Hive Case Study :- E-Commerce Sales

Contents

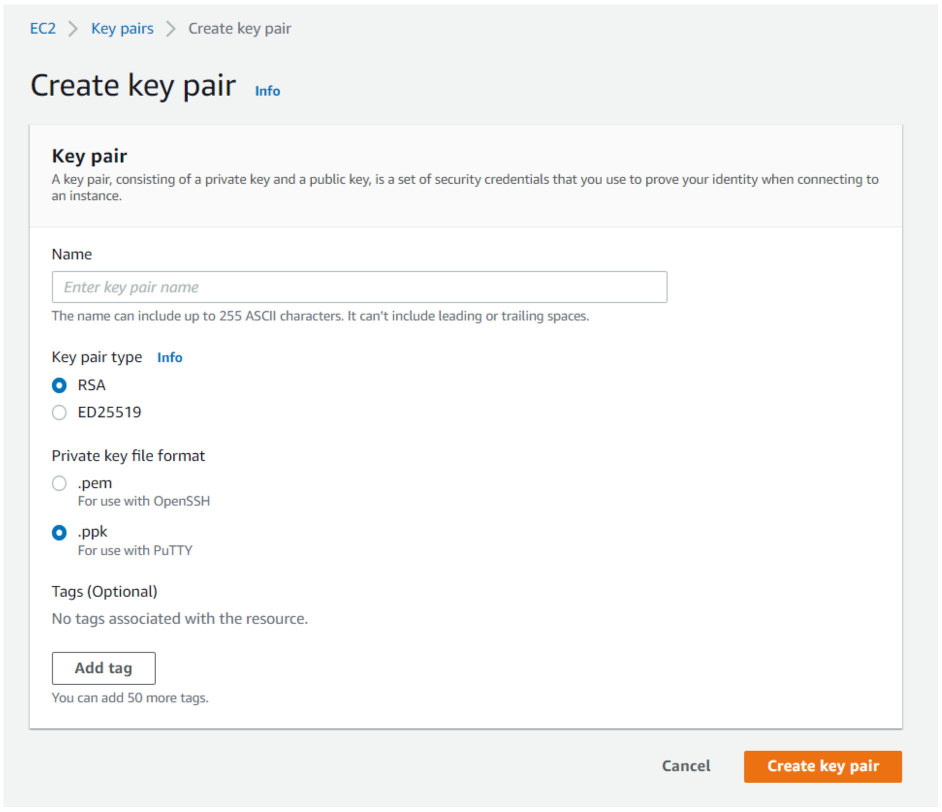
1. Problem Statement:

A tech companies is exploring ways to improve their sales. They want to start by analysing customer behaviour and gaining insights about product trends. Here, the role of big data analysts is among the most sought-after to gain the insights from abundance of data. Objective Through this assignment, as a big data analyst, we will extract data and gather insights from a real-life data set of an e-commerce company. We will analyse and gain insights about the clickstream data from a website so that we can extract insights about the customers behaviour.

2. Steps Involved:

2.1. Create Key Pair in AWS :

Open EC2 Management Console and select Create Key Pair. Enter a name for Key Pair, keeping the type as "RSA" and file format as ". ppk" and Select Create Key Pair. The Key Pair is downloaded.



EC2 > Key pairs > Create key pair

Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name

Enter key pair name

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info

[✓ RSA] ED25519

Private key file format

.pem For use with OpenSSH

[✓ .ppk For use with PuTTY]

Tags (Optional)

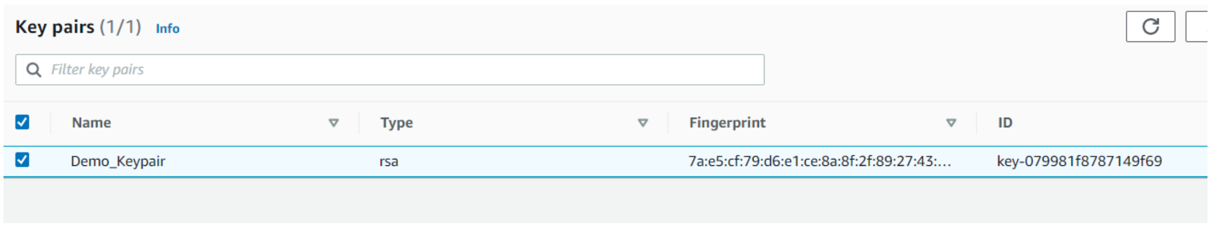
No tags associated with the resource.

Add tag

You can add 50 more tags.

Cancel | Create key pair

I have created a key pair with name Demo Keypair :

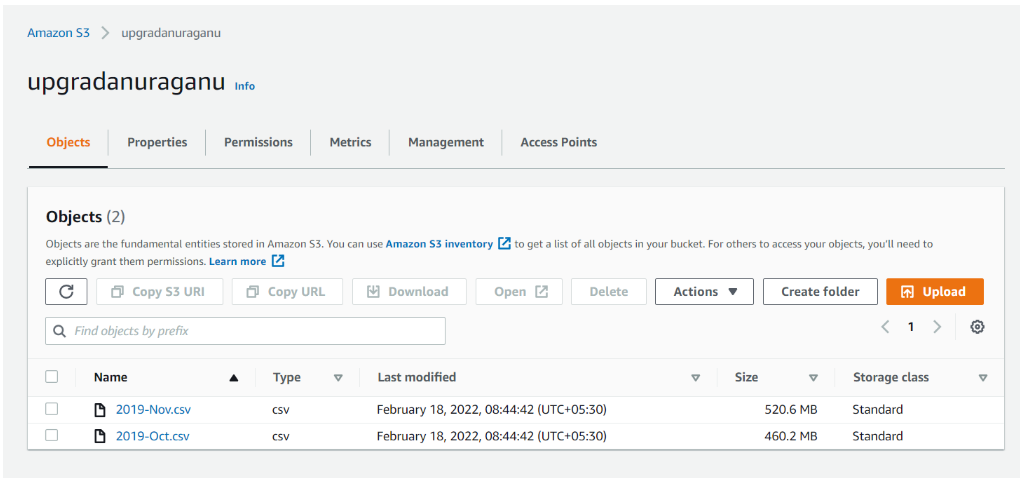

Key pairs (1/1) Info

Name	Type	Fingerprint	ID
Demo_Keypair	rsa	7a:e5:cf:79:d6:e1:ce:8a:8f:2f:89:27:43:…	key-079981f8787149f69

2.2. Create S3 Bucket and upload the Data

Go to Amazon S3 > Select Create Bucket > Enter Bucket Name “upgradanuraganu” > Unselect “Block all public access” > Select Create Bucket.

Once the bucket is created then we will upload the files which is provided for the case study. We will click on our bucket and then click on upload to upload the files.



Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
2019-Nov.csv	csv	February 18, 2022, 08:44:42 (UTC+05:30)	520.6 MB	Standard
2019-Oct.csv	csv	February 18, 2022, 08:44:42 (UTC+05:30)	460.2 MB	Standard

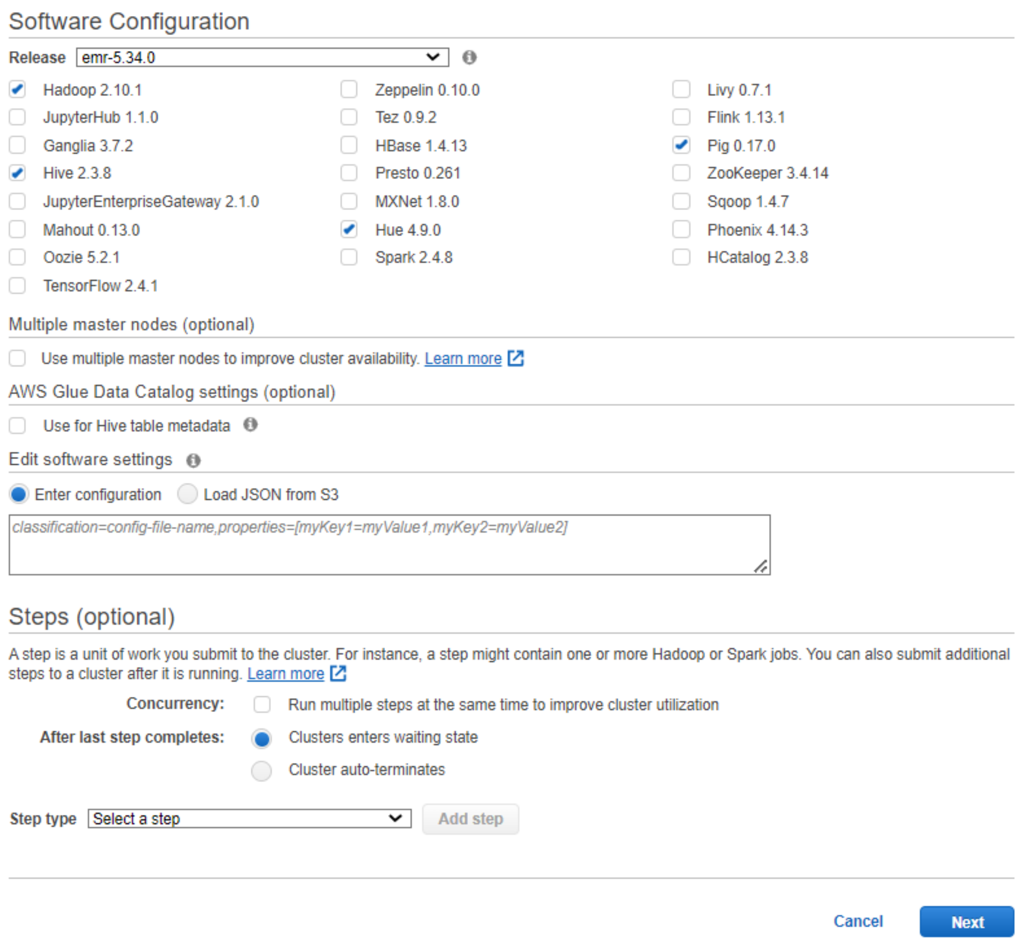
Once the file is upload, we will select one object and copy the S3 URL. Same with the same object. Below is the object URL cleated in S3.

s3://upgradanuraganu/2019-Nov.csv

s3://upgradanuraganu/2019-Oct.csv

2.3. We will now create the EMR cluster :

Create EMR Cluster • Go to EMR > Select Create Cluster. Selected the General Configuration, Software Configuration, and Hardware Configuration as shown in below images. Click "Create Cluster



Software Configuration

Release: emr-5.34.0

- [✓] Hadoop 2.10.1
- [ ] JupyterHub 1.1.0
- [ ] Ganglia 3.7.2
- [✓] Hive 2.3.8
- [ ] JupyterEnterpriseGateway 2.1.0
- [ ] Mahout 0.13.0
- [ ] Oozie 5.2.1
- [ ] TensorFlow 2.4.1
- [ ] Zeppelin 0.10.0
- [ ] Tez 0.9.2
- [ ] HBase 1.4.13
- [ ] Presto 0.261
- [ ] MXNet 1.8.0
- [✓] Hue 4.9.0
- [ ] Spark 2.4.8
- [ ] Livy 0.7.1
- [ ] Flink 1.13.1
- [✓] Pig 0.17.0
- [ ] ZooKeeper 3.4.14
- [ ] Sqoop 1.4.7
- [ ] Phoenix 4.14.3
- [ ] HCatalog 2.3.8

Multiple master nodes (optional)

- [ ] Use multiple master nodes to improve cluster availability. Learn more

AWS Glue Data Catalog settings (optional)

- [ ] Use for Hive table metadata

Edit software settings

[✓ Enter configuration] Load JSON from S3

classification=config-file-name,properties=[myKey1=myValue1,myKey2=myValue2]

Steps (optional)

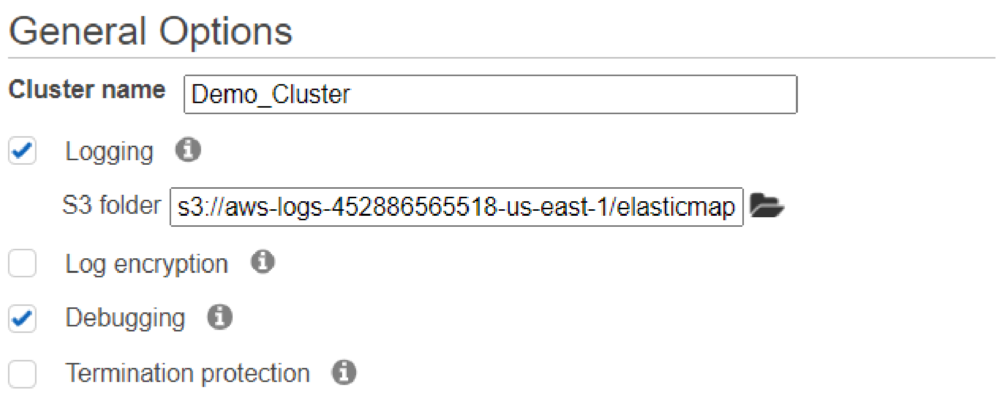
A step is a unit of work you submit to the cluster. For instance, a step might contain one or more Hadoop or Spark jobs. You can also submit additional steps to a cluster after it is running. Learn more

Concurrency: [ ] Run multiple steps at the same time to improve cluster utilization

After last step completes: [✓ Clusters enters waiting state] Cluster auto-terminates

Step type: Select a step

I have given the Cluster name as Demo_cluster



General Options

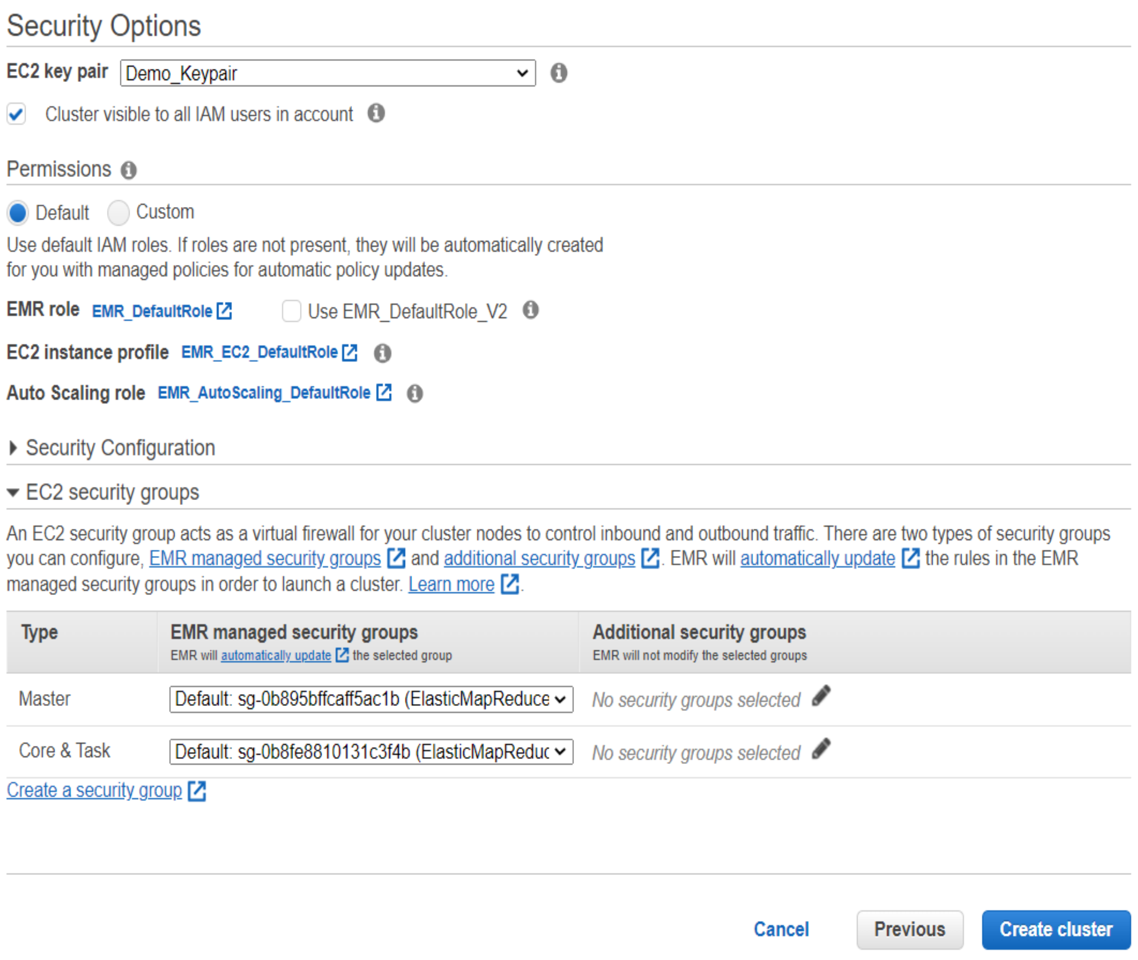
Cluster name: Demo_Cluster

- [✓] Logging

S3 folder: s3://aws-logs-452886565518-us-east-1/elasticmap

- [ ] Log encryption
- [✓] Debugging
- [ ] Termination protection

Now, in next step we will choose the EC2 key pair which we have created in first step, my key pair is Demo Keypair. Then we will click on Create Cluster.



