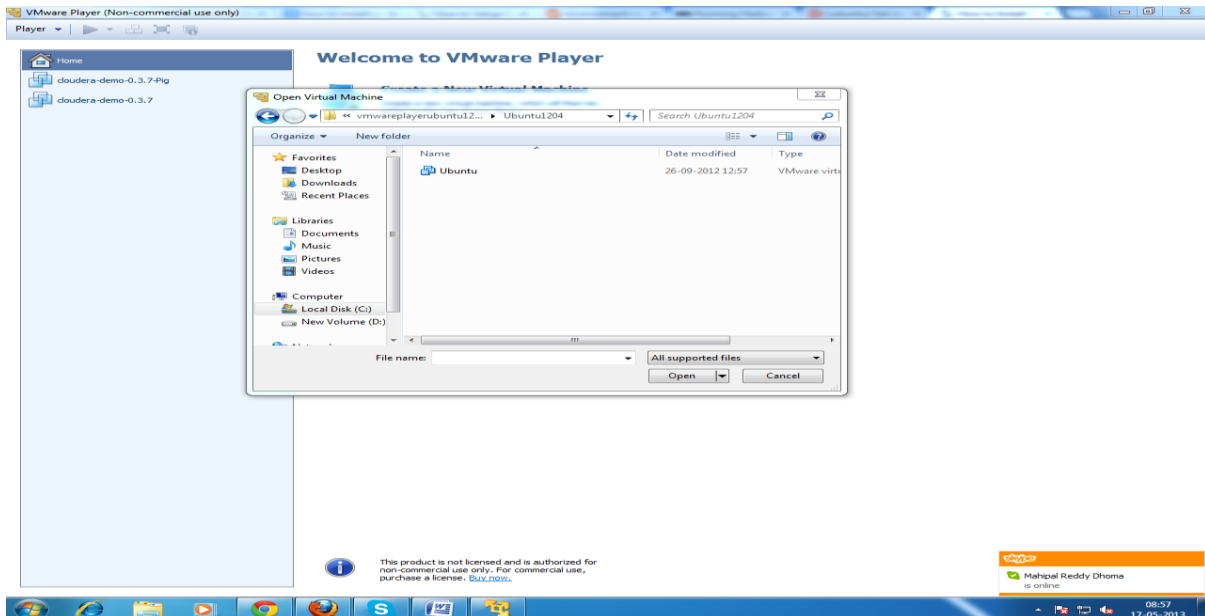
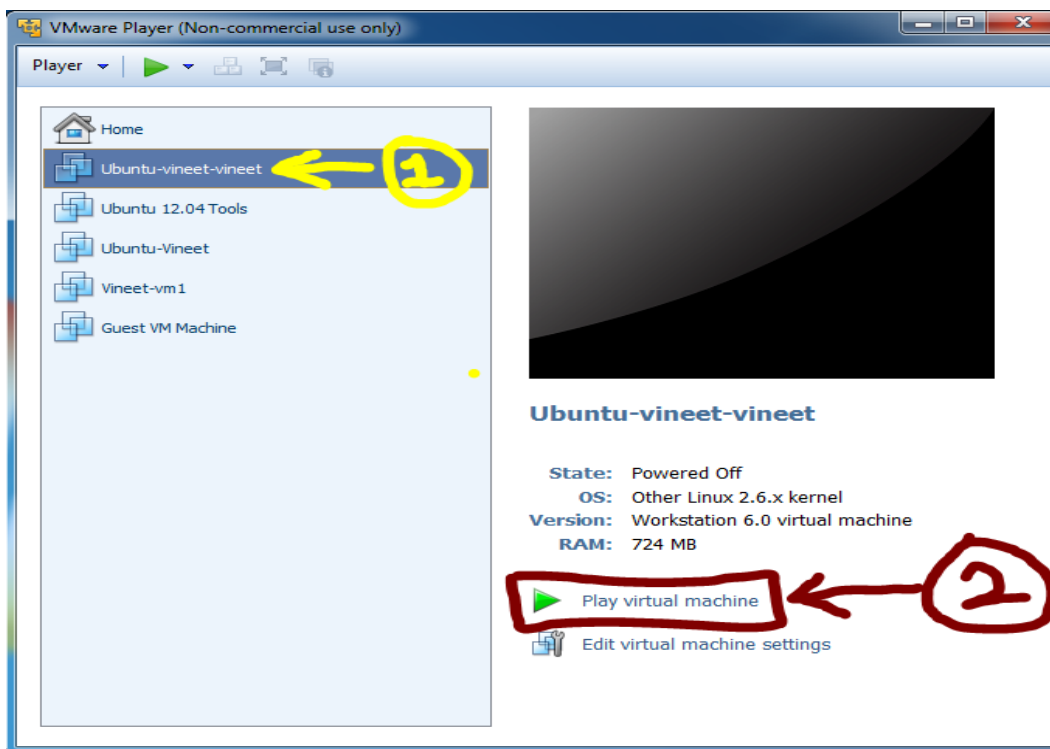


