HADOOP PSEUDO NODE SETUP

STEP 1 : Enable the network connection

To check if the connection is up or not

```
localhost login: root
Password:
Last failed login: Sun Aug 20 22:32:08 IST 2023 from localhost on ssh:notty
There were 3 failed login attempts since the last successful login.
Last login: Sun Aug 20 21:51:01 on tty1
[root@localhost ~]# nmcli con show
NAME   UUID                                  TYPE      DEVICE
ens33  d63ddcba-b44d-43c9-915f-504c9fe37d5d  ethernet  --
[root@localhost ~]# _
```

The connection is not up

To enable connection

```
[root@localhost ~]# nmcli con up ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/1)
[root@localhost ~]#
```

STEP 2 : check for IP address

```
[root@localhost ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:f3:a5:c2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.126.135/24 brd 192.168.126.255 scope global noprefixroute dynamic ens33
       valid_lft 1734sec preferred_lft 1734sec
    inet6 fe80::623:3b00:6092:fd5b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@localhost ~]#
```

STEP 3 : Installation of JDK

1) Download latest jdk
   http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
2) Extract tar file

```
jdk1.8.0_161/README.html
[root@localhost ~]# tar -xvf jdk-8u161-linux-x64.tar.gz
```

File is extracted

3) Moving the file into the directory /usr/local and renaming it as java

Now , the file is into the directory /usr/local with the name java

4) Setting the environment variable

Using the file /etc/profile

And writing into it

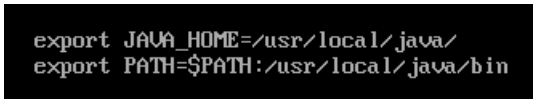
Write the following property into the file

Press i to insert content into file

Write the content at the end

Press esc to exit from file

Write :wq (i.e. write and quit) and press enter to save the changes



```
        pathmunge /usr/sbin after
    fi

    HOSTNAME=`/usr/bin/hostname 2>/dev/null`
    HISTSIZE=1000
    if [ "$HISTCONTROL" = "ignorespace" ] ; then
        export HISTCONTROL=ignoreboth
    else
        export HISTCONTROL=ignoredups
    fi

    export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

    # By default, we want umask to get set. This sets it for login shell
    # Current threshold for system reserved uid/gids is 200
    # You could check uidgid reservation validity in
    # /usr/share/doc/setup-*/uidgid file
    if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
        umask 002
    else
        umask 022
    fi

    for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
        if [ -r "$i" ]; then
            if [ "${-#*i}" != "$-" ]; then
                . "$i"
            else
                . "$i" >/dev/null
            fi
        fi
    done

    unset i
    unset -f pathmunge

    export JAVA_HOME=/usr/local/java/
    export PATH=$PATH:/usr/local/java/bin
```

The following statement shows that the file /etc/profile is written successfully

```
"/etc/profile" 88L, 2205C written
[root@localhost ~]#
```

STEP 4 : Check if JPS is running

```
[root@localhost ~]# source /etc/profile
[root@localhost ~]# jps
7213 Jps
[root@localhost ~]#
```

STEP 5 : Installing Apache Hadoop

1) Download Apache Hadoop tarball

https://archive.apache.org/dist/hadoop/core/hadoop-2.6.0/

2) Extracting the tar file

```
[root@localhost ~]# tar -xvf hadoop-2.7.2.tar.gz
```

