

# Apache Hadoop 2.6.0 Installation and Single Node Cluster Configuration on Ubuntu

A guide to install and setup Single-Node Apache Hadoop 2.x

## Table of Contents

Topic	Page
Software Requirements	
Hardware Requirements.	
Introduction	
1. Setting up the Ubuntu Desktop	
1.1 Creating an Ubuntu VM Player instance	
1.1.1 Download the VMware image	
1.1.2 Open the image file	
1.1.3 Play the Virtual Machine	
1.1.4 Update the OS packages and their dependencies	
1.1.5 Install the Java and openssh server for Hadoop 2.6.0	
1.2 Download the Apache Hadoop 2.6.0 binaries	
1.2.1 Download the Hadoop package	
2. Configure the Apache Hadoop 2.6.0 Single Node Server	
2.1 Update the Configuration files	
2.1.1 Update ".bashrc" file for user 'ubuntu'.	

- |   |  |
|---|--|
| <ul style="list-style-type: none"><li>2.2 Setup the Hadoop Cluster<ul style="list-style-type: none"><li>2.2.1 Configure JAVA_HOME</li><li>2.2.2 Create NameNode and DataNode directory</li><li>2.2.3 Configure the Default File system</li><li>2.2.4 Configure the HDFS</li><li>2.2.5 Configure YARN framework</li><li>2.2.6 Configure MapReduce framework</li><li>2.2.7 Edit /etc/hosts file</li><li>2.2.9 Creating ssh</li><li>2.2.10 Moving the key to authorized key:</li><li>2.2.11 Start the DFS services</li><li>2.2.12 Perform the Health Check</li></ul></li></ul> |  |
|---|--|



## **Introduction**

This setup and configuration document is a guide to setup a Single-Node Apache Hadoop 2.0 cluster on an Ubuntu virtual machine on your PC. If you are new to both Ubuntu and Hadoop, this guide comes handy to quickly setup a Single-Node Apache Hadoop 2.0 Cluster on Ubuntu and start your Big Data and Hadoop learning journey.

**The guide describes the whole process in two parts:**

### **Section 1: Setting up the Ubuntu OS for Hadoop 2.0**

This section describes step by step guide to download, configure an Ubuntu Virtual Machine image in VMPlayer, and provides steps to install pre-requisites for Hadoop Installation on Ubuntu.

### **Section 2: Installing Apache Hadoop 2.0 and Setting up the Single Node Cluster**

This section explains primary Hadoop 2.0 configuration files, Single-Node cluster configuration and Hadoop daemons start and stop process in detail.

#### **1. Setting up the Ubuntu Desktop**

This section describes the steps to download and create an Ubuntu image on VMware Player.

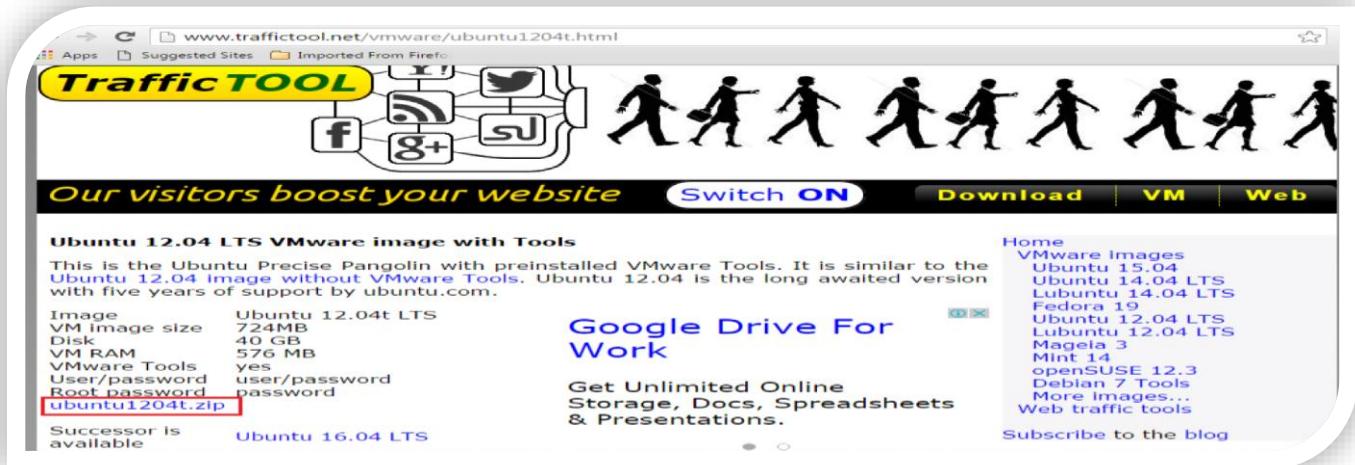
##### **1.1 Creating an Ubuntu VM Player instance**

The first step is to download an Ubuntu image and create an Ubuntu VMPlayer instance.

###### **1.1.1 Download the VMware image**

Access the following link and download the 12.0.4 Ubuntu image:

