HIVE CASE STUDY (Retail Store)
(DSC41)

Submitted by : SaiTeja , Kailash

PROBLEM STATEMENT:

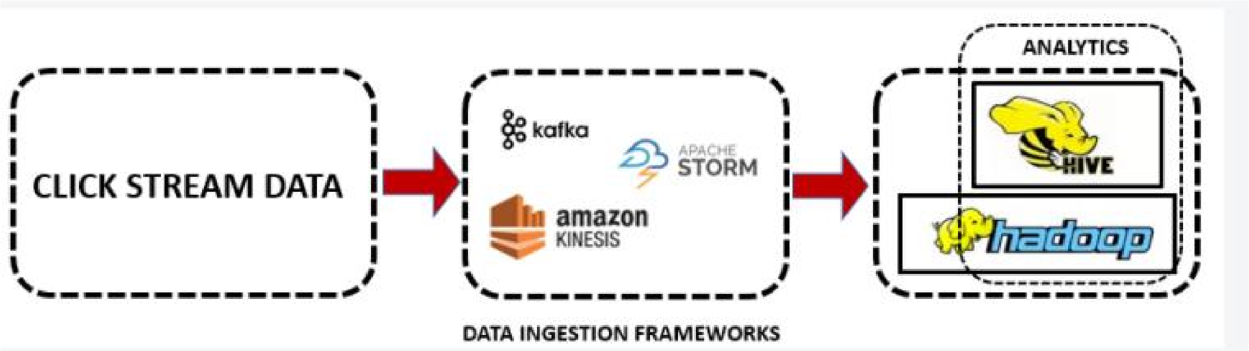
With online sales gaining popularity, tech companies are exploring ways to improve their sales by analyzing customer behavior and gaining insights about product trends. Furthermore, the websites make it easier for customers to find the products they require without much scavenging. Needless to say, the role of big data analysts is among the most sought-after job profiles of this decade. Therefore, as part of this assignment, we will be challenging you, as a big data analyst, to extract data and gather insights from a real-life data set of an e-commerce company.

OBJECTIVE:

The aim is to extract data and gather insights from a real-life data set of an e-commerce Company.

DATA:

The data used for this assignment is a public clickstream dataset of a cosmetic store. The clickstream data contains all the logs as to how one navigated through the e-commerce website. It also contains other data such as customer time spent on every page, a number of
clicks made, adding items to the cart, customer id, etc.



OVERVIEW OF STEPS:

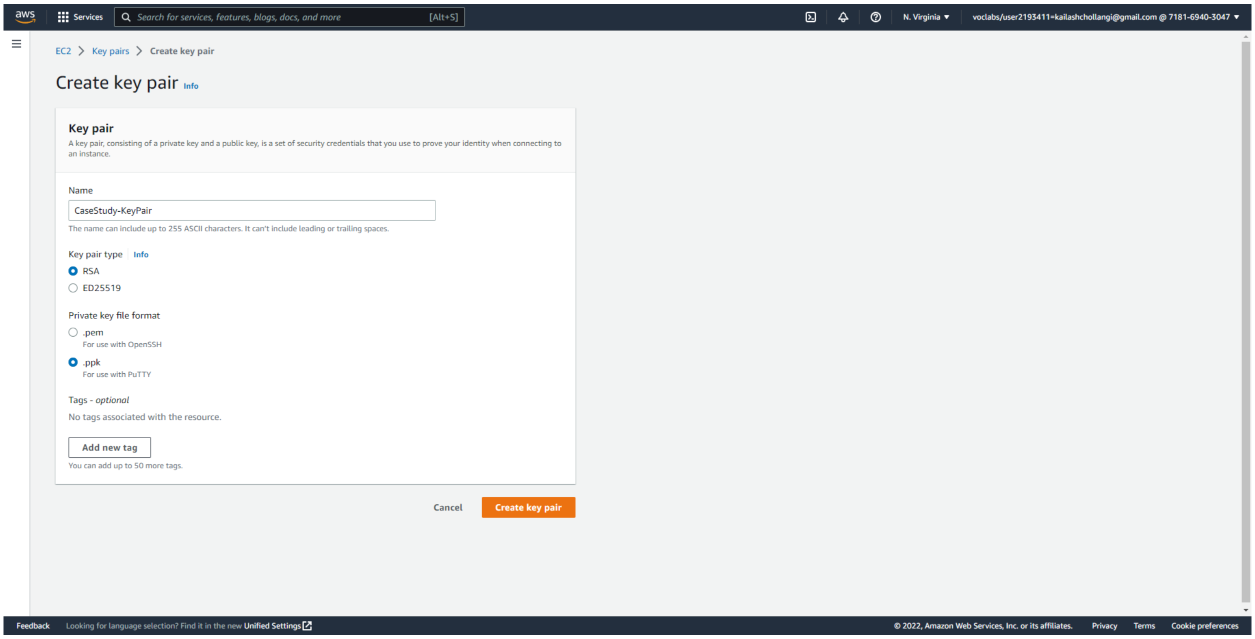
➢ Copying the data set into HDFS:
- Launch an EMR cluster that utilizes the hive services, and
- Move the data from S3 bucket into the HDFS

➢ 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 effectively as possible
- Show the improvement in performance after optimizing
- Run hive queries to answer the given questions.

➢ Cleaning up:
- Drop your database and
- Terminate your cluster

KEY-PAIR CREATION:

Creating Key Pair with ppk file format directly to use with putty


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia ▾
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047 ▾
EC2 > Key pairs > Create key pair
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create key pair Info

Key pair

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

Name

CaseStudy-KeyPair

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

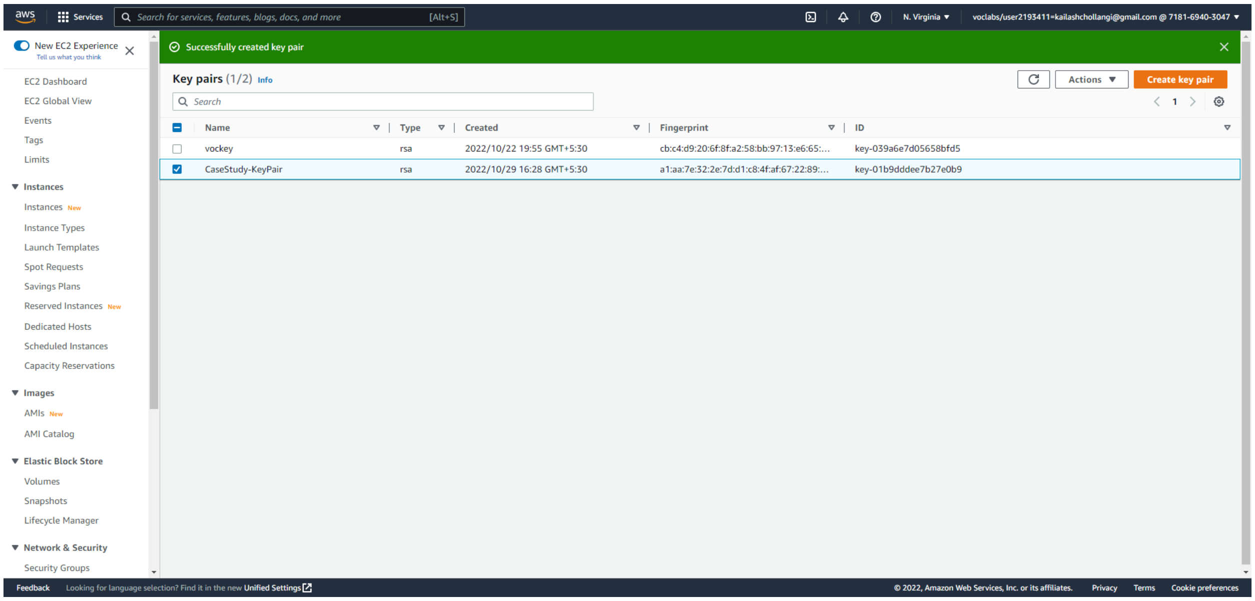
Tags - *optional*

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

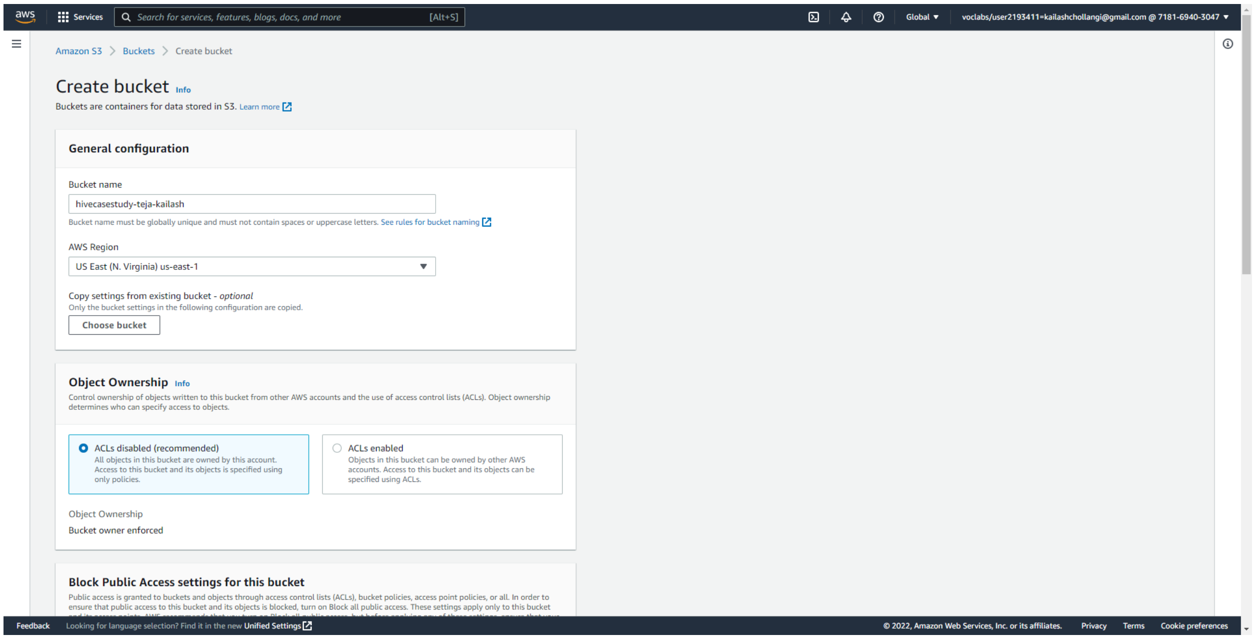
Key pairs (1/2) Info

	Name	Type	Created	Fingerprint	ID
□	vockey	rsa	2022/10/22 19:55 GMT+5:30	cb:c4:d9:20:6f:8f:a2:58:bb:97:13:e6:65:...	key-039a6e7d05658bfd5
☑	CaseStudy-KeyPair	rsa	2022/10/29 16:28 GMT+5:30	a1:aa:7e:32:2e:7d:d1:c8:4f:af:67:22:89:...	key-01b9dddee7b27e0b9

CaseStudy-KeyPair successfully created and downloaded (refer above screenshot)

S3 BUCKET:

To Store the data – Click on “Create Bucket”


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▼
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047 ▼
Amazon S3 > Buckets > Create bucket
Info
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create bucket

Buckets are containers for data stored in S3. Learn more

General configuration

Bucket name: hivecasestudy-teja-kailash

Bucket name must be globally unique and must not contain spaces or uppercase letters. See rules for bucket naming

AWS Region: US East (N. Virginia) us-east-1 ▼

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

Object Ownership

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

● ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

○ ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

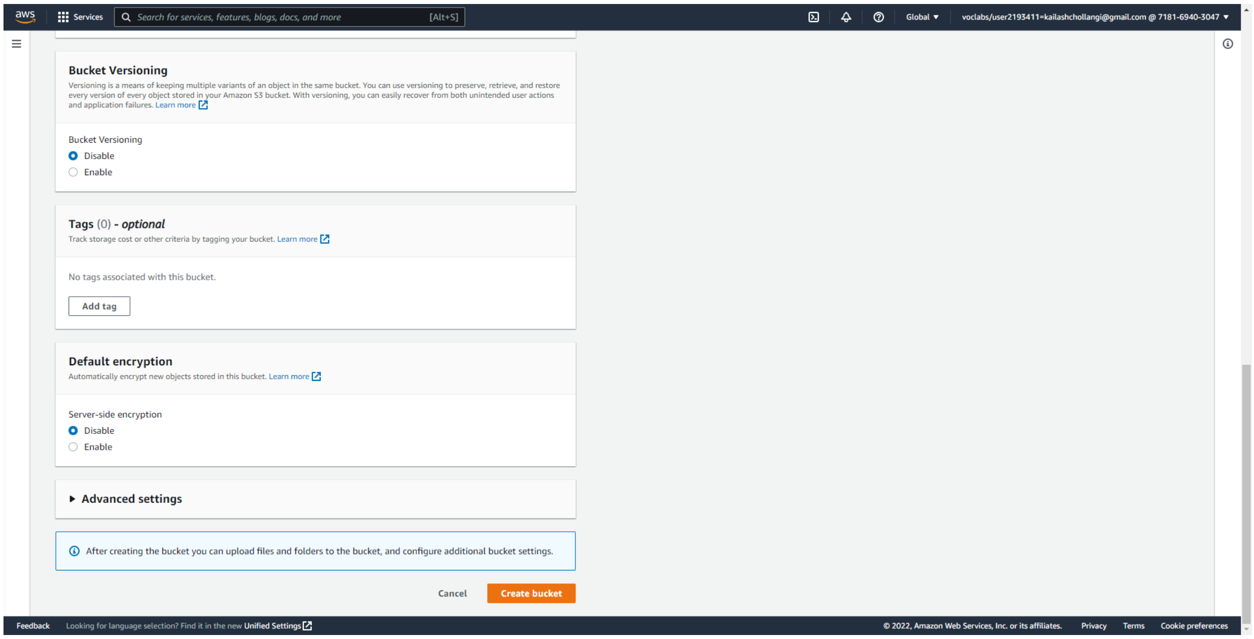
Object Ownership

Bucket owner enforced

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket

Creating “**hivecasestudy-teja-kailash**” with all default options


aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
Global ▼
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047 ▼
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning
● Disable
○ Enable

Tags (0) - *optional*

Track storage cost or other criteria by tagging your bucket. Learn more

No tags associated with this bucket.

Add tag

Default encryption

Automatically encrypt new objects stored in this bucket. Learn more

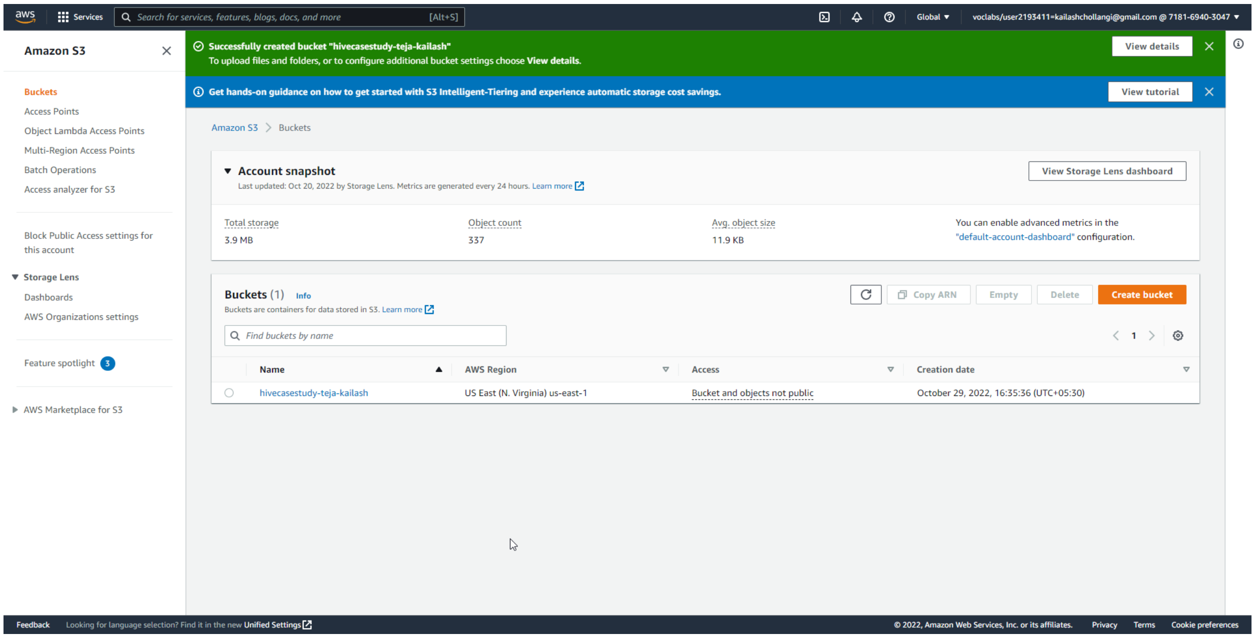
Server-side encryption
● Disable
○ Enable

▸ Advanced settings

After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Cancel   Create bucket

Bucket Successfully created.


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
Access analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
View details
View tutorial
Amazon S3 > Buckets
View Storage Lens dashboard
Copy ARN
Empty
Delete
Create bucket
Info
Find buckets by name
1
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Successfully created bucket "hivecasestudy-teja-kailash"
To upload files and folders, or to configure additional bucket settings choose **View details.**

Get hands-on guidance on how to get started with S3 Intelligent-Tiering and experience automatic storage cost savings.

Account snapshot

Last updated: Oct 20, 2022 by Storage Lens. Metrics are generated every 24 hours. Learn more

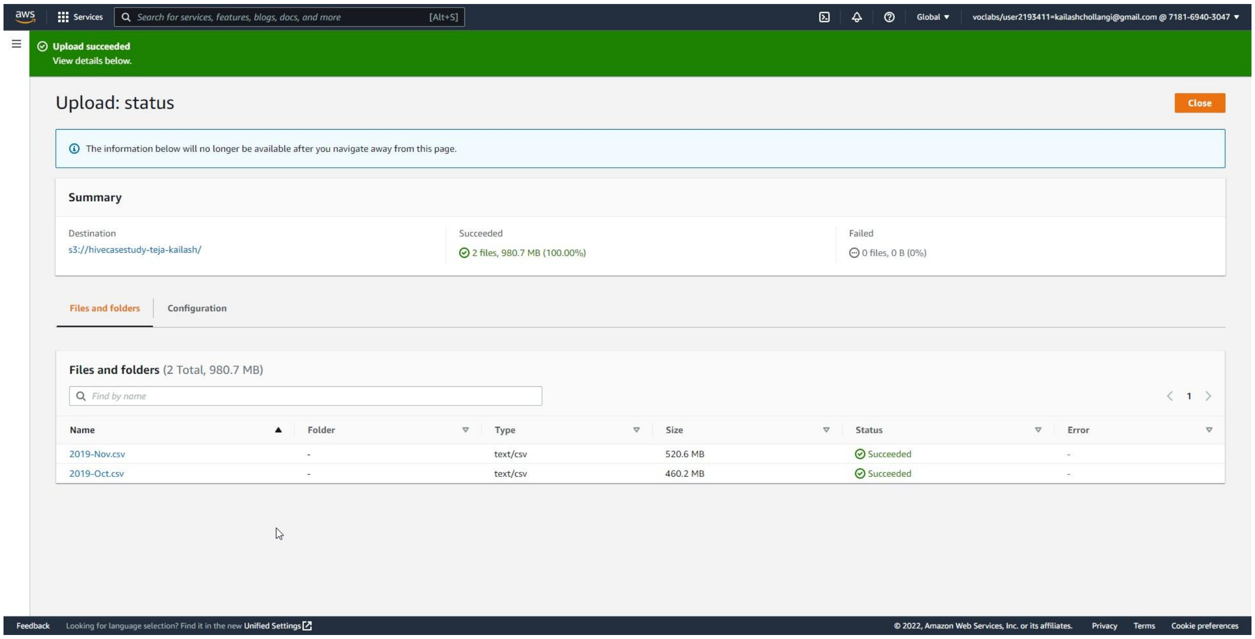
Total storage	Object count	Avg. object size	You can enable advanced metrics in the "default-account-dashboard" configuration.
3.9 MB	337	11.9 KB	

Buckets (1)

Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	hivecasestudy-teja-kailash	US East (N. Virginia) us-east-1	Bucket and objects not public	October 29, 2022, 16:35:36 (UTC+05:30)

Successfully uploaded the 2019 October and 2019 November csv files to S3 bucket


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047
Close
Files and folders
Configuration
Find by name
1
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

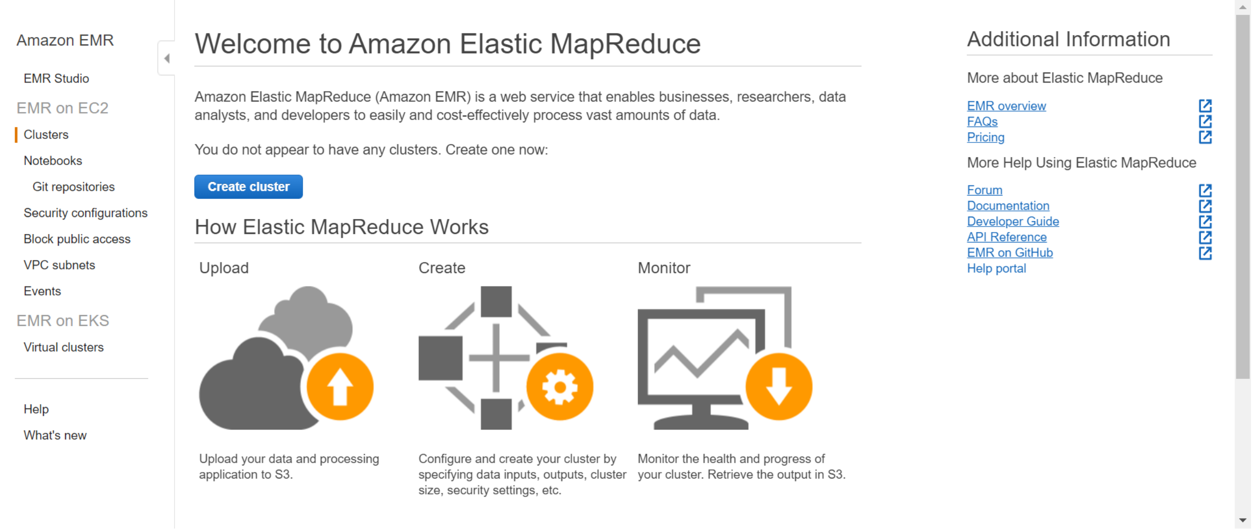
Destination	Succeeded	Failed
s3://hivecasestudy-teja-kailash/	2 files, 980.7 MB (100.00%)	0 files, 0 B (0%)

Files and folders (2 Total, 980.7 MB)

Name	Folder	Type	Size	Status	Error
2019-Nov.csv	-	text/csv	520.6 MB	Succeeded	-
2019-Oct.csv	-	text/csv	460.2 MB	Succeeded	-

EMR CLUSTER CREATION:

Click on “Create cluster” button to create the EMR cluster



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

Welcome to Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now:

Create cluster

How Elastic MapReduce Works

Upload

Upload your data and processing application to S3.

