Exploring Hadoop Ecosystem with Simple Linux Commands

Overview: This assignment is intended to get you more familiar with the Hadoop Ecosystem.

Prerequisites:

1. Google account OR Google Gmail account
   Before proceeding:
   - The user should have a Google account or Google Gmail account at hand.
   - If not, the user should create a new one first

2. Access Google Cloud Platform (GCP) console
   • The user should be able to access his/her Google Cloud Platform (GCP) console

3. An existing project to host the Hadoop-Spark cluster
   • The user has an existing project under this account to host the to-be-created cluster

4. GCP storage bucket ready for use
   • The user has created a GCP storage bucket and have it ready for use.

5. GCP Hadoop and Spark Cluster create with 1 Master Node and 2 Worker Node. The nodes must be turned on for this assignment.

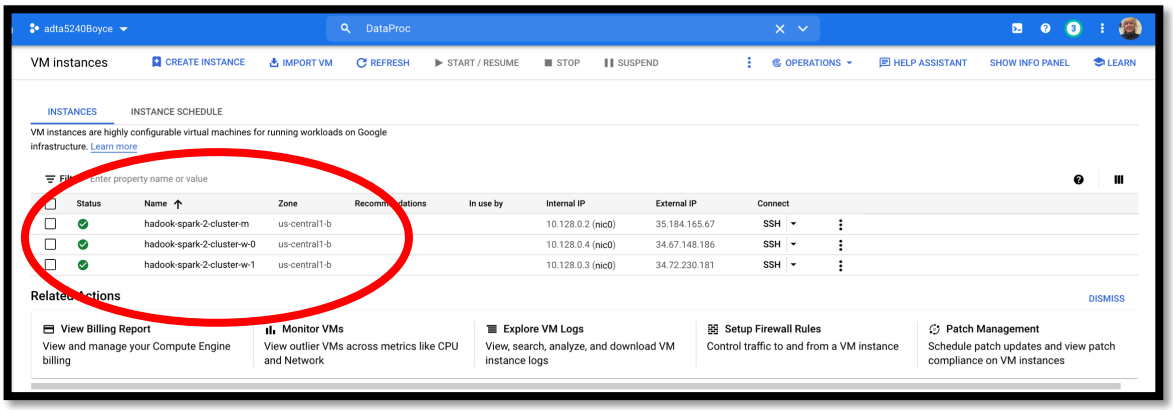
NOTES: Please see the following documents, if you need a refresher.

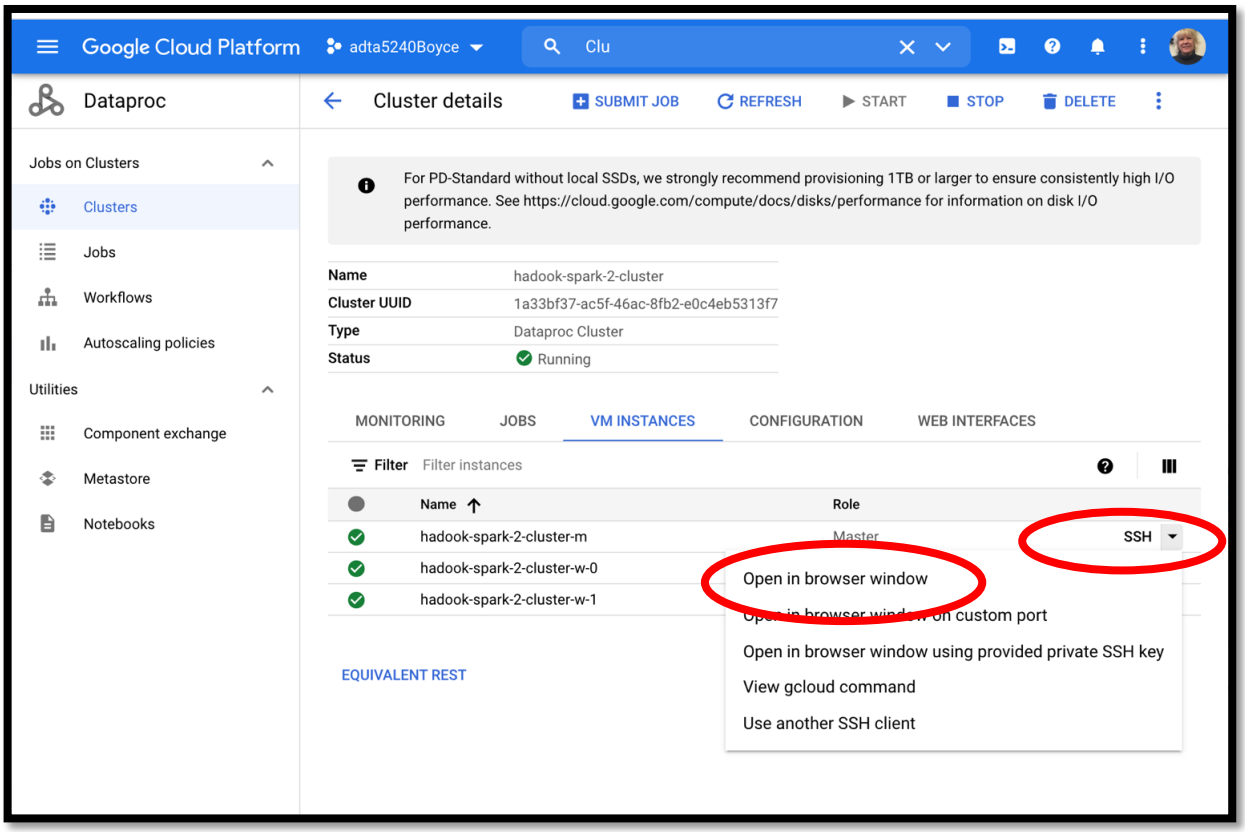
- How to Setup a GCP Account with Free Credit
- How to Create Projects in GCP
- How to Create New Storage Buckets in GCP
- How to Create a Hadoop and Spark Cluster in GCP

