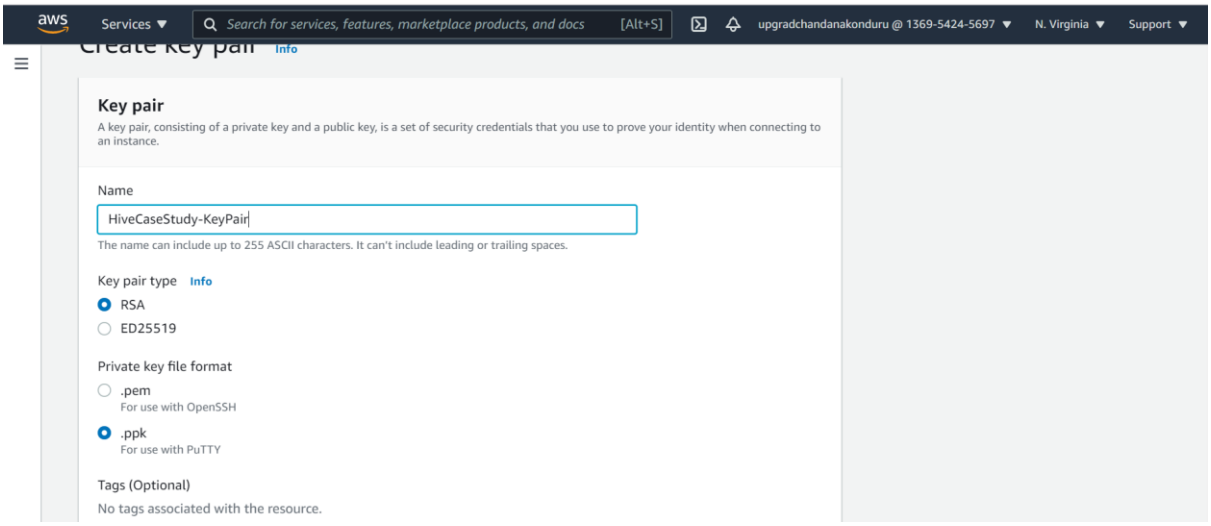
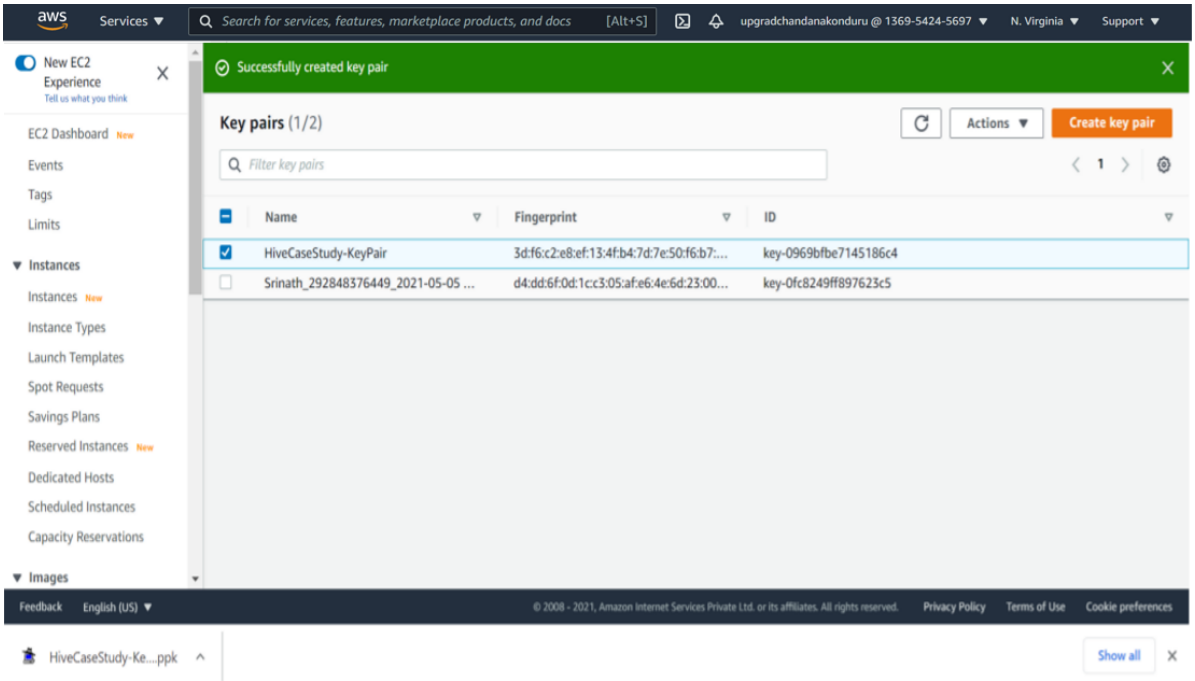
HIVE Case Study

By Chandana, Giresh

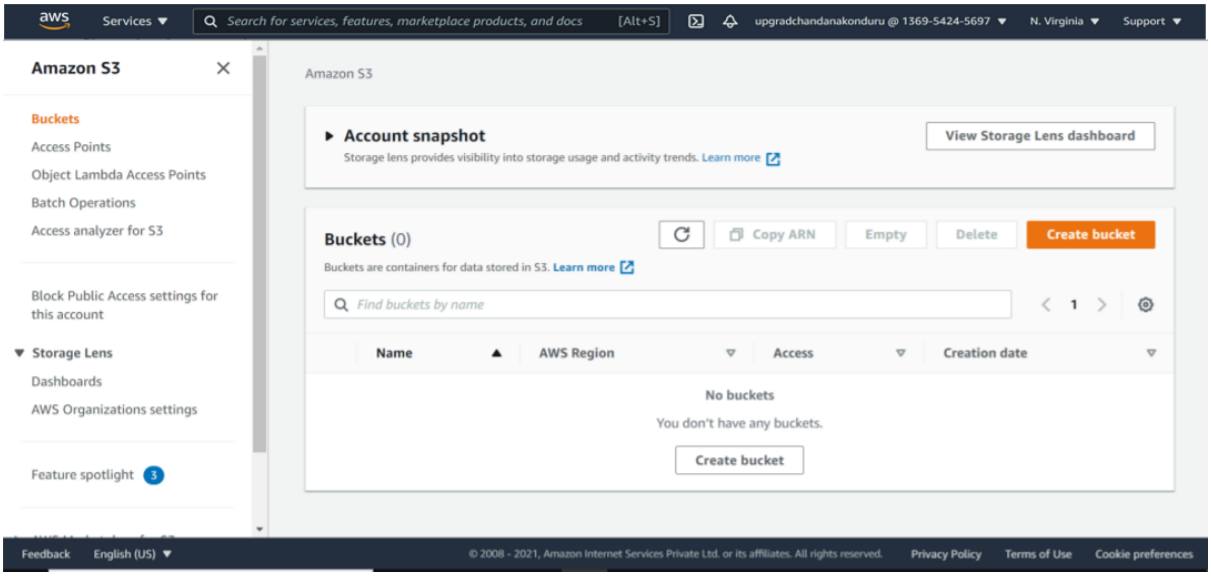
Key-pair creation

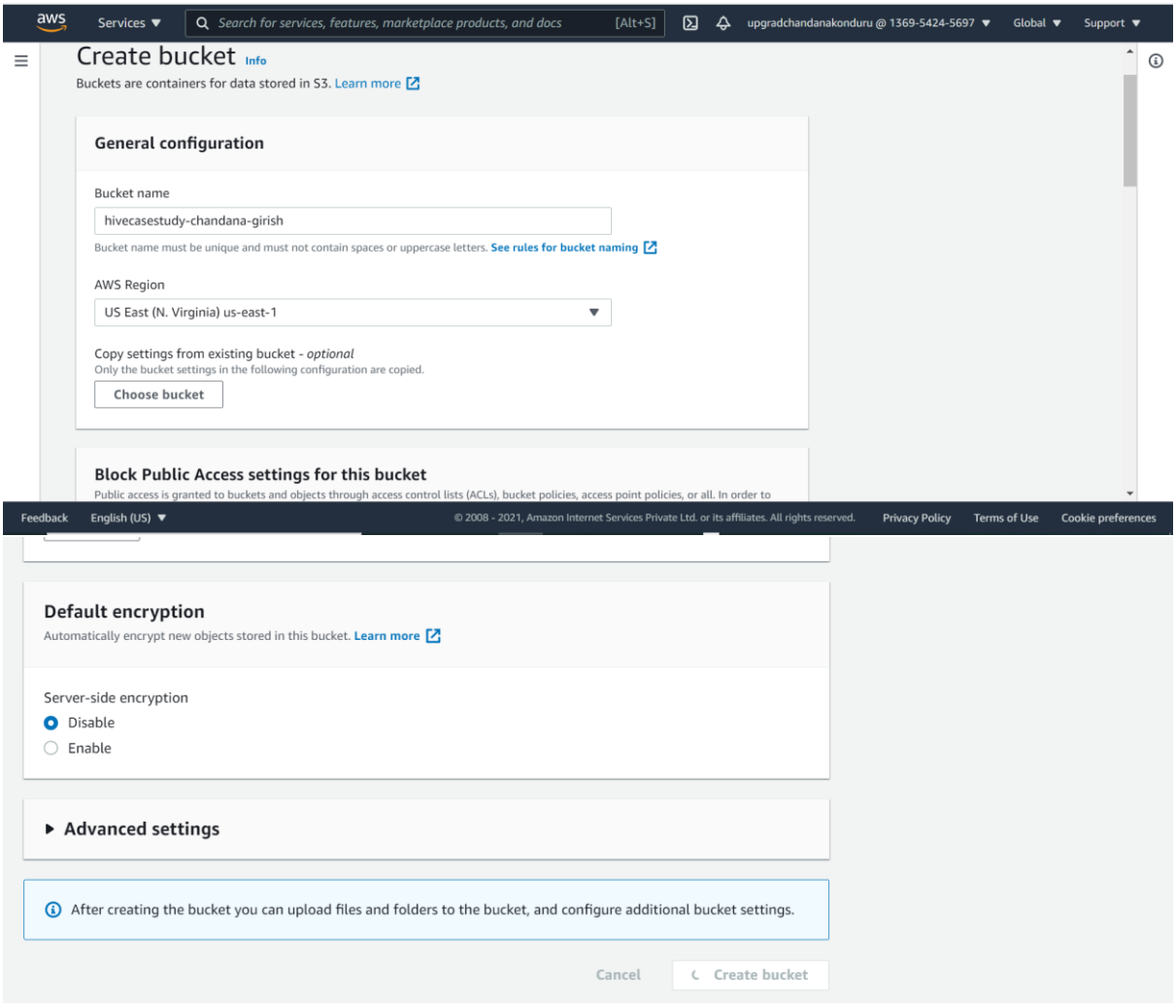


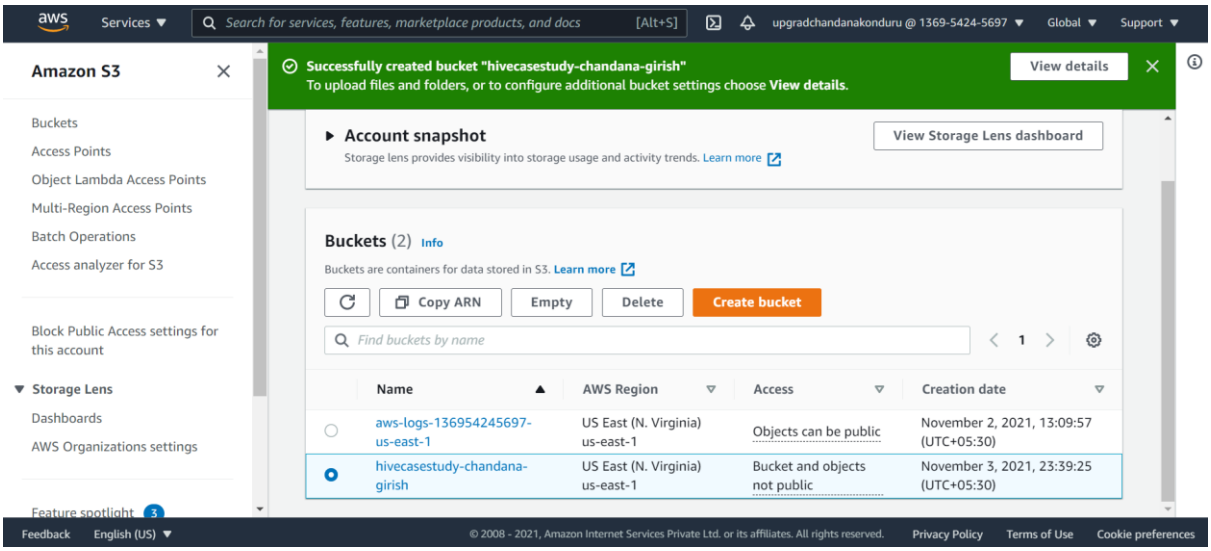


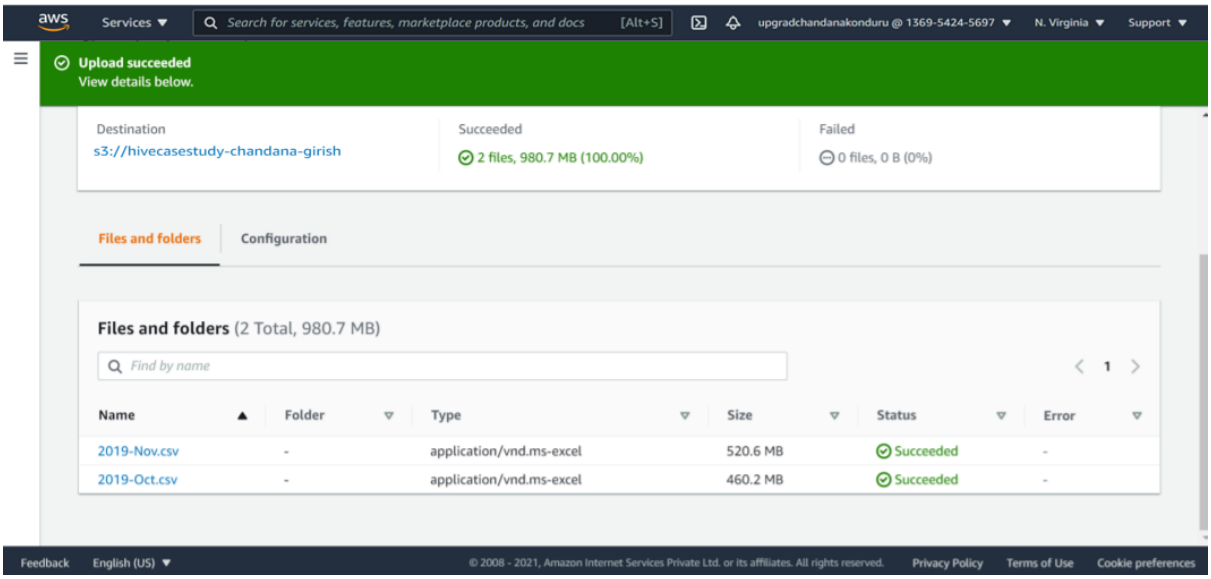
HiveCaseStudy-KeyPair successfully created and downloaded