Create

Configure and create your cluster by specifying data inputs, outputs, cluster size, security settings, etc.

Monitor

Monitor the health and progress of your cluster. Retrieve the output in S3.

Additional Information

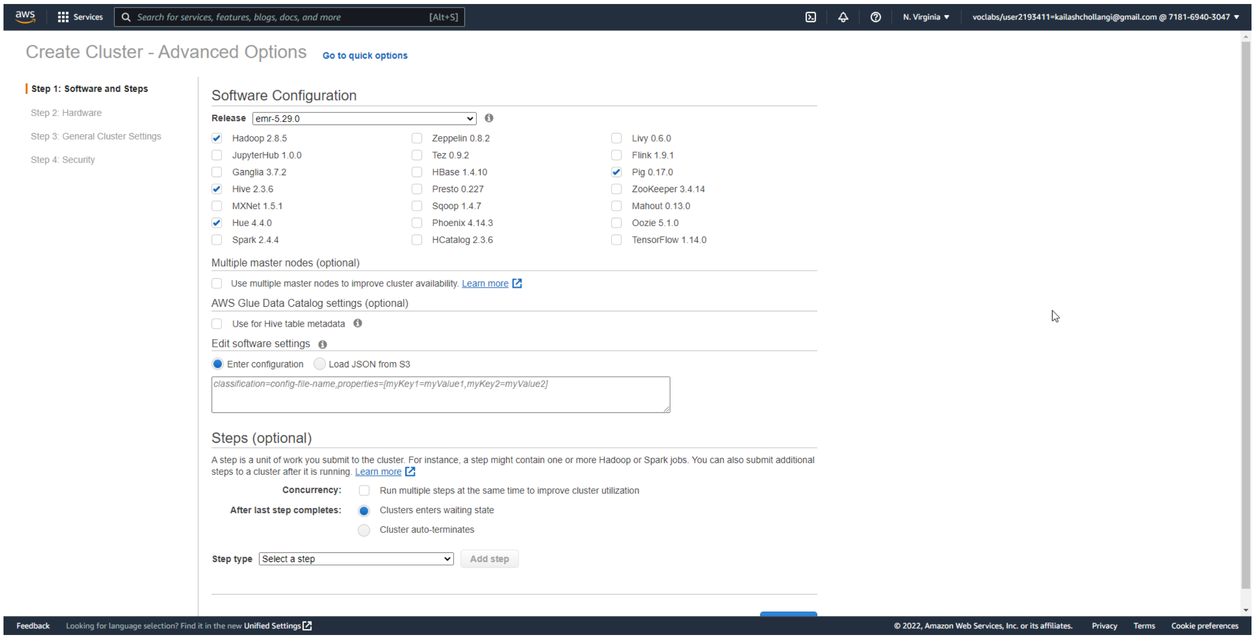
More about Elastic MapReduce

EMR overview
FAQs
Pricing

More Help Using Elastic MapReduce

Forum
Documentation
Developer Guide
API Reference
EMR on GitHub
Help portal

Creating cluster with advanced options


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047

Create Cluster - Advanced Options   Go to quick options

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

Software Configuration

Release: emr-5.29.0

☑ 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
☐ TensorFlow 1.14.0

Multiple master nodes (optional)

☐ Use multiple master nodes to improve cluster availability. Learn more

AWS Glue Data Catalog settings (optional)

☐ Use for Hive table metadata

Edit software settings

● Enter configuration ○ Load JSON from S3

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

Steps (optional)

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

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

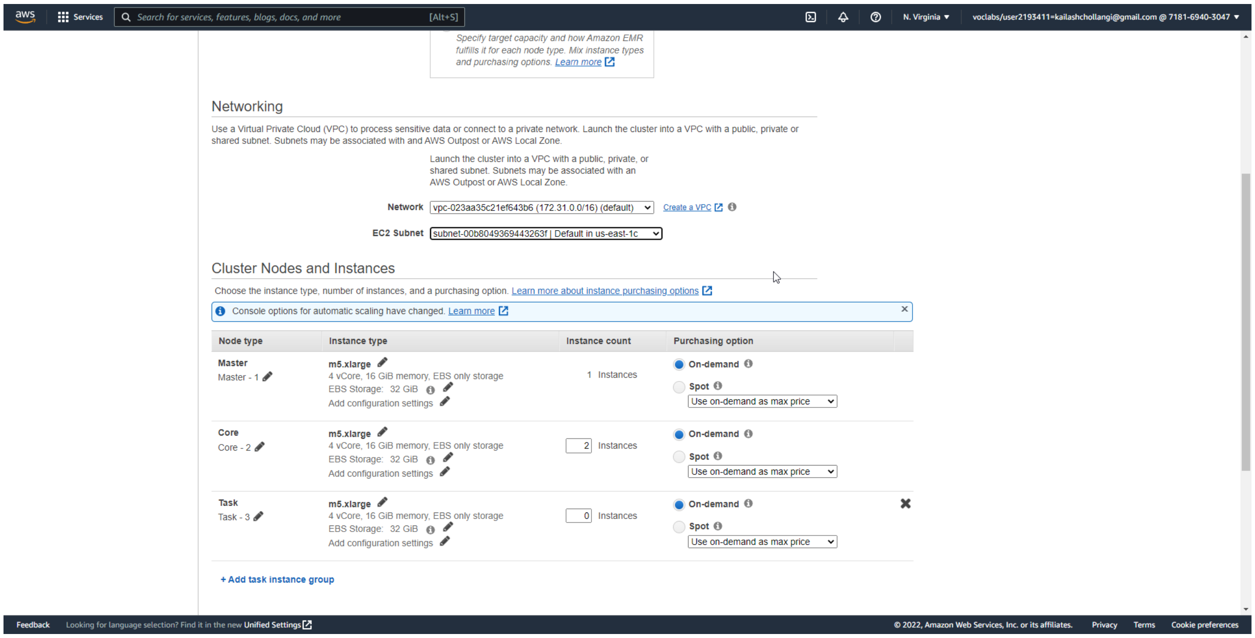
After last step completes: ● Clusters enters waiting state
○ Cluster auto-terminates

Step type Select a step   Add step

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Software and Steps page: Changed the Release from emr-5.36.0 to “**emr-5.29.0**”

Hardware page: Changing the Master and Core nodes from m5.xlarge to “**m4.large**”


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia ▾
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047 ▾

Specify target capacity and how Amazon EMR fulfills it for each node type. Mix instance types and purchasing options. Learn more

Networking

Use a Virtual Private Cloud (VPC) to process sensitive data or connect to a private network. Launch the cluster into a VPC with a public, private or shared subnet. Subnets may be associated with and AWS Outpost or AWS Local Zone.

Launch the cluster into a VPC with a public, private, or shared subnet. Subnets may be associated with an AWS Outpost or AWS Local Zone.

Network vpc-023aa35c21ef643b6 (172.31.0.0/16) (default) Create a VPC

EC2 Subnet subnet-00b8049369443263f | Default in us-east-1c

Cluster Nodes and Instances

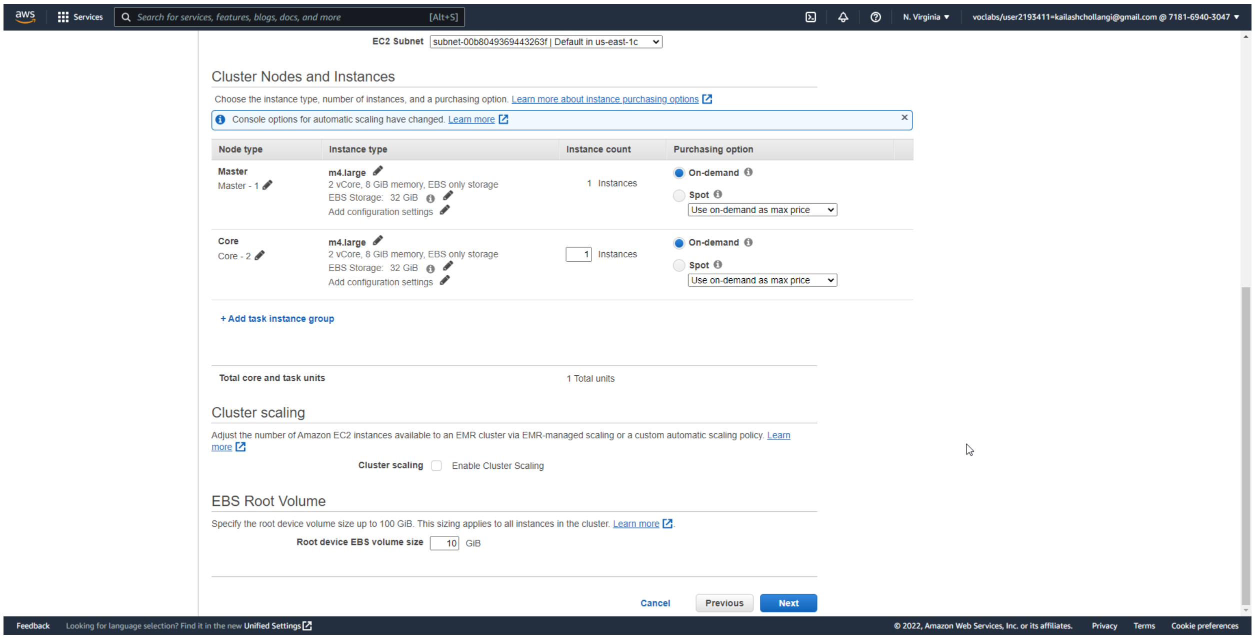
Choose the instance type, number of instances, and a purchasing option. Learn more about instance purchasing options

Console options for automatic scaling have changed. Learn more

Node type	Instance type	Instance count	Purchasing option
Master Master - 1	**m5.xlarge** 4 vCore, 16 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	**m5.xlarge** 4 vCore, 16 GiB memory, EBS only storage EBS Storage: 32 GiB Add configuration settings	2 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

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia ▾
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047 ▾

EC2 Subnet subnet-00b8049369443263f | Default in us-east-1c

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

+ Add task instance group

Total core and task units 1 Total units

Cluster scaling

Adjust the number of Amazon EC2 instances available to an EMR cluster via EMR-managed scaling or a custom automatic scaling policy. Learn more

Cluster scaling □ Enable Cluster Scaling

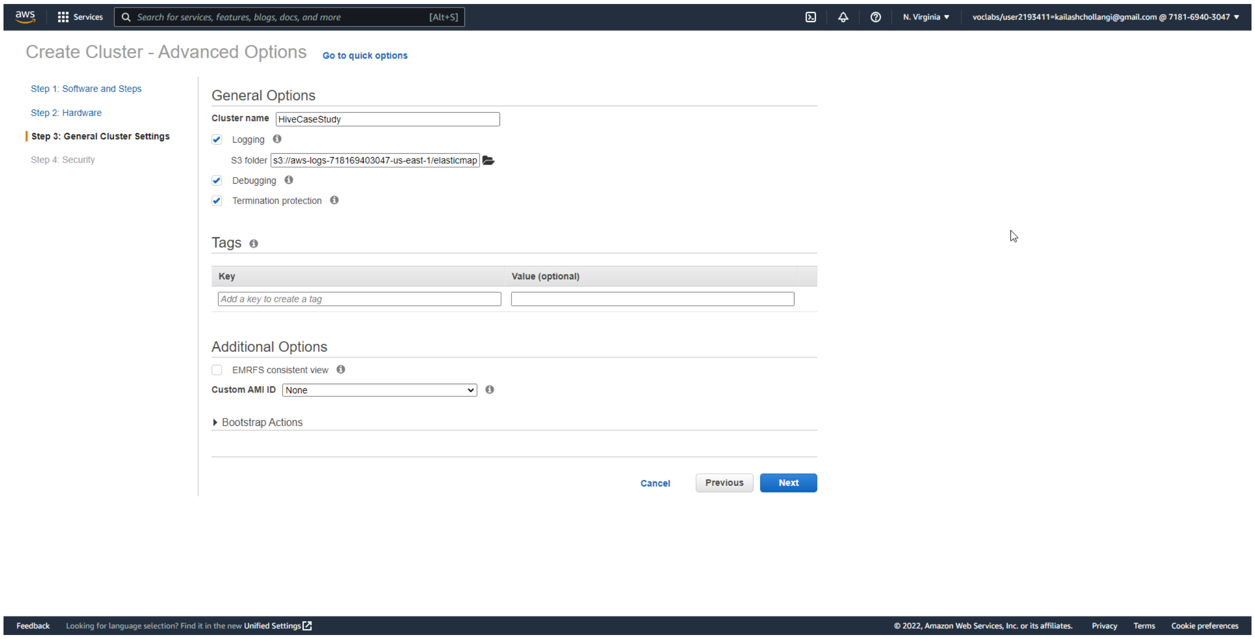
EBS Root Volume

Specify the root device volume size up to 100 GiB. This sizing applies to all instances in the cluster. Learn more.

Root device EBS volume size 10 GiB

Cancel
Previous
Next
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

General Cluster Settings page: Giving the name to cluster “HiveCaseStudy”


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
Add a key to create a tag
Bootstrap Actions
Cancel
Previous
Next
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create Cluster - Advanced Options

General Options

Cluster name: HiveCaseStudy

☑ Logging

S3 folder: s3://aws-logs-718169403047-us-east-1/elasticmap

☑ Debugging

☑ Termination protection

Tags

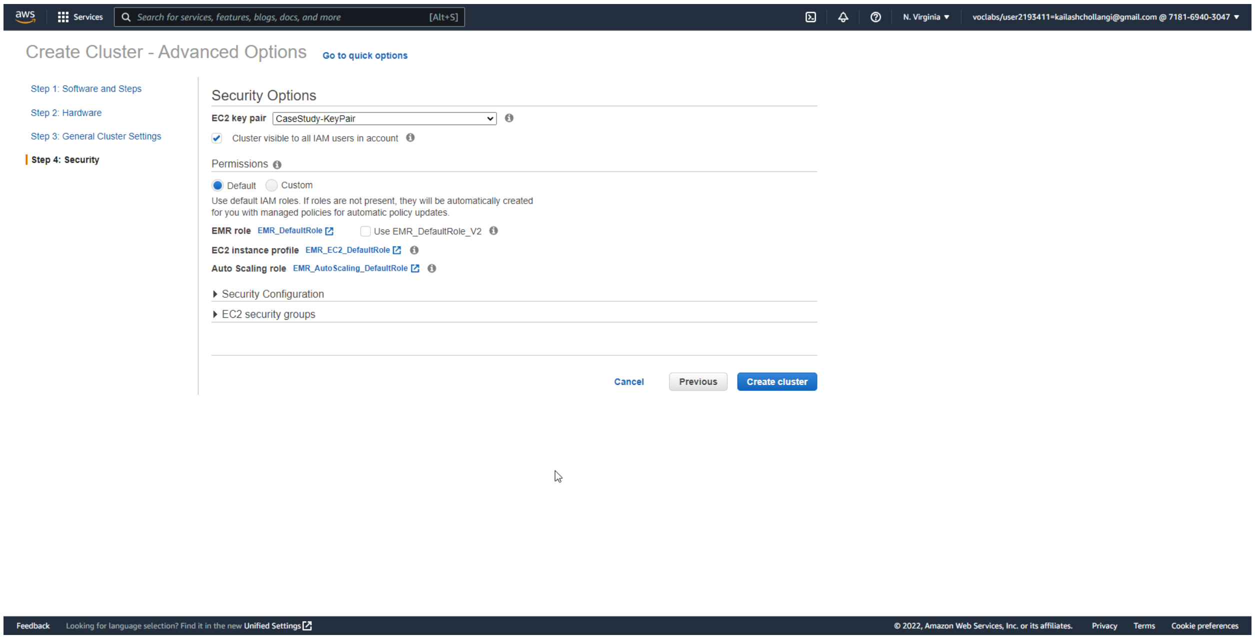
Key	Value (optional)

Additional Options

☐ EMRFS consistent view

Custom AMI ID: None

Security page: changing the EC2 key pair option to our created key pair – “**CaseStudy-KeyPair**”


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047
Go to quick options
Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security
Security Configuration
EC2 security groups
Cancel
Previous
Create cluster
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create Cluster - Advanced Options

Security Options

EC2 key pair: CaseStudy-KeyPair

☑ Cluster visible to all IAM users in account

Permissions

● Default ○ Custom

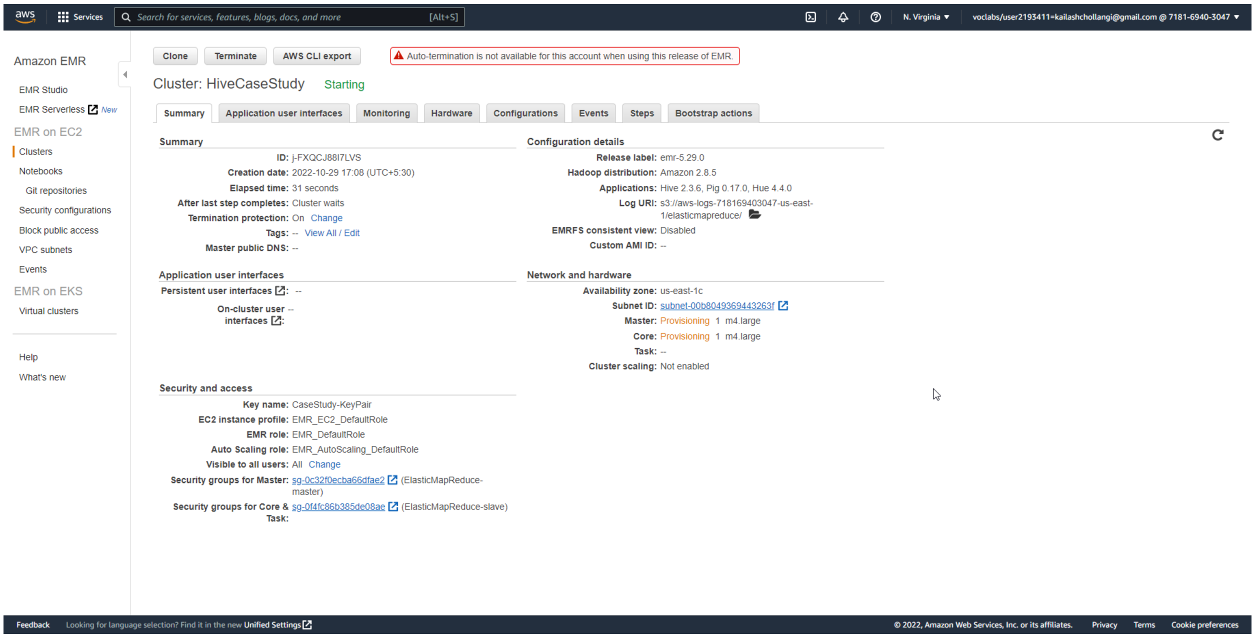
Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role: EMR_DefaultRole ☐ Use EMR_DefaultRole_V2

EC2 instance profile: EMR_EC2_DefaultRole

Auto Scaling role: EMR_AutoScaling_DefaultRole

Click on “Create Cluster” button



Auto-termination is not available for this account when using this release of EMR.

Cluster: HiveCaseStudy Starting

Summary

ID: j-FXQCJ88I7LVS
Creation date: 2022-10-29 17:08 (UTC+5:30)
Elapsed time: 31 seconds
After last step completes: Cluster waits
Termination protection: On 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-718169403047-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

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

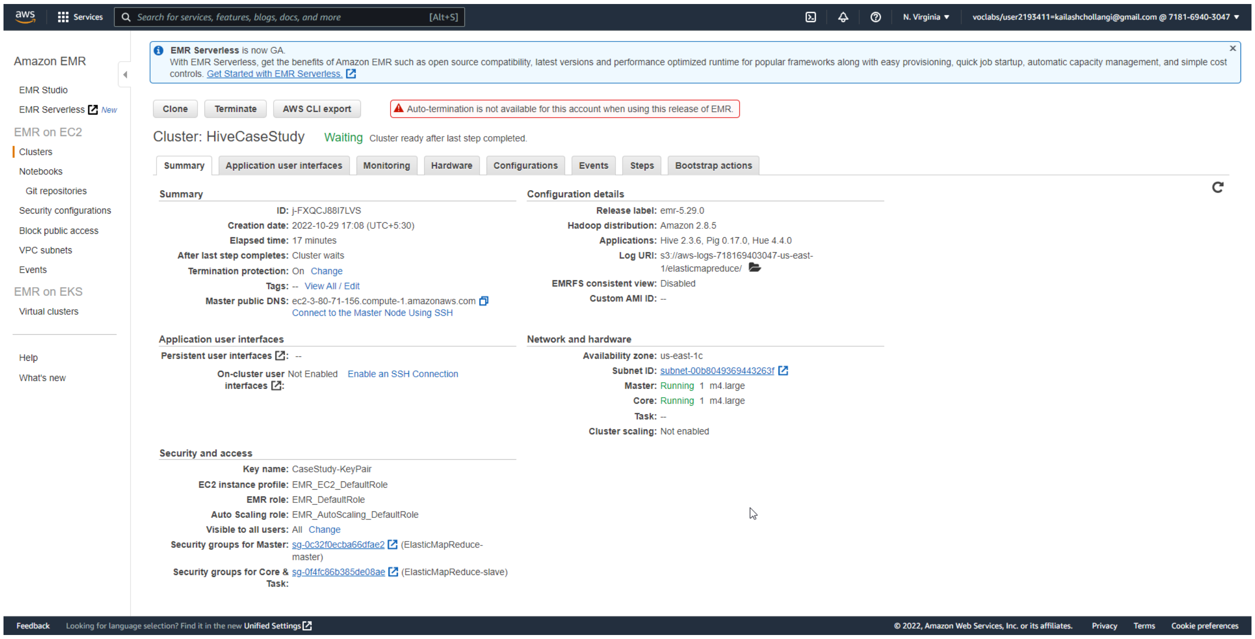
Network and hardware

Availability zone: us-east-1c
Subnet ID: subnet-00b8049369443263f
Master: Provisioning 1 m4.large
Core: Provisioning 1 m4.large
Task: --
Cluster scaling: Not enabled

Security and access

Key name: CaseStudy-KeyPair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)
Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)

Cluster is ready with status “Waiting”



EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.

