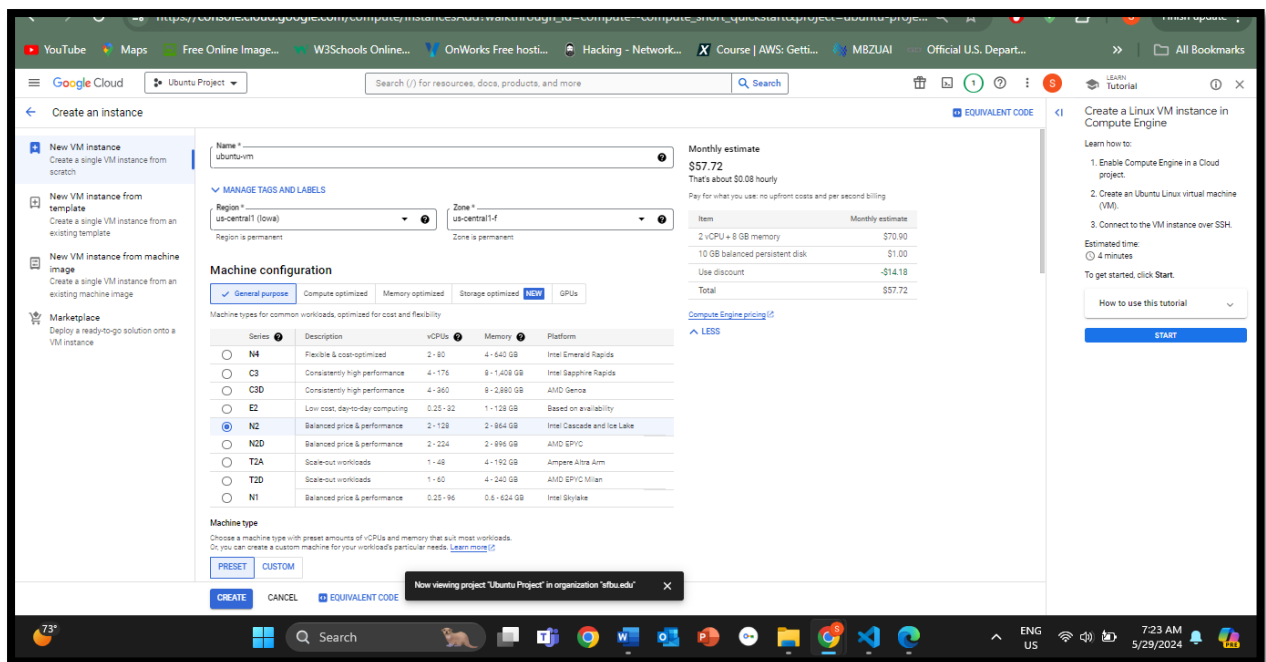


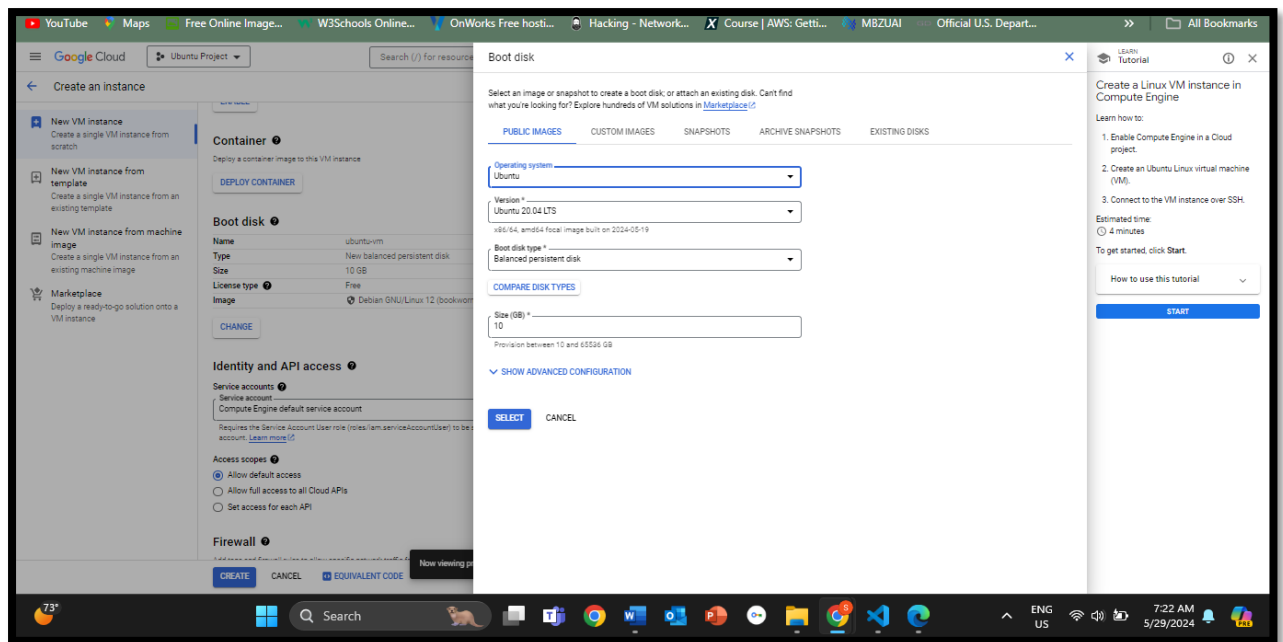
HOMEWORK

1. Provision a VM instance with Ubuntu OS

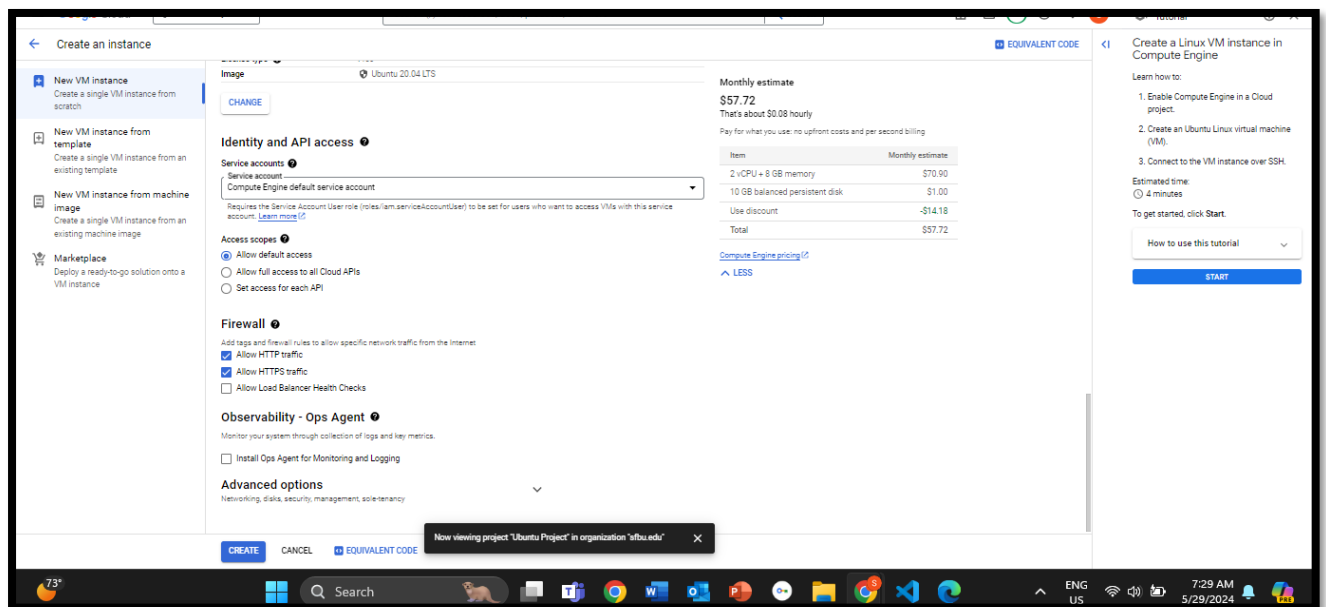
In the Google Cloud Platform, go to the Dashboard, select your project from the navigation menu, then navigate to Compute Engine > Virtual Instances. Click on "CREATE INSTANCE" to begin setting up a new virtual machine instance.



