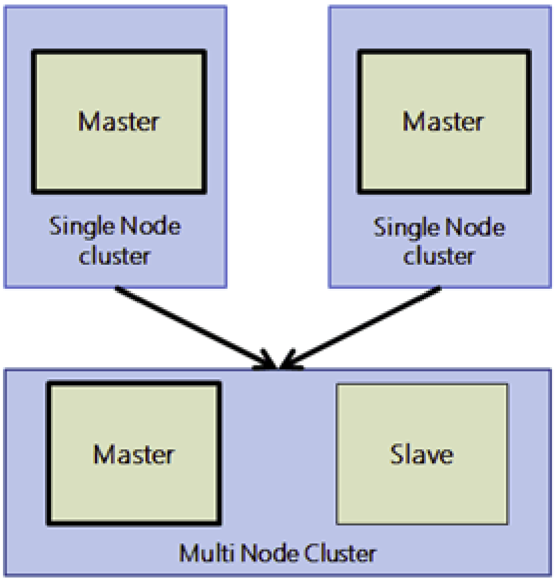
Install Multi-Node Cluster on Hadoop

The following document describes the required steps for setting up a distributed multi-node Apache Hadoop cluster on two Ubuntu machines, the best way to install and setup a multi node cluster is to start installing two individual single node Hadoop clusters by following my previous tutorial and merge them together with minimal configuration changes in which one Ubuntu box will become the designated master and the other box's will become a slave, we can add n number of slaves as per our future request.

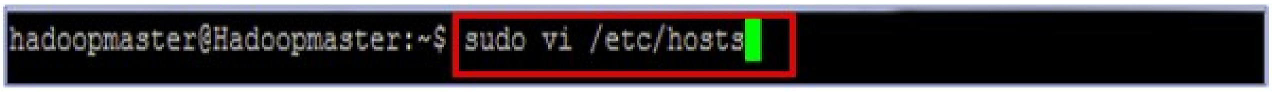


1. Prerequisites

i.Networking

Networking plays an important role here, before merging both single node servers into a multi node cluster we need to make sure that both the node pings each other(they need to be connected on the same network / hub or both the machines can speak to each other). Once we are done with this process, we will be moving to the next step in selecting the master node and slave node, here we are selecting 172.16.17.68 as the master machine(Hadoopmaster) and 172.16.17.61 as a slave (hadoopnode) . Then we need to add them in '/etc/hosts' file on each machine as follows.

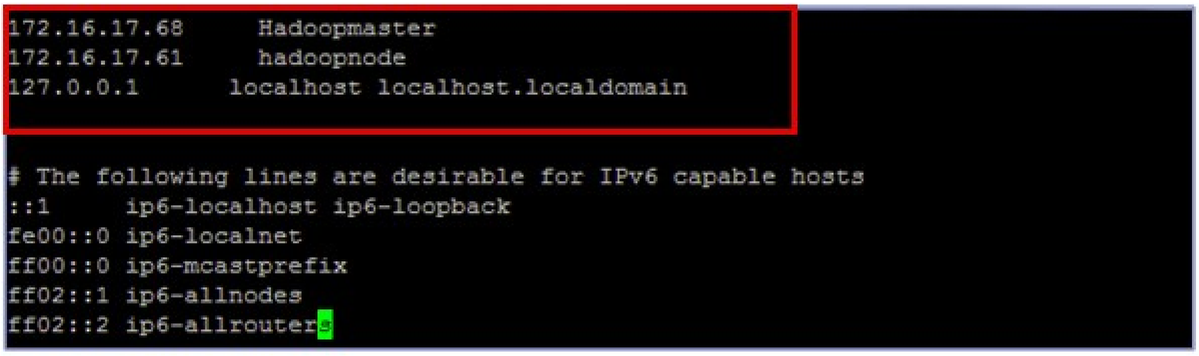
```
sudo vi /etc/hosts
```



```
hadoopmaster@Hadoopmaster:~$ sudo vi /etc/hosts
```

```
172.16.17.68      Haadoopmaster
172.16.17.61      hadoopnode
```

Note: The addition of more slaves should be updated here in each machine using unique names for slaves (e.g.: 172.16.17.xx hadoonode01, 172.16.17.xy slave02 so on..).

