
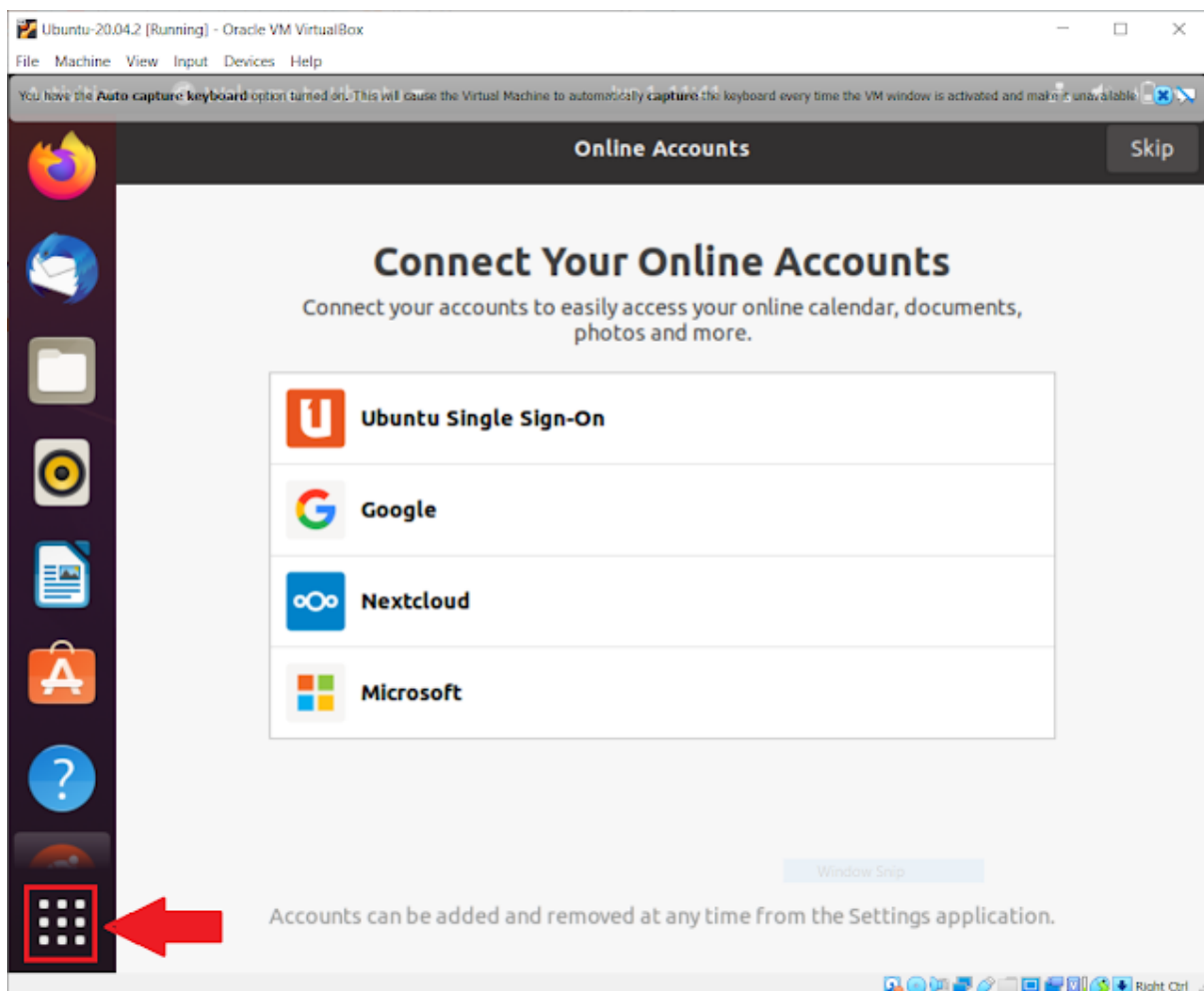


Hadoop Installation in Ubuntu

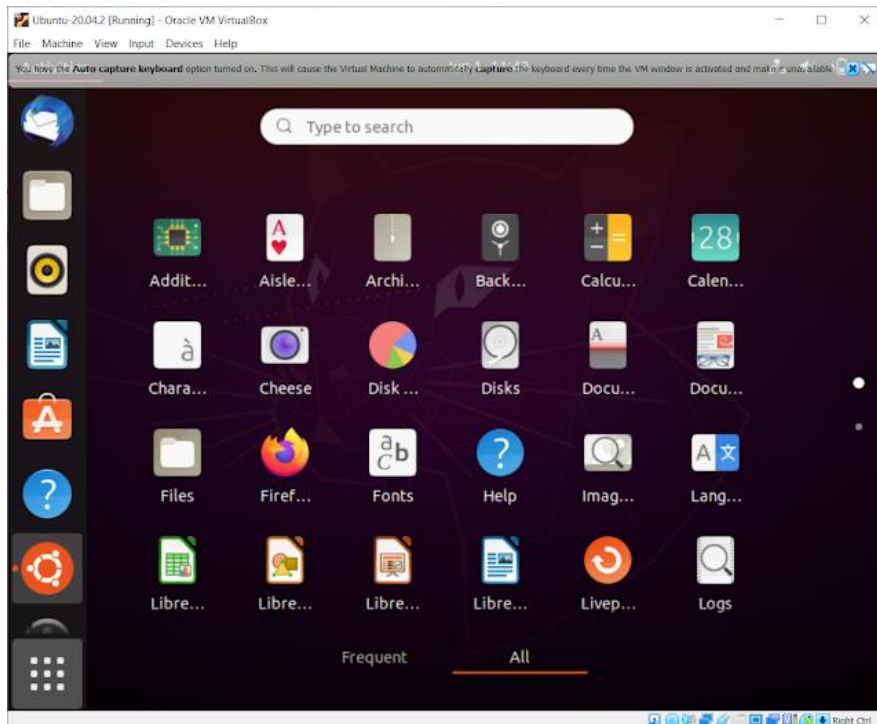
To install Hadoop on Ubuntu operating system we have to do some various tasks with the help of Ubuntu terminal. Terminal is the Command Line Interface (CLI) program in Ubuntu that allows you to create and delete files, run programs, and navigate through folders and files. It is same as command prompt in windows.