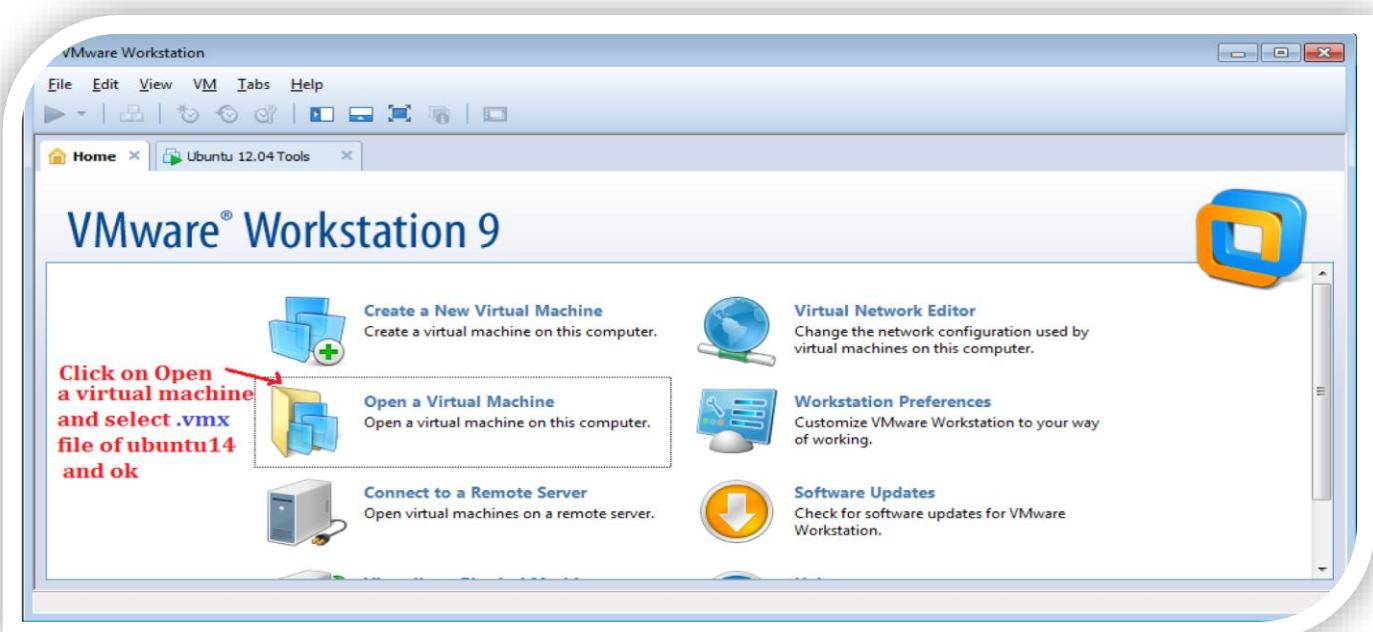
<http://www.traffictool.net/vmware/ubuntu1204t.html>

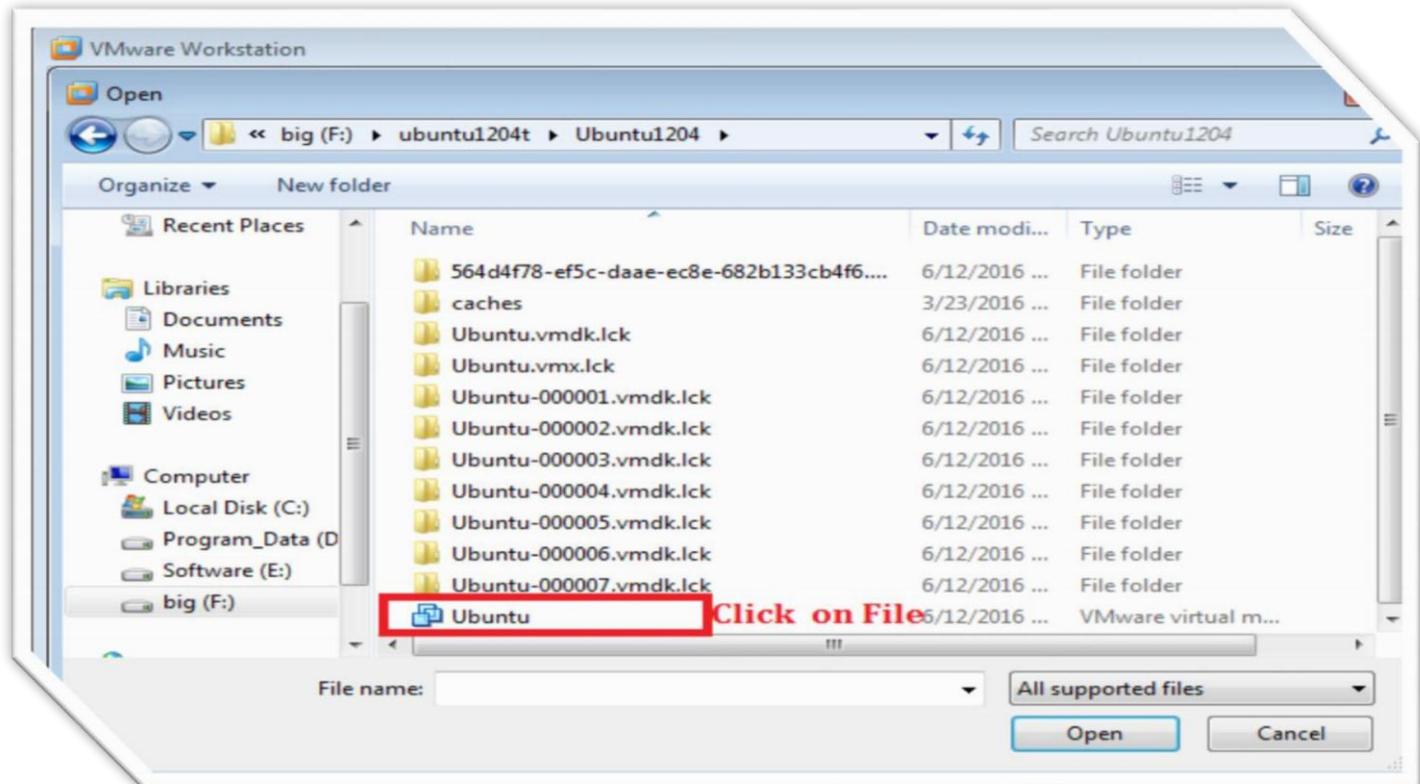


### 1.1.2 Open the image file

Extract the Ubuntu VM image and Open it in VMware Workstation.

