Single-node Hadoop Instalation:

<http://doctuts.readthedocs.io/en/latest/hadoop.html#installation>

**Creado**: 6 Diciembre

**Última modificación**: 6 Diciembre

## **SINGLE-NODE INSTALLATION**

### **Running Hadoop on Ubuntu (Single node cluster setup)**

The report here will describe the required steps for setting up a single-node Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux. Hadoop is a framework written in Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the Google File System (GFS) and of the MapReduce computing paradigm. Hadoop’s HDFS is a highly fault-tolerant distributed file system and, like Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications that have large data sets.

Before we start, we will understand the meaning of the following:

#### **DataNode:**

A DataNode stores data in the Hadoop File System. A functional file system has more than one DataNode, with the data replicated across them.

#### **NameNode:**

The NameNode is the centrepiece of an HDFS file system. It keeps the directory of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these file itself.

#### **Jobtracker:**

The Jobtracker is the service within hadoop that farms out MapReduce to specific nodes in the cluster, ideally the nodes that have the data, or atleast are in the same rack.

#### **TaskTracker:**

A TaskTracker is a node in the cluster that accepts tasks- Map, Reduce and Shuffle operatons – from a Job Tracker.

#### **Secondary Namenode:**

Secondary Namenode whole purpose is to have a checkpoint in HDFS. It is just a helper node for namenode.

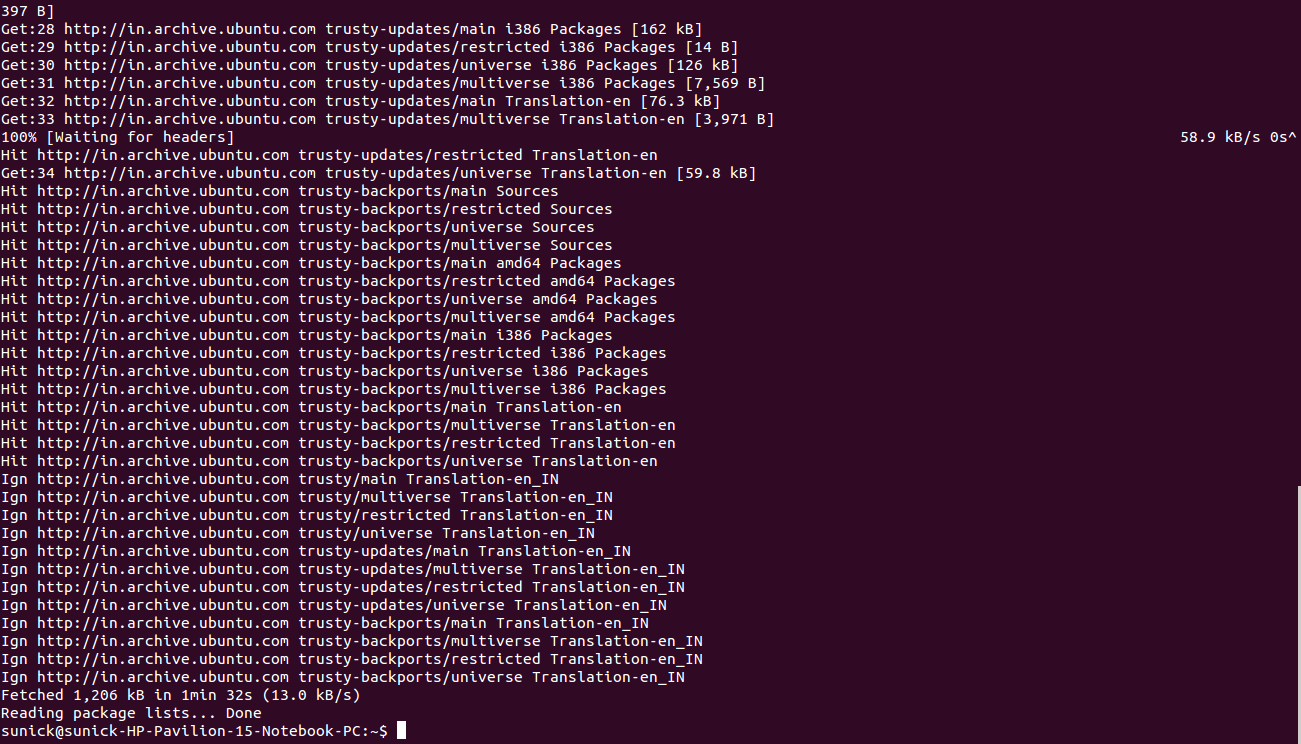
### **Prerequisites**

#### **Java 6 JDK**

Hadoop requires a working Java 1.5+ (aka Java 5) installation.

