Amirkabir University of Technology
(Tehran Polytechnic)

Cloud Computing

Hadoop Installation and Beyond

Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Spring 2021

Course Logistics

- **Midterm exam is on 1400/02/18**
  - Closed booked and closed everything indeed ☺
  - See **the study guide** in the course web page.
  - If you want to get high score, you should study the book and solve some practice exercises.

➢ Time to defense your work for HW1

  - Arrange a time with the assigned TA.
  - Good luck ☺

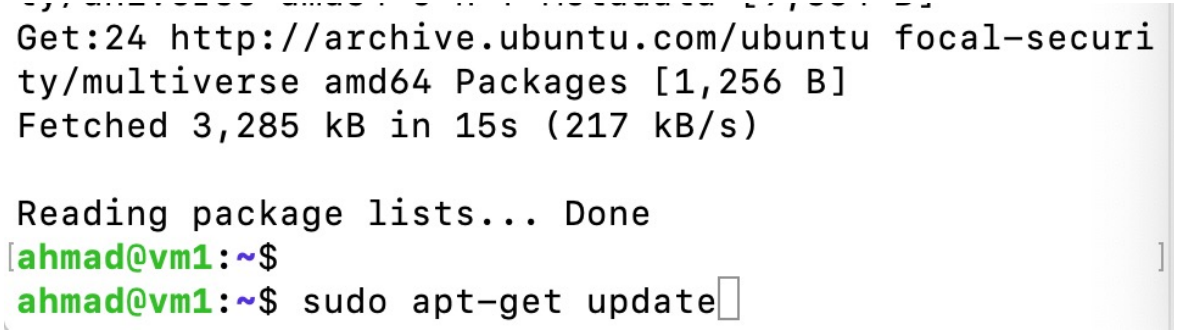
What We Learned from HW1

- Launching VMs using VirtualBox
- Configuring VM specification from the host
- Implementing a simple VM management API
- **What we do next?**
  - We install Hadoop Yarn on **one VM**
    - **You should do it in 3 VMs for your HW2**
  - We run Hadoop jobs on our small cluster

There are Many Guides Online

- I am following the link below
  - https://www.howtoforge.com/how-to-install-and-configure-apache-hadoop-on-ubuntu-2004/
- First, launch one VM and run the update

```
$sudo apt-get update -y
```



Make life easier using host OS terminal and SSH

Do all the installation on a VM and then clone it to save time & bandwidth

Install Java

- Apache Hadoop is a Java-based application.

```
$sudo apt-get install default-jdk default-jre -y
```

```
ahmad@vm1:~$ java -version
openjdk version "11.0.9.1" 2020-11-04
OpenJDK Runtime Environment (build 11.0.9.1+
OpenJDK 64-Bit Server VM (build 11.0.9.1+1-U
ahmad@vm1:~$
```

Create Hadoop User and Setup Passwordless SSH

- Add a new user and set its group

```
$ sudo adduser hadoop
$ sudo usermod -aG sudo hadoop
```

- Logout and login into the VM
- Login with hadoop user and generate an SSH key pair

```
$ sudo su - hadoop
$ ssh-keygen -t rsa
```

Create Hadoop User and Setup Passwordless SSH (Cont.)

```
ahmad@vm1:~$ sudo su - hadoop
[sudo] password for ahmad:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

hadoop@vm1:~$
hadoop@vm1:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Created directory '/home/hadoop/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:2E1eyIBm9RN3Cdjmnonk9R4rMu2o7JT0u8u8zfroYX4 hadoop@vm1
The key's randomart image is:
+---[RSA 3072]----+
|       oo .oo... |
|      +  +.+o..  |
|     o    *o.    |
|       o +.oo    |
|      . Soo+ +   |
|       . oo + o  |
|        o +. . o |
|       o =oBE o  |
|       .+o&@=.   |
+----[SHA256]-----+
hadoop@vm1:~$
```