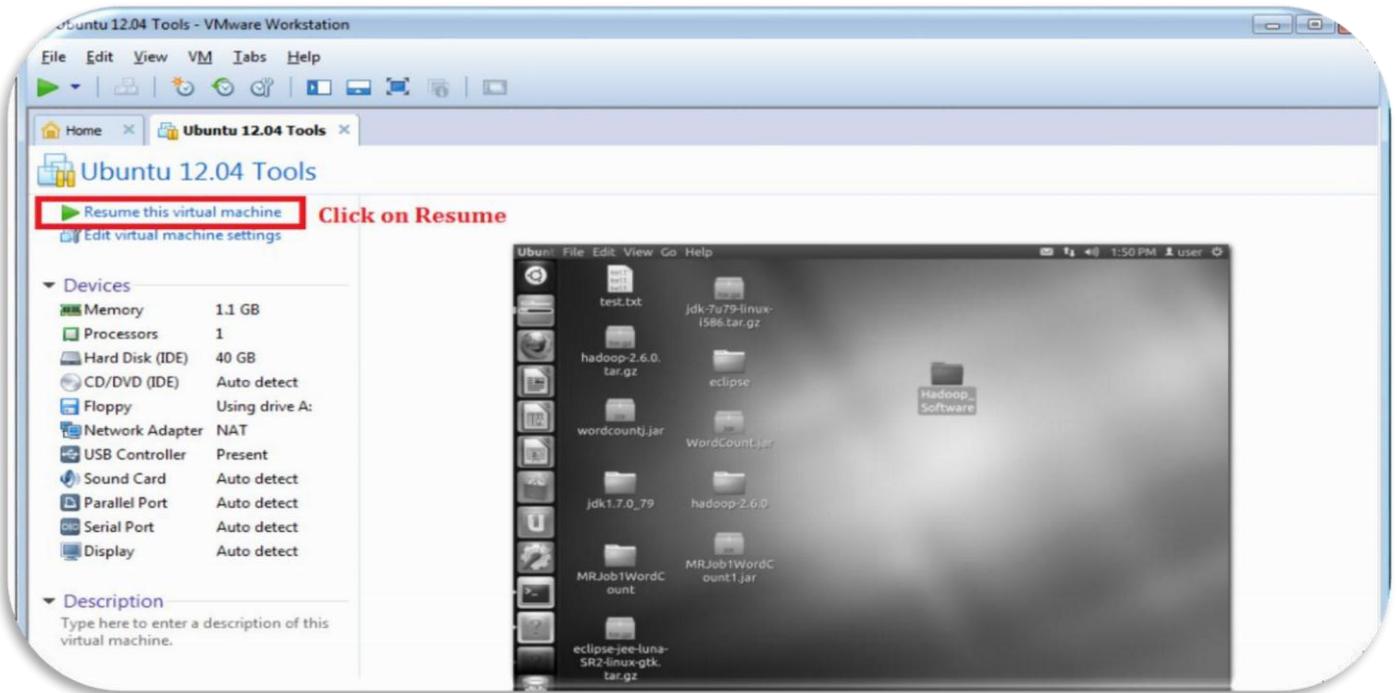
Click open virtual machine and select path where you have extracted the image





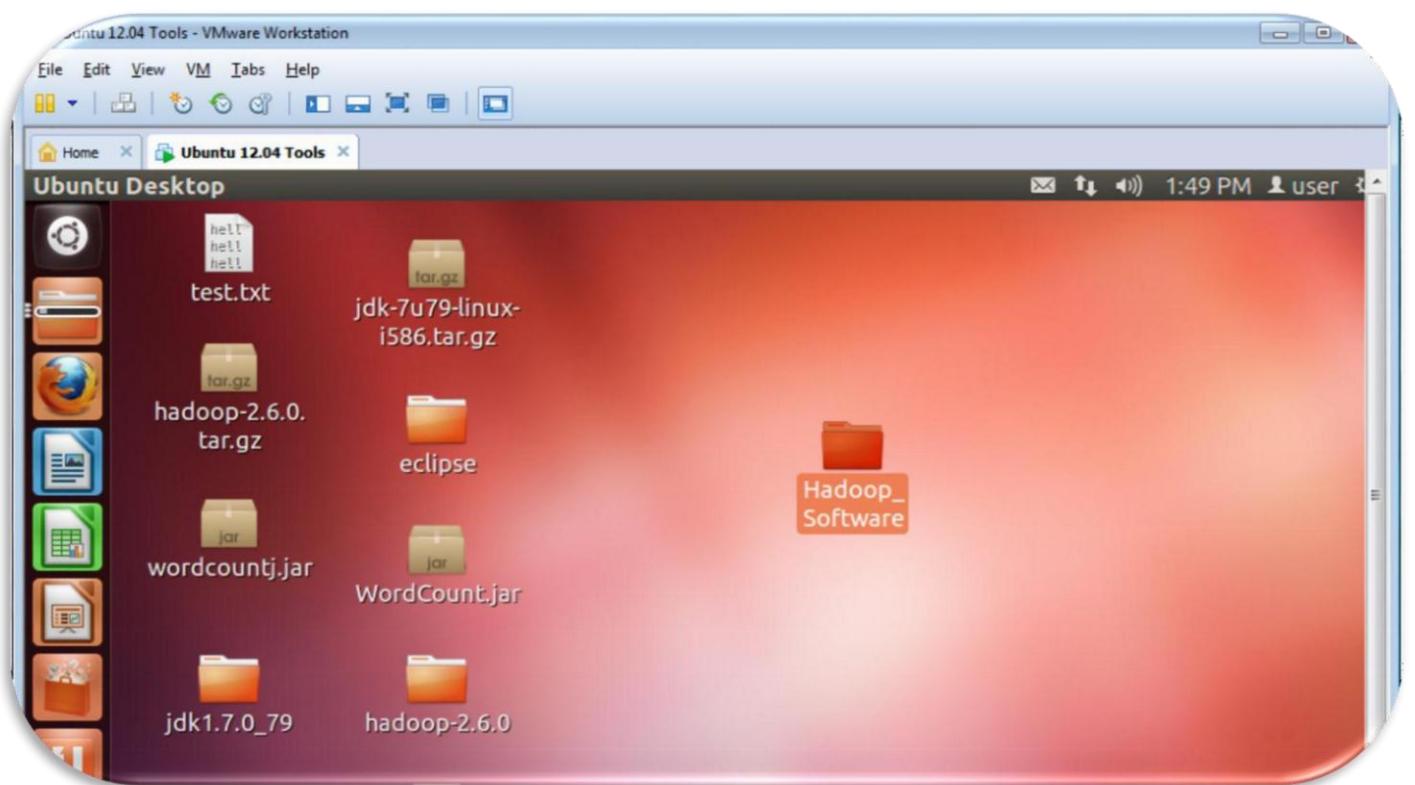
### 1.1.3 Play the Virtual Machine

You would see the below screen in VMware Workstation after the VM image creation completes.



You will get the home screen with the following image.





The user details for the Virtual instance is:

**Username : user**

**Password : password**

Open the terminal to access the file system. (**CTRL+ALT+T**) short cut key to open terminal.