Auto-termination is not available for this account when using this release of EMR.

Cluster: HiveCaseStudy Waiting Cluster ready after last step completed.

Summary

ID: j-FXQCJ88I7LVS
Creation date: 2022-10-29 17:08 (UTC+5:30)
Elapsed time: 17 minutes
After last step completes: Cluster waits
Termination protection: On Change
Tags: -- View All / Edit
Master public DNS: ec2-3-80-71-156.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-718169403047-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

Persistent user interfaces: --
On-cluster user interfaces: Not Enabled Enable an SSH Connection

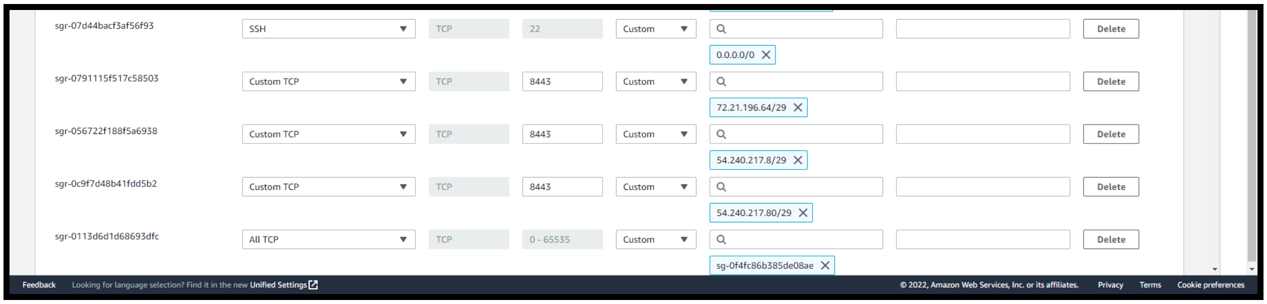
Network and hardware

Availability zone: us-east-1c
Subnet ID: subnet-00b8049369443263f
Master: Running 1 m4.large
Core: Running 1 m4.large
Task: --
Cluster scaling: Not enabled

Security and access

Key name: CaseStudy-KeyPair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)
Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)

- Both the Master and Core nodes are running
- 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.
- Click on edit Inbound rules.
- Add a new rule by selecting SSH and change the IP address to Anywhere



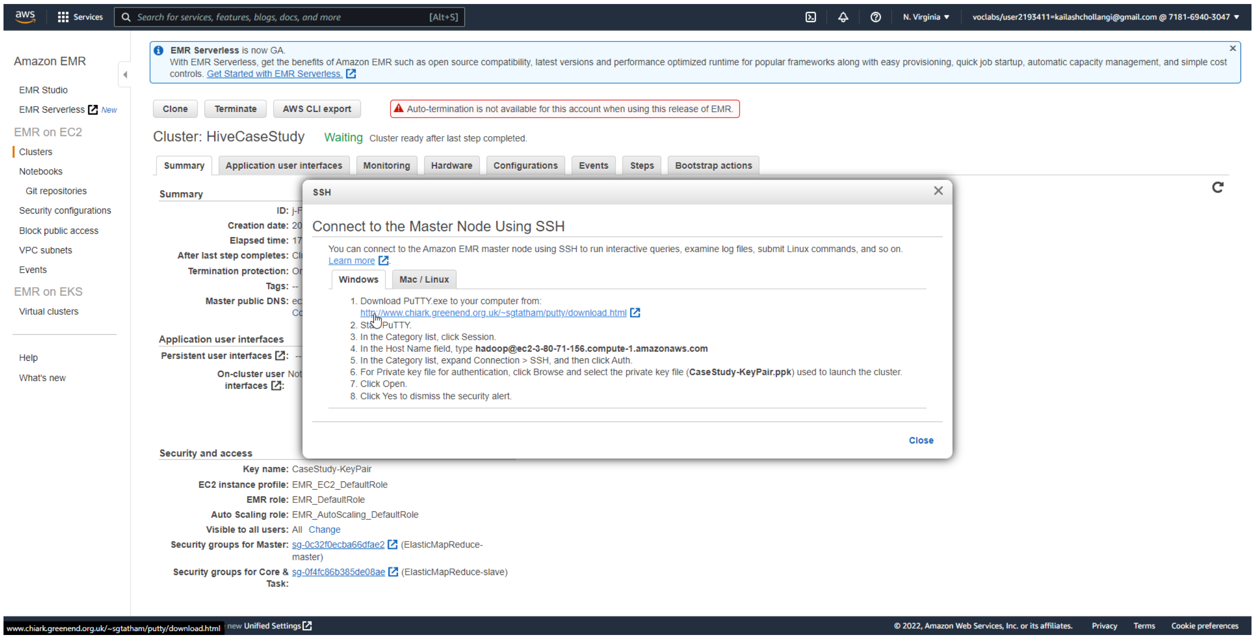
sgr-07d44bacf3af56f93	SSH	TCP	22	Custom	0.0.0.0/0		Delete
sgr-0791115f517c58503	Custom TCP	TCP	8443	Custom	72.21.196.64/29		Delete
sgr-056722f188f5a6938	Custom TCP	TCP	8443	Custom	54.240.217.8/29		Delete
sgr-0c9f7d48b41fdd5b2	Custom TCP	TCP	8443	Custom	54.240.217.80/29		Delete
sgr-0113d6d1d68693dfc	All TCP	TCP	0 - 65535	Custom	sg-0f4fc86b385de08ae		Delete

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

- Then save the SSH rule to the inbound rules

CONNECT TO MASTER NODE:

Open the putty and enter the Host Name as hadoop@ec2-3-80-71-156.compute-1.amazonaws.com and navigate to Connection > SSH > Auth then browse and select the private key, which we created initially.



EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.

Auto-termination is not available for this account when using this release of EMR.

Cluster: HiveCaseStudy Waiting Cluster ready after last step completed.

Summary

ID: j-F
Creation date: 20
Elapsed time: 17
After last step completes: Cl
Termination protection: Or
Tags: --
Master public DNS: ec
Co

Application user interfaces

Persistent user interfaces: --
On-cluster user interfaces: Not

SSH

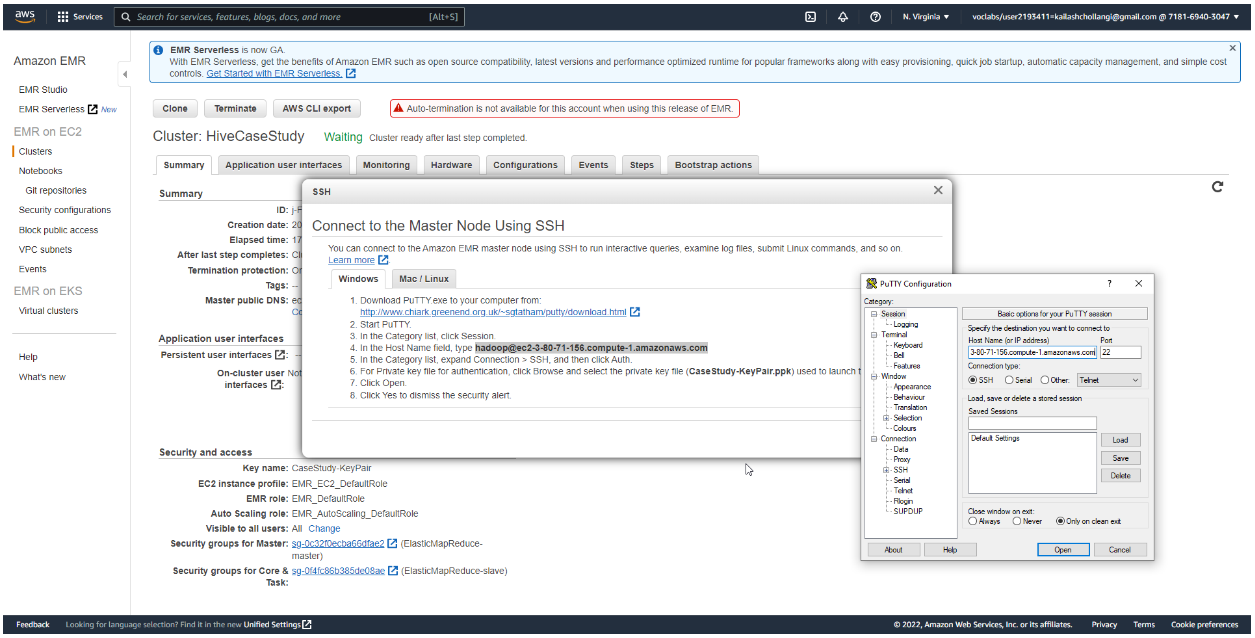
Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. Learn more.

1. Download PuTTY.exe to your computer from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Sta PuTTY.
3. In the Category list, click Session.
4. In the Host Name field, type **hadoop@ec2-3-80-71-156.compute-1.amazonaws.com**
5. In the Category list, expand Connection > SSH, and then click Auth.
6. For Private key file for authentication, click Browse and select the private key file (**CaseStudy-KeyPair.ppk**) used to launch the cluster.
7. Click Open.
8. Click Yes to dismiss the security alert.

Security and access

Key name: CaseStudy-KeyPair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)
Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)



Cluster: HiveCaseStudy Waiting Cluster ready after last step completed.

Auto-termination is not available for this account when using this release of EMR.

SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. Learn more.

Windows | Mac / Linux

1. Download PuTTY.exe to your computer from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Start PuTTY.
3. In the Category list, click Session.
4. In the Host Name field, type **hadoop@ec2-3-80-71-156.compute-1.amazonaws.com**
5. In the Category list, expand Connection > SSH, and then click Auth.
6. For Private key file for authentication, click Browse and select the private key file (**CaseStudy-KeyPair.ppk**) used to launch t
7. Click Open.
8. Click Yes to dismiss the security alert.

PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 3-80-71-156.compute-1.amazonaws.com

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Security and access

Key name: CaseStudy-KeyPair

EC2 instance profile: EMR_EC2_DefaultRole

EMR role: EMR_DefaultRole

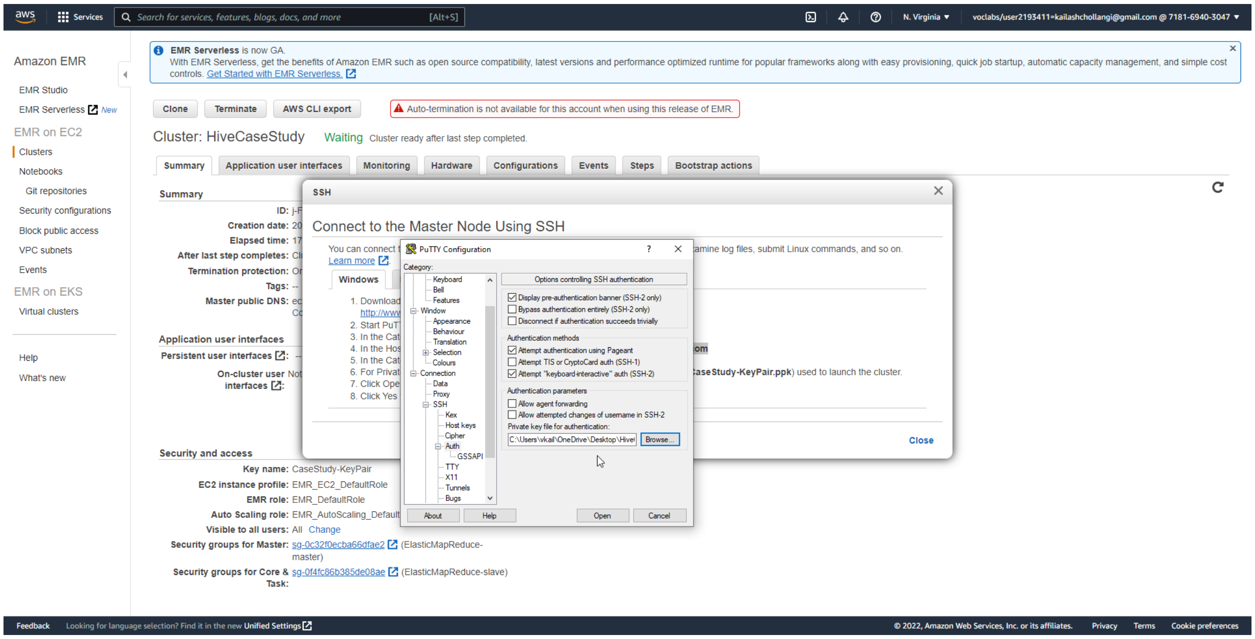
Auto Scaling role: EMR_AutoScaling_DefaultRole

Visible to all users: All Change

Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)

Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)

aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047
EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.
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
Clone
Terminate
AWS CLI export
Summary
Application user interfaces
Monitoring
Hardware
Configurations
Events
Steps
Bootstrap actions
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Serial
Telnet
Rlogin
SUPDUP
Load
Save
Delete
About
Help
Open
Cancel
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



PuTTY Configuration

Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)

☐ Bypass authentication entirely (SSH-2 only)

☐ Disconnect if authentication succeeds trivially

Authentication methods

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

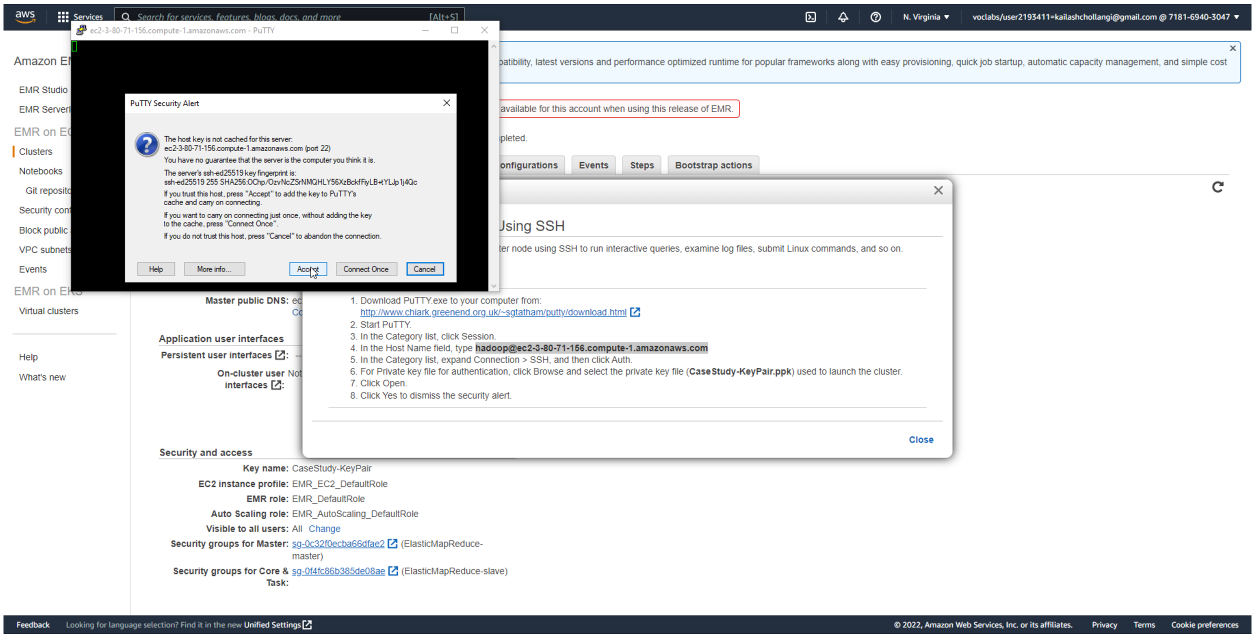
Authentication parameters

☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication: C:\Users\vkail\OneDrive\Desktop\Hive

Click on “open” and then Accept the connection



ec2-3-80-71-156.compute-1.amazonaws.com - PuTTY

PuTTY Security Alert

The host key is not cached for this server:
ec2-3-80-71-156.compute-1.amazonaws.com (port 22)
You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:OChp/OzvNcZSrNMQHLY56XzBckfFiyLB+tYLJp1j4Qc
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".
If you do not trust this host, press "Cancel" to abandon the connection.

Help | More info... | Accept | Connect Once | Cancel

patibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost

available for this account when using this release of EMR.

pleted.

Using SSH

ter node using SSH to run interactive queries, examine log files, submit Linux commands, and so on.

1. Download PuTTY.exe to your computer from: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
2. Start PuTTY.
3. In the Category list, click Session.
4. In the Host Name field, type **hadoop@ec2-3-80-71-156.compute-1.amazonaws.com**
5. In the Category list, expand Connection > SSH, and then click Auth.
6. For Private key file for authentication, click Browse and select the private key file (**CaseStudy-KeyPair.ppk**) used to launch the cluster.
7. Click Open.
8. Click Yes to dismiss the security alert.

Close

Master public DNS: ec

Application user interfaces

Persistent user interfaces: --

On-cluster user interfaces: Not

Security and access

Key name: CaseStudy-KeyPair

EC2 instance profile: EMR_EC2_DefaultRole

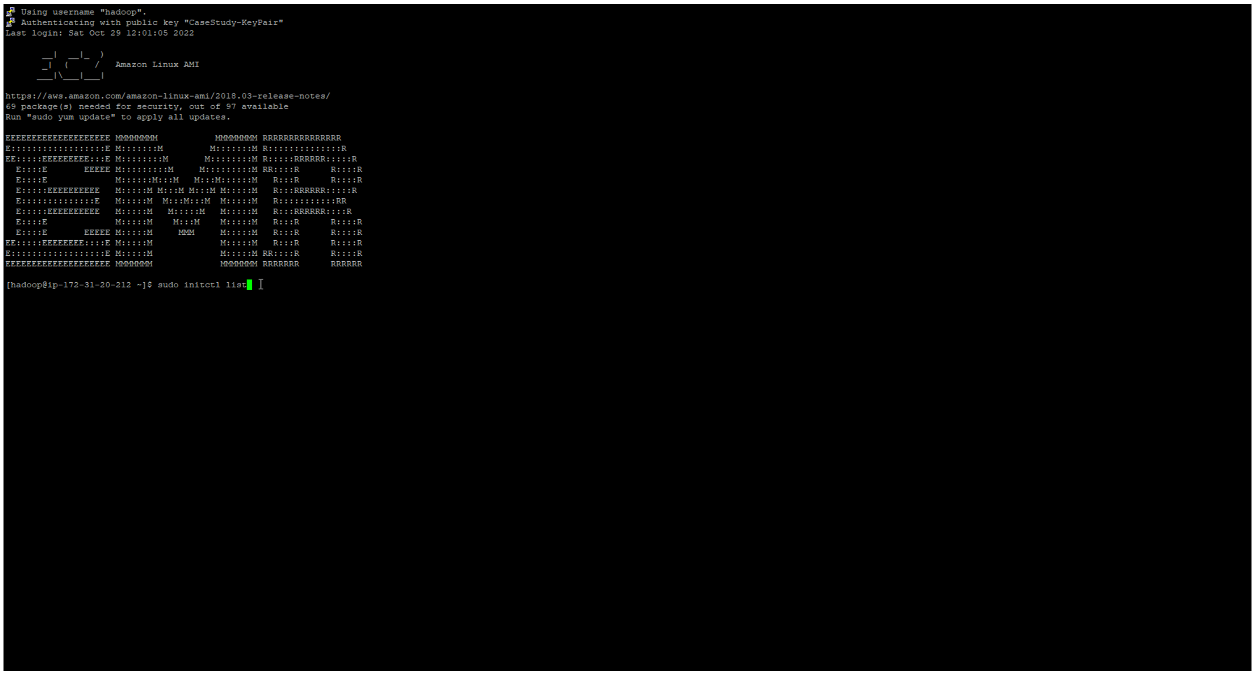
EMR role: EMR_DefaultRole

Auto Scaling role: EMR_AutoScaling_DefaultRole

Visible to all users: All Change

Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)

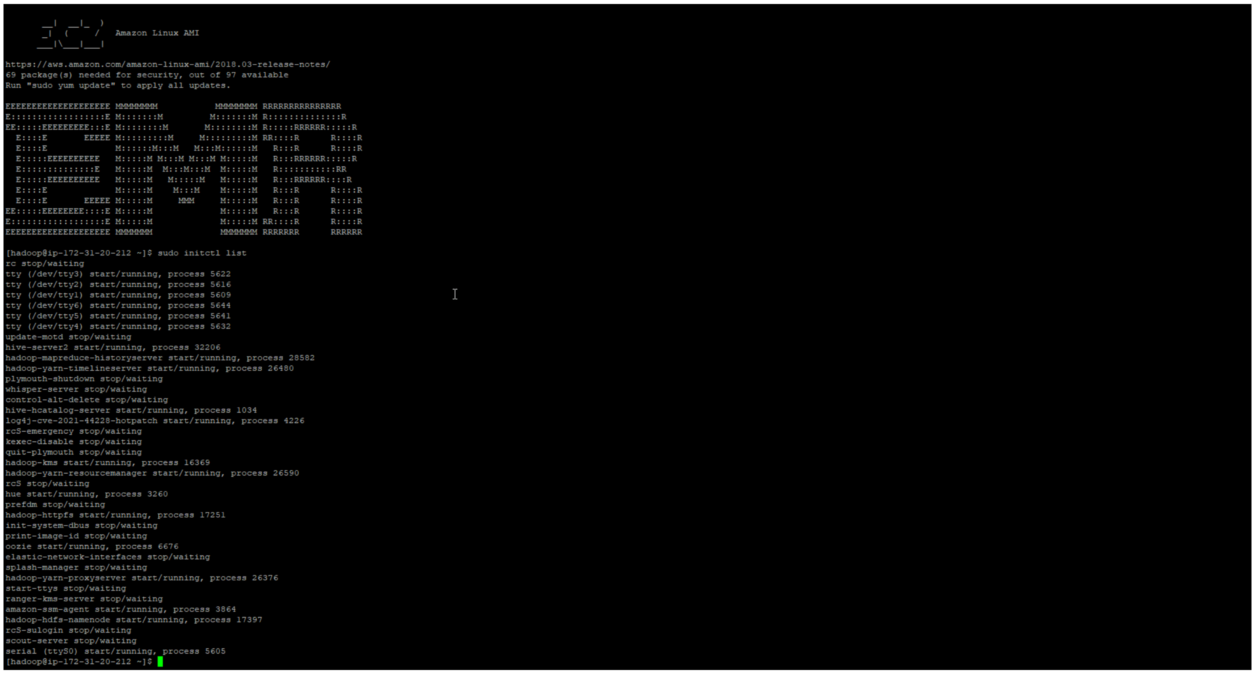
Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)



```
Using username "hadoop".
Authenticating with public key "CaseStudy-KeyPair"
Last login: Sat Oct 29 12:01:05 2022

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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
69 package(s) needed for security, out of 97 available
Run "sudo yum update" to apply all updates.

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

[hadoop@ip-172-31-20-212 ~]$ sudo initctl list
```

EMR CLI is launched

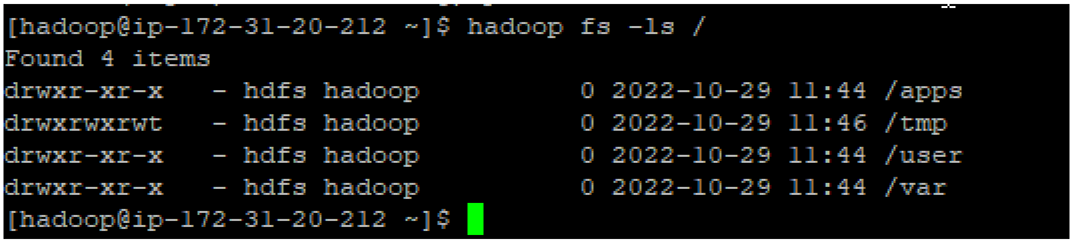
Verifying the services that are running on Hadoop cluster with command “**sudo initctl list**”



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

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
69 package(s) needed for security, out of 97 available
Run "sudo yum update" to apply all updates.

