Manual for installing and running single node Hadoop cluster on Ubuntu

This document explain required steps for setting up a *pseudo-distributed, single-node* Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux.

Here are the steps needed:

1. Install Java
2. Add User and User Group
3. Configure password less SSH authentication
4. Generate private public rsa key pair
5. Install Hadoop
6. bashrc configuration
7. Configure Hadoop
8. Starting single node cluster
9. Stopping single node cluster
10. Running some command on hadoop

Step 1 - Install Java:

Please run following commands on command prompt. This will require sudo access for the user.

```
//Install python-software-properties This will manage the repositories
sudo apt-get install python-software-properties

//add repository for java
sudo add-apt-repository ppa:ferramroberto/java

//update repository
sudo apt-get update

//install sun-java6
sudo apt-get install sun-java6-jdk

//update java-alternatives
sudo update-java-alternatives -s java-6-sun
```

After installing java JDK directory will be placed here **`/usr/lib/jvm/java-6-sun`**.

Step 2 - Add User and User Group:

We will use hadoop user group and hduser as user for all assignments and tutorials.

```
//Add a group hadoop
sudo addgroup hadoop

//add user hduser and set group as hadoop, this will ask for password for
setup
sudo adduser --ingroup hadoop hduser
```

Step 3 - Configure password less SSH authentication:

Password less access through SSH is required by Hadoop for communication between nodes. Assuming that SSH is already setup and running.

Run following command for setting up password less authentication

```
//Login as hduser, this will ask for password for hduser
user@hadoop:~$ su - hduser
```

Step 4 - Generate public/private rsa key-pair

1) Generate the key

Use the command below. This will ask for file in which to save the key, leave it blank

```
hduser@hadoop:~$  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:
9b:82:ea:58:b4:e0:35:d7:ff:19:66:a6:ef:ae:0e:d2 hduser@ubuntu
The key's randomart image is:
```

2) Copy the key

Key is generated in file /home/hduser/.ssh/id_rsa.pub.

This should be copied in file /home/hduser/.ssh/authorized_keys. For that run below command

```
hduser@hadoop:~$ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

Password less authentication is done.

To verify just use "`ssh localhost`".

Step 5 - Installing Hadoop:

Download Hadoop from here

(http://mirror.cogentco.com/pub/apache/hadoop/core/hadoop-1.2.1/hadoop-1.2.1.tar.gz)

or select your nearest mirror from here

(http://www.apache.org/dyn/closer.cgi/hadoop/core)

and then select hadoop-1.2.1/hadoop-1.2.1.tar.gz

Use your sudo user to execute following command for installing apache Hadoop.

```
//change directory to /usr/local folder
$ cd /usr/local

//untar hadoop
$ sudo tar xzf /tmp/hadoop-1.0.3.tar.gz

//change name
$ sudo mv hadoop-1.0.3 hadoop

//change ownership to hduser
$ sudo chown -R hduser:hadoop hadoop
```

Step 6 - bashrc configuration

1) Open vi editor

```
vi /home/hduser/.bashrc
```

2) Add following line of code at the end of your bashrc files.

```
# Hadoop Home Path
export HADOOP_HOME=/usr/local/hadoop

# Java home path
export JAVA_HOME=/usr/lib/jvm/java-6-sun

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

3) Save and exit vi editor by typing

```
:wq
```

4) Run source command to reflect the changes in .bashrc

```
hduser@ubuntu:~$source /home/hduser/.bashrc
```

Step 7 - Hadoop Configuration:

Before doing further configuration please do run following command for creating a directory which will be used by hadoop to keep the data. These command should be used with sudo user.

```
$ sudo mkdir -p /app/hadoop/data
$ sudo chown hduser:hadoop /app/hadoop/data
```

1) hadoop-env.sh:

Path: /usr/local/hadoop/conf/hadoop-env.sh

a) Start vi editor

```
vi /usr/local/hadoop/conf/hadoop-env.sh
```

b) Change only `$JAVA_HOME` variable to your java home.

~~# export JAVA_HOME=/usr/lib/j2sdk1.5-sun~~

```
export JAVA_HOME=/usr/lib/jvm/java-6-sun
```

c) Exit vi editor

```
:wq
```

2) conf/core-site.xml:

Path: /usr/local/hadoop/conf/core-site.xml

a) Start vi editor

```
vi /usr/local/hadoop/conf/core-site.xml
```

b) Add following lines between `<configuration> ... </configuration>` tags.

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/data</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>
```

c) Exit vi editor

```
:wq
```

3) conf/mapred-site.xml:

Path: /usr/local/hadoop/conf/mapred-site.xml

a) Start vi editor

```
vi /usr/local/hadoop/conf/mapred-site.xml
```

b) Add following lines between `<configuration> ... </configuration>` tags.

```
<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>
```

c) Exit vi editor

```
:wq
```

4) conf/hdfs-site.xml:

Path: /usr/local/hadoop/conf/hdfs-site.xml

a) Start vi editor

```
vi /usr/local/hadoop/conf/hdfs-site.xml
```

b) Add following lines between `<configuration> ... </configuration>` tags.

```
<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>
```

c) Exit vi editor

```
:wq
```

5) Format HDFS file system:

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

Step 8 - Starting single node cluster

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

1) Use command `jps` for checking hadoop processes running:

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

2287 TaskTracker

2149 JobTracker

1938 DataNode

2085 SecondaryNameNode

2349 Jps

1788 NameNode
```

Step 9 - Stopping single node cluster

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

Step 10 - Running some command on hadoop:

1) Creating directory in HDFS:

```
hduser@ubuntu:~$hadoop dfs -mkdir /test/
```

2) Create sample files

Create 2 sample files named as /home/hduser/test/test1.csv, /home/hduser/test/test2.csv

3) Upload two files in HDFS in directory /test/

```
hduser@ubuntu:~$ hadoop dfs -copyFromLocal /home/hduser/test/test1.csv /test/
```

```
hduser@ubuntu:~$ hadoop dfs -copyFromLocal /home/hduser/test/test2.csv /test/
```

4) List directory /test/

```
hduser@ubuntu:~$ hadoop dfs -ls  /test/
```

```
Found 2 items
-rw-r--r--   1 hduser supergroup         28 2014-03-30 19:13 /test/test1.csv
-rw-r--r--   1 hduser supergroup         24 2014-03-30 19:13 /test/test2.csv
```