To Store the data – Click on “Create Bucket”

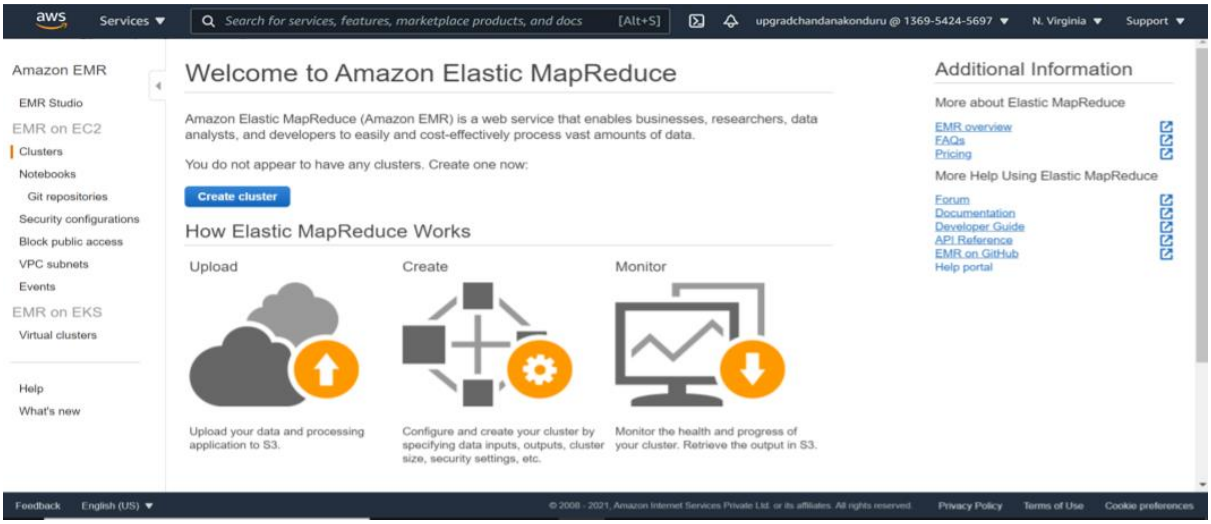




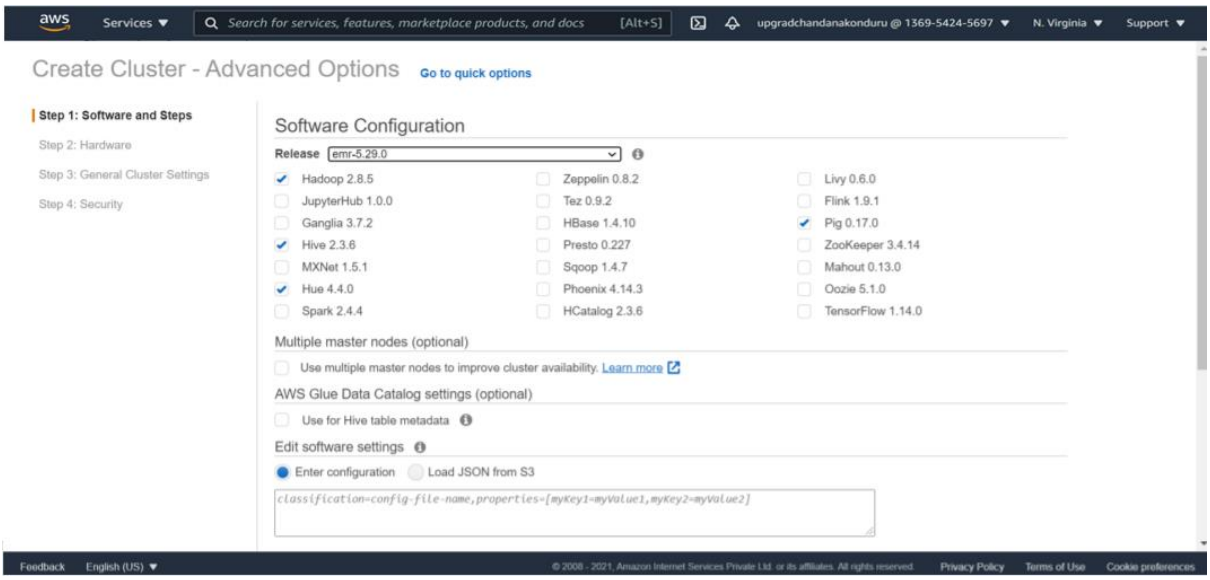




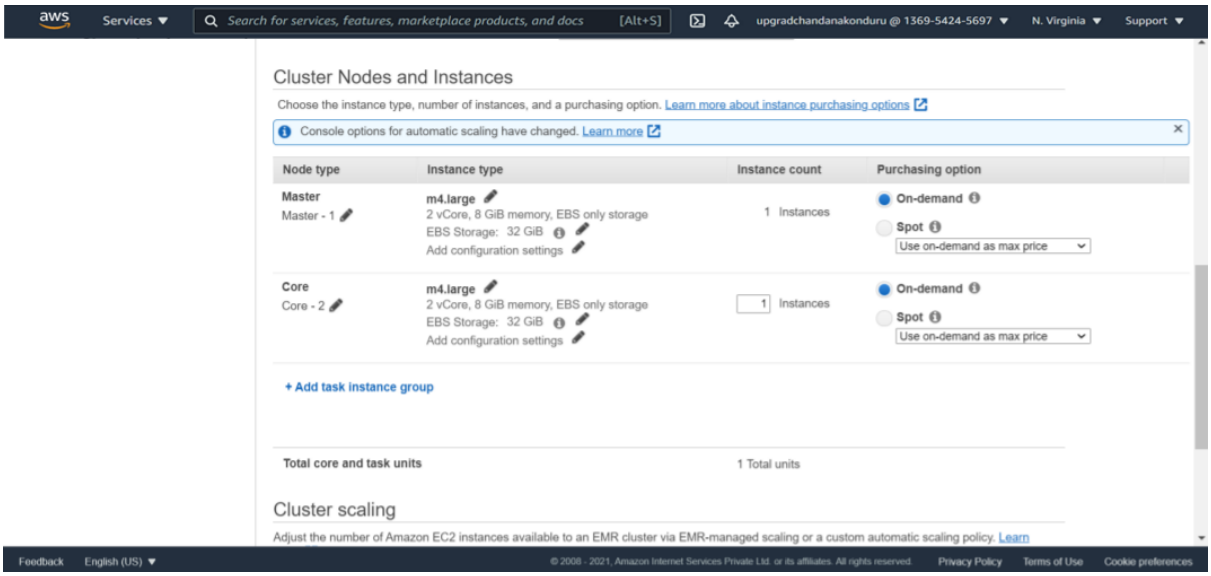
Created EMR cluster



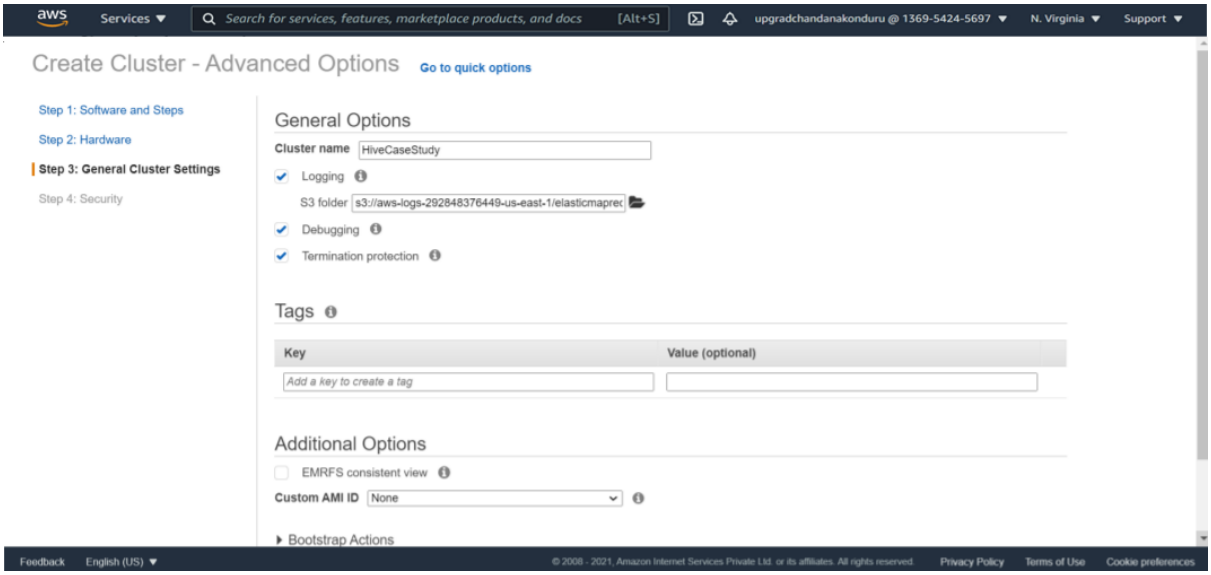
Use advance options



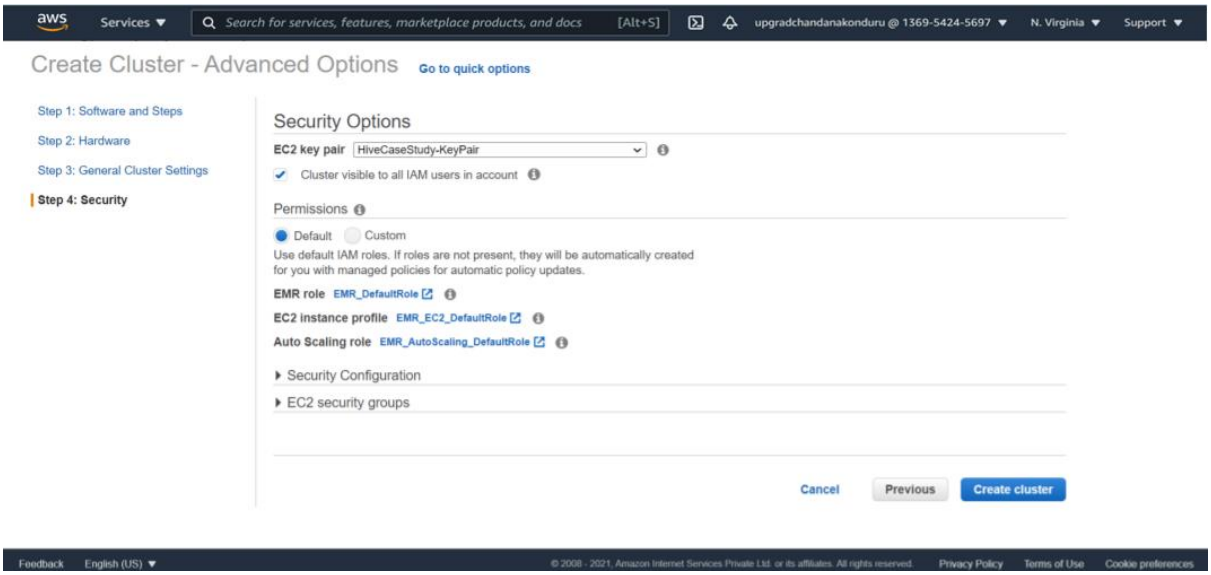
Software and Steps page: Changed the Release from emr-5.33.0 to “emr-5.29.0” Hardware page: Changing the Master and Core nodes from m5.xlarge to “m4.large”



