

# HIVE CASE STUDY

By - Yesh Thakur and Sabyasachi De (DSC31)

## PROBLEM STATEMENT

With online sales gaining popularity, tech companies are exploring ways to improve their sales by analysing customer behaviour and gaining insights about product trends. Furthermore, the websites make it easier for customers to find the products they require without much scavenging. This is done by tracking your clicks on their website and searching for patterns within them. This kind of data is called a clickstream data.



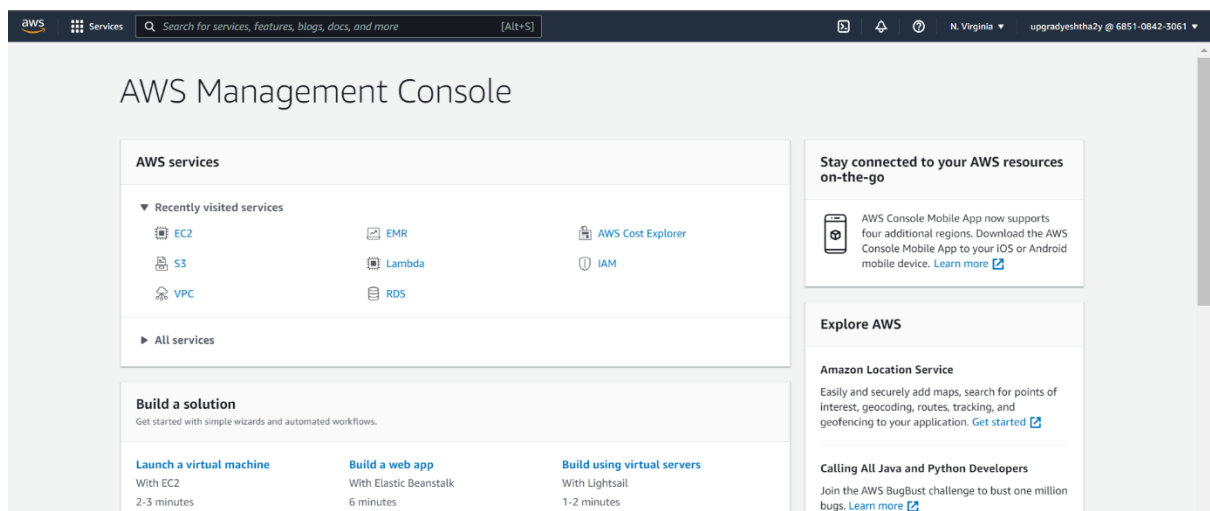
## OBJECTIVE:

To extract data and gather insights from a real-life data set of an e-commerce company.

## CLUSTER CREATION:

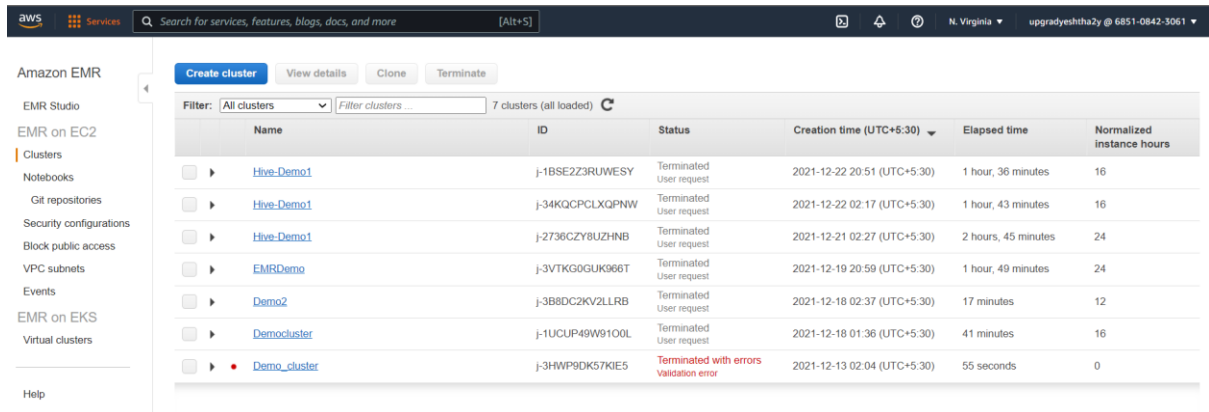
### Step 1

Login to AWS and look for EMR cluster either in the Recently Viewed Services tab or search for it in the Search tab.



## Step 2

Once you are in the EMR home page, click on ‘Create Cluster’ button.

