Hadoop Installation Steps:

Prerequisites

1. Sun Java 6

Hadoop requires a working Java 1.6+ (aka Java 6) installation. However, using Java 1.6 (aka Java 6) is recommended for running Hadoop.

```
root@ubuntu:~# java -version
java version "1.6.0_30"
OpenJDK Runtime Environment (IcedTea6 1.13.1) (6b30-1.13.1-1ubuntu2~0.10.04.2)
OpenJDK Client VM (build 23.25-b01, mixed mode, sharing)
```

2. Configuring SSH

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it .

```
root@ubuntu:~# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
f8:2c:a0:20:8f:22:07:4b:56:ff:a8:c1:5b:8f:32:d2 root@ubuntu
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|   .             |
|  . . .          |
|+o  ... S        |
|+*.. .oo         |
|= =o o..o        |
|oo E= o.         |
|  .oo. .         |
+-----------------+
root@ubuntu:~# cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
root@ubuntu:~# ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
RSA key fingerprint is 4f:8e:96:0a:26:22:94:12:ab:45:aa:f6:2c:8d:d2:2a.
```

```
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (RSA) to the list of known hosts.
Linux ubuntu 2.6.32-38-generic #83-Ubuntu SMP Wed Jan 4 11:13:04 UTC 2012 i686 GNU/Linux
Ubuntu 10.04.4 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/

247 packages can be updated.
219 updates are security updates.

New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Apr 27 19:06:47 2014 from localhost
```

Hadoop

- ## Installation

Download Hadoop from the Apache Download Mirrors and extract the contents of the Hadoop package to a location of your choice.

- ## Update $HOME/.bashrc

Add the following lines to the end of the `$HOME/.bashrc` file of user root. If you use a shell other than bash, you should of course update its appropriate configuration files instead of `.bashrc`.

Setting up Hadoop related environment variables into $HOME/.bashrc file of user root,

```
# Set Hadoop related environment variables
export HADOOP_HOME=/root/hadoop-1.1.1
export PATH=$PATH:$HADOOP_HOME/bin
export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
export PATH=$PATH:$JAVA_HOME/bin
```

- ## Configuration

Our goal is a single-node setup of Hadoop.

- ### Configuring hadoop-env.sh

The only required environment variable we have to configure for Hadoop in this practical is `JAVA_HOME`. Open `conf/hadoop-env.sh` in the editor of your choice and set the `JAVA_HOME` environment variable to the Sun JDK/JRE 6 directory.

Change
conf/hadoop-env.sh
```
1# The java implementation to use.  Required.
2# export JAVA_HOME=/usr/lib/j2sdk1.5-sun
```
to
conf/hadoop-env.sh
```
1# The java implementation to use.  Required.
2export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
```

- ### Add the following snippets between the `<configuration> ... </configuration>` tags in the respective configuration XML file.

  - #### In the core-site.xml

In this section, we will configure the directory where Hadoop will store its data files, the network ports it listens to, etc. Our setup will use Hadoop's Distributed File System, HDFS, even though our little "cluster" only contains our single local machine.

```
<configuration>
<property>
<name>hadoop.tmp.dir</name>
<value>/root/hadoop-1.1.1/temp</value>
<description>It store metadata about nodes </description>
</property>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:54310</value>
<description>Default hdfs file browser</description>
```

```
        </property>
</configuration>
```

- **In the hdfs-site.xml**

```
<configuration>
<property>
<name>dfs.replication</name>
<value>3</value>
<description>factor </description>
</property>
</configuration>
```

- **In the mapred-site.xml**

```
<configuration>
<property>
<name>mapred.job.tracker</name>
<value>localhost:54311</value>
<description> </description>
</property>
</configuration>
```

- **Formatting the HDFS filesystem via the NameNode**

The first step to starting up our Hadoop installation is formatting the Hadoop filesystem which is implemented on top of the local filesystem of our "cluster" .