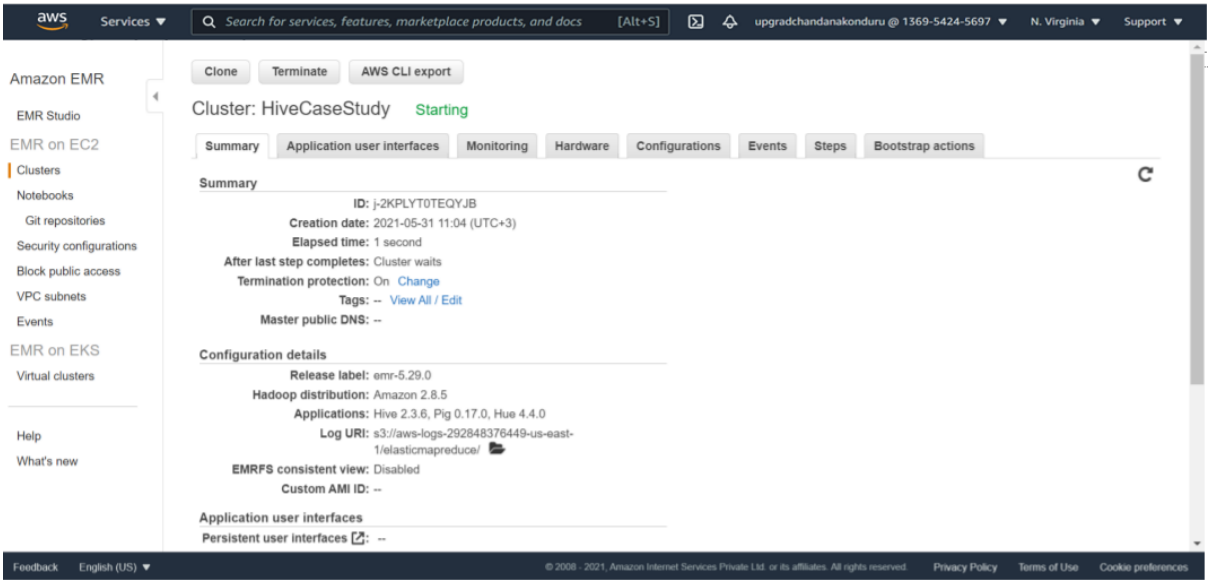
General Cluster Settings page: Giving the name to cluster “HiveCaseStudy”



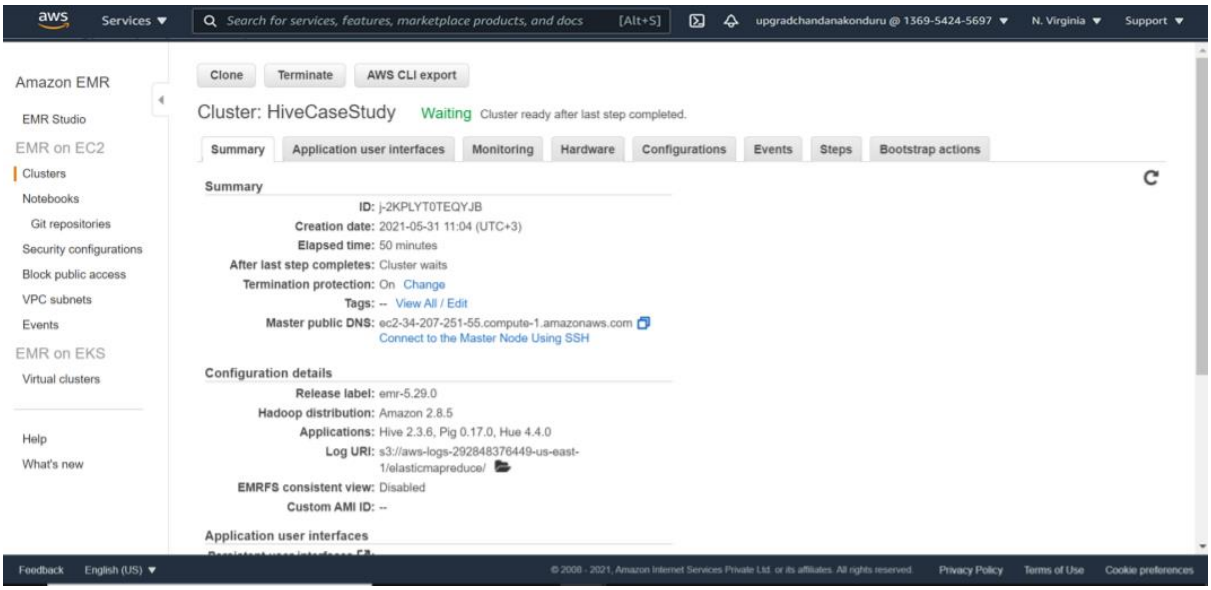
Security page: changing the EC2 key pair option to our created key pair – “HiveCaseStudy-KeyPair”

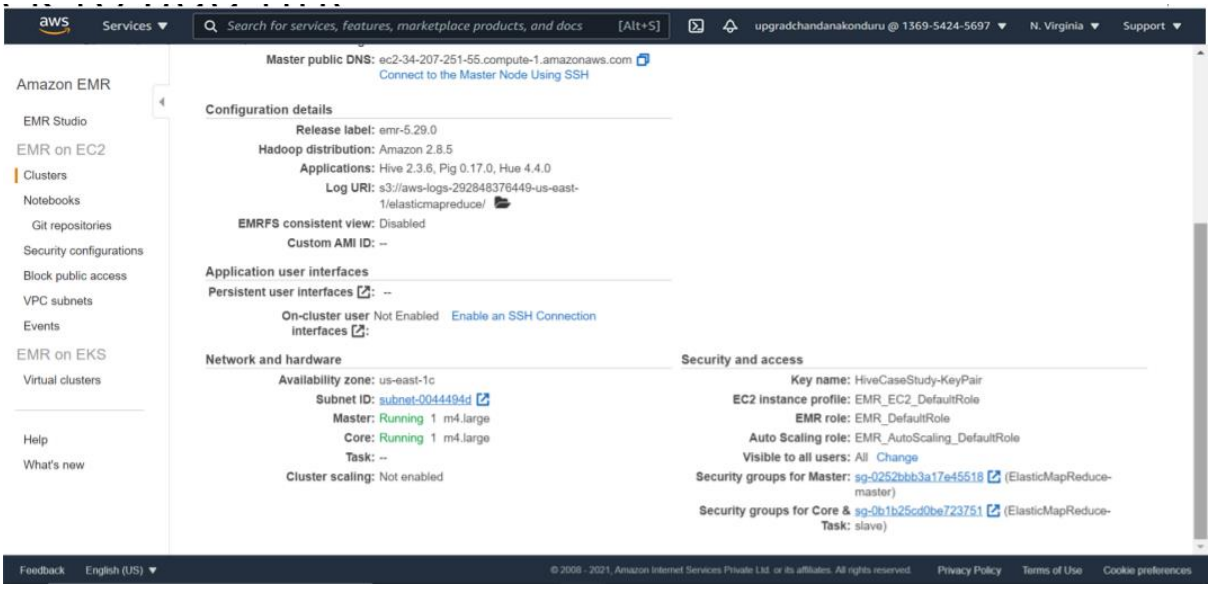


Click on “Create Cluster” button

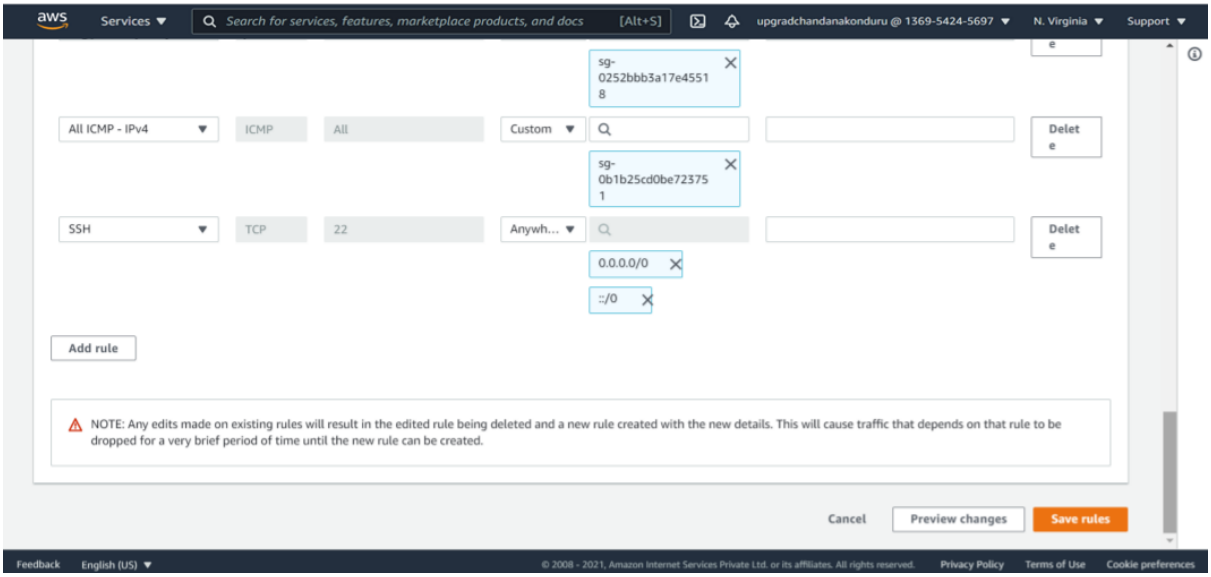


Cluster is ready with status “Waiting”





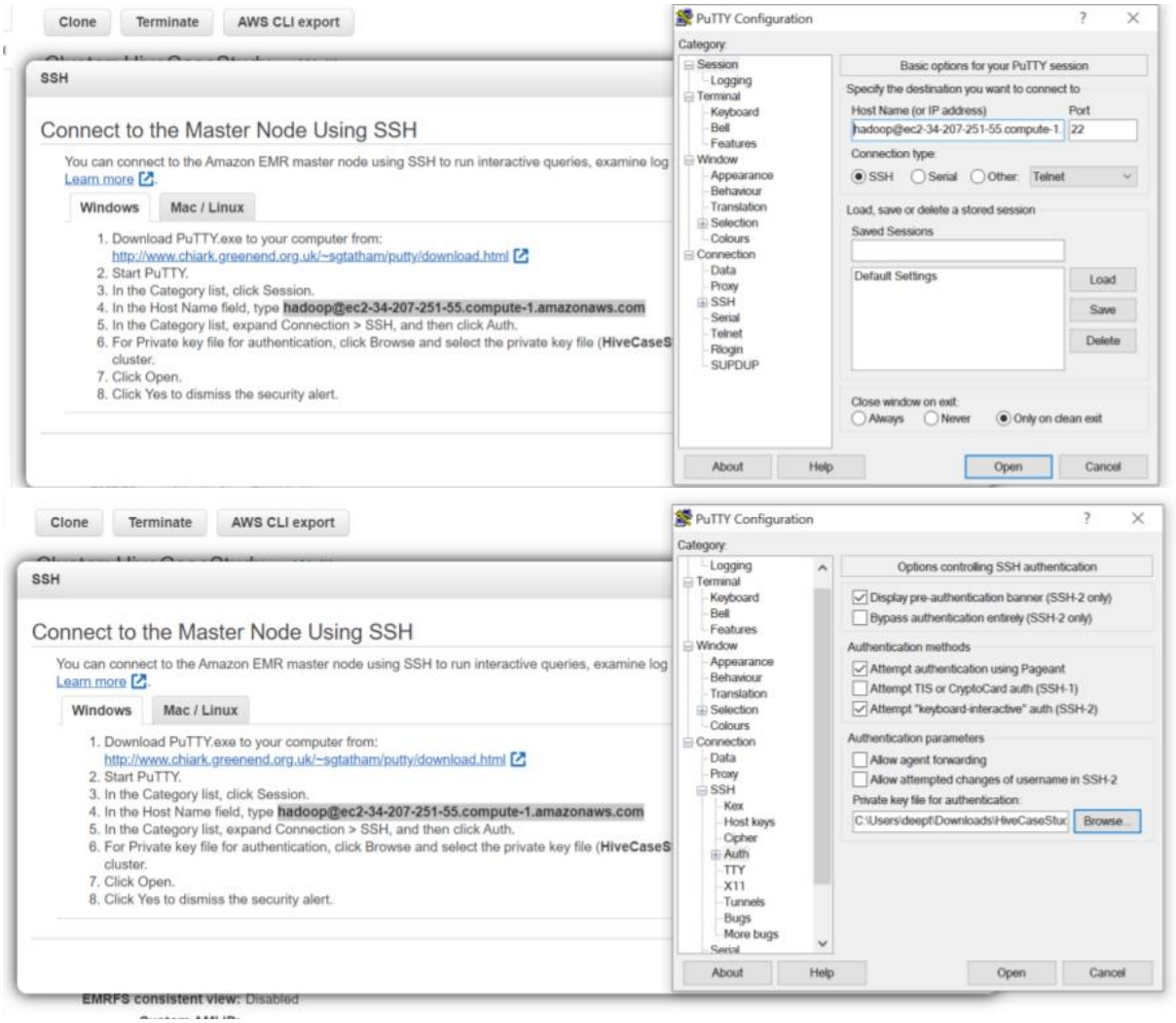
Both the Master and Core nodes are running



