

# Hadoop Installation Guide

---

**Prepared By: Jnaneshwar Bohara**

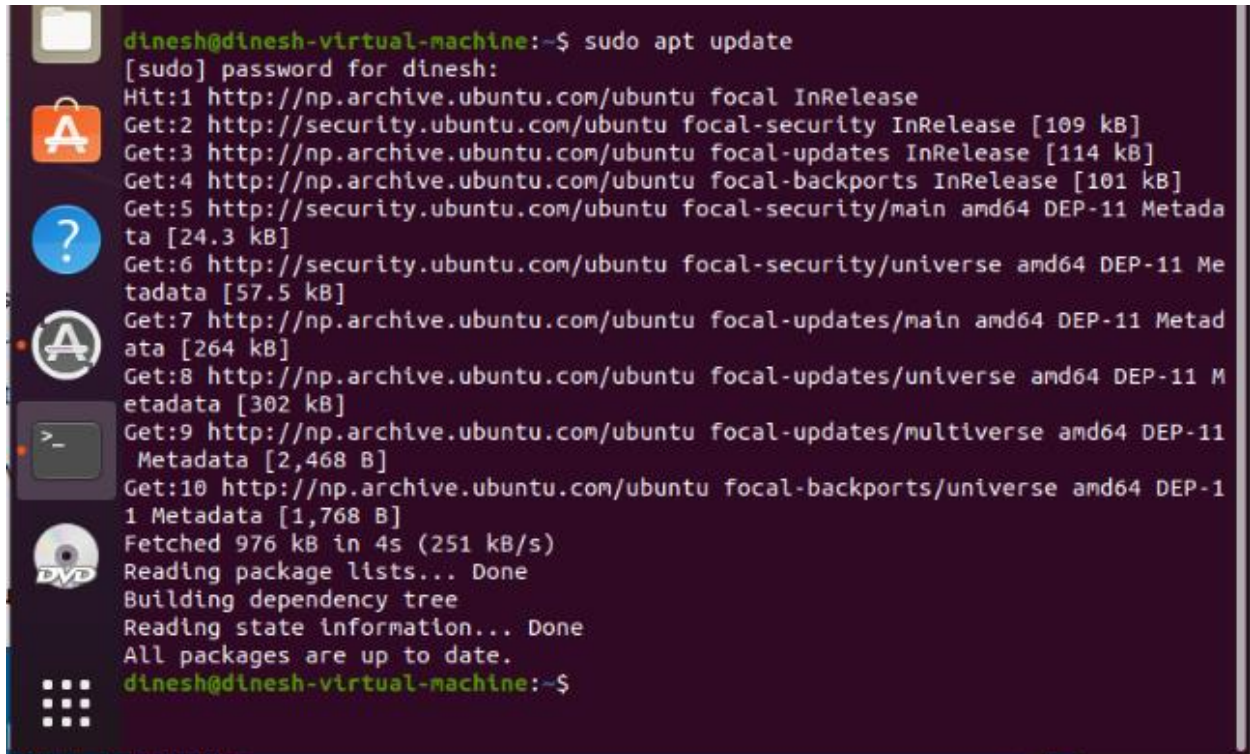
## Table of Contents

1. Java installation.....	2
2. Setting up a dedicated user for Hadoop user.....	4
2.1. First we want to do is install open SSH on Ubuntu .....	4
2.2. Creating Hadoop user.....	4
2.3. Enabling Password less SSH for Hadoop user .....	4
3. Downloading and installing Hadoop locally .....	6
3.1. Downloading the Hadoop file .....	6
3.2. Adding User to the sudoers group for privileges .....	6
3.3. Configuring Hadoop Environment Variables.....	7

