Hadoop Installation

Sandeep Prasad

1 Introduction

Hadoop is a system to manage large quantity of data. For this report hadoop-1.0.3 (Released, May 2012) is used and tested on Ubuntu-12.04. The system configuration is Memory (RAM) 4GB, Processor Intel®Core™i3-2120 CPU @3.30GHz X 4, OS Type 32 bit. The installation of hadoop in this [1] installation report is given as

1. Prerequisites
   - (a) Java
   - (b) Dedicated user
   - (c) Configuring ssh
   - (d) Disabling ipv6
2. Installation
   - (a) .bashrc
   - (b) Changes in hadoop-env.sh and *-site.xml file
   - (c) Formatting hdfs
   - (d) Starting and stopping single-node cluster

2 Prerequisites

2.1 Java

Hadoop requires Java 1.5 or above but all the Tutorials available on web insist on Java 1.6[1][2] and above. For this installation manual Java 1.7.0_25 is used. Java 1.7 is available in Ubuntu repository and can be installed using command given in Listing 1

```
$ sudo apt-get install openjdk-7-jdk
```

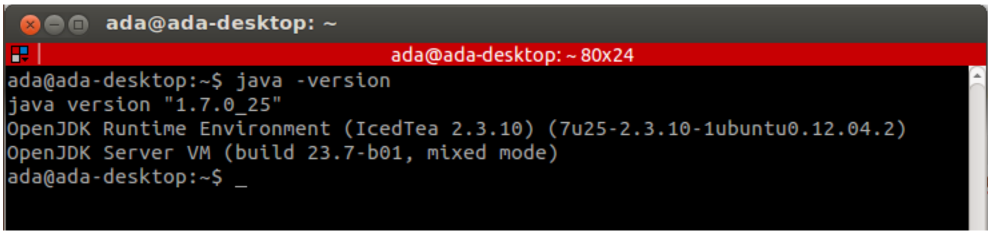
Listing 1: Installing Java 1.7

[1] http://hadoop.apache.org/docs/stable/single_node_setup.html#Required+Software

Java version can be checked using command in Listing 2 output of command shown in Figure 1

```
$ java −version
```

Listing 2: Checking Java Version



```
ada@ada-desktop:~$ java -version
java version "1.7.0_25"
OpenJDK Runtime Environment (IcedTea 2.3.10) (7u25-2.3.10-1ubuntu0.12.04.2)
OpenJDK Server VM (build 23.7-b01, mixed mode)
ada@ada-desktop:~$ _
```

Figure 1: Hadoop requires Java 1.5 or higher

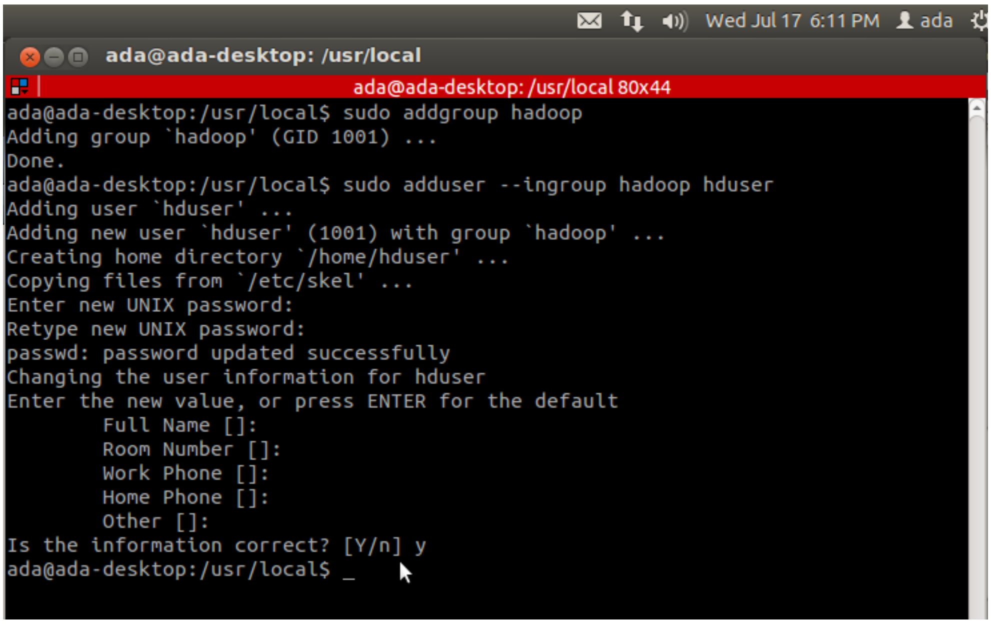
2.2 Creating dedicated user

Tutorials visited on internet advise creating a new dedicated user for using hadoop. New group is created (in this installation report new group created is “hadoop”) and user (in this installation report new user added is “hduser”) can be added to the newly created group using commands in Listing 3

```
$ sudo addgroup hadoop
$ sudo adduser −−ingroup hadoop hduser
```