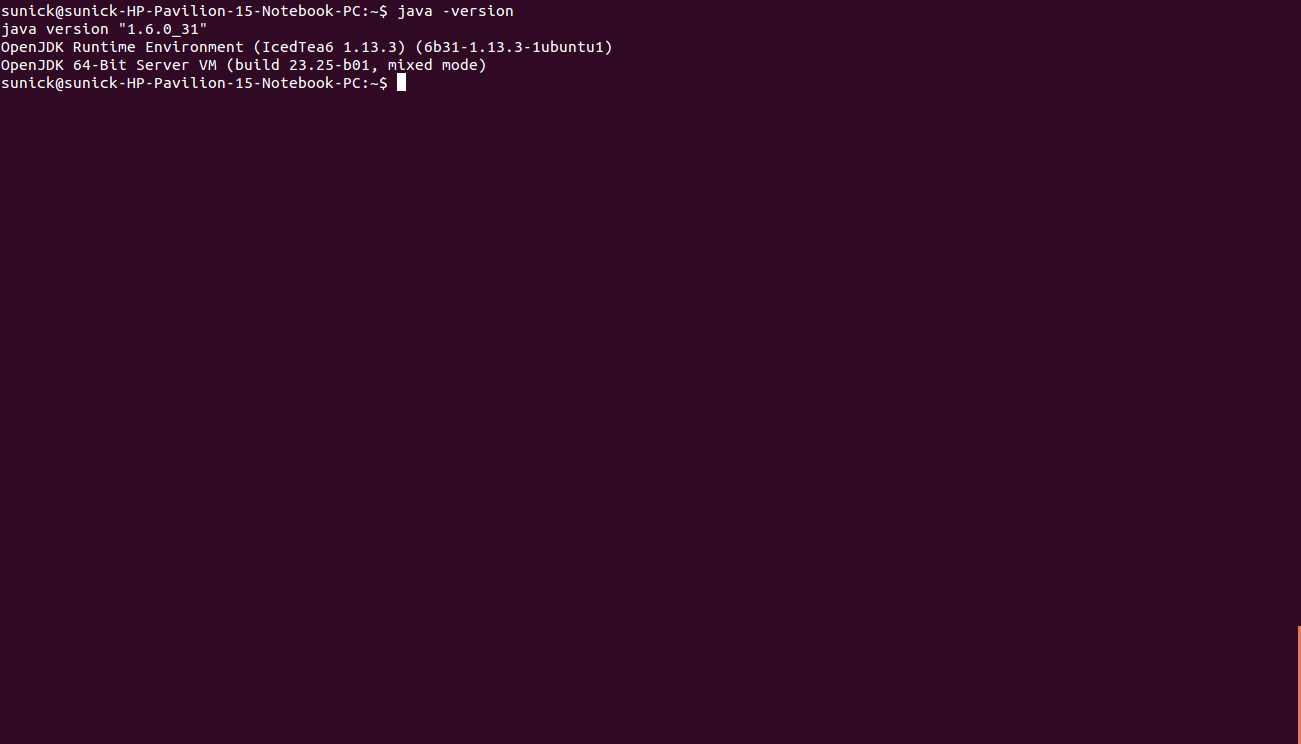
Update the source list

user@ubuntu:~$ sudo apt-get update



or

Install Sun Java 6 JDK



##### **Note:**

If you already have Java JDK installed on your system, then you need not run the above command.

