KISORE R G
210701124

EXP NO:1 Downloading and installing Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.

AIM:

To Download and install Hadoop, Understanding different Hadoop modes, Startup scripts, Configuration files.

PROCEDURE:

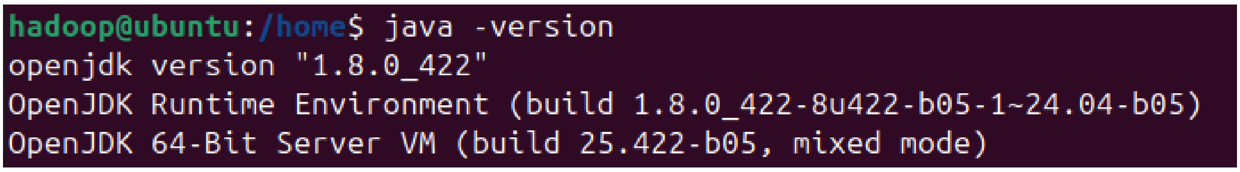
Step 1 : Install Java Development Kit
The default Ubuntu repositories contain Java 8 and Java 11 both. But, Install Java 8 because hive only works on this version. Use the following command to install it.

$sudo apt update&&sudo apt install openjdk-8-jdk

Step 2 : Verify the Java version
Once installed, verify the installed version of Java with the following command:

$ java -version

Output:



```
hadoop@ubuntu:/home$ java -version
openjdk version "1.8.0_422"
OpenJDK Runtime Environment (build 1.8.0_422-8u422-b05-1~24.04-b05)
OpenJDK 64-Bit Server VM (build 25.422-b05, mixed mode)
```

Step 3: Install SSH
SSH (Secure Shell) installation is vital for Hadoop as it enables secure communication between nodes in the Hadoop cluster. This ensures data integrity, confidentiality, and allows for efficient distributed processing of data across the cluster.

$sudo apt install ssh

Step 4 : Create the hadoop user :
All the Hadoop components will run as the user that you create for Apache Hadoop, and the user will also be used for logging in to Hadoop's web interface. Run the command to create user and set password.

$ sudo adduser hadoop

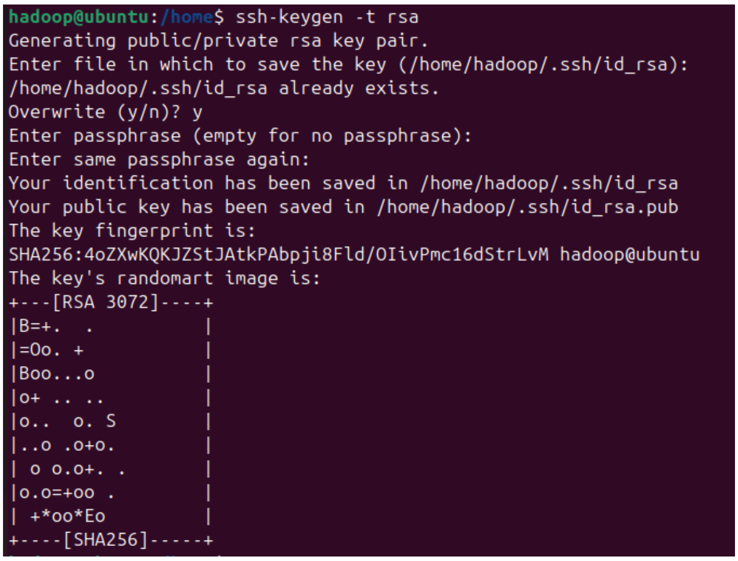
Step 5 : Switch user

Switch to the newly created hadoop user:

$ su - hadoop

Step 6 :
Configure SSH Now configure password-less SSH access for the newly created hadoop user, so didn’t enter the key to save file and passphrase. Generate an SSH keypair (generate Public and Private Key Pairs)first

