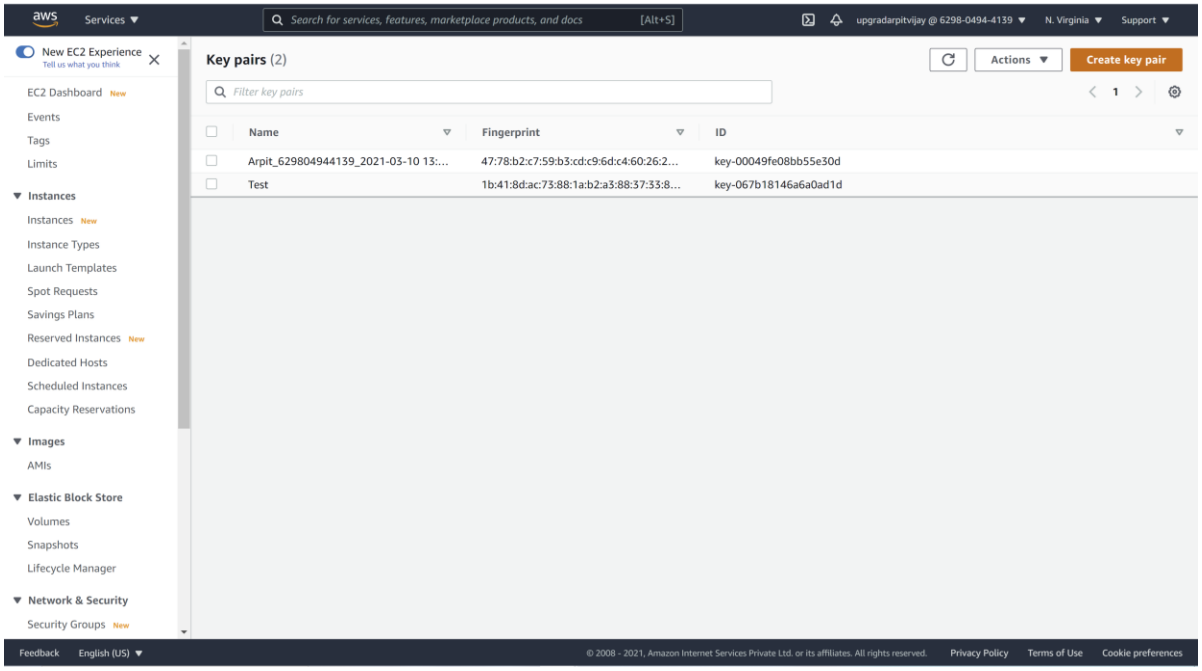
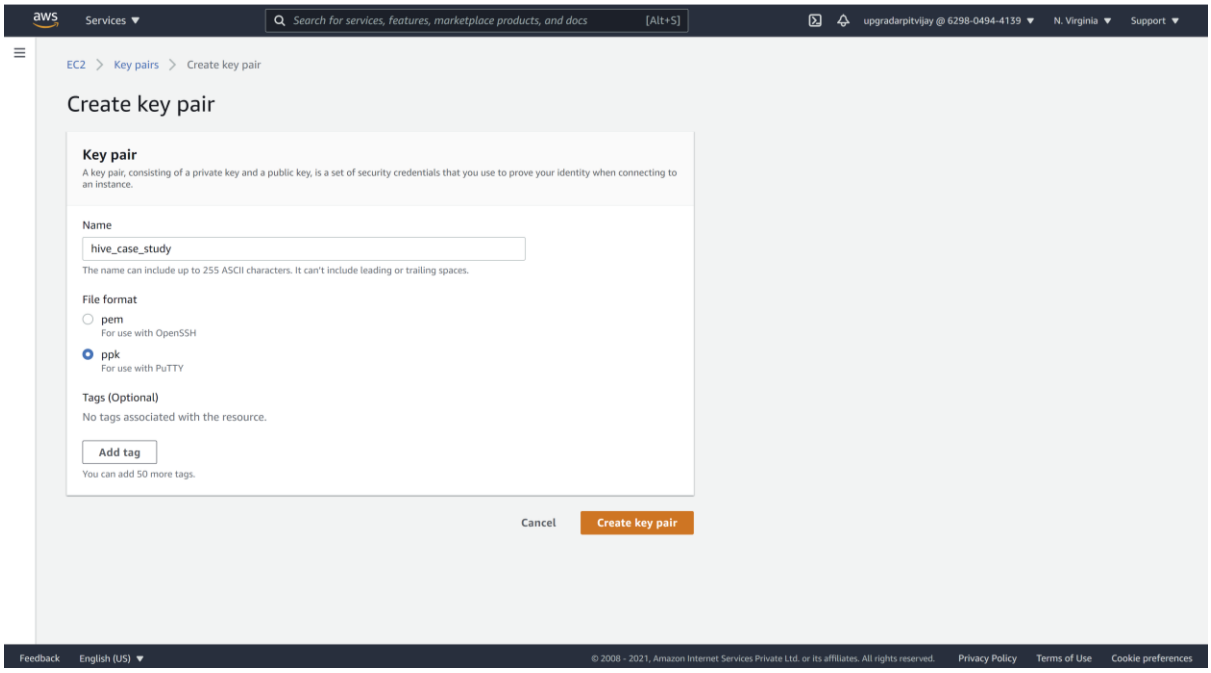
HIVE CASE STUDY STEPS

- **Key-pair creation and generation**

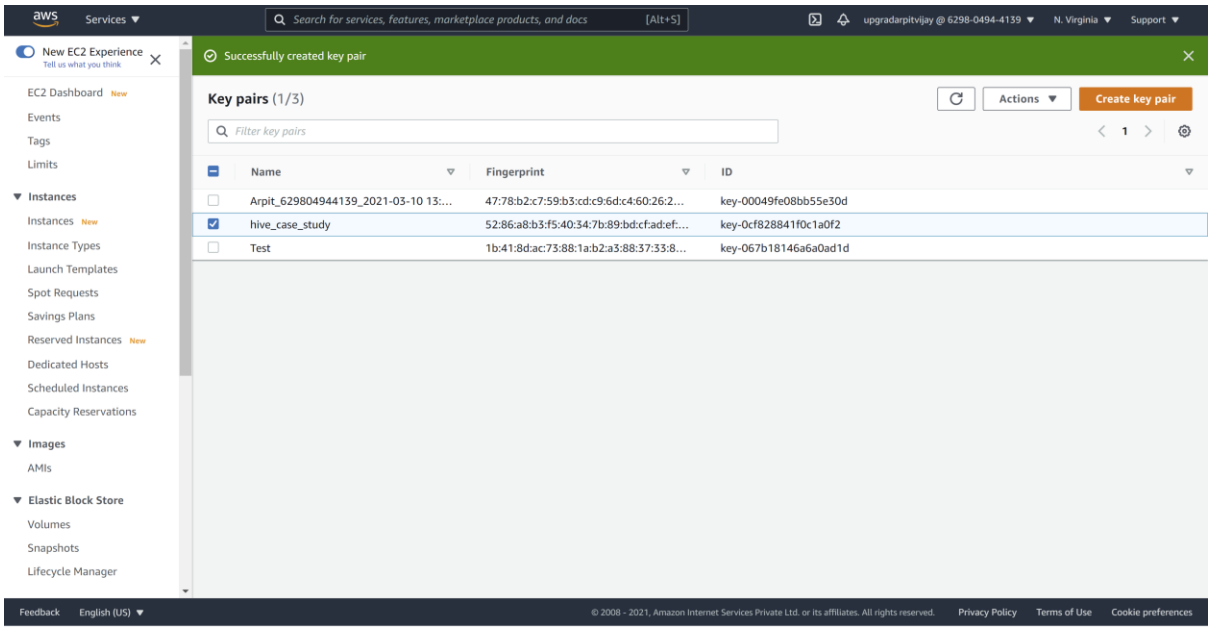
1.Click on Create Key pair



2.Mention the name of a Key Pair with ppk file format and click create.

