Hadoop Installation Guide

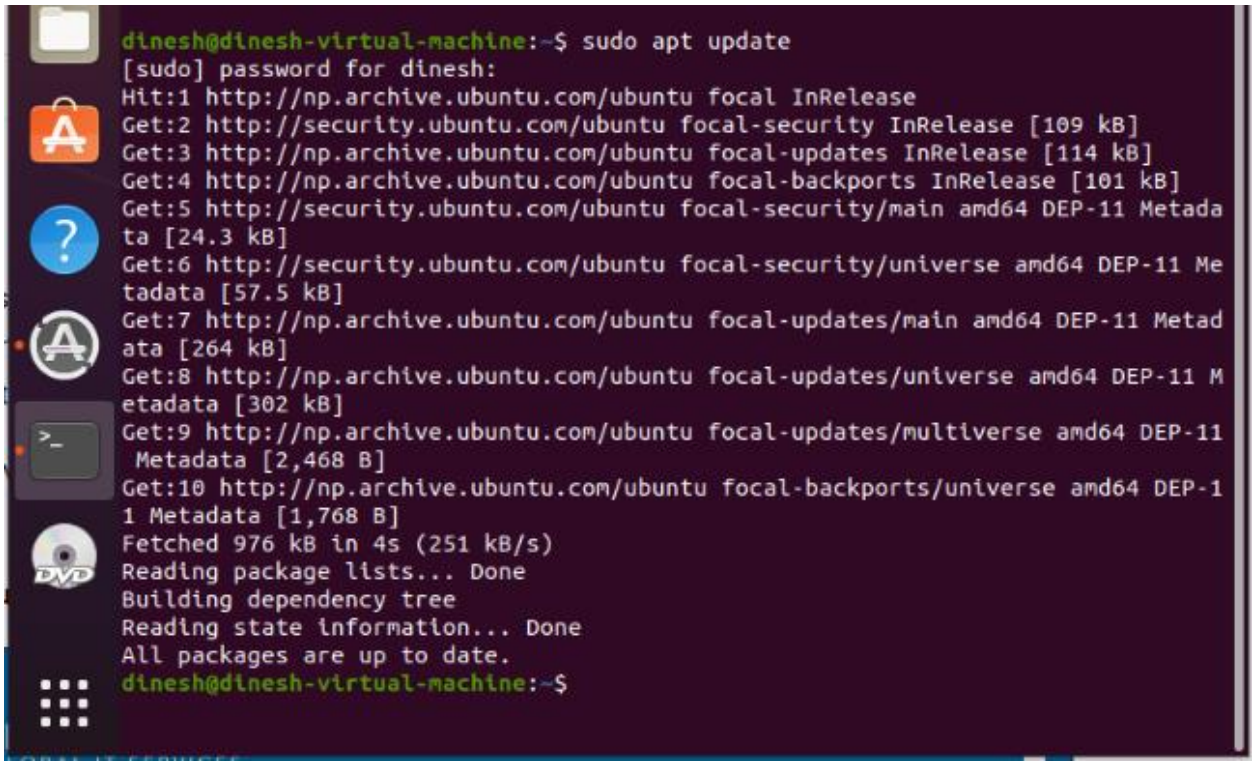
Prepared By: Jnaneshwar Bohara

Table of Contents

1. Java installation

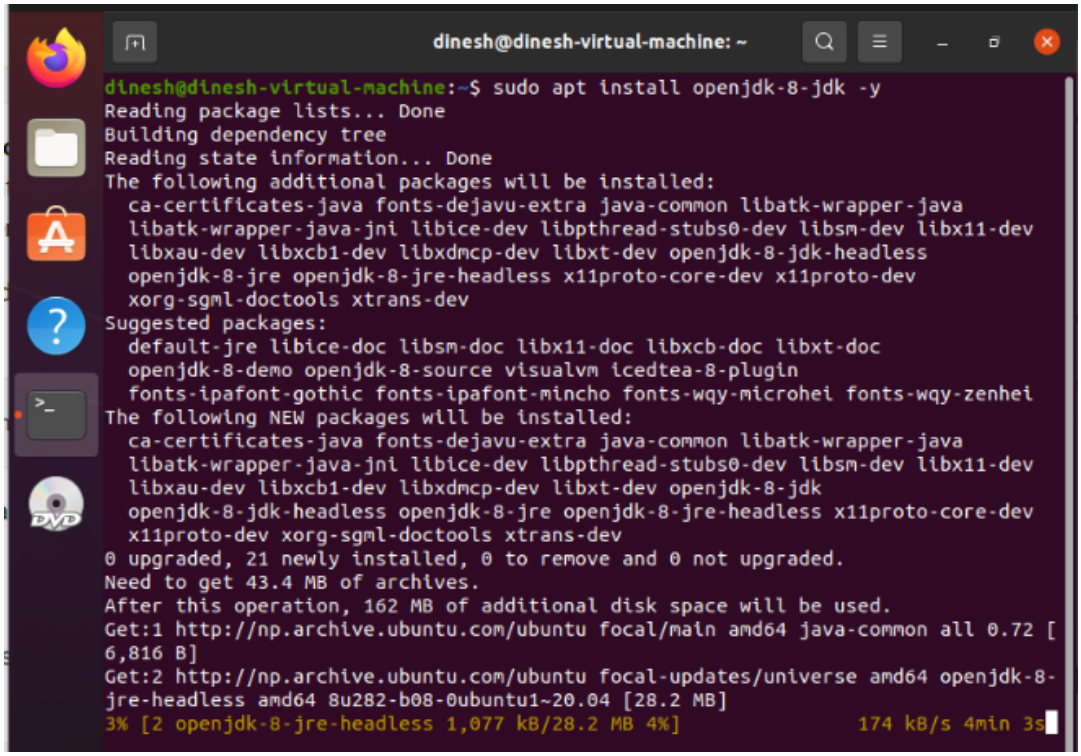
1.1. The Hadoop framework is written in Java, and its services require a compatible Java Runtime Environment (JRE) and Java Development Kit (JDK). Use the following command to update your system before initiating a new installation. First thing you want to do is to install updates.

```
sudo apt update
```

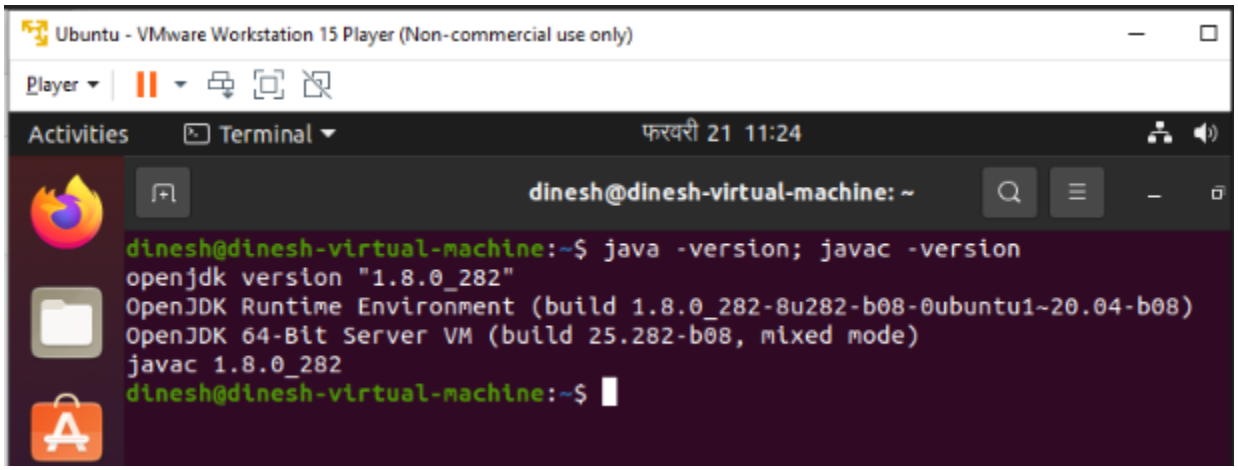


1.2. Here we are going to install open jdk 8 package as it contains both runtime and the development key.
Open the terminal and type the following code

- sudo apt install openjdk-8-jdk -y



1.3. Check if the java is installed in the pc.



2. Setting up a dedicated user for Hadoop user

A distinct user improves security and helps us arrange clusters more efficiently.

hdoop : It is the name of user account that you want to dedicate for Hadoop. Please replace the word with your specific username that you want to create.

2.1. First we want to do is install open SSH on Ubuntu

1) Install openssh server and client using the following command

```
sudo apt install openssh-server openssh-client -y
```

2.2. Creating Hadoop user

1) We are going to use the **adduser** command to add the user

```
sudo adduser hdoop
```