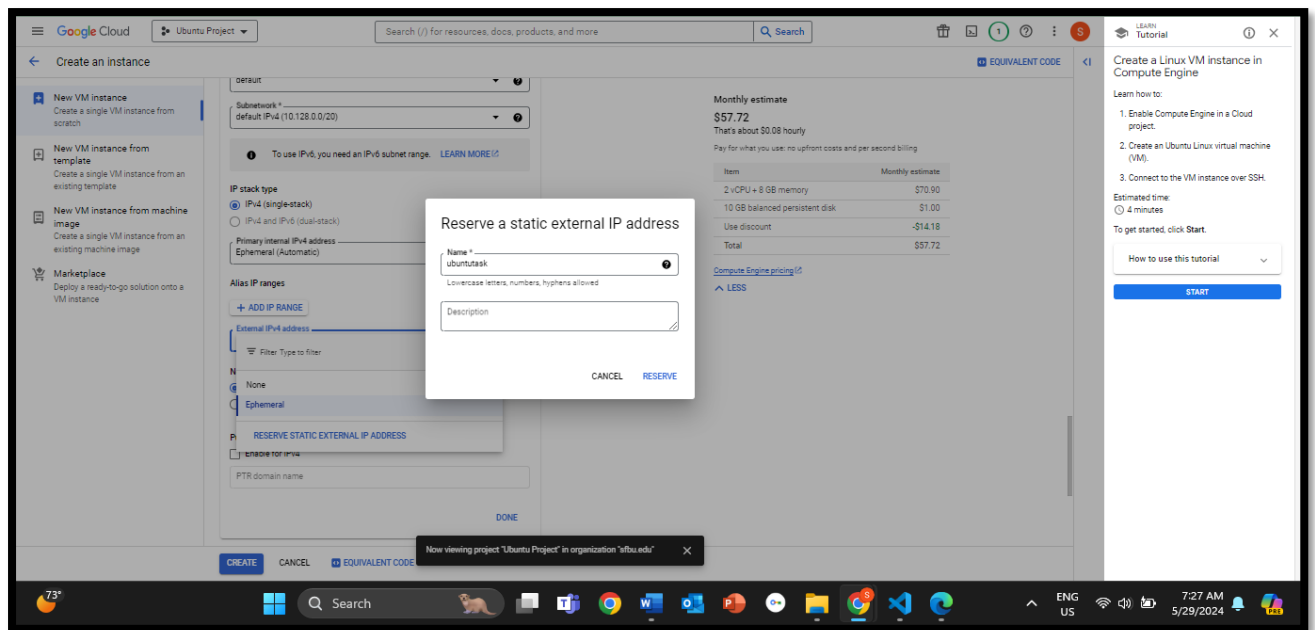
- ✓ In the Compute Engine > Virtual Instances section, click on the virtual machine instance you want to modify.
- ✓ In the instance details page, scroll down to the "Boot disk" section and locate the boot disk you want to change.
- ✓ Click the "Change" button next to the boot disk. In the "Boot disk" dialog that appears, select "Custom images" from the "OS images" dropdown menu. Choose "Ubuntu" from the list of available custom images.



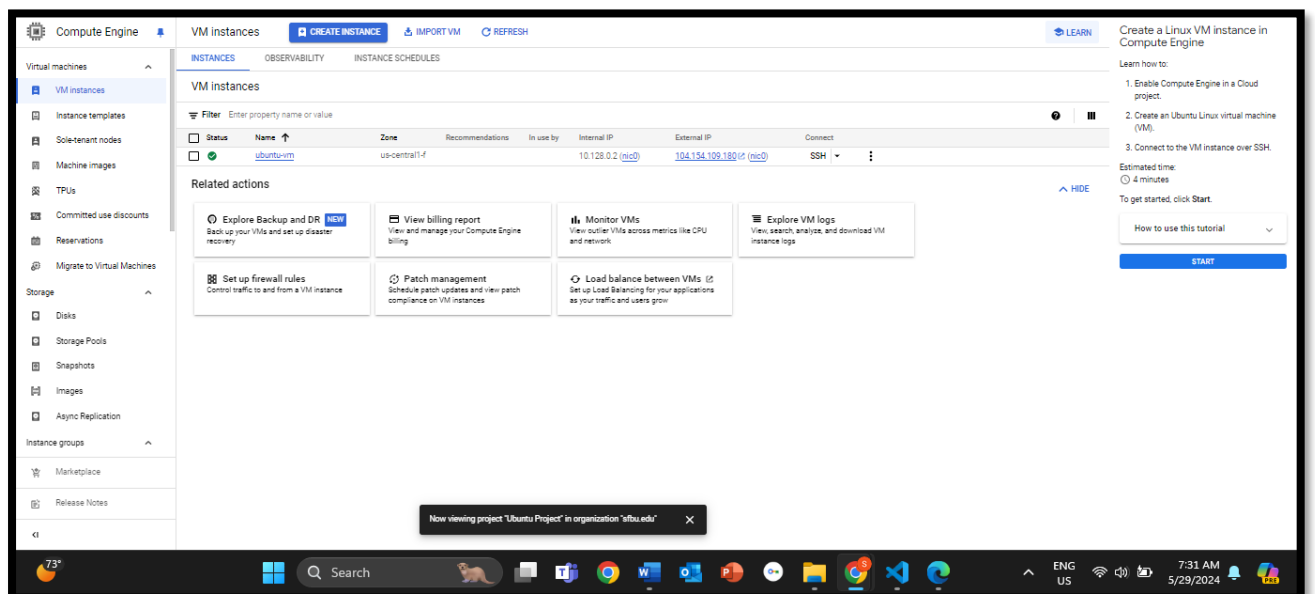
- ✓ Allow the the HTTP/HTTPS in the firewall category.

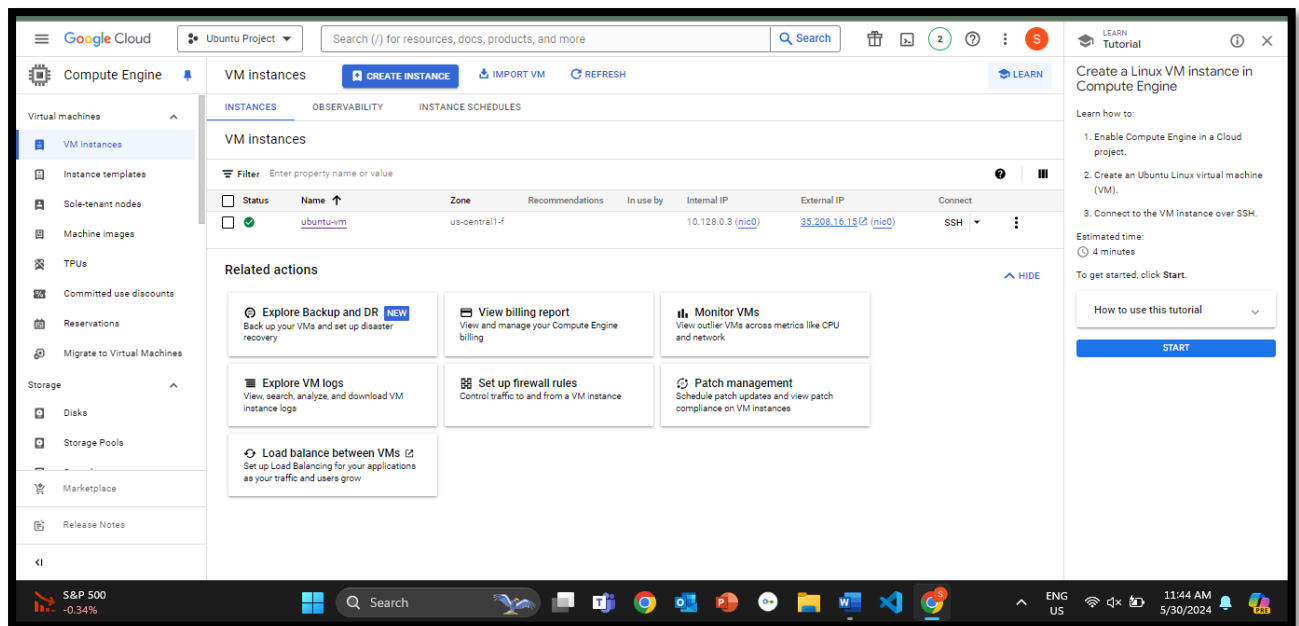


- ✓ We also need to create a static IP Address
For that we need to go to the Advanced Option -> Networking -> Network Interfaces and
Select Default , Add External IP address