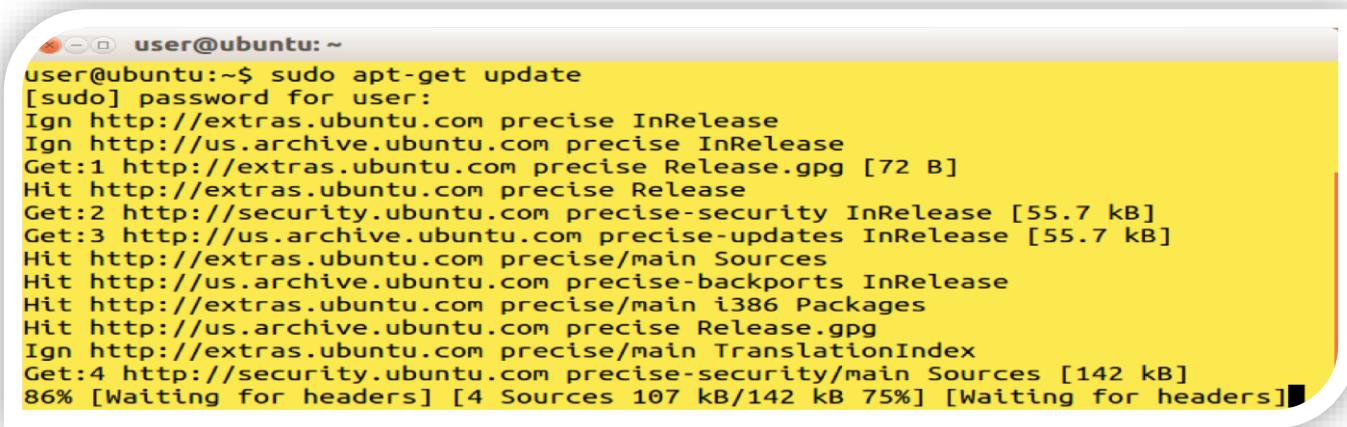




#### 1.1.4 Update the OS packages and their dependencies

The first task is to run '**apt-get update**' to download the package lists from the repositories and "update" them to get information on the newest versions of packages and their dependencies. Type command on terminal.

**\$sudo apt-get update**



**Problem Note : sudo apt-get update not working error related to lock file than**

**\$sudo rm /var/lib/apt/lists/lock**

**\$sudo rm /var/cache/apt/archives/lock**

#### 1.1.5 Install the Java and openssh server for Hadoop 2.6.0

Check java version installation.

**\$java -version**

```
user@ubuntu:~  
user@ubuntu:~/hadoop-2.6.0$ cd  
user@ubuntu:~$ java -version  
The program 'java' can be found in the following packages:  
* default-jre  
* gcj-4.6-jre-headless  
* openjdk-6-jre-headless  
* gcj-4.5-jre-headless  
* openjdk-7-jre-headless  
Try: sudo apt-get install <selected package>  
user@ubuntu:~$
```

Use apt-get to install the JDK 7.

**\$sudo apt-get install openjdk-7-jdk**

```
user@ubuntu:~  
user@ubuntu:~$ sudo apt-get install openjdk-7-jdk  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  openjdk-7-jre openjdk-7-jre-headless  
Suggested packages:  
  openjdk-7-demo openjdk-7-source visualvm icedtea-7-plugin  
  icedtea-7-jre-jamvm sun-java6-fon ts fonts-ipafont-gothic  
  fonts-ipafont-mincho ttf-telugu-fon ts ttf-oriya-fon ts ttf-kannada-fon ts  
  ttf-bengali-fon ts  
The following packages will be upgraded:  
  openjdk-7-jdk openjdk-7-jre openjdk-7-jre-headless  
3 upgraded, 0 newly installed, 0 to remove and 717 not upgraded.  
Need to get 60.0 MB of archives.  
After this operation, 152 kB disk space will be freed.  
Do you want to continue [Y/n]?
```

**\$java -version**

**Check the location of java folder where java install.**

**\$cd usr/lib/jvm**

```
user@ubuntu:~/usr/lib/jvm/java-7-openjdk-i386
user@ubuntu:~$ java -version
java version "1.7.0_101"
OpenJDK Runtime Environment (IcedTea 2.6.6) (7u101-2.6.6-0ubuntu0.12.04
.1)
OpenJDK Client VM (build 24.95-b01, mixed mode, sharing)
user@ubuntu:~$ cd /usr/l
lib/ local/
user@ubuntu:~$ cd /usr/lib/jvm/
user@ubuntu:/usr/lib/jvm$ ls
java-1.7.0-openjdk-i386  java-7-openjdk-i386
user@ubuntu:/usr/lib/jvm$ cd java-7-openjdk-i386/
user@ubuntu:/usr/lib/jvm/java-7-openjdk-i386$ ls
ASSEMBLY_EXCEPTION  docs      jre      man      THIRD_PARTY_README
bin                  include   lib      src.zip
user@ubuntu:/usr/lib/jvm/java-7-openjdk-i386$ 
user@ubuntu:/usr/lib/jvm/java-7-openjdk-i386$
```

\$sudo apt-get install openssh-server

```
user@ubuntu:~
user@ubuntu:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  openssh-client
Suggested packages:
  libpam-ssh keychain monkeysphere openssh-blacklist openssh-blacklist-extra
  rssh molly-guard
The following packages will be upgraded:
  openssh-client openssh-server
2 upgraded, 0 newly installed, 0 to remove and 718 not upgraded.
Need to get 1,302 kB of archives.
After this operation, 0 B of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

## 1.2 Download the Apache Hadoop 2.6.0 binaries

### 1.2.1 Download the Hadoop package

Download the binaries to your home directory. Use the default user 'user' for the installation.

In Live production instances a dedicated Hadoop user account for running Hadoop is used. Though, it's not mandatory to use a dedicated Hadoop user account but is recommended because this helps to separate the Hadoop installation from other software applications and user accounts running on the same machine (separating for security, permissions, backups, etc.). Click on following link to download **hadoop-2.6.0.tar.gz** file

<http://www.apache.org/dyn/closer.cgi/hadoop/common>

<http://www.apache.org/dyn/closer.cgi/hadoop/common>



We suggest the following mirror site for your download:

<http://mirror.fibergrid.in/apache/hadoop/common>

Other mirror sites are suggested below. Please use the backup mirrors only to download PGP and MD5 signatures working.

## HTTP

<http://mirror.fibergrid.in/apache/hadoop/common> [Click Here](#)

## Index of /apache/hadoop/common/hadoop-2.6.0

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#">hadoop-2.6.0-src.tar.gz</a>	2014-12-01 05:22	17M	
<a href="#">hadoop-2.6.0-src.tar.gz.mds</a>	2014-12-01 05:22	1.1K	
<a href="#">hadoop-2.6.0.tar.gz</a>	2014-12-01 05:22	186M	
<a href="#">hadoop-2.6.0.tar.gz.mds</a>	2014-12-01 05:22	958	

Apache/2.4.10 (Debian) Server at mirror.fibergrid.in Port 80