To install it

user@ubuntu:~$ sudo apt-get install sun-java6-jdk

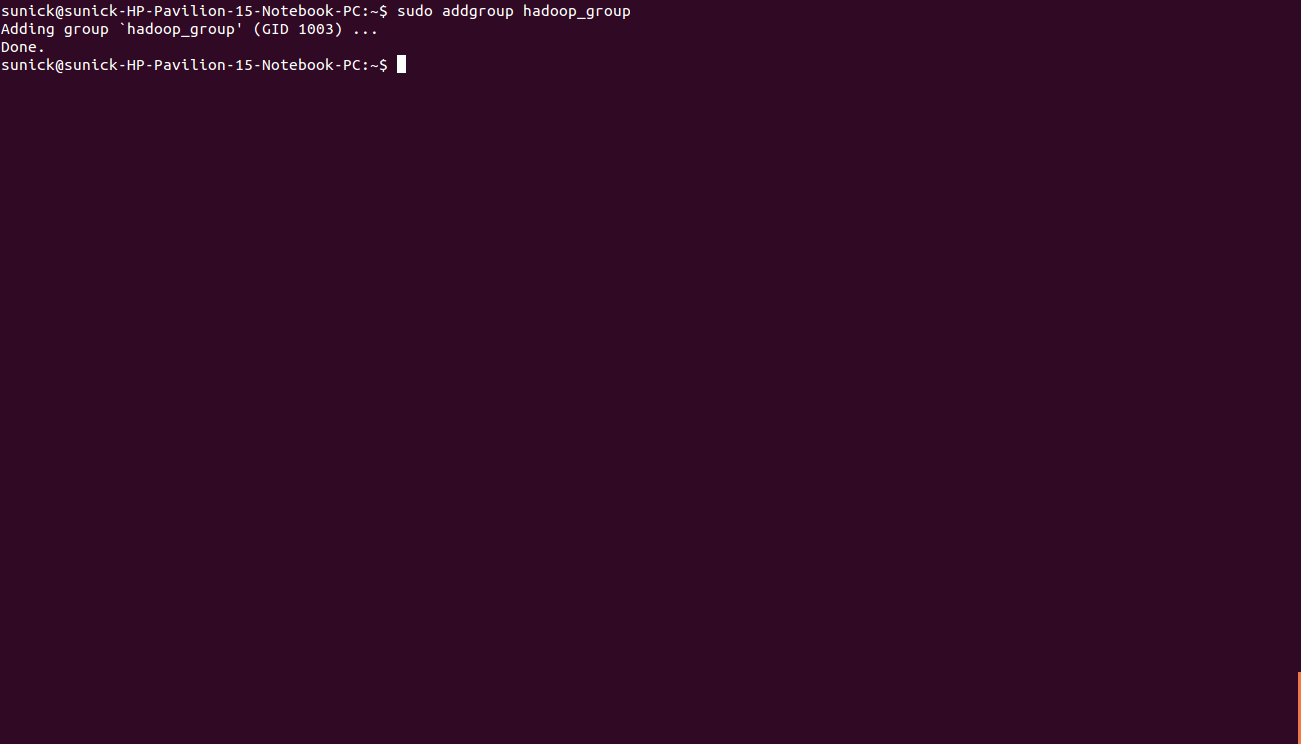
The full JDK which will be placed in /usr/lib/jvm/java-6-openjdk-amd64 After installation, check whether java JDK is correctly installed or not, with the following command