Hadoop Installation with Multiple DataNode

- Go to the below link and download the image of ubuntu 12.04
 - <http://www.traffictool.net/vmware/ubuntu1204t.html>
- Open VMware Player and click open virtual machine and select path where you have extracted image of Ubuntu. After that select the .vmx file and click ok.



- Now you can see the below screen in VMware Player.

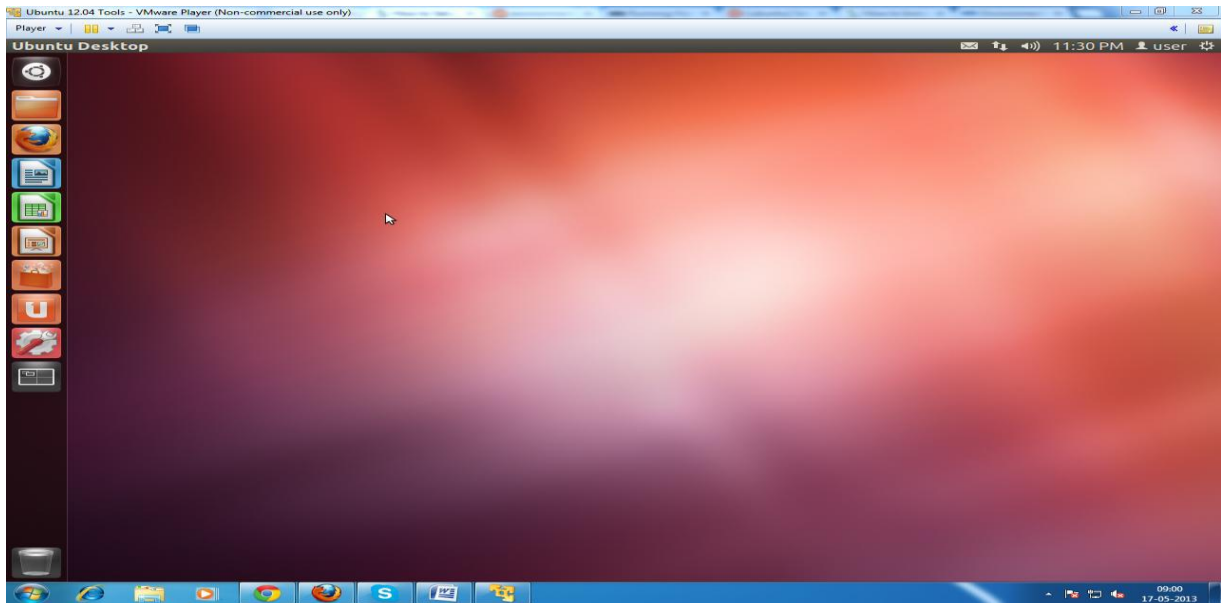


edureka!

