**Lab-01 Single Node Hadoop on 14.04**

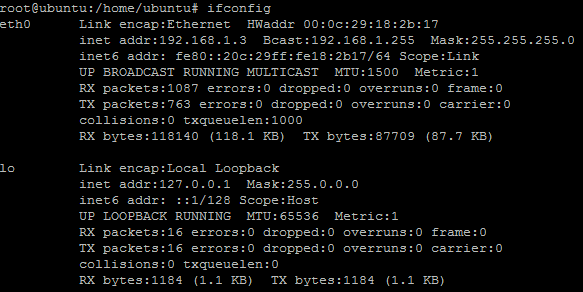
**Step 1. Preparing Ubuntu**

Change to super user mode for rest of the document

# sudo su

**Step 2. Identify what is your system IP address and netmask details:**

# ifconfig

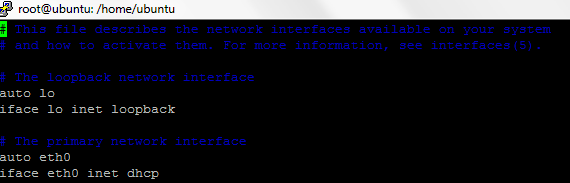


**In my scenario the IP address is 192.168.1.3 and netmask is 255.255.255.0**

**Step 3.Edit /etc/network/interfaces convert your ip address to static: on Cloud step 3not required)**

# vim /etc/network/interfaces

**Before editing the file ,the file content looks similar to below screenshot**

****

**Assing static IP and provides netmask ,gateway and dns server details as below ( as per your network)**

auto eth0

iface eth0inet static

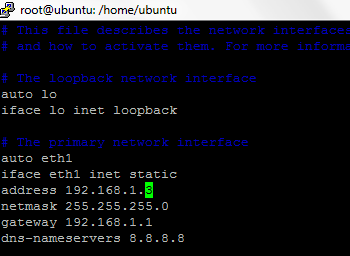
address 192.168.1.3

netmask 255.255.255.0

gateway 192.168.1.1

dns-nameservers 8.8.8.8

# After editing the file should look similar to below screenshot



**Step 3disable ipv6 and Restart the operating system**

vim /etc/sysctl.conf

# disable ipv6

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.lo.disable\_ipv6 = 1

# sysctl -p

##### **Step 4: Update your OS**

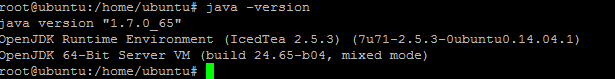
# apt-get update

**Step 5. Install Java Development Kit**

# apt-get install default-jdk

**Step 6.validate the Java version**

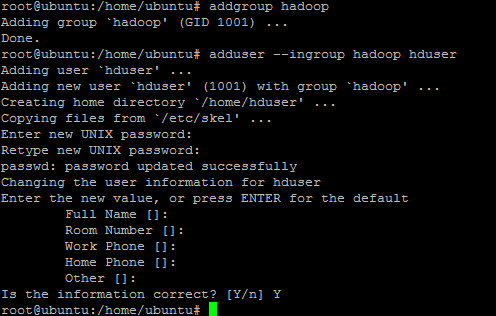
# java -version



**Step 7. Add Hadoop group and Hadoop user**

# addgroup hadoop

# adduser --ingroup hadoop hduser

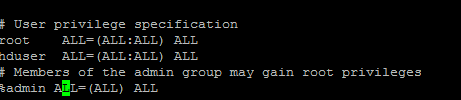


Note: Login as root user on linux machine and add hduser

# chmod 740 /etc/sudoers

# vim /etc/sudoers

# hduser ALL=(ALL:ALL) ALL



PlzNote : enter password, and hit enter when promoted for all questions for change user request

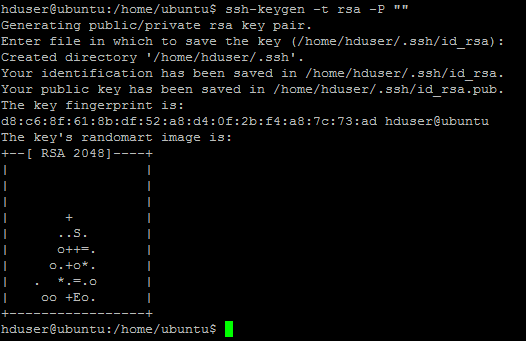
**Step 8. Install SSH and start sshd demon**

# apt-get install ssh

**Step 9.**Hadoop uses SSH (to access its nodes) which would normally require the user to enter a password. However, this requirement can be eliminated by creating and setting up SSH certificates using the following commands. If asked for a filename just leave it blank and press the enter key to continue.

