How to install Apache Hadoop 2.6.0 in Ubuntu (Single node setup)

Since we know it's the time for parallel computation to tackle large amount of dataset, we will require Apache Hadoop (here the name is derived from Elephant). As Apache Hadoop is the top most contributed Apache project, more and more features are implemented as well as more and more bugs are getting fixed in new coming versions. So, by considering this situation we need to follow slightly different steps than previous version. Here, I am trying to covering full fledge Hadoop installation steps for BigData enthusiasts who wish to install Apache Hadoop on their Ubuntu – Linux machine.

This blog post teaches how to install Apache Hadoop 2.6 over Ubuntu machine. (You can follow the same blog post for installation over Ubuntu server machine). To get started with Apache Hadoop install, I recommend that you should have knowledge of basic Linux commands which will be helpful in normal operations while installation task.

If you are looking for instructions over how to setup Hadoop Multinode cluster, visit my next post – http://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/

Prerequisites

1. **Installing Oracle Java 8**

   Apache Hadoop is java framework, we need java installed on our machine to get it run over operating system. Hadoop supports all java version greater than 5 (i.e. `Java 1.5`). So, Here you can also try Java 6, 7 instead of Java 8.

   ```
   vignesh@pingax:~$ sudo add-apt-repository ppa:webupd8team/java
   vignesh@pingax:~$ sudo apt-get update
   vignesh@pingax:~$ sudo apt-get install oracle-java8-installer
   ```

   It will install java source in your machine at `/usr/lib/jvm/java-8-oracle`

   To verify your java installation, you have to fire the following command like,

   ```
   vignesh@pingax:~$ java -version
   ```

2. **Creating a Hadoop user for accessing HDFS and MapReduce**
   To avoid security issues, we recommend to setup new Hadoop user group and user account to deal with all Hadoop related activities.

   We will create hadoop as system group and hduser as system user by,

   ```
   vignesh@pingax:~$ sudo addgroup hadoop
   vignesh@pingax:~$ sudo adduser --ingroup hadoop hduser
   ```

3. **Installing SSH**
   SSH ("Secure SHell") is a protocol for securely accessing one machine from another. Hadoop uses SSH for accessing another slaves nodes to start and manage all HDFS and MapReduce

daemons.

```
vignesh@pingax:~$ sudo apt-get install openssh-server
```

Now, we have installed SSH over Ubuntu machine so we will be able to connect with this machine as well as from this machine remotely.

Configuring SSH
Once you installed SSH on your machine, you can connect to other machine or allow other machines to connect with this machine. However we have this single machine, we can try connecting with this same machine by SSH. To do this, we need to copy generated RSA key (i.e. id_rsa.pub) pairs to authorized_keys folder of SSH installation of this machine by the following command,

```
# First login with hduser (and from now use only hduser account for further
steps)
vignesh@pingax:~$ sudo su hduser

# Generate ssh key for hduser account
hduser@pingax:~$ ssh-keygen -t rsa -P ""

## Copy id_rsa.pub to authorized keys from hduser
hduser@pingax:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

In case you are configuring SSH for another machine (i.e. from master node to slave node), you have to update the above command by adding the hostname of slave machine.

4. **Disabling IPv6**
   Since Hadoop doesn’t work on IPv6, we should disable it. One of another reason is also that it has been developed and tested on IPv4 stacks. Hadoop nodes will be able to communicate if we are having IPv4 cluster. (Once you have disabled IPV6 on your machine, you need to reboot your machine in order to check its effect. In case if you don’t know how to reboot with command use sudo reboot)

   For getting your IPv6 disable in your Linux machine, you need to update */etc/sysctl.conf* by adding following line of codes at end of the file,

```
# disable ipv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Tip:- You can use nano, gedit, and Vi editor for updating all text files for this configuration purpose.

