

Documentation For Hadoop Multi Node Cluster Setup

Yashpreet Singh (IIT2019124)

Aakash Bishnoi (IIT2019125)

Group No. 14

Follow the following steps :

DO THE FOLLOWING STEPS FOR BOTH MASTER AND SLAVE NODE

1. Install SSH using the following command:

```
sudo apt install ssh
```

2. Install PDSH using the following command:

```
sudo apt install pdsh
```

3. Open the .bashrc file with the following command:

```
sudo nano .bashrc
```

At the end of the file add the following line:

```
export PDSH_RCMD_TYPE=ssh
```

4. Now let's configure SSH. Let's create a new key using the following command:

```
ssh-keygen -t rsa -P ""
```

Just press Enter every time that is needed.

5. Now we need to copy the public key to the authorized_keys file with the following command:

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

6. Now we can verify the SSH configuration by connecting to the localhost:

```
ssh localhost
```

Just type “yes” and press Enter when needed.

```
hadoop@hadoop-master:~$ ssh localhost
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.13.0-27-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

1 device has a firmware upgrade available.
Run 'fwupdmgr get-upgrades' for more information.

560 updates can be installed immediately.
15 of these updates are security updates.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Fri Sep  9 04:51:44 2022 from 127.0.0.1
```

7. This is the step where we install Java 8. We use this command:

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

8. Download Hadoop using the following command:

```
sudo wget -P ~ https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz
```

If the above command doesn't work due to proxy, just manually download the hadoop binary version using the browser, just by clicking the above link.

9. Now, if go to the folder where this hadoop file is downloaded and apply the following command :

```
tar xzf hadoop-3.3.4.tar.gz
```

10. Now, change the `hadoop-3.3.4` folder name to `hadoop` (this makes it easier to use). Use this command:

```
mv hadoop-3.3.4 hadoop
```

11. Open the `hadoop-env.sh` file in the nano editor to edit `JAVA_HOME`. Make sure you are in the directory where `hadoop` folder is present.

```
nano hadoop/etc/hadoop/hadoop-env.sh
```

Paste this line to `JAVA_HOME`:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/
```

12. Change the `hadoop` folder directory to `/usr/local/hadoop`. This is the command (again in the same terminal):

```
sudo mv hadoop /usr/local/hadoop
```

13. Open the environment file on nano with this command:

```
sudo nano /etc/environment
```

Then, add the following configurations:

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"
```

14. Now we will add a user called `hadoop`, and we will set up its configurations:

```
sudo adduser hadoop
```

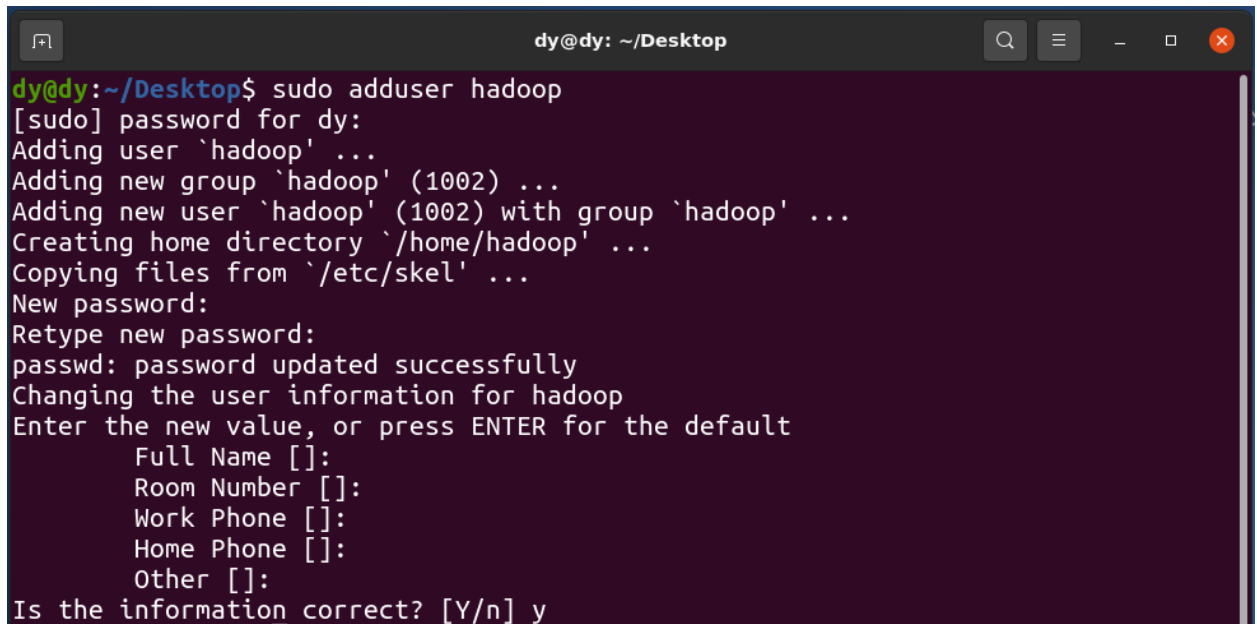
Provide the password and you can leave the rest blank, just press Enter.