# su hduser

# ssh-keygen -t rsa -P ""

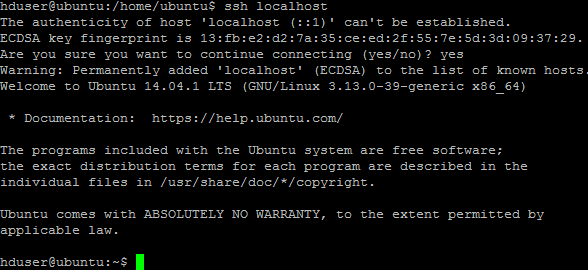


**Step 9. Allow hduser for password less ssh access**

# cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys

**Step 10. Validate if you are able to ssh**

# ssh localhost



**Step 11. Download Hadoop 2.4.1 on your Ubuntu machins**

# sudo wget http://archive.apache.org/dist/hadoop/core/hadoop-2.4.1/hadoop-2.4.1.tar.gz

# tar -xvf hadoop-2.4.1.tar.gz

**Step 12. Move the hadoop folder completely to /usr/local/hadoop**

# sudo mv hadoop-2.4.1 /usr/local/hadoop

# sudo su

# cd /usr/local

# ls -ld hadoop

# chown -R hduser:hadoophadoop

# pwd

/usr/local/

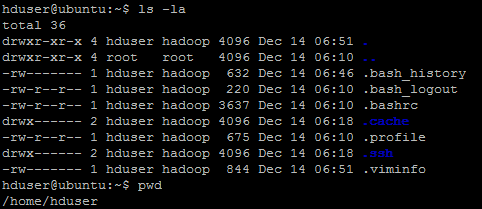
# ls -ld hadoop

**Step 13 : Let us now edit default configuration files of Hadoop ( /home/hduser/.bashrc)**

# cd /home/hduser

# ls-la

# vim .bashrc



# After the very last lines of the .bashrc file enter below details and save and exit

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export HADOOP\_INSTALL=/usr/local/hadoop

export PATH=$PATH:$HADOOP\_INSTALL/bin

export PATH=$PATH:$HADOOP\_INSTALL/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_HOME=$HADOOP\_INSTALL

export HADOOP\_HDFS\_HOME=$HADOOP\_INSTALL

export YARN\_HOME=$HADOOP\_INSTALL

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_INSTALL/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_INSTALL/lib"

**Step 14 : edit the /usr/local/hadoop/etc/hadoop/hadoop-env.sh and add java\_home directory details.Thiswil help java to be available for Hadoop during its startup**

**Heapsize should be mentioned in MB ( for x 500 MB means ,the name node will use 500 MB max memory during startup)**

# vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

**Stpe15 : ensure hduser has permission onf /app/hadoop/tmp directory**

**Excute this by logging in as root user fromhduser**

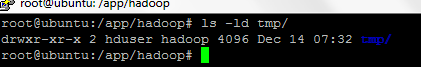
**$ exit**

**#** sudo mkdir -p /app/hadoop/tmp

#sudochownhduser:hadoop /app/hadoop/tmp

# cd /app/hadoop

# ls -ldtmp



Validate if hduser owns this folder /tmp or not

**Step 16 : /usr/local/hadoop/etc/hadoop/core-site.xml**:

The **/usr/local/hadoop/etc/hadoop/core-site.xml** file contains configuration properties that Hadoop uses when starting up. This file can be used to override the default settings that Hadoop starts with.

