

HIVE CASE STUDY

- DS C37 DA TRACK
- Crysl Lobo & Justin Benedict Dias

Launching of an EMR Cluster

Before creating an EMR cluster, we need to create a key-pair. Since the EMR cluster will be running on EC2 instances, we will require a key-pair to connect with the instance.

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

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 new tag](#)

You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

✓ Successfully created key pair

Key pairs (3) Info							Create key pair
<input type="text" value="Search"/>							1
<input type="checkbox"/>	Name	Type	Created	Fingerprint	ID		
<input type="checkbox"/>	Test-24/06	rsa	2022/06/24 22:04 GMT+5:30	a6:f3:aa:53:a2:38:3c:fd:67:13:bc:c8:56:b...	key-0a815ef4152a06f52		
<input type="checkbox"/>	Test-2606	rsa	2022/06/26 14:36 GMT+5:30	13:55:bca4:f3:56:9f:cc:3f:3e:a1:7c:61:4...	key-000399c3ec6f9560b		
<input type="checkbox"/>	hive_casestudy	rsa	2022/06/29 19:05 GMT+5:30	6d:ab:5ere5:c5:a8:f4:2f:e0:b9:a0:77:69:c...	key-09c18ea1e884de8e4		

Next, we create a S3 Bucket to store our datasets.

Buckets (2) Info

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

	Name	AWS Region	Access	Creation date
<input type="radio"/>	hive-case-study-ecom	US East (N. Virginia) us-east-1	Objects can be public	June 29, 2022, 19:18:56 (UTC+05:30)
<input type="radio"/>	test-demo-upgrad	US East (N. Virginia) us-east-1	Objects can be public	June 24, 2022, 20:14:12 (UTC+05:30)

hive-case-study-ecom Info

Objects

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](#)

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	2019-Nov.csv	csv	June 30, 2022, 12:08:17 (UTC+05:30)	520.6 MB	Standard
<input type="checkbox"/>	2019-Oct.csv	csv	June 30, 2022, 12:32:37 (UTC+05:30)	460.2 MB	Standard

After creation of Key-pair and S3 bucket, we now create the EMR cluster.
EMR – Create cluster – Advanced Options – Release emr-5.29.0

Cluster Nodes and Instances

Choose the instance type, number of instances, and a purchasing option. [Learn more about instance purchasing options](#)

Console options for automatic scaling have changed. [Learn more](#)

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

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: hive_casestudy

☒ 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

Cancel Previous **Create cluster**

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia

Amazon EMR

- EMR Studio
- EMR Serverless New
- EMR on EC2
 - Clusters
 - Notebooks
 - Git repositories
 - Security configurations
 - Block public access
 - VPC subnets
 - Events
- EMR on EKS
 - Virtual clusters
- Help
- What's new

Cluster: Hive-Case-Study Waiting Cluster ready after last step completed.

Summary

ID: j-10PFQX74XILQ6

Creation date: 2022-06-30 14:19 (UTC+5:30)

Elapsed time: 14 minutes

After last step completes: Cluster waits

Termination protection: On [Change](#)

Tags: -- [View All / Edit](#)

Master public DNS: ec2-18-206-195-241.compute-1.amazonaws.com [Connect to the Master Node Using SSH](#)

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://hive-case-study-ecom/ [View Log](#)

EMRFS consistent view: Disabled

Custom AMI ID: --

Application user interfaces

Persistent user interfaces: --

On-cluster user interfaces: Not Enabled [Enable an SSH Connection](#)

Network and hardware

Availability zone: us-east-1b

Subnet ID: [subnet-0aba3cea98526dec4](#)

Master: Running 1 m4.large

Core: Running 1 m4.large

Task: --

Cluster scaling: Not enabled

Security and access

Key name: hive_casestudy

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-0f9c7b02a147a14af](#) [View](#) [ElasticMapReduce](#)