## 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.

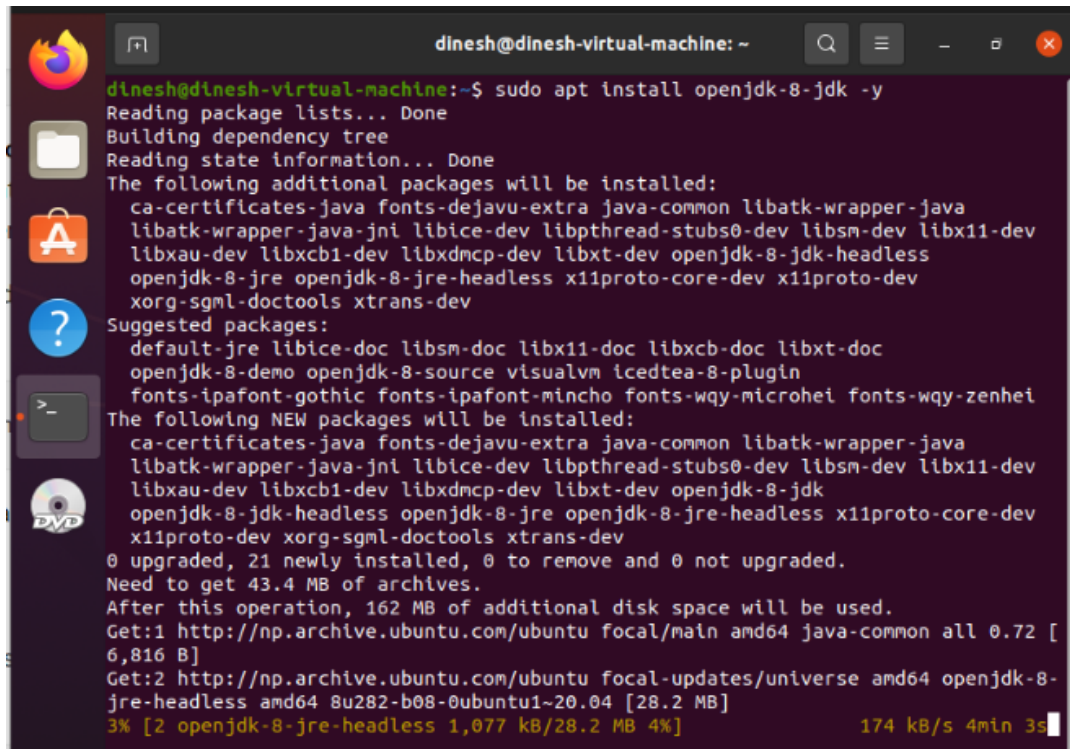
```
sudo apt update
```



```
dinesh@dinesh-virtual-machine:~$ sudo apt update
[sudo] password for dinesh:
Hit:1 http://np.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:3 http://np.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://np.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [57.5 kB]
Get:7 http://np.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [264 kB]
Get:8 http://np.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [302 kB]
Get:9 http://np.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:10 http://np.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]
Fetched 976 kB in 4s (251 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
dinesh@dinesh-virtual-machine:~$
```