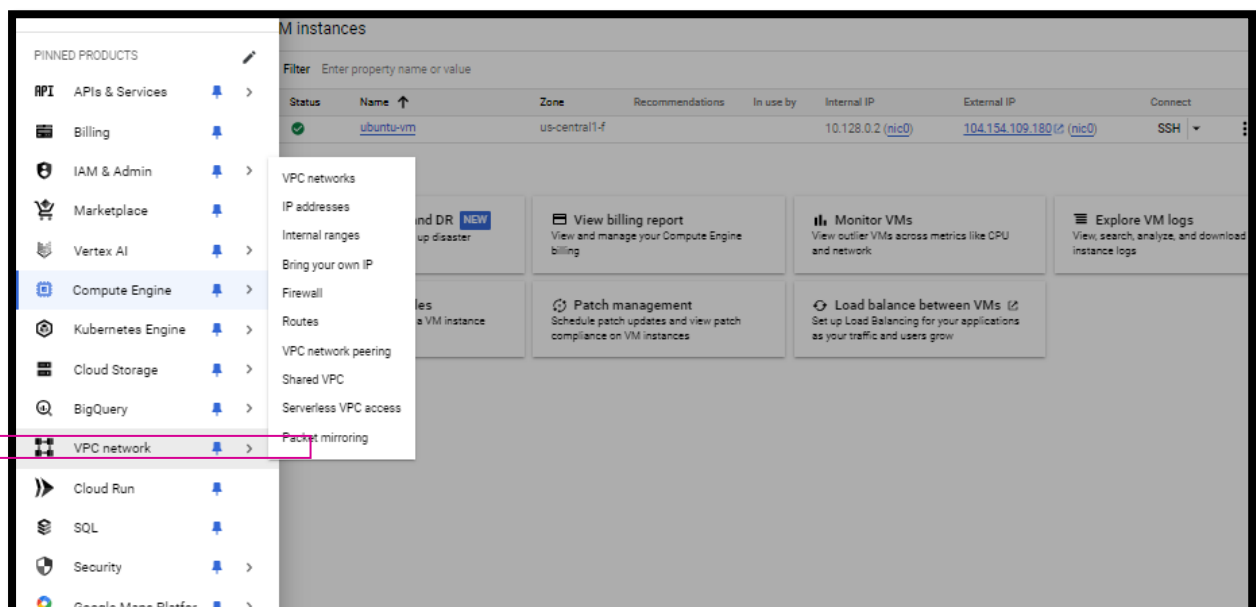
- ✓ After you complete the setup, click on the “CREATE” Button at the bottom. And your VM will be in the dashboard under the Compute Engine->VM instances.



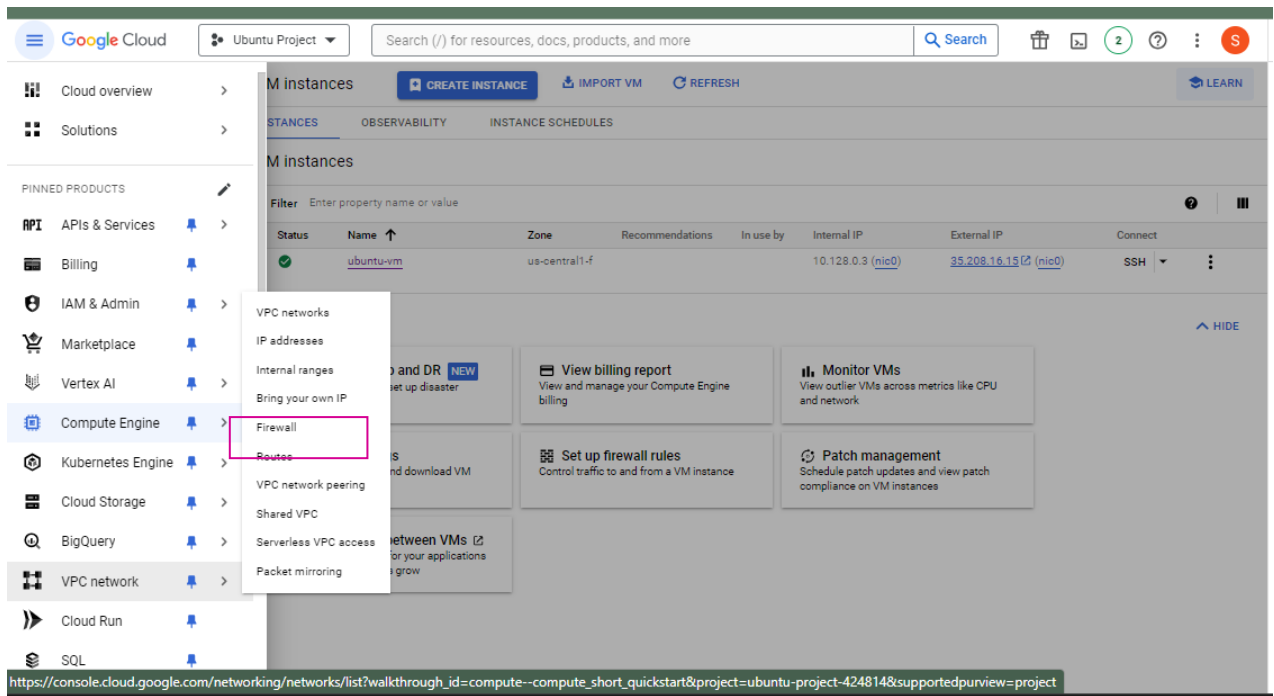


- ✓ To ensure access to Hadoop HDFS, NameNode, and ResourceManager information via a browser later, follow these steps to allow all protocols and ports for the HTTP server. If not needed, you can skip this step.

1. Click on the Navigation Menu icon and select "VPC network" from the options.

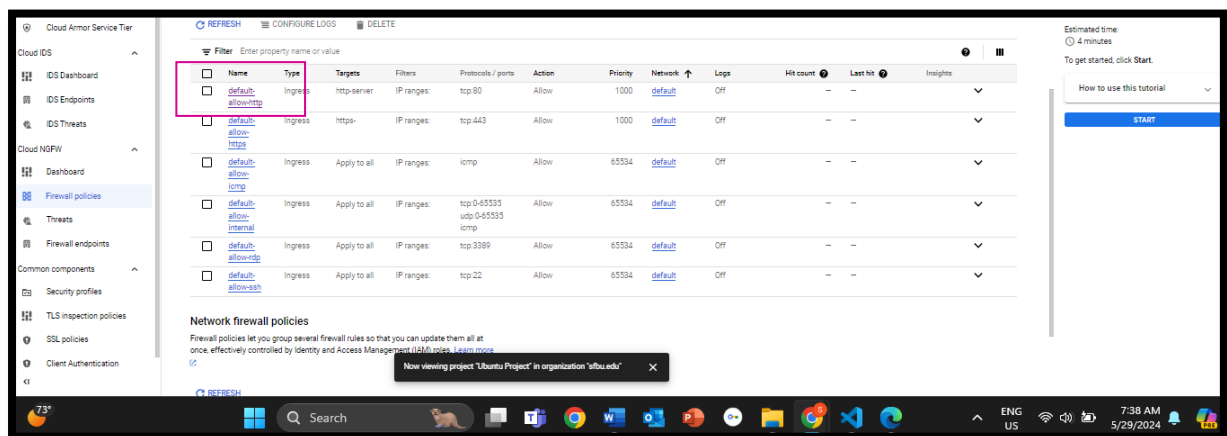


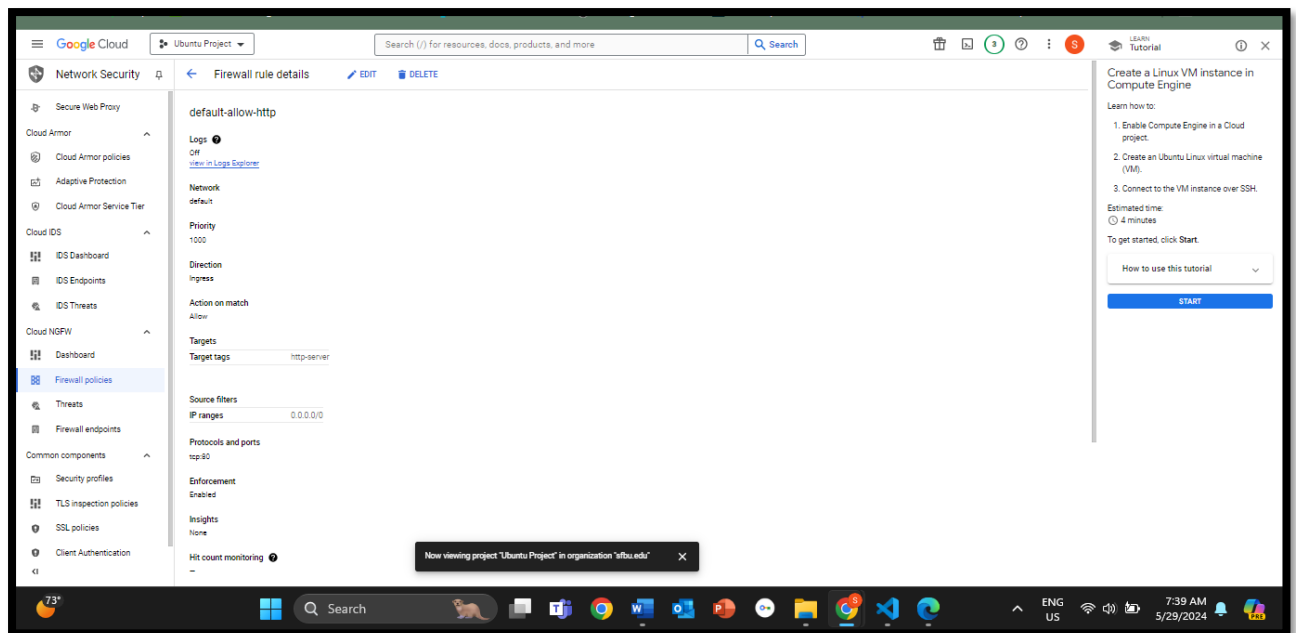
2. Navigate to "Firewall" settings in the VPC network section.