The tar file is extracted

```
hadoop-2.7.2/share/hadoop/common/lib/jsch-0.1.42.jar
hadoop-2.7.2/share/hadoop/common/lib/stax-api-1.0-2.jar
hadoop-2.7.2/share/hadoop/common/lib/jackson-jaxrs-1.9.13.jar
hadoop-2.7.2/share/hadoop/common/lib/api-util-1.0.0-M20.jar
hadoop-2.7.2/share/hadoop/common/lib/jsp-api-2.1.jar
hadoop-2.7.2/share/hadoop/common/lib/httpclient-4.2.5.jar
hadoop-2.7.2/share/hadoop/common/lib/guava-11.0.2.jar
hadoop-2.7.2/share/hadoop/common/lib/zookeeper-3.4.6.jar
hadoop-2.7.2/share/hadoop/common/lib/commons-lang-2.6.jar
hadoop-2.7.2/share/hadoop/common/lib/xz-1.0.jar
hadoop-2.7.2/share/hadoop/common/lib/jackson-xc-1.9.13.jar
hadoop-2.7.2/share/hadoop/common/lib/hadoop-annotations-2.7.2.jar
hadoop-2.7.2/share/hadoop/common/lib/jaxb-api-2.2.2.jar
hadoop-2.7.2/share/hadoop/common/lib/jersey-json-1.9.jar
hadoop-2.7.2/share/hadoop/common/lib/protobuf-java-2.5.0.jar
hadoop-2.7.2/share/hadoop/common/lib/httpcore-4.2.5.jar
hadoop-2.7.2/share/hadoop/common/lib/avro-1.7.4.jar
hadoop-2.7.2/share/hadoop/common/lib/commons-beanutils-core-1.8.0.jar
hadoop-2.7.2/share/hadoop/common/lib/servlet-api-2.5.jar
hadoop-2.7.2/share/hadoop/common/lib/api-asn1-api-1.0.0-M20.jar
hadoop-2.7.2/share/hadoop/common/lib/gson-2.2.4.jar
hadoop-2.7.2/share/hadoop/common/lib/commons-cli-1.2.jar
hadoop-2.7.2/share/hadoop/common/lib/junit-4.11.jar
hadoop-2.7.2/share/hadoop/common/lib/jettison-1.1.jar
hadoop-2.7.2/share/hadoop/common/lib/jsr305-3.0.0.jar
hadoop-2.7.2/share/hadoop/common/lib/commons-logging-1.1.3.jar
hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar
hadoop-2.7.2/share/hadoop/common/lib/hamcrest-core-1.3.jar
hadoop-2.7.2/share/hadoop/common/lib/slf4j-api-1.7.10.jar
hadoop-2.7.2/share/hadoop/common/lib/commons-httpclient-3.1.jar
hadoop-2.7.2/share/hadoop/common/lib/commons-beanutils-1.7.0.jar
hadoop-2.7.2/share/hadoop/common/lib/paranamer-2.3.jar
hadoop-2.7.2/share/hadoop/common/hadoop-common-2.7.2-tests.jar
hadoop-2.7.2/share/hadoop/common/sources/
hadoop-2.7.2/share/hadoop/common/sources/hadoop-common-2.7.2-sources.jar
hadoop-2.7.2/share/hadoop/common/sources/hadoop-common-2.7.2-test-sources.jar
hadoop-2.7.2/lib/
hadoop-2.7.2/lib/native/
hadoop-2.7.2/lib/native/libhdfs.so
hadoop-2.7.2/lib/native/libhadooputils.a
hadoop-2.7.2/lib/native/libhdfs.so.0.0.0
hadoop-2.7.2/lib/native/libhadoop.so.1.0.0
hadoop-2.7.2/lib/native/libhadoop.a
hadoop-2.7.2/lib/native/libhdfs.a
hadoop-2.7.2/lib/native/libhadoop.so
hadoop-2.7.2/lib/native/libhadooppipes.a
hadoop-2.7.2/LICENSE.txt
[root@localhost ~]# _
```

3) Moving the extracted file into the directory /usr/local and renaming it as Hadoop

```
[root@localhost ~]# mv hadoop-2.7.2 /usr/local/hadoop
[root@localhost ~]#
```

4) Setting Environment variables

Write into the file /etc/profile using the vi command and write the properties

```
[root@localhost ~]# vi /etc/profile
```

```
    pathmunge /usr/sbin after
fi

HOSTNAME=`/usr/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

export JAVA_HOME=/usr/local/java/
export PATH=$PATH:/usr/local/java/bin

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/bin
```

STEP 6 : Set JAVA path in hadoop-env.sh

Go into the directory

/usr/local/Hadoop/etc/Hadoop

And write into the file hadoop-env.sh using vi command

```
[root@localhost ~]# cd /usr/local/hadoop/etc/hadoop/
[root@localhost hadoop]# vi hadoop-env.sh
```