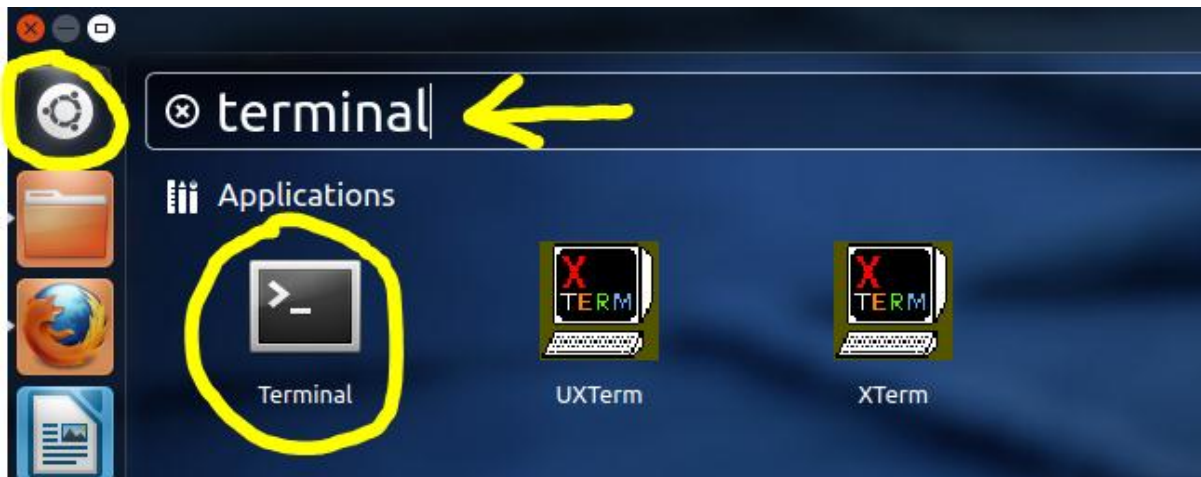
- Double click on ubuntu present in VMware Player. You will get a screen of the below image.

Username : **user**

Password : **password**



- Open a Terminal



- Update the repository:
 - **Command:** `sudo apt-get update`

```
user@ubuntu:~$ sudo apt-get update
```

- Once the Update is complete :
- **Command:** `sudo apt-get install openjdk-6-jdk`

```
user@ubuntu:~$ sudo apt-get install openjdk-6-jdk
```

- After Java has been Installed, To check whether Java is installed on your system or not give the below command :
- **Command:** `java -version`

```
user@ubuntu:~$ java -version
java version "1.6.0_27"
OpenJDK Runtime Environment (IcedTea6 1.12.5) (6b27-1.12.5-0ubuntu0.12.04.1)
OpenJDK Client VM (build 20.0-b12, mixed mode, sharing)
user@ubuntu:~$
```

- **Install openssh-server:**
- **Command:** `sudo apt-get install openssh-server`

```
user@ubuntu:~$ sudo apt-get install openssh-server
[sudo] password for user:
```

```
user@ubuntu:~$ sudo apt-get install openssh-server
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  openssh-client ssh-import-id
Suggested packages:
  libpam-ssh keychain monkeysphere openssh-blacklist openssh-blacklist-extra
  rssh molly-guard
The following NEW packages will be installed:
  openssh-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 2 newly installed, 0 to remove and 562 not upgraded.
Need to get 1,309 kB of archives.
After this operation, 891 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

- **Download and extract Hadoop:**
- **Command:** `wget http://archive.apache.org/dist/hadoop/core/hadoop-1.2.0/hadoop-1.2.0.tar.gz`

- Command: `tar -xvf hadoop-1.2.0.tar.gz`

```
user@ubuntu:~$ ls
Desktop      examples.desktop  Music          Templates
Documents    hadoop-1.2.0      Pictures       Videos
Downloads    hadoop-1.2.0.tar.gz Public
```

Configuration

- Add JAVA_HOME in hadoop-env.sh file:
- Command: `sudo gedit hadoop-1.2.0/conf/hadoop-env.sh`
- Type : `export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-i386`

Uncomment the below shown export and add the below the path to your JAVA_HOME:

```
*hadoop-env.sh ✕
# Set Hadoop-specific environment variables here.

# The only required environment variable is
# JAVA_HOME. All others are
# optional. When running a distributed
# configuration it is best to
# set JAVA_HOME in this file, so that it is
# correctly defined on
# remote nodes.

