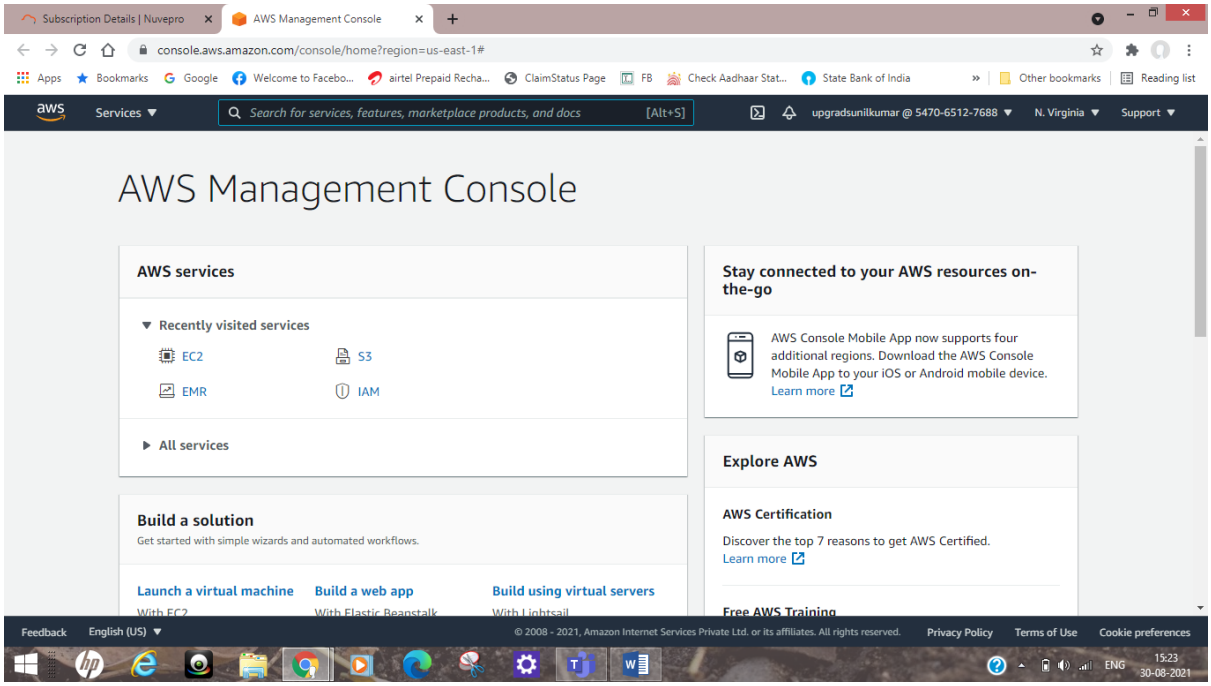
https://cloudlabs.nuvepro.com/company/Upgrad/home



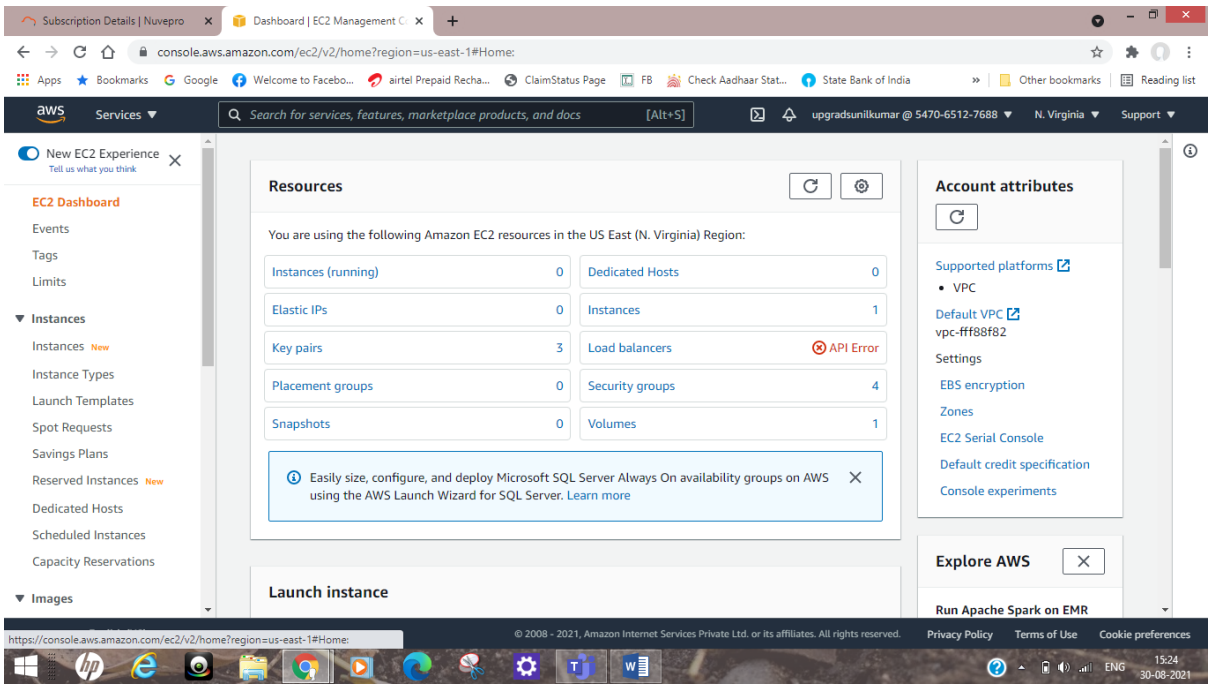
After Login: Clicked on Jump to console



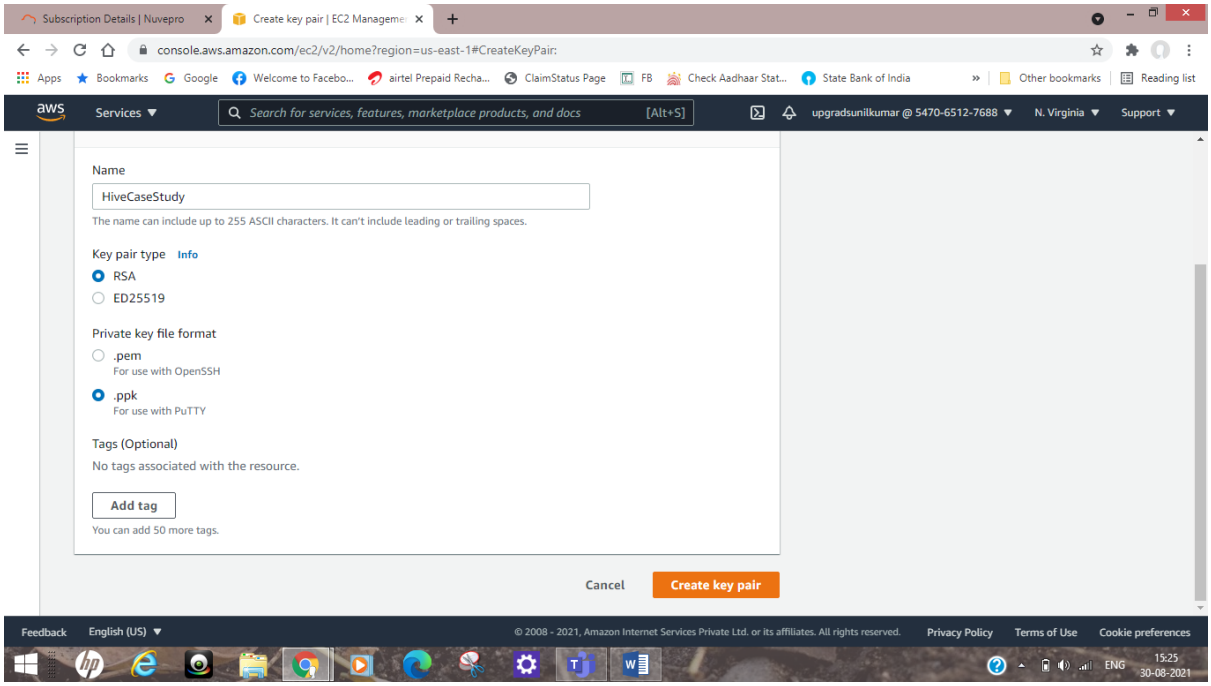
We have entered the Aws Console, First we will go to the EC2 – Elastic Compute cloud service to generate new Key Pair



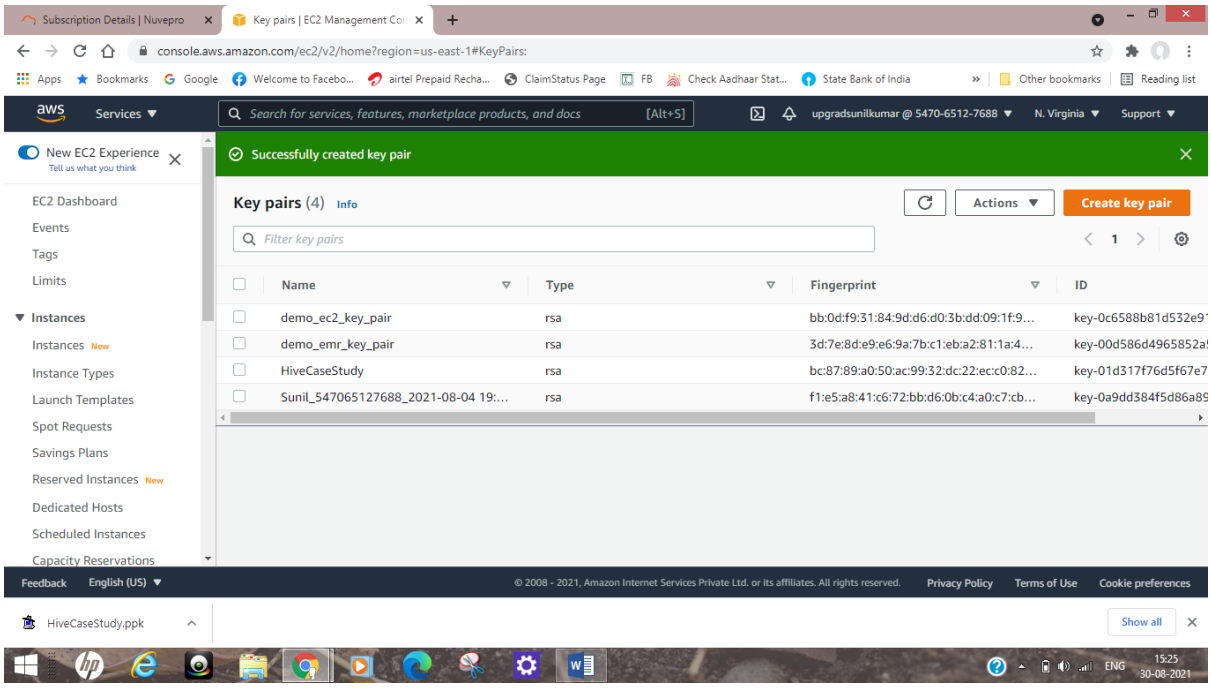
Click on the Key Pairs to generate new key Pair



Creating Key Pair with ppk file format to use directly with putty

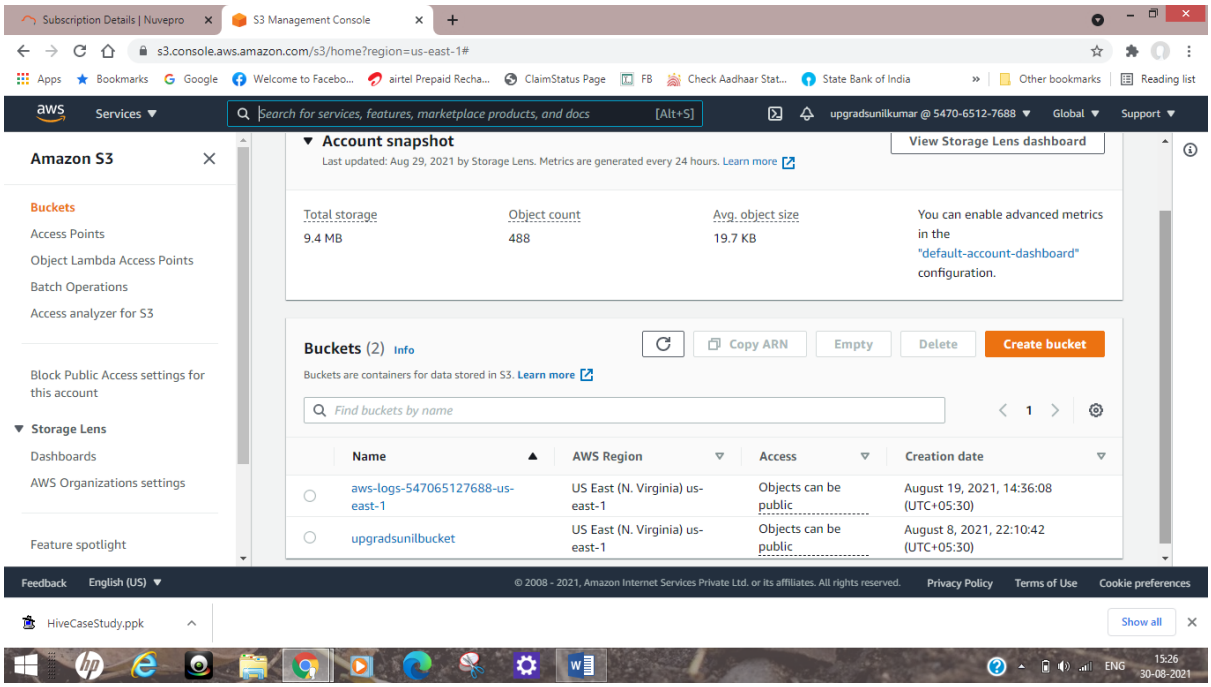


HiveCaseStudy- Key Pair successfully created and downloaded. It can be used to connect with putty.

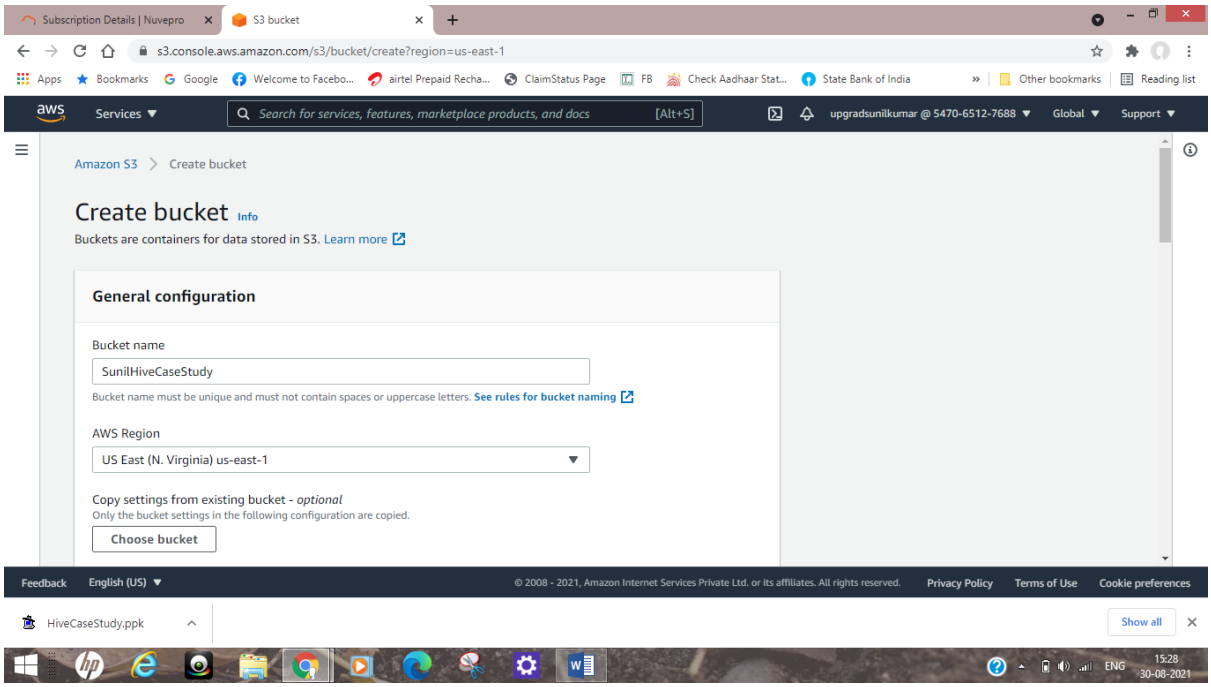


S3 BUCKET

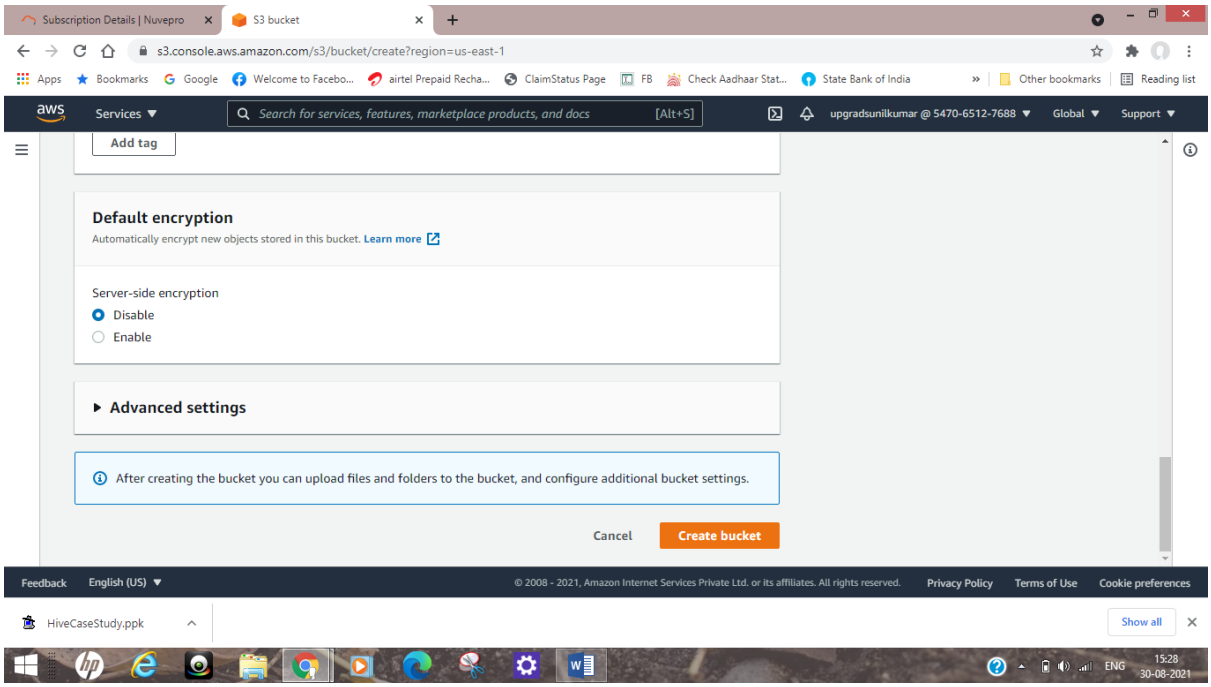
Click on services and enter S3 – Simple Storage Service