Now we need to add inbound security rule for the Master Node

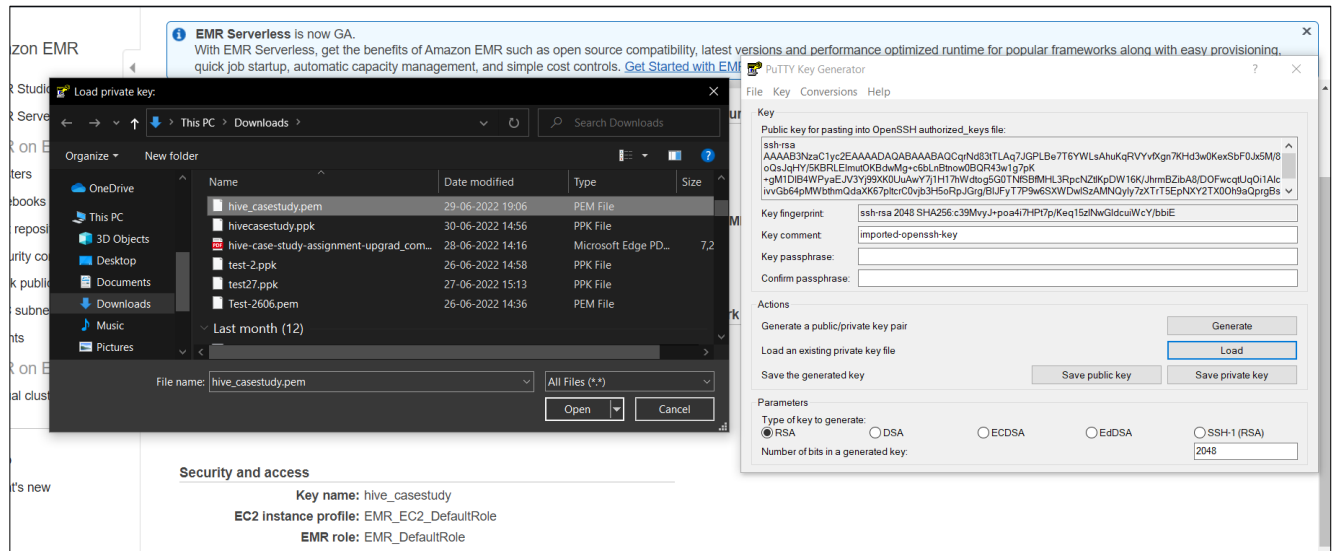
Security Group	Protocol	Port Range	Source	Action
sg-065f565de54eb4da5	Custom TCP	8443	207.171.167.25/32	Delete
sg-0b86c8467ffcbb8b	All UDP	0 - 65535	54.239.98.0/24	Delete
-	SSH	22	0.0.0.0/0	Delete

Add rule

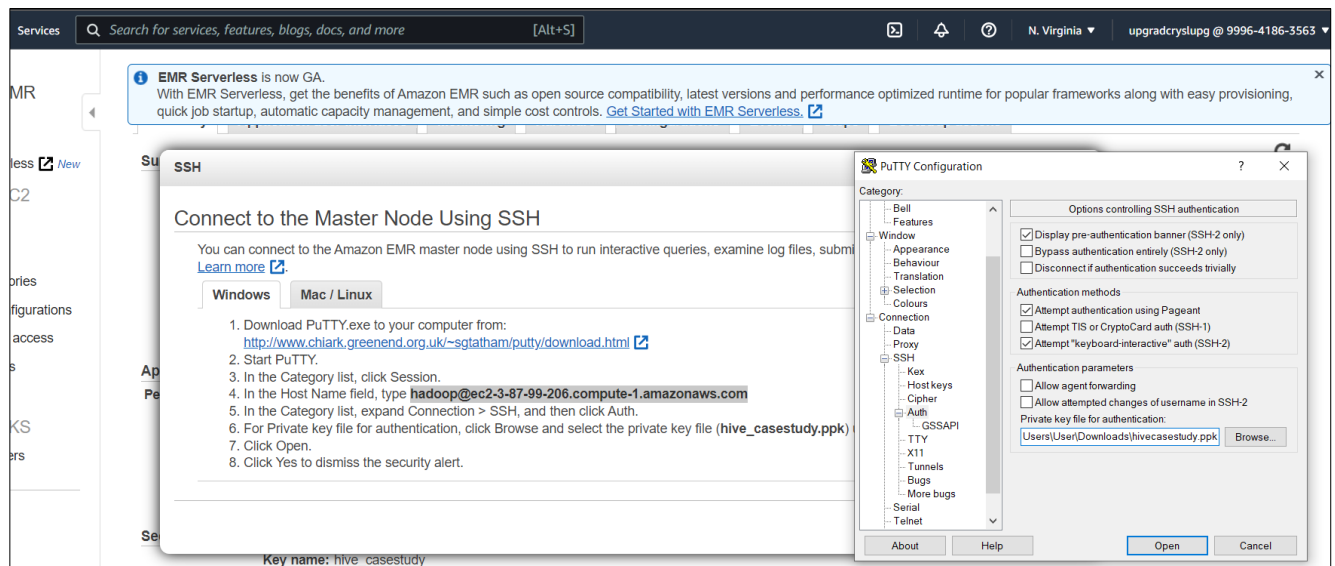
Now that the cluster is created, we are ready to move to the next stage.

