# HADOOP MULTINODE INSTALLATION

## Multi node-cluster setup

We will create a 3 node cluster system (1-Master, 2-slave).

### Systems should be connected to same network

#### Check if nodes are reachable

Find the ip Address of all 3 systems & try to ping each other

```
hduser@ubuntu:~$ifconfig
```

Master 192.168.220.180
Slave1 192.168.220.181
Slave2 192.168.220.182

```
Master  hduser@ubuntu:~$ ping 192.168.220.181     //Master pinging slave1
Master  hduser@ubuntu:~$ ping 192.168.220.182     //Master pinging slave2

Slave1   hduser@ubuntu:~$ ping 192.168.220.180     //Slave1 pinging master
Slave1   hduser@ubuntu:~$ ping 192.168.220.182     //Slave1 pinging slave1

Slave2   hduser@ubuntu:~$ ping 192.168.220.180     //Slave2 pinging master
Slave2   hduser@ubuntu:~$ ping 192.168.220.181     //Slave2 pinging slave1
```

### Change the hostname of all 3 system

Go to Master system.

```
$ sudo vim /etc/hostname
```

Replace ubuntu with master in file.
Press ESC on the keybord.
Save the configuration by :wq!
Repeat the above step with salve1 & slave2 VM and change the host name to **slave1, slave2**

### Update the hosts on all 3 nodes

Master SYSTEM:

```
$ sudo vim /etc/hosts
```

```
127.0.0.1        localhost            # dont make any changes to this line.
#127.0.1.1       master              # remove this line.
```

REFERENCE :KAUSHIK SHAKKARI
EMAIL : kaushik.shakkari@gmail.com

# HADOOP MULTINODE INSTALLATION

```
192.168.220.185   master          #Add this 3 lines.
192.168.220.186   slave1
192.168.220.187   slave2
```

Repeat the same in Slave1 and Slave2

## 1. Restart all systems

**Confirm the hostname of all 3 nodes.**

Execute below command in all three systems.

```
$hostname
```

It should print **master, slave1, slave2** in 3 machines respectively.

## 2. Ping Each other using Hostname

Start pinging each other system again using the hostname instead of ipaddress
Master->ping slave1 & slave2
Slave1->ping master only
Slave2->ping master only

```
hduser@master:~$  ping slave1
hduser@master:~$  ping slave2
hduser@slave1 :~$  ping master
hduser@slave2 :~$  ping master
```

## 3. Test SSH connectivity

```
hduser@master:~$ ssh master
hduser@master:~$ ssh slave1                  #Type Yes. It will connect to slave1
hduser@slave1:~$ exit                        #will exit slave1 and goes back to
master.
hduser@master:~$ ssh slave2                  # Type Yes. It will connect to slave2
hduser@slave2:~$ exit                        #will exit slave2 and goes back to
master.
hduser@master:~$
```

REFERENCE :KAUSHIK SHAKKARI
EMAIL : kaushik.shakkari@gmail.com

It will ask for yes or no and you should type 'yes'
We should be able to SSH master and SSH slaves without password prompt.
If it asks for password while connecting to master or slave using SSH, there is something went wrong and you need to fix it before proceeding further.

**Common Configurations (needs to be done in master as well as all slaves)**

1. **Update core-site.xml**(Master+ All Slave Nodes)

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

we have 2 changes

   a. Remove hadoop.tmp.dir configuration. We don’t require them
   b. change localhost  to master
      **replace**

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

**with**

```
 <property>
         <name> fs.default.name </name>
        <value>hdfs://master:9000</value>  <!--instead of localhost -->
</property>
```

2. **Update hdfs-site.xml**(Master + All slave Nodes)