Write the following property into the file and write and quit using :wq

```
export JAVA_HOME=/usr/local/java
```

STEP 7 : Configuring xml files

1)core-site.xml

```
[root@localhost hadoop]# cd /usr/local/hadoop/etc/hadoop/
[root@localhost hadoop]# vi core-site.xml
```

Write the following property

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

2) hdfs-site.xml

```
"core-site.xml" 24L, 863C written
[root@localhost hadoop]# vi hdfs-site.xml
```

Write the properties

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

<property>
<name>dfs.name.dir</name>
<value>file:///storage/name</value>
</property>

<property>
<name>dfs.data.dir</name>
<value>file:///storage/data</value>
</property>
</configuration>
```

3) mapred-site.xml

if the mapred-site.xml.template is present then rename it to mapres-site.xml

```
"hdfs-site.xml" 35L, 1016C written
[root@localhost hadoop]# mv mapred-site.xml.template mapred-site.xml
```

```
[root@localhost hadoop]# vi mapred-site.xml
```

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

</configuration>
```

4) yarn-site.xml

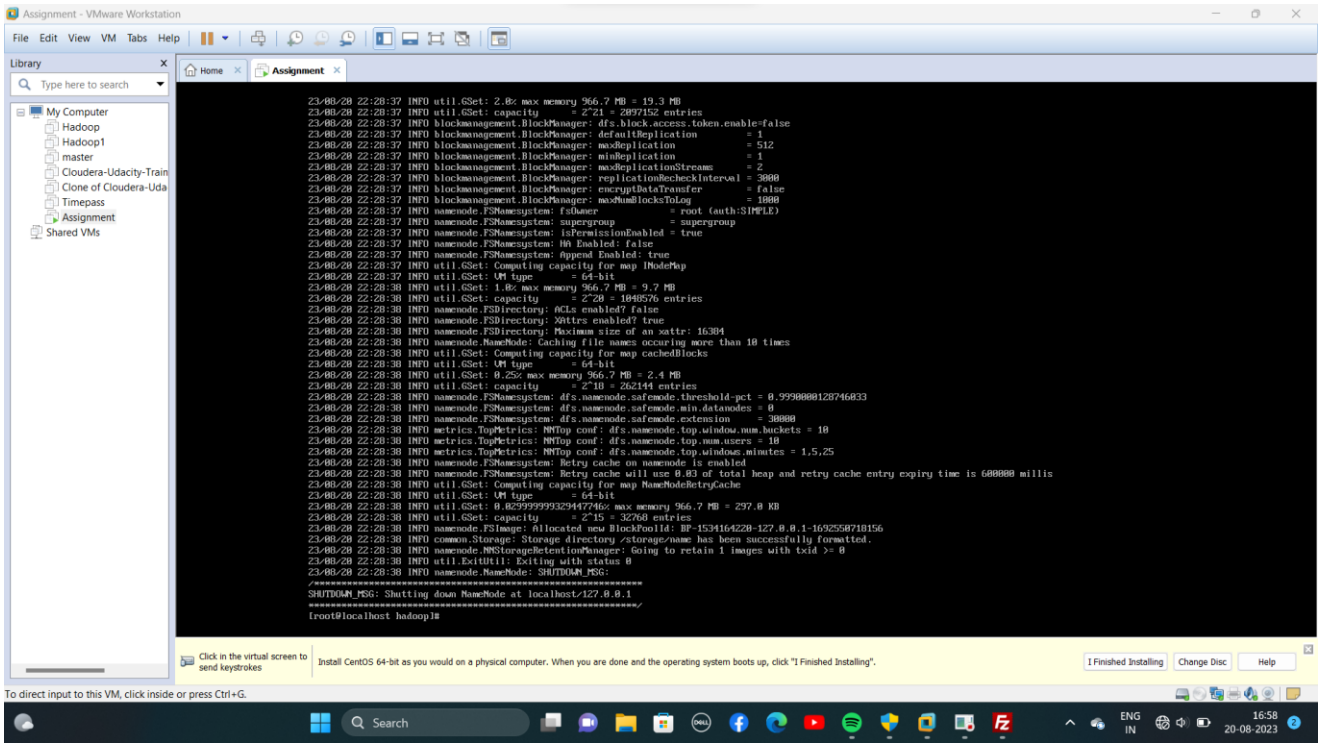
```
[root@localhost hadoop]# vi yarn-site.xml
-->
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>

</configuration>
```

