HIVE CASE STUDY

Submitted by : Abhas Ramteke, Renu Raj
Batch : DSC-42

PROBLEM STATEMENT :

With online sales gaining popularity, tech companies are exploring ways to improve their sales by analysing customer behaviour and gaining insights about product trends. Furthermore, the websites make it easier for customers to find the products they require without much scavenging. Needless to say, the role of big data analysts is among the most sought-after job profiles of this decade. Therefore, as part of this assignment, we will be challenging you, as a big data analyst, to extract data and gather insights from a real-life data set of an e-commerce company.

OBJECTIVE :

The goal is to extract the data and gather insights from a real-life data set of an e-commerce company.

DATA :

This project makes use of a public clickstream dataset from a cosmetics business. The clickstream data contains all of the logs of how a user navigated the ecommerce website. It also includes information such as client time spent on each page, amount of clicks made, things added to the cart, customer id, and so on.

The data is available from the link provided:

https://e-commerce-events-ml.s3.amazonaws.com/2019-Oct.csv

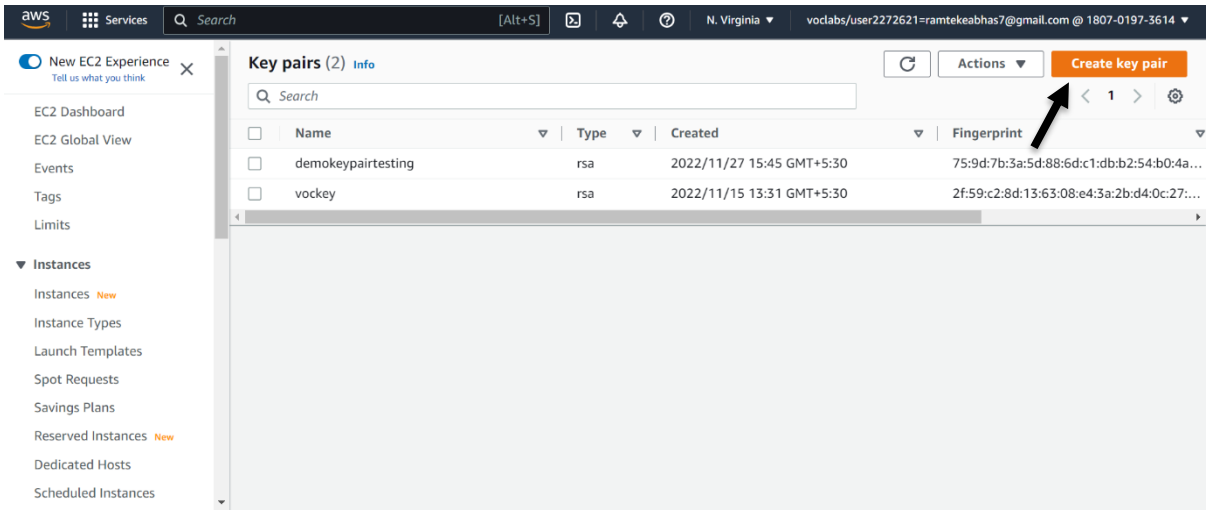
https://e-commerce-events-ml.s3.amazonaws.com/2019-Nov.csv

OVERVIEW OF STEPS :

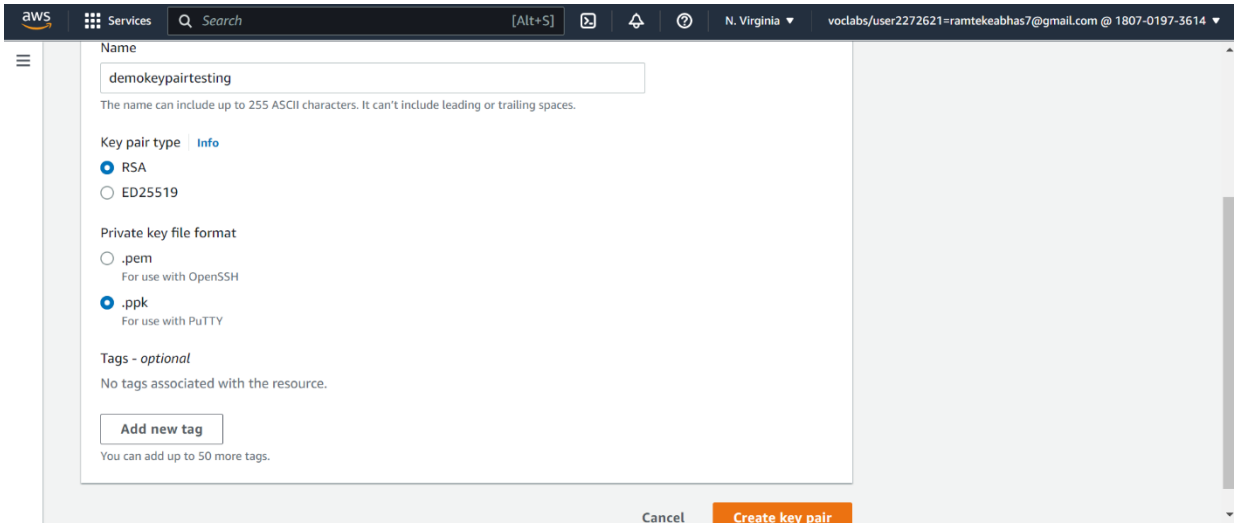
- Copying the data set into the HDFS:
    - Launch an EMR cluster that utilizes the Hive services.
    - Move the data from the S3 bucket into the HDFS.
- Creating the database and launching Hive queries on your EMR cluster:
    - Create the structure of your database,
    - Use optimized techniques to run your queries as efficiently as possible
    - Show the improvement of the performance after using optimization on any single query.
    - Run Hive queries to answer the questions given below.
- Cleaning up
    - Drop your database, and
    - Terminate your cluster

1. Key-pair creation and generation

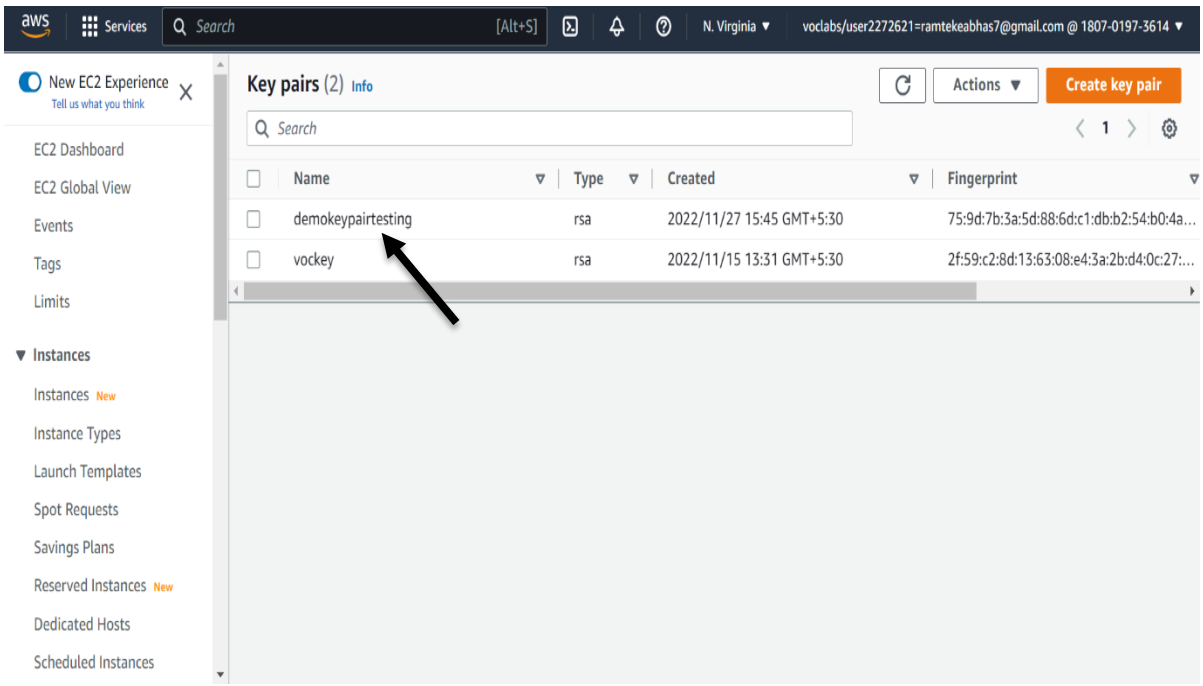
1.1 Click on Create Key pair



1.2 Providing name of Key & format as .ppk further for use with putty.

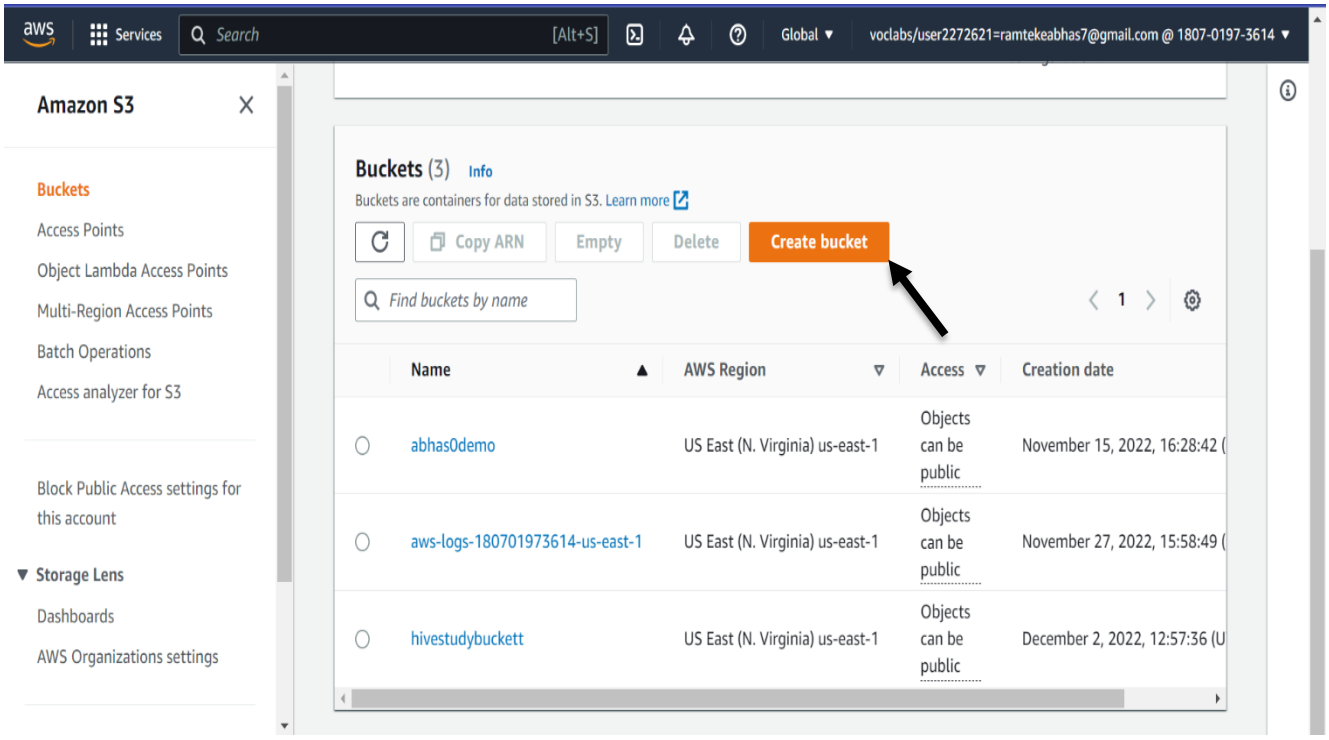


Successfully Key Pair created.
Name given to key-pair as shown below as “demokeypairtesting”.

