Yog Chaudhary

11727095

ADTA 5240 Week 5'th (harvesting, Storing, And Retrieving Data)

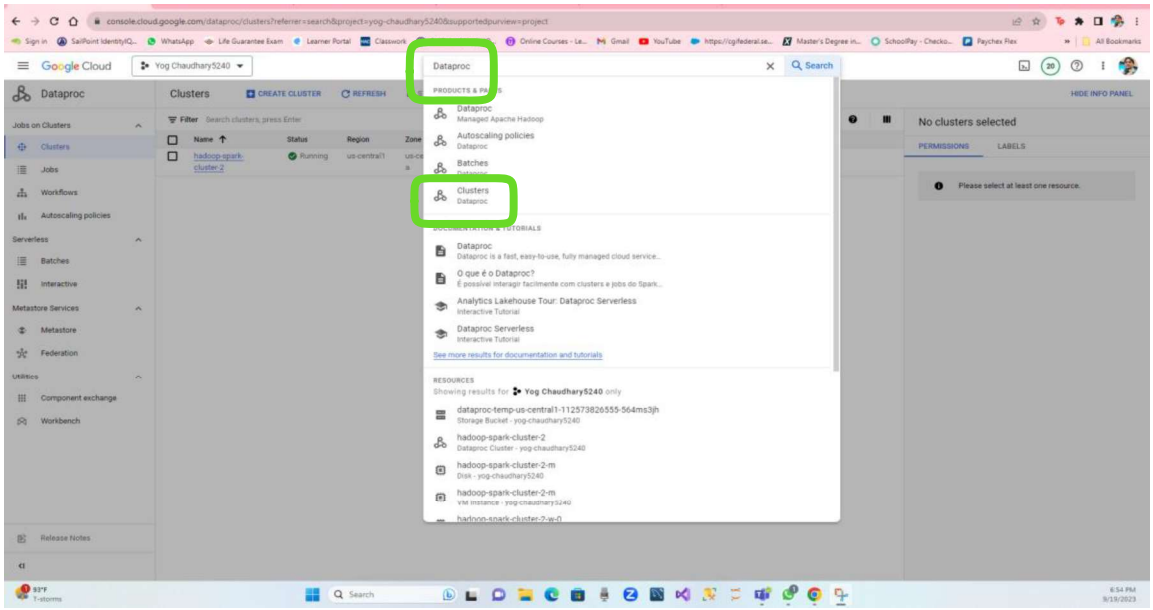
Professor: Dr. Zeynep Orhan **Sep 23, 2023**

University Of North Texas

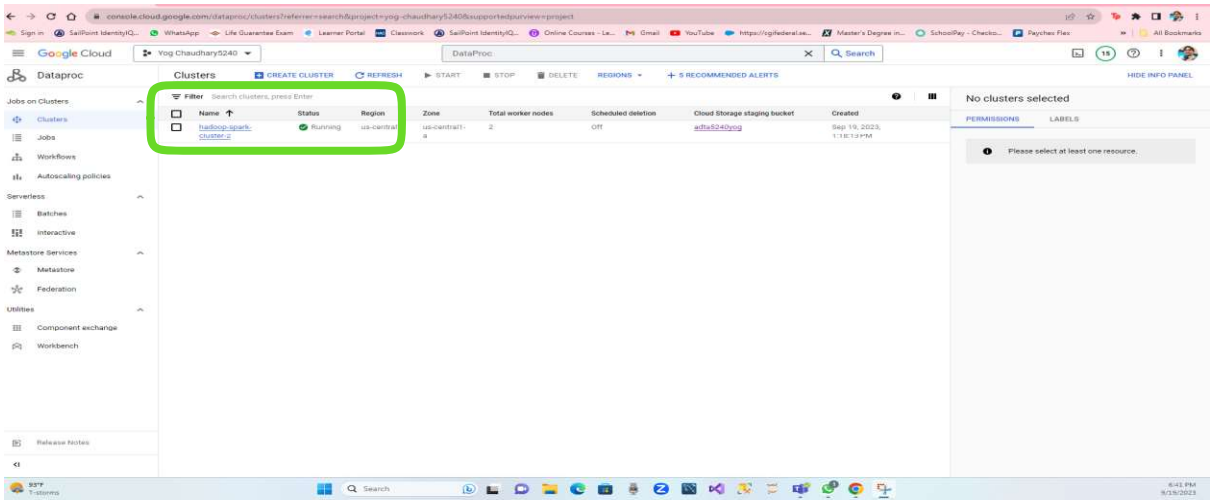
Setting up a Virtual Machine with Linux as the Operating System. Pdf

Step 1. For a Google console.

