

Hive Case Study

- Sanjay Tom Perayil

Problem Synopsis: 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 their clicks on the website and searching for patterns within them. The clickstream data contains all the logs as to how the customer navigated through the website. It also contains other details such as time spent on every page, etc. From this, tech companies make use of data ingesting frameworks such as Apache Kafka or AWS Kinesis in order to store it in frameworks such as Hadoop. From there, machine learning engineers or business analysts use this data to derive valuable insights.

Objective: To extract data and gather insights from a real-life data set of an e-commerce company for analysing and gaining insights about customer behaviour.

The steps involved in the entire process are as follows:

The implementation phase can be divided into the following parts:

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

Launching an EMR Cluster:

In order to launch a cluster, following steps are followed:

Key-pair creation:

we need to create an EC2 key-pair file - **private170498** and download it as .ppk file.

EMR Cluster creation:

Click on "Create cluster" button to create the EMR cluster.

Creating clusters with advanced options.

The screenshot shows the 'Create Cluster - Quick Options' page in the AWS Management Console. The page is divided into three main sections: General Configuration, Software configuration, and Hardware configuration. In the General Configuration section, the 'Cluster name' is 'My cluster', 'Logging' is checked, and the 'S3 folder' is 's3://aws-logs-815937943894-us-east-1/elasticmap'. The 'Launch mode' is set to 'Cluster'. In the Software configuration section, the 'Release' is 'emr-5.34.0'. In the Hardware configuration section, the 'Instance type' is 'm5.xlarge' and the 'Number of instances' is '3' (1 master and 2 core nodes). The page includes a search bar at the top and a footer with '© 2022, Amazon Internet Services Private Ltd. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Change the software release from **emr-5.34.0** to **emr-5.29.0**.

The screenshot shows the 'Create Cluster - Advanced Options' page in the AWS Management Console. The page is divided into four steps: Step 1: Software and Steps, Step 2: Hardware, Step 3: General Cluster Settings, and Step 4: Security. The 'Software Configuration' section is active, showing the 'Release' as 'emr-5.29.0'. Below this, there are checkboxes for various software packages: Hadoop 2.8.5, JupyterHub 1.0.0, Ganglia 3.7.2, Hive 2.3.6, MXNet 1.5.1, Hue 4.4.0, Spark 2.4.4, Zeppelin 0.8.2, Tez 0.9.2, HBase 1.4.10, Presto 0.227, Sqoop 1.4.7, Phoenix 4.14.3, HCatalog 2.3.6, Livy 0.6.0, Flink 1.9.1, Pig 0.17.0, ZooKeeper 3.4.14, Mahout 0.13.0, Oozie 5.1.0, and TensorFlow 1.14.0. The 'Multiple master nodes (optional)' section has a checkbox for 'Use multiple master nodes to improve cluster availability.' and a link to 'Learn more'. The 'AWS Glue Data Catalog settings (optional)' section has a checkbox for 'Use for Hive table metadata'. The 'Edit software settings' section has a link to 'Edit software settings'. The page includes a search bar at the top and a footer with '© 2022, Amazon Internet Services Private Ltd. or its affiliates.' and links for 'Privacy', 'Terms', and 'Cookie preferences'.

Change the Master and Core nodes from **m5.xlarge** to **m4.large**.

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	On-demand 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	On-demand Spot Use on-demand as max price
Task Task - 3	m5.xlarge 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	0 Instances	On-demand Spot Use on-demand as max price

+ Add task instance group

Create a new cluster named **Hivecasestudy**.

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

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

General Options

Cluster name: Hivecasestudy

Logging: ☒ [i](#)

S3 folder: s3://aws-logs-815937943894-us-east-1/elasticmap

Debugging: ☒ [i](#)

Termination protection: ☒ [i](#)

Tags [i](#)

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

Additional Options

Change the EC2 key pair option to our newly created key pair - **private170498.ppk**.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia upgradsanjaytomperayil @ 8159-3794-3894

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 [private170498](#) ⓘ

☒ 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

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

And finally, click on create cluster option.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia upgradsanjaytomperayil @ 8159-3794-3894

Amazon EMR

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

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

Cluster: Hivecasestudy Starting

Summary Application user interfaces Monitoring Hardware Configurations Events Steps Bootstrap actions

Summary

ID: j-2SX9LHCQCGDKO
Creation date: 2022-01-31 20:59 (UTC+5:30)
Elapsed time: 1 second
After last step completes: Cluster waits
Termination protection: Off [Change](#)
Tags: -- [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-815937943894-us-east-1/elasticmapreduce/ ⓘ
EMRFS consistent view: Disabled

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Once ready, the cluster it displays the message **Waiting**.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia upgradsanjaytomperayil @ 8159-3794-3894

Amazon EMR

EMR Studio

EMR on EC2

Clusters

Notebooks

Git repositories

Security configurations

Block public access

VPC subnets

Events

EMR on EKS

Virtual clusters

Help

What's new

Clone Terminate AWS CLI export

Cluster: Hivecasestudy Waiting Cluster ready after last step completed.

Summary Application user interfaces Monitoring Hardware Configurations Events Steps Bootstrap actions

Summary

ID: j-2SX9LHCQCGDKO

Creation date: 2022-01-31 20:59 (UTC+5:30)

Elapsed time: 17 minutes

After last step completes: Cluster waits

Termination protection: Off Change

Tags: -- View All / Edit

Master public DNS: ec2-54-89-41-12.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://aws-logs-815937943894-us-east-1/elasticmapreduce/

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

We need to make sure before connecting to SSH, ensure that the port is open to establish a connection. For this, click on **Security groups** for Master node.

aws Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia upgradsanjaytomperayil @ 8159-3794-3894

New EC2 Experience Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Tags

Limits

Instances

Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances New

Dedicated Hosts

Scheduled Instances

Capacity Reservations

Security Groups (1/3) Info

Filter security groups

Actions Export security groups to CSV Create security group

	Name	Security group ID	Security group name	VPC ID	Description
<input type="checkbox"/>	-	sg-02959d9beecb2e408	ElasticMapReduce-slave	vpc-0df9b1ab645212b29	Slave group for
<input type="checkbox"/>	-	sg-0315141e01225a8a0	default	vpc-0df9b1ab645212b29	default VPC sec
<input checked="" type="checkbox"/>	-	sg-098dccf7f2578cb72	ElasticMapReduce-mas...	vpc-0df9b1ab645212b29	Master group fo

sg-098dccf7f2578cb72 - ElasticMapReduce-master

Details Inbound rules Outbound rules Tags

You can now check network connectivity with Reachability Analyzer Run Reachability Analyzer

Feedback English (US) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Click on edit Inbound rules.