$ssh-keygen -t rsa

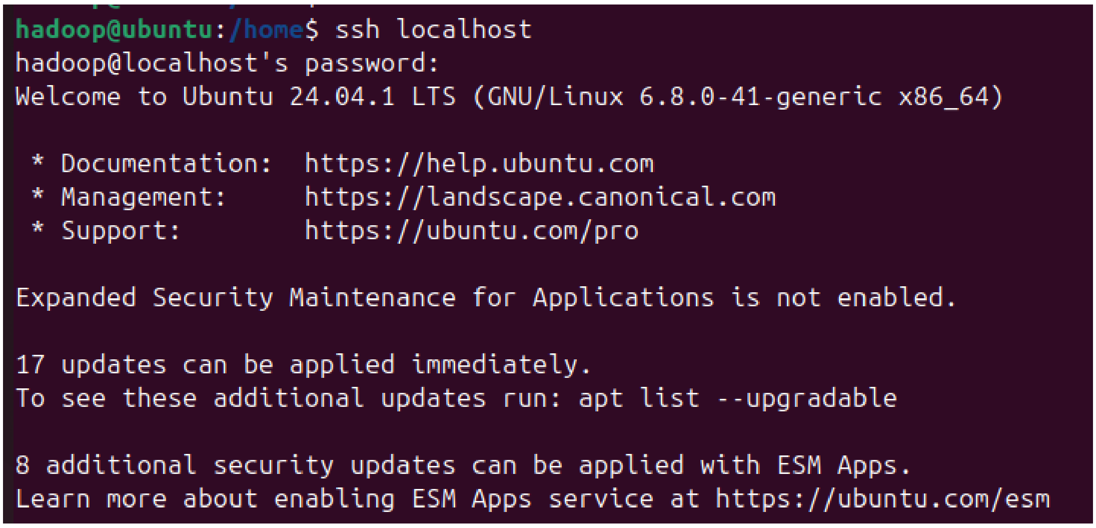


```
hadoop@ubuntu:/home$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
/home/hadoop/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has 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:
SHA256:4oZXwKQKJZStJAtkPAbpji8Fld/OIivPmc16dStrLvM hadoop@ubuntu
The key's randomart image is:
+---[RSA 3072]----+
|B=+.  .          |
|=Oo. +           |
|Boo...o          |
|o+ .. ..         |
|o..  o. S        |
|..o .o+o.        |
| o o.o+. .       |
|o.o=+oo .        |
| +*oo*Eo         |
+----[SHA256]-----+
```

Step 7 : Set permissions :
Next, append the generated public keys from id_rsa.pub to authorized_keys and set proper permission:
$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$ chmod 640 ~/.ssh/authorized_keys

Step 8 : SSH to the localhost
Next, verify the password less SSH authentication with the following command:

$ ssh localhost



```
hadoop@ubuntu:/home$ ssh localhost
hadoop@localhost's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-41-generic x86_64)

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

Expanded Security Maintenance for Applications is not enabled.

17 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

8 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
```

Step 9 : Switch user
Again switch to hadoop. So, First, change the user to hadoop with the following command:

$ su–hadoop

Step 10 : Install hadoop
Next, download the latest version of Hadoop using the wget command:

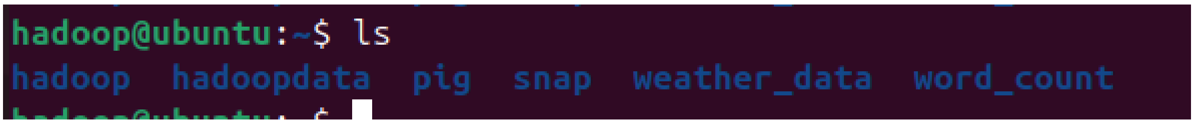
$ wgethttps://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz

Once downloaded, extract the downloaded file:

$ tar -xvzf hadoop-3.3.6.tar.gz

Next, rename the extracted directory to hadoop:

$ mv hadoop-3.3.6 hadoop

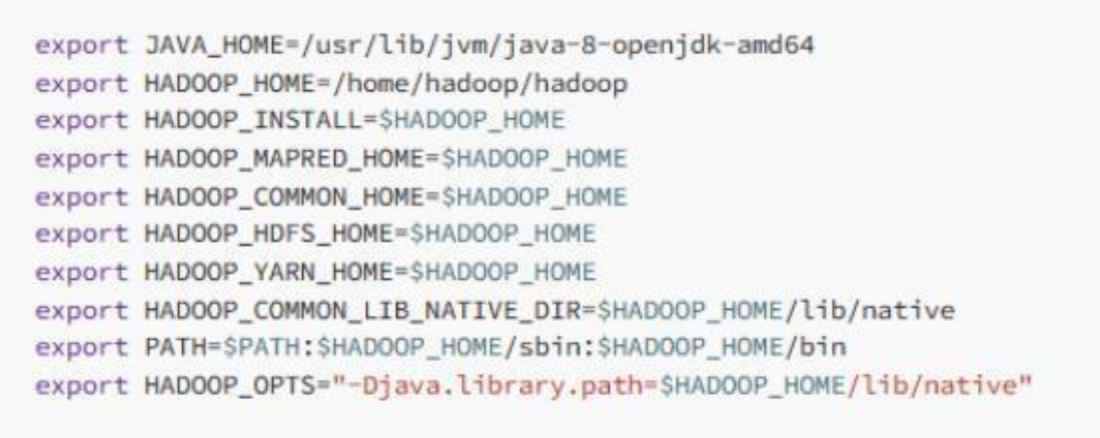


```
hadoop@ubuntu:~$ ls
hadoop  hadoopdata  pig  snap  weather_data  word_count
```