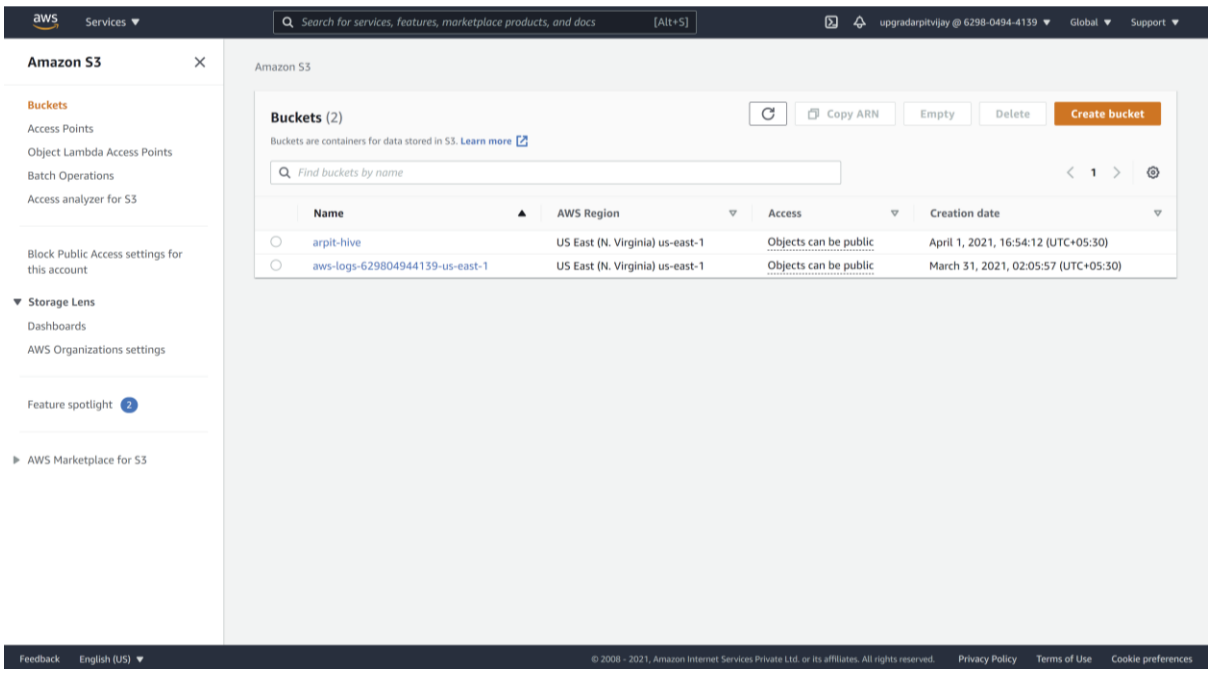


3. Successfully Key Pair created.

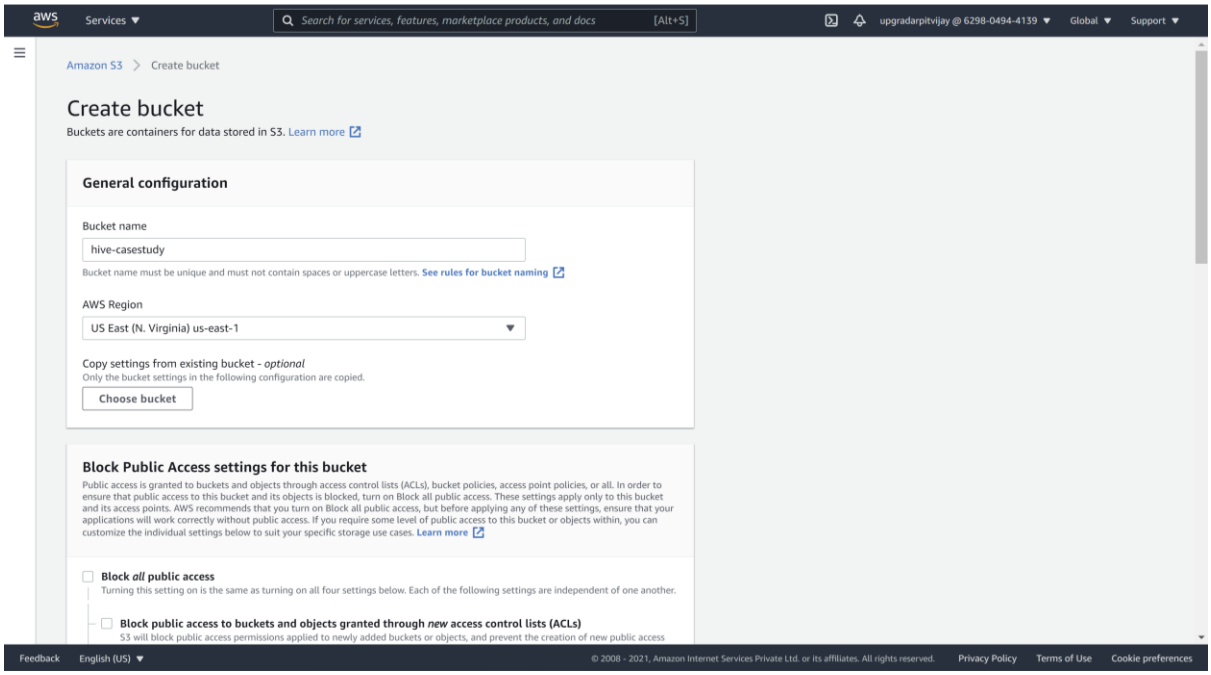


- **S3 Bucket to store data files**

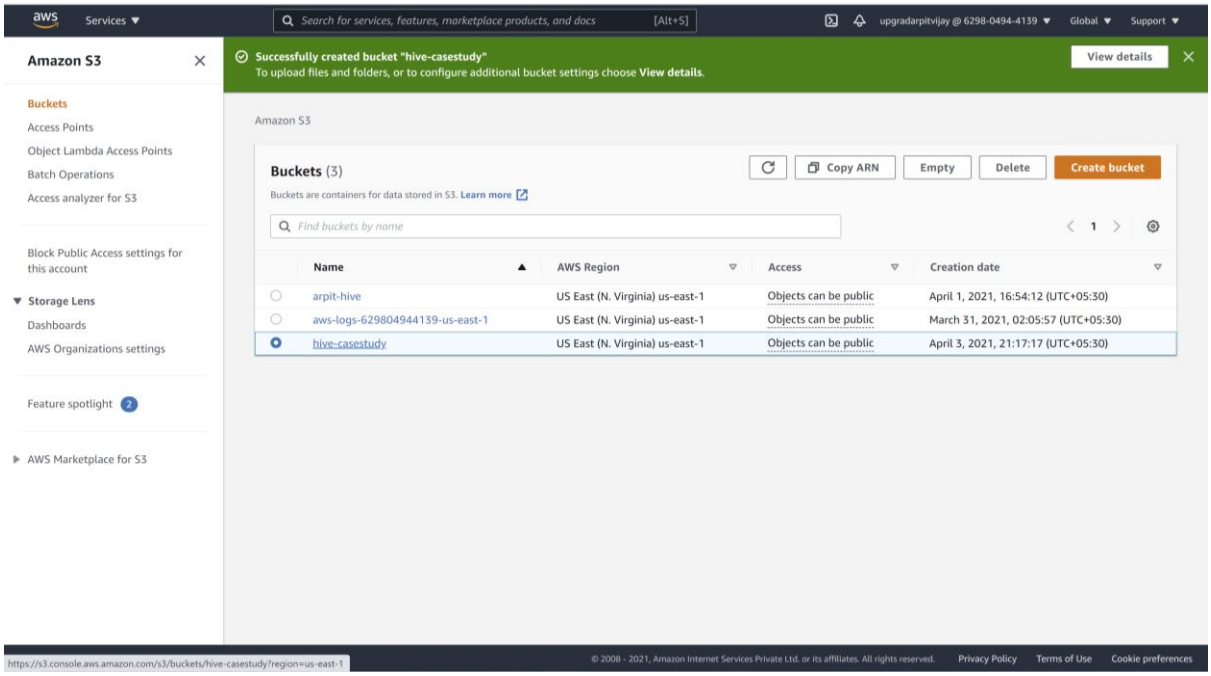
1.First we will create a new bucket by clicking on Create Bucket



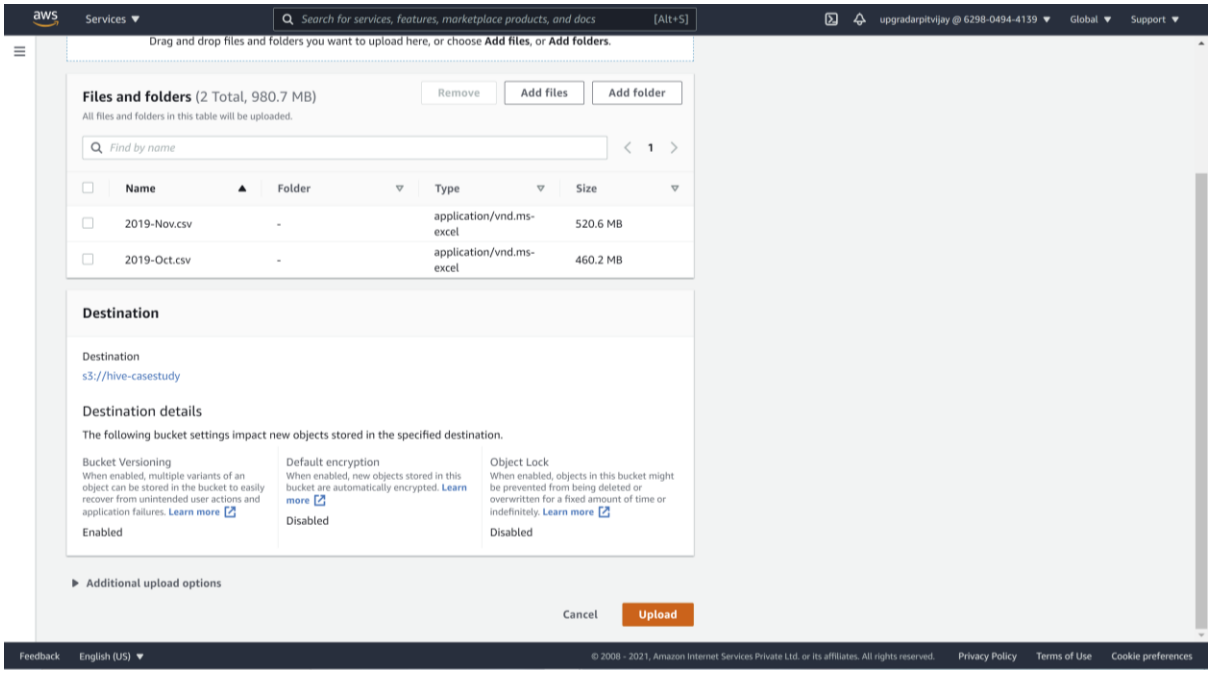
2. Provide the unique name to the Bucket and uncheck the block all public access box for this case study.

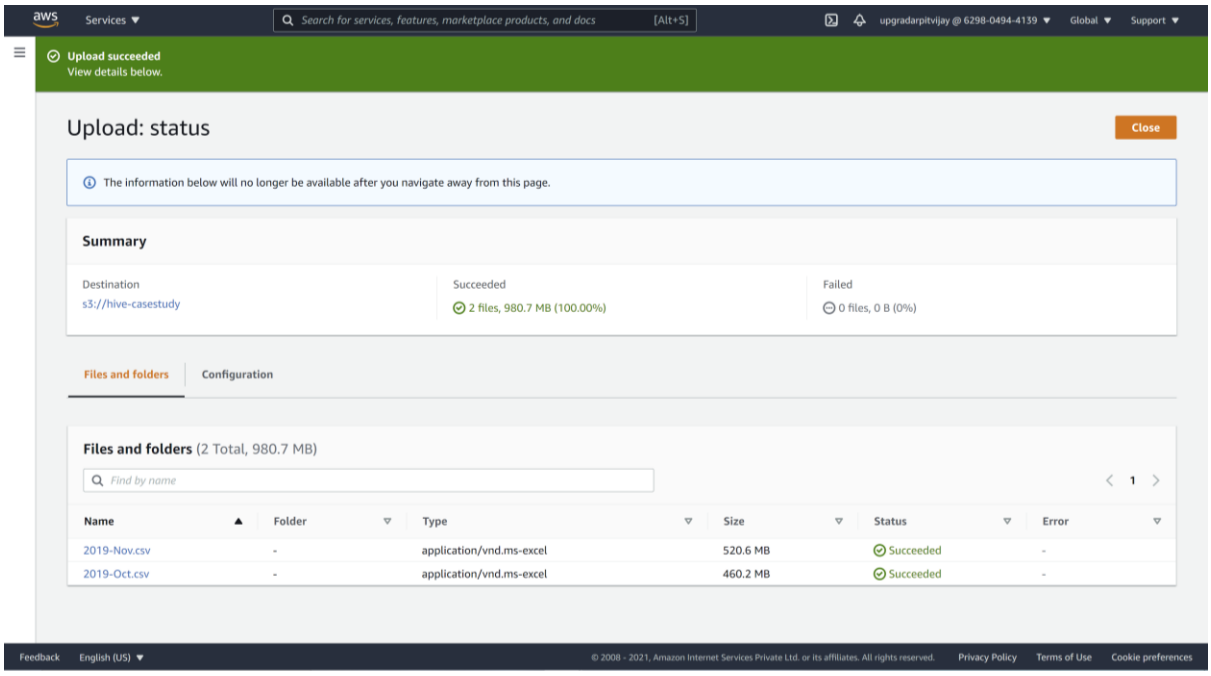


3. New Bucket created successfully with name “hive-casestudy”.



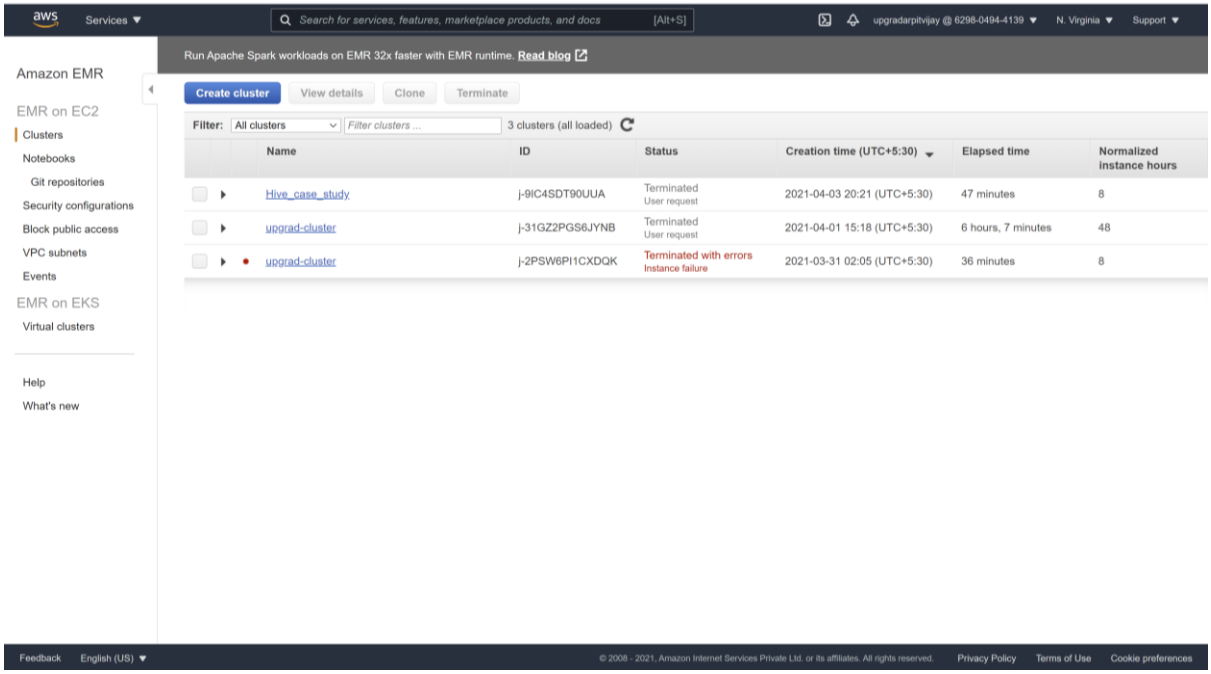
4. Upload the csv files to the new bucket created.