Now type these commands:

```
sudo usermod -aG hadoop hadoop
```

```
sudo chown hadoop:root -R /usr/local/hadoop/
```

```
sudo chmod g+rx -R /usr/local/hadoop/  
sudo adduser hadoop sudo
```

A terminal window titled 'dy@dy: ~/Desktop' with a dark purple background. It shows the execution of 'sudo adduser hadoop'. The process includes prompts for a password, adding a new group 'hadoop' (1002), adding a new user 'hadoop' (1002) with that group, creating a home directory '/home/hadoop', and copying files from '/etc/skel'. It then prompts for a new password and its retype, followed by updating the password successfully. Finally, it prompts for user information (Full Name, Room Number, Work Phone, Home Phone, Other) and asks if the information is correct, with 'y' being entered.

```
dy@dy:~/Desktop$ sudo adduser hadoop  
[sudo] password for dy:  
Adding user `hadoop' ...  
Adding new group `hadoop' (1002) ...  
Adding new user `hadoop' (1002) with group `hadoop' ...  
Creating home directory `/home/hadoop' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for hadoop  
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
```

ONCE YOU HAVE ADDED THE USER, DO ALL THE FOLLOWING COMMANDS AFTER RUNNING THE FOLLOWING COMMAND (FOR BOTH THE MACHINES):

```
su hadoop
```

15. Open the .bashrc file with the following command:

```
sudo nano .bashrc
```

At the end of the file add the following lines:

```
#Hadoop Related Options  
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/sbin:$HADOOP_HOME/bin
```

```
export  
HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
```

```
# Hadoop  
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 HADOOP_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"  
  
export PDSH_RCMD_TYPE=ssh
```

Save the file.

It is vital to apply the changes to the current running environment by using the following command:

```
source ~/.bashrc
```

16. Now we will add the machine ip addresses into the hosts file for both the nodes. Use the following command to get the ip address of each machine.

MAKE SURE BOTH THE MACHINES ARE ON THE SAME NETWORK

```
hadoop@hadoop-master:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    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: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 3c:2c:30:ef:e9:c0 brd ff:ff:ff:ff:ff:ff  
    inet 172.19.19.237/22 brd 172.19.19.255 scope global dynamic noprefixroute enp2s0  
        valid_lft 84283sec preferred_lft 84283sec  
    inet6 fe80::2493:5f9c:d7dc:8f17/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: wlp3s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noqueue state DOWN group default qlen 1000  
    link/ether 48:5f:99:2b:37:cf brd ff:ff:ff:ff:ff:ff
```

