PRACTICAL NO: 01

Aim: Write a program to Install, configure & run Hadoop and HDFS on Ubuntu (Basic).

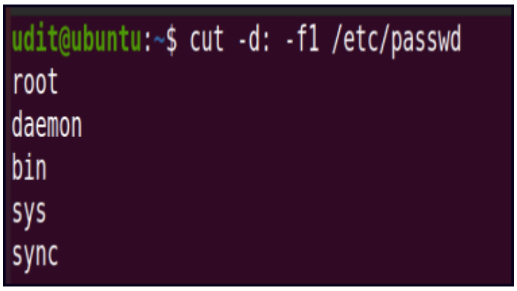
Pre-Requisites: An Ubuntu server VM with a user having sudo privileges

Code:

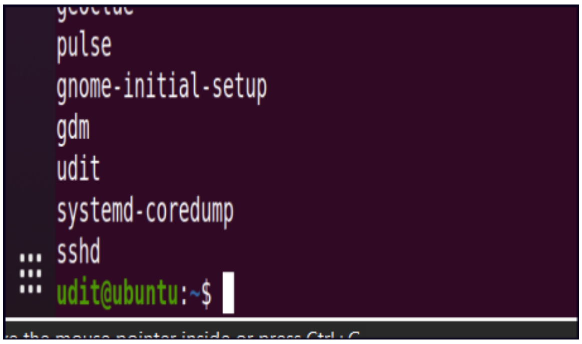
# Executed by Gulzarina Shaikh

# Check existing users on ubuntu

```
cut -d: -f1 /etc/passwd
```



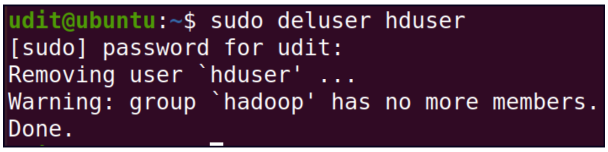
```
udit@ubuntu:~$ cut -d: -f1 /etc/passwd
root
daemon
bin
sys
sync
```



```
pulse
gnome-initial-setup
gdm
udit
systemd-coredump
sshd
udit@ubuntu:~$
```

# Remove hadoop user

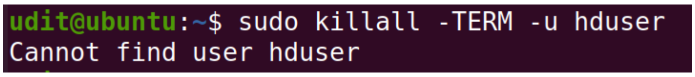
```
sudo deluser hduser
```



```
udit@ubuntu:~$ sudo deluser hduser
[sudo] password for udit:
Removing user `hduser' ...
Warning: group `hadoop' has no more members.
Done.
```

# ps aux | grep

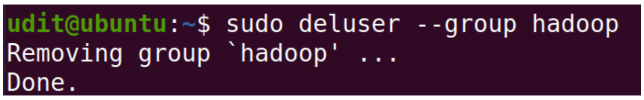
```
sudo killall -TERM -u hduser
```



```
udit@ubuntu:~$ sudo killall -TERM -u hduser
Cannot find user hduser
```

# Remove hadoop group

```
sudo deluser --group hadoop
```



```
udit@ubuntu:~$ sudo deluser --group hadoop
Removing group `hadoop' ...
Done.
```

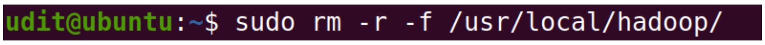
# Check presence of hadoop

# Go to location /usr/local/

# If you see a hadoop folder then hadoop installation was attempted and needs to be removed before fresh installation

# Remove hadoop

```
sudo rm -r -f /usr/local/hadoop/
```



```
udit@ubuntu:~$ sudo rm -r -f /usr/local/hadoop/
```

===

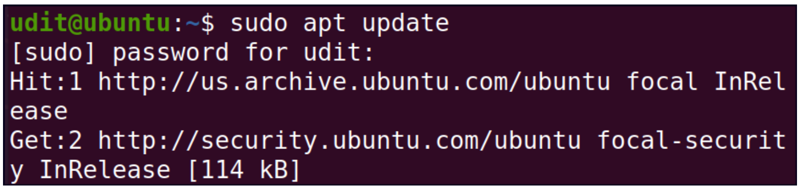
Step 1 — Installing Java
Step 2 — Installing Hadoop
Step 3 — Configuring Hadoop
Step 4 — Running Hadoop

==

Step 1 — Installing Java

#To get started, we'll update our package list:

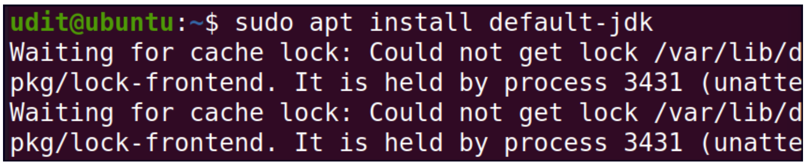
sudo apt update



```
udit@ubuntu:~$ sudo apt update
[sudo] password for udit:
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRel
ease
Get:2 http://security.ubuntu.com/ubuntu focal-securit
y InRelease [114 kB]
```

#Next, we'll install OpenJDK, the default Java Development Kit on Ubuntu 18.04:

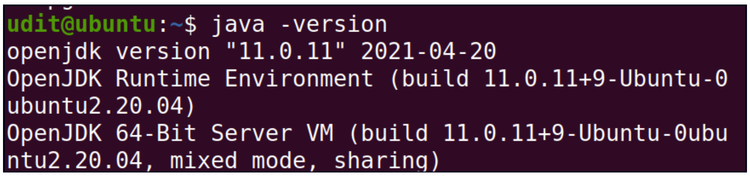
sudo apt install default-jdk



```
udit@ubuntu:~$ sudo apt install default-jdk
Waiting for cache lock: Could not get lock /var/lib/d
pkg/lock-frontend. It is held by process 3431 (unatte
Waiting for cache lock: Could not get lock /var/lib/d
pkg/lock-frontend. It is held by process 3431 (unatte
```

# Check folder for java installation at location -
# Java path -/usr/lib/jvm/java-11-openjdk-amd64
# Once the installation is complete, let's check the version.

java -version



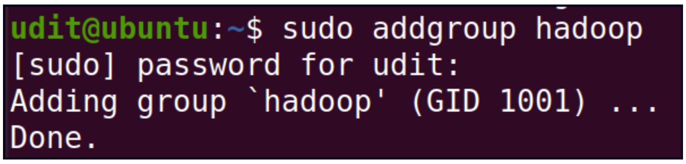
```
udit@ubuntu:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0
ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubu
ntu2.20.04, mixed mode, sharing)
```

This output verifies that OpenJDK has been successfully installed.

==

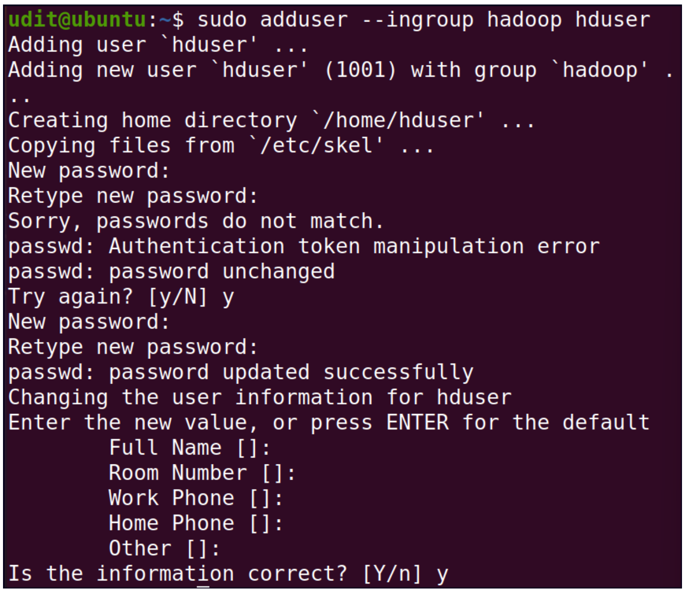
#Add new user to new user group
# group-hadoop, # user - hduser

sudo addgroup hadoop



```
udit@ubuntu:~$ sudo addgroup hadoop
[sudo] password for udit:
Adding group `hadoop' (GID 1001) ...
Done.
```

