Hadoop Distributive Cluster:

Insight:

To manage Big Data, we need high computational speed and high storage capacity too. Particularly for the companies like Facebook, Instagram, or the banking industry, petabytes of data is shared on daily basis. For managing this data efficiently, the system needs to overcome the velocity and the volume problem of Big Data.

For increasing the storage, usually two techniques come in our mind. They are horizontal scaling and vertical scaling.

- Vertical scaling refers to increasing the storage of your system by adding HD, PD etc. However, for business perspective, it is not a good idea because it can lead to the issue known as data unavailability if the system goes down.
- For managing the data efficiently, the Big Data is handled using horizontal scaling. In this technique, a system is set up to be the master. This system can also be called the namenode. It is connected to several slaves or the datanode that contribute its storage space to the master. Now, when master receives any data, it stripes it into blocks and sends each block to its master. This increases the efficiency of the whole case and simultaneously solves the Volume and the Velocity issue of Big Data.

I have created the master-slave topology or the “distributive storage cluster” using Hadoop. In this topology, the whole setup combining the master and the slave is referred to as the “cluster”. The master and the slaves are referred to as the nodes. Master is the namenode and the slaves are the datanode.

Requirements:

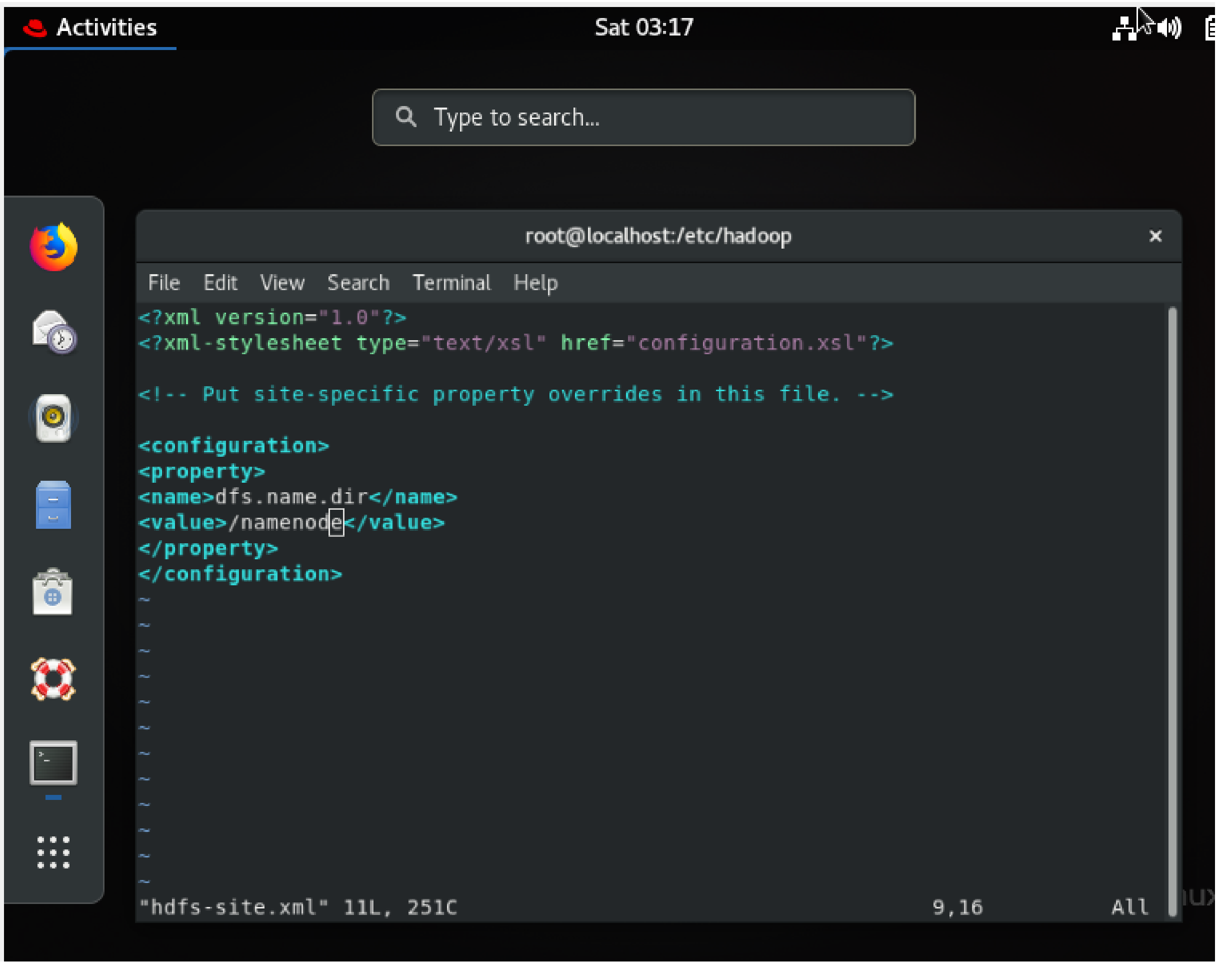
For completing this project, at least 2 laptops or PCs are required. However, I used virtualization to solve the issue of multiple laptops. 2 RedHat Linux OS is the minimum requirement. One shall act as the master or the data node and the other as the slave or the datanode.