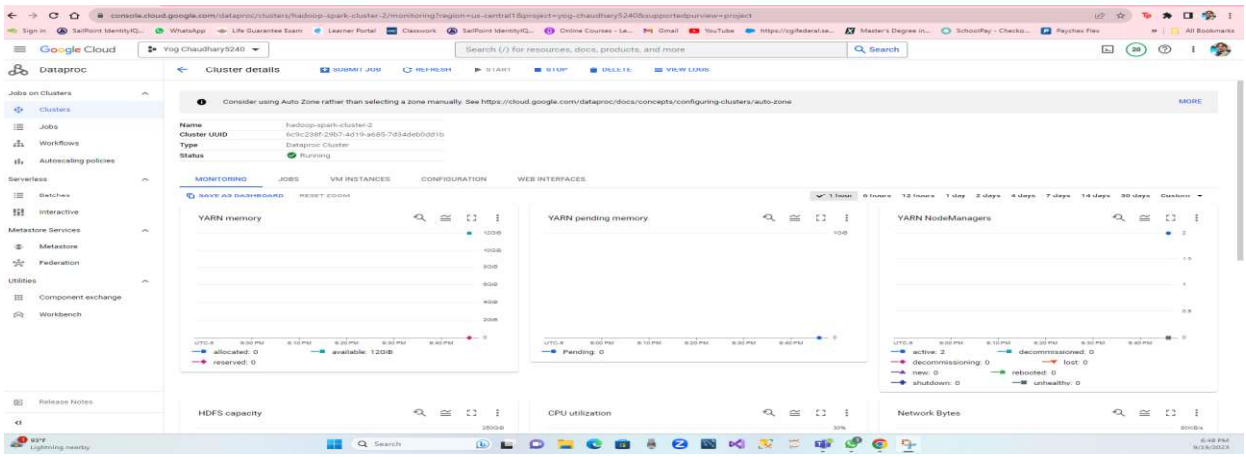
- **Then I clicked on three horizontal lines.**
- **In the search bar I typed "Cluster" and clicked on "Clusters Dataproc."**
- **Hare is screenshot**

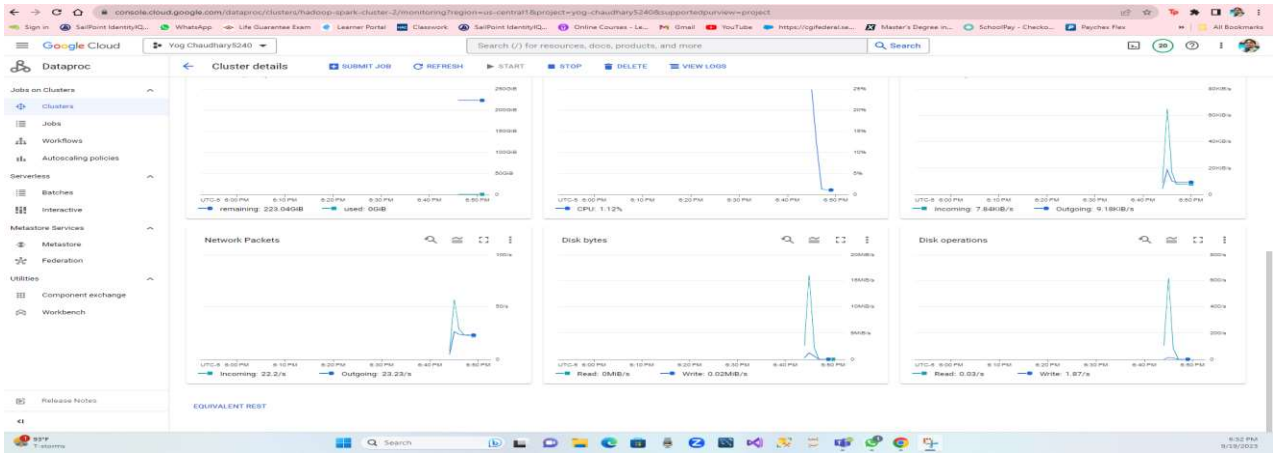


- **After clicking on clusters data proc it will mention the cluster that I have previously created, and it will show that it was running (with a green check mark).**
- **Below shows a screenshot of that.**



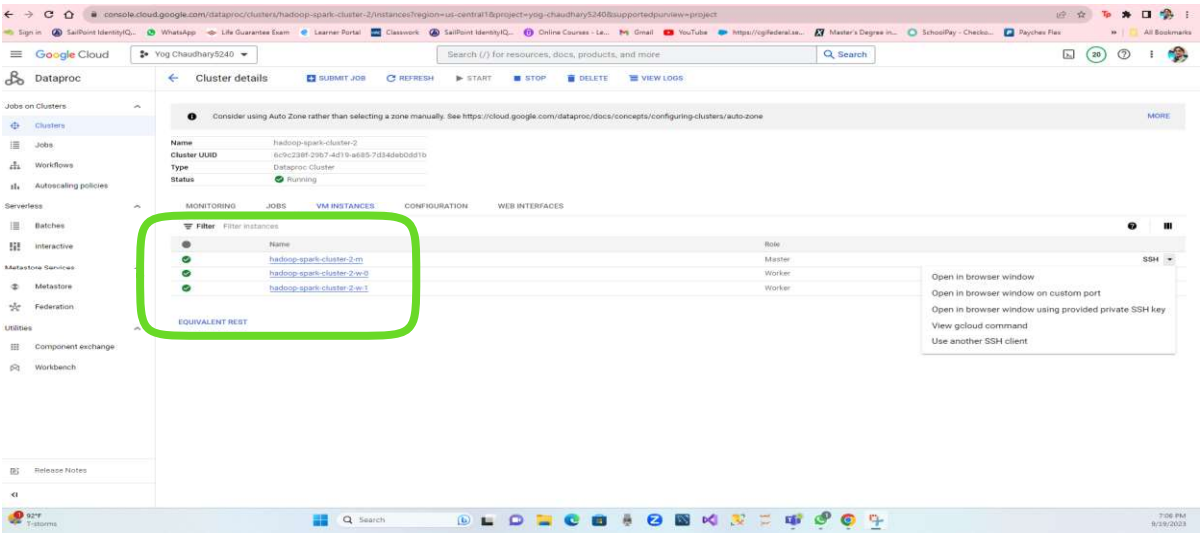
- **While doing this we need to start nodes by checking on the navigation panel.**
- **Clicked on computing engine.**
- **Then clicked on three vertical dots and clicked on start/resume.**
- **Now click.**
- **Below screenshot of that.**