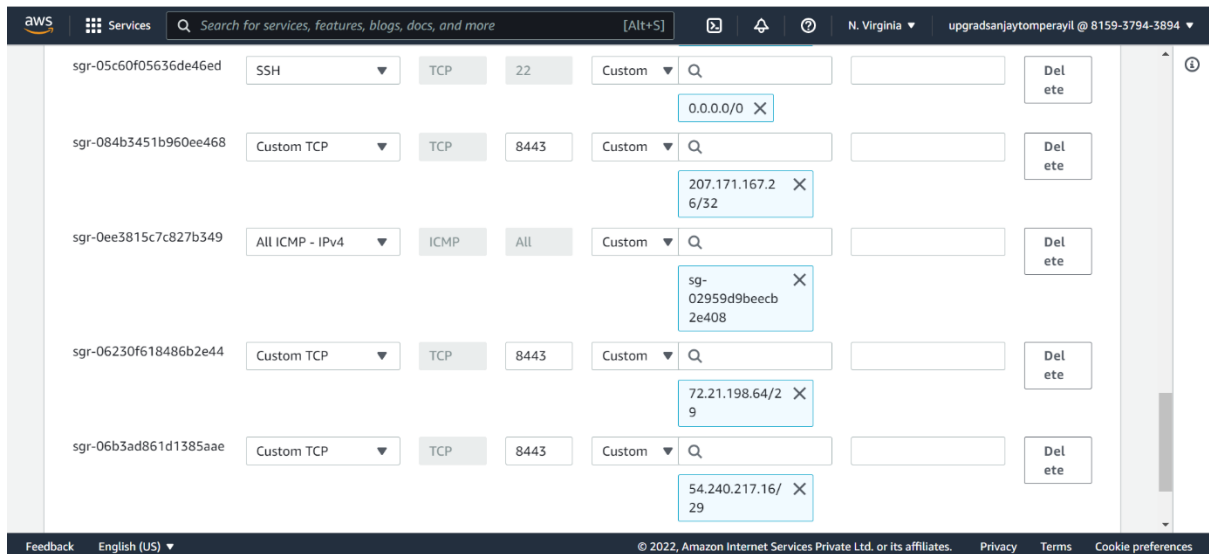
Inbound rules (19)

Filter security group rules

Manage tags Edit inbound rules

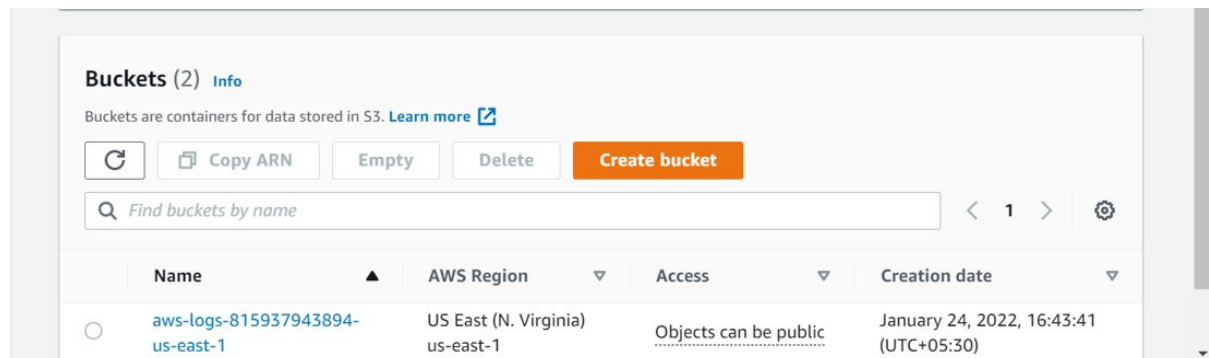
	Name	Security group rule...	IP version	Type	Protocol
<input type="checkbox"/>	-	sg-0880c05566ed18f48	-	All ICMP - IPv4	ICMP

Add a new rule by selecting SSH and change the IP address to Anywhere.

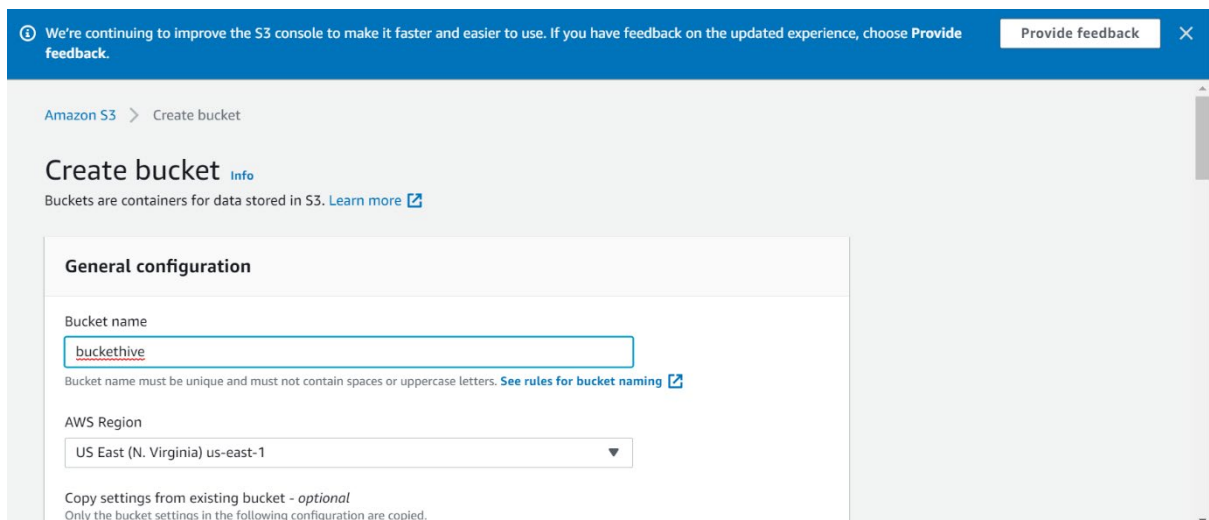


S3 bucket creation:

Click on create bucket.



Creating a new bucket named **buckethive** with default options.



New bucket named **buckethive** successfully created.

The screenshot shows the 'Buckets (2)' page in the Amazon S3 console. It includes a search bar 'Find buckets by name' and a table of buckets. The 'buckethive' bucket is listed with the following details:

Name	AWS Region	Access	Creation date
aws-logs-815937943894-us-east-1	US East (N. Virginia) us-east-1	Objects can be public	January 24, 2022, 16:43:41 (UTC+05:30)
buckethive	US East (N. Virginia) us-east-1	Bucket and objects not public	January 24, 2022, 16:22:56 (UTC+05:30)

Create a new folder under the bucket – **buckethive**.