STEP 8 : Formating Namenode

```
"yarn-site.xml" 23L, 789C written
[root@localhost hadoop]# hdfs namenode -format
```



The namenode is formatted without any error

STEP 9 : Starting the services

Go to sbin

```
[root@localhost hadoop]# cd /usr/local/hadoop/sbin/
```

STEP 10 : Starting dfs components

```
[root@localhost sbin]# start dfs.sh_
```

```
[root@localhost sbin]# sh start-dfs.sh
which: no start-dfs.sh in (/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/local/java/bin:/usr/local/java/bin:/usr/local/java/bin:/usr/local/hadoop/bin)
23/08/20 22:30:56 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:tSaXHr1CBIk+wQ93joVc0kzeMIwh9TniAS/Vb5zRFKU.
ECDSA key fingerprint is MD5:94:a1:b0:0f:29:38:77:f2:b0:41:1c:d1:aa:93:9e:67.
Are you sure you want to continue connecting (yes/no)? yes
localhost: Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
root@localhost's password: _
```

Checking if the services have started by typing jps on terminal

```
23/08/20 22:32:47 WARN util.NativeCodeLoader: Unable t
[root@localhost sbin]# jps
8226 SecondaryNameNode
8075 DataNode
7790 NameNode
8335 Jps
[root@localhost sbin]#
```

The services has started successfully

STEP 11 : Starting yarn components

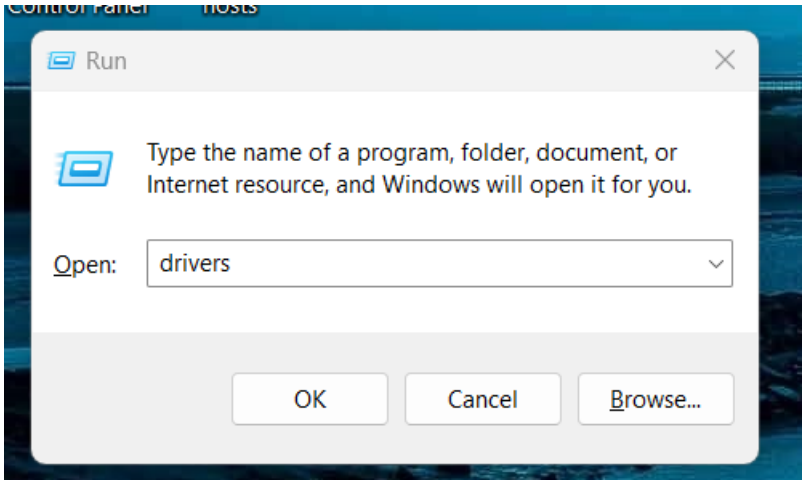
```
[root@localhost sbin]# sh start-yarn.sh
```

```
[root@localhost sbin]# sh start-yarn.sh
starting yarn daemons
which: no start-yarn.sh in (/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin:/usr/local/java/bin:/usr/local/java/bin:/usr/local/java/bin:/usr/local/hadoop/bin)
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-root-resourcemanager-localhost.localdomain.out
root@localhost's password:
```