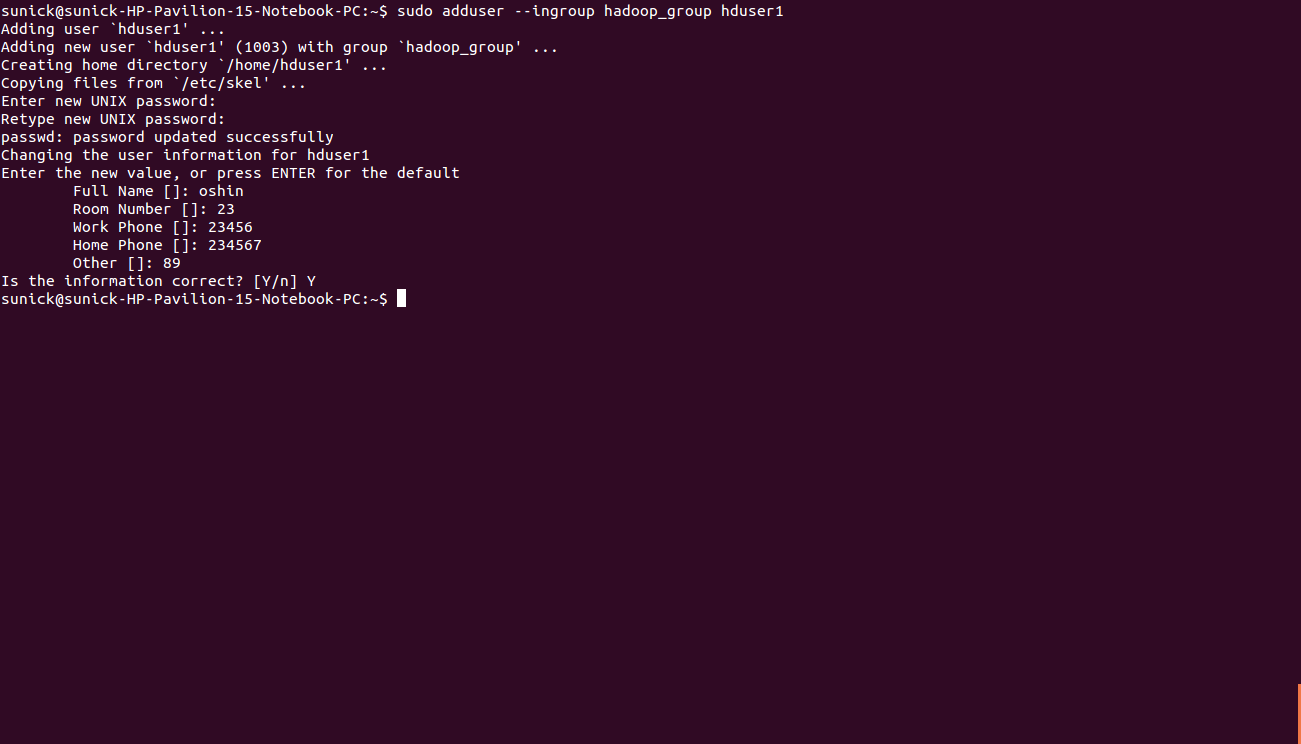
user@ubuntu:~$ java -version

#### **Adding a dedicated Hadoop system user**

We will use a dedicated Hadoop user account for running Hadoop.

user@ubuntu:~$ sudo addgroup hadoop\_group  
user@ubuntu:~$ sudo adduser --ingroup hadoop\_group hduser1





This will add the user hduser1 and the group hadoop\_group to the local machine. Add hduser1 to the sudo group