The screenshot shows the 'Objects (1)' page in the Amazon S3 console. It includes a search bar 'Find objects by prefix' and a table of objects. A folder named 'casestudy/' is listed with the following details:

Name	Type	Last modified	Size	Storage class
casestudy/	Folder	-	-	-

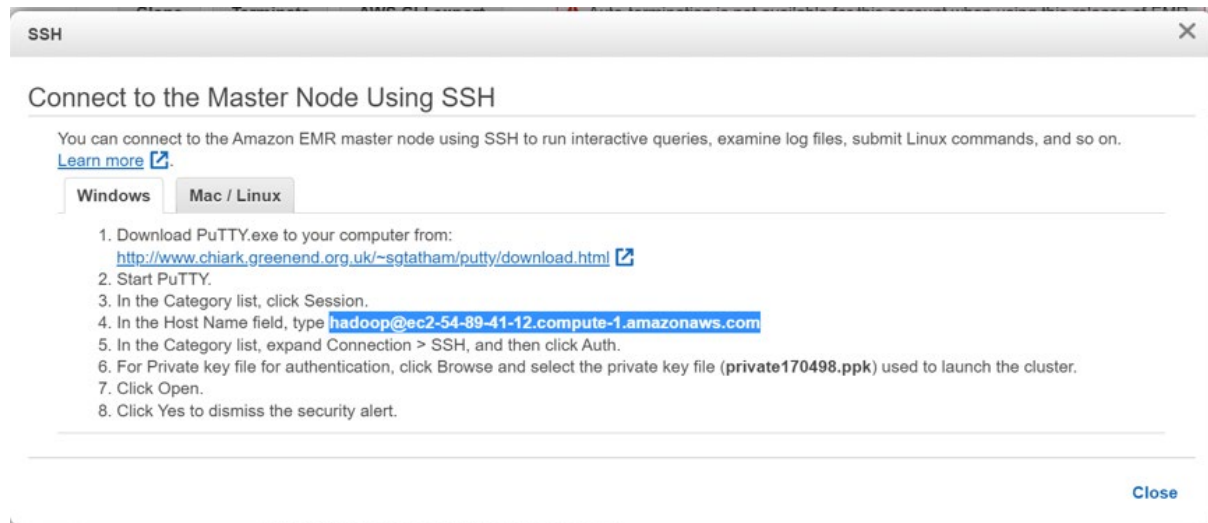
Upload 2 csv files 2019-Oct.csv and 2019-Nov.csv under the folder **casestudy**.

The screenshot shows the Amazon S3 console with the 'Objects (2)' page. Two CSV files have been uploaded under the 'casestudy/' folder:

Name	Type	Last modified	Size	Storage class
2019-Nov.csv	csv	January 24, 2022, 18:31:36 (UTC+05:30)	520.6 MB	Standard
2019-Oct.csv	csv	January 24, 2022, 18:31:36 (UTC+05:30)	460.2 MB	Standard

Connect to Master node using SSH:

Open Putty application and connect to host by using the highlighted code shown on the screenshot.



Navigate to Connection and move to Auth under SSH, then provide the private key pair - **private170498.ppk** which was created earlier.

Master node connected and Hive shell Launched.

```
hadoop@ip-172-31-42-136:~  
Using username "hadoop".  
Authenticating with public key "private170498"  
Last login: Fri Jan 28 13:04:29 2022  
  
      _|   _|   )  
      _| ( _ - /    Amazon Linux AMI  
      _|\__|_|_|_|_|  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
68 package(s) needed for security, out of 106 available  
Run "sudo yum update" to apply all updates.  
  
EEEEEEEEEEEEEEEEEEEEEE MMMMMMMM          MMMMMMMM RRRRRRRRRRRRRRRR  
E::::::::::::::::::E M::::::::M           M::::::::M R:::::::::::::R  
EE::::::::EEEEEEEEEE E M::::::::M         M::::::::M R:::RRRRRR:::R  
  E:::E       EEEE M::::::::M           M::::::::M RR:::R        R:::R  
  E:::E       M::::M:M::M     M::M:M::M     R:::R        R:::R  
  E:::EEEEEEEEEE M::::M M::M M::M M::M M::M R:::RRRRRR:::R  
  E::::::::::::E M::::M M::M:M::M M::M:M M::M R:::::::::RR  
  E:::EEEEEEEEEE M::::M M:::M M::M M::M M::M R:::RRRRRR:::R  
  E:::E       M::::M M::M M::M M::M M::M R:::R        R:::R  
  E:::E       EEEE M::::M     MMM     M::M M::M R:::R        R:::R  
EE::::::::EEEEEEEE::E M::::M           M::M:M M::M R:::R        R:::R  
E::::::::::::::::::E M::::M           M::::M RR:::R        R:::R  
EEEEEEEEEEEEEEEEEEEEEE MMMMMMMM          MMMMMMMM RRRRRRRR      RRRRRR  
  
[hadoop@ip-172-31-42-136 ~]$ hadoop fs -ls /  
Found 4 items  
drwxr-xr-x   - hdfs hadoop            0 2022-01-28 12:29 /apps  
drwxrwxrwt   - hdfs hadoop            0 2022-01-28 12:32 /tmp  
drwxr-xr-x   - hdfs hadoop            0 2022-01-28 12:29 /user  
drwxr-xr-x   - hdfs hadoop            0 2022-01-28 12:29 /var
```


All of the above are inbuilt directories in the HDFS.

Creating a temporary directory in the HDFS

```
hadoop fs -mkdir /user/hivecasestudy
```

```
hadoop fs -ls /user/
```

```
[hadoop@ip-172-31-42-136 ~]$ hadoop fs -mkdir /user/hivecasestudy
[hadoop@ip-172-31-42-136 ~]$ hadoop fs -ls /user/
Found 7 items
drwxrwxrwx - hadoop hadoop      0 2022-01-28 12:29 /user/hadoop
drwxr-xr-x - mapred mapred      0 2022-01-28 12:29 /user/history
drwxrwxrwx - hdfs hadoop        0 2022-01-28 12:29 /user/hive
drwxr-xr-x - hadoop hadoop      0 2022-01-28 13:27 /user/hivecasestudy
drwxrwxrwx - hue hue            0 2022-01-28 12:29 /user/hue
drwxrwxrwx - oozie oozie        0 2022-01-28 12:29 /user/oozie
drwxrwxrwx - root hadoop        0 2022-01-28 12:29 /user/root
```

New directory – **hivecasestudy** is successfully created.

Loading the data into the HDFS:

```
hadoop distcp s3://buckethive/casestudy/2019-Oct.csv /user/hivecasestudy/2019-Oct.csv
```

```
hadoop@ip-172-31-42-136~$ hadoop distcp s3://buckethive/casestudy/2019-Oct.csv /user/hivecasestudy/2019-Oct.csv
22/01/28 13:27:28 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://buckethive/casestudy/2019-Oct.csv], targetPath=/user/hivecasestudy/2019-Oct.csv, targetPathExists=false, filtersFile='null'}
22/01/28 13:27:28 INFO client.RMPProxy: Connecting to ResourceManager at ip-172-31-42-136.ec2.internal/172.31.42.136:8032
22/01/28 13:27:36 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/01/28 13:27:36 INFO tools.SimpleCopyListing: Build file listing completed.
22/01/28 13:27:36 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
22/01/28 13:27:36 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
22/01/28 13:27:37 INFO tools.DistCp: Number of paths in the copy list: 1
22/01/28 13:27:37 INFO tools.DistCp: Number of paths in the copy list: 1
22/01/28 13:27:37 INFO client.RMPProxy: Connecting to ResourceManager at ip-172-31-42-136.ec2.internal/172.31.42.136:8032
22/01/28 13:27:37 INFO mapreduce.JobSubmitter: number of splits:1
22/01/28 13:27:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1643373050143_0001
22/01/28 13:27:38 INFO impl.YarnClientImpl: Submitted application application_1643373050143_0001
22/01/28 13:27:39 INFO mapreduce.Job: The url to track the job: http://ip-172-31-42-136.ec2.internal:20888/proxy/application_1643373050143_0001/
22/01/28 13:27:39 INFO tools.DistCp: DistCp job-id: job_1643373050143_0001
22/01/28 13:27:39 INFO mapreduce.Job: Running job: job_1643373050143_0001
22/01/28 13:27:50 INFO mapreduce.Job: Job job_1643373050143_0001 running in uber mode : false
22/01/28 13:27:50 INFO mapreduce.Job: map 0% reduce 0%
22/01/28 13:28:08 INFO mapreduce.Job: map 100% reduce 0%
22/01/28 13:28:12 INFO mapreduce.Job: Job job_1643373050143_0001 completed successfully
22/01/28 13:28:12 INFO mapreduce.Job: Counters: 38
```