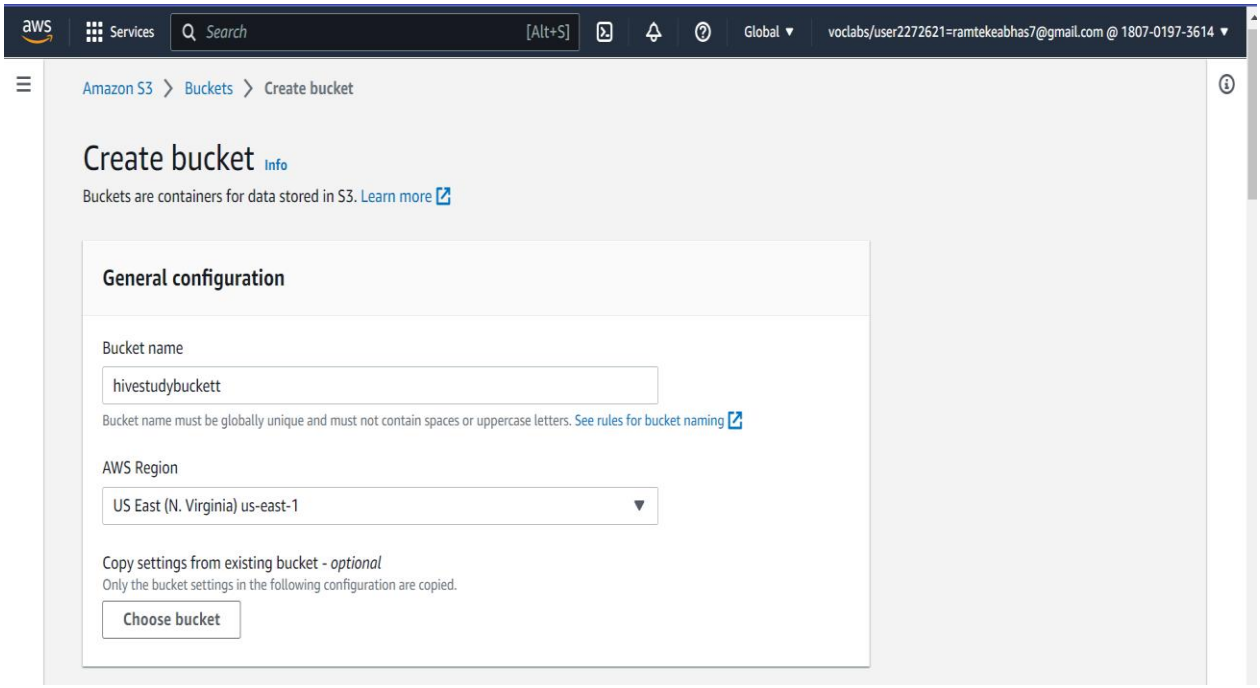


2. Create S3 Bucket to store data files :

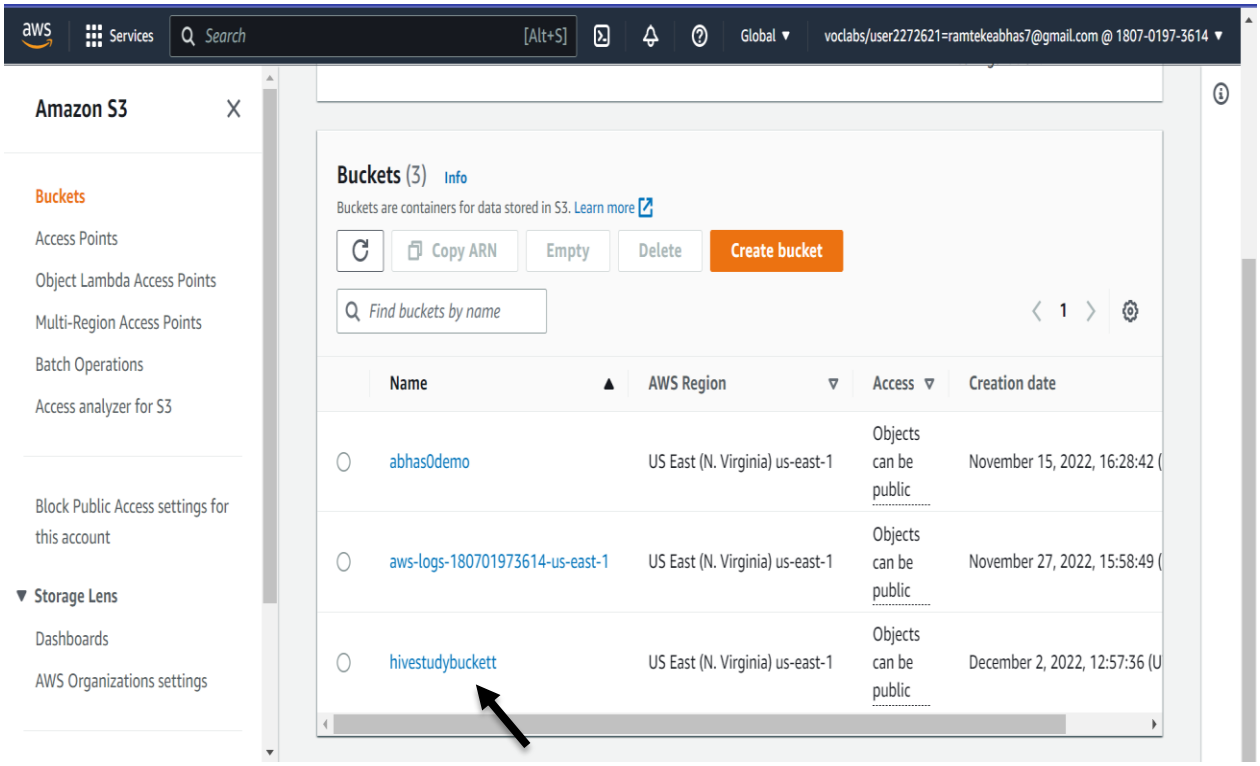
2.1 In search bar as we select s3 and clicked on the same, the below screen comes up in order to create bucket.



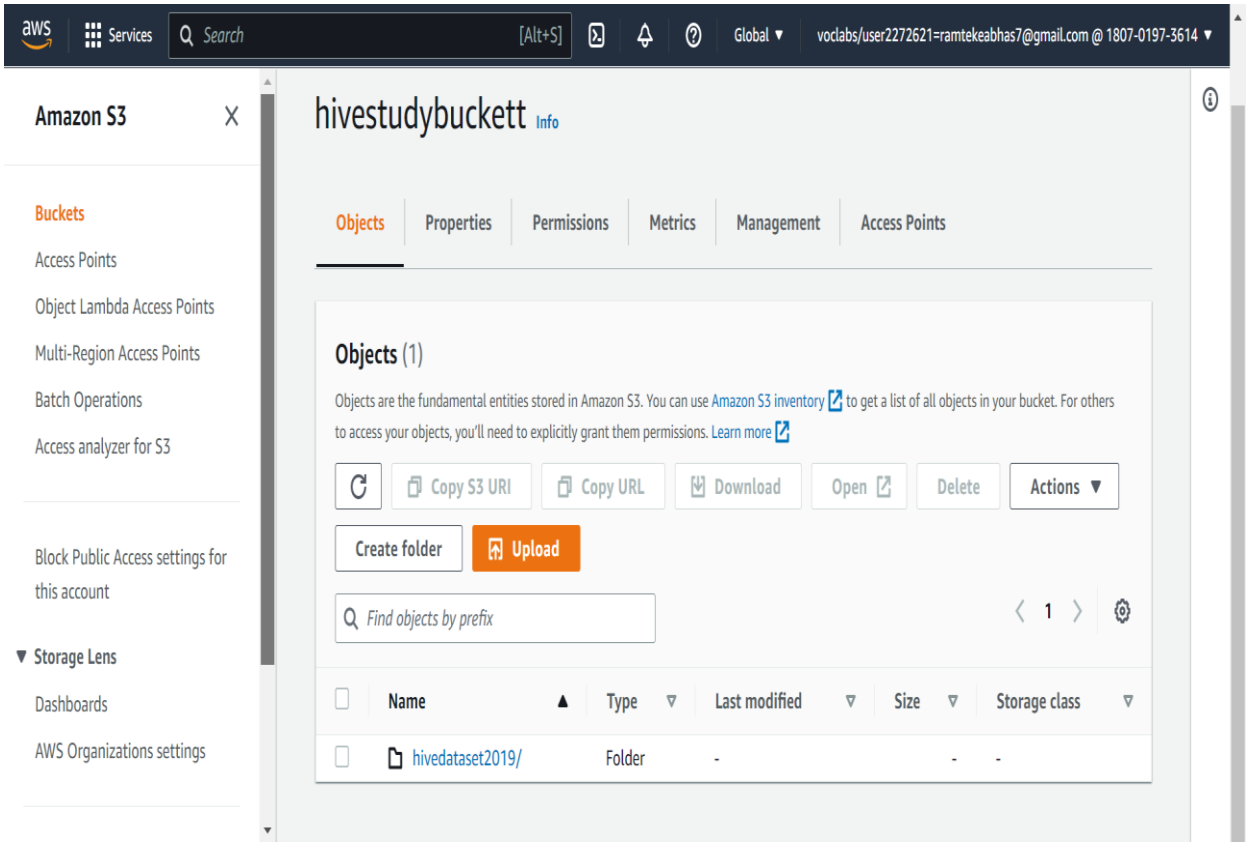
2.2 A unique name was given to create bucket which was not used earlier as instructed. Clicked on block to uncheck “block all public access”.

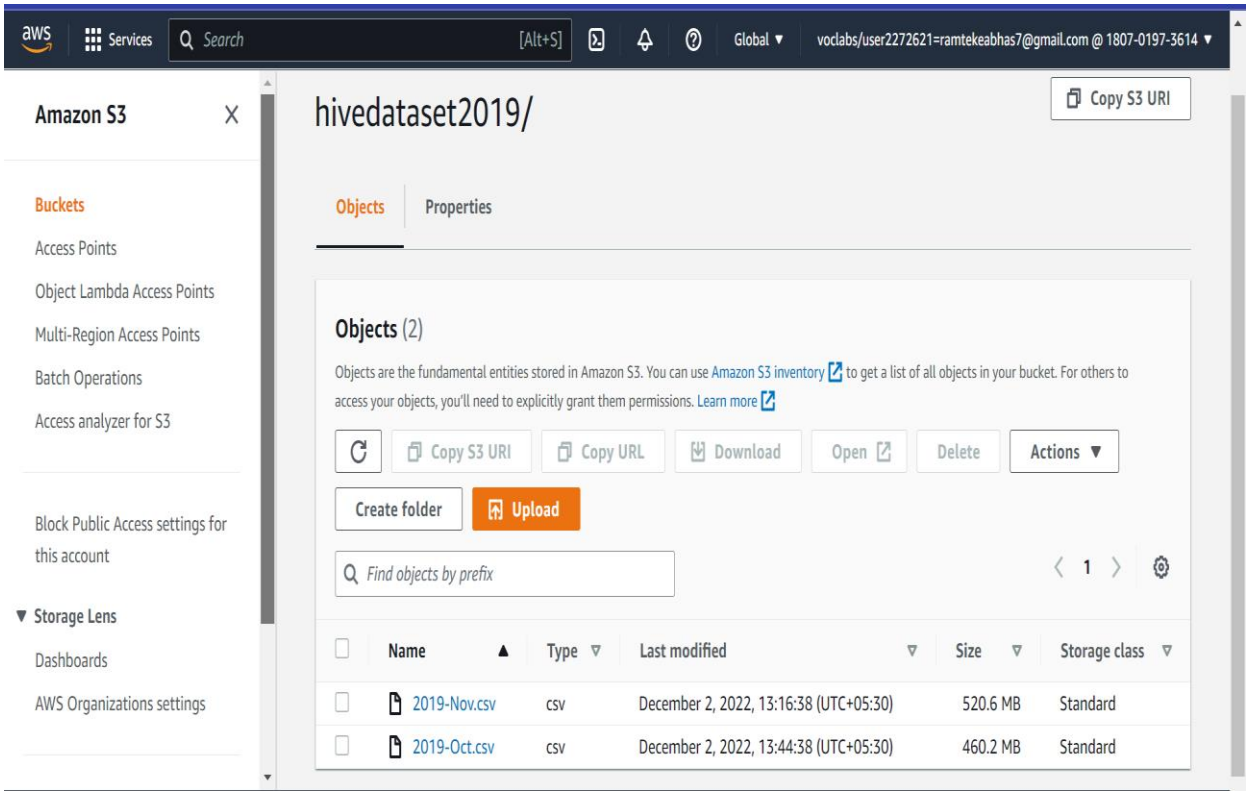


2.3 New Bucket created successfully with name “hivestudybuckett”.



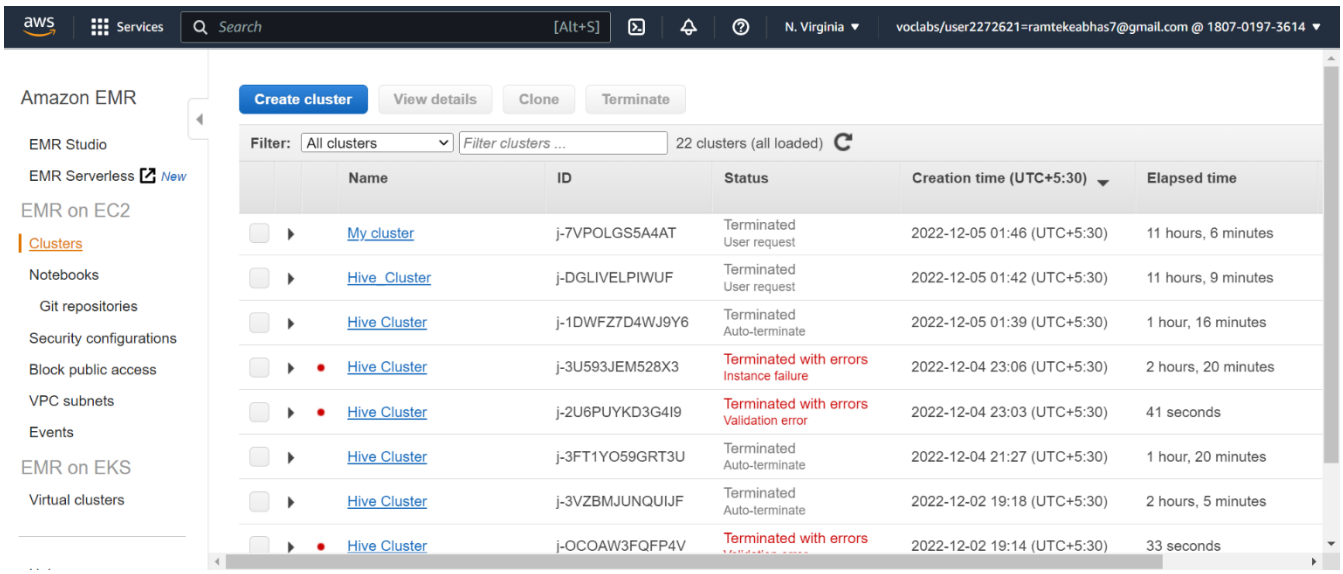
2.4 Firstly, a folder “hivedataset2019” is created under bucket & then uploaded both dataset in the folder.