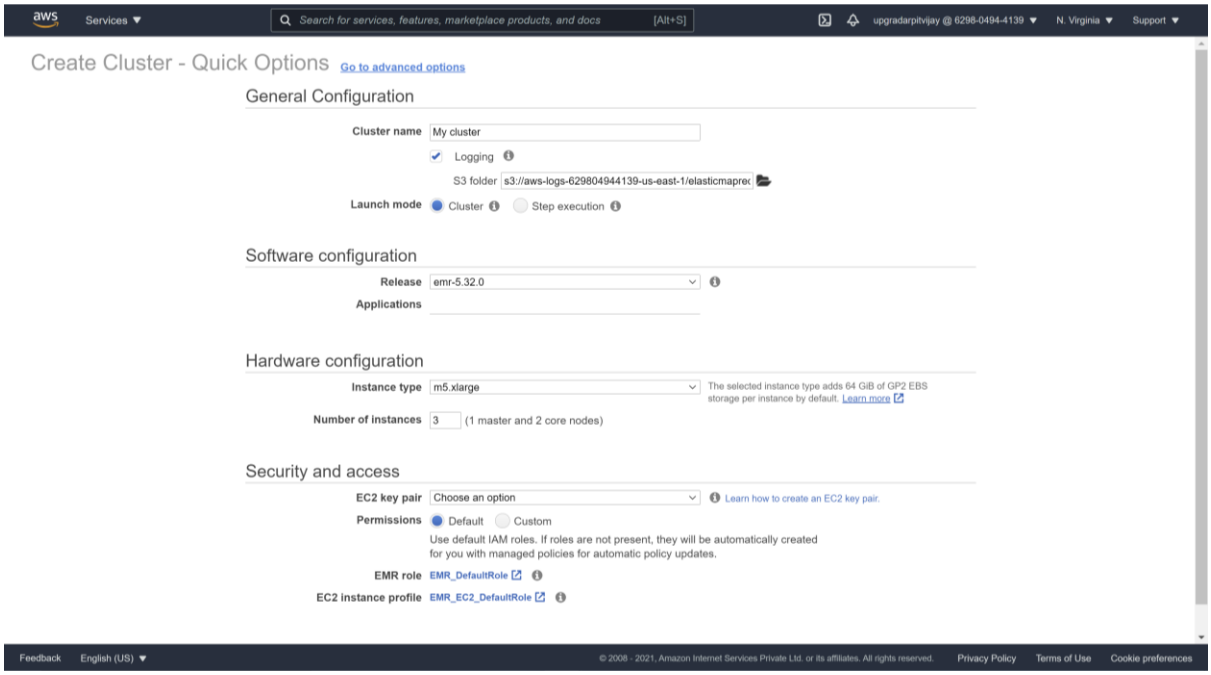


- **CLUSTER CREATION**

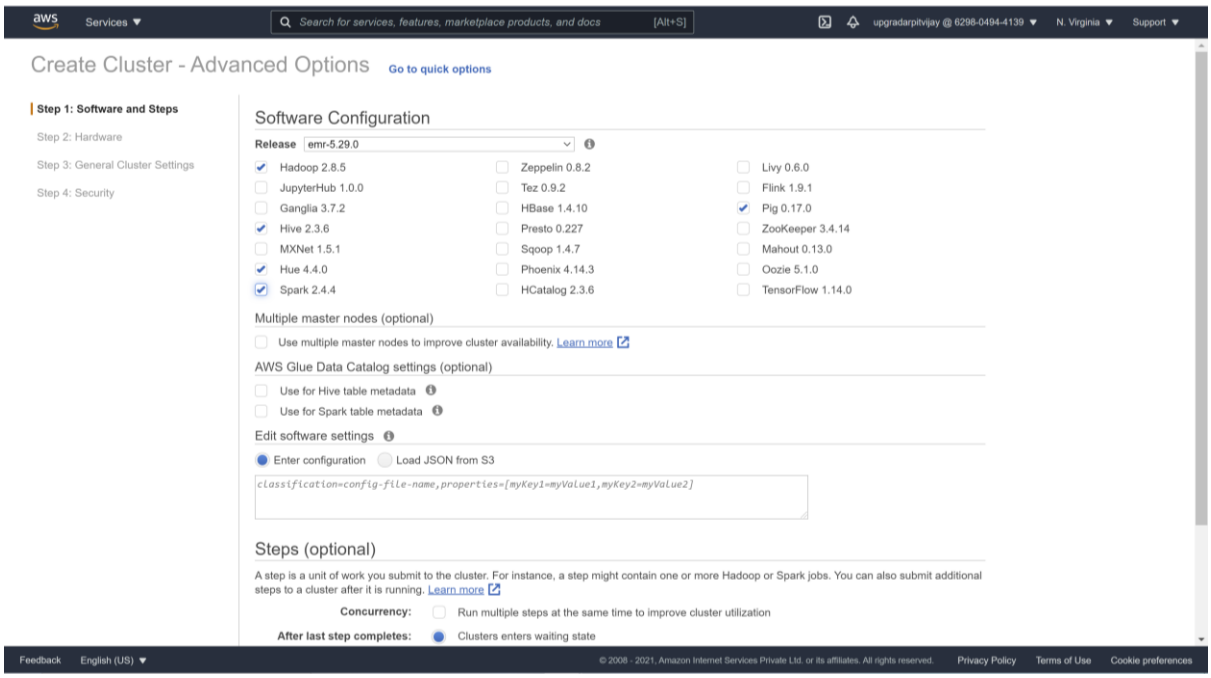
1. In Amazon EMR home page click on create cluster button.



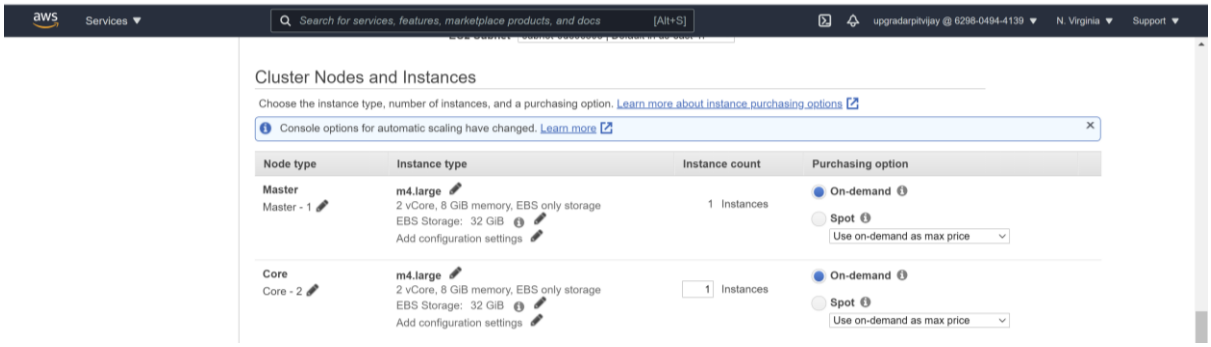
2. Then click on 'Go to advanced options'.



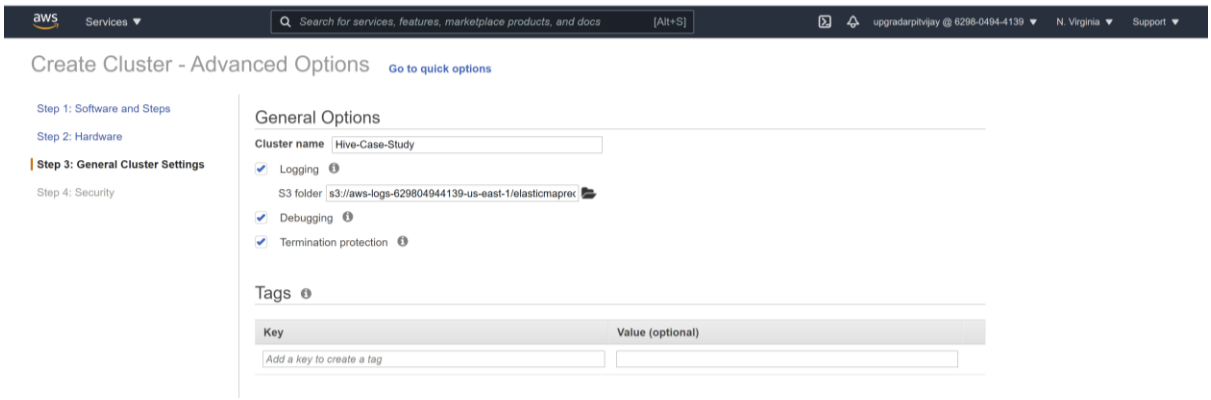
3. In software and steps page we have to select all the required services needed to complete this analysis. We are selecting emr-5.29.0 release as per the instruction with required services.