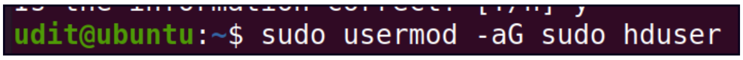
sudo adduser --ingroup hadoop hduser



```
udit@ubuntu:~$ sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' .
..
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

# Add new user to listed groups

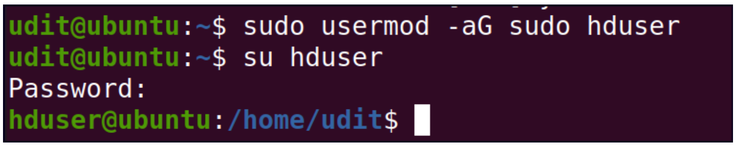
sudo usermod -aG sudo hduser



```
udit@ubuntu:~$ sudo usermod -aG sudo hduser
```

# Change to new user

su hduser



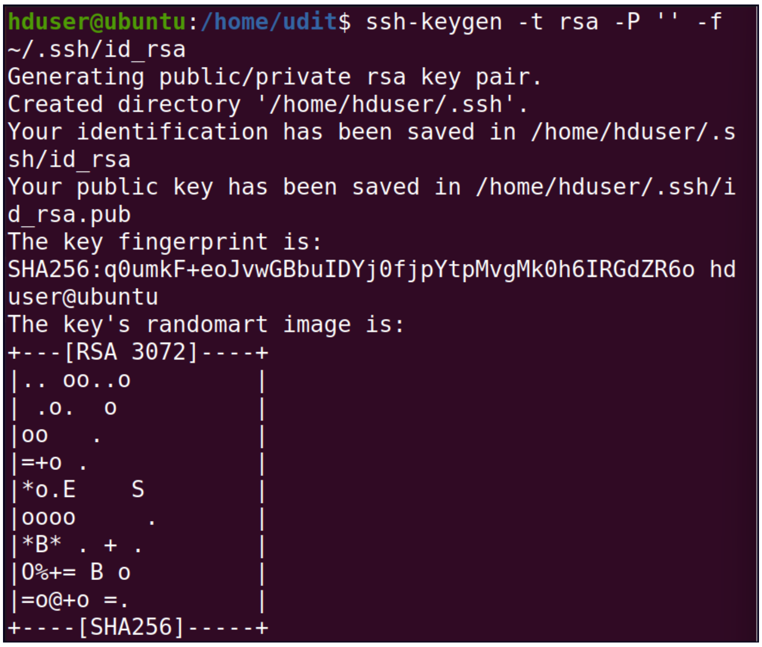
```
udit@ubuntu:~$ sudo usermod -aG sudo hduser
udit@ubuntu:~$ su hduser
Password:
hduser@ubuntu:/home/udit$ 
```

# The prompt shoud look like this - hduser@ubuntu:/

===

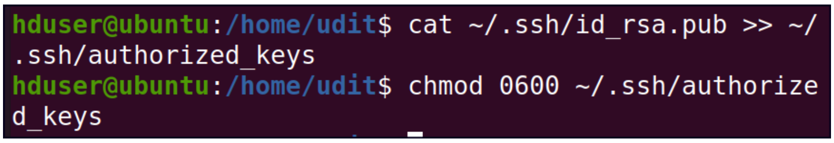
# Create a ssh keygen for the user.

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



```
hduser@ubuntu:/home/udit$ ssh-keygen -t rsa -P '' -f
~/.ssh/id_rsa
Generating public/private rsa key pair.
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.s
sh/id_rsa
Your public key has been saved in /home/hduser/.ssh/i
d_rsa.pub
The key fingerprint is:
SHA256:q0umkF+eoJvwGBbuIDYj0fjpYtpMvgMk0h6IRGdZR6o hd
user@ubuntu
The key's randomart image is:
+---[RSA 3072]----+
|.. oo..o         |
| .o.  o          |
|oo   .           |
|=+o .            |
|*o.E    S        |
|oooo     .       |
|*B* . + .        |
|O%+= B o         |
|=o@+o =.         |
+----[SHA256]-----+
```

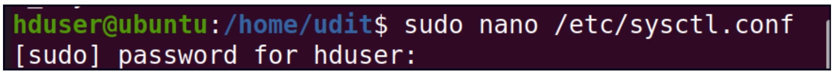
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys