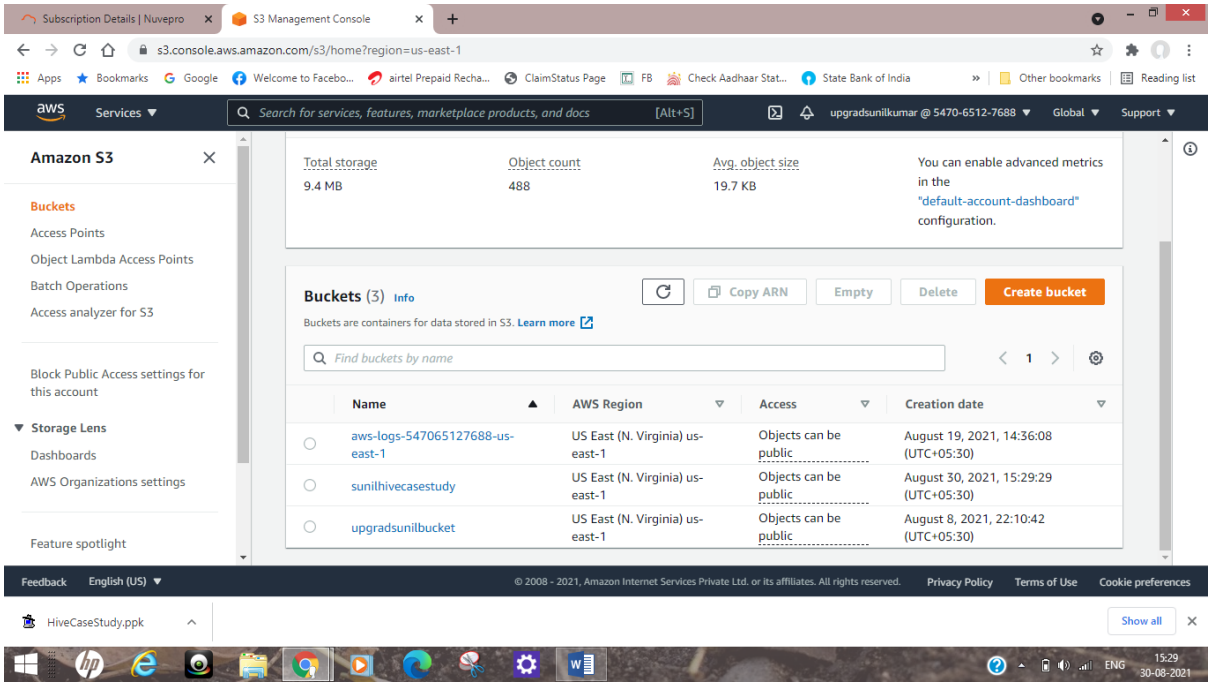
To Store the data in S3 – Click on “Create Bucket”, given Global unique name to the bucket



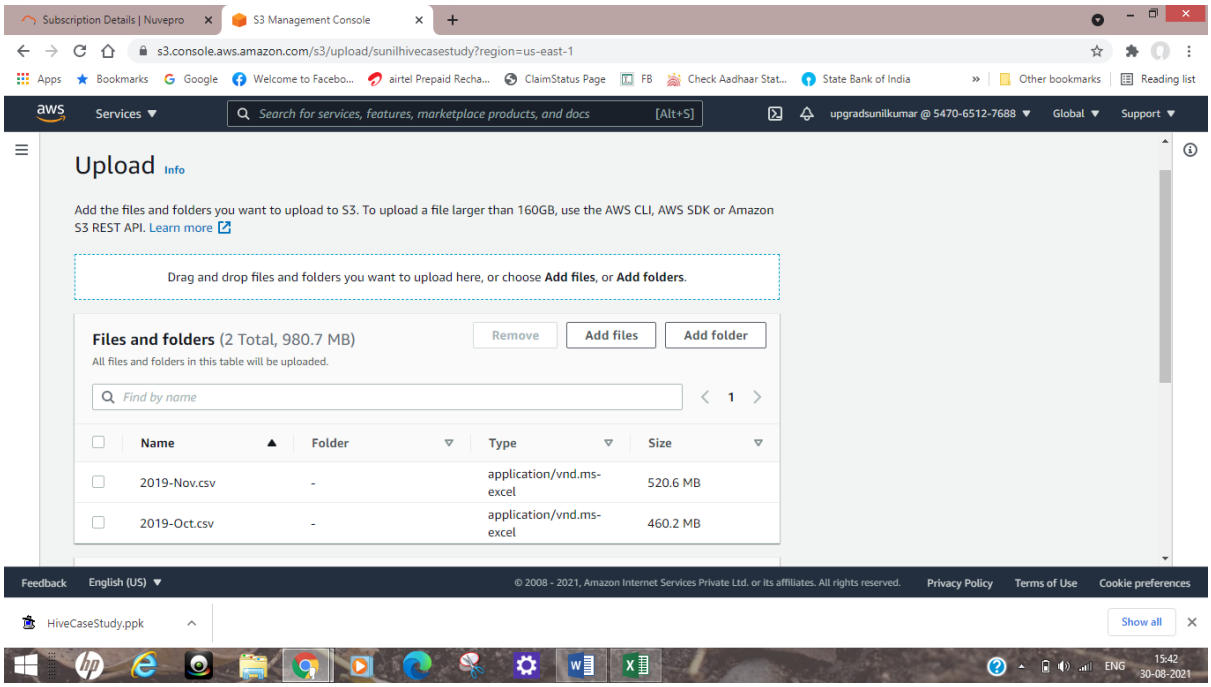
Keeping the Default settings and allowed the public access to the bucket.



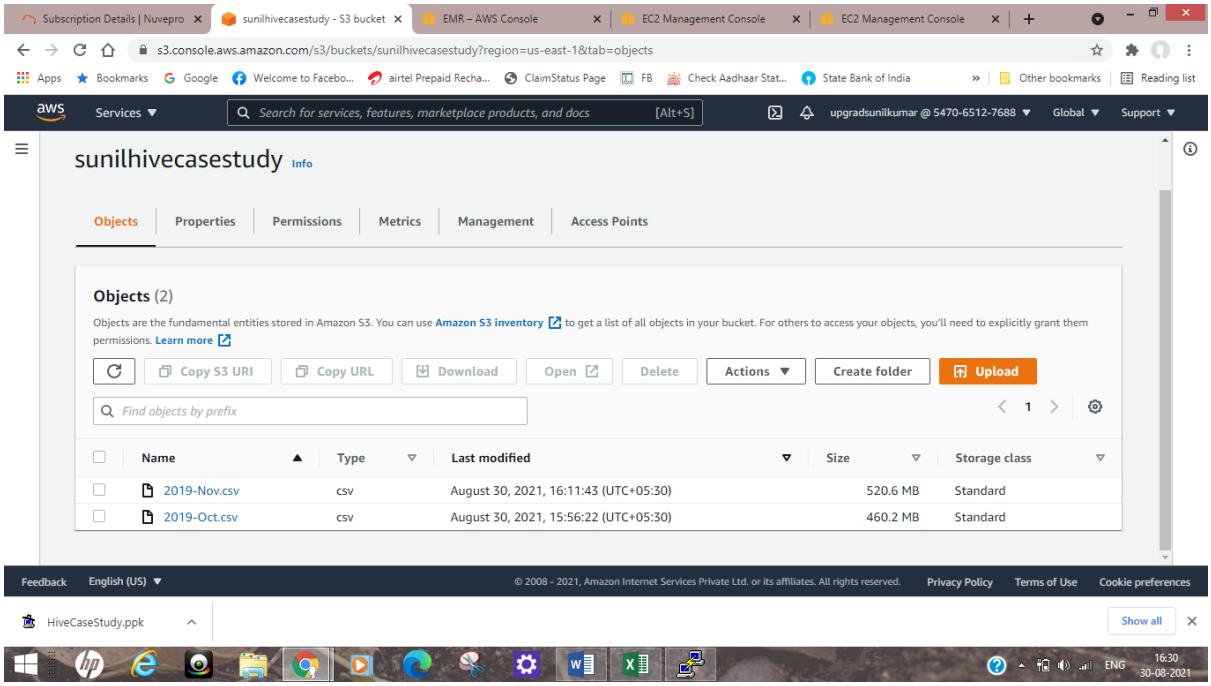
Successfully created the sunilhivecasestudy bucket. Now it's ready for data upload



Uploading the 2019 October and 2019 November csv files to S3 bucket

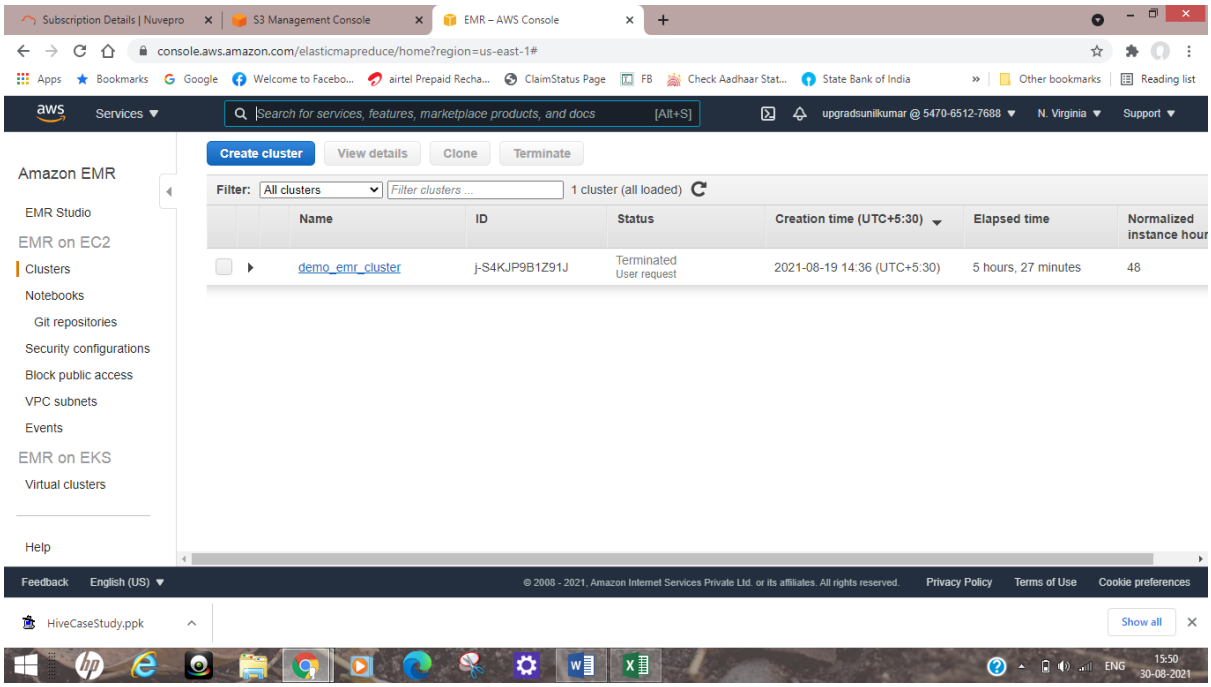


Successfully Uploaded the Files and now ready to move to hdfs.



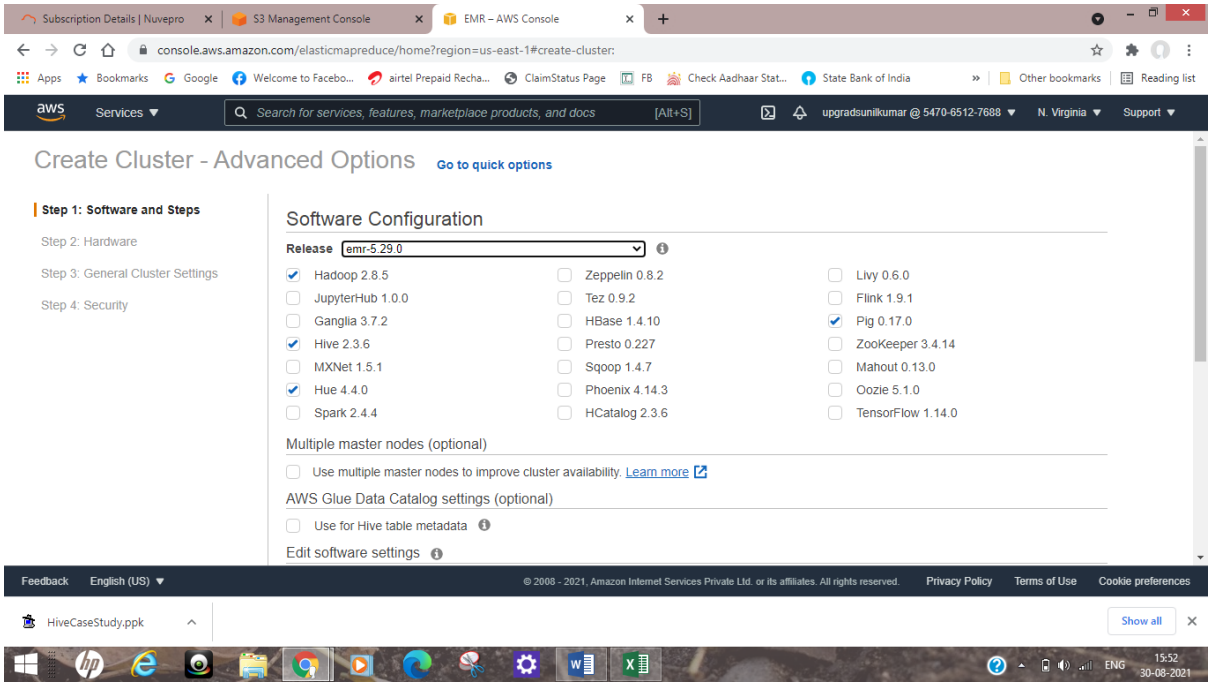
EMR CLUSTER CREATION:

Click on AWS Services and select EMR – Elastic Map Reduce

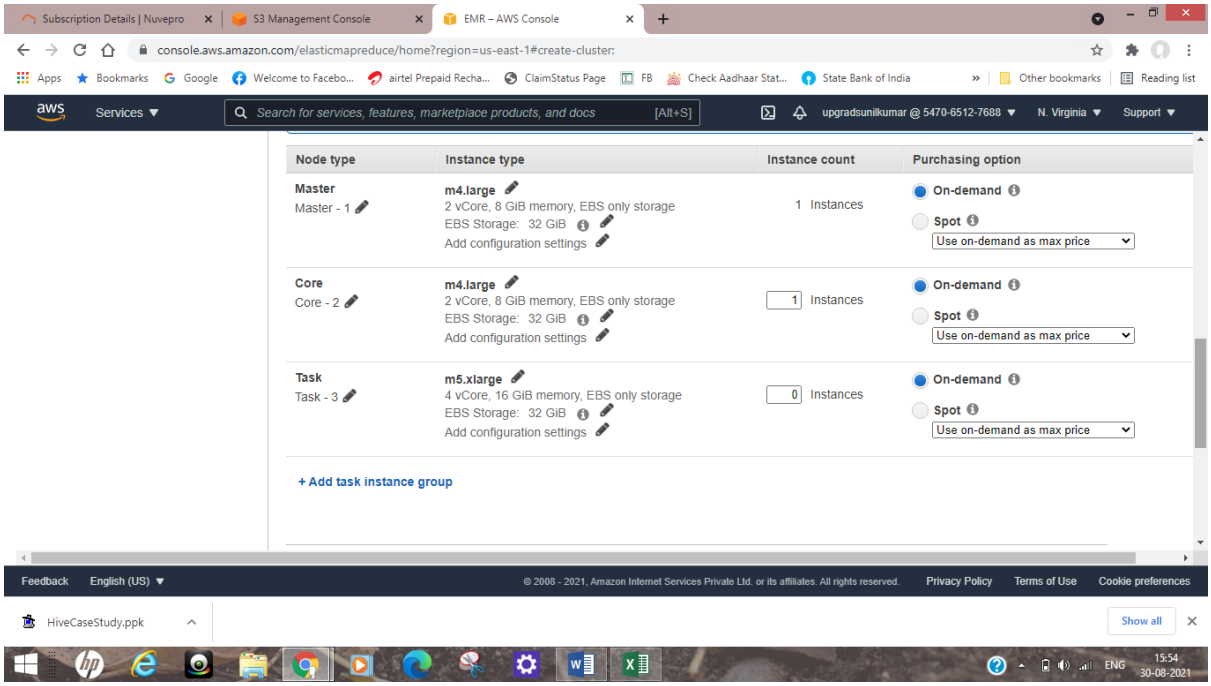


Click on “Create cluster” button to create the EMR cluster and select advances options.