Move the data from the S3 bucket into the HDFS

We need to open PuTTYgen application for windows and we have to load the .pem key-pair file and save the private key in the extension .ppk



Now we need to open PuTTY and give the host name and then browse for the private key which we generated above.



```
hadoop@ip-172-31-33-164:~  
  
  _ | _ | _ )  
  _ | ( _ /  Amazon Linux AMI  
  _ | \ _ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
68 package(s) needed for security, out of 97 available  
Run "sudo yum update" to apply all updates.  
  
EEEEEEEEEEEEEEEEEEEEEEEE MMMMMMMM MMMMMMMM RRRRRRRRRRRRRRRR  
E::::::::::::::::::::E M::::::::M M::::::::M R::::::::::::R  
EE::::EEEEEEEE::::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 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::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 MMMMMMMM MMMMMMMM RRRRRRR      RRRRRR  
  
[hadoop@ip-172-31-33-164 ~]$
```

```
[hadoop@ip-172-31-33-164 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x  - hdfs hadoop          0 2022-07-03 09:19 /apps
drwxrwxrwt  - hdfs hadoop          0 2022-07-03 09:22 /tmp
drwxr-xr-x  - hdfs hadoop          0 2022-07-03 09:19 /user
drwxr-xr-x  - hdfs hadoop          0 2022-07-03 09:19 /var
```

```
[hadoop@ip-172-31-33-164 ~]$ hadoop fs -mkdir /hivecasestudy
[hadoop@ip-172-31-33-164 ~]$ hadoop fs -ls /
Found 5 items
drwxr-xr-x   - hdfs    hadoop          0 2022-07-03 09:19 /apps
drwxr-xr-x   - hadoop  hadoop          0 2022-07-03 10:35 /hivecasestudy
drwxrwxrwt   - hdfs    hadoop          0 2022-07-03 09:22 /tmp
drwxr-xr-x   - hdfs    hadoop          0 2022-07-03 09:19 /user
drwxr-xr-x   - hdfs    hadoop          0 2022-07-03 09:19 /var
```

```
[hadoop@ip-172-31-33-164 ~]$ hadoop distcp 's3://hive-case-study-ecom/2019-Oct.csv' /hivecasestudy/2019-Oct.csv
Bytes Written=0
DistCp Counters
  Bytes Copied=482542278
  Bytes Expected=482542278
  Files Copied=1
```

```
Files Copied=1
[hadoop@ip-172-31-33-164 ~]$ hadoop distcp 's3://hive-case-study-ecom/2019-Nov.csv' /hivecasestudy/2019-Nov.csv

DistCp Counters
  Bytes Copied=545839412
  Bytes Expected=545839412
  Files Copied=1
```

Checking whether the data is uploaded in the directory

```
[hadoop@ip-172-31-33-164 ~]$ hadoop fs -ls /hivecasestudy
Found 2 items
-rw-r--r-- 1 hadoop hadoop 545839412 2022-07-03 10:48 /hivecasestudy/2019-Nov.csv
-rw-r--r-- 1 hadoop hadoop 482542278 2022-07-03 10:45 /hivecasestudy/2019-Oct.csv
```

As the data is successfully added, we can launch hive

```
[hadoop@ip-172-31-33-164 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive>
```

Checking if any databases exist

```
hive> show databases;
OK
default
Time taken: 1.005 seconds, Fetched: 1 row(s)
hive>
```

Creating a new database and using the same

```
hive> create database if not exists Ecommerce;
OK
Time taken: 0.551 seconds
hive> show databases;
OK
default
ecommerce
Time taken: 0.021 seconds, Fetched: 2 row(s)
hive> use ecommerce;
OK
Time taken: 0.064 seconds
```

Creating an external table from the raw data

```
hive> create External table if not exists ecom_sales(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' WITH SERDEPROPERTIES ("separatorChar"=",","quot
eChar"="\","escapeChar"="\")stored as textfile Location '/hivecasestudy' TBLPROPERTIES("skip.header.line.
count"="1");
OK
Time taken: 0.34 seconds
```

Describing the table

```
hive> describe ecom_sales;
OK
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
Time taken: 0.042 seconds, Fetched: 9 row(s)
```

Loading data from both files into this table

```
hive> LOAD DATA INPATH '/hivecasestudy/2019-Oct.csv' into table ecom_sales;
Loading data to table ecommerce.ecom_sales
OK
Time taken: 2.301 seconds
hive> LOAD DATA INPATH '/hivecasestudy/2019-Nov.csv' into table ecom_sales;
Loading data to table ecommerce.ecom_sales
OK
Time taken: 0.795 seconds
```