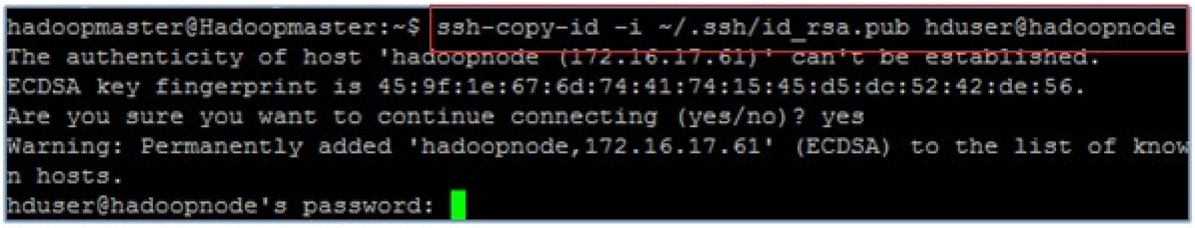


```
172.16.17.68      Hadoopmaster
172.16.17.61      hadoopnode
127.0.0.1       localhost localhost.localdomain

# 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
```

ii. Enabling SSH:

hduser on master(Hadoopmaster) machine need to able to connect to its own master (Hadoopmaster) account user and also need to connect hduser to the slave (hadoopnode) machine via password-less SSH login.

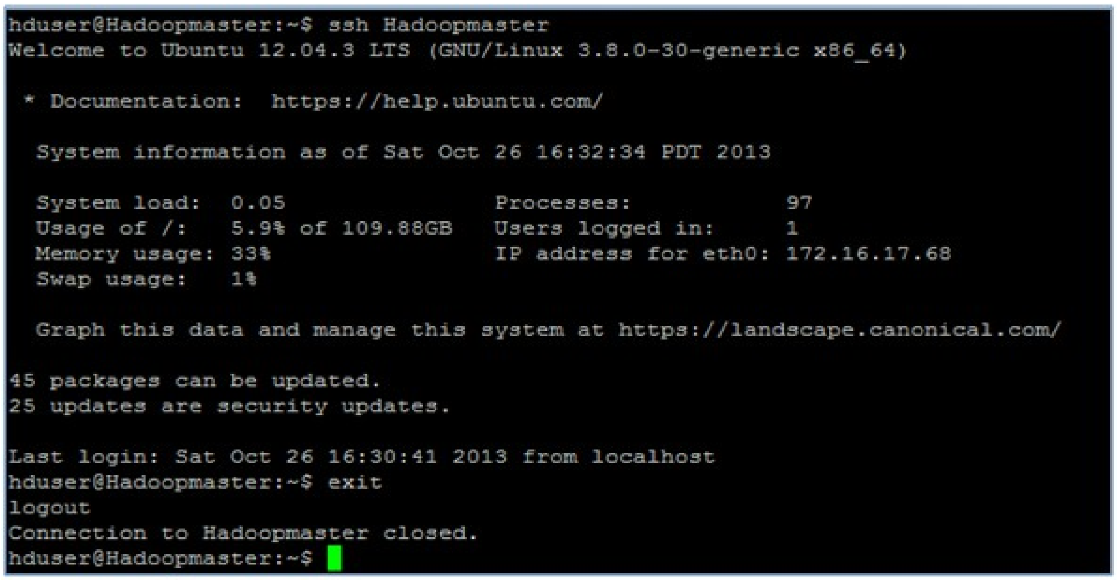
```
hduser@Hadoopmaster:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub hduser@hadoonode
```