hadoop@ip-172-31-42-136:~

```
File System Counters
  FILE: Number of bytes read=0
  FILE: Number of bytes written=172494
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=358
  HDFS: Number of bytes written=482542278
  HDFS: Number of read operations=12
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=4
  S3: Number of bytes read=482542278
  S3: Number of bytes written=0
  S3: Number of read operations=0
  S3: Number of large read operations=0
  S3: Number of write operations=0

Job Counters
  Launched map tasks=1
  Other local map tasks=1
  Total time spent by all maps in occupied slots (ms)=595360
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=18605
  Total vcore-milliseconds taken by all map tasks=18605
  Total megabyte-milliseconds taken by all map tasks=19051520

Map-Reduce Framework
  Map input records=1
  Map output records=0
  Input split bytes=135
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=396
  CPU time spent (ms)=19120
  Physical memory (bytes) snapshot=577859584
  Virtual memory (bytes) snapshot=3296989184
  Total committed heap usage (bytes)=503316480

File Input Format Counters
  Bytes Read=223

File Output Format Counters
  Bytes Written=0

DistCp Counters
  Bytes Copied=482542278
  Bytes Expected=482542278
  Files Copied=1
```

hadoop distcp s3://buckethive/casestudy/2019-Nov.csv /user/hivecasestudy/2019-Nov.csv

```
hadoop@ip-172-31-42-136:~$ hadoop distcp s3://buckethive/casestudy/2019-Nov.csv /user/hivecasestudy/2019-Nov.csv
22/01/28 13:28:22 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://buckethive/casestudy/2019-Nov.csv], targetPath=/user/hivecasestudy/2019-Nov.csv, targetPathExists=false, filtersFile='null']
22/01/28 13:28:22 INFO client.RMPProxy: Connecting to ResourceManager at ip-172-31-42-136.ec2.internal/172.31.42.136:8032
22/01/28 13:28:27 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/01/28 13:28:27 INFO tools.SimpleCopyListing: Build file listing completed.
22/01/28 13:28:27 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
22/01/28 13:28:27 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
22/01/28 13:28:27 INFO tools.DistCp: Number of paths in the copy list: 1
22/01/28 13:28:27 INFO tools.DistCp: Number of paths in the copy list: 1
22/01/28 13:28:27 INFO client.RMPProxy: Connecting to ResourceManager at ip-172-31-42-136.ec2.internal/172.31.42.136:8032
22/01/28 13:28:28 INFO mapreduce.JobSubmitter: number of splits:1
22/01/28 13:28:28 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1643373050143_0002
22/01/28 13:28:28 INFO impl.YarnClientImpl: Submitted application application_1643373050143_0002
22/01/28 13:28:28 INFO mapreduce.Job: The url to track the job: http://ip-172-31-42-136.ec2.internal:20888/proxy/application_1643373050143_0002/
22/01/28 13:28:28 INFO tools.DistCp: DistCp job-id: job_1643373050143_0002
22/01/28 13:28:28 INFO mapreduce.Job: Running job: job_1643373050143_0002
22/01/28 13:28:36 INFO mapreduce.Job: Job job_1643373050143_0002 running in uber mode : false
22/01/28 13:28:36 INFO mapreduce.Job: map 0% reduce 0%
22/01/28 13:28:54 INFO mapreduce.Job: map 100% reduce 0%
22/01/28 13:28:58 INFO mapreduce.Job: Job job_1643373050143_0002 completed successfully
22/01/28 13:28:58 INFO mapreduce.Job: Counters: 38
```

hadoop@ip-172-31-42-136:~

```
File System Counters
  FILE: Number of bytes read=0
  FILE: Number of bytes written=172500
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=360
  HDFS: Number of bytes written=545839412
  HDFS: Number of read operations=12
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=4
  S3: Number of bytes read=545839412
  S3: Number of bytes written=0
  S3: Number of read operations=0
  S3: Number of large read operations=0
  S3: Number of write operations=0
Job Counters
  Launched map tasks=1
  Other local map tasks=1
  Total time spent by all maps in occupied slots (ms)=605632
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=18926
  Total vcore-milliseconds taken by all map tasks=18926
  Total megabyte-milliseconds taken by all map tasks=19380224
Map-Reduce Framework
  Map input records=1
  Map output records=0
  Input split bytes=137
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=352
  CPU time spent (ms)=20120
  Physical memory (bytes) snapshot=574844928
  Virtual memory (bytes) snapshot=3296317440
  Total committed heap usage (bytes)=471859200
File Input Format Counters
  Bytes Read=223
File Output Format Counters
  Bytes Written=0
DistCp Counters
  Bytes Copied=545839412
  Bytes Expected=545839412
  Files Copied=1
```

We have successfully loaded both the files into HDFS.

Checking the successful loading of data files:

```
hadoop fs -ls /user/hivecasestudy
```

```
[hadoop@ip-172-31-42-136 ~]$ hadoop fs -ls /user/hivecasestudy
Found 2 items
-rw-r--r--  1 hadoop hadoop  545839412 2022-01-28 13:28 /user/hivecasestudy/2019-Nov.csv
-rw-r--r--  1 hadoop hadoop  482542278 2022-01-28 13:28 /user/hivecasestudy/2019-Oct.csv
```

Now, that we can see the 2 uploaded files present in the directory.

Inspecting the table:

```
hadoop fs -cat /user/hivecasestudy/2019-Oct.csv | head
```

```
[hadoop@ip-172-31-42-136 ~]$ hadoop fs -cat /user/hivecasestudy/2019-Oct.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-10-01 00:00:00 UTC, cart, 5773203, 1487580005134238553, ,runail, 2.62, 463240011, 26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:03 UTC, cart, 5773353, 1487580005134238553, ,runail, 2.62, 463240011, 26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:07 UTC, cart, 5881589, 2151191071051219817, ,lovely, 13.48, 429681830, 49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:07 UTC, cart, 5723490, 1487580005134238553, ,runail, 2.62, 463240011, 26dd6e6e-4dac-4778-8d2c-92e149dab885
2019-10-01 00:00:15 UTC, cart, 5881449, 1487580013522845895, ,lovely, 0.56, 429681830, 49e8d843-adf3-428b-a2c3-fe8bc6a307c9
2019-10-01 00:00:16 UTC, cart, 5857269, 1487580005134238553, ,runail, 2.62, 430174032, 73deale7-664e-43f4-8b30-d32b9d5af04f
2019-10-01 00:00:19 UTC, cart, 5739055, 1487580008246412266, ,kapous, 4.75, 377667011, 81326ac6-daa4-4f0a-b488-fd0956a78733
2019-10-01 00:00:24 UTC, cart, 5825598, 1487580009445982239, , ,0.56, 467916806, 2f5b5546-b8cb-9ee7-7ecd-84276f8ef486
2019-10-01 00:00:25 UTC, cart, 5698989, 1487580006317032337, , ,1.27, 385985999, d30965e8-1101-44ab-b45d-cc1bb9fae694
cat: Unable to write to output stream.
```