Security Options

EC2 key pair: Demo_Keypair

[✓] Cluster visible to all IAM users in account

Permissions

[✓ Default] Custom

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

EMR role: EMR_DefaultRole   [ ] Use EMR_DefaultRole_V2

EC2 instance profile: EMR_EC2_DefaultRole

Auto Scaling role: EMR_AutoScaling_DefaultRole

Security Configuration

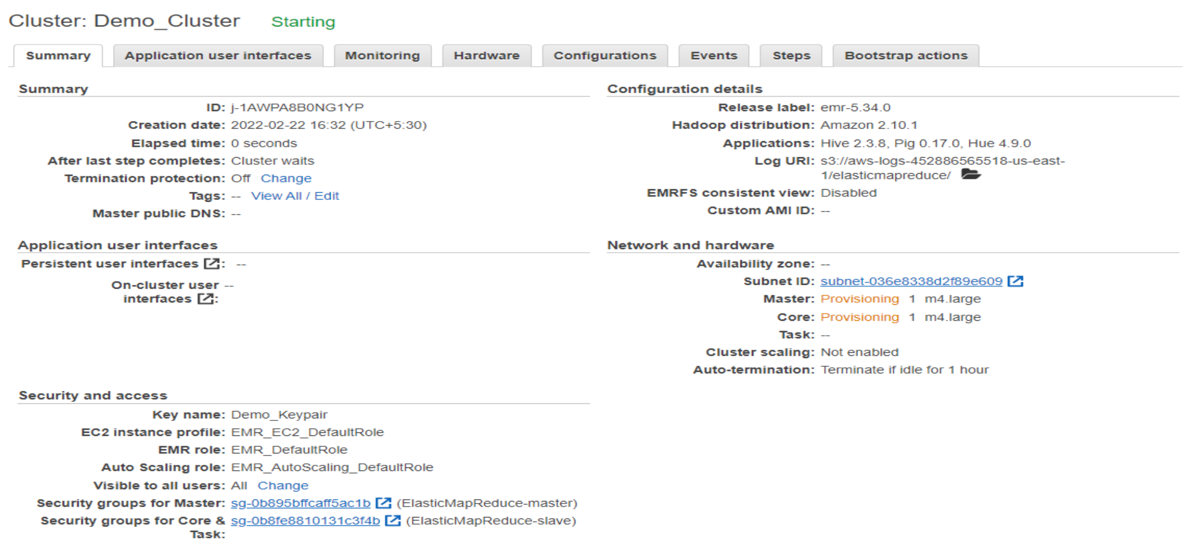
EC2 security groups

An EC2 security group acts as a virtual firewall for your cluster nodes to control inbound and outbound traffic. There are two types of security groups you can configure, EMR managed security groups and additional security groups. EMR will automatically update the rules in the EMR managed security groups in order to launch a cluster. Learn more.

Type	EMR managed security groups EMR will automatically update the selected group	Additional security groups EMR will not modify the selected groups
Master	Default: sg-0b895bffcaff5ac1b (ElasticMapReduce	*No security groups selected*
Core & Task	Default: sg-0b8fe8810131c3f4b (ElasticMapReduc	*No security groups selected*

Create a security group

- Now, Cluster is in Starting phase which means then the cluster is going to start it is looking for the free system for provision.



Cluster: Demo_Cluster   Starting

Summary

ID:	j-1AWPA8B0NG1YP
Creation date:	2022-02-22 16:32 (UTC+5:30)
Elapsed time:	0 seconds
After last step completes:	Cluster waits
Termination protection:	Off Change
Tags:	-- View All / Edit
Master public DNS:	--

Application user interfaces

Persistent user interfaces:	--
On-cluster user interfaces:	--

Security and access

Key name:	Demo_Keypair
EC2 instance profile:	EMR_EC2_DefaultRole
EMR role:	EMR_DefaultRole
Auto Scaling role:	EMR_AutoScaling_DefaultRole
Visible to all users:	All Change
Security groups for Master:	sg-0b895bffcaff5ac1b (ElasticMapReduce-master)
Security groups for Core & Task:	sg-0b8fe8810131c3f4b (ElasticMapReduce-slave)

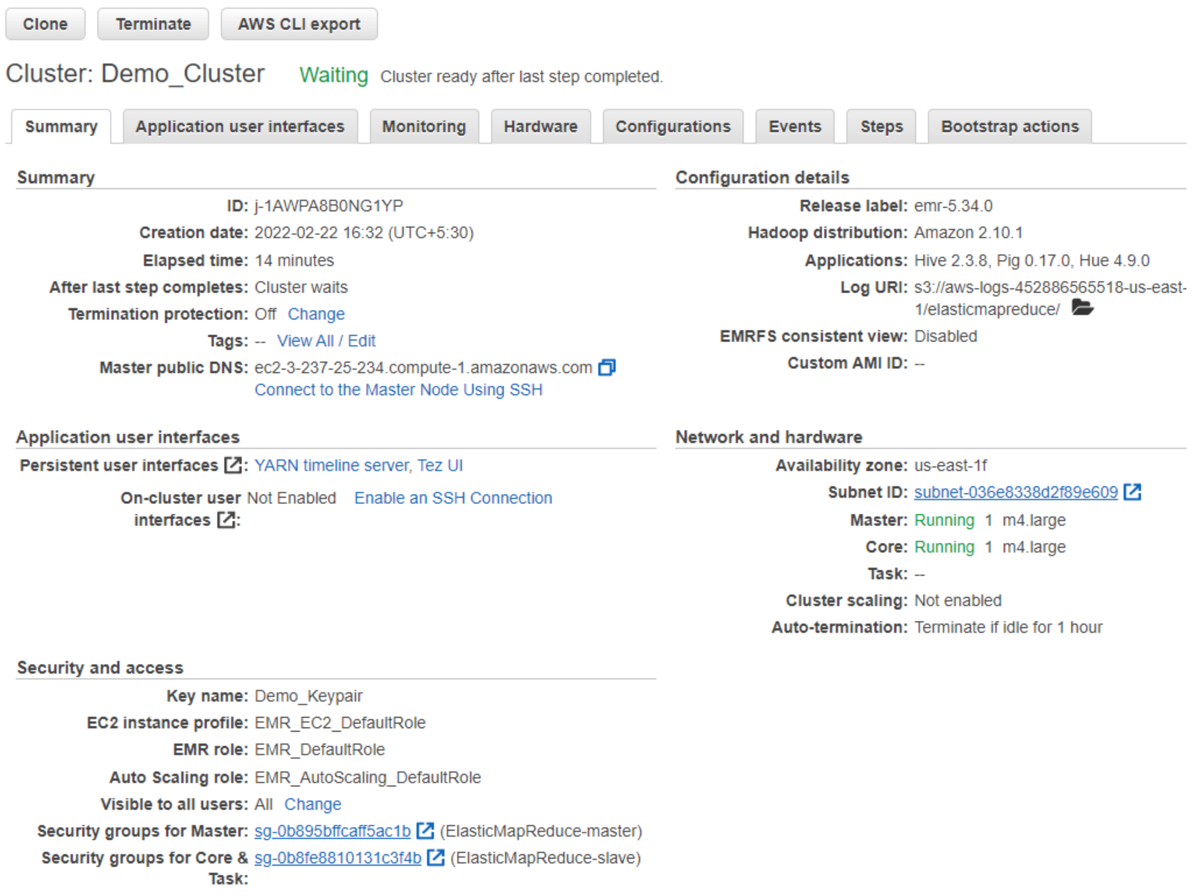
Configuration details

Release label:	emr-5.34.0
Hadoop distribution:	Amazon 2.10.1
Applications:	Hive 2.3.8, Pig 0.17.0, Hue 4.9.0
Log URI:	s3://aws-logs-452886565518-us-east-1/elasticmapreduce/
EMRFS consistent view:	Disabled
Custom AMI ID:	--

Network and hardware

Availability zone:	--
Subnet ID:	subnet-036e8338d2f89e609
Master:	Provisioning 1 m4.large
Core:	Provisioning 1 m4.large
Task:	--
Cluster scaling:	Not enabled
Auto-termination:	Terminate if idle for 1 hour

- Now the cluster is in witing state which means it is running now we can perform the operation.



Cluster: Demo_Cluster Waiting Cluster ready after last step completed.

Summary

ID:	j-1AWPA8B0NG1YP
Creation date:	2022-02-22 16:32 (UTC+5:30)
Elapsed time:	14 minutes
After last step completes:	Cluster waits
Termination protection:	Off Change
Tags:	-- View All / Edit
Master public DNS:	ec2-3-237-25-234.compute-1.amazonaws.com Connect to the Master Node Using SSH

Configuration details

Release label:	emr-5.34.0
Hadoop distribution:	Amazon 2.10.1
Applications:	Hive 2.3.8, Pig 0.17.0, Hue 4.9.0
Log URI:	s3://aws-logs-452886565518-us-east-1/elasticmapreduce/
EMRFS consistent view:	Disabled
Custom AMI ID:	--

Application user interfaces

Persistent user interfaces:	YARN timeline server, Tez UI
On-cluster user interfaces:	Not Enabled Enable an SSH Connection

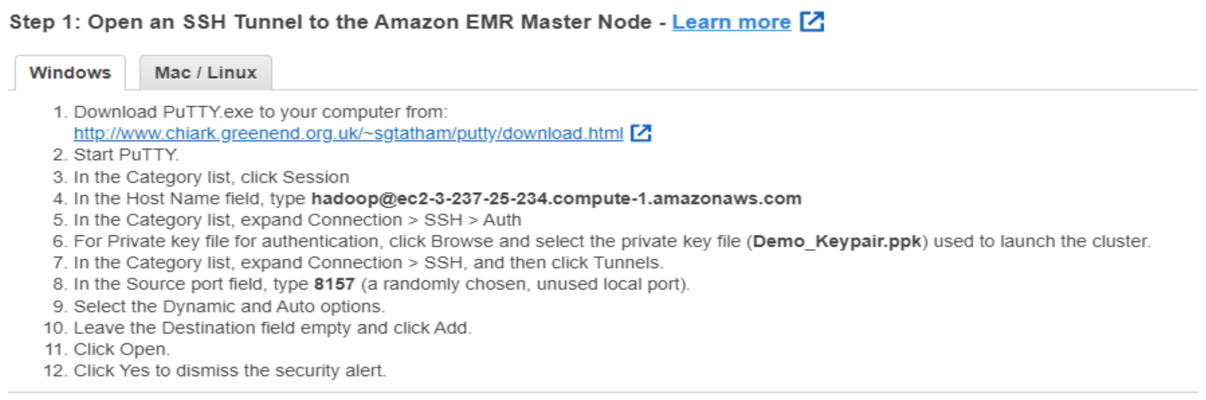
Network and hardware

Availability zone:	us-east-1f
Subnet ID:	subnet-036e8338d2f89e609
Master:	Running 1 m4.large
Core:	Running 1 m4.large
Task:	--
Cluster scaling:	Not enabled
Auto-termination:	Terminate if idle for 1 hour

Security and access

Key name:	Demo_Keypair
EC2 instance profile:	EMR_EC2_DefaultRole
EMR role:	EMR_DefaultRole
Auto Scaling role:	EMR_AutoScaling_DefaultRole
Visible to all users:	All Change
Security groups for Master:	sg-0b895bffcaff5ac1b (ElasticMapReduce-master)
Security groups for Core & Task:	sg-0b8fe8810131c3f4b (ElasticMapReduce-slave)

- We will open PuTTY to download PuTTY the steps to download is already provided, enable the SSH connection to open the CLI.



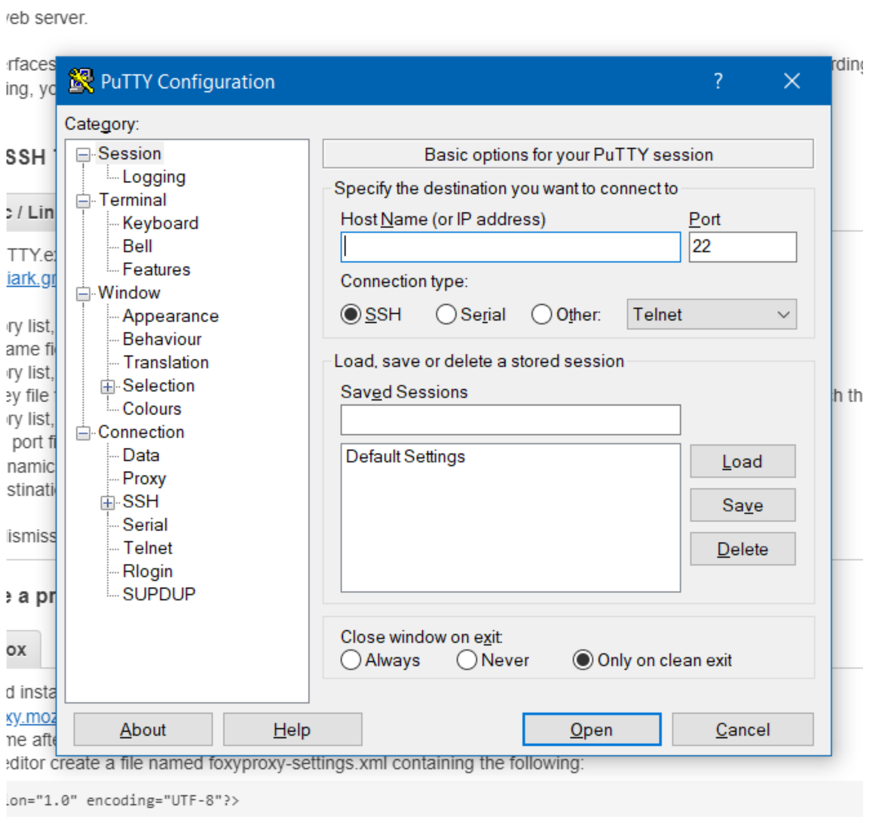
Step 1: Open an SSH Tunnel to the Amazon EMR Master Node - Learn more

Windows | Mac / Linux

1. Download PuTTY.exe to your computer from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Start PuTTY.
3. In the Category list, click Session
4. In the Host Name field, type **hadoop@ec2-3-237-25-234.compute-1.amazonaws.com**
5. In the Category list, expand Connection > SSH > Auth
6. For Private key file for authentication, click Browse and select the private key file (**Demo_Keypair.ppk**) used to launch the cluster.
7. In the Category list, expand Connection > SSH, and then click Tunnels.
8. In the Source port field, type **8157** (a randomly chosen, unused local port).
9. Select the Dynamic and Auto options.
10. Leave the Destination field empty and click Add.
11. Click Open.
12. Click Yes to dismiss the security alert.

- Now, we will copy the HOST name hadoop@ec2-3-237-25-234.compute-1.amazonaws.com **and paste it in PuTTY configuration in session tab.**

Once cluster in running state we have to click on Master public DNS. 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. Now we need to open PuTTY and connect to the master node by selecting the .ppk file.