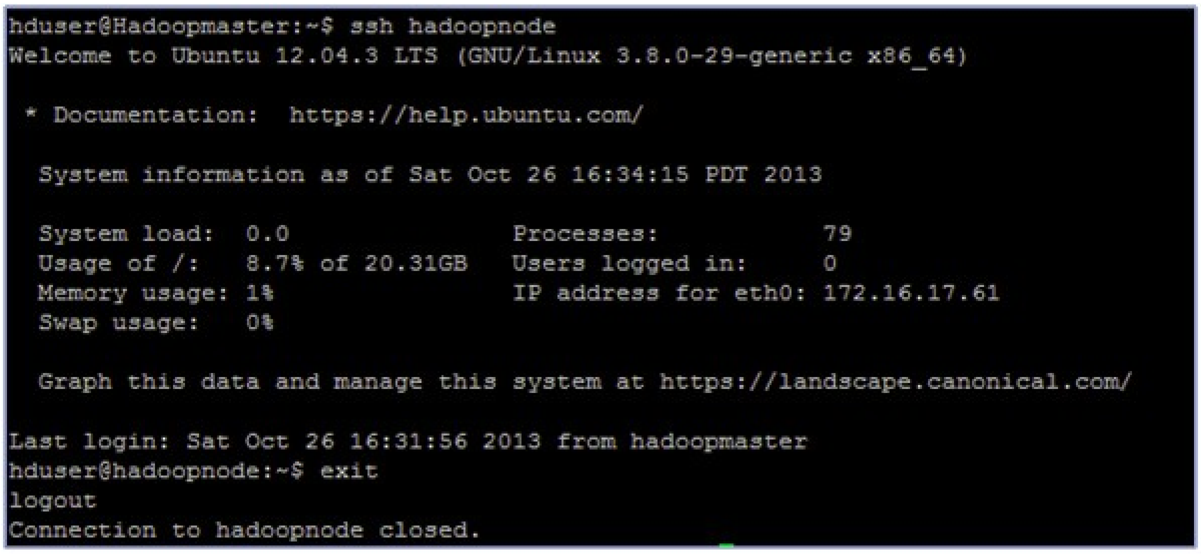
```
hadoopmaster@Hadoopmaster:~$ ssh-copy-id -i ~/.ssh/id_rsa.pub hduser@hadoopnode
The authenticity of host 'hadoopnode (172.16.17.61)' can't be established.
ECDSA key fingerprint is 45:9f:1e:67:6d:74:41:74:15:45:d5:dc:52:42:de:56.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'hadoopnode,172.16.17.61' (ECDSA) to the list of know
n hosts.
hduser@hadoopnode's password:
```

If you can see the below output when you run the given command on both master and slave, then we configured it correctly.

```
ssh Hadoopmaster
ssh hadoopnode
```



```
hduser@Hadoopmaster:~$ ssh Hadoopmaster
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Oct 26 16:32:34 PDT 2013

  System load:  0.05               Processes:           97
  Usage of /:   5.9% of 109.88GB   Users logged in:     1
  Memory usage: 33%                IP address for eth0: 172.16.17.68
  Swap usage:   1%

  Graph this data and manage this system at https://landscape.canonical.com/

45 packages can be updated.
25 updates are security updates.

Last login: Sat Oct 26 16:30:41 2013 from localhost
hduser@Hadoopmaster:~$ exit
logout
Connection to Hadoopmaster closed.
hduser@Hadoopmaster:~$
```



```
hduser@Hadoopmaster:~$ ssh hadoopnode
Welcome to Ubuntu 12.04.3 LTS (GNU/Linux 3.8.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

  System information as of Sat Oct 26 16:34:15 PDT 2013

  System load:  0.0               Processes:           79
  Usage of /:   8.7% of 20.31GB   Users logged in:     0
  Memory usage: 1%                IP address for eth0: 172.16.17.61
  Swap usage:   0%

  Graph this data and manage this system at https://landscape.canonical.com/

Last login: Sat Oct 26 16:31:56 2013 from hadoopmaster
hduser@hadoopnode:~$ exit
logout
Connection to hadoopnode closed.
```

2. Configurations:

The following are the required files we will use for the perfect configuration of the multi node Hadoop cluster.