3 changes

   a. Replication is set to 2
   b. Namenode configured only in master
   c. Datanode configured only in slave

```
$ sudo vim /usr/local/hadoop/etc/hadoop/hdfs-site.xml
<property>
   <name>dfs.replication</name>
   <value>2</value> <!--changing replication from 1 to 2 -->
  </property>

<!--Keep this entry in Master only, and delete from slaves-->
<property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
```

```
</property>

<!--Remove this entry From Master, and  Keep  this entry in slaves-->
<property>
   <name>dfs.datanode.data.dir</name>
   <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
```

3. **Update yarn-site.xml**(Master + All Slave Nodes)

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



<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>master:8025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>master:8030</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>master:8050</value>
</property>
```

4. **Update mapred-site.xml**(Master + All Slave Nodes)

```
$sudo vim  /usr/local/hadoop/etc/hadoop/mapred-site.xml
  <property>
          <name>mapreduce.jobhistory.address</name>
                   <value>master:10020</value><!-- Pointing jobhistory to master-->
                   <description>Host and port for Job History Server (default

                   0.0.0.0:10020)</description>
</property>
```

**Master only Configuration**

5. **Update Masters and slaves file(Master Node only)**

If you see any entry related to localhost, feel free to delete it. This file is just helper file that are used by hadoop scripts to start appropriate services on master and slave nodes.

```
hduser@master$ sudo vim  /usr/local/hadoop/etc/hadoop/slaves
```

```
slave1
slave2
```

Below masters file does not exists by default. It gets created the file.

```
hduser@master$ sudo vim  /usr/local/hadoop/etc/hadoop/masters
master
```

**Note**: You don’t need to configure them in slave nodes

### 6. Recreate Namenode folder(Master Only)

```
hduser@master$
sudo rm -rf /usr/local/hadoop_tmp
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/namenode
```

### Recreate Datanode folder(All Slave Nodes Only)

```
hduser@slave1$
sudo rm -rf /usr/local/hadoop_tmp
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
sudo chmod 777 /usr/local/hadoop_tmp/hdfs/datanode
```

### 7. Format the Name node(Master only)

Before starting the cluster, we need to format the Name node. Use the following command only on master node:

```
$ hdfs namenode -format
```

### 8. Start the DFS & Yarn (Master Only)

```
$ start-dfs.sh
$ start-yarn.sh
```

or

```
$ start-all.sh
```

#deprecated but internally calls start-dfs.sh && start-yarn.shType Yes when asked for.

You should observe that it tries to start data node on slave nodes one by one.
Once it is started,Do a Jps on Master and slaves.

Jps on Master node

# HADOOP MULTINODE INSTALLATION

```
hduser@master$ jps
3379 NameNode                    #because of start-dfs.sh
3175 SecondaryNameNode         #because of start-dfs.sh
3539 ResourceManager           #because of start-yarn.sh
```

Jps on slave nodes(slave1 and slave2)

```
hduser@slave1$ jps
2484 DataNode                    #because of start-dfs.sh
2607 NodeManager                 #because of start-yarn.sh
```

## 9. Review Yarn console:

If all the services started successfully on all nodes, then you should see all of your nodes listed under Yarn nodes. You can hit the following url on your browser and verify that:

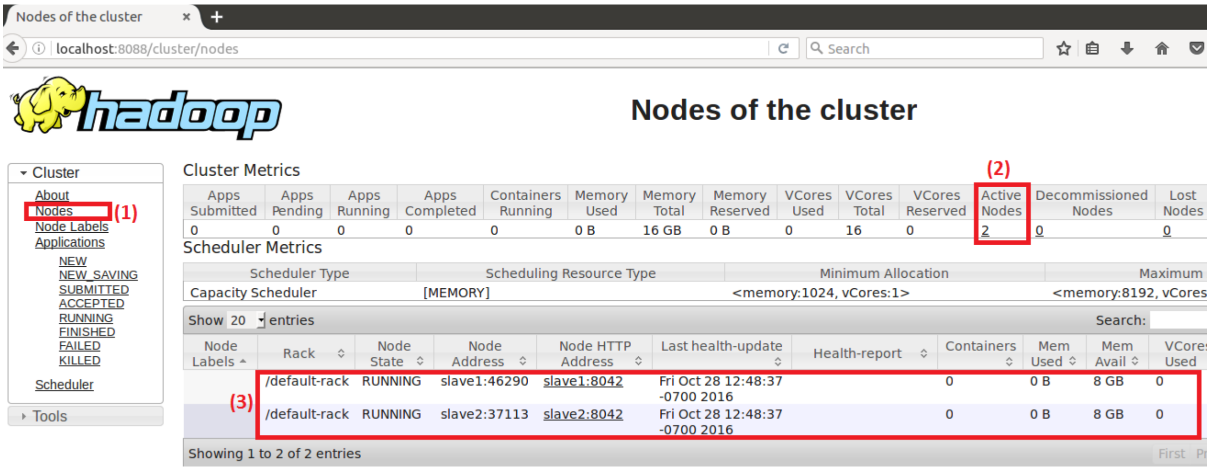
http://master:8088/cluster/nodes
http://master:50070     # can show live node count and info about each live nodes.

You can also get the report of your cluster by issuing the below commands