Open the ~/.bashrc file in your favorite text editor:

$ nano ~/.bashrc

Append the below lines to file.



```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_HOME=/home/hadoop/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 HADOOP_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"
```

Save and close the file. Then, activate the environment variables with the following command:

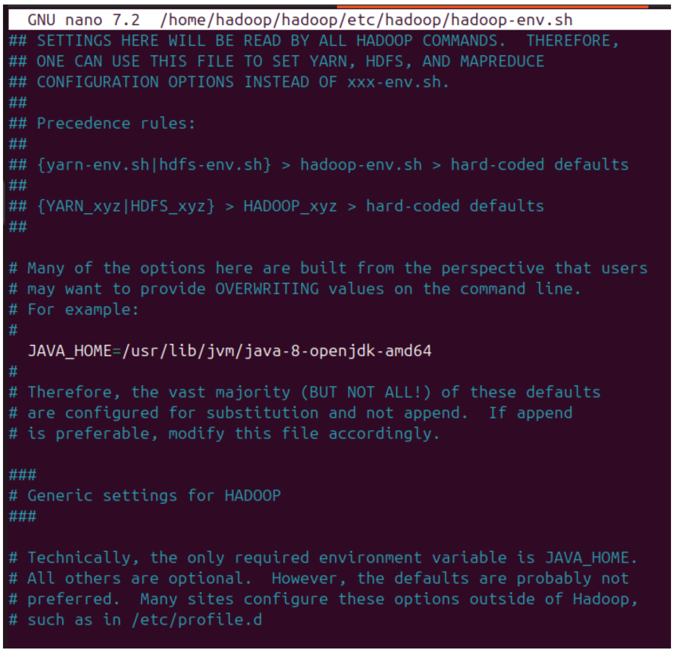
s$ source ~/.bashrc

Next, open the Hadoop environment variable file:

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

Search for the “export JAVA_HOME” and configure it.

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



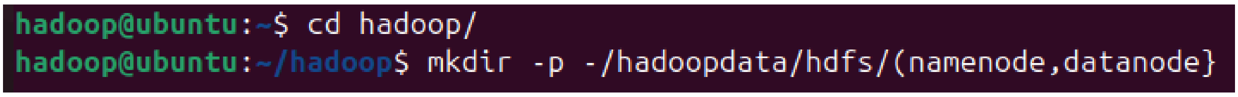
```
  GNU nano 7.2   /home/hadoop/hadoop/etc/hadoop/hadoop-env.sh
## SETTINGS HERE WILL BE READ BY ALL HADOOP COMMANDS.  THEREFORE,
## ONE CAN USE THIS FILE TO SET YARN, HDFS, AND MAPREDUCE
## CONFIGURATION OPTIONS INSTEAD OF xxx-env.sh.
##
## Precedence rules:
##
## {yarn-env.sh|hdfs-env.sh} > hadoop-env.sh > hard-coded defaults
##
## {YARN_xyz|HDFS_xyz} > HADOOP_xyz > hard-coded defaults
##

# Many of the options here are built from the perspective that users
# may want to provide OVERWRITING values on the command line.
# For example:
#
  JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
#
# Therefore, the vast majority (BUT NOT ALL!) of these defaults
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.

###
# 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
```

Step 11 : Configuring Hadoop :

First, you will need to create the namenode and datanode directories inside the Hadoop user home directory. Run the following command to create both directories:

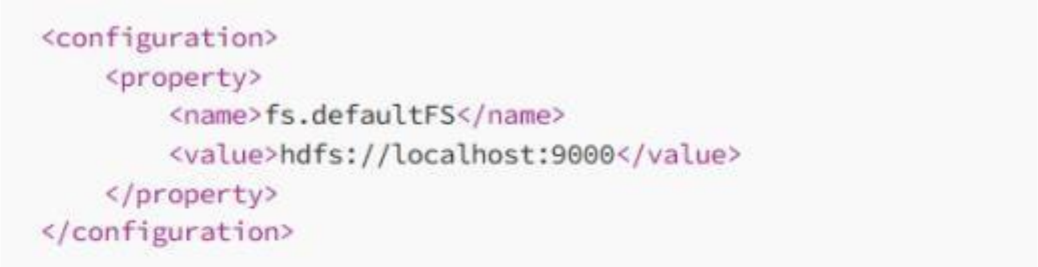
$ cd hadoop/

$mkdir -p ~/hadoopdata/hdfs/{namenode,datanode}



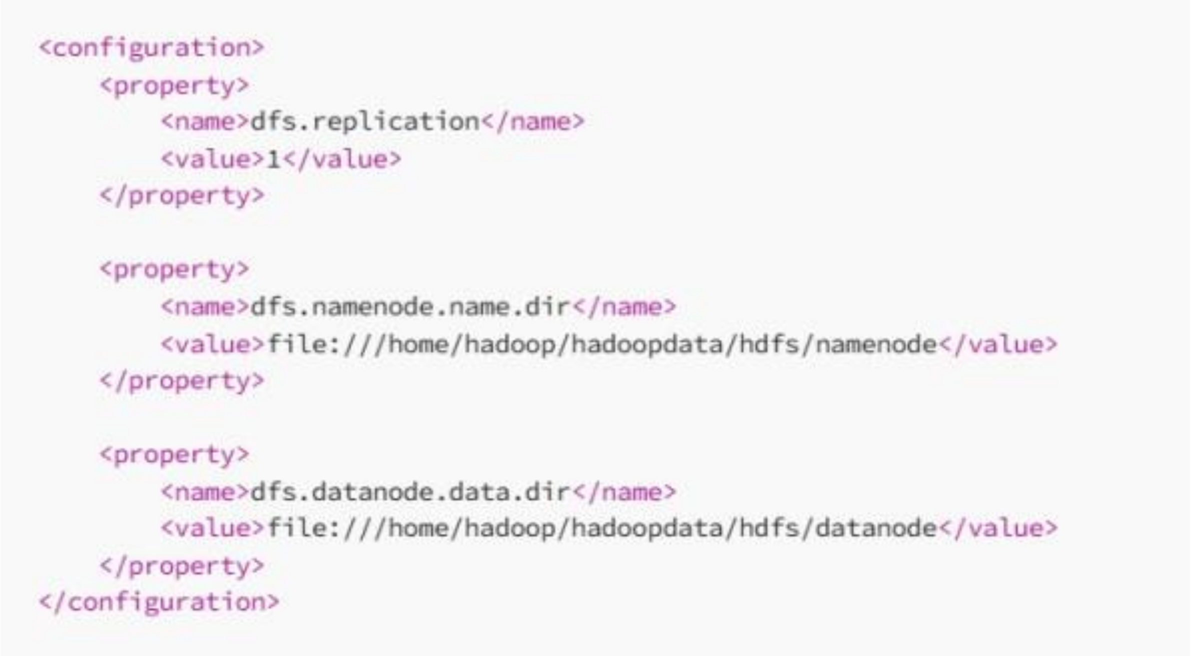
```
hadoop@ubuntu:~$ cd hadoop/
hadoop@ubuntu:~/hadoop$ mkdir -p -/hadoopdata/hdfs/(namenode,datanode}
```

