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

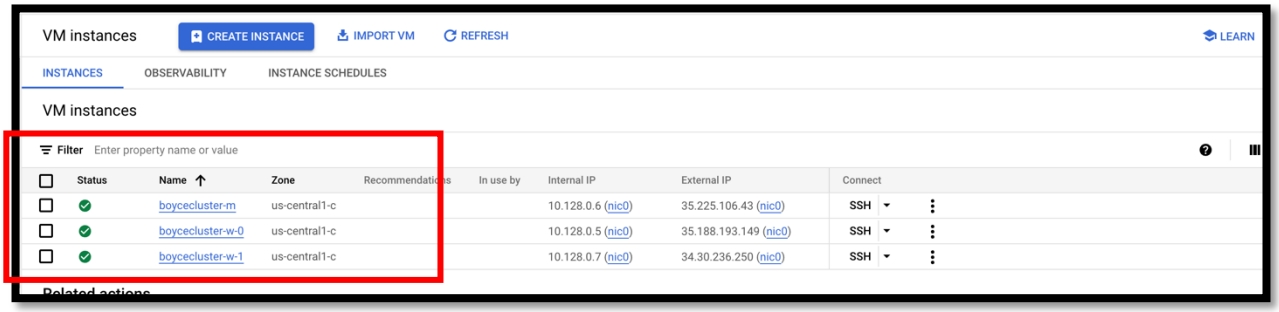
4. GCP storage bucket ready for use
   - The user has created a GCP storage bucket and have it ready for use.

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.

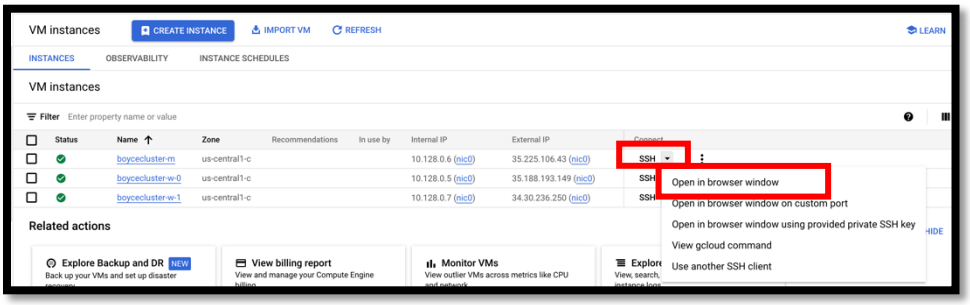
- How to Setup a GCP Account with Free Credit
- How to Create Projects in GCP
- How to Create New Storage Buckets in GCP
- How to Create a Hadoop and Spark Cluster in GCP

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

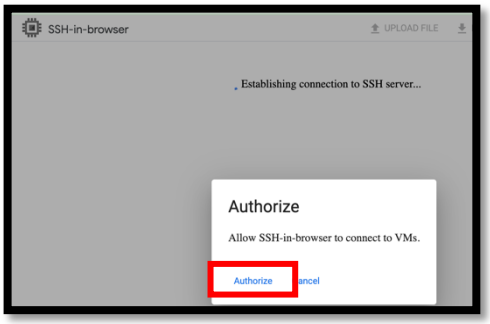


Step One: If you have not started the cluster, start all 3 nodes in the cluster you have already created.