user@ubuntu:~$ sudo adduser hduser1 sudo

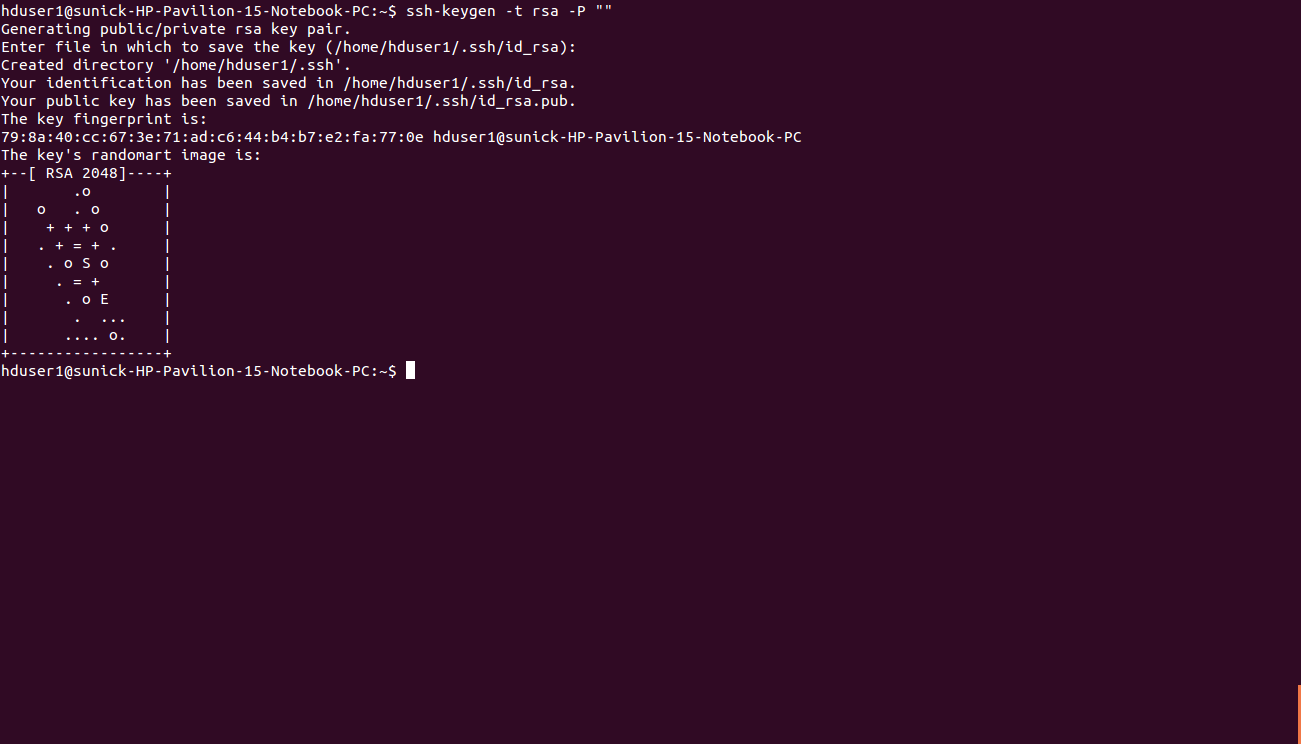
#### **Configuring SSH**

The hadoop control scripts rely on SSH to peform cluster-wide operations. For example, there is a script for stopping and starting all the daemons in the clusters. To work seamlessly, SSH needs to be setup to allow password-less login for the hadoop user from machines in the cluster. The simplest way to achive this is to generate a public/private key pair, and it will be shared across the cluster.

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine. For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost for the hduser user we created in the earlier.

We have to generate an SSH key for the hduser user.

user@ubuntu:~$ su – hduser1  
hduser1@ubuntu:~$ ssh-keygen -t rsa -P ""



The second line will create an RSA key pair with an empty password.