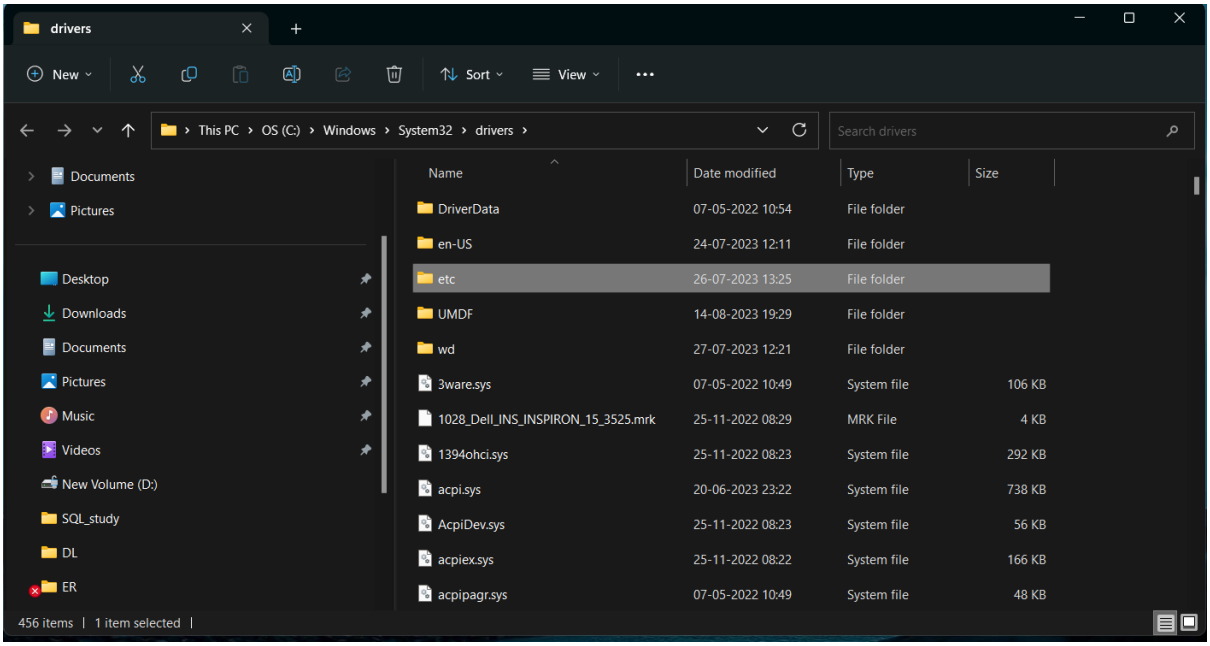
```
[root@localhost sbin]# jps
8226 SecondaryNameNode
8755 Jps
8663 NodeManager
8075 DataNode
8381 ResourceManager
7790 NameNode
[root@localhost sbin]#
```

STEP 12 : Accessing Hadoop GUI

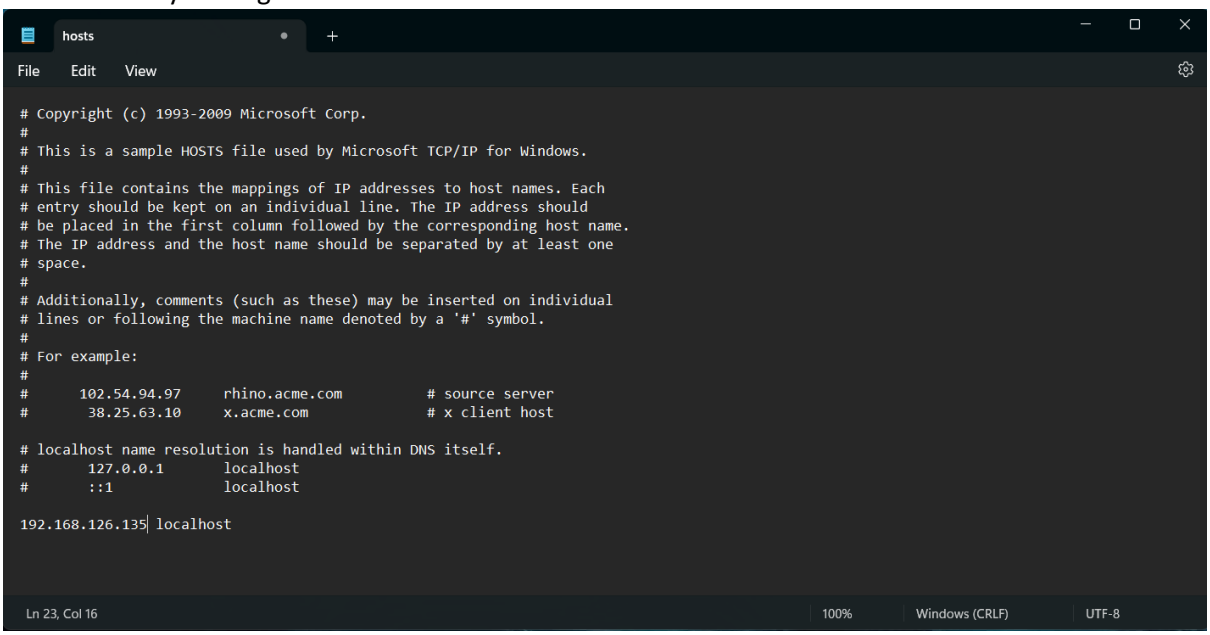
1) open windows Run Box and type drivers and press OK