Listing 3: adding group and user for hadoop

Figure 2 displays the above mentioned commands ‘for creating group and user’ executed on my system. When hduser is added it asks for new UNIX password. This password is password for hduser. Retype the password when prompted and enter the details asked (details are optional). In the end enter ‘y’ to complete the procedure.


Wed Jul 17 6:11 PM
ada
ada@ada-desktop: /usr/local
ada@ada-desktop: /usr/local 80x44

```
ada@ada-desktop:/usr/local$ sudo addgroup hadoop
Adding group `hadoop' (GID 1001) ...
Done.
ada@ada-desktop:/usr/local$ sudo adduser --ingroup hadoop hduser
Adding user `hduser' ...
Adding new user `hduser' (1001) with group `hadoop' ...
Creating home directory `/home/hduser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for hduser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
ada@ada-desktop:/usr/local$ _
```

Figure 2: Adding group hadoop and user hduser

Some steps mentioned in this manual require sudo permission. hduser can be added to sudo list using command mentioned in Listing 4.

```
$ sudo adduser hduser sudo
```

Listing 4: adding hduser to sudoers list

2.3 Configuring ssh

Ssh access is required for hadoop to run. In this installation report we will configure ssh access for localhost to user hduser. If ssh server is not installed on the machine, for Ubuntu it can be installed using command given in Listing 5

```
$ sudo apt-get install openssh-server
```

Listing 5: Installing ssh server

To allow ssh access a SSHKey has to be generated which can be generated for user hduser as followed

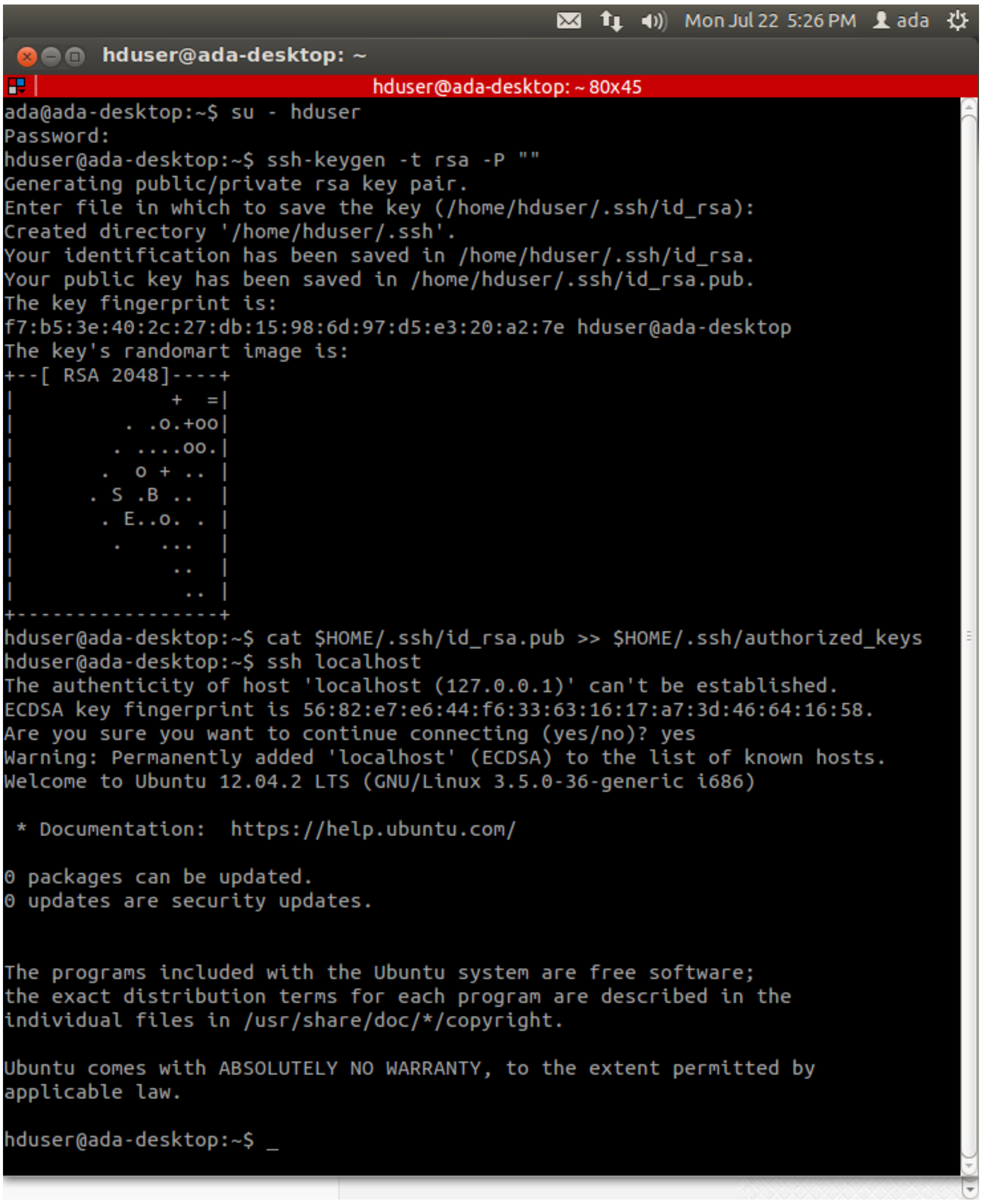
```
$ su - hduser
$ ssh-keygen -t rsa -P ""
$ cat $HOME/.ssh.id_rsa.pub >> $HOME/.ssh/authorized_keys
$ ssh localhost
```