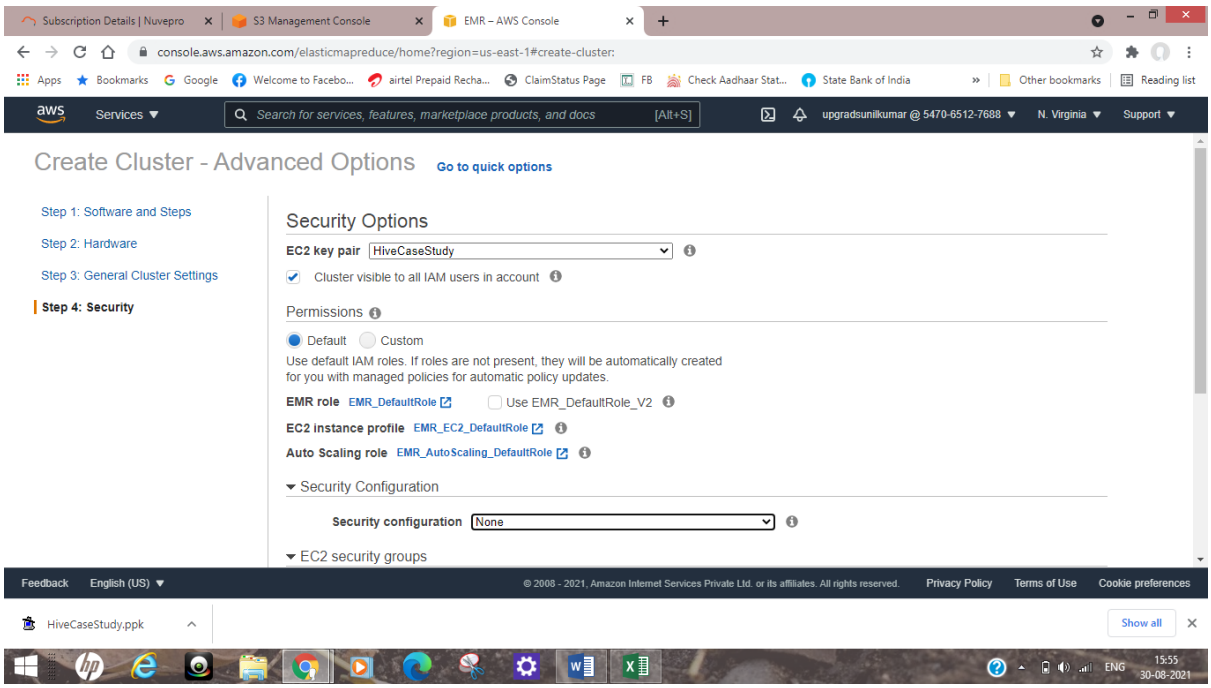
Selected emr 5.29.0 release for this case study



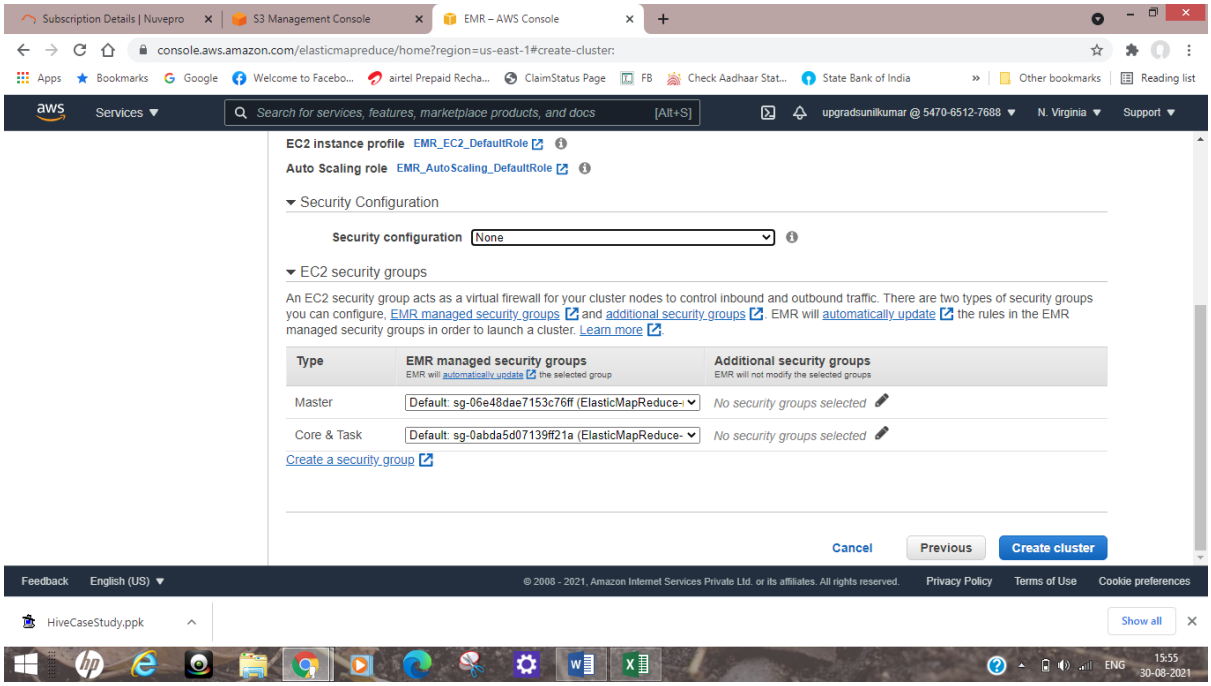
We are using 2 node cluster with m4.large for Master node and M4.large for Core node.



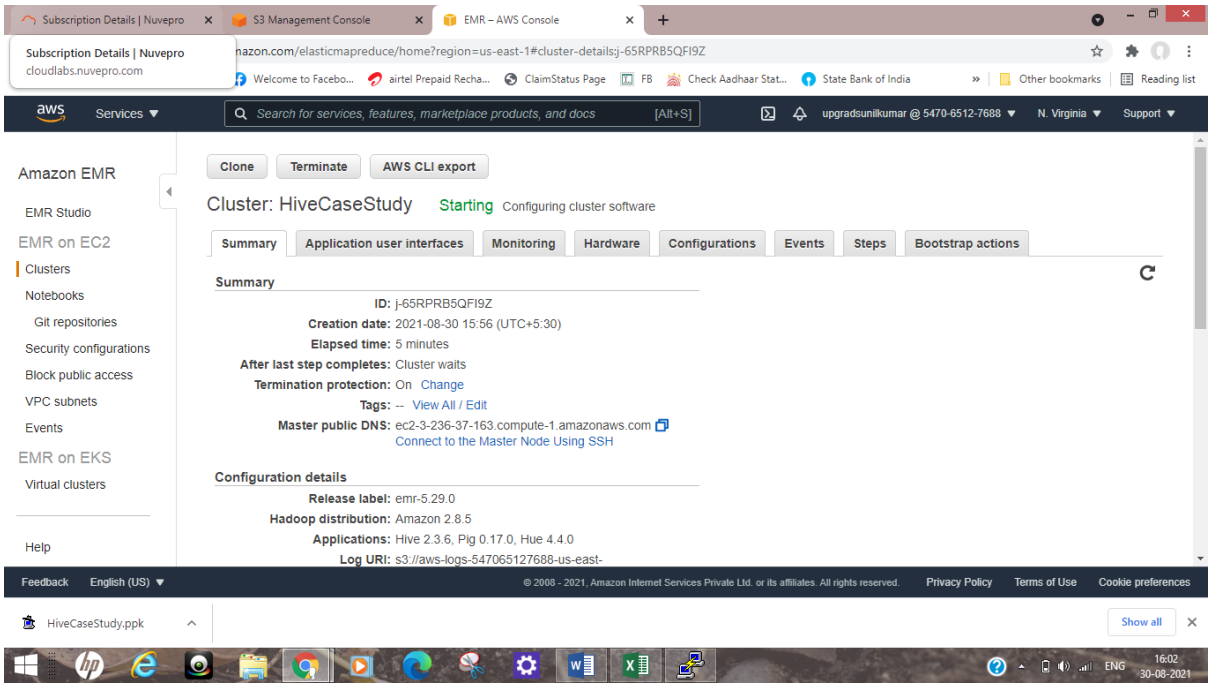
We will be using the HiveCaseStudy Key Pair which we created for connecting the cluster.



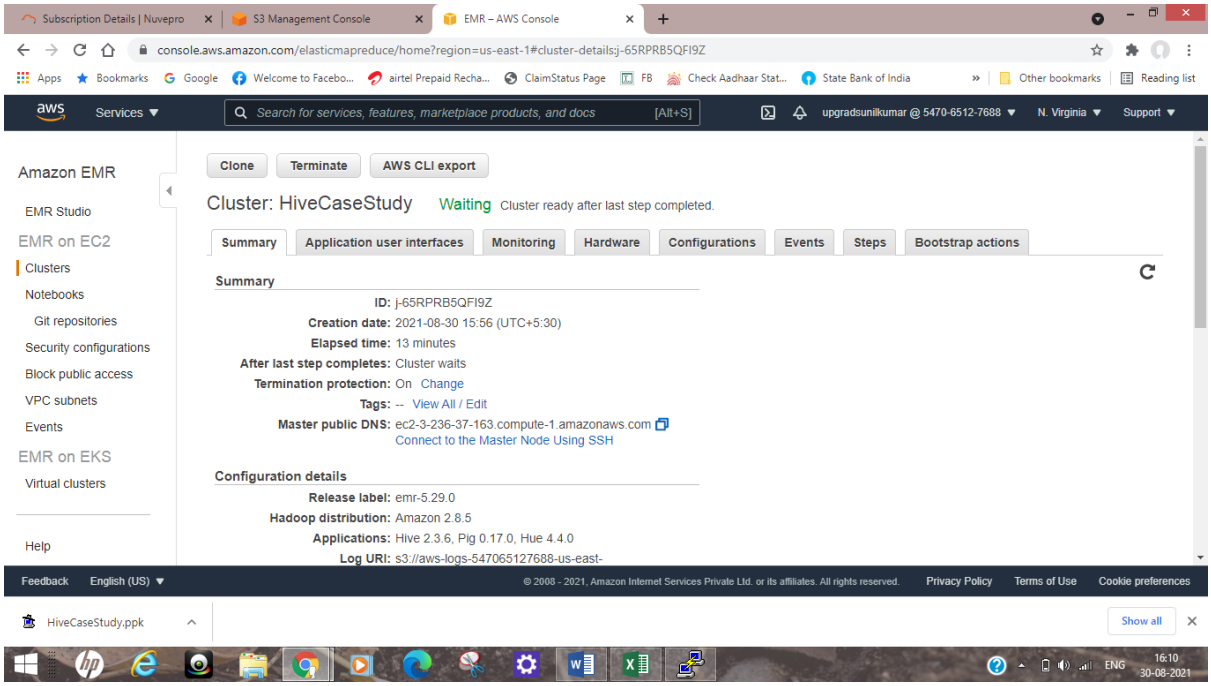
After selecting all the specifications. Click on Create Cluster



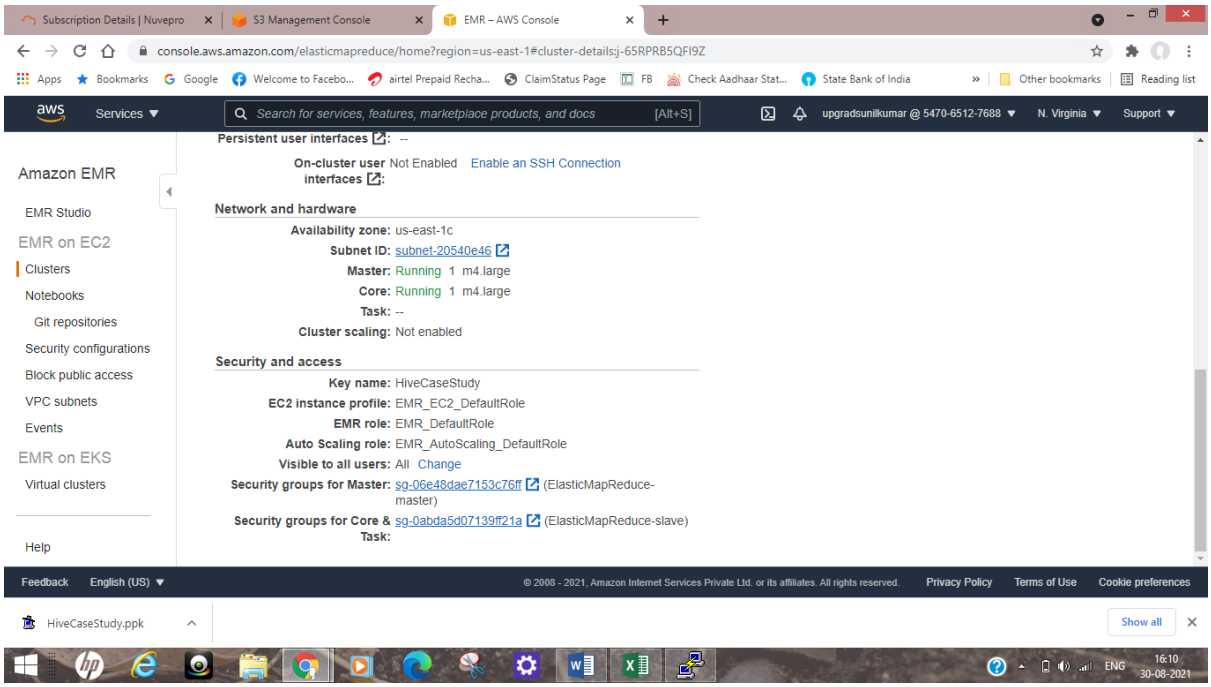
EMR Cluster Creation has started.



Cluster is ready and status has changed from starting to “Waiting”

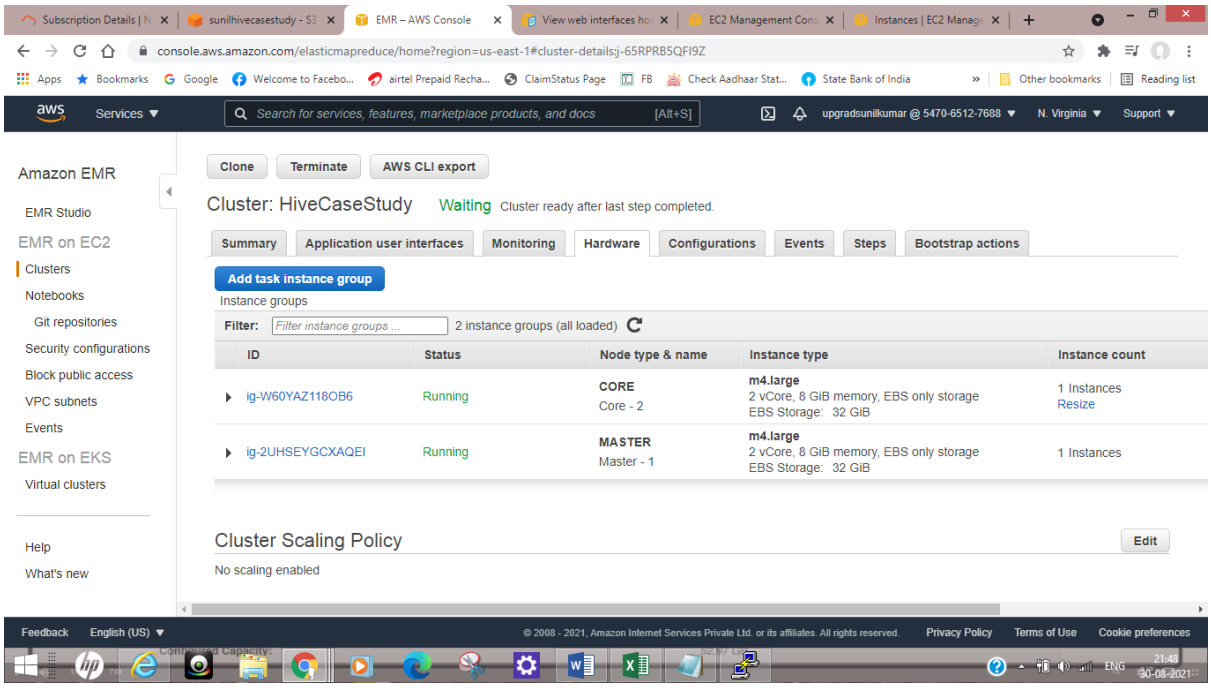


Master and Core nodes are also in running status.

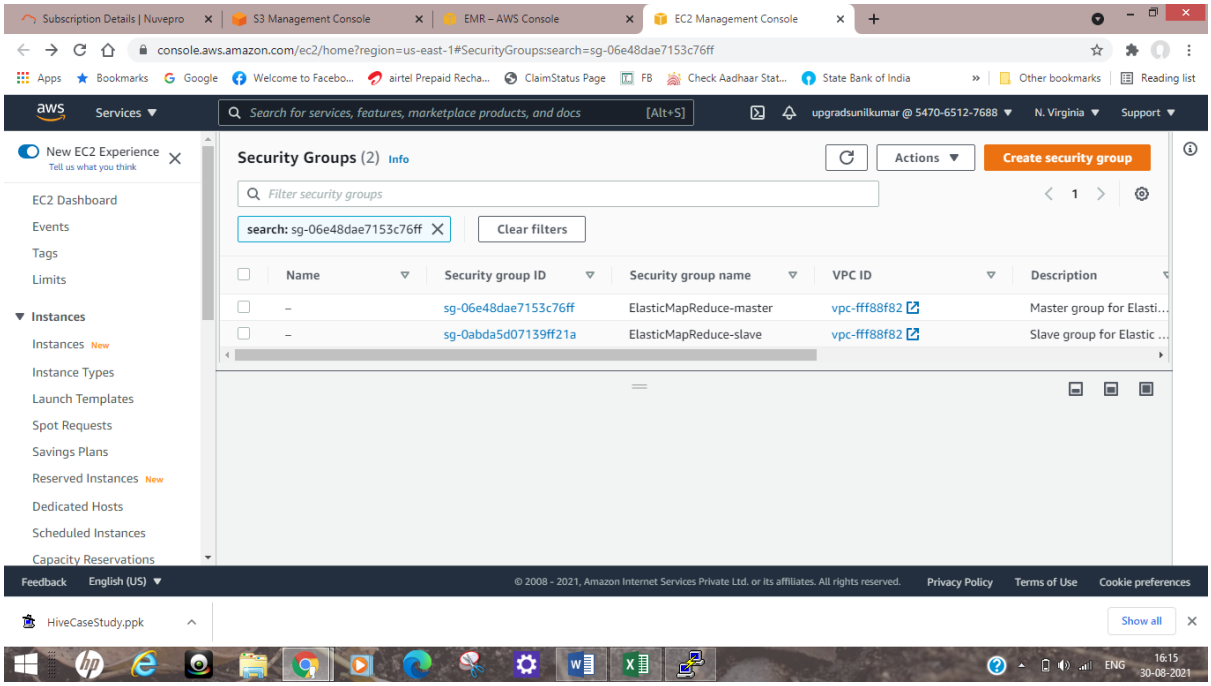


We can check the status of the nodes in the hardware section.

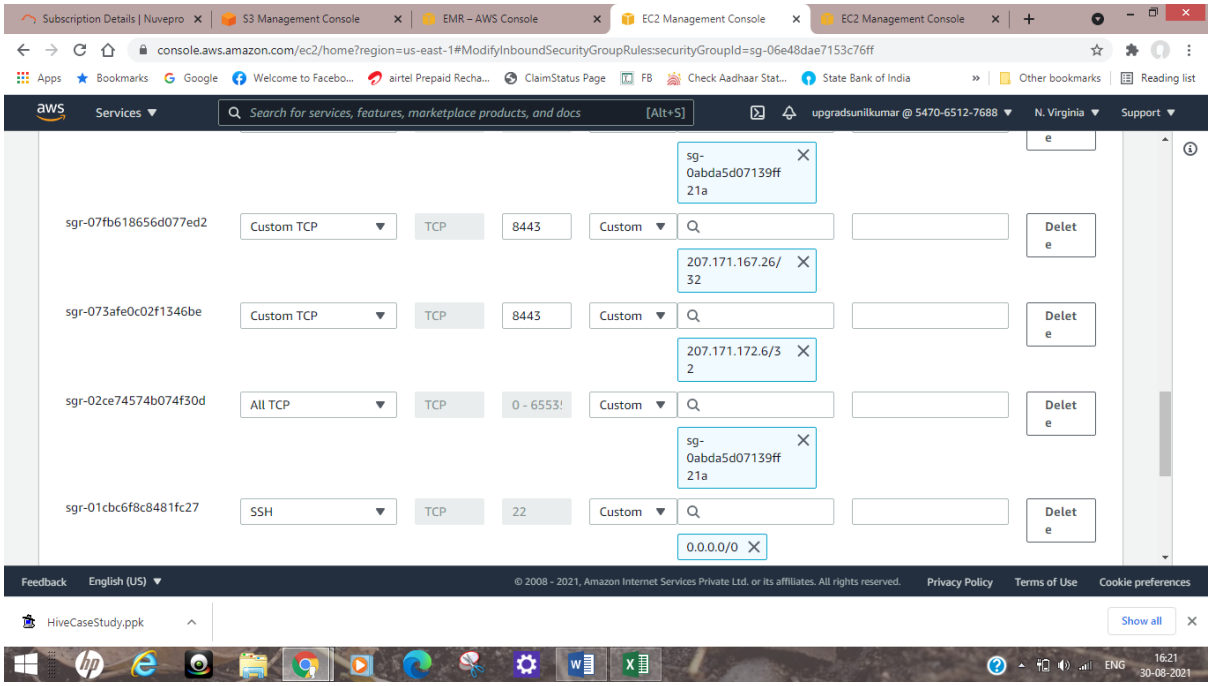
Master and Core node both are running.