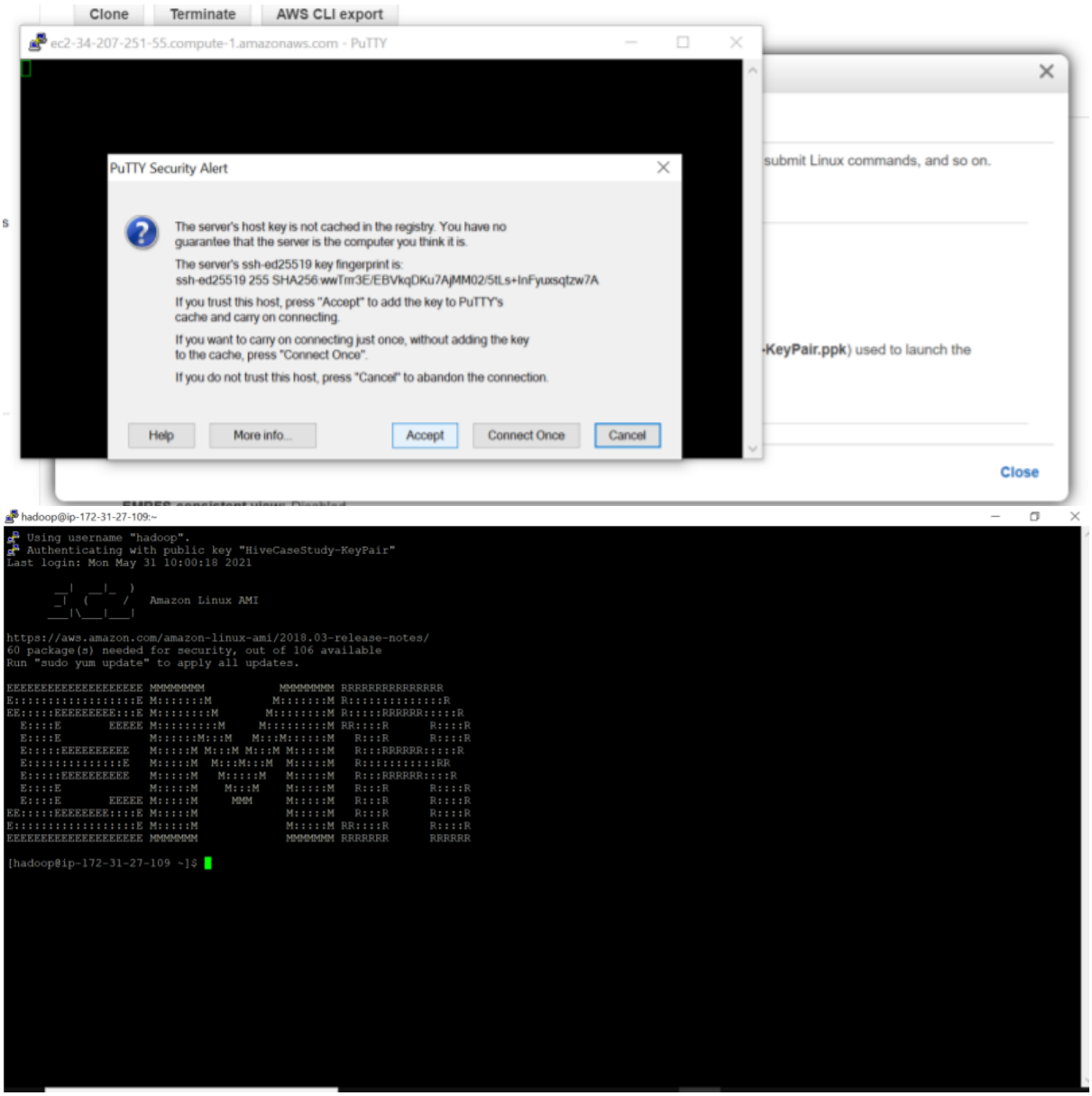
Then save the SSH rule to the inbound rules

CONNECT TO MASTER NODE:

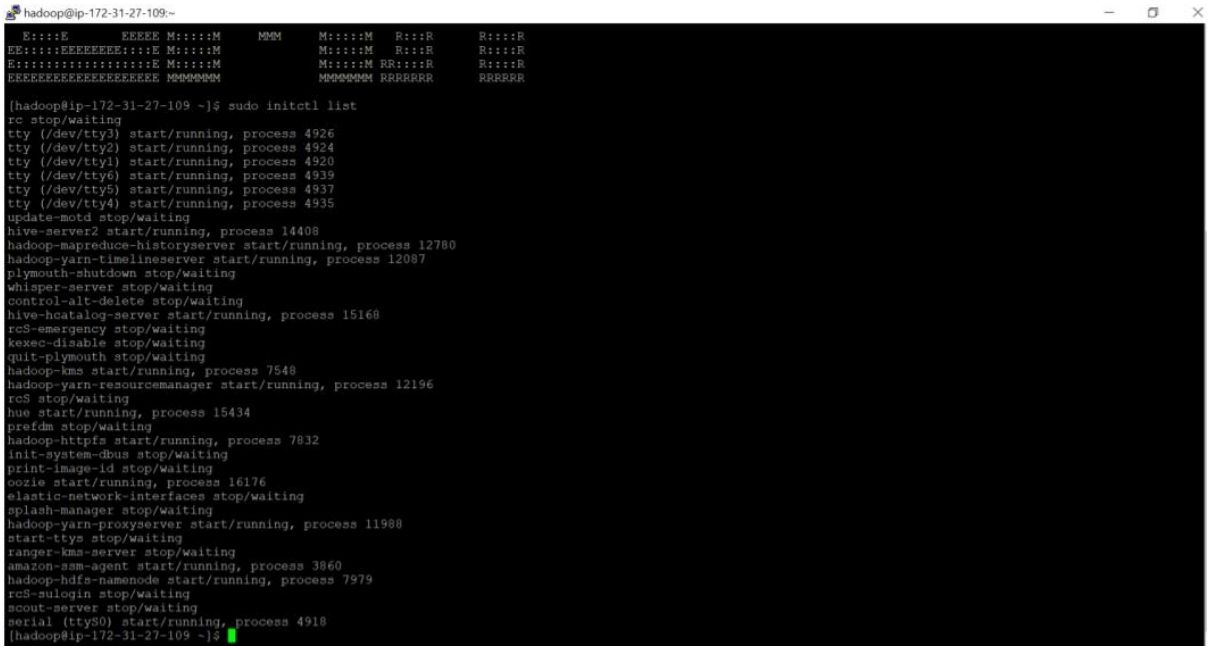
Open the putty and enter the Host Name as hadoop@ec2-34-207-251-55.compute1.amazonaws.com and navigate to Connection > SSH > Auth then browse and select the private key, which we creating initially.



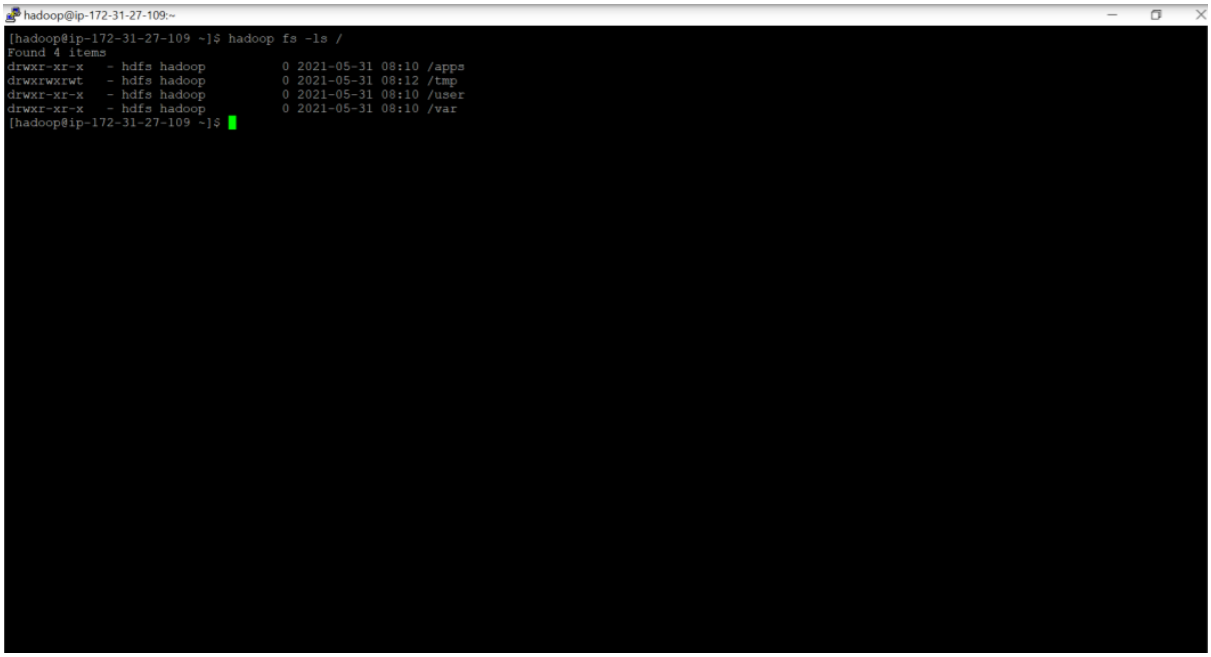
Click on “open” and then Accept the connection



EMR CLI is launched Verifying the services that are running on Hadoop cluster with command “sudo initctl list”



We can see that Hive services are running Verifying the Hadoop file system with command “hadoop fs -ls /”



All the above are inbuilt directories in HDFS.

CREATING A NEW DIRECTORY FOR HIVE CASE STUDY:

Creating a new directory under user>hive for Hive case study to store the data files and directory name creating is “hive-casestudy” and verifying whether the new directory is listed in Hadoop file system>user>hive

hadoop fs -mkdir /user/hive/hive-casestudy

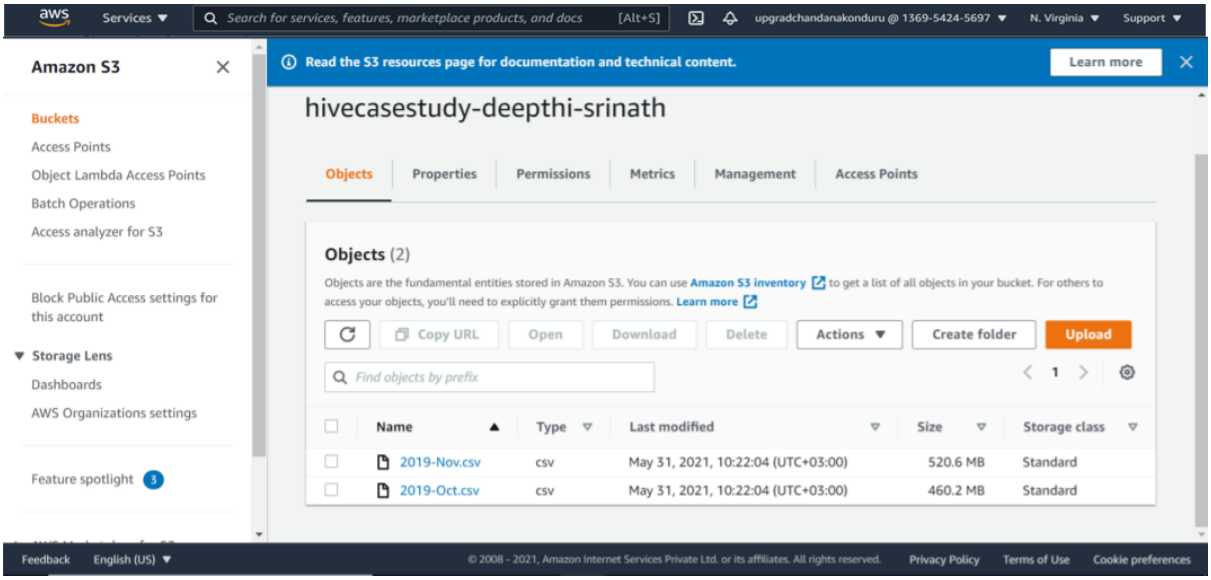
hadoop fs -ls /user/hive/

```
hadoop@ip-172-31-27-109:~
[hadoop@ip-172-31-27-109 ~]$ hadoop fs -mkdir /user/hive/hive-casestudy
[hadoop@ip-172-31-27-109 ~]$ hadoop fs -ls /user/hive/
Found 2 items
drwxr-xr-x   - hadoop hadoop          0 2021-05-31 10:50 /user/hive/hive-casestudy
drwxrwxrwt   - hdfs   hadoop          0 2021-05-31 08:10 /user/hive/warehouse
[hadoop@ip-172-31-27-109 ~]$
```

New directory is successfully created

LOADING THE DATA FROM S3 BUCKET to HDFS:

Copying the file path from S3



Distributed copy command is using to copy the data from S3 to HDFS –

For 2019 October:

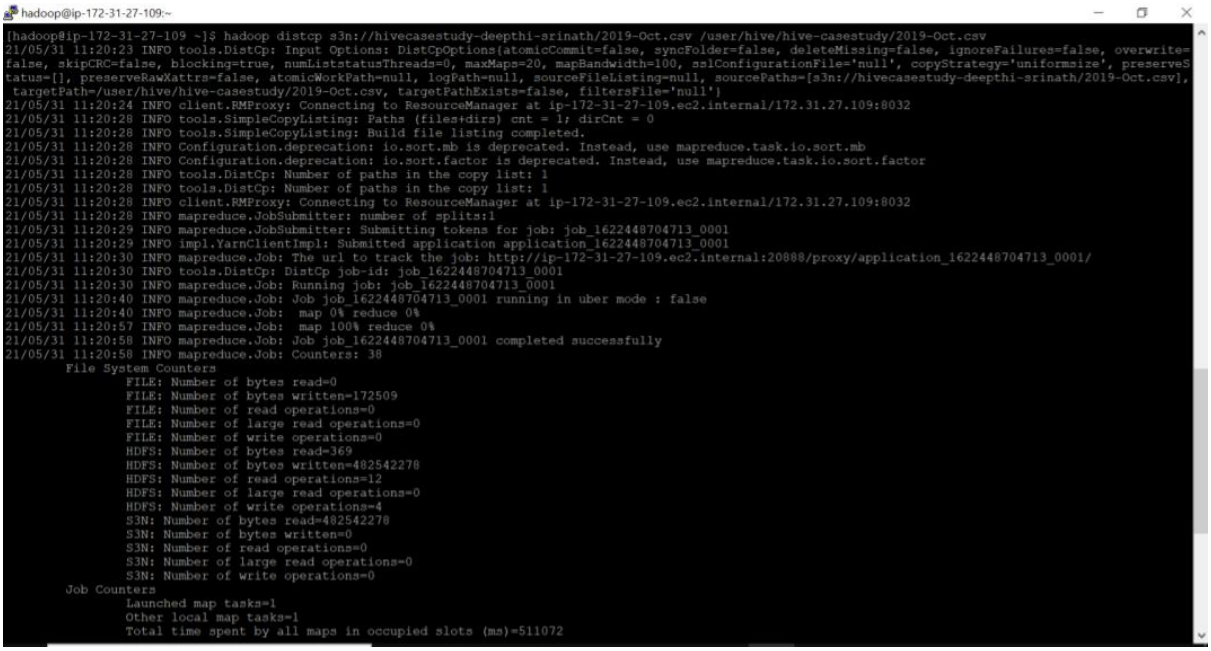
hadoop distcp s3n://hivecasestudy-deepthi-srinath/2019-Oct.csv /user/hive/hivecasestudy/2019-Oct.csv