Setting the header

```
hive> set hive.cli.print.header=true;
```

Checking if the data was inserted into the table

```
hive> select * from ecom_sales limit 5;
OK
ecom_sales.event_time  ecom_sales.event_type  ecom_sales.product_id  ecom_sales.category_id  ecom_sales.
category_code  ecom_sales.brand      ecom_sales.price      ecom_sales.user_id      ecom_sales.user_ses
sion
2019-11-01 00:00:02 UTC view      5802432 1487580009286598681      0.32      562076640      09f
afd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart      5844397 1487580006317032337      2.38      553329724      206
7216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC view      5837166 1783999064103190764      pnb      22.22      556138645      57e
d222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart      5876812 1487580010100293687      jessnail      3.16      564506666 1
86c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC remove_from_cart      5826182 1487580007483048900      3.33      553
329724 2067216c-31b5-455d-a1cc-af0575a34ffb
Time taken: 2.364 seconds, Fetched: 5 row(s)
```

QUESTIONS AND ANSWERS USING HIVE QUERY LANGUAGE

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

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

Ans: 1211538.4299 is the total revenue generated due to purchases in October 2019.

The query is performed on the static table and the time taken is 71 seconds.

```
hive> select sum(price) as total_revenue from ecom_sales where month(event_time) =10 and event_type = 'purchase';
Query ID = hadoop_20220704130008_f05413a4-616a-47bb-a316-f76cd9a29052
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0014)
```

	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: 70.58 s
OK
total_revenue
1211538.4299997438
Time taken: 71.423 seconds, Fetched: 1 row(s)
```

We now create a dynamic table with optimization techniques of partitioning and bucketing for quick results.

We name the table dynamic_sales and partition the table on “event_type” and clustered the table on “user_id” into 7 buckets.

```
hive> set hive.exec.dynamic.partition=true;
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> create external table if not exists dynamic_sales(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 7 buckets ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde' stored as textfile;
OK
Time taken: 0.178 seconds
```


Describing the table

```
hive> describe dynamic_sales;
OK
event_time          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
event_type          string

# Partition Information
# col_name          data_type      comment

event_type          string
Time taken: 0.106 seconds, Fetched: 14 row(s)
```

Loading the data in the bucket table

```
hive> insert into dynamic_sales partition (event_type) select event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type from ecom_sales;
```

Checking if the data was inserted into the table dynamic_sales

```
hive> select * from dynamic_sales limit 5;
OK
dynamic_sales.event_time      dynamic_sales.product_id      dynamic_sales.category_id      d
dynamic_sales.category_code   dynamic_sales.brand           dynamic_sales.price            dynamic_sale
s.user_id                    dynamic_sales.user_session    dynamic_sales.event_type
2019-10-10 13:00:56 UTC 5810479 1487580005268456287      22.22      410940282 2
fe5d5c9-0c45-4ff5-8a3c-245dd5d5961e      cart
2019-10-08 08:16:43 UTC 5888521 1597770225539875791      laboratoriu      7.14      5579
83520      961b81d9-8e7f-465a-8a7a-c40847fd216c      cart
2019-10-10 00:00:23 UTC 5745314 1487580010628776014      1.43      553999552 9
986365b-eb78-4b42-98ba-b90141b92c00      cart
2019-10-09 07:33:01 UTC 5817689 1487580005092295511      10.32      541115674 f
7924d95-94a8-473a-b947-970c54ac60ad      cart
2019-10-08 13:25:19 UTC 5796094 1487580005754995573      4.44      318432695 8
c8ae9d3-7228-4f7a-ab0b-f9afb1a004d7      cart
Time taken: 0.199 seconds, Fetched: 5 row(s)
```

Checking the partitions created in hive

```
hive> show partitions dynamic_sales;
OK
event_type=cart
event_type=purchase
event_type=remove_from_cart
event_type=view
Time taken: 0.08 seconds, Fetched: 4 row(s)
```

Checking the partitions created in Hadoop

```
[hadoop@ip-172-31-47-94 ~]$ hadoop fs -ls /user/hive/warehouse/ecommerce.db/dyna
mic_sales;
Found 4 items
drwxrwxrwt - hadoop hadoop 0 2022-07-04 07:42 /user/hive/warehouse/ec
ommerce.db/dynamic_sales/event_type=cart
drwxrwxrwt - hadoop hadoop 0 2022-07-04 07:42 /user/hive/warehouse/ec
ommerce.db/dynamic_sales/event_type=purchase
drwxrwxrwt - hadoop hadoop 0 2022-07-04 07:42 /user/hive/warehouse/ec
ommerce.db/dynamic_sales/event_type=remove_from_cart
drwxrwxrwt - hadoop hadoop 0 2022-07-04 07:42 /user/hive/warehouse/ec
ommerce.db/dynamic sales/event_type=view
```

We run the same query on the optimized table. The time taken has reduced to 25 seconds.