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

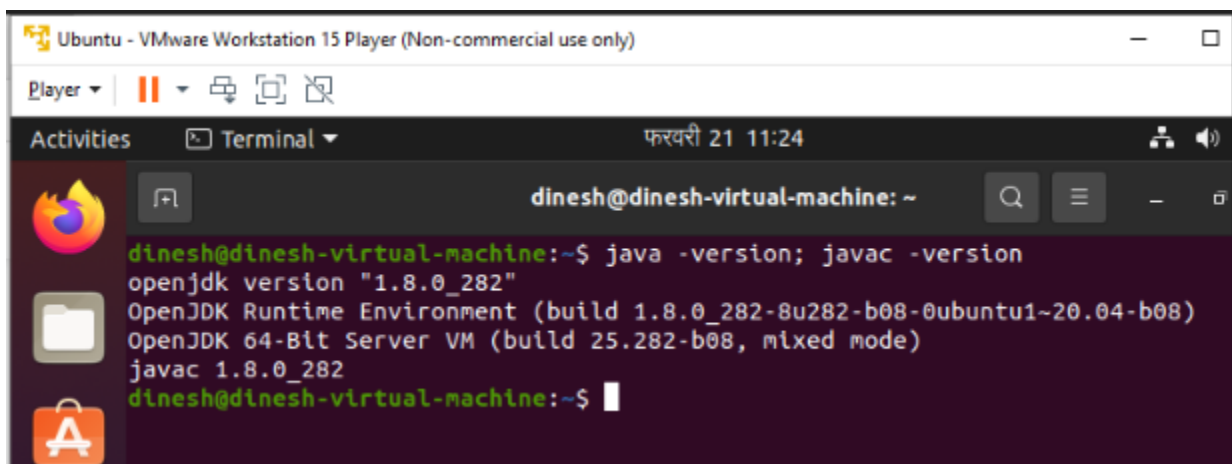
- `sudo apt install openjdk-8-jdk -y`



A terminal window titled "dinesh@dinesh-virtual-machine: ~" showing the command `sudo apt install openjdk-8-jdk -y` and its output. The output includes package lists, dependency tree building, and a list of additional packages to be installed. It also shows the disk space requirements and the progress of the download.

```
dinesh@dinesh-virtual-machine:~$ sudo apt install openjdk-8-jdk -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-dev
libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk-headless
openjdk-8-jre openjdk-8-jre-headless x11proto-core-dev x11proto-dev
xorg-sgml-doctools xtrans-dev
Suggested packages:
default-jre libice-doc libsm-doc libx11-doc libxcb-doc libxt-doc
openjdk-8-demo openjdk-8-source visualvm icedtea-8-plugin
fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei fonts-wqy-zenhei
The following NEW packages will be installed:
ca-certificates-java fonts-dejavu-extra java-common libatk-wrapper-java
libatk-wrapper-java-jni libice-dev libpthread-stubs0-dev libsm-dev libx11-dev
libxau-dev libxcb1-dev libxdmcp-dev libxt-dev openjdk-8-jdk
openjdk-8-jdk-headless openjdk-8-jre openjdk-8-jre-headless x11proto-core-dev
x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 21 newly installed, 0 to remove and 0 not upgraded.
Need to get 43.4 MB of archives.
After this operation, 162 MB of additional disk space will be used.
Get:1 http://np.archive.ubuntu.com/ubuntu focal/main amd64 java-common all 0.72 [
6,816 B]
Get:2 http://np.archive.ubuntu.com/ubuntu focal-updates/universe amd64 openjdk-8-
jre-headless amd64 8u282-b08-0ubuntu1~20.04 [28.2 MB]
3% [2 openjdk-8-jre-headless 1,077 kB/28.2 MB 4%] 174 kB/s 4min 3s
```

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



A terminal window titled "dinesh@dinesh-virtual-machine: ~" showing the command `java -version; javac -version` and its output. The output displays the OpenJDK version and runtime environment details.

```
dinesh@dinesh-virtual-machine:~$ java -version; javac -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-8u282-b08-0ubuntu1~20.04-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
javac 1.8.0_282
dinesh@dinesh-virtual-machine:~$
```

## 2. Setting up a dedicated user for Hadoop user

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

**hadoop** : 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 hadoop
```

**Note: User name is very curial part here**

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

```
su - hadoop
```

### 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
```

```
hadoopuser@dinesh-virtual-machine: ~  
hadoopuser@dinesh-virtual-machine:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa  
Generating public/private rsa key pair.  
Created directory '/home/hadoopuser/.ssh'.  
Your identification has been saved in /home/hadoopuser/.ssh/id_rsa  
Your public key has been saved in /home/hadoopuser/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:RKElhIbkbVHZJ5C87MJM65V4RaSk15xTgZ770Qf8UUA hadoopuser@dinesh-virtual-machine  
The key's randomart image is:  
+---[RSA 3072]-----+  
|oo..o+B++..E.|  
|+O. ==* . .|  
|...++B= . .|  
|.o.*.o o .|  
|++ + S o .|  
|* * . . o|  
|. + . . .|  
|. . . .|  
+-----[SHA256]-----+  
hadoopuser@dinesh-virtual-machine:~$
```