Steps to open Terminal in Ubuntu are:

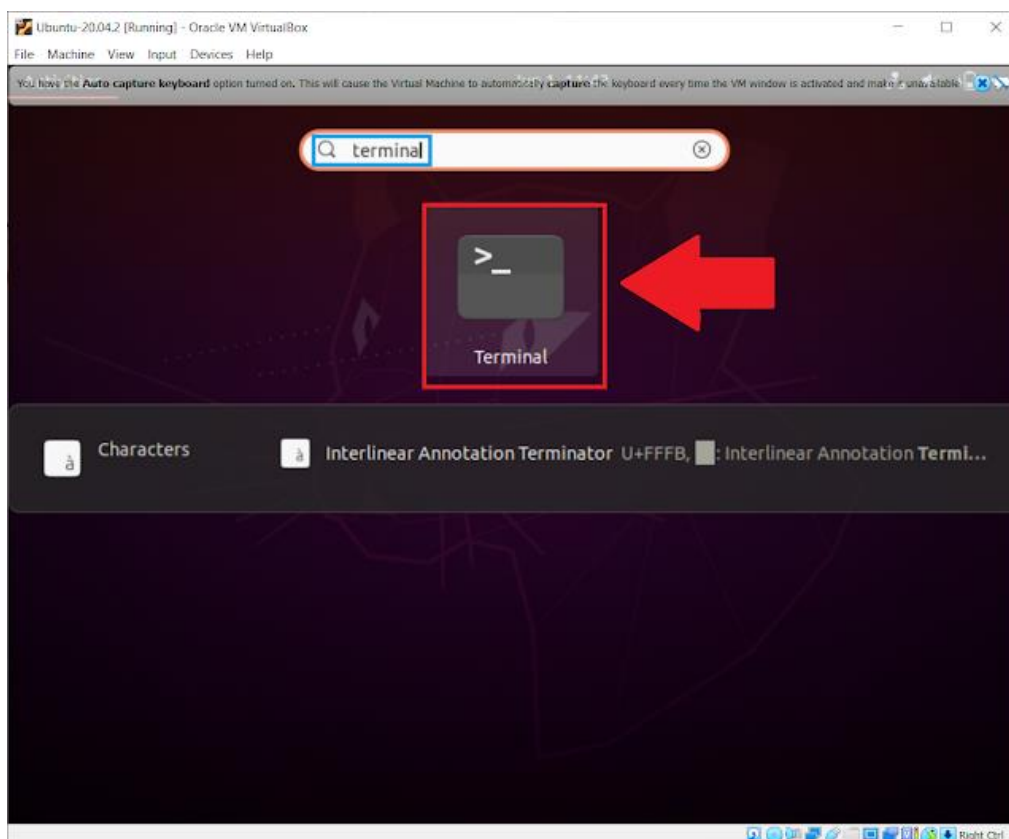
- 1) On main screen of **Ubuntu OS**, you will find  icon (Menu icon) on bottom-left side, click on it.



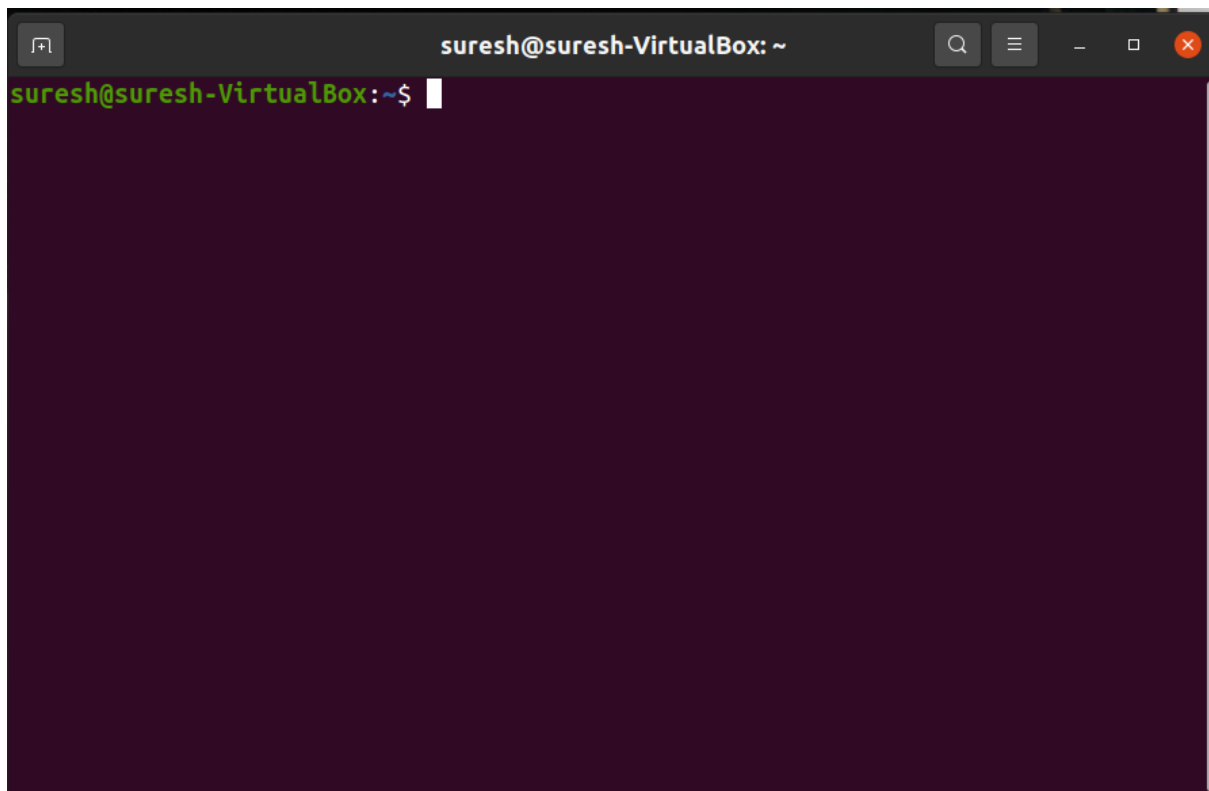
2) It will show different apps present in Ubuntu.



3) Now search for 'terminal' through search bar present on top and open it.



It will look like:



So Various tasks for installing Hadoop on ubuntu are:

Task 1: Update Ubuntu OS

Task 2: Installation of OpenJDK

Task 3: Installation of OpenSSH

Task 4: Enabling Passwordless SSH

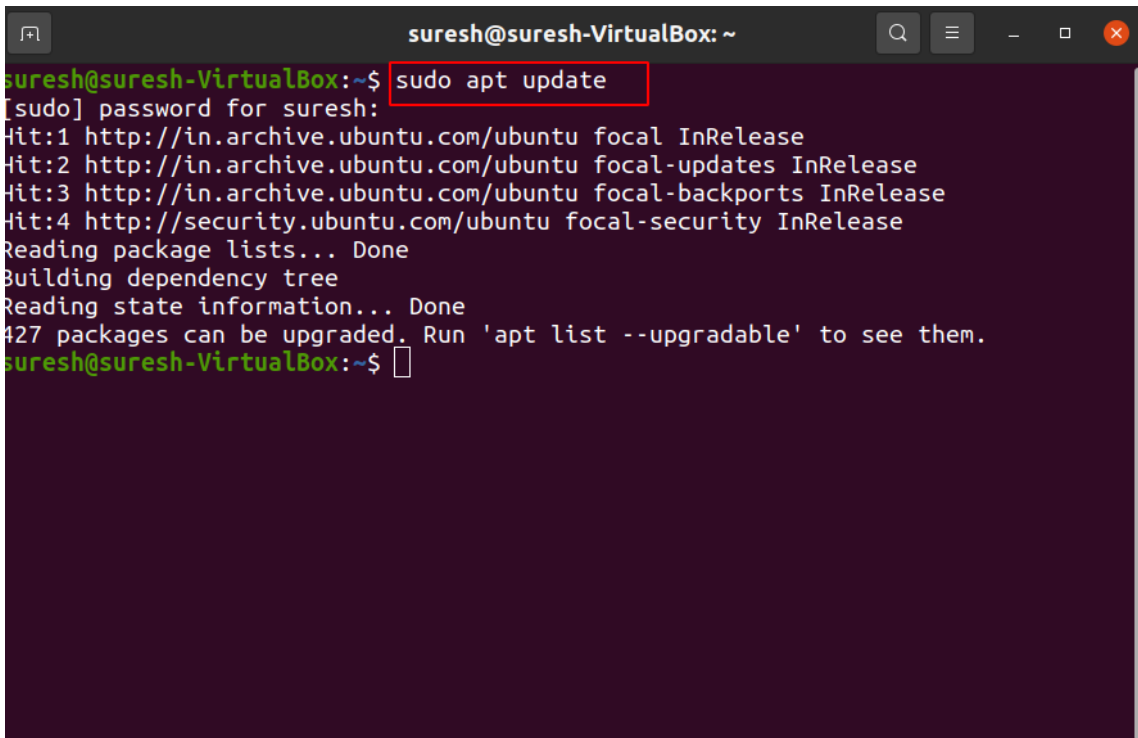
Task 5: Download, Install and configure Hadoop

Installation of Hadoop

Task 1: Update Ubuntu OS

To update Ubuntu OS, Type the following command on terminal

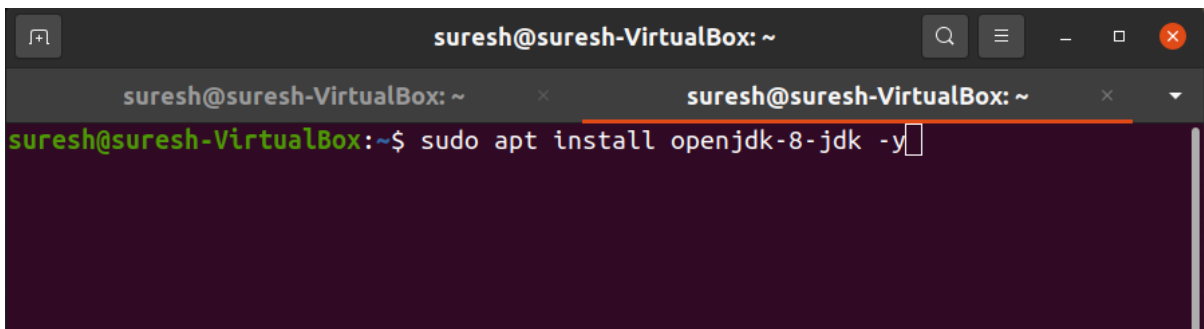
Command: sudo apt update

A terminal window titled 'suresh@suresh-VirtualBox: ~' with standard window controls. The command 'sudo apt update' is entered and highlighted with a red box. The output shows the system fetching package lists from various mirrors (in.archive.ubuntu.com and security.ubuntu.com), reading package lists, building a dependency tree, and reading state information. It concludes that 427 packages can be upgraded and suggests running 'apt list --upgradable' to see them.

```
suresh@suresh-VirtualBox:~$ sudo apt update
[sudo] password for suresh:
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
427 packages can be upgraded. Run 'apt list --upgradable' to see them.
suresh@suresh-VirtualBox:~$
```

Task 2: Installation of OpenJDK

Command: sudo apt install openjdk-8-jdk -y

A terminal window titled 'suresh@suresh-VirtualBox: ~' with standard window controls. The command 'sudo apt install openjdk-8-jdk -y' is entered at the prompt. The window has two tabs, both titled 'suresh@suresh-VirtualBox: ~'.

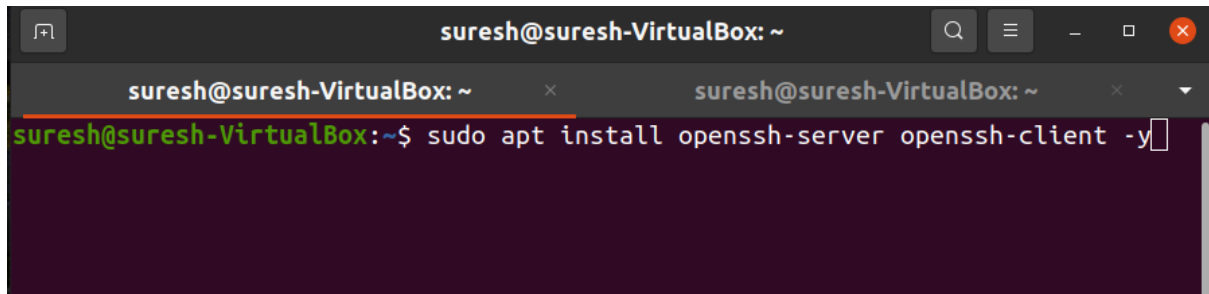
```
suresh@suresh-VirtualBox:~$ sudo apt install openjdk-8-jdk -y
```

If you want to check java versions by using

Command: java -version; javac -version

Task 3: Installation of OpenSSH

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



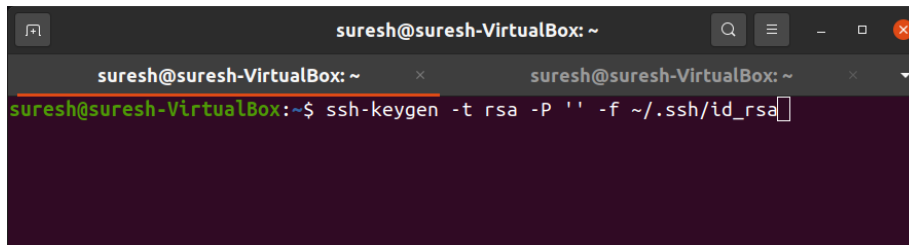
```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ sudo apt install openssh-server openssh-client -y
```

Task 4: Enabling Passwordless SSH

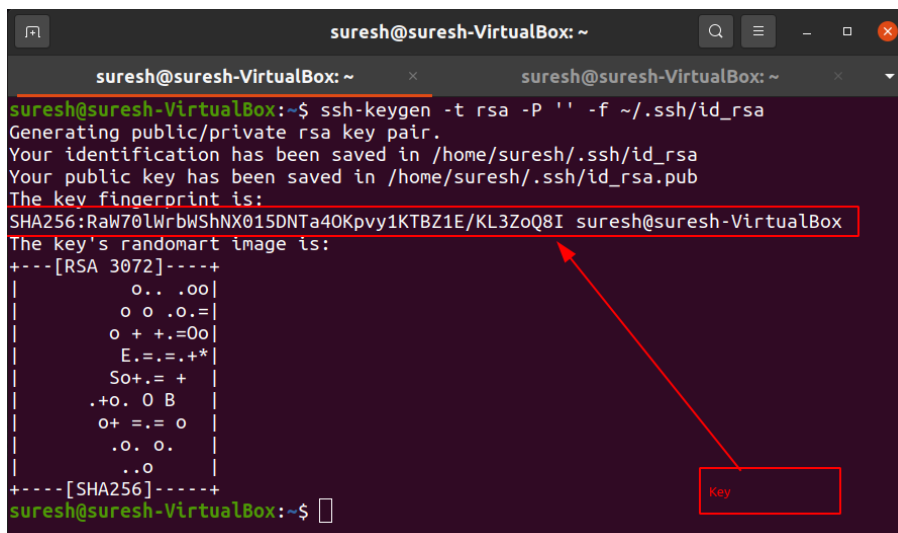
This task contains 4 steps and 4 commands