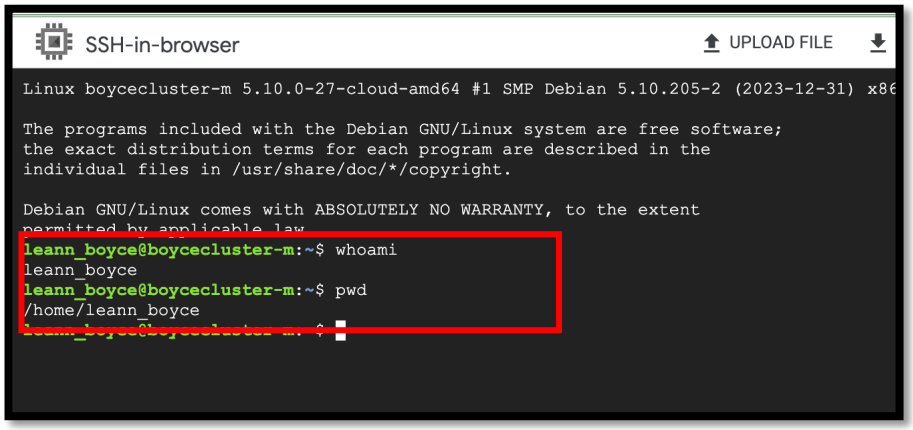
- You will then Click on the chevron next to SSH
- Click on "Open in browser window."



- You may need to click "Authorize" to proceed in opening terminal via SSH in GCP