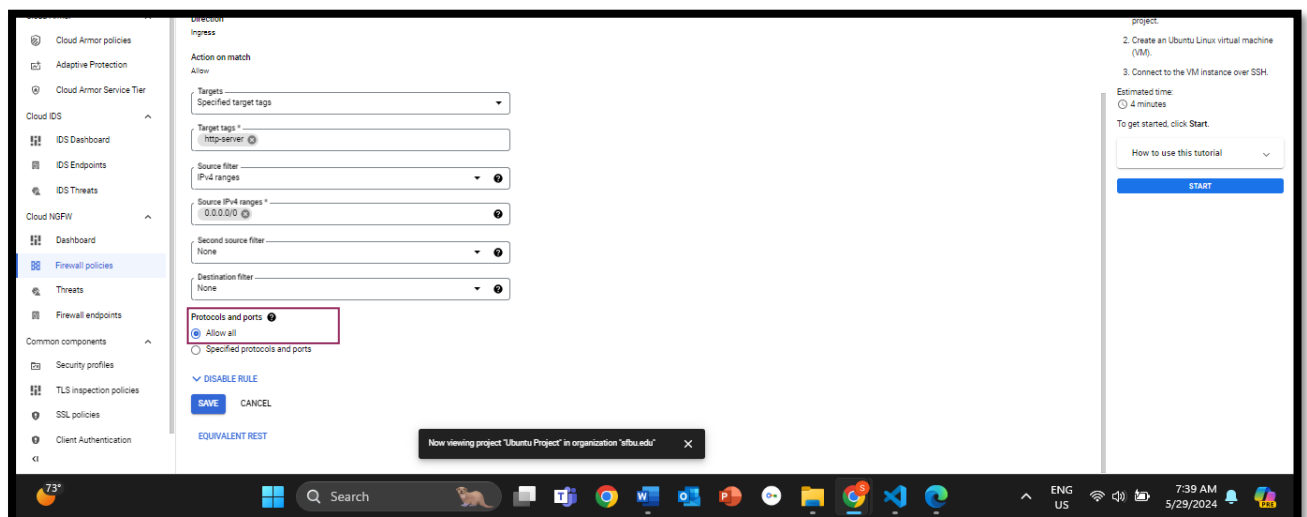
3. Locate the firewall rules applicable to your network configuration.
4. Edit the firewall rule associated with HTTP traffic, typically named "http-server" or similar.
5. Update the firewall rule to allow all protocols and ports for inbound HTTP traffic.

Click on “Edit” from the top



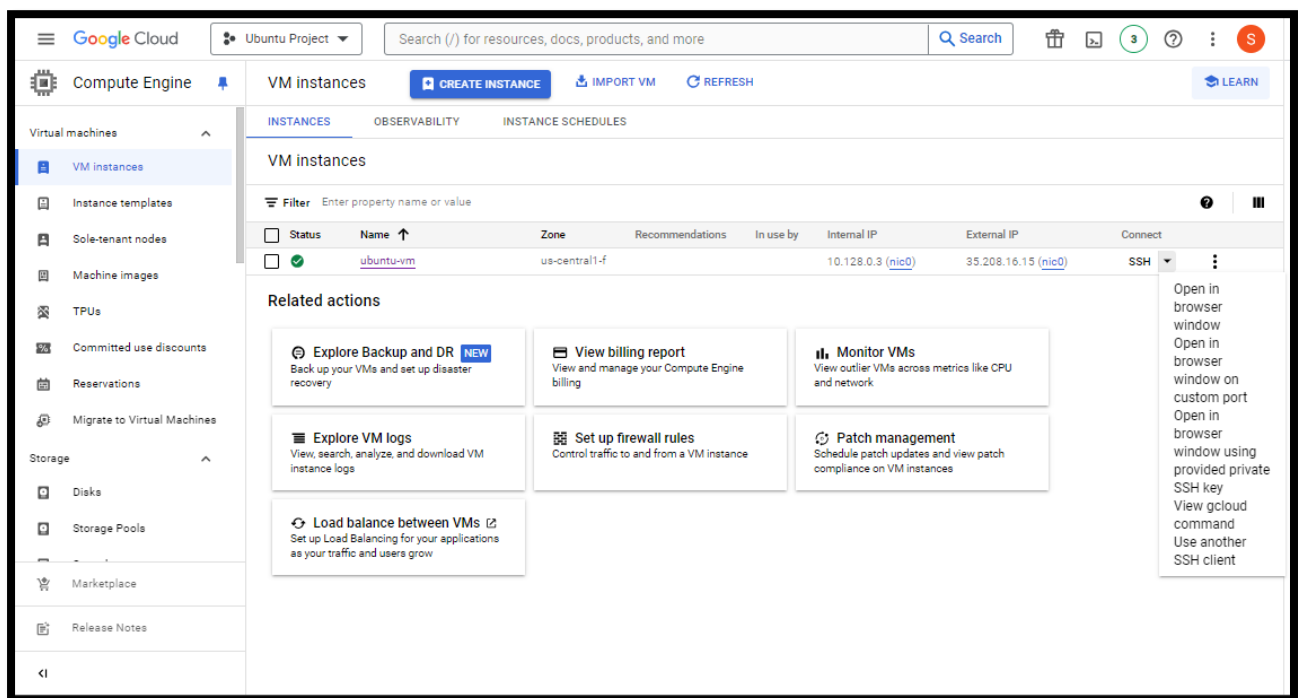
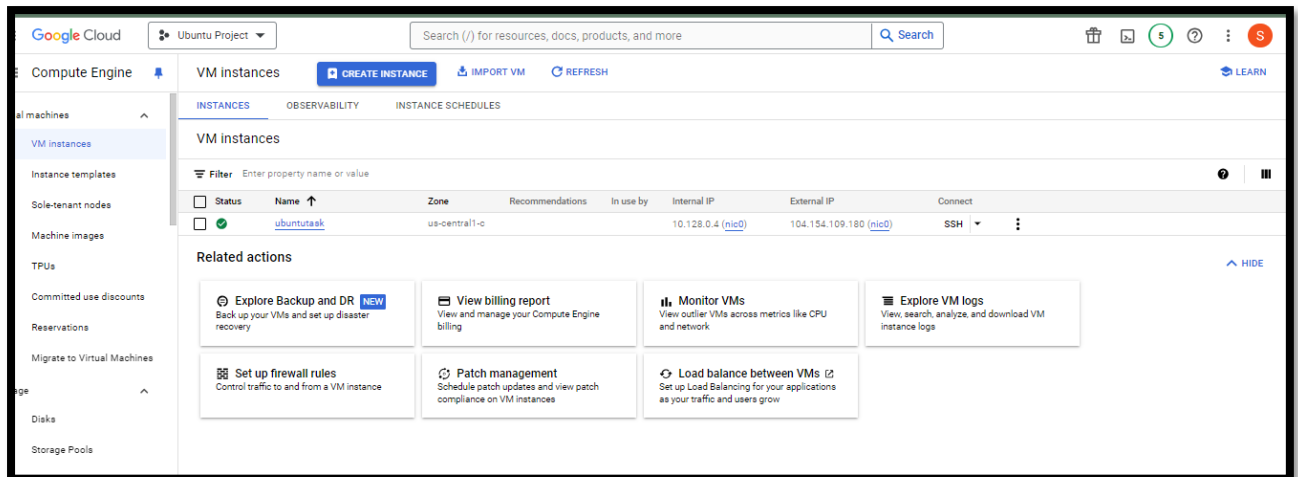


And make all Protocol and Ports “Allow all”.