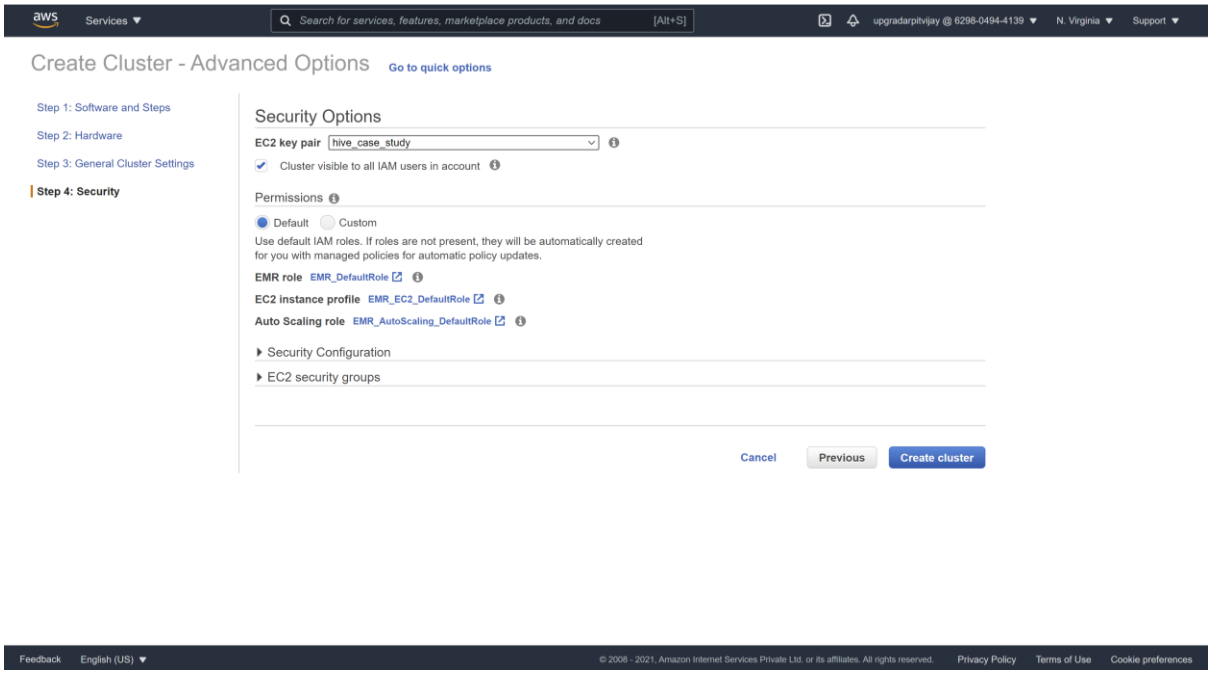
4. Then we will proceed to next page, in this step we have to specify the required configuration needed for the cluster. Here we have to select the instance type and instance count for master and core node. We have selected instance type as m4.large and 1 instance count for both master and core node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



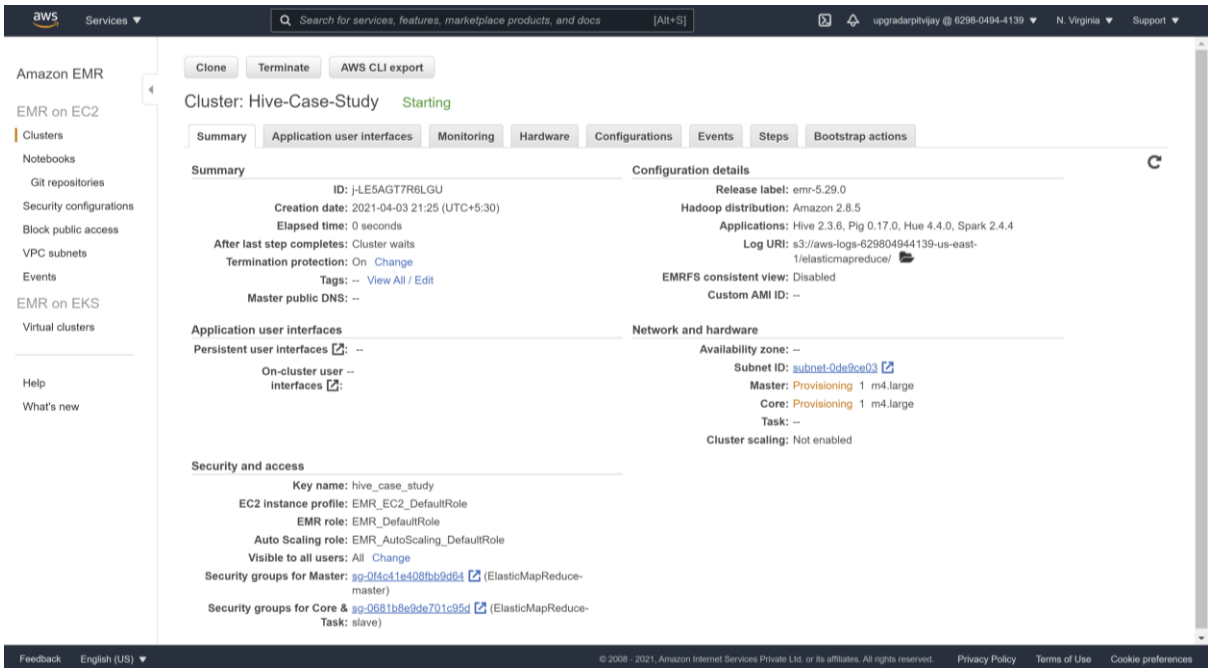
5. Here we have given the name of the cluster as “Hive-Case-Study”.



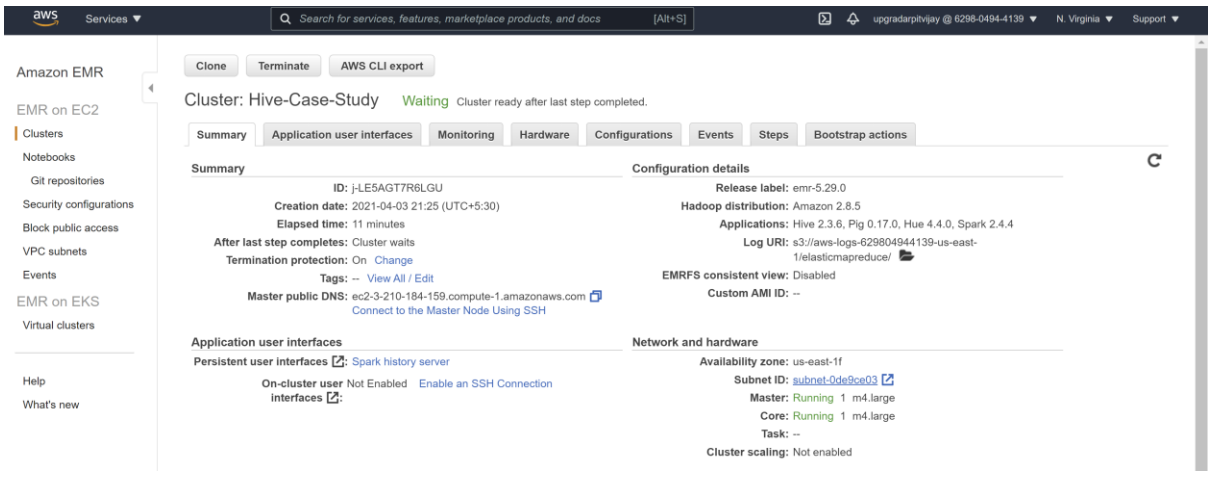
6. We have selected the created EC2 key pair and then click on the create cluster option.



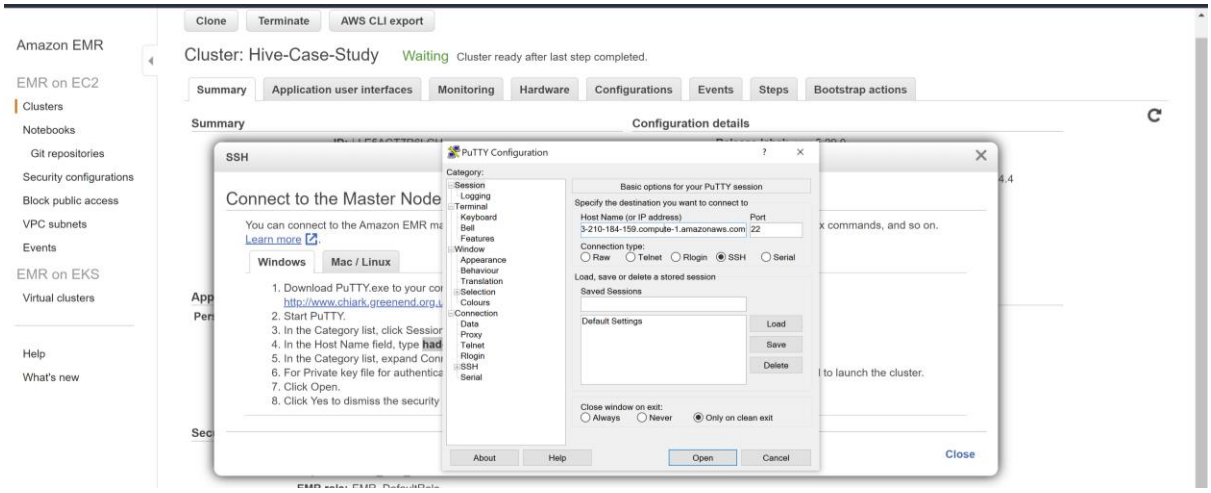
7. Then cluster has been created and will be in the “Starting” state and will change to “Waiting“ state after sometime.

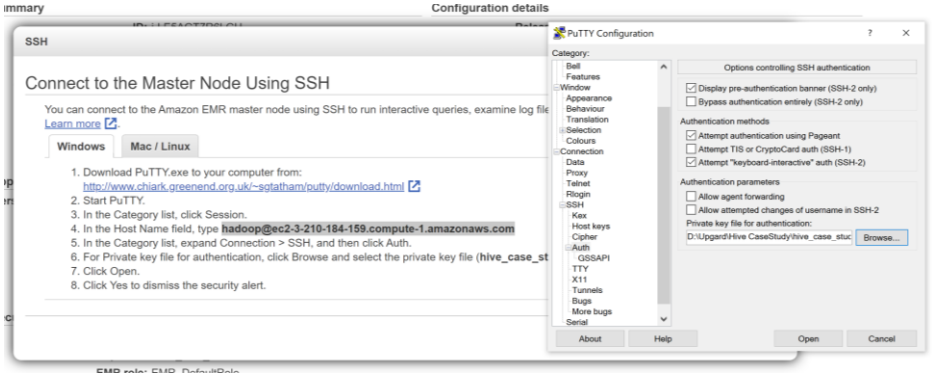


8. Now cluster is started as Master and Core both are in “Running” State.

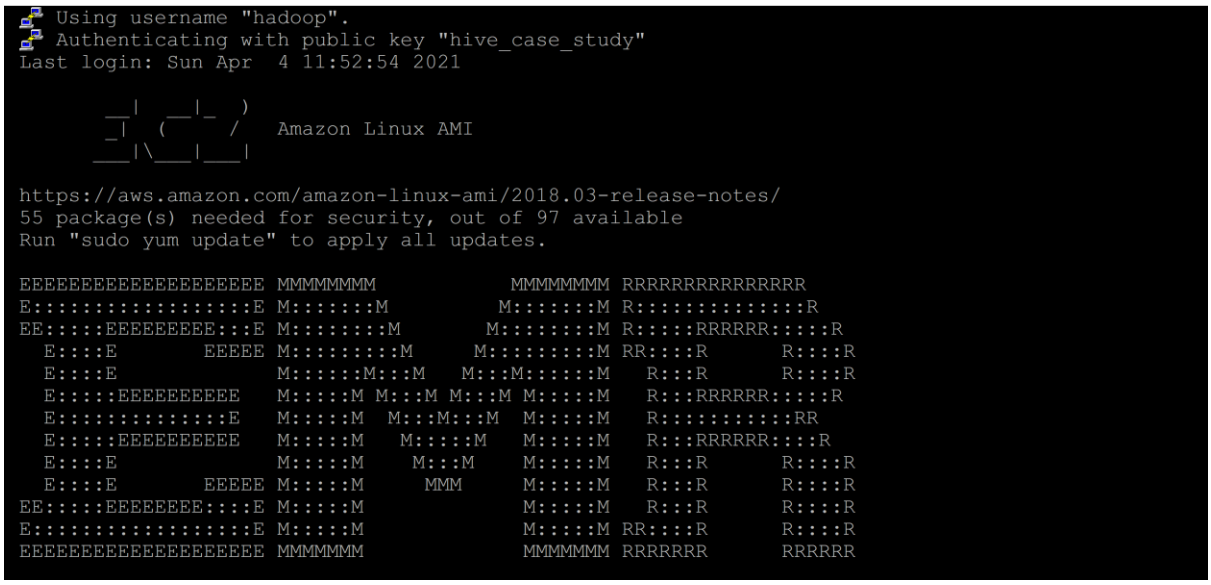


9. Once cluster in running state we have to click on Master public DNS. We have to open the putty configuration and then give the host name (master node DNS) and then browse to the private key file location by clicking on Connection → SSH → Auth.