VERY IMPORTANT: Be sure all nodes are running in GCP.

Step One: Start all 3 nodes in the cluster you have already created.

- You will then Click on the chevron next to SSH
- Click on “Open in browser window.”





Step Two: Explore the Cluster in Hadoop

- Open terminal via SSH in GCP
- See all the services of Hadoop in our cluster
- Use the command
    - whoami

- pwd

```
leannboyce19@hadoop-spark-2-cluster-m:~$
leannboyce19@hadoop-spark-2-cluster-m:~$
leannboyce19@hadoop-spark-2-cluster-m:~$
leannboyce19@hadoop-spark-2-cluster-m:~$ whoami
leannboyce19
leannboyce19@hadoop-spark-2-cluster-m:~$
leannboyce19@hadoop-spark-2-cluster-m:~$ pwd
/home/leannboyce19
leannboyce19@hadoop-spark-2-cluster-m:~$
```

  - These command lines show you your user name and the home directory
- Enter the command
  - ps -ef | grep -i hadoop
  - This will list all the processing currently running
  - Remember when we set up Hadoop all of these services were setup when setup Hadoop and Spark Cluster with Dataproc

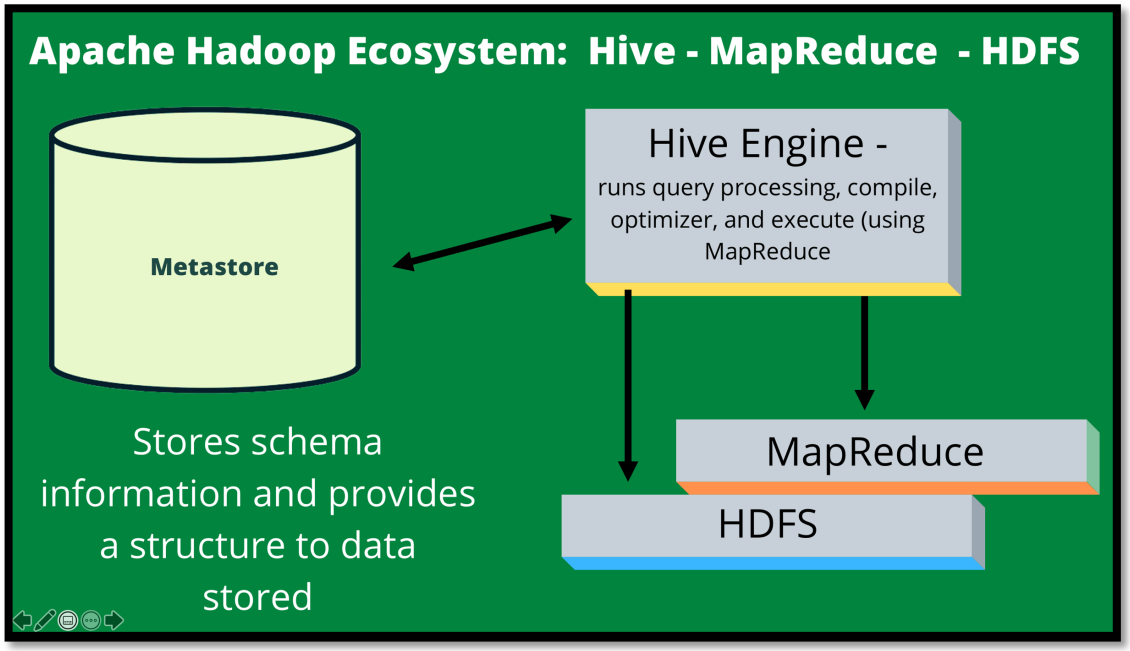
ssh.cloud.google.com/projects/double-port-320104/zones/us-central1-a/instances/hadoop-spark-2-cluster-m?authuser=0&hl=en_US&prc

```
Connected, host fingerprint: ssh-rsa 0 10:A6:BA:44:F6:14:6F:ED:5C:8B:3F:C8:AA:52
:7F:72:E8:31:CA:28:43:B4:35:D4:10:16:A1:74:8D:45:19:9F
Linux hadoop-spark-2-cluster-m 5.10.0-0.bpo.7-amd64 #1 SMP Debian 5.10.40-1~bpo1
0+1 (2021-06-04) 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 Jul 21 05:02:44 2021 from 35.235.241.18
leannboyce19@hadoop-spark-2-cluster-m:~$ whoami
leannboyce19
leannboyce19@hadoop-spark-2-cluster-m:~$ pwd
/home/leannboyce19
leannboyce19@hadoop-spark-2-cluster-m:~$ ps -ef | grep -i hadoop
root       819     1  0 03:17 ?        00:00:36 /usr/bin/java -XX:+AlwaysPreTouch -Xms1605m -Xmx1605m -XX:+CrashOnO
utOfMemoryError -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/var/crash/google-dataproc-agent.hprof -Djava.util
.logging.config.file=/etc/google-dataproc/logging.properties -cp /usr/local/share/google/dataproc/dataproc-agent.ja
r:/etc/hadoop/conf:/usr/lib/hadoop/lib/*:/usr/lib/hadoop/.//*:/usr/lib/hadoop-hdfs/./:/usr/lib/hadoop-hdfs/lib/*:/u
sr/lib/hadoop-hdfs/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoop-yarn/.//*:/usr/lib/hadoop-mapreduce/lib/*:/usr/l
ib/hadoop-mapreduce/.//*:/usr/local/share/google/dataproc/lib/* com.google.cloud.hadoop.services.agent.AgentMain /u
sr/local/share/google/dataproc/startup-script.sh /usr/local/share/google/dataproc/post-hdfs-startup-script.sh
mapred     3124     1  0 03:17 ?        00:00:28 /usr/lib/jvm/adoptopenjdk-8-hotspot-amd64/bin/java -Dproc_historyse