We have to edit the Security Group of Master Node

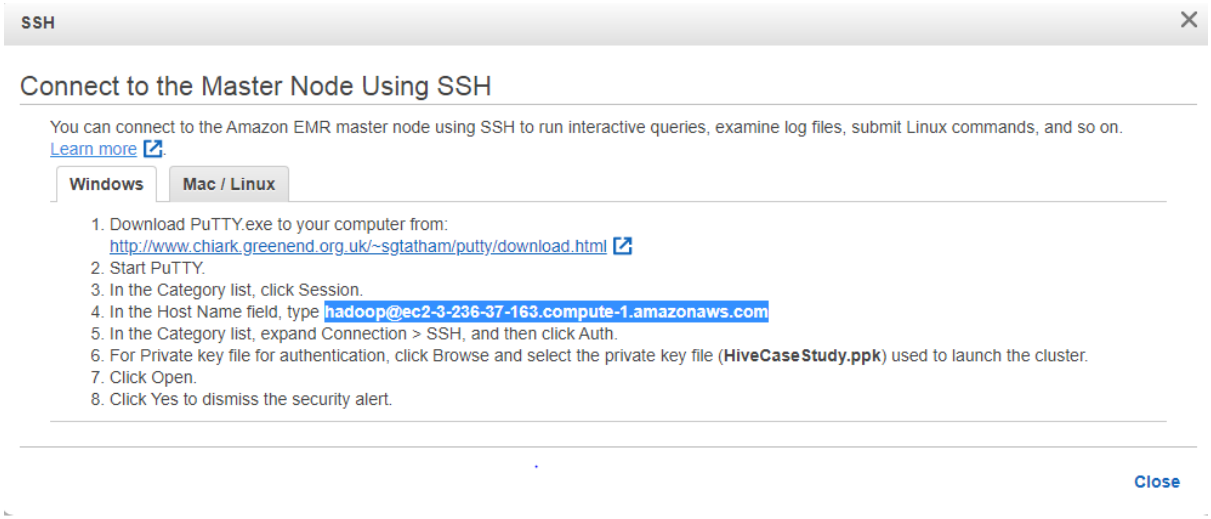


Then save the SSH rule to anywhere in the inbound rules

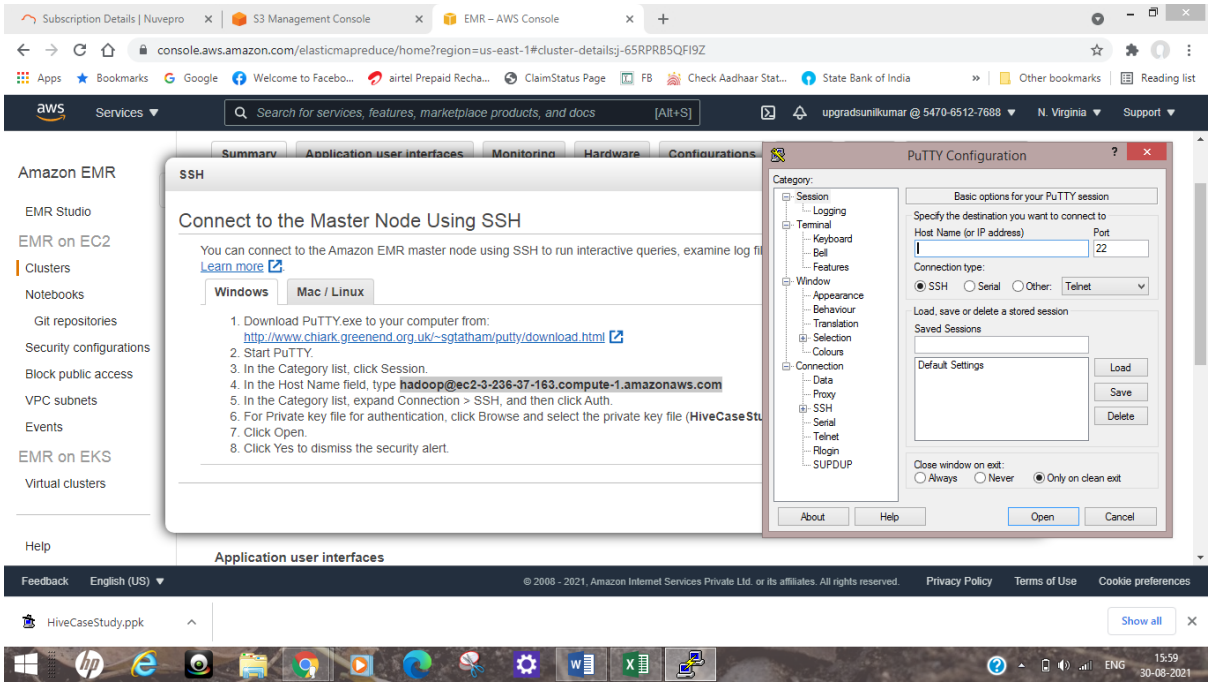


CONNECTING TO MASTER NODE:

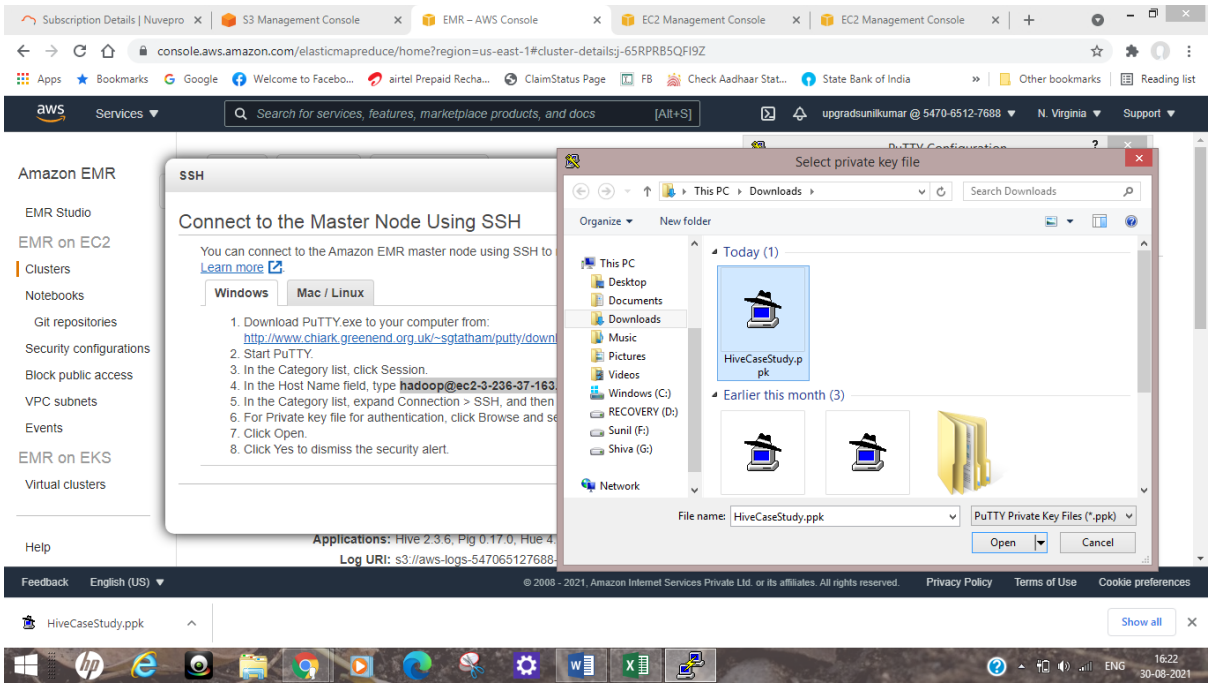
Host: hadoop@ec2-3-236-37-163.compute-1.amazonaws.com



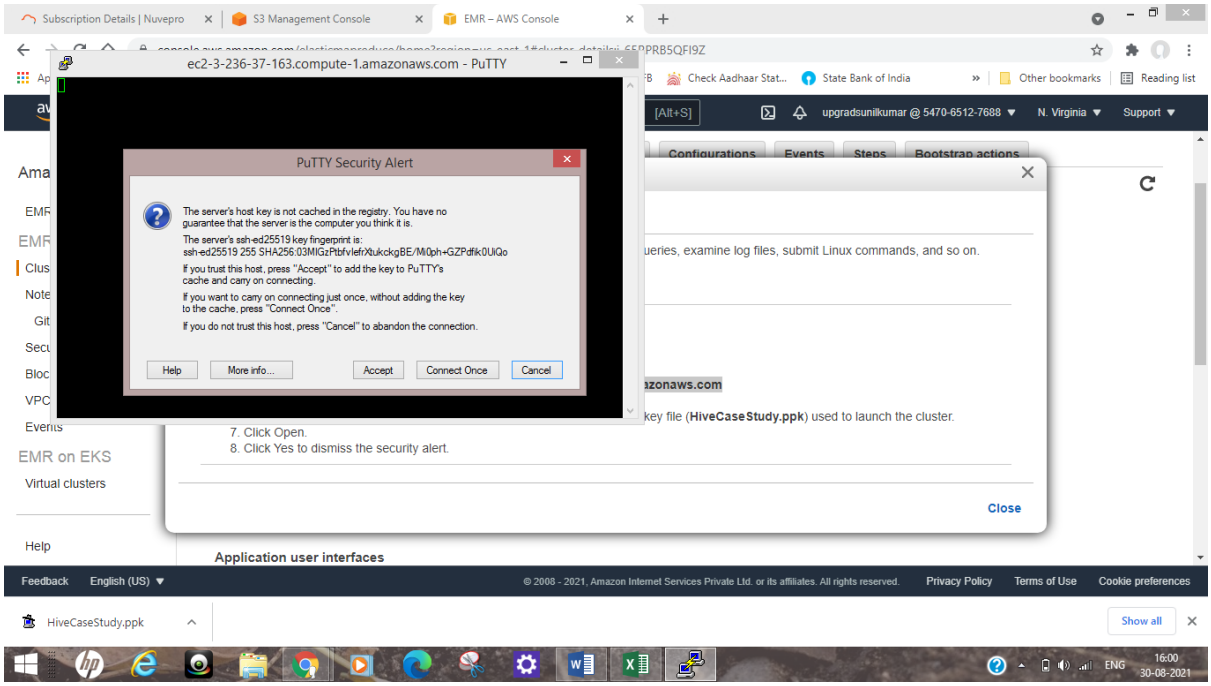
Open the putty and enter the Host Name and click on SSH→ Auth



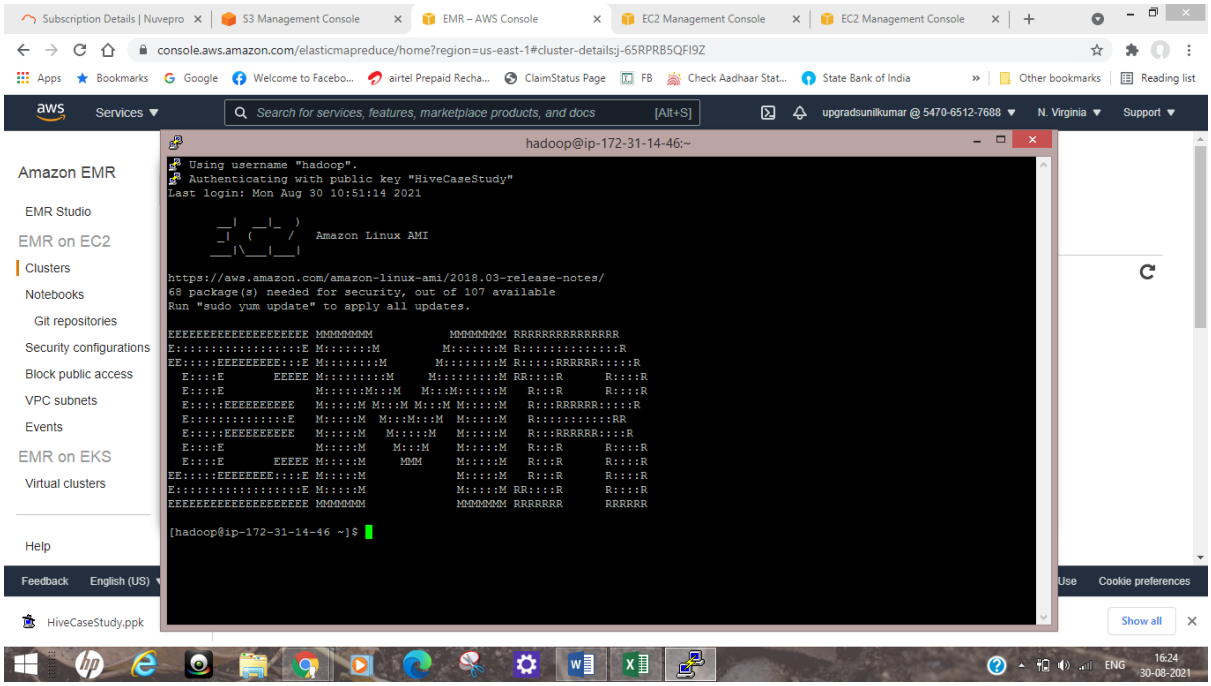
Browse and select the private key HiveCaseStudy.ppk file



Click on Open and accept the connection

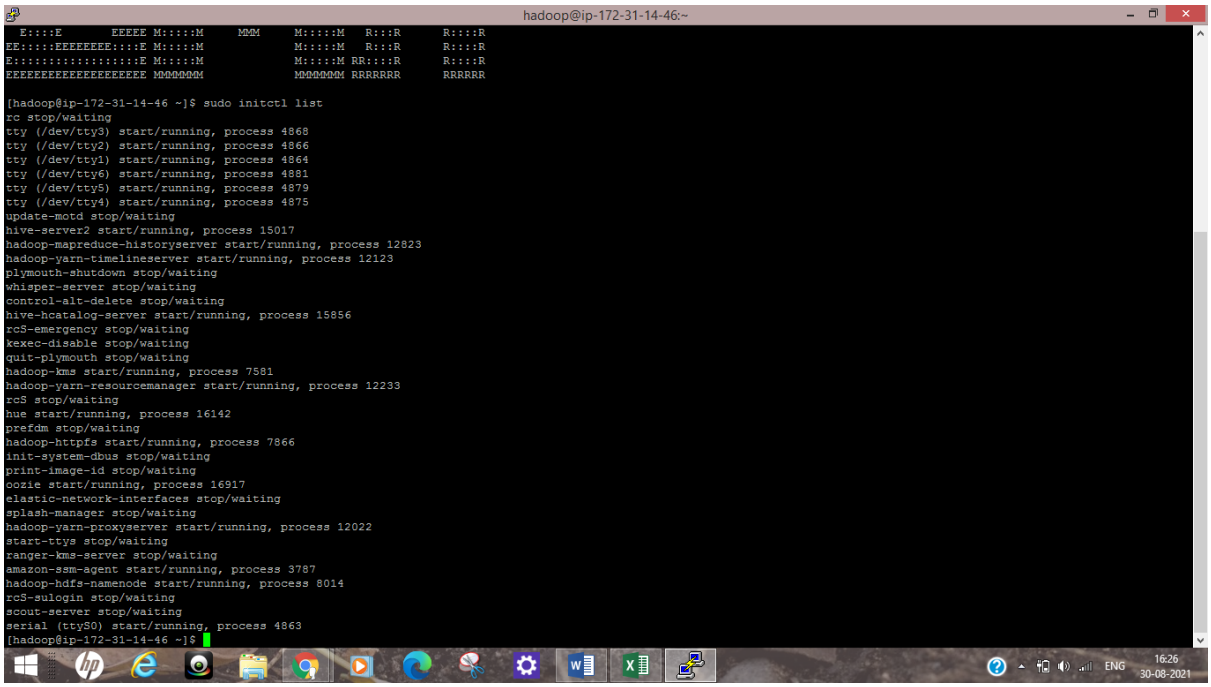


EMR – Elastic Map Reduce CLI is launched successfully.

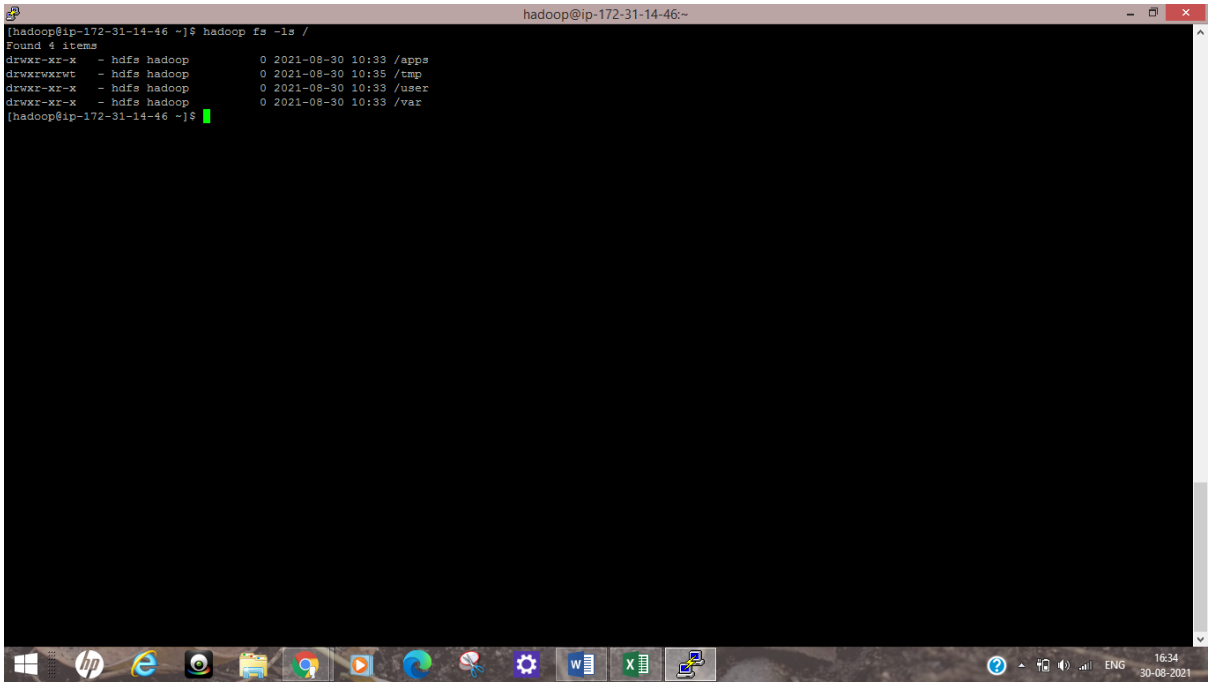


Enter “sudo initctl list” to check the services running in the EMR.

Hive services are running.



