Hadoop Installation & Configuration

This installation and configuration is tested on Ubuntu{16.04} using the Hadoop version 2.7.3

1 Pre-Installation

1.1 Upate and Upgrade the packages

Update and upgrade all the current packages that are installed in your operating system

```
sudo apt-get udate
sudo apt-get upgrade
```

1.2 Install Java

This includes installing **jdk** (Java Development Kit: To compile your java programs) and **jre** (Java Runtime Environment: To Excecute the java programs)

```
sudo apt-get install default-jdk
```

The above command must alone install jdk as well as jre
Verify this by excecuting the following commands:

```
javac
java
```

Both these commands must produce output similar to as follows:

```
"Usage: java ......
.
.
."
```

In case in **java** command does not work, then execute the following command:

```
sudo apt-get install default-jre
```

1.3 Install openssh-server

SSH (**S**ecure **SH**ell) is a protocol for securely accessing one machine from another. **Hadoop** uses SSH for accessing another slaves nodes to start and manage all **HDFS** and **MapReduce** daemons.
It can be installed by using following command:

```
sudo apt-get install ssh
```

1.4 Create Hadoop User

To avoid security issues, it is recommend to setup new Hadoop user group and user account to deal with all Hadoop related activities.
Following are the commands to create Hadoop group as **hadoop** and hadoop user as **hduser** within the **hadoop** user group.

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

1.5 Give super user previleges to “hduser"

Certain commands requires superuser previleges, these commands can be executed using the command **sudo** prepended to other commands. Initially the OS may not allow hduser to excecute these commands. To allow this to happen, perform the following from the user account using which **hduser** is created:
open the visudo file using the following:

```
sudo visudo
```

Add the following contents at the end of the sudoer file

```
hduser ALL=(ALL:ALL) ALL
```

Note: Now Switch your user account to *hduser* and open the terminal

1.6 Configure SSH

Once you have installed SSH on your machine, you can connect to other machine or allow other machines to connect with this machine. However we have this single machine, we can try connecting with this same machine by SSH. To do this, we need to copy generated RSA key (i.e. id_rsa.pub) pairs to authorized_keys folder of SSH installation of this machine by the following command:

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

1.7 Disabling IPv6

Since Hadoop doesn?t work on IPv6, we should disable it. One of another reason is also that it has been developed and tested on IPv4 stacks. Hadoop nodes will be able to communicate if we are having IPv4 cluster. (Once you have disabled IPV6 on your machine, you need to reboot your machine in order to check its effect)

To disable IPv6 open the file **/etc/sysctl.conf** and add the following contents at the end of the file.

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

2 Installation Steps

2.1 Download Hadoop

Download the lates hadoop from **http://hadoop.apache.org/releases.html**
Click here to download

2.2 Move downloaded hadoop file to home directory

The downloaded files are usually present in the *Downloads* directory, which can be moved to the *Home* directory for the convinience. The **cp** command can be used for this purpose. Usage is as follows:

```
cp <Source> <Destination>
```

2.3 Extract the Hadoop files and move it to “/usr/local" directory

The Downloaded Hadoop file are in the compressed with **.tar.gz** format. It has to be extracted and moved to **/usr/local** directory as follows:

```
sudo tar -xzvf hadoop-2.7.3.tar.gz
sudo mv hadoop-2.7.3 /usr/local/hadoop
```

2.4 Assign ownership of *Hadoop* folder to Hadoop user

The ownership of the */usr/local/hadoop* directory is given to ***hduser*** as follows:

```
sudo chown hduser:hadoop -R /usr/local/hadoop
```

2.5 Create Hadoop temp directories for Namenode and Datanode

Use the following commands to create the temporary directories:

```
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/namenode
sudo mkdir -p /usr/local/hadoop_tmp/hdfs/datanode
```

2.6 Assign ownership of temporary folders to Hadoop user

The temporary directories of namenode and datanode must be assigned the ownership of *hadoop* user. Use the following commands to assign the ownlership to *hadoop* user:

```
sudo chown hduser:hadoop -R /usr/local/hadoop_tmp/
```

3 Update User and Hadoop configuration files

3.1 Update the user profile configuration file- *.bashrc*

.bashrc is a shell script that Bash runs whenever it is started interactively. You can put any command in that file that you could type at the command prompt. You put commands here to set up the shell for use in your particular environment, or to customize things to your preferences.

Add the following contents as the end of **~/.bashrc** file.

Note: ~ is a shortcut for “Home" directory, altetrnatively $HOME can be used instead of ~

```
# – HADOOP ENVIRONMENT VARIABLES START – #
export JAVA_HOME=/usr/lib/jvm/default-java
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
# – HADOOP ENVIRONMENT VARIABLES END – #
```

3.2 Configure the file : hadoop-env.sh

Open the file ***/usr/local/hadoop/etc/hadoop/hadoop-env.sh*** and modify the **JAVA_HOME** variable as follows:

```
JAVA_HOME=/usr/lib/jvm/default-java
```

3.3 Configure the file : core-site.xml

Open the file ***/usr/local/hadoop/etc/hadoop/core-site.xml*** and Paste the below lines in between <configuration> tag