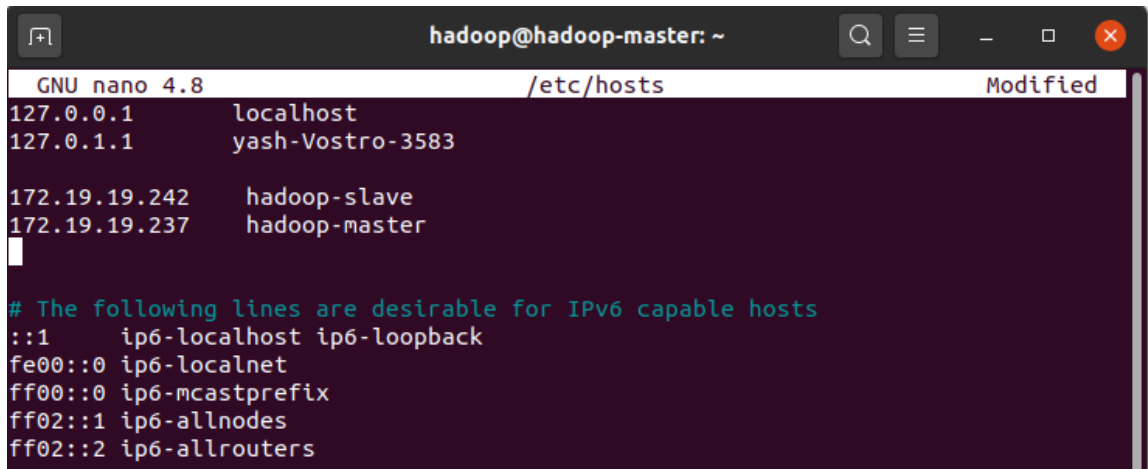
The ip addr of the above machine is 172.19.19.237

Write down the ip addresses of both the machines.

Now, run the following command on both machines:

```
sudo nano /etc/hosts
```

Now write down the ip addresses for the master and slave node as shown

A screenshot of a terminal window titled 'hadoop@hadoop-master: ~'. The terminal shows the nano 4.8 editor editing the file /etc/hosts. The file content is as follows:

```
127.0.0.1    localhost
127.0.1.1    yash-Vostro-3583

172.19.19.242  hadoop-slave
172.19.19.237  hadoop-master

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

Save the file.

17. Now, in the **master node**, apply the following command:

```
ssh-keygen -t rsa
```

Now, run the following commands:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@hadoop-master
ssh-copy-id -i ~/.ssh/id_rsa.pub hadoop@hadoop-slave
```