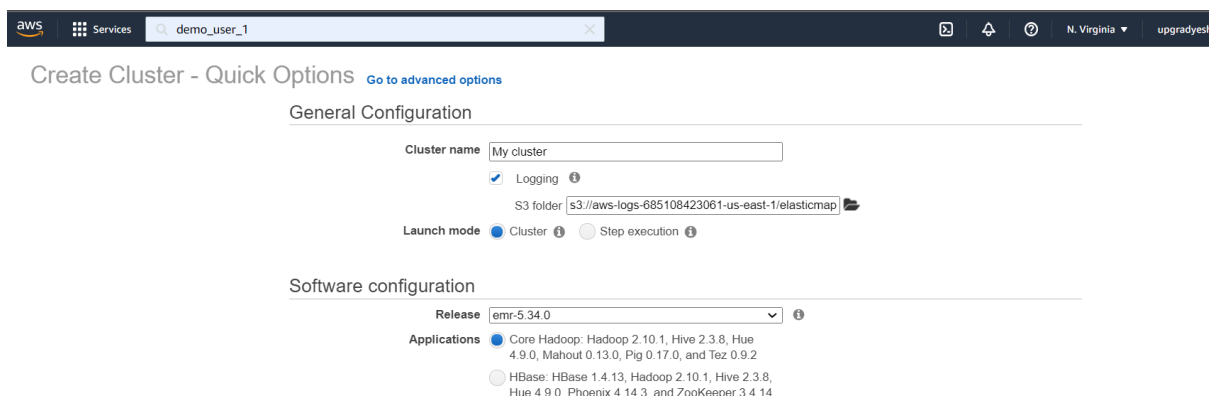


The screenshot shows the Amazon EMR console. On the left is a navigation menu with options like Amazon EMR, EMR Studio, EMR on EC2, Clusters, Notebooks, Git repositories, Security configurations, Block public access, VPC subnets, Events, EMR on EKS, Virtual clusters, and Help. The main area displays a table of clusters. At the top, there are buttons for 'Create cluster', 'View details', 'Clone', and 'Terminate'. Below these is a filter bar showing 'All clusters' and '7 clusters (all loaded)'. The table has columns for Name, ID, Status, Creation time (UTC+5:30), Elapsed time, and Normalized instance hours. The clusters listed are:

Name	ID	Status	Creation time (UTC+5:30)	Elapsed time	Normalized instance hours
Hive-Demo1	j-1BSE2Z3RUWESY	Terminated User request	2021-12-22 20:51 (UTC+5:30)	1 hour, 36 minutes	16
Hive-Demo1	j-34KQCPCLXQPNW	Terminated User request	2021-12-22 02:17 (UTC+5:30)	1 hour, 43 minutes	16
Hive-Demo1	j-2736CZ8UZHNB	Terminated User request	2021-12-21 02:27 (UTC+5:30)	2 hours, 45 minutes	24
EMRDemo	j-3VTKG0GUK966T	Terminated User request	2021-12-19 20:59 (UTC+5:30)	1 hour, 49 minutes	24
Demo2	j-3B8DC2KV2LLRB	Terminated User request	2021-12-18 02:37 (UTC+5:30)	17 minutes	12
Democluster	j-1UCUP49W91OOL	Terminated User request	2021-12-18 01:36 (UTC+5:30)	41 minutes	16
Demo_cluster	j-3HWP9DK57KIE5	Terminated with errors Validation error	2021-12-13 02:04 (UTC+5:30)	55 seconds	0

## Step 3

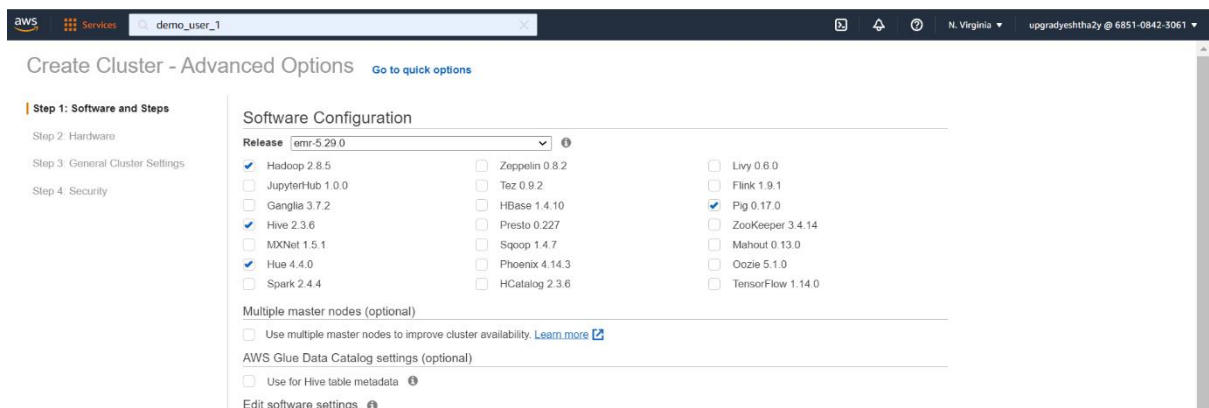
Click on ‘Go to Advanced Options’.



The screenshot shows the 'Create Cluster - Quick Options' page. It has a search bar with 'demo\_user\_1' and a 'Go to advanced options' link. The 'General Configuration' section includes a 'Cluster name' field with 'My cluster', a checked 'Logging' checkbox, and an 'S3 folder' field with 's3://aws-logs-685108423061-us-east-1/elasticmap'. The 'Launch mode' section has 'Cluster' selected. The 'Software configuration' section shows a 'Release' dropdown set to 'emr-5.34.0' and 'Applications' with 'Core Hadoop' selected. Below the applications, it lists the software stack: 'Core Hadoop: Hadoop 2.10.1, Hive 2.3.8, Hue 4.9.0, Mahout 0.13.0, Pig 0.17.0, and Tez 0.9.2'. Other options include 'HBase: HBase 1.4.13, Hadoop 2.10.1, Hive 2.3.8, Hue 4.9.0, Phoenix 4.14.3, and ZooKeeper 3.4.14'.

## Step 4

From the Software Configuration, select the ones which are required along with the release of the EMR and then click next. For this case study, we have chosen emr-5.29.0 as per the instructions.



The screenshot shows the 'Create Cluster - Advanced Options' page. It has a search bar with 'demo\_user\_1' and a 'Go to quick options' link. The 'Software Configuration' section shows a 'Release' dropdown set to 'emr-5.29.0'. Below this, there are three columns of checkboxes for various software packages. The first column has 'Hadoop 2.8.5' checked. The second column has 'Zeppelin 0.8.2', 'Tez 0.9.2', 'HBase 1.4.10', 'Presto 0.227', 'Sqoop 1.4.7', 'Phoenix 4.14.3', and 'HCatalog 2.3.6'. The third column has 'Livvy 0.6.0', 'Flink 1.9.1', 'Pig 0.17.0' checked, 'ZooKeeper 3.4.14', 'Mahout 0.13.0', 'Oozie 5.1.0', and 'TensorFlow 1.14.0'. Below these are sections for 'Multiple master nodes (optional)', 'AWS Glue Data Catalog settings (optional)', and 'Edit software settings'.