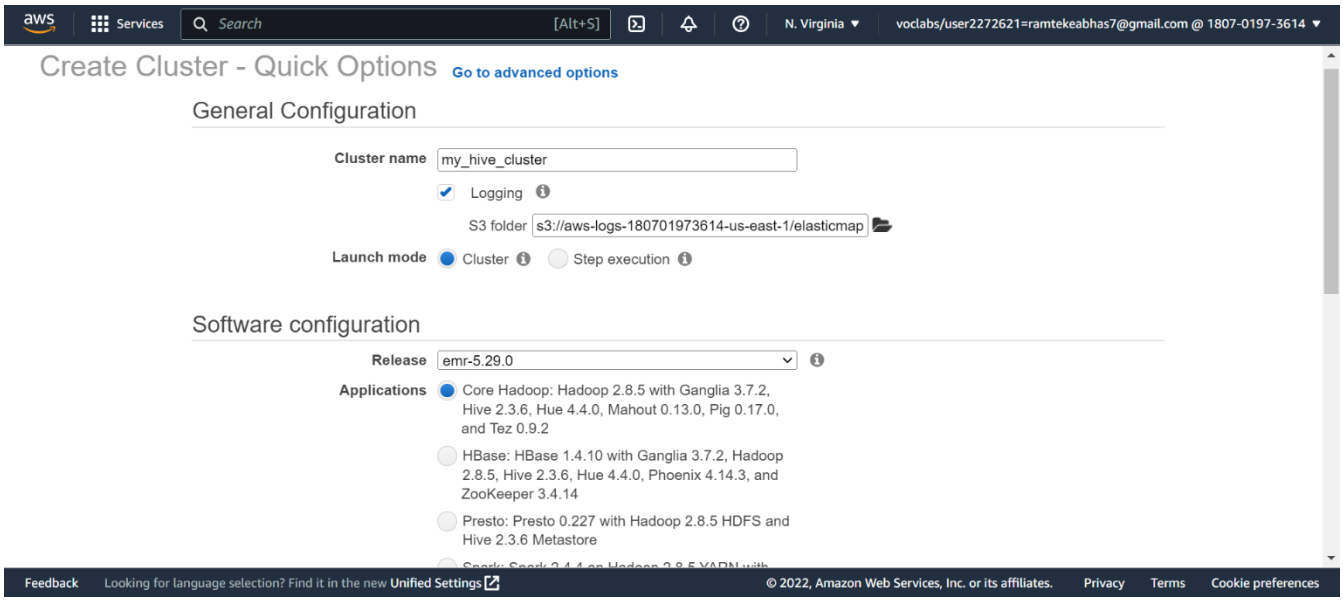


3. EMR CLUSTER CREATION:

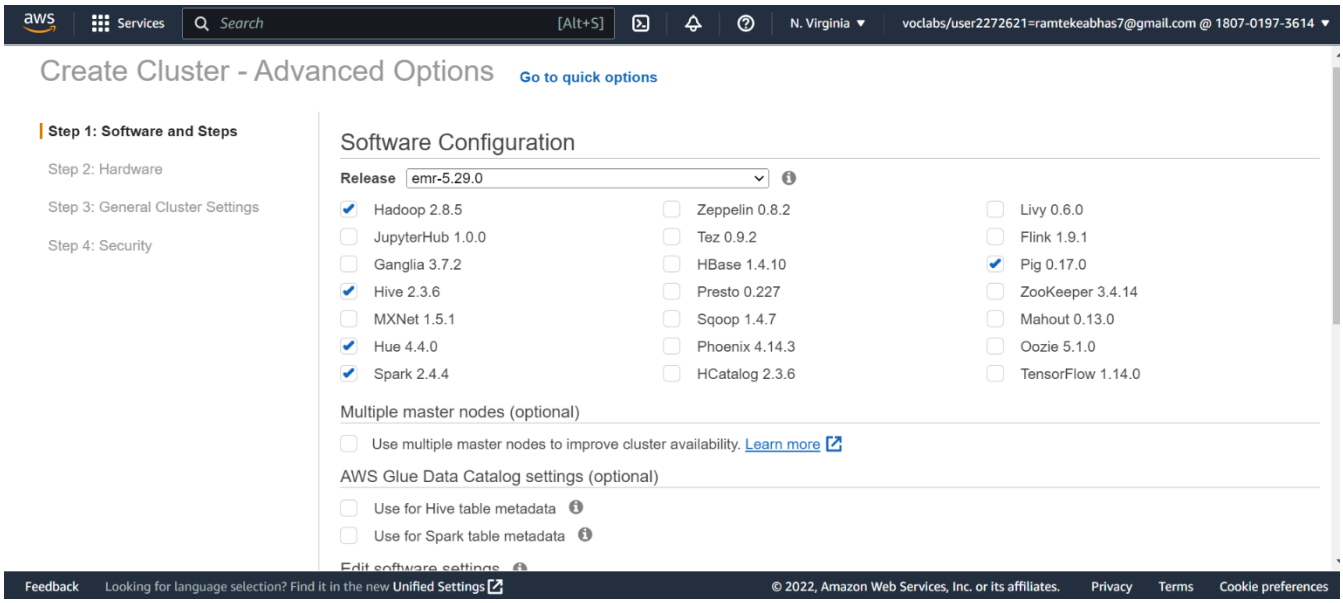
3.1 In Amazon EMR home page click on create cluster button.



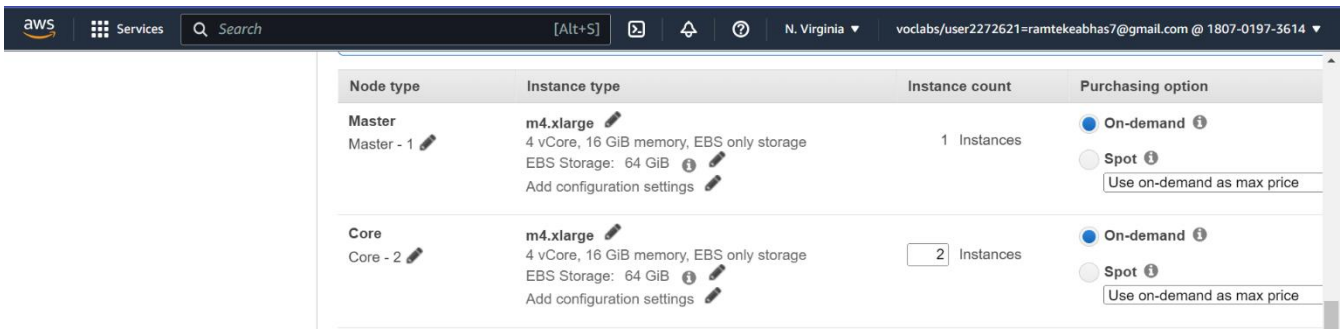
3.2 Then click on 'Go to advanced options'.



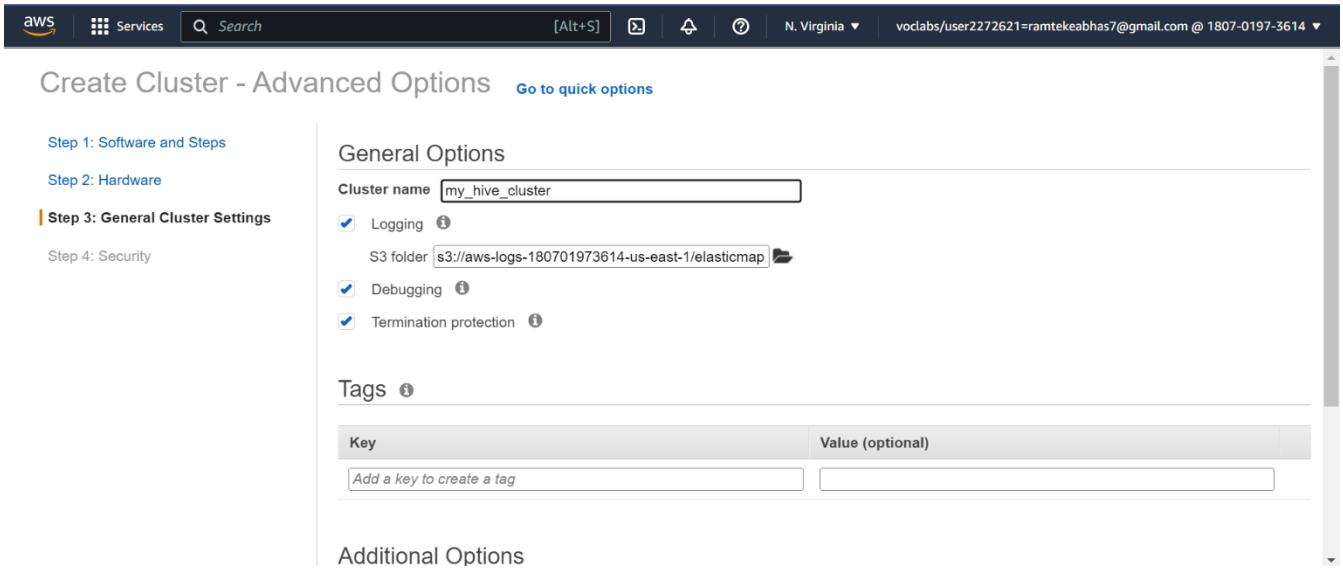
3.3 In software and steps page we have to select all the required services needed to complete this analysis. We are selecting emr-5.29.0 release as per the instruction with required services.



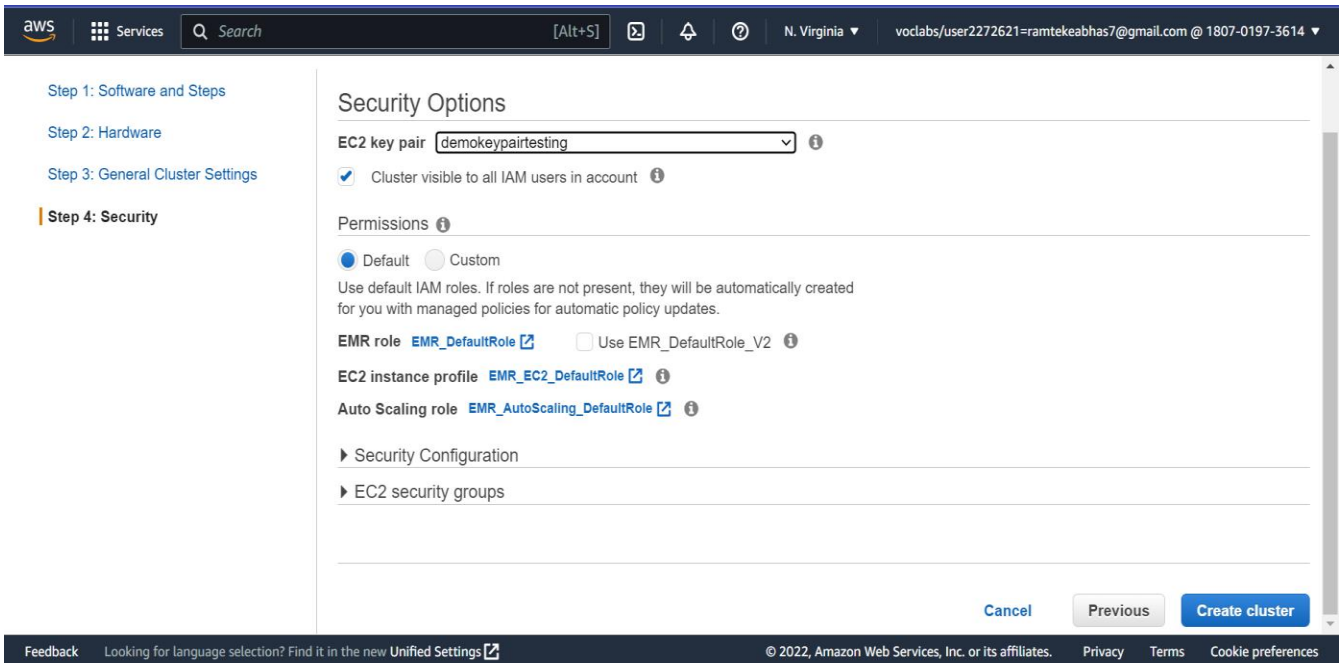
3.4 Then we will proceed to next page, in this step we have to specify the required configuration needed for the cluster. Here we have to select the instance type and instance count for master and core node. We have selected instance type as m4.large and 1 instance count for master and 2 for core node.



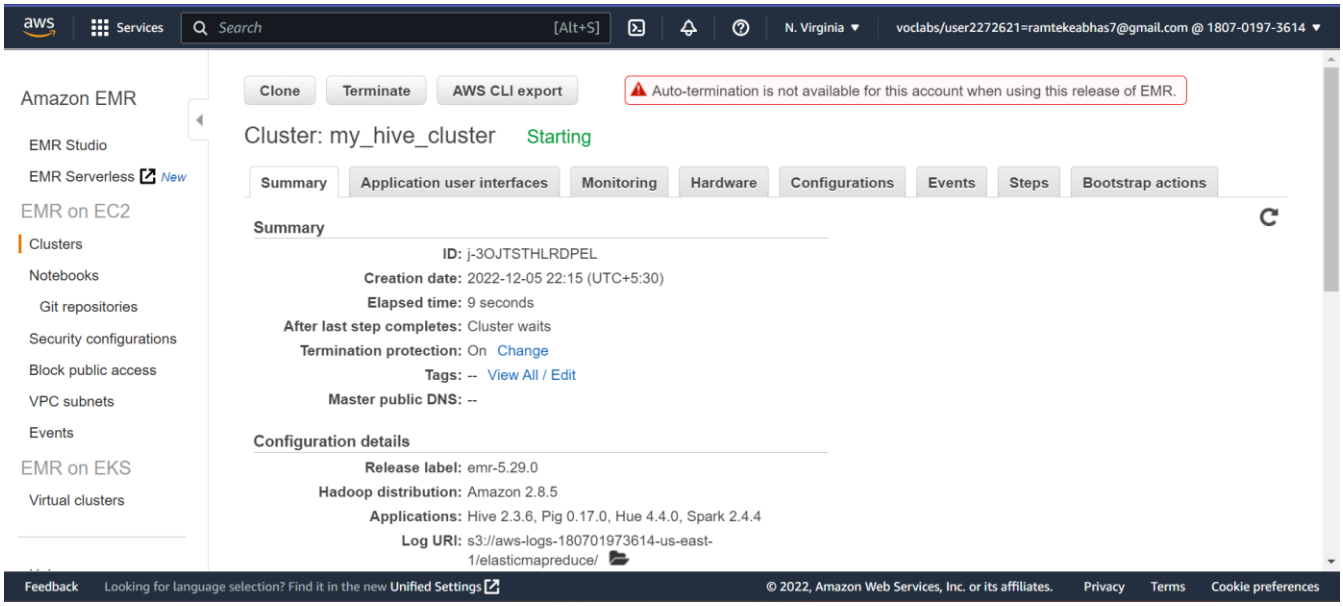
3.5 Here we have given the name of the cluster as “my_hive_cluster”.