rver -Xmx4000m -Dhadoop.log.dir=/usr/lib/hadoop/logs -Dhadoop.log.file=hadoop.log -Dhadoop.home.dir=/usr/lib/hadoop
 -Dhadoop.id.str= -Dhadoop.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.policy.
file=hadoop-policy.xml -Djava.net.preferIPv4Stack=true -Dhadoop.log.dir=/var/log/hadoop-mapreduce -Dhadoop.log.file
=hadoop.log -Dhadoop.root.logger=INFO,console -Dhadoop.id.str=mapred -Dhadoop.log.dir=/usr/lib/hadoop/logs -Dhadoop
.log.file=hadoop.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str= -Dhadoop.root.logger=INFO,console -Djava.li
brary.path=/usr/lib/hadoop/lib/native -Dhadoop.policy.file=hadoop-policy.xml -Djava.net.preferIPv4Stack=true -Dhado
op.log.dir=/var/log/hadoop-mapreduce -Dhadoop.log.file=mapred-mapred-historyserver-hadoop-spark-2-cluster-m.log -Dh
adoop.root.logger=INFO,RFA -Dmapred.jobsummary.logger=INFO,JSA -XX:+UseConcMarkSweepGC -XX:+PrintGCTimeStamps -XX:+
PrintGCDateStamps -XX:+PrintGCDetails -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.mapreduce.v2.hs.
JobHistoryServer
yarn       3134     1  0 03:17 ?        00:01:03 /usr/lib/jvm/adoptopenjdk-8-hotspot-amd64/bin/java -Dproc_resourcem
anager -Xmx4000m -Dhadoop.log.dir=/var/log/hadoop-yarn -Dyarn.log.dir=/var/log/hadoop-yarn -Dhadoop.log.file=yarn-y
arn-resourcemanager-hadoop-spark-2-cluster-m.log -Dyarn.log.file=yarn-yarn-resourcemanager-hadoop-spark-2-cluster-m
.log -Dyarn.home.dir= -Dyarn.id.str=yarn -Dhadoop.root.logger=INFO,RFA -Dyarn.root.logger=INFO,RFA -Djava.library.p
ath=/usr/lib/hadoop/lib/native -Dyarn.policy.file=hadoop-policy.xml -Xmx12845m -Dhadoop.log.dir=/var/log/hadoop-yar
n -Dyarn.log.dir=/var/log/hadoop-yarn -Dhadoop.log.file=yarn-yarn-resourcemanager-hadoop-spark-2-cluster-m.log -Dya
rn.log.file=yarn-yarn-resourcemanager-hadoop-spark-2-cluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dhadoop.ho
me.dir=/usr/lib/hadoop -Dhadoop.root.logger=INFO,RFA -Dyarn.root.logger=INFO,RFA -Djava.library.path=/usr/lib/hadoo
p/lib/native -classpath /etc/hadoop/conf:/etc/hadoop/conf:/etc/hadoop/conf:/usr/lib/hadoop/lib/*:/usr/lib/hadoop/./
/*:/usr/lib/hadoop-hdfs/./:/usr/lib/hadoop-hdfs/lib/*:/usr/lib/hadoop-hdfs/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib
/hadoop-yarn/.//*:/usr/lib/hadoop-mapreduce/lib/*:/usr/lib/hadoop-mapreduce/.//*:/usr/lib/spark/yarn/*::/usr/local/
share/google/dataproc/lib/*:/usr/local/share/google/dataproc/lib/*:/usr/local/share/google/dataproc/lib/*:/usr/lib/
hadoop-yarn/.//*:/usr/lib/hadoop-yarn/lib/*:/etc/hadoop/conf/rm-config/log4j.properties:/usr/lib/hadoop-yarn/.//tim
elineservice/*:/usr/lib/hadoop-yarn/.//timelineservice/lib/* org.apache.hadoop.yarn.server.resourcemanager.Resource
Manager
```