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



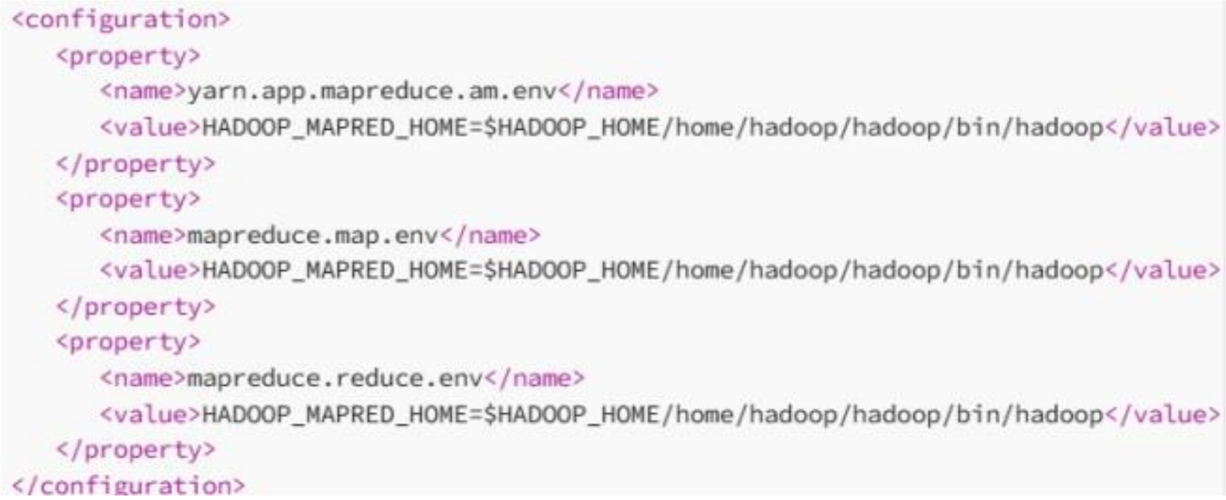
```
<configuration>
    <property>
        <name>fs.defaultFS</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

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



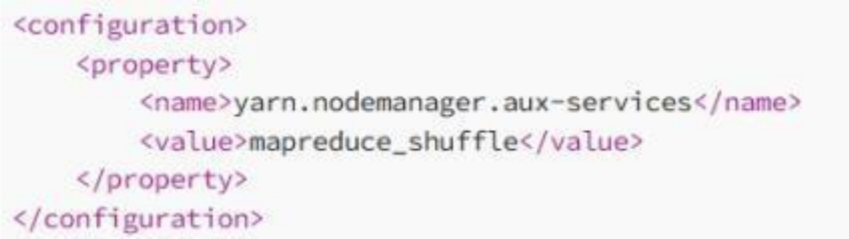
```
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>

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

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

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



```
<configuration>
   <property>
      <name>yarn.app.mapreduce.am.env</name>
      <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
   </property>
   <property>
      <name>mapreduce.map.env</name>
      <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
   </property>
   <property>
      <name>mapreduce.reduce.env</name>
      <value>HADOOP_MAPRED_HOME=$HADOOP_HOME/home/hadoop/hadoop/bin/hadoop</value>
   </property>
</configuration>
```

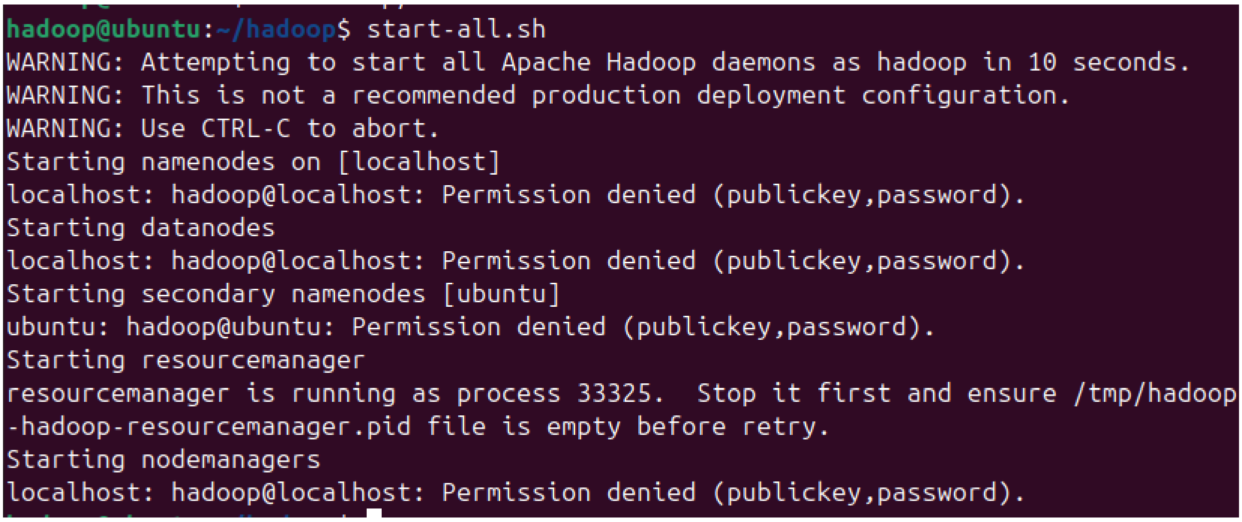
$nano $HADOOP_HOME/etc/hadoop/yarn-site.xml