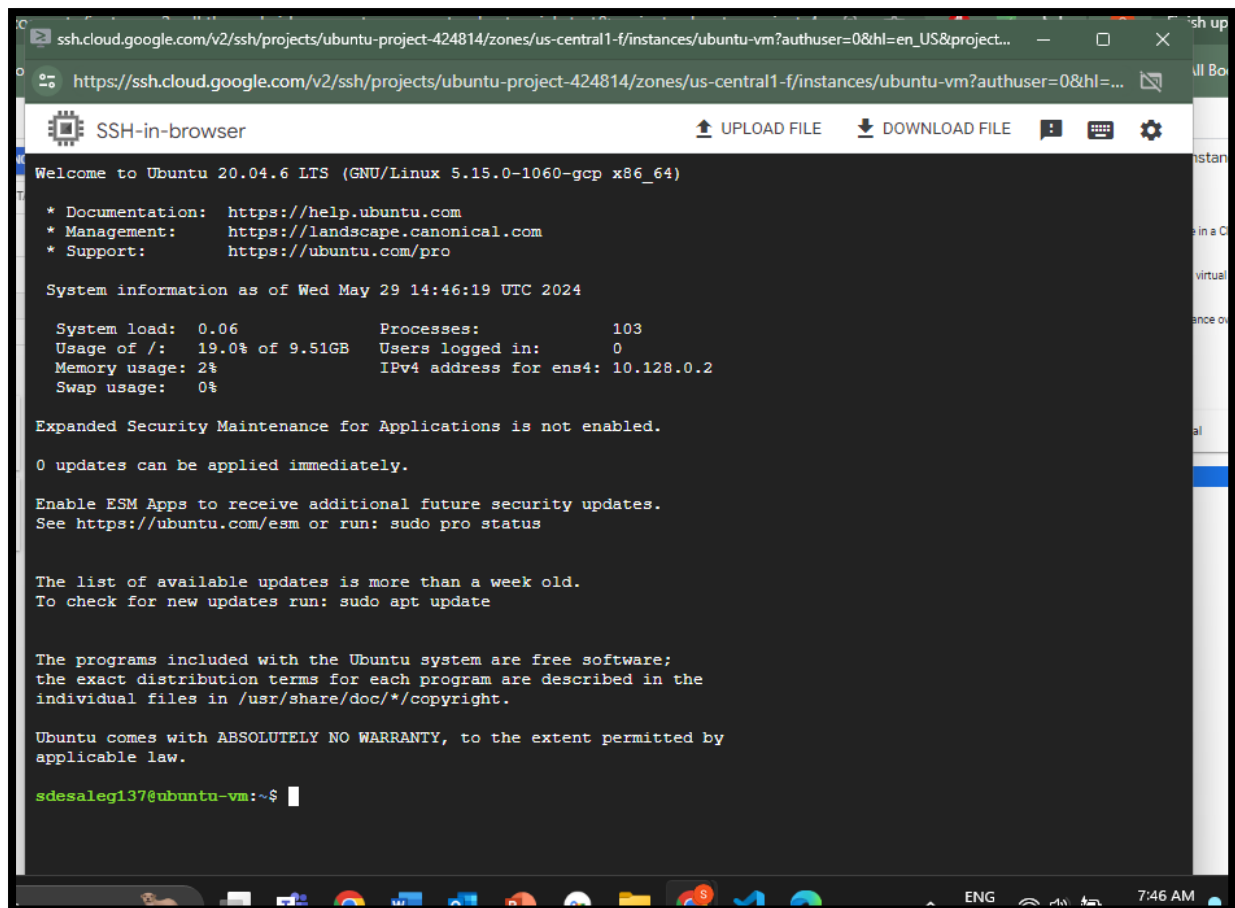


6. Save the changes to the firewall rule.

- ✓ Login in ubuntu VM to install Hadoop including requisite software step by step.
- Click the arrow near the SSH, to open in browser window.



✓ Then a new page will be opened:



The screenshot shows a web browser window titled "SSH-in-browser" with a URL from google.com. The terminal displays the Ubuntu 20.04.6 LTS login screen. It shows system information as of Wednesday, May 29, 2024, at 14:46:19 UTC. The system load is 0.06, with 103 processes and 19.0% disk usage. Memory usage is 2% and swap usage is 0%. It also shows that no updates can be applied immediately and that the list of available updates is more than a week old. The terminal ends with the prompt `sdesaleg137@ubuntu-vm:~$`.

```
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1060-gcp x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Wed May 29 14:46:19 UTC 2024

System load:  0.06          Processes:      103
Usage of /:   19.0% of 9.51GB Users logged in: 0
Memory usage: 2%           IPv4 address for ens4: 10.128.0.2
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

sdesaleg137@ubuntu-vm:~$
```

2. Hadoop: Setting up a Single Node Cluster.

- ✓ Java is required for Hadoop because the Hadoop framework, including its core processing engine MapReduce, is implemented in Java.
- ✓ Follow the commands below to install Java.
 - `$ sudo apt-get update`
 - `$ sudo apt-get install openjdk-8-jdk`
 - `$ java -version` (This command will display the java version info if installation is successful)
 - `$ sudo update-alternatives --config java`
 - (To know the java path, write down this path for later path setting)

```
Reading package lists... Done
sdesaleg137@ubuntu-vm:~$ sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

```
Error: A fatal exception has occurred. Program will exit.
sdesaleg137@ubuntu-vm:~$ java -version
openjdk version "1.8.0_402"
OpenJDK Runtime Environment (build 1.8.0_402-8u402-ga-2ubuntu1~20.04-b06)
OpenJDK 64-Bit Server VM (build 25.402-b06, mixed mode)
sdesaleg137@ubuntu-vm:~$ sudo update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin/java): /usr/lib/jvm/java-8-openjdk-amd64/j
re/bin/java
Nothing to configure.
sdesaleg137@ubuntu-vm:~$
```


- ✓ Check if ssh/sshd/pdsh exists already, If not, install them:

```
$ which ssh
$ which sshd
$ which pdsh
```

- ✓ For Installing ssh & pdsh follow these commands:

```
$ sudo apt-get install ssh
$ sudo apt-get install pdsh
```

```
sdesaleg137@ubuntu-vm:~$ which ssh
/usr/bin/ssh
sdesaleg137@ubuntu-vm:~$ which sshd
/usr/sbin/sshd
sdesaleg137@ubuntu-vm:~$ which pdsh
sdesaleg137@ubuntu-vm:~$ sudo apt-get install pdsh
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

- ✓ After installation to confirm:

```
sdesaleg137@ubuntu-vm:~$ which pdsh
/usr/bin/pdsh
sdesaleg137@ubuntu-vm:~$
```

- ✓ To download and unpack Hadoop 3.4.0, follow these steps:

1. Download Hadoop 3.4.0:

- Visit the official Apache Hadoop website or use a mirror site to download the Hadoop 3.4.0 distribution package.
- You can use a web browser or a command-line tool like wget to download the package.
- \$ wget <https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz>

2. Unpack the Hadoop Distribution:

- Navigate to the directory where the downloaded package is located.
- \$ tar xvzf hadoop-3.3.4.tar.gz

```
sdesaleg137@ubuntu-vm:~$ wget https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
--2024-05-29 15:04:20-- https://dlcdn.apache.org/hadoop/common/hadoop-3.4.0/hadoop-3.4.0.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 965537117 (921M) [application/x-gzip]
Saving to: 'hadoop-3.4.0.tar.gz'

hadoop-3.4.0.tar.gz      100%[=====>] 920.81M   263MB/s   in 3.6s

2024-05-29 15:04:52 (259 MB/s) - 'hadoop-3.4.0.tar.gz' saved [965537117/965537117]

sdesaleg137@ubuntu-vm:~$ tar xvzf hadoop-3.4.0.tar.gz
hadoop-3.4.0/
```

- ✓ Set java path for Hadoop:

```
$ cd hadoop-3.4.0/
```

- ✓ Then open the file:

```
$ vi etc/hadoop/hadoop-env.sh
```

- After, you open the file, you can click “i” to insert and for saving use “:wq”
Add “ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

```
World 2
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ vi /etc/hadoop/hadoop-env.sh
sdesaleg137@ubuntutask:~/hadoop-3.4.0$
```

```
###
# Generic settings for HADOOP
###

# Technically, the only required environment variable is JAVA_HOME.
# All others are optional. However, the defaults are probably not
# preferred. Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d

# The java implementation to use. By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
# export JAVA_HOME=

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

# The language environment in which Hadoop runs. Use the English
# environment to ensure that logs are printed as expected.
export LANG=en_US.UTF-8

"hadoop-env.sh" 437L, 16839C
```

- ✓ Set other path/variables for Hadoop

```
$ vi ~/.bashrc
```

Add the following lines:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
```

```
export PDSH_RCMD_TYPE=ssh
```

```
export PATH=${JAVA_HOME}/bin:${PATH}
```

```
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

```
sdesaleg137@ubuntu-vm:~$ cd hadoop-3.4.0/
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ vi ~/.bashrc
```

```
fi
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
export PDSH_RCMD_TYPE=ssh
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
~
~
~
```

- ✓ Source the .bashrc file to apply the changes to your current shell session:

```
$source ~/.bashrc
```

```
export PATH=$PATH:$HADOOP_HOME/bin
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ vi ~/.bashrc
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ source ~/.bashrc
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$
```

- ✓ Test the Hadoop command:

```
$ bin/Hadoop
```

This will display the usage documentation for the Hadoop script if Hadoop is installed successfully

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hadoop
Usage: hadoop [OPTIONS] SUBCOMMAND [SUBCOMMAND OPTIONS]
or hadoop [OPTIONS] CLASSNAME [CLASSNAME OPTIONS]
where CLASSNAME is a user-provided Java class

OPTIONS is none or any of:

buildpaths          attempt to add class files from build tree
--config dir        Hadoop config directory
--debug            turn on shell script debug mode
--help            usage information
hostnames list[,of,host,names] hosts to use in worker mode
hosts filename      list of hosts to use in worker mode
loglevel level      set the log4j level for this command
workers            turn on worker mode

SUBCOMMAND is one of:

Admin Commands:

daemonlog          get/set the log level for each daemon

Client Commands:
```

Standalone Operations:

- ✓ By default, Hadoop is configured to run in a non-distributed mode, as a single Java process. This is useful for debugging. The following example copies the unpacked conf directory to use as input and then finds and displays every match of the given regular expression. Output is written to the given output directory.