PuTTY Configuration

Category:

- Session
  - Logging
- Terminal
  - Keyboard
  - Bell
  - Features
- Window
  - Appearance
  - Behaviour
  - Translation
  - Selection
  - Colours
- Connection
  - Data
  - Proxy
  - SSH
  - Serial
  - Telnet
  - Rlogin
  - SUPDUP

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: [✓ SSH] Serial Other: Telnet

Load, save or delete a stored session

Saved Sessions

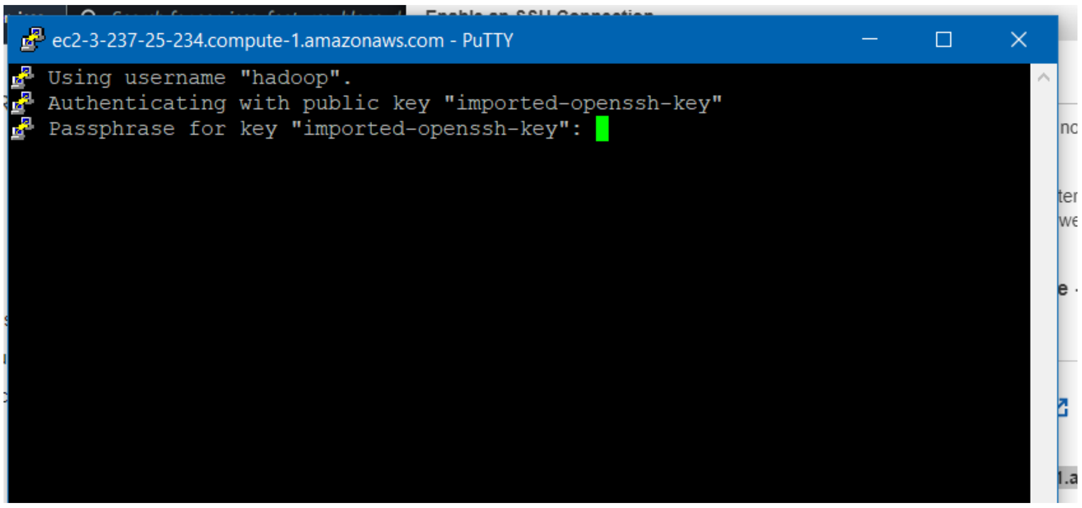
Default Settings

Load | Save | Delete

Close window on exit: Always Never [✓ Only on clean exit]

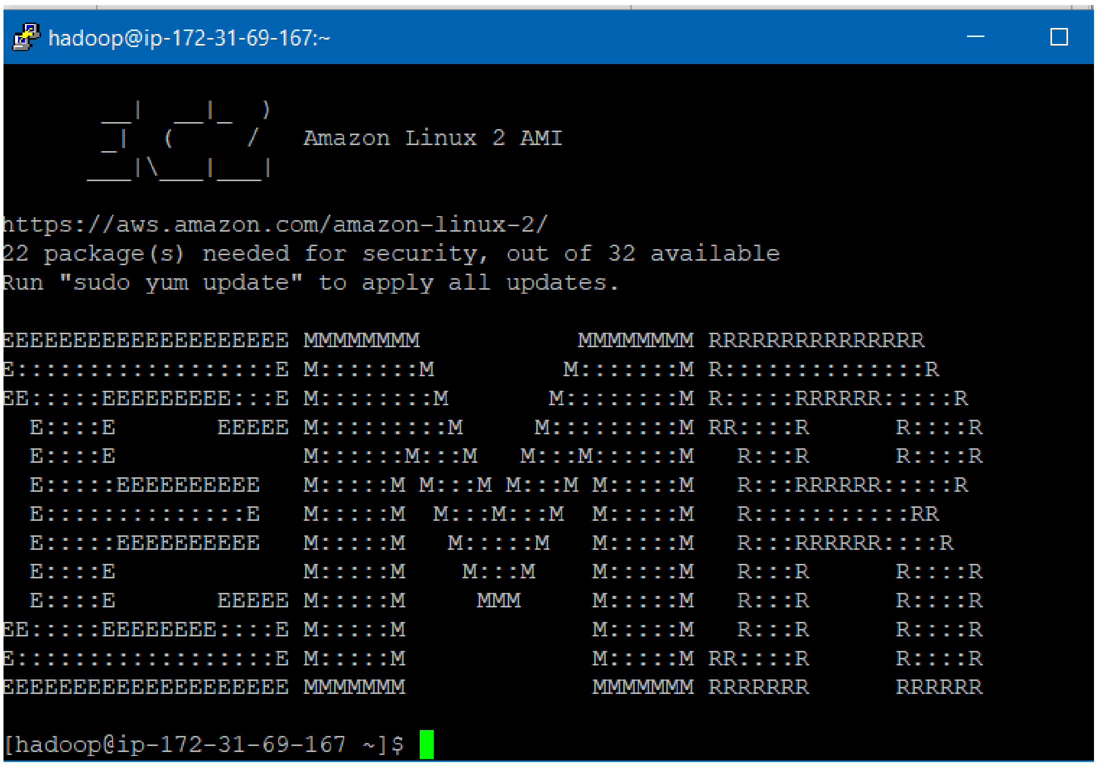
About | Help | Open | Cancel

- Below is the security code which was enabled at the time of creation of private key pair.



```
Using username "hadoop".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
```

EMR Opens, and below screen reflects that Connection to Hadoop is successful:



```
       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
22 package(s) needed for security, out of 32 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-69-167 ~]$
```

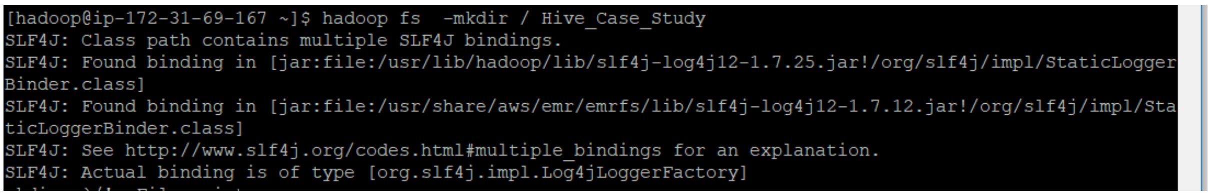
- Create a file directory :

Check the existing directories, and create a directory named ‘Hive_Case_Study’ in Hadoop. > ‘Hive_Case_Study’ is now created in Hadoop:

- Command for create a directory :

Query :

hadoop fs -mkdir /Hive_Case_Study

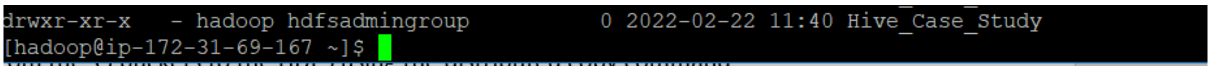


```
[hadoop@ip-172-31-69-167 ~]$ hadoop fs  -mkdir / Hive_Case_Study
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLogger
Binder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/emr/emrfs/lib/slf4j-log4j12-1.7.12.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
```

- To view the directory :

Query:

hadoop fs -ls /



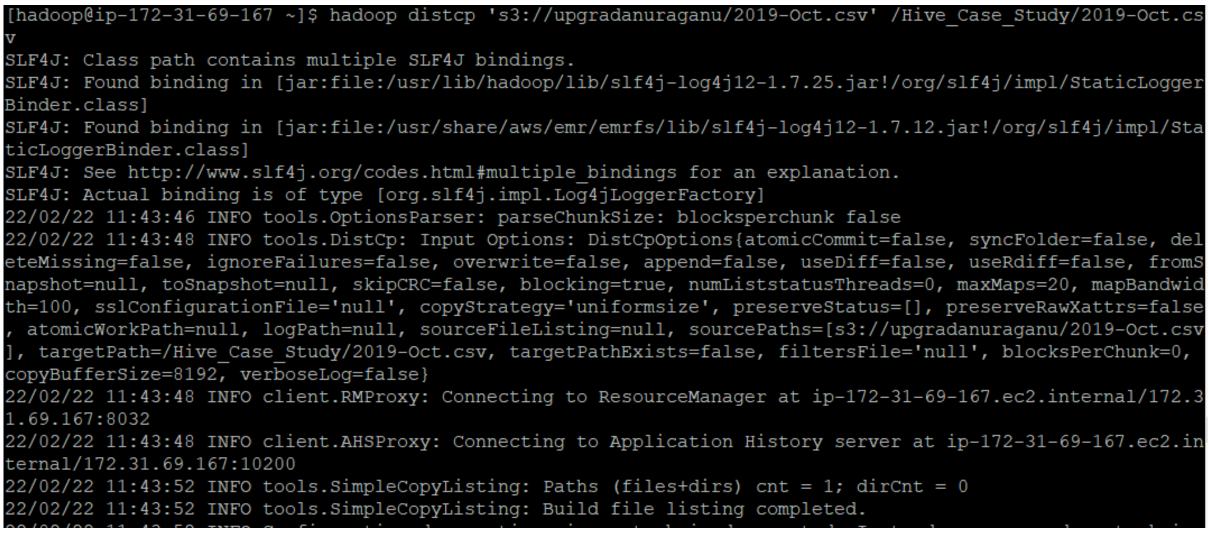
```
drwxr-xr-x   - hadoop hdfsadmingroup          0 2022-02-22 11:40 Hive_Case_Study
[hadoop@ip-172-31-69-167 ~]$
```

2.4. Move the data from the s3 buckets to the HDFS using the distributed copy command.

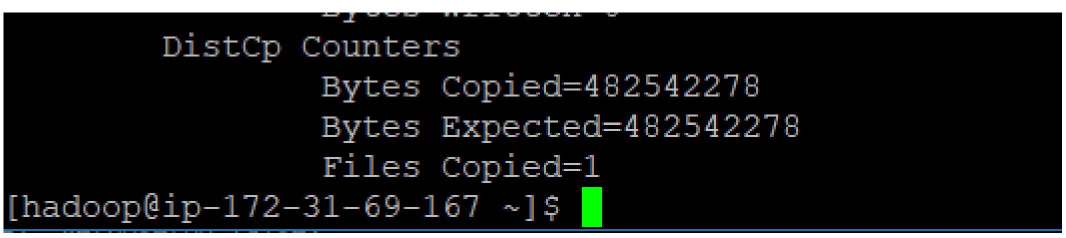
Loading the s3 public data set to created directory “Hive_Case_Study” in hadoop .

Query:

Hadoop distcp ‘s3://upgradanuraganu/2019-Oct.csv’ / Hive_Case_Study/2019-Oct.csv

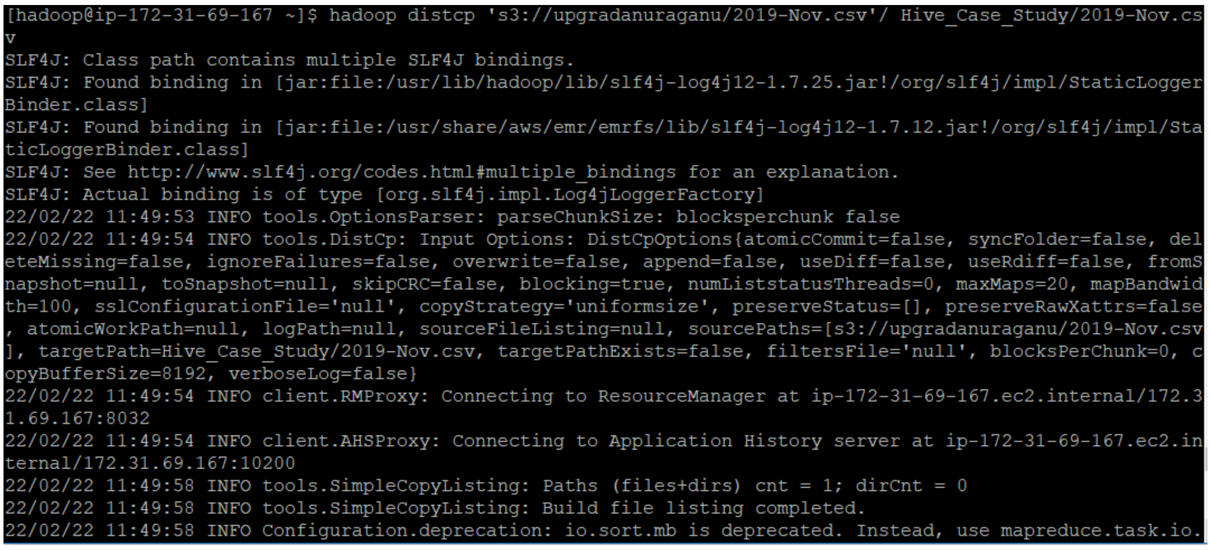


```
[hadoop@ip-172-31-69-167 ~]$ hadoop distcp 's3://upgradanuraganu/2019-Oct.csv' /Hive_Case_Study/2019-Oct.cs
v
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLogger
Binder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/emr/emrfs/lib/slf4j-log4j12-1.7.12.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/02/22 11:43:46 INFO tools.OptionsParser: parseChunkSize: blocksperchunk false
22/02/22 11:43:48 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, del
eteMissing=false, ignoreFailures=false, overwrite=false, append=false, useDiff=false, useRdiff=false, fromS
napshot=null, toSnapshot=null, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwid
th=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false
, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3://upgradanuraganu/2019-Oct.csv
], targetPath=/Hive_Case_Study/2019-Oct.csv, targetPathExists=false, filtersFile='null', blocksPerChunk=0,
copyBufferSize=8192, verboseLog=false}
22/02/22 11:43:48 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-69-167.ec2.internal/172.3
1.69.167:8032
22/02/22 11:43:48 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-69-167.ec2.in
ternal/172.31.69.167:10200
22/02/22 11:43:52 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/02/22 11:43:52 INFO tools.SimpleCopyListing: Build file listing completed.
```

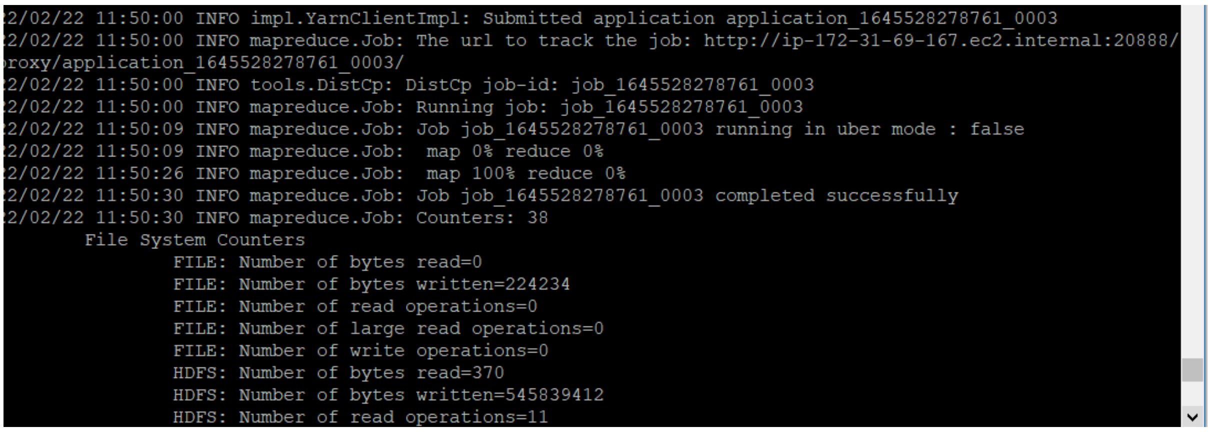


```
        DistCp Counters
                Bytes Copied=482542278
                Bytes Expected=482542278
                Files Copied=1
[hadoop@ip-172-31-69-167 ~]$
```