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

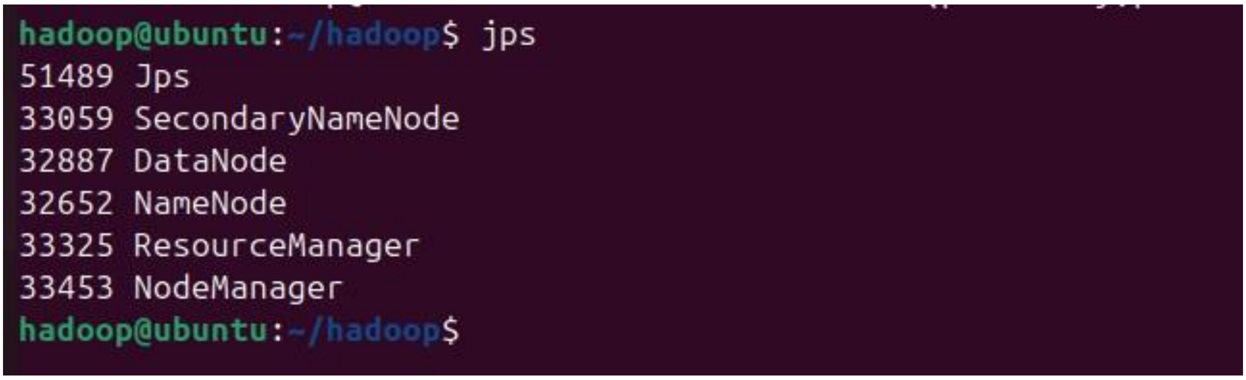
Step 12 – Start Hadoop Cluster:
Run the following command to format the Hadoop Namenode:
$hdfs namenode –format

Then start the Hadoop cluster with the following command.
$ start-all.sh



```
hadoop@ubuntu:~/hadoop$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
localhost: hadoop@localhost: Permission denied (publickey,password).
Starting datanodes
localhost: hadoop@localhost: Permission denied (publickey,password).
Starting secondary namenodes [ubuntu]
ubuntu: hadoop@ubuntu: Permission denied (publickey,password).
Starting resourcemanager
resourcemanager is running as process 33325.  Stop it first and ensure /tmp/hadoop
-hadoop-resourcemanager.pid file is empty before retry.
Starting nodemanagers
localhost: hadoop@localhost: Permission denied (publickey,password).
```

$ jps

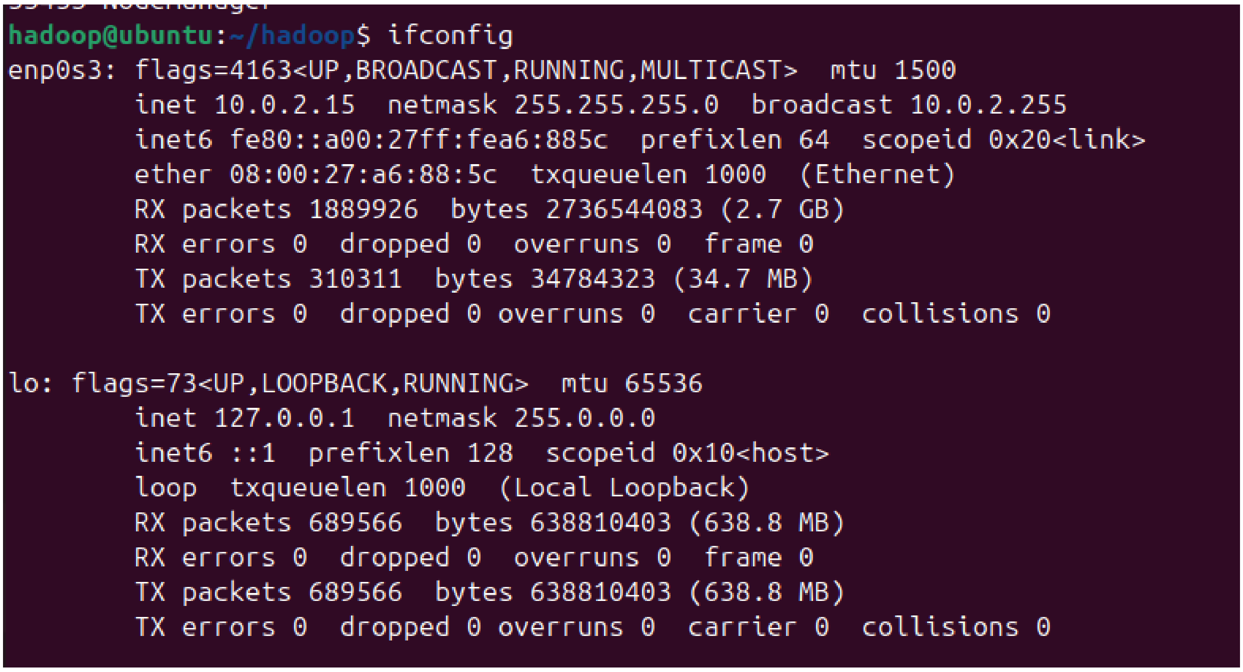


```
hadoop@ubuntu:~/hadoop$ jps
51489 Jps
33059 SecondaryNameNode
32887 DataNode
32652 NameNode
33325 ResourceManager
33453 NodeManager
hadoop@ubuntu:~/hadoop$
```