Listing 6: Creating keygen and adding localhost to known hosts

The command given above can be explained as

1. Changing from default user to hduser, given in line 1 of Listing 6.
2. Generating keygen, when asked to enter the file to save the key, press enter and key will be saved in default /home/hduser/.ssh/id_rsa file, given in line 2 of Listing 6.
3. Authorizing public key generated as in line 3 of Listing 6.
4. Adding localhost to list of known hosts using ssh, when prompted for 'yes/no', write 'yes' and press enter, given in line 4 of Listing 6.
5. all the above steps is carried out by hduser.

Figure 3 shows the configuration steps for ssh executed on my system.



```
ada@ada-desktop:~$ su - hduser
Password:
hduser@ada-desktop:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hduser/.ssh/id_rsa):
Created directory '/home/hduser/.ssh'.
Your identification has been saved in /home/hduser/.ssh/id_rsa.
Your public key has been saved in /home/hduser/.ssh/id_rsa.pub.
The key fingerprint is:
f7:b5:3e:40:2c:27:db:15:98:6d:97:d5:e3:20:a2:7e hduser@ada-desktop
The key's randomart image is:
+--[ RSA 2048]----+
|             +  =|
|         . .o.+oo|
|        . ....oo.|
|       .  o + .. |
|      . S .B ..  |
|       . E..o. . |
|        .   ...  |
|             ..  |
|              .. |
+-----------------+
hduser@ada-desktop:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
hduser@ada-desktop:~$ ssh localhost
The authenticity of host 'localhost (127.0.0.1)' can't be established.
ECDSA key fingerprint is 56:82:e7:e6:44:f6:33:63:16:17:a7:3d:46:64:16:58.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com/

0 packages can be updated.
0 updates are security updates.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

hduser@ada-desktop:~$ _
```

Figure 3: Configuring ssh on localhost

2.4 ipv6

For hadoop to run ipv6 has to be disabled which can be done by editing /etc/sysctl.conf file. Editing sysctl.conf file requires sudo permission. Lines added in /etc/sysctl.conf file is shown in Listing 7

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

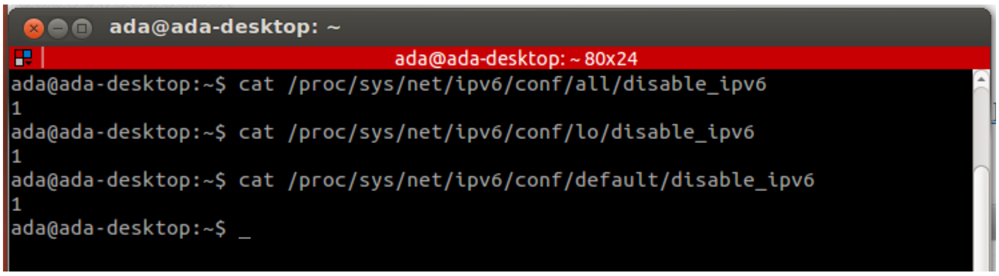
Listing 7: Lines added in /etc/sysctl.conf file