- 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
```

```
hadoopuser@dinesh-virtual-machine:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
hadoopuser@dinesh-virtual-machine:~$ chmod 0600 ~/.ssh/authorized_keys  
hadoopuser@dinesh-virtual-machine:~$
```

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

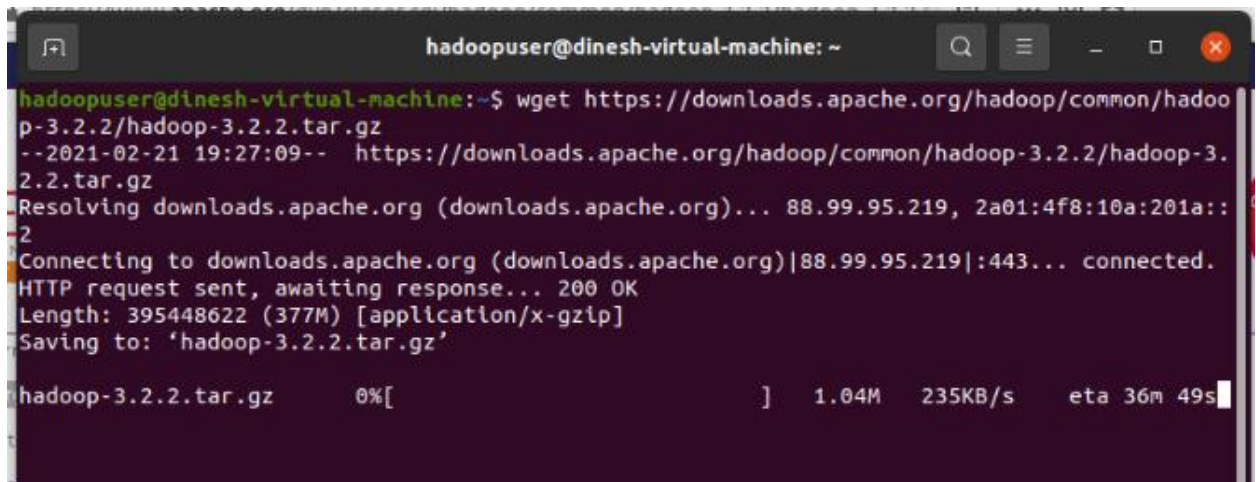
```
ssh localhost
```

### 3. Downloading and installing Hadoop locally

#### 3.1. Downloading the Hadoop file

- 1) We have to specify the version we want to 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
```



```
hadoopuser@dinesh-virtual-machine: ~  
hadoopuser@dinesh-virtual-machine:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.tar.gz  
--2021-02-21 19:27:09-- https://downloads.apache.org/hadoop/common/hadoop-3.2.2/hadoop-3.2.2.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: 395448622 (377M) [application/x-gzip]  
Saving to: 'hadoop-3.2.2.tar.gz'  
hadoop-3.2.2.tar.gz      0%[          ] 1.04M  235KB/s  eta 36m 49s
```

- 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

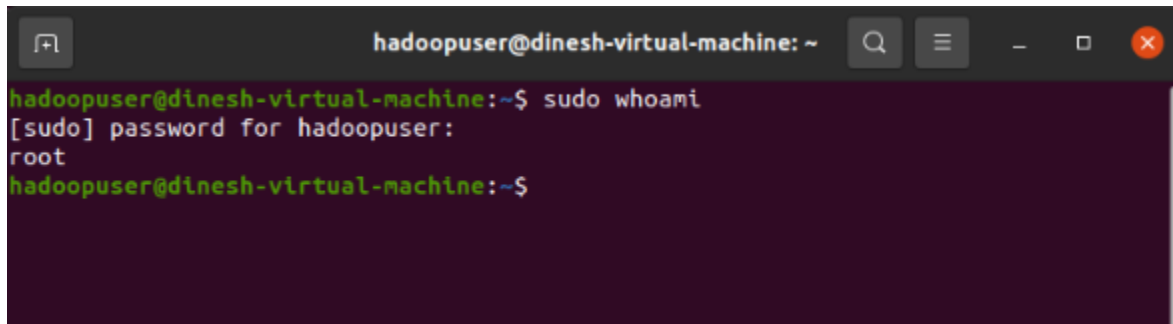
Now switch to your Hadoop user with the following code

```
su - hadoop
```