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



- **Copying the data into HDFS**

1.To access the public s3 bucket.
Command: aws s3 ls hive-casestudy

```
[hadoop@ip-172-31-78-144 ~]$ aws s3 ls hive-casestudy
2021-04-03 15:48:16  545839412 2019-Nov.csv
2021-04-03 15:48:16  482542278 2019-Oct.csv
[hadoop@ip-172-31-78-144 ~]$
```

2.Creating a directory in HDFS.
Command: hadoop fs -mkdir /user/hive/ecomcasestudy

3.Checking the available directory.
Command: hadoop fs -ls /user/hive

```
hadoop@ip-172-31-72-97:~
[hadoop@ip-172-31-72-97 ~]$ hadoop fs -mkdir /user/hive/ecomcasestudy
[hadoop@ip-172-31-72-97 ~]$ hadoop fs -ls /user/hive
Found 2 items
drwxr-xr-x   - hadoop hadoop          0 2021-04-04 11:56 /user/hive/ecomcasestudy
drwxrwxrwt   - hdfs   hadoop          0 2021-04-04 11:48 /user/hive/warehouse
```

4.Loading the s3 public data set to created directory “ecomcasestudy” in hadoop .

Command: hadoop distcp 's3://hive-casestudy/2019-Oct.csv' /user/hive/ecomcasestudy/oct.csv

hadoop distcp 's3://hive-casestudy/2019-Nov.csv' /user/hive/ecomcasestudy/nov.csv

```
[hadoop@ip-172-31-72-97 ~]$ hadoop distcp 's3://hive-casestudy/2019-Nov.csv' /user/hive/ecomcasestudy/nov.csv
21/04/04 12:00:22 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=fa
lse, ignoreFailures=false, overwrite=false, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwid
th=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false, atomicWorkP
ath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3://hive-casestudy/2019-Nov.csv], targetPath=/user/hive/ec
omcasestudy/nov.csv, targetPathExists=false, filtersFile='null'}
21/04/04 12:00:22 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-72-97.ec2.internal/172.31.72.97:8032
21/04/04 12:00:26 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
21/04/04 12:00:26 INFO tools.SimpleCopyListing: Build file listing completed.
21/04/04 12:00:26 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
21/04/04 12:00:26 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.fact
or
21/04/04 12:00:26 INFO tools.DistCp: Number of paths in the copy list: 1
21/04/04 12:00:26 INFO tools.DistCp: Number of paths in the copy list: 1
21/04/04 12:00:26 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-72-97.ec2.internal/172.31.72.97:8032
21/04/04 12:00:26 INFO mapreduce.JobSubmitter: number of splits:1
21/04/04 12:00:27 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1617537007743_0002
21/04/04 12:00:27 INFO impl.YarnClientImpl: Submitted application application_1617537007743_0002
21/04/04 12:00:27 INFO mapreduce.Job: The url to track the job: http://ip-172-31-72-97.ec2.internal:20888/proxy/applicat
ion_1617537007743_0002/
21/04/04 12:00:27 INFO tools.DistCp: DistCp job-id: job_1617537007743_0002
21/04/04 12:00:27 INFO mapreduce.Job: Running job: job_1617537007743_0002
21/04/04 12:00:35 INFO mapreduce.Job: Job job_1617537007743_0002 running in uber mode : false
21/04/04 12:00:35 INFO mapreduce.Job:  map 0% reduce 0%
21/04/04 12:00:52 INFO mapreduce.Job:  map 100% reduce 0%
21/04/04 12:00:54 INFO mapreduce.Job: Job job_1617537007743_0002 completed successfully
21/04/04 12:00:54 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=172837
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=352
                HDFS: Number of bytes written=545839412
                HDFS: Number of read operations=12
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                S3: Number of bytes read=545839412
                S3: Number of bytes written=0
```

```
                FILE: Number of write operations=0
                HDFS: Number of bytes read=352
                HDFS: Number of bytes written=545839412
                HDFS: Number of read operations=12
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                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)=545376
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=17043
                Total vcore-milliseconds taken by all map tasks=17043
                Total megabyte-milliseconds taken by all map tasks=17452032
        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)=299
                CPU time spent (ms)=21030
                Physical memory (bytes) snapshot=617218048
                Virtual memory (bytes) snapshot=3300544512
                Total committed heap usage (bytes)=515375104
        File Input Format Counters
                Bytes Read=217
        File Output Format Counters
                Bytes Written=0
        DistCp Counters
                Bytes Copied=545839412
                Bytes Expected=545839412
                Files Copied=1
[hadoop@ip-172-31-72-97 ~]$
```

5.After loading the data set we have used below command to check the data set file and data set in the hadoop directory.

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

Command: hadoop fs -cat /user/hive/ecomcasestudy/oct.csv | head

```
hadoop@ip-172-31-72-97:~
[hadoop@ip-172-31-72-97 ~]$ hadoop fs -cat /user/hive/ecomcasestudy/oct.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-10-01 00:00:00 UTC,cart,5773203,1487580005134238553,,runail,2.62,463240011,26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC,cart,5773353,1487580005134238553,,runail,2.62,463240011,26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC,cart,5881589,2151191071051219817,,lovely,13.48,429681830,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC,cart,5723490,1487580005134238553,,runail,2.62,463240011,26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:15 UTC,cart,5881449,1487580013522845895,,lovely,0.56,429681830,49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:16 UTC,cart,5857269,1487580005134238553,,runail,2.62,430174032,73dea1e7-664e-43f4-8b30-d32b9d5af04f
2019-10-01 00:00:19 UTC,cart,5739055,1487580008246412266,,kapous,4.75,377667011,81326ac6-daa4-4f0a-b488-fd0956a78733
2019-10-01 00:00:24 UTC,cart,5825598,1487580009445982239,,,0.56,467916806,2f5b5546-b8cb-9ee7-7ecd-84276f8ef486
2019-10-01 00:00:25 UTC,cart,5698989,1487580006317032337,,,1.27,385985999,d30965e8-1101-44ab-b45d-cc1bb9fae694
cat: Unable to write to output stream.
[hadoop@ip-172-31-72-97 ~]$
```

Command: hadoop fs -cat /user/hive/ecomcasestudy/nov.csv | head

```
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-11-01 00:00:02 UTC,view,5802432,1487580009286598681,,,0.32,562076640,09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC,cart,5844397,1487580006317032337,,,2.38,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC,view,5837166,1783999064103190764,,pnb,22.22,556138645,57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC,cart,5876812,1487580010100293687,,jessnail,3.16,564506666,186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34f
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34f
2019-11-01 00:00:25 UTC,view,5856189,1487580009026551821,,runail,15.71,562076640,09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:32 UTC,view,5837835,1933472286753424063,,,3.49,514649199,432a4e95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC,remove_from_cart,5870838,1487580007675986893,,milv,0.79,429913900,2f0bff3c-252f-4fe6-afcd-5d8a6a
```

- ## Database & Table Creation

  Create the Database

```
hive> create database ECOMDB;
OK
Time taken: 0.225 seconds
hive>
```

1. After moving the data to the directory, we create the base table(ecom_data) and check for the data in the table.

Command:
CREATE TABLE IF NOT EXISTS ecom_data (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/ecomcasestudy/' tblproperties('skip.header.line.count'='1');

```
hive> CREATE TABLE IF NOT EXISTS ecom_data (event_time timestamp, event_type string , product_id string , category_id st
ring , 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/ecomcasestudy/' tblproperties('skip.header.line.count'='1');
OK
Time taken: 1.04 seconds
hive>
```

```
hive> select * from ecom_data limit 5;
OK
event_time      event_type      product_id      category_id     category_code   brand   price   user_id user_session
2019-11-01 00:00:02 UTC view    5802432 1487580009286598681                     0.32    562076640       09fafd6c-6c99-46
b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart    5844397 1487580006317032337                     2.38    553329724       2067216c-31b5-45
5d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC view    5837166 1783999064103190764             pnb     22.22   556138645       57ed222e-a54a-49
07-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart    5876812 1487580010100293687             jessnail        3.16    564506666       186c1951
-8052-4b37-adce-dd9644b1d5f7
Time taken: 2.096 seconds, Fetched: 5 row(s)
```

2. Once the base table is created, we need to optimize the table for quick query result through partitioning and bucketing. Our optimized table name is **ecom_data_part**.

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

Command:

CREATE TABLE IF NOT EXISTS ecom_data_part (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 (category_code) into 12 buckets
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS TEXTFILE
LOCATION '/user/hive/ecomcasestudy/' tblproperties('skip.header.line.count'='1');

```
hive> CREATE TABLE IF NOT EXISTS ecom_data_part (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 (category_code) into 12 buckets
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > STORED AS TEXTFILE
    > LOCATION '/user/hive/ecomcasestudy/' tblproperties('skip.header.line.count'='1');
OK
Time taken: 0.11 seconds
hive>
```

3. Loading the data into optimize table from base table.

Command:

Insert into table ecom_data_part partition (event_type)
select event_time , product_id , category_id , category_code , brand , price , user_id , user_session , event_type from ecom_data ;

```
hive> Insert into table ecom_data_part partition (event_type)
    > select event_time , product_id , category_id , category_code , brand , price , user_id , user_session , event_type
 from ecom_data ;
Query ID = hadoop_20210404121016_accd5532-d54a-45f1-8df1-b77b1b0ac3e4
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_1617537007743_0004)

--------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 159.79 s
--------------------------------------------------------------------------------
Loading data to table default.ecom_data_part partition (event_type=null)

Loaded : 5/5 partitions.
         Time taken to load dynamic partitions: 1.199 seconds
         Time taken for adding to write entity : 0.004 seconds
OK
Time taken: 173.433 seconds
hive>
```