Ipv6 status can be checked using command in listing 8.

```
$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6
```

Listing 8: Checking ipv6 status after restarting system

The output will be either 0 or 1. 0 means ipv6 is enabled and 1 means it is disabled as shown in figure 4



```
ada@ada-desktop:~$ cat /proc/sys/net/ipv6/conf/all/disable_ipv6
1
ada@ada-desktop:~$ cat /proc/sys/net/ipv6/conf/lo/disable_ipv6
1
ada@ada-desktop:~$ cat /proc/sys/net/ipv6/conf/default/disable_ipv6
1
ada@ada-desktop:~$ _
```

Figure 4: Checking status of ipv6 after restarting computer

Alternatively ipv6 can be disabled only for hadoop by adding line below in /usr/local/hadoop/conf/hadoop-env.sh.

```
export HADOOP_OPTS=-Djava.net.preferIPv4Stack=true
```

3 Installation

3.1 Hadoop's folder

Copy hadoop-1.0.3.tar.gz file in /usr/local directory and untar it. Also create temporary folder which will be used by hadoop's hdfs file system (in my case temporary folder created is 'tmp' in /usr/local folder). After that we have to change ownership of hadoop and temporary directory just created. Copying the file in /usr/local, untaring, creating temporary folder and changing owner requires sudo permission. The commands executed is given in figure 5

```
1 $ sudo tar -xzf hadoop-1.0.3.tar.gz
2 $ sudo mv hadoop-1.0.3 hadoop
3 $ sudo mkdir tmp
4 $ sudo chown -R hduser:hadoop hadoop
5 $ sudo chown -R hduser:hadoop tmp
```

Listing 9: Steps to be followed before using hadoop

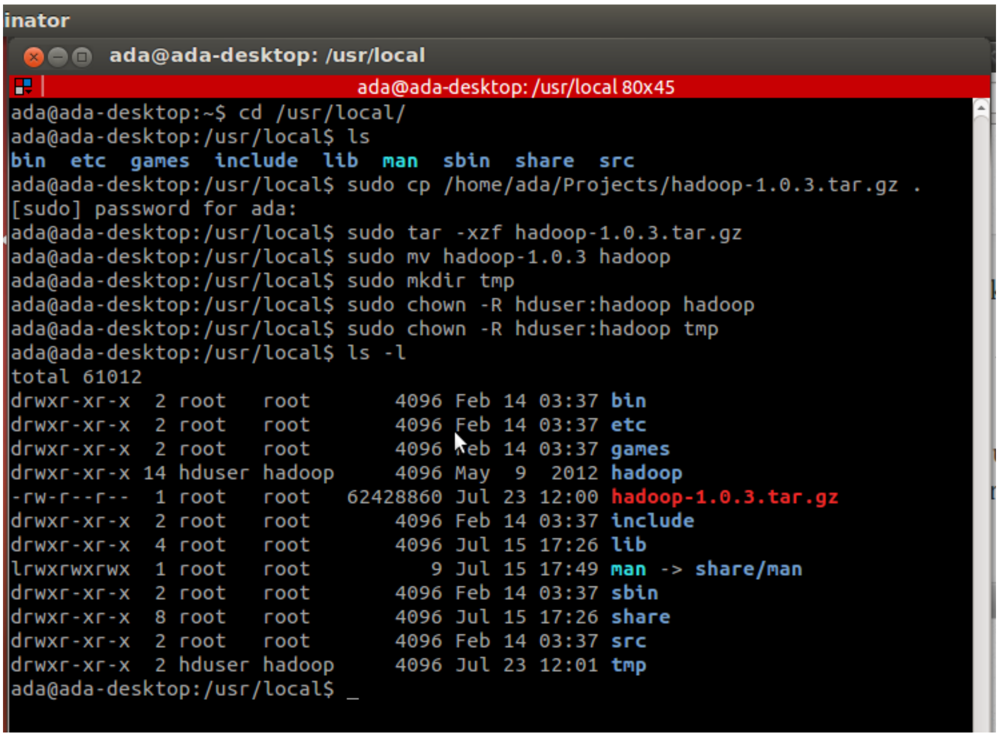
The steps mentioned in Listing 9 assumes that hadoop's tar file has been copied in /usr/local folder and user with sudo permission is in /usr/local folder (check the working folder using 'pwd' command on terminal, now the steps can be explained as below

1. untar hadoop-1.0.3.tar.gz file line 1 of Listing 9. It will create a folder called hadoop-1.0.3.