`hadoop fs -cat /user/hivecasestudy/2019-Nov.csv | head`

```
[hadoop@ip-172-31-42-136 ~]$ hadoop fs -cat /user/hivecasestudy/2019-Nov.csv | head
event_time,event_type,product_id,category_id,category_code,brand,price,user_id,user_session
2019-11-01 00:00:02 UTC, view, 5802432, 1487580009286598681, , ,0.32, 562076640, 09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC, cart, 5844397, 1487580006317032337, , ,2.38, 553329724, 2067216c-31b5-455d-alcc-af0575a34ffb
2019-11-01 00:00:10 UTC, view, 5837166, 1783999064103190764, , ,pnb, 22.22, 556138645, 57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC, cart, 5876812, 1487580010100293687, , ,jessnail, 3.16, 564506666, 186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC, remove_from_cart, 5826182, 1487580007483048900, , , ,3.33, 553329724, 2067216c-31b5-455d-alcc-af0575a34ffb
2019-11-01 00:00:24 UTC, remove_from_cart, 5826182, 1487580007483048900, , , ,3.33, 553329724, 2067216c-31b5-455d-alcc-af0575a34ffb
2019-11-01 00:00:25 UTC, view, 5856189, 1487580009026551821, , ,runail, 15.71, 562076640, 09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:32 UTC, view, 5837835, 1933472286753424063, , , ,3.49, 514649199, 432a4e95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC, remove_from_cart, 5870838, 1487580007675986893, , ,milv, 0.79, 429913900, 2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
cat: Unable to write to output stream.
```

Creating Hive Schema & Optimization of tables:

hive

```
[hadoop@ip-172-31-42-136 ~]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
```

Creating a new Database for the casestudy:

create database if not exists casestudy;
use casestudy;

```
hive> create database if not exists casestudy;
OK
Time taken: 0.841 seconds
hive> use casestudy;
OK
Time taken: 0.066 seconds
```

Creating an external table in hive to load the data:

CREATE EXTERNAL TABLE IF NOT EXISTS store (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/hivecasestudy/'
TBLPROPERTIES ("skip.header.line.count"="1");

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS store (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/hivecasestudy/'
> TBLPROPERTIES ("skip.header.line.count"="1");
OK
Time taken: 0.338 seconds
```

Checking the table store:

describe store;

```
hive> describe store;
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: 0.461 seconds, Fetched: 9 row(s)
```

LOAD DATA INPATH '/user/hivecasestudy/2019-Oct.csv' INTO TABLE store;

```
hive> LOAD DATA INPATH '/user/hivecasestudy/2019-Oct.csv' INTO TABLE store;
Loading data to table casestudy.store
OK
Time taken: 1.719 seconds
```

LOAD DATA INPATH '/user/hivecasestudy/2019-Oct.csv' INTO TABLE store;

```
hive> LOAD DATA INPATH '/user/hivecasestudy/2019-Nov.csv' INTO TABLE store;
Loading data to table casestudy.store
OK
Time taken: 0.706 seconds
```

Checking the successful creation of table and transfer of data into the table:

SELECT * FROM store
LIMIT 10;

```
hive> SELECT * FROM store
> LIMIT 10;
OK
2019-11-01 00:00:02 UTC view 5802432 1487580009286598681 0.32 562076640 09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart 5844397 1487580006317032337 2.38 553329724 2067216c-31b5-455d-alcc-af0575a34ffb
2019-11-01 00:00:10 UTC view 5837166 1783999064103190764 pnb 22.22 556138645 57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart 5876812 1487580010100293687 jessnail 3.16 564506666 186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC remove_from_cart 5826182 1487580007483048900 3.33 553329724 2067216c-31b5-455d-alcc-af0575a34ffb
2019-11-01 00:00:24 UTC remove_from_cart 5826182 1487580007483048900 3.33 553329724 2067216c-31b5-455d-alcc-af0575a34ffb
2019-11-01 00:00:25 UTC view 5856189 1487580009026551821 runail 15.71 562076640 09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:32 UTC view 5837835 1933472286753424063 3.49 514649199 432a4e95-375c-4b40-bd36-0fc039e77580
2019-11-01 00:00:34 UTC remove_from_cart 5870838 1487580007675986893 milv 0.79 429913900 2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
2019-11-01 00:00:37 UTC view 5870803 1487580007675986893 milv 0.79 429913900 2f0bff3c-252f-4fe6-afcd-5d8a6a92839a
Time taken: 3.443 seconds, Fetched: 10 row(s)
```

Enabling Dynamic partitioning and Bucketing:

```
set hive.cli.print.header=true;
set hive.exec.dynamic.partition=true;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.enforce.bucketing=true;
```

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

Optimization of tables - Partitioning and Bucketing:

We will be using partitioning on **event_type**.

We will be using bucketing and clustering simultaneously on price and will divide it into **20 buckets**.

```
CREATE EXTERNAL TABLE IF NOT EXISTS store_1 (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 (price) INTO 20 buckets
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS TEXTFILE LOCATION '/user/hivecasestudy/'
TBLPROPERTIES ("skip.header.line.count"="1");
```

```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS store_1 (event_time timestamp, product_id string, category_id string, category_code string, brand string, price flo
at, user_id bigint, user_session string)
> PARTITIONED BY (event_type string)
> CLUSTERED BY (price) INTO 20 buckets
> ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
> STORED AS TEXTFILE LOCATION '/user/hivecasestudy/'
> TBLPROPERTIES ("skip.header.line.count"="1");
OK
Time taken: 0.099 seconds
```

Checking the new table store_1:

describe store_1;

```
hive> describe store_1;
OK
col_name      data_type      comment
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: 0.112 seconds, Fetched: 14 row(s)
```

Inserting data into partitioned and bucketed table:

```
INSERT INTO TABLE store_1 partition(event_type)
SELECT event_time, product_id, category_id, category_code, brand, price, user_id,
user_session, event_type
FROM store;
```

```
hive> INSERT INTO TABLE store_1 partition(event_type)
> SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type
> FROM store;
Query ID = hadoop_20220128133434_30a762aa-c5be-4677-86b0-f170a14ac4fd
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)

-----
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: 178.48 s
-----
Loading data to table casestudy.store_1 partition (event_type=null)

Loaded : 4/4 partitions.
Time taken to load dynamic partitions: 0.507 seconds
Time taken for adding to write entity : 0.006 seconds
OK
event_time      product_id      category_id      category_code      brand      price      user_id user_session      event_type
Time taken: 182.455 seconds
```

Based on the above screenshot, we can verify that 4 partitions have been created.

Querying Questions:

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

Non-optimized table

```
SELECT SUM(price) as Oct_revenue
FROM store
WHERE event_type = 'purchase' AND month(event_time) = 10;
```