##### **Note:**

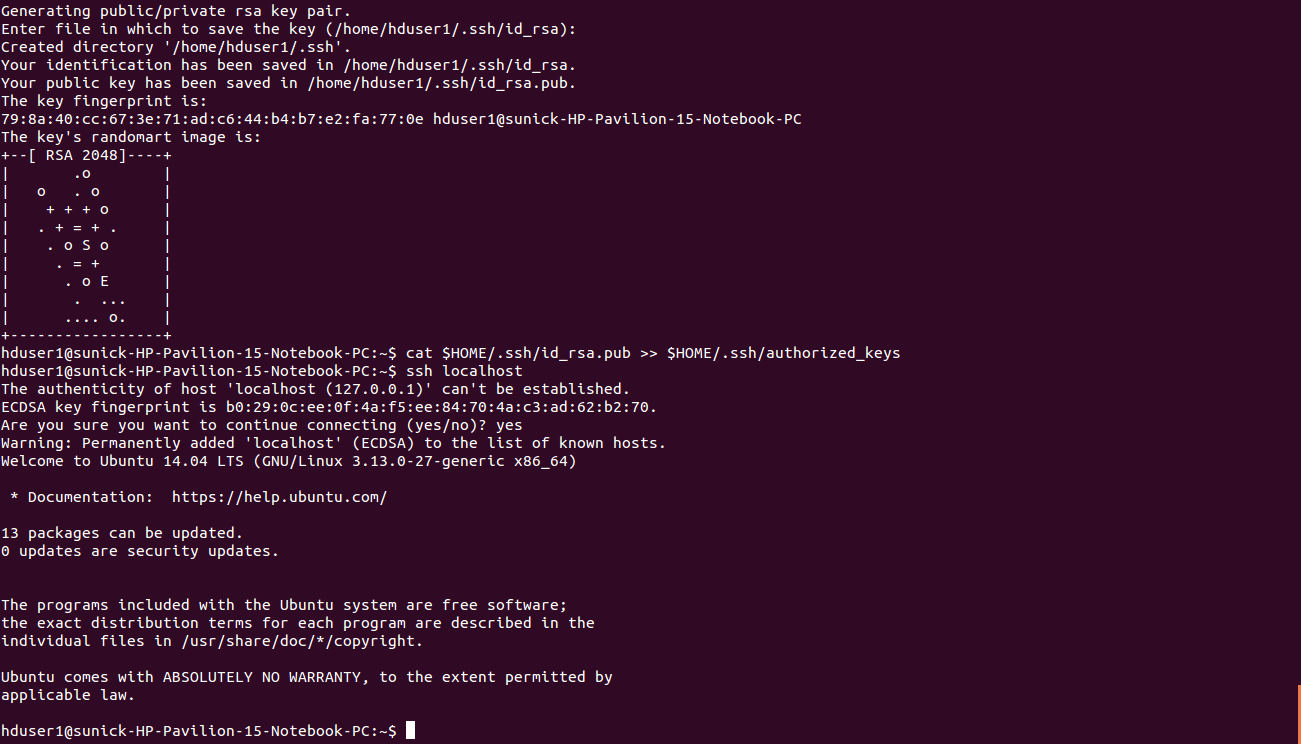
P “”, here indicates an empty password

You have to enable SSH access to your local machine with this newly created key which is done by the following command.

hduser1@ubuntu:~$ cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys

The final step is to test the SSH setup by connecting to the local machine with the hduser1 user. The step is also needed to save your local machine’s host key fingerprint to the hduser user’s known hosts file.

hduser@ubuntu:~$ ssh localhost



If the SSH connection fails, we can try the following (optional):

* Enable debugging with ssh -vvv localhost and investigate the error in detail.
* Check the SSH server configuration in /etc/ssh/sshd\_config. If you made any changes to the SSH server configuration file, you can force a configuration reload with sudo /etc/init.d/ssh reload.

### **INSTALLATION**

#### **Main Installation**

* Now, I will start by switching to hduser
* hduser@ubuntu:~$ su - hduser1
* Now, download and extract Hadoop 1.2.0
* Setup Environment Variables for Hadoop

Add the following entries to .bashrc file

# Set Hadoop-related environment variables  
export HADOOP\_HOME=/usr/local/hadoop  
# Add Hadoop bin/ directory to PATH  
export PATH= $PATH:$HADOOP\_HOME/bin

#### **Configuration**

##### **hadoop-env.sh**

Change the file: conf/hadoop-env.sh

#export JAVA\_HOME=/usr/lib/j2sdk1.5-sun

to in the same file

# export JAVA\_HOME=/usr/lib/jvm/java-6-openjdk-amd64 (for 64 bit)  
# export JAVA\_HOME=/usr/lib/jvm/java-6-openjdk-amd64 (for 32 bit)

##### **conf/**[**\***](http://doctuts.readthedocs.io/en/latest/hadoop.html#id2)**-site.xml**

Now we create the directory and set the required ownerships and permissions

hduser@ubuntu:~$ sudo mkdir -p /app/hadoop/tmp  
hduser@ubuntu:~$ sudo chown hduser:hadoop /app/hadoop/tmp  
hduser@ubuntu:~$ sudo chmod 750 /app/hadoop/tmp

The last line gives reading and writing permissions to the /app/hadoop/tmp directory

* Error: If you forget to set the required ownerships and permissions, you will see a java.io.IO Exception when you try to format the name node.