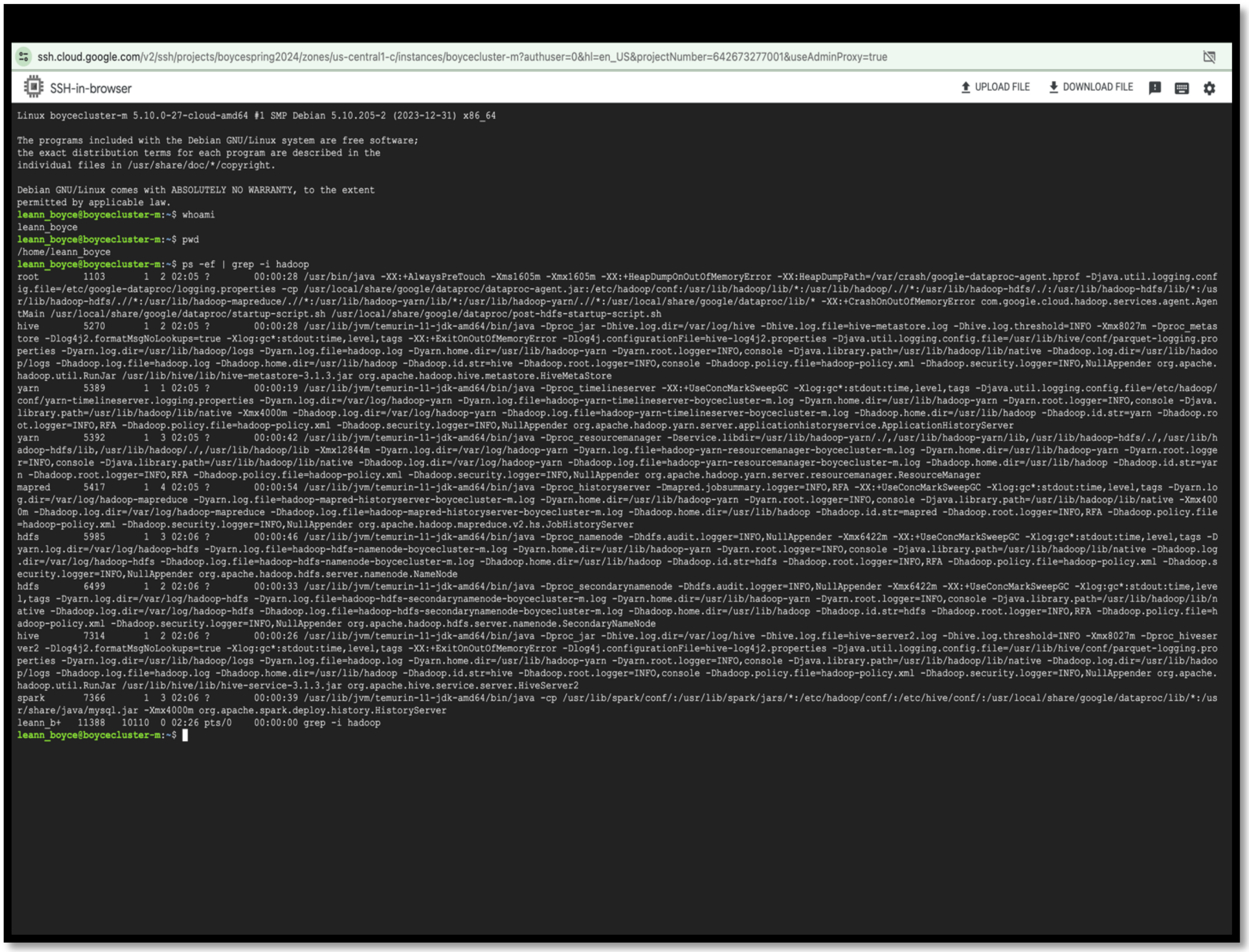


- See all the services of Hadoop in our cluster
- Use the command
    - whoami
    - pwd