```
user@ubuntu:~/Desktop
user@ubuntu:~$ cd Desktop/
user@ubuntu:~/Desktop$ ls
eclipse
eclipse-jee-luna-SR2-linux-gtk.tar.gz
hadoop-2.6.0.tar.gz
jdk1.7.0_79
jdk-7u79-linux-i586.tar.gz
MRJobWordCount
test.txt
WordCount.jar
wordcountj.jar
```

Unzip the files and review the package content and configuration files.

\$tar -xvf hadoop-2.6.0.tar.gz

```
user@ubuntu:~/Desktop
user@ubuntu:~/Desktop$ tar -xvf hadoop-2.6.0.tar.gz
```



```
user@ubuntu:~/hadoop-2.6.0
user@ubuntu:~/Desktop$ cd
user@ubuntu:~$ pwd
/home/user
user@ubuntu:~$ cd Desktop/
user@ubuntu:~/Desktop$ mv hadoop-2.6.0 /home/user/ mv command to move hadoop-2.6.0 to /home/user
user@ubuntu:~/Desktop$ cd
user@ubuntu:~$ cd /home/user/
user@ubuntu:~$ ls
Desktop  Downloads      hadoop-2.6.0  Pictures  Templates
Documents examples.desktop  Music      Public    Videos
user@ubuntu:~$ cd $HOME
user@ubuntu:~$ pwd
/home/user
user@ubuntu:~$ cd /home/user/hadoop-2.6.0/
user@ubuntu:~/hadoop-2.6.0$ ls
bin  include  libexec  NOTICE.txt  sbin
etc  lib      LICENSE.txt  README.txt  share
user@ubuntu:~/hadoop-2.6.0$
```

```
user@ubuntu:~/Desktop/hadoop-2.6.0
user@ubuntu:~/Desktop$ cd hadoop-2.6.0/
user@ubuntu:~/Desktop/hadoop-2.6.0$ ls
bin  include  libexec  NOTICE.txt  sbin
etc  lib      LICENSE.txt  README.txt  share
user@ubuntu:~/Desktop/hadoop-2.6.0$ ls -l folder structure of hadoop2.6.0
total 52
drwxr-xr-x 2 user user 4096 Nov 13 2014 bin
drwxr-xr-x 3 user user 4096 Nov 13 2014 etc
drwxr-xr-x 2 user user 4096 Nov 13 2014 include
drwxr-xr-x 3 user user 4096 Nov 13 2014 lib
drwxr-xr-x 2 user user 4096 Nov 13 2014 libexec
-rw-r--r-- 1 user user 15429 Nov 13 2014 LICENSE.txt
-rw-r--r-- 1 user user 101 Nov 13 2014 NOTICE.txt
-rw-r--r-- 1 user user 1366 Nov 13 2014 README.txt
drwxr-xr-x 2 user user 4096 Nov 13 2014 sbin
drwxr-xr-x 4 user user 4096 Nov 13 2014 share
user@ubuntu:~/Desktop/hadoop-2.6.0$
```

Review the Hadoop configurations files.

```
user@ubuntu:~/Desktop/hadoop-2.6.0/etc/hadoop
user@ubuntu:~/Desktop/hadoop-2.6.0$ cd etc/hadoop/
user@ubuntu:~/Desktop/hadoop-2.6.0/etc/hadoop$ ls
capacity-scheduler.xml      httpfs-env.sh          mapred-env.sh
configuration.xsl           httpfs-log4j.properties  mapred-queues.xml.template
container-executor.cfg       httpfs-signature.secret mapred-site.xml.template
core-site.xml                httpfs-site.xml        slaves
hadoop-env.cmd              kms-acls.xml          ssl-client.xml.example
hadoop-env.sh                kms-env.sh            ssl-server.xml.example
hadoop-metrics2.properties   kms-log4j.properties  yarn-env.cmd
hadoop-metrics.properties    kms-site.xml          yarn-env.sh
hadoop-policy.xml            log4j.properties     yarn-site.xml
hdfs-site.xml                mapred-env.cmd
user@ubuntu:~/Desktop/hadoop-2.6.0/etc/hadoop$
```

After creating and configuring your virtual servers, the Ubuntu instance is now ready to start installation and configuration of Apache Hadoop 2.6.0 Single Node Cluster. This section describes the steps in details to install Apache Hadoop 2.6.0 and configure a Single Node Apache Hadoop cluster.

## 2. Configure the Apache Hadoop 2.6.0 Single Node Server

This section explains the steps to configure the Single Node Apache Hadoop 2.6.0 Server on Ubuntu.

### 2.1 Update the Configuration files

#### 2.1.1 Update “.bashrc” file for user ‘ubuntu’.

Must be move to ‘**user**’ \$HOME directory and edit ‘.bashrc’ file.