Verifying the Hadoop file system with command **“hadoop fs -ls /”**



CREATING A DIRECTORY FOR CASE STUDY:

Creating a new directory under user>hive for Hive case study and to store the data files

hadoop fs -mkdir /user/hive/hivecasestudy

We have also verified the directory using the command

hadoop fs -ls /user/hive/

New directory is successfully created

LOADING DATA into HDFS:

S3 Path for data files:

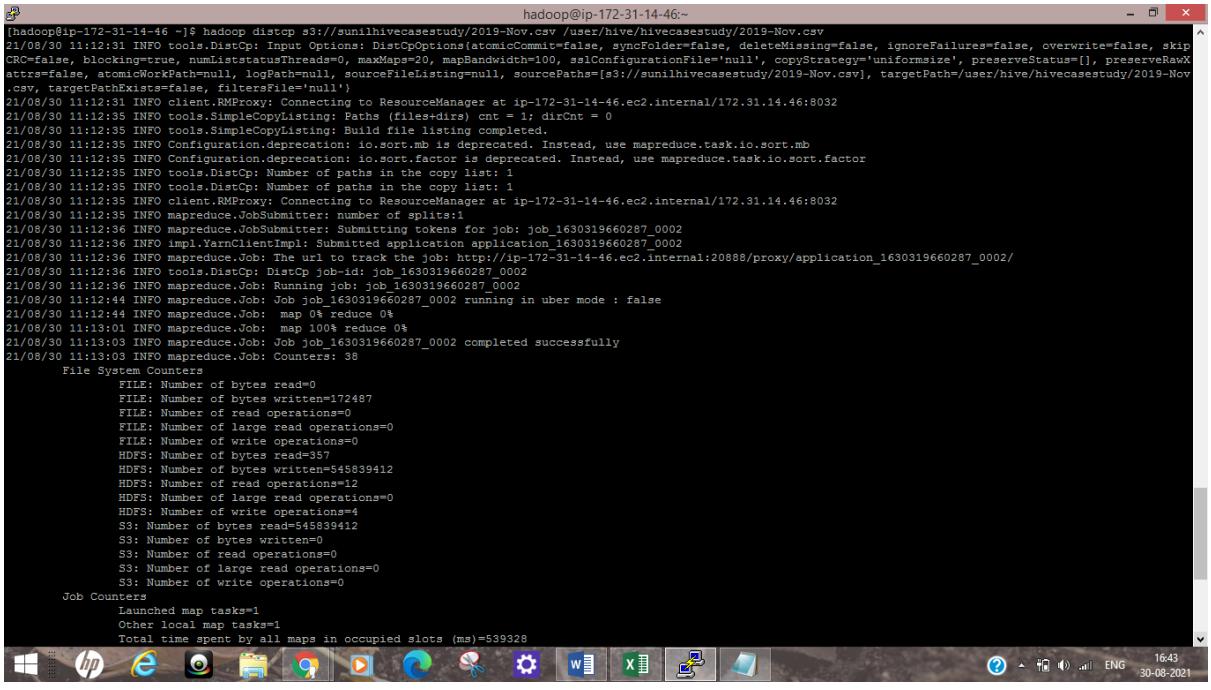
s3://sunilhivecasestudy/2019-Nov.csv

s3://sunilhivecasestudy/2019-Oct.csv

Copying the data from S3 to HDFS Using distributed copy command:

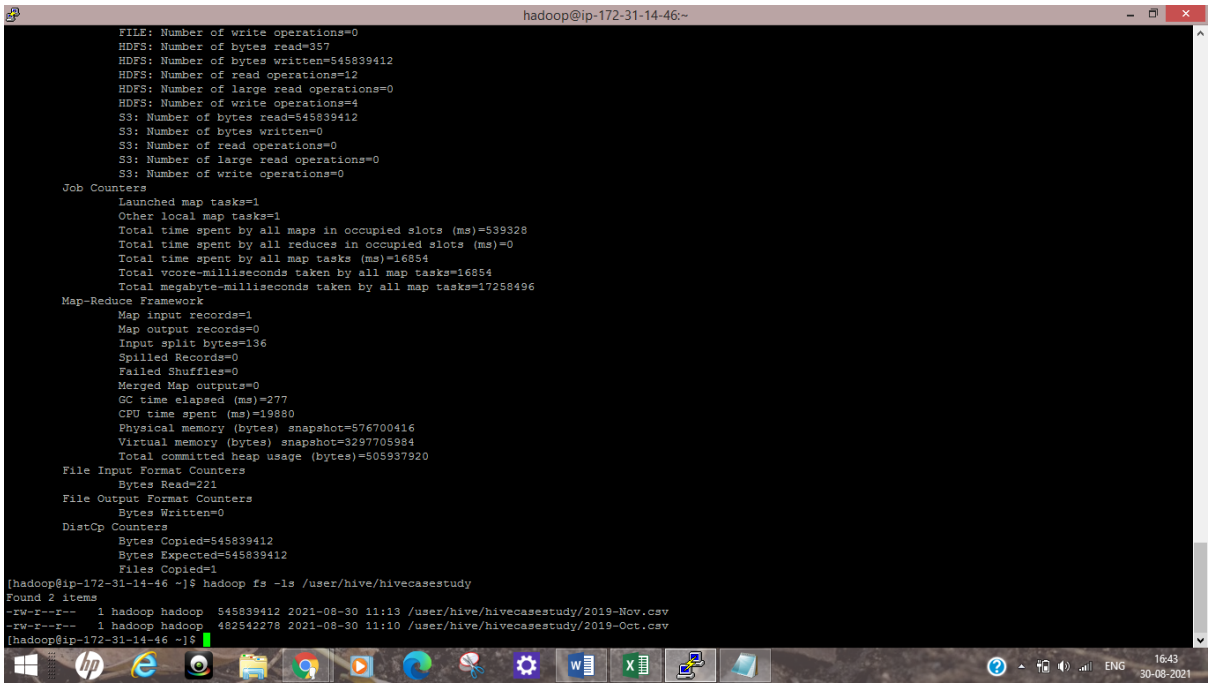
hadoop distcp s3://sunilhivecasestudy/2019-Oct.csv /user/hive/hivecasestudy/2019-Oct.csv

hadoop distcp s3://sunilhivecasestudy/2019-Nov.csv /user/hive/hivecasestudy/2019-Nov.csv



Both the files are successfully copied into the HDFS System.

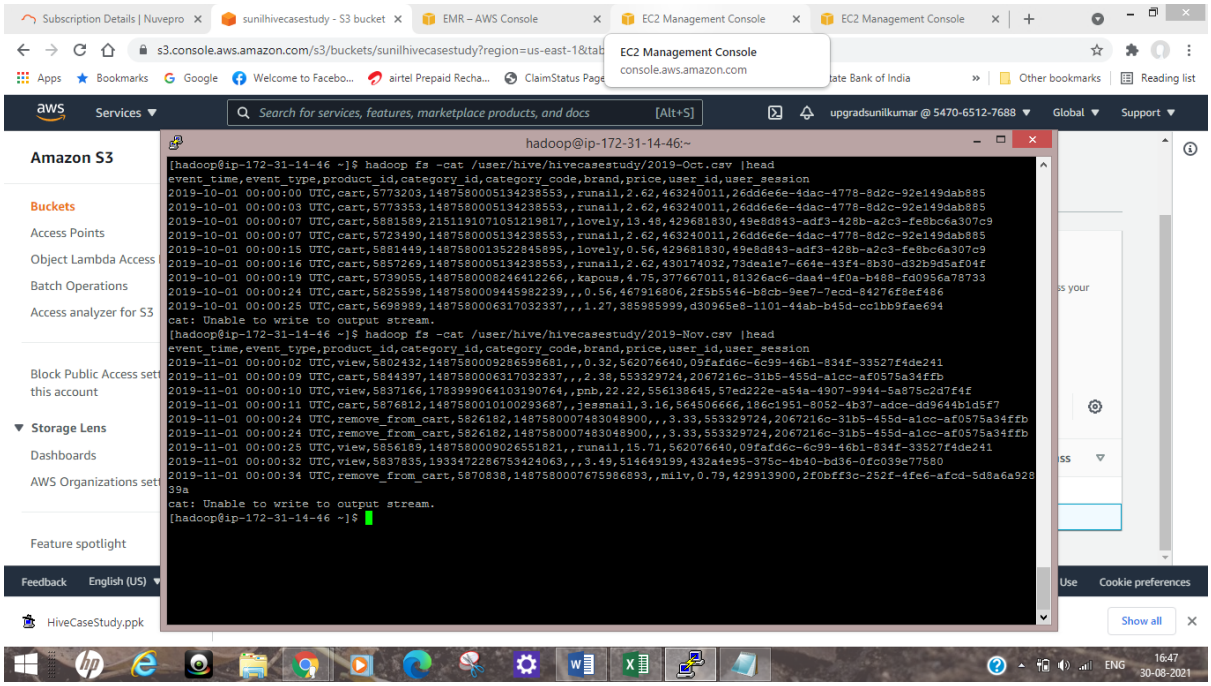
hadoop fs -ls /user/hive/hivecasestudy



Checking the top rows of the copied files in HDFS using

hadoop fs -cat /user/hive/hivecasestudy/2019-Oct.csv |head

hadoop fs -cat /user/hive/hivecasestudy/2019-Nov.csv |head

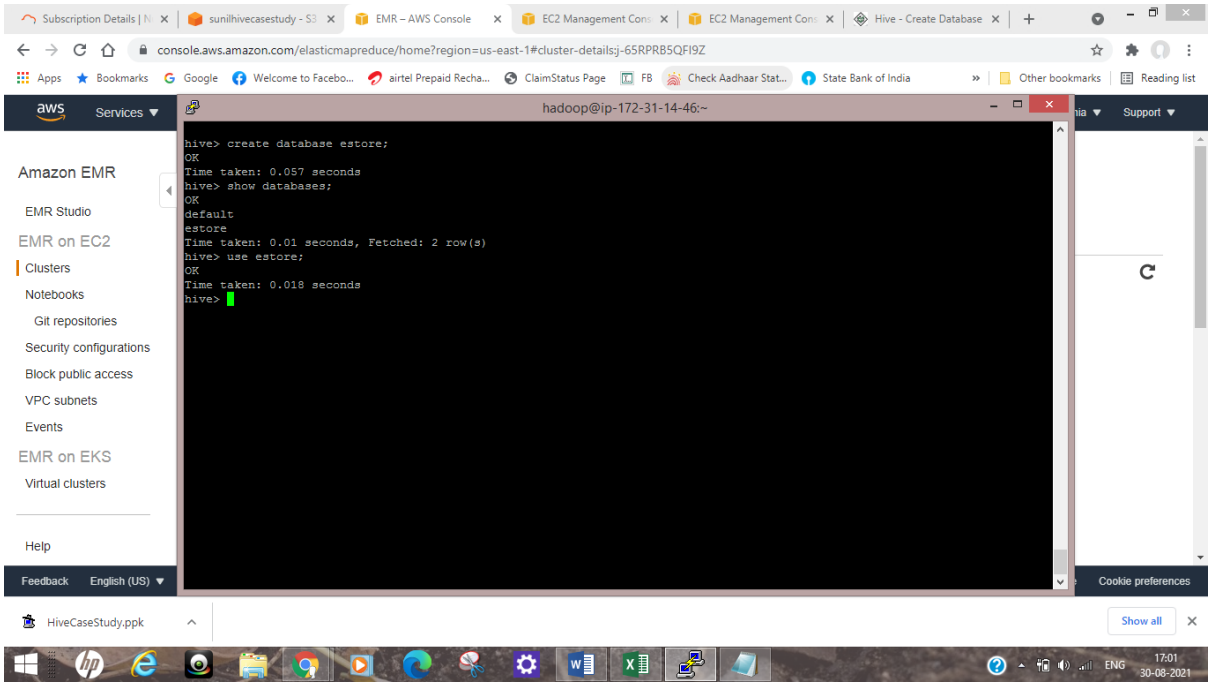


We will be entering the Hive CLI and creating database to store data

Create database estore;

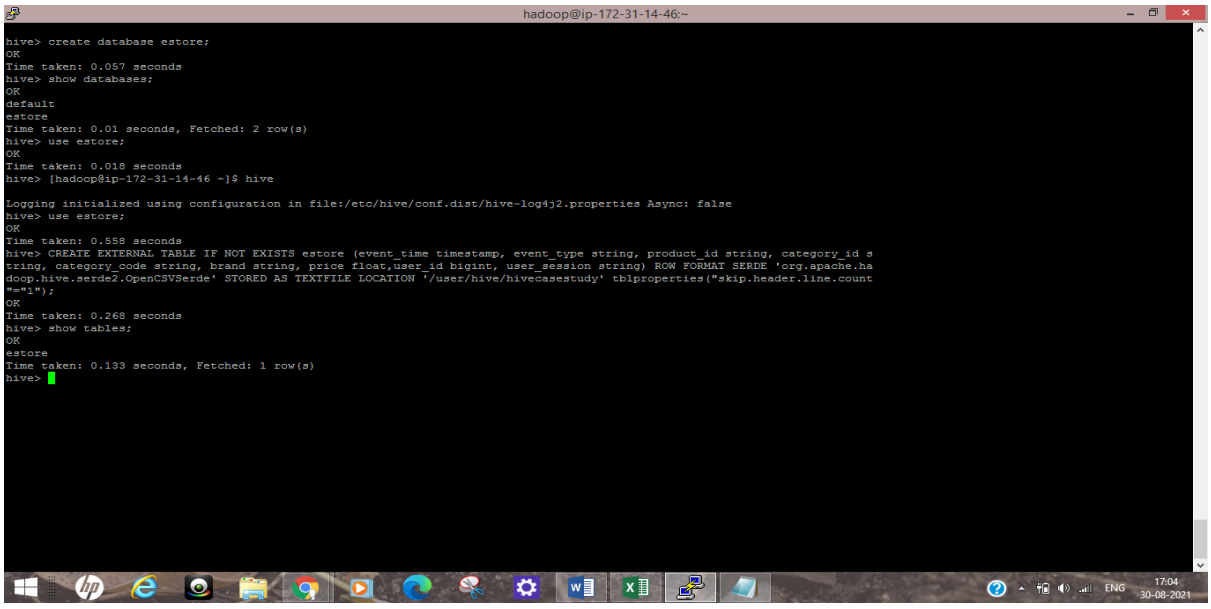
Use estore;

We will use estore database to load data and do the analysis.



Creating External Table Using CSVSerde:

CREATE EXTERNAL TABLE IF NOT EXISTS estore (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price float,user_id bigint, user_session string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hivecasestudy' tblproperties("skip.header.line.count"="1");

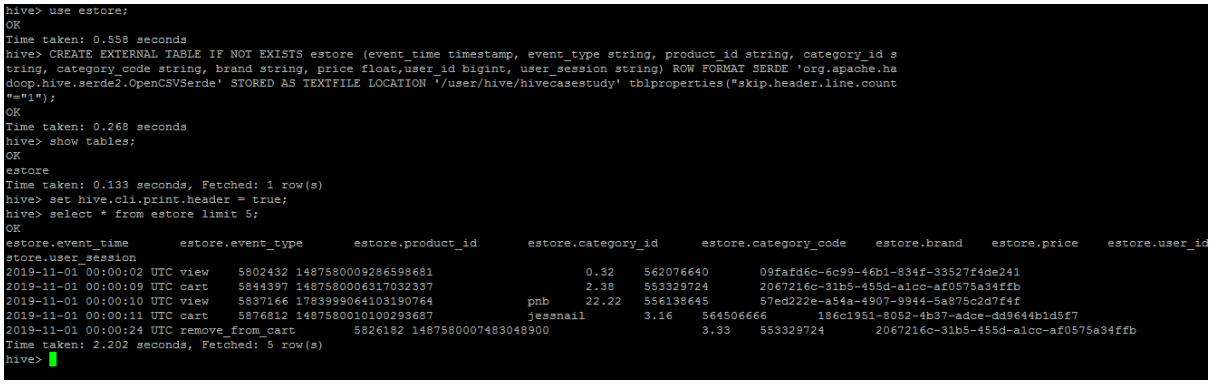


To display the header columns

set hive.cli.print.header = true;

We are checking the top 5 rows in the table using

select * from estore limit 5;



Creating Optimised Table:

We are creating the optimised tables using partitioning and Bucketing Techniques:

set hive.exec.dynamic.partition.mode = nonstrict;

set hive.exec.dynamic.partition = true;

set hive.enforce.bucketing = true;

Creating an optimized table by partitioning on “event_type” and bucketing on “price”

CREATE TABLE IF NOT EXISTS dynpart_bucket_estore(event_time timestamp, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string) PARTITIONED BY (event_type string) CLUSTERED BY (price) INTO 10 BUCKETS ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hivecasestudy' tblproperties('skip.header.line.count' = '1');

Copying the data from the estore table:

INSERT INTO TABLE dynpart_bucket_estore PARTITION (event_type) SELECT event_time,product_id, category_id, category_code, brand, price, user_id, user_session, event_type FROM estore;

```
hive> CREATE TABLE IF NOT EXISTS dynpart_bucket_estore(event_time timestamp, product_id string, category_id string, category_code string, brand string, price float, use
r_id bigint, user_session string) PARTITIONED BY (event_type string) CLUSTERED BY (price) INTO 10 BUCKETS ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS TEXTFILE LOCATION '/user/hive/hivecasestudy' tblproperties('skip.header.line.count' = '1');
OK
Time taken: 0.078 seconds
hive> INSERT INTO TABLE dynpart_bucket_estore PARTITION (event_type) SELECT event_time,product_id, category_id, category_code, brand, price, user_id, user_session, even
t_type FROM estore;
Query ID = hadoop_20210830120840_0792a952-8bb6-4c6c-b82a-b825f2015bb4
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1630319660287_0005)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 160.49 s
--------------------------------------------------------------------------------
Loading data to table estore.dynpart_bucket_estore partition (event_type=null)

Loaded : 4/4 partitions.
         Time taken to load dynamic partitions: 1.043 seconds
         Time taken for adding to write entity : 0.003 seconds
OK
event_time      product_id      category_id     category_code   brand   price   user_id user_session    event_type
Time taken: 172.636 seconds
hive>
```

Checking the tables created

```
hive> show tables;
OK
tab_name
dynpart_bucket_estore
estore
Time taken: 0.043 seconds, Fetched: 2 row(s)
hive>
```

Verifying the partitioned files in the Hadoop file system

hadoop fs -ls /user/hive/hivecasestudy

Verifying the bucketed files in the Hadoop file system

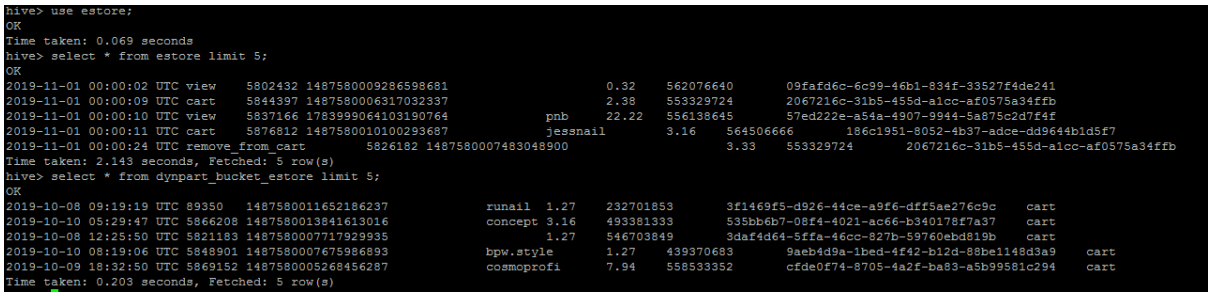
hadoop fs -ls /user/hive/hivecasestudy/event_type=cart

```
[hadoop@ip-172-31-14-46 ~]$ hadoop fs -ls /user/hive/hivecasestudy
Found 6 items
-rw-r--r--   1 hadoop hadoop  545839412 2021-08-30 11:13 /user/hive/hivecasestudy/2019-Nov.csv
-rw-r--r--   1 hadoop hadoop  482542278 2021-08-30 11:10 /user/hive/hivecasestudy/2019-Oct.csv
drwxr-xr-x   - hadoop hadoop          0 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=cart
drwxr-xr-x   - hadoop hadoop          0 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=purchase
drwxr-xr-x   - hadoop hadoop          0 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=remove_from_cart
drwxr-xr-x   - hadoop hadoop          0 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=view
[hadoop@ip-172-31-14-46 ~]$ hadoop fs -ls /user/hive/hivecasestudy/event_type=cart
Found 10 items
-rwxr-xr-x   1 hadoop hadoop   27512579 2021-08-30 12:10 /user/hive/hivecasestudy/event_type=cart/000000_0
-rwxr-xr-x   1 hadoop hadoop   32190447 2021-08-30 12:10 /user/hive/hivecasestudy/event_type=cart/000001_0
-rwxr-xr-x   1 hadoop hadoop   33302805 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=cart/000002_0
-rwxr-xr-x   1 hadoop hadoop   32602023 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=cart/000003_0
-rwxr-xr-x   1 hadoop hadoop   34104132 2021-08-30 12:10 /user/hive/hivecasestudy/event_type=cart/000004_0
-rwxr-xr-x   1 hadoop hadoop   32538513 2021-08-30 12:10 /user/hive/hivecasestudy/event_type=cart/000005_0
-rwxr-xr-x   1 hadoop hadoop   39257340 2021-08-30 12:10 /user/hive/hivecasestudy/event_type=cart/000006_0
-rwxr-xr-x   1 hadoop hadoop   24825787 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=cart/000007_0
-rwxr-xr-x   1 hadoop hadoop   28504487 2021-08-30 12:11 /user/hive/hivecasestudy/event_type=cart/000008_0
-rwxr-xr-x   1 hadoop hadoop   35410315 2021-08-30 12:10 /user/hive/hivecasestudy/event_type=cart/000009_0
[hadoop@ip-172-31-14-46 ~]$
```

Checking the optimised tables using

select * from estore limit 5;

select * from dynpart_bucket_estore limit 5;



Time taken by normal estore table is 2.143 Seconds, whereas partitioned and bucked table could retrieve the data faster in .203 Seconds.