Password-less SSH

```
$cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$chmod 0600 ~/.ssh/authorized_keys
```

```
hadoop@vm1:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
hadoop@vm1:~$ chmod 0600 ~/.ssh/authorized_keys
hadoop@vm1:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:hhwPaoAhQTlUqkUiXnPa77CCFUxG62fzu6zzJn
EVQGY.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
es
Warning: Permanently added 'localhost' (ECDSA) to the list of known ho
sts.
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)
```

Install Hadoop

```
$sudo su - Hadoop
$wget
https://downloads.apache.org/hadoop/common/hadoop-3.2.1/hadoop-3.2.1.tar.gz
```

```
hadoop@vm1:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3
.2.1/hadoop-3.2.1.tar.gz
--2020-11-21 08:45:28--  https://downloads.apache.org/hadoop/common/ha
doop-3.2.1/hadoop-3.2.1.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219,
 2a01:4f8:10a:201a::2
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 359196911 (343M) [application/x-gzip]
Saving to: 'hadoop-3.2.1.tar.gz'

hadoop-3.2.1.tar. 100%[===========>] 342.56M  2.40MB/s    in 2m 42s

2020-11-21 08:48:11 (2.11 MB/s) - 'hadoop-3.2.1.tar.gz' saved [3591969
11/359196911]

hadoop@vm1:~$
```

Install Hadoop (Cont.)

```
$tar -xvzf hadoop-3.2.1.tar.gz

$sudo mv hadoop-3.2.1 /usr/local/Hadoop

$sudo mkdir /usr/local/hadoop/logs

$sudo chown -R hadoop:hadoop /usr/local/hadoop
```

```
hadoop@vm1:~$ ls -lh /usr/local/hadoop/
total 208K
drwxr-xr-x 2 hadoop hadoop 4.0K Sep 10  2019 bin
drwxr-xr-x 3 hadoop hadoop 4.0K Sep 10  2019 etc
drwxr-xr-x 2 hadoop hadoop 4.0K Sep 10  2019 include
drwxr-xr-x 3 hadoop hadoop 4.0K Sep 10  2019 lib
drwxr-xr-x 4 hadoop hadoop 4.0K Sep 10  2019 libexec
-rw-rw-r-- 1 hadoop hadoop 148K Sep 10  2019 LICENSE.txt
drwxr-xr-x 2 hadoop hadoop 4.0K Nov 21 09:09 logs
-rw-rw-r-- 1 hadoop hadoop  22K Sep 10  2019 NOTICE.txt
-rw-rw-r-- 1 hadoop hadoop 1.4K Sep 10  2019 README.txt
drwxr-xr-x 3 hadoop hadoop 4.0K Sep 10  2019 sbin
drwxr-xr-x 4 hadoop hadoop 4.0K Sep 10  2019 share
hadoop@vm1:~$
```

Configuring Environmental Variables

```
$nano ~/.bashrc
```

Add the following lines:

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

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

Configuring Environmental Variables (Cont.)

```
hadoop@vm1:~$ vim ~/.bashrc
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$ source ~/.bashrc
hadoop@vm1:~$ echo $HADOOP_HOME
/usr/local/hadoop
hadoop@vm1:~$
```

Configure Hadoop

- We will learn how to setup Hadoop on a single node.
  - Configuring Hadoop for multiple nodes is your second homework assignment (good luck ☺).
- First, locate the correct Java path using the following command:

```
$which javac
```

  - You should see the following output:

```
/usr/bin/javac
```

- Next, find the OpenJDK directory with the following command:

```
$readlink -f /usr/bin/javac
```

  - You should see the following output:

```
/usr/lib/jvm/java-11-openjdk-amd64/bin/javac
```

Configure Hadoop (Cont.)

```
hadoop@vm1:~$ which javac
/usr/bin/javac
hadoop@vm1:~$ readlink -f /usr/bin/javac
/usr/lib/jvm/java-11-openjdk-amd64/bin/javac
hadoop@vm1:~$
hadoop@vm1:~$
```

Configure Hadoop (Cont.)