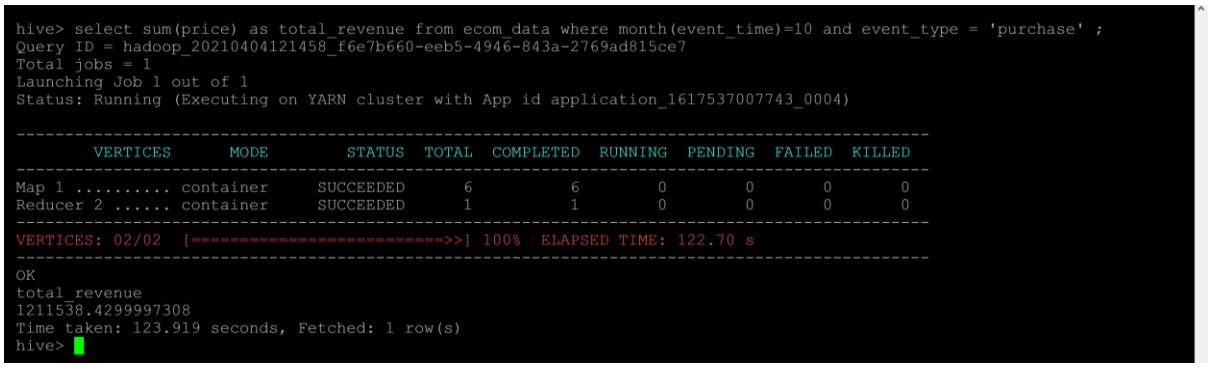
```
hive> set hive.cli.print.header=true ;
hive> select * from ecom_data_part limit 5;
OK
ecom_data_part.event_time       ecom_data_part.product_id       ecom_data_part.category_id      ecom_data_part.category_
code    ecom_data_part.brand    ecom_data_part.price    ecom_data_part.user_id  ecom_data_part.user_session     ecom_dat
a_part.event_type
2019-10-27 10:48:31 UTC 5812230 1487580005427839846             irisk   2.48    556028109       61ead6df-aac2-4167-8d4d-
484a3337aa1c    cart
2019-10-28 03:29:28 UTC 5787019 1487580005595612013                     4.11    556453136       e1971dce-73f1-484b-a105-
10fad2538d78    cart
2019-10-28 03:29:29 UTC 47610   1487580007717929935                     0.79    178186931       1ab7b37f-9d6e-4bc7-8df4-
287eaf05153f    cart
2019-10-26 22:25:01 UTC 5896615 1487580005427839846             irisk   2.48    468176825       14b007c6-468f-477d-82e0-
56359e34c1fd    cart
2019-10-29 13:30:53 UTC 5817692 1487580010872045658                     0.79    72529230        02a293d0-75c4-428e-a05b-
8239bd9121d3    cart
Time taken: 0.19 seconds, Fetched: 5 row(s)
hive>
```

- **Query Analysis**

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

Code:

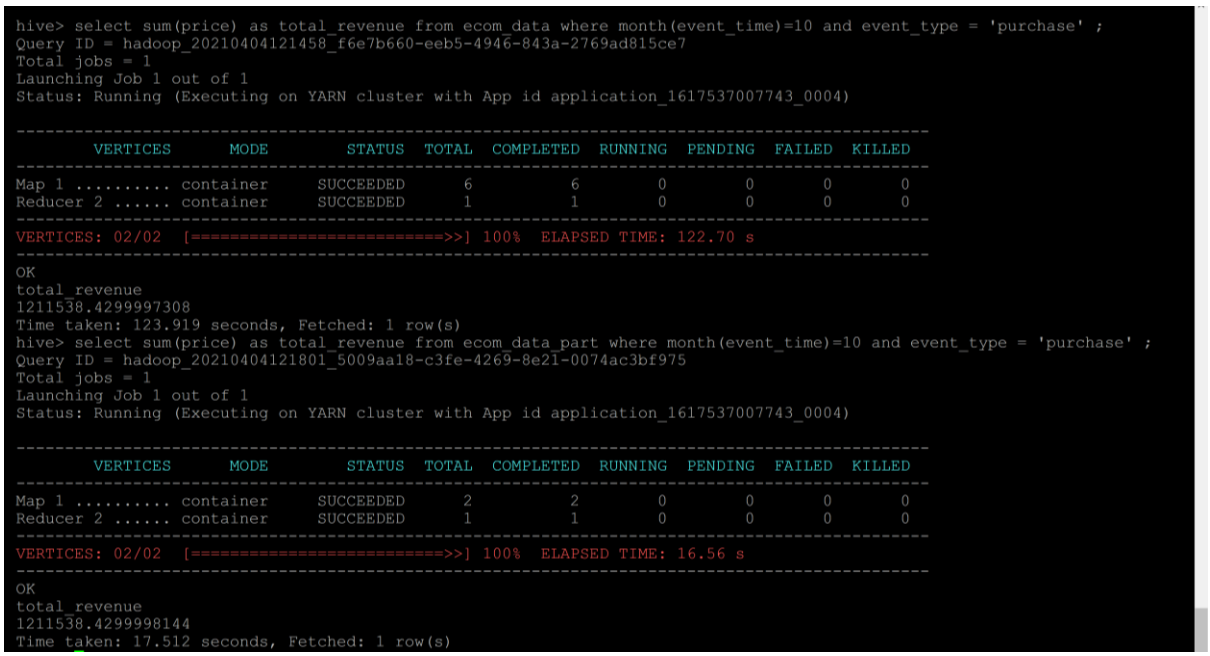
select sum(price) as total_revenue from ecom_data where month(event_time)=10 and event_type = 'purchase' ;



Note: The below screenshot of the same query from both the base table and the bucketed table. When compared the bucketed table takes less time to query the result than the base table. This is the use of partitioning and bucketing the data.

Code:

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



Insights:

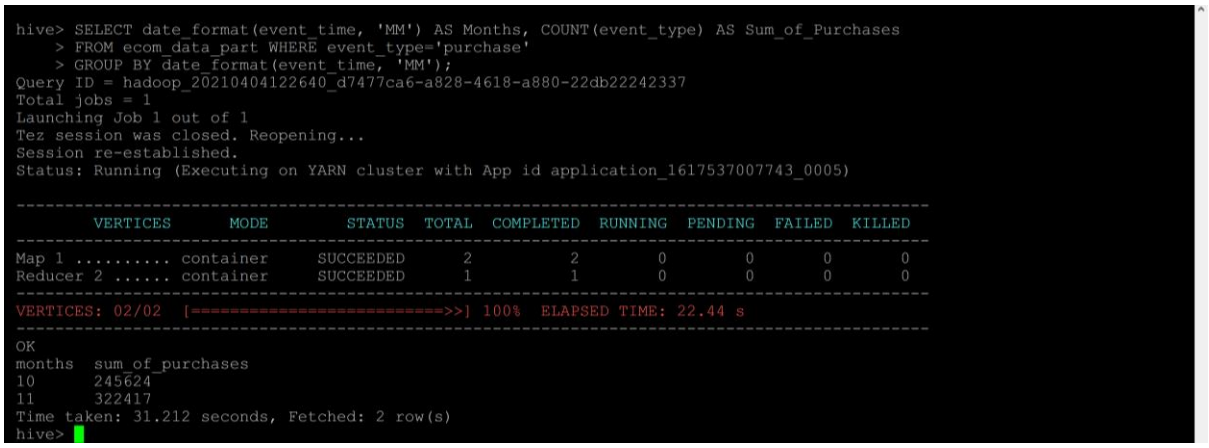
- The Total revenue generated based on Purchase in the month of October 2019 was **1,211,538.43/-**. We tried getting the result using both base table[ecom_data] and optimized table[ecom_data_part]. We figured out that Bucketed table reduces the query time when compared to the base table by approx. 107 seconds.

Hence moving forward, we will use the commands using optimized table for less query time.

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

Code:

```
SELECT date_format(event_time, 'MM') AS Months, COUNT(event_type) AS Sum_of_Purchases
FROM ecom_data_part WHERE event_type='purchase'
GROUP BY date_format(event_time, 'MM');
```



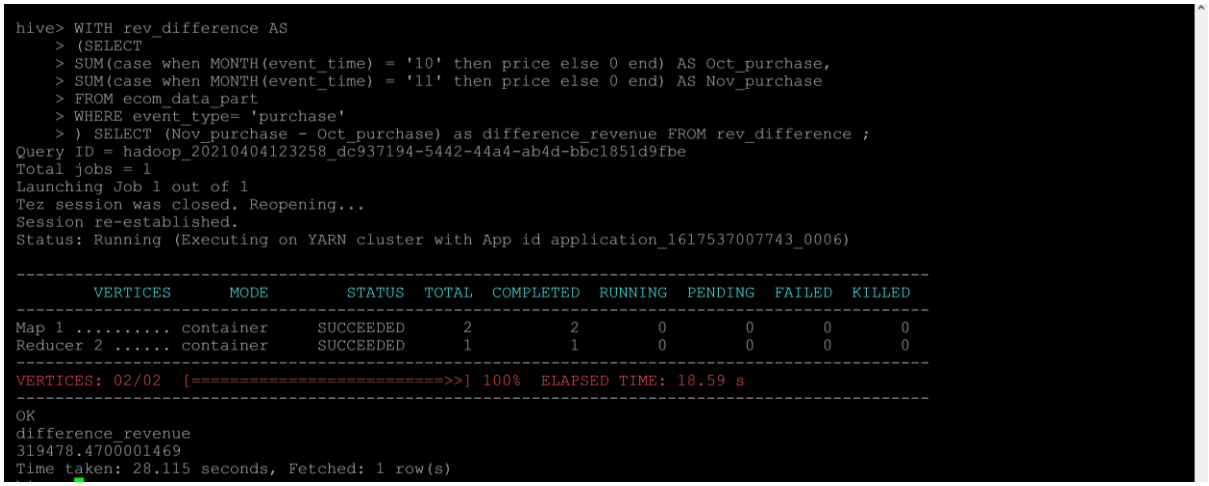
Insights:

- It seems to be that there was more purchase made in the month of November (11) i.e., 322,417 than in the month of October (10) i.e., 245,624.
- Looking at these figures we could assume that the month of November must be more profitable than the month of October.

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

Code:

```
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 ecom_data_part
WHERE event_type= 'purchase')
SELECT (Nov_purchase - Oct_purchase) as difference_revenue FROM rev_difference ;
```



```
hive> 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 ecom_data_part
    > WHERE event_type= 'purchase'
    > ) SELECT (Nov_purchase - Oct_purchase) as difference_revenue FROM rev_difference ;
Query ID = hadoop_20210404123258_dc937194-5442-44a4-ab4d-bbc1851d9fbe
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_1617537007743_0006)

--------------------------------------------------------------------------------
        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: 18.59 s
--------------------------------------------------------------------------------
OK
difference_revenue
319478.4700001469
Time taken: 28.115 seconds, Fetched: 1 row(s)
```

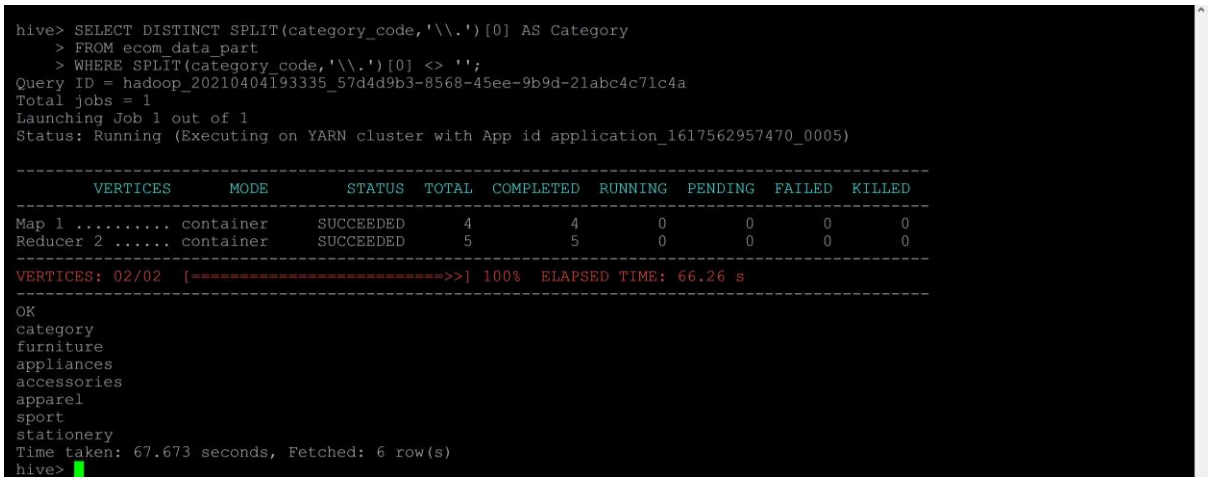
Insights:

- It is seen that the difference in revenue between October and November month is 319478.4700.
- As per the result, we could conclude that the revenue generated in November of 2019 was more than the revenue generated in the month of October. In other words, November was more profitable and had a better sale.