# The java implementation to use. Required.
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-i386|

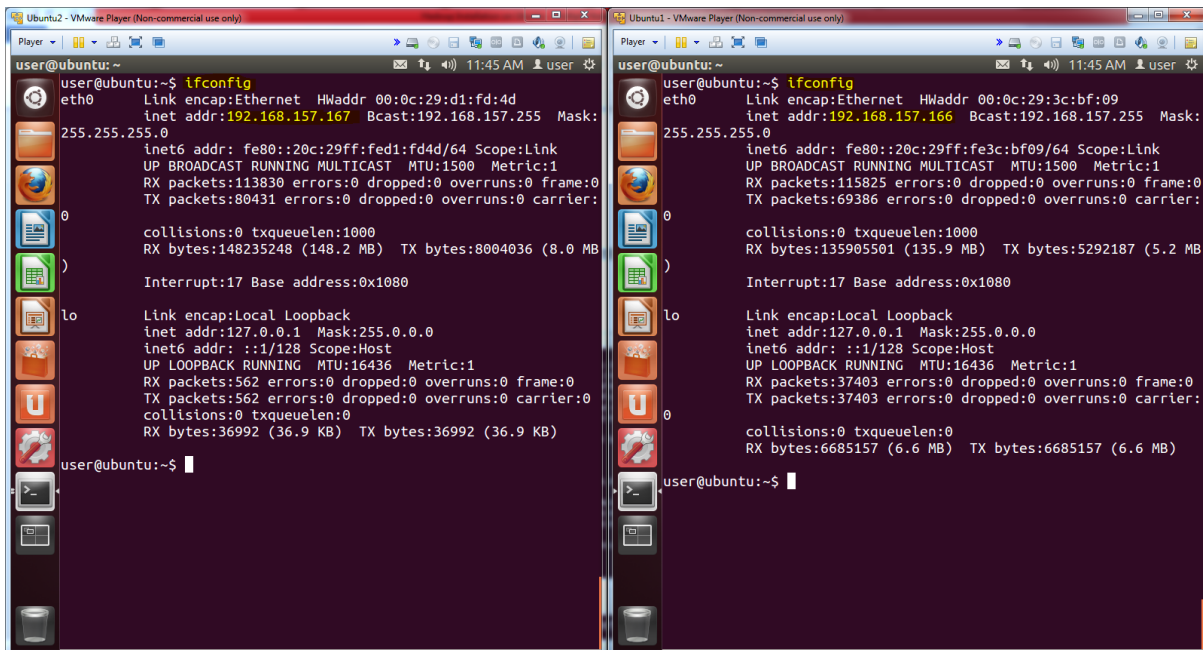
# Extra Java CLASSPATH elements. Optional.
# export HADOOP_CLASSPATH=
```

Now create another instance of Ubuntu VM and start it on VMPlayer.

[Important – Do all the above steps for second Image of Ubuntu]

edureka!

- Getting ip addresses for both VMs:
- Command: ifconfig
- Command: sudo gedit /etc/hosts



The image shows two side-by-side terminal windows from the Ubuntu2 - VMware Player. Both windows display the output of the 'ifconfig' command. The left window shows the network configuration for 'eth0' with IP address 192.168.157.167 and 'lo' with IP address 127.0.0.1. The right window shows the network configuration for 'eth0' with IP address 192.168.157.166 and 'lo' with IP address 127.0.0.1. Both windows also show statistics for RX and TX packets and bytes.

```
user@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:d1:fd:4d
          inet addr:192.168.157.167  Bcast:192.168.157.255  Mask:
          255.255.255.0
          inet6 addr: fe80::20c:29ff:fed1:fd4d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:113830 errors:0 dropped:0 overruns:0 frame:0
          TX packets:80431 errors:0 dropped:0 overruns:0 carrier:
          0
          collisions:0 txqueuelen:1000
          RX bytes:148235248 (148.2 MB)  TX bytes:8004036 (8.0 MB)

          Interrupt:17 Base address:0x1080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:562 errors:0 dropped:0 overruns:0 frame:0
          TX packets:562 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:36992 (36.9 KB)  TX bytes:36992 (36.9 KB)