```
$ mkdir input
```

```
$ cp etc/hadoop/*.xml input
```

```
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar grep input
output 'dfs[a-z.]+'
```

```
$ cat output/*
```

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ ls
LICENSE-binary  NOTICE-binary  README.txt  etc  lib  licenses-1
LICENSE.txt     NOTICE.txt    bin         include  libexec  sbin
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ mkdir input
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ cp etc/hadoop/*.xml input
```

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ ls ./input
capacity-scheduler.xml  hadoop-policy.xml  hdfs-site.xml  kms-acls.xml  mapred-site.xml
core-site.xml           hdfs-rbf-site.xml  https-site.xml  kms-site.xml  yarn-site.xml
```

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar
grep input output 'dfs[a-z.]+'
2024-05-29 17:28:03,488 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-05-29 17:28:03,576 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-05-29 17:28:03,576 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-05-29 17:28:03,788 INFO input.FileInputFormat: Total input files to process : 10
2024-05-29 17:28:03,818 INFO mapreduce.JobSubmitter: number of splits:10
2024-05-29 17:28:03,977 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local13506703_0001
2024-05-29 17:28:03,977 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-05-29 17:28:04,149 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2024-05-29 17:28:04,150 INFO mapreduce.Job: Running job: job_local13506703_0001
2024-05-29 17:28:04,156 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2024-05-29 17:28:04,162 INFO output.PathOutputCommitterFactory: No output committer factory defined, defaulting
to FileOutputCommitterFactory
```

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ cat output/*
1      dfsadmin
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$
```

Pseudo-Distributed Operation

- ✓ Hadoop can also be run on a single-node in a pseudo-distributed mode where each Hadoop daemon runs in a separate Java process. Configure parameters:

\$ vi etc/hadoop/core-site.xml

\$ vi etc/hadoop/hdfs-site.xml

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ vi core-site.xml
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ vi hdfs-site.xml
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$
```



SSH-in-browser

UPLOAD FILE

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
~
~
```

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
~
~
~
```

- ✓ To set up SSH passphraseless authentication, follow these steps:


```
$ ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 600 ~/.ssh/authorized_keys
$ ssh localhost
```
- ✓ Following these steps will enable SSH passphraseless authentication, allowing you to connect to remote hosts securely and conveniently without entering a passphrase each time.

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ ssh-keygen -t rsa -P "" -f ~/.ssh/id_rsa
Generating public/private rsa key pair.
Your identification has been saved in /home/sdesaleg137/.ssh/id_rsa
Your public key has been saved in /home/sdesaleg137/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:eJKtpreOiPAJfQA6naWVZWlKId9rvCatuSWMhbG3f8k sdesaleg137@ubuntu-vm
The key's randomart image is:
+---[RSA 3072]-----+
|      o+.          |
|    .++ .         |
|  .+o . . .       |
|.o =. B . .       |
|o =  * S +        |
| o .  O + .       |
|o . . + = = . .   |
|. + + +. O E       |
|. + ooo.+...       |
+---[SHA256]-----+
```

If the following codes are not working for you, when you do ssh on the local host, try to make the following changes:

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ ssh localhost
sdesaleg137@localhost: Permission denied (publickey).
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ ^C
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ chmod 700 ~/.ssh
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ chmod 600 ~/.ssh/authorized_keys
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ chmod 600 ~/.ssh/id_rsa
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ chmod 644 ~/.ssh/id_rsa.pub
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ sudo systemctl restart ssh
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ ssh localhost
```

- ✓ If it still makes an issue :
- ✓ Verify SSH Configuration:
- ✓ Open the SSH configuration file and ensure the settings allow for public key authentication:

sudo nano /etc/ssh/sshd_config

Make sure the following lines are present and not commented out:

PubkeyAuthentication yes

AuthorizedKeysFile %h/.ssh/authorized_keys

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ ssh localhost
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1060-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed May 29 17:58:10 UTC 2024

System load:  0.0           Processes:    106
Usage of /:   53.7% of 9.51GB Users logged in:  1
Memory usage: 5%          IPv4 address for ens4: 10.128.0.2
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
4 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed May 29 17:35:44 2024 from 35.235.241.16
sdesaleg137@ubuntu-vm:~$
```

- ✓ The following instructions are to run a MapReduce job locally.

a. Format file System

\$ bin/hdfs namenode -format

```
sdesaleg137@ubuntu-vm:~$ cd hadoop-3.4.0
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hdfs namenode -format
WARNING: /home/sdesaleg137/hadoop-3.4.0/logs does not exist. Creating.
2024-05-29 18:13:07,135 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:  host = ubuntu-vm.us-centrall-f.c.ubuntu-project-424814.internal/10.128.0.2
STARTUP_MSG:  args = [-format]
STARTUP_MSG:  version = 3.4.0
STARTUP_MSG:  classpath = /home/sdesaleg137/hadoop-3.4.0/etc/hadoop:/home/sdesaleg137/hadoop-3.4.0/share/hadoop/common/lib/kerb-client-2.0.3.jar:/home/sdesaleg137/hadoop-3.4.0/share/hadoop/common/lib/curat
```

b. Start NameNode and DataNode daemon:

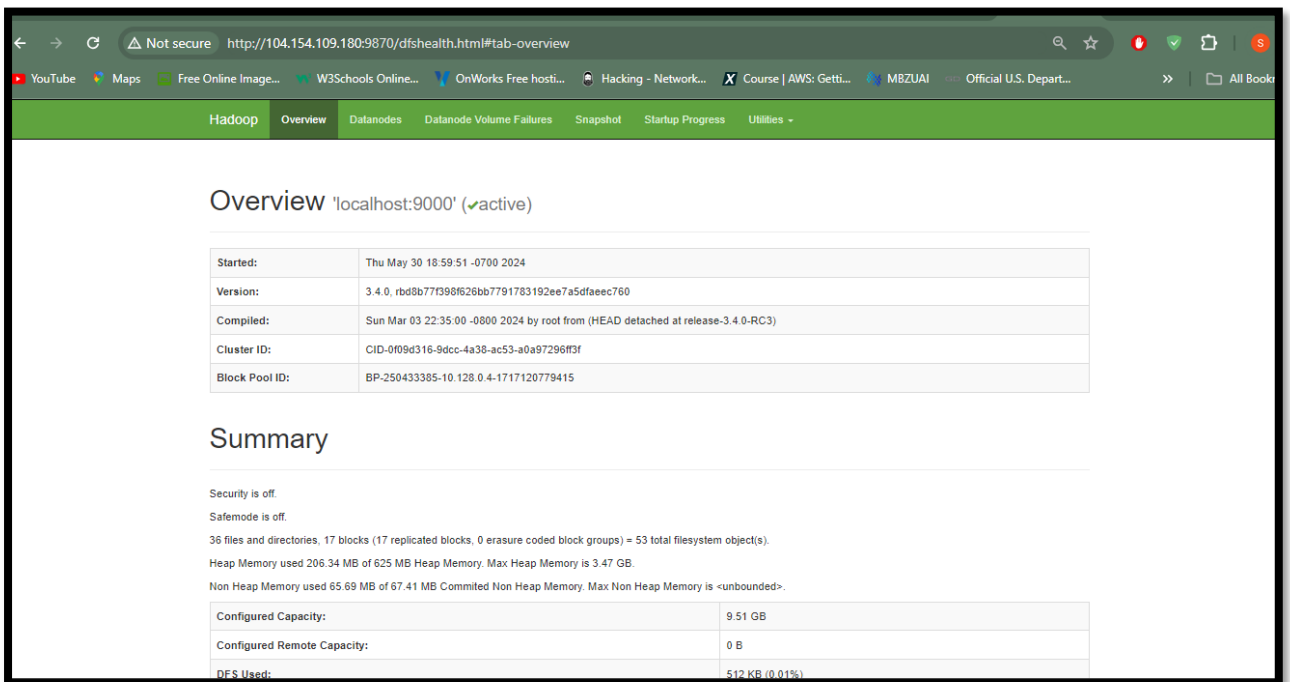
\$ sbin/start-dfs.sh


```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu-vm]
ubuntu-vm: Warning: Permanently added 'ubuntu-vm,10.128.0.2' (ECDSA) to the list of known hosts.
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$
```

- c. Browser the web interface for the NameNode; By default it's available at port 9870.

Check the NameNode through browser:

Note: We can NOT use this URL: <http://localhost:8088/>, we must replace localhost with the external IP address created in GCP VM instance. e.g. <http://104.154.109.180:9870/> in my case.



- d. Make the HDFS directories required to execute MapReduce jobs:

Note: We cannot create username randomly, we must use the username with which we login this Linux server, For example, my username is sdesaleg137.