4. Find the distinct categories of products. Categories with null value can be ignored.

Code:

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



Insights:

- There are total 6 different categories under which company sells their different products. i.e Furniture, appliances, accessories, apparel, sport and stationery.

5. Find the total number of products available in each category.
Code:

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

```
hive> SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS No_of_products
    > FROM ecom_data_part
    > WHERE SPLIT(category_code,'\\.')[0] <> ''
    > GROUP BY SPLIT(category_code,'\\.')[0]
    > ORDER BY No_of_products DESC;
Query ID = hadoop_20210404124606_137fdfbb-5d99-47ec-b964-62c5f5170308
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617537007743_0006)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        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: 71.98 s
----------------------------------------------------------------------------------------------
OK
category        no_of_products
appliances      61736
stationery      26722
furniture       23604
apparel 18232
accessories     12929
sport   2
```

Insights:

- Appliance with i.e., 61,736 products is the leading category, followed by stationery as second furniture as third.
- Sports category has only 2 products registered. This must be due to low cosmetic products in the sports market.

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

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

```
hive> SELECT brand,sum(price) as total_price from ecom_data_part where brand !=''
    > and event_type ='purchase'
    > group by brand order by total_price desc limit 1;
Query ID = hadoop_20210404125208_739cc8ec-3af2-4162-bab1-1319aabf45b7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617537007743_0006)

--------------------------------------------------------------------------------
        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
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 15.83 s
--------------------------------------------------------------------------------
OK
brand   total_price
runail  148297.94000000478
Time taken: 16.523 seconds, Fetched: 1 row(s)
hive>
```

Insights:

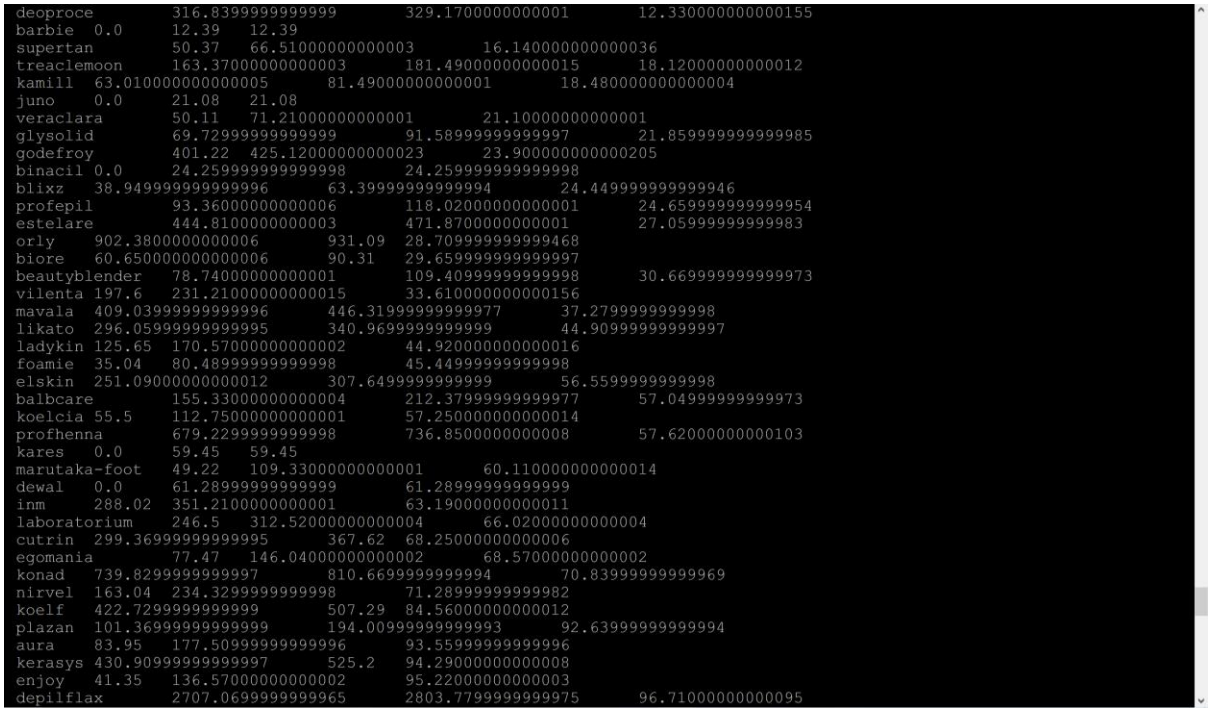
- Runail is the brand that has highest / maximum sales in the month of October and November of 2019 combined.
- Reason being cosmetic lovers are loyal towards Runail brand and hence increasing the products related to Runail could further increase the profit.

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

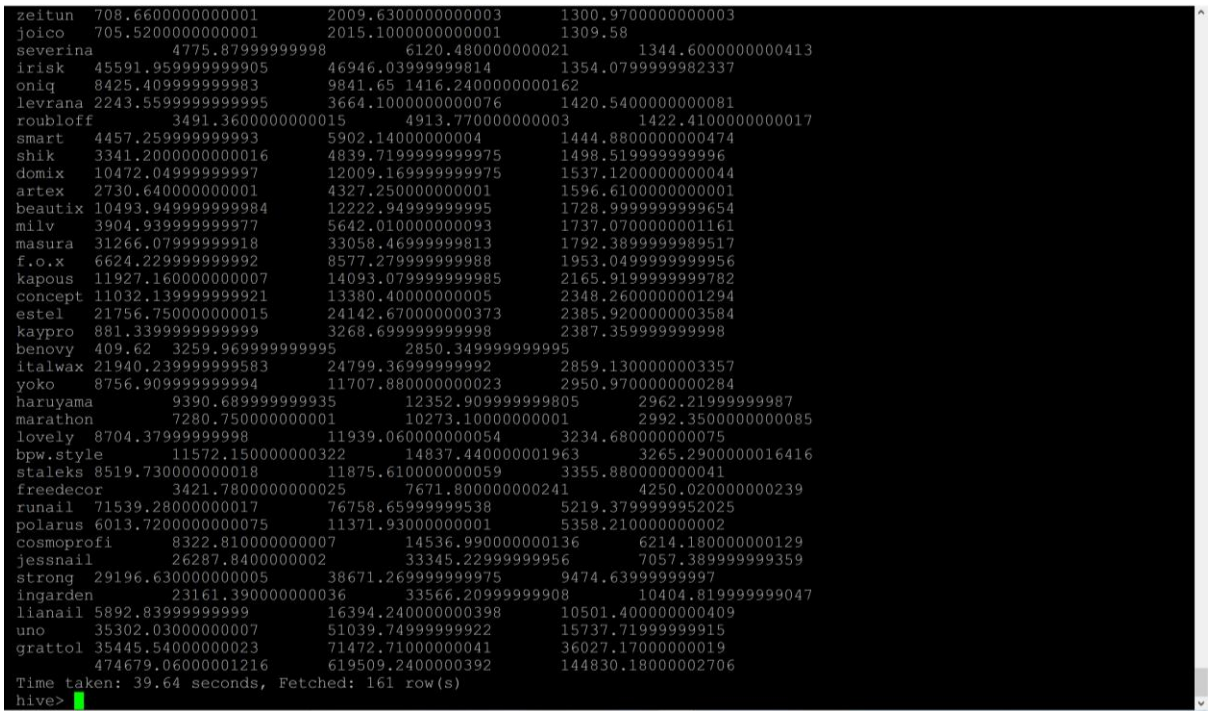
```
WITH Monthly_Revenue AS (
SELECT brand,
SUM(CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS Oct_Revenue,
SUM(CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS Nov_Revenue
FROM ecom_data_part
WHERE event_type='purchase' AND date_format(event_time, 'MM') IN ('10', '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;
```

```
hive> 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 ecom_data_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;
Query ID = hadoop_20210404133419_16157ed6-e57c-4680-bdd4-ed08fedb4e6e
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_1617537007743_0008)

--------------------------------------------------------------------------------
        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
Reducer 3 ..... container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 33.01 s
--------------------------------------------------------------------------------
OK
brand   oct_revenue     nov_revenue     sales_difference
ovale   2.54    3.1     0.56
cosima  20.229999999999997      20.93   0.7000000000000028
grace   100.91999999999996      102.61000000000003      1.6900000000000688
helloganic      0.0     3.1     3.1
skinity 8.88    12.440000000000001      3.5600000000000005
bodyton 1376.3400000000004      1380.6399999999976      4.299999999997226
moyou   5.71    10.280000000000001      4.570000000000001
neoleor 43.41   51.7    8.290000000000006
soleo   204.19999999999953      212.53000000000034      8.330000000000808
jaguar  1102.1100000000001      1110.6500000000005      8.540000000000418
tertio  236.16000000000003      245.7999999999999       9.639999999999873
```



```
deoproce        316.8399999999999       329.1700000000001       12.330000000000155
barbie  0.0     12.39   12.39
supertan        50.37   66.51000000000003       16.140000000000036
treaclemoon     163.37000000000003      181.49000000000015      18.12000000000012
kamill  63.010000000000005      81.49000000000001       18.480000000000004
juno    0.0     21.08   21.08
veraclara       50.11   71.21000000000001       21.10000000000001
glysolid        69.72999999999999       91.58999999999997       21.859999999999985
godefroy        401.22  425.12000000000023      23.900000000000205
binacil 0.0     24.259999999999998      24.259999999999998
blixz   38.949999999999996      63.39999999999994       24.449999999999946
profepil        93.36000000000006       118.02000000000001      24.659999999999954
estelare        444.8100000000003       471.8700000000001       27.05999999999983
orly    902.3800000000006       931.09  28.709999999999468
biore   60.650000000000006      90.31   29.659999999999997
beautyblender   78.74000000000001       109.40999999999998      30.669999999999973
vilenta 197.6   231.21000000000015      33.610000000000156
mavala  409.03999999999996      446.31999999999977      37.2799999999998
likato  296.05999999999995      340.9699999999999       44.90999999999997
ladykin 125.65  170.57000000000002      44.920000000000016
foamie  35.04   80.48999999999998       45.44999999999998
elskin  251.09000000000012      307.6499999999999       56.5599999999998
balbcare        155.33000000000004      212.37999999999977      57.04999999999973
koelcia 55.5    112.75000000000001      57.250000000000014
profhenna       679.2299999999998       736.8500000000008       57.62000000000103
kares   0.0     59.45   59.45
marutaka-foot   49.22   109.33000000000001      60.110000000000014
dewal   0.0     61.28999999999999       61.28999999999999
inm     288.02  351.2100000000001       63.19000000000011
laboratorium    246.5   312.52000000000004      66.02000000000004
cutrin  299.36999999999995      367.62  68.25000000000006
egomania        77.47   146.04000000000002      68.57000000000002
konad   739.8299999999997       810.6699999999994       70.83999999999969
nirvel  163.04  234.3299999999998       71.28999999999982
koelf   422.7299999999999       507.29  84.56000000000012
plazan  101.36999999999999      194.00999999999993      92.63999999999994
aura    83.95   177.50999999999996      93.55999999999996
kerasys 430.90999999999997      525.2   94.29000000000008
enjoy   41.35   136.57000000000002      95.22000000000003
depilflax       2707.0699999999965      2803.7799999999975      96.71000000000095
```



```
zeitun  708.6600000000001       2009.6300000000003      1300.9700000000003
joico   705.5200000000001       2015.1000000000001      1309.58
severina        4775.87999999998        6120.480000000021       1344.6000000000413
irisk   45591.959999999905      46946.03999999814       1354.0799999982337
oniq    8425.409999999983       9841.65 1416.2400000000162
levrana 2243.5599999999995      3664.1000000000076      1420.5400000000081
roubloff        3491.3600000000015      4913.770000000003       1422.4100000000017
smart   4457.259999999993       5902.14000000004        1444.8800000000474
shik    3341.2000000000016      4839.7199999999975      1498.519999999996
domix   10472.04999999997       12009.169999999975      1537.1200000000044
artex   2730.640000000001       4327.250000000001       1596.6100000000001
beautix 10493.949999999984      12222.94999999995       1728.9999999999654
milv    3904.939999999977       5642.010000000093       1737.0700000001161
masura  31266.07999999918       33058.46999999813       1792.3899999989517
f.o.x   6624.229999999992       8577.279999999988       1953.0499999999956
kapous  11927.160000000007      14093.079999999985      2165.9199999999782
concept 11032.139999999921      13380.40000000005       2348.2600000001294
estel   21756.750000000015      24142.670000000373      2385.9200000003584
kaypro  881.3399999999999       3268.699999999998       2387.359999999998
benovy  409.62  3259.969999999995        2850.349999999995
italwax 21940.239999999583      24799.36999999992       2859.1300000003357
yoko    8756.909999999994       11707.880000000023      2950.9700000000284
haruyama        9390.689999999935       12352.909999999805      2962.21999999987
marathon        7280.750000000001       10273.10000000001       2992.3500000000085
lovely  8704.37999999998        11939.060000000054      3234.680000000075
bpw.style       11572.150000000322      14837.440000001963      3265.2900000016416
staleks 8519.730000000018       11875.610000000059      3355.880000000041
freedecor       3421.7800000000025      7671.800000000241       4250.020000000239
runail  71539.28000000017       76758.65999999538       5219.3799999952025
polarus 6013.7200000000075      11371.93000000001       5358.210000000002
cosmoprofi      8322.810000000007       14536.990000000136      6214.180000000129
jessnail        26287.8400000002        33345.22999999956       7057.389999999359
strong  29196.630000000005      38671.269999999975      9474.63999999997
ingarden        23161.390000000036      33566.20999999908       10404.819999999047
lianail 5892.83999999999        16394.240000000398      10501.400000000409
uno     35302.03000000007       51039.74999999922       15737.71999999915
grattol 35445.54000000023       71472.71000000041       36027.17000000019
        474679.06000001216      619509.2400000392       144830.18000002706
Time taken: 39.64 seconds, Fetched: 161 row(s)
hive>
```