- Next, edit the hadoop-env.sh file and define the Java path:

```
$ sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

- Add the following lines:

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_CLASSPATH+=" $HADOOP_HOME/lib/*.jar"
```

- You also need to download the Javax activation file. You can download it with the following command:

```
$cd /usr/local/hadoop/lib
$sudo wget
https://jcenter.bintray.com/javax/activation/javax.activ
ation-api/1.2.0/javax.activation-api-1.2.0.jar
```

- You can now verify the Hadoop version using the following command:

```
$hadoop version
```

Configure Hadoop (Cont.)

```
hadoop@vm1:~$ sudo vim $HADOOP_HOME/etc/hadoop/hadoop-env.sh
[sudo] password for hadoop:
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$ cd /usr/local/hadoop/lib
hadoop@vm1:/usr/local/hadoop/lib$ sudo wget https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar
--2020-11-21 09:42:53--  https://jcenter.bintray.com/javax/activation/javax.activation-api/1.2.0/javax.activation-api-1.2.0.jar
Resolving jcenter.bintray.com (jcenter.bintray.com)... 52.43.200.1, 52.88.32.158, 35.161.162.245, ...
Connecting to jcenter.bintray.com (jcenter.bintray.com)|52.43.200.1|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 56674 (55K) [application/content-stream]
Saving to: 'javax.activation-api-1.2.0.jar'

javax.activation- 100%[===========>]  55.35K   102KB/s    in 0.5s

2020-11-21 09:42:58 (102 KB/s) - 'javax.activation-api-1.2.0.jar' saved [56674/56674]
```

Configure Hadoop (Cont.)

- Verifying Hadoop version

```
hadoop@vm1:/usr/local/hadoop/lib$ hadoop version
Hadoop 3.2.1
Source code repository https://gitbox.apache.org/repos/asf/hadoop.git
-r b3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled by rohithsharmaks on 2019-09-10T15:56Z
Compiled with protoc 2.5.0
From source with checksum 776eaf9eee9c0ffc370bcbc1888737
This command was run using /usr/local/hadoop/share/hadoop/common/hadoo
p-common-3.2.1.jar
hadoop@vm1:/usr/local/hadoop/lib$
```

Configure core-site.xml File

- Next, you will need to specify the URL for your NameNode. You can do it by editing core-site.xml file.

```
$sudo nano $HADOOP_HOME/etc/hadoop/core-site.xml
```

- Add the following lines:

```
<configuration>
 <property>
    <name>fs.default.name</name>
    <value>hdfs://0.0.0.0:9000</value>
    <description>The default file system URI</description>
 </property>
</configuration>
```

Configure core-site.xml File (Cont.)

```
<?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 impl
ied.
  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.default.name</name>
        <value>hdfs://0.0.0.0:9000</value>
        <description>The default file system URI</description>
</property>

</configuration>
```

Configure hdfs-site.xml File

- Next, you will need to define the location for storing node metadata, fsimage file, and edit log file.
- You can do it by editing hdfs-site.xml file.
- First, create a directory for storing node metadata.

```
$sudo mkdir -p /home/hadoop/hdfs/{namenode,datanode}
$sudo chown -R hadoop:hadoop /home/hadoop/hdfs
```

- Next, edit the hdfs-site.xml file and define the location of the directory.

```
$sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

Configure hdfs-site.xml File (Cont.)

- Add the following lines:

```
<configuration>
     <property>
          <name>dfs.replication</name>
          <value>1</value>
     </property>
     <property>
          <name>dfs.name.dir</name>
          <value>file:///home/hadoop/hdfs/namenode</value>
     </property>
     <property>
          <name>dfs.data.dir</name>
          <value>file:///home/hadoop/hdfs/datanode</value>
     </property>
</configuration>
```

Configure hdfs-site.xml File (Cont.)

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

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

   <property>
      <name>dfs.name.dir</name>
      <value>file:///home/hadoop/hdfs/namenode</value>
   </property>

   <property>
      <name>dfs.data.dir</name>
      <value>file:///home/hadoop/hdfs/datanode</value>
   </property>
</configuration>
```

Configure mapred-site.xml File

- Next, you will need to define MapReduce values. You can define it by editing mapred-site.xml file.

```
sudo nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

- Add the following lines:

```
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

Configure mapred-site.xml File (Cont.)

```
<?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 impl
ied.
  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>

</configuration>
```

Configure yarn-site.xml File

- Next, you will need to edit the yarn-site.xml file and define YARN related settings.

```
$sudo nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

- Add the following lines:

```
<configuration>
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
</configuration>
```

Configure yarn-site.xml File (Cont.)