- How to move up and down on the terminal window
  - Click on the Setting icon in the upper right-hand side of the terminal
  - Click on “How to Copy and Paste” This will take you to the directions on how to scroll.
- You can scroll the terminal using your mouse wheel or trackpad. Alternatively, the `Ctrl+Shift+PageUp`/`Ctrl+Shift+PageDn` keyboard shortcuts scroll the terminal on Windows and Linux, and `Fn+Shift+Up`/`Fn+Shift+Down` scroll the terminal on macOS.
- So, what does this all mean? These are all the components of the Hadoop Ecosystem.
  - At the top, you see root. The process number is 819. The process is what is needed to run a program, a Hadoop component. The process number is an ID for that program, if you will. It is very important in the Ecosystem as you could shut down a process with a command using the process ID number.
  - Then you see mapred. The process number is 3124 that is running JobHisotry Server.
  - Next is yarn. The process number is 3134 that is running the Resource Manager

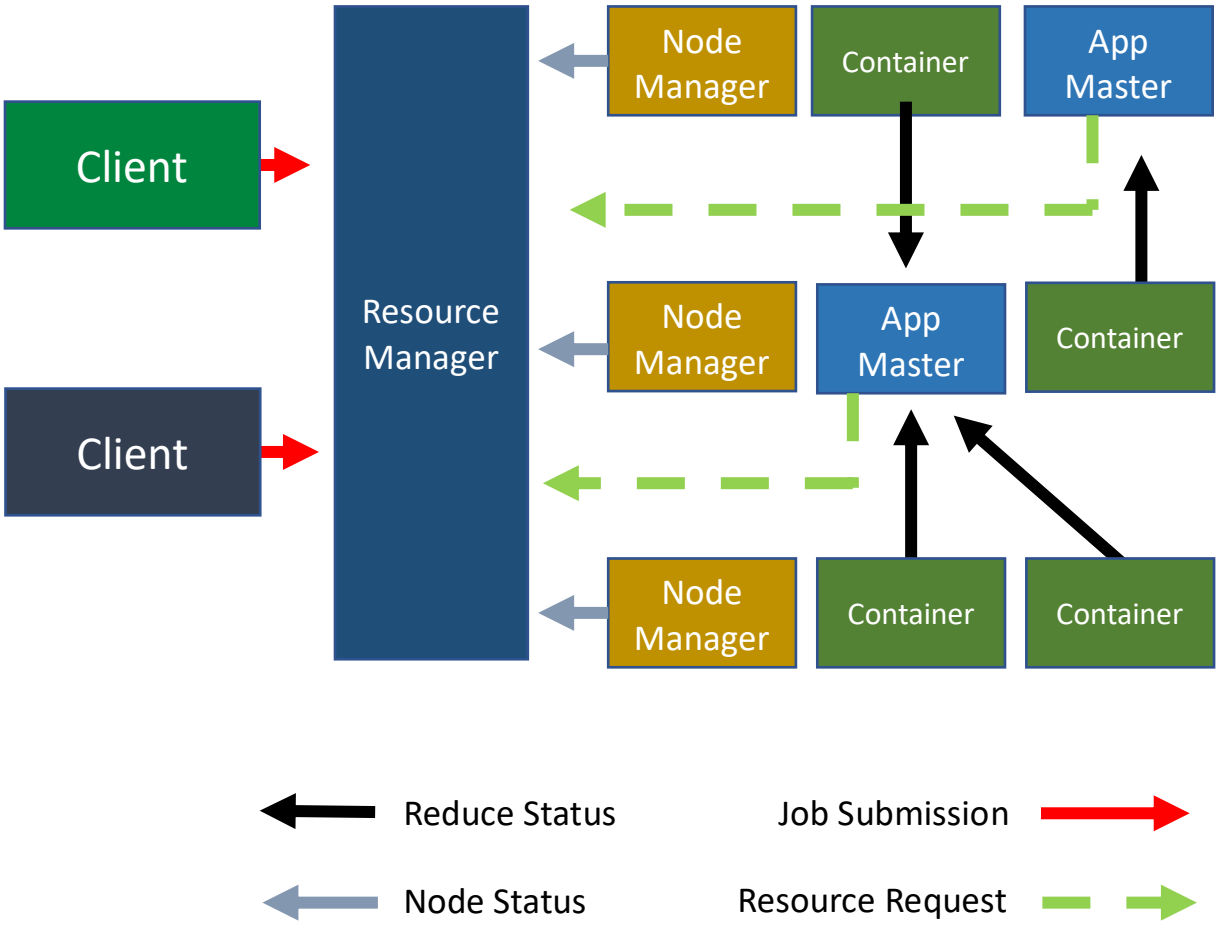
- Next is yarn. The process number is 3135 that is running the Application History Server
- Next is hive with a process number 3383 that is running the HiveMetastore (see below)
- Then you see hdfs with a process number 3749 that is running the NameNode
- Then there is another hdfs with a process number of 4400 that is running the Secondary NameNode
- Then you see another hive with a process number of 5641 that is running the HiveServer 2
- And lastly you see spark with a process number of 5687 that is running HistoryServer.
- Is this sounding familiar?
- Take note of each service, process ID of each service and what each is running.

- Let's look back at Hive from our lecture



- Metastore and Hive Server (Engine) are critical to run Hive.

- Let's look back at YARN Architecture from our lecture.
    - The Resource Manager (Master Node) is a major component of Yarn
    - This is so that it can work with the Application Master and Node Master or worker nodes



- Let's once again look at the HDFS Architecture from the same lecture