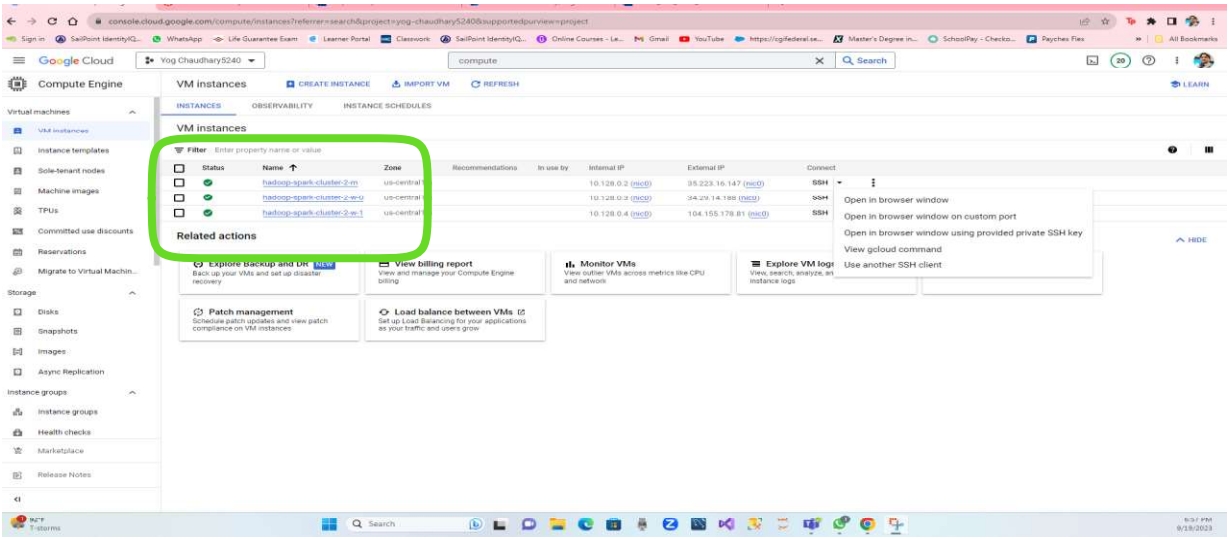


- **When I scroll down it will show how the monitoring is changing according to the usage.**

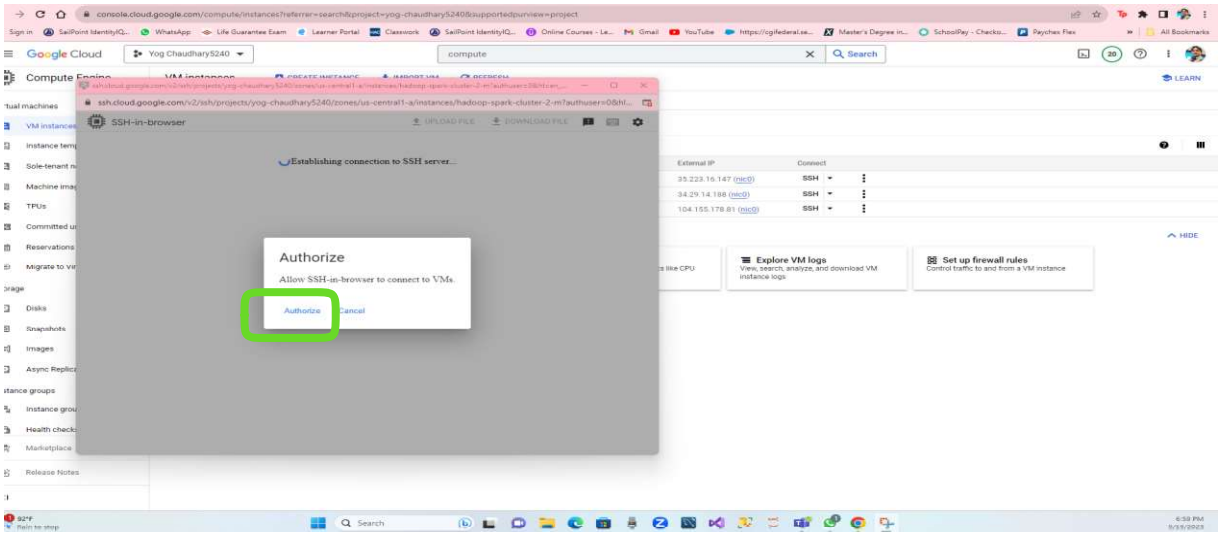
- **By scrolling up and clicking on the virtual machines the screen will show the cluster details.**
- **We will see one master node and one worker node.**
- **After that I accessed the master node through "SSH."**
- **Click on "SSH."**
- **Click on "Open in browser."**