Note: User name is very curial part here

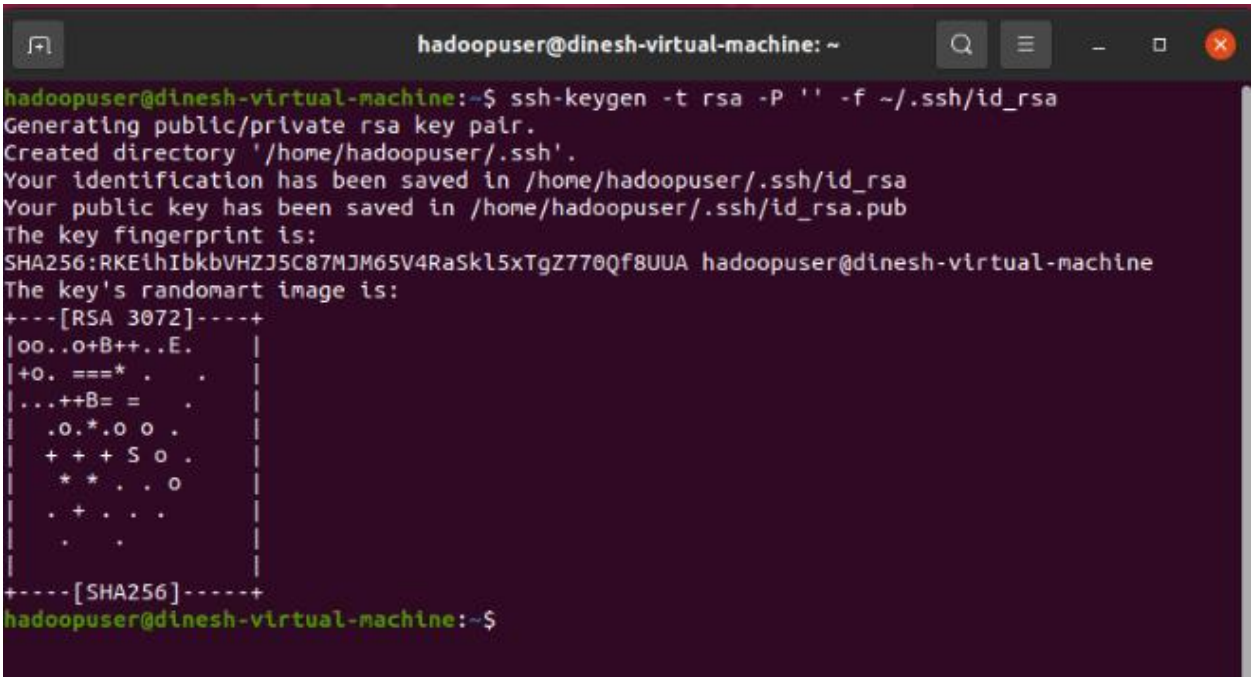
2) Lets switch the user to newly created user with the following command

```
su - hdoop
```

2.3. Enabling Password less SSH for Hadoop user

1) Generating an ssh key pair and define the location to be stored

```
ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```

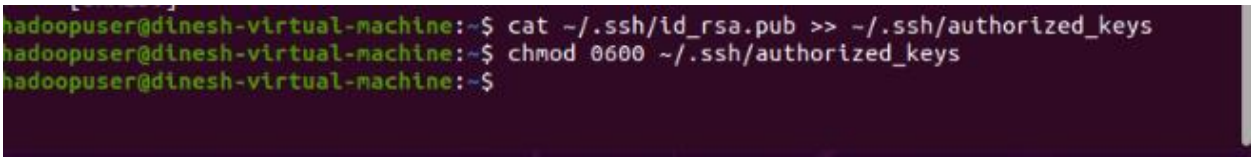


2) Using the cat command to store the public key as authorized_key in the ssh directory by the following command

```
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

3) Setting the permissions for the user with the chmod command

```
chmod 0600 ~/.ssh/authorized_keys
```