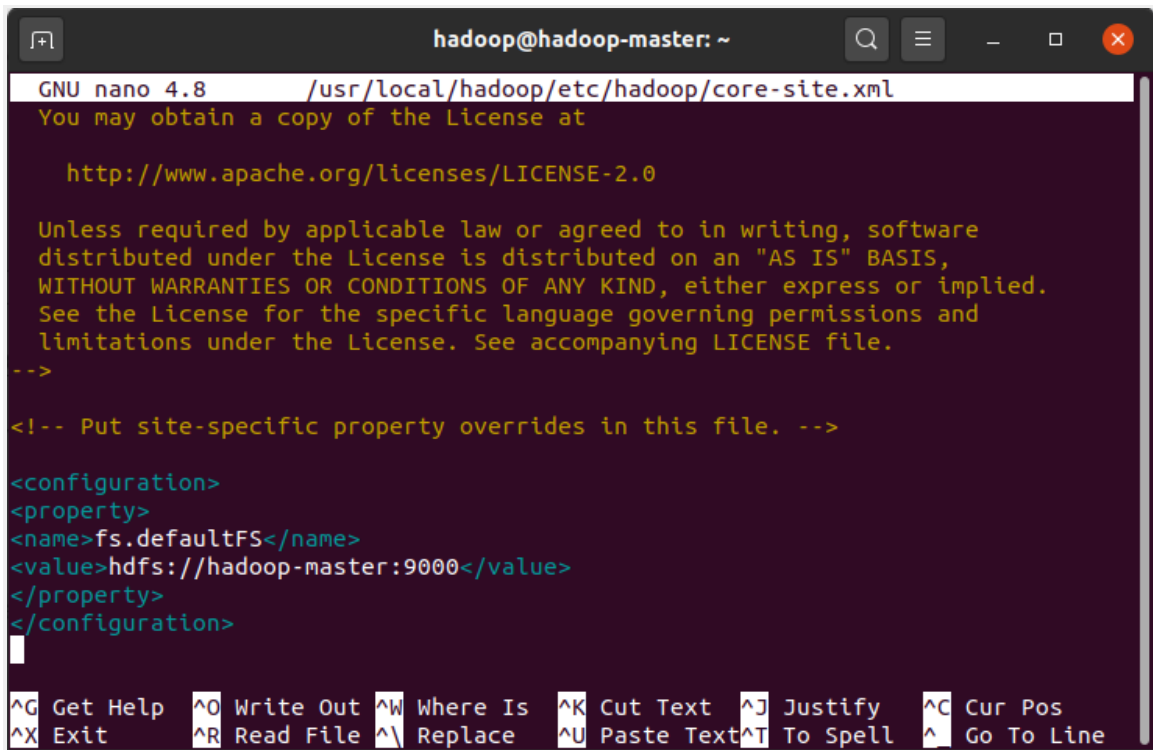
DO THE FOLLOWING STEPS FOR BOTH MASTER AND SLAVE NODE IN ORDER TO PROCESS FASTER (the process mentioned in the tutorials make changes only in the master node and the copies all the files into slave node which takes a lot of time)

18. Open core-site.xml file on nano:

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

Then add the following configurations:

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://hadoop-master:9000</value>
</property>
</configuration>
```



```
hadoop@hadoop-master: ~
GNU nano 4.8 /usr/local/hadoop/etc/hadoop/core-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.
-->

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

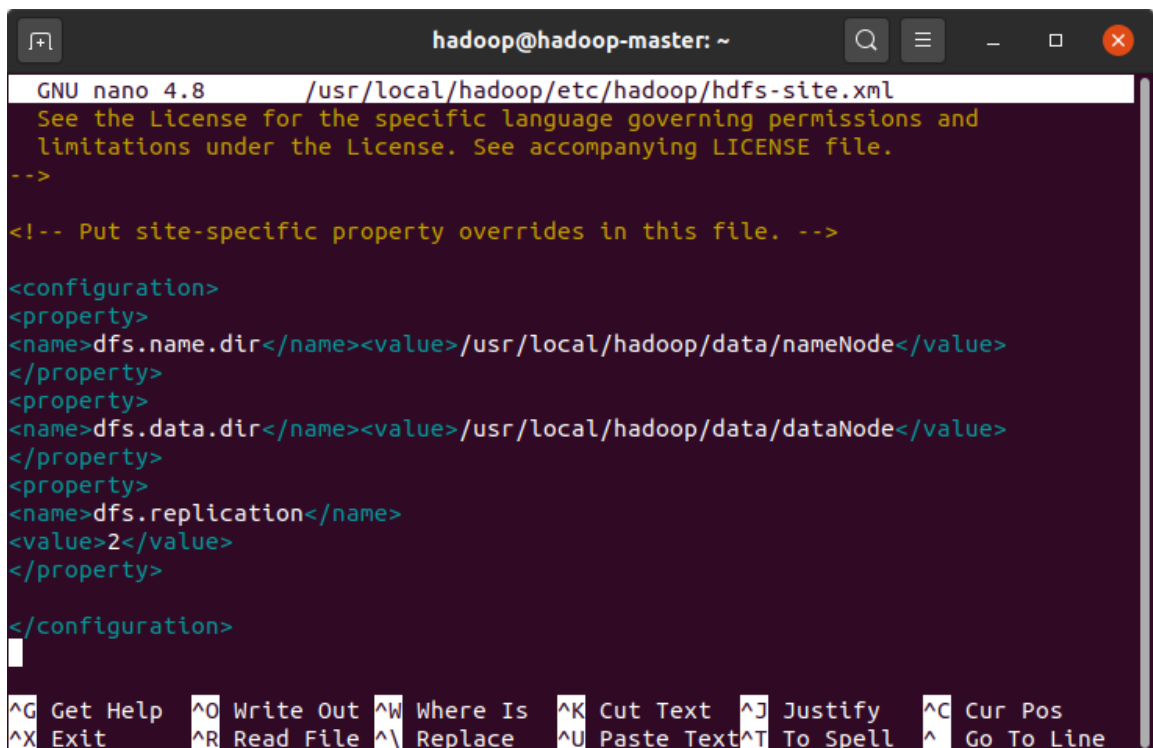
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://hadoop-master:9000</value>
</property>
</configuration>
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