Optimised Tables are used for better performance and faster analysis.

Answers to the questions given below.

1) Find the total revenue generated due to purchases made in October.

Base Table: Estore

SELECT SUM(price) AS tot_revenue_oct FROM estore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';

Optimised Table: Dynpart_bucket_estore

SELECT SUM(price) AS tot_revenue_oct FROM dynpart_bucket_estore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';

```
hive> SELECT SUM(price) AS tot_revenue_oct FROM dynpart_bucket_estore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';
Query ID = hadoop_20210830131019_7df1b9a2-368a-44e6-b660-31546468606b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1630319660287_0007)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 22.38 s
--------------------------------------------------------------------------------
OK
tot_revenue_oct
1211532.4500002791
Time taken: 23.489 seconds, Fetched: 1 row(s)
hive> SELECT SUM(price) AS tot_revenue_oct FROM estore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';
Query ID = hadoop_20210830131143_aa3b800f-db98-4946-a44a-4ec8a1a9c635
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1630319660287_0007)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      5          5        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 125.93 s
--------------------------------------------------------------------------------
OK
tot_revenue_oct
1211538.4299997438
Time taken: 126.554 seconds, Fetched: 1 row(s)
hive>
```

Output/Observations:

Total revenue generated based on purchase made in the month of October is 1,211,538.43 rupees.

Base table query took the execution time of 126 seconds whereas optimized table query took execution time of 23 seconds.

We can observe that there is drop in the execution time of the same query. Optimized table gives better performance in execution time.

---

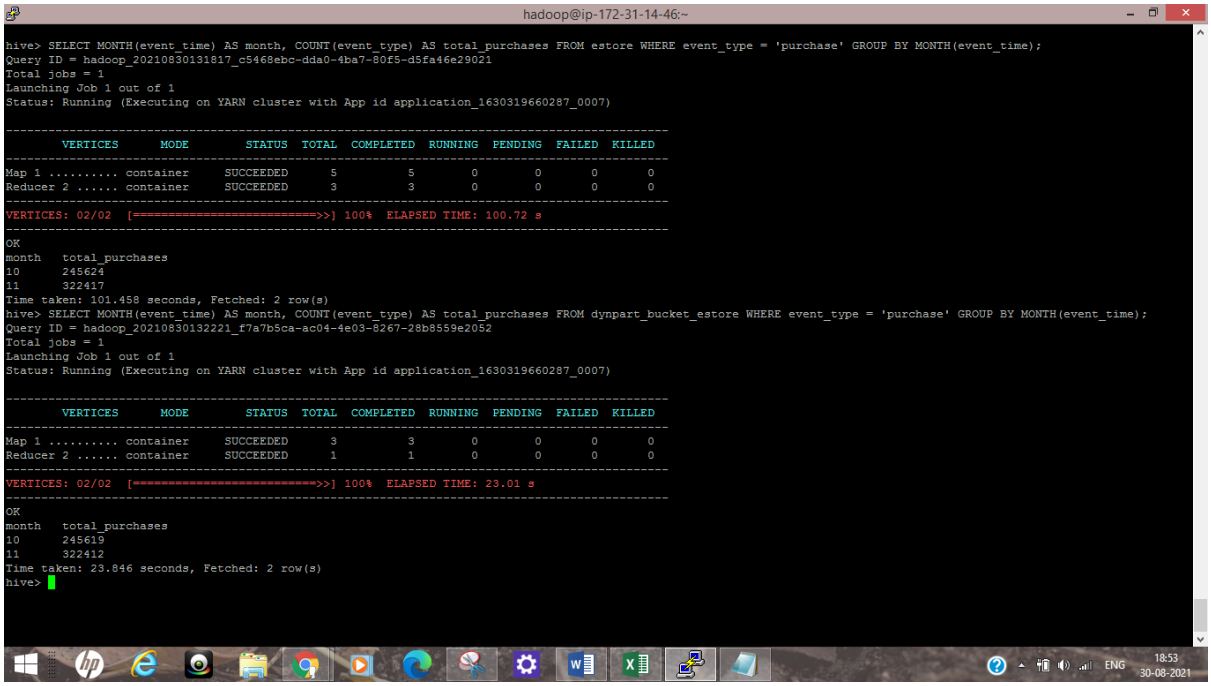
2) Write a query to yield the total sum of purchases per month in a single output.

Base Table: Estore

SELECT MONTH(event_time) AS month, COUNT(event_type) AS total_purchases FROM estore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);

Optimised Table: Dynpart_bucket_estore

SELECT MONTH(event_time) AS month, COUNT(event_type) AS total_purchases FROM dynpart_bucket_estore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);



Output/Observations:

Count of purchases made in the month of October is 245624 and in the month of November 322417, we observe that there is 31% Increase in the November Purchases compared to October Purchases.

Base table query took the execution time of 101 seconds whereas optimized table query took execution time of 24 seconds.

We also observe that partitioning and bucketing techniques on table can reduce execution time substantially.

---

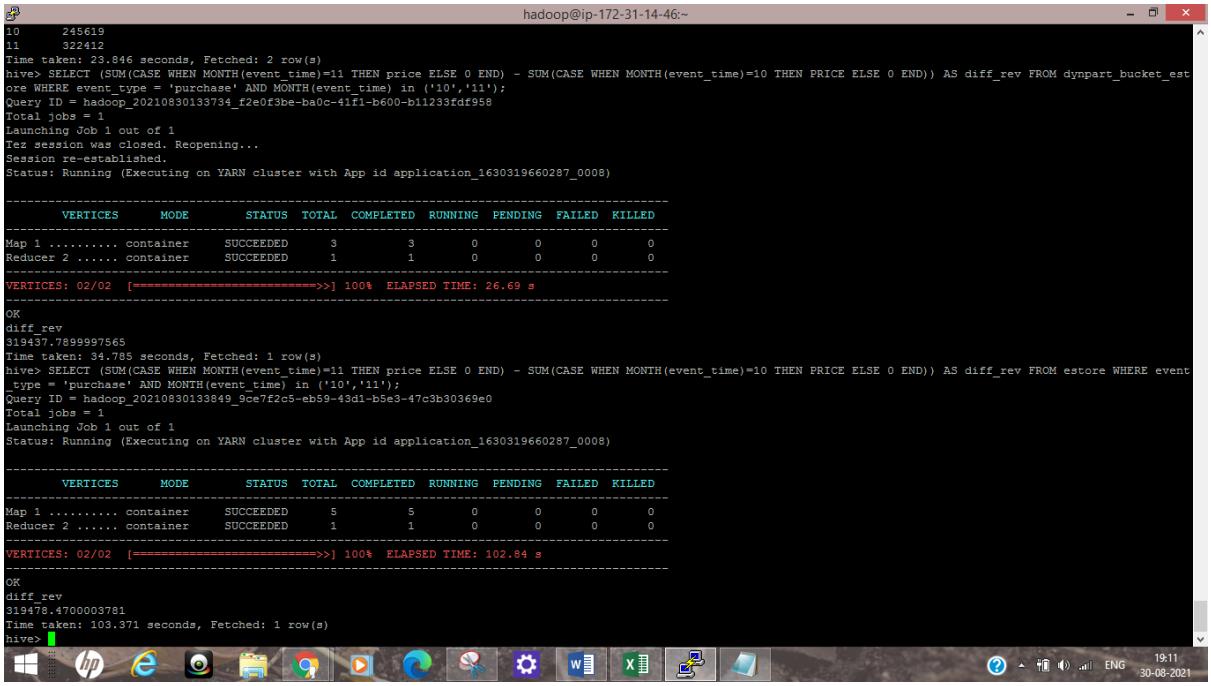
3) Write a query to find the change in revenue generated due to purchases from October to November.

Base Table: Estore

SELECT (SUM(CASE WHEN MONTH(event_time)=11 THEN price ELSE 0 END) - SUM(CASE WHEN MONTH(event_time)=10 THEN PRICE ELSE 0 END)) AS diff_rev FROM dynpart_bucket_estore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11');

Optimised Table: Dynpart_bucket_estore

SELECT (SUM(CASE WHEN MONTH(event_time)=11 THEN price ELSE 0 END) - SUM(CASE WHEN MONTH(event_time)=10 THEN PRICE ELSE 0 END)) AS diff_rev FROM estore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11');



Output/Observations:

We can observe that the difference in revenue between October purchases and November Purchases is 319478.

Base table query took the execution time of 103 seconds whereas optimized table query took execution time of 34 seconds

We also observe that partitioning and bucketing techniques on table can reduce execution time substantially.

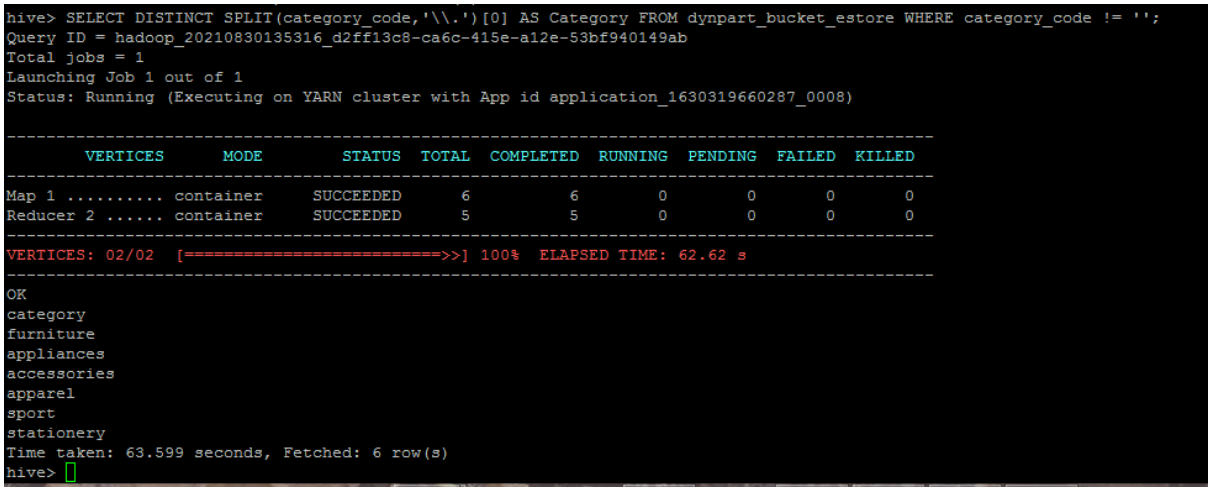
We will be using the Optimised Table for further analysis.

---

4) Find distinct categories of products. Categories with null category code can be ignored.

Optimised Table: Dynpart_bucket_estore

SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Category FROM dynpart_bucket_estore WHERE category_code != '';



Output/Observations:

We have 6 unique categories are –

Furniture	Apparel
Appliances	Sport
Accessories	Stationary

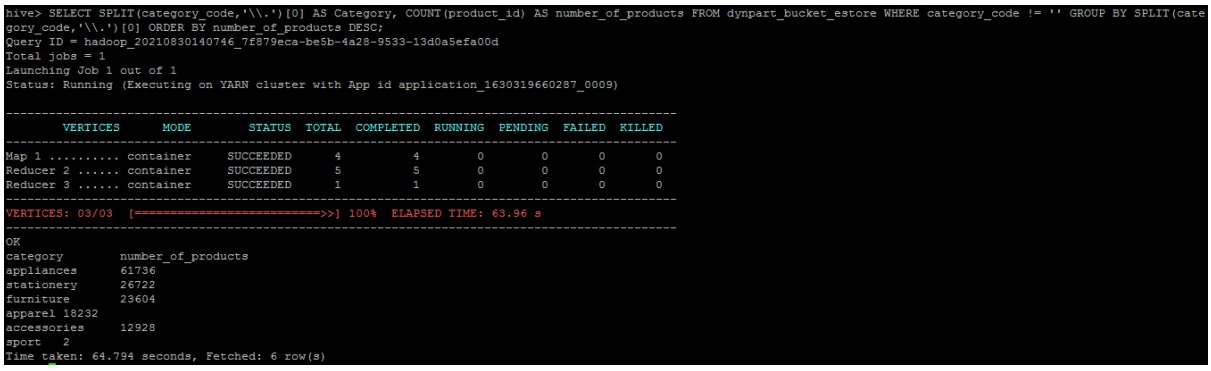
Time taken to execute the query is 63 seconds

---

5) Find the total number of products available under each category.

Optimised Table: Dynpart_bucket_estore

SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS number_of_products FROM dynpart_bucket_estore WHERE category_code != '' GROUP BY SPLIT(category_code,'\\.')[0] ORDER BY number_of_products DESC;



Output/Observations:

Appliances	61736
Stationary	26722
Furniture	23604
Apparel	18232
Accessories	12928
Sport	2

Appliances are the most popular category with 61K Products, Stationary and Furniture are in the range of 26K to 23 K Products, Followed by Apparel with 18K Products, Accessories with almost 13K Product and Sports with 2 products.

Time Taken for execution of query is 65 Seconds

---

6) Which brand had the maximum sales in October and November combined?

Optimised Table: Dynpart_bucket_estore

WITH total_sales_summary AS(

SELECT brand, round((SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) + SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)),2) AS total_sales FROM dynpart_bucket_estore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)

SELECT brand, total_sales FROM total_sales_summary ORDER BY total_sales DESC LIMIT 1;

```
hive> WITH total_sales_summary AS(
    > SELECT brand, round((SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) + SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)),2) AS total_sales
    > FROM dynpart_bucket_estore
    > WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)
    > SELECT brand, total_sales FROM total_sales_summary ORDER BY total_sales DESC LIMIT 1;
Query ID = hadoop_20210830141719_88610c8c-8d63-4537-889a-88f0b8269e1a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1630319660287_0010)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 25.80 s
--------------------------------------------------------------------------------
OK
brand   total_sales
runail  148292.46
Time taken: 26.876 seconds, Fetched: 1 row(s)
```

Runail is the brand that has the highest sales in total of both the months October and November

It seems that Runail brand has high popularity among the buyers and could help in increasing their profit.

Time Taken for execution of query is 27 Seconds

---

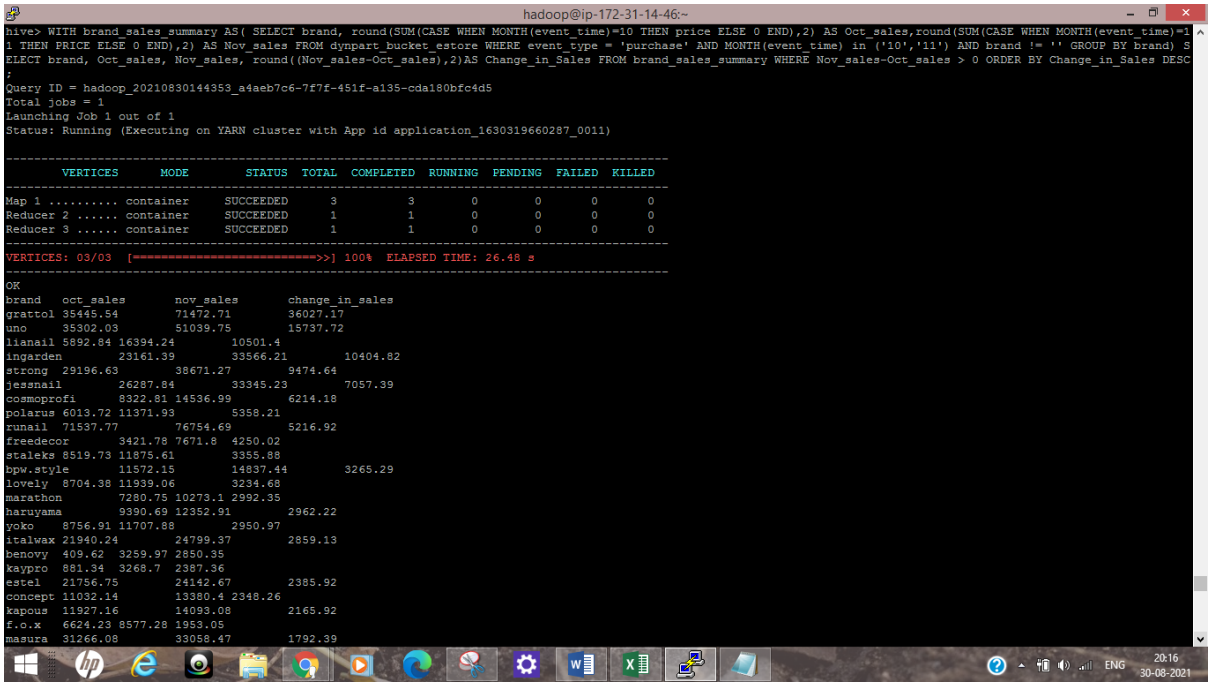
7) Which brands increased their sales from October to November?