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

[hadoop@ip-172-31-20-212 ~]$ sudo initctl list
rc stop/waiting
tty (/dev/tty3) start/running, process 5622
tty (/dev/tty2) start/running, process 5616
tty (/dev/tty1) start/running, process 5609
tty (/dev/tty6) start/running, process 5644
tty (/dev/tty5) start/running, process 5641
tty (/dev/tty4) start/running, process 5632
update-motd stop/waiting
hive-server2 start/running, process 32206
hadoop-mapreduce-historyserver start/running, process 28582
hadoop-yarn-timelineserver start/running, process 26480
plymouth-shutdown stop/waiting
whisper-server stop/waiting
control-alt-delete stop/waiting
hive-hcatalog-server start/running, process 1034
log4j-cve-2021-44228-hotpatch start/running, process 4226
rcS-emergency stop/waiting
kexec-disable stop/waiting
quit-plymouth stop/waiting
hadoop-kms start/running, process 16369
hadoop-yarn-resourcemanager start/running, process 26590
rcS stop/waiting
hue start/running, process 3260
prefdm stop/waiting
hadoop-httpfs start/running, process 17251
init-system-dbus stop/waiting
print-image-id stop/waiting
oozie start/running, process 6676
elastic-network-interfaces stop/waiting
splash-manager stop/waiting
hadoop-yarn-proxyserver start/running, process 26376
start-ttys stop/waiting
ranger-kms-server stop/waiting
amazon-ssm-agent start/running, process 3864
hadoop-hdfs-namenode start/running, process 17397
rcS-sulogin stop/waiting
scout-server stop/waiting
serial (ttyS0) start/running, process 5605
[hadoop@ip-172-31-20-212 ~]$
```

We can see that Hive services are running

Verifying the Hadoop file system with command “**hadoop fs -ls /**”



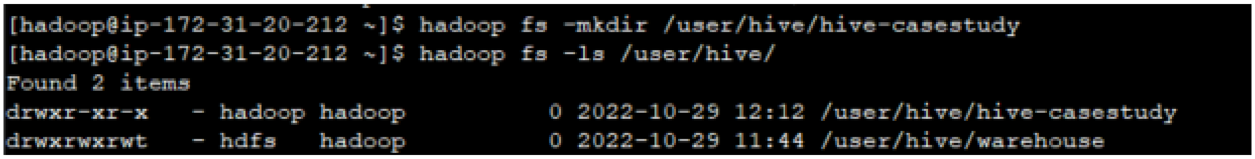
```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x   - hdfs hadoop          0 2022-10-29 11:44 /apps
drwxrwxrwt   - hdfs hadoop          0 2022-10-29 11:46 /tmp
drwxr-xr-x   - hdfs hadoop          0 2022-10-29 11:44 /user
drwxr-xr-x   - hdfs hadoop          0 2022-10-29 11:44 /var
[hadoop@ip-172-31-20-212 ~]$
```

All the above are inbuilt directories in HDFS.

CREATING A NEW DIRECTORY FOR HIVE CASE STUDY:

Creating a new directory under user>hive for Hive case study to store the data files and directory name creating is “**hive-casestudy**” and verifying whether the new directory is listed in Hadoop file system>user>hive

hadoop fs -mkdir /user/hive/hive-casestudy
hadoop fs -ls /user/hive/

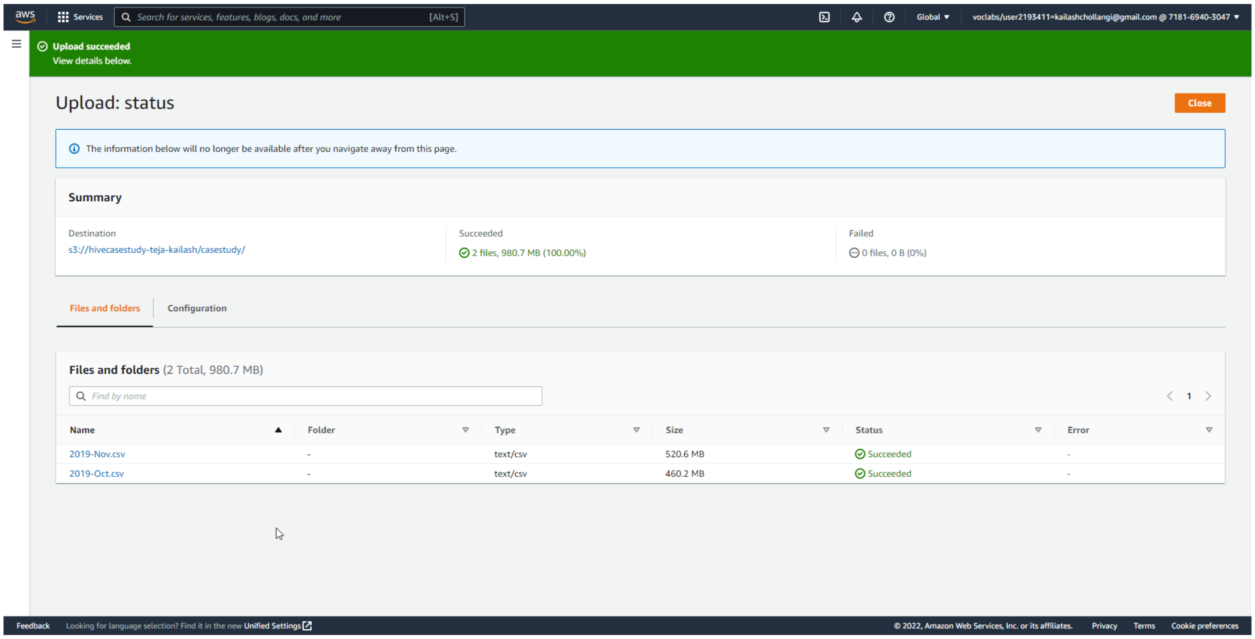


```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -mkdir /user/hive/hive-casestudy
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /user/hive/
Found 2 items
drwxr-xr-x   - hadoop hadoop          0 2022-10-29 12:12 /user/hive/hive-casestudy
drwxrwxrwt   - hdfs   hadoop          0 2022-10-29 11:44 /user/hive/warehouse
```

New directory is successfully created

LOADING THE DATA FROM S3 BUCKET to HDFS:

Copying the file path from S3


aws
Services
Search for services, features, blogs, docs, and more [Alt+S]
Global ▾
voclabs/user2193411=kailashchollangi@gmail.com @ 7181-6940-3047 ▾
Close
Files and folders
Configuration
Find by name
1
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Upload succeeded
View details below.

Upload: status

The information below will no longer be available after you navigate away from this page.

Summary

Destination: s3://hivecasestudy-teja-kailash/casestudy/

Succeeded: ✓ 2 files, 980.7 MB (100.00%)

Failed: 0 files, 0 B (0%)

Files and folders (2 Total, 980.7 MB)

Name	Folder	Type	Size	Status	Error
2019-Nov.csv	-	text/csv	520.6 MB	✓ Succeeded	-
2019-Oct.csv	-	text/csv	460.2 MB	✓ Succeeded	-

Distributed copy command is using to copy the data from S3 to HDFS –

For 2019 October:

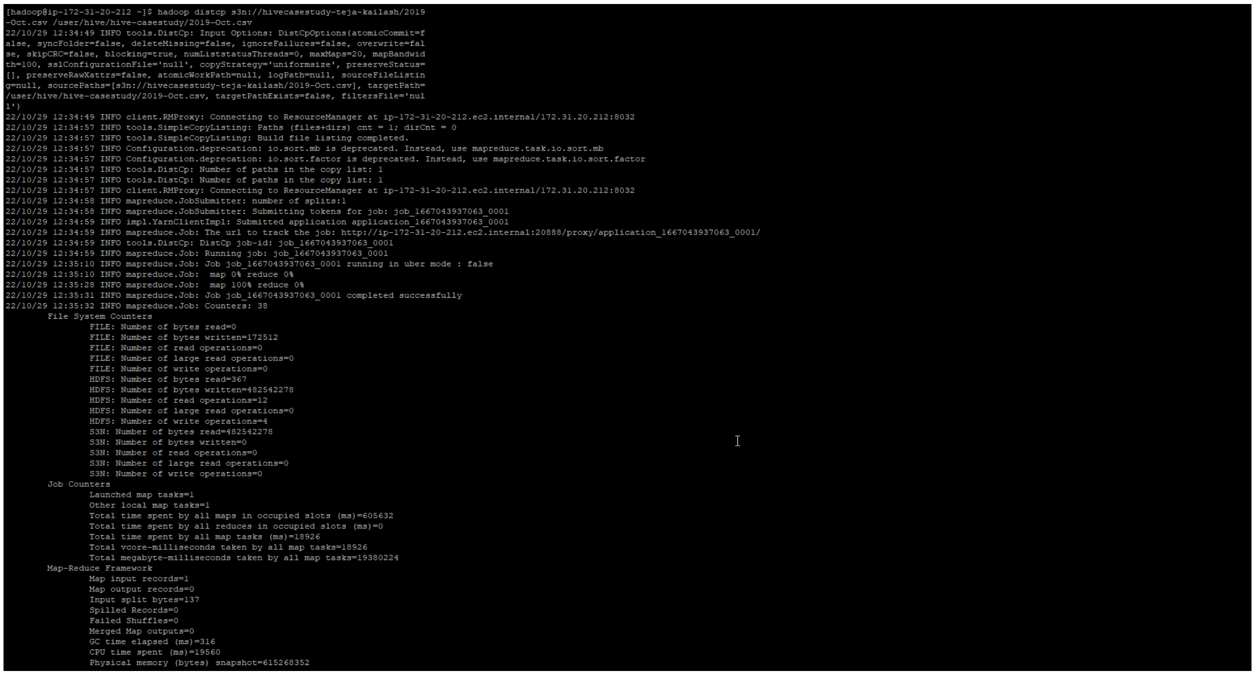
```
hadoop distcp s3n://hivecasestudy-teja-kailash/2019-Oct.csv /user/hive/hive-casestudy/2019-Oct.csv
```

For 2019 November:

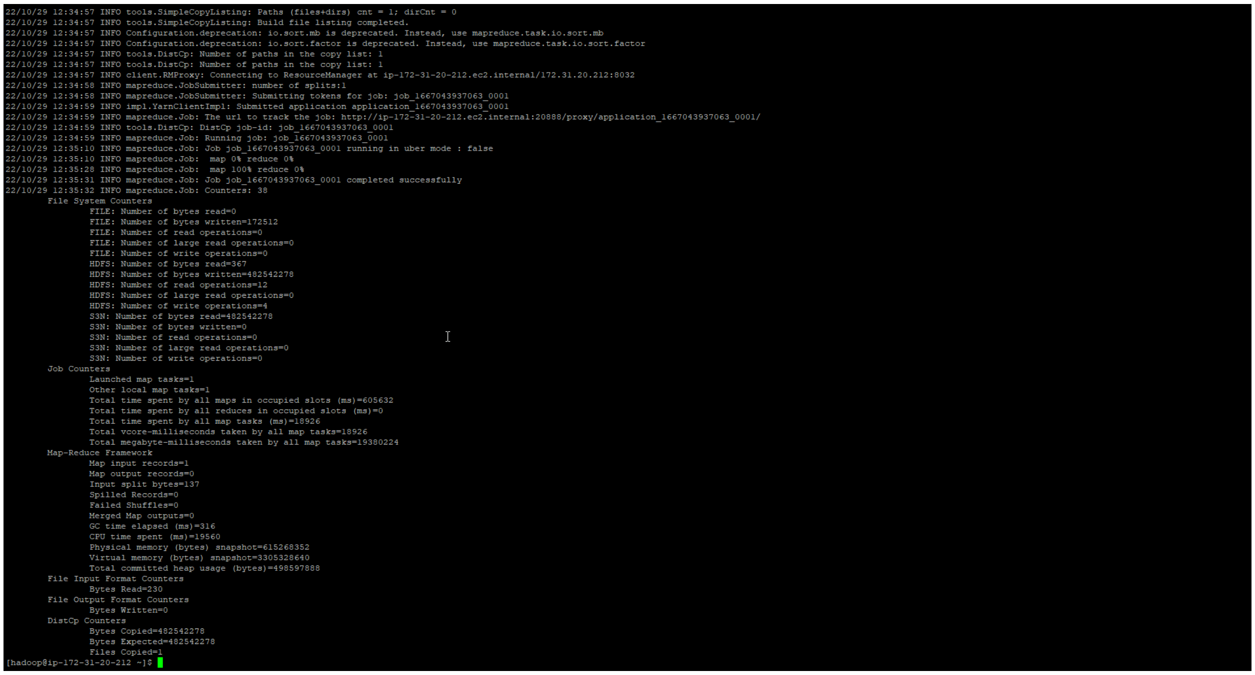
```
hadoop distcp s3n://hivecasestudy-teja-kailash/2019-Nov.csv /user/hive/hive-casestudy/2019-Nov.csv
```

Below are the screenshots for copying October 2019 and November 2019 data individually

October 2019:

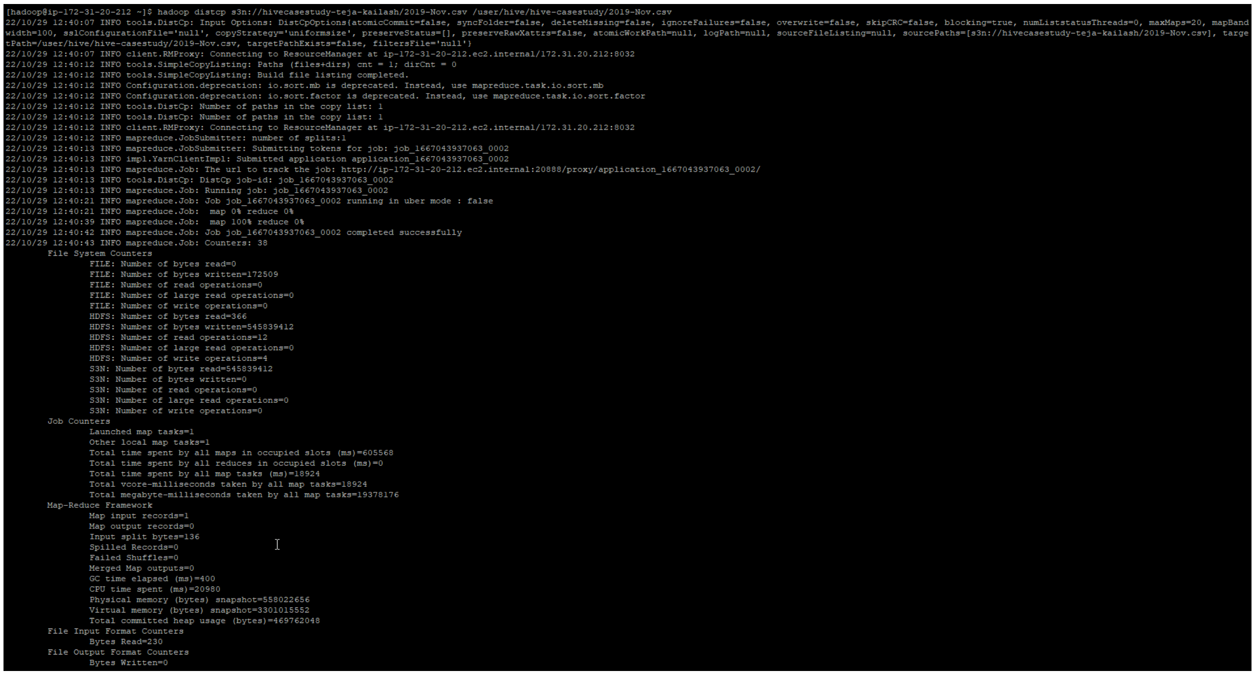


```
[hadoop@ip-172-31-20-212 ~]$ hadoop distcp s3n://hivecasestudy-teja-kailash/2019
-Oct.csv /user/hive/hive-casestudy/2019-Oct.csv
22/10/29 12:34:49 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=f
alse, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=fal
se, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBandwid
th=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=
[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListin
g=null, sourcePaths=[s3n://hivecasestudy-teja-kailash/2019-Oct.csv], targetPath=
/user/hive/hive-casestudy/2019-Oct.csv, targetPathExists=false, filtersFile='nul
l'}
22/10/29 12:34:49 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-20-212.ec2.internal/172.31.20.212:8032
22/10/29 12:34:57 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/10/29 12:34:57 INFO tools.SimpleCopyListing: Build file listing completed.
22/10/29 12:34:57 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
22/10/29 12:34:57 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
22/10/29 12:34:57 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:34:57 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:34:57 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-20-212.ec2.internal/172.31.20.212:8032
22/10/29 12:34:58 INFO mapreduce.JobSubmitter: number of splits:1
22/10/29 12:34:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1667043937063_0001
22/10/29 12:34:59 INFO impl.YarnClientImpl: Submitted application application_1667043937063_0001
22/10/29 12:34:59 INFO mapreduce.Job: The url to track the job: http://ip-172-31-20-212.ec2.internal:20888/proxy/application_1667043937063_0001/
22/10/29 12:34:59 INFO tools.DistCp: DistCp job-id: job_1667043937063_0001
22/10/29 12:34:59 INFO mapreduce.Job: Running job: job_1667043937063_0001
22/10/29 12:35:10 INFO mapreduce.Job: Job job_1667043937063_0001 running in uber mode : false
22/10/29 12:35:10 INFO mapreduce.Job:  map 0% reduce 0%
22/10/29 12:35:28 INFO mapreduce.Job:  map 100% reduce 0%
22/10/29 12:35:31 INFO mapreduce.Job: Job job_1667043937063_0001 completed successfully
22/10/29 12:35:32 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=172512
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=367
                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
                S3N: Number of bytes read=482542278
                S3N: Number of bytes written=0
                S3N: Number of read operations=0
                S3N: Number of large read operations=0
                S3N: 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)=316
                CPU time spent (ms)=19560
                Physical memory (bytes) snapshot=615268352
```

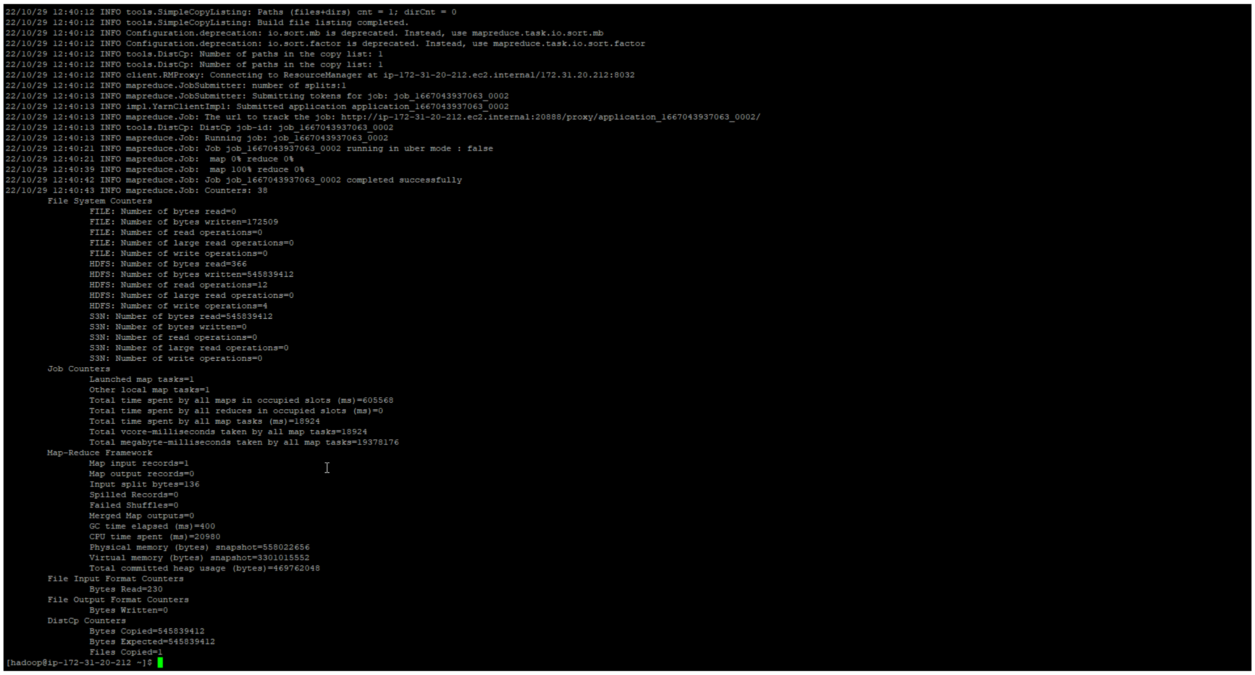