Step 13 – Access Hadoop Namenode and Resource Manager
If you installing net-tools for the first time switch to default user:

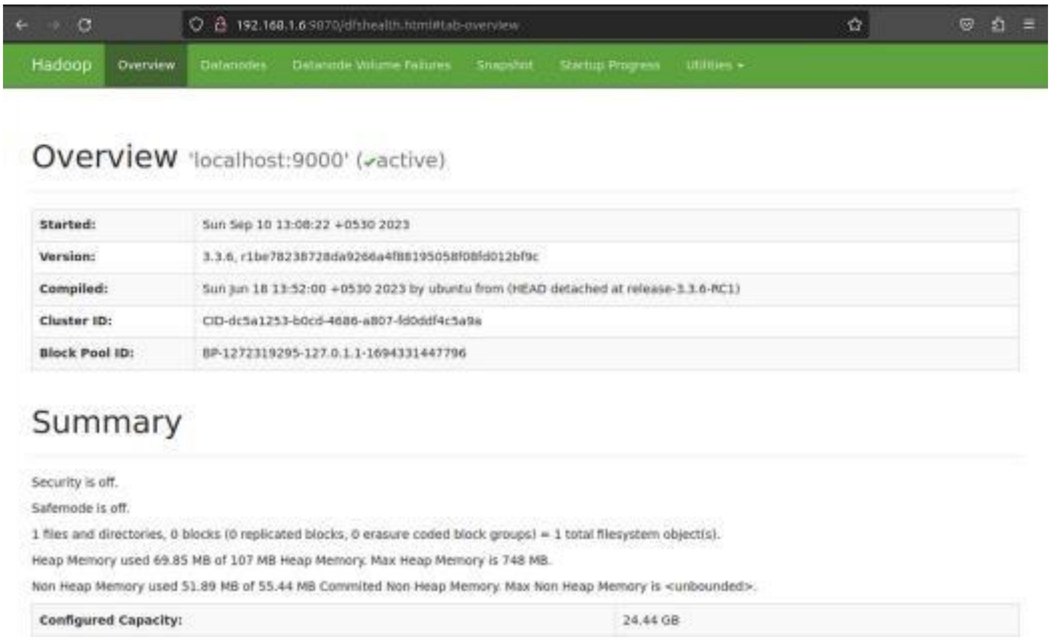
$sudo apt install net-tools

Then run ifconfig command to know our ip address: **ifconfig**



```
hadoop@ubuntu:~/hadoop$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fea6:885c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a6:88:5c  txqueuelen 1000  (Ethernet)
        RX packets 1889926  bytes 2736544083 (2.7 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 310311  bytes 34784323 (34.7 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 689566  bytes 638810403 (638.8 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 689566  bytes 638810403 (638.8 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

To access the Namenode, open your web browser and visit the URL http://your-serverip:9870.
You should see the following screen: http://192.168.1.6:9870



Overview 'localhost:9000' (✓active)

Started:	Sun Sep 10 13:08:22 +0530 2023
Version:	3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c
Compiled:	Sun Jun 18 13:52:00 +0530 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1)
Cluster ID:	CID-dc5a1253-b0cd-4686-a807-fd0ddf4c5a9a
Block Pool ID:	BP-1272319295-127.0.1.1-1694331447796

Summary

Security is off.

Safemode is off.