Installation Steps

1. **Download latest Apache Hadoop source from Apache mirrors**
   First you need to download Apache Hadoop 2.6.0 (i.e. *hadoop-2.6.0.tar.gz*)or latest version

source from Apache download Mirrors. You can also try stable hadoop to get all latest features as well as recent bugs solved with Hadoop source. Choose location where you want to place all your hadoop installation, I have chosen */usr/local/hadoop*

```
## Locate to hadoop installation parent dir
hduser@pingax:~$ cd /usr/local/

## Extract Hadoop source
sudo tar -xzvf hadoop-2.6.0.tar.gz

## Move hadoop-2.6.0 to hadoop folder
sudo mv hadoop-2.6.0 /usr/local/hadoop

## Assign ownership of this folder to Hadoop user
sudo chown hduser:hadoop -R /usr/local/hadoop

## Create Hadoop temp directories for Namenode and Datanode
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode

## Again assign ownership of this Hadoop temp folder to Hadoop user
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

2. **Update Hadoop configuration files**

User profile : Update $HOME/.bashcr

```
## User profile : Update $HOME/.bashrc
hduser@pingax:~$ sudo gedit .bashrc

## Update hduser configuration file by appending the
## following environment variables at the end of this file.

# -- HADOOP ENVIRONMENT VARIABLES START -- #
export JAVA_HOME=/usr/lib/jvm/java-8-oracle
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
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=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
# -- HADOOP ENVIRONMENT VARIABLES END -- #
```

Configuration file : hadoop-env.sh

```
## To edit file, fire the below given command
hduser@pingax:/usr/local/hadoop/etc/hadoop$ sudo gedit hadoop-env.sh

## Update JAVA_HOME variable,
JAVA_HOME=/usr/lib/jvm/java-8-oracle
```

Configuration file : core-site.xml

```
## To edit file, fire the below given command
hduser@pingax:/usr/local/hadoop/etc/hadoop$ sudo gedit core-site.xml

## Paste these lines into <configuration> tag
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
```

Configuration file : hdfs-site.xml

```
## To edit file, fire the below given command
hduser@pingax:/usr/local/hadoop/etc/hadoop$ sudo gedit hdfs-site.xml

## Paste these lines into <configuration> tag
<property>
      <name>dfs.replication</name>
      <value>1</value>
 </property>
 <property>
      <name>dfs.namenode.name.dir</name>
      <value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
 </property>
 <property>
      <name>dfs.datanode.data.dir</name>
      <value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
 </property>
```

Configuration file : yarn-site.xml

```
## To edit file, fire the below given command
hduser@pingax:/usr/local/hadoop/etc/hadoop$ sudo gedit yarn-site.xml

## Paste these lines into <configuration> tag
<property>
      <name>yarn.nodemanager.aux-services</name>
      <value>mapreduce_shuffle</value>
</property>
<property>
      <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
      <value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>
```

Configuration file : mapred-site.xml

```
## Copy template of mapred-site.xml.template file
cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/mapred-site.xml