Insights:

- Total of 161 brands had an increase in the selling from October to November.
- ‘Grattol’ brand has the highest total increment i.e., 36,027 /- and ‘Ovale’ seems to have least increment of 0.56 /- from October to November.
- Among all these brands list, ‘Runail’ which was the best brand in terms of selling in October 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 /-.
- This implies that ‘Runail’ is the best and popular brand among all other brands within people.

8.Your Company wants to reward the top 10 users in its website with a Golden Customer plan. Write a query to generate a list of top 10 users who spent the most.
Code:

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

```
hive> SELECT user_id, SUM(price) as Total_Expense
    > FROM ecom_data_part
    > WHERE event_type='purchase'
    > GROUP BY user_id
    > ORDER BY Total_Expense DESC
    > LIMIT 10;
Query ID = hadoop_20210404131249_41788d71-42b8-49f2-b6a1-8fa5c3b43544
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1617537007743_0007)

--------------------------------------------------------------------------------
        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
Reducer 3 ......   container     SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 17.63 s
--------------------------------------------------------------------------------
OK
user_id total_expense
557790271       2715.8699999999963
150318419       1645.9699999999998
562167663       1352.8500000000004
531900924       1329.45
557850743       1295.4799999999998
522130011       1185.3900000000012
561592095       1109.700000000001
431950134       1097.5899999999997
566576008       1056.3600000000001
521347209       1040.9099999999999
Time taken: 18.433 seconds, Fetched: 10 row(s)
hive>
```

Insights:

- Here is the list of the top 10 users or buyers who have spend the most and could be rewarded with a Golden Customer plan to attract more people in the coming future.

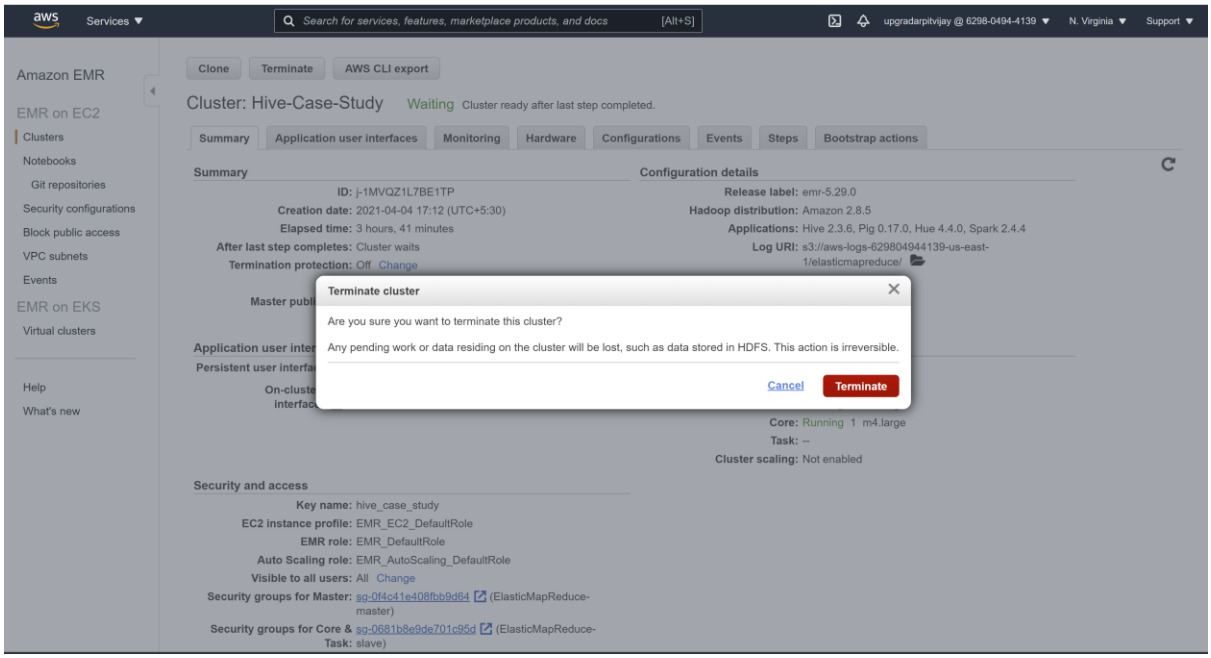
- **Cleaning Up**

1. Drop the database ECOMDB

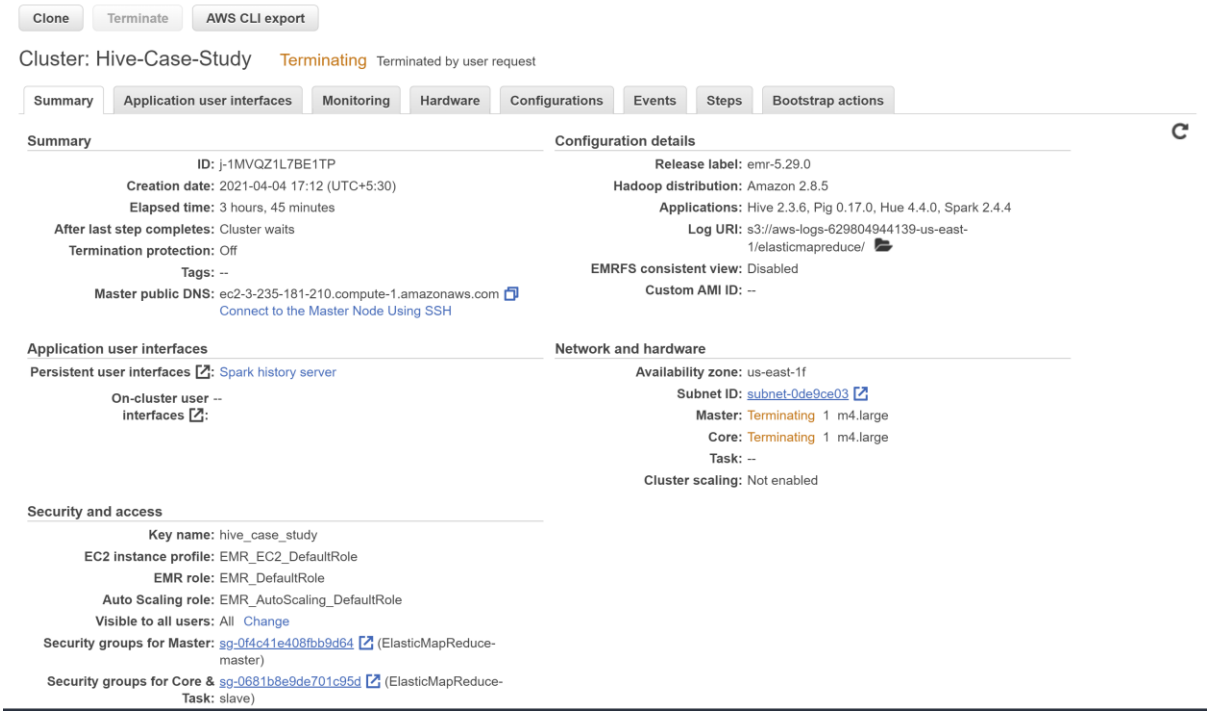
```
hive> DROP DATABASE IF EXISTS ECOMDB;
OK
Time taken: 0.218 seconds
hive>
```

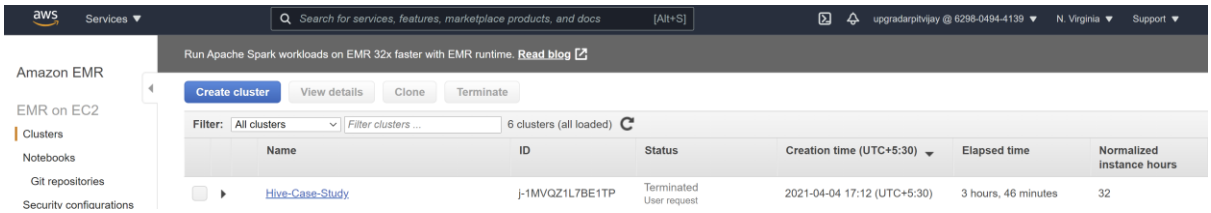
2. Once the analysis is done, we can terminate the cluster by changing the Termination protection from **ON** to **OFF** and then click on the terminate button.

3. Click on Terminate.



4. Once we click on the terminate button the status of the cluster changes to terminating and then the cluster is terminated.





SUMMARY

For this case study we have worked with public clickstream dataset of a cosmetics store.

Steps taken are:

1.Copying the data set into the HDFS from S3 Bucket.
2.Creating the database and launching Hive queries on your EMR cluster.
[we made necessary optimizations, such as selecting the appropriate table format and using partitioned/bucketed tables]
[use CSVSerde with the default properties value for loading the dataset into a Hive table]
3.Cleaning up.

Overall, the increase in sales happened from October to November.