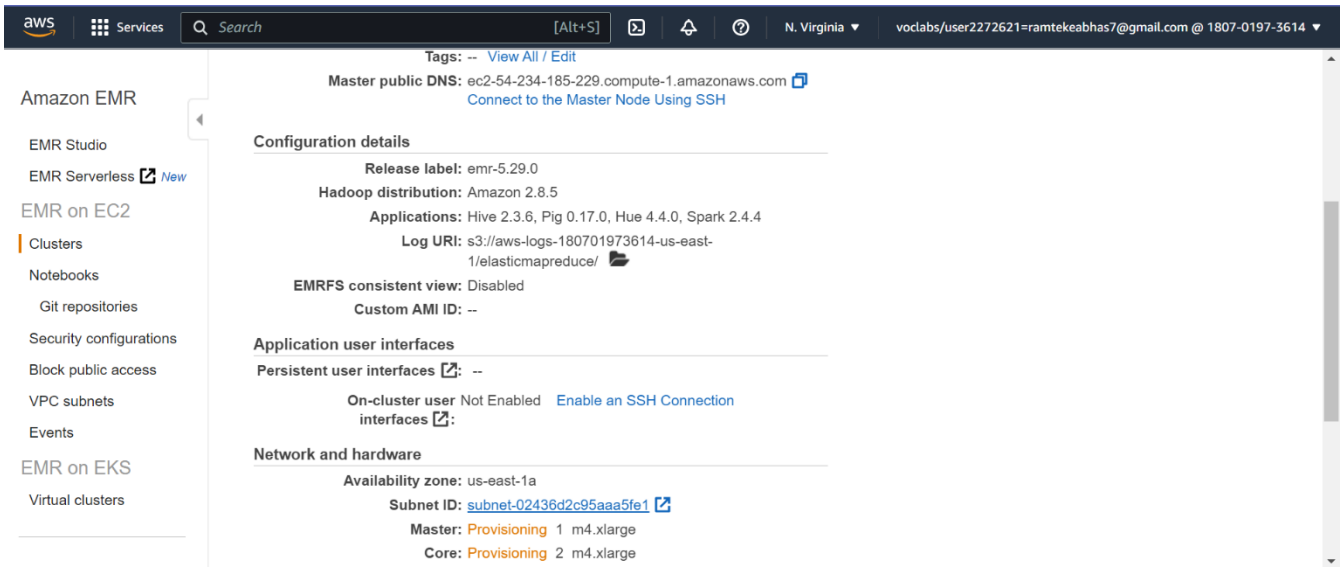
3.6 We have selected the created EC2 key pair named “demokeypairtesting” and then click on the create cluster option.



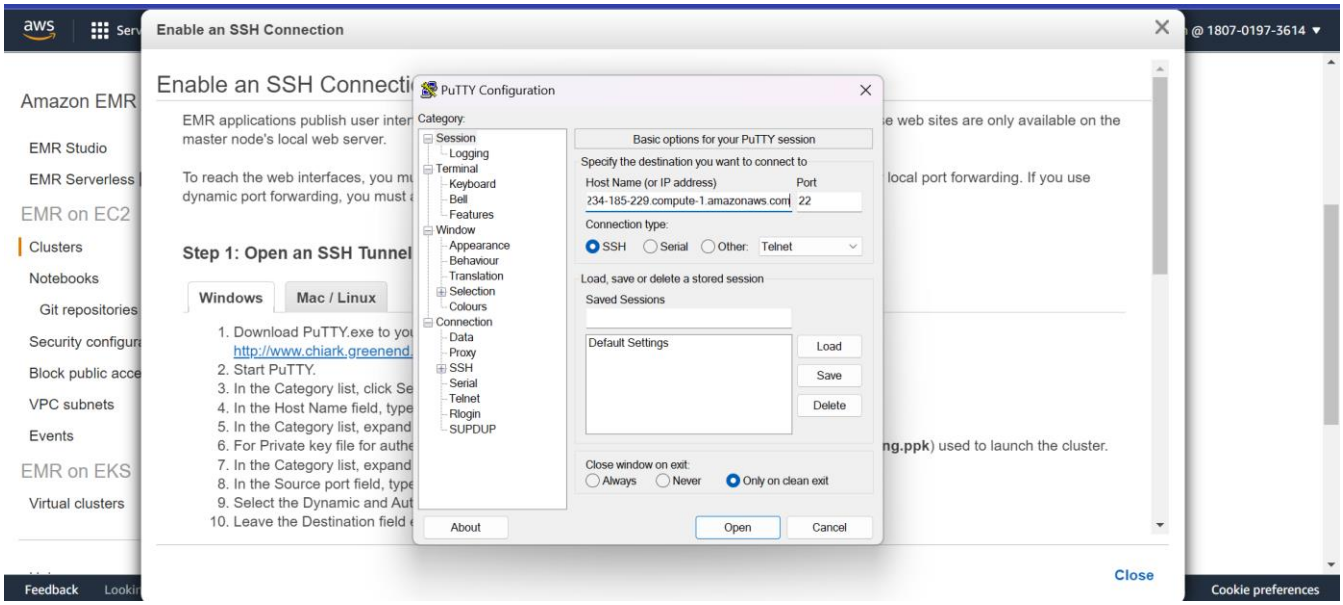
3.7 Then cluster has been created and will be in the “Starting” state and will change to ‘Waiting’ state after sometime.

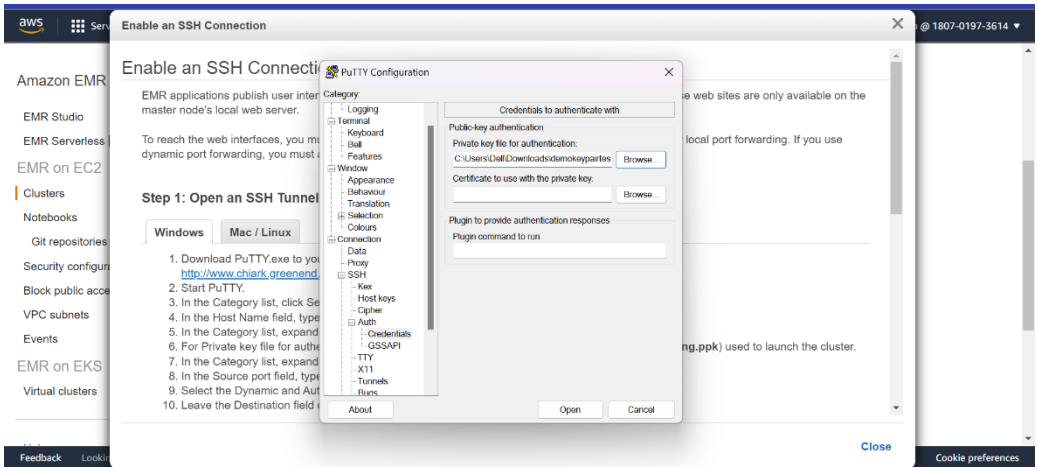


3.8 Now cluster is started & connected to Master Node using SSH connection.

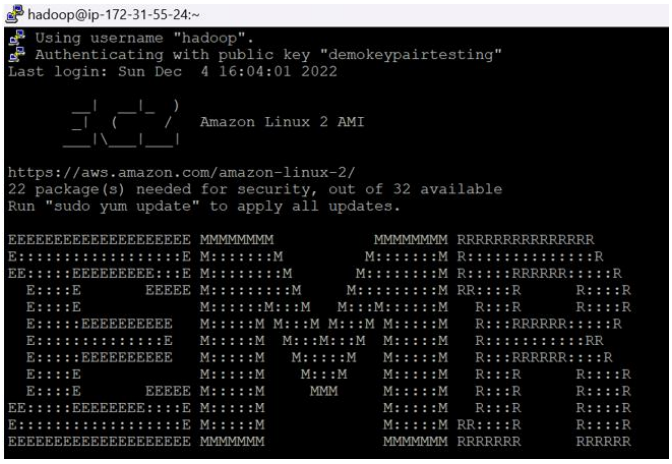


3.9 Once cluster in running state we have to click on Master public DNS(Connect to master stroke using SSH). We have to open the putty configuration and then give the host name (master node DNS) and then browse to the private key file location by clicking on Connection → SSH → Auth → credentials. Selected browse option for selecting key pair.





3.10 Then EMR command line interface will be open and login as Hadoop.



4. Copying the data into HDFS

4.1 To access the public s3bucket.

Command : aws s3 ls hivestudybuckett

Output : PRE hivedataset2019/ (this is folder where both the dataset is uploaded).

```
[hadoop@ip-172-31-55-24 ~]$ aws s3 ls hivestudybuckett
                           PRE hivedataset2019/
```

4.2 Command to check already present directories in HDFS.

Command: hadoop fs -ls /

```
[hadoop@ip-172-31-55-24 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x   - hdfs hdfsadmingroup          0 2022-12-04 16:04 /apps
drwxrwxrwt   - hdfs hdfsadmingroup          0 2022-12-04 16:07 /tmp
drwxr-xr-x   - hdfs hdfsadmingroup          0 2022-12-04 16:04 /user
drwxr-xr-x   - hdfs hdfsadmingroup          0 2022-12-04 16:04 /var
```

4.3 Checking the available directory.

Command: hadoop fs -ls /user/

```
[hadoop@ip-172-31-55-24 ~]$ hadoop fs -mkdir /user/hive/hive_case_study_abhas
[hadoop@ip-172-31-55-24 ~]$ hadoop fs -ls /user/
Found 6 items
drwxrwxrwx   - hadoop hdfsadmingroup          0 2022-12-04 16:04 /user/hadoop
drwxr-xr-x   - mapred mapred                  0 2022-12-04 16:04 /user/history
drwxrwxrwx   - hdfs   hdfsadmingroup          0 2022-12-04 16:40 /user/hive
drwxrwxrwx   - hue    hue                     0 2022-12-04 16:04 /user/hue
drwxrwxrwx   - oozie  oozie                   0 2022-12-04 16:06 /user/oozie
drwxrwxrwx   - root   hdfsadmingroup          0 2022-12-04 16:04 /user/root
[hadoop@ip-172-31-55-24 ~]$
```

4.4 Creating a directory in HDFS.

Command: hadoop fs -mkdir /user/hive/hive_case_study_abhas

4.5 Load the s3 public data set to created directory “hive_case_study_abhas” in hadoop .

Command: hadoop distcp s3://hivestudybuckett/hivedataset2019/2019-Oct.csv /user/new_hive/2019-Oct.csv

```
[hadoop@ip-172-31-53-223 ~]$ hadoop distcp s3://hivestudybuckett/hivedataset2019/2019-Oct.csv  /user
/new_hive/2019-Oct.csv
22/12/04 18:00:45 INFO tools.OptionsParser: parseChunkSize: blocksperchunk false
22/12/04 18:00:47 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=fal
se, deleteMissing=false, ignoreFailures=false, overwrite=false, append=false, useDiff=false, useRdif
f=false, fromSnapshot=null, toSnapshot=null, skipCRC=false, blocking=true, numListstatusThreads=0, m
axMaps=20, mapBandwidth=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus
=[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths
=[s3://hivestudybuckett/hivedataset2019/2019-Oct.csv], targetPath=/user/new_hive/2019-Oct.csv, targe
tPathExists=false, filtersFile='null', blocksPerChunk=0, copyBufferSize=8192, verboseLog=false}
22/12/04 18:00:47 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-53-223.ec2.interna
l/172.31.53.223:8032
22/12/04 18:00:47 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-53-223
.ec2.internal/172.31.53.223:10200
22/12/04 18:00:54 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/12/04 18:00:54 INFO tools.SimpleCopyListing: Build file listing completed.
22/12/04 18:00:54 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.t
ask.io.sort.mb
22/12/04 18:00:54 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapredu
ce.task.io.sort.factor
22/12/04 18:00:54 INFO tools.DistCp: Number of paths in the copy list: 1
22/12/04 18:00:54 INFO tools.DistCp: Number of paths in the copy list: 1
22/12/04 18:00:54 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-53-223.ec2.interna
l/172.31.53.223:8032
22/12/04 18:00:54 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-53-223
.ec2.internal/172.31.53.223:10200
22/12/04 18:00:55 INFO mapreduce.JobSubmitter: number of splits:1
22/12/04 18:00:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1670175863837_0001
22/12/04 18:00:55 INFO conf.Configuration: resource-types.xml not found
22/12/04 18:00:55 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/12/04 18:00:55 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi,
type = COUNTABLE
22/12/04 18:00:55 INFO resource.ResourceUtils: Adding resource type - name = vcores, units = , type
= COUNTABLE
22/12/04 18:00:56 INFO impl.YarnClientImpl: Submitted application application_1670175863837_0001
22/12/04 18:00:56 INFO mapreduce.Job: The url to track the job: http://ip-172-31-53-223.ec2.internal
:20888/proxy/application_1670175863837_0001/
22/12/04 18:00:56 INFO tools.DistCp: DistCp job-id: job_1670175863837_0001
22/12/04 18:00:56 INFO mapreduce.Job: Running job: job_1670175863837_0001
22/12/04 18:01:07 INFO mapreduce.Job: Job job_1670175863837_0001 running in uber mode : false
22/12/04 18:01:07 INFO mapreduce.Job:  map 0% reduce 0%
22/12/04 18:01:25 INFO mapreduce.Job:  map 100% reduce 0%
22/12/04 18:01:29 INFO mapreduce.Job: Job job_1670175863837_0001 completed successfully
22/12/04 18:01:29 INFO mapreduce.Job: Counters: 38
```

```
                Bytes Read=250
        File Output Format Counters
                Bytes Written=0
        DistCp Counters
                Bytes Copied=482542278
                Bytes Expected=482542278
                Files Copied=1
```