```
22/10/29 12:34:57 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/10/29 12:34:57 INFO tools.SimpleCopyListing: Build file listing completed.
22/10/29 12:34:57 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
22/10/29 12:34:57 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
22/10/29 12:34:57 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:34:57 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:34:57 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-20-212.ec2.internal/172.31.20.212:8032
22/10/29 12:34:58 INFO mapreduce.JobSubmitter: number of splits:1
22/10/29 12:34:58 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1667043937063_0001
22/10/29 12:34:59 INFO impl.YarnClientImpl: Submitted application application_1667043937063_0001
22/10/29 12:34:59 INFO mapreduce.Job: The url to track the job: http://ip-172-31-20-212.ec2.internal:20888/proxy/application_1667043937063_0001/
22/10/29 12:34:59 INFO tools.DistCp: DistCp job-id: job_1667043937063_0001
22/10/29 12:34:59 INFO mapreduce.Job: Running job: job_1667043937063_0001
22/10/29 12:35:10 INFO mapreduce.Job: Job job_1667043937063_0001 running in uber mode : false
22/10/29 12:35:10 INFO mapreduce.Job:  map 0% reduce 0%
22/10/29 12:35:28 INFO mapreduce.Job:  map 100% reduce 0%
22/10/29 12:35:31 INFO mapreduce.Job: Job job_1667043937063_0001 completed successfully
22/10/29 12:35:32 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=172512
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=367
                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
                S3N: Number of bytes read=482542278
                S3N: Number of bytes written=0
                S3N: Number of read operations=0
                S3N: Number of large read operations=0
                S3N: 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)=316
                CPU time spent (ms)=19560
                Physical memory (bytes) snapshot=615268352
                Virtual memory (bytes) snapshot=3305328640
                Total committed heap usage (bytes)=498597888
        File Input Format Counters
                Bytes Read=230
        File Output Format Counters
                Bytes Written=0
        DistCp Counters
                Bytes Copied=482542278
                Bytes Expected=482542278
                Files Copied=1
[hadoop@ip-172-31-20-212 ~]$
```

November 2019

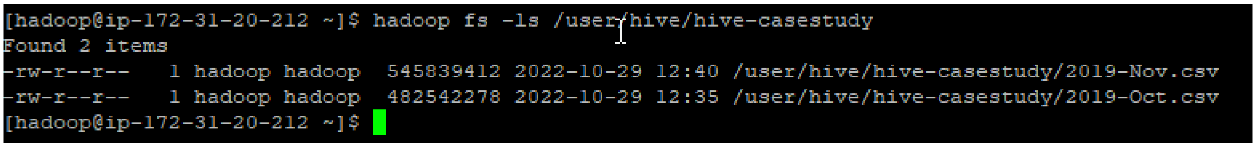


```
[hadoop@ip-172-31-20-212 ~]$ hadoop distcp s3n://hivecasestudy-teja-kailash/2019-Nov.csv /user/hive/hive-casestudy/2019-Nov.csv
22/10/29 12:40:07 INFO tools.DistCp: Input Options: DistCpOptions{atomicCommit=false, syncFolder=false, deleteMissing=false, ignoreFailures=false, overwrite=false, skipCRC=false, blocking=true, numListstatusThreads=0, maxMaps=20, mapBand
width=100, sslConfigurationFile='null', copyStrategy='uniformsize', preserveStatus=[], preserveRawXattrs=false, atomicWorkPath=null, logPath=null, sourceFileListing=null, sourcePaths=[s3n://hivecasestudy-teja-kailash/2019-Nov.csv], targe
tPath=/user/hive/hive-casestudy/2019-Nov.csv, targetPathExists=false, filtersFile='null'}
22/10/29 12:40:07 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-20-212.ec2.internal/172.31.20.212:8032
22/10/29 12:40:12 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/10/29 12:40:12 INFO tools.SimpleCopyListing: Build file listing completed.
22/10/29 12:40:12 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
22/10/29 12:40:12 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
22/10/29 12:40:12 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:40:12 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:40:12 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-20-212.ec2.internal/172.31.20.212:8032
22/10/29 12:40:12 INFO mapreduce.JobSubmitter: number of splits:1
22/10/29 12:40:13 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1667043937063_0002
22/10/29 12:40:13 INFO impl.YarnClientImpl: Submitted application application_1667043937063_0002
22/10/29 12:40:13 INFO mapreduce.Job: The url to track the job: http://ip-172-31-20-212.ec2.internal:20888/proxy/application_1667043937063_0002/
22/10/29 12:40:13 INFO tools.DistCp: DistCp job-id: job_1667043937063_0002
22/10/29 12:40:13 INFO mapreduce.Job: Running job: job_1667043937063_0002
22/10/29 12:40:21 INFO mapreduce.Job: Job job_1667043937063_0002 running in uber mode : false
22/10/29 12:40:21 INFO mapreduce.Job:  map 0% reduce 0%
22/10/29 12:40:39 INFO mapreduce.Job:  map 100% reduce 0%
22/10/29 12:40:42 INFO mapreduce.Job: Job job_1667043937063_0002 completed successfully
22/10/29 12:40:43 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=172509
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=366
                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
                S3N: Number of bytes read=545839412
                S3N: Number of bytes written=0
                S3N: Number of read operations=0
                S3N: Number of large read operations=0
                S3N: 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)=605568
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=18924
                Total vcore-milliseconds taken by all map tasks=18924
                Total megabyte-milliseconds taken by all map tasks=19378176
        Map-Reduce Framework
                Map input records=1
                Map output records=0
                Input split bytes=136
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=400
                CPU time spent (ms)=20980
                Physical memory (bytes) snapshot=558022656
                Virtual memory (bytes) snapshot=3301015552
                Total committed heap usage (bytes)=469762048
        File Input Format Counters
                Bytes Read=230
        File Output Format Counters
                Bytes Written=0
```



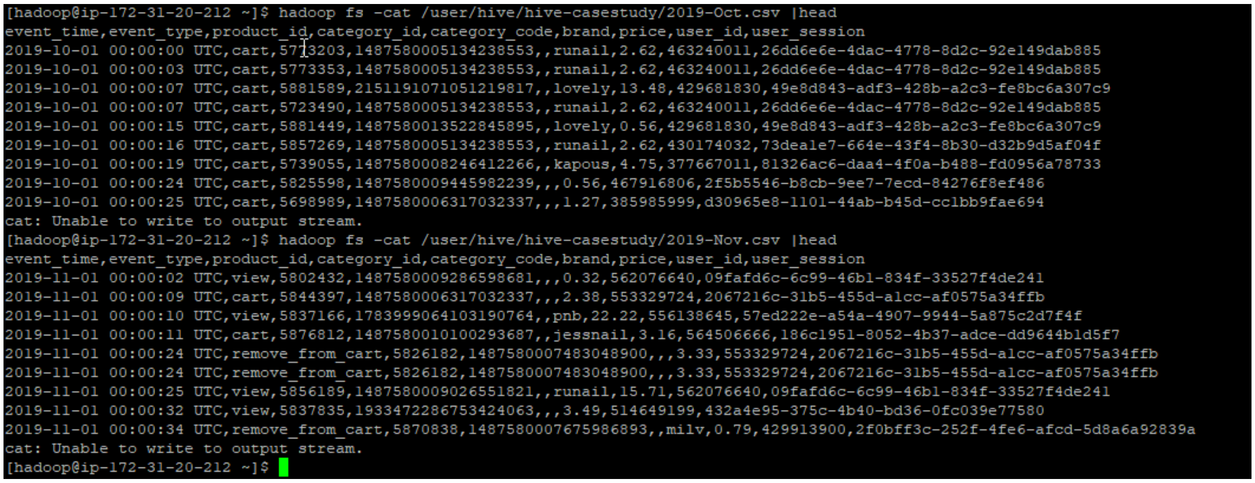
```
22/10/29 12:40:12 INFO tools.SimpleCopyListing: Paths (files+dirs) cnt = 1; dirCnt = 0
22/10/29 12:40:12 INFO tools.SimpleCopyListing: Build file listing completed.
22/10/29 12:40:12 INFO Configuration.deprecation: io.sort.mb is deprecated. Instead, use mapreduce.task.io.sort.mb
22/10/29 12:40:12 INFO Configuration.deprecation: io.sort.factor is deprecated. Instead, use mapreduce.task.io.sort.factor
22/10/29 12:40:12 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:40:12 INFO tools.DistCp: Number of paths in the copy list: 1
22/10/29 12:40:12 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-20-212.ec2.internal/172.31.20.212:8032
22/10/29 12:40:12 INFO mapreduce.JobSubmitter: number of splits:1
22/10/29 12:40:13 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1667043937063_0002
22/10/29 12:40:13 INFO impl.YarnClientImpl: Submitted application application_1667043937063_0002
22/10/29 12:40:13 INFO mapreduce.Job: The url to track the job: http://ip-172-31-20-212.ec2.internal:20888/proxy/application_1667043937063_0002/
22/10/29 12:40:13 INFO tools.DistCp: DistCp job-id: job_1667043937063_0002
22/10/29 12:40:13 INFO mapreduce.Job: Running job: job_1667043937063_0002
22/10/29 12:40:21 INFO mapreduce.Job: Job job_1667043937063_0002 running in uber mode : false
22/10/29 12:40:21 INFO mapreduce.Job:  map 0% reduce 0%
22/10/29 12:40:39 INFO mapreduce.Job:  map 100% reduce 0%
22/10/29 12:40:42 INFO mapreduce.Job: Job job_1667043937063_0002 completed successfully
22/10/29 12:40:43 INFO mapreduce.Job: Counters: 38
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=172509
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=366
                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
                S3N: Number of bytes read=545839412
                S3N: Number of bytes written=0
                S3N: Number of read operations=0
                S3N: Number of large read operations=0
                S3N: 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)=605568
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=18924
                Total vcore-milliseconds taken by all map tasks=18924
                Total megabyte-milliseconds taken by all map tasks=19378176
        Map-Reduce Framework
                Map input records=1
                Map output records=0
                Input split bytes=136
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=400
                CPU time spent (ms)=20980
                Physical memory (bytes) snapshot=558022656
                Virtual memory (bytes) snapshot=3301015552
                Total committed heap usage (bytes)=469762048
        File Input Format Counters
                Bytes Read=230
        File Output Format Counters
                Bytes Written=0
        DistCp Counters
                Bytes Copied=545839412
                Bytes Expected=545839412
                Files Copied=1
[hadoop@ip-172-31-20-212 ~]$
```

Verifying whether the data is successfully copied into HDFS from S3 buckets
Command: **hadoop fs -ls /user/hive/hive-casestudy**



```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /user/hive/hive-casestudy
Found 2 items
-rw-r--r--   1 hadoop hadoop  545839412 2022-10-29 12:40 /user/hive/hive-casestudy/2019-Nov.csv
-rw-r--r--   1 hadoop hadoop  482542278 2022-10-29 12:35 /user/hive/hive-casestudy/2019-Oct.csv
[hadoop@ip-172-31-20-212 ~]$
```

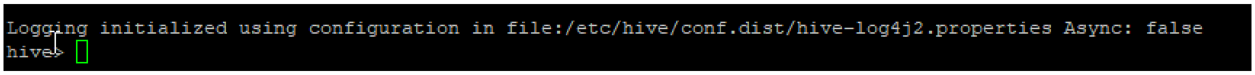
Inspecting the table data to know which columns are available before creating the hive table with command **“hadoop fs -cat /user/hive/hive-casestudy/2019-Oct.csv |head” and “hadoop fs -cat /user/hive/hive-casestudy/2019-Nov.csv |head”**



```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -cat /user/hive/hive-casestudy/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@ip-172-31-20-212 ~]$ hadoop fs -cat /user/hive/hive-casestudy/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-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC,view,5837166,1783999064103190764,,pnb,22.22,556138645,57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC,cart,5876812,1487580010100293687,,jessnail,3.16,564506666,186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:24 UTC,remove_from_cart,5826182,1487580007483048900,,,3.33,553329724,2067216c-31b5-455d-a1cc-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.
[hadoop@ip-172-31-20-212 ~]$
```

Both the tables are having same columns of data

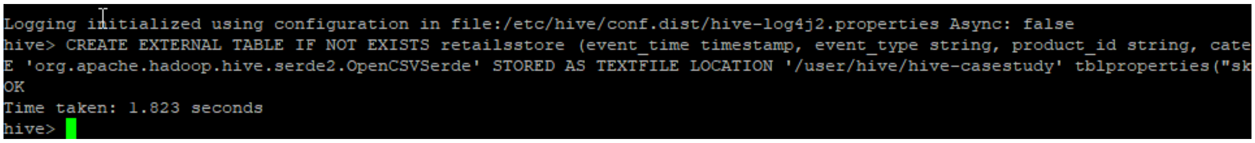
Moving to hive:



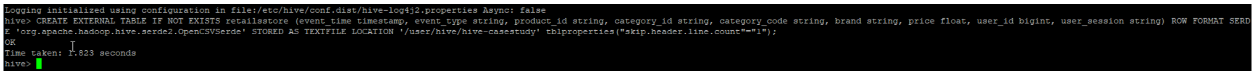
```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive>
```

CREATING AN EXTERNAL TABLE IN HIVE:

CREATE EXTERNAL TABLE IF NOT EXISTS retailsstore (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/hive-casestudy' tblproperties("skip.header.line.count"="1");



```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retailsstore (event_time timestamp, event_type string, product_id string, cate
E 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hive-casestudy' tblproperties("sk
OK
Time taken: 1.823 seconds
hive>
```



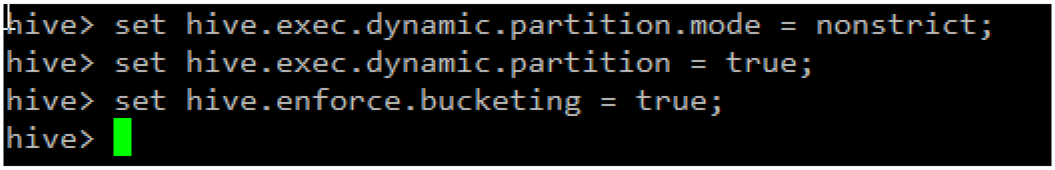
```
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.properties Async: false
hive> CREATE EXTERNAL TABLE IF NOT EXISTS retailsstore (event_time timestamp, event_type string, product_id string, category_id string, category_code string, brand string, price float, user_id bigint, user_session string) ROW FORMAT SERD
E 'org.apache.hadoop.hive.serde2.OpenCSVSerde' STORED AS TEXTFILE LOCATION '/user/hive/hive-casestudy' tblproperties("skip.header.line.count"="1");
OK
Time taken: 1.823 seconds
hive>
```

Below command is used to set the display the header columns

set hive.cli.print.header = true;

APPLYING OPTIMIZATION TECHNIQUES - PARTITIONING AND BUCKETING:

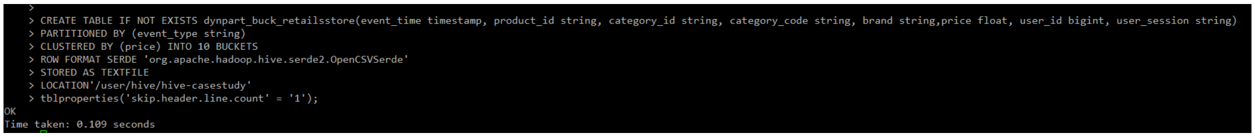
Below commands are to enable the dynamic partitioning and bucketing
hive> set hive.exec.dynamic.partition.mode = nonstrict;
hive> set hive.exec.dynamic.partition = true;
hive> set hive.enforce.bucketing = true;



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

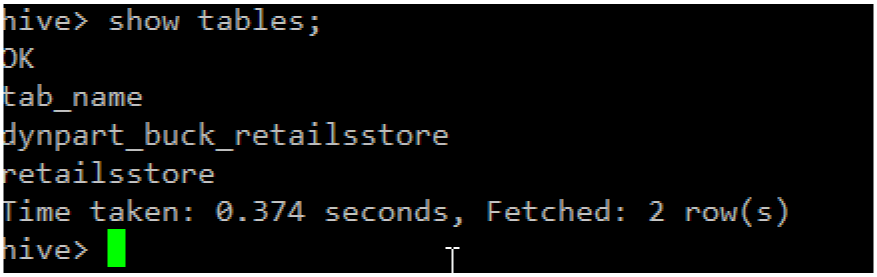
Creating an optimized table by applying partitioning on “event_type” and bucketing on “price”

CREATE TABLE IF NOT EXISTS dynpart_buck_retailsstore(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 10 BUCKETS
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
STORED AS TEXTFILE
LOCATION '/user/hive/hive-casestudy'
tblproperties('skip.header.line.count' = '1');



```
    >
    > CREATE TABLE IF NOT EXISTS dynpart_buck_retailsstore(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 10 BUCKETS
    > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > STORED AS TEXTFILE
    > LOCATION'/user/hive/hive-casestudy'
    > tblproperties('skip.header.line.count' = '1');
OK
Time taken: 0.109 seconds
```

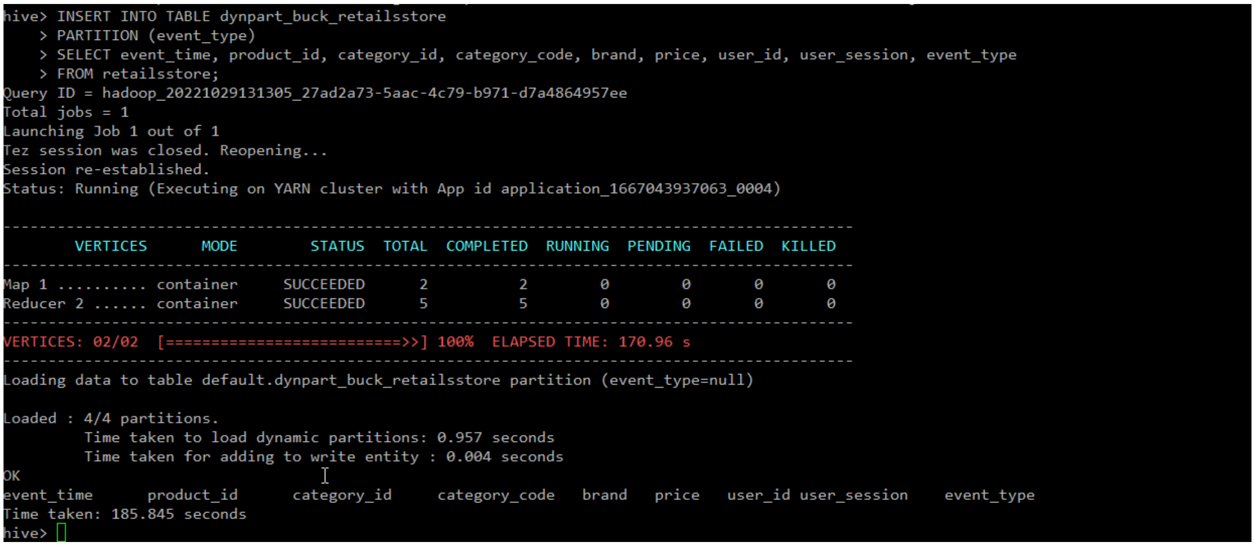
Verifying the created table



```
hive> show tables;
OK
tab_name
dynpart_buck_retailsstore
retailsstore
Time taken: 0.374 seconds, Fetched: 2 row(s)
hive>
```

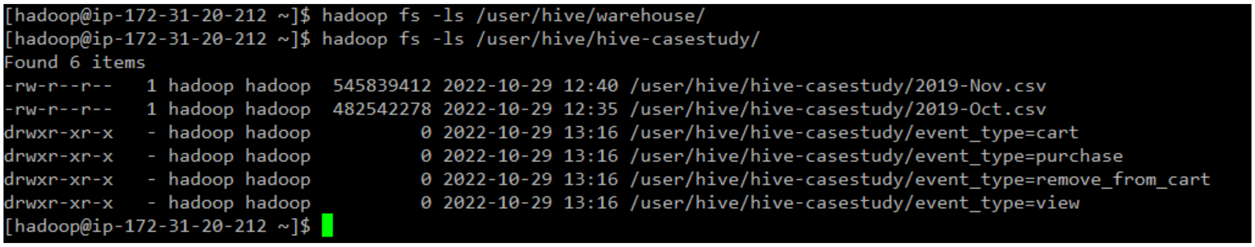
INSERTING THE DATA INTO NEWLY CREATED OPTIMIZED TABLE (dynpart_buck_retailsstore) FROM EXISTING TABLE(retailsstore):

INSERT INTO TABLE dynpart_buck_retailsstore
PARTITION (event_type)
SELECT event_time,
product_id, category_id, category_code, brand, price, user_id, user_session, event_type
FROM retailsstore;



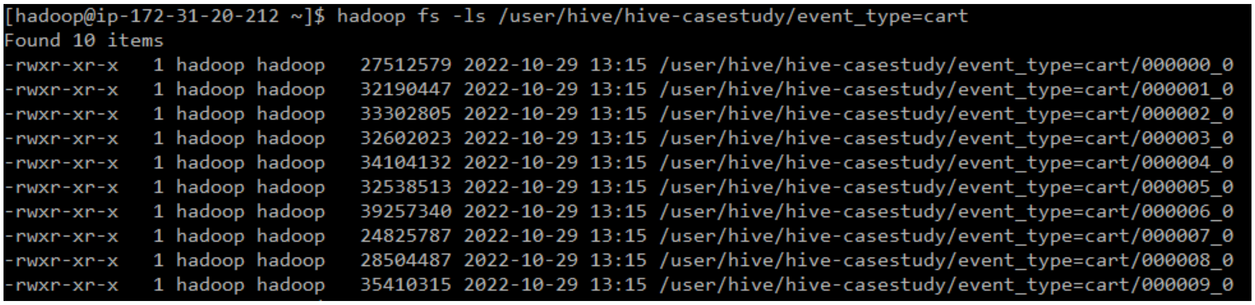
```
hive> INSERT INTO TABLE dynpart_buck_retailsstore
    > PARTITION (event_type)
    > SELECT event_time, product_id, category_id, category_code, brand, price, user_id, user_session, event_type
    > FROM retailsstore;
Query ID = hadoop_20221029131305_27ad2a73-5aac-4c79-b971-d7a4864957ee
Total jobs = 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_1667043937063_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: 170.96 s
----------------------------------------------------------------------------------------------
Loading data to table default.dynpart_buck_retailsstore partition (event_type=null)