Step 1: Generating key pair

Command 1: `ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa`



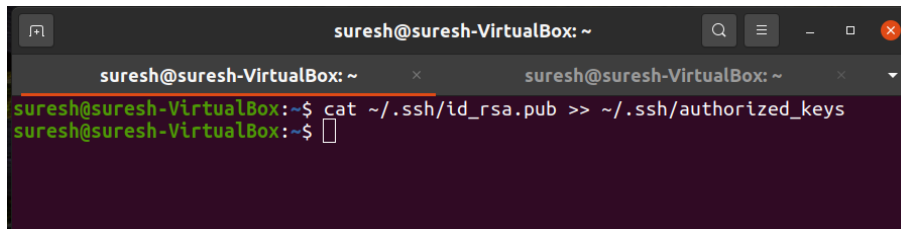
```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
```



```
suresh@suresh-VirtualBox:~$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa  
Generating public/private rsa key pair.  
Your identification has been saved in /home/suresh/.ssh/id_rsa  
Your public key has been saved in /home/suresh/.ssh/id_rsa.pub  
The key fingerprint is:  
SHA256:RaW70LWrbWShNX015DNTa40Kpvy1KTBZ1E/KL3ZoQ8I suresh@suresh-VirtualBox  
The key's randomart image is:  
+---[RSA 3072]-----+  
|  
|  o.. .oo|  
| o o .o.=|  
| o + +.=0o|  
| E.=.=.*|  
| So+.= +|  
|. +o. O B|  
| o+ =. o |  
|.o. o. |  
| ..o |  
+---[SHA256]-----+  
suresh@suresh-VirtualBox:~$
```

Step 2: Copying public key into authorized_keys file

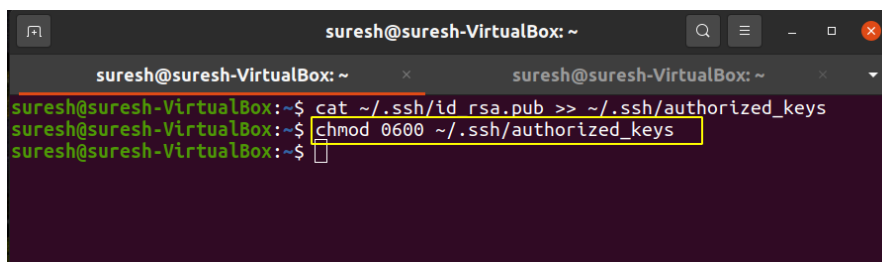
Command 2: `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`



```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
suresh@suresh-VirtualBox:~$
```

Step 3: Changing mod of authorized_keys file to Read and Write only for owner

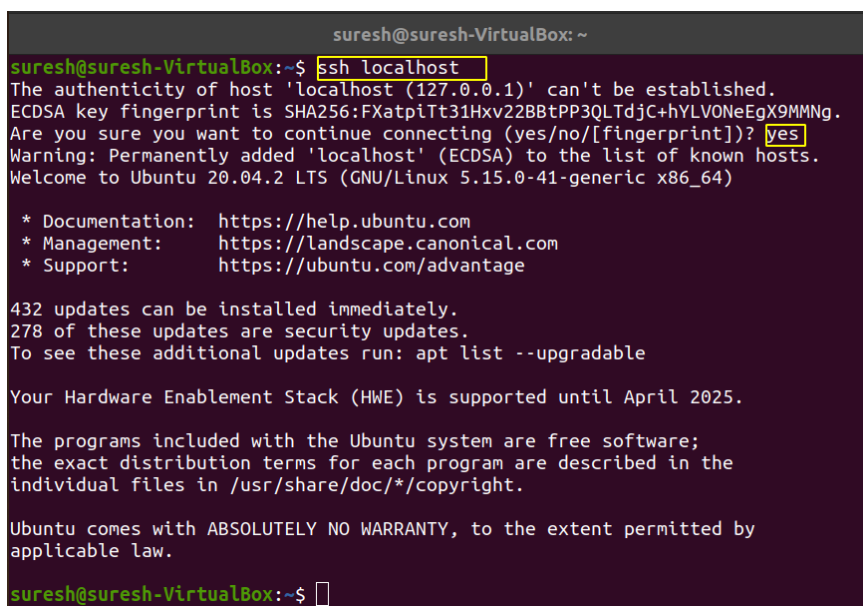
Command 3: `chmod 0600 ~/.ssh/authorized_keys`



```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys  
suresh@suresh-VirtualBox:~$ chmod 0600 ~/.ssh/authorized_keys  
suresh@suresh-VirtualBox:~$
```

Step 4: Starting localhost

Command 4: `ssh localhost`

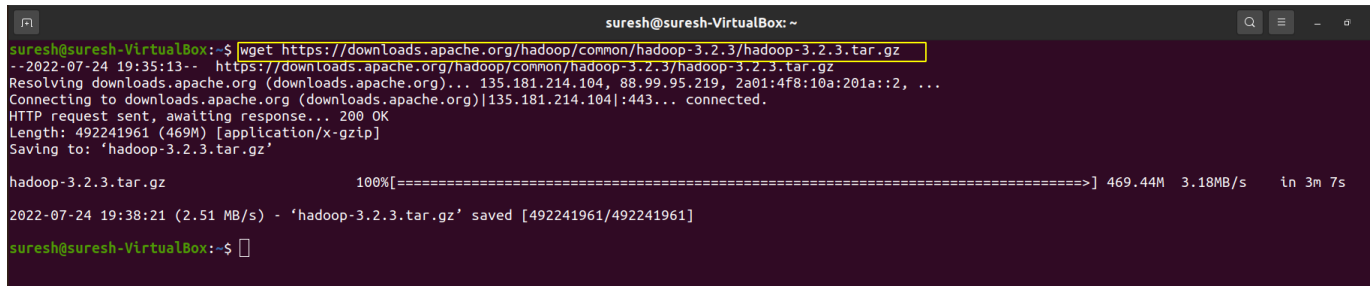


```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ ssh localhost  
The authenticity of host 'localhost (127.0.0.1)' can't be established.  
ECDSA key fingerprint is SHA256:FXatpiTt31Hxv22BBtPP3QLTdjC+hYLV0NeEgX9MMNg.  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.  
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.15.0-41-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
432 updates can be installed immediately.  
278 of these updates are security updates.  
To see these additional updates run: apt list --upgradable  
  
Your Hardware Enablement Stack (HWE) is supported until April 2025.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
suresh@suresh-VirtualBox:~$
```