Command: hadoop distcp s3://hivestudybuckett/hivedataset2019/2019-Nov.csv /user/new_hive/2019-Nov.csv

```
[hadoop@ip-172-31-53-223 ~]$ hadoop distcp s3://hivestudybuckett/hivedataset2019/2019-Nov.csv  /user
/new_hive/2019-Nov.csv
22/12/04 18:04:09 INFO tools.OptionsParser: parseChunkSize: blocksperchunk false
22/12/04 18:04:11 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=fal
se, deleteMissing=false, ignoreFailures=false, overwrite=false, append=false, useDiff=false, useRdif
f=false, fromSnapshot=null, toSnapshot=null, skipCRC=false, blocking=true, numListstatusThreads=0, m
axMaps=20, mapBandwidth=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus
=[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths
=[s3://hivestudybuckett/hivedataset2019/2019-Nov.csv], targetPath=/user/new_hive/2019-Nov.csv, targe
tPathExists=false, filtersFile='null', blocksPerChunk=0, copyBufferSize=8192, verboseLog=false}
22/12/04 18:04:11 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-53-223.ec2.interna
l/172.31.53.223:8032
22/12/04 18:04:11 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-53-223
.ec2.internal/172.31.53.223:10200
22/12/04 18:04:16 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/12/04 18:04:16 INFO tools.SimpleCopyListing: Build file listing completed.
22/12/04 18:04:16 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.t
ask.io.sort.mb
22/12/04 18:04:16 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapredu
ce.task.io.sort.factor
22/12/04 18:04:16 INFO tools.DistCp: Number of paths in the copy list: 1
22/12/04 18:04:16 INFO tools.DistCp: Number of paths in the copy list: 1
```

```
ce.task.io.sort.factor
22/12/04 18:04:16 INFO tools.DistCp: Number of paths in the copy list: 1
22/12/04 18:04:16 INFO tools.DistCp: Number of paths in the copy list: 1
22/12/04 18:04:16 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-53-223.ec2.interna
l/172.31.53.223:8032
22/12/04 18:04:16 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-53-223
.ec2.internal/172.31.53.223:10200
22/12/04 18:04:17 INFO mapreduce.JobSubmitter: number of splits:1
22/12/04 18:04:17 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1670175863837_0002
22/12/04 18:04:18 INFO conf.Configuration: resource-types.xml not found
22/12/04 18:04:18 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
22/12/04 18:04:18 INFO resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi,
type = COUNTABLE
22/12/04 18:04:18 INFO resource.ResourceUtils: Adding resource type - name = vcores, units = , type
= COUNTABLE
22/12/04 18:04:18 INFO impl.YarnClientImpl: Submitted application application_1670175863837_0002
22/12/04 18:04:18 INFO mapreduce.Job: The url to track the job: http://ip-172-31-53-223.ec2.internal
:20888/proxy/application_1670175863837_0002/
22/12/04 18:04:18 INFO tools.DistCp: DistCp job-id: job_1670175863837_0002
22/12/04 18:04:18 INFO mapreduce.Job: Running job: job_1670175863837_0002
22/12/04 18:04:26 INFO mapreduce.Job: Job job_1670175863837_0002 running in uber mode : false
22/12/04 18:04:26 INFO mapreduce.Job:  map 0% reduce 0%
22/12/04 18:04:44 INFO mapreduce.Job:  map 100% reduce 0%
22/12/04 18:04:48 INFO mapreduce.Job: Job job_1670175863837_0002 completed successfully
22/12/04 18:04:48 INFO mapreduce.Job: Counters: 38
```

```
22/12/04 18:04:48 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=224554
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=385
                HDFS: Number of bytes written=545839412
                HDFS: Number of read operations=11
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=5
                S3: Number of bytes read=545839412
                S3: Number of bytes written=0
                S3: Number of read operations=0
                S3: Number of large read operations=0
                S3: Number of write operations=0
        Job Counters
                Launched map tasks=1
                Other local map tasks=1
                Total time spent by all maps in occupied slots (ms)=646624
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=20207
                Total vcore-milliseconds taken by all map tasks=20207
                Total megabyte-milliseconds taken by all map tasks=20691968
        Map-Reduce Framework
                Map input records=1
                Map output records=0
                Input split bytes=135
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=359
                CPU time spent (ms)=21250
                Physical memory (bytes) snapshot=618848256
```

```
                Physical memory (bytes) snapshot=618848256
                Virtual memory (bytes) snapshot=3334119424
                Total committed heap usage (bytes)=500695040
        File Input Format Counters
                Bytes Read=250
        File Output Format Counters
                Bytes Written=0
        DistCp Counters
                Bytes Copied=545839412
                Bytes Expected=545839412
                Files Copied=1
```

5. Hive query > hive

5.1 create database ABDB;

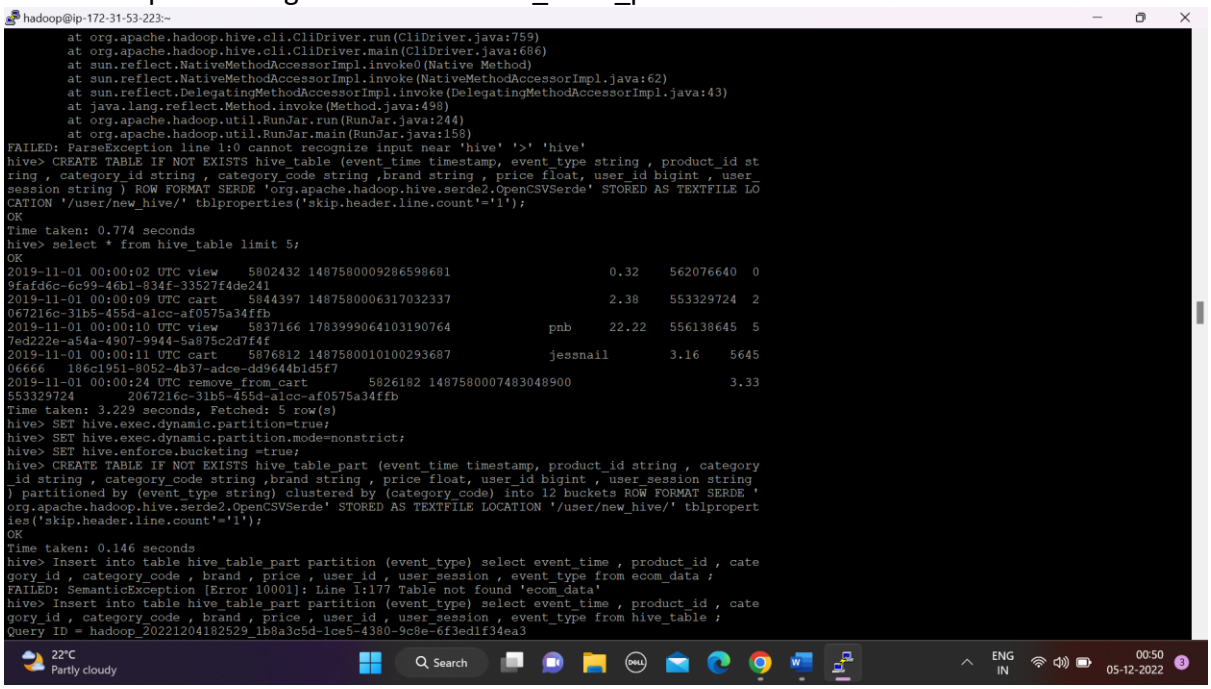
5.2 use ABDB;

5.3 Create table hive_table;

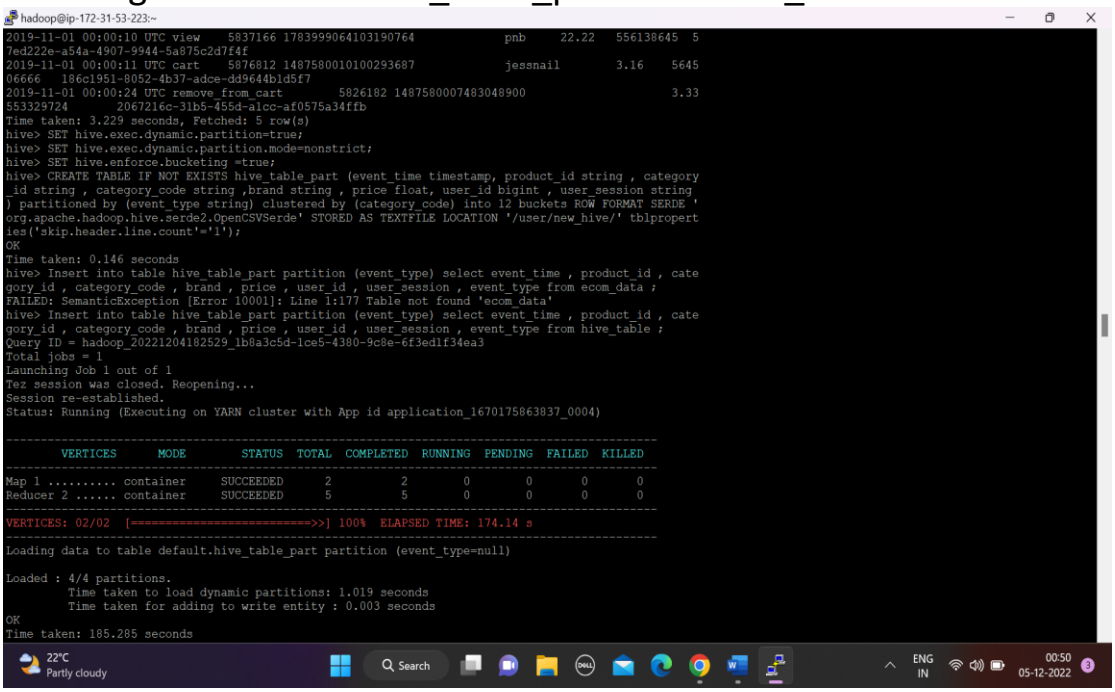
5.4 Once base table is created we need to go for optimizing the table for quick query result.
Optimised table name is hive_table_part.

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

5.5 Created partitioning table named hive_table_part.



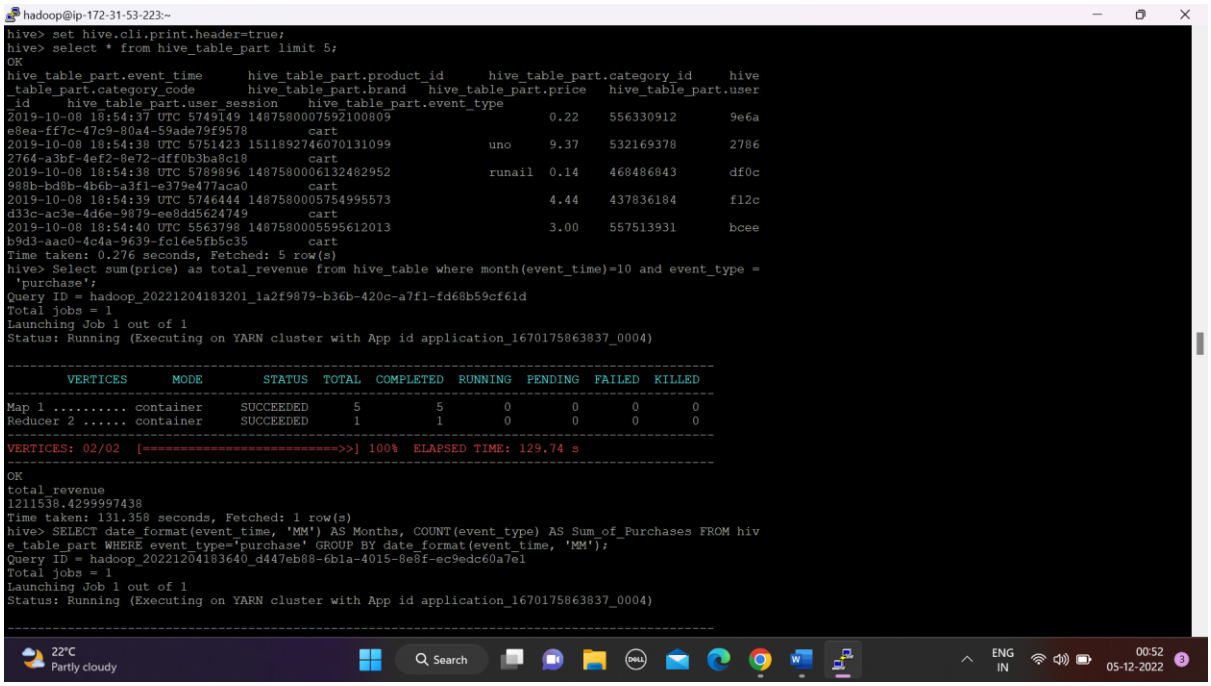
5.5 Inserting records into hive_table_part from hive_table.



5.6 Print the column headers for the queries.

Command:

Set hive.cli.print.header=true;



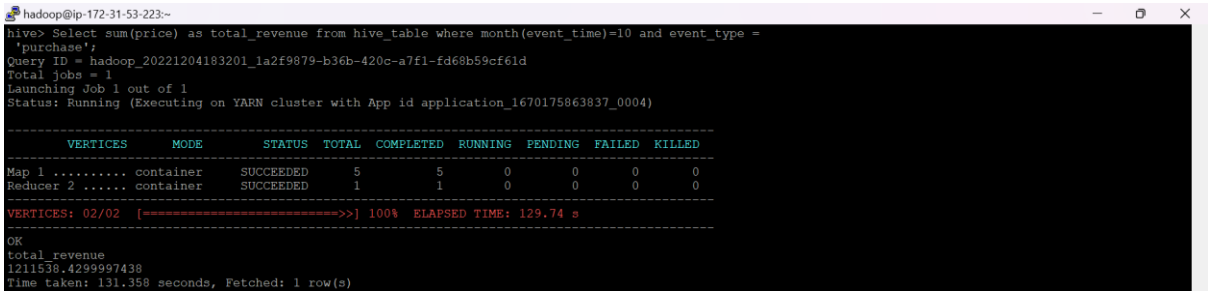
Following are the screen shots of the query outputs:

Question 1:

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

Ans 1- Total revenue generated with the help of base table created as in below image.