a) **masters**

b) **slaves**

c) **core-site.xml**

d) **mapred-site.xml**

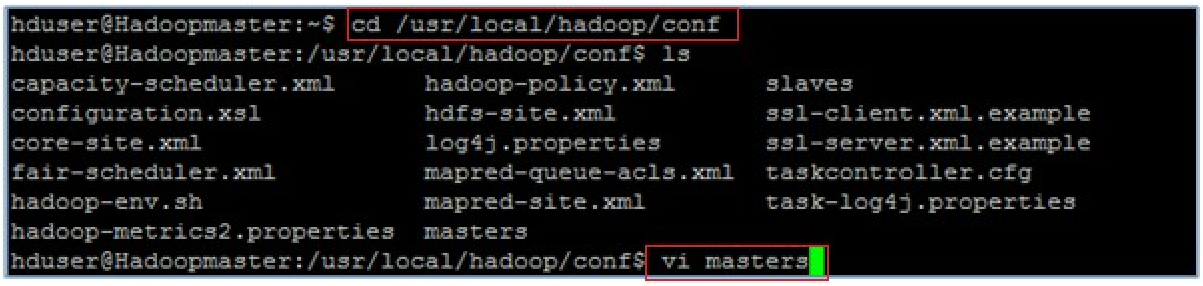
e) **hdfs-site.xml**

Lets configure each and every config file accordingly:

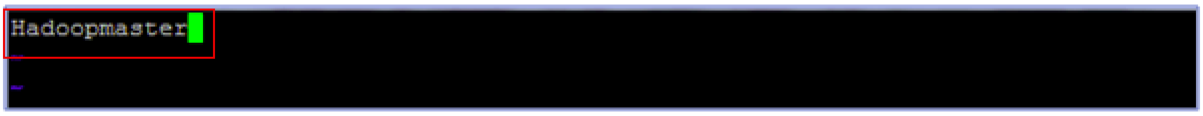
a. masters:

In master (Hadoopmaster) machine we need to configure masters file accordingly as shown in the image and add the master (Hadoopmaster) node name.

```
vi masters
Hadoopmaster
```



```
hduser@Hadoopmaster:~$ cd /usr/local/hadoop/conf
hduser@Hadoopmaster:/usr/local/hadoop/conf$ ls
capacity-scheduler.xml       hadoop-policy.xml      slaves
configuration.xsl            hdfs-site.xml          ssl-client.xml.example
core-site.xml                log4j.properties       ssl-server.xml.example
fair-scheduler.xml           mapred-queue-acls.xml  taskcontroller.cfg
hadoop-env.sh                mapred-site.xml        task-log4j.properties
hadoop-metrics2.properties   masters
hduser@Hadoopmaster:/usr/local/hadoop/conf$ vi masters
```

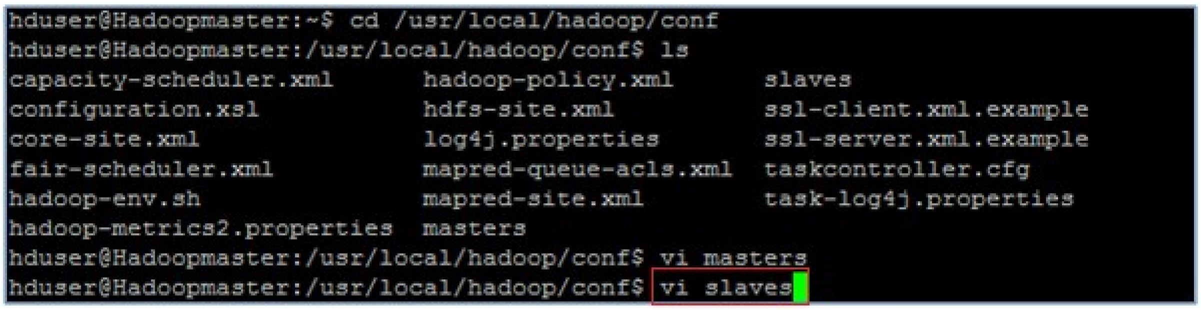


```
Hadoopmaster
~
~
```

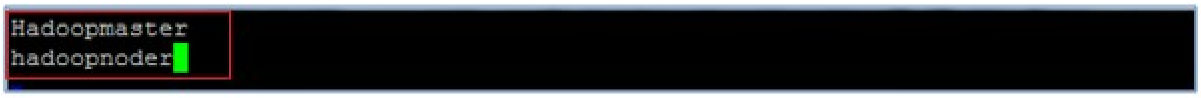
b. slaves:

Lists the hosts, one per line, where the Hadoop slave daemons (DataNodes and TaskTrackers) will be running as shown:

```
Hadoomaster
hadoopnode
```



```
hduser@Hadoopmaster:~$ cd /usr/local/hadoop/conf
hduser@Hadoopmaster:/usr/local/hadoop/conf$ ls
capacity-scheduler.xml      hadoop-policy.xml      slaves
configuration.xsl           hdfs-site.xml          ssl-client.xml.example
core-site.xml               log4j.properties       ssl-server.xml.example
fair-scheduler.xml          mapred-queue-acls.xml  taskcontroller.cfg
hadoop-env.sh               mapred-site.xml        task-log4j.properties
hadoop-metrics2.properties  masters
hduser@Hadoopmaster:/usr/local/hadoop/conf$ vi masters
hduser@Hadoopmaster:/usr/local/hadoop/conf$ vi slaves
```



```
Hadoopmaster
hadoopnode
```

If you have additional slave nodes, just add them to the conf/slaves file, one hostname per line.

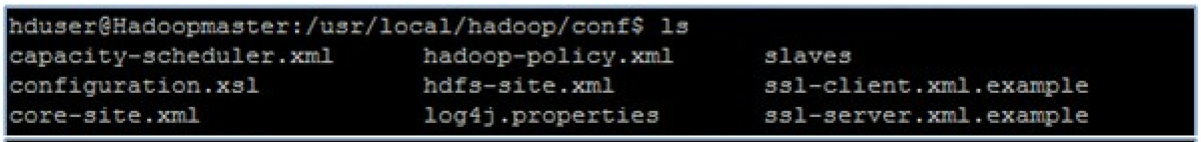
Configuring all *-site.xml files:

We need to use the same configurations on all the nodes of hadoop cluster, i.e. we need to edit all *-site.xml files on each and every server accordingly.

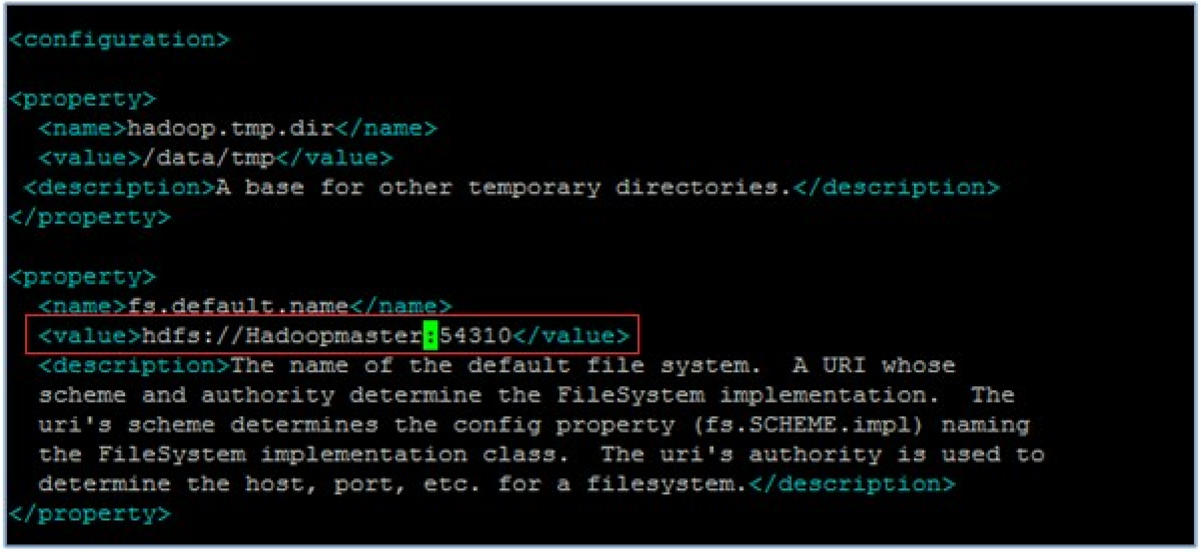
c. core-site.xml:

We are changing the host name from ‘localhost’ to Hadoopmaster, which specifies the NameNode (the HDFS master) host and port.

```
vi core-site.xml
```



```
hduser@Hadoopmaster:/usr/local/hadoop/conf$ ls
capacity-scheduler.xml      hadoop-policy.xml      slaves
configuration.xsl           hdfs-site.xml          ssl-client.xml.example
core-site.xml               log4j.properties       ssl-server.xml.example
```

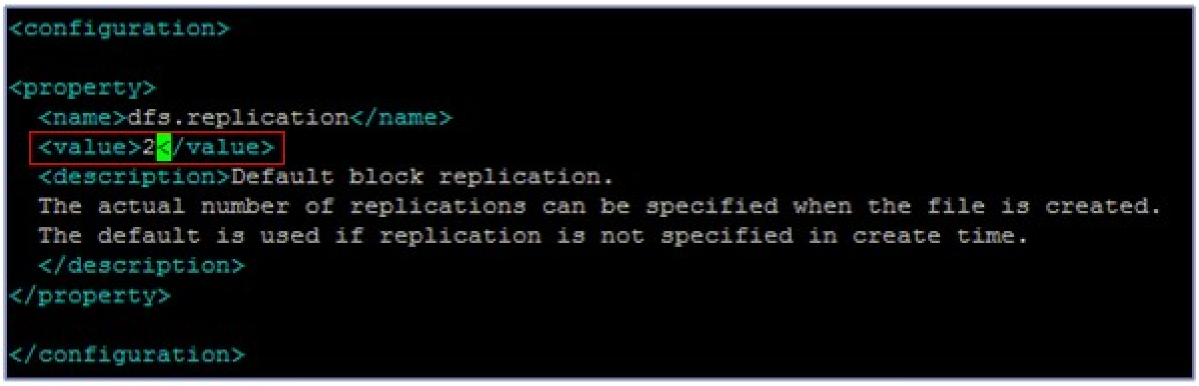


```
<configuration>

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

<property>
  <name>fs.default.name</name>
  <value>hdfs://Hadoopmaster:54310</value>
  <description>The name of the default file system.  A URI whose
  scheme and authority determine the FileSystem implementation.  The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class.  The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>
```

d. hdfs-site.xml:

We are changing the replication factor to “2”, The default value of dfs.replication is 3. However, we have only two nodes available, so we set dfs.replication to 2.

```
vi hdfs-site.xml
```

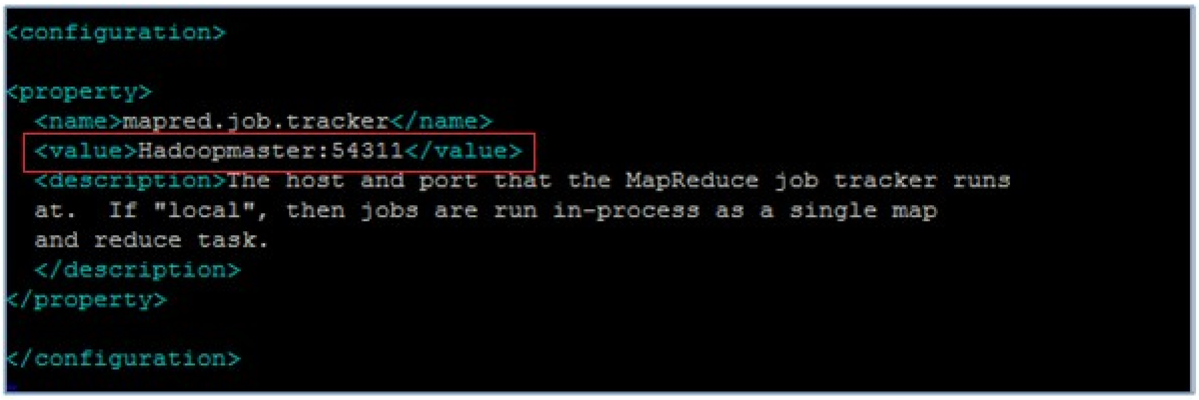


```
<configuration>

<property>
  <name>dfs.replication</name>
  <value>2</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>

</configuration>
```

e. mapred-site.xml:

We are changing the host name from ‘localhost’ to Hadoopmaster, which specifies the JobTracker (MapReduce master) host and port

```
vi mapred-site.xml
```

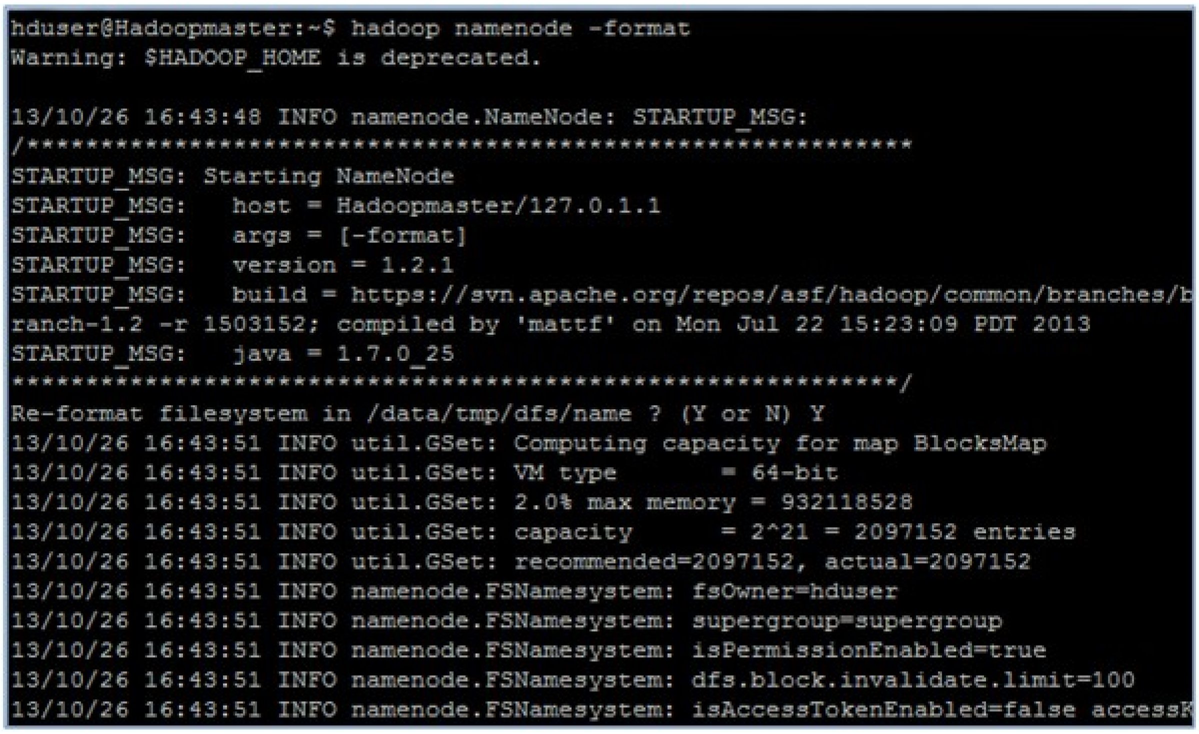


```
<configuration>

<property>
  <name>mapred.job.tracker</name>
  <value>Hadoopmaster: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>
```

3. Formatting and Starting/Stopping the HDFS filesystem via the NameNode:

The first step to starting up your multi–node Hadoop cluster is formatting the Hadoop filesystem which is implemented on top of the local filesystem of your cluster. To format the filesystem (which simply initializes the directory specified by the dfs.name.dir variable), run the given command.

```
hadoop namenode –format
```



```
hduser@Hadoopmaster:~$ hadoop namenode -format
Warning: $HADOOP_HOME is deprecated.

13/10/26 16:43:48 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = Hadoopmaster/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.2.1
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/b
ranch-1.2 -r 1503152; compiled by 'mattf' on Mon Jul 22 15:23:09 PDT 2013
STARTUP_MSG:   java = 1.7.0_25
************************************************************/
Re-format filesystem in /data/tmp/dfs/name ? (Y or N) Y
13/10/26 16:43:51 INFO util.GSet: Computing capacity for map BlocksMap
13/10/26 16:43:51 INFO util.GSet: VM type       = 64-bit
13/10/26 16:43:51 INFO util.GSet: 2.0% max memory = 932118528
13/10/26 16:43:51 INFO util.GSet: capacity      = 2^21 = 2097152 entries
13/10/26 16:43:51 INFO util.GSet: recommended=2097152, actual=2097152
13/10/26 16:43:51 INFO namenode.FSNamesystem: fsOwner=hduser
13/10/26 16:43:51 INFO namenode.FSNamesystem: supergroup=supergroup
13/10/26 16:43:51 INFO namenode.FSNamesystem: isPermissionEnabled=true
13/10/26 16:43:51 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
13/10/26 16:43:51 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessK
```