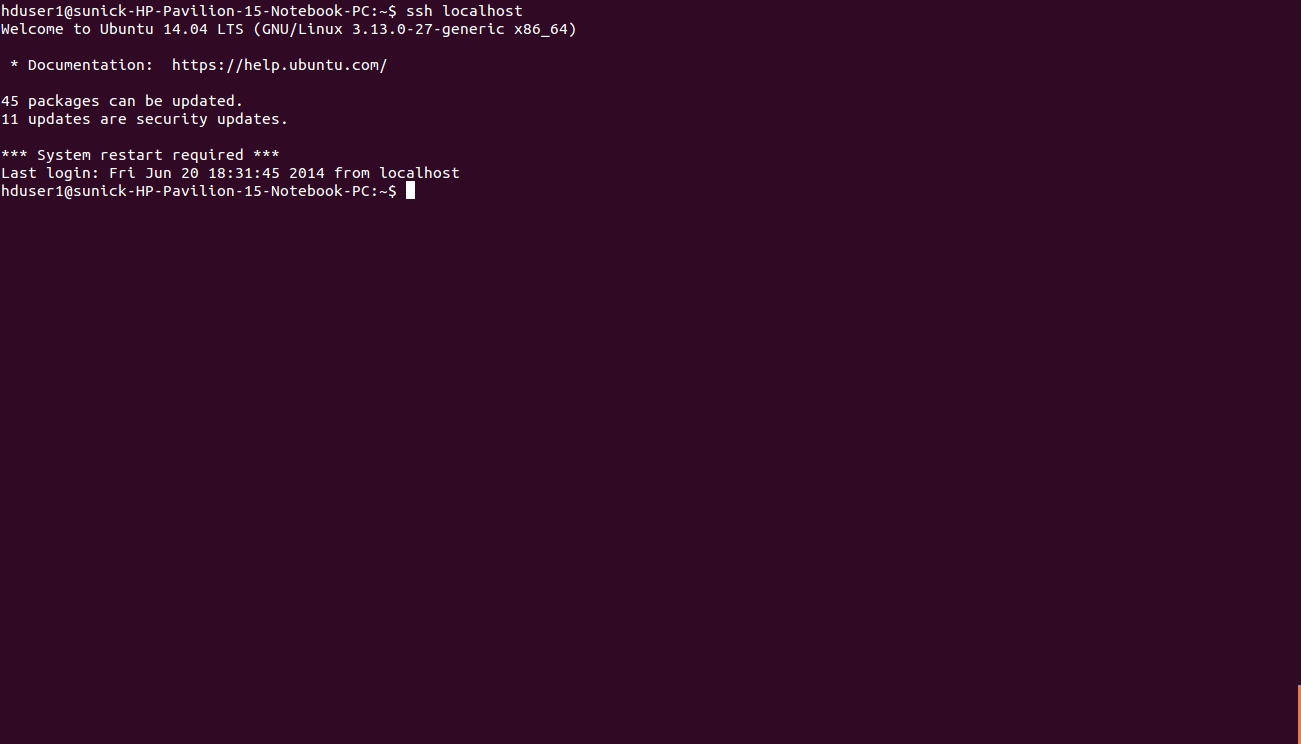
Paste the following between <configuration>

* In file conf/core-site.xml
* <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>
* In file conf/mapred-site.xml
* <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>
* In file conf/hdfs-site.xml
* <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>

#### **Formatting the HDFS filesystem via the NameNode**

To format the filesystem (which simply initializes the directory specified by the dfs.name.dir variable). Run the command

hduser@ubuntu:~$ /usr/local/hadoop/bin/hadoop namenode –format



#### **Starting your single-node cluster**

Before starting the cluster, we need to give the required permissions to the directory with the following command