user@ubuntu:~$
```

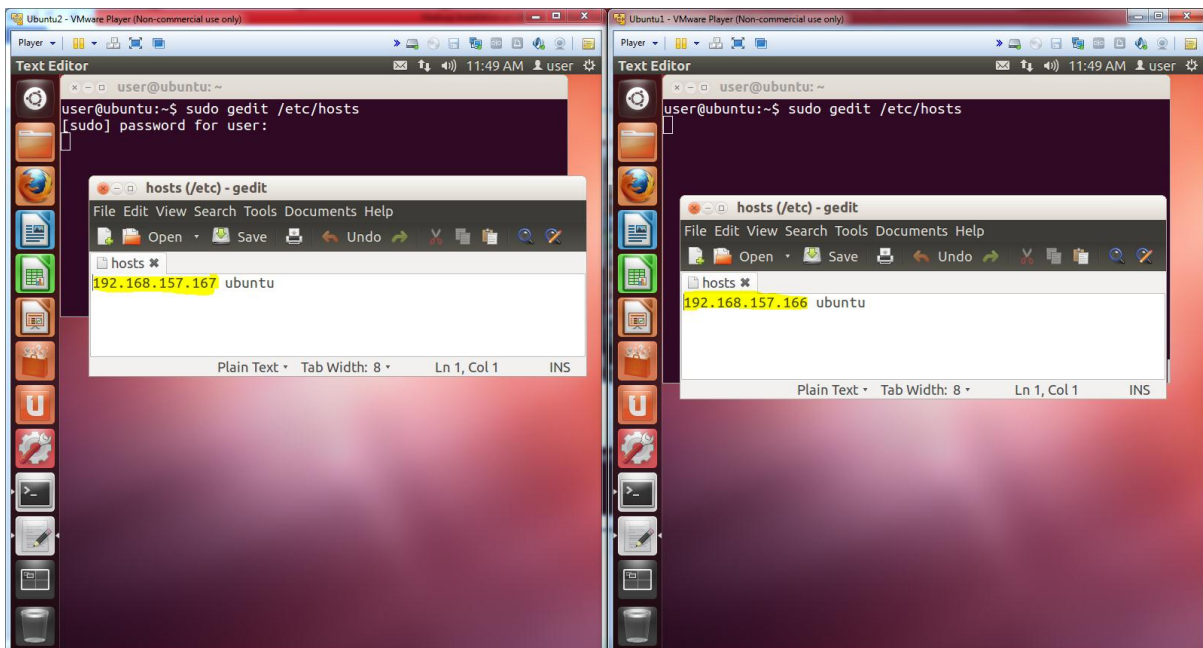
```
user@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:3c:bf:09
          inet addr:192.168.157.166  Bcast:192.168.157.255  Mask:
          255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3c:bf09/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115825 errors:0 dropped:0 overruns:0 frame:0
          TX packets:69386 errors:0 dropped:0 overruns:0 carrier:
          0
          collisions:0 txqueuelen:1000
          RX bytes:135905501 (135.9 MB)  TX bytes:5292187 (5.2 MB)

          Interrupt:17 Base address:0x1080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:37403 errors:0 dropped:0 overruns:0 frame:0
          TX packets:37403 errors:0 dropped:0 overruns:0 carrier:
          0
          collisions:0 txqueuelen:0
          RX bytes:6685157 (6.6 MB)  TX bytes:6685157 (6.6 MB)

user@ubuntu:~$
```

Getting ip addresses for both VMs:



The image shows two side-by-side terminal windows from the Ubuntu2 - VMware Player. Both windows display the output of the 'sudo gedit /etc/hosts' command. The left window shows the 'hosts' file being edited with the IP address 192.168.157.167 assigned to 'ubuntu'. The right window shows the 'hosts' file being edited with the IP address 192.168.157.166 assigned to 'ubuntu'.

```
user@ubuntu:~$ sudo gedit /etc/hosts
[sudo] password for user:
hosts (/etc) - gedit
File Edit View Search Tools Documents Help
192.168.157.167 ubuntu
Plain Text  Tab Width: 8  Ln 1, Col 1  INS
```

```
user@ubuntu:~$ sudo gedit /etc/hosts
hosts (/etc) - gedit
File Edit View Search Tools Documents Help
192.168.157.166 ubuntu
Plain Text  Tab Width: 8  Ln 1, Col 1  INS
```

Select a VM to assign as master and note its ip address.

I have chosen VM 1 whose ip address is 192.168.157.166

edureka!

[Note: core-site.xml, mapred-site.xml and hdfs-site.xml are same for both VMs, you need to set above 3 files on both VM]

➤ Edit core-site.xml:

- Command: `sudo gedit Hadoop-1.2.0/conf/core-site.xml`

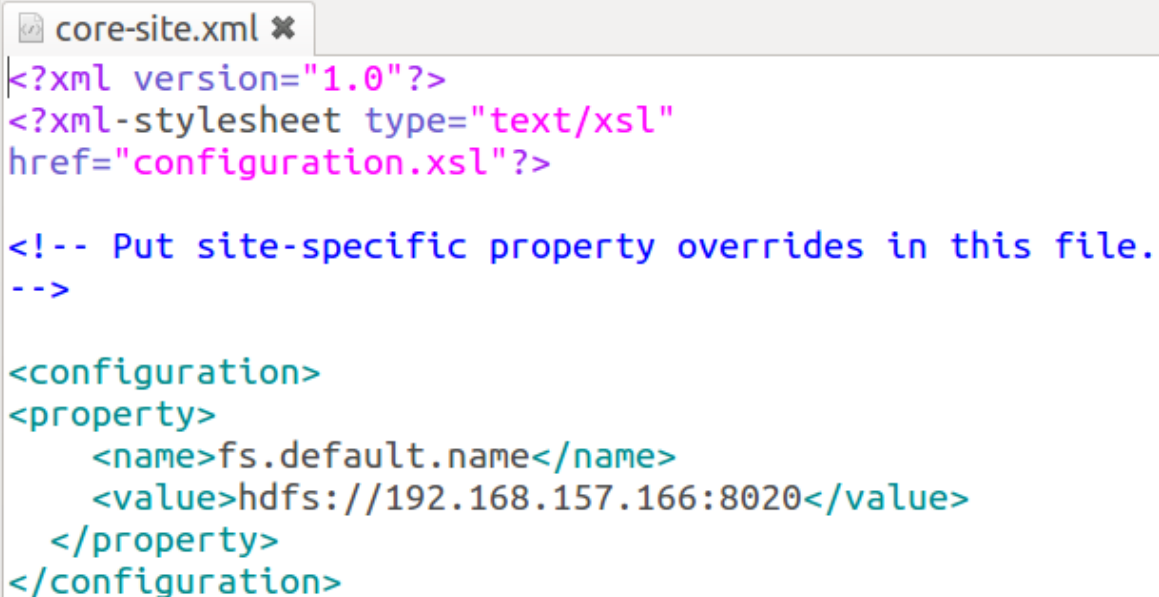
`<property>`

`<name>fs.default.name</name>`

`<value>hdfs:// 192.168.157.166:8020</value>`

`</property>`