Query - Select sum(price) as total_revenue from hive_table where month(event_time)=10 and event_type = 'purchase' ;



Ans2 - Total revenue generated with the help of partition table created as in below image.

Select sum(price) as total_revenue from hive_table_part where month(event_time)=10 and event_type = 'purchase' ;

```
hive> select sum(price) as total_revenue from hive_table_part where month(event_time)=10 and event_type = 'purchase' ;
Query ID = hadoop_20221207190213_ddb111bc-144a-4fec-9e10-bf9dbe0dcda7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1670438307348_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: 9.07 s
--------------------------------------------------------------------------------
OK
total_revenue
1211525.179999663
Time taken: 9.958 seconds, Fetched: 1 row(s)
hive>
```

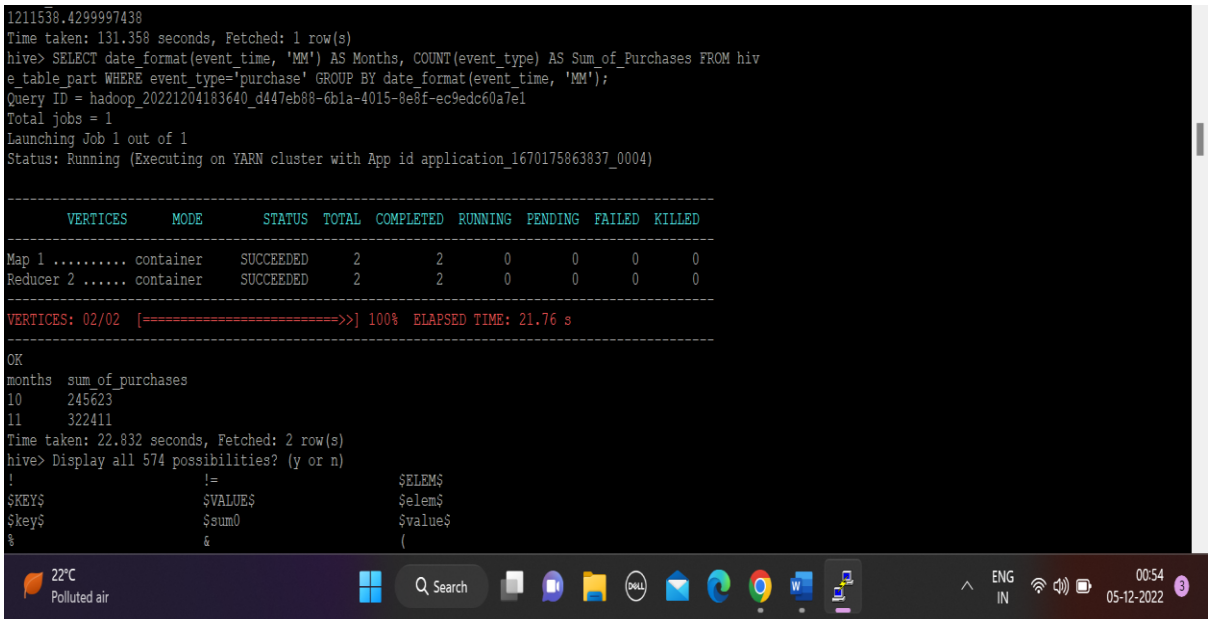
Inference:-

The total revenue made by the ECOM company in the month of October is 1211538/- & We attempted to obtain the result by using both the base table hive_table and the optimised table hive_table_part. We discovered that the Bucketed table improves query time by around 120 seconds when compared to the base table. Hence using hive_table_part partitioning table for further query for optimization.

Question 2)

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

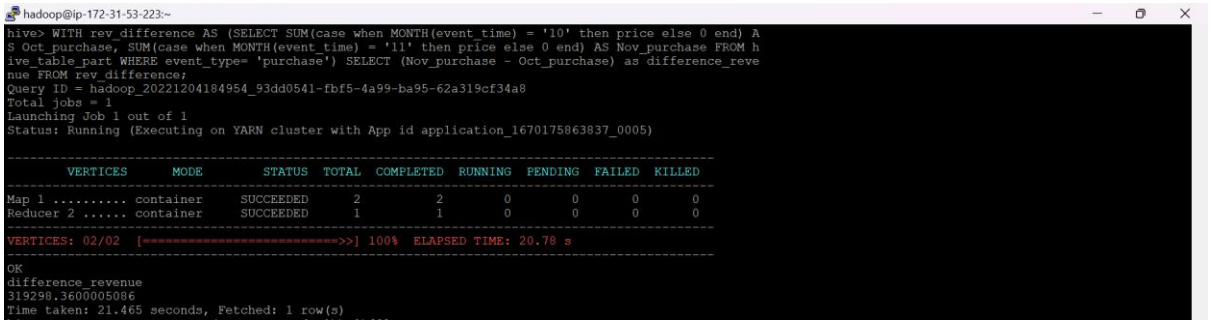
Query 2:
select date_format(event_time, 'MM') as month, count(event_type) as Sum_of_purchases from hive_table_part where event_type='purchase' group by date_format(event_time, 'MM');



Inference:
More purchases are done in the month of November compared to October, with a difference of 76788/-

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

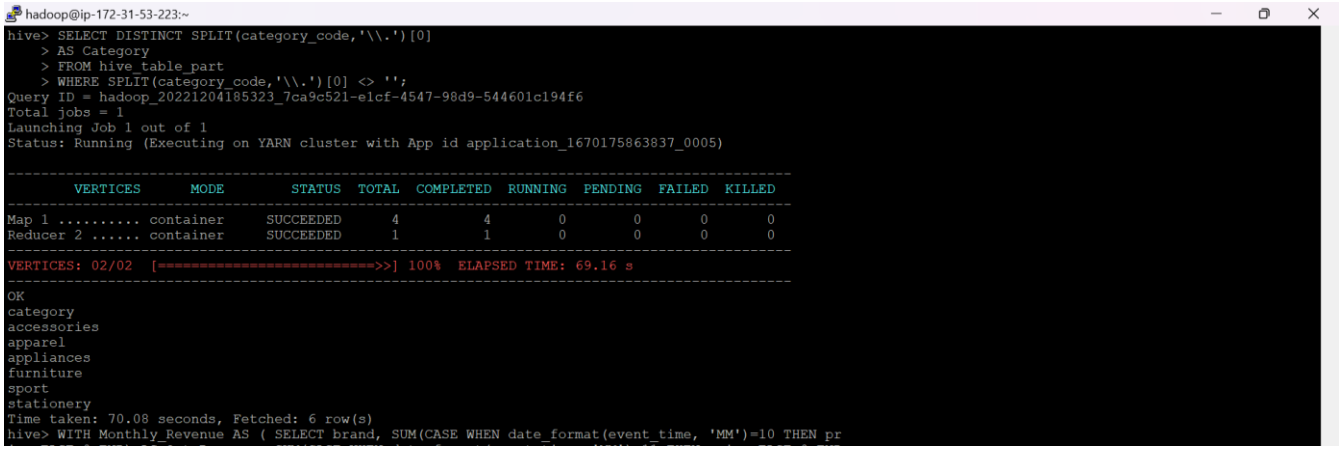
Query- WITH rev_difference AS
(SELECT SUM(case when MONTH(event_time) = '10' then price else 0 end) AS Oct_purchase, SUM(case when MONTH(event_time) = '11' then price else 0 end) AS Nov_purchase FROM hive_table_part
WHERE event_type= 'purchase') SELECT (Nov_purchase - Oct_purchase) as difference_revenue FROM rev_difference ;



Inference:
The difference in revenue between October and November is 319298/-. The month of November is more profitable.

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

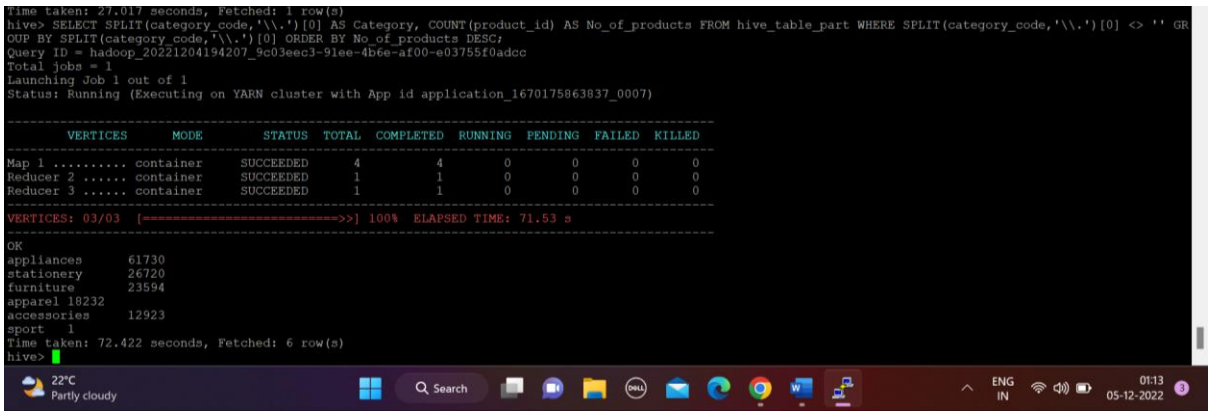
Query- SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Category FROM hive_table_part WHERE SPLIT(category_code,'\\.')[0] <> '';



Furniture, appliances, accessories, apparel, sport and stationery are the different categories in the table.

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

SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS No_of_products FROM hive_table_part WHERE SPLIT(category_code,'\\.')[0] <> '' GROUP BY SPLIT(category_code,'\\.')[0] ORDER BY No_of_products DESC;

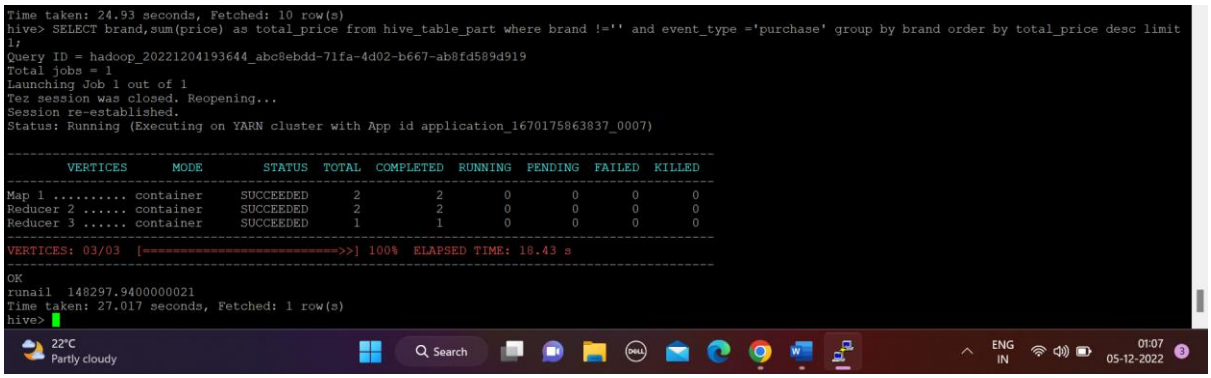


appliances	61730
stationery	26720
furniture	23594
apparel	18232
accessories	12923
sport	1

Inference: - Appliances category has the highest number of products 61730, and the least is sport which is only has 1.

