DeZyre

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 2.x cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux. Steps are almost same as Hadoop 1 with few changes.

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 openjdk-7-jdk

//update java-alternatives
sudo update-java-alternatives -s openjdk-7-jdk
```

After installing java JDK directory will be placed here **`/usr/lib/jvm/`**

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://www.trieuvan.com/apache/hadoop/common/hadoop-2.2.0/hadoop-2.2.0.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-2.2.0.tar.gz

//change name
$ sudo mv hadoop-2.2.0 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.

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
```

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:**

6. Hadoop Configuration:

All configuration files for hadoop 2 exist at path /usr/local/hadoop/etc/hadoop.

So perform following command

$ cd /usr/local/hadoop/etc/hadoop

a) **hadoop-env.sh**: Change only $JAVA_HOME variable to your java home.

gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh

~~# export JAVA_HOME=/usr/lib/j2sdk1.5-sun~~
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64

b) **core-site.xml**

$ gedit /usr/local/hadoop/etc/hadoop/core-site.xml

#Paste following between <configuration>

```
<property>
   <name>fs.default.name</name>
   <value>hdfs://localhost:9000</value>
</property>
```

c) **yarn-site.xml**

$ gedit /usr/local/hadoop/etc/hadoop/yarn-site.xml

#Paste following between <configuration>

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

d) **mapred-site.xml**

Perform following commands

$ mv mapred-site.xml.template mapred-site.xml

$ gedit mapred-site.xml

#Paste following between <configuration>

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

e) **hdfs-site.xml**

Perform following command to create two directory.

$ cd ~

$ mkdir -p mydata/hdfs/namenode

$ mkdir -p mydata/hdfs/datanode

$ cd /usr/local/hadoop/etc/hadoop

$ gedit hdfs-site.xml

Paste following between <configuration> tag

```
<property>
   <name>dfs.replication</name>
   <value>1</value>
 </property>
 <property>
   <name>dfs.namenode.name.dir</name>
   <value>file:/home/hduser/mydata/hdfs/namenode</value>
 </property>
 <property>
   <name>dfs.datanode.data.dir</name>
   <value>file:/home/hduser/mydata/hdfs/datanode</value>
 </property>
```

Format HDFS file system:

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

7. Starting single node cluster

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

Checking hadoop processes running:

hduser@ubuntu:/usr/local/hadoop$ jps
2287 TaskTracker
2149 JobTracker
1938 DataNode
2085 SecondaryNameNode
2349 Jps
1788 NameNode

8. Stopping single node cluster

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

9. **Running some command on hadoop**:

Creating directory in HDFS:

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

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

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/

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