```
user@ubuntu:~$ sudo gedit hadoop-1.2.0/conf/core-site.xml
[sudo] password for user:
```



```
core-site.xml ✕
<?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://192.168.157.166:8020</value>
</property>
</configuration>
```

➤ Edit hdfs-site.xml:

- Command: `sudo gedit Hadoop-1.2.0/conf/hdfs-site.xml`

```
<property>

  <name>dfs.replication</name>

  <value>2</value>

</property>

<property>

  <name>dfs.permissions</name>

  <value>false</value>

</property>
```

```
user@ubuntu:~$ sudo gedit hadoop-1.2.0/conf/hdfs-site.xml
user@ubuntu:~$
```



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

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

<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.permissions</name>
  <value>>false</value>
</property>
</configuration>
```

➤ Edit mapred-site.xml:

- Command: `sudo gedit Hadoop-1.2.0/conf/mapred -site.xml`

```
<property>

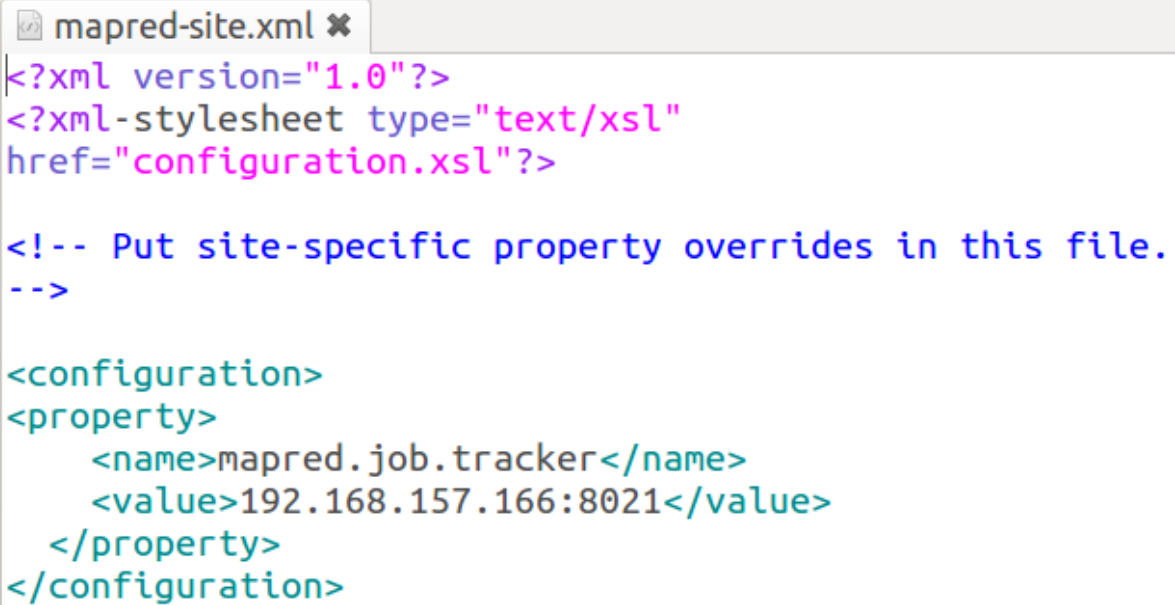
  <name>mapred.job.tracker</name>

  <value>192.168.157.166:8021</value>

</property>
```

edureka!

```
user@ubuntu:~$ sudo gedit hadoop-1.2.0/conf/mapred-site.xml
user@ubuntu:~$
```