## Step 5

On the Hardware Configuration page, we have to specify the required configuration i.e., instance type and instance count for the master and core nodes.

As per the instructions, we have selected m4.large for both the master and core node as it is a 2 node cluster.

Click next.

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	m4.large 2 vCore, 8 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Core Core - 2	m4.large 2 vCore, 8 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	1 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 64 GiB Add configuration settings	0 Instances	<input checked="" type="radio"/> On-demand <input type="radio"/> Spot Use on-demand as max price

## Step 6

Here, give a suitable name to your cluster.

General Options

Cluster name: Hive-Case-Study

☒ Logging  
S3 folder: s3://aws-logs-685108423061-us-east-1/elasticmapreduce

☒ Debugging

☒ Termination protection

Tags

Key	Value (optional)
Name	test
Add a key to create a tag	

## Step 7

Select an EC2 key pair (created before the cluster creation) and click on 'Create Cluster'.

**Create Cluster - Advanced Options** [Go to quick options](#)

Step 1: Software and Steps  
Step 2: Hardware  
Step 3: General Cluster Settings  
**Step 4: Security**

**Security Options**

EC2 key pair: **d191221**

☒ Cluster visible to all IAM users in account

Permissions: **Default** (Selected) | 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

[Cancel](#) [Previous](#) [Create cluster](#)

## Step 8

Wait for your cluster to start and then we can go to putty.

**Amazon EMR**

Cluster: **Hive-Case-Study** **Starting**

[Clone](#) [Terminate](#) [AWS CLI export](#) ⚠ Auto-termination is not available for this account when using this release of EMR.

**Summary** | Application user interfaces | Monitoring | Hardware | Configurations | Events | Steps | Bootstrap actions

**Summary**

ID: j-ZBX8CS03LSA  
Creation date: 2021-12-24 03:04 (UTC+5:30)  
Elapsed time: 0 seconds  
After last step completes: Cluster waits  
Termination protection: On [Change](#)  
Tags: Name = test [View All / Edit](#)  
Master public DNS: --

**Configuration details**

Release label: emr-5.29.0  
Hadoop distribution: Amazon 2.8.5  
Applications: Hive 2.3.6, Pig 0.17.0, Hue 4.4.0  
Log URI: s3://aws-logs-685108423061-us-east-1/elasticmapreduce/  
EMRFS consistent view: Disabled  
Custom AMI ID: --

**Application user interfaces**

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

**Network and hardware**

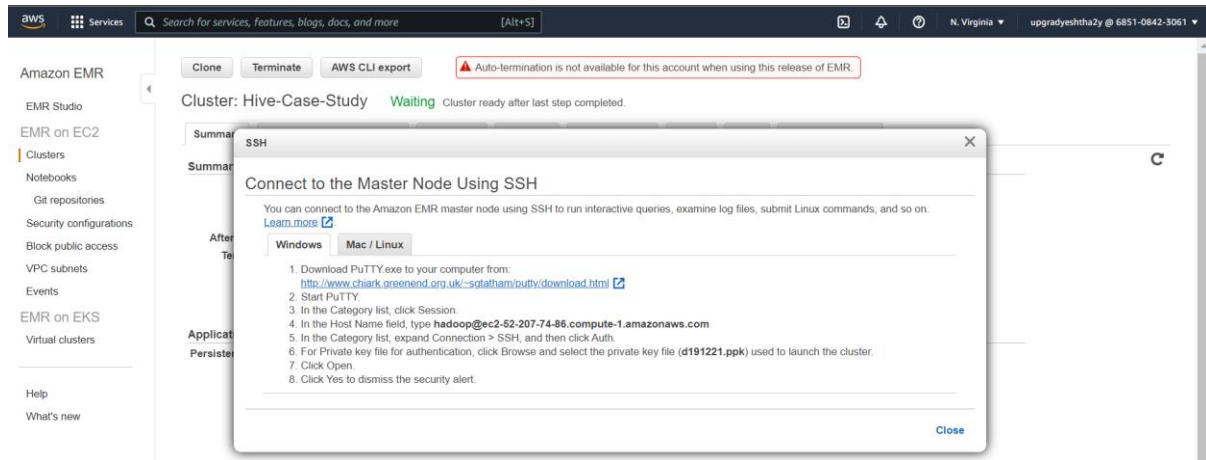
Availability zone: --  
Subnet ID: [subnet-1278ae23](#)  
Master: **Provisioning** 1 m4.large  
Core: **Provisioning** 1 m4.large  
Task: --  
Cluster scaling: Not enabled

**Security and access**

Key name: d191221  
EC2 instance profile: EMR\_EC2\_DefaultRole  
EMR role: EMR\_DefaultRole