2. line 2 of Listing 9 changes the name of hadoop folder from hadoop-1.0.3 to hadoop. This step is not required but is carried out as convenience.

3. Line 3 of Listing 9 makes 'tmp' directory that will be used by hdfs as it's temporary folder and it's location will be mentioned in core-site.xml file.

4. Line 4 and line 5 of Listing 9 changes the ownership of hadoop and tmp folder from root:root to hduser:hadoop.


inator
ada@ada-desktop: /usr/local
ada@ada-desktop: /usr/local 80x45

```
ada@ada-desktop:~$ cd /usr/local/
ada@ada-desktop:/usr/local$ ls
bin  etc  games  include  lib  man  sbin  share  src
ada@ada-desktop:/usr/local$ sudo cp /home/ada/Projects/hadoop-1.0.3.tar.gz .
[sudo] password for ada:
ada@ada-desktop:/usr/local$ sudo tar -xzf hadoop-1.0.3.tar.gz
ada@ada-desktop:/usr/local$ sudo mv hadoop-1.0.3 hadoop
ada@ada-desktop:/usr/local$ sudo mkdir tmp
ada@ada-desktop:/usr/local$ sudo chown -R hduser:hadoop hadoop
ada@ada-desktop:/usr/local$ sudo chown -R hduser:hadoop tmp
ada@ada-desktop:/usr/local$ ls -l
total 61012
drwxr-xr-x  2 root   root       4096 Feb 14 03:37 bin
drwxr-xr-x  2 root   root       4096 Feb 14 03:37 etc
drwxr-xr-x  2 root   root       4096 Feb 14 03:37 games
drwxr-xr-x 14 hduser hadoop     4096 May  9  2012 hadoop
-rw-r--r--  1 root   root   62428860 Jul 23 12:00 hadoop-1.0.3.tar.gz
drwxr-xr-x  2 root   root       4096 Feb 14 03:37 include
drwxr-xr-x  4 root   root       4096 Jul 15 17:26 lib
lrwxrwxrwx  1 root   root          9 Jul 15 17:49 man -> share/man
drwxr-xr-x  2 root   root       4096 Feb 14 03:37 sbin
drwxr-xr-x  8 root   root       4096 Jul 15 17:26 share
drwxr-xr-x  2 root   root       4096 Feb 14 03:37 src
drwxr-xr-x  2 hduser hadoop     4096 Jul 23 12:01 tmp
ada@ada-desktop:/usr/local$ _
```

Figure 5: Steps executed with sudo user

3.2 Updating .bashrc for hduser

We can edit .bashrc file for hduser. The edited .bashrc file is shown in Listing 10

```
#Set Hadoop-related environment variables
export HADOOP_HOME=/usr/local/hadoop

#Set JAVA_HOME= (we will also configure JAVA_HOME directly for Hadoop later on)
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386

#Some convenient aliases and functions for running Hadoop-related commands
unalias fs &> /dev/null
alias fs="hadoop fs"
unalias hls &> /dev/null
alias hls="fs -ls"

#If you have LZO compression enabled in your hadoop cluster and
#compress job outputs with LZOP (not covered in this tutorial)
#Conveniently inspect an LZOP compressed file from the command
#line: run via:
#
# $ lzohead /hdfs/path/to/lzop/compressed/file.zo
#
#Requires installed 'lzop' command
lzohead () {
        hadoop fs -cat $1 | lzop -dc | head -1000 | less
}

#Add Hadoop bin/ directory to PATH
export PATH=$PATH:$HADOOP_HOME/bin
```

Listing 10: Changes made in .bashrc file for hduser

Hadoop uses lzop which is a compression tool. In Ubuntu lzop can be installed using command in Listing 11

```
 $ sudo apt-get install lzop
```

Listing 11: Installing lzop in Ubuntu

3.3 Changes in Hadoop folder

In Hadoop's folder we have to edit few files for hadoop to run. The files can be found in /usr/local/hadoop/conf directory. The files are hadoop-env.sh, core-site.xml, hdfs-site.xml and mapred-site.xml. This changes can be done using user 'hduser'.