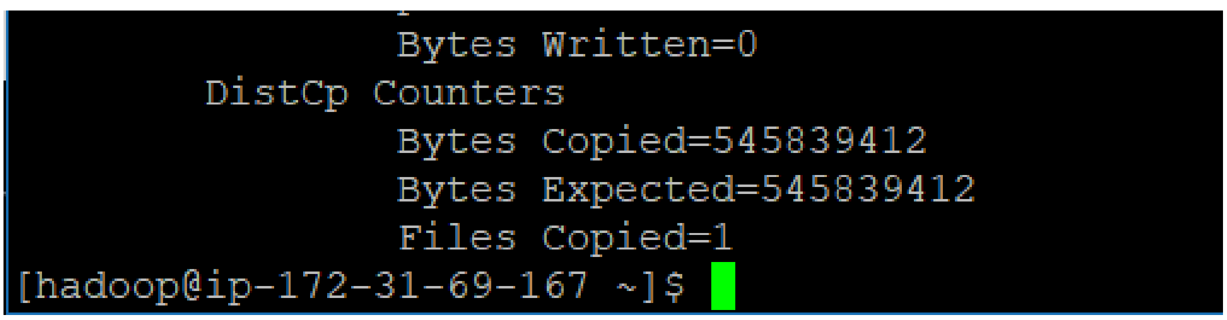
Hadoop distcp ‘s3://upgradanuraganu/2019-Nov.csv’ /Hive_Case_Study/2019-Nov.csv



```
[hadoop@ip-172-31-69-167 ~]$ hadoop distcp 's3://upgradanuraganu/2019-Nov.csv'/ Hive_Case_Study/2019-Nov.cs
v
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLogger
Binder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/emr/emrfs/lib/slf4j-log4j12-1.7.12.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
22/02/22 11:49:53 INFO tools.OptionsParser: parseChunkSize: blocksperchunk false
22/02/22 11:49:54 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, del
eteMissing=false, ignoreFailures=false, overwrite=false, append=false, useDiff=false, useRdiff=false, fromS
napshot=null, toSnapshot=null, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwid
th=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false
, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3://upgradanuraganu/2019-Nov.csv
], targetPath=Hive_Case_Study/2019-Nov.csv, targetPathExists=false, filtersFile='null', blocksPerChunk=0, c
opyBufferSize=8192, verboseLog=false}
22/02/22 11:49:54 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-69-167.ec2.internal/172.3
1.69.167:8032
22/02/22 11:49:54 INFO client.AHSProxy: Connecting to Application History server at ip-172-31-69-167.ec2.in
ternal/172.31.69.167:10200
22/02/22 11:49:58 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/02/22 11:49:58 INFO tools.SimpleCopyListing: Build file listing completed.
22/02/22 11:49:58 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.
```



```
2/02/22 11:50:00 INFO impl.YarnClientImpl: Submitted application application_1645528278761_0003
2/02/22 11:50:00 INFO mapreduce.Job: The url to track the job: http://ip-172-31-69-167.ec2.internal:20888/
roxy/application_1645528278761_0003/
2/02/22 11:50:00 INFO tools.DistCp: DistCp job-id: job_1645528278761_0003
2/02/22 11:50:00 INFO mapreduce.Job: Running job: job_1645528278761_0003
2/02/22 11:50:09 INFO mapreduce.Job: Job job_1645528278761_0003 running in uber mode : false
2/02/22 11:50:09 INFO mapreduce.Job:  map 0% reduce 0%
2/02/22 11:50:26 INFO mapreduce.Job:  map 100% reduce 0%
2/02/22 11:50:30 INFO mapreduce.Job: Job job_1645528278761_0003 completed successfully
2/02/22 11:50:30 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=224234
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=370
                HDFS: Number of bytes written=545839412
                HDFS: Number of read operations=11
```

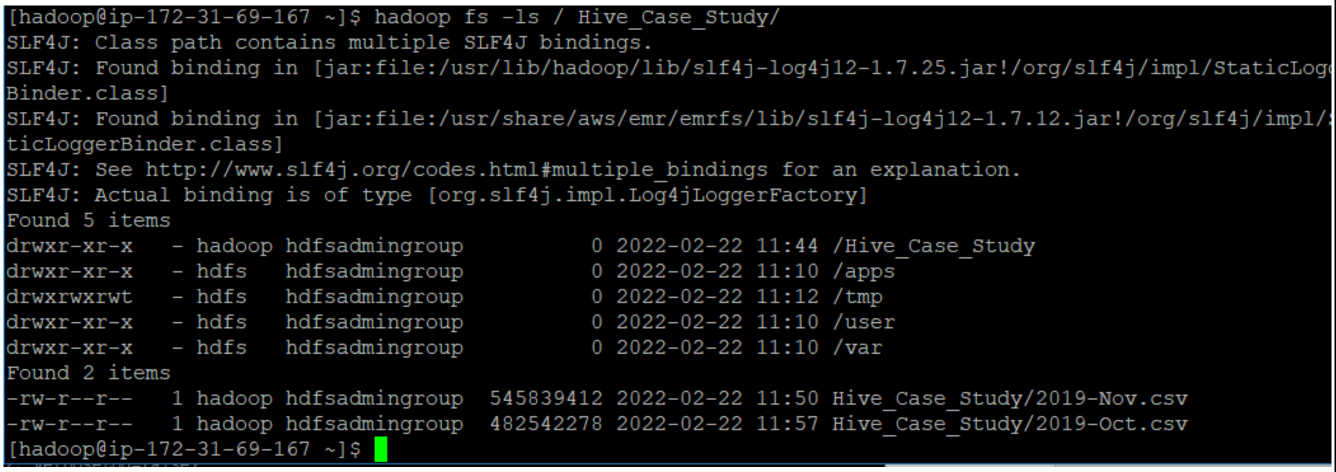


```
                Bytes Written=0
        DistCp Counters
                Bytes Copied=545839412
                Bytes Expected=545839412
                Files Copied=1
[hadoop@ip-172-31-69-167 ~]$
```

- To check if both the files are copied from S3 to HDFS.

Query:

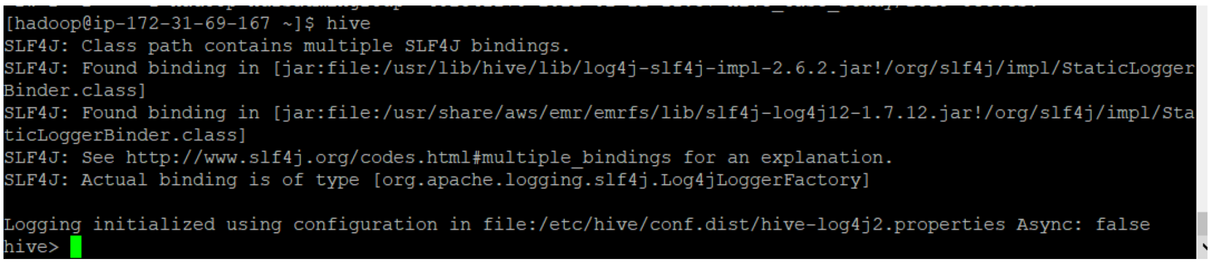
Hadoop fs -ls / Hive_case_study/



```
[hadoop@ip-172-31-69-167 ~]$ hadoop fs -ls / Hive_Case_Study/
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLogg
Binder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/emr/emrfs/lib/slf4j-log4j12-1.7.12.jar!/org/slf4j/impl/
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Found 5 items
drwxr-xr-x   - hadoop hdfsadmingroup          0 2022-02-22 11:44 /Hive_Case_Study
drwxr-xr-x   - hdfs   hdfsadmingroup          0 2022-02-22 11:10 /apps
drwxrwxrwt   - hdfs   hdfsadmingroup          0 2022-02-22 11:12 /tmp
drwxr-xr-x   - hdfs   hdfsadmingroup          0 2022-02-22 11:10 /user
drwxr-xr-x   - hdfs   hdfsadmingroup          0 2022-02-22 11:10 /var
Found 2 items
-rw-r--r--   1 hadoop hdfsadmingroup  545839412 2022-02-22 11:50 Hive_Case_Study/2019-Nov.csv
-rw-r--r--   1 hadoop hdfsadmingroup  482542278 2022-02-22 11:57 Hive_Case_Study/2019-Oct.csv
[hadoop@ip-172-31-69-167 ~]$
```

2.5. Starting Hive:

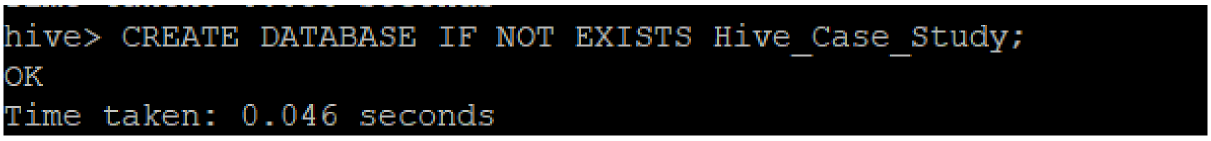
Query :



```
[hadoop@ip-172-31-69-167 ~]$ hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/hive/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLogger
Binder.class]
SLF4J: Found binding in [jar:file:/usr/share/aws/emr/emrfs/lib/slf4j-log4j12-1.7.12.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive>
```

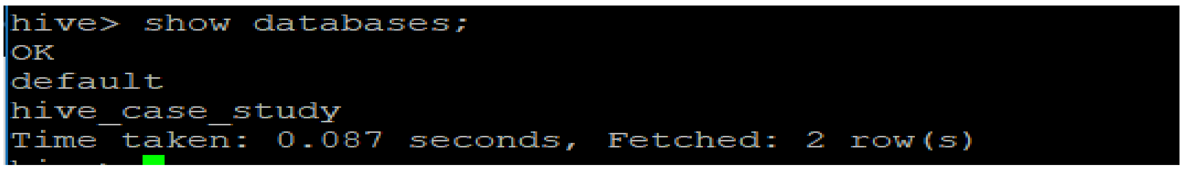
- We will create a database create a table and then load the data which was provided and execute the query to find/analyse the data which was provided.

1. CREATE DATABASE IF NOT EXISTS Hive_Case_Study;



```
hive> CREATE DATABASE IF NOT EXISTS Hive_Case_Study;
OK
Time taken: 0.046 seconds
```

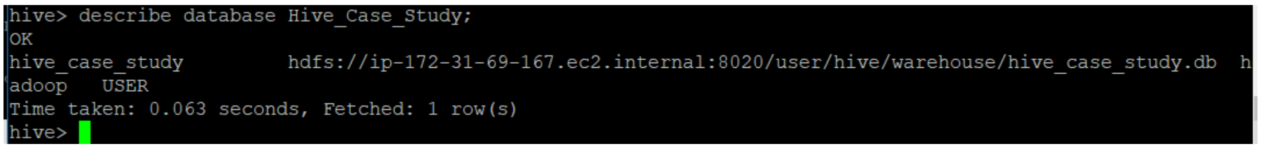
2. Show databases; command to view the data base



```
hive> show databases;
OK
default
hive_case_study
Time taken: 0.087 seconds, Fetched: 2 row(s)
```

3. To describe the database command use is

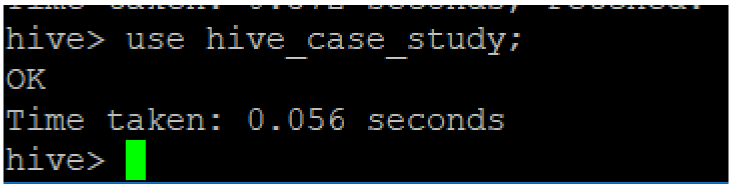
Describe database Hive_Case_Study;



```
hive> describe database Hive_Case_Study;
OK
hive_case_study         hdfs://ip-172-31-69-167.ec2.internal:8020/user/hive/warehouse/hive_case_study.db  h
adoop   USER
Time taken: 0.063 seconds, Fetched: 1 row(s)
hive>
```

4. To use the same database:

Query:



```
hive> use hive_case_study;
OK
Time taken: 0.056 seconds
hive>
```

2.6. Now the database is created we will now Create an external table "cosmeticDB":

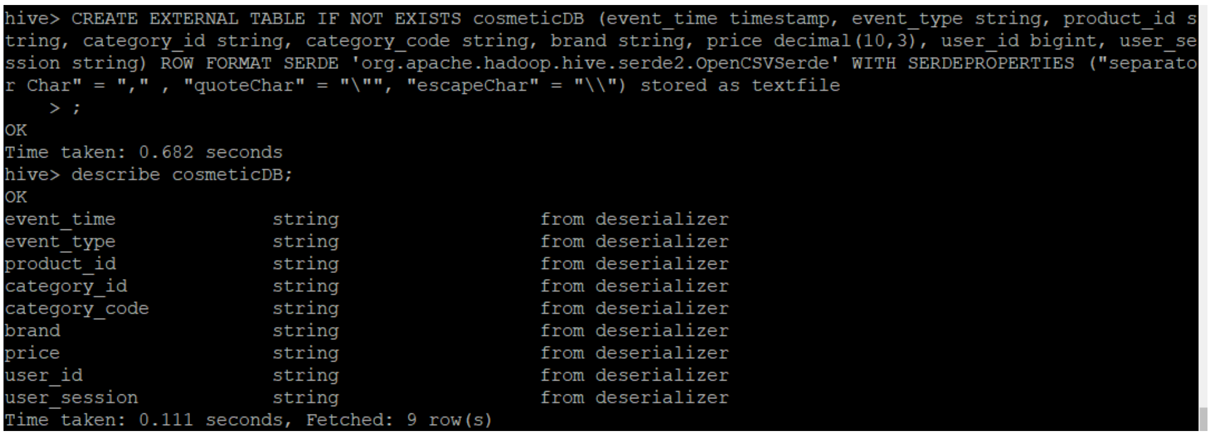
Query:

CREATE EXTERNAL TABLE IF NOT EXISTS cosmeticDB (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_session string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ("separatorChar" = "," , "quoteChar" = "\"", "escapeChar" = "\\") stored as textfile;

1. To view the table command used is (describe table name)

Query:

Describe cometicDB;



```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS cosmeticDB (event_time timestamp, event_type string, product_id s
tring, category_id string, category_code string, brand string, price decimal(10,3), user_id bigint, user_se
ssion string) ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' WITH SERDEPROPERTIES ("separato
r Char" = "," , "quoteChar" = "\"", "escapeChar" = "\\") stored as textfile
    > ;
OK
Time taken: 0.682 seconds
hive> describe cosmeticDB;
OK
event_time              string                  from deserializer
event_type              string                  from deserializer
product_id              string                  from deserializer
category_id             string                  from deserializer
category_code           string                  from deserializer
brand                   string                  from deserializer
price                   string                  from deserializer
user_id                 string                  from deserializer
user_session            string                  from deserializer
Time taken: 0.111 seconds, Fetched: 9 row(s)
```

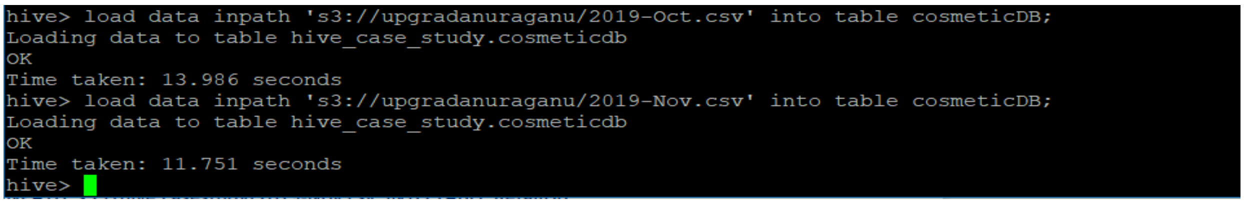
Here if we observe all the data types are in ‘string ‘ by default. This is one of the limitations in serde. We need to cast the data types into desired ones.