- **Click on the drop-down button next to "SSH."**
- **Click on open in a new browser window.**



- Click authorized

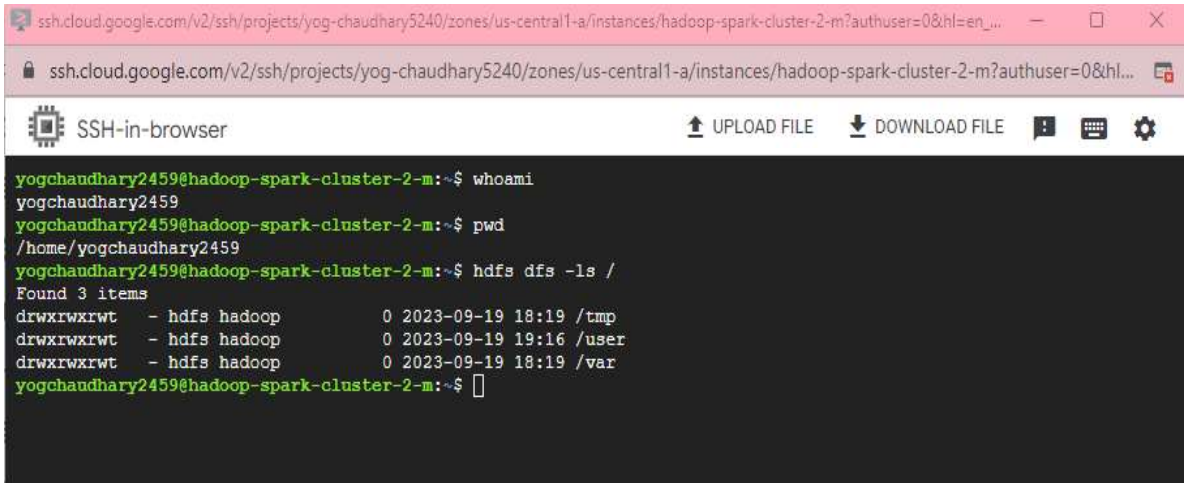


Step 2: We open the SSH terminal. This connects our local computer to the remote server. We use SSH terminal to work with the cluster

- We are going to access the Hadoop Distributed File System (HDFS) of our cluster.
- First, let us go over some of the basic Linux Commands.

→ clear
→ whoami
→ pwd
→ hdfs dfs –ls /

- Then See 3 Folders



Step 3: Let see use folder in hdfs.

→ Hdfs dfs –ls /user

- We use the command in hdfs.
- We see there are many folders in the user directory: hbase (tmp and user) and hive (var).
- All these folders were created for us by the system.

- Now, we see 12 instead created.

```
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user
Found 12 items
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/dataproc
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/hbase
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/hdfs
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/hive
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/mapred
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/pig
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/solr
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/spark
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/yarn
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-19 19:16 /user/yogchaudhary
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/zeppelin
drwxrwxrwt   - hdfs             hadoop          0 2023-09-19 18:19 /user/zookeeper
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```

- We created a new folder with command: hdfs dfs –mkdir /user/yogchaudhary

```
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```

Step 4: Create a subfolder "data" to hold new subfolder which will eventually hold the data files, we created in GCP (userdata and weblog)

- We create another subfolder that is called "data." Remember in hdfs to include the full path.