Task 5: Downloading and configuring Hadoop

Step 1: Downloading Hadoop version 3.2.3 zip file

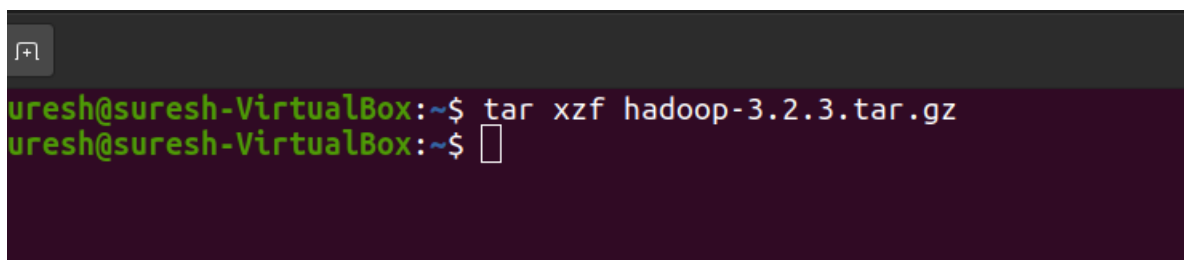
Command: `wget https://downloads.apache.org/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz`



```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ wget https://downloads.apache.org/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz  
--2022-07-24 19:35:13-- https://downloads.apache.org/hadoop/common/hadoop-3.2.3/hadoop-3.2.3.tar.gz  
Resolving downloads.apache.org (downloads.apache.org)... 135.181.214.104, 88.99.95.219, 2a01:4f8:10a:201a::2, ...  
Connecting to downloads.apache.org (downloads.apache.org)|135.181.214.104|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 492241961 (469M) [application/x-gzip]  
Saving to: 'hadoop-3.2.3.tar.gz'  
  
hadoop-3.2.3.tar.gz          100%[=====>] 469.44M  3.18MB/s   in 3m 7s  
  
2022-07-24 19:38:21 (2.51 MB/s) - 'hadoop-3.2.3.tar.gz' saved [492241961/492241961]  
  
suresh@suresh-VirtualBox:~$
```

Step 2: Extracting file of hadoop

Command: `tar xzf hadoop-3.2.3.tar.gz`



```
uresh@suresh-VirtualBox:~$ tar xzf hadoop-3.2.3.tar.gz  
uresh@suresh-VirtualBox:~$
```

Step 3: Configuring Hadoop

You have to configure 6 files for installing Hadoop, That are :

1. **.bashrc** : This files contain environment variables as we have in windows
2. **hadoop-env.sh** : Its serve as a master file to configure different files
3. **core-site.xml** : Its contain properties of HDFS and Hadoop core
4. **hdfs-site.xml** : Its contain the location for storing node metadata
5. **mapred-site.xml** : Its contain the MapReduce configuration

6. **yarn-site.xml** : Its contain Node manager and Resource manager configuration

Configuring Files:

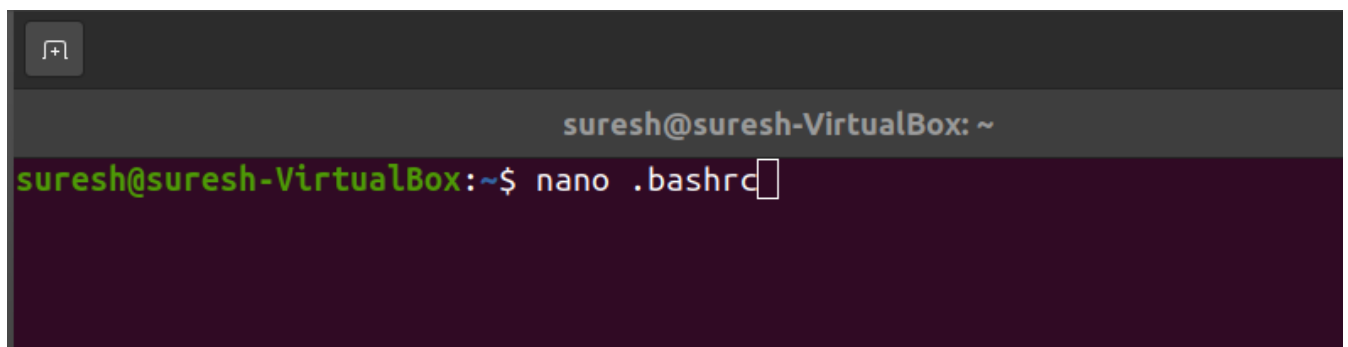
Note:

- 1) Enter your Ubuntu user name in place of 'suresh'.
- 2) To save and exit file from nano editor user 'ctrl+o' to save the file and 'ctrl+x' to exit the file

1) .bashrc

Step 1: To open '.bashrc' file in nano editor

Command : nano .bashrc



```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ nano .bashrc
```

Step 2: Copy the following data to the end of the .bashrc file


```
suresh@suresh-VirtualBox: ~
GNU nano 4.8 .bashrc
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -aLF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error)" "$(history|tail -n 1|sed -n -e s/^ *[0-9]* //g)"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.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
```

#Add below lines in this file

#Hadoop Related Options

export HADOOP_HOME=/home/suresh/hadoop-3.2.3

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"**

Note:

Enter your Ubuntu user name in place of 'suresh'.

```
suresh@suresh-VirtualBox: ~
GNU nano 4.8 .bashrc

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.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/suresh/hadoop-3.2.3
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      M-U Undo
^X Exit          ^R Read File    ^_ Replace      ^U Paste Text   ^T To Spell     ^_ Go To Line    M-E Redo
```

Save **(Ctrl+O)** then Press the Enter button and Exit **(Ctrl+X)** the file.

Step 3: Apply the changes to the running environment variable

Command: source ~/.bashrc

```
suresh@suresh-VirtualBox: ~
suresh@suresh-VirtualBox:~$ nano .bashrc
suresh@suresh-VirtualBox:~$ source ~/.bashrc
suresh@suresh-VirtualBox:~$
```

2) hadoop-env.sh

Step 1: Retrieve javac file location

Command: which javac

Step 2: Copy the path you get as an output

Step 3: Retrieving openjdk-8 file location

Command: readlink -f /usr/bin/javac

Step 4: Copy the path you get as an output from starting to '....amd64'

Step 5: Open hadoop-env.sh file using nano editor to configure it.

Command: nano \$HADOOP_HOME/etc/hadoop/hadoop-env.sh

```
suresh@suresh-VirtualBox: ~  
suresh@suresh-VirtualBox:~$ which javac  
/usr/bin/javac  
suresh@suresh-VirtualBox:~$ readlink -f /usr/bin/javac  
/usr/lib/jvm/java-8-openjdk-amd64/bin/javac  
suresh@suresh-VirtualBox:~$ nano $HADOOP_HOME/etc/hadoop/hadoop-env.sh
```

Step 6: In hadoop-env.sh file, look for 'export JAVA_HOME=' text in it and remove the '#' symbol from the front of line and paste the copied path to it after

'=' sign.