2. Load the input Data into “cosmeticDB” table:

Query:

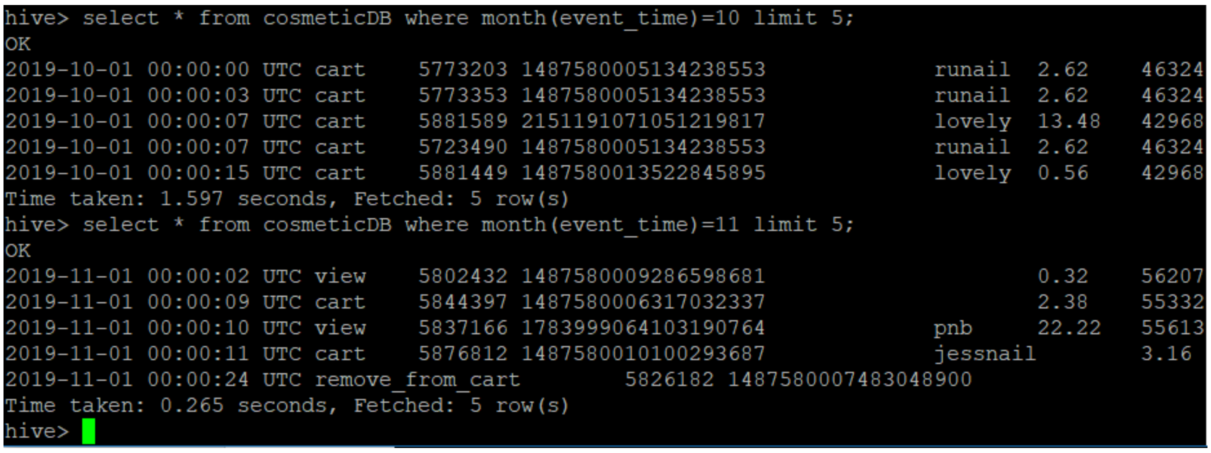
LOAD DATA INPATH ‘s3://upgradanuraganu/2019-Oct.csv’ INTO TABLE cosmeticDB;

LOAD DATA INPATH ‘s3://upgradanuraganu/2019-Nov.csv’ INTO TABLE cosmeticDB;



```
hive> load data inpath 's3://upgradanuraganu/2019-Oct.csv' into table cosmeticDB;
Loading data to table hive_case_study.cosmeticdb
OK
Time taken: 13.986 seconds
hive> load data inpath 's3://upgradanuraganu/2019-Nov.csv' into table cosmeticDB;
Loading data to table hive_case_study.cosmeticdb
OK
Time taken: 11.751 seconds
hive>
```

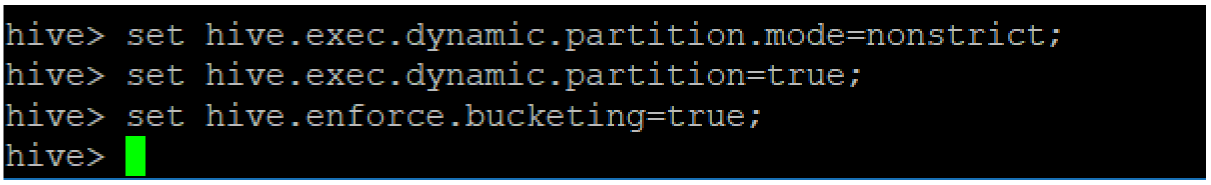
To check if the load data is correct for this, we will write a query to check the month(event_time) because the data is of 11(Nov) and 10(Oct).



```
hive> select * from cosmeticDB where month(event_time)=10 limit 5;
OK
2019-10-01 00:00:00 UTC cart    5773203 1487580005134238553             runail  2.62    46324
2019-10-01 00:00:03 UTC cart    5773353 1487580005134238553             runail  2.62    46324
2019-10-01 00:00:07 UTC cart    5881589 2151191071051219817             lovely  13.48   42968
2019-10-01 00:00:07 UTC cart    5723490 1487580005134238553             runail  2.62    46324
2019-10-01 00:00:15 UTC cart    5881449 1487580013522845895             lovely  0.56    42968
Time taken: 1.597 seconds, Fetched: 5 row(s)
hive> select * from cosmeticDB where month(event_time)=11 limit 5;
OK
2019-11-01 00:00:02 UTC view    5802432 1487580009286598681                     0.32    56207
2019-11-01 00:00:09 UTC cart    5844397 1487580006317032337                     2.38    55332
2019-11-01 00:00:10 UTC view    5837166 1783999064103190764             pnb     22.22   55613
2019-11-01 00:00:11 UTC cart    5876812 1487580010100293687             jessnail        3.16
2019-11-01 00:00:24 UTC remove_from_cart        5826182 1487580007483048900
Time taken: 0.265 seconds, Fetched: 5 row(s)
hive>
```

2.7. Applying Optimization Techniques Columns Partitioning and Bucketing:

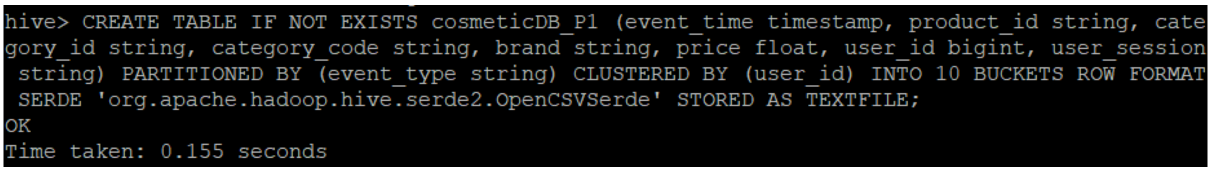
1. Start with enabling the dynamic partitioning and bucketing.



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

2. Create a Table "cosmeticDB_P1", PARTITIONED BY(event_type string) CLUSTERED BY (user _id) INTO 10 buckets o Create a Table "RetailDB_EC2", PARTITIONED BY(event_type string) CLUSTERED BY (user_id) INTO 10 buckets

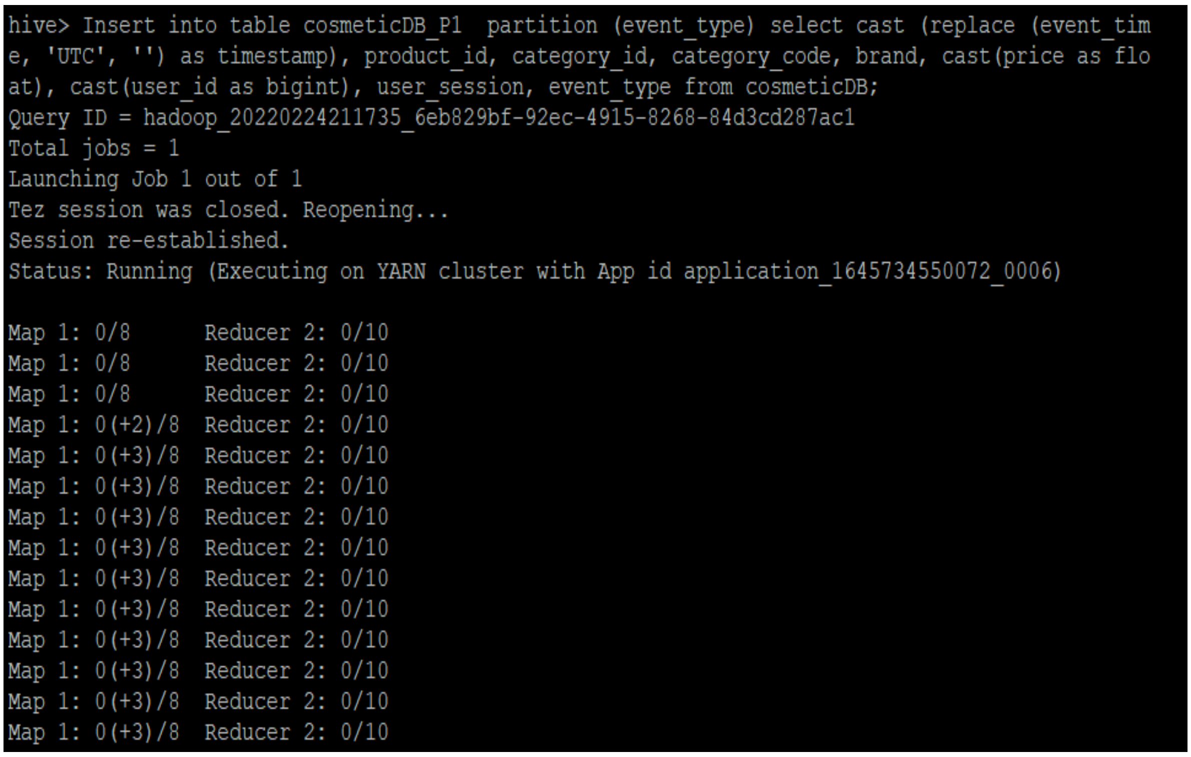
CREATE TABLE IF NOT EXISTS cosmeticDB_P1 (event_time timestamp, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string) PARTITIONED BY (event_type string) CLUSTERED BY (user_id) INTO 10 BUCKETS ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE;



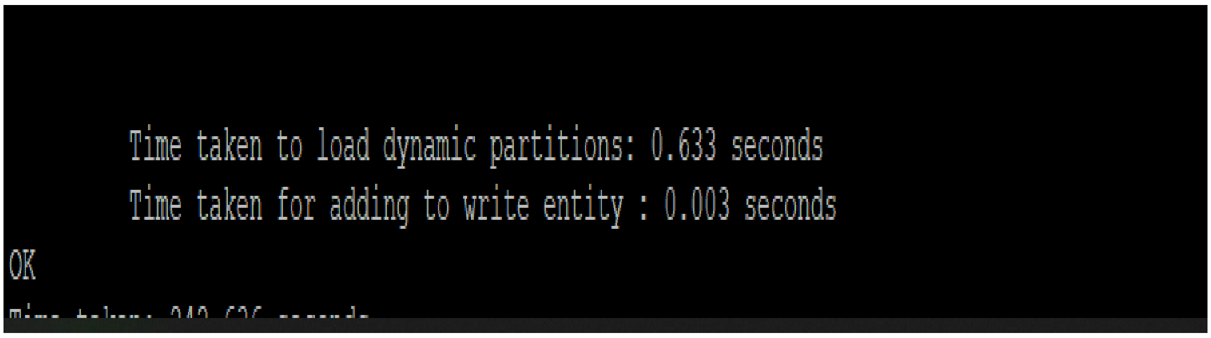
```
hive> CREATE TABLE IF NOT EXISTS cosmeticDB_P1 (event_time timestamp, product_id string, cate
gory_id string, category_code string, brand string, price float, user_id bigint, user_session
 string) PARTITIONED BY (event_type string) CLUSTERED BY (user_id) INTO 10 BUCKETS ROW FORMAT
 SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE;
OK
Time taken: 0.155 seconds
```

3. Load the Data into the tables:

Insert into table cosmeticDB_P1 partition (event_type) select cast (replace (event_time, 'UTC', '') as timestamp), product_id, category_id, category_code, brand, cast(price as float), cast(user_id as bigint), user_session, event_type from cosmeticDB;



```
hive> Insert into table cosmeticDB_P1  partition (event_type) select cast (replace (event_tim
e, 'UTC', '') as timestamp), product_id, category_id, category_code, brand, cast(price as flo
at), cast(user_id as bigint), user_session, event_type from cosmeticDB;
Query ID = hadoop_20220224211735_6eb829bf-92ec-4915-8268-84d3cd287ac1
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_1645734550072_0006)

Map 1: 0/8      Reducer 2: 0/10
Map 1: 0/8      Reducer 2: 0/10
Map 1: 0/8      Reducer 2: 0/10
Map 1: 0(+2)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
Map 1: 0(+3)/8  Reducer 2: 0/10
```



```
        Time taken to load dynamic partitions: 0.633 seconds
        Time taken for adding to write entity : 0.003 seconds
OK
Time taken: 240.636 seconds
```

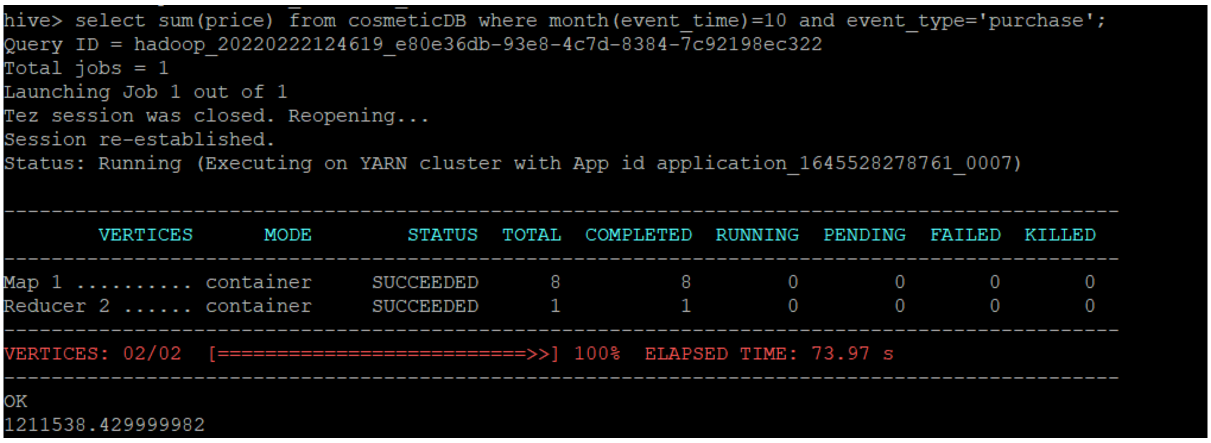
3. Question and Answer:

Q1. Find the total revenue generated due to purchases made in October.

Query:

SELECT sum(price) FROM cosmeticDB WHERE month(event_time)=10 and event_type='purchase';

Normal Table : (Without Partitioning)

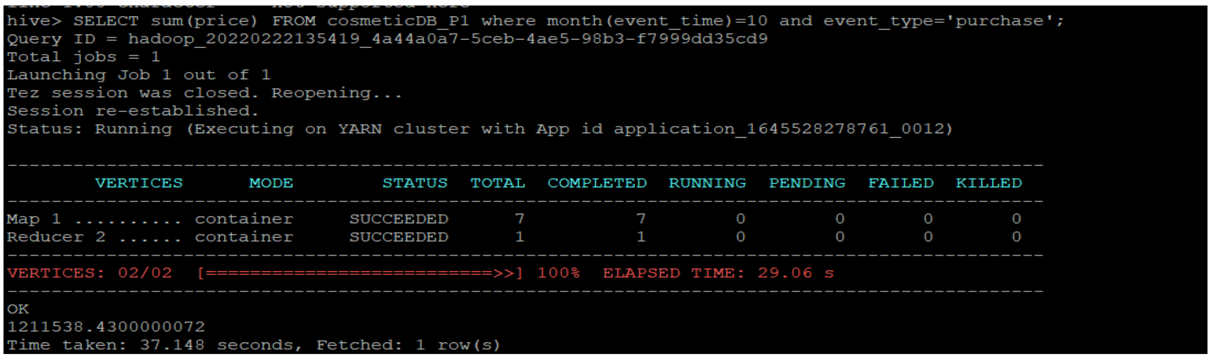


```
hive> select sum(price) from cosmeticDB where month(event_time)=10 and event_type='purchase';
Query ID = hadoop_20220222124619_e80e36db-93e8-4c7d-8384-7c92198ec322
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1645528278761_0007)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      8          8        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 73.97 s
----------------------------------------------------------------------------------------------
OK
1211538.429999982
```

Partition Table :

SELECT sum(price) FROM cosmeticDB_P1 where month(event_time)=10 and event_time='purchase';

If we compare the query optimization of partitioned table it is almost 43 second's less than normal table.