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

```
sudo whoami
```

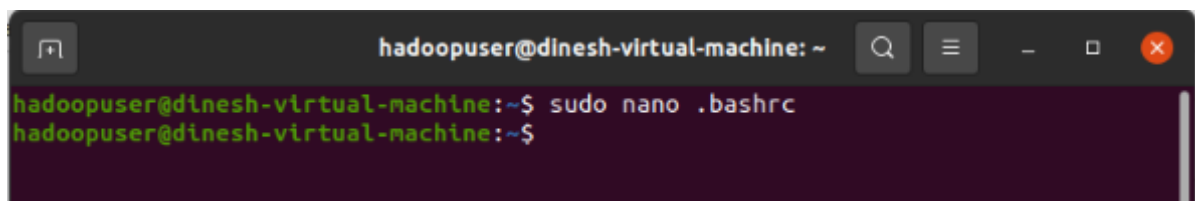
the result should return **root**



```
hadoopuser@dinesh-virtual-machine: ~  
hadoopuser@dinesh-virtual-machine:~$ sudo whoami  
[sudo] password for hadoopuser:  
root  
hadoopuser@dinesh-virtual-machine:~$
```

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

```
sudo nano .bashrc
```



```
hadoopuser@dinesh-virtual-machine: ~  
hadoopuser@dinesh-virtual-machine:~$ sudo nano .bashrc  
hadoopuser@dinesh-virtual-machine:~$
```