Hadoop and Java needs to get installed on both the OS before moving further.

- **Creating Master:**

Select one OS to be your namenode. Check whether the java and Hadoop versions are compatible and successfully running. (java –version   hadoop version)

```
cd /etc/hadoop            //get into the hadoop directory

mkdir /namenode          //created a directory for the namenode.

cd/

pwd/

ls                             //check if the namenode directory you created is visible.

cd namenode          //get into the namenode directory

vim hdfs-site.xml   //here we shall set up the namenode as follows:
```


Activities
Sat 03:17
Type to search...
root@localhost:/etc/hadoop
File Edit View Search Terminal Help

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
<property>
<name>dfs.name.dir</name>
<value>/namenode</value>
</property>
</configuration>
~
~
~
~
~
~
~
~
~
~
~
~
"hdfs-site.xml" 11L, 251C                              9,16          All
```

➢ **Creating Slave:**

Go to the OS which will act as the slave. Follow some similar steps to create the datanode.

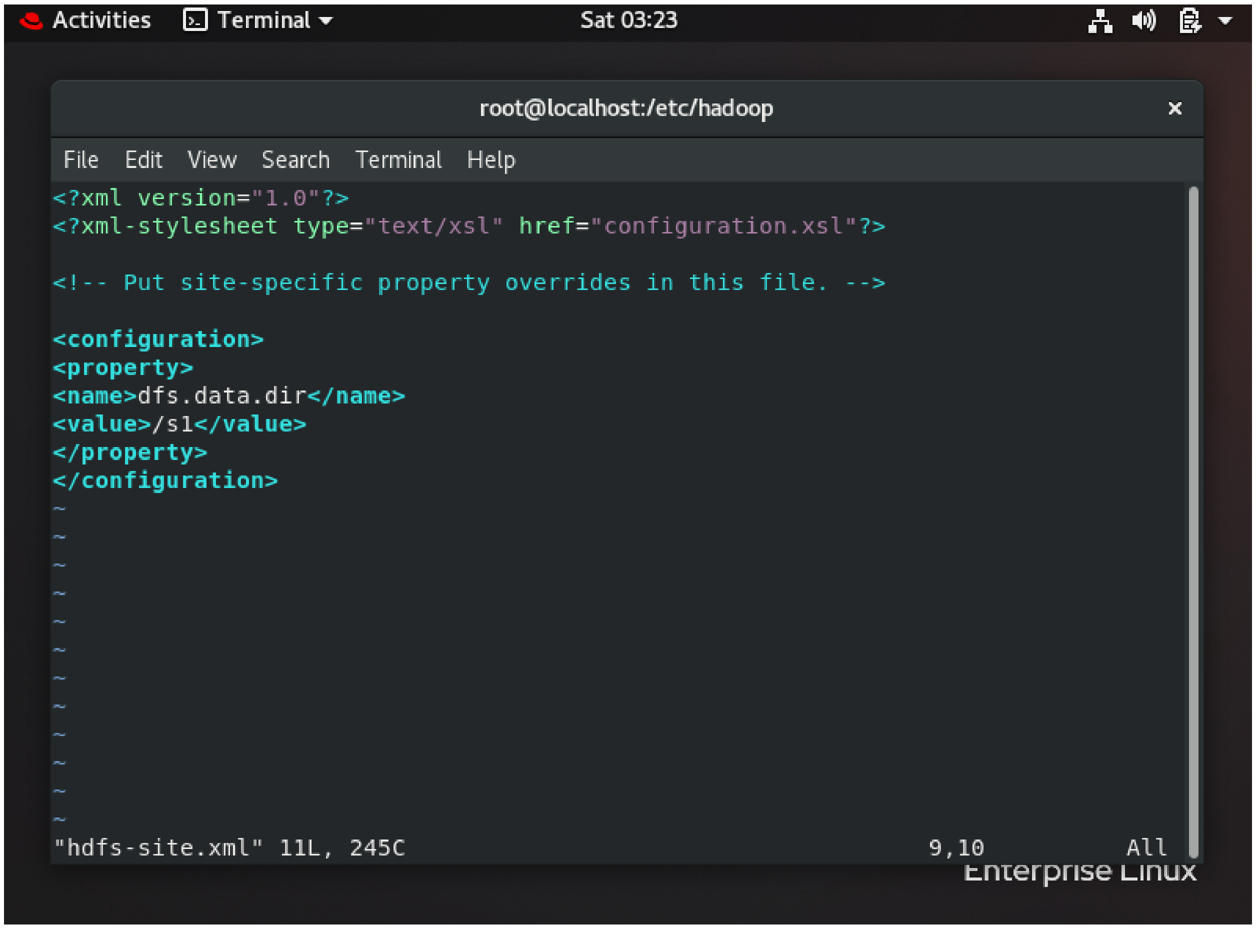
mkdir /s1

cd /

ls

cd s1/

vim hdfs-site.xml //be careful while putting the name and value


Activities
Terminal
Sat 03:23
root@localhost:/etc/hadoop
File Edit View Search Terminal Help
Enterprise Linux

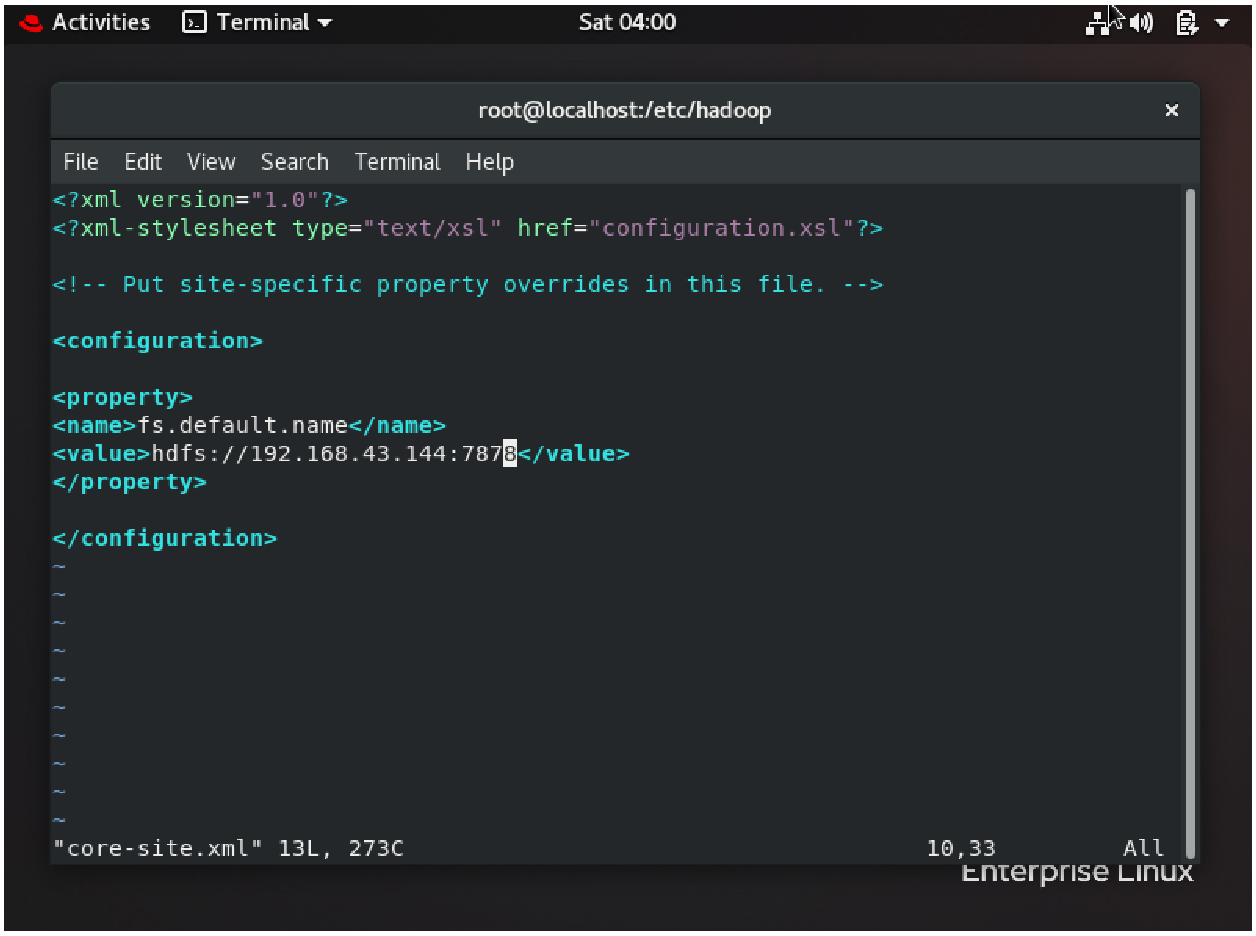
```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
<property>
<name>dfs.data.dir</name>
<value>/s1</value>
</property>
</configuration>
~
~
~
~
~
~
~
~
~
~
~
~
"hdfs-site.xml" 11L, 245C                                9,10          All
```

There should not be any indentation as some versions do not support it. So, now, the namenode and the datanode are ready. The slave is ready to share its storage space to its master, but currently it is not sharing. We shall now make the datanode to contribute its storage to the master.

- **Activating the master:**

cd /etc/hadoop

vim core-site.xml   //edit the file as follows:


Activities
Terminal
Sat 04:00
root@localhost:/etc/hadoop
File Edit View Search Terminal Help
Enterprise Linux

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>

<property>
<name>fs.default.name</name>
<value>hdfs://192.168.43.144:7878</value>
</property>

</configuration>
~
~
~
~
~
~
~
~
~
~
"core-site.xml" 13L, 273C                                10,33          All
```