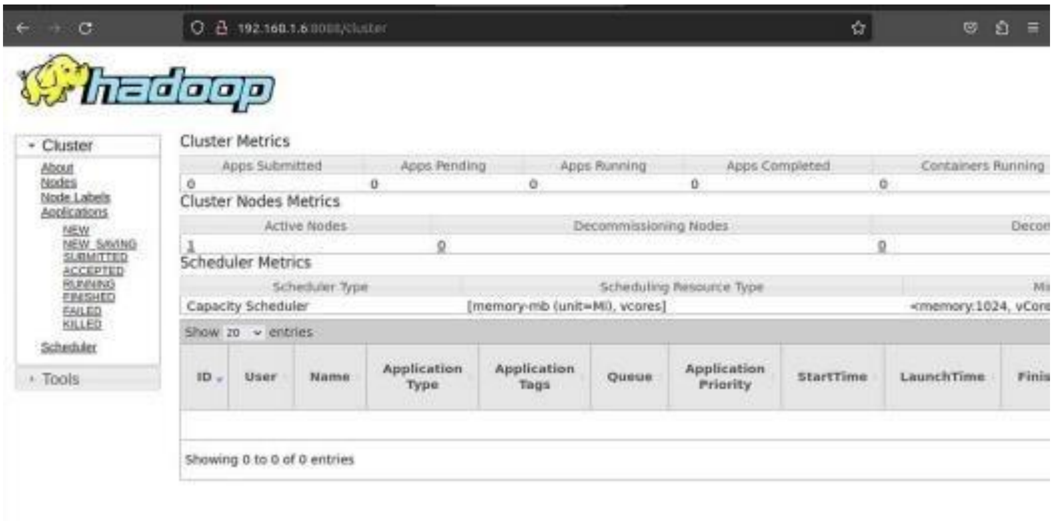
1 files and directories, 0 blocks (0 replicated blocks, 0 erasure coded block groups) = 1 total filesystem object(s).

Heap Memory used 69.85 MB of 107 MB Heap Memory. Max Heap Memory is 748 MB.

Non Heap Memory used 51.89 MB of 55.44 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	24.44 GB

To access Resource Manage, open your web browser and visit the URL http://your-serverip:8088. You should see the following screen: http://192.168.16:8088



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decom
1	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Mi
Capacity Scheduler	[memory-mb (unit=Mi), vcores]	<memory:1024, vCore

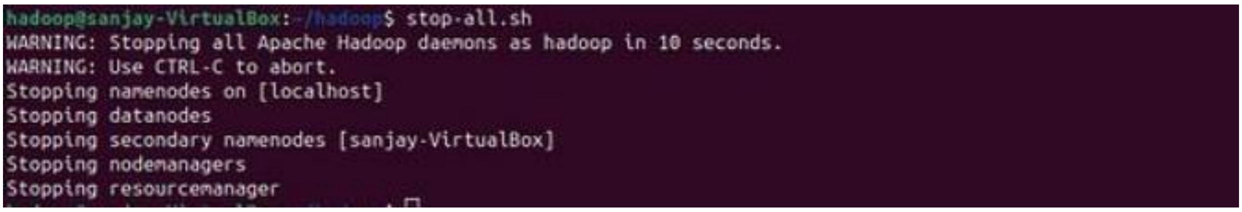
ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	Finis

Showing 0 to 0 of 0 entries

Step 14 – Stop Hadoop Cluster

To stop the Hadoop all services, run the following command:

$ stop-all.sh



```
hadoop@sanjay-VirtualBox:~/hadoop$ stop-all.sh
WARNING: Stopping all Apache Hadoop daemons as hadoop in 10 seconds.
WARNING: Use CTRL-C to abort.
Stopping namenodes on [localhost]
Stopping datanodes
Stopping secondary namenodes [sanjay-VirtualBox]
Stopping nodemanagers
Stopping resourcemanager
```

RESULT:

The step-by-step installation and configuration of Hadoop on Ubutu linux system have been successfully completed.