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

3.4 Configure the file : hdfs-site.xml.xml

Open the file ***/usr/local/hadoop/etc/hadoop/hdfs-site.xml*** and Paste the below lines in between <configuration> tag

```
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:/usr/local/hadoop_tmp/hdfs/datanode</value>
</property>
```

3.5 Configure the file : yarn-site.xml

Open the file ***/usr/local/hadoop/etc/hadoop/yarn-site.xml*** and Paste the below lines in between <configuration> tag

```
<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>
```

3.6 Configure the file : mapred-site.xml

Copy the file **/usr/local/hadoop/etc/hadoop/mapred-site.xml.template** to **/usr/local/hadoop/etc/hadoop/mapr site.xml**

Open the file ***/usr/local/hadoop/etc/hadoop/mapred-site.xml*** and Paste the below lines in between <configuration> tag

```
cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
/usr/local/hadoop/etc/hadoop/mapred-site.xml
```

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

3.7 Update the current Shell

Use the folloeing command to update the current shell to take the values changed in *~/.bashrc* file.

```
source ~/.bashrc
```

4 Post Hadoop Installation Steps

These steps verify the installation of hadoop.

4.1 Format the name node

The *namenode* can be formatted as follows:

```
hdfs namenode -format
```

4.2 Start all Hadoop daemons

The *HDFS* and *MapReduce* deamons can be started as follows:

```
start-dfs.sh
start-yarn.sh
```

OR

It may also be started as follows.

```
start-all.sh
```

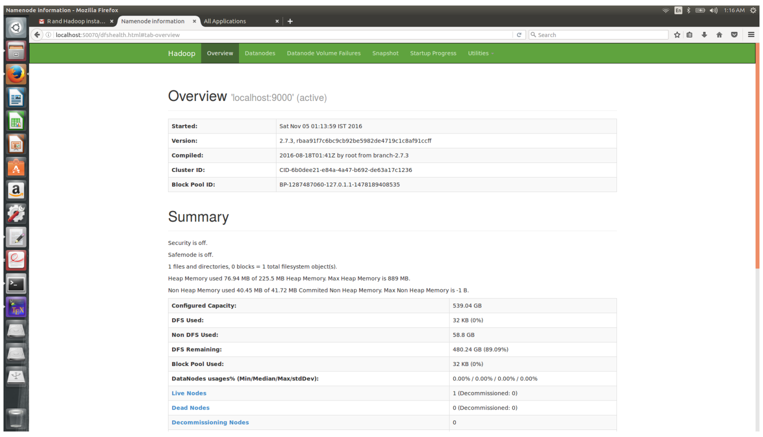
4.3 Track/Monitor/Verify

Verify Hadoop daemons:

```
jps
```

4.4 Accessing Hadoop on Browser

```
http://localhost:50070/
```



Overview 'localhost:9000' (active)

Started:	Sat Nov 05 01:13:59 IST 2016
Version:	2.7.3, rbaa91f7c6bc9cb92be5982de4719c1c8af91ccff
Compiled:	2016-08-18T01:41Z by root from branch-2.7.3
Cluster ID:	CID-6b0dee21-e84a-4a47-b692-de63a17c1236
Block Pool ID:	BP-1287487060-127.0.1.1-1478189408535

Summary

Security is off.

Safemode is off.

1 files and directories, 0 blocks = 1 total filesystem object(s).

Heap Memory used 76.94 MB of 225.5 MB Heap Memory. Max Heap Memory is 889 MB.

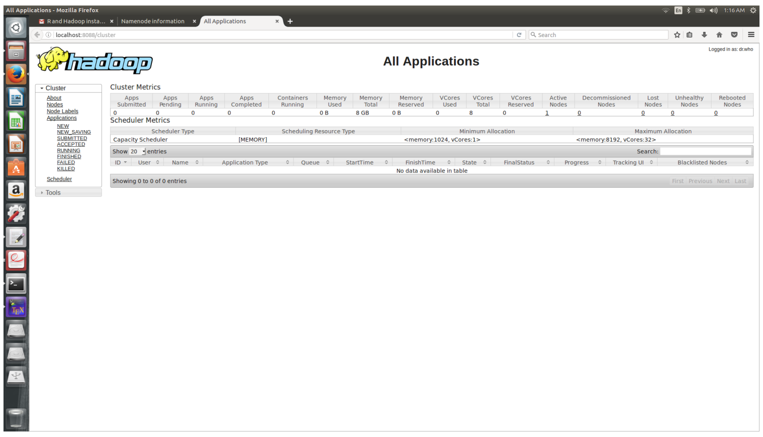
Non Heap Memory used 40.45 MB of 41.72 MB Commited Non Heap Memory. Max Non Heap Memory is -1 B.

Configured Capacity:	539.04 GB
DFS Used:	32 KB (0%)
Non DFS Used:	58.8 GB
DFS Remaining:	480.24 GB (89.09%)
Block Pool Used:	32 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0

Figure 1: Accessing Hadoop on Browser

4.5 Verify All Applications for Cluster

http://localhost:8088/



All Applications

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	8 GB	0 B	0	8	0	1	0	0	0	0

Scheduler Metrics

Scheduler Type	Scheduling Resource Type	Minimum Allocation	Maximum Allocation
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:32>

Show 20 entries

Search:

ID	User	Name	Application Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklisted Nodes

No data available in table

Showing 0 to 0 of 0 entries

Figure 2: All Applications