- 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

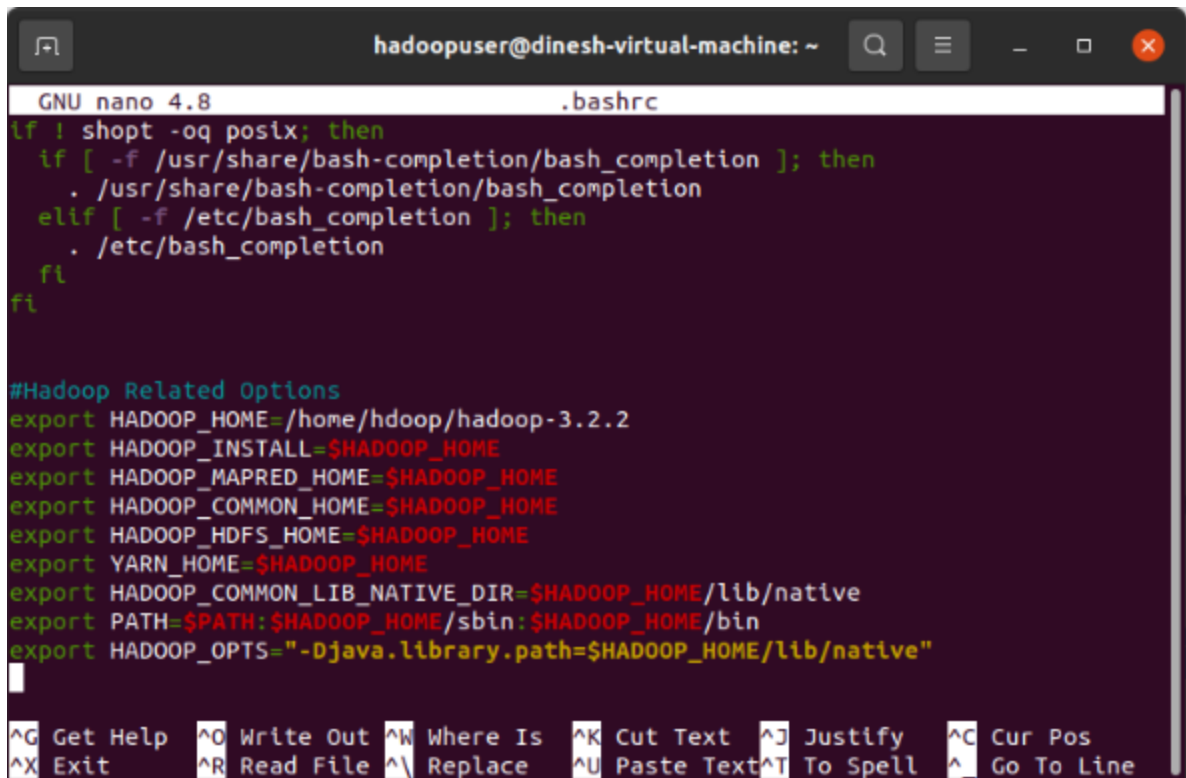
export HADOOP_HDFS_HOME=$HADOOP_HOME

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"
```





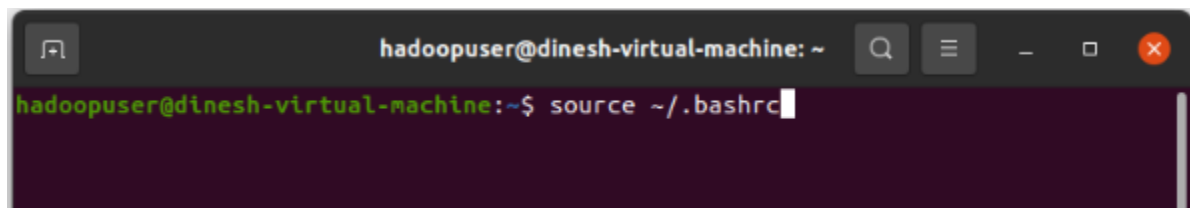
```
GNU nano 4.8 .bashrc
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

#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/native"

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

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

```
source ~/.bashrc
```



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

- Lets find the open JDK directory

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

```
hadoop@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) Search 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
```

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

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

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

```

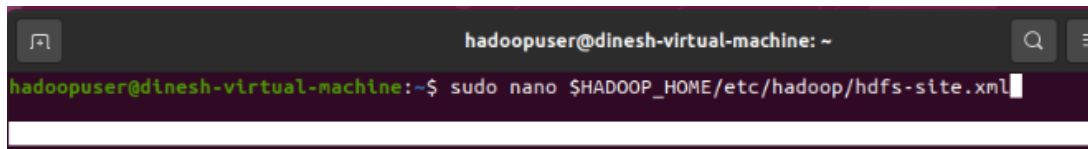
<!-- Put site-specific property overrides in this file. -->

<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
```



- 10) Add the following command in the file.

```

<configuration>

<property>

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

  <value>/home/hadoop/dfsdata/namenode</value>

</property>

<property>

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

  <value>/home/hadoop/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 mapred-site.xml file.

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

12) Add the following configuration at last of the file