19. open the hdfs-site.xml file.

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

Add the following configurations:

```
<configuration>
<property>
<name>dfs.name.dir</name><value>/usr/local/hadoop/data/name
Node</value>
</property>
<property>
<name>dfs.data.dir</name><value>/usr/local/hadoop/data/data
Node</value>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>
```



```
hadoop@hadoop-master: ~
GNU nano 4.8 /usr/local/hadoop/etc/hadoop/hdfs-site.xml
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.name.dir</name><value>/usr/local/hadoop/data/nameNode</value>
</property>
<property>
<name>dfs.data.dir</name><value>/usr/local/hadoop/data/dataNode</value>
</property>
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
</configuration>
^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
```

20. Open the mapred-site.xml file.

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


Add the following configurations:

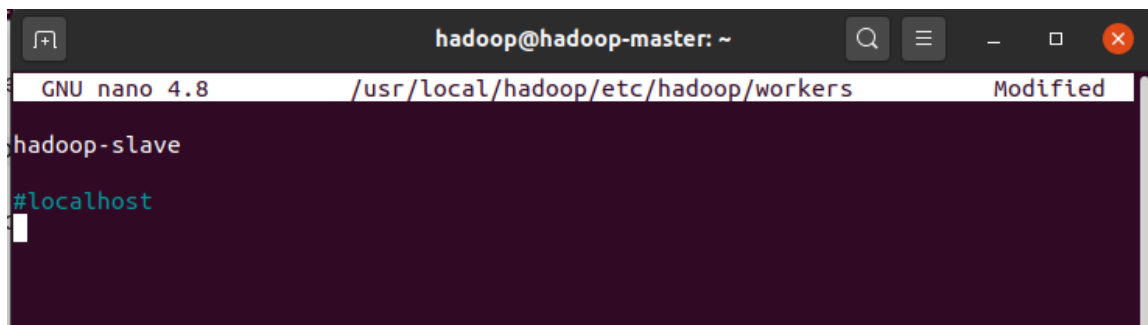
```
<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>hadoop-master:9001</value>
  </property>
</configuration>
```

21. Now, on **hadoop-master**, let's open the workers file:

```
sudo nano /usr/local/hadoop/etc/hadoop/workers
```

Add the following lines: (the slave name)

```
hadoop-slave
```



22. Now, in the **slave node**, open the yarn-site.xml file:

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

add the following configurations:

```
<property>
<name>yarn.resourcemanager.hostname</name>
<value>hadoop-master</value>
</property>
```