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

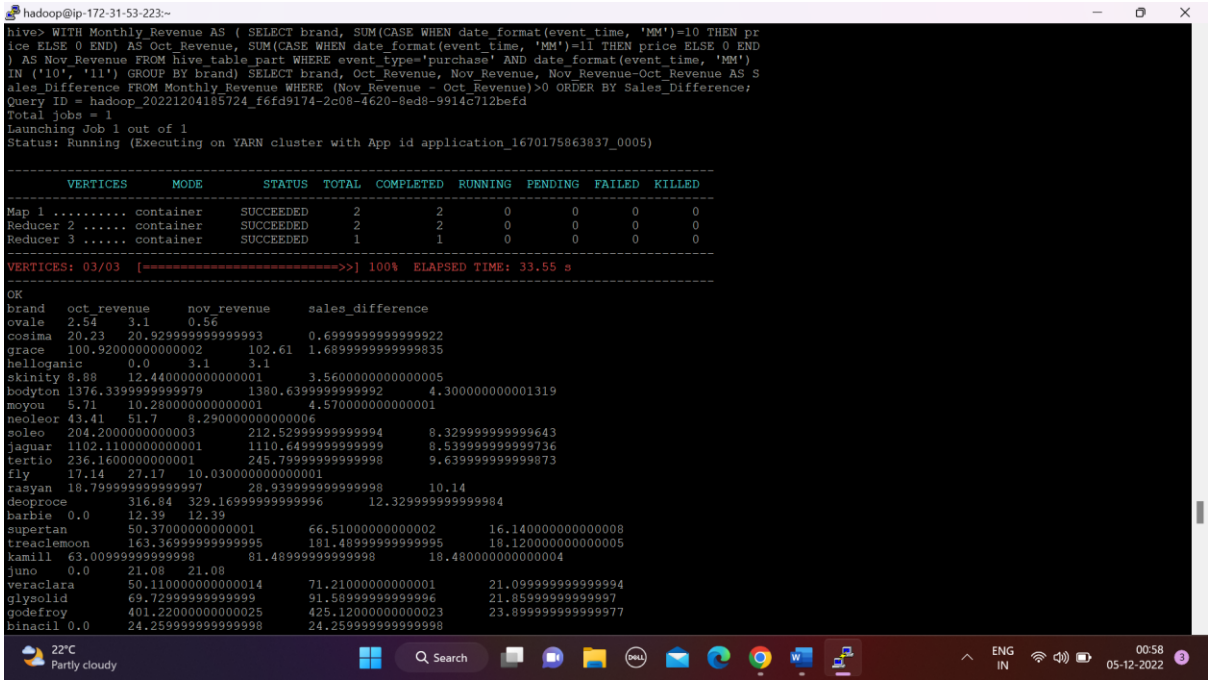
SELECT brand,sum(price) as total_price from hive_table_part where brand !='' and event_type ='purchase' group by brand order by total_price desc limit 1;

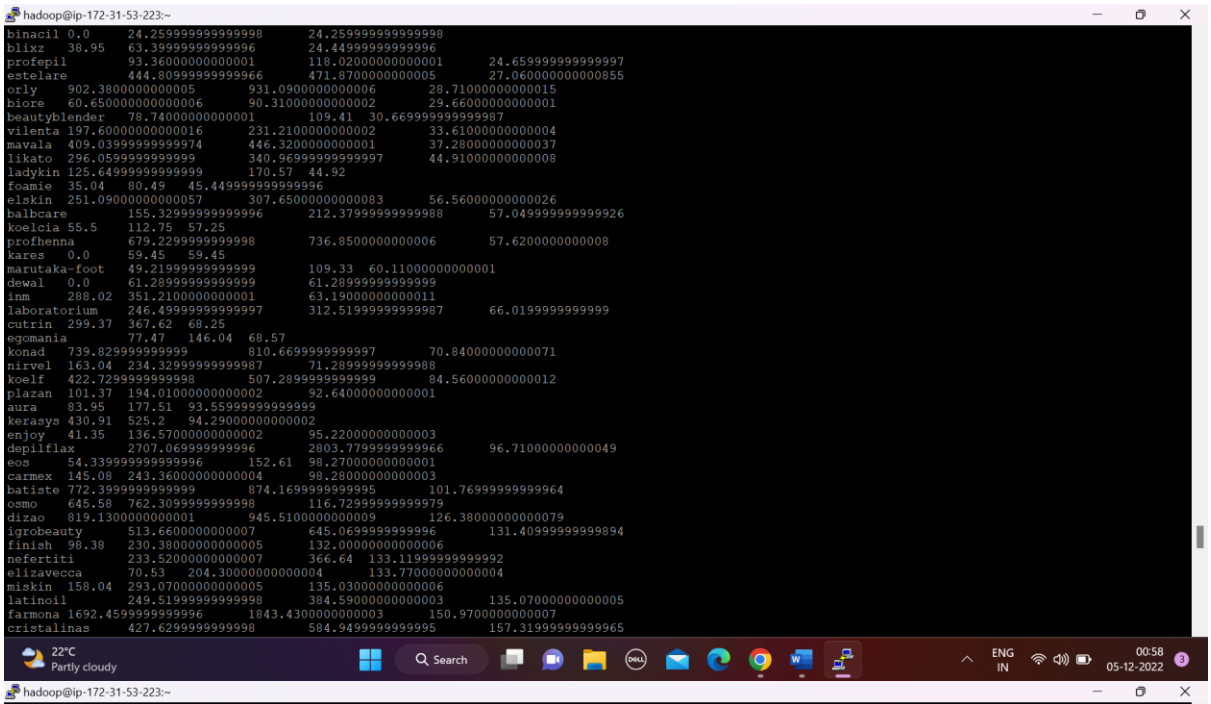
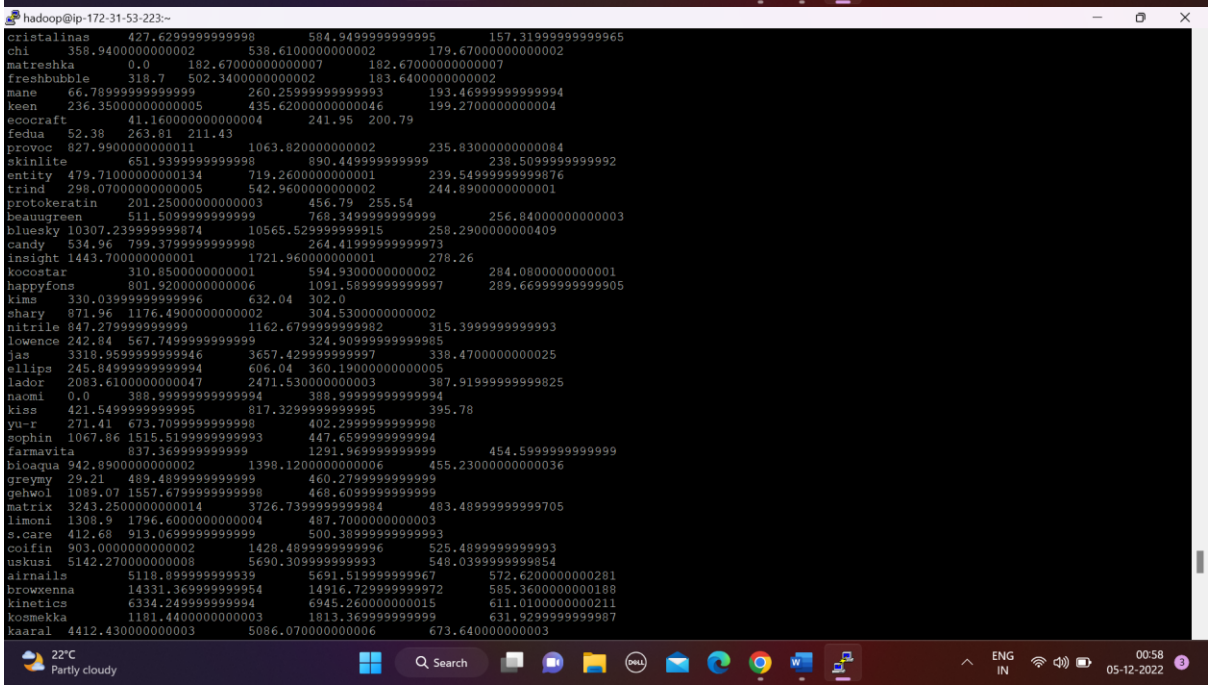
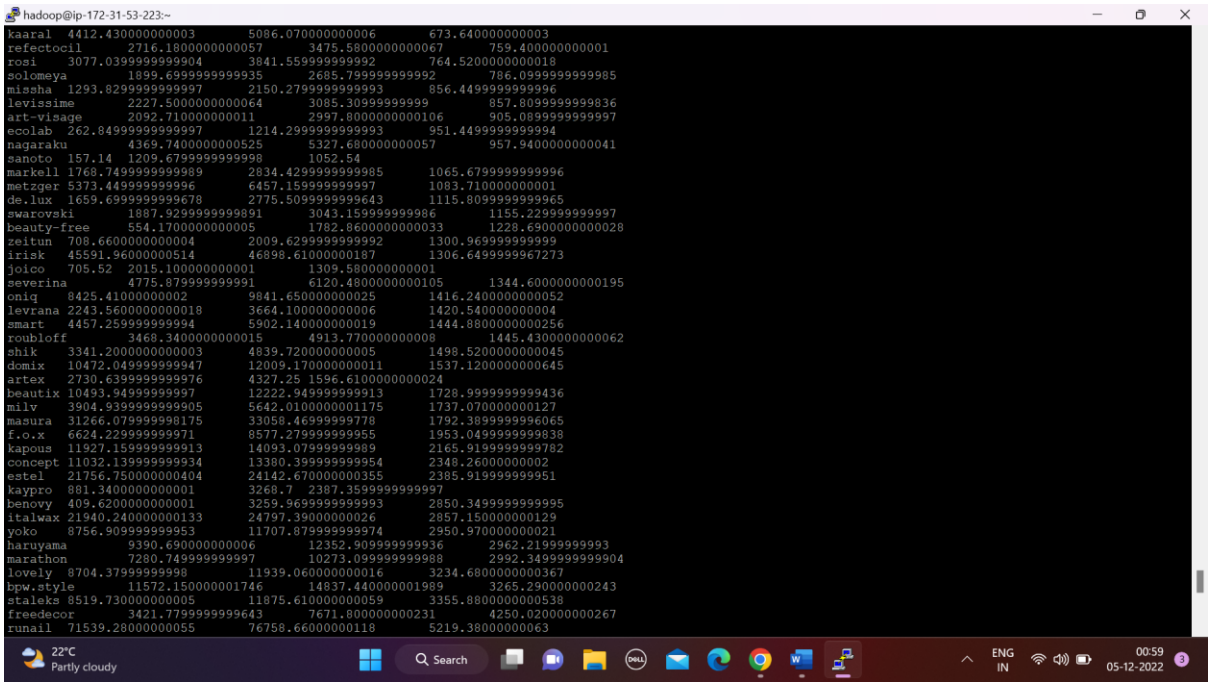


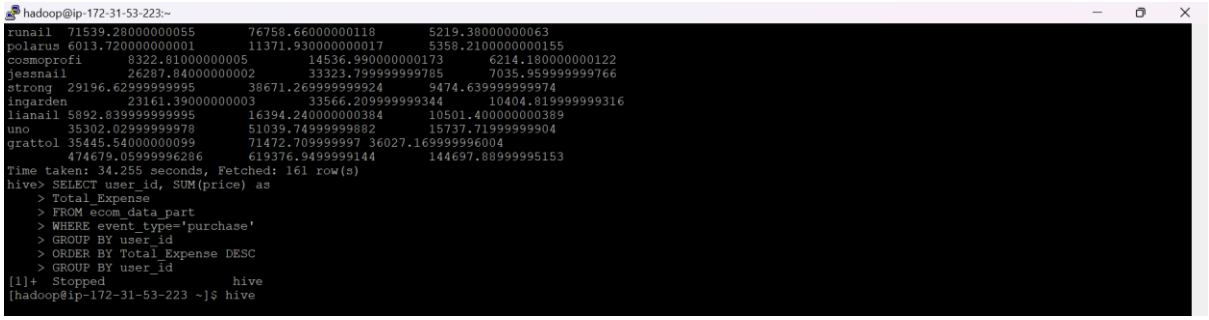
Inference:-
Runail is the brand which has the maximum price of 148297 /- in the two months October and November. Looks like cosmetics addicts more towards runail brand.

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

WITH Monthly_Revenue AS (
SELECT brand,
SUM(CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END)
AS Oct_Revenue, SUM(CASE WHEN date_format(event_time, 'MM')=11 THEN
price ELSE 0 END) AS Nov_Revenue FROM hive_table_part WHERE event_type='purchase' AND
date_format(event_time, 'MM') IN ('10', '11') GROUP BY brand) SELECT brand, Oct_Revenue, Nov_Revenue,
Nov_Revenue-Oct_Revenue AS Sales_Difference FROM Monthly_Revenue WHERE (Nov_Revenue -
Oct_Revenue)>0 ORDER BY Sales_Difference;




hadoop@ip-172-31-53-223:~
binacil 0.0 24.259999999999998 24.259999999999998
blixz 38.95 63.39999999999996 24.44999999999996
profepil 93.36000000000001 118.02000000000001 24.659999999999997
estelare 444.80999999999966 471.8700000000005 27.060000000000855
orly 902.3800000000005 931.0900000000006 28.71000000000015
biore 60.650000000000006 90.31000000000002 29.66000000000001
beautyblender 78.74000000000001 109.41 30.669999999999987
vilenta 197.60000000000016 231.2100000000002 33.61000000000004
mavala 409.03999999999974 446.3200000000001 37.28000000000037
likato 296.0599999999999 340.96999999999997 44.91000000000008
ladykin 125.64999999999999 170.57 44.92
foamie 35.04 80.49 45.449999999999996
elskin 251.09000000000057 307.65000000000083 56.56000000000026
balbcare 155.32999999999996 212.37999999999988 57.049999999999926
koelcia 55.5 112.75 57.25
profhenna 679.2299999999998 736.8500000000006 57.6200000000008
kares 0.0 59.45 59.45
marutaka-foot 49.21999999999999 109.33 60.11000000000001
dewal 0.0 61.28999999999999 61.28999999999999
inm 288.02 351.2100000000001 63.19000000000011
laboratorium 246.49999999999997 312.51999999999987 66.0199999999999
cutrin 299.37 367.62 68.25
egomania 77.47 146.04 68.57
konad 739.8299999999999 810.6699999999997 70.84000000000071
nirvel 163.04 234.32999999999987 71.28999999999988
koelf 422.7299999999998 507.2899999999999 84.56000000000012
plazan 101.37 194.01000000000002 92.64000000000001
aura 83.95 177.51 93.55999999999999
kerasys 430.91 525.2 94.29000000000002
enjoy 41.35 136.57000000000002 95.22000000000003
depilflax 2707.069999999996 2803.7799999999966 96.71000000000049
eos 54.339999999999996 152.61 98.27000000000001
carmex 145.08 243.36000000000004 98.28000000000003
batiste 772.3999999999999 874.1699999999995 101.76999999999964
osmo 645.58 762.3099999999998 116.72999999999979
dizao 819.1300000000001 945.5100000000009 126.38000000000079
igrobeauty 513.6600000000007 645.0699999999996 131.40999999999894
finish 98.38 230.38000000000005 132.00000000000006
nefertiti 233.52000000000007 366.64 133.11999999999992
elizavecca 70.53 204.30000000000004 133.77000000000004
miskin 158.04 293.07000000000005 135.03000000000006
latinoil 249.51999999999998 384.59000000000003 135.07000000000005
farmona 1692.4599999999996 1843.4300000000003 150.9700000000007
cristalinas 427.6299999999998 584.9499999999995 157.31999999999965
22°C
Partly cloudy
Search
ENG
IN
00:58
05-12-2022