Loaded : 4/4 partitions.
         Time taken to load dynamic partitions: 0.957 seconds
         Time taken for adding to write entity : 0.004 seconds
OK
event_time      product_id      category_id     category_code   brand   price   user_id user_session    event_type
Time taken: 185.845 seconds
hive> 
```

Output: Based on the above results, it partitioned into 4
Verifying the partitioned in the Hadoop file system

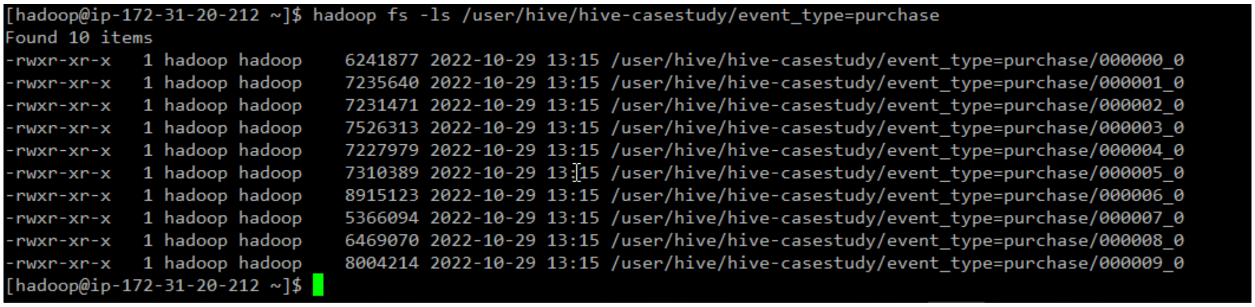


```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /user/hive/warehouse/
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /user/hive/hive-casestudy/
Found 6 items
-rw-r--r--   1 hadoop hadoop  545839412 2022-10-29 12:40 /user/hive/hive-casestudy/2019-Nov.csv
-rw-r--r--   1 hadoop hadoop  482542278 2022-10-29 12:35 /user/hive/hive-casestudy/2019-Oct.csv
drwxr-xr-x   - hadoop hadoop          0 2022-10-29 13:16 /user/hive/hive-casestudy/event_type=cart
drwxr-xr-x   - hadoop hadoop          0 2022-10-29 13:16 /user/hive/hive-casestudy/event_type=purchase
drwxr-xr-x   - hadoop hadoop          0 2022-10-29 13:16 /user/hive/hive-casestudy/event_type=remove_from_cart
drwxr-xr-x   - hadoop hadoop          0 2022-10-29 13:16 /user/hive/hive-casestudy/event_type=view
[hadoop@ip-172-31-20-212 ~]$ 
```

Randomly verifying the partitioned data in hadoop



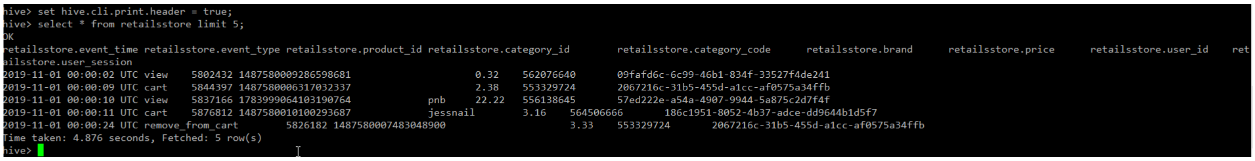
```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /user/hive/hive-casestudy/event_type=cart
Found 10 items
-rwxr-xr-x   1 hadoop hadoop   27512579 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000000_0
-rwxr-xr-x   1 hadoop hadoop   32190447 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000001_0
-rwxr-xr-x   1 hadoop hadoop   33302805 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000002_0
-rwxr-xr-x   1 hadoop hadoop   32602023 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000003_0
-rwxr-xr-x   1 hadoop hadoop   34104132 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000004_0
-rwxr-xr-x   1 hadoop hadoop   32538513 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000005_0
-rwxr-xr-x   1 hadoop hadoop   39257340 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000006_0
-rwxr-xr-x   1 hadoop hadoop   24825787 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000007_0
-rwxr-xr-x   1 hadoop hadoop   28504487 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000008_0
-rwxr-xr-x   1 hadoop hadoop   35410315 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=cart/000009_0
```



```
[hadoop@ip-172-31-20-212 ~]$ hadoop fs -ls /user/hive/hive-casestudy/event_type=purchase
Found 10 items
-rwxr-xr-x   1 hadoop hadoop    6241877 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000000_0
-rwxr-xr-x   1 hadoop hadoop    7235640 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000001_0
-rwxr-xr-x   1 hadoop hadoop    7231471 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000002_0
-rwxr-xr-x   1 hadoop hadoop    7526313 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000003_0
-rwxr-xr-x   1 hadoop hadoop    7227979 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000004_0
-rwxr-xr-x   1 hadoop hadoop    7310389 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000005_0
-rwxr-xr-x   1 hadoop hadoop    8915123 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000006_0
-rwxr-xr-x   1 hadoop hadoop    5366094 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000007_0
-rwxr-xr-x   1 hadoop hadoop    6469070 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000008_0
-rwxr-xr-x   1 hadoop hadoop    8004214 2022-10-29 13:15 /user/hive/hive-casestudy/event_type=purchase/000009_0
[hadoop@ip-172-31-20-212 ~]$ 
```

VERIFYING THE PERFORMANCE OF BOTH THE TABLES – BEFORE AND AFTER OPTIMIZED TECHNIQUES:

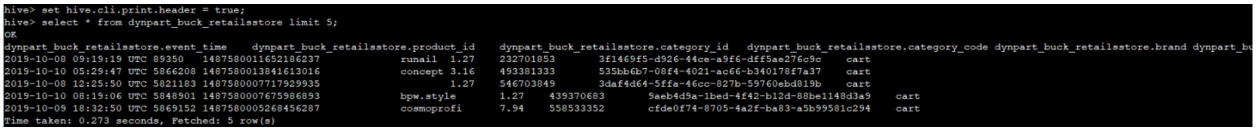
select * from retailsstore limit 5;



```
hive> set hive.cli.print.header = true;
hive> select * from retailsstore limit 5;
OK
retailsstore.event_time retailsstore.event_type retailsstore.product_id retailsstore.category_id       retailsstore.category_code      retailsstore.brand      retailsstore.price      retailsstore.user_id    ret
ailsstore.user_session
2019-11-01 00:00:02 UTC view    5802432 1487580009286598681                     0.32    562076640       09fafd6c-6c99-46b1-834f-33527f4de241
2019-11-01 00:00:09 UTC cart    5844397 1487580006317032337                     2.38    553329724       2067216c-31b5-455d-a1cc-af0575a34ffb
2019-11-01 00:00:10 UTC view    5837166 1783999064103190764             pnb     22.22   556138645       57ed222e-a54a-4907-9944-5a875c2d7f4f
2019-11-01 00:00:11 UTC cart    5876812 1487580010100293687             jessnail        3.16    564506666       186c1951-8052-4b37-adce-dd9644b1d5f7
2019-11-01 00:00:24 UTC remove_from_cart        5826182 1487580007483048900                     3.33    553329724       2067216c-31b5-455d-a1cc-af0575a34ffb
Time taken: 4.876 seconds, Fetched: 5 row(s)
hive> 
```

Time taken to retrieve first 5 rows of data **before optimization is 4.876 seconds (above)**

select * from dynpart_buck_retailsstore limit 5;



```
hive> set hive.cli.print.header = true;
hive> select * from dynpart_buck_retailsstore limit 5;
OK
dynpart_buck_retailsstore.event_time    dynpart_buck_retailsstore.product_id    dynpart_buck_retailsstore.category_id   dynpart_buck_retailsstore.category_code dynpart_buck_retailsstore.brand dynpart_bu
2019-10-08 09:19:19 UTC 89350   1487580011652186237             runail  1.27    232701853       3f1469f5-d926-44ce-a9f6-dff5ae276c9c    cart
2019-10-10 05:29:47 UTC 5866208 1487580013841613016             concept 3.16    493381333       535bb6b7-08f4-4021-ac66-b340178f7a37    cart
2019-10-08 12:25:50 UTC 5821183 1487580007717929935                     1.27    546703849       3daf4d64-5ffa-46cc-827b-59760ebd819b    cart
2019-10-10 08:19:06 UTC 5848901 1487580007675986893             bpw.style       1.27    439370683       9aeb4d9a-1bed-4f42-b12d-88be1148d3a9    cart
2019-10-09 18:32:50 UTC 5869152 1487580005268456287             cosmoprofi      7.94    558533352       cfde0f74-8705-4a2f-ba83-a5b99581c294    cart
Time taken: 0.273 seconds, Fetched: 5 row(s)
```

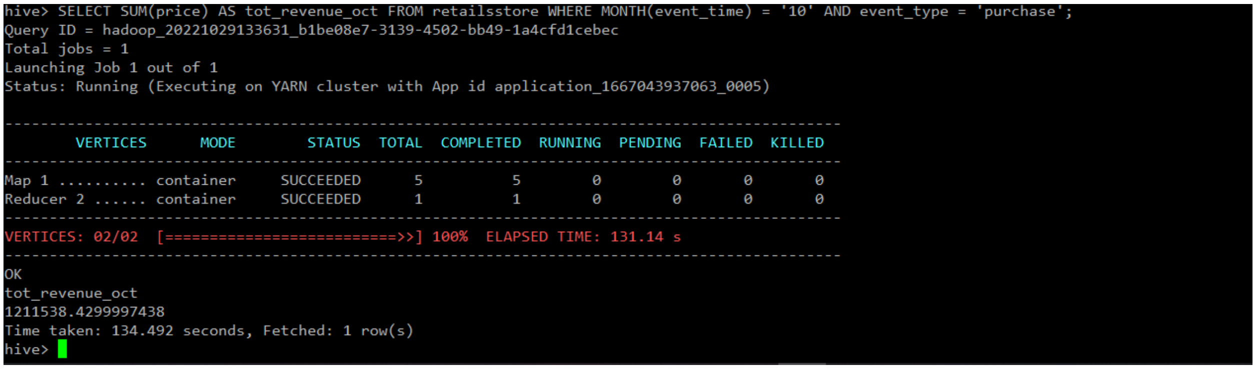
Time taken to retrieve the first 5 rows of data **after** optimization is **0.273 seconds (above screenshot)**

ANSWERING GIVEN QUESTIONS:

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

Base table:

SELECT SUM(price) AS tot_revenue_oct FROM retailsstore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';

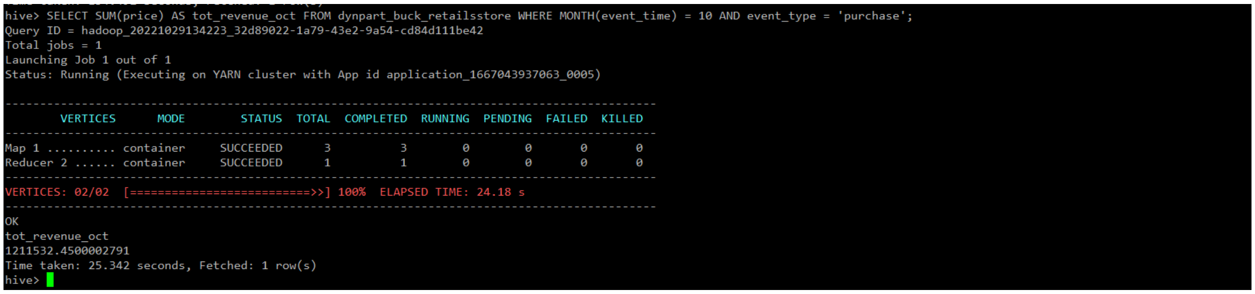


```
hive> SELECT SUM(price) AS tot_revenue_oct FROM retailsstore WHERE MONTH(event_time) = '10' AND event_type = 'purchase';
Query ID = hadoop_20221029133631_b1be08e7-3139-4502-bb49-1a4cfd1cebec
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 131.14 s
----------------------------------------------------------------------------------------------
OK
tot_revenue_oct
1211538.4299997438
Time taken: 134.492 seconds, Fetched: 1 row(s)
hive> 
```

Time taken is **134.492 seconds**

Optimized table:

SELECT SUM(price) AS tot_revenue_oct FROM dynpart_buck_retailsstore WHERE MONTH(event_time) = 10 AND event_type = 'purchase';



```
hive> SELECT SUM(price) AS tot_revenue_oct FROM dynpart_buck_retailsstore WHERE MONTH(event_time) = 10 AND event_type = 'purchase';
Query ID = hadoop_20221029134223_32d89022-1a79-43e2-9a54-cd84d111be42
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 24.18 s
----------------------------------------------------------------------------------------------
OK
tot_revenue_oct
1211532.4500002791
Time taken: 25.342 seconds, Fetched: 1 row(s)
hive>
```

Time taken with optimized table is **25.342 seconds**

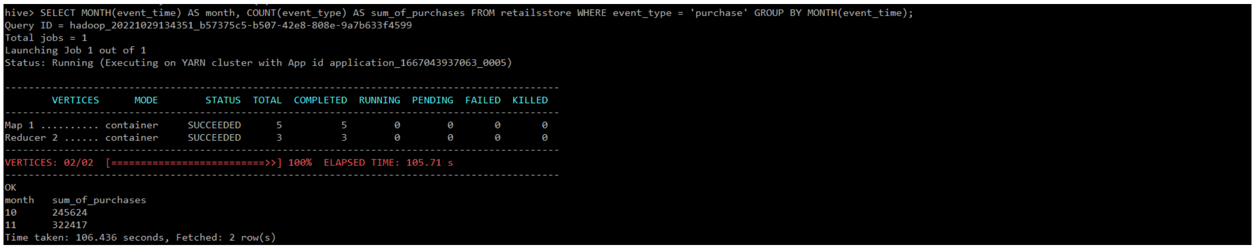
Insights:

1. The total revenue generated based on Purchase made in the month of October is 1,211,538.43 /-
2. Non-optimized table query took the execution time of 134.492 seconds whereas optimized table query took execution time of 25.342 seconds. We can see there is a significant drop in the execution time of the same query.
3. Hence, optimized table gives better performance in execution time.

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

Base Query:

SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM retailsstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);

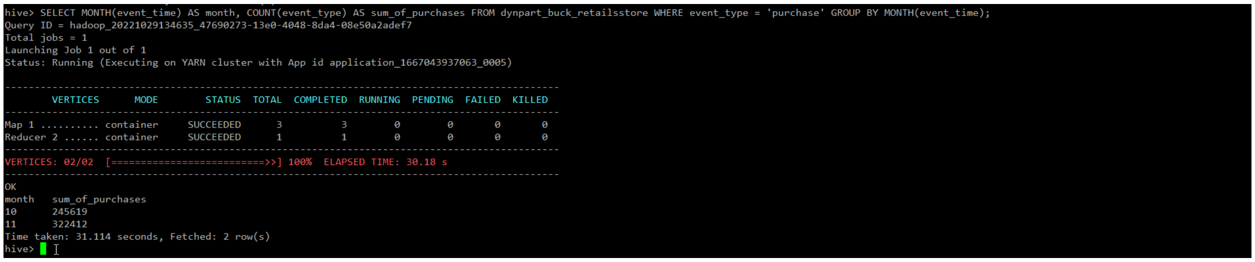


```
hive> SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM retailsstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);
Query ID = hadoop_20221029134351_b57375c5-b507-42e8-808e-9a7b633f4599
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 105.71 s
----------------------------------------------------------------------------------------------
OK
month   sum_of_purchases
10      245624
11      322417
Time taken: 106.436 seconds, Fetched: 2 row(s)
```

Time taken is **106.436 seconds**

Optimized table:

SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM dynpart_buck_retailsstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);



```
hive> SELECT MONTH(event_time) AS month, COUNT(event_type) AS sum_of_purchases FROM dynpart_buck_retailsstore WHERE event_type = 'purchase' GROUP BY MONTH(event_time);
Query ID = hadoop_20221029134635_47690273-13e0-4048-8da4-08e50a2adef7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 30.18 s
----------------------------------------------------------------------------------------------
OK
month   sum_of_purchases
10      245619
11      322412
Time taken: 31.114 seconds, Fetched: 2 row(s)
hive>
```

Time taken is **31.114 seconds**

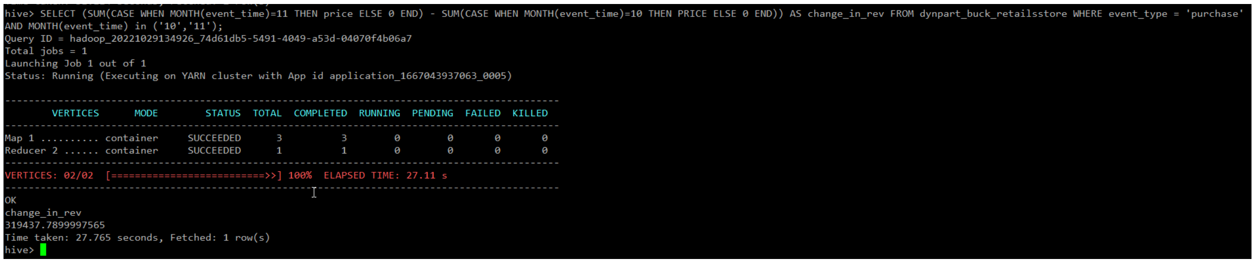
<u>Insights:</u>

- Sum of purchases made in the month of October is 245624 and in the month of November 322417, which means number of purchases are increased in November month
- Non-optimized table query took the execution time of 106.436 seconds whereas optimized table query took execution time of 31.114 seconds. We can see there is a significant drop in the execution time of the same query.
- Hence, with proper partitioning and bucketing on table we can reduce execution time.

Using Optimized table from below questions onwards:

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

SELECT (SUM(CASE WHEN MONTH(event_time)=11 THEN price ELSE 0 END) - SUM(CASE WHEN MONTH(event_time)=10 THEN PRICE ELSE 0 END)) AS change_in_rev FROM dynpart_buck_retailsstore WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11');



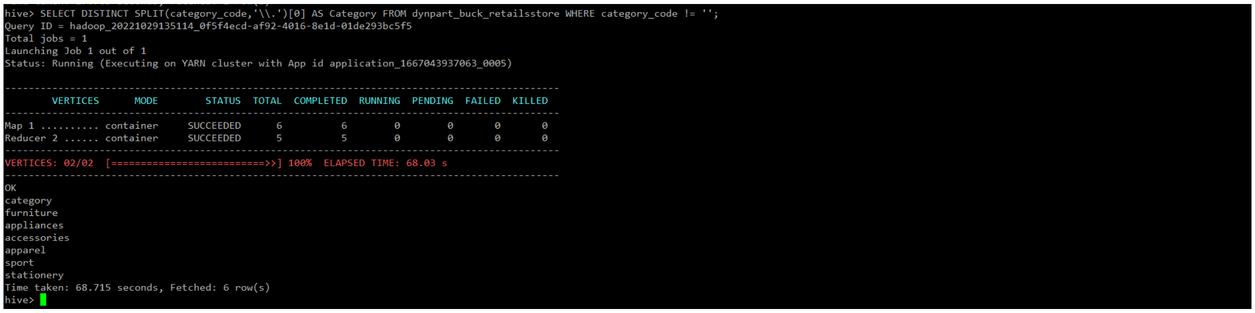
```
hive> SELECT (SUM(CASE WHEN MONTH(event_time)=11 THEN price ELSE 0 END) - SUM(CASE WHEN MONTH(event_time)=10 THEN PRICE ELSE 0 END)) AS change_in_rev FROM dynpart_buck_retailsstore WHERE event_type = 'purchase'
AND MONTH(event_time) in ('10','11');
Query ID = hadoop_20221029134926_74d61db5-5491-4049-a53d-04070f4b06a7
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 27.11 s
----------------------------------------------------------------------------------------------
OK
change_in_rev
319437.7899997565
Time taken: 27.765 seconds, Fetched: 1 row(s)
hive>
```

<u>Insights:</u>

1. Time taken to execute the query is 27.765 seconds
2. Revenue increased in November by 319437.789 from October

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

```
SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Category
FROM dynpart_buck_retailsstore
WHERE category_code != '';
```



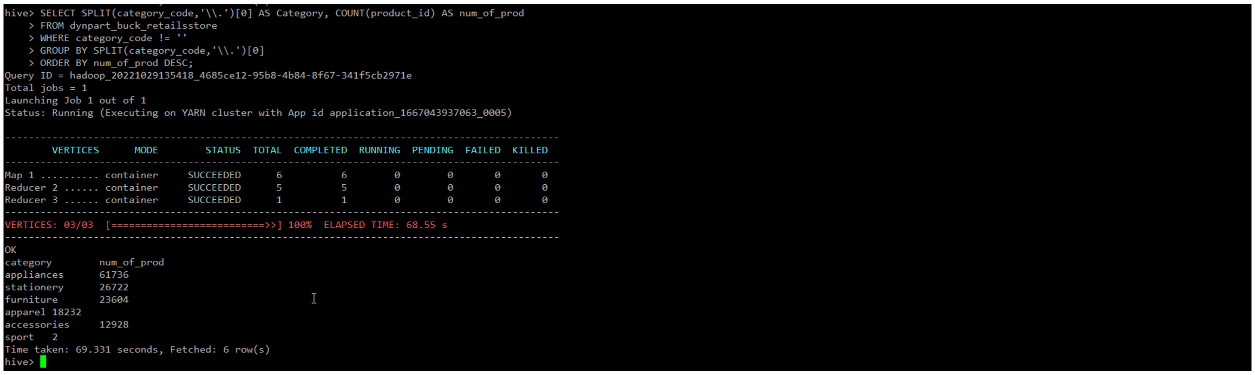
```
hive> SELECT DISTINCT SPLIT(category_code,'\\.')[0] AS Category FROM dynpart_buck_retailsstore WHERE category_code != '';
Query ID = hadoop_20221029135114_0f5f4ecd-af92-4016-8e1d-01de293bc5f5
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 68.03 s
----------------------------------------------------------------------------------------------
OK
category
furniture
appliances
accessories
apparel
sport
stationery
Time taken: 68.715 seconds, Fetched: 6 row(s)
hive>
```

<u>Insights:</u>

1. Time taken to execute the query is 68.715 seconds
2. Total we got 6 distinct categories are – furniture, appliances, accessories, apparel, sport, stationary.

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