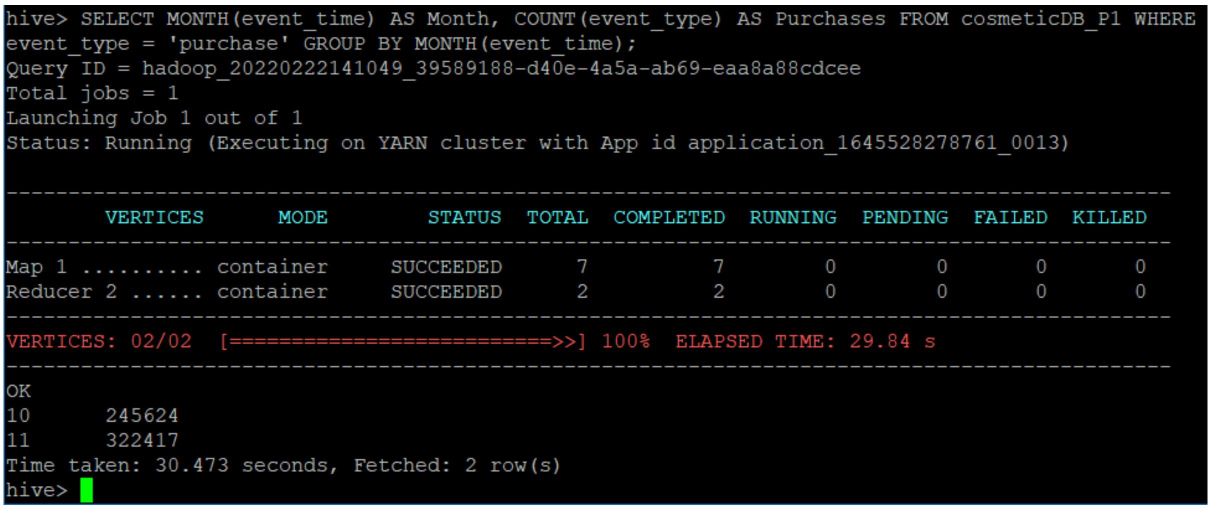


```
hive> SELECT sum(price) FROM cosmeticDB_P1 where month(event_time)=10 and event_type='purchase';
Query ID = hadoop_20220222135419_4a44a0a7-5ceb-4ae5-98b3-f7999dd35cd9
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1645528278761_0012)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 29.06 s
----------------------------------------------------------------------------------------------
OK
1211538.4300000072
Time taken: 37.148 seconds, Fetched: 1 row(s)
```

Q2. Write a query to yield the total sum of purchases per month in a single output.

Query:

SELECT MONTH(event_time) AS Month, COUNT(event_type) AS Purchases FROM cosmeticDB_P1 WHERE event_type = 'purchase' GROUP BY MONTH(event_time);

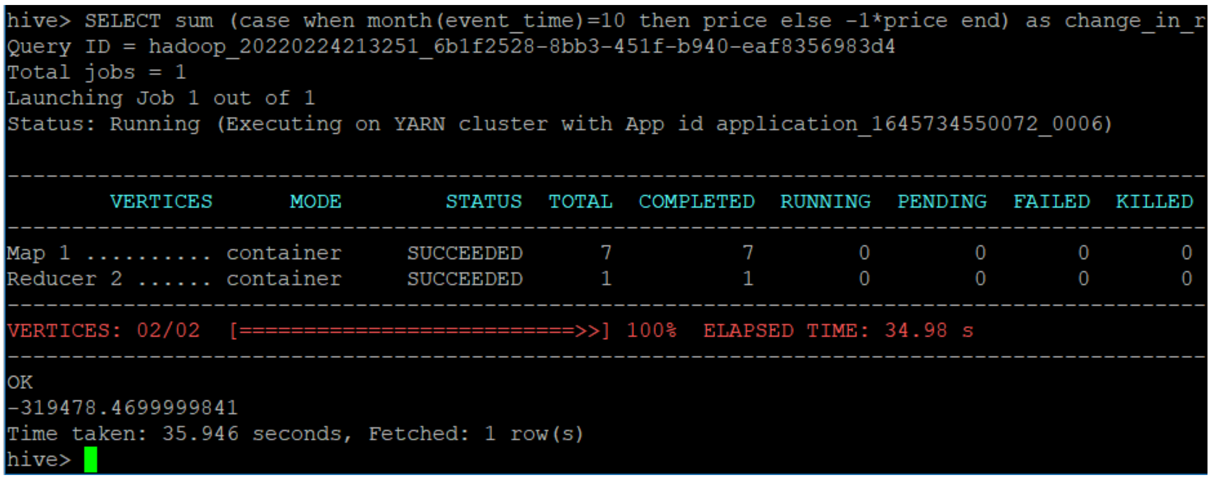


```
hive> SELECT MONTH(event_time) AS Month, COUNT(event_type) AS Purchases FROM cosmeticDB_P1 WHERE
event_type = 'purchase' GROUP BY MONTH(event_time);
Query ID = hadoop_20220222141049_39589188-d40e-4a5a-ab69-eaa8a88cdcee
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1645528278761_0013)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 29.84 s
----------------------------------------------------------------------------------------------
OK
10      245624
11      322417
Time taken: 30.473 seconds, Fetched: 2 row(s)
hive>
```

Q3. Find the total revenue generated due to purchases made in October.

Query:

SELECT sum (case when month(event_time)=10 then price else -1*price end) as change_in_revenue FROM cosmeticDB_P1 WHERE month(event_time) in (10,11) and event_type='purchase';

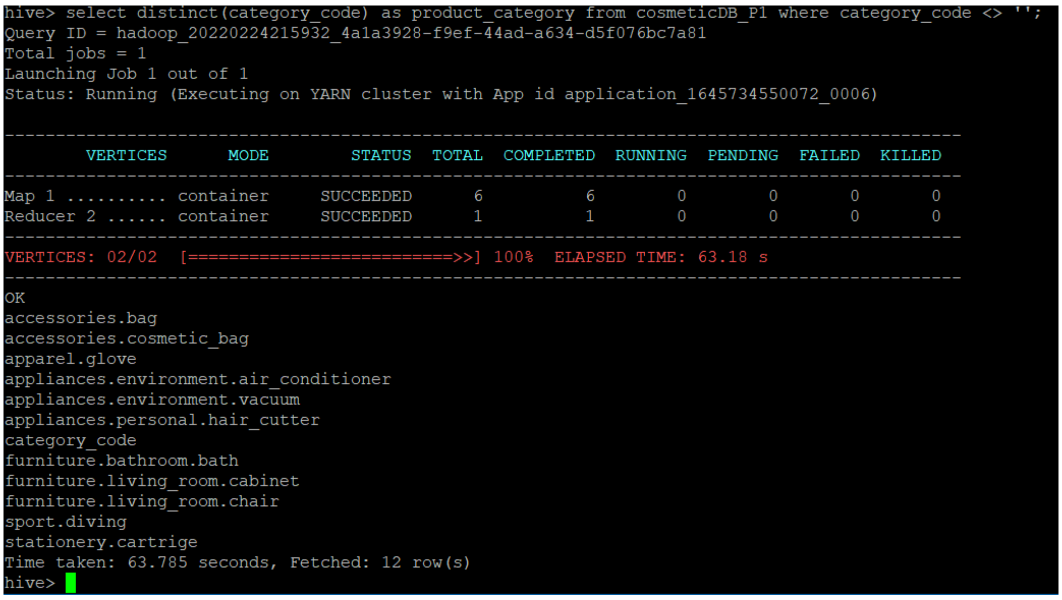


```
hive> SELECT sum (case when month(event_time)=10 then price else -1*price end) as change_in_r
Query ID = hadoop_20220224213251_6b1f2528-8bb3-451f-b940-eaf8356983d4
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1645734550072_0006)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 34.98 s
----------------------------------------------------------------------------------------------
OK
-319478.4699999841
Time taken: 35.946 seconds, Fetched: 1 row(s)
hive>
```

Q4. Find distinct categories of products. Categories with null category code can be ignored.

Query :

select distinct category_code as product_category from cosmeticDB_P1 where category_code <> ‘ ‘

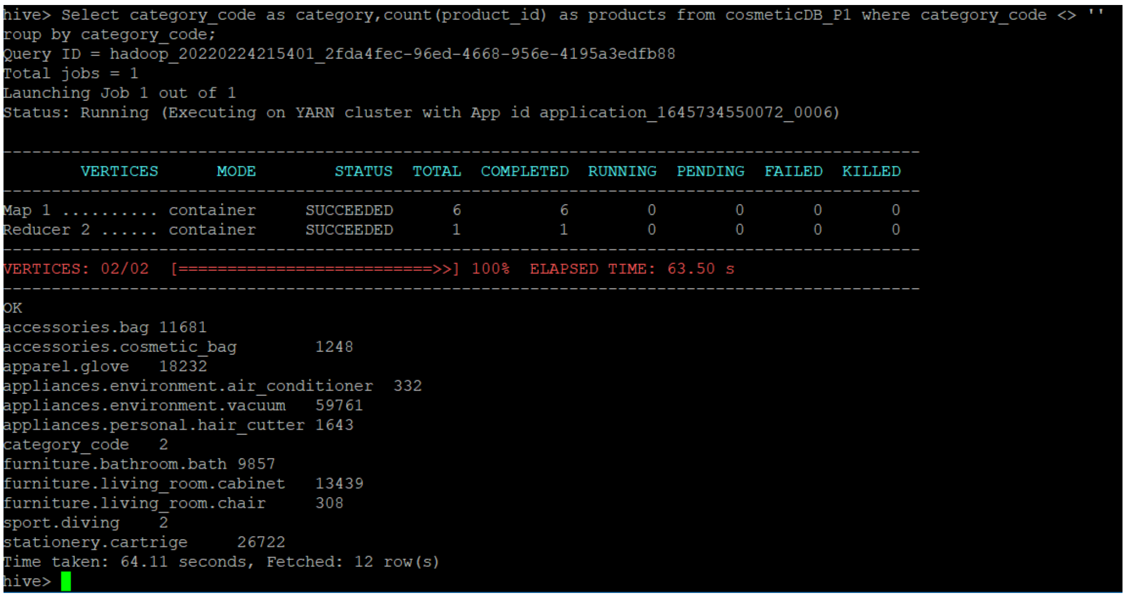


```
hive> select distinct(category_code) as product_category from cosmeticDB_P1 where category_code <> '';
Query ID = hadoop_20220224215932_4a1a3928-f9ef-44ad-a634-d5f076bc7a81
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1645734550072_0006)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 63.18 s
----------------------------------------------------------------------------------------------
OK
accessories.bag
accessories.cosmetic_bag
apparel.glove
appliances.environment.air_conditioner
appliances.environment.vacuum
appliances.personal.hair_cutter
category_code
furniture.bathroom.bath
furniture.living_room.cabinet
furniture.living_room.chair
sport.diving
stationery.cartrige
Time taken: 63.785 seconds, Fetched: 12 row(s)
hive>
```

Q5. Find the total number of products available under each category.

Query :

Select category_code as category,count(product_id) as products from cosmeticDB_P1 where category_code <> ‘ ’ group by category_code;

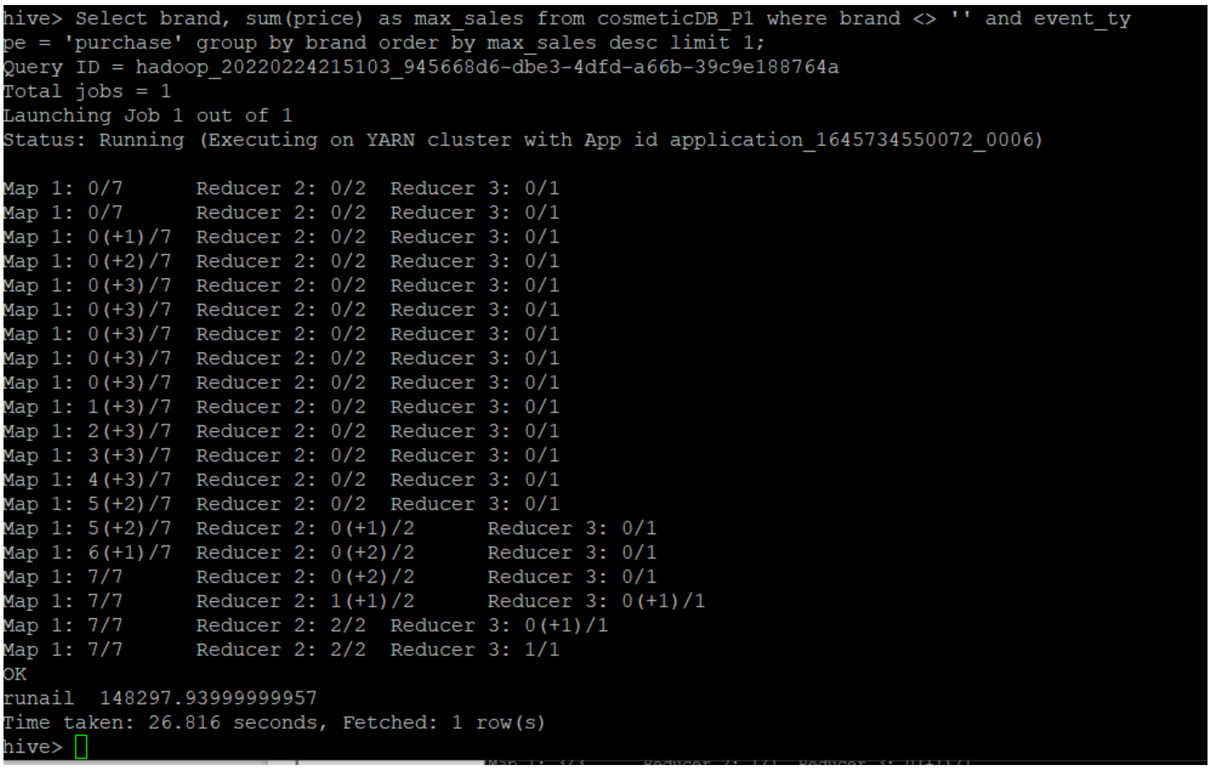


```
hive> Select category_code as category,count(product_id) as products from cosmeticDB_P1 where category_code <> ''
roup by category_code;
Query ID = hadoop_20220224215401_2fda4fec-96ed-4668-956e-4195a3edfb88
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1645734550072_0006)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 63.50 s
----------------------------------------------------------------------------------------------
OK
accessories.bag 11681
accessories.cosmetic_bag        1248
apparel.glove   18232
appliances.environment.air_conditioner  332
appliances.environment.vacuum   59761
appliances.personal.hair_cutter 1643
category_code   2
furniture.bathroom.bath 9857
furniture.living_room.cabinet   13439
furniture.living_room.chair     308
sport.diving    2
stationery.cartrige     26722
Time taken: 64.11 seconds, Fetched: 12 row(s)
hive>
```

Q6. Which brand had the maximum sales in October and November combined?

Query :

Select brand, sum(price) as max_sales from cosmeticDB_P1 where brand <> ‘ ’ and event_type = 'purchase' group by brand order by max_sales desc limit 1;



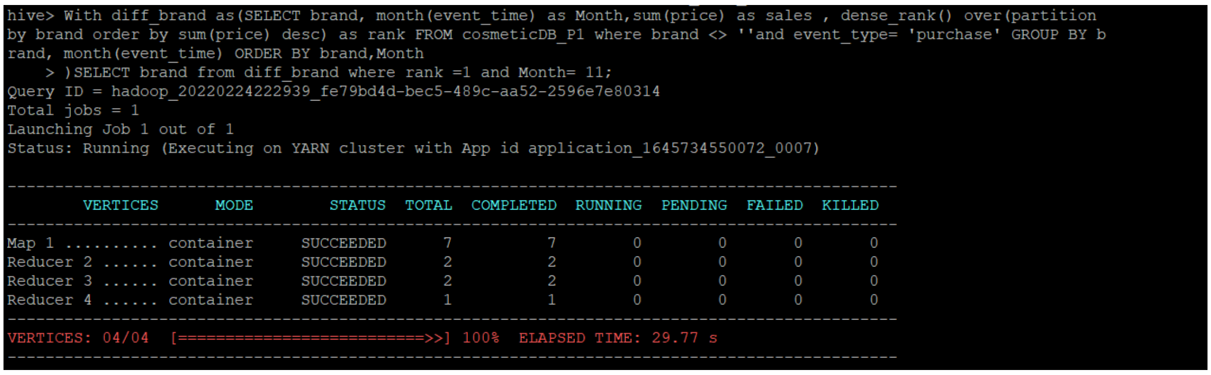
```
hive> Select brand, sum(price) as max_sales from cosmeticDB_P1 where brand <> '' and event_ty
pe = 'purchase' group by brand order by max_sales desc limit 1;
Query ID = hadoop_20220224215103_945668d6-dbe3-4dfd-a66b-39c9e188764a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1645734550072_0006)

Map 1: 0/7      Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0/7      Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+1)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+2)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 0(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 1(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 2(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 3(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 4(+3)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 5(+2)/7  Reducer 2: 0/2  Reducer 3: 0/1
Map 1: 5(+2)/7  Reducer 2: 0(+1)/2      Reducer 3: 0/1
Map 1: 6(+1)/7  Reducer 2: 0(+2)/2      Reducer 3: 0/1
Map 1: 7/7      Reducer 2: 0(+2)/2      Reducer 3: 0/1
Map 1: 7/7      Reducer 2: 1(+1)/2      Reducer 3: 0(+1)/1
Map 1: 7/7      Reducer 2: 2/2  Reducer 3: 0(+1)/1
Map 1: 7/7      Reducer 2: 2/2  Reducer 3: 1/1
OK
runail  148297.93999999957
Time taken: 26.816 seconds, Fetched: 1 row(s)
hive>
```

Q7. Which brands increased their sales from October to November?