    - 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



```
Linux boycecluster-m 5.10.0-27-cloud-amd64 #1 SMP Debian 5.10.205-2 (2023-12-31) 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.
leann_boyce@boycecluster-m:~$ whoami
leann_boyce
leann_boyce@boycecluster-m:~$ pwd
/home/leann_boyce
leann_boyce@boycecluster-m:~$ ps -ef | grep -i hadoop
root        1103       1  2 02:05 ?        00:00:28 /usr/bin/java -XX:+AlwaysPreTouch -Xms1605m -Xmx1605m -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/var/crash/google-dataproc-agent.hprof -Djava.util.logging.conf
ig.file=/etc/google-dataproc/logging.properties -cp /usr/local/share/google/dataproc/dataproc-agent.jar:/etc/hadoop/conf:/usr/lib/hadoop/lib/*:/usr/lib/hadoop/.//*:/usr/lib/hadoop-hdfs/./:/usr/lib/hadoop-hdfs/lib/*:/us
r/lib/hadoop-hdfs/.//*:/usr/lib/hadoop-mapreduce/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoop-yarn/.//*:/usr/local/share/google/dataproc/lib/* -XX:+CrashOnOutOfMemoryError com.google.cloud.hadoop.services.agent.Agen
tMain /usr/local/share/google/dataproc/startup-script.sh /usr/local/share/google/dataproc/post-hdfs-startup-script.sh
hive        5270       1  2 02:05 ?        00:00:28 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_jar -Dhive.log.dir=/var/log/hive -Dhive.log.file=hive-metastore.log -Dhive.log.threshold=INFO -Xmx8027m -Dproc_metas
tore -Dlog4j2.formatMsgNoLookups=true -Xlog:gc*:stdout:time,level,tags -XX:+ExitOnOutOfMemoryError -Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/usr/lib/hive/conf/parquet-logging.pro
perties -Dyarn.log.dir=/usr/lib/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log.dir=/usr/lib/hadoo
p/logs -Dhadoop.log.file=hadoop.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=hive -Dhadoop.root.logger=INFO,console -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.
hadoop.util.RunJar /usr/lib/hive/lib/hive-metastore-3.1.3.jar org.apache.hadoop.hive.metastore.HiveMetaStore
yarn        5389       1  1 02:05 ?        00:00:19 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_timelineserver -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,level,tags -Djava.util.logging.config.file=/etc/hadoop/
conf/yarn-timelineserver.logging.properties -Dyarn.log.dir=/var/log/hadoop-yarn -Dyarn.log.file=hadoop-yarn-timelineserver-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.
library.path=/usr/lib/hadoop/lib/native -Xmx4000m -Dhadoop.log.dir=/var/log/hadoop-yarn -Dhadoop.log.file=hadoop-yarn-timelineserver-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=yarn -Dhadoop.ro
ot.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.yarn.server.applicationhistoryservice.ApplicationHistoryServer
yarn        5392       1  3 02:05 ?        00:00:42 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_resourcemanager -Dservice.libdir=/usr/lib/hadoop-yarn/./,/usr/lib/hadoop-yarn/lib,/usr/lib/hadoop-hdfs/./,/usr/lib/h
adoop-hdfs/lib,/usr/lib/hadoop/./,/usr/lib/hadoop/lib -Xmx12844m -Dyarn.log.dir=/var/log/hadoop-yarn -Dyarn.log.file=hadoop-yarn-resourcemanager-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logge
r=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log.dir=/var/log/hadoop-yarn -Dhadoop.log.file=hadoop-yarn-resourcemanager-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=yar
n -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.yarn.server.resourcemanager.ResourceManager
mapred      5417       1  4 02:05 ?        00:00:54 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_historyserver -Dmapred.jobsummary.logger=INFO,RFA -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,level,tags -Dyarn.lo
g.dir=/var/log/hadoop-mapreduce -Dyarn.log.file=hadoop-mapred-historyserver-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Xmx400
0m -Dhadoop.log.dir=/var/log/hadoop-mapreduce -Dhadoop.log.file=hadoop-mapred-historyserver-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=mapred -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file
=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.mapreduce.v2.hs.JobHistoryServer
hdfs        5985       1  3 02:06 ?        00:00:46 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_namenode -Dhdfs.audit.logger=INFO,NullAppender -Xmx6422m -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,level,tags -D
yarn.log.dir=/var/log/hadoop-hdfs -Dyarn.log.file=hadoop-hdfs-namenode-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log
.dir=/var/log/hadoop-hdfs -Dhadoop.log.file=hadoop-hdfs-namenode-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=hdfs -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.s
ecurity.logger=INFO,NullAppender org.apache.hadoop.hdfs.server.namenode.NameNode
hdfs        6499       1  2 02:06 ?        00:00:33 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_secondarynamenode -Dhdfs.audit.logger=INFO,NullAppender -Xmx6422m -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,leve
l,tags -Dyarn.log.dir=/var/log/hadoop-hdfs -Dyarn.log.file=hadoop-hdfs-secondarynamenode-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/n
ative -Dhadoop.log.dir=/var/log/hadoop-hdfs -Dhadoop.log.file=hadoop-hdfs-secondarynamenode-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=hdfs -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=h
adoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.hdfs.server.namenode.SecondaryNameNode
hive        7314       1  2 02:06 ?        00:00:26 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_jar -Dhive.log.dir=/var/log/hive -Dhive.log.file=hive-server2.log -Dhive.log.threshold=INFO -Xmx8027m -Dproc_hiveser
ver2 -Dlog4j2.formatMsgNoLookups=true -Xlog:gc*:stdout:time,level,tags -XX:+ExitOnOutOfMemoryError -Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/usr/lib/hive/conf/parquet-logging.pro
perties -Dyarn.log.dir=/usr/lib/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log.dir=/usr/lib/hadoo
p/logs -Dhadoop.log.file=hadoop.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=hive -Dhadoop.root.logger=INFO,console -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.
hadoop.util.RunJar /usr/lib/hive/lib/hive-service-3.1.3.jar org.apache.hive.service.server.HiveServer2
spark       7366       1  3 02:06 ?        00:00:39 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -cp /usr/lib/spark/conf/:/usr/lib/spark/jars/*:/etc/hadoop/conf/:/etc/hive/conf/:/usr/local/share/google/dataproc/lib/*:/us
r/share/java/mysql.jar -Xmx4000m org.apache.spark.deploy.history.HistoryServer
leann_b+   11388   10110  0 02:26 pts/0    00:00:00 grep -i hadoop
leann_boyce@boycecluster-m:~$ 
```

- 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.