A screenshot of a text editor window titled 'mapred-site.xml'. The window displays XML configuration code for MapReduce. The code includes an XML declaration, a stylesheet reference, a comment about site-specific overrides, and a configuration block with a single property for the mapred.job.tracker.

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

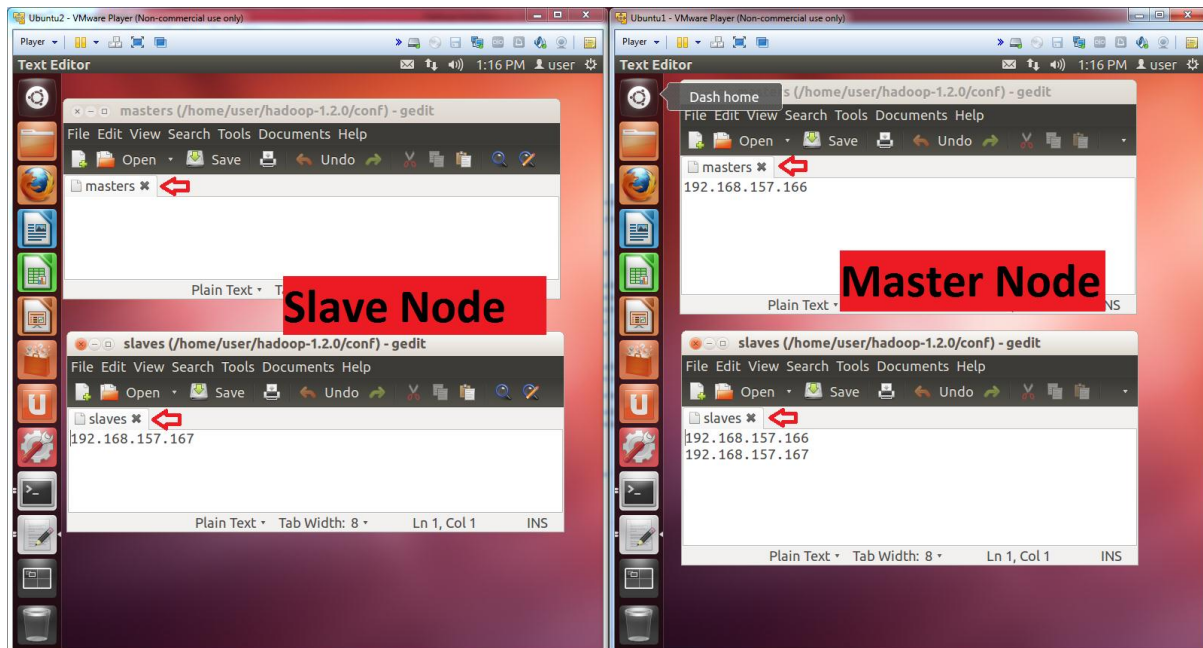
<!-- Put site-specific property overrides in this file.
-->

<configuration>
<property>
  <name>mapred.job.tracker</name>
  <value>192.168.157.166:8021</value>
</property>
</configuration>
```


Now there is a slight difference in slaves and masters file for both VM.

On master node, masters file contains master node's ip address only and slaves file contains ip addresses of both vms.

On slave node, master file is blank and slaves file contains slave VM's ip address. See the image below.



➤ **Create a ssh key:**

- **Command:** `ssh-keygen -t rsa -P ""`