```
hduser@master$ hdfs dfsadmin -report
```

In case you don’t get to see the live nodes on the browsers, there are some mistakes somewhere. You need to figure it out. You can  look into the logs in the below location
/usr/local/hadoop/logs/
Check in all the log files of both master and slaves and it should hint you the problem.

Screenshots:



Nodes of the cluster

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Total | VCores Reserved | Active Nodes | Decommissioned Nodes | Lost Nodes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 B | 16 GB | 0 B | 0 | 16 | 0 | 2 | 0 | 0 |

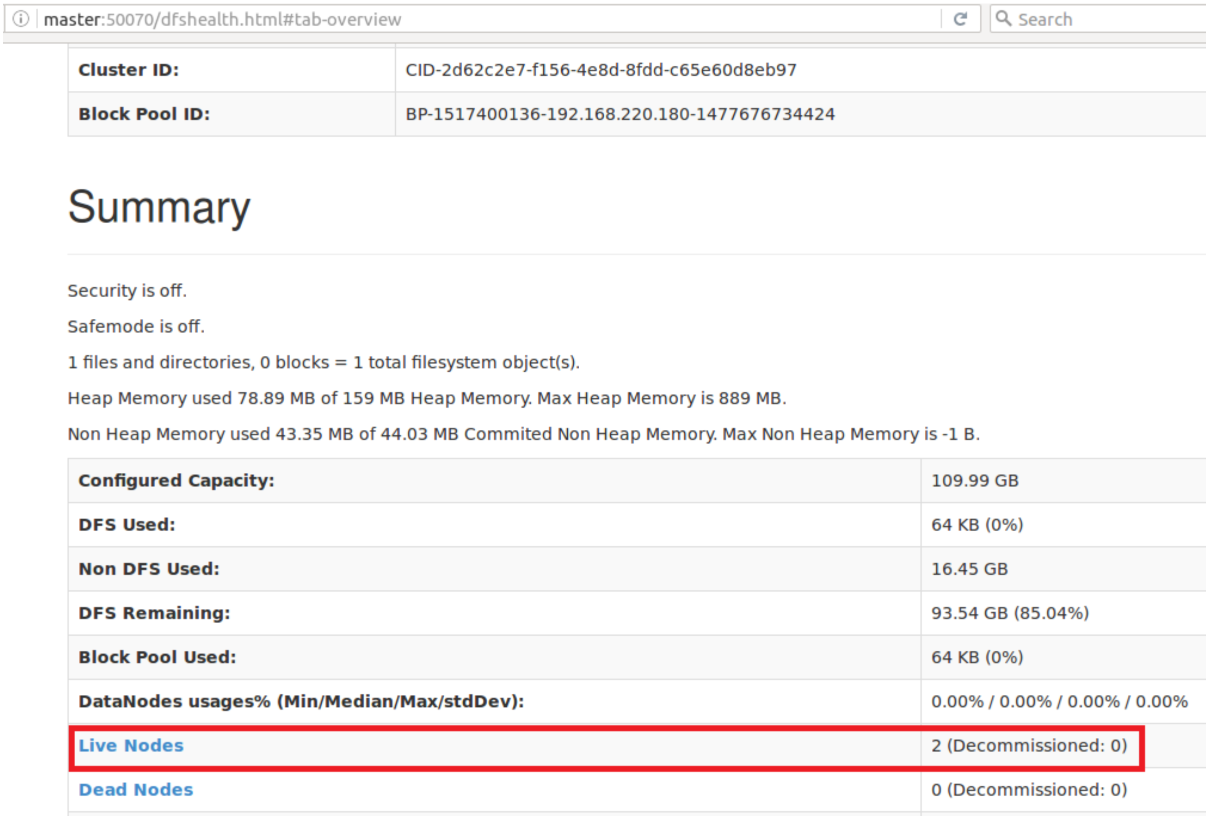
Scheduler Metrics

| Scheduler Type | Scheduling Resource Type | Minimum Allocation | Maximum |
|---|---|---|---|
| Capacity Scheduler | [MEMORY] | <memory:1024, vCores:1> | <memory:8192, vCores |

Show 20 entries Search:

| Node Labels | Rack | Node State | Node Address | Node HTTP Address | Last health-update | Health-report | Containers | Mem Used | Mem Avail | VCores Used |