```
<?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 impl
ied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->

<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>

</configuration>
```

Format HDFS NameNode

- Next, you will need to validate the Hadoop configuration and format the HDFS NameNode.
- First, log in with Hadoop user and format the HDFS NameNode with the following command.

```
$su - hadoop
$hdfs namenode -format
```

```
2020-11-21 10:50:10,000 INFO namenode.FSImage: FSImageSaver clean chec
kpoint: txid=0 when meet shutdown.
2020-11-21 10:50:10,002 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at vm1/127.0.0.1
************************************************************/
hadoop@vm1:~$
hadoop@vm1:~$ hdfs namenode -format
```

Start the Hadoop Cluster

- First, start the NameNode and DataNode with the following command:

```
$start-dfs.sh
```

- You should get the following output:

```
Starting namenodes on [0.0.0.0]
Starting datanodes
Starting secondary namenodes [ubuntu2004]
```

Start the Hadoop Cluster (Cont.)

- Next, start the YARN resource and nodemanagers by running the following command.

```
$start-yarn.sh
```

- You should get the following output:

```
Starting resourcemanager
Starting nodemanagers
```

Start the Hadoop Cluster (Cont.)

- What is the expexted result of the following command?

```
$jps
```

Start the Hadoop Cluster (Cont.)

- What is the result of the following command?

```
$jps
```

```
5047 NameNode
5850 Jps
5326 SecondaryNameNode
5151 DataNode
```

Start the Hadoop Cluster (Cont.)

```
[hadoop@vm1:~$                                                                  ]
[hadoop@vm1:~$ start-dfs.sh                                                     ]
Starting namenodes on [0.0.0.0]
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of k
nown hosts.
Starting datanodes
Starting secondary namenodes [vm1]
vm1: Warning: Permanently added 'vm1' (ECDSA) to the list of known hos
ts.
[hadoop@vm1:~$ start-yarn.sh                                                    ]
Starting resourcemanager
Starting nodemanagers
[hadoop@vm1:~$                                                                  ]
```

Hadoop Installation
Part 2

Course Logistics

- Contact TA assigned to your group
- https://docs.google.com/spreadsheets/d/1kTXpmvETSXBOeNb2Crh7D0SNCK5yFDKYE-nZ3hBJUKs/edit?usp=sharing
- Select a time slot in the course website and present HW1 to the assigned TA
  - Good luck ☺
- This should be done by 99/09/14

Facing an Error

```
[hadoop@vm1:~/hdfs$ jps
 9755 Jps
[hadoop@vm1:~/hdfs$
```

HDFS and Yarn are not started

```
$cat logs/hadoop-hadoop-datanode-vm1.log
```

```
…
ERROR org.apache.hadoop.hdfs.server.datanode.DataNode:
Exception in secureMain
java.lang.ExceptionInInitializerError

…
```

We solve this problem in our next session.

Logs Provide Hints

```
        at org.apache.hadoop.http.HttpServer2.start(HttpServer2.java:1155)
        at org.apache.hadoop.hdfs.server.namenode.NameNodeHttpServer.start
(NameNodeHttpServer.java:181)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.startHttpServer
(NameNode.java:885)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(Name
Node.java:707)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode
.java:953)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode
.java:926)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(
NameNode.java:1692)
        at org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.j
ava:1759)
Caused by: java.lang.IllegalArgumentException: Invalid Java version 11.0.9
.1
        at org.eclipse.jetty.util.JavaVersion.parseJDK9(JavaVersion.java:7
1)
        at org.eclipse.jetty.util.JavaVersion.parse(JavaVersion.java:49)
        at org.eclipse.jetty.util.JavaVersion.<clinit>(JavaVersion.java:43
)
        ... 23 more
2020-11-22 18:36:09,392 INFO org.apache.hadoop.util.ExitUtil: Exiting with
 status 1: java.lang.ExceptionInInitializerError
2020-11-22 18:36:09,418 INFO org.apache.hadoop.hdfs.server.namenode.NameNo
de: SHUTDOWN_MSG:
```

Solution for the Error

- Hadoop 3.2 is not compatible with Java11
- Let's revert back to Java 8