3.3.1 hadoop-env.sh

In hadoop-env.sh we have to define path for JAVA_HOME. By default it will be commented and it's value will be set to j2sdk1.5-sun as shown in Listing 12, un-comment it and change it's value to the Java to be used. Original and edited hadoop-env.sh files are given in Listing 12 and Listing 13 respectively.

```
# The java implementation to use.  Required.
# export JAVA_HOME=/usr/lib/j2sdk1.5-sun
```

Listing 12: Java path in original hadoop-env.sh

```
# The java implementation to use.  Required.
 export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-i386
```

Listing 13: Java path provided in hdfs-env.sh

3.3.2 core-site.xml

Edited core-site.xml is given in Listing 14. Notice the <value> field in first <property> tag, the value points to 'tmp' folder we created earlier as mentioned in Line 3 of Listing 9.

```
<?xml version=”1.0”?>
<?xml-stylesheet type=”text/xsl” href=”configuration.xsl”?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
        <property>
                <name>hadoop.tmp.dir</name>
                <value>/usr/local/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>
```

Listing 14: Edited core-site.xml

3.3.3 hdfs-site.xml

Edited hdfs-site.xml file is given in Listing 15

```
<?xml version=”1.0”?>
<?xml-stylesheet type=”text/xsl” href=”configuration.xsl”?>

<!-- Put site-specific property overrides in this file. -->

<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>
</configuration>
```

Listing 15: Edited hdfs-site.xml

3.3.4 mapred-site.xml

Edited mapred-site.xml file is given in Listing 16

```
<?xml version=”1.0”?>
<?xml-stylesheet type=”text/xsl” href=”configuration.xsl”?>

<!-- Put site-specific property overrides in this file. -->

<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>
```