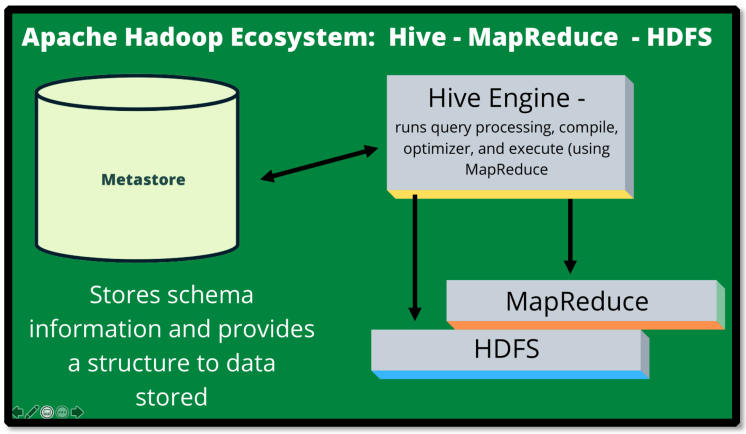
```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
leann_boyce@boycecluster-m:~$ whoami
leann_boyce
leann_boyce@boycecluster-m:~$ pwd
/home/leann_boyce
leann_boyce@boycecluster-m:~$ ps -ef | grep -i hadoop
root        1103       1  2 02:05 ?        00:00:28 /usr/bin/java -XX:+AlwaysPreTouch -Xms1605m -Xmx1605m -XX:+HeapDumpOnOutOfMemoryError -XX:
HeapDumpPath=/var/crash/google-dataproc-agent.hprof -Djava.util.logging.config.file=/etc/google-dataproc/logging.properties -cp /usr/local/sha
re/google/dataproc/dataproc-agent.jar:/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-mapreduce/.//*:/usr/lib/hadoop-yarn/lib/*:/usr/lib/hadoop-yarn/.//*:/usr/local/share/google/d
ataproc/lib/* -XX:+CrashOnOutOfMemoryError com.google.cloud.hadoop.services.agent.AgentMain /usr/local/share/google/dataproc/startup-script.sh
 /usr/local/share/google/dataproc/post-hdfs-startup-script.sh
hive        5270       1  2 02:05 ?        00:00:28 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_jar -Dhive.log.dir=/var/log/hive -Dhive.
log.file=hive-metastore.log -Dhive.log.threshold=INFO -Xmx8027m -Dproc_metastore -Dlog4j2.formatMsgNoLookups=true -Xlog:gc*:stdout:time,level,
tags -XX:+ExitOnOutOfMemoryError -Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/usr/lib/hive/conf/parquet-l
ogging.properties -Dyarn.log.dir=/usr/lib/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,
console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log.dir=/usr/lib/hadoop/logs -Dhadoop.log.file=hadoop.log -Dhadoop.home.dir=/u
sr/lib/hadoop -Dhadoop.id.str=hive -Dhadoop.root.logger=INFO,console -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,Null
Appender org.apache.hadoop.util.RunJar /usr/lib/hive/lib/hive-metastore-3.1.3.jar org.apache.hadoop.hive.metastore.HiveMetaStore
yarn        5389       1  1 02:05 ?        00:00:19 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_timelineserver -XX:+UseConcMarkSweepGC -
Xlog:gc*:stdout:time,level,tags -Djava.util.logging.config.file=/etc/hadoop/conf/yarn-timelineserver.logging.properties -Dyarn.log.dir=/var/lo
g/hadoop-yarn -Dyarn.log.file=hadoop-yarn-timelineserver-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,consc
le -Djava.library.path=/usr/lib/hadoop/lib/native -Xmx4000m -Dhadoop.log.dir=/var/log/hadoop-yarn -Dhadoop.log.file=hadoop-yarn-timelineserver
-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=yarn -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xm
l -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.yarn.server.applicationhistoryservice.ApplicationHistoryServer
yarn        5392       1  3 02:05 ?        00:00:42 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_resourcemanager -Dservice.libdir=/usr/li
b/hadoop-yarn/./,/usr/lib/hadoop-yarn/lib,/usr/lib/hadoop-hdfs/./,/usr/lib/hadoop-hdfs/lib,/usr/lib/hadoop/./,/usr/lib/hadoop/lib -Xmx12844m -
Dyarn.log.dir=/var/log/hadoop-yarn -Dyarn.log.file=hadoop-yarn-resourcemanager-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.
root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log.dir=/var/log/hadoop-yarn -Dhadoop.log.file=hadoop-yarn-re
sourcemanager-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.str=yarn -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hac
oop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.yarn.server.resourcemanager.ResourceManager
mapred      5417       1  4 02:05 ?        00:00:54 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_historyserver -Dmapred.jobsummary.logger
=INFO,RFA -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,level,tags -Dyarn.log.dir=/var/log/hadoop-mapreduce -Dyarn.log.file=hadoop-mapred-hist
oryserver-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/nati
ve -Xmx4000m -Dhadoop.log.dir=/var/log/hadoop-mapreduce -Dhadoop.log.file=hadoop-mapred-historyserver-boycecluster-m.log -Dhadoop.home.dir=/us
r/lib/hadoop -Dhadoop.id.str=mapred -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullApp
ender org.apache.hadoop.mapreduce.v2.hs.JobHistoryServer
```