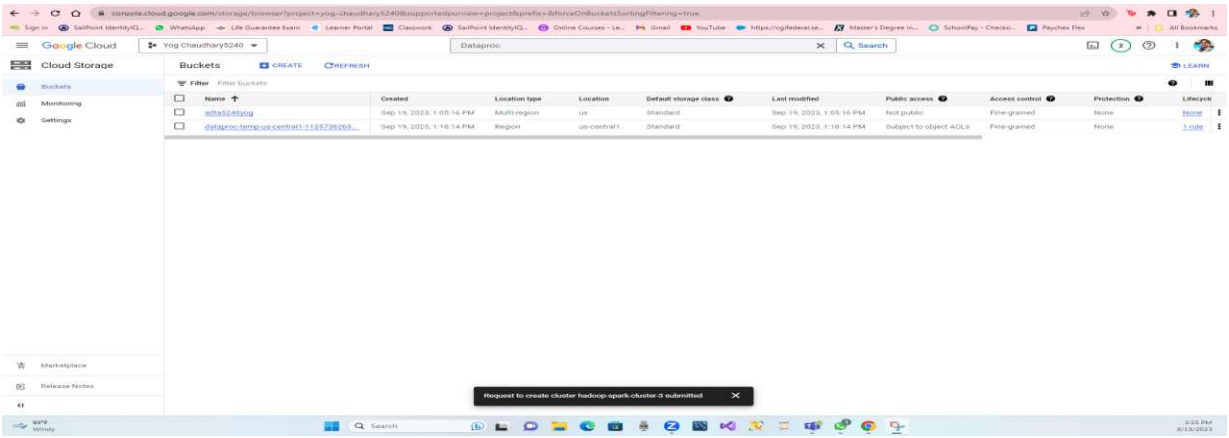
→ hdfs dfs -mkdir /yogchaudhary/data

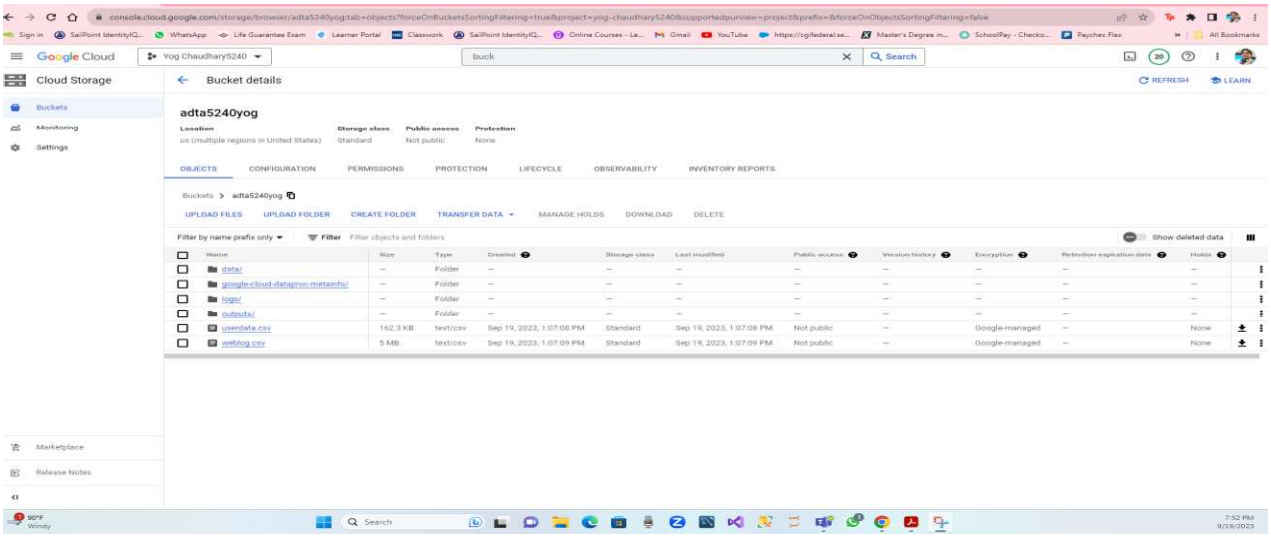
- There is nothing in this subfolder, but let us check to be sure

→ **hdfs dfs -ls /user/yogchaudhary/data**

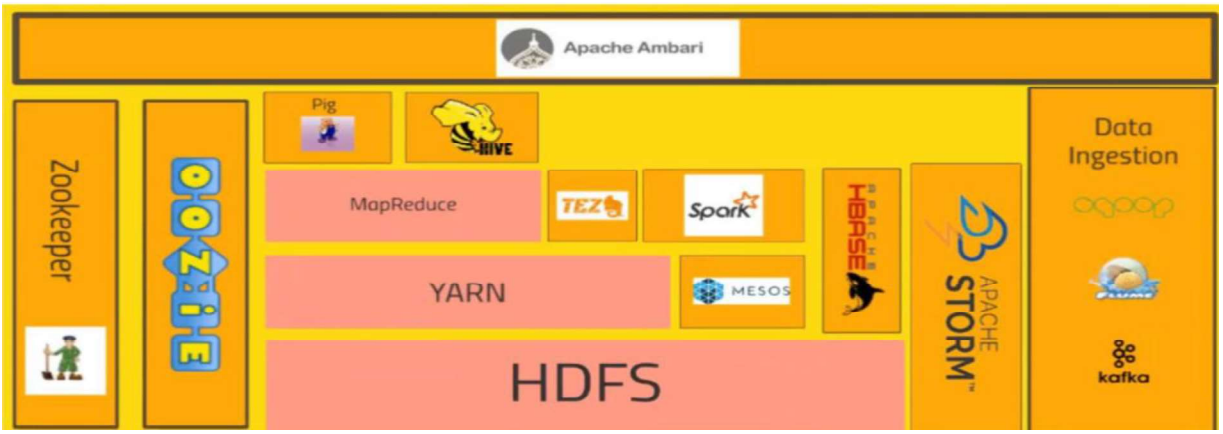
```
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -mkdir /yogchaudhary/data
mkdir: `/yogchaudhary/data': No such file or directory
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary/data
ls: `/user/yogchaudhary/data': No such file or directory
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```

- Now, we fill the "data" subfolder to create a bucket in GCP. This subfolder will associate with that bucket. We loaded two datasets into the GCP Cloud storage Buckets: (userdata, weblog).





Step 5: We copy the data files from the bucket into our newly created subfolder in hdfs. This will allow us to run MapReduce, Spark, or Hive



- We create a subfolder for each dataset (userdata and weblog) in the folder "data."
- First create "userdata"

→ **hdfs dfs -mkdir /user/yogchaudhary/data**

- Let us see that subfolder “userdata” was created.

→ **hdfs dfs -ls /user/yogchaudhary**

- We see there is 1 item found in the data folder.