```
hduser@ubuntu:/home/udit$ cat ~/.ssh/id_rsa.pub >> ~/
.ssh/authorized_keys
hduser@ubuntu:/home/udit$ chmod 0600 ~/.ssh/authorize
d_keys
```

# Disable IPv6.

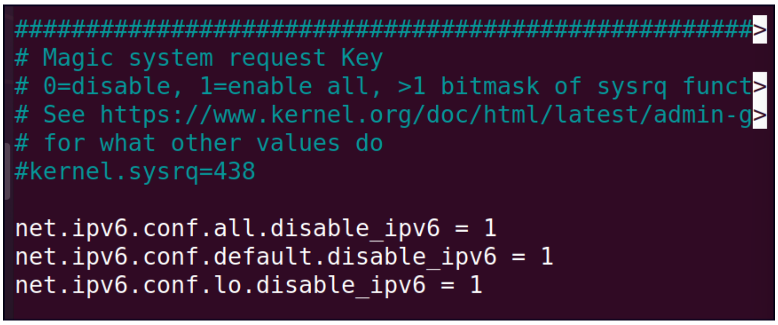
sudo nano /etc/sysctl.conf



```
hduser@ubuntu:/home/udit$ sudo nano /etc/sysctl.conf
[sudo] password for hduser:
```

#add the following lines to the end of the file

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1



```
####################################################>
# Magic system request Key
# 0=disable, 1=enable all, >1 bitmask of sysrq funct>
# See https://www.kernel.org/doc/html/latest/admin-g>
# for what other values do
#kernel.sysrq=438

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

===

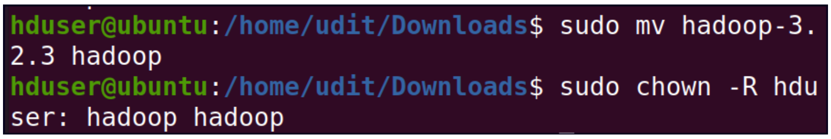
Step 2 — Downloading & Installing Hadoop

===

#Extract Hadoop and move to group hadoop

cd /usr/local

sudo tar xvf /home/udit/Downloads/hadoop-3.2.3.tar.gz
sudo mv hadoop-3.2.3 hadoop
sudo chown -R hduser: hadoop hadoop

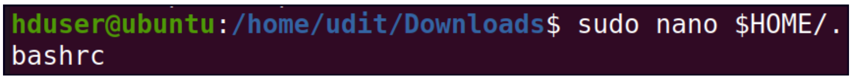


```
hduser@ubuntu:/home/udit/Downloads$ sudo mv hadoop-3.
2.3 hadoop
hduser@ubuntu:/home/udit/Downloads$ sudo chown -R hdu
ser: hadoop hadoop
```

===

#Now open $HOME/.bashrc

sudo nano $HOME/.bashrc

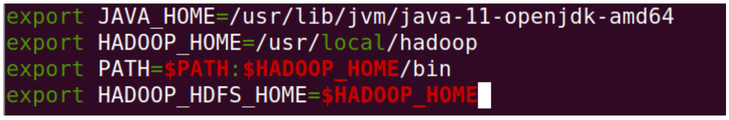


```
hduser@ubuntu:/home/udit/Downloads$ sudo nano $HOME/.
bashrc
```

# Add the following lines

export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop

```
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_HDFS_HOME=$HADOOP_HOME
```



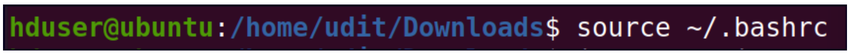
```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_HDFS_HOME=$HADOOP_HOME
```

# Save file - Ctrl + s

# Close file - Ctrl + x

# Run the following command to make changes through the .bashrc file.

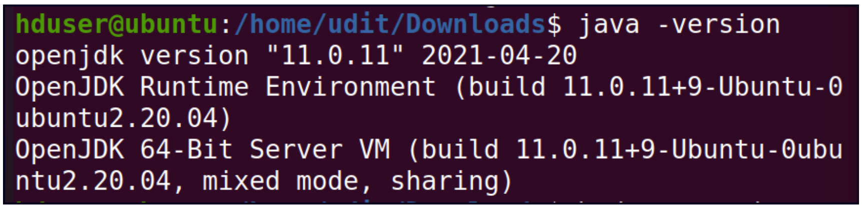
```
source ~/.bashrc
```



```
hduser@ubuntu:/home/udit/Downloads$ source ~/.bashrc
```