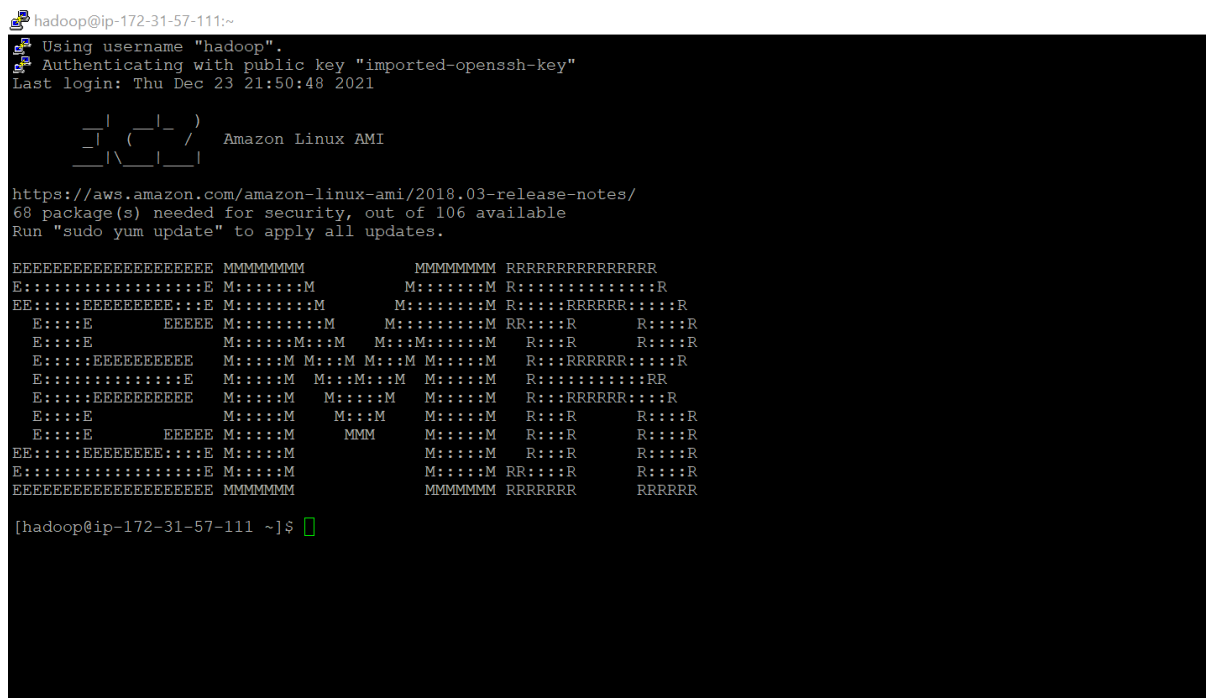
## Step 9

When the cluster is in the Waiting state which means it's running, click on Master public DNS.



## Step 10

- Open Putty and paste the Host Name.
- Click and expand SSH and select AUTH.
- Browse Key pair file and click Open.



## Step 11

Copying the data into HDFS.

1. Creating a directory 'casestudy' in HDFS.  
Command: `hadoop fs -mkdir /user/hive/casestudy`
2. Let's check this directory.  
Command: `hadoop fs -ls /user/hive`

```
[hadoop@ip-172-31-48-166 ~]$ hadoop fs -ls /user/hive
Found 2 items
drwxr-xr-x - hadoop hadoop          0 2021-12-24 20:28 /user/hive/casestudy
drwxrwxrwt - hdfs hadoop            0 2021-12-24 19:34 /user/hive/warehouse
[hadoop@ip-172-31-48-166 ~]$
```

3. Loading the data from S3 bucket to HDFS.

Command:

`hadoop distcp s3://casestudyhive1/2019-Oct.csv /user/hive/casestudy/2019-Oct.csv`

```
[hadoop@ip-172-31-58-247 ~]$ hadoop distcp s3://casestudyhive1/2019-Oct.csv /user/hive/casestudy/2019-Oct.csv
21/12/29 20:44:30 INFO tools.DistCp: Input Options: DistCpOptions(atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=false, skipCRC=false, blocking=true, numListStatusThreads=0, maxMaps=20, mapBandwidth=100, sslConfigurationFile=null, copyStrategy='uniformsize', preserveStatus=[], preserveRawAttrs=false, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3://casestudyhive1/2019-Oct.csv], targetPath=/user/hive/casestudy/2019-Oct.csv, targetPathExists=false, filtersFile='null')
21/12/29 20:44:31 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-58-247.ec2.internal/172.31.58.247:8032
21/12/29 20:44:37 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
21/12/29 20:44:37 INFO tools.SimpleCopyListing: Build file listing completed.
21/12/29 20:44:37 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
21/12/29 20:44:37 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
21/12/29 20:44:37 INFO tools.DistCp: Number of paths in the copy list: 1
21/12/29 20:44:37 INFO tools.DistCp: Number of paths in the copy list: 1
21/12/29 20:44:37 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-58-247.ec2.internal/172.31.58.247:8032
21/12/29 20:44:37 INFO mapreduce.JobSubmitter: number of splits:1
21/12/29 20:44:38 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1640810275048_0001
21/12/29 20:44:38 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1640810275048_0001
```

```
GC time elapsed (ms)=360
CPU time spent (ms)=18770
Physical memory (bytes) snapshot=584773632
Virtual memory (bytes) snapshot=3310522368
Total committed heap usage (bytes)=503316480
File Input Format Counters
  Bytes Read=217
File Output Format Counters
  Bytes Written=0
DistCp Counters
  Bytes Copied=482542278
  Bytes Expected=482542278
  Files Copied=1
```

`hadoop distcp s3://casestudyhive1/2019-Nov.csv /user/hive/casestudy/2019-Nov.csv`

```
[hadoop@ip-172-31-58-247 ~]$ hadoop distcp s3://casestudyhive1/2019-Nov.csv /user/hive/casestudy/2019-Nov.csv
21/12/29 20:48:23 INFO tools.DistCp: Input Options: DistCpOptions(atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=false, skipCRC=false, blocking=true, numListStatusThreads=0, maxMaps=20, mapBandwidth=100, sslConfigurationFile=null, copyStrategy='uniformsize', preserveStatus=[], preserveRawAttrs=false, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3://casestudyhive1/2019-Nov.csv], targetPath=/user/hive/casestudy/2019-Nov.csv, targetPathExists=false, filtersFile='null')
21/12/29 20:48:23 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-58-247.ec2.internal/172.31.58.247:8032
21/12/29 20:48:29 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
21/12/29 20:48:29 INFO tools.SimpleCopyListing: Build file listing completed.
21/12/29 20:48:29 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
21/12/29 20:48:29 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
21/12/29 20:48:29 INFO tools.DistCp: Number of paths in the copy list: 1
21/12/29 20:48:29 INFO tools.DistCp: Number of paths in the copy list: 1
21/12/29 20:48:29 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-58-247.ec2.internal/172.31.58.247:8032
21/12/29 20:48:30 INFO mapreduce.JobSubmitter: number of splits:1
```

```

Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=299
CPU time spent (ms)=20170
Physical memory (bytes) snapshot=594878464
Virtual memory (bytes) snapshot=3308191744
Total committed heap usage (bytes)=506462208

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