```
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -mkdir /user/yogchaudhary/data
mkdir: `/user/yogchaudhary/data': File exists
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary/data/userdata
ls: `/user/yogchaudhary/data/userdata': No such file or directory
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary/data
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary
Found 1 items
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-19 20:33 /user/yogchaudhary/data
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```

- Now, we create a subfolder named “weblog.”

→ **hdfs dfs -mkdir / user/yogchaudhary/data/userdata**

- There is nothing in this subfolder, but let us check to be sure
- Let us see that subfolder “weblog” was created.

→ **hdfs dfs -ls /user/yogchaudhary/data/userdata**

- We see there are 2 items now found in the data folder

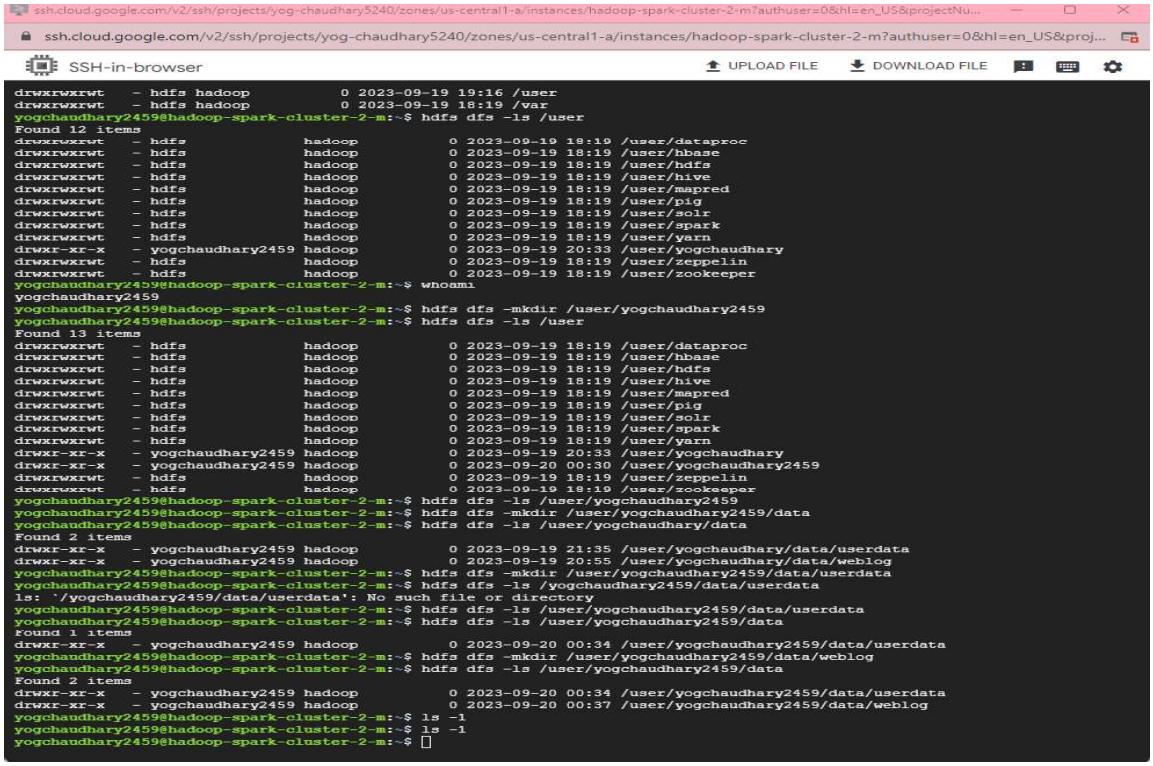
→ **hdfs dfs -ls /user/yogchaudhary/data**

ssh.cloud.google.com/v2/ssh/projects/yog-chaudhary5240/zones/us-central1-a/instances/hadoop-spark-cluster-2-m?authuser=0&hl=en_...

ssh.cloud.google.com/v2/ssh/projects/yog-chaudhary5240/zones/us-central1-a/instances/hadoop-spark-cluster-2-m?authuser=0&hl...