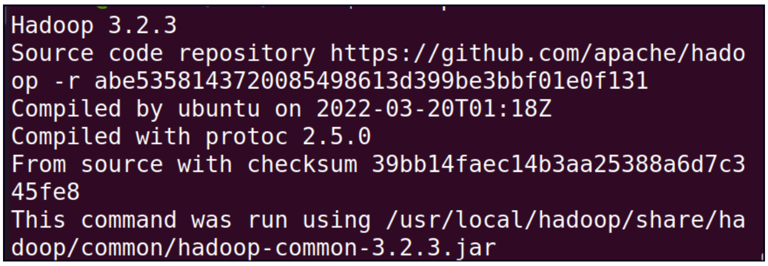
# Check version of java and hadoop

Command: java -version



```
hduser@ubuntu:/home/udit/Downloads$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0
ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubu
ntu2.20.04, mixed mode, sharing)
```

Command: hadoop version

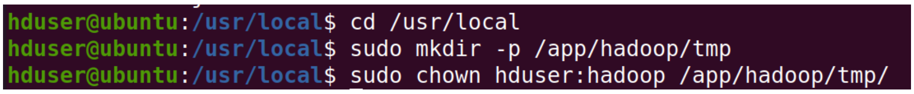


```
Hadoop 3.2.3
Source code repository https://github.com/apache/hado
op -r abe5358143720085498613d399be3bbf01e0f131
Compiled by ubuntu on 2022-03-20T01:18Z
Compiled with protoc 2.5.0
From source with checksum 39bb14faec14b3aa25388a6d7c3
45fe8
This command was run using /usr/local/hadoop/share/ha
doop/common/hadoop-common-3.2.3.jar
```

===

#Create a tmp folder in /app/hadoop/tmp and change the owner to hduser.

```
cd /usr/local
sudo mkdir -p /app/hadoop/tmp
sudo chown hduser:hadoop /app/hadoop/tmp/
```



```
hduser@ubuntu:/usr/local$ cd /usr/local
hduser@ubuntu:/usr/local$ sudo mkdir -p /app/hadoop/tmp
hduser@ubuntu:/usr/local$ sudo chown hduser:hadoop /app/hadoop/tmp/
```

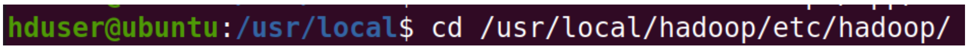
===

Step 3 — Configuring Hadoop

===

# Hadoop requires that you set the path to Java, either as an environment variable or in the Hadoop configuration file.hadoop-env.sh

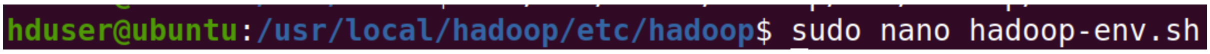
```
cd /usr/local/hadoop/etc/hadoop/
```



```
hduser@ubuntu:/usr/local$ cd /usr/local/hadoop/etc/hadoop/
```

#To Configure Hadoop’s Java Home, begin by opening hadoop-env.sh

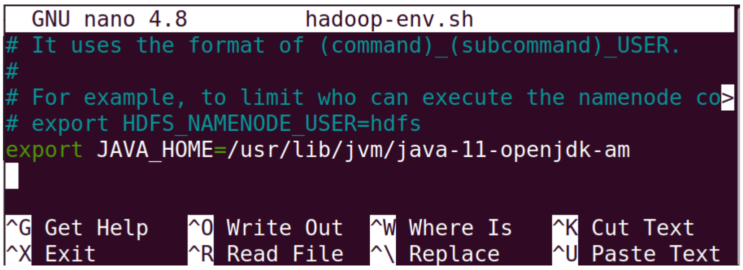
```
sudo nano hadoop-env.sh
```



```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano hadoop-env.sh
```

# Add the following line at the end of .sh file

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```

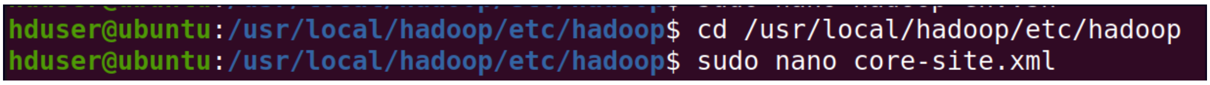


```
  GNU nano 4.8           hadoop-env.sh
# It uses the format of (command)_(subcommand)_USER.
#
# For example, to limit who can execute the namenode co>
# export HDFS_NAMENODE_USER=hdfs
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-am

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text
```