```
$ bin/hdfs dfs -mkdir /user
```

```
$ bin/hdfs dfs -mkdir /user/
```

```
$ bin/hdfs dfs -mkdir input $ bin/hdfs dfs -put etc/hadoop/*.xml input
```

```
$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar grep
input output 'dfs[a-z.]+'
```

```
sdesaleg137@ubuntu-vm:~$ ls
hadoop-3.4.0  hadoop-3.4.0.tar.gz
sdesaleg137@ubuntu-vm:~$ cd hadoop-3.4.0
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user/sdesaleg137
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir input
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hdfs dfs -put etc/hadoop/*.xml input
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.4.0.jar
grep input output 'dfs[a-z.]+'
2024-05-29 22:35:37,247 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2024-05-29 22:35:37,376 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2024-05-29 22:35:37,376 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2024-05-29 22:35:37,750 INFO input.FileInputFormat: Total input files to process : 10
```

- ✓ Copy and display the output files

```
$ bin/hdfs dfs -get output output
```

```
$ cat output/*
```

```
Bytes Written=29
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hdfs dfs -get output output
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ cat output/*
cat: output/output: Is a directory
1
dfsadmin
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ ls -rtl output/output/*
-rw-r--r-- 1 sdesaleg137 sdesaleg137 0 May 29 22:37 output/output/_SUCCESS
-rw-r--r-- 1 sdesaleg137 sdesaleg137 29 May 29 22:37 output/output/part-r-00000
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$
```

- ✓ If you want to stop here, before you leave, you'd better stop the daemons, otherwise, skip this step.

```
$ sbin/stop-dfs.sh
```

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ sbin/stop-dfs.sh
Stopping namenodes on [localhost]
localhost: sdesaleg137@localhost: Permission denied (publickey).
pdsh@ubuntu-vm: localhost: ssh exited with exit code 255
Stopping datanodes
```

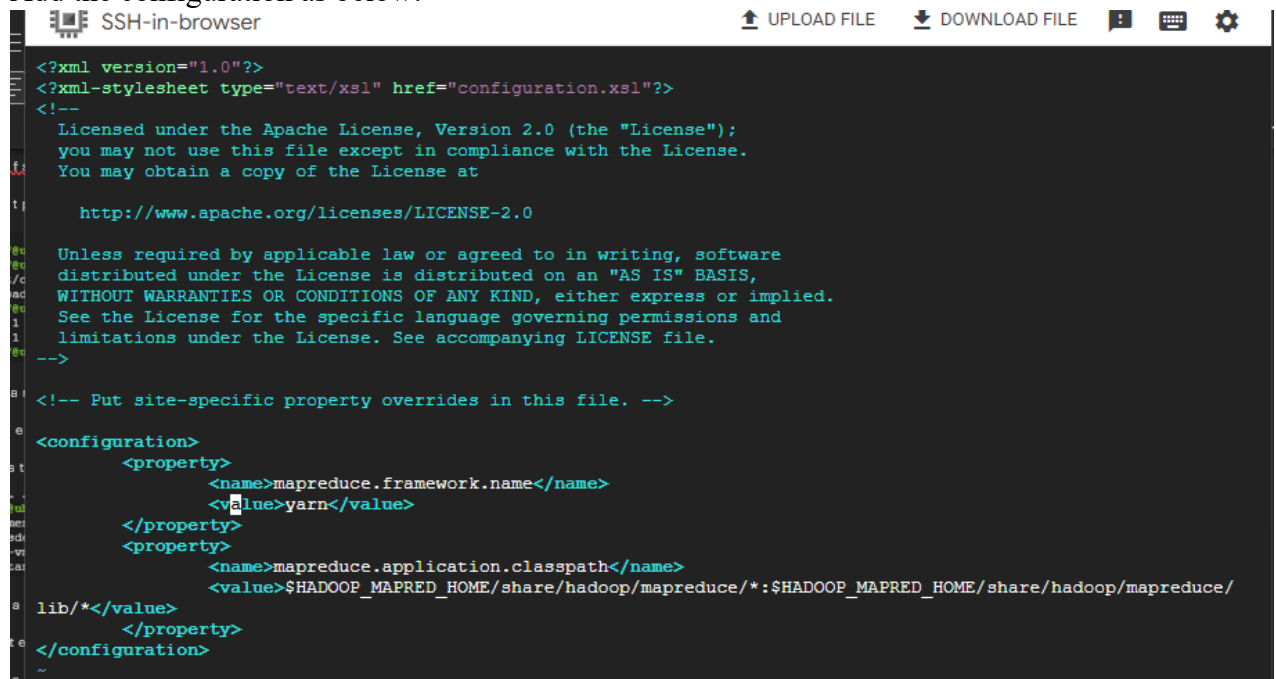
- ✓ To run YARN in pseudo-distributed mode and configure the necessary parameters, follow these steps:

- Configure parameters as shown

```
$ vi etc/hadoop/mapred-site.xml
```

```
pdsh@ubuntu-vm: ubuntu-vm: ssh exited with exit code 255
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ vi etc/hadoop/mapred-site.xml
```

- ✓ Add the configuration as below:



```
SSH-in-browser
[+] UPLOAD FILE [v] DOWNLOAD FILE [i] [k] [g]

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/
lib/*</value>
  </property>
</configuration>
```

Again, add or modify configuration for the yarn.xml.

```
$ vi yarn-site.xml
```

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$ vi yarn-site.xml
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc/hadoop$
```

Added the configuration :


```
<?xml version="1.0"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<configuration>

    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.nodemanager.env-whitelist</name>
        <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME</value>
    </property>
</configuration>
~
```

- b. Start ResourceManager daemon and NodeManager Daemon

\$ sbin/start-yarn.sh

```
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
localhost: sdesaleg137@localhost: Permission denied (publickey).
pdsh@ubuntu-vm: localhost: ssh exited with exit code 255
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$
```

The permission denied is for the local host.

- ✓ Browser the web interface – It's available at port 8088, URL like this: <http://localhost:8088/> Here, we use <http://104.154.109.180:8088/> for GCP Hadoop environment in my case.

The screenshot shows the Hadoop web interface at <http://104.154.109.180:8088/cluster>. The interface displays various cluster metrics and application status.

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes
0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Cluster Application Priority
Capacity Scheduler	[memory-mb (unit=M), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

All Applications

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB
No data available in table														

Showing 0 to 0 of 0 entries

3. Map Reduce Tutorial: Word Count

- ✓ WordCount Example (Reference link: [Apache Hadoop 3.3.4 – MapReduce Tutorial](#)) Create Java source code for WordCount. (Source codes are copied from the above link)
- \$ vi WordCount.java
- ✓ Then from the link above copy the following information to the WordCount.java.

```

import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context
            ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }

    public static class IntSumReducer
        extends Reducer<Text, IntWritable, Text, IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
            Context context
            ) throws IOException, InterruptedException {

-- INSERT --

```

- ✓ Compile java code and create a jar
\$ bin/hadoop com.sun.tools.javac.Main WordCount.java
\$ jar cf wc.jar WordCount*.class

```

sdesaleg137@ubuntu-vm:~/hadoop-3.4.0/etc$ cd ..
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ vi WordCount.java
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ export PATH=${JAVA_HOME}/bin:${PATH}
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ bin/hadoop com.sun.tools.javac.Main WordCount.java
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$ jar cf wc.jar WordCount*.class
sdesaleg137@ubuntu-vm:~/hadoop-3.4.0$

```

- ✓ Create hdfs directories, file01 and file02 . Follow the commands below to create the files.
\$ bin/hdfs dfs -mkdir wordcount
\$ bin/hdfs dfs -mkdir wordcount/input
\$ echo "Hello World Bye World" > wordcount/input/file01
\$ echo "Hello Hadoop Goodbye Hadoop" > wordcount/input/file02
\$ bin/hdfs dfs -mkdir -p /user/sdesaleg137/wordcount/input
\$ bin/hdfs dfs -put wordcount/input/* /user/sdesaleg137/wordcount/input/
\$ bin/hdfs dfs -ls /user/sdesaleg137/wordcount/input/
\$ bin/hdfs dfs -cat /user/sdesaleg137/wordcount/input/file01
\$ bin/hdfs dfs -cat /user/sdesaleg137/wordcount/input/file02

In summary, the commands create directories wordcount and wordcount/input in HDFS, create two text files file01 and file02 with specified content, ensure the existence of a directory /user/sdesaleg137/wordcount/input in HDFS, upload the local files to the HDFS directory, list the contents of the HDFS input directory, and finally, display the content of both file01 and file02 from the HDFS input directory. These commands collectively set up a basic Hadoop environment for running MapReduce jobs, with sample input files ready for processing.