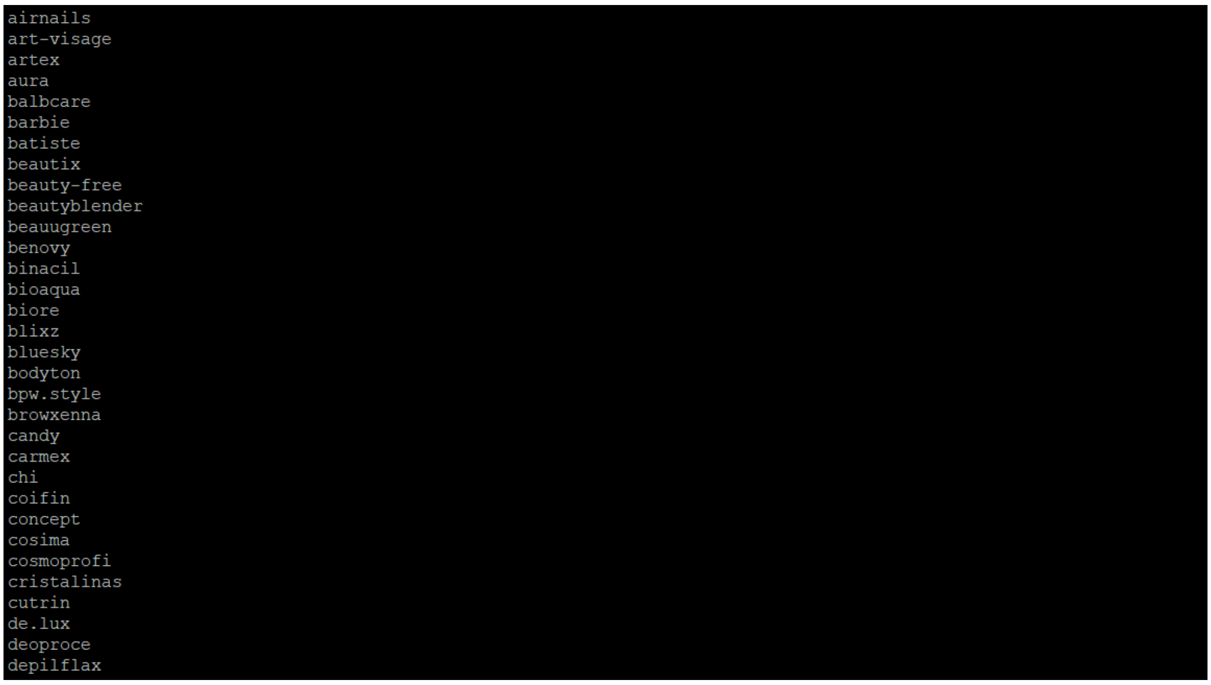
Query:

With diff_brand as (SELECT brand, month(event_time) as Month,sum(price) as sales , dense_rank() over(partition by brand order by sum(price) desc) as rank FROM cosmeticDB_P1 where brand <> ''and event_type= 'purchase' GROUP BY brand, month(event_time) ORDER BY brand,Month)

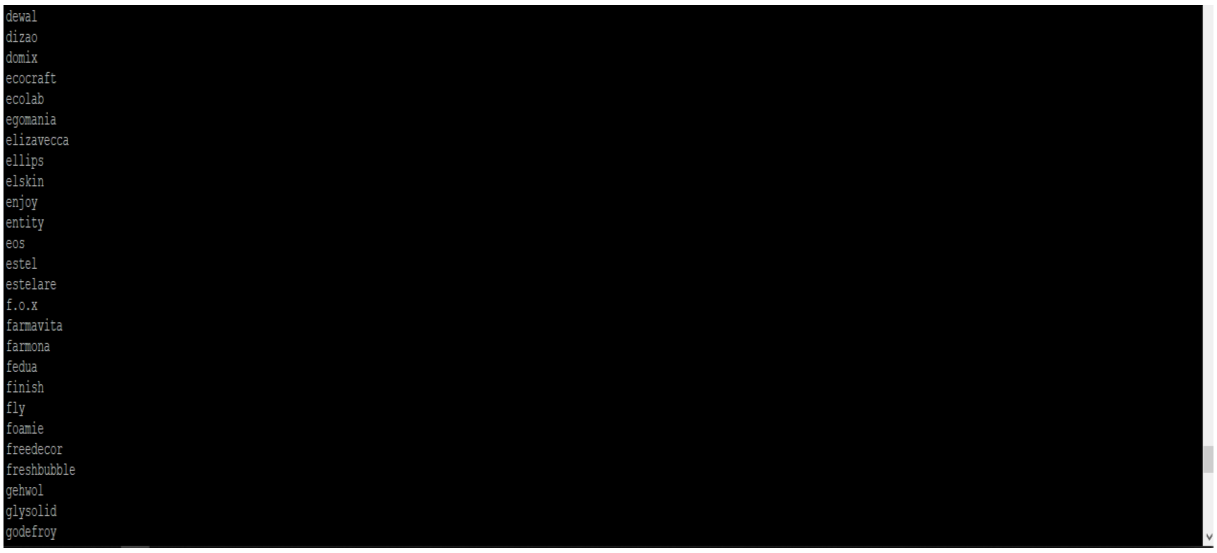
SELECT brand from diff_brand where rank =1 and Month= 11



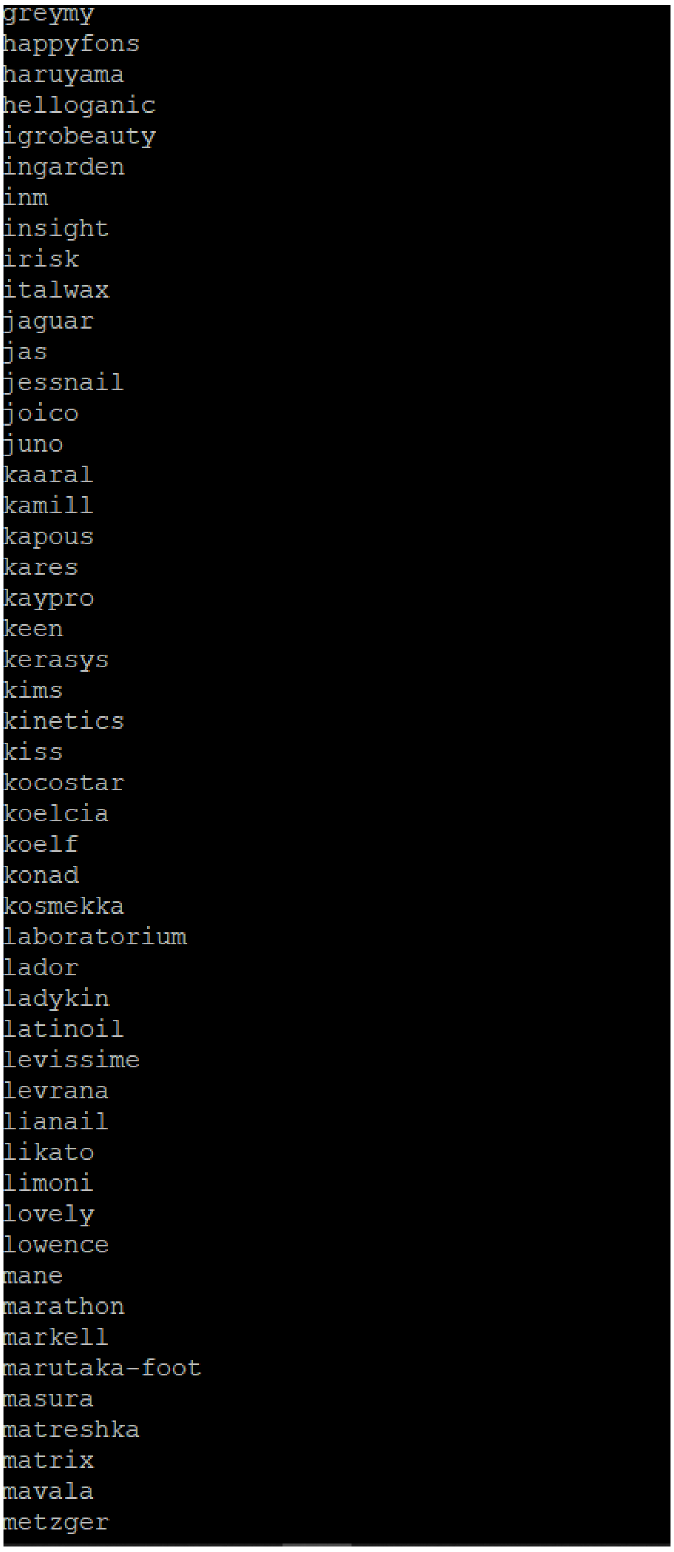
```
hive> With diff_brand as(SELECT brand, month(event_time) as Month,sum(price) as sales , dense_rank() over(partition
by brand order by sum(price) desc) as rank FROM cosmeticDB_P1 where brand <> ''and event_type= 'purchase' GROUP BY b
rand, month(event_time) ORDER BY brand,Month
    > )SELECT brand from diff_brand where rank =1 and Month= 11;
Query ID = hadoop_20220224222939_fe79bd4d-bec5-489c-aa52-2596e7e80314
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1645734550072_0007)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      2          2        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 04/04  [==========================>>] 100%  ELAPSED TIME: 29.77 s
----------------------------------------------------------------------------------------------
```



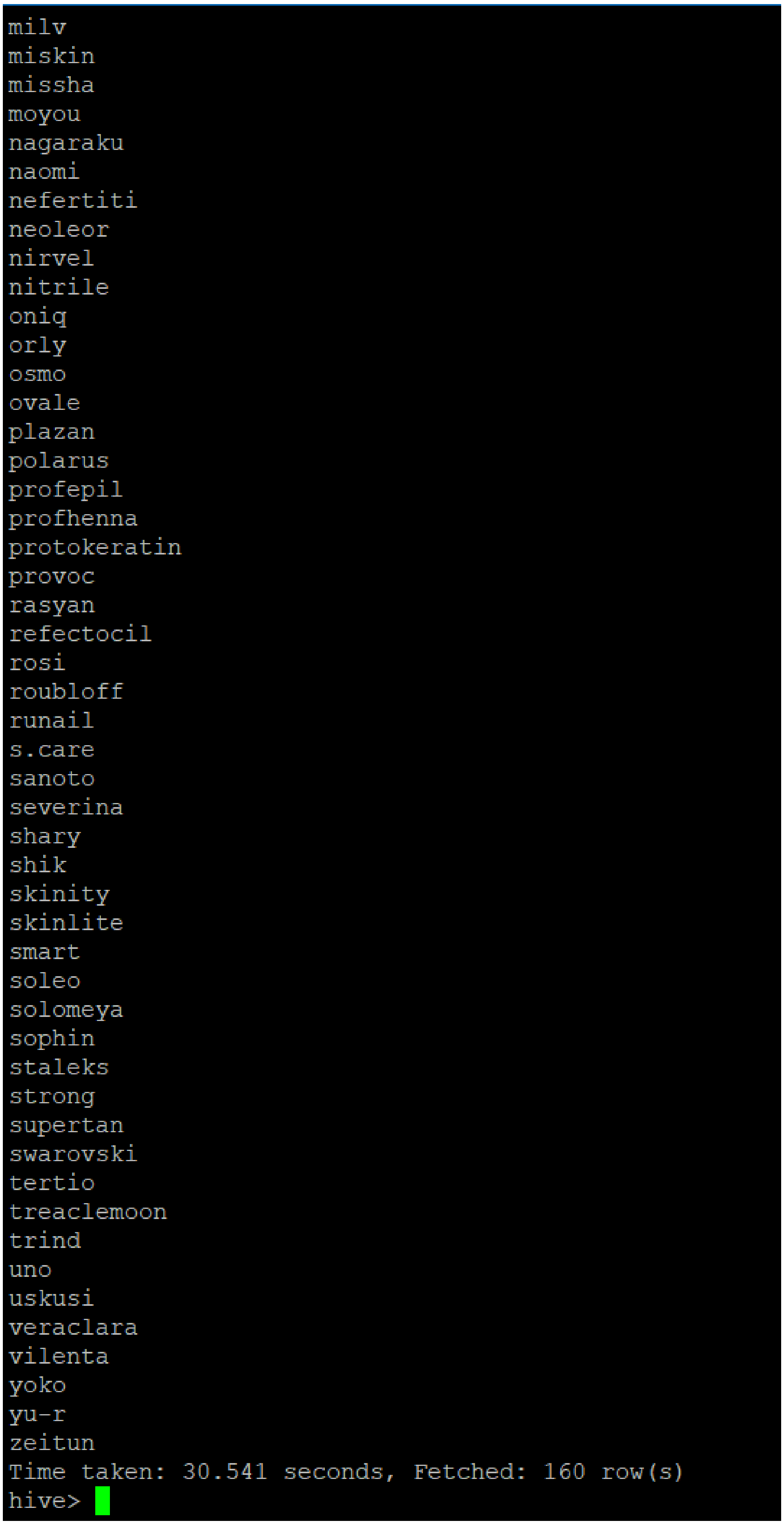
```
airnails
art-visage
artex
aura
balbcare
barbie
batiste
beautix
beauty-free
beautyblender
beauugreen
benovy
binacil
bioaqua
biore
blixz
bluesky
bodyton
bpw.style
browxenna
candy
carmex
chi
coifin
concept
cosima
cosmoprofi
cristalinas
cutrin
de.lux
deoproce
depilflax
```



```
dewal
dizao
domix
ecocraft
ecolab
egomania
elizavecca
ellips
elskin
enjoy
entity
eos
estel
estelare
f.o.x
farmavita
farmona
fedua
finish
fly
foamie
freedecor
freshbubble
gehwol
glysolid
godefroy
```



```
greymy
happyfons
haruyama
helloganic
igrobeauty
ingarden
inm
insight
irisk
italwax
jaguar
jas
jessnail
joico
juno
kaaral
kamill
kapous
kares
kaypro
keen
kerasys
kims
kinetics
kiss
kocostar
koelcia
koelf
konad
kosmekka
laboratorium
lador
ladykin
latinoil
levissime
levrana
lianail
likato
limoni
lovely
lowence
mane
marathon
markell
marutaka-foot
masura
matreshka
matrix
mavala
metzger
```