4. Starting the multi-node cluster:

Starting the cluster is performed in two steps.

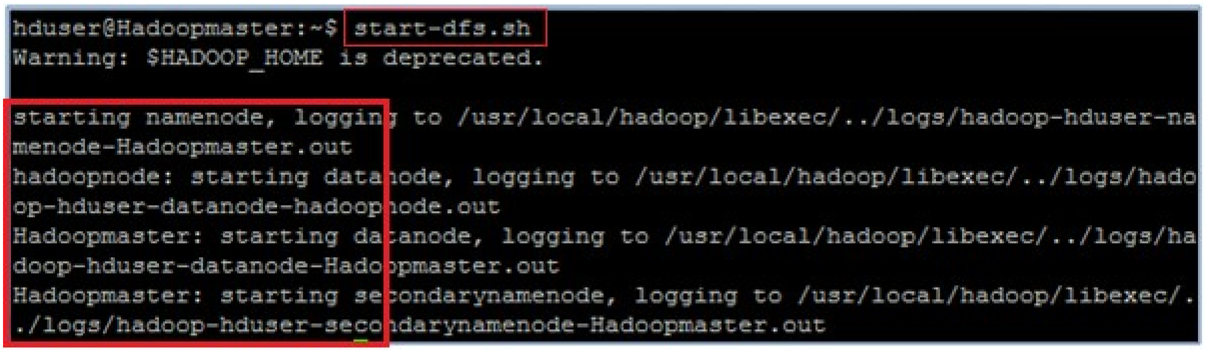
We begin by starting the HDFS daemons first, the NameNode daemon is started on Hadoopmaster and DataNode daemons are started on all nodes(slaves).

Then we will start the MapReduce daemons, the JobTracker is started on Hadoomaster and TaskTracker daemons are started on all nodes (slaves).

a. To start HDFS daemons:

```
start-dfs.sh
```

This will get NameNode up and DataNodes up listed in conf/slaves.



```
hduser@Hadoopmaster:~$ start-dfs.sh
Warning: $HADOOP_HOME is deprecated.

starting namenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-na
menode-Hadoopmaster.out
hadoopnode: starting datanode, logging to /usr/local/hadoop/libexec/../logs/hado
op-hduser-datanode-hadoopnode.out
Hadoopmaster: starting datanode, logging to /usr/local/hadoop/libexec/../logs/ha
doop-hduser-datanode-Hadoopmaster.out
Hadoopmaster: starting secondarynamenode, logging to /usr/local/hadoop/libexec/.
./logs/hadoop-hduser-secondarynamenode-Hadoopmaster.out
```

b. To start Map Red daemons:

```
start-mapred.sh
```

This will bring up the MapReduce cluster with the JobTracker running on the machine you ran the previous command on, and TaskTrackers on the machines listed in the conf/slaves file.

5. Running a Map-reduce Job:

Use a much larger volume of data as inputs as we are running in a cluster.

```
hadoop jar hadoop *examples*. jar wordcount  /user/hduser/demo 
/user/hduser/demo-output
```

we can observe namenode,mapreduce,tasktracker process on the webinterface by following given url's

- http://Hadoopmaster:50070/ – web UI of the NameNode daemon
- http://Hadoopmaster:50030/ – web UI of the JobTracker daemon
- http://Hadoopmaster:50060/ – web UI of the TaskTracker daemon

*Hadoopmaster can be replaced with the machine ip