## To edit file, fire the below given command
hduser@pingax:/usr/local/hadoop/etc/hadoop$ sudo gedit mapred-site.xml

## Paste these lines into <configuration> tag
<property>
      <name>mapreduce.framework.name</name>
      <value>yarn</value>
</property>
```

3. **Format Namenode**

   ```
   hduser@pingax:hdfs namenode -format
   ```

4. **Start all Hadoop daemons**

   Start hdfs daemons

   ```
   hduser@pingax:/usr/local/hadoop$ start-dfs.sh
   ```

   Start MapReduce daemons:

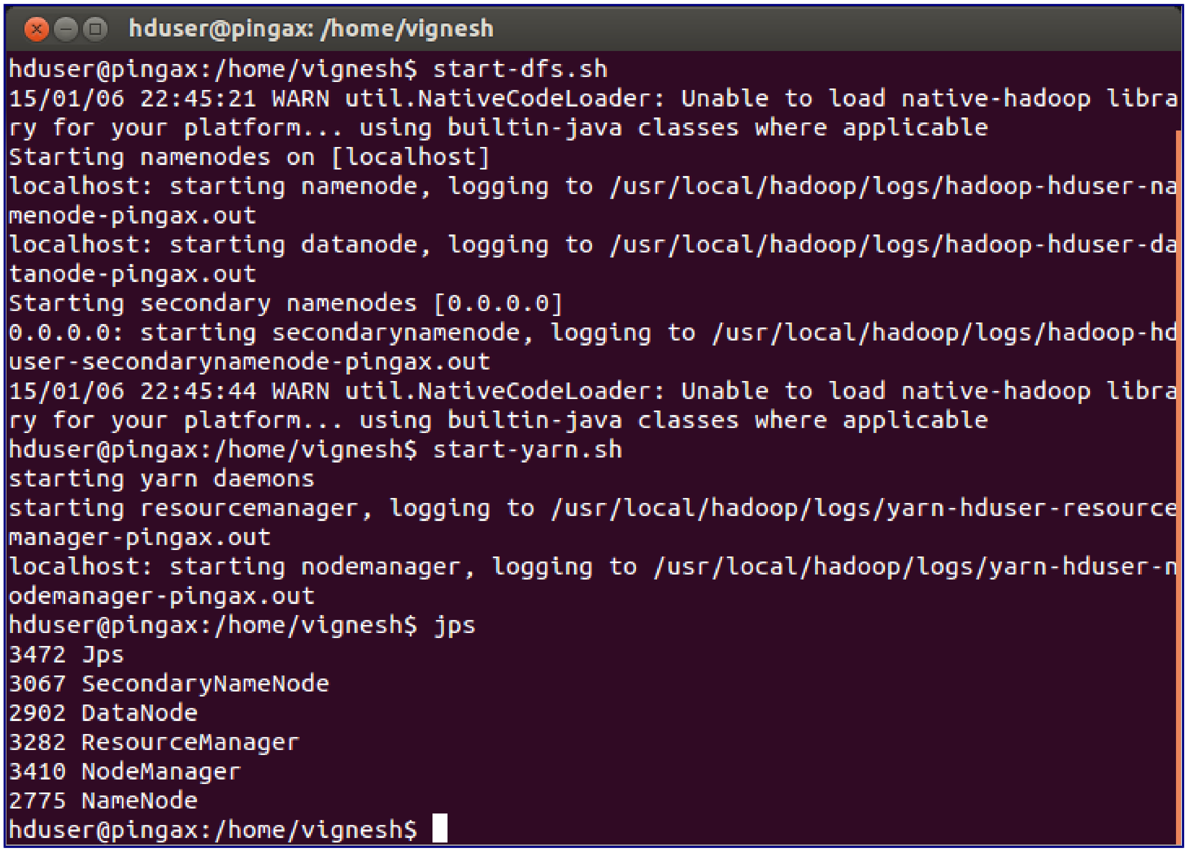
   ```
   hduser@pingax:/usr/local/hadoop$ start-yarn.sh
   ```

   Instead both of these above command you can also use *start-all.sh*, but its now deprecated so its not recommended to be used for better Hadoop operations.

5. **Track/Monitor/Verify**

   Verify Hadoop daemons:

   ```
   hduser@pingax: jps
   ```



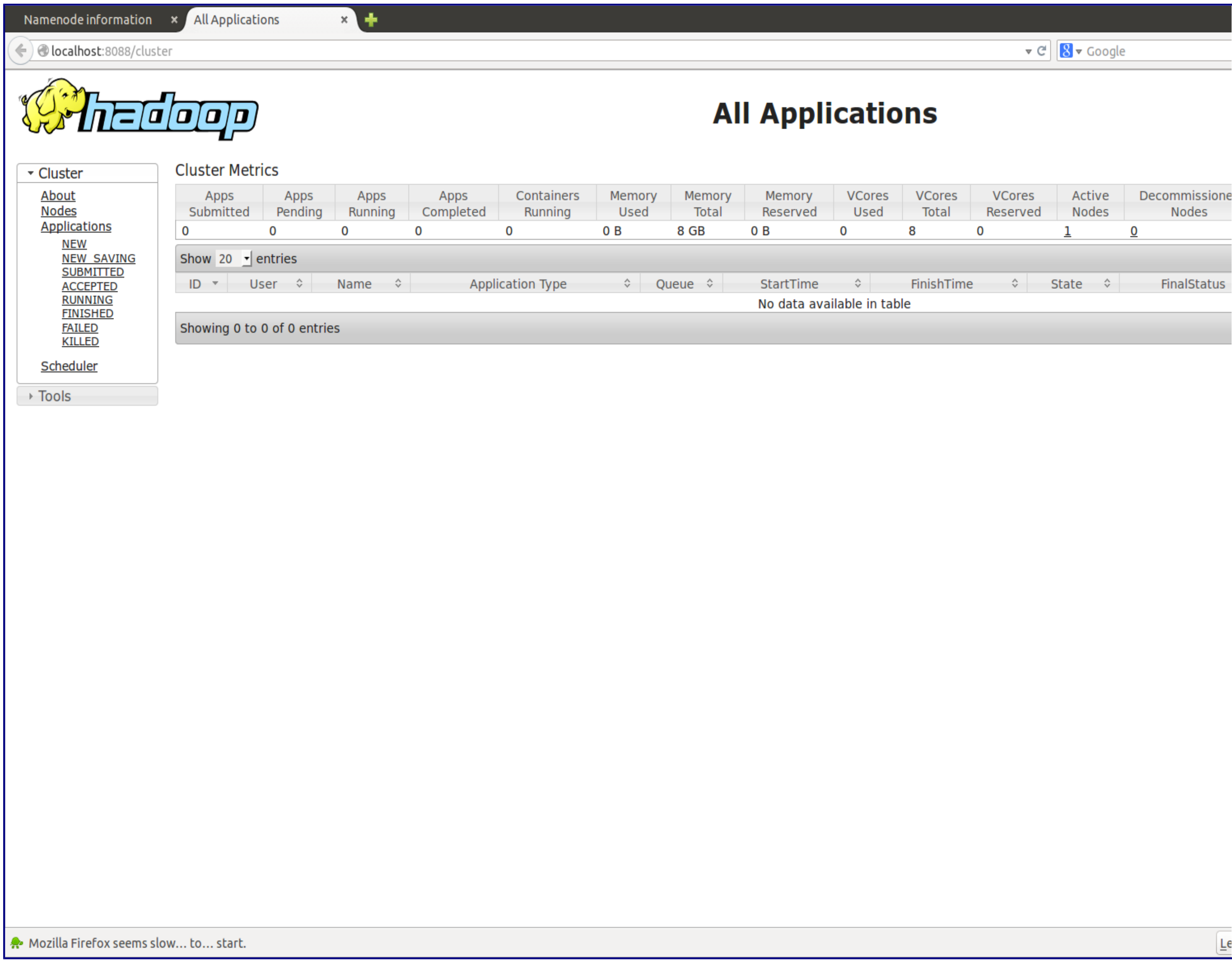
```
hduser@pingax:/home/vignesh$ start-dfs.sh
15/01/06 22:45:21 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hduser-na
menode-pingax.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hduser-da
tanode-pingax.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hd