4) Now switch the user and ssh the local host with the following command

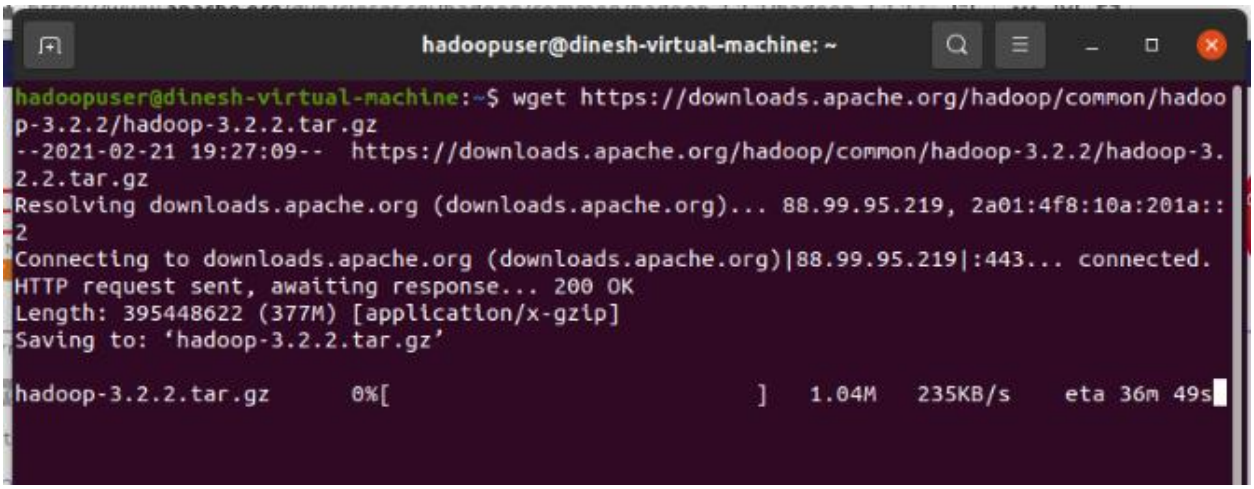
```
ssh localhost
```

3. Downloading and installing Hadoop locally

3.1. Downloading the Hadoop file

1) We have to specify the version we want do download properly. **Hadoop 3.2.2 is recommended**.

```
wget https://downloads.apache.org/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz
```



2) Now with the tar we will have to extract the Hadoop for initializing the installation

```
tar xzf hadoop-3.2.2.tar.gz
```

All the Hadoop files are now located in Hadoop

3.2. Adding User to the sudoers group for privileges

1) To edit the configuration file we must provide proper root privileges. So we have to add the newly created user to the admin group. **Switch to the main root user**

```
su – main_root_user
```

2) Run the following command

```
sudo usermod -aG sudo hdoop
```

3.3. Configuring Hadoop Environment Variables

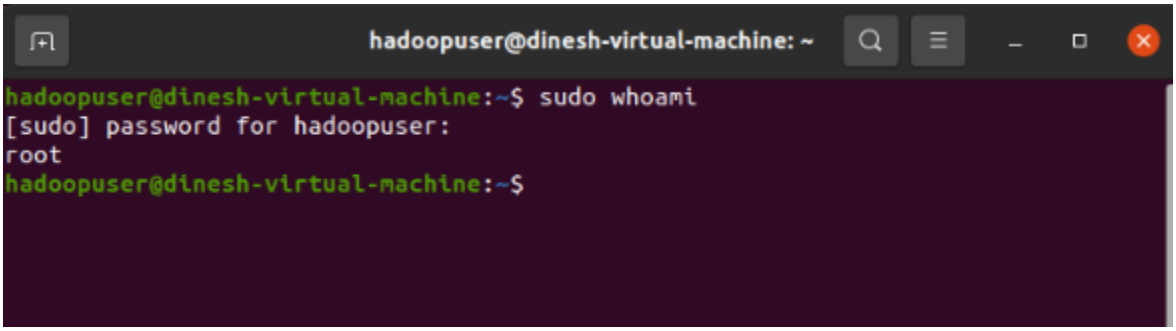
Now switch to your Hadoop user with the following code

```
su  - hdoop
```

1) To check if the Hadoop user is in the soders. (Checking for root privileges)