```
hive> SELECT SUM(price) as Oct_revenue
> FROM store
> WHERE event_type = 'purchase' AND month(event_time) = 10;
Query ID = hadoop_20220128134242_8a254e32-08b3-495e-8381-c0d37b94b9b7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)

-----
VERTICES      MODE           STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
-----
Map 1 ..... container  SUCCEEDED   5       5         0         0         0         0
Reducer 2 ..... container  SUCCEEDED   1       1         0         0         0         0
-----
VERTICES: 02/02  [=====>>] 100%  ELAPSED TIME: 106.88 s
-----
OK
oct_revenue
1211538.4299997438
Time taken: 109.374 seconds, Fetched: 1 row(s)
```


Optimized table

```
SELECT SUM(price) as Oct_revenue
FROM store_1
WHERE event_type = 'purchase' AND month(event_time) = 10;
```

```
hive> SELECT SUM(price) as Oct_revenue
> FROM store_1
> WHERE event_type = 'purchase' AND month(event_time) = 10;
Query ID = hadoop_20220128134507_3abbd413-4af4-4805-a7c1-b63fc7e2dd0b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)

-----
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: 27.00 s
-----
OK
oct_revenue
1211489.6700001915
Time taken: 28.262 seconds, Fetched: 1 row(s)
```

- We see almost similar results obtained with slight variation through both the ways as the data used is the same for both the cases.
- We observed that, the query execution time for non-optimized table was 109.374 seconds, while incase for optimized query is 28.262 seconds, with a difference of 81.112 seconds. This is a huge variation, as the data size increase the execution time will also increase.

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

Non-optimized table

```
SELECT month(event_time) as Month, SUM(price) as Total_revenue
FROM store
WHERE event_type = 'purchase'
GROUP BY month(event_time);
```



```
hive> SELECT month(event_time) as Month, SUM(price) as Total_revenue
> FROM store
> WHERE event_type = 'purchase'
> GROUP BY month(event_time);
Query ID = hadoop_20220128134626_595cd7ee-991d-4403-8b00-62f9b6985063
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	5	5	0	0	0	0
Reducer 2	container	SUCCEEDED	3	3	0	0	0	0

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 107.41 s
OK
month    total_revenue
10       1211538.4299997438
11       1531016.900000122
Time taken: 108.244 seconds, Fetched: 2 row(s)
```

Optimized table

```
SELECT month(event_time) as Month, SUM(price) as Total_revenue
FROM store_1
WHERE event_type = 'purchase'
GROUP BY month(event_time);
```

```
hive> SELECT month(event_time) as Month, SUM(price) as Total_revenue
> FROM store_1
> WHERE event_type = 'purchase'
> GROUP BY month(event_time);
Query ID = hadoop_20220128135138_07fbd7fe-d96f-44e0-a3bf-49ad3a67701f
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)
```

	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: 26.21 s
OK
month    total_revenue
10       1211489.6700001918
11       1530983.7700001802
Time taken: 26.819 seconds, Fetched: 2 row(s)
```

- Sum of purchases made in the month of October is 1211489 while in the month of November, the sum of purchases is 1530983, which means number of purchases are increased in November month
- We observed that, the query execution time for non-optimized table was 108.244 seconds, while incase for optimized query is 26.819 seconds, with a difference of 81.425 seconds. We can see there is a huge difference in the execution time of the same query.

Hence, with proper partitioning and bucketing on table we can reduce execution time.

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

```
WITH Monthly_revenue AS (  
  SELECT  
    SUM (CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS Oct_Revenue,  
    SUM (CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS Nov_Revenue  
  FROM store_1  
  WHERE event_type= 'purchase'  
  AND date_format(event_time, 'MM') in ('10','11')  
)  
SELECT Nov_revenue, Oct_revenue, (Nov_revenue-Oct_revenue) AS  
Difference_Of_revenue FROM Monthly_revenue;
```

```
hive> WITH Monthly_revenue AS (  
  > SELECT  
  > SUM (CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS Oct_Revenue,  
  > SUM (CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS Nov_Revenue  
  > FROM store_1  
  > WHERE event_type= 'purchase'  
  > AND date_format(event_time, 'MM') in ('10','11')  
  > )  
  > SELECT Nov_revenue, Oct_revenue, (Nov_revenue-Oct_revenue) AS  
  > Difference_Of_revenue FROM Monthly_revenue;  
Query ID = hadoop_20220128135256_09b5af88-20a1-4017-b382-d882bfc8d228  
Total jobs = 1  
Launching Job 1 out of 1  
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)  
  
-----  
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: 42.30 s  
-----  
OK  
nov_revenue      oct_revenue      difference_of_revenue  
1530983.7700001802      1211489.6700001915      319494.0999999887  
Time taken: 42.933 seconds, Fetched: 1 row(s)
```

The change in revenue generated due to purchases from October to November is 319494.099.

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

```
SELECT DISTINCT SPLIT(category_code, '\\.')[0] AS category  
FROM store_1  
WHERE SPLIT(category_code, '\\.')[0]<>;
```

```
hive> SELECT DISTINCT SPLIT(category_code,'\\\.')[0] AS category
> FROM store_1
> WHERE SPLIT(category_code,'\\\.')[0]<>'';
Query ID = hadoop_20220128135452_67d31252-2294-4f82-9355-590clab5441a
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	6	6	0	0	0	0	0
Reducer 2	container	SUCCEEDED	5	5	0	0	0	0	0

```
VERTICES: 02/02 [=====>>] 100% ELAPSED TIME: 72.71 s
OK
category
furniture
appliances
accessories
apparel
sport
stationery
Time taken: 73.399 seconds, Fetched: 6 row(s)
```

There are 6 Distinct categories of products.

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

```
SELECT SPLIT(category_code,'\\\.')[0] AS category, COUNT(product_id) AS Total_product
FROM store_1
GROUP BY SPLIT(category_code,'\\\.')[0]
ORDER BY Total_product DESC;
```

```
hive> SELECT SPLIT(category_code,'\\\.')[0] AS category, COUNT(product_id) AS Total_product
> FROM store_1
> GROUP BY SPLIT(category_code,'\\\.')[0]
> ORDER BY Total_product DESC;
Query ID = hadoop_20220128135639_d4aaa55b-5aff-4e9b-abb9-1b81229e4b12
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)
```

	VERTICES	MODE	STATUS	TOTAL	COMPLETED	RUNNING	PENDING	FAILED	KILLED
Map 1	container	SUCCEEDED	6	6	0	0	0	0	0
Reducer 2	container	SUCCEEDED	5	5	0	0	0	0	0
Reducer 3	container	SUCCEEDED	1	1	0	0	0	0	0

```
VERTICES: 03/03 [=====>>] 100% ELAPSED TIME: 73.95 s
OK
category      total_product
8594817
appliances    61735
stationery    26721
furniture     23604
apparel 18232
accessories   12929
sport         2
Time taken: 74.596 seconds, Fetched: 7 row(s)
```

There are 8594817 total number of products.

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

```
SELECT brand, SUM(price) as Total_sales
FROM store_1
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 store_1
> WHERE event_type = 'purchase' AND brand!= ''
> GROUP BY brand
> ORDER BY Total_sales DESC
> LIMIT 1;
Query ID = hadoop_20220128135815_daaa97fa-1d81-4de7-a647-944cc9809d8e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)