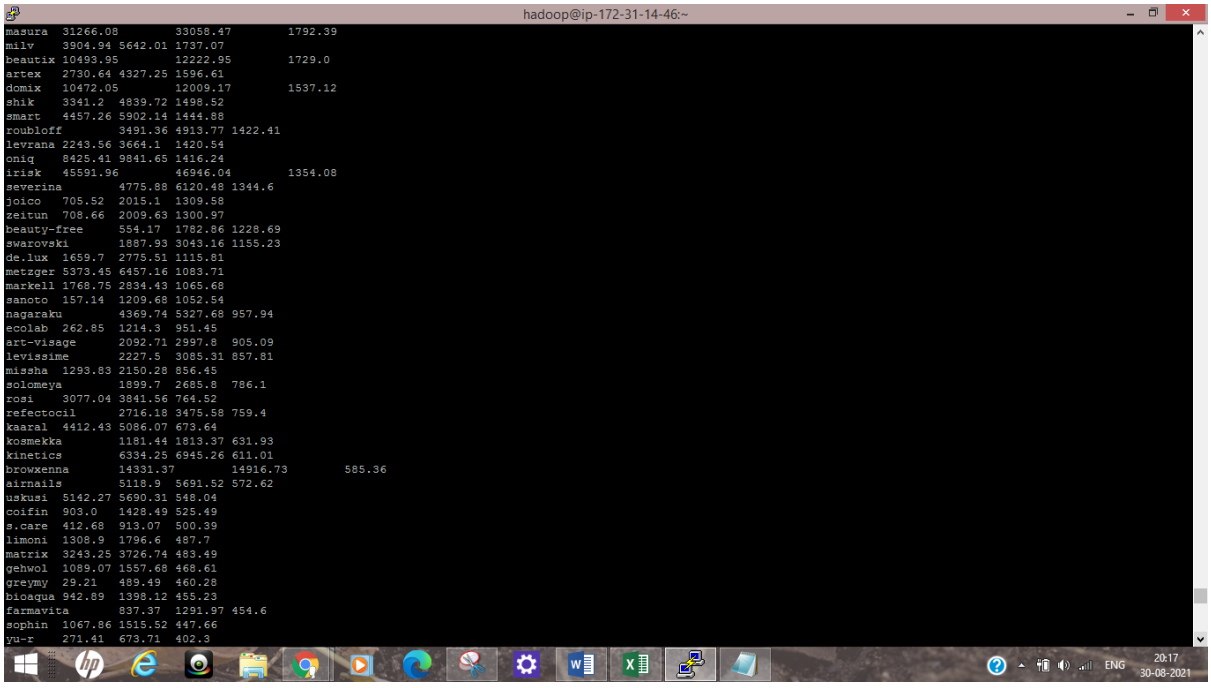
Optimised Table: Dynpart_bucket_estore

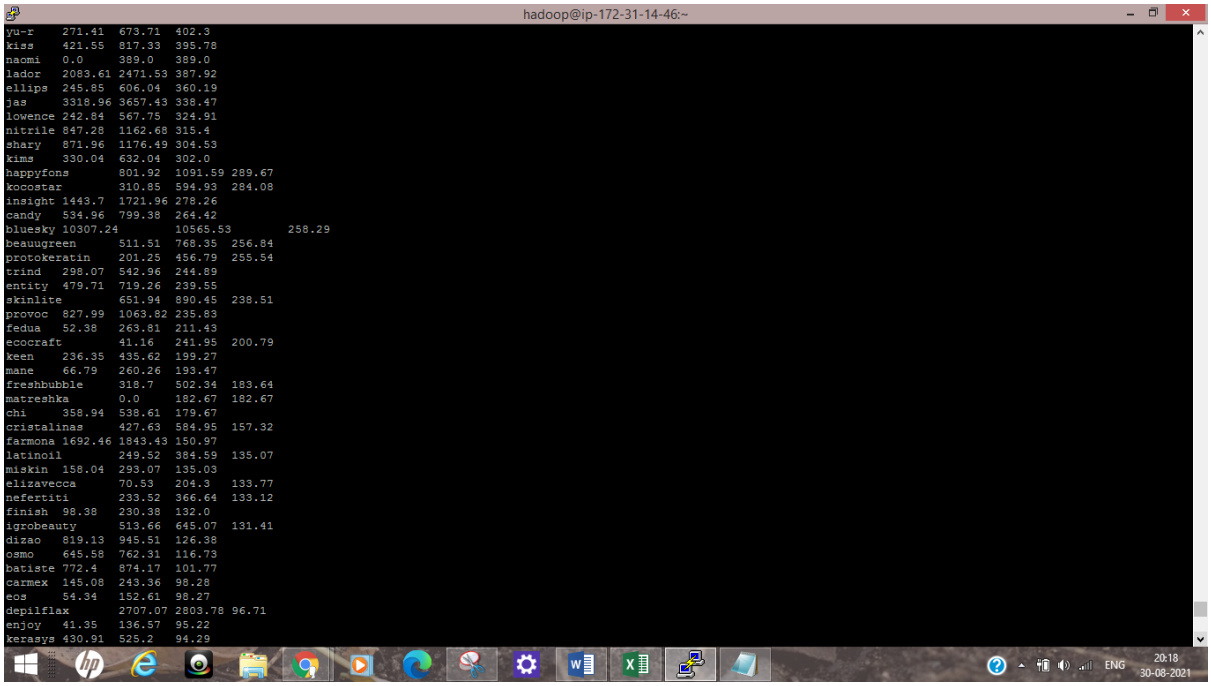
WITH brand_sales_summary AS(

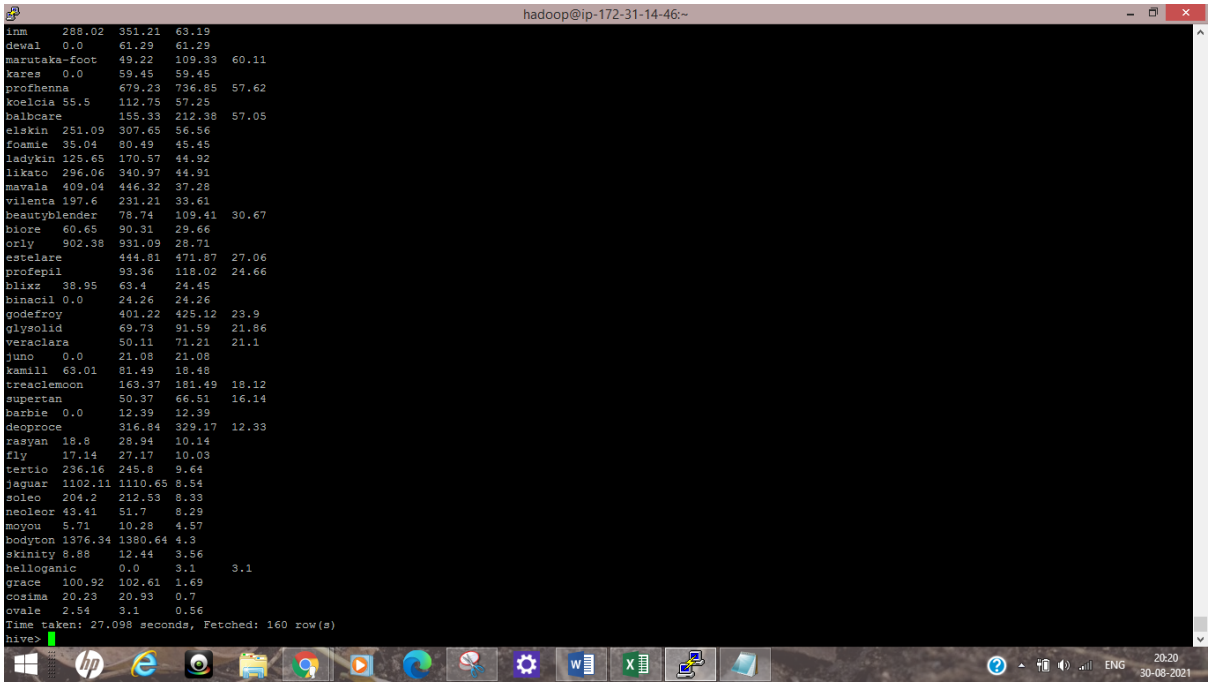
SELECT brand, round(SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END),2) AS Oct_sales,round(SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END),2) AS Nov_sales FROM dynpart_bucket_estore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)

SELECT brand, Oct_sales, Nov_sales, round((Nov_sales-Oct_sales),2)AS Change_in_Sales FROM brand_sales_summary WHERE Nov_sales-Oct_sales > 0 ORDER BY Change_in_Sales DESC;


hadoop@ip-172-31-14-46:~
hive> WITH brand_sales_summary AS(SELECT brand, round(SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END),2) AS Oct_sales,round(SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END),2) AS Nov_sales FROM dynpart_bucket_estore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand) SELECT brand, Oct_sales, Nov_sales, round((Nov_sales-Oct_sales),2)AS Change_in_Sales FROM brand_sales_summary WHERE Nov_sales-Oct_sales > 0 ORDER BY Change_in_Sales DESC;
Query ID = hadoop_20210830144353_a4aeb7c6-7f7f-451f-a135-cda180bfc4d5
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1630319660287_0011)
VERTICES MODE STATUS TOTAL COMPLETED RUNNING PENDING FAILED KILLED
Map 1 ..... container SUCCEEDED 3 3 0 0 0 0
Reducer 2 ..... container SUCCEEDED 1 1 0 0 0 0
Reducer 3 ..... container SUCCEEDED 1 1 0 0 0 0
VERTICES: 03/03 [==========>>] 100% ELAPSED TIME: 26.48 s
OK
brand oct_sales nov_sales change_in_sales
grattol 35445.54 71472.71 36027.17
uno 35302.03 51039.75 15737.72
lianail 5892.84 16394.24 10501.4
ingarden 23161.39 33566.21 10404.82
strong 29196.63 38671.27 9474.64
jessnail 26287.84 33345.23 7057.39
cosmoprofi 8322.81 14536.99 6214.18
polarus 6013.72 11371.93 5358.21
runail 71537.77 76754.69 5216.92
freedecor 3421.78 7671.8 4250.02
staleks 8519.73 11875.61 3355.88
bpw.style 11572.15 14837.44 3265.29
lovely 8704.38 11939.06 3234.68
marathon 7280.75 10273.1 2992.35
haruyama 9390.69 12352.91 2962.22
yoko 8756.91 11707.88 2950.97
italwax 21940.24 24799.37 2859.13
benovy 409.62 3259.97 2850.35
kaypro 881.34 3268.7 2387.36
estel 21756.75 24142.67 2385.92
concept 11032.14 13380.4 2348.26
kapous 11927.16 14093.08 2165.92
f.o.x 6624.23 8577.28 1953.05
masura 31266.08 33058.47 1792.39
ENG 20:16 30-08-2021


hadoop@ip-172-31-14-46:~
masura 31266.08 33058.47 1792.39
milv 3904.94 5642.01 1737.07
beautix 10493.95 12222.95 1729.0
artex 2730.64 4327.25 1596.61
domix 10472.05 12009.17 1537.12
shik 3341.2 4839.72 1498.52
smart 4457.26 5902.14 1444.88
roubloff 3491.36 4913.77 1422.41
levrana 2243.56 3664.1 1420.54
oniq 8425.41 9841.65 1416.24
irisk 45591.96 46946.04 1354.08
severina 4775.88 6120.48 1344.6
joico 705.52 2015.1 1309.58
zeitun 708.66 2009.63 1300.97
beauty-free 554.17 1782.86 1228.69
swarovski 1887.93 3043.16 1155.23
de.lux 1659.7 2775.51 1115.81
metzger 5373.45 6457.16 1083.71
markell 1768.75 2834.43 1065.68
sanoto 157.14 1209.68 1052.54
nagaraku 4369.74 5327.68 957.94
ecolab 262.85 1214.3 951.45
art-visage 2092.71 2997.8 905.09
levissime 2227.5 3085.31 857.81
missha 1293.83 2150.28 856.45
solomeya 1899.7 2685.8 786.1
rosi 3077.04 3841.56 764.52
refectocil 2716.18 3475.58 759.4
kaaral 4412.43 5086.07 673.64
kosmekka 1181.44 1813.37 631.93
kinetics 6334.25 6945.26 611.01
browxenna 14331.37 14916.73 585.36
airnails 5118.9 5691.52 572.62
uskusi 5142.27 5690.31 548.04
coifin 903.0 1428.49 525.49
s.care 412.68 913.07 500.39
limoni 1308.9 1796.6 487.7
matrix 3243.25 3726.74 483.49
gehwol 1089.07 1557.68 468.61
greymy 29.21 489.49 460.28
bioaqua 942.89 1398.12 455.23
farmavita 837.37 1291.97 454.6
sophin 1067.86 1515.52 447.66
yu-r 271.41 673.71 402.3
ENG 20:17 30-08-2021





Output/Observations:

160 brands have increased the selling from October to November.

‘Grattol’ brand has the highest total increment i.e., 36,027 /- and ‘Ovale’ seems to have the least increment of 0.56 /- from October to November.

‘Runail’ is the best and popular brand among the buyers with highest sales of around 70K in October and November.

Time Taken for execution of query is 27 Seconds

---

8) Your company wants to reward the top 10 users of its website with a Golden Customer plan. Write a query to generate a list of top 10 users who spend the most.

SELECT user_id, round(SUM(price),2) AS total_spends FROM dynpart_bucket_estore WHERE event_type = 'purchase' GROUP BY user_id ORDER BY total_spends DESC LIMIT 10;

```
hive> SELECT user_id, round(SUM(price),2) AS total_spends FROM dynpart_bucket_estore WHERE event_type = 'purchase' GROUP BY user_id ORDER BY total_spends DESC LIMIT 10;

Query ID = hadoop_20210830150031_aac74086-3154-43c2-9b2f-54b97b544353
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1630319660287_0012)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      3          3        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 26.19 s
----------------------------------------------------------------------------------------------
OK
user_id total_spends
557790271       2715.87
150318419       1645.97
562167663       1352.85
531900924       1329.45
557850743       1295.48
522130011       1185.39
561592095       1109.7
431950134       1097.59
566576008       1056.36
521347209       1040.91
Time taken: 27.111 seconds, Fetched: 10 row(s)
hive>
```

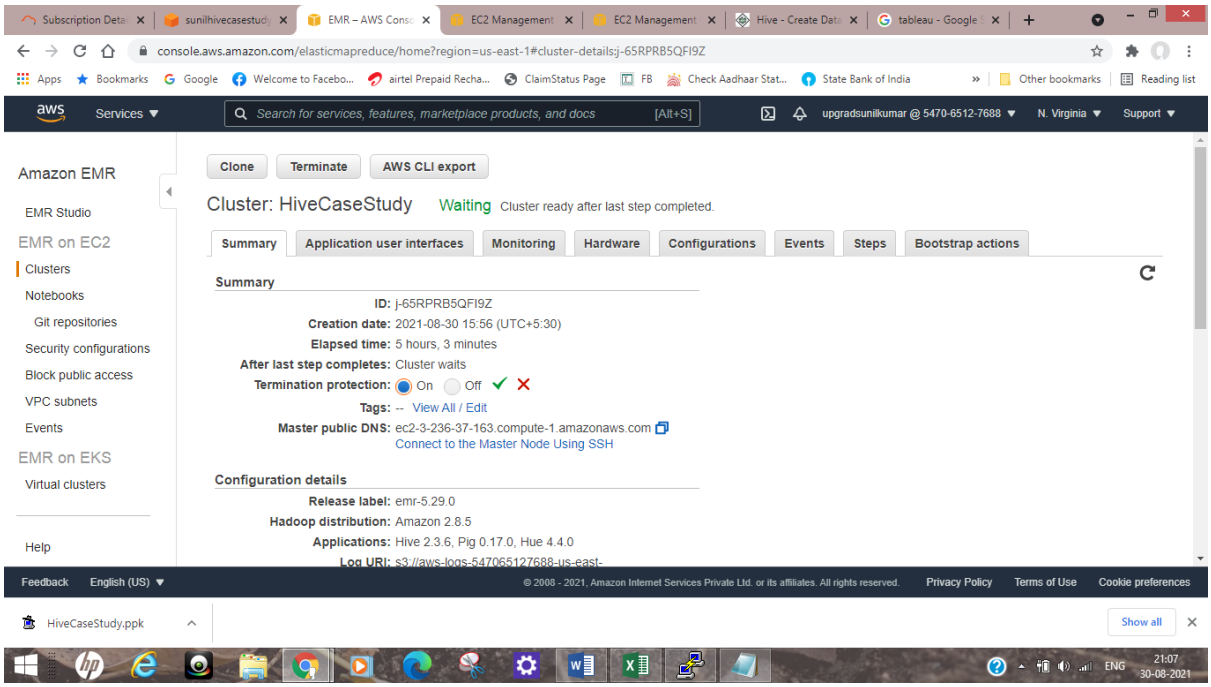
Output/Observations:

Extracted the list of Top 10 Customers who has spent most in the website and shall be rewarded with Golden Customers to improve the customer engagement.

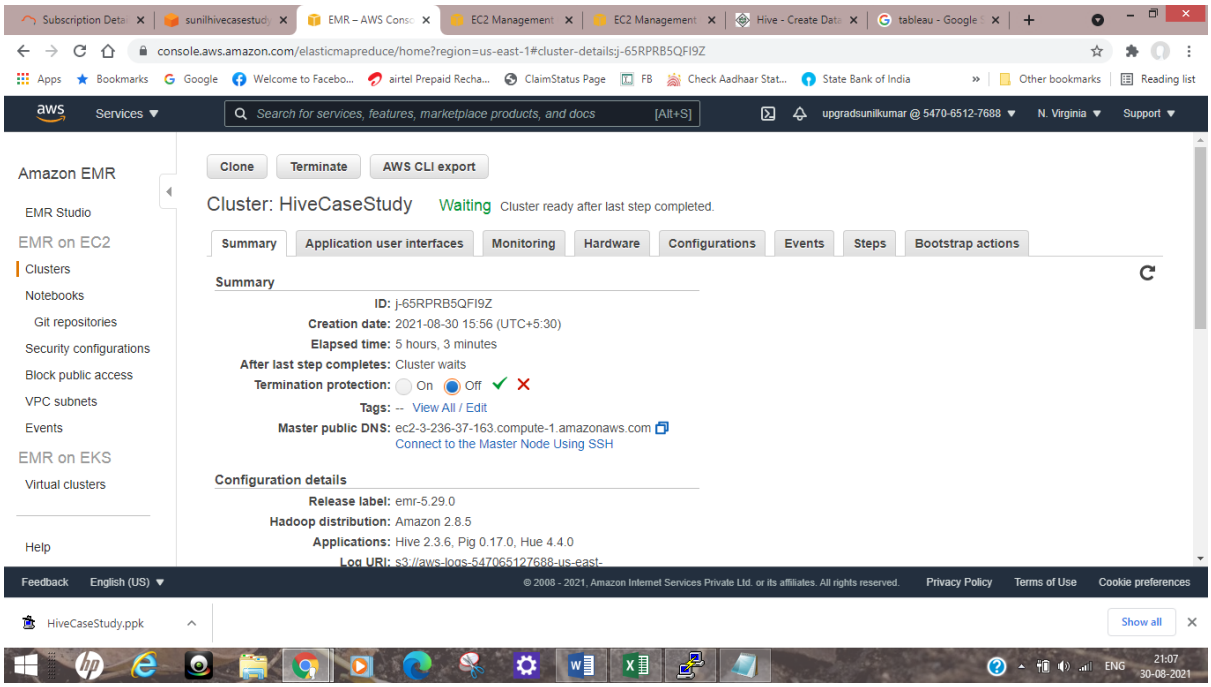
Time Taken for execution of query is 27 Seconds

We have completed the analysis, Let us terminate the EMR Cluster.