```
<configuration>

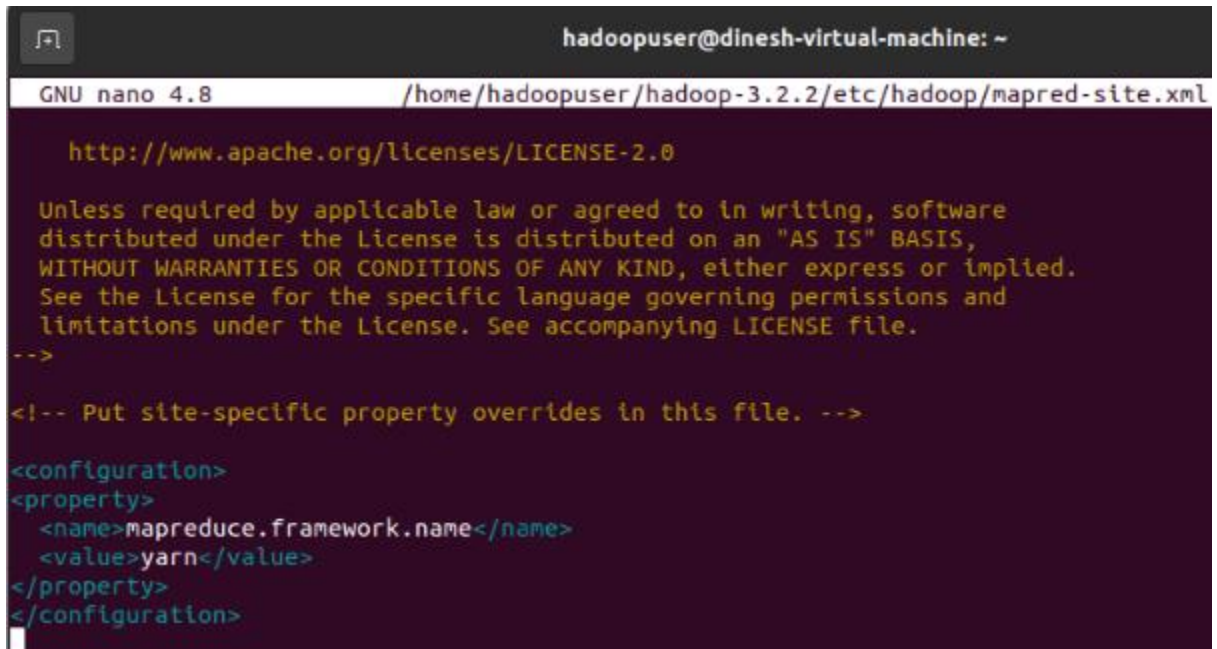
<property>

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

  <value>yarn</value>

</property>
```

```
</configuration>
```



The screenshot shows a terminal window with the prompt `hadoopuser@dinesh-virtual-machine: ~`. The nano text editor is open, editing the file `/home/hadoopuser/hadoop-3.2.2/etc/hadoop/mapred-site.xml`. The editor's status bar at the top indicates `GNU nano 4.8`. The content of the file is as follows:

```
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 implied.
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>
```

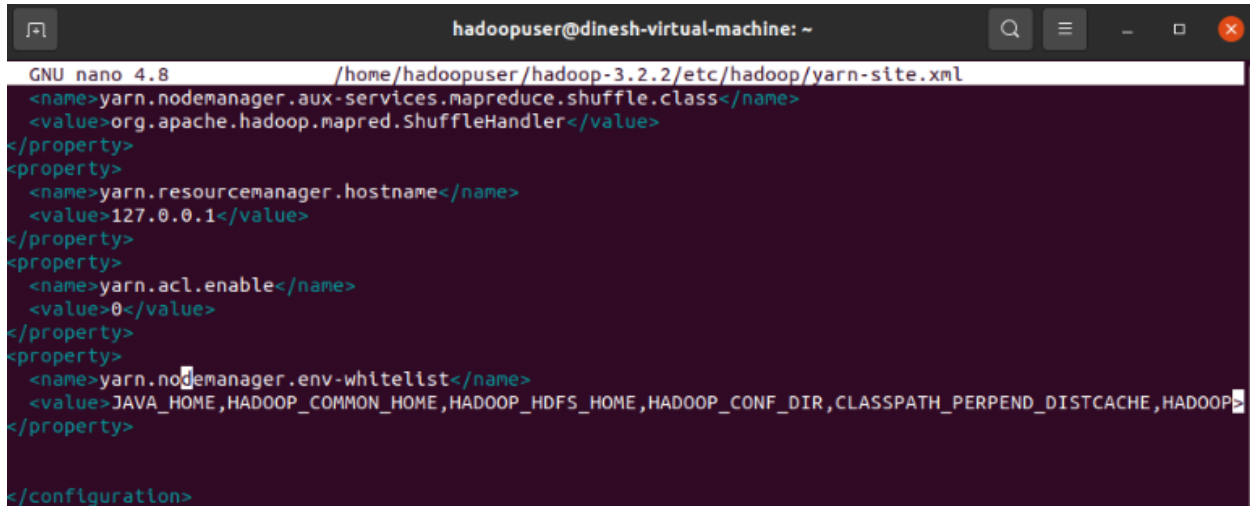
### 13) Editing yarn-site.xml file