To format the filesystem (which simply initializes the directory specified by the dfs.name.dirvariable), run the command,

```
root@ubuntu:~# cd hadoop-1.1.1/
root@ubuntu:~/hadoop-1.1.1# bin/hadoop namenode -format
```

The output will look like this:

```
14/04/27 19:15:36 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ubuntu/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.1.1
```

```
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.1 -r 1411108; compiled by 'hortonfo' on Mon Nov 19 10:48:11 UTC 2012
************************************************************/
14/04/27 19:15:36 INFO util.GSet: VM type       = 32-bit
14/04/27 19:15:36 INFO util.GSet: 2% max memory = 19.33375 MB
14/04/27 19:15:36 INFO util.GSet: capacity      = 2^22 = 4194304 entries
14/04/27 19:15:36 INFO util.GSet: recommended=4194304, actual=4194304
14/04/27 19:15:37 INFO namenode.FSNamesystem: fsOwner=root
14/04/27 19:15:37 INFO namenode.FSNamesystem: supergroup=supergroup
14/04/27 19:15:37 INFO namenode.FSNamesystem: isPermissionEnabled=true
14/04/27 19:15:37 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
14/04/27 19:15:37 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
14/04/27 19:15:37 INFO namenode.NameNode: Caching file names occuring more than 10 times
14/04/27 19:15:37 INFO common.Storage: Image file of size 110 saved in 0 seconds.
14/04/27 19:15:37 INFO namenode.FSEditLog: closing edit log: position=4, editlog=/root/hadoop-1.1.1/temp/dfs/name/current/edits
14/04/27 19:15:37 INFO namenode.FSEditLog: close success: truncate to 4, editlog=/root/hadoop-1.1.1/temp/dfs/name/current/edits
14/04/27 19:15:37 INFO common.Storage: Storage directory /root/hadoop-1.1.1/temp/dfs/name has been successfully formatted.
14/04/27 19:15:37 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
************************************************************/
```

- **Starting single-node cluster**

Run the command:

```
root@ubuntu:~/hadoop-1.1.1# bin/start-all.sh
```

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on your machine. The output will look like this:

```
Warning: $HADOOP_HOME is deprecated.

starting namenode, logging to /root/hadoop-1.1.1/libexec/../logs/hadoop-root-namenode-ubuntu.out
localhost: starting datanode, logging to /root/hadoop-1.1.1/libexec/../logs/hadoop-
```

```
root-datanode-ubuntu.out
localhost: starting secondarynamenode, logging to /root/hadoop-
1.1.1/libexec/../logs/hadoop-root-secondarynamenode-ubuntu.out
starting jobtracker, logging to /root/hadoop-1.1.1/libexec/../logs/hadoop-root-
jobtracker-ubuntu.out
localhost: starting tasktracker, logging to /root/hadoop-1.1.1/libexec/../logs/hadoop-
root-tasktracker-ubuntu.out
```

A nifty tool for checking whether the expected Hadoop processes are running is `jps` (part of Sun's Java since v1.5.0).

```
root@ubuntu:~/hadoop-1.1.1# jps
9057 DataNode
9561 Jps
9231 SecondaryNameNode
9486 TaskTracker
8891 NameNode
9313 JobTracker
```

You can also check with `netstat` if Hadoop is listening on the configured ports.

```
root@ubuntu:~# netstat -plten | grep java
tcp6       0      0 :::50070                :::*                    LISTEN      0          54916      8891/java
tcp6       0      0 :::49623                :::*                    LISTEN      0          54345      9057/java
tcp6       0      0 :::50010                :::*                    LISTEN      0          55022      9057/java
tcp6       0      0 :::50075                :::*                    LISTEN      0          55069      9057/java
tcp6       0      0 :::52956                :::*                    LISTEN      0          54547      9231/java
tcp6       0      0 :::33343                :::*                    LISTEN      0          54727      9313/java
tcp6       0      0 127.0.0.1:37218         :::*                    LISTEN      0          55090      9486/java
tcp6       0      0 :::57282                :::*                    LISTEN      0          53913      8891/java
tcp6       0      0 :::50020                :::*                    LISTEN      0          55746      9057/java
tcp6       0      0 127.0.0.1:54310         :::*                    LISTEN      0          54388      8891/java
tcp6       0      0 127.0.0.1:54311         :::*                    LISTEN      0          55027      9313/java
tcp6       0      0 :::50090                :::*                    LISTEN      0          55043      9231/java
tcp6       0      0 :::50060                :::*                    LISTEN      0          55352      9486/java
tcp6       0      0 :::50030                :::*                    LISTEN      0          55073      9313/java
```

- **Stopping single-node cluster**

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

```
root@ubuntu:~/hadoop-1.1.1# cd bin
root@ubuntu:~/hadoop-1.1.1/bin# ./stop-all.sh
```

The output will look like this:

```
Warning: $HADOOP_HOME is deprecated.

stopping jobtracker
localhost: stopping tasktracker
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode
```