```
user@ubuntu:~$ cd  
user@ubuntu:~$ pwd present working directory  
/home/user  
user@ubuntu:~$ ls -al .b* displat list of hidden file which start b alphabets  
-rw----- 1 user user 3159 Mar 29 02:37 .bash_history  
-rw-r--r-- 1 user user 220 Apr 28 2012 .bash_logout  
-rw-r--r-- 1 user user 3840 Mar 26 09:44 .bashrc  
-rw-r--r-- 1 user user 3840 Mar 26 09:44 .bashrc~  
user@ubuntu:~$
```

Update the '.bashrc' file to add important Apache Hadoop environment variables for user.

a) Change directory to home.

**\$ cd**

b) Edit the file

**\$ sudo gedit .bashrc**

```
user@ubuntu:~$ sudo gedit .bashrc  
[sudo] password for user:  
user@ubuntu:~$
```

Add below lines in the .bashrc file.

**# Set Hadoop-related environment variables**

```
export HADOOP_HOME=$HOME/hadoop-2.6.0  
export HADOOP_CONF_DIR=$HOME/hadoop-2.6.0/etc/hadoop  
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.6.0  
export HADOOP_COMMON_HOME=$HOME/hadoop-2.6.0  
export HADOOP_HDFS_HOME=$HOME/hadoop-2.6.0  
export YARN_HOME=$HOME/hadoop-2.6.0  
# Set JAVA_HOME  
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386  
# Add Hadoop bin/ directory to PATH  
export PATH=$PATH:$HOME/hadoop-2.6.0/bin
```

```

export HADOOP_HOME=$HOME/hadoop-2.6.0
export HADOOP_CONF_DIR=$HOME/hadoop-2.6.0/etc/hadoop
export HADOOP_MAPRED_HOME=$HOME/hadoop-2.6.0
export HADOOP_COMMON_HOME=$HOME/hadoop-2.6.0
export HADOOP_HDFS_HOME=$HOME/hadoop-2.6.0
export YARN_HOME=$HOME/hadoop-2.6.0

export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
export PATH=$PATH:$HOME/hadoop-2.6.0/bin

```

Hadoop related environment variables

Set JAVA\_HOME

Hadoop bin/directory path

c) Source the **.bashrc** file to set the hadoop environment variables without having to invoke a new shell:

```

user@ubuntu: ~
user@ubuntu:~$ source .bashrc
user@ubuntu:~$

```



## 2.2 Setup the Hadoop Cluster

This section describes the detail steps needed for setting up the Hadoop Cluster and configuring the core Hadoop configuration files.

### 2.2.1 Configure JAVA\_HOME

Configure JAVA\_HOME in '**hadoop-env.sh**'. This file specifies environment variables that affect the JDK used by Apache Hadoop 2.6.0 daemons started by the Hadoop start-up scripts:

```
/hadoop-2.6.0/etc/hadoop$ sudo gedit hadoop-env.sh
```

```

user@ubuntu: ~/hadoop-2.6.0/etc/hadoop
user@ubuntu:~$ cd $HADOOP_CONF_DIR
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ pwd
/home/user/hadoop-2.6.0/etc/hadoop
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ sudo gedit hadoop-env.sh

```

Copy this line in haoop-env.sh

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

```
hadoop-env.sh ✘  
# 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.  
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

## 2.2.2 Create NameNode and DataNode directory

Check the \$HADOOP\_HOME value .

```
user@ubuntu: ~  
user@ubuntu:~$ echo $HADOOP_HOME  
/home/user/hadoop-2.6.0      namenode and datanode folder create in  
user@ubuntu:~$                          hadoop-2.6.0 folder
```

Create DataNode and NameNode directories to store HDFS data.

```
$sudo mkdir -p $HADOOP_HOME/hadoop2_data/hdfs/namenode
```

```
$sudo mkdir -p $HADOOP_HOME/hadoop2_data/hdfs/datanode
```

```
user@ubuntu: ~  
user@ubuntu:~/hadoop-2.6.0$ cd  
user@ubuntu:~/hadoop-2.6.0$ mkdir -p $HADOOP_HOME/hadoop2_data/hdfs/namenode  
  
user@ubuntu: ~  
user@ubuntu:~/hadoop-2.6.0$ cd  
user@ubuntu:~/hadoop-2.6.0$ mkdir -p $HADOOP_HOME/hadoop2_data/hdfs/datanode
```

Check by hadoop2\_data folder create inside /home/user/hadoop-2.6.0

```
user@ubuntu:~/hadoop-2.6.0/hadoop2_data/hdfs$ cd  
user@ubuntu:~$ cd /home/user/hadoop-2.6.0/  
user@ubuntu:~/hadoop-2.6.0$ ls  
bin  hadoop2_data  lib  LICENSE.txt  NOTICE.txt  README.txt  sbin  
etc  include       libexec  NOTICE.txt  README.txt  share  
user@ubuntu:~/hadoop-2.6.0$
```

### 2.2.3 Configure the Default File system

The 'core-site.xml' file contains the configuration settings for Apache Hadoop Core such as I/O settings that are common to HDFS, YARN and MapReduce. Configure default files-system (Parameter: fs.default.name) used by clients in core-site.xml

```
user@ubuntu:~/hadoop-2.6.0/etc/hadoop  
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ sudo gedit core-site.xml  
[sudo] password for user:  
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$
```

Note : copy <configuration> tag to </configuration> and replace same tag in core-site.xml file

```
<!-- core-site.xml -->  
<configuration>  
  <property>  
    <name>fs.default.name</name>  
    <value>hdfs://localhost:9000</value>  
  </property>  
</configuration>
```

Where *hostname* and *port* are the machine and port on which Name Node daemon runs and listens. It also informs the Name Node as to which IP and port it should bind. The commonly used port is 9000 and you can also specify IP address rather than hostname.

## 2.2.4 Configure the HDFS

This file contains the configuration settings for HDFS daemons; the Name Node and the data nodes.

Configure **hdfs-site.xml** and specify default block replication, and NameNode and DataNode directories for HDFS. 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.

```
user@ubuntu:~/hadoop-2.6.0/etc/hadoop
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ sudo gedit hdfs-site.xml
```

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.permissions</name>
<value>false</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/home/user/hadoop-2.6.0/hadoop2_data/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/home/user/hadoop-2.6.0/hadoop2_data/hdfs/datanode</value>
</property>
</configuration>
```

## 2.2.5 Configure YARN framework

This file contains the configuration settings for YARN the NodeManager.

```
user@ubuntu:~/hadoop-2.6.0/etc/hadoop
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ sudo gedit yarn-site.xml
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ █
```

```
<!-- yarn-site.xml -->
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
    <value>org.apache.hadoop.mapred.ShuffleHandler</value>
  </property>
</configuration>
```

## 2.6 Configure MapReduce framework

This file contains the configuration settings for MapReduce. Configure **mapred-site.xml** and specify framework details.