```
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_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_MAPRED_HOME</value>
</property>
</configuration>
```

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

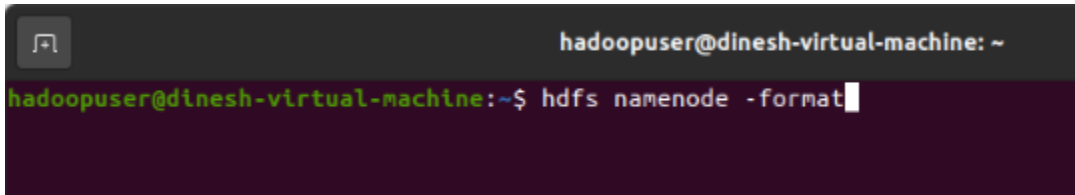


The screenshot shows a terminal window with the title 'hadoopuser@dinesh-virtual-machine: ~'. The nano editor is open, editing the file '/home/hadoopuser/hadoop-3.2.2/etc/hadoop/yarn-site.xml'. The content of the file is as follows:

```
GNU nano 4.8 /home/hadoopuser/hadoop-3.2.2/etc/hadoop/yarn-site.xml
<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_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_
</property>
</configuration>
```

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

```
hdfs namenode -format
```



The screenshot shows a terminal window with the title 'hadoopuser@dinesh-virtual-machine: ~'. The command prompt is 'hadoopuser@dinesh-virtual-machine:~\$' and the command 'hdfs namenode -format' has been entered, with the cursor at the end of the line.



```
hadoopuser@dinesh-virtual-machine: ~  
s set to 000:00:00:00.000  
2021-02-22 18:01:17,538 INFO blockmanagement.BlockManager: The block deletion will start around 2021 Feb 22 18:01:17  
2021-02-22 18:01:17,542 INFO util.GSet: Computing capacity for map BlocksMap  
2021-02-22 18:01:17,542 INFO util.GSet: VM type = 64-bit  
2021-02-22 18:01:17,557 INFO util.GSet: 2.0% max memory 1.7 GB = 35.3 MB  
2021-02-22 18:01:17,558 INFO util.GSet: capacity = 2^22 = 4194304 entries  
2021-02-22 18:01:17,922 INFO blockmanagement.BlockManager: Storage policy satisfier is disabled  
2021-02-22 18:01:17,922 INFO blockmanagement.BlockManager: dfs.block.access.token.enable = false  
2021-02-22 18:01:18,010 INFO Configuration.deprecation: No unit for dfs.namenode.safemode.extension(30000) assuming MILLISECONDS  
2021-02-22 18:01:18,010 INFO blockmanagement.BlockManagerSafeMode: dfs.namenode.safemode.threshold-pct = 0.9990000128746033  
2021-02-22 18:01:18,011 INFO blockmanagement.BlockManagerSafeMode: dfs.namenode.safemode.min.datanodes = 0  
2021-02-22 18:01:18,011 INFO blockmanagement.BlockManagerSafeMode: dfs.namenode.safemode.extension = 30000  
2021-02-22 18:01:18,015 INFO blockmanagement.BlockManager: defaultReplication = 1  
2021-02-22 18:01:18,015 INFO blockmanagement.BlockManager: maxReplication = 512  
2021-02-22 18:01:18,016 INFO blockmanagement.BlockManager: minReplication = 1  
2021-02-22 18:01:18,016 INFO blockmanagement.BlockManager: maxReplicationStreams = 2  
2021-02-22 18:01:18,016 INFO blockmanagement.BlockManager: redundancyRecheckInterval = 3000ms  
2021-02-22 18:01:18,016 INFO blockmanagement.BlockManager: encryptDataTransfer = false  
2021-02-22 18:01:18,016 INFO blockmanagement.BlockManager: maxNumBlocksToLog = 1000
```

- 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
```

```
hadoopuser@dinesh-virtual-machine: ~/hadoop-3.2.2/sbin  
hadoopuser@dinesh-virtual-machine:~$ ls  
hadoop-3.2.2  hadoop-3.2.2.tar.gz  tempdata  
hadoopuser@dinesh-virtual-machine:~$ cd hadoop-3.2.2  
hadoopuser@dinesh-virtual-machine:~/hadoop-3.2.2$ cd sbin  
hadoopuser@dinesh-virtual-machine:~/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 \*\*!!**