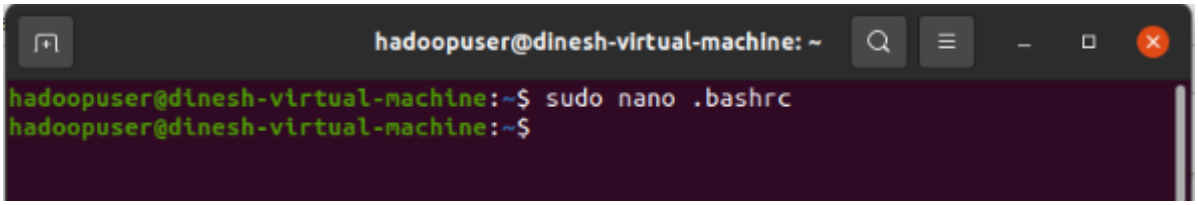
```
sudo whoami
```

the result should return **root**



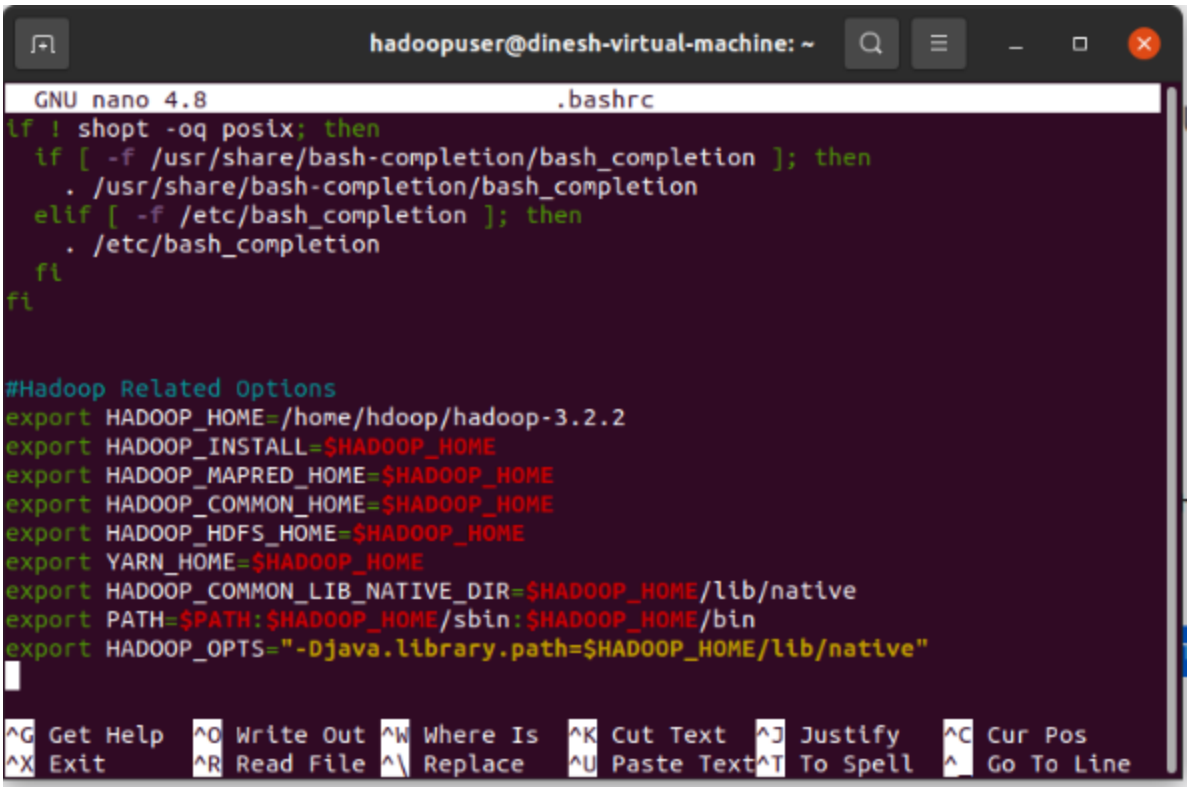
2) We have to first edit the .bashrc file.

```
sudo nano .bashrc
```



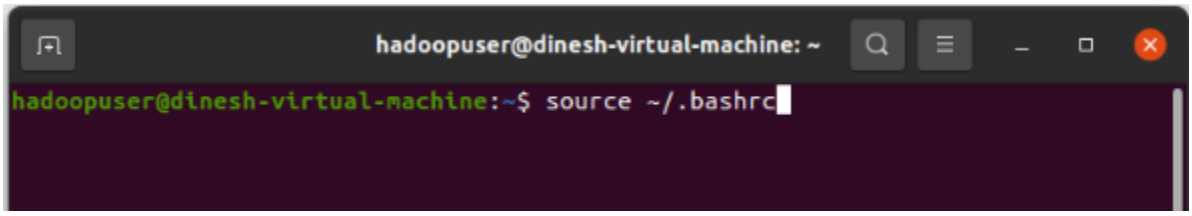
3) After the bashrc file opens, add the following code. Make sure you check your variable accordingly. **Navigate to the end of the file and add the following code.**

```
#Hadoop Related Options

export HADOOP_HOME=/home/hdoop/hadoop-3.2.2

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/nativ"
```



3) Now we must apply the changes to currently running environment. To do that type the following command.

```
source ~/.bashrc
```



4) Now we have to edit the Hadoop-env.sh file.
Note: If you have installed the Open JDK 8 you can direct go to step 5
   - Here first we have to know the location of file where our java is installed

```
which javac
```

```
hadoopuser@dinesh-virtual-machine: ~
hadoopuser@dinesh-virtual-machine:~$ which javac
/usr/bin/javac
hadoopuser@dinesh-virtual-machine:~$
```

- Lets find the open JDK directory

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

```
hdoop@pnap-VirtualBox:~$ readlink -f /usr/bin/javac
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac
```