-----
      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: 24.63 s
-----
OK
brand    total_sales
runail  148296.74000000622
Time taken: 25.536 seconds, Fetched: 1 row(s)
```

runail has the maximum number of sales in October and November combined which has collected 148296.74 number of sales.

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

```
WITH Brand_sales AS (
SELECT brand,
SUM(CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS
Oct_revenue,
SUM(CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS
Nov_revenue
FROM store_1
WHERE event_type = 'purchase' AND
date_format(event_time, 'MM') IN ('10', '11')
GROUP BY brand
)
SELECT brand, Oct_revenue, Nov_revenue, (Nov_revenue - Oct_revenue) AS
Difference_Of_revenue
FROM Brand_sales
WHERE (Nov_revenue-Oct_revenue) > 0
ORDER BY Difference_Of_revenue DESC;
```

```

hive> WITH Brand_sales AS (
  > SELECT brand,
  > SUM(CASE WHEN date_format(event_time, 'MM')=10 THEN price ELSE 0 END) AS Oct_revenue,
  > SUM(CASE WHEN date_format(event_time, 'MM')=11 THEN price ELSE 0 END) AS Nov_revenue
  > FROM store_1
  > WHERE event_type = 'purchase' AND
  > date_format(event_time, 'MM') IN ('10', '11')
  > GROUP BY brand
  > )
  > SELECT brand, Oct_revenue, Nov_revenue, (Nov_revenue - Oct_revenue) AS Difference_Of_revenue
  > FROM Brand_sales
  > WHERE (Nov_revenue-Oct_revenue) > 0
  > ORDER BY Difference_Of_revenue DESC;
Query ID = hadoop_20220128135847_945f7169-9d92-48b5-9712-a58b117d5d20
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)

```

	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: 44.74 s

```

OK
brand  oct_revenue  nov_revenue  difference_of_revenue
474655.96999994945  619492.2199999489  144836.24999999942
grattol 35445.53999999818  71466.44000000754  36020.90000000936
uno 35302.02999999994  51039.750000000175  15737.720000000234
lianail 5892.840000000006  16394.24000000062  10501.400000000614
ingarden 23161.390000000323  33566.21000000024  10404.819999999916
strong 29196.630000000012  38671.26999999993  9474.63999999992
jessnail 26287.83999999997  33345.22999999998  7057.389999999985
cosmoprofi 8322.810000000038  14536.990000000143  6214.180000000106
polarus 6013.71999999999  11371.92999999997  5358.209999999997
runail 71538.32999999075  76758.40999999596  5220.0800000005211
freedecor 3421.7800000000334  7671.799999999905  4250.019999999871
staleks 8519.730000000027  11869.580000000027  3349.8500000000004
bpw.style 11572.150000000098  14837.4400000002  3265.2900000001027
lovely 8704.379999999957  11939.060000000001  3234.6800000000044
marathon 7280.749999999997  10273.099999999997  2992.3499999999995
haruyama 9390.6900000000088  12352.9100000000198  2962.2200000001103
yoko 8756.909999999993  11707.880000000005  2950.9700000000012
italwax 21940.239999999816  24797.239999999816  2857.0
benovy 409.62000000000023  3259.9700000000057  2850.3500000000054
kaypro 881.3399999999999  3268.7000000000007  2387.3600000000006
estel 21756.749999999924  24142.670000000012  2385.9200000001947
concept 11032.140000000043  13380.400000000027  2348.259999999984
kapous 11927.159999999902  14093.079999999994  2165.920000000093
f.o.x 6624.2300000000092  8577.280000000022  1953.0499999999302
masura 31261.97999999773  33058.46999999869  1796.4900000000962
milv 3904.9399999999728  5642.009999999998  1737.0700000000256
beautix 10493.94999999998  12222.94999999997  1729.0000000000164
artex 2730.639999999998  4327.250000000007  1596.6100000000092
domix 10472.049999999988  12009.170000000062  1537.1200000000736
shik 3341.200000000002  4839.720000000004  1498.5200000000018
smart 4457.259999999988  5902.139999999993  1444.8800000000047
roubloff 3491.359999999993  4913.769999999995  1422.4100000000067
levrana 2243.559999999968  3664.100000000054  1420.5400000000086
oniq 8425.410000000001  9841.650000000061  1416.2400000000507
irisk 45591.959999998624  46946.03999999895  1354.0800000003292
severina 4775.880000000005  6120.480000000001  1344.600000000005
joico 705.52 2015.100000000004  1309.5800000000004
zeitun 708.659999999999  2009.630000000001  1300.9700000000012
beauty-free 554.1700000000002  1782.860000000001  1228.690000000001
swarovski 1887.9299999999814  3043.159999999984  1155.2300000000025
de.lux 1659.6999999999678  2775.5099999999807  1115.810000000013
metzger 5373.450000000001  6457.160000000009  1083.7100000000082
markell 1768.749999999977  2834.430000000044  1065.6800000000067
sanoto 157.14 1209.679999999998  1052.54
nagaraku 4369.740000000019  5327.680000000017  957.9399999999978
ecolab 262.85 1214.299999999997  951.449999999997
art-visage 2092.710000000001  2997.8000000000056  905.0899999999956
levissime 2227.500000000064  3085.3099999999804  857.8099999999974
missha 1293.829999999992  2150.28 856.450000000001
solomeya 1899.6999999999957  2685.799999999994  786.0999999999981
rosi 3077.03999999999  3841.5599999999863  764.5199999999963
refectocil 2716.1800000000076  3475.5800000000095  759.4000000000019
kaaral 4412.430000000004  5086.070000000001  673.6400000000058
kosmekka 1181.4399999999996  1813.3700000000008  631.9300000000012
kinetics 6334.250000000048  6945.260000000029  611.0099999999811
browxenna 14331.369999999994  14916.729999999952  585.3599999999587
airnails 5118.9000000000095  5691.5200000000056  572.6199999999608
uskusi 5142.269999999936  5690.30999999998  548.0400000000445
coifin 903.0 1428.489999999998  525.4899999999998

```