|---|---|---|---|---|---|---|---|---|---|---|
|  | /default-rack | RUNNING | slave1:46290 | slave1:8042 | Fri Oct 28 12:48:37 -0700 2016 |  | 0 | 0 B | 8 GB | 0 |
|  | /default-rack | RUNNING | slave2:37113 | slave2:8042 | Fri Oct 28 12:48:37 -0700 2016 |  | 0 | 0 B | 8 GB | 0 |

Showing 1 to 2 of 2 entries



master:50070/dfshealth.html#tab-overview

| | |
|---|---|
| **Cluster ID:** | CID-2d62c2e7-f156-4e8d-8fdd-c65e60d8eb97 |
| **Block Pool ID:** | BP-1517400136-192.168.220.180-1477676734424 |

## Summary

Security is off.

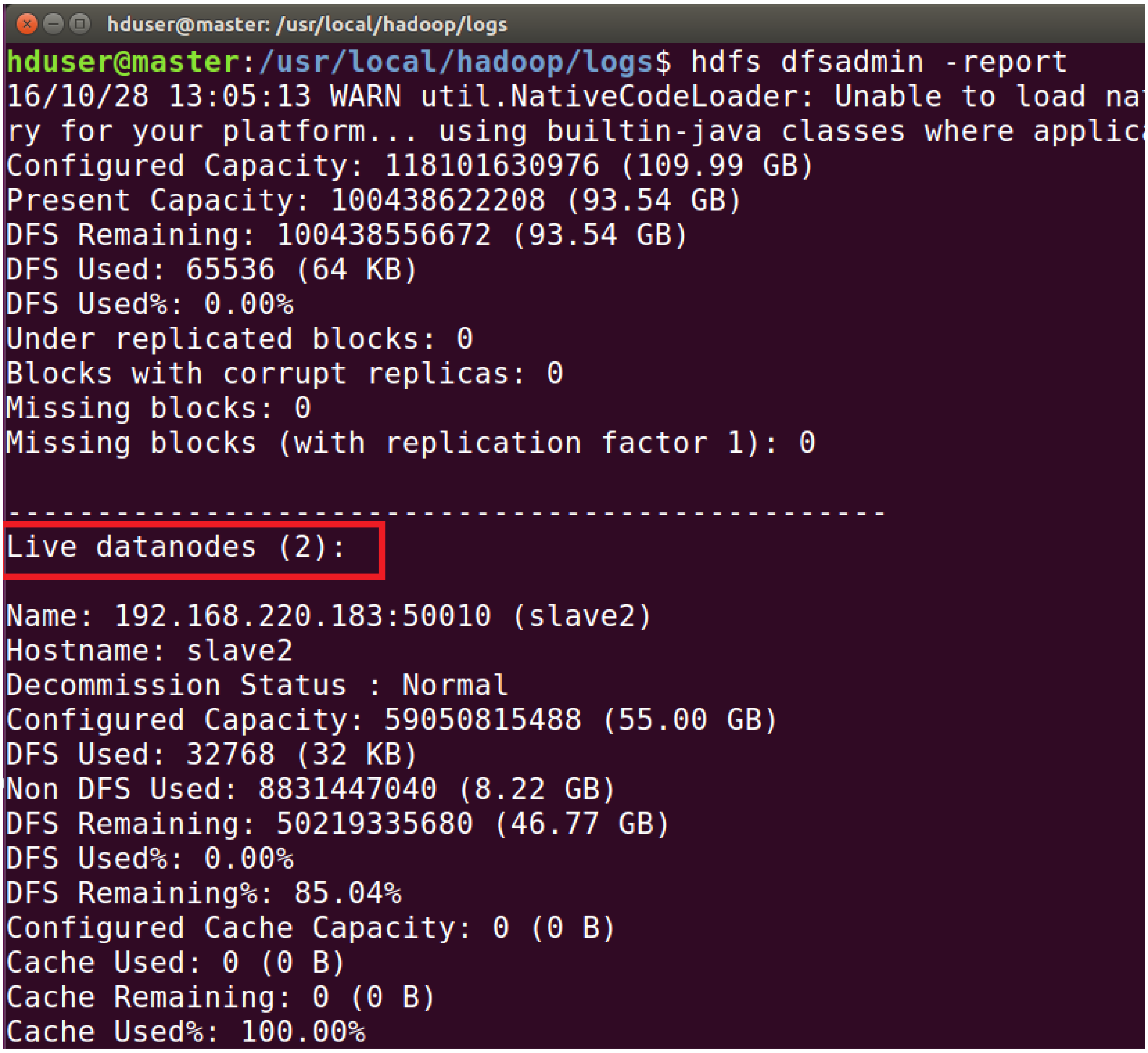
Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 78.89 MB of 159 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 43.35 MB of 44.03 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

| | |
|---|---|
| **Configured Capacity:** | 109.99 GB |
| **DFS Used:** | 64 KB (0%) |
| **Non DFS Used:** | 16.45 GB |
| **DFS Remaining:** | 93.54 GB (85.04%) |
| **Block Pool Used:** | 64 KB (0%) |
| **DataNodes usages% (Min/Median/Max/stdDev):** | 0.00% / 0.00% / 0.00% / 0.00% |
| **Live Nodes** | 2 (Decommissioned: 0) |
| **Dead Nodes** | 0 (Decommissioned: 0) |



hduser@master: /usr/local/hadoop/logs

```
hduser@master:/usr/local/hadoop/logs$ hdfs dfsadmin -report
16/10/28 13:05:13 WARN util.NativeCodeLoader: Unable to load na
ry for your platform... using builtin-java classes where applic
Configured Capacity: 118101630976 (109.99 GB)
Present Capacity: 100438622208 (93.54 GB)
DFS Remaining: 100438556672 (93.54 GB)
DFS Used: 65536 (64 KB)
DFS Used%: 0.00%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (2):

Name: 192.168.220.183:50010 (slave2)
Hostname: slave2
Decommission Status : Normal
Configured Capacity: 59050815488 (55.00 GB)
DFS Used: 32768 (32 KB)
Non DFS Used: 8831447040 (8.22 GB)
DFS Remaining: 50219335680 (46.77 GB)
DFS Used%: 0.00%
DFS Remaining%: 85.04%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
```