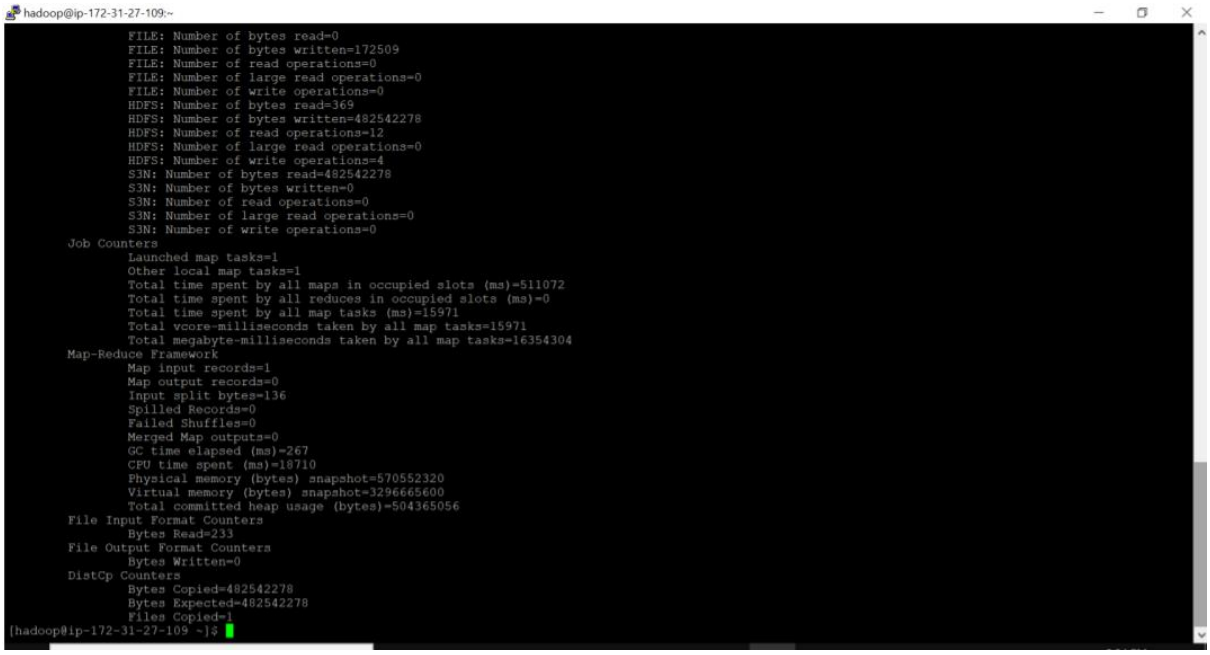
For 2019 November:

hadoop distcp s3n://hivecasestudy-deepthi-srinath/2019-Nov.csv /user/hive/hivecasestudy/2019-Nov.csv

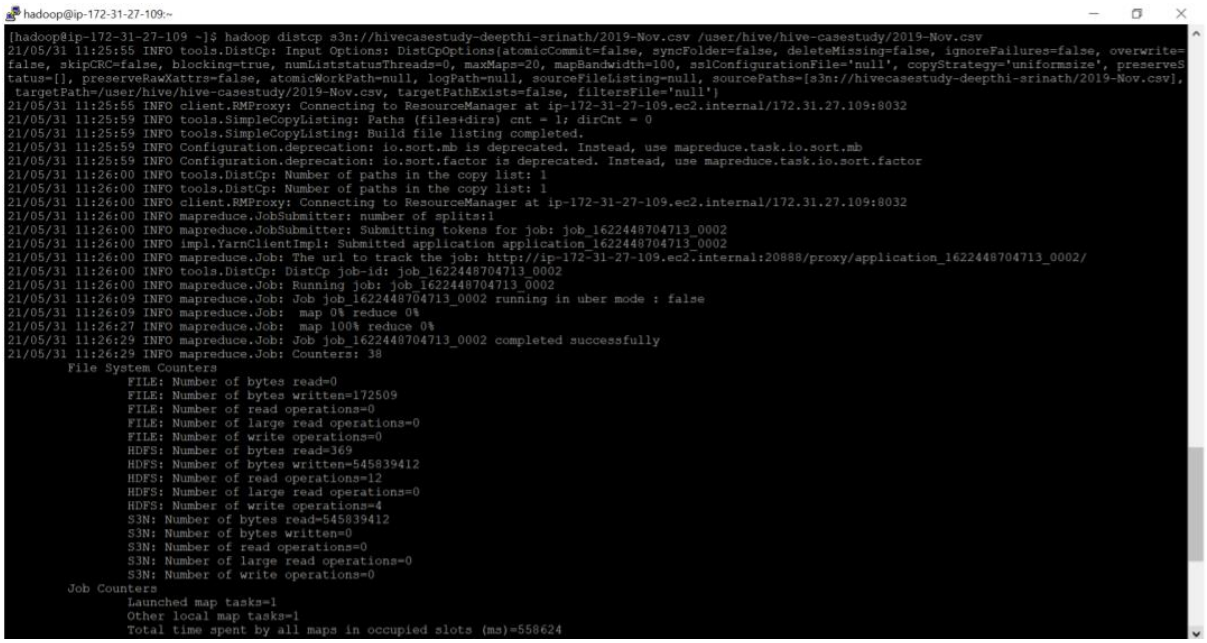
screenshots for copying October 2019 and November 2019 data individually:

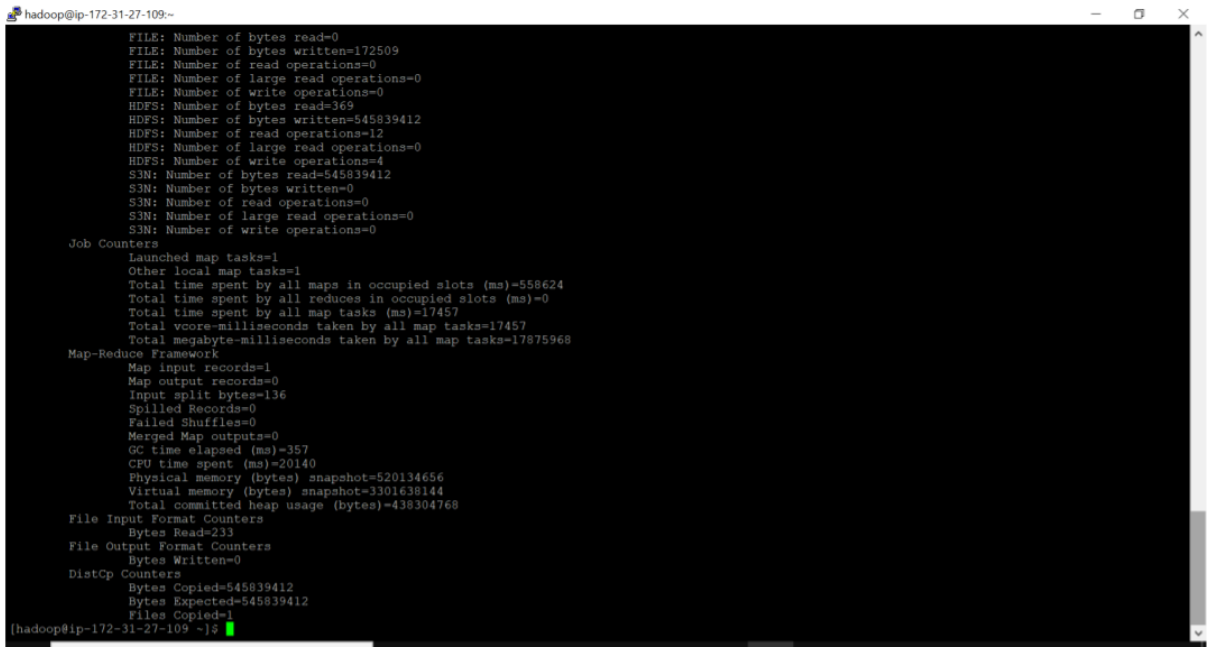
October 2019:





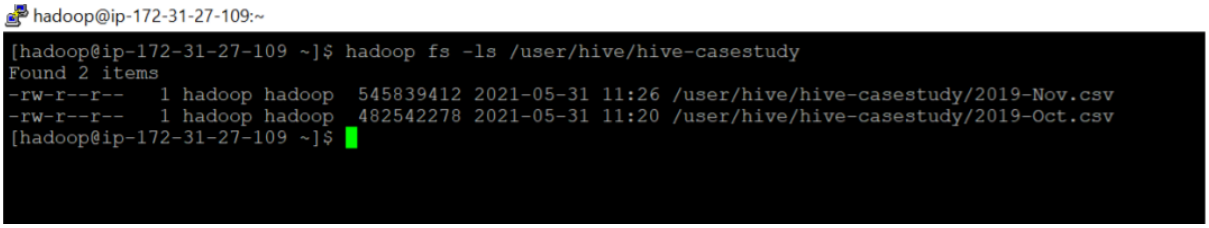
November 2019





Verifying whether the data is successfully copied into HDFS from S3 buckets

Command: hadoop fs -ls /user/hive/hive-casestudy



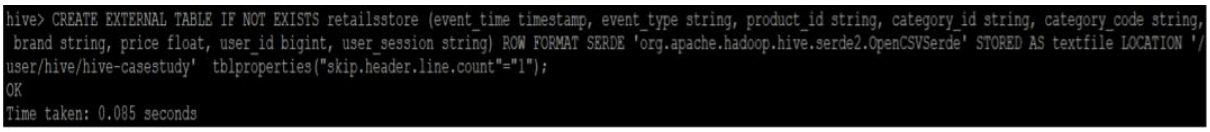
Inspecting the table data to know which columns are available before creating the hive table with command “hadoop fs -cat /user/hive/hive-casestudy/2019-Oct.csv |head” and “hadoop fs -cat /user/hive/hive-casestudy/2019-Nov.csv |head”

Both the tables are having same columns of data

Moving to hive:

CREATING AN EXTERNAL TABLE IN HIVE:

CREATE EXTERNAL TABLE IF NOT EXISTS retailsstore (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/hive-casestudy' tblproperties("skip.header.line.count"="1");



Below command is used to set the display the header columns

set hive.cli.print.header = true;

APPLYING OPTIMIZATION TECHNIQUES - PARTITIONING AND BUCKETING:

Below commands are to enable the dynamic partitioning and bucketing

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

hive> set hive.exec.dynamic.partition = true;

hive> set hive.enforce.bucketing = true;

```
hive> set hive.exec.dynamic.partition.mode = nonstrict;
hive> set hive.exec.dynamic.partition = true;
hive> set hive.enforce.bucketing = true;
hive>
```

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

CREATE TABLE IF NOT EXISTS dynpart_buck_retailsstore(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.serde

2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hive-casestudy' tblproperties('skip.header.line.count' = '1');

```
hive> CREATE TABLE IF NOT EXISTS dynpart_buck_retailsstore(event_time timestamp, product_id string, category_id string, category_code string, brand string, p
rice 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/hive-casestudy'
    > tblproperties('skip.header.line.count' = '1');
OK
Time taken: 0.061 seconds
hive>
```

Verifying the created table

```
hive> show tables;
OK
tab_name
dynpart_buck_retailsstore
retailsstore
Time taken: 0.021 seconds, Fetched: 2 row(s)
hive>
```

INSERTING THE DATA INTO NEWLY CREATED OPTIMIZED TABLE (dynpart_buck_retailsstore) FROM EXISTING TABLE(retailsstore):

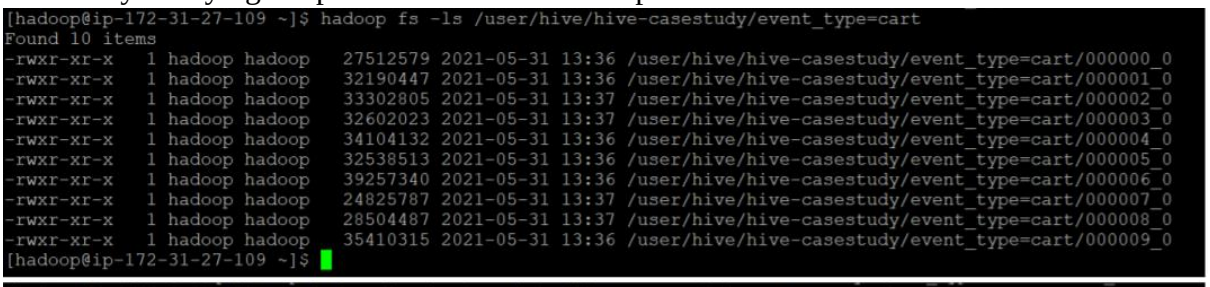
INSERT INTO TABLE dynpart_buck_retailsstore PARTITION (event_type) SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type FROM retailsstore;

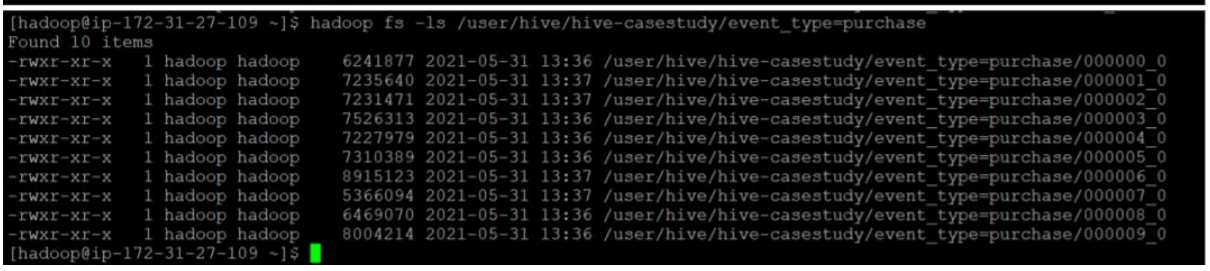
```
hive> INSERT INTO TABLE dynpart_buck_retailsstore
    > PARTITION (event_type)
    > SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type
    > FROM retailsstore;
Query ID = hadoop_20210531133455_970f6238-6993-4be5-a751-93e4bad61d78
Total jobs = 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_1622448704713_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: 168.08 s
--------------------------------------------------------------------------------
Loading data to table default.dynpart_buck_retailsstore partition (event_type=null)

Loaded : 4/4 partitions.
         Time taken to load dynamic partitions: 0.41 seconds
         Time taken for adding to write entity : 0.007 seconds
OK
event_time      product_id      category_id     category_code   brand   price   user_id user_session    event_type
Time taken: 177.824 seconds
hive>
```