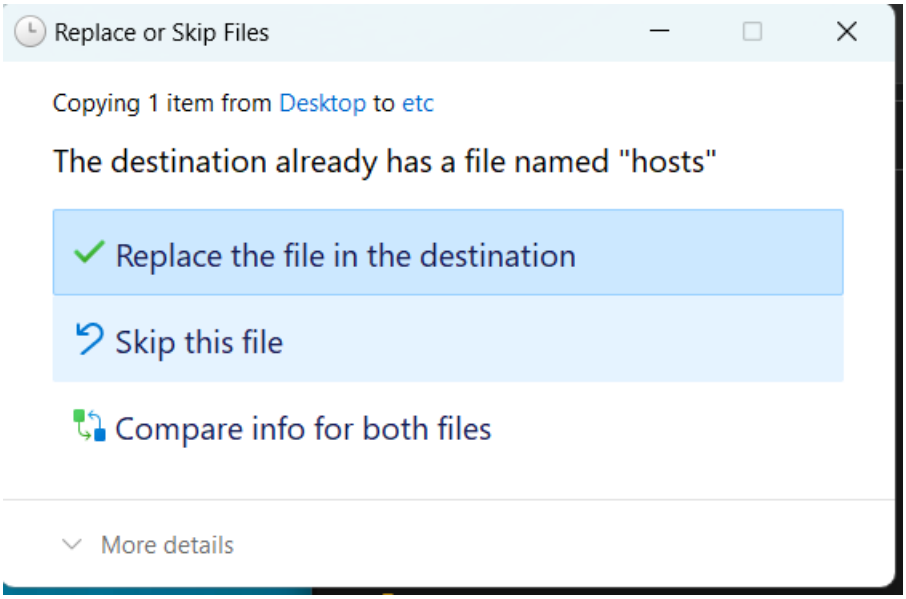
2) Go into the etc folder



3) Copy the hosts file to Desktop
4) Edit the file by writing machine's IP address



5) Copy the file and paste it into etc folder



STEP 13 : Disable Firewall

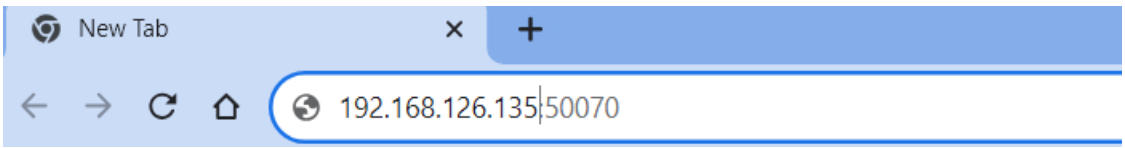
Check the status of firewall
If active then disable it

```
[root@localhost sbin]# sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2023-08-20 21:50:12 IST; 4h 40min left
     Docs: man:firewalld(1)
 Main PID: 6170 (firewalld)
   CGroup: /system.slice/firewalld.service
           └─6170 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

Aug 20 21:50:11 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Aug 20 21:50:12 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
[root@localhost sbin]#
```

```
[root@localhost sbin]# sudo systemctl stop firewalld
[root@localhost sbin]# sudo systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@localhost sbin]# _
```

STEP 14 : Type <ip address>:50070 in your browser



Hadoop GUI is accessed