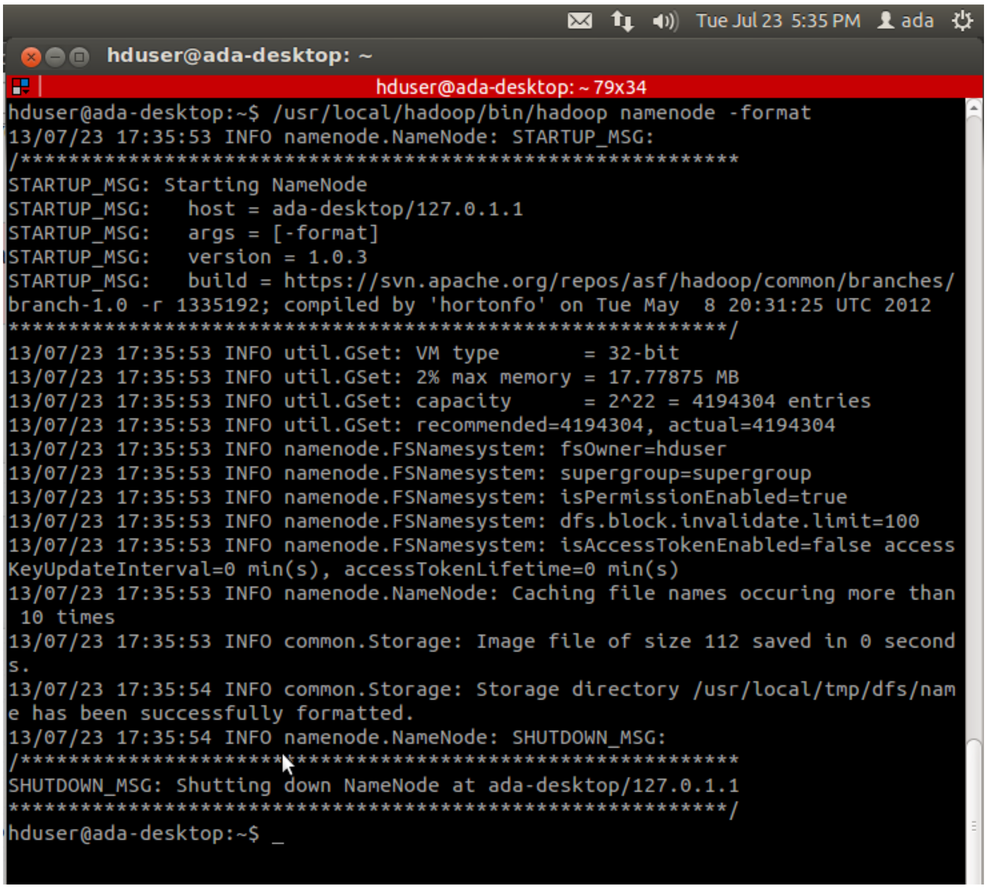
Listing 16: Edited mapred-site.xml

3.4 Formating hdfs FileSystem

Formatting hdfs FileSystem will format the virtually created File System. Anything stored in the cluster will be lost. hdfs can be formatted using command given in Listing 17. Figure 6 shows the the output obtained by formatting hdfs on my system.

```
$ /usr/local/hadoop/bin/hadoop namenode −format
```

Listing 17: formatting hdfs

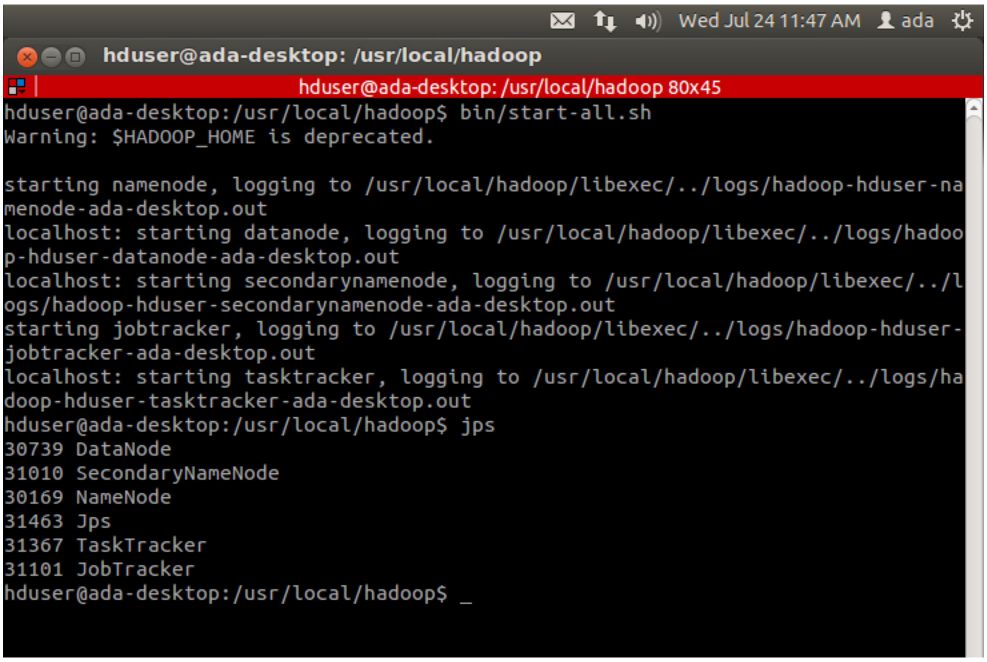


```
hduser@ada-desktop:~$ /usr/local/hadoop/bin/hadoop namenode -format
13/07/23 17:35:53 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = ada-desktop/127.0.1.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 1.0.3
STARTUP_MSG:   build = https://svn.apache.org/repos/asf/hadoop/common/branches/
branch-1.0 -r 1335192; compiled by 'hortonfo' on Tue May  8 20:31:25 UTC 2012
************************************************************/
13/07/23 17:35:53 INFO util.GSet: VM type       = 32-bit
13/07/23 17:35:53 INFO util.GSet: 2% max memory = 17.77875 MB
13/07/23 17:35:53 INFO util.GSet: capacity      = 2^22 = 4194304 entries
13/07/23 17:35:53 INFO util.GSet: recommended=4194304, actual=4194304
13/07/23 17:35:53 INFO namenode.FSNamesystem: fsOwner=hduser
13/07/23 17:35:53 INFO namenode.FSNamesystem: supergroup=supergroup
13/07/23 17:35:53 INFO namenode.FSNamesystem: isPermissionEnabled=true
13/07/23 17:35:53 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
13/07/23 17:35:53 INFO namenode.FSNamesystem: isAccessTokenEnabled=false access
KeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
13/07/23 17:35:53 INFO namenode.NameNode: Caching file names occuring more than
 10 times
13/07/23 17:35:53 INFO common.Storage: Image file of size 112 saved in 0 second
s.
13/07/23 17:35:54 INFO common.Storage: Storage directory /usr/local/tmp/dfs/nam
e has been successfully formatted.
13/07/23 17:35:54 INFO namenode.NameNode: SHUTDOWN_MSG:
/************************************************************
SHUTDOWN_MSG: Shutting down NameNode at ada-desktop/127.0.1.1
************************************************************/
hduser@ada-desktop:~$ _
```