```

4. Now, checking the files in the directory.

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

```

[hadoop@ip-172-31-48-166 ~]$ hadoop fs -ls /user/hive/casestudy/
Found 2 items
-rw-r--r--  1 hadoop hadoop  482542278  2021-12-24  20:34  /user/hive/casestudy/2019_Nov.csv
-rw-r--r--  1 hadoop hadoop  482542278  2021-12-24  20:33  /user/hive/casestudy/2019_Oct.csv
[hadoop@ip-172-31-48-166 ~]$

```

## Step 12

Data sets are loaded now let's launch Hive.

### Creating a new database 'hive\_casestudy'

Command:

create database if not exists hive\_casestudy ;

show databases;

use hive\_casestudy;

```

[hadoop@ip-172-31-58-247 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.p
hive> create database if not exists hive_casestudy ;
OK
Time taken: 1.095 seconds
hive> show databases;
OK
default
hive_casestudy
Time taken: 0.214 seconds, Fetched: 2 row(s)
hive> use hive_casestudy;
OK
Time taken: 0.122 seconds

```

## Creating a table.

create external table if not exists clickstream (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/casestudy/' tblproperties ("skip.header.line.count"="1");

```
hive> create external table if not exists clickstream (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/casestudy/' tblproperties ("skip.header.line.count"="1");
OK
Time taken: 0.367 seconds
```

Now, we need to optimize the table 'clickstream' through partitioning and bucketing for faster query results.

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

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

```
set hive.enforce.bucketing = true;
```

```
Time taken: 0.381 seconds
hive> set hive.exec.dynamic.partition=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.enforce.bucketing = true;
hive> █
```

## Creating a new table with dynamic partitions and buckets and then inserting the data.

create table if not exists dyn\_part\_buck\_clickstream (event\_time string, 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 13 buckets row format delimited fields terminated by ',' lines terminated by '\n' stored as textfile;

```
hive> create table if not exists dyn_part_buck_clickstream (event_time string, 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 13 buckets row format delimited fields terminated by ',' lines terminated by '\n' stored as textfile;
OK
Time taken: 0.115 seconds
```

insert into table dyn\_part\_buck\_clickstream partition (event\_type) select event\_time, product\_id, category\_id, category\_code, brand, price, user\_id, user\_session, event\_type from clickstream;

```
hive> insert into table dyn_part_buck_clickstream partition (event_type) select event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type from clickstream;
Query ID = hadoop_20211230195527_667535ca-bb1d-4589-96ad-1da7720aeb02
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_1640892795316_0004)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    2         2         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    5         5         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 169.28 s
Loading data to table hive_casestudy.dyn_part_buck_clickstream partition (event_type=null)
Loaded : 4/4 partitions.
Time taken to load dynamic partitions: 0.94 seconds
Time taken for adding to write entity : 0.003 seconds
OK
Time taken: 186.607 seconds
```



### Checking both the tables:

describe clickstream;

describe dyn\_part\_buck\_clickstream;

```
hive> describe clickstream;
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: 0.047 seconds, Fetched: 9 row(s)
hive> describe dyn_part_buck_clickstream;
OK
event_time          string
product_id          string
category_id         string
category_code       string
brand                string
price                float
user_id              bigint
user_session        string
event_type           string

# Partition Information
# col_name           data_type           comment
event_type           string
Time taken: 0.133 seconds, Fetched: 14 row(s)
```

### Getting the headers back.

set hive.cli.print.header=true;

We have created two tables:

1<sup>st</sup> table which contains data for both October and November.

2<sup>nd</sup> table which has the same data but optimized with partitions and buckets.

## Step 13

### Query Analysis

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

Unoptimized Query:

SELECT SUM(price) as total\_rev FROM clickstream WHERE MONTH(event\_time) = 10 and event\_type = 'purchase';

```
hive> SELECT SUM(price) as total_rev FROM clickstream WHERE MONTH(event_time) = 10 and event_type = 'purchase';
Query ID = hadoop_20211229214117_0b1ba86c-3f37-4863-be5e-0bc94e251182
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640810275048_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: 65.03 s
-----
OK
total_rev
1211538.4299997438
Time taken: 65.675 seconds, Fetched: 1 row(s)
```

Optimized Query using dyn\_part\_buck\_clickstream:

SELECT SUM(price) as total\_rev FROM dyn\_part\_buck\_clickstream WHERE MONTH(event\_time) = 10 and event\_type = 'purchase';

```
hive> SELECT SUM(price) as total_rev FROM dyn_part_buck_clickstream WHERE MONTH(event_time) = 10 and event_type = 'purchase';
Query ID = hadoop_20211230231835_29779f34-5d56-4b25-8de4-41f6ce79c843
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640905004182_0003)

-----
      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: 15.13 s
-----
OK
total_rev
1211538.4295325726
Time taken: 15.879 seconds, Fetched: 1 row(s)
```

The total revenue generated is 1211538.4295325726. Unoptimized query took 65.675 secs while optimized query took 15.879 secs.