```
user@ubuntu:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
77:2e:8e:c8:21:69:1e:04:b4:14:b5:12:6a:1a:9a:5f user@ubuntu
The key's randomart image is:
+--[ RSA 2048 ]-----+
|  =O.                  |
| + O .                 |
| O.+ .                 |
|+O O                   |
|+ E S . .              |
| . O . . . O           |
| . = . . .             |
|  O + O O .            |
| . O . .               |
+-----+
user@ubuntu:~$
```

➤ **Creating a password-less ssh login:**

- **Command:** `ssh-copy-id -i $HOME/.ssh/id_rsa.pub user@192.168.157.166`
- **Command:** `ssh-copy-id -i $HOME/.ssh/id_rsa.pub user@192.168.157.167`

Run the below commands on Master node.

➤ **Format the name node**

- **Command:** `bin/hadoop namenode -format`

➤ **Start the namenode, datanode and job tracker**

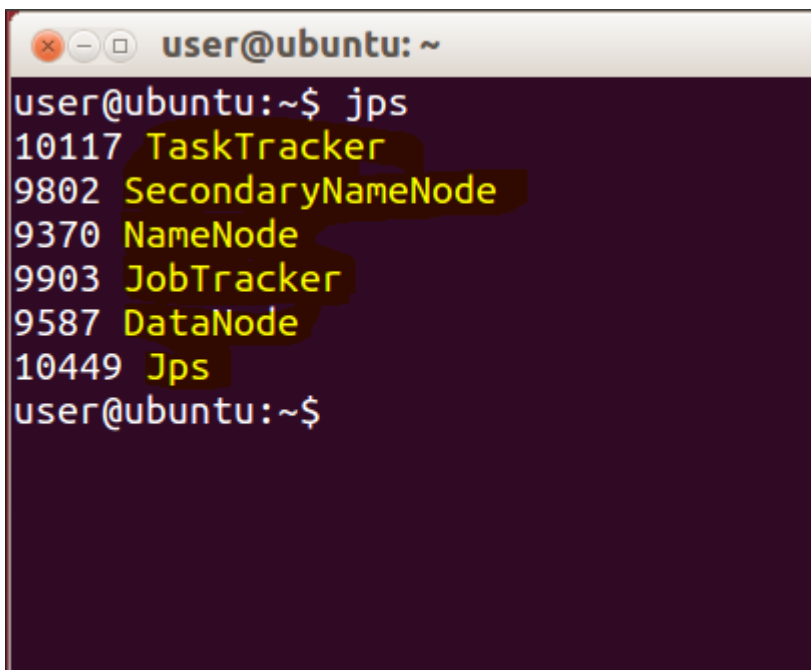
- **Command:** `bin/start-dfs.sh`

➤ **Start the task tracker**

- **Command:** `bin/start-mapred.sh`

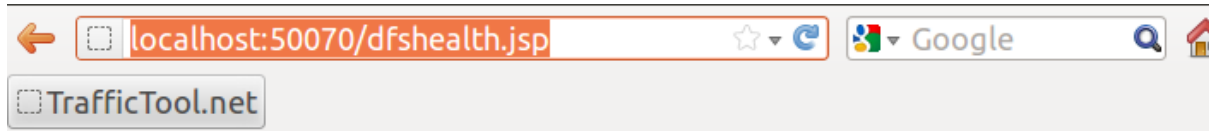
➤ **To check if Hadoop started correctly**

- **Command:** `jps`

A screenshot of a terminal window titled 'user@ubuntu: ~'. The terminal shows the command 'jps' being executed, resulting in the following output: '10117 TaskTracker', '9802 SecondaryNameNode', '9370 NameNode', '9903 JobTracker', '9587 DataNode', and '10449 Jps'. The prompt 'user@ubuntu:~\$' is visible at the bottom.

```
user@ubuntu:~$ jps
10117 TaskTracker
9802 SecondaryNameNode
9370 NameNode
9903 JobTracker
9587 DataNode
10449 Jps
user@ubuntu:~$
```

Open browser and type <http://localhost:50070/dfshealth.jsp> to see the current live node.



NameNode 'ubuntu:8020'

Started: Mon Jun 03 10:50:11 EDT 2013
Version: 1.2.0, r1479473
Compiled: Mon May 6 06:59:37 UTC 2013 by hortonfo
Upgrades: There are no upgrades in progress.

[Browse the filesystem](#)
[Namenode Logs](#)

Cluster Summary

6 files and directories, 1 blocks = 7 total. Heap Size is 31.57 MB / 966.69 MB (3%)

Configured Capacity	:	78.56 GB
DFS Used	:	68 KB
Non DFS Used	:	9.9 GB
DFS Remaining	:	68.66 GB
DFS Used%	:	0 %
DFS Remaining%	:	87.39 %
Live Nodes	:	2
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

That's it....You did it. You can see both nodes running.