```
/hadoop-2.6.0/etc/hadoop$ cp mapred-site.xml.template mapred-site.xml
```

```
user@ubuntu:~/hadoop-2.6.0/etc/hadoop
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ cp mapred-site.xml.template mapred-site.xml
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ sudo gedit mapred-site.xml
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ █
```

```
/hadoop-2.6.0/etc/hadoop$ sudo gedit mapred-site.xml
```

```

<!-- mapread-site.xml -->

<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>

```

## 2.2.7 Edit /etc/hosts file

Give **ifconfig** in the terminal and note down the ip address. Then put this ip address in **/etc/hosts** file as mentioned in below snapshots, save the file and then close it.

```

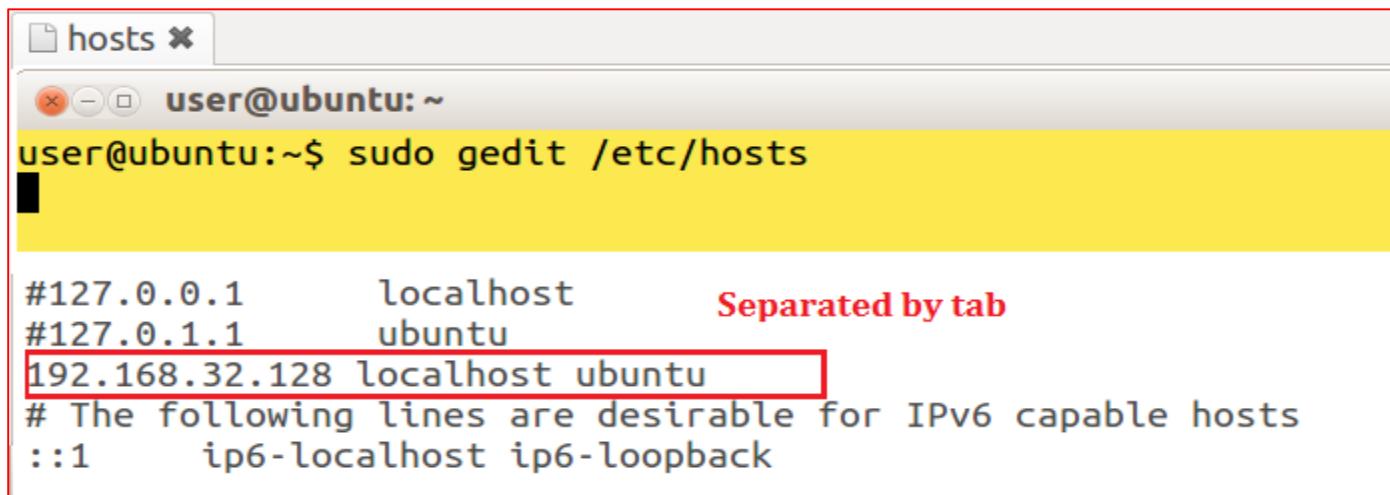
$cd
$ifconfig
user@ubuntu:~/hadoop-2.6.0/etc/hadoop$ cd
user@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:3c:b4:f6
          inet addr:192.168.32.128 Bcast:192.168.32.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe3c:b4f6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:4186 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1989 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4805186 (4.8 MB) TX bytes:147089 (147.0 KB)
          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:90 errors:0 dropped:0 overruns:0 frame:0
          TX packets:90 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7812 (7.8 KB) TX bytes:7812 (7.8 KB)

```

The **ip address** in this file, **localhost** and **ubuntu** are **separated by tab**.

```
$sudo gedit /etc/hosts
```



```
hosts * user@ubuntu: ~
user@ubuntu:~$ sudo gedit /etc/hosts
[REDACTED]

#127.0.0.1      localhost      Separated by tab
#127.0.1.1      ubuntu
192.168.32.128 localhost ubuntu
# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
```

**Note:** if not change anything in this etc/hosts file only first two line mention than also correct. etc/hosts File generally use in Multimode cluster.

## 2.2.9 Creating ssh

```
$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):  
Created directory '/home/user/.ssh'.  
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:  
38:e0:40:b9:9f:e1:c0:b4:c6:31:27:d5:12:73:b4:c0 user@ubuntu  
The key's randomart image is:  
+--[ RSA 2048]----+  
| .+=+o |  
| .* E.. |  
| +.B... |  
| Bo... . |  
| . +oo S |  
| + . |  
+-----+  
user@ubuntu:~$
```

Do not give any  
file name, just  
press enter

## 2.2.10 Moving the key to authorized key:

```
$cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

```
user@ubuntu:~$ echo $HOME  
/home/user  
user@ubuntu:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys  
user@ubuntu:~$ █
```

## 2.2. 11 start the DFS services

The first step in starting up your Hadoop installation is formatting the Hadoop file-system, which is implemented on top of the local file-systems of your cluster. This is required on the first time Hadoop installation. Do not format a running Hadoop file-system, this will cause all your data to be erased.

To format the file-system, run the command:

```
$cd  
$shadoop namenode -format
```



A screenshot of a terminal window titled "user@ubuntu: ~". The window shows the command "hadoop namenode -format" being typed into the input field. The background of the window is yellow.

```
user@ubuntu:~$ hadoop namenode -format
```

## -----Reboot the system-----

You are now all set to start the HDFS services i.e. Name Node, Resource Manager, Node Manager and Data Nodes on your Apache Hadoop Cluster.

```
$cd hadoop-2.6.0/sbin/  
$./hadoop-daemon.sh start namenode  
$./hadoop-daemon.sh start datanode
```



```
user@ubuntu:~/hadoop-2.6.0/sbin
user@ubuntu:~$ cd $HADOOP_HOME
user@ubuntu:~/hadoop-2.6.0$ cd sbin/
user@ubuntu:~/hadoop-2.6.0/sbin$ ls
distribute-exclude.sh    slaves.sh        stop-all.sh
hadoop-daemon.sh         start-all.cmd   stop-balancer.sh
hadoop-daemons.sh        start-all.sh    stop-dfs.cmd
hdfs-config.cmd          start-balancer.sh stop-dfs.sh
hdfs-config.sh            start-dfs.cmd   stop-secure-dns.sh
hs_err_pid3807.log        start-dfs.sh    stop-yarn.cmd
httpfs.sh                 start-secure-dns.sh stop-yarn.sh
kms.sh                    start-yarn.cmd   yarn-daemon.sh
mr-jobhistory-daemon.sh  start-yarn.sh    yarn-daemons.sh
refresh-namenodes.sh      stop-all.cmd
user@ubuntu:~/hadoop-2.6.0/sbin$ ./hadoop-daemon.sh start namenode
starting namenode, logging to /home/user/hadoop-2.6.0/logs/hadoop-user-namenode-ubuntu.out
user@ubuntu:~/hadoop-2.6.0/sbin$ ./hadoop-daemon.sh start datanode
starting datanode, logging to /home/user/hadoop-2.6.0/logs/hadoop-user-datanode-ubuntu.out
user@ubuntu:~/hadoop-2.6.0/sbin$ jps
4138 DataNode
4090 NameNode
4171 Jps
```