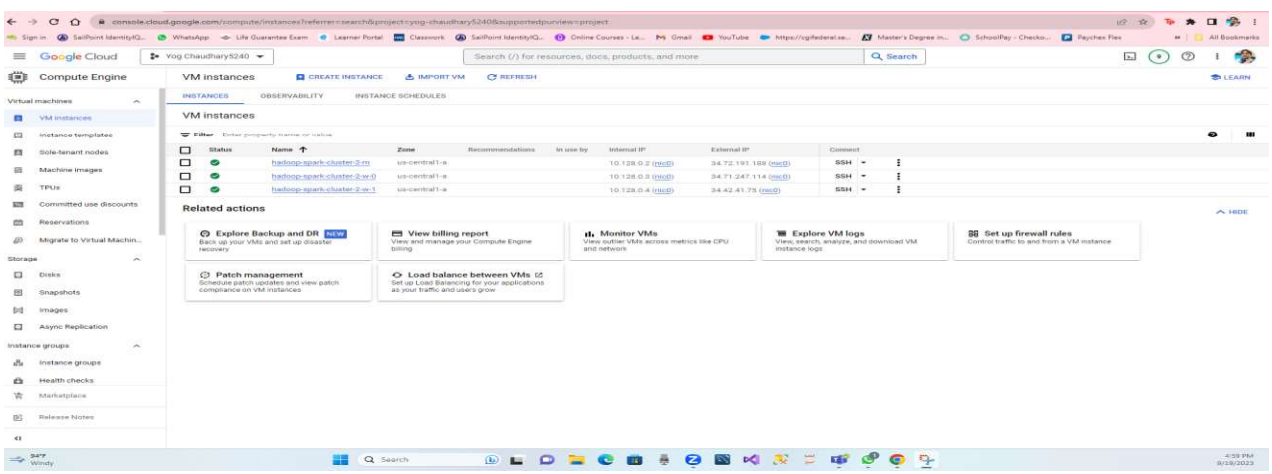
SSH-in-browser UPLOAD FILE DOWNLOAD FILE

```
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary
Found 1 items
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-19 21:35 /user/yogchaudhary/data
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -mkdir / user/yogchaudhary/data/userdata
mkdir: `/': File exists
mkdir: `user/yogchaudhary/data/userdata': No such file or directory
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary/data/userdata
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ hdfs dfs -ls /user/yogchaudhary/data
Found 2 items
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-19 21:35 /user/yogchaudhary/data/userdata
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-19 20:55 /user/yogchaudhary/data/weblog
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```



Step 6: Now we must open another new SSH terminal. Then I have access to the master node another SSH terminal and types "Compute Engine "in the search bar – OR- use the Navigation pane and scroll down to "Compute Engine ".

- Then click on "SSH" and click on "open in browser"



Step 7 In this new SSH terminal, we will see what is in the directory.

→ **Ls –1**
→ **We also see there is "data"**
→ **Now we will move to the data for this use cd DATA**

```
ssh.cloud.google.com/v2/ssh/projects/yog-chaudhary5240/zones/us-central1-a/instances/hadoop-spark-cluster-2-m?authuser=0&hl=en_...
ssh.cloud.google.com/v2/ssh/projects/yog-chaudhary5240/zones/us-central1-a/instances/hadoop-spark-cluster-2-m?authuser=0&hl...
SSH-in-browser                                   UPLOAD FILE   DOWNLOAD FILE

Linux hadoop-spark-cluster-2-m 5.10.0-0.deb10.16-cloud-amd64 #1 SMP Debian 5.10.127-2~bpo10+1 (2022-07-28) x86_
64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Sep 20 01:27:10 2023 from 35.235.244.34
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ ls -1
DATA
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ cs DATA/
-bash: cs: command not found
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ ls -1
DATA
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ gsutil cp gs://adta5240yog/userdata.csv userdata.csv
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023. Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Copying gs://adta5240yog/userdata.csv...
/ [1 files][162.3 KiB/162.3 KiB]
Operation completed over 1 objects/162.3 KiB.
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```

→ Copying two files from buckets to hdfs:

Step 8 Now I am copying the data files (userdata and weblog) from the GCP bucket to hdfs.

- For this we will use these commands
- Userdata:
- gsutil cp gs://adta5240yog/userdata.csv userdata.csv

```
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ ls -1
DATA
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ gsutil cp gs://adta5240yog/userdata.csv userdata.csv
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023. Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Copying gs://adta5240yog/userdata.csv...
/ [1 files][162.3 KiB/162.3 KiB]
Operation completed over 1 objects/162.3 KiB.
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ ls -1
DATA
userdata.csv
```

- The file “userdata” has been copied from bucket to master node. The file is 162.3 KIB.
- Now I am going to do the same thing for “weblog.”
- Weblog
- gsutil cp gs://adta5240yog/weblog.csv weblog.csv
- Below shows that this was also successfully copied and now we have two data files in the “DATA” folder

```
userdata.csv
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ gsutil cp gs://adta5240yog/weblog.csv weblog.csv
WARNING:  Python 3.5-3.7 will be deprecated on August 8th, 2023. Please use Python version 3.8 and up.

If you have a compatible Python interpreter installed, you can use it by setting
the CLOUDSDK_PYTHON environment variable to point to it.

Copying gs://adta5240yog/weblog.csv...
/ [1 files][  5.0 MiB/  5.0 MiB]
Operation completed over 1 objects/5.0 MiB.
yogchaudhary2459@hadoop-spark-cluster-2-m:~$ ls -l
DATA
userdata.csv
weblog.csv
yogchaudhary2459@hadoop-spark-cluster-2-m:~$
```