# vim /usr/local/hadoop/etc/hadoop/core-site.xml

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/app/hadoop/tmp</value>

<description>A base for other temporary directories.</description>

</property>

<property>

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

<value>hdfs://localhost:54310</value>

<description>The name of the default file system. A URI whose

scheme and authority determine the FileSystem implementation. The

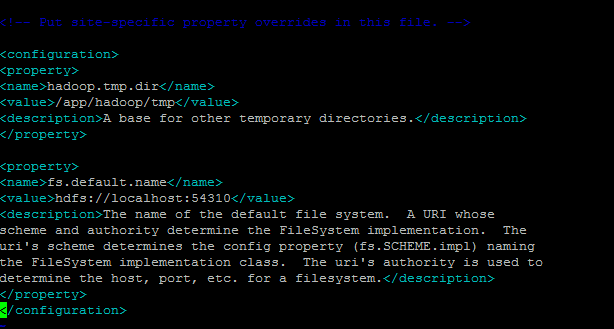
uri's scheme determines the config property (fs.SCHEME.impl) naming

the FileSystem implementation class. The uri's authority is used to

determine the host, port, etc. for a filesystem.</description>

</property>

</configuration>



Explaination : <name>fs.default.name</name>.

The value your specify where Primary name node will run. For example multi node setup repalce local host with IP address of your master server or hostnem of master server

<value>hdfs://localhost:54310</value>

**Step 16. /usr/local/hadoop/etc/hadoop/mapred-site.xml**

By default, the **/usr/local/hadoop/etc/hadoop/** folder contains the **/usr/local/hadoop/etc/hadoop/mapred-site.xml.template** file which has to be renamed/copied with the name **mapred-site.xml**:

$ cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/etc/hadoop/mapred-site.xml

$ vim**/usr/local/hadoop/etc/hadoop/mapred-site.xml**

<configuration>

<property>

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

<value>localhost:54311</value>

<description>The host and port that the MapReduce job tracker runs

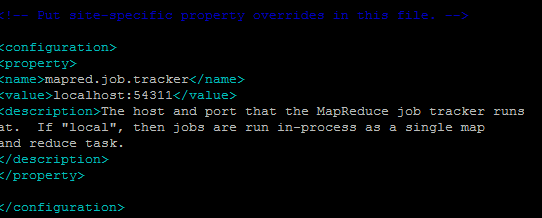
at. If "local", then jobs are run in-process as a single map

and reduce task.

</description>

</property>

</configuration>



**Explaination :mapred.job.trackervalue here is on which node job tracker will run**

Step17 : we need to create new directories namenode and datanode directories

The **/usr/local/hadoop/etc/hadoop/hdfs-site.xml** file needs to be configured for each host in the cluster that is being used. It is used to specify the directories which will be used as the **namenode** and the **datanode** on that host

Perform all operatiosn below as root user

$ sudo root

# sudo mkdir -p /usr/local/hadoop\_store/hdfs/namenode

# sudo mkdir -p /usr/local/hadoop\_store/hdfs/datanode

# sudo chown -R hduser:hadoop /usr/local/hadoop\_store

# su hduser

Step18 :Edit hdfs-site.xml file

$ vim **/usr/local/hadoop/etc/hadoop/hdfs-site.xml**

<configuration>

<property>

<name>dfs.replication</name>

<value>1</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>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop\_store/hdfs/namenode</value>

</property>

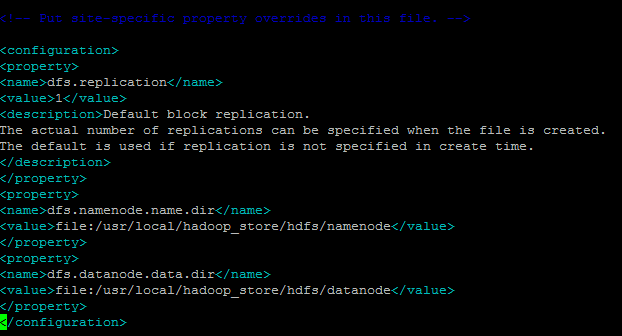
<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop\_store/hdfs/datanode</value>

</property>

</configuration>

****

Explainationdfs.replciation value is number of replicas of blocks

Df.namenode.name.dir value is where your metadata info is saved

Dfs.namenode.data.dir value states where my data is going to be stored

Format the New HadoopFilesystem

Now, the Hadoopfilesystem needs to be formatted so that we can start to use it. The format command should be issued with write permission since it creates **current** directory under **/usr/local/hadoop\_store/hdfs/namenode** folder:

hduser@k:~$ hadoop namenode -format

DEPRECATED: Use of this script to execute hdfs command is deprecated.