Here hdfs is the protocol used to establish connection between master and slave. Ip provided is the ip of the master. 7878 is the port number. Avoid indentation. The configuration of the master node is completed. Before activating it, we need to

format it ones.

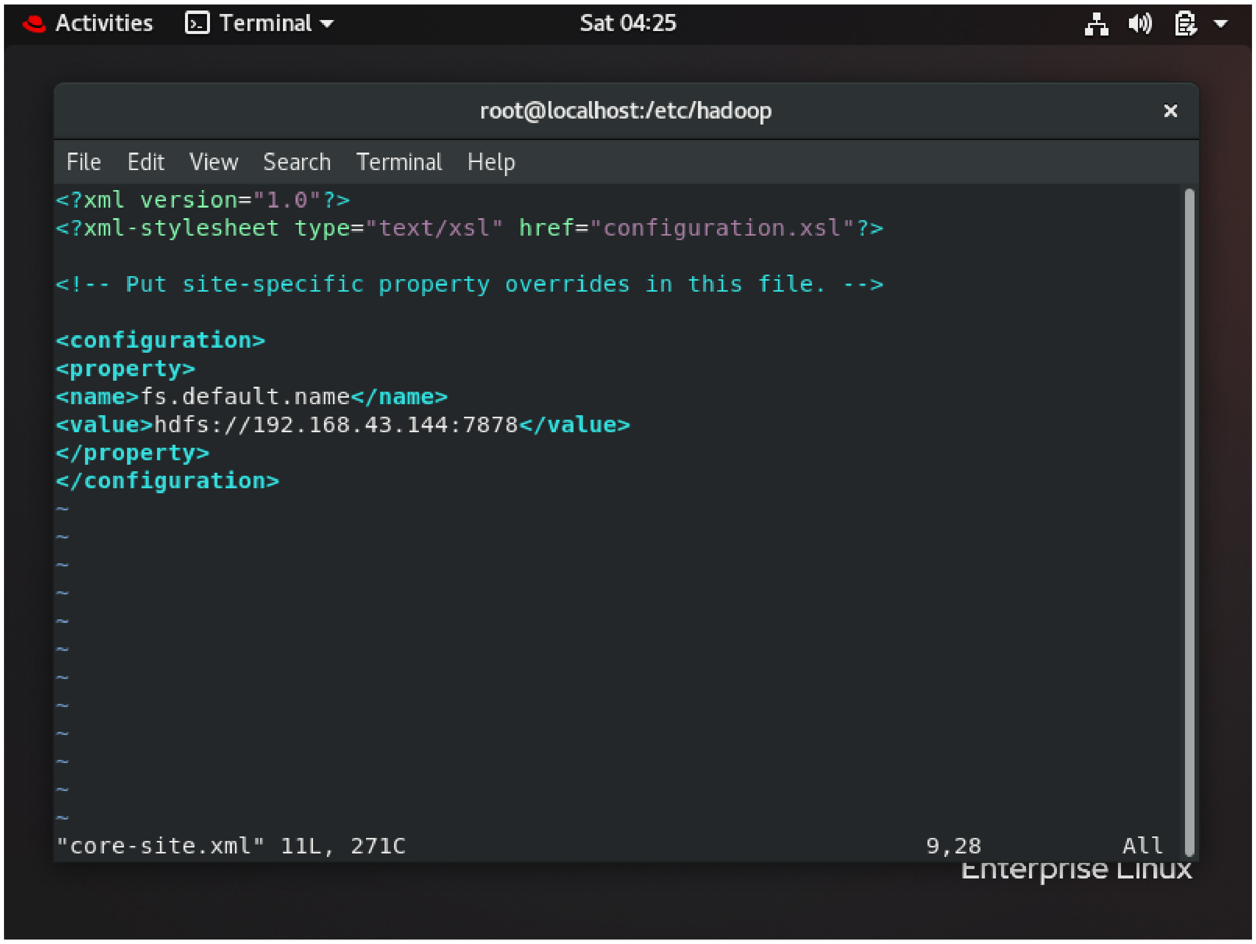
Follow the mentioned steps to format and start the namenode.

cd

```
hadoop namenode –format
hadoop-daemon.sh start namenode    //namenode is started
jps    //it shows that the namenode of port number 7878 is active.
systemctl stop firewalld    //disables firewall to enable slave connection.
```

- **Activate the slave**

```
cd /etc/hadoop
vim core-site.xml
```


Activities
Terminal
Sat 04:25
root@localhost:/etc/hadoop
File Edit View Search Terminal Help
Enterprise Linux