Copy the path upto amd64

5) Now lets open the Hadoop –env.sh file

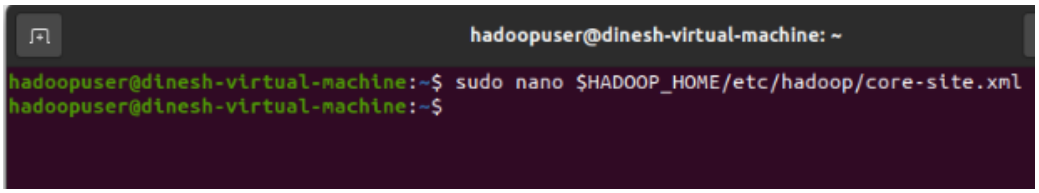
```
sudo nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

6) Seach for JAVA_HOME and uncomment it (delete the #) and paste the copied location

```
###

# 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=/usr/lib/jvm/java-8-openjdk-amd64

# Location of Hadoop.  By default, Hadoop will attempt to determine
# this location based upon its execution path.
# export HADOOP_HOME=

# Location of Hadoop's configuration information.  i.e., where this
# file is living. If this is not defined, Hadoop will attempt to
# locate it based upon its execution path.
#
# NOTE: It is recommend that this variable not be set here but in
```

7) Now lets edit Core-site.xml file

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



8) At the end of the file add the following configuration.

```
<configuration>

<property>

  <name>hadoop.tmp.dir</name>

  <value>/home/hdoop/tmpdata</value>

</property>

<property>

  <name>fs.default.name</name>

  <value>hdfs://127.0.0.1:9000</value>

</property>
</configuration>
```

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

<configuration>
<property>
  <name>hadoop.tmp.dir</name>
  <value>/home/hadoopuser/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>
```

9) Now we have to edit hdfs-site.xml file. To open the file type the following command

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

```
hadoopuser@dinesh-virtual-machine: ~
hadoopuser@dinesh-virtual-machine:~$ sudo nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml
```

10) Add the following command in the file.

```
<configuration>

<property>

  <name>dfs.data.dir</name>

  <value>/home/hdoop/dfsdata/namenode</value>

</property>

<property>

  <name>dfs.data.dir</name>

  <value>/home/hdoop/dfsdata/datanode</value>

</property>

<property>
```

```
  <name>dfs.replication</name>

  <value>1</value>

</property>
</configuration>
```

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

<configuration>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hadoopuser/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/hadoopuser/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>

</configuration>
```

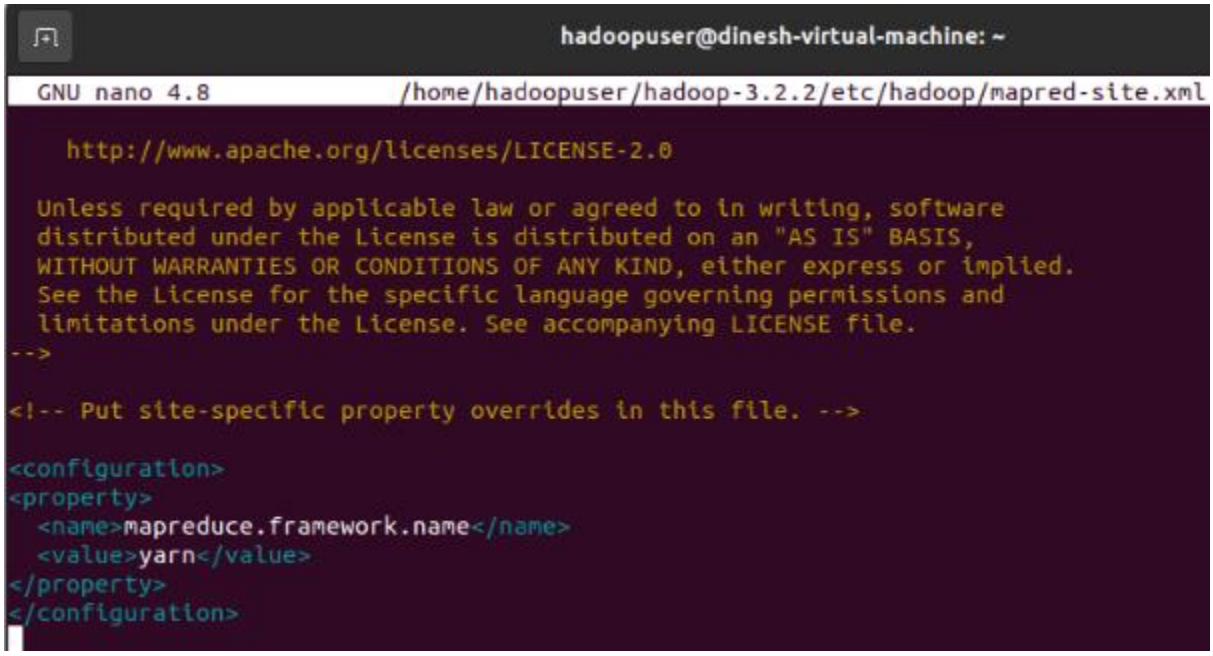
11) Edit mapered-site.xml file.

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

12) Add the following configuration at last of the file