23. Now we need to format the HDFS file system. Run this command in the **master node**:

```
hdfs namenode -format
```

24. Now, in the **master node**, start HDFS with this command:

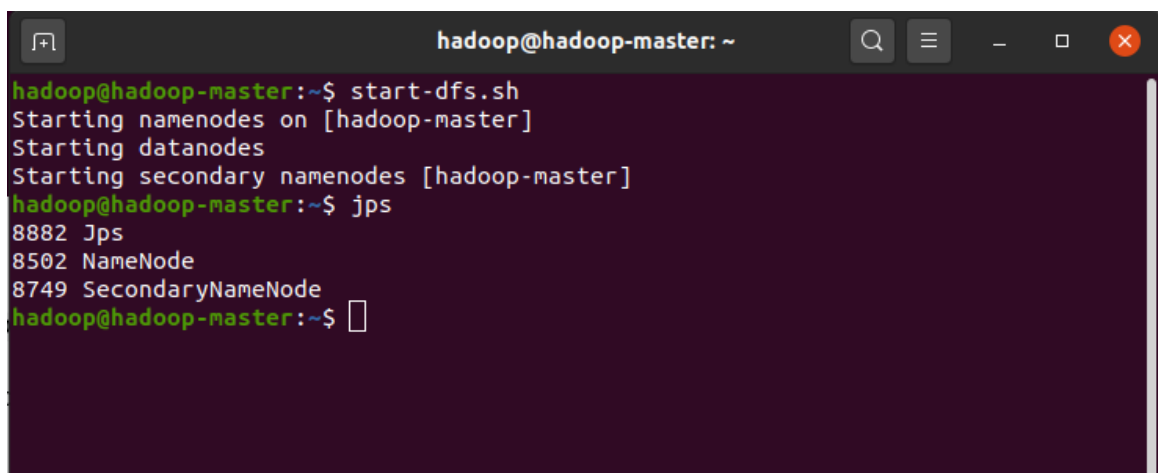
```
start-dfs.sh
```

Verify the same by applying the following command in both the machines:

```
jps
```

The output will look like this:

For master node:

A terminal window titled 'hadoop@hadoop-master: ~' showing the execution of 'start-dfs.sh' and 'jps'. The output of 'start-dfs.sh' includes 'Starting namenodes on [hadoop-master]', 'Starting datanodes', and 'Starting secondary namenodes [hadoop-master]'. The output of 'jps' lists '8882 Jps', '8502 NameNode', and '8749 SecondaryNameNode'.

```
hadoop@hadoop-master:~$ start-dfs.sh
Starting namenodes on [hadoop-master]
Starting datanodes
Starting secondary namenodes [hadoop-master]
hadoop@hadoop-master:~$ jps
8882 Jps
8502 NameNode
8749 SecondaryNameNode
hadoop@hadoop-master:~$
```

For slave node :