Output: Based on the above results, it partitioned into 4

Verifying the partitioned in the Hadoop file system

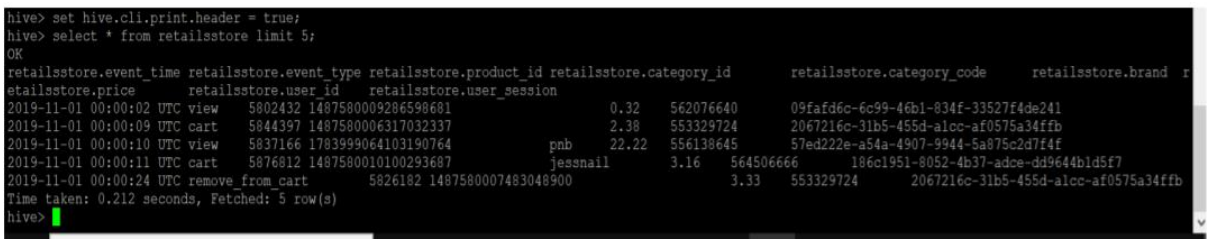
Randomly verifying the partitioned data in hadoop





VERIFYING THE PERFORMANCE OF BOTH THE TABLES – BEFORE AND AFTER OPTIMIZED TECHNIQUES:

select * from retailsstore limit 5;



Time taken to retrieve first 5 rows of data before optimization is 0.212 seconds (above)

select * from dynpart_buck_retailsstore limit 5;

```
hive> set hive.cli.print.header = true;
hive> select * from dynpart_buck_retailsstore limit 5;
OK
dynpart_buck_retailsstore.event_time    dynpart_buck_retailsstore.product_id    dynpart_buck_retailsstore.category_id   dynpart_buck_retailsstore.category_co
de      dynpart_buck_retailsstore.brand dynpart_buck_retailsstore.price dynpart_buck_retailsstore.user_id       dynpart_buck_retailsstore.user_session  dynpa
rt_buck_retailsstore.event_type
2019-10-08 09:19:19 UTC 89350   1487580011652186237             runail  1.27    232701853       3f1469f5-d926-44ce-a9f6-dff5ae276c9c    cart
2019-10-10 05:29:47 UTC 5866208 1487580013841613016             concept 3.16    493381333       535bb6b7-08f4-4021-ac66-b340178f7a37    cart
2019-10-08 12:25:50 UTC 5821183 1487580007717929935                     1.27    546703849       3daf4d64-5ffa-46cc-827b-59760ebd819b    cart
2019-10-10 08:19:06 UTC 5848901 1487580007675986893             bpw.style       1.27    439370683       9aeb4d9a-1bed-4f42-b12d-88be1148d3a9    cart
2019-10-09 18:32:50 UTC 5869152 1487580005268456287             cosmoprofi      7.94    558533352       cfde0f74-8705-4a2f-ba83-a5b99581c294    cart
Time taken: 0.185 seconds, Fetched: 5 row(s)
hive>
```

Time taken to retrieve the first 5 rows of data after optimization is 0.185 seconds (above screenshot)

GIVEN QUESTIONS:

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

Base table:

SELECT SUM(price) AS tot_revenue_oct FROM retailsstore WHERE MONTH(event_time) = ‘10’ AND event_type = ‘purchase’;

```
hive> SELECT SUM(price) AS tot_revenue_oct FROM retailsstore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';
Query ID = hadoop_20210531123157_a77aa0ec-fc06-4a02-a121-1222590c3741
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1622448704713_0004)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..... container     SUCCEEDED      2          2        0        0       0       0
Reducer 2 ..... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 69.72 s
----------------------------------------------------------------------------------------------
OK
tot_revenue_oct
1211538.4299997438
Time taken: 70.448 seconds, Fetched: 1 row(s)
hive>
```

Time taken is 70.448 seconds

Optimized table:

SELECT SUM(price) AS tot_revenue_oct FROM dynpart_buck_retailsstore WHERE MONTH(event_time) = 10 AND event_type = ‘purchase’;

```
hive> SELECT SUM(price) AS tot_revenue_oct FROM dynpart_buck_retailsstore WHERE MONTH(event_time) = 10 AND event_type = 'purchase';
Query ID = hadoop_20210531143211_c0a7491d-6e53-457c-a681-b3373f3e7e72
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_1622448704713_0006)

----------------------------------------------------------------------------------------------
        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: 24.67 s
----------------------------------------------------------------------------------------------
OK
tot_revenue_oct
1211532.4500002791
Time taken: 33.181 seconds, Fetched: 1 row(s)
hive>
```

Time taken with optimized table is 33.181 seconds Insights:

1. The total revenue generated based on Purchase made in the month of October is 1,211,538.43 /-

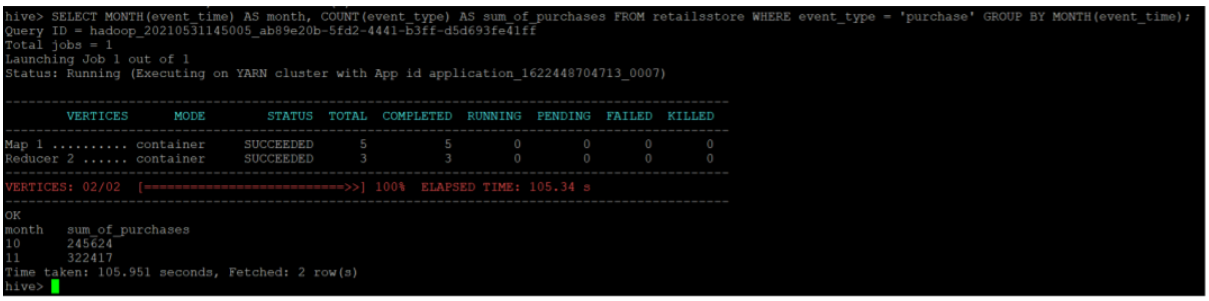
2. Non-optimized table query took the execution time of 70.448 seconds whereas optimized table query took execution time of 33.181 seconds. We can see there is a significant drop in the execution time of the same query.

3. Hence, 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 Query:

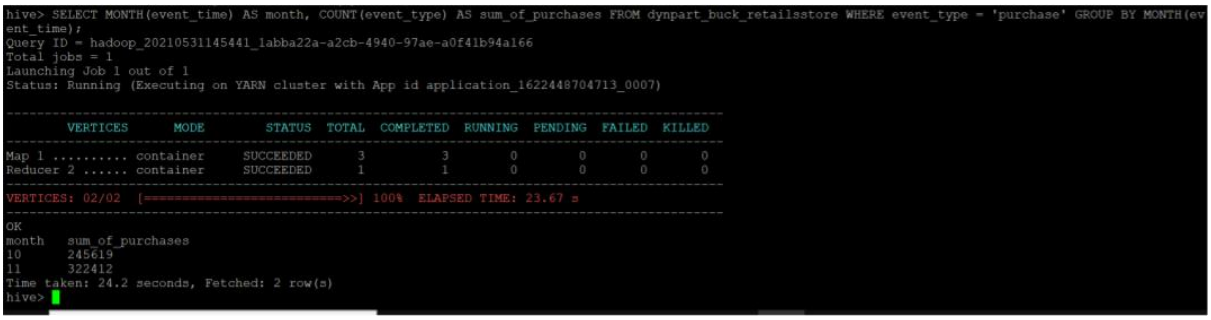
SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM retailsstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);



Time taken is 105.951 seconds

Optimized table:

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



Time taken is 24.2 seconds

Insights:

1. Sum of purchases made in the month of October is 245624 and in the month of November 322417, which means number of purchases are increased in November month

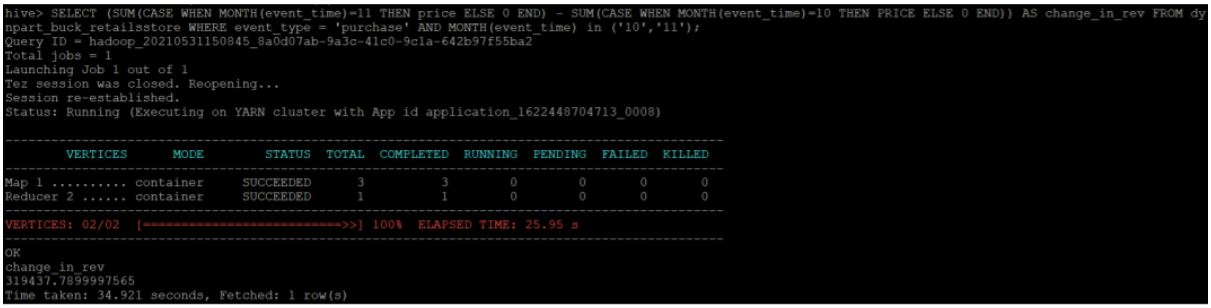
2. Non-optimized table query took the execution time of 105.951 seconds whereas optimized table query took execution time of 24.2 seconds. We can see there is a significant drop in the execution time of the same query.

3. Hence, with proper partitioning and bucketing on table we can reduce execution time.

Using Optimized table from below questions onwards:

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

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 change_in_rev FROM dynpart_buck_retailsstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11');



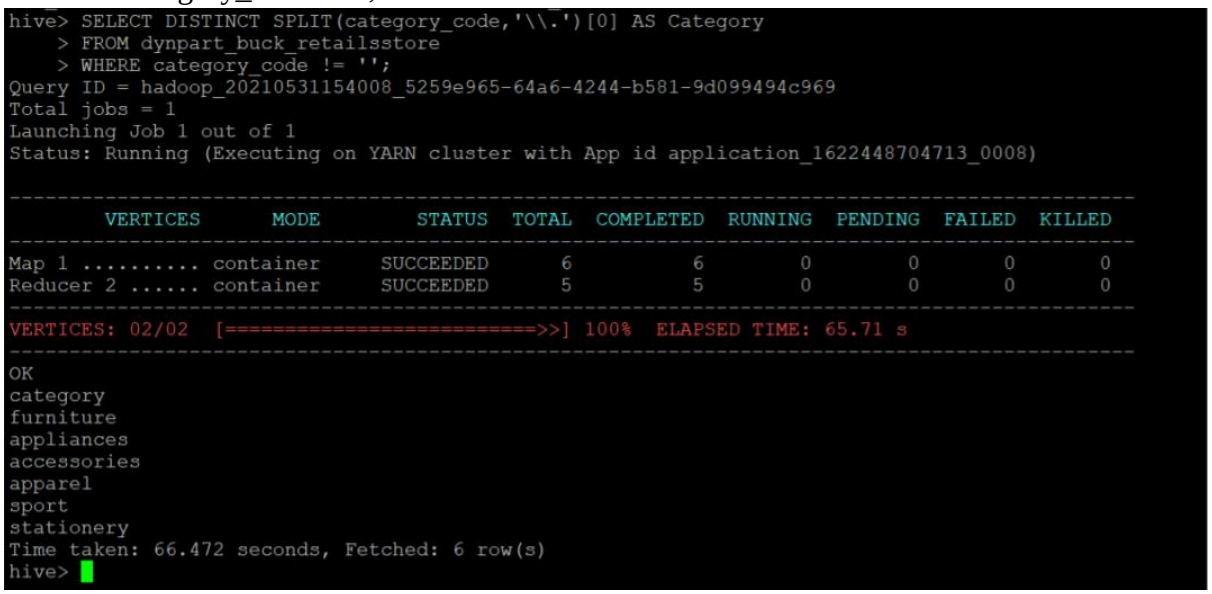
Insights:

1. Time taken to execute the query is 34.921 seconds

2. Revenue increased in November by 319437.789 from October

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

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



Insights:

1. Time taken to execute the query is 66.472 seconds
2. Total we got 6 distinct categories are – furniture, appliances, accessories, apparel,sport, stationary.

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

SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS num_of_prodFROM

dynpart_buck_retailsstore

WHERE category_code != ''

```
GROUP BY SPLIT(category_code,'\\.')[0]
ORDER BY num_of_prod DESC;
```

```
hive> SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS num_of_prod
    > FROM dynpart_buck_retailsstore
    > WHERE category_code != ''
    > GROUP BY SPLIT(category_code,'\\.')[0]
    > ORDER BY num_of_prod DESC;
Query ID = hadoop_20210531154545_6b1e9ab2-632e-48df-a837-caf8d76d0406
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1622448704713_0008)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 66.49 s
----------------------------------------------------------------------------------------------
OK
category        num_of_prod
appliances      61736
stationery      26722
furniture       23604
apparel 18232
accessories     12928
sport   2
Time taken: 67.111 seconds, Fetched: 6 row(s)
hive>
```

Insights:

a. Time taken to execute the query is 67.111 seconds
b. Appliances are having highest number of products available with 61736 compared to other categories.
c. Stationary and Furniture categories are almost equally registered with availableranges from 23000 to 27000.
d. Sports category is least available with 2 products.