Start the YARN daemons i.e. Resource Manager and Node Manager. Cross check the service start-up using JPS (Java Process Monitoring Tool).

**\$./yarn-daemon.sh start resourcemanager  
\$./yarn-daemon.sh start nodemanager**

```
user@ubuntu:~/hadoop-2.6.0/sbin$ ./yarn-daemon.sh start resourcemanager
starting resourcemanager, logging to /home/user/hadoop-2.6.0/logs/yarn-user-resourcemanager-ubuntu.out
user@ubuntu:~/hadoop-2.6.0/sbin$ ./yarn-daemon.sh start nodemanager
starting nodemanager, logging to /home/user/hadoop-2.6.0/logs/yarn-user-nodemanager-ubuntu.out
user@ubuntu:~/hadoop-2.6.0/sbin$ jps
4692 NodeManager
4730 Jps
4138 DataNode
4090 NameNode
4635 ResourceManager
user@ubuntu:~/hadoop-2.6.0/sbin$
```

Start the History server.

**./mr-jobhistory-daemon.sh start historyserver**

```
user@ubuntu:~/hadoop-2.6.0/sbin$ ./mr-jobhistory-daemon.sh start historyserver
starting historyserver, logging to /home/user/hadoop-2.6.0/logs/mapred-user-historyserver-ubuntu.out
user@ubuntu:~/hadoop-2.6.0/sbin$ jps
4692 NodeManager
5135 Jps
4138 DataNode
4090 NameNode
5108 JobHistoryServer
4635 ResourceManager
user@ubuntu:~/hadoop-2.6.0/sbin$
```

**Note:** Always suspend your VMware Workstation, do not shut it down. So that when you open your VM again, your cluster will be up. In case you shut it down, so when you start your VM all your daemons will be down (not running). So again start all your daemons starting from namenode, do not format the namenode again.

## 2.2. 12 Perform the Health Check

### a) Check the NameNode status:

Praveen Kumar Chandaliya, DoAI, SVNIT

<http://localhost:50070/dfshealth.jsp>

The screenshot shows the HDFS健康检查页面（NameNode 'localhost:9000' (active)）。顶部显示了启动时间、版本、编译信息、集群ID和块池ID。下方有链接“Browse the filesystem”和“NameNode Logs”。  
**Cluster Summary**  
Security is OFF  
25 files and directories, 10 blocks = 35 total.  
Heap Memory used 42.37 MB is 54% of Committed Heap Memory 78.25 MB. Max Heap Memory is 966.69 MB.  
Non Heap Memory used 28.89 MB is 76% of Committed Non Heap Memory 37.97 MB. Max Non Heap Memory is 118 MB.  
以下是从该页面截取的内存使用数据表：

Configured Capacity	:	39.28 GB
DFS Used	:	148.77 MB
Non DFS Used	:	7.65 GB
DFS Remaining	:	31.48 GB
DFS Used%	:	0.37%
DFS Remaining%	:	80.15%
Block Pool Used	:	148.77 MB
Block Pool Used%	:	0.37%
DataNodes usages	:	Min %
		0.37%
		Median %
		0.37%
		Max %
		0.37%
		stdev %
Live Nodes	:	1 (Decommissioned: 0)

**b) JobHistory status:**

<http://localhost:19888/jobhistory>

The screenshot shows the Hadoop JobHistory interface at [localhost:19888/jobhistory](http://localhost:19888/jobhistory). The page title is "JobHistory". On the left, there's a sidebar with "Application" (About, Jobs), "Tools", and a search bar. The main area is titled "Retired Jobs" and displays two completed jobs:

Submit Time	Start Time	Finish Time	Job ID	Name	User	Queue	State	Maps Total	Maps Completed	Reduces Total	Reduces Completed
2016.03.29 02:47:04 EDT	2016.03.29 02:48:15 EDT	2016.03.29 03:02:45 EDT	<a href="#">job_1459222578593_0003</a>	word count job	user	default	SUCCEEDED	2	2	1	1
2016.03.29 02:35:36 EDT	2016.03.29 02:39:05 EDT	2016.03.29 02:47:50 EDT	<a href="#">job_1459222578593_0002</a>	word count job	user	default	SUCCEEDED	2	2	1	1

At the bottom, it says "Showing 1 to 2 of 2 entries" and has navigation links: First, Previous, 1, Next, Last.

### c) Browse HDFS input and output file and log information.

<http://localhost:50070>

The screenshot shows the HDFS Explorer interface at [localhost:50070/explorer.html#/home/user](http://localhost:50070/explorer.html#/home/user). The top navigation bar includes "Hadoop", "Overview", "Datanodes", "Snapshot", "Startup Progress", "Utilities" (with a note "click on utilities link ->Browse Dir"), and "Logout". The main area is titled "Browse Directory" and shows a list of files in the "/home/user" directory:

Permission	Owner	Group	Size	Replication	Block Size	Name
drwxr-xr-x	user	supergroup	0 B	0	0 B	<a href="#">input</a>
drwxr-xr-x	user	supergroup	0 B	0	0 B	<a href="#">output</a>
drwxr-xr-x	user	supergroup	0 B	0	0 B	<a href="#">output1</a>

Annotations highlight specific files:

- A red box surrounds the "input" file with the text "input directory inside hdfs use for put input file".
- A black box surrounds the "output" file with the text "output director inside hdfs use for output of program generate part-r-00000".