```
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ vi WordCount.java
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hadoop com.sun.tools.javac.Main WordCount.java
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ jar cf wc.jar WordCount*.class
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir wordcount/input
mkdir: `hdfs://localhost:9000/user/sdesaleg137/wordcount': No such file or directory
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user
mkdir: `/user': File exists
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir /user/sdesaleg137
mkdir: `/user/sdesaleg137': File exists
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir wordcount
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir wordcount/input
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -put wordcount/input/* wordcount/input/
put: `wordcount/input/*': No such file or directory
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ ^C
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ echo "Hello World Bye World" > wordcount/input/file01
-bash: wordcount/input/file01: No such file or directory
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ mkdir -p wordcount/input
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ echo "Hello World Bye World" > wordcount/input/file01
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ echo "Hello Hadoop Goodbye Hadoop" > wordcount/input/file02
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -put wordcount/input/* /user/sdesaleg137/wordcount/input/
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hadoop fs -ls /user/sdesaleg137/wordcount/input/
Found 2 items
-rw-r--r-- 1 sdesaleg137 supergroup          22 2024-05-31 02:23 /user/sdesaleg137/wordcount/input/file01
-rw-r--r-- 1 sdesaleg137 supergroup          28 2024-05-31 02:23 /user/sdesaleg137/wordcount/input/file02
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hdfs dfs -mkdir wordcount/output
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hadoop fs -cat /user/sdesaleg137/wordcount/input/file01
Hello World Bye World
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hadoop fs -cat /user/sdesaleg137/wordcount/input/file02
Hello Hadoop Goodbye Hadoop
```

- ✓ Then run the application,
bin/hadoop jar wc.jar WordCount /user/sdesaleg137/wordcount/input
/user/sdesaleg137/wordcount/new_output

```
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hadoop jar wc.jar WordCount /user/sdesaleg137/wordcount/input /user/sdesaleg137/wordcount/new_output
2024-05-31 02:27:43,012 INFO client.DefaultNoHARMAFailoverProxyProvider: Connecting to ResourceManager at /0.0.0.0:8032
2024-05-31 02:27:43,339 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application wi
th ToolRunner to remedy this.
2024-05-31 02:27:43,347 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/sdesaleg137/.staging/job_1717121553430_0001
2024-05-31 02:27:43,396 INFO input.FileInputFormat: Total input files to process : 2
2024-05-31 02:27:44,839 INFO mapreduce.JobSubmitter: number of splits:2
2024-05-31 02:27:45,352 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1717121553430_0001
2024-05-31 02:27:45,352 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-05-31 02:27:45,483 INFO conf.Configuration: resource-types.xml not found
2024-05-31 02:27:45,483 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-05-31 02:27:45,890 INFO impl.YarnClientImpl: Submitted application application_1717121553430_0001
2024-05-31 02:27:45,914 INFO mapreduce.Job: The url to track the job: http://ubuntutask.us-central1-c.c.ubuntu-project-424814.internal:8088/proxy/application_1717121553430_0001/
2024-05-31 02:27:45,914 INFO mapreduce.Job: Running job: job_1717121553430_0001
2024-05-31 02:27:52,002 INFO mapreduce.Job: Job job_1717121553430_0001 running in uber mode : false
2024-05-31 02:27:52,003 INFO mapreduce.Job: map 0% reduce 0%
2024-05-31 02:27:57,057 INFO mapreduce.Job: map 100% reduce 0%
2024-05-31 02:28:01,080 INFO mapreduce.Job: map 100% reduce 100%
2024-05-31 02:28:01,087 INFO mapreduce.Job: Job job_1717121553430_0001 completed successfully
2024-05-31 02:28:01,157 INFO mapreduce.Job: Counters: 54
File System Counters
  FILE: Number of bytes read=79
  FILE: Number of bytes written=925795
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=302
  HDFS: Number of bytes written=41
  HDFS: Number of read operations=11
  HDFS: Number of large read operations=0
```

```
Total megabyte-milliseconds taken by all reduce tasks=1930240
Map-Reduce Framework
  Map input records=2
  Map output records=8
  Map output bytes=82
  Map output materialized bytes=85
  Input split bytes=252
  Combine input records=8
  Combine output records=6
  Reduce input groups=5
  Reduce shuffle bytes=85
  Reduce input records=6
  Reduce output records=5
  Spilled Records=12
  Shuffled Maps =2
  Failed Shuffles=0
  Merged Map outputs=2
  GC time elapsed (ms)=174
  CPU time spent (ms)=1320
  Physical memory (bytes) snapshot=845148160
  Virtual memory (bytes) snapshot=7646793728
  Total committed heap usage (bytes)=935854080
  Peak Map Physical memory (bytes)=317845504
  Peak Map Virtual memory (bytes)=2547949568
  Peak Reduce Physical memory (bytes)=256045056
  Peak Reduce Virtual memory (bytes)=2553839616
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=50
File Output Format Counters
  Bytes Written=41
```

- ✓ Display the output

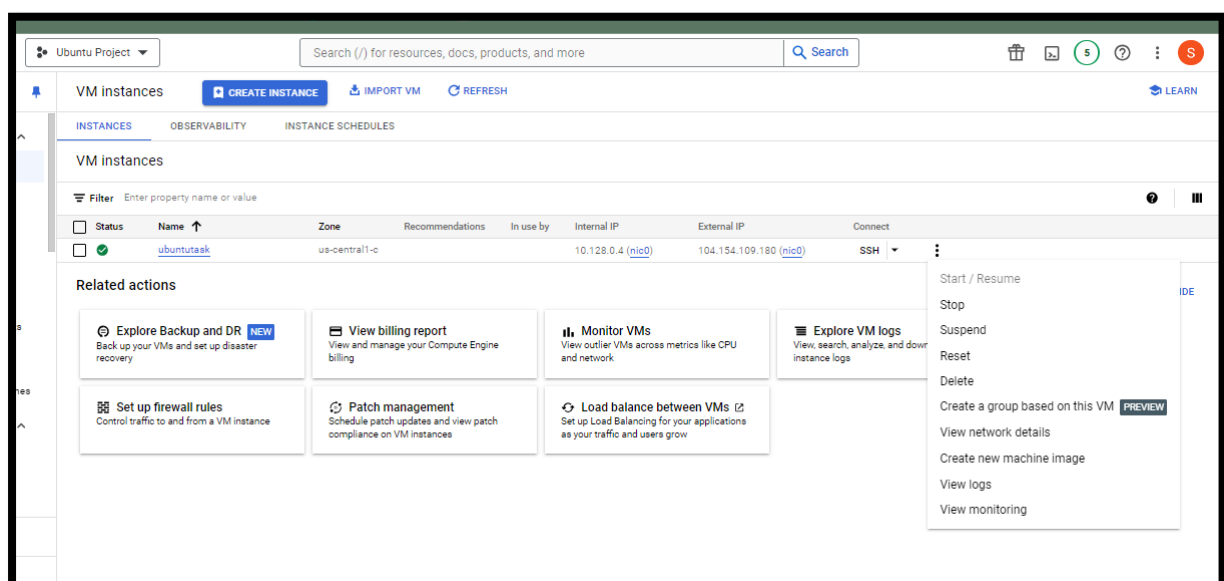
```
$ bin/hadoop fs -cat /user/sdesaleg137/wordcount/new_output/part-r-00000
```

```
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ bin/hadoop fs -cat /user/sdesaleg137/wordcount/new_output/part-r-00000
Bye      1
Goodbye  1
Hadoop   2
Hello    2
World    2
sdesaleg137@ubuntutask:~/hadoop-3.4.0$
```

- ✓ Stop the daemons
- ```
$ sbin/stop-dfs.sh
$ sbin/stop-yarn.sh Or
$ sbin/stop-all.sh
```

```
sdesaleg137@ubuntutask:~/hadoop-3.4.0/etc/hadoop$ cd ../..
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ sbin/stop-dfs.sh
Stopping namenodes on [localhost]
localhost: sdesaleg137@localhost: Permission denied (publickey).
pdsh@ubuntutask: localhost: ssh exited with exit code 255
Stopping datanodes
localhost: sdesaleg137@localhost: Permission denied (publickey).
pdsh@ubuntutask: localhost: ssh exited with exit code 255
Stopping secondary namenodes [ubuntutask]
ubuntutask: sdesaleg137@ubuntutask: Permission denied (publickey).
pdsh@ubuntutask: ubuntutask: ssh exited with exit code 255
sdesaleg137@ubuntutask:~/hadoop-3.4.0$ sbin/stop-yarn.sh
Stopping nodemanagers
localhost: sdesaleg137@localhost: Permission denied (publickey).
pdsh@ubuntutask: localhost: ssh exited with exit code 255
Stopping resourcemanager
sdesaleg137@ubuntutask:~/hadoop-3.4.0$
```

Then at last, you can stop the VM instance you have been working on.



**DONE!!!**