```
hive> select sum(price) as total_revenue from dynamic_sales where month(event_time) =10 and event_type = 'purchase';
Query ID = hadoop_20220704130354_23f2f319-a7d1-45f8-a59c-31e60381e46b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0014)
```

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

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 24.65 s
OK
total_revenue
1211538.4299998758
Time taken: 25.288 seconds, Fetched: 1 row(s)
```

As we get quick results with the optimized table, we use the same table for further queries.

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

```
select month(event_time) as purchase_month, count(product_id) as total_purchases
from dynamic_sales
where event_type = 'purchase'
group by month(event_time);
```

Ans: 10 – October 245624
 11 – November 322417

There was more purchase made in the month of November than in the month of October.

```
hive> select month(event_time) as purchase_month, count(product_id) as total_purchases from dynamic_sales where event_type = 'purchase' group by month(event_time);
Query ID = hadoop_20220704130550_104a169b-5a28-48ca-9656-6ac20baf54b7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0014)
```

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

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 26.24 s
OK
purchase_month  total_purchases
10             245624
11             322417
Time taken: 26.95 seconds, Fetched: 2 row(s)
```

Q3. Write a query to find the change in revenue generated due to purchases from October to November.

```
WITH monthly_sales as
(select round (sum (case when date_format (event_time, 'MM')=10 then price else 0
end),2) as oct_sales,
round (sum (case when date_format (event_time, 'MM') =11 then price else 0 end),2) as
nov_sales
from dynamic_sales
where event_type = 'purchase' and date_format (event_time, 'MM') in ('10', '11'))
select nov_sales, oct_sales, (nov_sales - oct_sales) as change_in_revenue
from monthly_sales;
```

Ans: nov_sales	oct_sales	change_in_revenue
1531016.9	1211538.43	319478.47

There is an increase of 319478.47 in the revenue generated due to purchase from October to November.

```
hive> WITH monthly_sales as ( select round ( sum (case when date_format (event_time, 'MM') = 10 th
en price else 0 end),2) as oct_sales, round (sum (case when date_format (event_time, 'MM') =11 the
n price else 0 end),2) as nov_sales from dynamic_sales where event_type = 'purchase' and date_form
at (event_time, 'MM') in ('10', '11') ) select nov_sales, oct_sales, (nov_sales - oct_sales) as ch
ange_in_revenue from monthly_sales;
Query ID = hadoop_20220704131633_4ac8a8b1-12c3-4eed-84a3-63e0a1d23f53
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0015)

-----
VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    3         3         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    1         1         0         0         0         0
-----
VERTICES: 02/02  [=====>>] 100%  ELAPSED TIME: 39.80 s
-----
OK
nov_sales      oct_sales      change_in_revenue
1531016.9      1211538.43      319478.47
Time taken: 41.134 seconds, Fetched: 1 row(s)
```

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

```
select distinct split(category_code,'\\\.')[0] as product_category from dynamic_sales
where split(category_code,'\\\.')[0] IS NOT NULL;
```

Ans: product_category:-

- furniture
- appliances
- accessories
- apparel
- sport
- stationery

There are 6 different categories under which the company sells products.

```
hive> select distinct split(category_code,'\\\.')[0] as product_category from dynamic_sales where
split(category_code,'\\\.')[0] IS NOT NULL;
Query ID = hadoop_20220704132702_d7592de7-f492-4b57-b235-a2d14a402d8b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0017)

-----
      VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    4         4         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    5         5         0         0         0         0
-----
VERTICES: 02/02  [=====>>] 100%  ELAPSED TIME: 74.85 s
-----
OK
product_category

furniture
appliances
accessories
apparel
sport
stationery
Time taken: 75.795 seconds, Fetched: 7 row(s)
```

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

```
select split(category_code,'\\.')[0] as product_category, count(product_id) as
total_products
from dynamic_sales
where split(category_code,'\\.')[0] IS NOT NULL
group by split(category_code,'\\.')[0];
```

Ans:

product_category	total_products
furniture	23604
appliances	61736
accessories	12929
apparel	18232
sport	2
stationery	26722

The category of Appliances has the highest number of products followed by furniture, stationery, apparel, accessories and sports.