```
$sudo apt-get remove --purge icedtea-* openjdk-*
```

```
sudo apt install openjdk-8-jdk
```

- I can now start HDFS and Yarn
- Let's review the configuration files once again

/etc/hosts

```
sudo vim /etc/hosts
```

```
#127.0.0.1 localhost
192.168.8.104 vm1
#127.0.1.1 vm1
# The following lines are desirable for IPv6 capable hosts
#::1     ip6-localhost ip6-loopback
#fe00::0 ip6-localnet
#ff00::0 ip6-mcastprefix
#ff02::1 ip6-allnodes
#ff02::2 ip6-allrouters
```

$HADOOP_HOME/etc/hadoop/hadoop-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_CLASSPATH+=" $HADOOP_HOME/lib/*.jar"
```

$HADOOP_HOME/etc/hadoop/core-site.xml

```
<!-- Put site-specific property overrides in this file. -->
<configuration>
<property>
        <name>fs.defaultFS</name>
        <value>hdfs://vm1:9000</value>
        <description>The default file system URI</description>
</property>
<property>
        <name>hadoop.tmp.dir</name>
        <value>/home/hadoop/hadooptmpdata</value>
</property>

</configuration>
```

$HADOOP_HOME/etc/hadoop/hdfs-site.xml

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

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

   <property>
      <name>dfs.name.dir</name>
      <value>file:///home/hadoop/hdfs/namenode</value>
   </property>

   <property>
      <name>dfs.data.dir</name>
      <value>file:///home/hadoop/hdfs/datanode</value>
   </property>
<property>
        <name>dfs.permissions</name>
        <value>false</value>
</property>


</configuration>
```

$HADOOP_HOME/etc/hadoop/mapred-site.xml

```
<configuration>
<property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
</property>

</configuration>
```

$HADOOP_HOME/etc/hadoop/yarn-site.xml

```
-->
<configuration>

<!-- Site specific YARN configuration properties -->

<property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
</property>

</configuration>
```

$HADOOP_HOME/etc/hadoop/workers

Starting the Cluster

```
[hadoop@vm1:~$
[hadoop@vm1:~$ rm -rf hdfs/namenode/*
[hadoop@vm1:~$ rm -rf hdfs/datanode/*
 hadoop@vm1:~$ hdfs namenode -format
```

```
[hadoop@vm1:~$ start-dfs.sh
 Starting namenodes on [vm1]
 Starting datanodes
 Starting secondary namenodes [vm1]
[hadoop@vm1:~$ jps
```

Starting the Cluster (Cont.)

```
hadoop@vm1:~$ start-dfs.sh
Starting namenodes on [vm1]
Starting datanodes
Starting secondary namenodes [vm1]
hadoop@vm1:~$ jps
```

```
hadoop@vm1:~$ jps
3441 SecondaryNameNode
3266 DataNode
3555 Jps
3124 NameNode
```

Starting the Cluster (Cont.)

```
[hadoop@vm1:~$ start-yarn.sh
 Starting resourcemanager
 Starting nodemanagers
[hadoop@vm1:~$ jps
```

Starting the Cluster (Cont.)