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://192.168.43.144:7878</value>
</property>
</configuration>
~
~
~
~
~
~
~
~
~
~
~
~
"core-site.xml" 11L, 271C                              9,28          All
```

The ip address should be of the master. It will tell the salve which ip address to connect. Similarly, the port number will also be the same.

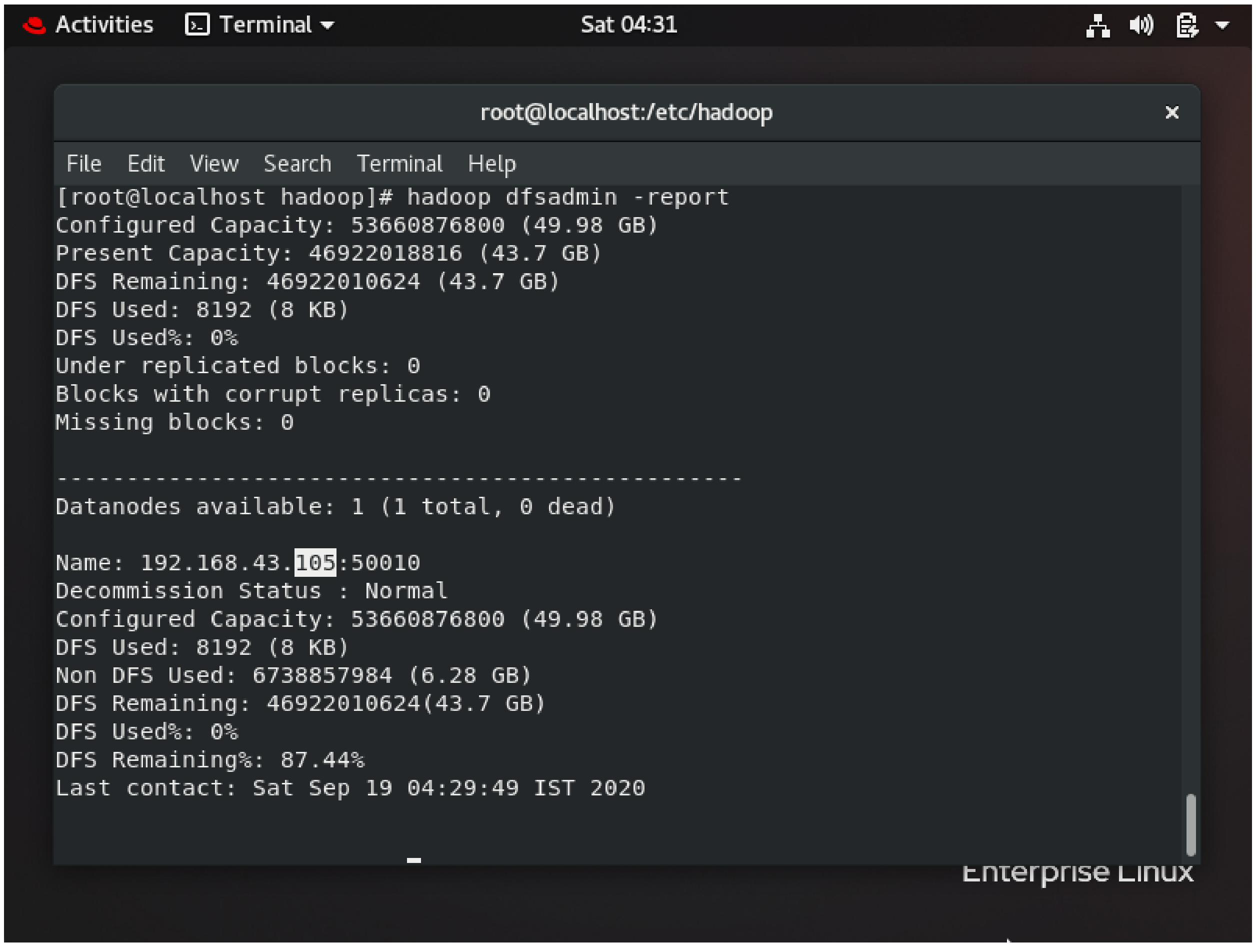
hadoop-daemon.sh start datanode

jps   //it shows that the datanode is active.

- **Testing**

For testing if the connection is successfully establishes and the slave is sharing its data with the master, go to the namenode and type the following command.

hadoop dfsadmin –report


Activities
Terminal
Sat 04:31
root@localhost:/etc/hadoop
File Edit View Search Terminal Help
Enterprise Linux

```
[root@localhost hadoop]# hadoop dfsadmin -report
Configured Capacity: 53660876800 (49.98 GB)
Present Capacity: 46922018816 (43.7 GB)
DFS Remaining: 46922010624 (43.7 GB)
DFS Used: 8192 (8 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-------------------------------------------------
Datanodes available: 1 (1 total, 0 dead)

Name: 192.168.43.105:50010
Decommission Status : Normal
Configured Capacity: 53660876800 (49.98 GB)
DFS Used: 8192 (8 KB)
Non DFS Used: 6738857984 (6.28 GB)
DFS Remaining: 46922010624(43.7 GB)
DFS Used%: 0%
DFS Remaining%: 87.44%
Last contact: Sat Sep 19 04:29:49 IST 2020
```

It shows that the datanode available is 1. It also shows the ip of the datanode. Similarly, you can add many slaves to the master and increase its efficiency. This makes the implementation of hadoop distributive storage cluster successful.