```
<configuration>

<property>

  <name>mapreduce.framework.name</name>

  <value>yarn</value>

</property>
```

```
</configuration>
```



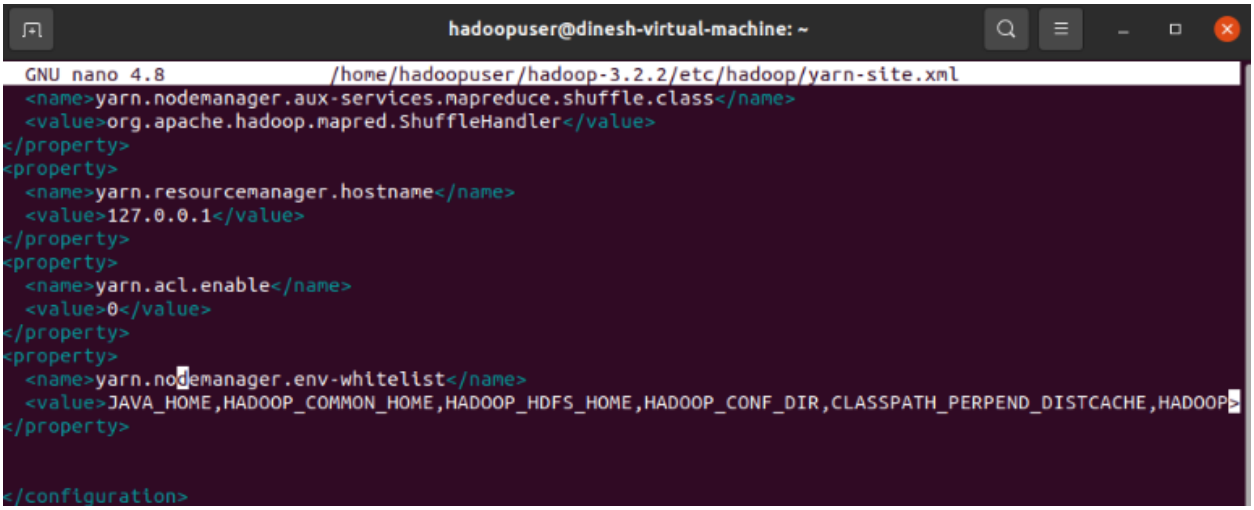
13) Editing yarn-site.xml file

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

14) Add the following configuration in the file.

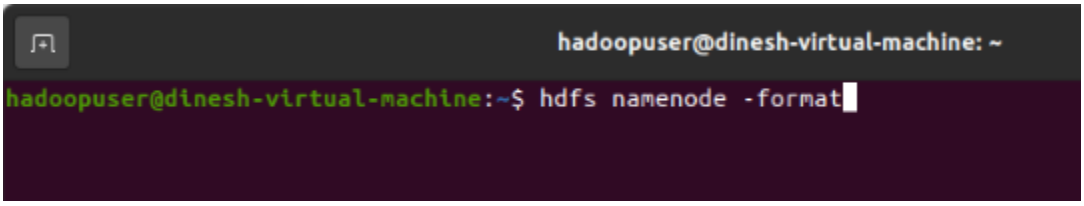
```
<configuration>