Instead use the hdfs command for it.

Note that **hadoopnamenode -format** command should be executed once before we start using Hadoop. If this command is executed again after Hadoop has been used, it'll destroy all the data on the Hadoop file system.

tartingHadoop

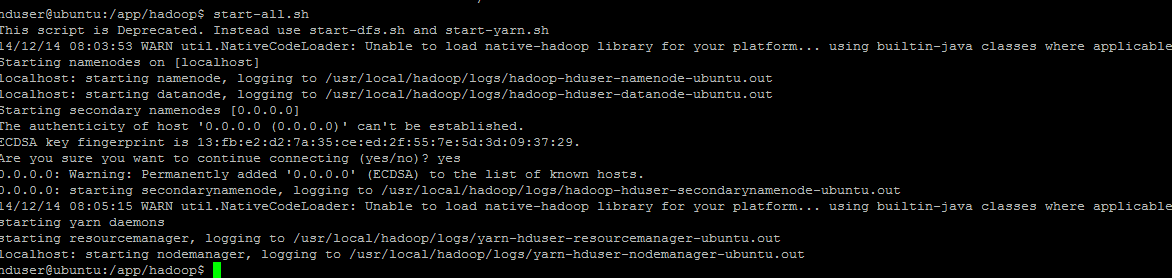
Now it's time to start the newly installed single node cluster. We can use

$ **start-all.sh** or

$ **start-dfs.sh**

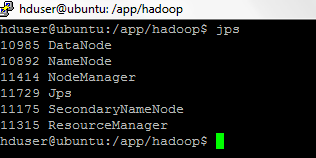
**$ start-yarn.sh**

The or put will look like this .,when promoted for response type yes and hit enter



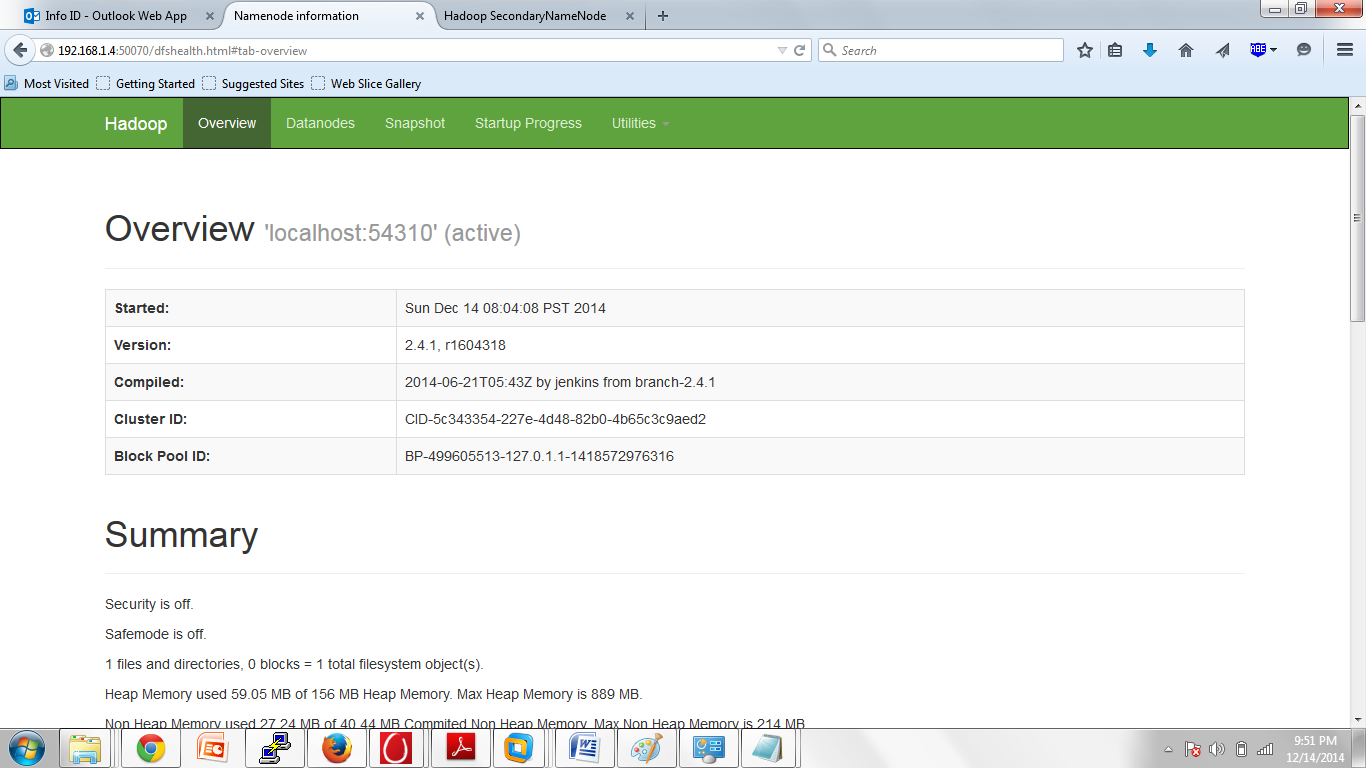
Step 19 : validate thehadoop demon services