```
SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS num_of_prod
FROM dynpart_buck_retailsstore
WHERE category_code != ''
GROUP BY SPLIT(category_code,'\\.')[0]
ORDER BY num_of_prod DESC;
```



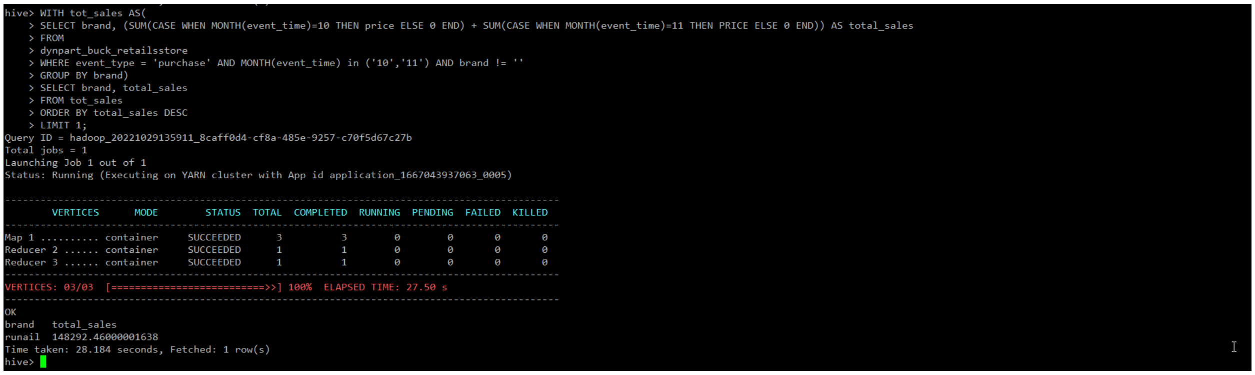
```
hive> SELECT SPLIT(category_code,'\\.')[0] AS Category, COUNT(product_id) AS num_of_prod
    > FROM dynpart_buck_retailsstore
    > WHERE category_code != ''
    > GROUP BY SPLIT(category_code,'\\.')[0]
    > ORDER BY num_of_prod DESC;
Query ID = hadoop_20221029135418_4685ce12-95b8-4b84-8f67-341f5cb2971e
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      6          6        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      5          5        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 68.55 s
----------------------------------------------------------------------------------------------
OK
category        num_of_prod
appliances      61736
stationery      26722
furniture       23604
apparel 18232
accessories     12928
sport   2
Time taken: 69.331 seconds, Fetched: 6 row(s)
hive>
```

<u>Insights:</u>

1. Time taken to execute the query is 69.331 seconds
2. Appliances are having highest number of products available with 61736 compared to other categories.
3. Stationary and Furniture categories are almost equally registered with available ranges from 23000 to 27000.
4. Sports category is least available with 2 products.

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

```
WITH tot_sales AS(
SELECT brand, (SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) +
SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)) AS total_sales
FROM dynpart_buck_retailsstore
WHERE event_type = 'purchase' AND MONTH(event_time) in
('10','11') AND brand != ''
GROUP BY brand)
SELECT brand, total_sales
FROM tot_sales
ORDER BY total_sales DESC
LIMIT 1;
```



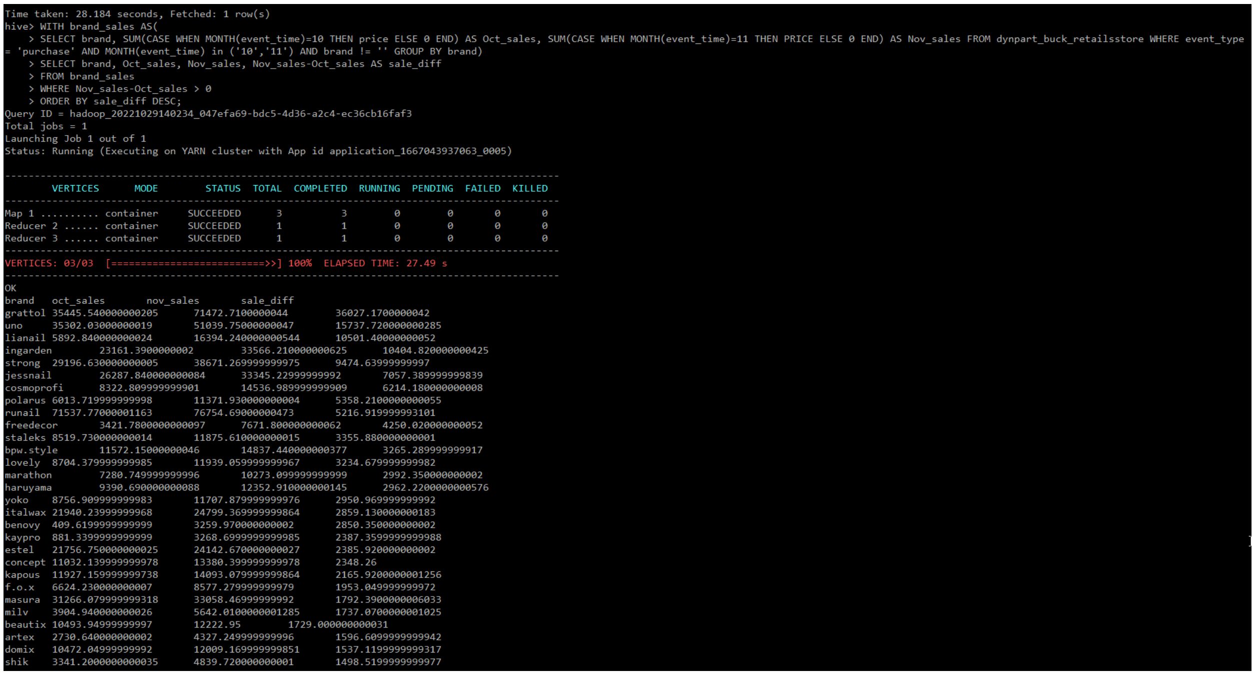
```
hive> WITH tot_sales AS(
    > SELECT brand, (SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) + SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END)) AS total_sales
    > FROM
    > dynpart_buck_retailsstore
    > WHERE event_type = 'purchase' AND MONTH(event_time) in ('10','11') AND brand != ''
    > GROUP BY brand)
    > SELECT brand, total_sales
    > FROM tot_sales
    > ORDER BY total_sales DESC
    > LIMIT 1;
Query ID = hadoop_20221029135911_8caff0d4-cf8a-485e-9257-c70f5d67c27b
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 27.50 s
----------------------------------------------------------------------------------------------
OK
brand   total_sales
runail  148292.46000001638
Time taken: 28.184 seconds, Fetched: 1 row(s)
hive>
```

Insights:

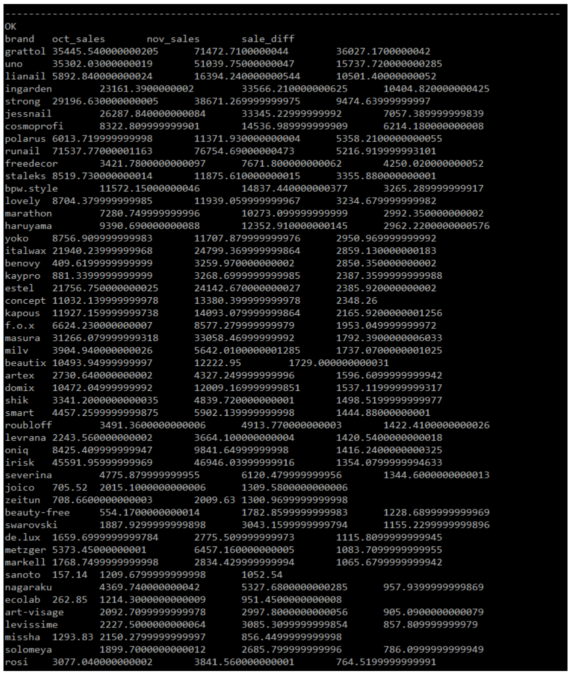
1. Runail is the brand that has the highest sales in total of both the months October and November.
2. It seems that Runail brand has high popularity among cosmetic lovers and bringing in more.
3. Products related to Runail brand could help in increasing their profit.

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

```
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 dynpart_buck_retailsstore WHERE event_type = 'purchase' AND
MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)
SELECT brand, Oct_sales, Nov_sales, Nov_sales-Oct_sales AS sale_diff
FROM brand_sales
WHERE Nov_sales-Oct_sales > 0
ORDER BY sale_diff DESC;
```

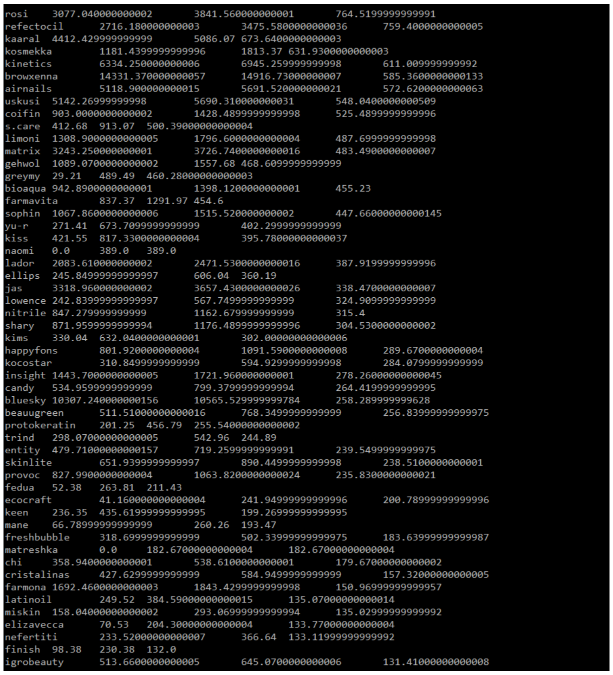


```
Time taken: 28.184 seconds, Fetched: 1 row(s)
hive> WITH brand_sales AS(
    > SELECT brand, SUM(CASE WHEN MONTH(event_time)=10 THEN price ELSE 0 END) AS Oct_sales, SUM(CASE WHEN MONTH(event_time)=11 THEN PRICE ELSE 0 END) AS Nov_sales FROM dynpart_buck_retailsstore WHERE event_type
= 'purchase' AND MONTH(event_time) in ('10','11') AND brand != '' GROUP BY brand)
    > SELECT brand, Oct_sales, Nov_sales, Nov_sales-Oct_sales AS sale_diff
    > FROM brand_sales
    > WHERE Nov_sales-Oct_sales > 0
    > ORDER BY sale_diff DESC;
Query ID = hadoop_20221029140234_047efa69-bdc5-4d36-a2c4-ec36cb16faf3
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 27.49 s
----------------------------------------------------------------------------------------------
OK
brand   oct_sales       nov_sales       sale_diff
grattol 35445.540000000205      71472.7100000044        36027.1700000042
uno     35302.03000000019       51039.75000000047       15737.720000000285
lianail 5892.840000000024       16394.240000000544      10501.40000000052
ingarden        23161.3900000002        33566.210000000625      10404.820000000425
strong  29196.630000000005      38671.269999999975      9474.63999999997
jessnail        26287.840000000084      33345.22999999992       7057.389999999839
cosmoprofi      8322.809999999901       14536.989999999909      6214.180000000008
polarus 6013.719999999998       11371.930000000004      5358.2100000000055
runail  71537.77000001163       76754.69000000473       5216.919999993101
freedecor       3421.7800000000097      7671.800000000062       4250.020000000052
staleks 8519.730000000014       11875.610000000015      3355.880000000001
bpw.style       11572.15000000046       14837.440000000377      3265.289999999917
lovely  8704.379999999985       11939.059999999967      3234.679999999982
marathon        7280.749999999996       10273.099999999999      2992.350000000002
haruyama        9390.690000000088       12352.910000000145      2962.2200000000576
yoko    8756.909999999983       11707.879999999976      2950.969999999992
italwax 21940.23999999968       24799.369999999864      2859.130000000183
benovy  409.6199999999999       3259.970000000002       2850.350000000002
kaypro  881.3399999999999       3268.6999999999985      2387.3599999999988
estel   21756.750000000025      24142.670000000027      2385.920000000002
concept 11032.139999999978      13380.399999999978      2348.26
kapous  11927.159999999738      14093.079999999864      2165.9200000001256
f.o.x   6624.230000000007       8577.279999999979       1953.049999999972
masura  31266.079999999318      33058.46999999992       1792.3900000006033
milv    3904.940000000026       5642.0100000001285      1737.0700000001025
beautix 10493.94999999997       12222.95        1729.000000000031
artex   2730.640000000002       4327.249999999996       1596.6099999999942
domix   10472.04999999992       12009.169999999851      1537.1199999999317
shik    3341.2000000000035      4839.720000000001       1498.5199999999977
```

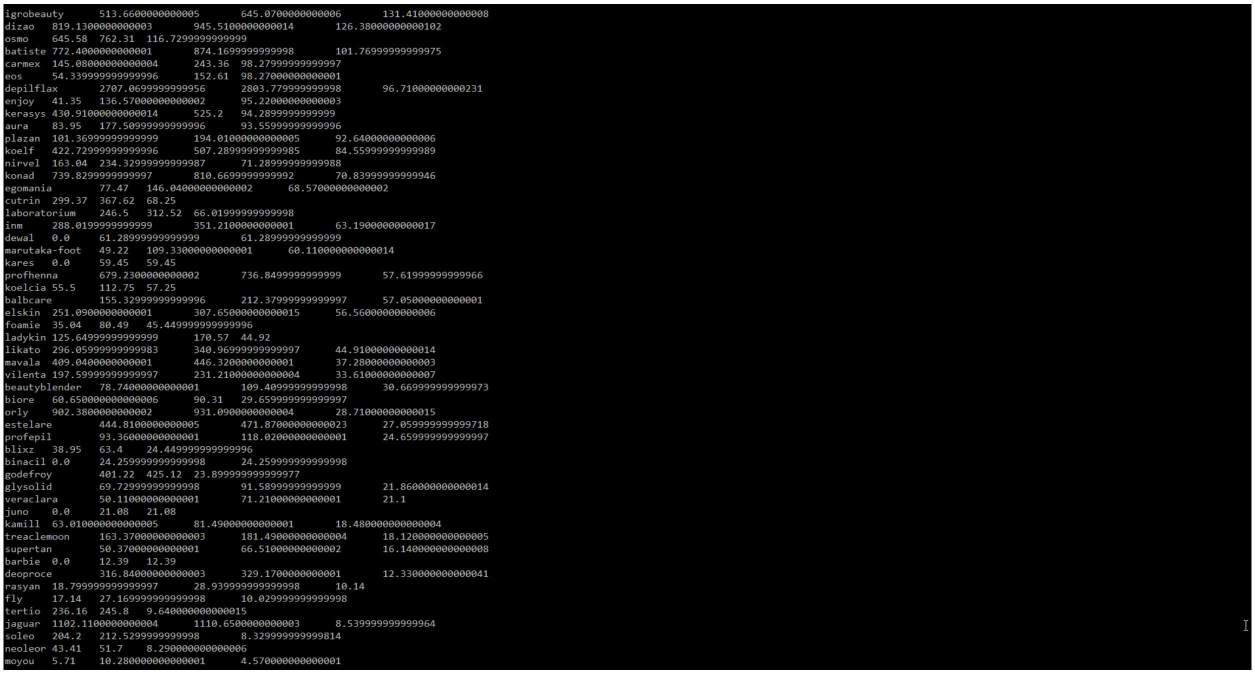


```
----------------------------------------------------------------------------------------------
OK
brand   oct_sales       nov_sales       sale_diff
grattol 35445.540000000205      71472.7100000044        36027.1700000042
uno     35302.03000000019       51039.75000000047       15737.720000000285
lianail 5892.840000000024       16394.240000000544      10501.40000000052
ingarden        23161.3900000002        33566.210000000625      10404.820000000425
strong  29196.630000000005      38671.269999999975      9474.63999999997
jessnail        26287.840000000084      33345.22999999992       7057.389999999839
cosmoprofi      8322.809999999901       14536.989999999909      6214.180000000008
polarus 6013.719999999998       11371.930000000004      5358.2100000000055
runail  71537.77000001163       76754.69000000473       5216.919999993101
freedecor       3421.7800000000097      7671.800000000062       4250.020000000052
staleks 8519.730000000014       11875.610000000015      3355.880000000001
bpw.style       11572.15000000046       14837.440000000377      3265.289999999917
lovely  8704.379999999985       11939.059999999967      3234.679999999982
marathon        7280.749999999996       10273.099999999999      2992.350000000002
haruyama        9390.690000000088       12352.910000000145      2962.2200000000576
yoko    8756.909999999983       11707.879999999976      2950.969999999992
italwax 21940.23999999968       24799.369999999864      2859.130000000183
benovy  409.6199999999999       3259.970000000002       2850.350000000002
kaypro  881.3399999999999       3268.6999999999985      2387.3599999999988
estel   21756.750000000025      24142.670000000027      2385.920000000002
concept 11032.139999999978      13380.399999999978      2348.26
kapous  11927.159999999738      14093.079999999864      2165.9200000001256
f.o.x   6624.230000000007       8577.279999999979       1953.049999999972
masura  31266.079999999318      33058.46999999992       1792.3900000006033
milv    3904.940000000026       5642.0100000001285      1737.0700000001025
beautix 10493.94999999997       12222.95        1729.000000000031
artex   2730.640000000002       4327.249999999996       1596.6099999999942
domix   10472.04999999992       12009.169999999851      1537.1199999999317
shik    3341.2000000000035      4839.720000000001       1498.5199999999977
```

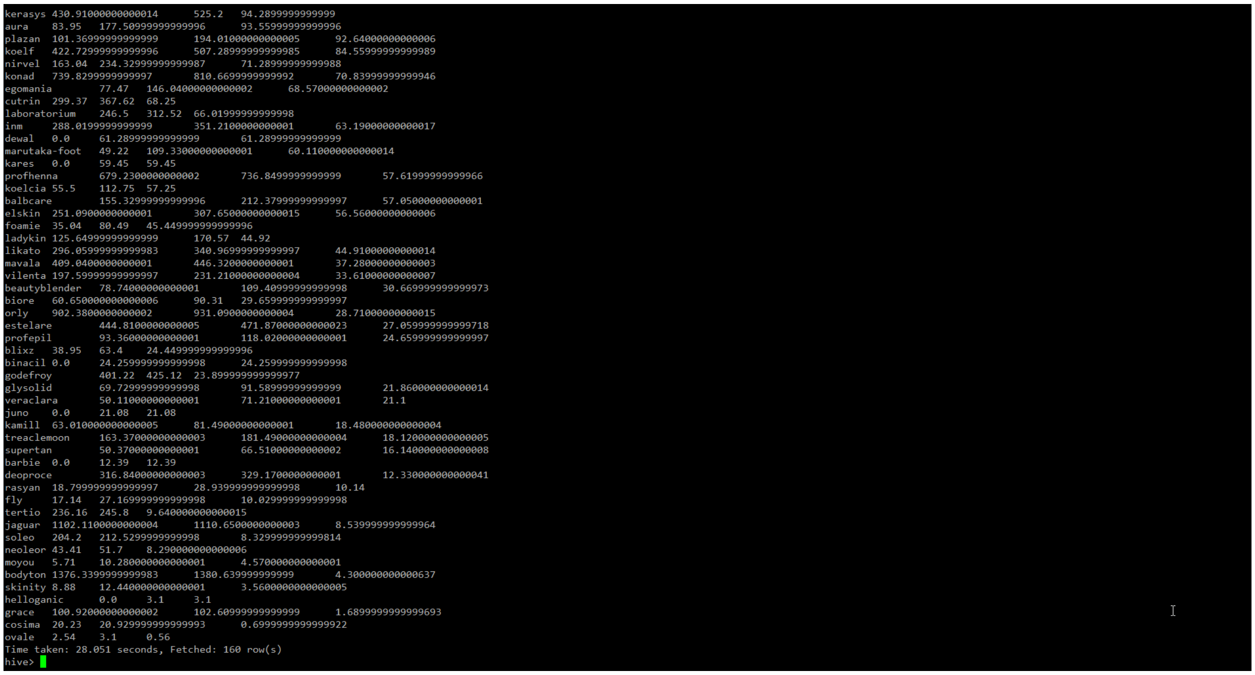
```
smart   4457.2599999999875      5902.139999999998       1444.88000000001
roubloff        3491.3600000000006      4913.770000000003       1422.4100000000026
levrana 2243.560000000002       3664.100000000004       1420.5400000000018
oniq    8425.409999999947       9841.64999999998        1416.2400000000325
irisk   45591.95999999969       46946.03999999916       1354.0799999994633
severina        4775.879999999955       6120.479999999956       1344.6000000000013
joico   705.52  2015.1000000000006      1309.5800000000006
zeitun  708.6600000000003       2009.63 1300.9699999999998
beauty-free     554.1700000000014       1782.8599999999983      1228.6899999999969
swarovski       1887.9299999999898      3043.1599999999794      1155.2299999999896
de.lux  1659.6999999999784      2775.509999999973       1115.8099999999945
metzger 5373.45000000001        6457.160000000005       1083.7099999999955
markell 1768.7499999999998      2834.429999999994       1065.6799999999942
sanoto  157.14  1209.6799999999998      1052.54
nagaraku        4369.740000000042       5327.6800000000285      957.9399999999869
ecolab  262.85  1214.3000000000009      951.4500000000008
art-visage      2092.7099999999978      2997.8000000000056      905.0900000000079
levissime       2227.5000000000064      3085.3099999999854      857.809999999979
missha  1293.83 2150.2799999999997      856.4499999999998
solomeya        1899.7000000000012      2685.799999999996       786.0999999999949
rosi    3077.040000000002       3841.560000000001       764.5199999999991
```



```
rosi    3077.040000000002       3841.560000000001       764.5199999999991
refectocil      2716.180000000003       3475.5800000000036      759.4000000000005
kaaral  4412.429999999999       5086.07 673.6400000000003
kosmekka        1181.4399999999996      1813.37 631.9300000000003
kinetics        6334.250000000006       6945.259999999998       611.009999999992
browxenna       14331.370000000057      14916.73000000007       585.3600000000133
airnails        5118.900000000015       5691.520000000021       572.6200000000063
uskusi  5142.26999999998        5690.310000000031       548.0400000000509
coifin  903.0000000000002       1428.4899999999998      525.4899999999996
s.care  412.68  913.07  500.39000000000004
limoni  1308.9000000000005      1796.6000000000004      487.6999999999998
matrix  3243.250000000001       3726.7400000000016      483.4900000000007
gehwol  1089.0700000000002      1557.68 468.6099999999999
greymy  29.21   489.49  460.28000000000003
bioaqua 942.8900000000001       1398.1200000000001      455.23
farmavita       837.37  1291.97 454.6
sophin  1067.8600000000006      1515.520000000002       447.66000000000145
yu-r    271.41  673.7099999999999       402.2999999999999
kiss    421.55  817.3300000000004       395.78000000000037
naomi   0.0     389.0   389.0
lador   2083.610000000002       2471.5300000000016      387.9199999999996
ellips  245.84999999999997      606.04  360.19
jas     3318.960000000002       3657.4300000000026      338.4700000000007
lowence 242.83999999999997      567.7499999999999       324.9099999999999
nitrile 847.279999999999        1162.679999999999       315.4
shary   871.9599999999994       1176.4899999999996      304.5300000000002
kims    330.04  632.0400000000001       302.00000000000006
happyfons       801.9200000000004       1091.5900000000008      289.6700000000004
kocostar        310.8499999999999       594.9299999999998       284.0799999999999
insight 1443.7000000000005      1721.960000000001       278.26000000000045
candy   534.9599999999999       799.3799999999994       264.4199999999995
bluesky 10307.240000000156      10565.529999999784      258.289999999628
beauugreen      511.51000000000016      768.3499999999999       256.83999999999975
protokeratin    201.25  456.79  255.54000000000002
trind   298.07000000000005      542.96  244.89
entity  479.71000000000157      719.2599999999991       239.5499999999975
skinlite        651.9399999999997       890.4499999999998       238.5100000000001
provoc  827.9900000000004       1063.8200000000024      235.8300000000021
fedua   52.38   263.81  211.43
ecocraft        41.160000000000004      241.94999999999996      200.78999999999996
keen    236.35  435.6199999999995       199.26999999999995
mane    66.78999999999999       260.26  193.47
freshbubble     318.6999999999999       502.33999999999975      183.63999999999987
matreshka       0.0     182.67000000000004      182.67000000000004
chi     358.9400000000001       538.6100000000001       179.67000000000002
cristalinas     427.6299999999999       584.9499999999999       157.32000000000005
farmona 1692.4600000000003      1843.4299999999998      150.96999999999957
latinoil        249.52  384.59000000000015      135.07000000000014
miskin  158.04000000000002      293.06999999999994      135.02999999999992
elizavecca      70.53   204.30000000000004      133.77000000000004
nefertiti       233.52000000000007      366.64  133.11999999999992
finish  98.38   230.38  132.0
igrobeauty      513.6600000000005       645.0700000000006       131.41000000000008
```