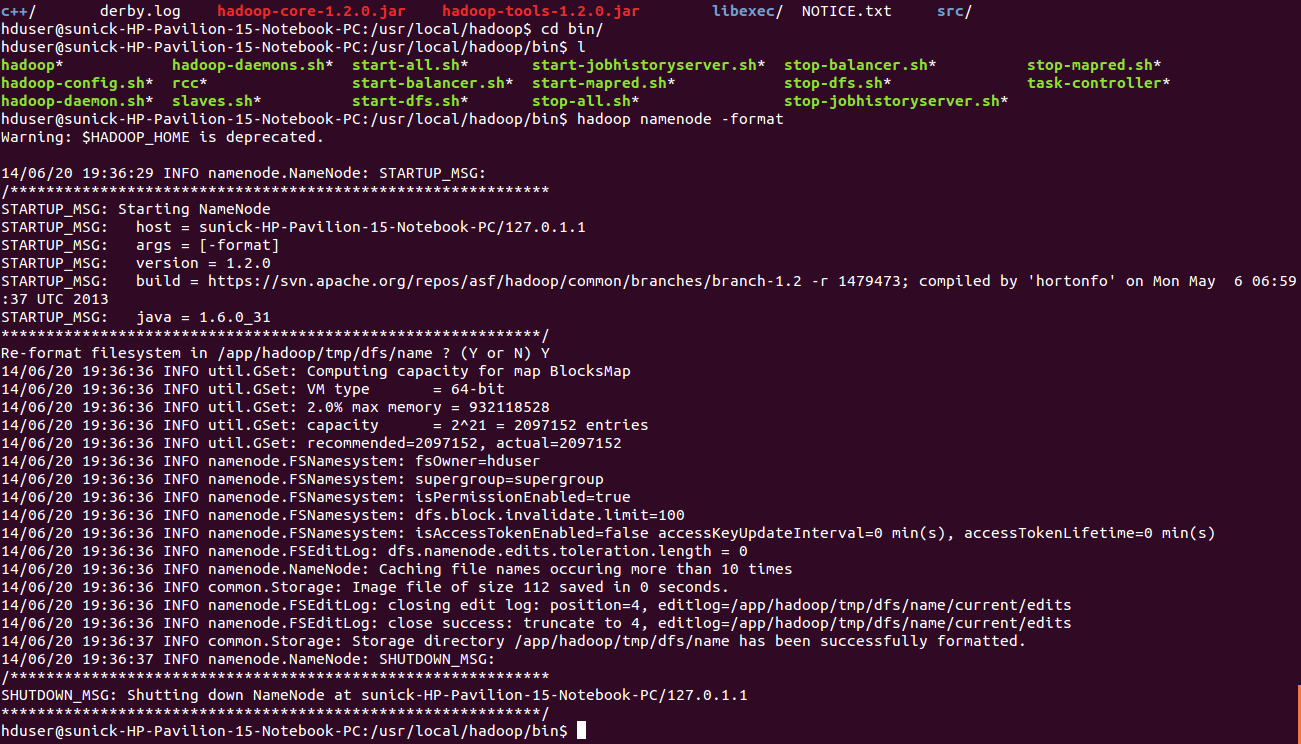
hduser@ubuntu:~$ sudo chmod -R 777 /usr/local/hadoop

Run the command

hduser@ubuntu:~$ /usr/local/hadoop/bin/start-all.sh

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on the machine.

hduser@ubuntu:/usr/local/hadoop$ jps



##### **Errors:**

1. If by chance your datanode is not starting, then you have to erase the contents of the folder /app/hadoop/tmp
2. The command that can be used
3. hduser@ubuntu:~:$ sudo rm –Rf /app/hadoop/tmp/\*
4. You can also check with netstat if Hadoop is listening on the configured ports.
5. The command that can be used
6. hduser@ubuntu:~$ sudo netstat -plten | grep java
7. Errors if any, examine the log files in the /logs/ directory.

#### **Stopping your single-node cluster**

Run the command to stop all the daemons running on your machine.

hduser@ubuntu:~$ /usr/local/hadoop/bin/stop-all.sh

##### **ERROR POINTS:**

If datanode is not starting, then clear the tmp folder before formatting the namenode using the following command

hduser@ubuntu:~$ rm -Rf /app/hadoop/tmp/\*

##### **Note:**

* The masters and slaves file should contain localhost.
* In /etc/hosts, the ip of the system should be given with the alias as localhost.
* Set the java home path in hadoop-env.sh as well bashrc.