```
[hadoop@vm1:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
[hadoop@vm1:~$ jps
[hadoop@vm1:~$ jps
3441 SecondaryNameNode
3266 DataNode
3907 NodeManager
3124 NameNode
3974 Jps
3755 ResourceManager
```

Hadoop NameNode Interface

Overview 'vm1:9000' (active)

Started:	Mon Nov 23 11:54:39 +0330 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 20:26:00 +0430 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-a4eb41fb-f575-45fe-9032-c2ad5fce380d
Block Pool ID:	BP-1211547759-192.168.8.104-1606119831429

Hadoop NameNode Interface (Cont.)

Overview 'vm1:9000' (active)

Started:	Mon Nov 23 11:54:39 +0330 2020
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 20:26:00 +0430 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-a4eb41fb-f575-45fe-9032-c2ad5fce380d
Block Pool ID:	BP-1211547759-192.168.8.104-1606119831429

Individual DataNodes URL

DataNode on vm1:9866

Cluster ID:	CID-a4eb41fb-f575-45fe-9032-c2ad5fce380d
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842

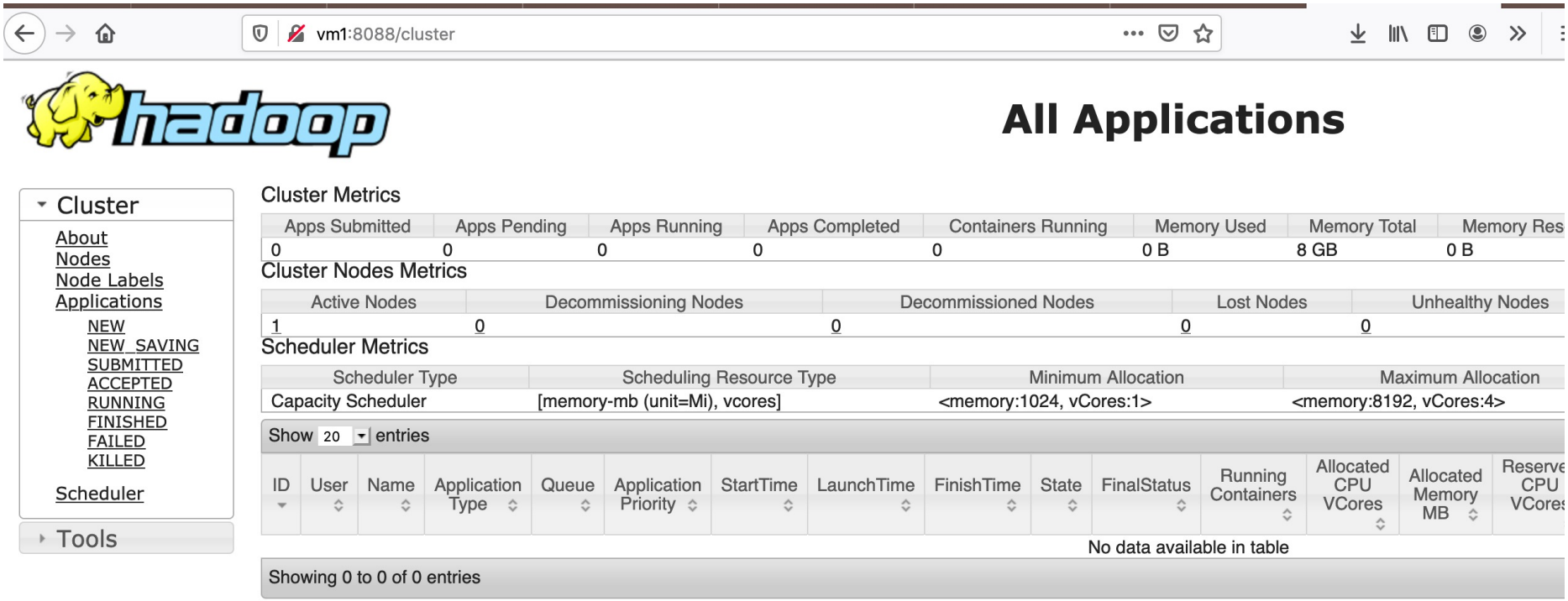
Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
vm1:9000	BP-1211547759-192.168.8.104-1606119831429	RUNNING	0s	44 minutes	0 B (64 MB)

Volume Information

Directory	StorageType	Capacity Used	Capacity Left	Capacity Reserved	Reserved Space for Replicas	Blocks
/home/hadoop/hdfs/datanode	DISK	28 KB	11.06 GB	0 B	0 B	0

YARN Resource Manager URL

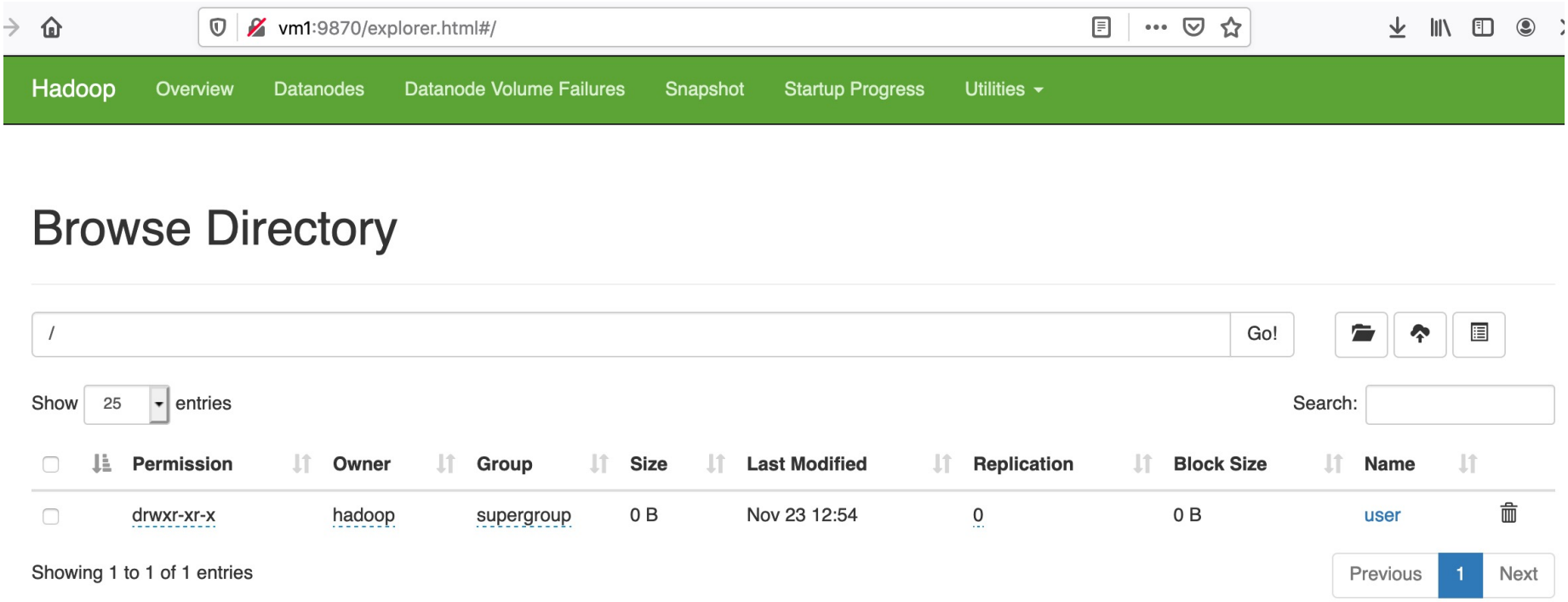


HDFS Terminal Interface