- At the top, you see root. The process number is 1103. 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 hive. The process number is 5270 that is running HiveMetaStore
- Next is yarn. The process number is 5389 that is running the Application History Server
- Next is yarn. The process number is 5392 that is running the Resource Manager
- Next is mapred with a process number 5417 that is running the JobHistoryServer

```
hdfs        5985       1  3 02:06 ?        00:00:46 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_namenode -Dhdfs.audit.logger=INFO,NullAp
pender -Xmx6422m -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,level,tags -Dyarn.log.dir=/var/log/hadoop-hdfs -Dyarn.log.file=hadoop-hdfs-name
node-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/hadoop/lib/native -D
hadoop.log.dir=/var/log/hadoop-hdfs -Dhadoop.log.file=hadoop-hdfs-namenode-boycecluster-m.log -Dhadoop.home.dir=/usr/lib/hadoop -Dhadoop.id.st
r=hdfs -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppender org.apache.hadoop.hdfs.
server.namenode.NameNode
hdfs        6499       1  2 02:06 ?        00:00:33 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_secondarynamenode -Dhdfs.audit.logger=IN
FO,NullAppender -Xmx6422m -XX:+UseConcMarkSweepGC -Xlog:gc*:stdout:time,level,tags -Dyarn.log.dir=/var/log/hadoop-hdfs -Dyarn.log.file=hadoop-
hdfs-secondarynamenode-boycecluster-m.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,console -Djava.library.path=/usr/lib/ha
doop/lib/native -Dhadoop.log.dir=/var/log/hadoop-hdfs -Dhadoop.log.file=hadoop-hdfs-secondarynamenode-boycecluster-m.log -Dhadoop.home.dir=/us
r/lib/hadoop -Dhadoop.id.str=hdfs -Dhadoop.root.logger=INFO,RFA -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,NullAppen
der org.apache.hadoop.hdfs.server.namenode.SecondaryNameNode
hive        7314       1  2 02:06 ?        00:00:26 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -Dproc_jar -Dhive.log.dir=/var/log/hive -Dhive.
log.file=hive-server2.log -Dhive.log.threshold=INFO -Xmx8027m -Dproc_hiveserver2 -Dlog4j2.formatMsgNoLookups=true -Xlog:gc*:stdout:time,level,
tags -XX:+ExitOnOutOfMemoryError -Dlog4j.configurationFile=hive-log4j2.properties -Djava.util.logging.config.file=/usr/lib/hive/conf/parquet-l
ogging.properties -Dyarn.log.dir=/usr/lib/hadoop/logs -Dyarn.log.file=hadoop.log -Dyarn.home.dir=/usr/lib/hadoop-yarn -Dyarn.root.logger=INFO,
console -Djava.library.path=/usr/lib/hadoop/lib/native -Dhadoop.log.dir=/usr/lib/hadoop/logs -Dhadoop.log.file=hadoop.log -Dhadoop.home.dir=/u
sr/lib/hadoop -Dhadoop.id.str=hive -Dhadoop.root.logger=INFO,console -Dhadoop.policy.file=hadoop-policy.xml -Dhadoop.security.logger=INFO,Null
Appender org.apache.hadoop.util.RunJar /usr/lib/hive/lib/hive-service-3.1.3.jar org.apache.hive.service.server.HiveServer2
spark       7366       1  3 02:06 ?        00:00:39 /usr/lib/jvm/temurin-11-jdk-amd64/bin/java -cp /usr/lib/spark/conf/:/usr/lib/spark/jars/*:
/etc/hadoop/conf/:/etc/hive/conf/:/usr/local/share/google/dataproc/lib/*:/usr/share/java/mysql.jar -Xmx4000m org.apache.spark.deploy.history.H
istoryServer
```