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

```
suresh@suresh-VirtualBox: ~
GNU nano 4.8 /home/suresh/hadoop-3.2.3/etc/hadoop/
##
## {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/java/testing hdfs dfs -ls
#
# 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
#
# The java implementation to use.  By default, this environment
# variable is REQUIRED on ALL platforms except OS X!
export JAVA_HOME=
# Location of Hadoop.  By default, Hadoop will attempt to determine
# this location based upon its execution path.
^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
```

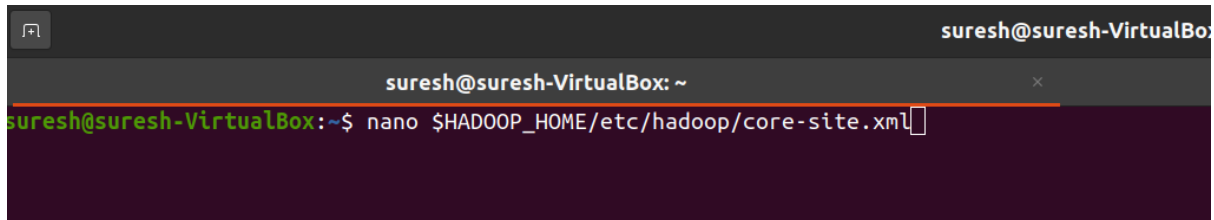
```
suresh@suresh-VirtualBox: ~
GNU nano 4.8 /home/suresh/hadoop-3.2.3/etc/hadoop/
# are configured for substitution and not append.  If append
# is preferable, modify this file accordingly.
###
# Generic settings for HADOOP
###
# Technically, the only required environment variable is JAVA_HOME.
# All others are optional.  However, the defaults are probably not
# preferred.  Many sites configure these options outside of Hadoop,
# such as in /etc/profile.d
#
# 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
# /etc/profile.d or equivalent.  Some options (such as
# --config) may react strangely otherwise.
#
# export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop
#
# The maximum amount of heap to use (Java -Xmx).  If no unit
```

Save **(Ctrl+O)** then Press the Enter button and Exit **(Ctrl+X)** the file.

3) core-site.xml

Step 1: Open core-site.xml file in nano editor

Command: nano \$HADOOP_HOME/etc/hadoop/core-site.xml

A terminal window titled 'suresh@suresh-VirtualBox' showing the command 'nano \$HADOOP_HOME/etc/hadoop/core-site.xml' being entered at the prompt 'suresh@suresh-VirtualBox:~\$'.

Step 2: Copy the following text given below between **<configuration>** and **</configuration>** tags



```
<property>

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

  <value>/home/suresh/tmpdata</value>

</property>

<property>

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

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

</property>
```

Note:

Enter your Ubuntu user name in place of 'suresh'.

```

suresh@suresh-VirtualBox: ~
GNU nano 4.8 /home/suresh/hado
<?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 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>hadoop.tmp.dir</name>
  <value>/home/suresh/tmpdata</value>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://127.0.0.1:9000</value>
</property>
</configuration>

```



Save **(Ctrl+O)** then Press the Enter button and Exit **(Ctrl+X)** the file.

4) hdfs-site.xml

Step 1: Open hdfs-site.xml file in nano editor

Command: nano \$HADOOP_HOME/etc/hadoop/hdfs-site.xml

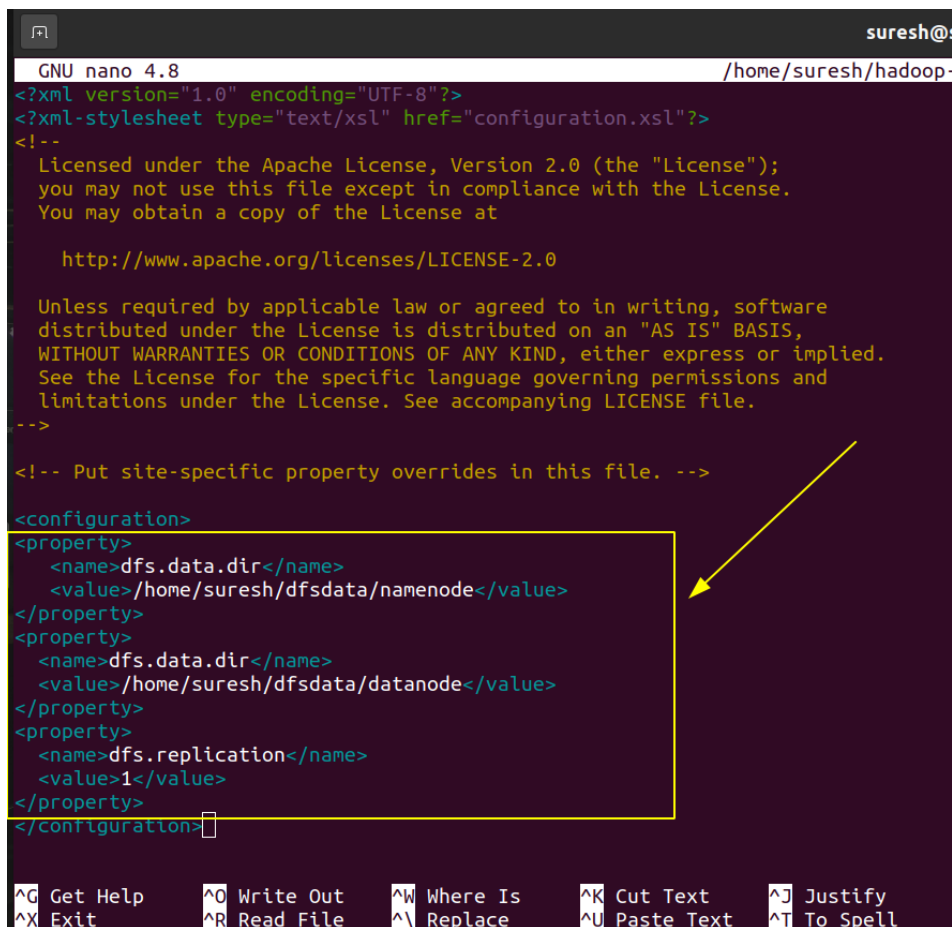
```

suresh@suresh-VirtualBox: ~
suresh@suresh-VirtualBox: ~$ nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

```

Step 2: Copy the following text given below between <configuration> and </configuration> tags

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



```
GNU nano 4.8 /home/suresh/hadoop-
<?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 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>dfs.data.dir</name>
  <value>/home/suresh/dfsdata/namenode</value>
</property>
<property>
  <name>dfs.data.dir</name>
  <value>/home/suresh/dfsdata/datanode</value>
</property>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
</configuration>
```

Save **(Ctrl+O)** then Press the Enter button and Exit **(Ctrl+X)** the file.