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

WITH tot_sales AS(

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

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

```
hive> WITH tot_sales AS(
    > SELECT brand, (SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) + SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)) AS total_sales
    > FROM dynpart_buck_retailsstore
    > WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != ''
    > GROUP BY brand)
    > SELECT brand, total_sales
    > FROM tot_sales
    > ORDER BY total_sales DESC
    > LIMIT 1;
Query ID = hadoop_20210531155143_f7fa1dff-3002-4826-bf02-b56a4e73f8fb
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1622448704713_0008)

----------------------------------------------------------------------------------------------
        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: 27.25 s
----------------------------------------------------------------------------------------------
OK
brand   total_sales
runail  148292.46000001638
Time taken: 27.83 seconds, Fetched: 1 row(s)
hive>
```

Insights:

a. Runail is the brand that has the highest sales in total of both the months October andNovember
b. It seems that Runail brand has high popularity among cosmetic lovers and bringingin more
c. products related to Runail brand could help in increasing their profit.

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

WITH brand_sales AS(

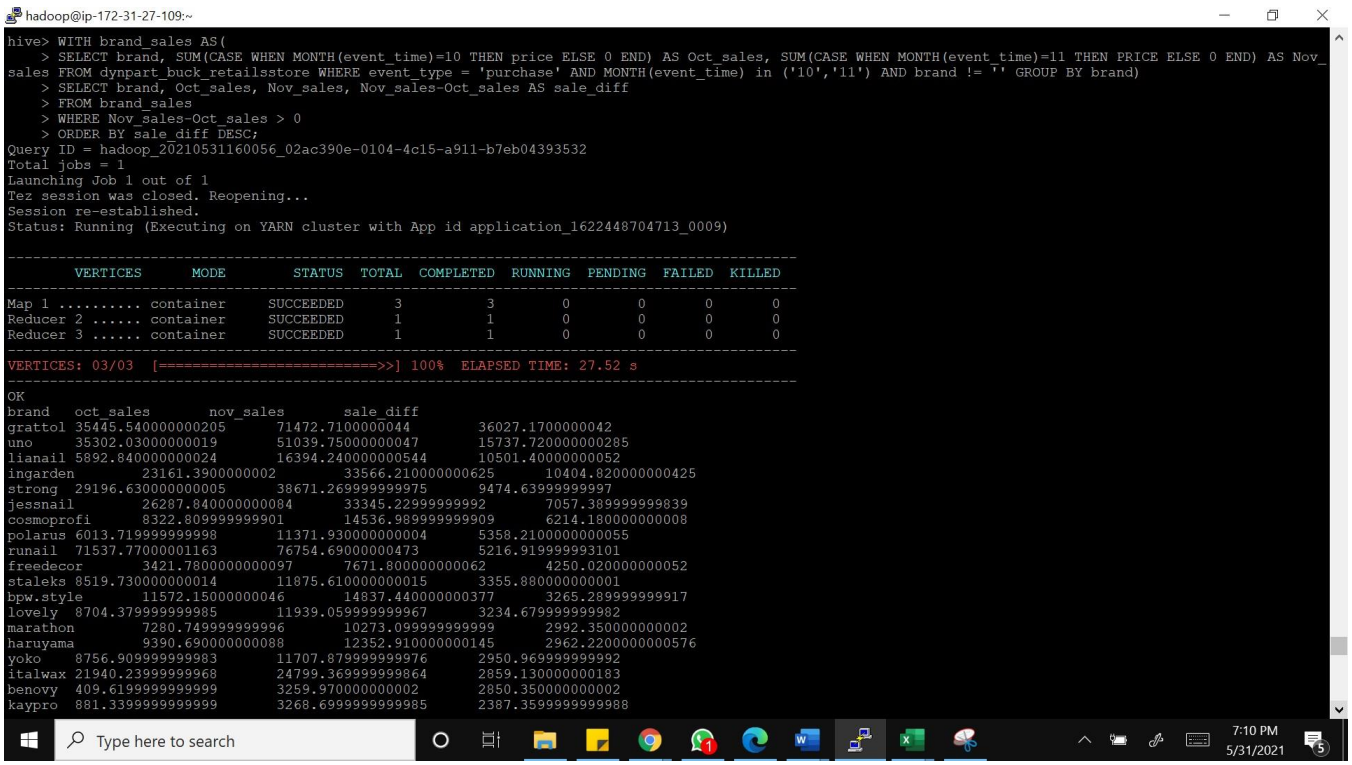
SELECT brand, SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) AS Oct_sales, SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END) AS Nov_sales FROM dynpart_buck_retailsstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)

SELECT brand, Oct_sales, Nov_sales, Nov_sales-Oct_sales AS sale_diffFROM

brand_sales

WHERE Nov_sales-Oct_sales > 0

ORDER BY sale_diff DESC;


hadoop@ip-172-31-27-109:~
hive> WITH brand_sales AS(
> SELECT brand, SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) AS Oct_sales, SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END) AS Nov_sales FROM dynpart_buck_retailsstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)
> SELECT brand, Oct_sales, Nov_sales, Nov_sales-Oct_sales AS sale_diff
> FROM brand_sales
> WHERE Nov_sales-Oct_sales > 0
> ORDER BY sale_diff DESC;
Query ID = hadoop_20210531160056_02ac390e-0104-4c15-a911-b7eb04393532
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_1622448704713_0009)
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: 27.52 s
OK
brand oct_sales nov_sales sale_diff
grattol 35445.540000000205 71472.7100000044 36027.1700000042
uno 35302.03000000019 51039.75000000047 15737.720000000285
lianail 5892.840000000024 16394.240000000544 10501.40000000052
ingarden 23161.3900000002 33566.210000000625 10404.820000000425
strong 29196.630000000005 38671.269999999975 9474.63999999997
jessnail 26287.840000000084 33345.22999999992 7057.389999999839
cosmoprofi 8322.809999999901 14536.989999999909 6214.180000000008
polarus 6013.719999999998 11371.930000000004 5358.2100000000055
runail 71537.77000001163 76754.69000000473 5216.919999993101
freedecor 3421.7800000000097 7671.800000000062 4250.020000000052
staleks 8519.730000000014 11875.610000000015 3355.880000000001
bpw.style 11572.15000000046 14837.440000000377 3265.289999999917
lovely 8704.379999999985 11939.059999999967 3234.679999999982
marathon 7280.749999999996 10273.099999999999 2992.350000000002
haruyama 9390.690000000088 12352.910000000145 2962.2200000000576
yoko 8756.909999999983 11707.879999999976 2950.969999999992
italwax 21940.23999999968 24799.369999999864 2859.130000000183
benovy 409.6199999999999 3259.970000000002 2850.350000000002
kaypro 881.3399999999999 3268.6999999999985 2387.3599999999988
Type here to search
7:10 PM
5/31/2021

hadoop@ip-172-31-27-109:~

```
kaypro  881.3399999999999       3268.6999999999985      2387.3599999999988
estel   21756.750000000025      24142.670000000027      2385.920000000002
concept 11032.139999999978      13380.399999999978      2348.26
kapous  11927.159999999738      14093.079999999864      2165.9200000001256
f.o.x   6624.230000000007       8577.279999999979       1953.049999999972
masura  31266.079999999318      33058.46999999992       1792.3900000006033
milv    3904.940000000026       5642.0100000001285      1737.0700000001025
beautix 10493.94999999997       12222.95        1729.000000000031
artex   2730.640000000002       4327.249999999996       1596.6099999999942
domix   10472.04999999992       12009.169999999851      1537.1199999999317
shik    3341.2000000000035      4839.720000000001       1498.5199999999977
smart   4457.2599999999875      5902.139999999998       1444.88000000001
roubloff        3491.3600000000006      4913.770000000003       1422.4100000000026
levrana 2243.560000000002       3664.100000000004       1420.5400000000018
oniq    8425.409999999947       9841.64999999998        1416.2400000000325
irisk   45591.95999999969       46946.03999999916       1354.0799999994633
severina        4775.879999999955       6120.479999999956       1344.6000000000013
joico   705.52  2015.1000000000006      1309.5800000000006
zeitun  708.6600000000003       2009.63 1300.9699999999998
beauty-free     554.1700000000014       1782.8599999999983      1228.6899999999969
swarovski       1887.9299999999898      3043.1599999999794      1155.2299999999896
de.lux  1659.6999999999784      2775.509999999973       1115.8099999999945
metzger 5373.45000000001        6457.160000000005       1083.7099999999955
markell 1768.7499999999998      2834.429999999994       1065.6799999999942
sanoto  157.14  1209.6799999999998       1052.54
nagaraku        4369.740000000042       5327.6800000000285      957.9399999999869
ecolab  262.85  1214.3000000000009       951.4500000000008
art-visage      2092.7099999999978      2997.8000000000056      905.0900000000079
levissime       2227.5000000000064      3085.3099999999854      857.809999999979
missha  1293.83 2150.2799999999997      856.4499999999998
solomeya        1899.7000000000012      2685.799999999996       786.0999999999949
rosi    3077.040000000002       3841.560000000001       764.5199999999991
refectocil      2716.180000000003       3475.5800000000036      759.4000000000005
kaaral  4412.429999999999       5086.07 673.6400000000003
kosmekka        1181.4399999999996      1813.37 631.9300000000003
kinetics        6334.250000000006       6945.259999999998       611.009999999992
browxenna       14331.370000000057      14916.73000000007       585.3600000000133
airnails        5118.900000000015       5691.520000000021       572.6200000000063
uskusi  5142.26999999998        5690.310000000031       548.0400000000509
coifin  903.0000000000002       1428.4899999999998      525.4899999999996
s.care  412.68  913.07  500.39000000000004
limoni  1308.9000000000005      1796.6000000000004      487.6999999999998
matrix  3243.250000000001       3726.7400000000016      483.4900000000007
gehwol  1089.0700000000002      1557.68 468.6099999999999
```

hadoop@ip-172-31-27-109:~

```
greymy  29.21   489.49  460.28000000000003
bioaqua 942.8900000000001       1398.1200000000001      455.23
farmavita       837.37  1291.97 454.6
sophin  1067.8600000000006      1515.520000000002       447.66000000000145
yu-r    271.41  673.7099999999999       402.2999999999999
kiss    421.55  817.3300000000004       395.78000000000037
naomi   0.0     389.0   389.0
lador   2083.610000000002       2471.5300000000016      387.9199999999996
ellips  245.84999999999997      606.04  360.19
jas     3318.960000000002       3657.4300000000026      338.4700000000007
lowence 242.83999999999997      567.7499999999999       324.9099999999999
nitrile 847.279999999999        1162.679999999999       315.4
shary   871.9599999999994       1176.4899999999996      304.5300000000002
kims    330.04  632.0400000000001       302.00000000000006
happyfons       801.9200000000004       1091.5900000000008      289.6700000000004
kocostar        310.8499999999999       594.9299999999998       284.0799999999999
insight 1443.7000000000005      1721.960000000001       278.26000000000045
candy   534.9599999999999       799.3799999999994       264.4199999999995
bluesky 10307.240000000156      10565.529999999784      258.289999999628
beauugreen      511.51000000000016      768.3499999999999       256.83999999999975
protokeratin    201.25  456.79  255.54000000000002
trind   298.07000000000005      542.96  244.89
entity  479.71000000000157      719.2599999999991       239.5499999999975
skinlite        651.9399999999997       890.4499999999998       238.5100000000001
provoc  827.9900000000004       1063.8200000000024      235.8300000000021
fedua   52.38   263.81  211.43
ecocraft        41.160000000000004      241.94999999999996      200.78999999999996
keen    236.35  435.61999999999995      199.26999999999995
mane    66.78999999999999       260.26  193.47
freshbubble     318.6999999999999       502.33999999999975      183.63999999999987
matreshka       0.0     182.67000000000004      182.67000000000004
chi     358.9400000000001       538.6100000000001       179.67000000000002
cristalinas     427.6299999999999       584.9499999999999       157.32000000000005
farmona 1692.4600000000003      1843.4299999999998      150.96999999999957
latinoil        249.52  384.59000000000015      135.07000000000014
miskin  158.04000000000002      293.06999999999994      135.02999999999992
elizavecca      70.53   204.30000000000004      133.77000000000004
nefertiti       233.52000000000007      366.64  133.11999999999992
finish  98.38   230.38  132.0
igrobeauty      513.6600000000005       645.0700000000006       131.41000000000008
dizao   819.1300000000003       945.5100000000014       126.38000000000102
osmo    645.58  762.31  116.7299999999999
batiste 772.4000000000001       874.1699999999998       101.76999999999975
carmex  145.08000000000004      243.36  98.27999999999997
```

hadoop@ip-172-31-27-109:~

```
eos     54.339999999999996      152.61  98.27000000000001
depilflax       2707.0699999999956      2803.779999999998       96.71000000000231
enjoy   41.35   136.57000000000002      95.22000000000003
kerasys 430.91000000000014      525.2   94.2899999999999
aura    83.95   177.50999999999996      93.55999999999996
plazan  101.36999999999999      194.01000000000005      92.64000000000006
koelf   422.72999999999996      507.28999999999985      84.55999999999989
nirvel  163.04  234.32999999999987      71.28999999999988
konad   739.8299999999997       810.6699999999992       70.83999999999946
egomania        77.47   146.04000000000002      68.57000000000002
cutrin  299.37  367.62  68.25
laboratorium    246.5   312.52  66.01999999999998
inm     288.0199999999999       351.2100000000001       63.19000000000017
dewal   0.0     61.28999999999999       61.28999999999999
marutaka-foot   49.22   109.33000000000001      60.110000000000014
kares   0.0     59.45   59.45
profhenna       679.2300000000002       736.8499999999999       57.61999999999966
koelcia 55.5    112.75  57.25
balbcare        155.32999999999996      212.37999999999997      57.05000000000001
elskin  251.0900000000001       307.65000000000015      56.56000000000006
foamie  35.04   80.49   45.449999999999996
ladykin 125.64999999999999      170.57  44.92
likato  296.05999999999983      340.96999999999997      44.91000000000014
mavala  409.0400000000001       446.3200000000001       37.28000000000003
vilenta 197.59999999999997      231.21000000000004      33.61000000000007
beautyblender   78.74000000000001       109.40999999999998      30.669999999999973
biore   60.650000000000006      90.31   29.659999999999997
orly    902.3800000000002       931.0900000000004       28.71000000000015
estelare        444.8100000000005       471.87000000000023      27.059999999999718
profepil        93.36000000000001       118.02000000000001      24.659999999999997
blixz   38.95   63.4    24.449999999999996
binacil 0.0     24.259999999999998      24.259999999999998
godefroy        401.22  425.12  23.899999999999977
glysolid        69.72999999999998       91.58999999999999       21.860000000000014
veraclara       50.11000000000001       71.21000000000001       21.1
juno    0.0     21.08   21.08
kamill  63.010000000000005      81.49000000000001       18.480000000000004
treaclemoon     163.37000000000003      181.49000000000004      18.120000000000005
supertan        50.37000000000001       66.51000000000002       16.140000000000008
barbie  0.0     12.39   12.39
deoproce        316.84000000000003      329.1700000000001       12.330000000000041
rasyan  18.799999999999997      28.939999999999998      10.14
fly     17.14   27.169999999999998      10.029999999999998
tertio  236.16  245.8   9.640000000000015
```

```
jaguar  1102.1100000000004      1110.6500000000003      8.539999999999964
soleo   204.2   212.5299999999998       8.329999999999814
neoleor 43.41   51.7    8.290000000000006
moyou   5.71    10.280000000000001      4.570000000000001
bodyton 1376.3399999999983      1380.639999999999       4.300000000000637
skinity 8.88    12.440000000000001      3.5600000000000005
helloganic      0.0     3.1     3.1
grace   100.92000000000002      102.60999999999999      1.6899999999999693
cosima  20.23   20.929999999999993      0.6999999999999922
ovale   2.54    3.1     0.56
Time taken: 36.317 seconds, Fetched: 160 row(s)
hive>
```