```
hive> select split(category_code,'\\.')[0] as product_category, count(product_id) as total_products from dynamic_sales where split(category_code,'\\.')[0]<>' ' group by split(category_code,'\\.')[0];
Query ID = hadoop_20220704134113_a8cd17f8-fe02-4be2-b68f-096ecd93e2e3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0017)

-----
      VERTICES      MODE      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED    4         4         0         0         0         0
Reducer 2 ..... container  SUCCEEDED    5         5         0         0         0         0
-----
VERTICES: 02/02  [=====>>] 100%  ELAPSED TIME: 69.35 s
-----
OK
product_category      total_products
furniture             23604
appliances            61736
accessories           12929
apparel 18232
sport 2
stationery            26722
Time taken: 69.999 seconds, Fetched: 6 row(s)
```

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

```
select brand, round(sum(price),2) as Sales from dynamic_sales
where brand <>'' and event_type = 'purchase'
group by brand
order by Sales desc
limit 1;
```

Ans: brand sales
 runail 148297.94

Runail is the brand having maximum sales in October and November combined.

```
hive> select brand, round(sum(price),2) as Sales from dynamic_sales where brand <>'' and event
_type = 'purchase' group by brand order by Sales desc limit 1;
Query ID = hadoop_20220704134927_1bca9723-4f5e-4677-aff9-6bb75abe4d11
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0018)
```

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

```
VERTICES: 03/03  [=====>>>] 100%  ELAPSED TIME: 30.43 s
OK
brand     sales
runail    148297.94
Time taken: 37.508 seconds, Fetched: 1 row(s)
```

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

WITH monthly_sales as

(select brand, round (sum (case when date_format (event_time, 'MM') = 10 then price else 0 end),2) as oct_sales,

round (sum (case when date_format (event_time, 'MM') =11 then price else 0 end),2) as nov_sales

from dynamic_sales

where event_type = 'purchase' and date_format (event_time, 'MM') in ('10', '11')

group by brand)

select brand, oct_sales, nov_sales, (nov_sales – oct_sales) as difference_in_sales

from monthly_sales

where (nov_sales – oct_sales) > 0

order by difference_in_sales desc;

Ans: 161 brands have increased their sales from October to November.

Grattol has the highest increment of 36027 and Ovale has the lowest increment of 0.56.