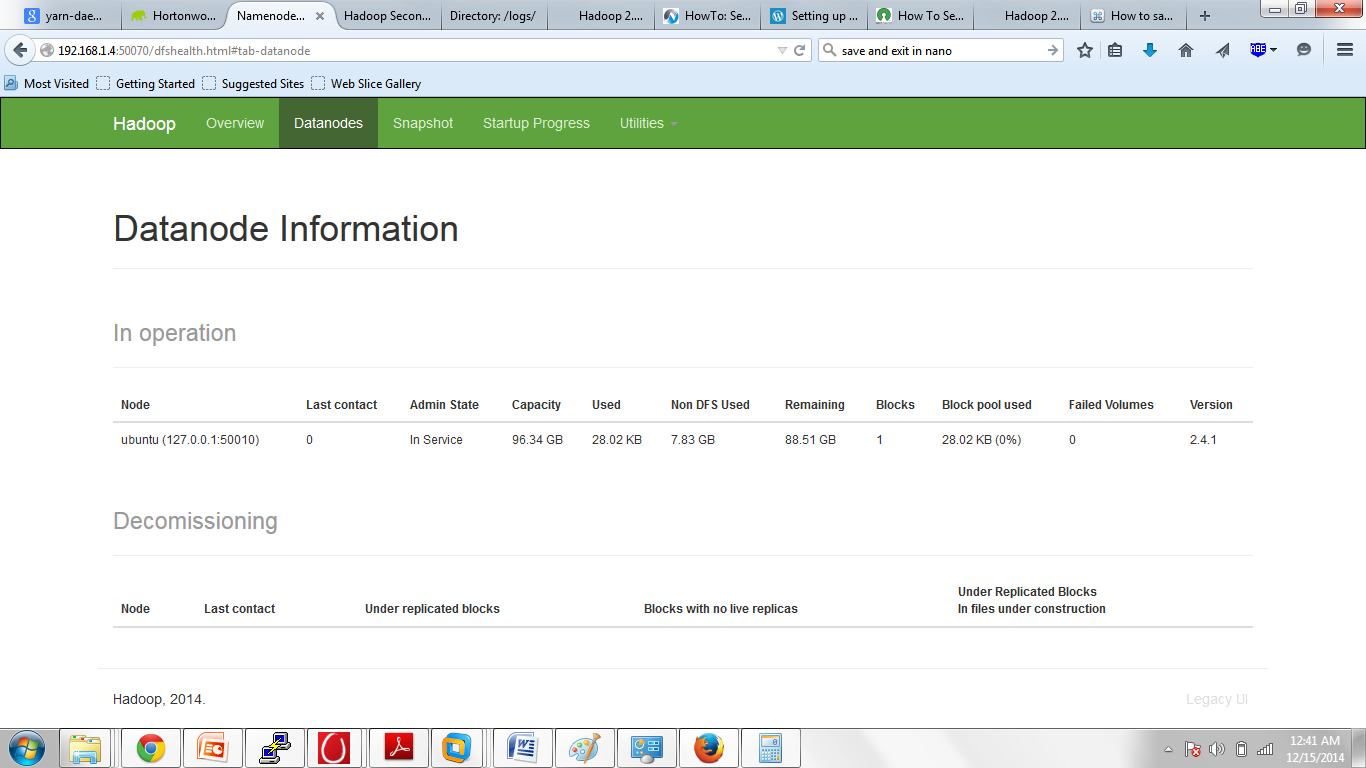
$ jps



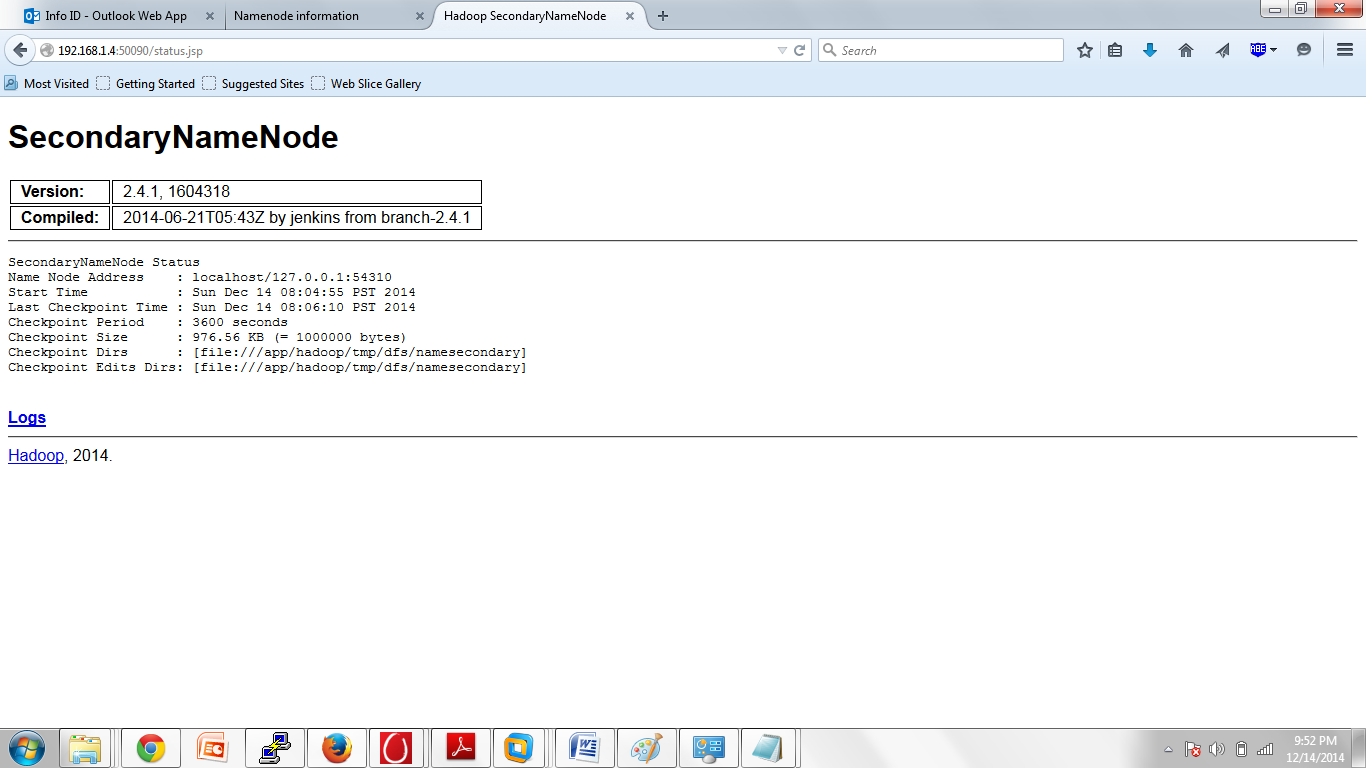
**Test on browser**

[**http://192.168.1.4:50070/**](http://192.168.1.4:50070/)