user-secondarynamenode-pingax.out
15/01/06 22:45:44 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
hduser@pingax:/home/vignesh$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hduser-resource
manager-pingax.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hduser-n
odemanager-pingax.out
hduser@pingax:/home/vignesh$ jps
3472 Jps
3067 SecondaryNameNode
2902 DataNode
3282 ResourceManager
3410 NodeManager
2775 NameNode
hduser@pingax:/home/vignesh$ 
```

Monitor Hadoop ResourseManage and Hadoop NameNode

If you wish to track Hadoop MapReduce as well as HDFS, you can try exploring Hadoop web view of ResourceManager and NameNode which are usually used by hadoop administrators.

Open your default browser and visit to the following links.

For ResourceManager – `Http://localhost:8088`



All Applications

Cluster Metrics

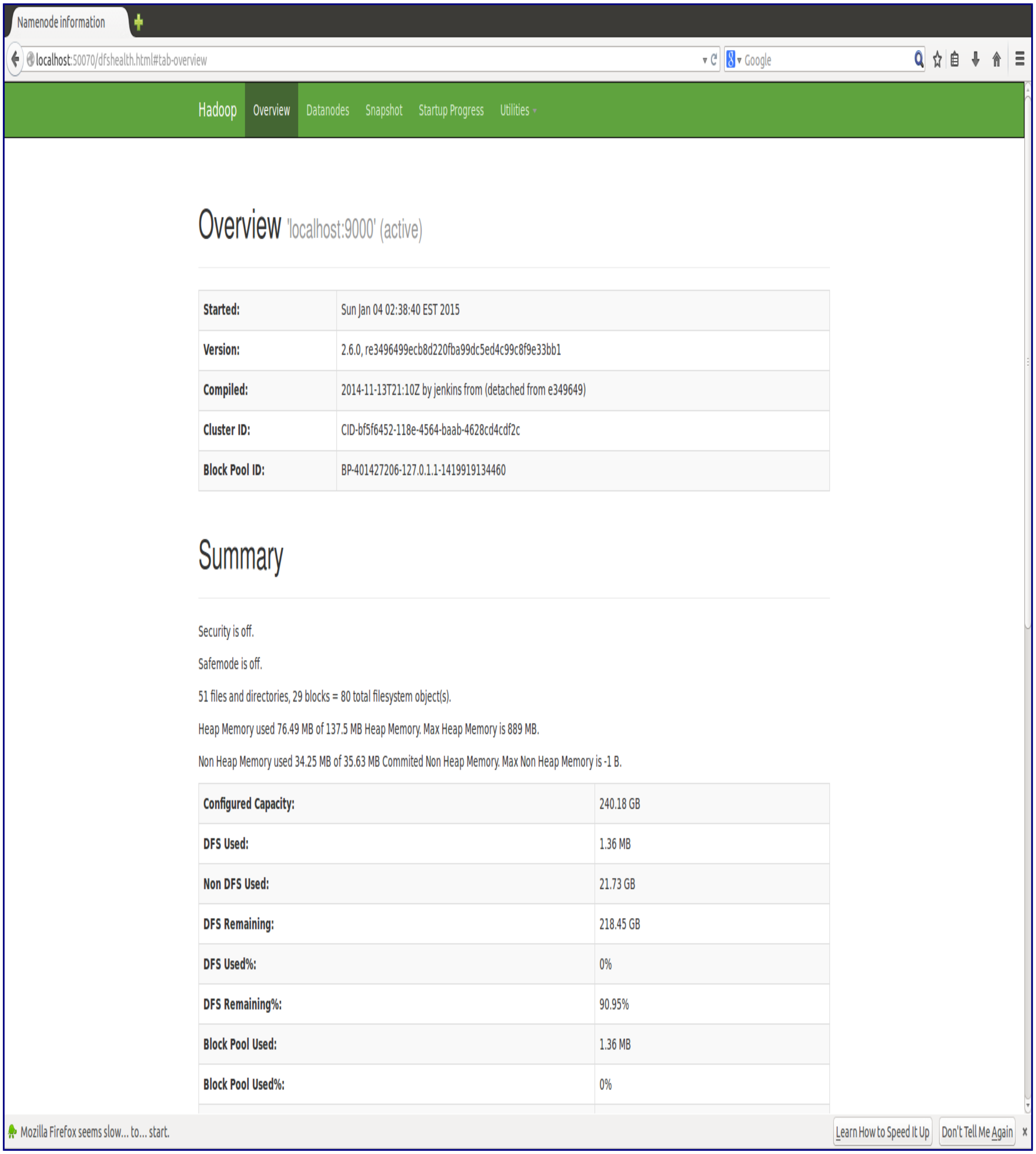
Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissione Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0

Show 20 entries

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus
No data available in table								

Showing 0 to 0 of 0 entries

For NameNode – Http://localhost:50070