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

Unoptimized Query:

```
SELECT MONTH(event_time), COUNT(event_type) AS pur_cnt from clickstream WHERE  
event_type = 'purchase' GROUP BY MONTH(event_time);
```

```
hive> SELECT MONTH(event_time), COUNT(event_type) AS pur_cnt from clickstream WHERE event_type = 'purchase' GROUP BY MONTH(event_time);  
Query ID = hadoop_20211229214456_b24ed3ca-3716-456d-ba31-818c9efbale2  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1640810275048_0006)  
  
-----  
VERTICES    MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED  
-----  
Map 1 ..... container  SUCCEEDED  2      2          0        0        0      0  
Reducer 2 ..... container  SUCCEEDED  3      3          0        0        0      0  
-----  
VERTICES: 02/02 [=====]>>] 100% ELAPSED TIME: 60.79 s  
-----  
OK  
_c0      pur_cnt  
_l0      245624  
_l1      322417  
Time taken: 61.451 seconds, Fetched: 2 row(s)
```

Optimized Query using dyn\_part\_buck\_clickstream:

```
SELECT MONTH(event_time), COUNT(event_type) AS pur_cnt from dyn_part_buck_clickstream  
WHERE event_type = 'purchase' GROUP BY MONTH(event_time);
```

```
hive> SELECT MONTH(event_time), COUNT(event_type) AS pur_cnt from dyn_part_buck_clickstream WHERE event_type = 'purchase' GROUP BY MONTH(event_time);  
Query ID = hadoop_20211231181315_749c5841-8f64-4520-82ec-ef590b8de71a  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1640973564708_0003)  
  
-----  
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: 16.96 s  
-----  
OK  
_c0      pur_cnt  
_l0      245624  
_l1      322417  
Time taken: 18.047 seconds, Fetched: 2 row(s)
```

The total sum of the purchase for October is 245624 and for November it is 322417. Unoptimized query took 61.451 secs while optimized query took 18.047 secs.

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

Unoptimized Query:

```
WITH diff_revenue AS (SELECT SUM(CASE WHEN MONTH(event_time) = 10 THEN price ELSE 0 END) AS oct_pur, SUM(CASE WHEN MONTH(event_time) = 11 THEN price ELSE 0 END) AS nov_pur FROM clickstream WHERE event_type = 'purchase') SELECT (nov_pur - oct_pur) AS difference_revenue FROM diff_revenue;
```

```
hive> WITH diff_revenue AS (SELECT SUM(CASE WHEN MONTH(event_time) = 10 THEN price ELSE 0 END) AS oct_pur, SUM(CASE WHEN MONTH(event_time) = 11 THEN price ELSE 0 END) AS nov_pur FROM clickstream WHERE event_type = 'purchase') SELECT (nov_pur - oct_pur) AS difference_revenue FROM diff_revenue;
Query ID = hadoop_20211229213710_8d21ecff-8c55-479a-b196-dde72d32ca14
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_1640810275049_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: 56.08 s
-----
OK
difference_revenue
319478.4700003781
Time taken: 65.493 seconds, Fetched: 1 row(s)
```

Optimized Query:

```
WITH diff_revenue AS (SELECT SUM(CASE WHEN MONTH(event_time) = 10 THEN price ELSE 0 END) AS oct_pur, SUM(CASE WHEN MONTH(event_time) = 11 THEN price ELSE 0 END) AS nov_pur FROM dyn_part_buck_clickstream WHERE event_type = 'purchase') SELECT (nov_pur - oct_pur) AS difference_revenue FROM diff_revenue;
```

```
hive> WITH diff_revenue AS (SELECT SUM(CASE WHEN MONTH(event_time) = 10 THEN price ELSE 0 END) AS oct_pur, SUM(CASE WHEN MONTH(event_time) = 11 THEN price ELSE 0 END) AS nov_pur FROM dyn_part_buck_clickstream WHERE event_type = 'purchase') SELECT (nov_pur - oct_pur) AS difference_revenue FROM diff_revenue;
Query ID = hadoop_20211231181909_65b2c46c-8246-46a5-b5ce-e71eb74329f1
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640973564708_0003)

-----
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: 18.73 s
-----
OK
difference_revenue
319478.469592195
Time taken: 19.568 seconds, Fetched: 1 row(s)
```

The change in revenue is 319478.469592195. Unoptimized query took 65.493 secs while optimized query took 19.560 secs.

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

Unoptimized Query:

```
SELECT DISTINCT(category_code) FROM clickstream where category_code !='';
```

```
hive> SELECT DISTINCT(category_code) FROM clickstream where category_code !='';
Query ID = hadoop_20211229220013_0fb8e7f1-cfb9-43f0-95d9-83db0652deaf
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640810275048_0007)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1 .....	container	SUCCEEDED	2	2	0	0	0	0	0
Reducer 2 .....	container	SUCCEEDED	5	5	0	0	0	0	0

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 58.03 s
```

```
OK
category_code
accessories.cosmetic_bag
stationery.cartridge
accessories.bag
appliances.environment.vacuum
furniture.living_room.chair
sport.diving
appliances.personal.hair_cutter
appliances.environment.air_conditioner
apparel.glove
furniture.bathroom.bath
furniture.living_room.cabinet
Time taken: 58.732 seconds, Fetched: 11 row(s)
```

Optimized Query:

```
SELECT DISTINCT(category_code) FROM dyn_part_buck_clickstream where category_code !='';
```

```
hive> SELECT DISTINCT(category_code) FROM dyn_part_buck_clickstream where category_code !='';
Query ID = hadoop_20211230201428_39b977e8-2c7d-4f51-bc92-fd15ebe79d47
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640892795316_0005)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1 .....	container	SUCCEEDED	7	7	0	0	0	0	0
Reducer 2 .....	container	SUCCEEDED	4	4	0	0	0	0	0

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 29.47 s
```

```
OK
category_code
accessories.bag
appliances.environment.vacuum
appliances.personal.hair_cutter
sport.diving
apparel.glove
furniture.bathroom.bath
furniture.living_room.cabinet
stationery.cartridge
accessories.cosmetic_bag
appliances.environment.air_conditioner
furniture.living_room.chair
Time taken: 30.289 seconds, Fetched: 11 row(s)
```

There are 11 distinct categories in total. Unoptimized query took 58.732 secs while optimized query took 30.289 secs.