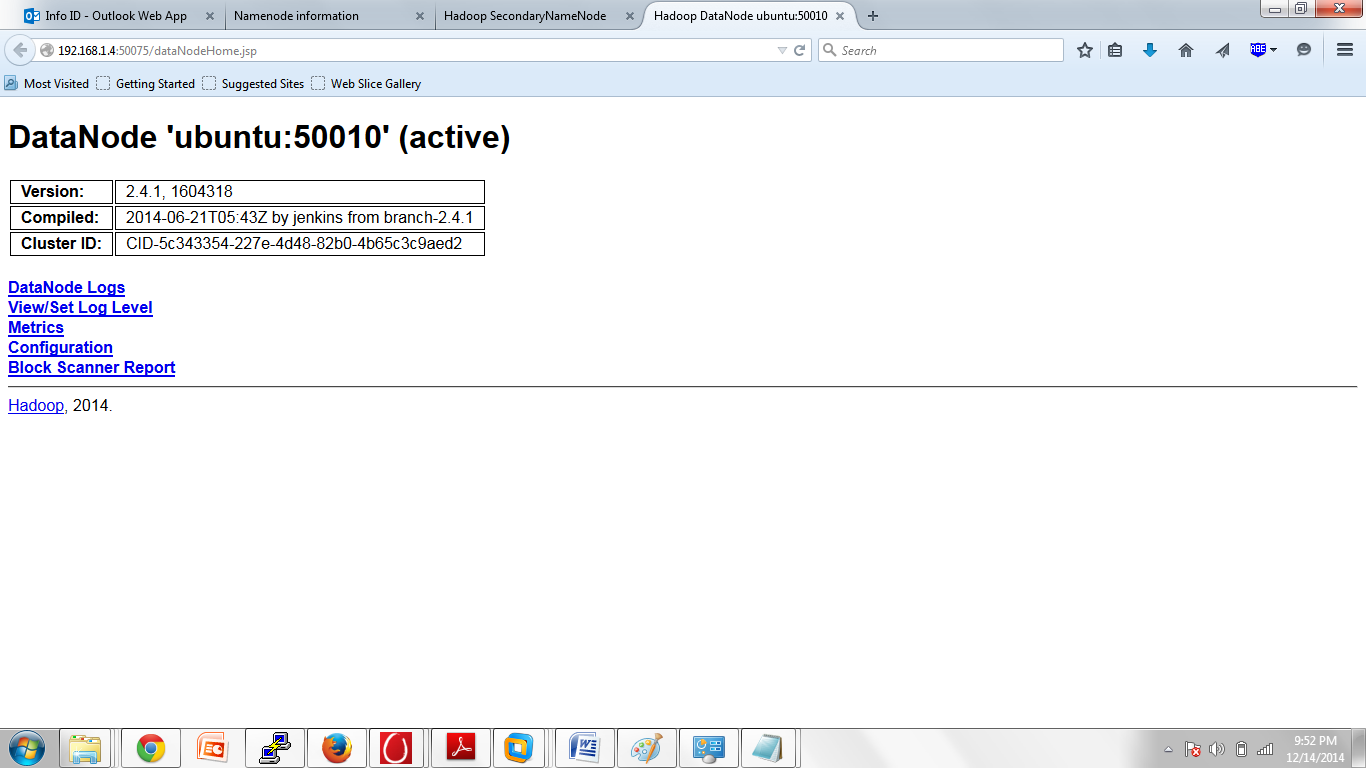
****

****

[**http://192.168.1.4:50090/**](http://192.168.1.4:50090/)

****

[**http://192.168.1.4:50075/**](http://192.168.1.4:50075/)

****