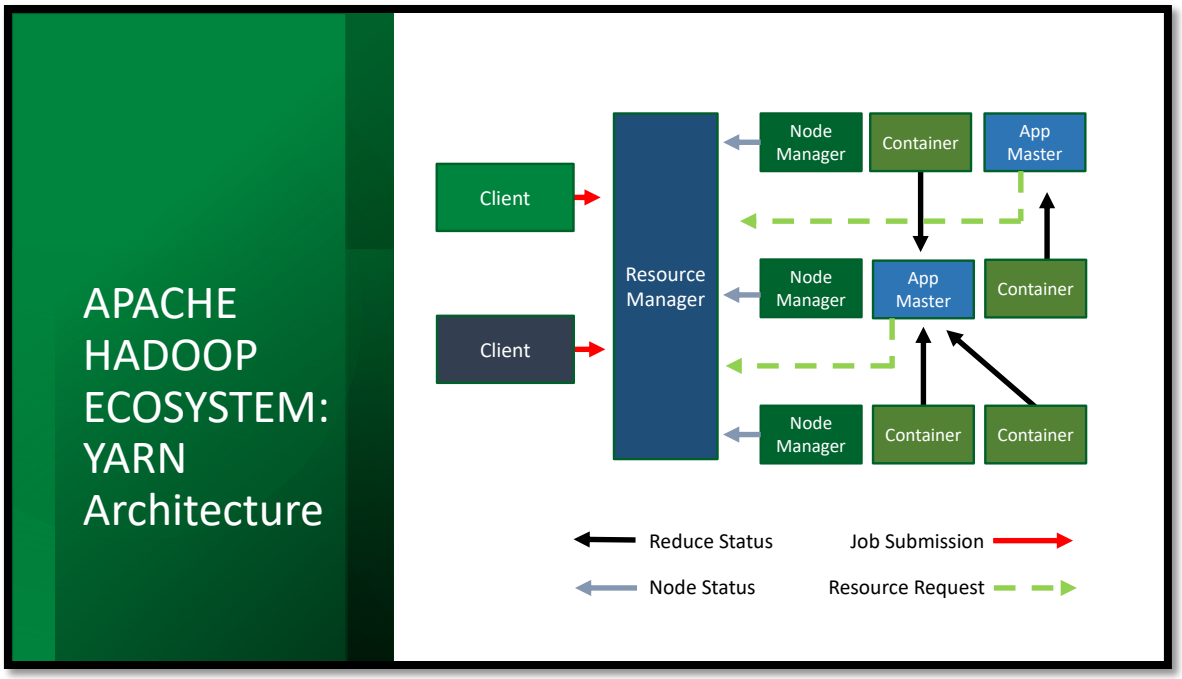
- Then you see hdfs. The process number is 5985 that is running NameNode
- Next is hdfs. The process number is 6499 that is running the Secondary NameNode
- Next is hive. The process number is 7314 that is running the HiveServer2
- Next is spark with a process number 7366 that is running the HistoryServer

- Is this sounding familiar? You will have different process numbers that I noted but you will have the same services.
- 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.