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

Unoptimized Query:

```
select count(product_id) as prod_id, category_code from clickstream where category_code !='' group by category_code;
```

```
hive> select count(product_id) as prod_id, category_code from clickstream where category_code !='' group by category_code;
Query ID = hadoop_20211230203546_7f6e9328-094e-45a6-b5f9-790eb78a9fb4
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_1640892795316_0006)

-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container    SUCCEEDED    2         2         0         0         0         0
Reducer 2 ..... container    SUCCEEDED    5         5         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 60.36 s
-----
OK
prod_id category_code
1248 accessories.cosmetic_bag
26722 stationery.cartridge
11681 accessories.bag
59761 appliances.environment.vacuum
308 furniture.living_room.chair
2 sport.diving
1643 appliances.personal.hair_cutter
332 appliances.environment.air_conditioner
18232 apparel.glove
9857 furniture.bathroom.bath
13439 furniture.living_room.cabinet
Time taken: 71.505 seconds, Fetched: 11 row(s)
```

Optimized Query:

```
select count(product_id) as prod_id, category_code from dyn_part_buck_clickstream where category_code !='' group by category_code;
```

```
hive> select count(product_id) as prod_id, category_code from dyn_part_buck_clickstream where category_code !='' group by category_code;
Query ID = hadoop_20211230204322_17600065-97f4-42ab-ba24-4f98dbbffd57
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640892795316_0006)

-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container    SUCCEEDED    7         7         0         0         0         0
Reducer 2 ..... container    SUCCEEDED    4         4         0         0         0         0
-----
VERTICES: 02/02 [=====>>>] 100% ELAPSED TIME: 29.58 s
-----
OK
prod_id category_code
11681 accessories.bag
59761 appliances.environment.vacuum
1643 appliances.personal.hair_cutter
2 sport.diving
18232 apparel.glove
9857 furniture.bathroom.bath
13439 furniture.living_room.cabinet
26722 stationery.cartridge
1248 accessories.cosmetic_bag
332 appliances.environment.air_conditioner
308 furniture.living_room.chair
Time taken: 30.228 seconds, Fetched: 11 row(s)
```

Unoptimized query took 71.505 secs while optimized query took 30.228 secs.

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

Unoptimized Query:

select brand, sum(price) as total\_sales from clickstream where event\_type = 'purchase' and brand !='' group by brand order by total\_sales desc limit 1;

```
hive> select brand, sum(price) as total_sales from clickstream where event_type = 'purchase' and brand !='' group by brand order by total_sales desc limit 1;
Query ID = hadoop_20211230210315_ddf68b29-051d-4a18-bf1c-8d941c79df66
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_1640892795316_0007)

-----
VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    2         2         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    3         3         0         0         0         0
Reducer 3 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 03/03  [=====] 100% ELAPSED TIME: 63.24 s
-----
OK
brand    total_sales
runail  148297.9400000003
Time taken: 73.686 seconds, Fetched: 1 row(s)
```

Optimized Query:

select brand, sum(price) as total\_sales from dyn\_part\_buck\_clickstream where event\_type = 'purchase' and brand !='' group by brand order by total\_sales desc limit 1;

```
hive> select brand, sum(price) as total_sales from dyn_part_buck_clickstream where event_type = 'purchase' and brand !='' group by brand order by total_sales desc limit 1;
Query ID = hadoop_20211230210644_5f55f170-8e2f-4381-872a-eb377f73a49e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640892795316_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: 20.18 s
-----
OK
brand    total_sales
runail  148297.93996394053
Time taken: 21.105 seconds, Fetched: 1 row(s)
```

Top brand is runail with total sales as 148297.93996394053. Unoptimized query took 73.686 secs while optimized query took 21.105 secs.

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

Unoptimized Query:

WITH brand\_sales AS (SELECT brand, sum(CASE WHEN month(event\_time)=10 THEN price else 0 END) AS oct\_sales, sum(CASE WHEN month(event\_time)=11 THEN price else 0 END) AS nov\_sales from clickstream where event\_type = 'purchase' group by brand) select brand from brand\_sales where (nov\_sales - oct\_sales)>0;

```
hive> WITH brand_sales AS (SELECT brand, sum(CASE WHEN month(event_time)=10 THEN price else 0 END) AS oct_sales, sum(CASE WHEN month(event_time)=11 THEN price else 0 END) AS nov_sales from clickstream where event_type = 'purchase' group by brand) select brand from brand_sales where (nov_sales - oct_sales)>0;
Query ID = hadoop_20211230213144_47b9fc60-c5f3-4fea-9573-5bd6fe761b8f
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_1640892795316_0008)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	.....	container	SUCCEEDED	2	2	0	0	0	0
Reducer 2	.....	container	SUCCEEDED	3	3	0	0	0	0

```
VERTICES: 02/02 [=====] 100% ELAPSED TIME: 69.79 s
OK
brand
airnails
artex
binacil
biosqua
blixz
bluesky
bpw.style
carmex
chi
concept
cosima
cosmoprofi
deoproce
depilflax
dewal
dizao
egomania
elizavecca
ellips
finish
freshbubble
grattol
haruyama
helloqanic
```