# Sve & Close

# Make the changes in core-site.xml file

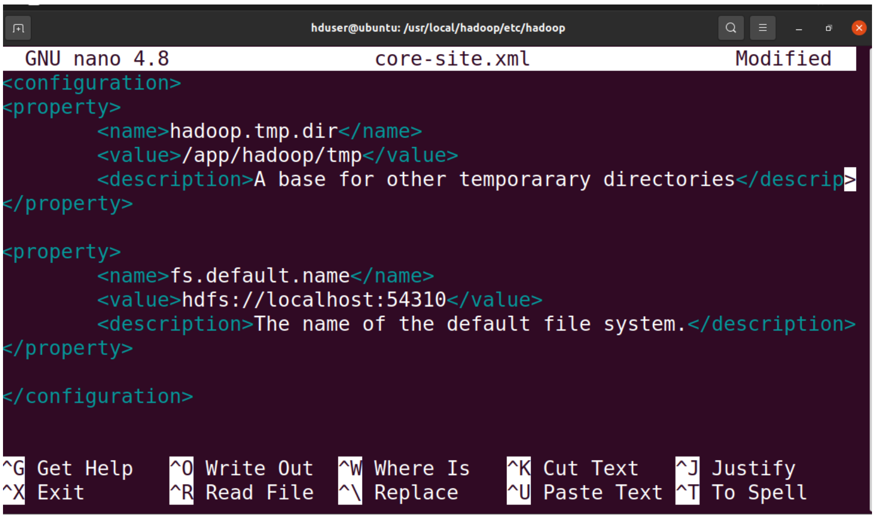
```
cd /usr/local/hadoop/etc/hadoop
sudo nano core-site.xml
```



```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ cd /usr/local/hadoop/etc/hadoop
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano core-site.xml
```

#Add the following lines

```
<property>
        <name>hadoop.tmp.dir</name>
        <value>/app/hadoop/tmp</value>
        <description>A base for other temporarary directories</description>
</property>

<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:54310</value>
        <description>The name of the default file system.</description>
</property>
```



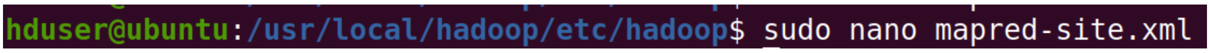
```
  GNU nano 4.8                  core-site.xml                    Modified
<configuration>
<property>
        <name>hadoop.tmp.dir</name>
        <value>/app/hadoop/tmp</value>
        <description>A base for other temporarary directories</descrip>
</property>

<property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:54310</value>
        <description>The name of the default file system.</description>
</property>

</configuration>

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

# save & Close

===

#Make the changes in mapred-site.xml

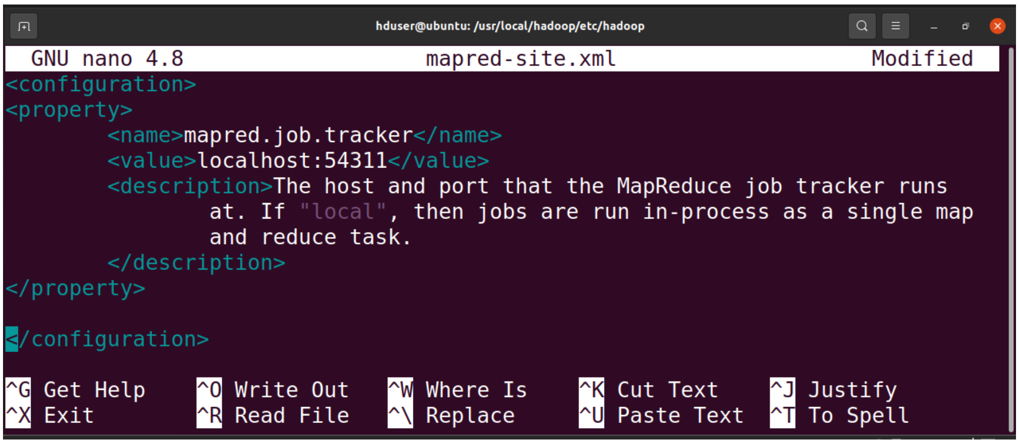
```
	sudo nano mapred-site.xml
```



```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano mapred-site.xml
```

#Add the following lines