<property>

  <name>yarn.nodemanager.aux-services</name>

  <value>mapreduce_shuffle</value>

</property>

<property>

  <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

  <value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

<property>

  <name>yarn.resourcemanager.hostname</name>

  <value>127.0.0.1</value>

</property>

<property>

  <name>yarn.acl.enable</name>

  <value>0</value>

</property>

<property>

  <name>yarn.nodemanager.env-whitelist</name>

  <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_C
ONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MA
PRED_HOME</value>
```

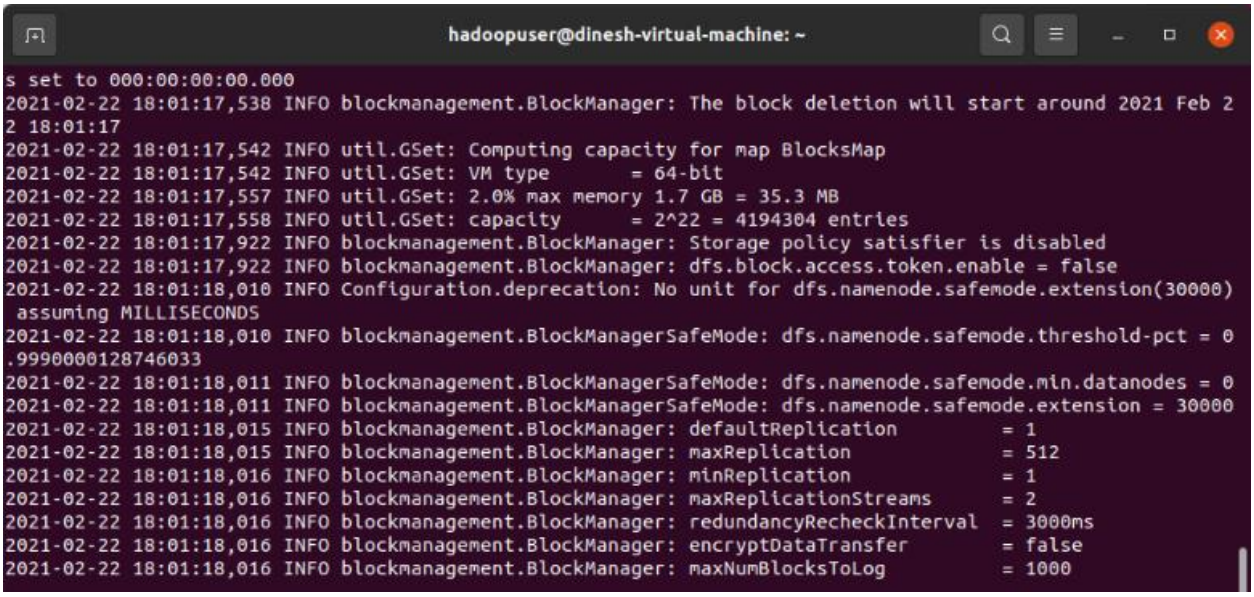
```
</property>
</configuration>
```



15) We now have to format hdfs name node. To do that type the following command.

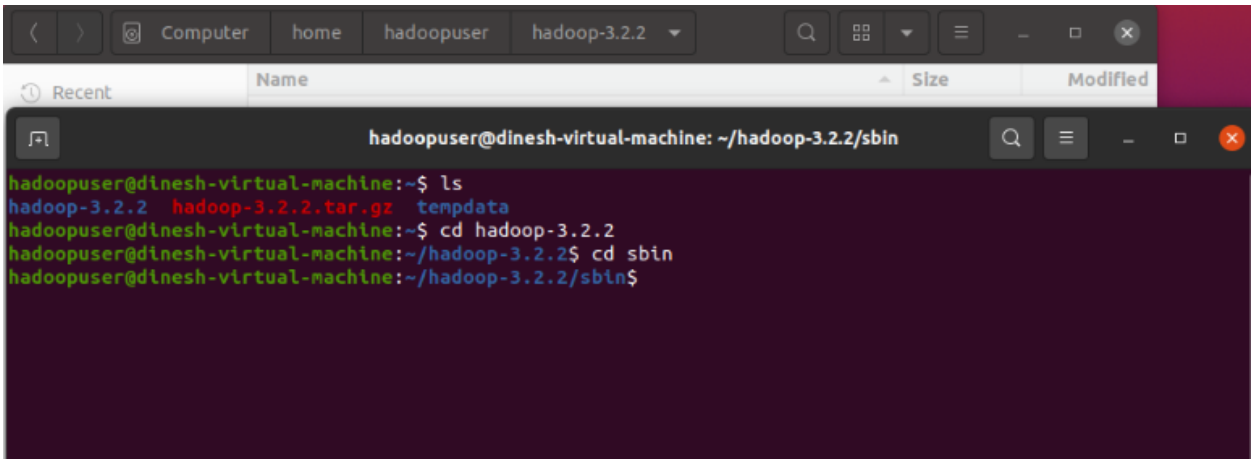
```
hdfs namenode -format
```





16) To start Hadoop cluster type the following command.
First you have to go to the **sbin** folder which is located at your Hadoop sbin folder.
Hadoop -> sbin .

```
cd hadoop-3.2.2/sbin
```




```
./start-dfs.sh
```

```
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [dinesh-virtual-machine]
dinesh-virtual-machine: Warning: Permanently added 'dinesh-virtual-machine' (ECDSA) to the list of known h
osts.
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2/sbin$ jps
5969 Jps
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2/sbin$
```

17) Starting the yarn resource manager.

```
./start-yarn.sh
```

```
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2/sbin$ ./start-yarn.sh
Starting resourcemanager
Starting nodemanagers
```

18) Now check if all the java services are running.

```
jps
```

```
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2/sbin$ jps
2576 NameNode
2704 DataNode
3059 Jps
2060 NodeManager
2878 SecondaryNameNode
1823 ResourceManager
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2/sbin$
```

!!** Congratulations Hadoop installation Successful **!!