hadoop@ip-172-31-53-223:~
cristalinas 427.6299999999998 584.9499999999995 157.31999999999965
chi 358.9400000000002 538.6100000000002 179.67000000000002
matreshka 0.0 182.67000000000007 182.67000000000007
freshbubble 318.7 502.3400000000002 183.6400000000002
mane 66.78999999999999 260.25999999999993 193.46999999999994
keen 236.35000000000005 435.62000000000046 199.2700000000004
ecocraft 41.160000000000004 241.95 200.79
fedua 52.38 263.81 211.43
provoc 827.9900000000011 1063.8200000000002 235.83000000000084
skinlite 651.9399999999998 890.4499999999999 238.50999999999992
entity 479.71000000000134 719.2600000000001 239.54999999999876
trind 298.07000000000005 542.9600000000002 244.8900000000001
protokeratin 201.25000000000003 456.79 255.54
beauugreen 511.5099999999999 768.3499999999999 256.84000000000003
bluesky 10307.239999999874 10565.529999999915 258.2900000000409
candy 534.96 799.3799999999998 264.41999999999973
insight 1443.7000000000001 1721.9600000000001 278.26
kocostar 310.8500000000001 594.9300000000002 284.0800000000001
happyfons 801.9200000000006 1091.5899999999997 289.66999999999905
kims 330.03999999999996 632.04 302.0
shary 871.96 1176.4900000000002 304.5300000000002
nitrile 847.279999999999 1162.6799999999982 315.3999999999993
lowence 242.84 567.7499999999999 324.90999999999985
jas 3318.9599999999946 3657.429999999997 338.4700000000025
ellips 245.84999999999994 606.04 360.19000000000005
lador 2083.6100000000047 2471.530000000003 387.91999999999825
naomi 0.0 388.99999999999994 388.99999999999994
kiss 421.5499999999995 817.3299999999995 395.78
yu-r 271.41 673.7099999999998 402.2999999999998
sophin 1067.86 1515.5199999999993 447.6599999999994
farmavita 837.369999999999 1291.969999999999 454.5999999999999
bioaqua 942.8900000000002 1398.1200000000006 455.23000000000036
greymy 29.21 489.4899999999999 460.2799999999999
gehwol 1089.07 1557.6799999999998 468.6099999999999
matrix 3243.2500000000014 3726.7399999999984 483.48999999999705
limoni 1308.9 1796.6000000000004 487.7000000000003
s.care 412.68 913.0699999999999 500.38999999999993
coifin 903.0000000000002 1428.4899999999996 525.4899999999993
uskusi 5142.270000000008 5690.309999999993 548.0399999999854
airnails 5118.899999999939 5691.519999999967 572.6200000000281
browxenna 14331.369999999954 14916.729999999972 585.3600000000188
kinetics 6334.249999999994 6945.260000000015 611.0100000000211
kosmekka 1181.4400000000003 1813.369999999999 631.9299999999987
kaaral 4412.430000000003 5086.070000000006 673.6400000000003
22°C
Partly cloudy
Search
ENG
IN
00:58
05-12-2022

hadoop@ip-172-31-53-223:~
kaaral 4412.430000000003 5086.070000000006 673.6400000000003
refectocil 2716.1800000000057 3475.5800000000067 759.4000000000001
rosi 3077.0399999999904 3841.5599999999992 764.5200000000018
solomeya 1899.6999999999935 2685.7999999999992 786.0999999999985
missha 1293.8299999999997 2150.2799999999993 856.4499999999996
levissime 2227.5000000000064 3085.309999999999 857.8099999999836
art-visage 2092.7100000000011 2997.8000000000106 905.0899999999997
ecolab 262.84999999999997 1214.2999999999993 951.4499999999994
nagaraku 4369.7400000000525 5327.680000000057 957.9400000000041
sanoto 157.14 1209.6799999999998 1052.54
markell 1768.7499999999989 2834.4299999999985 1065.6799999999996
metzger 5373.449999999996 6457.159999999997 1083.7100000000001
de.lux 1659.6999999999678 2775.5099999999643 1115.8099999999965
swarovski 1887.9299999999891 3043.1599999999986 1155.2299999999997
beauty-free 554.1700000000005 1782.8600000000033 1228.6900000000028
zeitun 708.6600000000004 2009.6299999999992 1300.969999999999
irisk 45591.96000000514 46898.61000000187 1306.6499999967273
joico 705.52 2015.1000000000001 1309.5800000000001
severina 4775.879999999991 6120.4800000000105 1344.6000000000195
oniq 8425.41000000002 9841.650000000025 1416.2400000000052
levrana 2243.5600000000018 3664.100000000006 1420.540000000004
smart 4457.259999999994 5902.140000000019 1444.8800000000256
roubloff 3468.3400000000015 4913.770000000008 1445.4300000000062
shik 3341.2000000000003 4839.720000000005 1498.5200000000045
domix 10472.049999999947 12009.170000000011 1537.1200000000645
artex 2730.6399999999976 4327.25 1596.6100000000024
beautix 10493.94999999997 12222.949999999913 1728.9999999999436
milv 3904.9399999999905 5642.0100000001175 1737.070000000127
masura 31266.079999998175 33058.46999999778 1792.3899999996065
f.o.x 6624.229999999971 8577.279999999955 1953.0499999999838
kapous 11927.159999999913 14093.07999999989 2165.9199999999782
concept 11032.139999999934 13380.399999999954 2348.26000000002
estel 21756.750000000404 24142.670000000355 2385.919999999951
kaypro 881.3400000000001 3268.7 2387.3599999999997
benovy 409.6200000000001 3259.9699999999993 2850.3499999999995
italwax 21940.240000000133 24797.39000000026 2857.150000000129
yoko 8756.909999999953 11707.879999999974 2950.970000000021
haruyama 9390.690000000006 12352.909999999936 2962.21999999993
marathon 7280.749999999997 10273.099999999988 2992.3499999999904
lovely 8704.37999999998 11939.060000000016 3234.6800000000367
bpw.style 11572.150000001746 14837.440000001989 3265.290000000243
staleks 8519.730000000005 11875.610000000059 3355.8800000000538
freedecor 3421.7799999999643 7671.800000000231 4250.020000000267
runail 71539.28000000055 76758.66000000118 5219.38000000063
22°C
Partly cloudy
Search
ENG
IN
00:59
05-12-2022

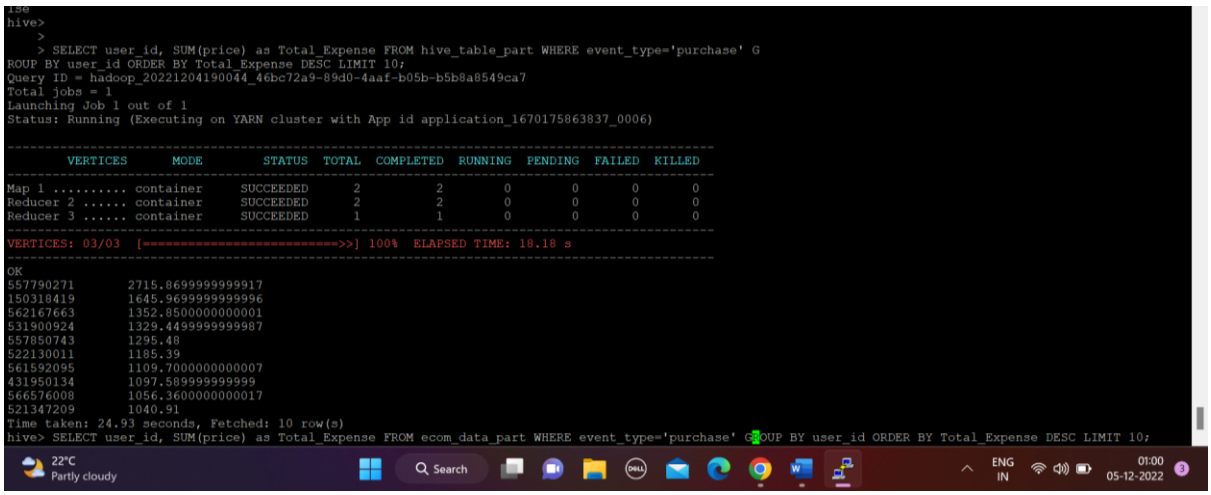


Inference: There are 161 brands for which the sales increased from October to November. Ovale showing the least difference and Grattol showing the maximum difference. Runail is also among the brands which showed increase in sales. So runail can be claimed as a popular brand.

Query 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.

Query – SELECT user_id, SUM(price) as Total_Expense FROM hive_table_part WHERE event_type='purchase' GROUP BY user_id ORDER BY Total_Expense DESC LIMIT 10;



Inference:

The output gives the list of top 10 customers who can be rewarded the Golden customer plan.

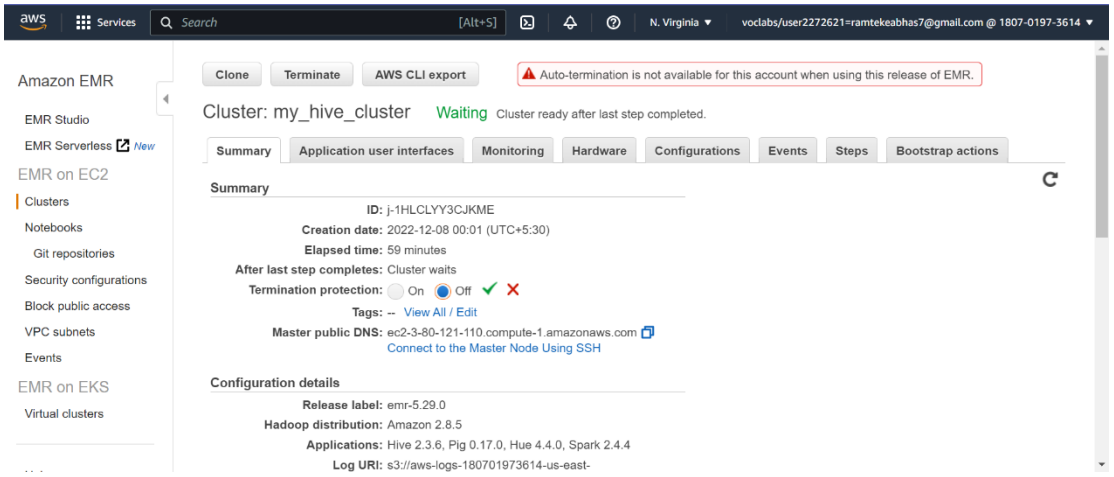
6. Cleaning Up:

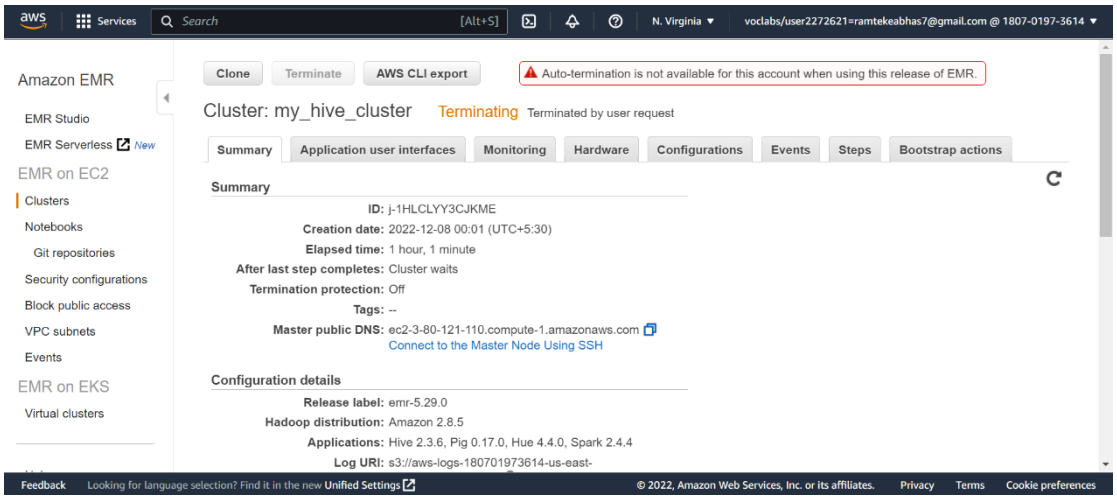
Dropping the database.

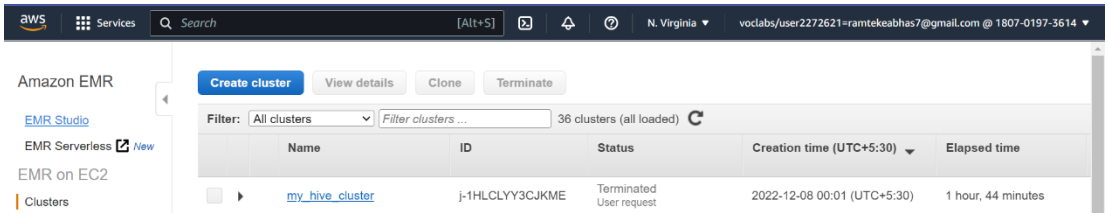
Drop database ABDB;

```
hive> DROP DATABASE IF EXISTS ABDB;
OK
Time taken: 0.159 seconds
hive>
```

6.1. Once the analysis is done, we can terminate the cluster by changing the Termination protection from **ON** to **OFF** and thenclick on the terminate button.







Summary:

The public clickstream dataset of a cosmetics store is used for this analysis.

Steps involved:

1. Copying the data set into the HDFS from S3 Bucket.
2. Creating the database and launching Hive queries on EMR cluster.
3. Necessary optimizations are done, such as selecting the appropriate table format and using partitioned/bucketed tables. CSV Serde is used with the default properties value for loading the dataset into a Hive table.
4. Runail seems to have increased sales from October to November
5. Cleaning up.

Overall, there is increase in sales from October to November.