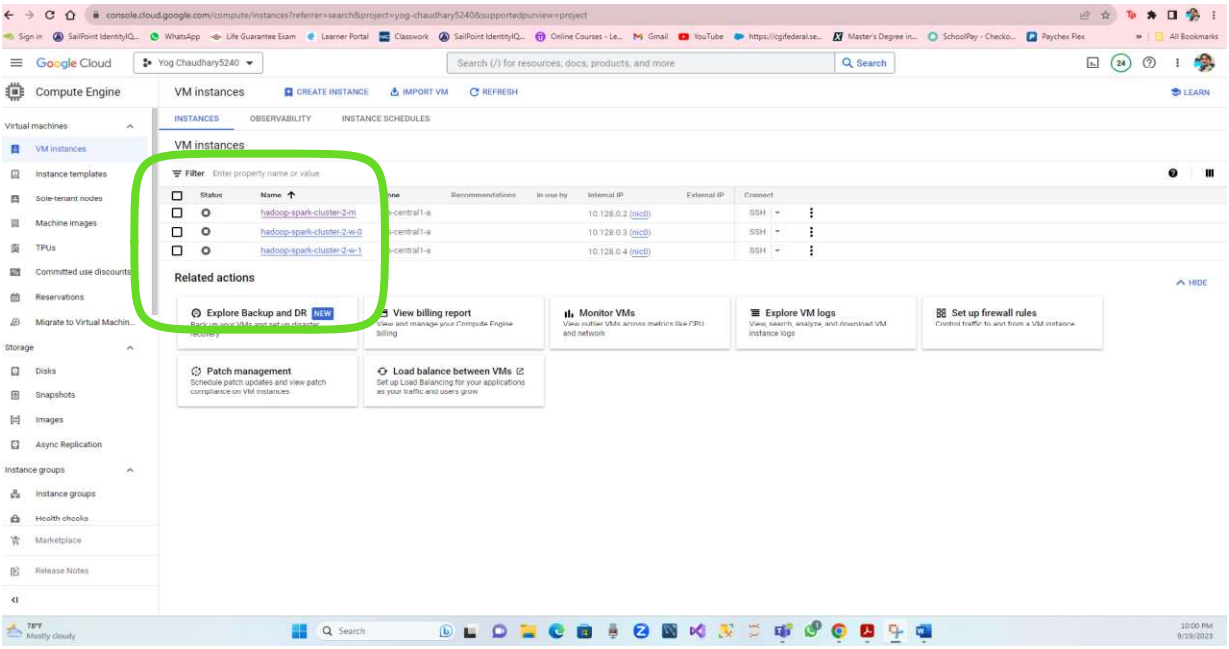
Step 9 Now I will take the file from the master node to the hdfs ecosystem.

- By using these commands, we can check the file "userdata" and "weblog" was copied to hdfs.
- hdfs dfs –ls /user/yogchaudhary2459/data
- hdfs dfs –put userdata.csv /user/yogchaudhary2459/data/userdata
- hdfs dfs –put weblog.csv /user/yogchaudhary2459/data/weblog
- hdfs dfs –ls /user/yogchaudhary2459/data/userdata
- hdfs dfs –ls /user/yogchaudhary2459/weblog

```
ssh.cloud.google.com/v2/ssh/projects/yog-chaudhary5240/zones/us-central1-a/instances/hadoop-spark-cluster-2-m?authuser=0&hl=en_...
ssh.cloud.google.com/v2/ssh/projects/yog-chaudhary5240/zones/us-central1-a/instances/hadoop-spark-cluster-2-m?authuser=0&hl...
SSH-in-browser                                   UPLOAD FILE   DOWNLOAD FILE
yogchaudhary2459@hadoop-spark-cluster-2-m:~/DATA$ hdfs dfs -put userdata.csv /user/yogchaudhary2459/data/userda
ta
yogchaudhary2459@hadoop-spark-cluster-2-m:~/DATA$ hdfs dfs -ls /user/yogchaudhary2459/data
Found 2 items
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-20 02:29 /user/yogchaudhary2459/data/userdata
drwxr-xr-x   - yogchaudhary2459 hadoop          0 2023-09-20 00:37 /user/yogchaudhary2459/data/weblog
yogchaudhary2459@hadoop-spark-cluster-2-m:~/DATA$ hdfs dfs -put weblog.csv /user/yogchaudhary2459/data/weblog
put: `weblog.csv': No such file or directory
yogchaudhary2459@hadoop-spark-cluster-2-m:~/DATA$
```

Hence, CSV files were successfully copied.

Step 10. Now finally all 3 virtual machine instances were stopped as shown below in GCP by selecting stop from three dots present at the top right to the SSH of each node.


console.cloud.google.com/compute/instances?referrer=search&project=yog-chaudhary5240&supportedpurview=project
Google Cloud
Yog Chaudhary5240
Search (/) for resources, docs, products, and more
Search
Compute Engine
VM instances
CREATE INSTANCE
IMPORT VM
REFRESH
LEARN
INSTANCES
OBSERVABILITY
INSTANCE SCHEDULES
Virtual machines
VM instances
Instance templates
Sole-tenant nodes
Machine images
TPUs
Committed use discounts
Reservations
Migrate to Virtual Machin...
Storage
Disks
Snapshots
Images
Async Replication
Instance groups
Instance groups
Health checks
Marketplace
Release Notes
VM instances
Filter
Enter property name or value
Status
Name
Recommendations
In use by
Internal IP
External IP
Connect
hadoop-spark-cluster-2-m
10.128.0.2 (nic0)
hadoop-spark-cluster-2-w-0
10.128.0.3 (nic0)
hadoop-spark-cluster-2-w-1
10.128.0.4 (nic0)
SSH
Related actions
HIDE
Explore Backup and DR
NEW
View billing report
View and manage your Compute Engine billing
Monitor VMs
View outlier VMs across metrics like CPU and network
Explore VM logs
View, search, analyze, and download VM instance logs
Set up firewall rules
Control traffic to and from a VM instance
Patch management
Schedule patch updates and view patch compliance on VM instances
Load balance between VMs
Set up Load Balancing for your applications as your traffic and users grow