```
candy
coffin
cristalinas
cutrin
domix
eocoraft
elekin
enjoy
entity
eos
estel
estelare
farmavita
fedua
foamie
glysolid
godefroy
inn
irisk
kamill
karex
kaypro
keen
kinetics
koelcia
lianail
lowence
matreshka
mavala
missha
moyou
nagaraku
profepil
rasyan
refectocil
skinity
smart
solomeya
swarovski
trind
uno
ya-r
Time taken: 80.648 seconds, Fetched: 161 row(s)
hive> █
```



## Optimized Query:

WITH brand\_sales AS (SELECT brand, sum(CASE WHEN month(event\_time)=10 THEN price else 0 END) AS oct\_sales, sum(CASE WHEN month(event\_time)=11 THEN price else 0 END) AS nov\_sales from dyn\_part\_buck\_clickstream where event\_type = 'purchase' group by brand) select brand from brand\_sales where (nov\_sales - oct\_sales)>0;

```
hive> WITH brand_sales AS (SELECT brand, sum(CASE WHEN month(event_time)=10 THEN price else 0 END) AS oct_sales, sum(CASE WHEN month(event_time)=11 THEN price else 0 END) AS nov_sales from dyn_part_buck_clickstream where event_type = 'purchase' group by brand) select brand from brand_sales where (nov_sales - oct_sales)>0;
Query ID = hadoop_20211230213810_f4a13410-789b-455a-8d5c-c9f28f3e408f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640892795316_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
-----
VERTICES: 02/02  [=====] 100% ELAPSED TIME: 19.02 s
-----
OK
Brand
airnails
art-visage
artex
aura
balbcare
barbie
batisse
beautix
beauty-free
beautyblender
beauugreen
benovy
binacil
biosqua
biore
blixz
bluesky
bodyton
bpw.style
browxenna
candy
carmex
chi
coffin
concept
```

```
nirvel
nitrile
oniq
orly
osmo
ovale
plazan
polaris
profepil
profhenna
protokeratin
provoc
rasyan
reflectocil
roci
roubloff
runail
s.care
sanoto
severina
shary
shik
skinlity
skinlite
smart
soleo
solomeya
sophin
staleks
strong
supertan
swarovski
tertio
treaclemoon
trind
uno
uakuei
veraclara
vilenta
yoko
yu-r
zeitun
Time taken: 19.634 seconds, Fetched: 161 row(s)
hive>
```

There are a total of 161 brands that have increased sales from October to November. Unoptimized query took 80.648 secs while optimized query took 19.634 secs.

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

Unoptimized Query:

WITH spending\_summary AS (select user\_id, sum(price) as total\_spending from clickstream where event\_type = 'purchase' group by user\_id order by total\_spending desc) select user\_id from spending\_summary limit 10;

```
hive> WITH spending_summary AS (select user_id, sum(price) as total_spending from clickstream where event_type = 'purchase' group by user_id order by total_spending desc) select user_id from spending_summary limit 10;
Query ID = hadoop_20211230215428_35c8adc7-978c-444c-a2c4-0f8b10cc71d2
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_1640892795316_0009)

-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    2         2         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    3         3         0         0         0         0
Reducer 3 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 03/03 [=====] 100% ELAPSED TIME: 62.65 s
-----
OK
user_id
557790271
150318419
562167663
531900924
557850743
522130011
561592095
431950134
566576008
521347209
Time taken: 72.797 seconds, Fetched: 10 row(s)
```

Optimized Query:

WITH spending\_summary AS (select user\_id, sum(price) as total\_spending from dyn\_part\_buck\_clickstream where event\_type = 'purchase' group by user\_id order by total\_spending desc) select user\_id from spending\_summary limit 10;

```
hive> WITH spending_summary AS (select user_id, sum(price) as total_spending from dyn_part_buck_clickstream where event_type = 'purchase' group by user_id order by total_spending desc) select user_id from spending_summary limit 10;
Query ID = hadoop_20211230220036_6406003b-2b0b-4cb4-a0d9-2cb2bfa61b72
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1640892795316_0009)

-----
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: 17.05 s
-----
OK
user_id
557790271
150318419
562167663
531900924
557850743
522130011
561592095
431950134
566576008
521347209
Time taken: 18.038 seconds, Fetched: 10 row(s)
hive>
```

Above are the top 10 customers who spend the most who should be awarded with Golden customer plan. Unoptimized query took 72.797 secs while optimized query took 18.038 secs.

## Dropping database:

drop database hive\_casestudy cascade;

```
hive> drop database hive_casestudy cascade;
OK
Time taken: 0.533 seconds
hive> show databases;
OK
database_name
default
Time taken: 0.012 seconds, Fetched: 1 row(s)
hive>
```

## Terminating the cluster:

The screenshot shows the AWS Management Console for an Amazon EMR cluster named "Hive-Case-Study". The cluster is in a "Waiting" state, indicating it is ready after the last step completed. The console displays various tabs for cluster management, including Summary, Application user interfaces, Monitoring, Hardware, Configurations, Events, Steps, and Bootstrap actions. The Summary tab is active, showing details such as the cluster ID (j-3H1SRC05CPX6P), creation date (2021-12-31 00:55 UTC+5:30), elapsed time (2 hours, 48 minutes), and configuration details like Release label (emr-5.29.0), Hadoop distribution (Amazon 2.8.5), and Applications (Hive 2.3.6, Pig 0.17.0, Hue 4.4.0). The console also shows the Master public DNS, Tags, and Application user interfaces.

The screenshot shows the same AWS Management Console page, but with a "Terminate cluster" dialog box open. The dialog box asks, "Are you sure you want to terminate this cluster?" and includes a warning: "Any pending work or data residing on the cluster will be lost, such as data stored in HDFS. This action is irreversible." The dialog box has "Cancel" and "Terminate" buttons.