```
igrobeauty      513.6600000000005       645.0700000000006       131.41000000000008
dizao   819.1300000000003       945.5100000000014       126.38000000000102
osmo    645.58  762.31  116.7299999999999
batiste 772.4000000000001       874.1699999999998       101.76999999999975
carmex  145.08000000000004      243.36  98.27999999999997
eos     54.339999999999996      152.61  98.27000000000001
depilflax       2707.0699999999956      2803.779999999998       96.71000000000231
enjoy   41.35   136.57000000000002      95.22000000000003
kerasys 430.91000000000014      525.2   94.2899999999999
aura    83.95   177.50999999999996      93.55999999999996
plazan  101.36999999999999      194.01000000000005      92.64000000000006
koelf   422.72999999999996      507.28999999999985      84.55999999999989
nirvel  163.04  234.32999999999987      71.28999999999988
konad   739.8299999999997       810.6699999999992       70.83999999999946
egomania        77.47   146.04000000000002      68.57000000000002
cutrin  299.37  367.62  68.25
laboratorium    246.5   312.52  66.01999999999998
inm     288.0199999999999       351.2100000000001       63.19000000000017
dewal   0.0     61.28999999999999       61.28999999999999
marutaka-foot   49.22   109.33000000000001      60.110000000000014
kares   0.0     59.45   59.45
profhenna       679.2300000000002       736.8499999999999       57.61999999999966
koelcia 55.5    112.75  57.25
balbcare        155.32999999999996      212.37999999999997      57.05000000000001
elskin  251.0900000000001       307.65000000000015      56.56000000000006
foamie  35.04   80.49   45.449999999999996
ladykin 125.64999999999999      170.57  44.92
likato  296.05999999999983      340.96999999999997      44.910000000000014
mavala  409.0400000000001       446.3200000000001       37.28000000000003
vilenta 197.59999999999997      231.21000000000004      33.61000000000007
beautyblender   78.74000000000001       109.40999999999998      30.669999999999973
biore   60.650000000000006      90.31   29.659999999999997
orly    902.3800000000002       931.0900000000004       28.71000000000015
estelare        444.8100000000005       471.87000000000023      27.059999999999718
profepil        93.36000000000001       118.02000000000001      24.659999999999997
blixz   38.95   63.4    24.449999999999996
binacil 0.0     24.259999999999998      24.259999999999998
godefroy        401.22  425.12  23.899999999999977
glysolid        69.72999999999998       91.58999999999999       21.860000000000014
veraclara       50.11000000000001       71.21000000000001       21.1
juno    0.0     21.08   21.08
kamill  63.010000000000005      81.49000000000001       18.480000000000004
treaclemoon     163.37000000000003      181.49000000000004      18.120000000000005
supertan        50.37000000000001       66.51000000000002       16.140000000000008
barbie  0.0     12.39   12.39
deoproce        316.84000000000003      329.1700000000001       12.330000000000041
rasyan  18.799999999999997      28.939999999999998      10.14
fly     17.14   27.169999999999998      10.029999999999998
tertio  236.16  245.8   9.640000000000015
jaguar  1102.1100000000004      1110.6500000000003      8.539999999999964
soleo   204.2   212.5299999999998       8.329999999999814
neoleor 43.41   51.7    8.290000000000006
moyou   5.71    10.280000000000001      4.570000000000001
```



```
kerasys 430.91000000000014      525.2   94.2899999999999
aura    83.95   177.50999999999996      93.55999999999996
plazan  101.36999999999999      194.01000000000005      92.64000000000006
koelf   422.72999999999996      507.28999999999985      84.55999999999989
nirvel  163.04  234.32999999999987      71.28999999999988
konad   739.8299999999997       810.6699999999992       70.83999999999946
egomania        77.47   146.04000000000002      68.57000000000002
cutrin  299.37  367.62  68.25
laboratorium    246.5   312.52  66.01999999999998
inm     288.0199999999999       351.2100000000001       63.19000000000017
dewal   0.0     61.28999999999999       61.28999999999999
marutaka-foot   49.22   109.33000000000001      60.110000000000014
kares   0.0     59.45   59.45
profhenna       679.2300000000002       736.8499999999999       57.61999999999966
koelcia 55.5    112.75  57.25
balbcare        155.32999999999996      212.37999999999997      57.05000000000001
elskin  251.0900000000001       307.65000000000015      56.56000000000006
foamie  35.04   80.49   45.449999999999996
ladykin 125.64999999999999      170.57  44.92
likato  296.05999999999983      340.96999999999997      44.910000000000014
mavala  409.0400000000001       446.3200000000001       37.28000000000003
vilenta 197.59999999999997      231.21000000000004      33.61000000000007
beautyblender   78.74000000000001       109.40999999999998      30.669999999999973
biore   60.650000000000006      90.31   29.659999999999997
orly    902.3800000000002       931.0900000000004       28.71000000000015
estelare        444.8100000000005       471.87000000000023      27.059999999999718
profepil        93.36000000000001       118.02000000000001      24.659999999999997
blixz   38.95   63.4    24.449999999999996
binacil 0.0     24.259999999999998      24.259999999999998
godefroy        401.22  425.12  23.899999999999977
glysolid        69.72999999999998       91.58999999999999       21.860000000000014
veraclara       50.11000000000001       71.21000000000001       21.1
juno    0.0     21.08   21.08
kamill  63.010000000000005      81.49000000000001       18.480000000000004
treaclemoon     163.37000000000003      181.49000000000004      18.120000000000005
supertan        50.37000000000001       66.51000000000002       16.140000000000008
barbie  0.0     12.39   12.39
deoproce        316.84000000000003      329.1700000000001       12.330000000000041
rasyan  18.799999999999997      28.939999999999998      10.14
fly     17.14   27.169999999999998      10.029999999999998
tertio  236.16  245.8   9.640000000000015
jaguar  1102.1100000000004      1110.6500000000003      8.539999999999964
soleo   204.2   212.5299999999998       8.329999999999814
neoleor 43.41   51.7    8.290000000000006
moyou   5.71    10.280000000000001      4.570000000000001
```

```
bodyton 1376.3399999999983      1380.639999999999       4.300000000000637
skinity 8.88    12.440000000000001      3.5600000000000005
helloganic      0.0     3.1     3.1
grace   100.92000000000002      102.60999999999999      1.6899999999999693
cosima  20.23   20.929999999999993      0.6999999999999922
ovale   2.54    3.1     0.56
Time taken: 28.051 seconds, Fetched: 160 row(s)
hive>
```

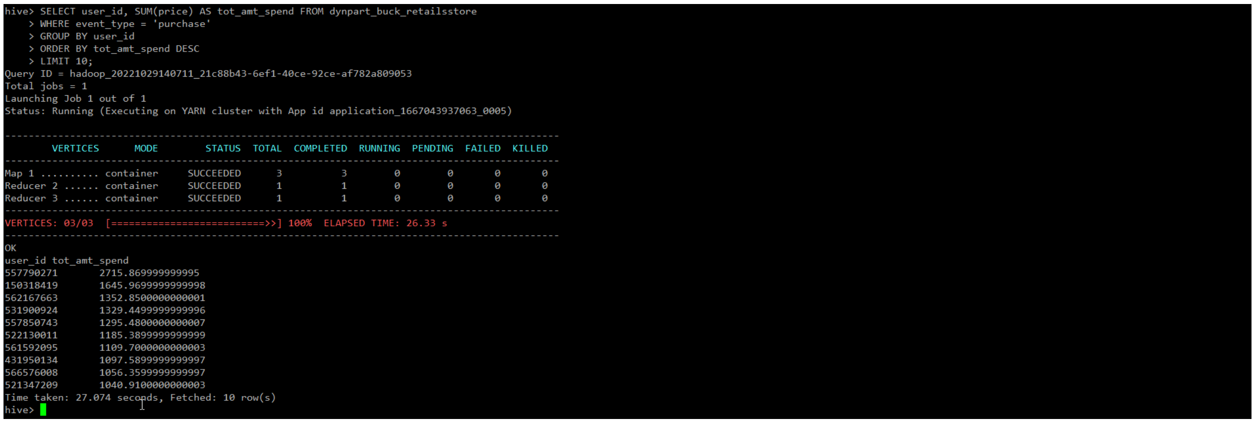
Insights:

1. Here are some 160 brands with increment in the selling from October to November.
2. ‘Grattol’ brand has the highest total increment i.e., 36,027 /- and ‘Ovale’ seems to have the least increment of 0.56 /- from October to November.
3. Among all these brands lists, ‘Runail’ which was the best brand in terms of selling in October and November combined is also in the top 10 brands with high increment for October (71539.28) to November (76758.61) i.e., increment of total 5219.38.

4. This implies that ‘Runail’ is the best and popular brand among all other brands within people.

8. Your company wants to reward the top 10 users of its websites 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 tot_amt_spend FROM dynpart_buck_retailsstore
WHERE event_type = 'purchase'
GROUP BY user_id
ORDER BY tot_amt_spend DESC
LIMIT 10;



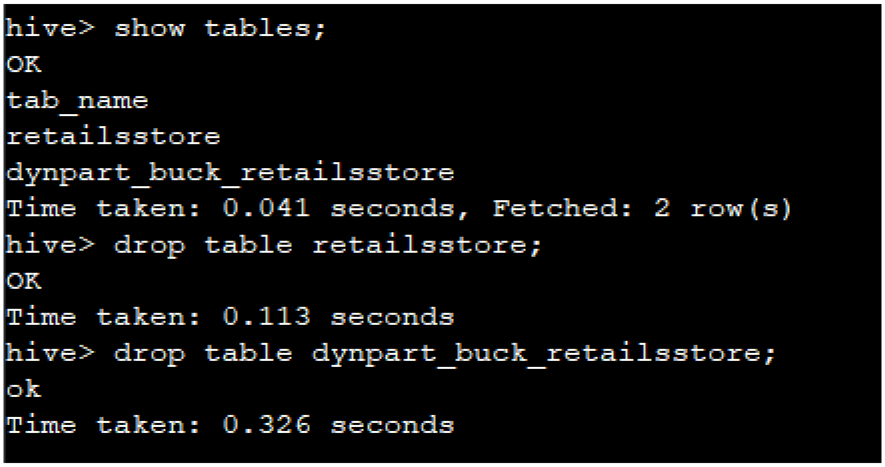
```
hive> SELECT user_id, SUM(price) AS tot_amt_spend FROM dynpart_buck_retailsstore
    > WHERE event_type = 'purchase'
    > GROUP BY user_id
    > ORDER BY tot_amt_spend DESC
    > LIMIT 10;
Query ID = hadoop_20221029140711_21c88b43-6ef1-40ce-92ce-af782a809053
Total jobs = 1
Launching Job 1 out of 1
Status: Running (Executing on YARN cluster with App id application_1667043937063_0005)

----------------------------------------------------------------------------------------------
        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: 26.33 s
----------------------------------------------------------------------------------------------
OK
user_id tot_amt_spend
557790271       2715.869999999995
150318419       1645.9699999999998
562167663       1352.8500000000001
531900924       1329.4499999999996
557850743       1295.4800000000007
522130011       1185.3899999999999
561592095       1109.7000000000003
431950134       1097.5899999999997
566576008       1056.3599999999997
521347209       1040.9100000000003
Time taken: 27.074 seconds, Fetched: 10 row(s)
hive> 
```

<u>Insights:</u>

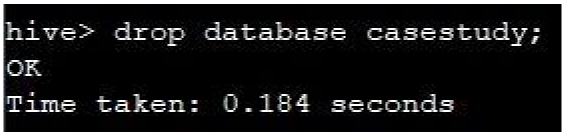
1. Here is the list of the top 10 users or buyers who have spent the most and could be rewarded with a Golden Customer plan to attract more people in the coming future.
2. With the Optimized table the execution time reduced with proper partitioning and bucketing.
3. Time taken to execute this query on optimized table is 27.874 seconds.

Cleaning up:

Once the analysis is completed, we should drop the tables and databases



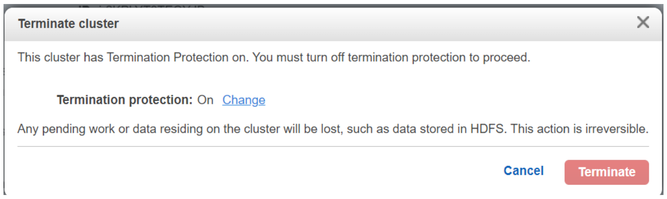
```
hive> show tables;
OK
tab_name
retailsstore
dynpart_buck_retailsstore
Time taken: 0.041 seconds, Fetched: 2 row(s)
hive> drop table retailsstore;
OK
Time taken: 0.113 seconds
hive> drop table dynpart_buck_retailsstore;
ok
Time taken: 0.326 seconds
```



```
hive> drop database casestudy;
OK
Time taken: 0.184 seconds
```

TERMINATION PROCESS:

After completing our analysis, we should terminate the EMR cluster

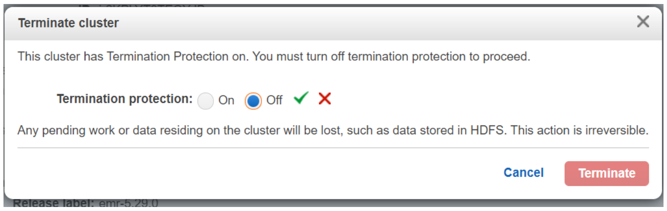


Terminate cluster

This cluster has Termination Protection on. You must turn off termination protection to proceed.

Termination protection: On Change

Any pending work or data residing on the cluster will be lost, such as data stored in HDFS. This action is irreversible.



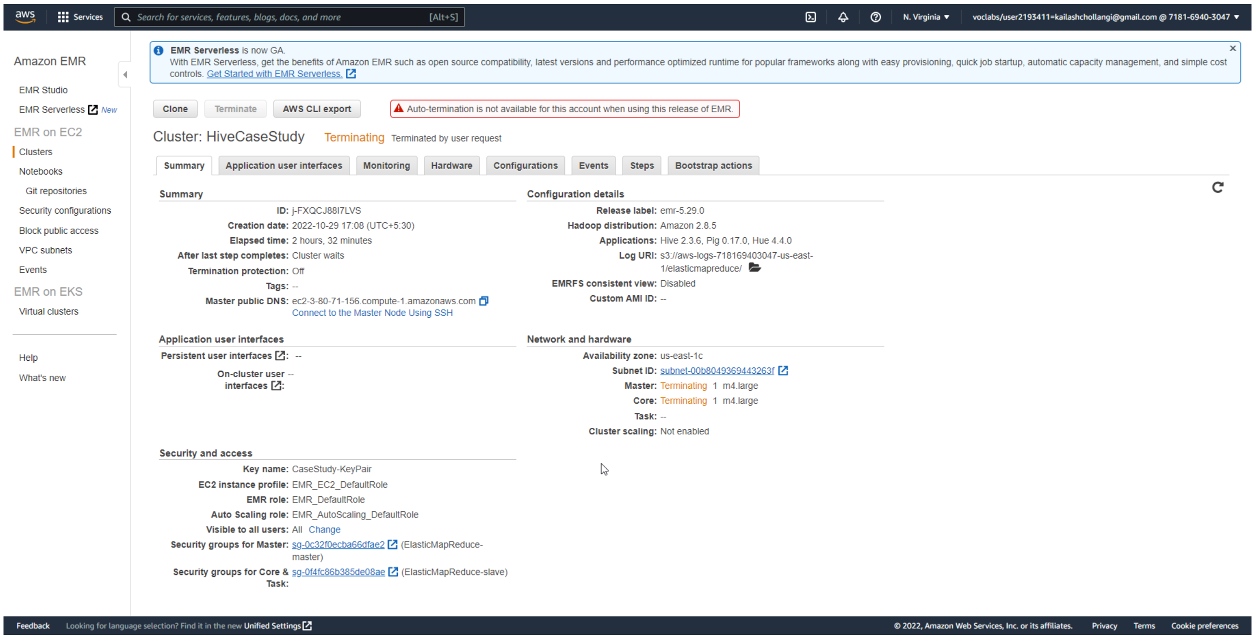
Terminate cluster

This cluster has Termination Protection on. You must turn off termination protection to proceed.

Termination protection: ○ On ● Off ✓ ✗

Any pending work or data residing on the cluster will be lost, such as data stored in HDFS. This action is irreversible.

Release label: emr-5.29.0



EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.

Auto-termination is not available for this account when using this release of EMR.

Cluster: HiveCaseStudy Terminating Terminated by user request

Summary

ID: j-FXQCJ88I7LVS
Creation date: 2022-10-29 17:08 (UTC+5:30)
Elapsed time: 2 hours, 32 minutes
After last step completes: Cluster waits
Termination protection: Off
Tags: --
Master public DNS: ec2-3-80-71-156.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-718169403047-us-east-1/elasticmapreduce/
EMRFS consistent view: Disabled
Custom AMI ID: --

Application user interfaces

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

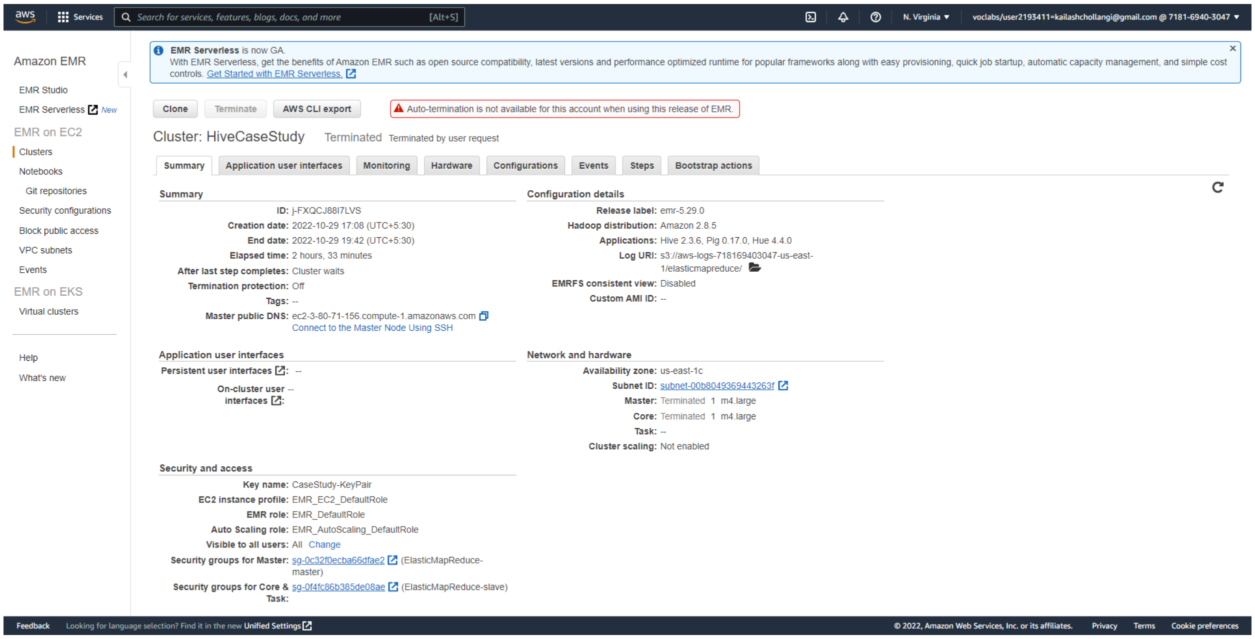
Network and hardware

Availability zone: us-east-1c
Subnet ID: subnet-00b8049369443263f
Master: Terminating 1 m4.large
Core: Terminating 1 m4.large
Task: --
Cluster scaling: Not enabled

Security and access

Key name: CaseStudy-KeyPair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)
Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)

Cluster Terminated!!



EMR Serverless is now GA.
With EMR Serverless, get the benefits of Amazon EMR such as open source compatibility, latest versions and performance optimized runtime for popular frameworks along with easy provisioning, quick job startup, automatic capacity management, and simple cost controls. Get Started with EMR Serverless.

Auto-termination is not available for this account when using this release of EMR.

Cluster: HiveCaseStudy Terminated Terminated by user request

Summary

ID: j-FXQCJ88I7LVS
Creation date: 2022-10-29 17:08 (UTC+5:30)
End date: 2022-10-29 19:42 (UTC+5:30)
Elapsed time: 2 hours, 33 minutes
After last step completes: Cluster waits
Termination protection: Off
Tags: --
Master public DNS: ec2-3-80-71-156.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-718169403047-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: us-east-1c
Subnet ID: subnet-00b8049369443263f
Master: Terminated 1 m4.large
Core: Terminated 1 m4.large
Task: --
Cluster scaling: Not enabled

Security and access

Key name: CaseStudy-KeyPair
EC2 instance profile: EMR_EC2_DefaultRole
EMR role: EMR_DefaultRole
Auto Scaling role: EMR_AutoScaling_DefaultRole
Visible to all users: All Change
Security groups for Master: sg-0c32f0ecba66dfae2 (ElasticMapReduce-master)
Security groups for Core & Task: sg-0f4fc86b385de08ae (ElasticMapReduce-slave)

Thankyou!!