Figure 6: Output when hdfs is formatted

3.5 Starting and stopping hdfs

After completing all prerequisites, installation steps mentioned and formatting hdfs, hadoop is ready for use. Hadoop can be started and stopped using the start and stop script available in bin directory (done using hduser). Script to start and stop hadoop when run on my system are given in figure 7 and figure 8 respectively. The command to start hadoop services is (it is assumed you are in /usr/local/hadoop directory). Figure 7 also mentions jps, jps is a tool available in Java used to check the services started. When start script is executed the services started are DataNode, SecondaryNameNode, NameNode, TaskTracker and JobTracker.

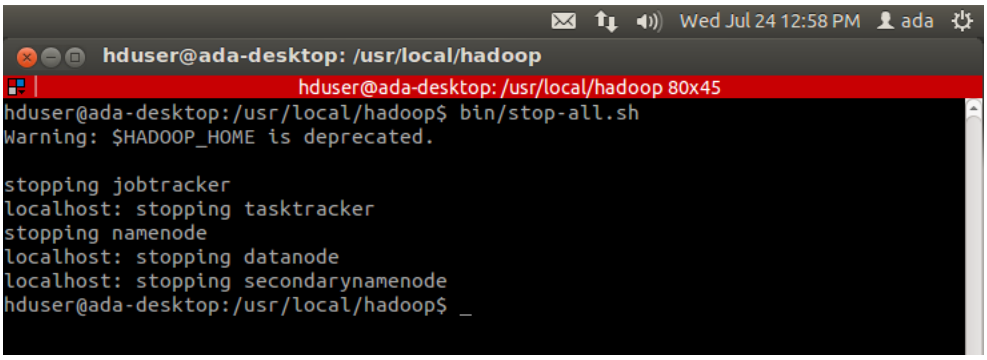


Wed Jul 24 11:47 AM
ada
hduser@ada-desktop: /usr/local/hadoop
hduser@ada-desktop: /usr/local/hadoop 80x45

```
hduser@ada-desktop:/usr/local/hadoop$ bin/start-all.sh
Warning: $HADOOP_HOME is deprecated.

starting namenode, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-na
menode-ada-desktop.out
localhost: starting datanode, logging to /usr/local/hadoop/libexec/../logs/hadoo
p-hduser-datanode-ada-desktop.out
localhost: starting secondarynamenode, logging to /usr/local/hadoop/libexec/../l
ogs/hadoop-hduser-secondarynamenode-ada-desktop.out
starting jobtracker, logging to /usr/local/hadoop/libexec/../logs/hadoop-hduser-
jobtracker-ada-desktop.out
localhost: starting tasktracker, logging to /usr/local/hadoop/libexec/../logs/ha
doop-hduser-tasktracker-ada-desktop.out
hduser@ada-desktop:/usr/local/hadoop$ jps
30739 DataNode
31010 SecondaryNameNode
30169 NameNode
31463 Jps
31367 TaskTracker
31101 JobTracker
hduser@ada-desktop:/usr/local/hadoop$ _
```

Figure 7: Starting hadoop and checking the status of started processes using jps



Wed Jul 24 12:58 PM
ada
hduser@ada-desktop: /usr/local/hadoop
hduser@ada-desktop: /usr/local/hadoop 80x45

```
hduser@ada-desktop:/usr/local/hadoop$ bin/stop-all.sh
Warning: $HADOOP_HOME is deprecated.

stopping jobtracker
localhost: stopping tasktracker
stopping namenode
localhost: stopping datanode
localhost: stopping secondarynamenode
hduser@ada-desktop:/usr/local/hadoop$ _
```

Figure 8: Stopping hadoop processes

References

[1] Michael G. Noll. Running hadoop on ubuntu linux (single-node cluster) - michael g. noll. http://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/.

[2] Hadoop 1.1.2 Documentation. Single node setup. http://hadoop.apache.org/docs/stable/index.html.