```
s.care 412.68 913.0699999999999 500.38999999999993
limoni 1308.90000000000005 1796.6 487.69999999999936
matrix 3243.2500000000002 3726.7400000000007 483.4899999999989
gehwol 1089.07 1557.6799999999994 468.60999999999945
greymy 29.21 489.49 460.28000000000003
bioaqua 942.89 1398.1199999999994 455.22999999999945
farmavita 837.37000000000001 1291.9699999999998 454.5999999999997
sophin 1062.32000000000006 1515.5200000000004 453.1999999999998
yu-r 271.41 673.71 402.3
kiss 421.55000000000001 817.3299999999992 395.7799999999991
naomi 0.0 388.99999999999994 388.99999999999994
lador 2083.61000000000024 2471.5299999999996 387.91999999999937
ellips 245.84999999999997 606.0399999999998 360.18999999999999
jas 3318.96 3657.43000000000026 338.47000000000025
lowence 242.84000000000003 567.7499999999999 324.90999999999985
nitrile 847.2799999999999 1162.6799999999999 315.4
shary 871.9599999999998 1176.4899999999999 304.52999999999993
kims 330.04 632.04000000000001 302.00000000000006
happyfons 801.92000000000004 1091.59000000000006 289.67000000000002
kocostar 310.84999999999997 594.93000000000004 284.08000000000004
insight 1443.70000000000007 1721.96000000000005 278.25999999999976
candy 534.9599999999998 799.3799999999994 264.41999999999996
bluesky 10307.2400000000105 10565.5300000000063 258.289999999999572
beaugreen 511.51000000000016 768.3499999999999 256.83999999999975
protokeratin 201.24999999999997 456.78999999999996 255.54
trind 298.07000000000001 542.96000000000003 244.890000000000016
entity 479.710000000000225 719.2599999999995 239.54999999999728
skinlite 651.94 889.0199999999998 237.0799999999997
provoc 827.99000000000004 1063.82000000000024 235.830000000000021
fedua 52.38 263.81 211.43
ecocraft 41.160000000000004 241.95000000000005 200.790000000000005
keen 236.35 435.62000000000006 199.27000000000007
mane 66.78999999999999 260.26 193.47
freshbubble 318.70000000000005 502.34000000000003 183.64
matreshka 0.0 182.67000000000007 182.67000000000007
chi 358.94000000000005 538.61 179.66999999999996
cristalinas 427.62999999999999 584.95 157.320000000000016
farmona 1692.46 1843.43 150.97000000000003
latinoil 249.52000000000004 384.59000000000015 135.07000000000001
miskin 158.04000000000002 293.06999999999994 135.02999999999992
elizavecca 70.53 204.3 133.77
nefertiti 233.52000000000007 366.64 133.11999999999992
finish 98.38 230.38 132.0
igrobeauty 513.66000000000001 645.07000000000008 131.40999999999985
dizao 819.13000000000001 945.51000000000023 126.380000000000125

cutrin 299.37000000000006 367.62 68.24999999999994
laboratorium 246.5 312.52000000000001 66.02000000000001
inm 288.01999999999997 351.21 63.190000000000011
dewal 0.0 61.29 61.29
marutaka-foot 49.21999999999999 109.33000000000001 60.11000000000002
kares 0.0 59.45 59.45
profhenna 679.2299999999994 736.8499999999997 57.620000000000023
koelcia 55.5 112.75 57.25
balbcare 155.32999999999996 212.37999999999997 57.050000000000001
elskin 251.09000000000043 307.65000000000003 56.55999999999989
foamie 35.04 80.49 45.449999999999996
ladykin 125.64999999999999 170.57 44.92
likato 296.0599999999999 340.96999999999997 44.910000000000008
mavala 409.04000000000001 446.32000000000005 37.27999999999997
vilenta 197.59999999999994 231.20999999999992 33.609999999999985
beautyblender 78.74000000000001 109.41 30.669999999999987
biore 60.650000000000006 90.31 29.659999999999997
orly 902.3799999999997 931.0899999999995 28.70999999999981
estelare 444.81000000000006 471.8699999999999 27.05999999999983
profepil 93.360000000000001 118.02000000000002 24.660000000000001
blixz 38.95 63.39999999999999 24.44999999999999
binacil 0.0 24.259999999999998 24.259999999999998
godefroy 401.22000000000001 425.12 23.89999999999992
glysolid 69.72999999999996 91.58999999999999 21.860000000000028
veraclara 50.10999999999985 71.20999999999998 21.099999999999994
juno 0.0 21.08 21.08
kamill 63.010000000000005 81.49000000000001 18.480000000000004
treaclemoon 163.37 181.48999999999998 18.119999999999976
supertan 50.37 66.51 16.140000000000008
barbie 0.0 12.39 12.39
deoproce 316.84000000000001 329.17000000000001 12.329999999999984
rasyan 18.799999999999997 28.939999999999998 10.14
fly 17.14 27.169999999999998 10.029999999999998
tertio 236.160000000000025 245.80000000000013 9.6399999999999873
jaguar 1102.11 1110.6499999999999 8.539999999999964
soleo 204.19999999999998 212.52999999999972 8.329999999999927
neoleor 43.41 51.7 8.290000000000006
moyou 5.71 10.280000000000001 4.570000000000001
bodyton 1376.34000000000001 1380.6400000000006 4.3000000000000409
skinity 8.88 12.440000000000001 3.5600000000000005
helloganic 0.0 3.1 3.1
grace 100.92000000000002 102.60999999999999 1.6899999999999693
cosima 20.23 20.929999999999996 0.699999999999957
ovale 2.54 3.1 0.56
Time taken: 45.378 seconds, Fetched: 161 row(s)
```


There are 161 brands which has increased their sales from October to November.

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.

```
SELECT user_id, SUM(price) as Total_spend
FROM store_1
WHERE event_type = 'purchase'
GROUP BY user_id
ORDER BY Total_spend DESC
LIMIT 10;
```

```
hive> SELECT user_id, SUM(price) as Total_spend
> FROM store_1
> WHERE event_type = 'purchase'
> GROUP BY user_id
> ORDER BY Total_spend DESC
> LIMIT 10;
Query ID = hadoop_20220128140006_a751f084-3ea5-4a1a-ba17-3fe0cfc40bae
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1643373050143_0003)

-----
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: 28.51 s
-----
OK
user_id total_spend
557790271      2715.869999999995
150318419      1645.9699999999996
562167663      1352.8499999999995
531900924      1329.4499999999998
557850743      1295.48
522130011      1185.3900000000003
561592095      1109.7000000000003
431950134      1097.5899999999995
566576008      1056.3600000000001
521347209      1040.9099999999996
Time taken: 29.214 seconds, Fetched: 10 row(s)
```

Cleaning up:

Once the analysis is completed, we should drop the tables.

```
hive> show tables;
OK
tab_name
store
store_1
Time taken: 0.041 seconds, Fetched: 2 row(s)
hive> drop table store;
OK
Time taken: 0.133 seconds
hive> drop table store_1;
OK
Time taken: 0.326 seconds
```

Drop the database.

```
hive> show databases;
OK
database_name
casestudy
default
Time taken: 0.042 seconds, Fetched: 2 row(s)
hive> drop database casestudy;
OK
Time taken: 0.199 seconds
hive> show databases;
OK
database_name
default
Time taken: 0.012 seconds, Fetched: 1 row(s)
```

Terminate your cluster:

After dropping the database, we should terminate the cluster.

The screenshot displays the AWS Management Console interface for an Amazon EMR cluster. The left sidebar shows the navigation menu with 'Amazon EMR' selected. The main content area shows the 'Cluster: Hivecasestudy' page, which is in a 'Terminated' state. A red warning banner at the top states: 'Auto-termination is not available for this account when using this release of EMR.' Below this, the 'Summary' tab is active, displaying the following details:

- ID: j-36XWM77GFVOW7
- Creation date: 2022-01-28 17:52 (UTC+5:30)
- End date: 2022-01-28 20:08 (UTC+5:30)
- Elapsed time: 2 hours, 15 minutes
- After last step completes: Cluster waits
- Termination protection: Off
- Tags: --
- Master public DNS: ec2-3-90-59-29.compute-1.amazonaws.com
- 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-815937943894-us-east-

Cluster is terminated.