Step 3: Creating of dfsdata, dfsdata/namenode and dfsdata/datanode directories

Command 1: mkdir dfsdata

Command 2: cd dfsdata/

Command 3: mkdir namenode

Command 4: mkdir datanode

```
suresh@suresh-VirtualBox:~$ mkdir dfsdata
suresh@suresh-VirtualBox:~$ cd dfsdata/
suresh@suresh-VirtualBox:~/dfsdata$ mkdir namenode
suresh@suresh-VirtualBox:~/dfsdata$ mkdir datanode
suresh@suresh-VirtualBox:~/dfsdata$
```

5. mapred-site.xml

Step1: Open mapred-site.xml in nano editor

Command: nano \$HADOOP_HOME/etc/hadoop/mapred-site.xml

```
suresh@suresh-VirtualBox:~/dfsdata$ ls
datanode  namenode
suresh@suresh-VirtualBox:~/dfsdata$ nano $HADOOP_HOME/etc/hadoop/mapred-site.xml
```

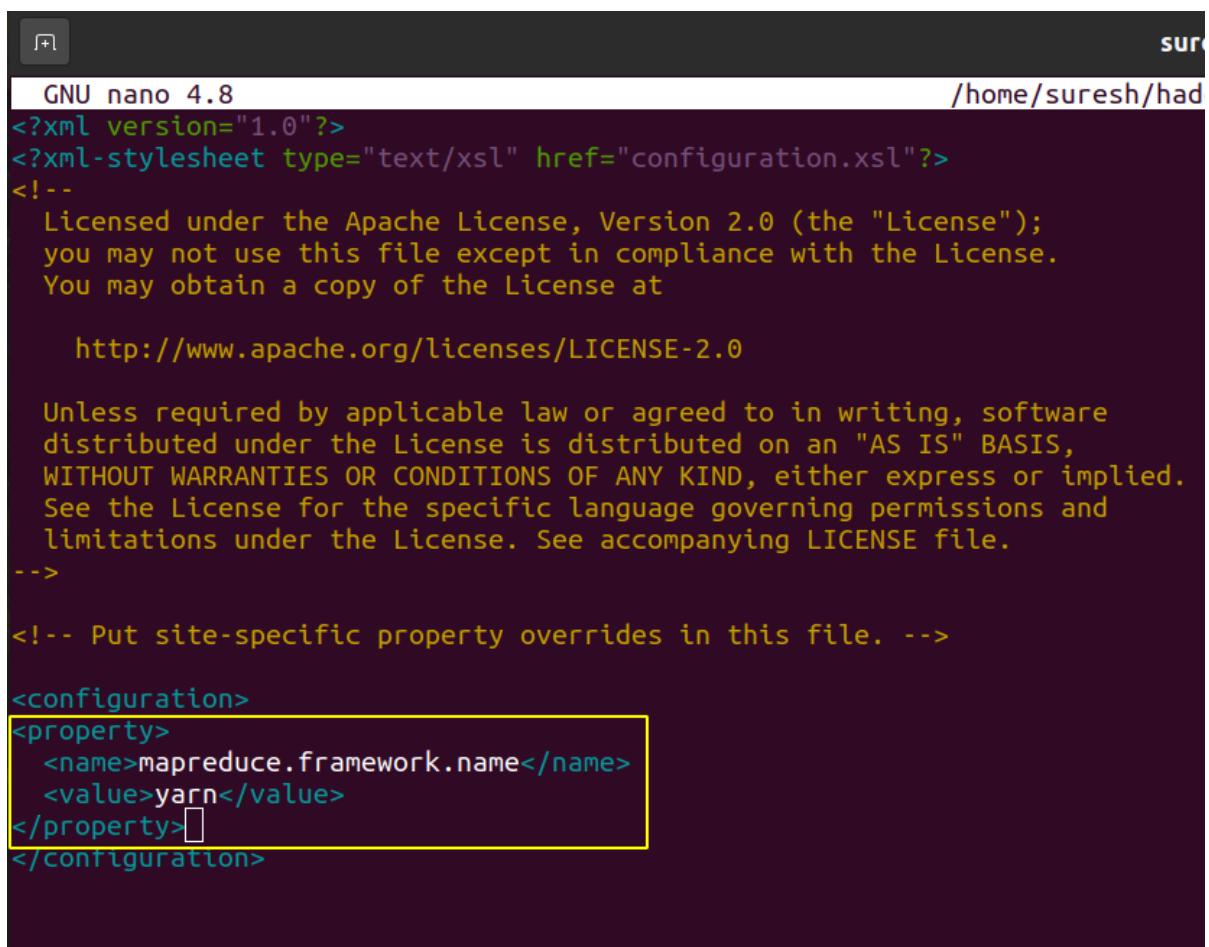

Step 2: Copy the following text between the `<configuration>` and `</configuration>` tags

`<property>`

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

`<value>yarn</value>`

`</property>`

A screenshot of a terminal window with the nano 4.8 text editor. The editor shows an XML configuration file. The content includes an XML declaration, a stylesheet reference, a license notice for Apache License 2.0, and a configuration block. The configuration block contains a property for 'mapreduce.framework.name' with the value 'yarn'. This property block is highlighted with a yellow rectangle. The cursor is positioned at the end of the property block.

```
GNU nano 4.8 /home/suresh/had
<?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 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>
```

Save **(Ctrl+O)** then Press the Enter button and Exit **(Ctrl+X)** the file.

6. yarn-site.xml

Step 1: Open yarn-site.xml in nano editor

Command: nano \$HADOOP_HOME/etc/hadoop/yarn-site.xml

```
suresh@suresh-VirtualBox:~/dfsdata$ nano $HADOOP_HOME/etc/hadoop/yarn-site.xml
```

Step 2: Copy the following text between the <configuration> and </configuration> tags

```
<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,H
ADOOP_CONF_DIR,CLASSPATH_PERPEND_DISTCACHE,HADOOP_YARN_
HOME,HADOOP_MAPRED_HOME</value>
</property>
```

```

suresh@suresh-VirtualBox: ~
GNU nano 4.8 /home/suresh/hadoop-3.2.3/etc/hadoop/yarn-site.xml
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 implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE 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>
<!-- Site specific YARN configuration properties -->

```

Save **(Ctrl+O)** then Press the Enter button and Exit **(Ctrl+X)** the file.