Insights:

a. Here are some 160 brands with increment in the selling from October to November.
b. '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.
c. Among all these brands lists, 'Runail' which was the best brand in terms of selling inOctober and November combined is also in the top 10 brands with high increment for October (71539.28) to November (76758.61) i.e., increment of total 5219.38.
d. This implies that 'Runail' is the best and popular brand among all other brandswithin people.

8. Your company wants to reward the top 10 users of its websites with a golden customer plan. Write a query to generate a list of top 10 users who spend the most.

```
SELECT user_id, SUM(price) AS tot_amt_spend FROM dynpart_buck_retailsstore

WHERE event_type = 'purchase'

GROUP BY user_id

ORDER BY tot_amt_spend DESC

LIMIT 10;
```

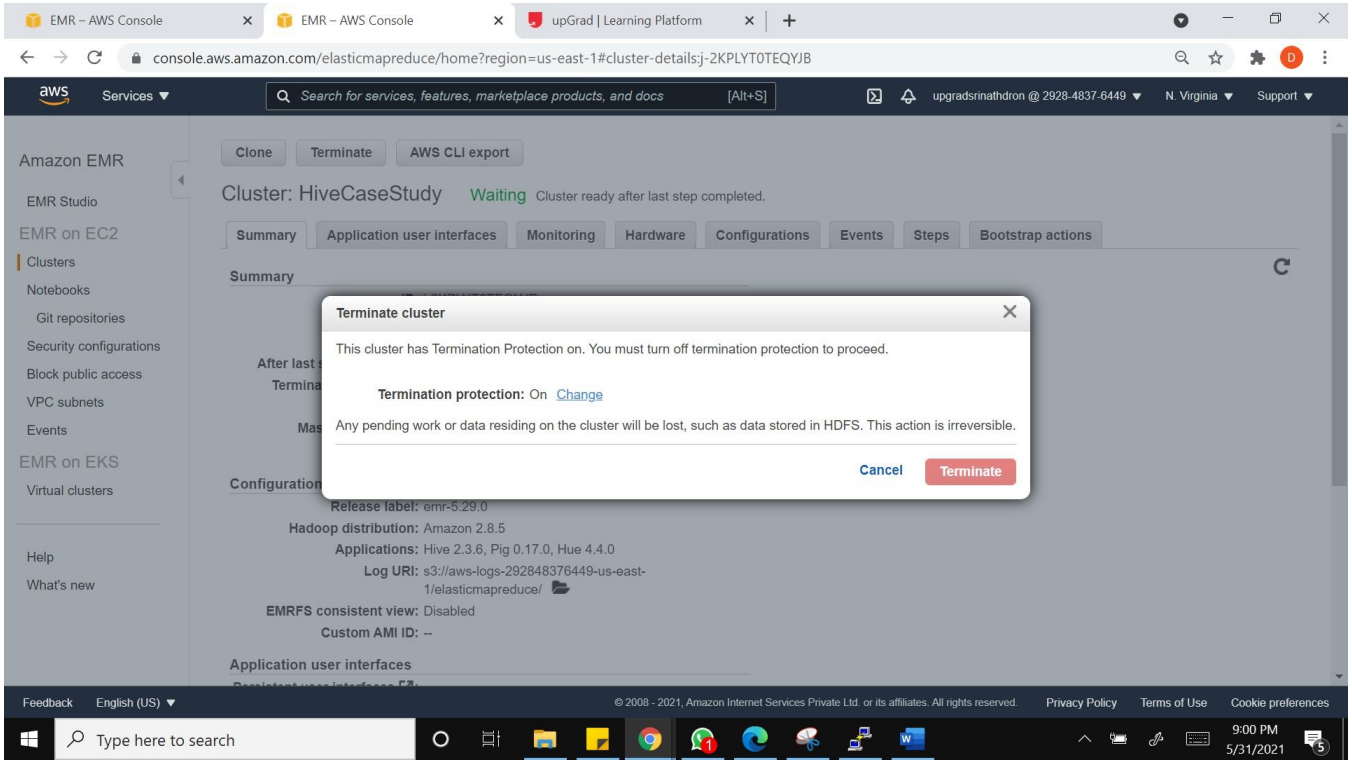
```
hive> SELECT user_id, SUM(price) AS tot_amt_spend FROM dynpart_buck_retailsstore
    > WHERE event_type = 'purchase'
    > GROUP BY user_id
    > ORDER BY tot_amt_spend DESC
    > LIMIT 10;
Query ID = hadoop_20210531161943_b6c52ee0-47d4-4a07-8fa9-a78e316b7fab
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1622448704713_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: 26.69 s
--------------------------------------------------------------------------------
OK
user_id tot_amt_spend
557790271       2715.869999999995
150318419       1645.9699999999998
562167663       1352.8500000000001
531900924       1329.4499999999996
557850743       1295.4800000000007
522130011       1185.3899999999999
561592095       1109.7000000000003
431950134       1097.5899999999997
566576008       1056.3599999999997
521347209       1040.9100000000003
Time taken: 34.875 seconds, Fetched: 10 row(s)
hive>
```

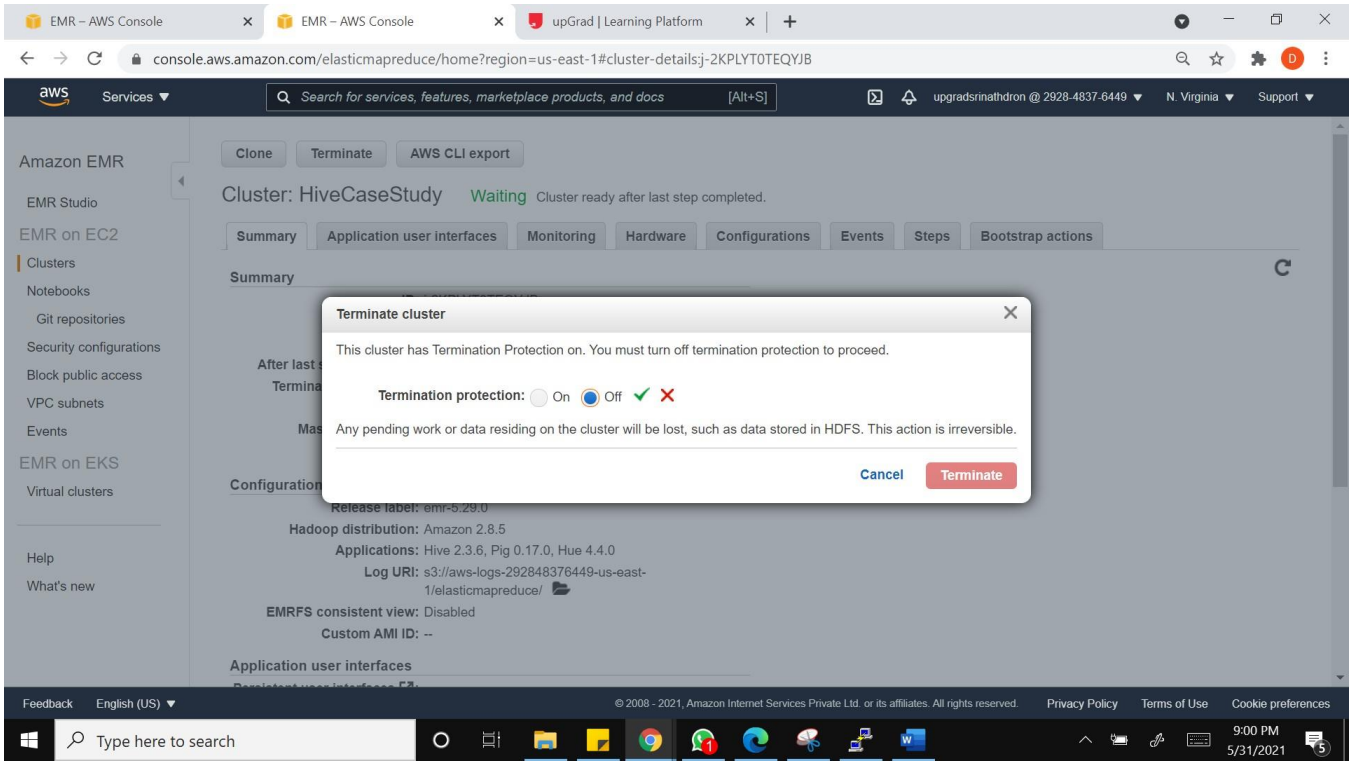
Insights:

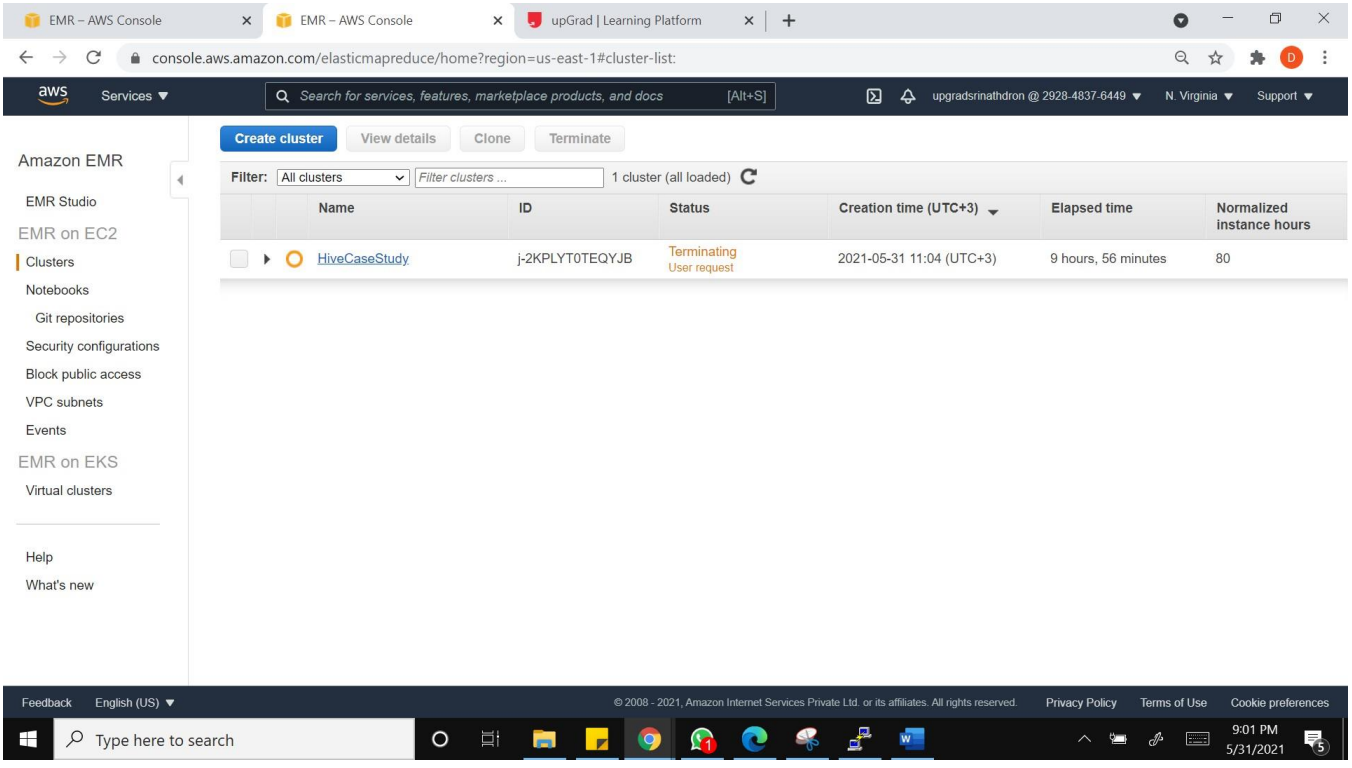
a. Here is the list of the top 10 users or buyers who have spent the most and could be rewarded with a Golden Customer plan to attract more people in the coming future.
b. With the Optimized table the execution time reduced with proper partitioning and bucketing.
c. Time taken to execute this query on optimized table is 34.875 seconds.

TERMINATION PROCESS:

After completing our analysis, we should terminate the EMR cluster







Cluster terminated!!