```
hive> WITH monthly_sales as (select brand, round(sum(case when date_format(event_time, 'MM') = 10 then price else 0 end),2) as oct_sales, round(sum(case when date_format(event_time, 'MM') = 11 then price else 0 end),2) as nov_sales from dynamic_sales where event_type = 'purchase' and date_format(event_time, 'MM') in ('10','11') group by brand) select brand, oct_sales, nov_sales, (nov_sales-oct_sales) as difference_in_sales from monthly_sales where (nov_sales-oct_sales) > 0 order by difference_in_sales desc;
```

Query ID = hadoop_20220704141005_9bef0698-91a9-4b93-8fbc-8e7b2adaeed9

Total jobs = 1

Launching Job 1 out of 1

Status: Running (Executing on YARN cluster with App id application_1656916573590_0020)

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

VERTICES: 03/03 [=====>>>] 100% ELAPSED TIME: 44.05 s

OK

	474679.06	619509.24	144830.18
grattol	35445.54	71472.71	36027.170000000006
uno	35302.03	51039.75	15737.720000000001
lianail	5892.84	16394.24	10501.400000000001
ingarden		23161.39	33566.21
strong	29196.63	38671.27	9474.639999999996
jessnail	26287.84		33345.23
cosmoprofi	8322.81	14536.99	6214.18
polarus	6013.72	11371.93	5358.21
runail	71539.28	76758.66	5219.380000000005
freedecor	3421.78	7671.8	4250.02
staleks	8519.73	11875.61	3355.880000000001
bpw.style	11572.15		14837.44
lovely	8704.38	11939.06	3234.680000000003
marathon	7280.75	10273.1	2992.350000000004
haruyama	9390.69	12352.91	2962.2199999999993
yoko	8756.91	11707.88	2950.9699999999993
italwax	21940.24	24799.37	2859.1299999999974
benovy	409.62	3259.97	2850.35
kaypro	881.34	3268.7	2387.3599999999997
estel	21756.75	24142.67	2385.9199999999983
concept	11032.14	13380.4	2348.26
kapous	11927.16	14093.08	2165.92
f.o.x	6624.23	8577.28	1953.050000000001
masura	31266.08	33058.47	1792.3899999999994
milv	3904.94	5642.01	1737.0700000000002
beautix	10493.95	12222.95	1729.0

artex	2730.64	4327.25	1596.6100000000001
domix	10472.05	12009.17	1537.1200000000008
shik	3341.2	4839.72	1498.5200000000004
smart	4457.26	5902.14	1444.88
roubloff		3491.36	4913.77 1422.4100000000003
levrana	2243.56	3664.1	1420.54
oniq	8425.41	9841.65	1416.2399999999998
irisk	45591.96		46946.04 1354.0800000000017
severina		4775.88	6120.48 1344.5999999999995
joico	705.52	2015.1	1309.58
zeitun	708.66	2009.63	1300.9700000000003
beauty-free		554.17	1782.86 1228.69
swarovski		1887.93	3043.16 1155.2299999999998
de.lux	1659.7	2775.51	1115.8100000000002
metzger	5373.45	6457.16	1083.71
markell	1768.75	2834.43	1065.6799999999998
sanoto	157.14	1209.68	1052.54
nagaraku		4369.74	5327.68 957.9400000000005
ecolab	262.85	1214.3	951.4499999999999
art-visage		2092.71	2997.8 905.0900000000001
levissime		2227.5	3085.31 857.81
missha	1293.83	2150.28	856.4500000000003
solomeya		1899.7	2685.8 786.1000000000001
rosi	3077.04	3841.56	764.52
refectocil		2716.18	3475.58 759.4000000000001
kaaral	4412.43	5086.07	673.6399999999994
kosmekka		1181.44	1813.37 631.9299999999998
kinetics		6334.25	6945.26 611.0100000000002
browxenna		14331.37	14916.73 585.3599999999998
airnails		5118.9	5691.52 572.6200000000008
uskusi	5142.27	5690.31	548.04
coifin	903.0	1428.49	525.49
s.care	412.68	913.07	500.3900000000004
limoni	1308.9	1796.6	487.6999999999998
matrix	3243.25	3726.74	483.4899999999998
gehwol	1089.07	1557.68	468.6100000000001
greymy	29.21	489.49	460.2800000000003
bioaqua	942.89	1398.12	455.2299999999999
farmavita		837.37	1291.97 454.6
sophin	1067.86	1515.52	447.6600000000001
yu-r	271.41	673.71	402.3
kiss	421.55	817.33	395.7800000000003
naomi	0.0	389.0	389.0
lador	2083.61	2471.53	387.9200000000001
ellips	245.85	606.04	360.18999999999994
jas	3318.96	3657.43	338.4699999999998
lowence	242.84	567.75	324.9099999999997
nitrile	847.28	1162.68	315.4000000000001
shary	871.96	1176.49	304.53
kims	330.04	632.04	301.99999999999994
happyfons		801.92	1091.59 289.66999999999996
kocostar		310.85	594.93 284.0799999999999
insight	1443.7	1721.96	278.26
candy	534.96	799.38	264.41999999999996
bluesky	10307.24		10565.53 258.2900000000009
beauugreen		511.51	768.35 256.8400000000003
protokeratin		201.25	456.79 255.5400000000002
trind	298.07	542.96	244.8900000000004
entity	479.71	719.26	239.55
skinlite		651.94	890.45 238.51
provoc	827.99	1063.82	235.82999999999993
fedua	52.38	263.81	211.43
ecocraft		41.16	241.95 200.79
keen	236.35	435.62	199.27
mane	66.79	260.26	193.46999999999997
freshbubble		318.7	502.34 183.64
matreshka		0.0	182.67 182.67
chi	358.94	538.61	179.67000000000002
cristalinas		427.63	584.95 157.3200000000005
farmona	1692.46	1843.43	150.9700000000003
latinoil		249.52	384.59 135.06999999999996
maskin	158.04	293.07	135.03
elizavecca		70.53	204.3 133.77
nefertiti		233.52	366.64 133.11999999999998
finish	98.38	230.38	132.0
igrobeauty		513.66	645.07 131.41000000000008
dizao	819.13	945.51	126.38
osmo	645.58	762.31	116.72999999999999
batiste	772.4	874.17	101.76999999999998
carmex	145.08	243.36	98.28
eos	54.34	152.61	98.27000000000001
depilflax		2707.07	2803.78 96.71000000000004
enjoy	41.35	136.57	95.22
kerasys	430.91	525.2	94.29000000000002
aura	83.95	177.51	93.55999999999999
plazan	101.37	194.01	92.63999999999999
koelf	422.73	507.29	84.56
nirvel	163.04	234.33	71.29000000000002
konad	739.83	810.67	70.83999999999992
egomania		77.47	146.04 68.57
cutrin	299.37	367.62	68.25
laboratorium		246.5	312.52 66.01999999999998
inm	288.02	351.21	63.19
dewal	0.0	61.29	61.29
marutaka-foot		49.22	109.33 60.11
kares	0.0	59.45	59.45
profhenna		679.23	736.85 57.62000000000005
koelcia	55.5	112.75	57.25

balbcare	155.33	212.38	57.04999999999998
elskin	251.09	307.65	56.559999999999974
foamie	35.04	80.49	45.449999999999996
ladykin	125.65	170.57	44.919999999999999
likato	296.06	340.97	44.9100000000000025
mavala	409.04	446.32	37.279999999999997
vilenta	197.6	231.21	33.6100000000000014
beautyblender	78.74	109.41	30.67
biore	60.65	90.31	29.6600000000000004
orly	902.38	931.09	28.7100000000000036
estelare	444.81	471.87	27.0600000000000002
profepil	93.36	118.02	24.659999999999997
blizx	38.95	63.4	24.449999999999996
binacil	0.0	24.26	24.26
godefroy	401.22	425.12	23.899999999999977
glysolid	69.73	91.59	21.86
veraclara	50.11	71.21	21.099999999999994
juno	0.0	21.08	21.08
kamill	63.01	81.49	18.479999999999997
treaclemoon	163.37	181.49	18.1200000000000005
supertan	50.37	66.51	16.1400000000000008
barbie	0.0	12.39	12.39
deoproce	316.84	329.17	12.3300000000000041
rasyan	18.8	28.94	10.14
fly	17.14	27.17	10.0300000000000001
tertio	236.16	245.8	9.6400000000000015
jaguar	1102.11	1110.65	8.5400000000000191
soleo	204.2	212.53	8.3300000000000013
neoleor	43.41	51.7	8.2900000000000006
moyou	5.71	10.28	4.569999999999999
bodyton	1376.34	1380.64	4.3000000000000182
skinity	8.88	12.44	3.5599999999999987
helloganic	0.0	3.1	3.1
grace	100.92	102.61	1.6899999999999977
cosima	20.23	20.93	0.6999999999999993
ovale	2.54	3.1	0.56

Time taken: 45.459 seconds, Fetched: 161 row(s)

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.