-----All configuration of files done-----

Now Format HDFS NameNode

Make sure to format namenode in
dfsdata folder

Command : hdfs namenode –format

```
suresh@suresh-VirtualBox:~$ hdfs namenode -format
```

```

suresh@suresh-VirtualBox: ~
2022-07-25 00:50:51,318 INFO util.GSet: 0.029999999329447746% max memory 2.3 GB = 720.4 KB
2022-07-25 00:50:51,318 INFO util.GSet: capacity = 2^16 = 65536 entries
2022-07-25 00:50:51,353 INFO namenode.FSImage: Allocated new BlockPoolId: BP-554215607-127.0.1.1-1658690451344
2022-07-25 00:50:51,377 INFO common.Storage: Storage directory /home/suresh/tmpdata/dfs/name has been successfully formatted.
2022-07-25 00:50:51,403 INFO namenode.FSImageFormatProtobuf: Saving image file /home/suresh/tmpdata/dfs/name/current/fsimage.ckpt_000000000000000000 using no compression
2022-07-25 00:50:51,523 INFO namenode.FSImageFormatProtobuf: Image file /home/suresh/tmpdata/dfs/name/current/fsimage.ckpt_000000000000000000 of size 401 bytes saved in 0 seconds.
2022-07-25 00:50:51,535 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2022-07-25 00:50:51,567 INFO namenode.FSNamesystem: Stopping services started for active state
2022-07-25 00:50:51,567 INFO namenode.FSNamesystem: Stopping services started for standby state
2022-07-25 00:50:51,571 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2022-07-25 00:50:51,571 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at suresh-VirtualBox/127.0.1.1
*****/

```

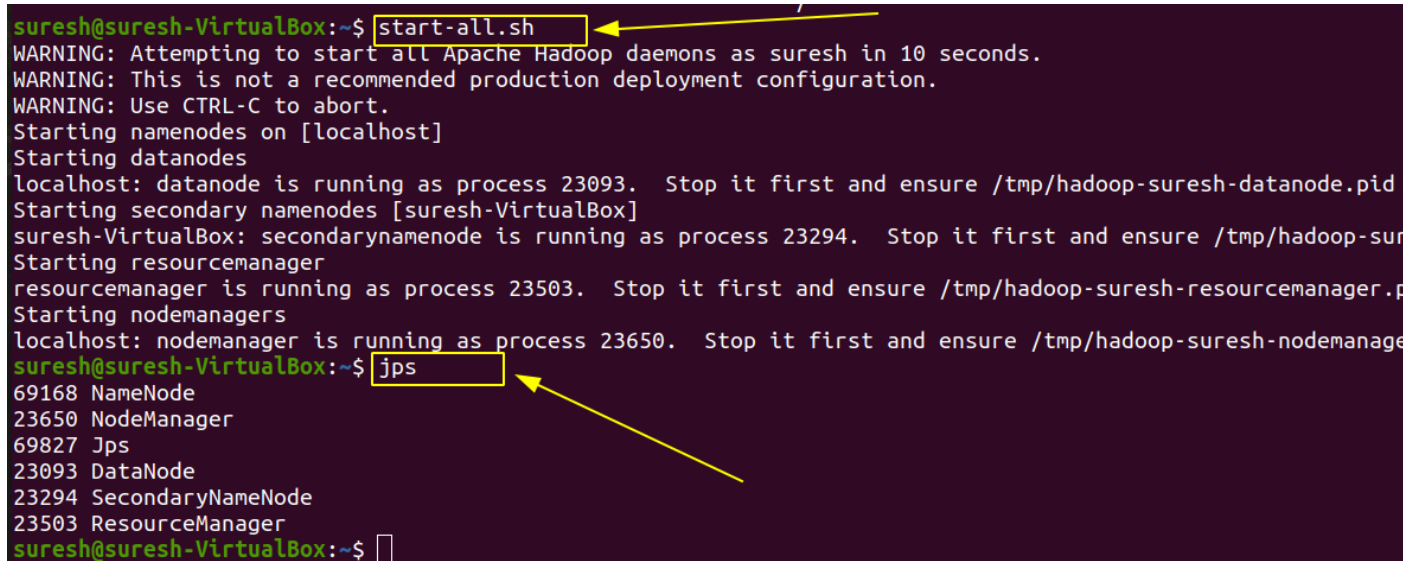
Start all the Services

Step 1: start all the .sh service

Command 1: start-all.sh

Step 2: Check all the services started

Command 2: jps



```
suresh@suresh-VirtualBox:~$ start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as suresh in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
localhost: datanode is running as process 23093. Stop it first and ensure /tmp/hadoop-suresh-datanode.pid
Starting secondary namenodes [suresh-VirtualBox]
suresh-VirtualBox: secondarynamenode is running as process 23294. Stop it first and ensure /tmp/hadoop-suresh-sec
Starting resourcemanager
resourcemanager is running as process 23503. Stop it first and ensure /tmp/hadoop-suresh-resourcemanager.p
Starting nodemanagers
localhost: nodemanager is running as process 23650. Stop it first and ensure /tmp/hadoop-suresh-nodemanager
suresh@suresh-VirtualBox:~$ jps
69168 NameNode
23650 NodeManager
69827 Jps
23093 DataNode
23294 SecondaryNameNode
23503 ResourceManager
suresh@suresh-VirtualBox:~$
```

c. Checking for NameNode, DataNode and Hadoop are working on browser

Step1) Open Browser and open 3 tabs and type following url in different tabs

- 1) <http://localhost:9870> // For NameNode
- 2) <http://localhost:9864> //For DataNode
- 3) <http://localhost:8088> //For Hadoop Working

If all tabs look same as shown below following pictures, then all thing is working properly

Namenode information × DataNode Information × All Applications × +

← → ↻ localhost:9870/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities ▾

Overview 'localhost:9000' (active)

Started:	Mon Jul 25 00:52:14 +0530 2022
Version:	3.2.3, rabe5358143720085498613d399be3bbf01e0f131
Compiled:	Sun Mar 20 06:48:00 +0530 2022 by ubuntu from branch-3.2.3
Cluster ID:	CID-dd6f35ef-9339-4159-a3b8-b4621764299f
Block Pool ID:	BP-554215607-127.0.1.1-1658690451344

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 174.72 MB of 296 MB Heap Memory. Max Heap Memory is 2.29 GB.

Namenode information × DataNode Information × All Applications × +

← → ↻ localhost:9864/datanode.html ☆

Hadoop Overview Utilities ▾

DataNode on suresh-VirtualBox:9866

Cluster ID:	CID-dd6f35ef-9339-4159-a3b8-b4621764299f
Version:	3.2.3, rabe5358143720085498613d399be3bbf01e0f131

Block Pools

Namenode Address	Block Pool ID	Actor State	Last Heartbeat	Last Block Report	Last Block Report Size (Max Size)
localhost:9000	BP-554215607-127.0.1.1-1658690451344	RUNNING	1s	21 minutes	0 B (64 MB)


Volume Information

Namenode information

DataNode Information

All Applications

localhost:8088/cluster



All Applications

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW_SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Used Resources	Total Resources	Reserved
0	0	0	0	0	<memory:0 B, vCores:0>	<memory:8 GB, vCores:8>	<memory:0 B, vCores:0>

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes
1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation	Maximum Resource
Capacity Scheduler	[memory-mb (unit=Mb), vcores]	<memory:1024, vCores:1>	<memory:8192, vCores:4>	0

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCoers	Allocated Memory MB	Allocated GPUs	Reserved CPU VCoers	Reserved Memory MB
No data available in table																

Showing 0 to 0 of 0 entries

If all localhost is working means Hadoop is successfully installed.