Overview 'localhost:9000' (active)

Started:	Sun Jan 04 02:38:40 EST 2015
Version:	2.6.0, re3496499ecb8d220fba99dc5ed4c99c8f9e33bb1
Compiled:	2014-11-13T21:10Z by jenkins from (detached from e349649)
Cluster ID:	CID-bf5f6452-118e-4564-baab-4628cd4cdf2c
Block Pool ID:	BP-401427206-127.0.1.1-1419919134460

Summary

Security is off.

Safemode is off.

51 files and directories, 29 blocks = 80 total filesystem object(s).

Heap Memory used 76.49 MB of 137.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 34.25 MB of 35.63 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	240.18 GB
DFS Used:	1.36 MB
Non DFS Used:	21.73 GB
DFS Remaining:	218.45 GB
DFS Used%:	0%
DFS Remaining%:	90.95%
Block Pool Used:	1.36 MB
Block Pool Used%:	0%

Mozilla Firefox seems slow... to... start.

If you are getting output as shown in the above snapshot then Congratulations! You have successfully installed Apache Hadoop in your Ubuntu and if not then post your error messages in comments. We will be happy to help you. Happy Hadooping.!!

For Hadoop Multinode/Cluster setup, visit my next blog – http://pingax.com/install-apache-hadoop-ubuntu-cluster-setup/