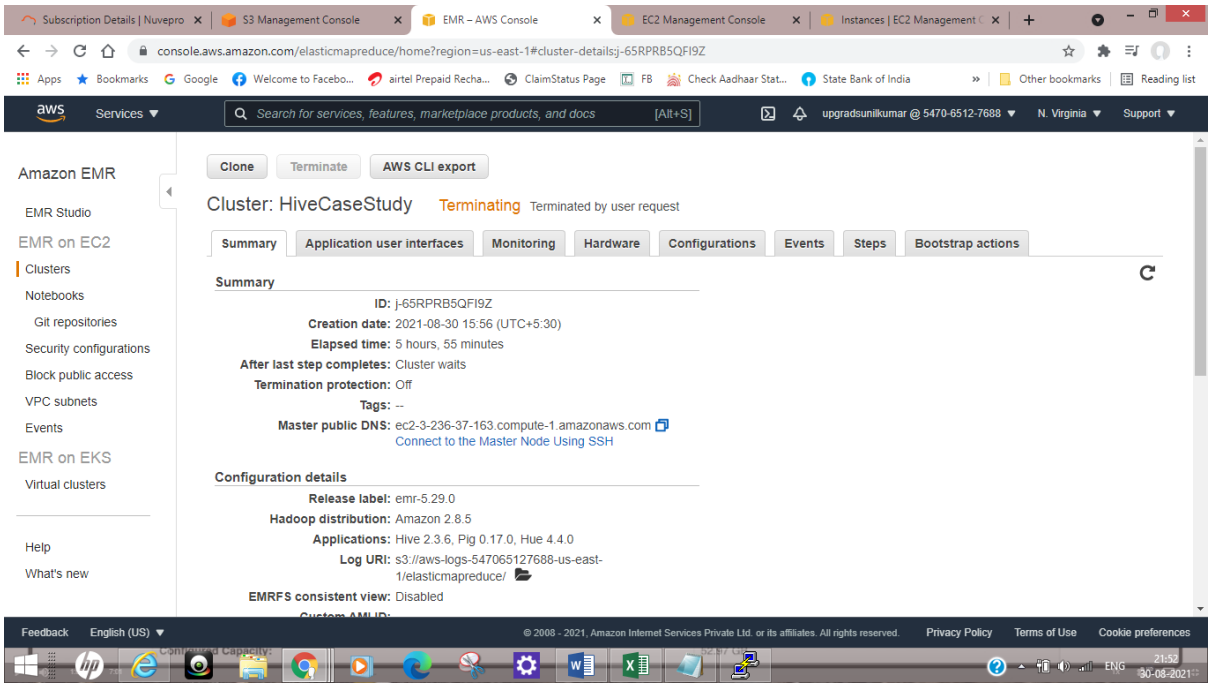
Click on the Terminate Protection and Disable the option and save it.

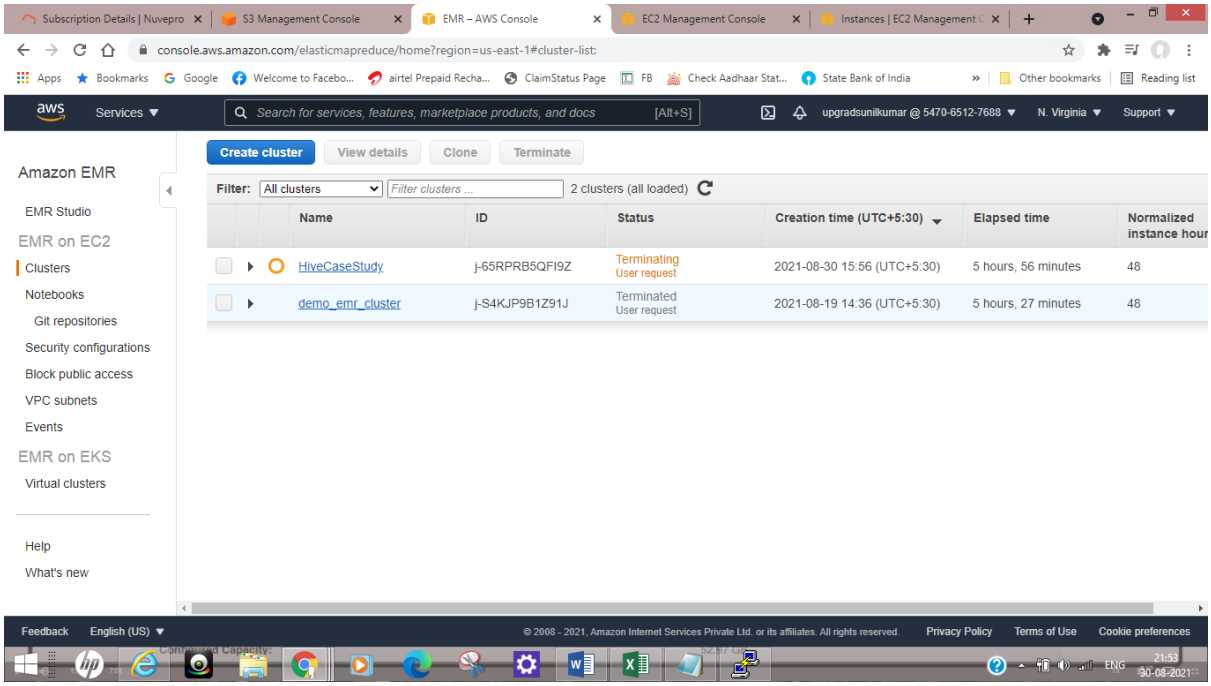


After saving, click on the Terminate

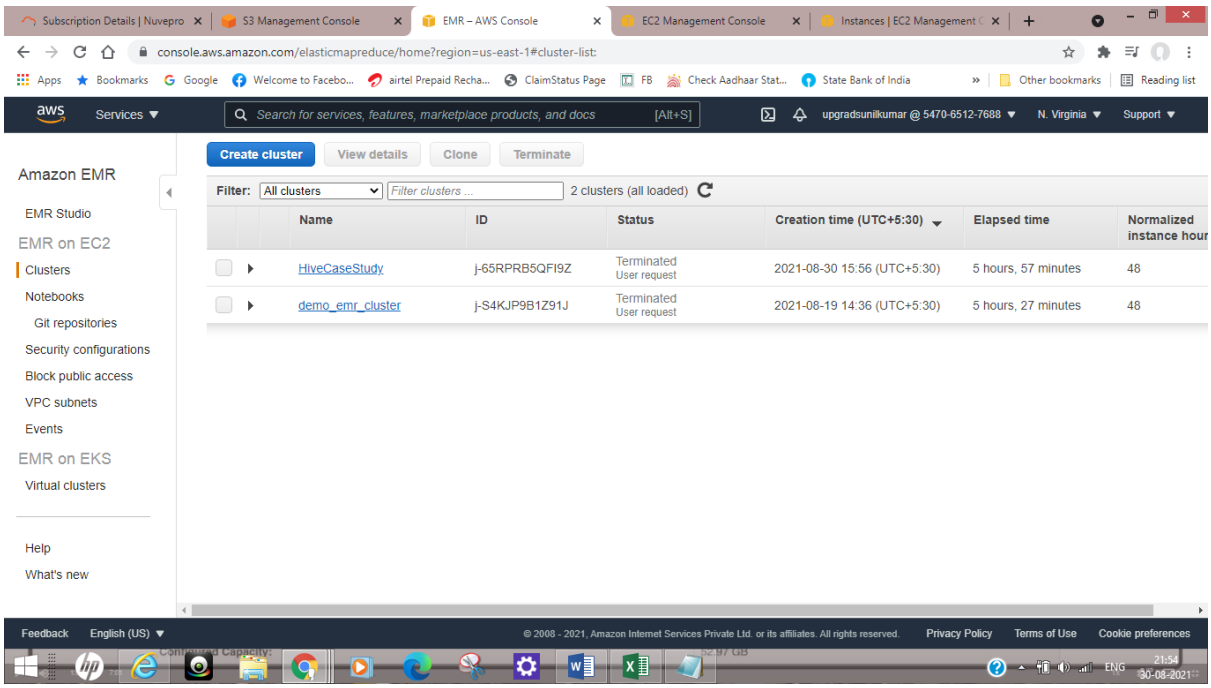


EMR Cluster status has changed from waiting to Terminating

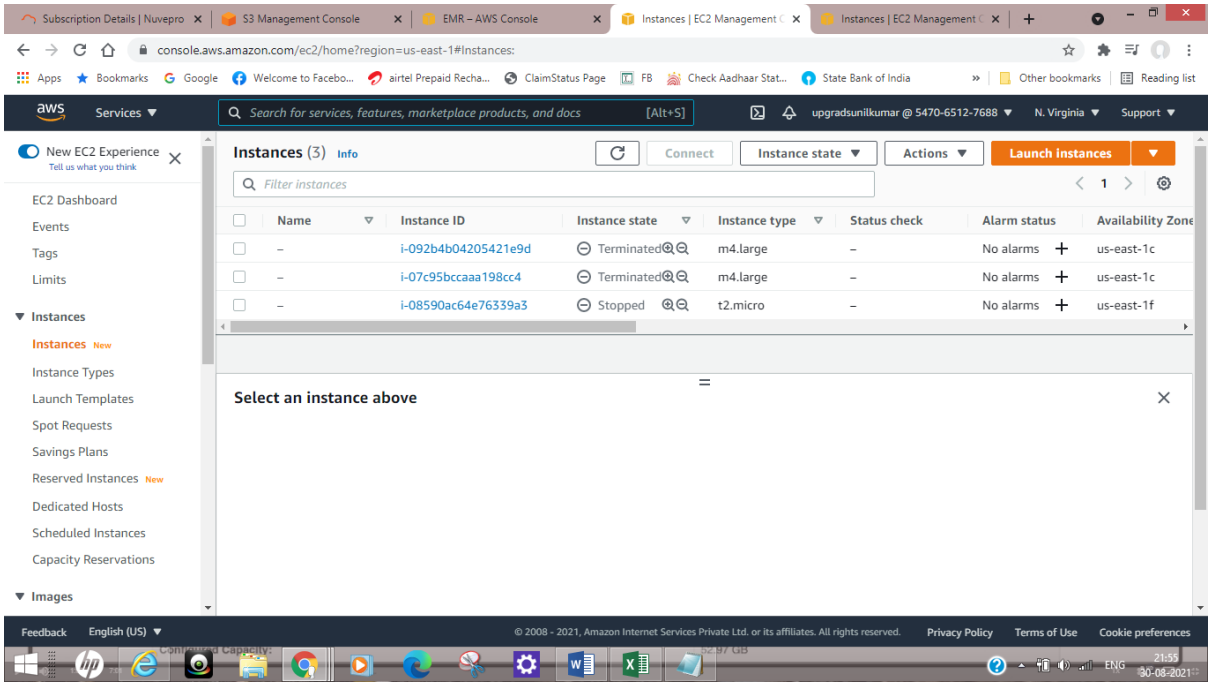




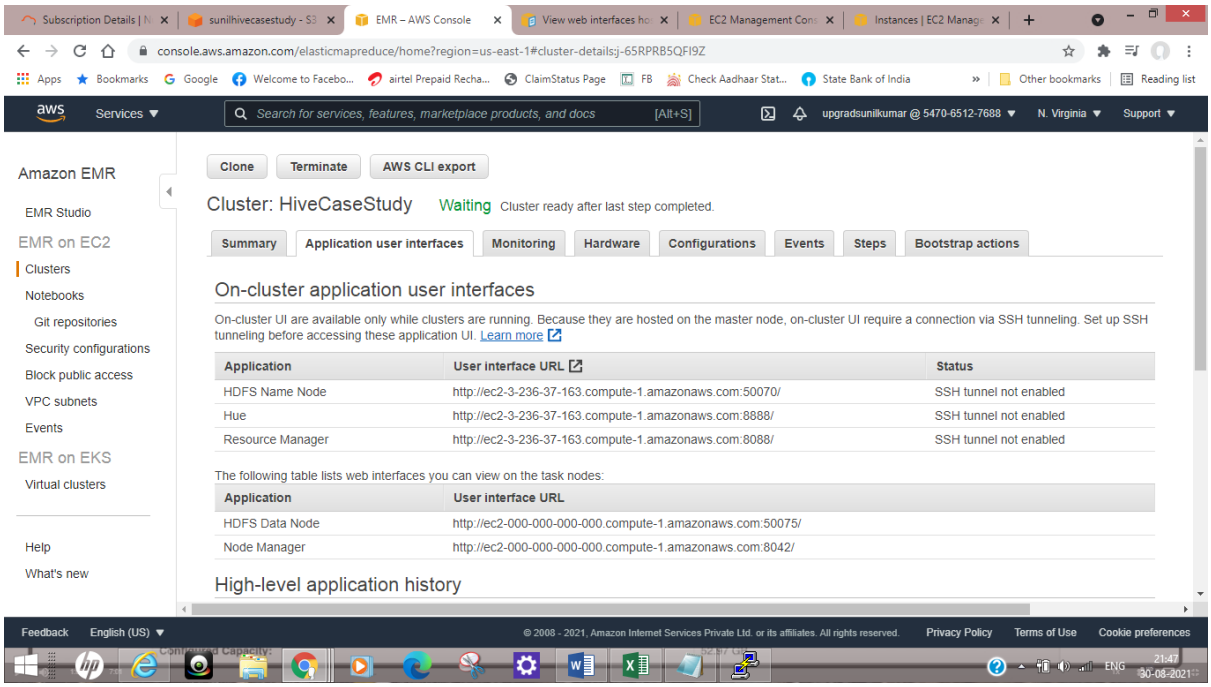
Cluster has Terminated Sccesfully



All the Instances are also Terminated Successfully

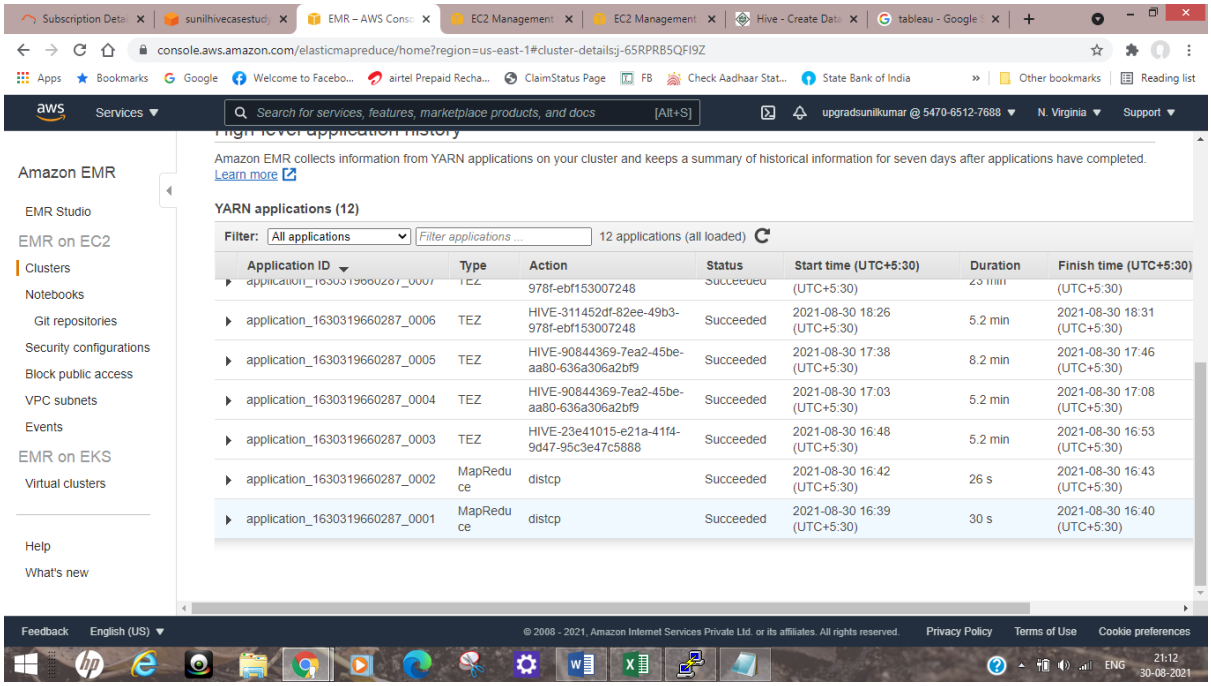


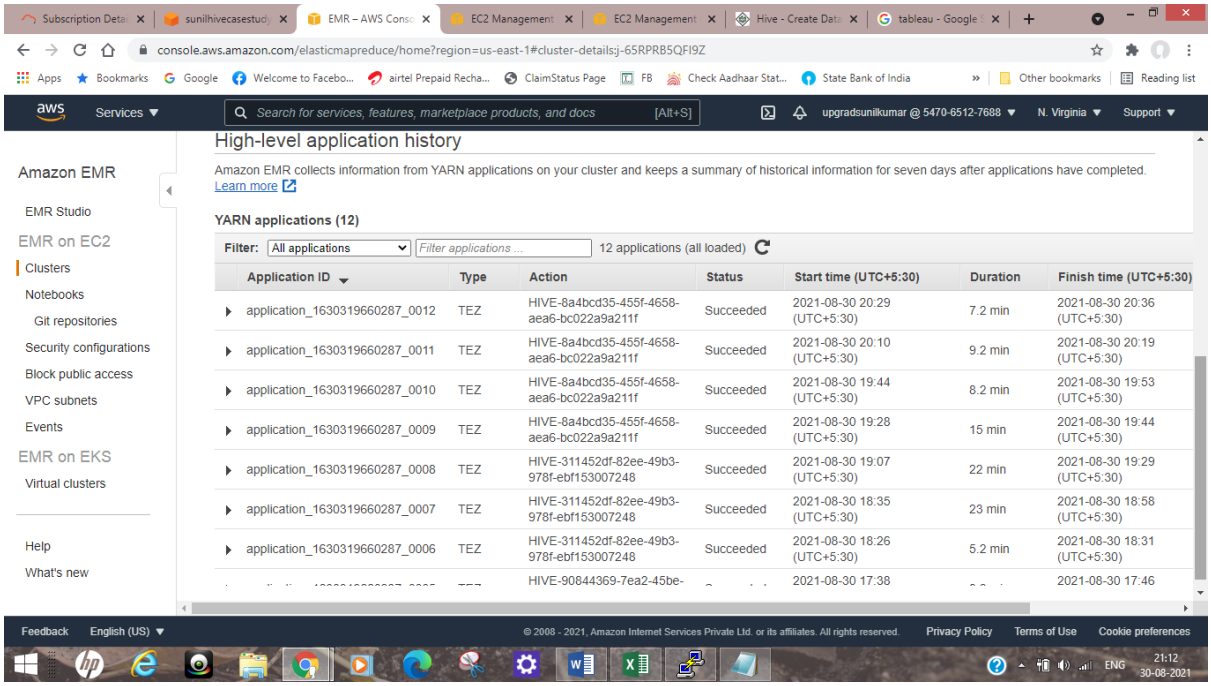
Links of Web Interface Applications of HDFS, MasterNode, Hue



List of Events that has done during the analysis:

List of Activities that have taken place in Yarn Applications





We have completed the analysis using Hive Successfully using EMR Cluster.

Thank You!