```
select user_id, round(sum(price),2) as total_purchase
from dynamic_sales
where event_type = 'purchase'
group by user_id
order by total_purchase desc limit 10;
```

Ans:

user_id	total_purchase
557790271	2715.87
150318419	1645.97
562167663	1352.85
531900924	1329.45
557850743	1295.48
522130011	1185.39
561592095	1109.7
431950134	1097.59
566576008	1056.36
521347209	1040.91

This is the list of the top 10 users who spend the most and can be rewarded with a Golden Customer Plan.

```
hive> select user_id, round(sum(price),2) as total_purchase from dynamic_sales where event_type = '
purchase' group by user_id order by total_purchase desc limit 10;
Query ID = hadoop_20220704145218_7b5d0319-a134-4750-a2e0-4f824b31873d
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1656916573590_0021)

-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED   3       3           0         0         0         0
Reducer 2 ..... container  SUCCEEDED   1       1           0         0         0         0
Reducer 3 ..... container  SUCCEEDED   1       1           0         0         0         0
-----
VERTICES: 03/03  [=====>>] 100%  ELAPSED TIME: 27.56 s
-----
OK
user_id total_purchase
557790271      2715.87
150318419      1645.97
562167663      1352.85
531900924      1329.45
557850743      1295.48
522130011      1185.39
561592095      1109.7
431950134      1097.59
566576008      1056.36
521347209      1040.91
Time taken: 31.435 seconds, Fetched: 10 row(s)
```

Cleaning up

Dropping the database

```
hive> DROP DATABASE ecommerce CASCADE;  
OK  
Time taken: 0.419 seconds
```

Terminating the cluster

The screenshot shows the Amazon EMR console interface. The cluster 'Hive-Case-Study' is in a 'Terminating' state, indicated by an orange label. The console displays various tabs for the cluster, 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-3FMT01UJAYKQ8), creation date (2022-07-04 11:59 UTC+5:30), and elapsed time (10 hours, 8 minutes). It also lists the master public DNS, application user interfaces, and security and access information. The Network and hardware section shows the availability zone (us-east-1b) and the subnet ID (subnet-0aba3cea98526dec4). The cluster scaling is noted as 'Not enabled'.

```
Broadcast message from root@ip-172-31-47-94  
(unknown) at 16:38 ...  
  
The system is going down for power off NOW!  
[ ]
```