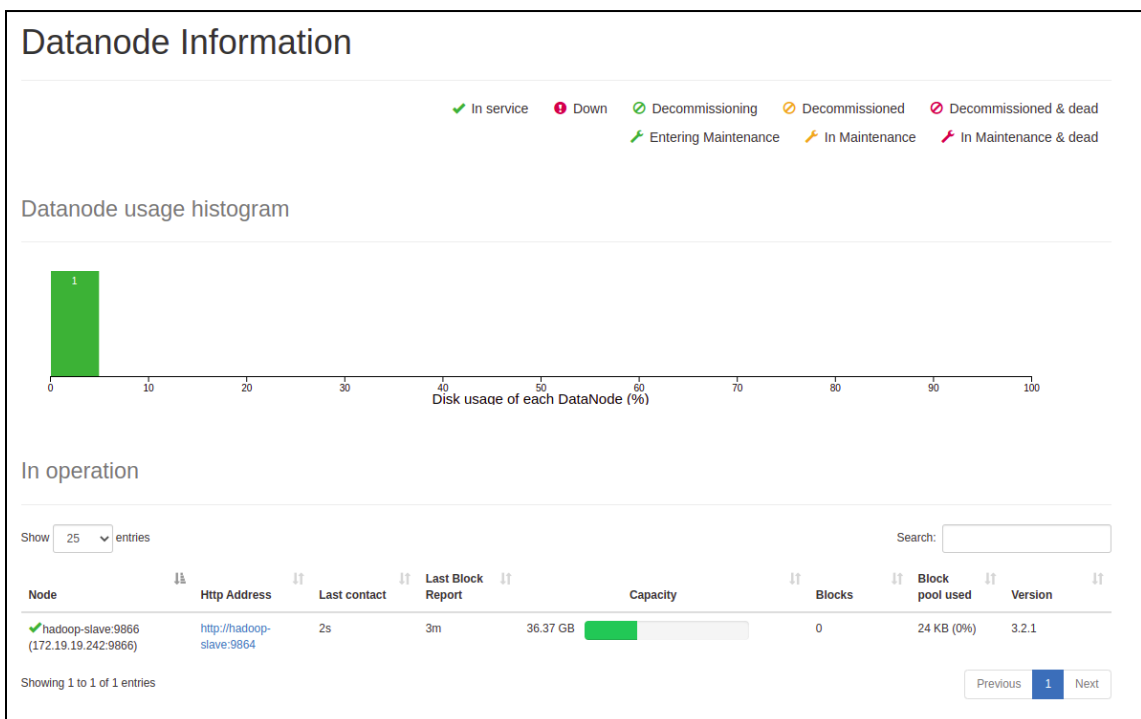
A terminal window titled 'hadoop@aakash: ~\$' showing the output of 'jps', which lists '6414 Jps' and '6351 DataNode'.

```
hadoop@aakash:~$ jps
6414 Jps
6351 DataNode
hadoop@aakash:~$
```

Also verify if everything is working fine by going to **localhost:9870** in the **master node**.

There it must be showing 1 live node

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 92.78 MB of 187 MB Heap Memory. Max Heap Memory is 1.7 GB.	
Non Heap Memory used 48.29 MB of 49.59 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.	
Configured Capacity:	36.37 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	11.47 GB
DFS Remaining:	23.02 GB (63.29%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)



25. Now, on the **master node**, let's start yarn. Use this command:

```
start-yarn.sh
```

Verify the same using the `jps` command on both the machines. The output will be :

For master node:

```
hadoop@hadoop-master:~$ start-yarn.sh
Starting resourcemanager
Starting nodemanagers
hadoop@hadoop-master:~$ jps
9587 Jps
8502 NameNode
8749 SecondaryNameNode
9279 ResourceManager
```

For slave node:

```
hadoop@aakash:~$ jps
6752 NodeManager
6861 Jps
6351 DataNode
hadoop@aakash:~$
```

Also verify the same on the **master node**, by going to **localhost:8088** in the browser.

You will find details of the network. There must be 1 active node, if everything went well.

localhost:8088/cluster

hadoop

Cluster

- About
- Nodes
- Node Labels
- Applications
 - NEW
 - NEW_SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed
0	0	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes
1	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type
Capacity Scheduler	[memory-mb (unit=M), vcores]

Show 20 entries

ID	User	Name	Application Type	Queue	Application Priority	StartTime	LaunchTime
----	------	------	------------------	-------	----------------------	-----------	------------

Showing 0 to 0 of 0 entries

26. Now, in order to stop the network, just use the following command on the master node:

```
stop-all.sh
```

27. Next time, whenever you want to use this network, just update the ip addresses in the /etc/hosts file (in both the machines) and you are good to go!!

Just apply the following command on the master node to start the network:

```
start-all.sh
```

References :

1. https://medium.com/@jootorres_11979/how-to-set-up-a-hadoop-3-2-1-multi-node-cluster-on-ubuntu-18-04-2-nodes-567ca44a3b12
2. <https://phoenixnap.com/kb/install-hadoop-ubuntu>
3. https://www.tutorialspoint.com/hadoop/hadoop_multi_node_cluster.htm