```
hadoop@vm1:~$ hdfs dfs -ls /
hadoop@vm1:~$ hdfs dfs -mkdir /user
hadoop@vm1:~$ hdfs dfs -mkdir /user/hadoop
hadoop@vm1:~$ hdfs dfs -ls /
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2020-11-23 09:24 /user
hadoop@vm1:~$
hadoop@vm1:~$ hdfs dfs -ls /user
Found 1 items
drwxr-xr-x   - hadoop supergroup          0 2020-11-23 09:24 /user/hadoop
hadoop@vm1:~$
```

https://www.geeksforgeeks.org/hdfs-commands/

Let’s Check the Web Interface



Upload a File to HDFS

```
hadoop@vm1:~$ vim test.txt
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$ cat test.txt
hi joe
hello world
bye trump
hadoop@vm1:~$ hdfs dfs -copyFromLocal test.txt /user/hadoop/
2020-11-23 09:33:41,687 INFO sasl.SaslDataTransferClient: SAS
L encryption trust check: localHostTrusted = false, remoteHos
tTrusted = false
hadoop@vm1:~$ hdfs dfs -ls /user/hadoop
Found 1 items
-rw-r--r--   1 hadoop supergroup         29 2020-11-23 09:33
/user/hadoop/test.txt
hadoop@vm1:~$
```

https://www.geeksforgeeks.org/hdfs-commands/

Let's Run WordCount Job

- Follow the following guide

  https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

- You will learn how to run MapReduce job
- You will do more in your homework

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;
```