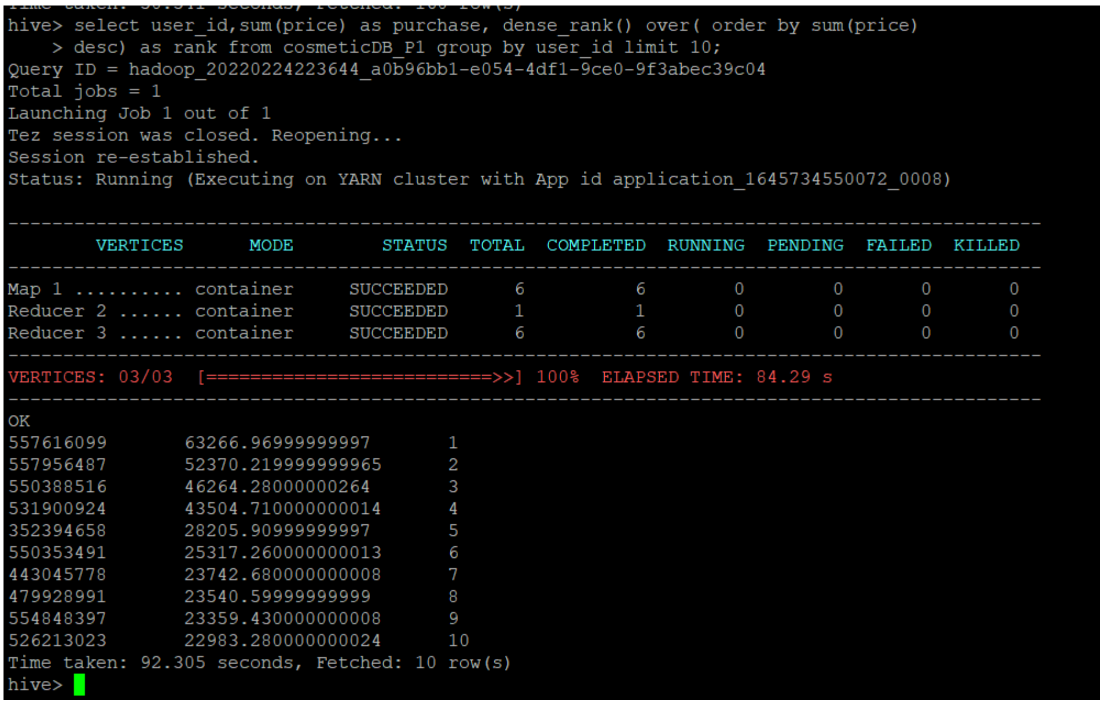


```
milv
miskin
missha
moyou
nagaraku
naomi
nefertiti
neoleor
nirvel
nitrile
oniq
orly
osmo
ovale
plazan
polarus
profepil
profhenna
protokeratin
provoc
rasyan
refectocil
rosi
roubloff
runail
s.care
sanoto
severina
shary
shik
skinity
skinlite
smart
soleo
solomeya
sophin
staleks
strong
supertan
swarovski
tertio
treaclemoon
trind
uno
uskusi
veraclara
vilenta
yoko
yu-r
zeitun
Time taken: 30.541 seconds, Fetched: 160 row(s)
hive>
```

Q8. 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 purchase, dense_rank() over(order by sum(price) desc) as rank from cosmeticDB_P1 where event_type= 'purchase' group by user_id limit 10;



```
hive> select user_id,sum(price) as purchase, dense_rank() over( order by sum(price)
    > desc) as rank from cosmeticDB_P1 group by user_id limit 10;
Query ID = hadoop_20220224223644_a0b96bb1-e054-4df1-9ce0-9f3abec39c04
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_1645734550072_0008)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      6          6        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 84.29 s
----------------------------------------------------------------------------------------------
OK
557616099       63266.96999999997       1
557956487       52370.219999999965      2
550388516       46264.28000000264       3
531900924       43504.710000000014      4
352394658       28205.90999999997       5
550353491       25317.260000000013      6
443045778       23742.680000000008      7
479928991       23540.59999999999       8
554848397       23359.430000000008      9
526213023       22983.280000000024      10
Time taken: 92.305 seconds, Fetched: 10 row(s)
hive>
```

Observations:

1. For less query time we should always partition the table and also make bucket.
2. In our data set also we observed that performance rate increased when we used the partitioned table.
3. In event_type table compared to all the other event types purchase is less than other two event types.
4. Highest number of products available under appliances. environment.vaccume category.
5. The total revenue is high in November month than October month.
6. Runail brand has highest sales compared with other brands.
7. The user_id: 557616099 spent most compared to others and selected for reward and recognition program.

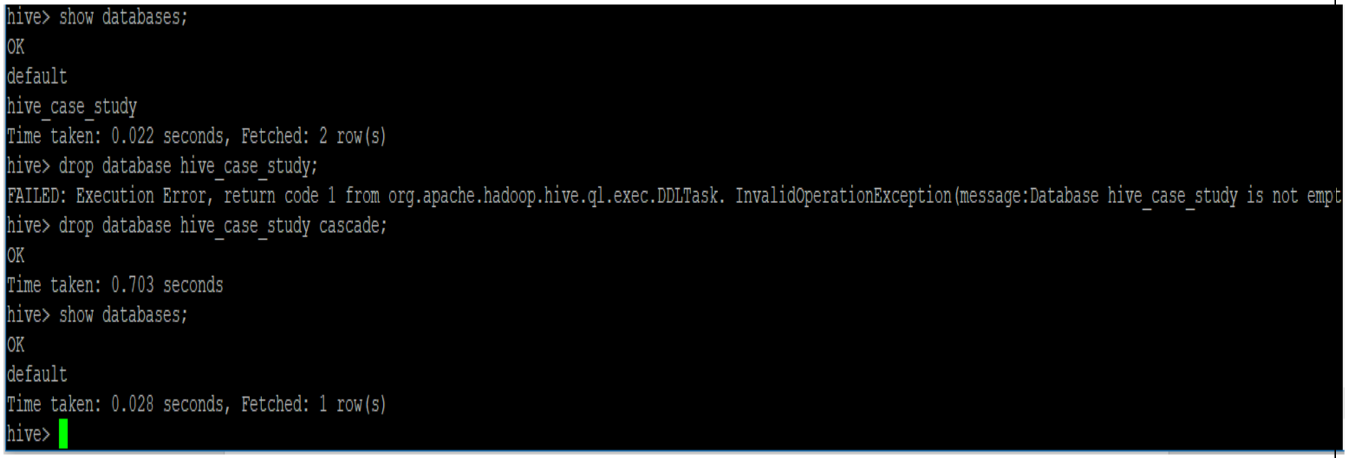
4. Dropping the Database and Terminating the Cluster

For dropping the database :

Query:

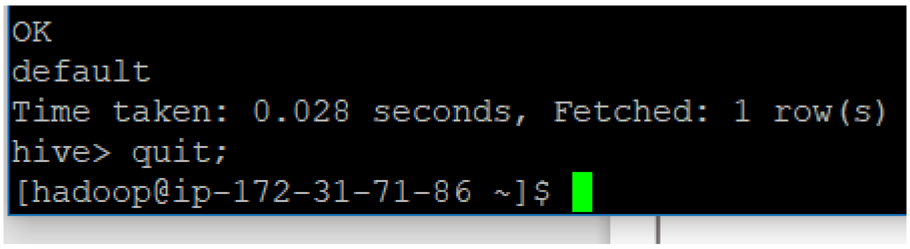
Show database;

Drop database hive_case_study cascade; **(we use cascade to delete the table which is created in database. Without that it will give an error.)**



```
hive> show databases;
OK
default
hive_case_study
Time taken: 0.022 seconds, Fetched: 2 row(s)
hive> drop database hive_case_study;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask. InvalidOperationException(message:Database hive_case_study is not empt
hive> drop database hive_case_study cascade;
OK
Time taken: 0.703 seconds
hive> show databases;
OK
default
Time taken: 0.028 seconds, Fetched: 1 row(s)
hive>
```

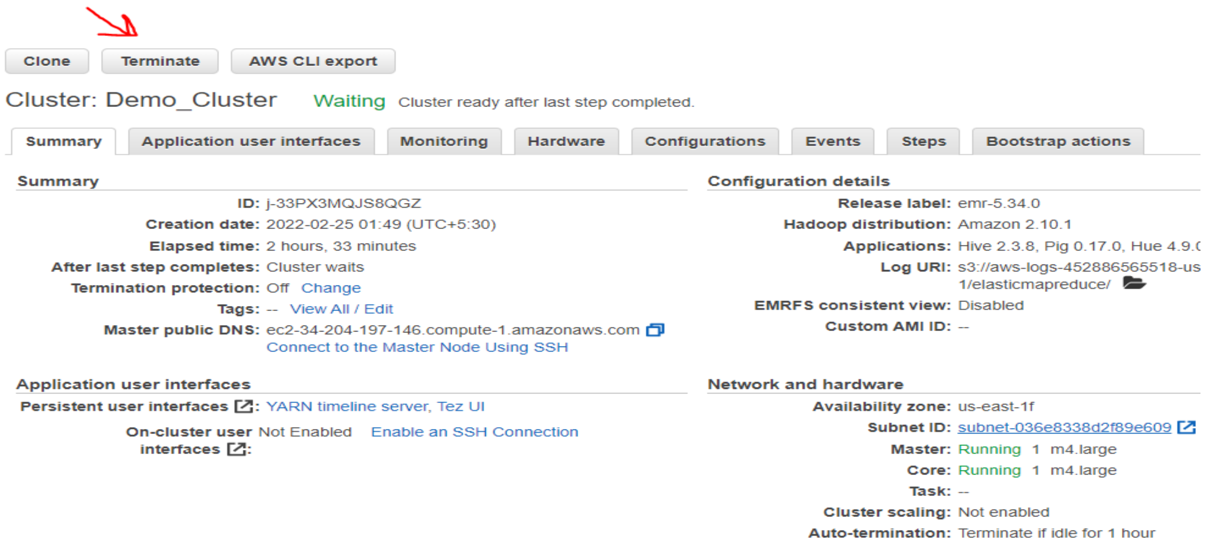
Quitting Hive from PuTTY :



```
OK
default
Time taken: 0.028 seconds, Fetched: 1 row(s)
hive> quit;
[hadoop@ip-172-31-71-86 ~]$
```

Terminate EMR Cluster

• Select Terminate > Select Terminate in the pop-up window >Status changes to Terminating.



Cluster: Demo_Cluster Waiting Cluster ready after last step completed.

Summary

ID: j-33PX3MQJS8QGZ
Creation date: 2022-02-25 01:49 (UTC+5:30)
Elapsed time: 2 hours, 33 minutes
After last step completes: Cluster waits
Termination protection: Off Change
Tags: -- View All / Edit
Master public DNS: ec2-34-204-197-146.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

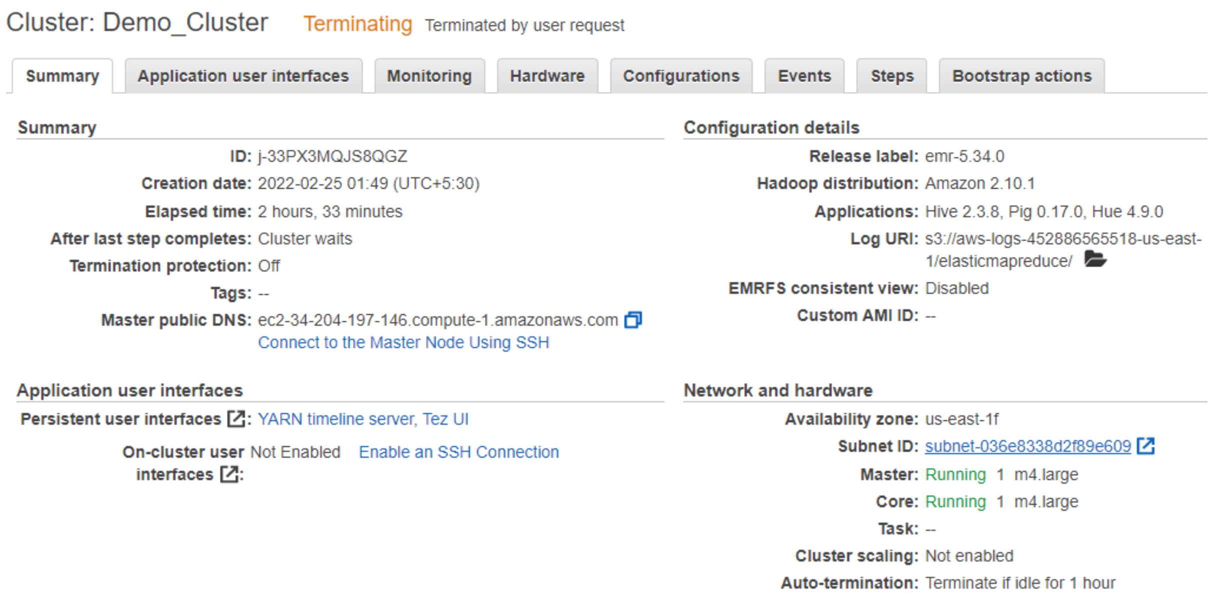
Release label: emr-5.34.0
Hadoop distribution: Amazon 2.10.1
Applications: Hive 2.3.8, Pig 0.17.0, Hue 4.9.(
Log URI: s3://aws-logs-452886565518-us
1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

Persistent user interfaces: YARN timeline server, Tez UI
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Network and hardware

Availability zone: us-east-1f
Subnet ID: subnet-036e8338d2f89e609
Master: Running 1 m4.large
Core: Running 1 m4.large
Task: --
Cluster scaling: Not enabled
Auto-termination: Terminate if idle for 1 hour



Cluster: Demo_Cluster Terminating Terminated by user request

Summary

ID: j-33PX3MQJS8QGZ
Creation date: 2022-02-25 01:49 (UTC+5:30)
Elapsed time: 2 hours, 33 minutes
After last step completes: Cluster waits
Termination protection: Off
Tags: --
Master public DNS: ec2-34-204-197-146.compute-1.amazonaws.com
Connect to the Master Node Using SSH

Configuration details

Release label: emr-5.34.0
Hadoop distribution: Amazon 2.10.1
Applications: Hive 2.3.8, Pig 0.17.0, Hue 4.9.0
Log URI: s3://aws-logs-452886565518-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

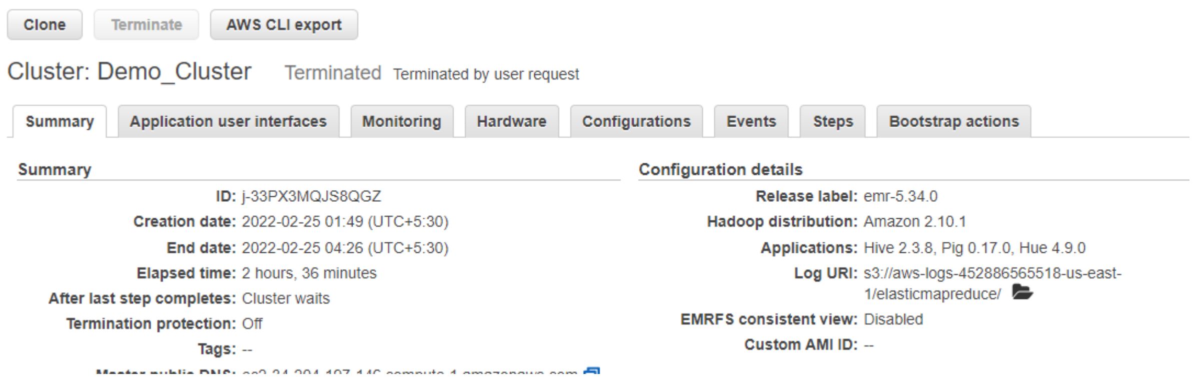
Application user interfaces

Persistent user interfaces: YARN timeline server, Tez UI
On-cluster user interfaces: Not Enabled Enable an SSH Connection

Network and hardware

Availability zone: us-east-1f
Subnet ID: subnet-036e8338d2f89e609
Master: Running 1 m4.large
Core: Running 1 m4.large
Task: --
Cluster scaling: Not enabled
Auto-termination: Terminate if idle for 1 hour

Cluster is successfully terminated :



Cluster: Demo_Cluster Terminated Terminated by user request

Summary

ID:	j-33PX3MQJS8QGZ
Creation date:	2022-02-25 01:49 (UTC+5:30)
End date:	2022-02-25 04:26 (UTC+5:30)
Elapsed time:	2 hours, 36 minutes
After last step completes:	Cluster waits
Termination protection:	Off
Tags:	--

Configuration details

Release label:	emr-5.34.0
Hadoop distribution:	Amazon 2.10.1
Applications:	Hive 2.3.8, Pig 0.17.0, Hue 4.9.0
Log URI:	s3://aws-logs-452886565518-us-east-1/elasticmapreduce/
EMRFS consistent view:	Disabled
Custom AMI ID:	--

Made By :

Anurag Aditya & Hazarathaiah G