```
	<property>
		<name>mapred.job.tracker</name>
		<value>localhost:54311</value>
		<description>The host and port that the MapReduce job tracker runs
			at. If "local", then jobs are run in-process as a single map
			and reduce task.
		</description>
	</property>
```



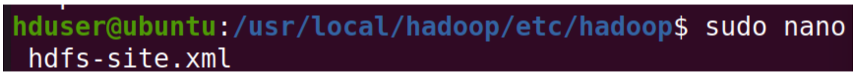
```
  GNU nano 4.8                  mapred-site.xml                   Modified
<configuration>
<property>
        <name>mapred.job.tracker</name>
        <value>localhost:54311</value>
        <description>The host and port that the MapReduce job tracker runs
                at. If "local", then jobs are run in-process as a single map
                and reduce task.
        </description>
</property>

</configuration>

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

# save & Close

===

#Make the changes in hdfs-site.xml

```
	sudo nano hdfs-site.xml
```

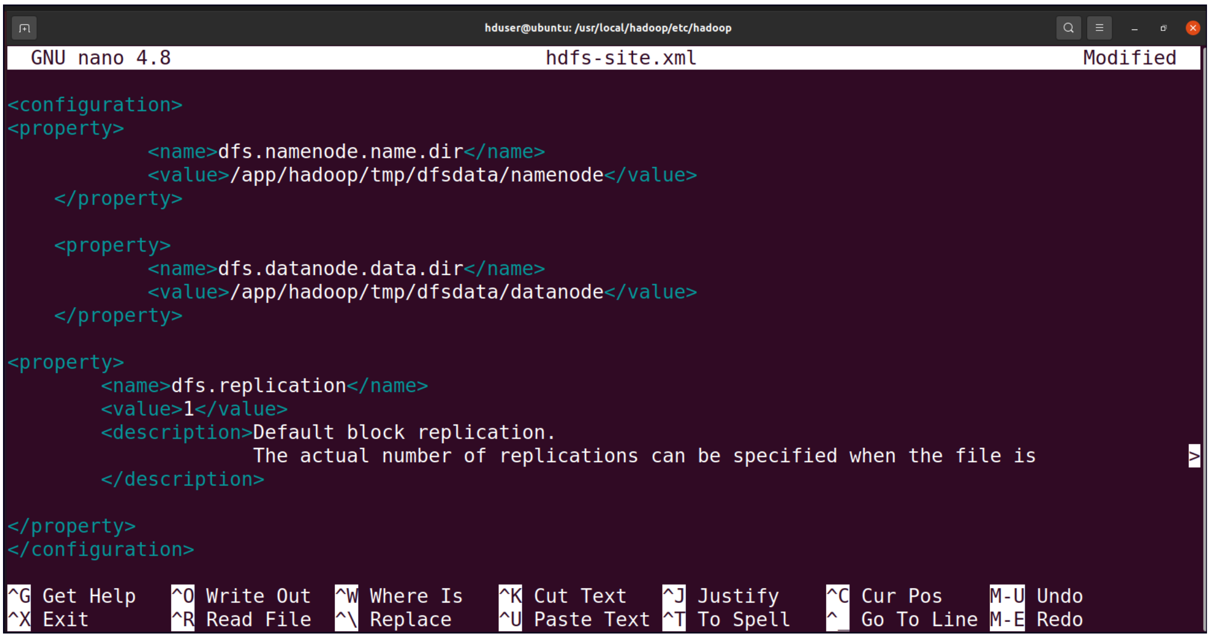


```
hduser@ubuntu:/usr/local/hadoop/etc/hadoop$ sudo nano
 hdfs-site.xml
```

#Add the following lines

```
	<property>
		<name>dfs.namenode.name.dir</name>
		<value>/app/hadoop/tmp/dfsdata/namenode</value>
	   </property>

	   <property>
		<name>dfs.datanode.data.dir</name>
		<value>/app/hadoop/tmp/dfsdata/datanode</value>
	   </property>
```

```
<property>
        <name>dfs.replication</name>
        <value>1</value>
        <description>Default block replication.
                The actual number of replications can be specified when the file is
                created. The default is used if replication is not specified in
                create time.
        </description>
</property>
```



hduser@ubuntu: /usr/local/hadoop/etc/hadoop

```
  GNU nano 4.8                          hdfs-site.xml                              Modified

<configuration>
<property>
            <name>dfs.namenode.name.dir</name>
            <value>/app/hadoop/tmp/dfsdata/namenode</value>
    </property>

    <property>
            <name>dfs.datanode.data.dir</name>
            <value>/app/hadoop/tmp/dfsdata/datanode</value>
    </property>

<property>
        <name>dfs.replication</name>
        <value>1</value>
        <description>Default block replication.
                     The actual number of replications can be specified when the file is            >
        </description>

</property>
</configuration>

^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 & Close

==

# Format namenode