WordCount.java (Cont.)

```
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);
      }
    }
  }
```

WordCount.java (Cont.)

```
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 {
    int sum = 0;
    for (IntWritable val : values) {
      sum += val.get();
    }
    result.set(sum);
    context.write(key, result);
  }
}
```

WordCount.java (Cont.)

```
  public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

Environment Variables

Assuming environment variables are set as follows:

```
export JAVA_HOME=/usr/java/default
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/tools.jar
```

Compile `WordCount.java` and create a jar:

```
$ bin/hadoop com.sun.tools.javac.Main WordCount.java
$ jar cf wc.jar WordCount*.class
```

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_CLASSPATH+=" $HADOOP_HOME/lib/*.jar"
export HADOOP_CLASSPATH+=" ${JAVA_HOME}/lib/tools.jar"
```

hadoop-env.sh

First Run Error

```
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$ hadoop jar wc.jar WordCount /user/hadoop/test.txt
 /user/hadoop/output
Last 4096 bytes of stderr :
Error: Could not find or load main class org.apache.hadoop.mapr
educe.v2.app.MRAppMaster

Please check whether your etc/hadoop/mapred-site.xml contains t
he below configuration:
<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=${full path of your hadoop distribu
tion directory}</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=${full path of your hadoop distribu
tion directory}</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=${full path of your hadoop distribu
tion directory}</value>
</property>
```

Just Do What is Asked

```
<configuration>
<property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
</property>

<property>
  <name>yarn.app.mapreduce.am.env</name>
  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
</property>
<property>
  <name>mapreduce.map.env</name>
  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
</property>
<property>
  <name>mapreduce.reduce.env</name>
  <value>HADOOP_MAPRED_HOME=/usr/local/hadoop</value>
</property>

</configuration>
```

WordCount Submission

```
hadoop@vm1:~$
hadoop@vm1:~$
hadoop@vm1:~$ hadoop jar wc.jar WordCount /user/hadoop/test.txt
 /user/hadoop/output
2020-11-23 10:20:32,662 INFO mapreduce.Job:  map 0% reduce 0%
2020-11-23 10:20:45,117 INFO mapreduce.Job:  map 100% reduce 0%
2020-11-23 10:21:04,349 INFO mapreduce.Job:  map 100% reduce 100
%
2020-11-23 10:21:08,416 INFO mapreduce.Job: Job job_160612578174
5_0002 completed successfully
2020-11-23 10:21:08,745 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=71
                FILE: Number of bytes written=451669
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=130
                HDFS: Number of bytes written=41
                HDFS: Number of read operations=8
```

WordCount Submission (Cont.)

```
[hadoop@vm1:~$ hdfs dfs -ls /user/hadoop/output
Found 2 items
-rw-r--r--   1 hadoop supergroup          0 2020-11-23 10:21 /us
er/hadoop/output/_SUCCESS
-rw-r--r--   1 hadoop supergroup         41 2020-11-23 10:21 /us
er/hadoop/output/part-r-00000
[hadoop@vm1:~$
[hadoop@vm1:~$
[hadoop@vm1:~$ hdfs dfs -cat /user/hadoop/output/part-r-00000
2020-11-23 10:50:23,776 INFO sasl.SaslDataTransferClient: SASL e
ncryption trust check: localHostTrusted = false, remoteHostTrust
ed = false
bye     1
hello   1
hi      1
joe     1
trump   1
world   1
[hadoop@vm1:~$
```

Homework Assignment 2

- You will install Hadoop Yarn on 3 VMs
  - 1 node for Resource Manager and NameNode
  - 2 nodes for Node Manager and DataNode
- You will implement example MapReduce programs
  - Stay tuned for the HW explanation
- Ask questions in the course webpage about Hadoop Yarn installation on multiple nodes