```
hdfs namenode -format
hdfs datanode -format
```

==

# Step 4 - Running Hadoop

==

# To start hadoop, we need to start localhost

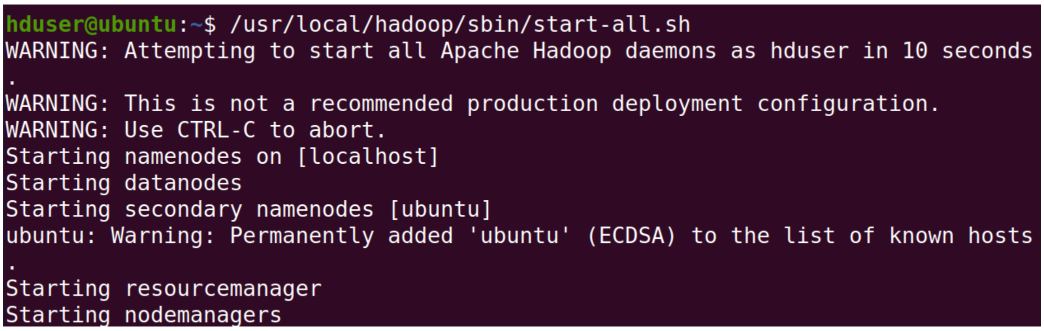
```
ssh localhost
```

# If connection is refused, it will result in error - run only if error & then run previous command

```
sudo apt-get install ssh
```

# Start all the hadoop services

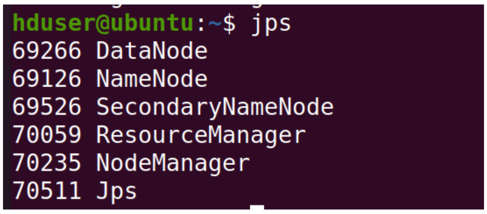
```
/usr/local/hadoop/sbin/start-all.sh
```



```
hduser@ubuntu:~$ /usr/local/hadoop/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds
.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
ubuntu: Warning: Permanently added 'ubuntu' (ECDSA) to the list of known hosts
.
Starting resourcemanager
Starting nodemanagers
```

# Check if that all hadoop services are running (6 services should appear)

jps



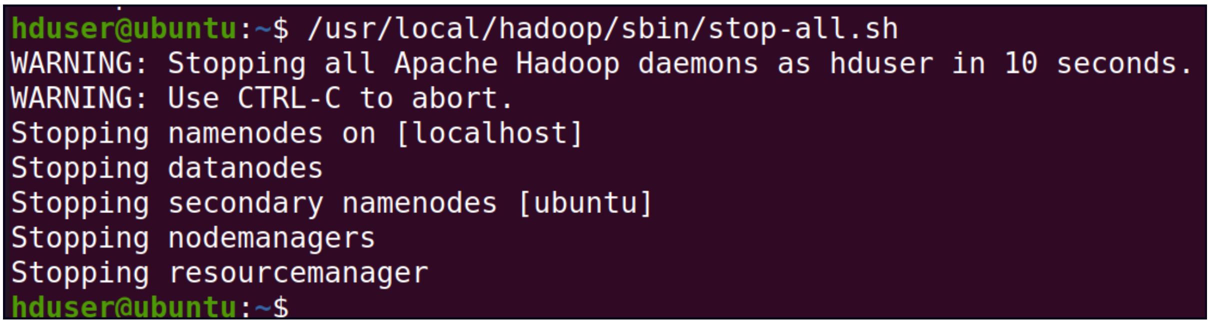
```
hduser@ubuntu:~$ jps
69266 DataNode
69126 NameNode
69526 SecondaryNameNode
70059 ResourceManager
70235 NodeManager
70511 Jps
```

# Access localhost:9870 to get namenode status, open browser and type

http://localhost:9870

# Stop all the hadoop services.

/usr/local/hadoop/sbin/stop-all.sh



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

Conclusion: The performed program to Install, configure & run Hadoop and HDFS on Ubuntu (Basic) of Hadoop on Ubuntu has been successfully demonstrated.