**Steps to follow when you want to setup a PSEUDO DISTRIBUTED MODE OF HADOOP CLUSTER**

Once your VM BOX is setup and multiple linux machines are setup in VM BOX. Make sure that you are using BRIDGED ADAPTER in network settings of your linux machine.

* Have Java setup, preferably java v6/7 ,JDK.

Ubuntu users: sudo apt-get install openjdk-7-jdk

Centos Users: yum install java-1.7.0-openjdk-devel

* Java gets installed in /usr/lib/jvm/xxxxxxxx
* Check in xxxxxxx, if it shows different directories like bin, JRE, etc..
* Check in bin if you have various programs like java,javac,jps etc.
* Have openssh-server installed, we will use SSH to setup password less access to local machine nd other machines of cluster.

Ubuntu users: sudo apt-get install openssh-server

Centos users: yum install openssh-server

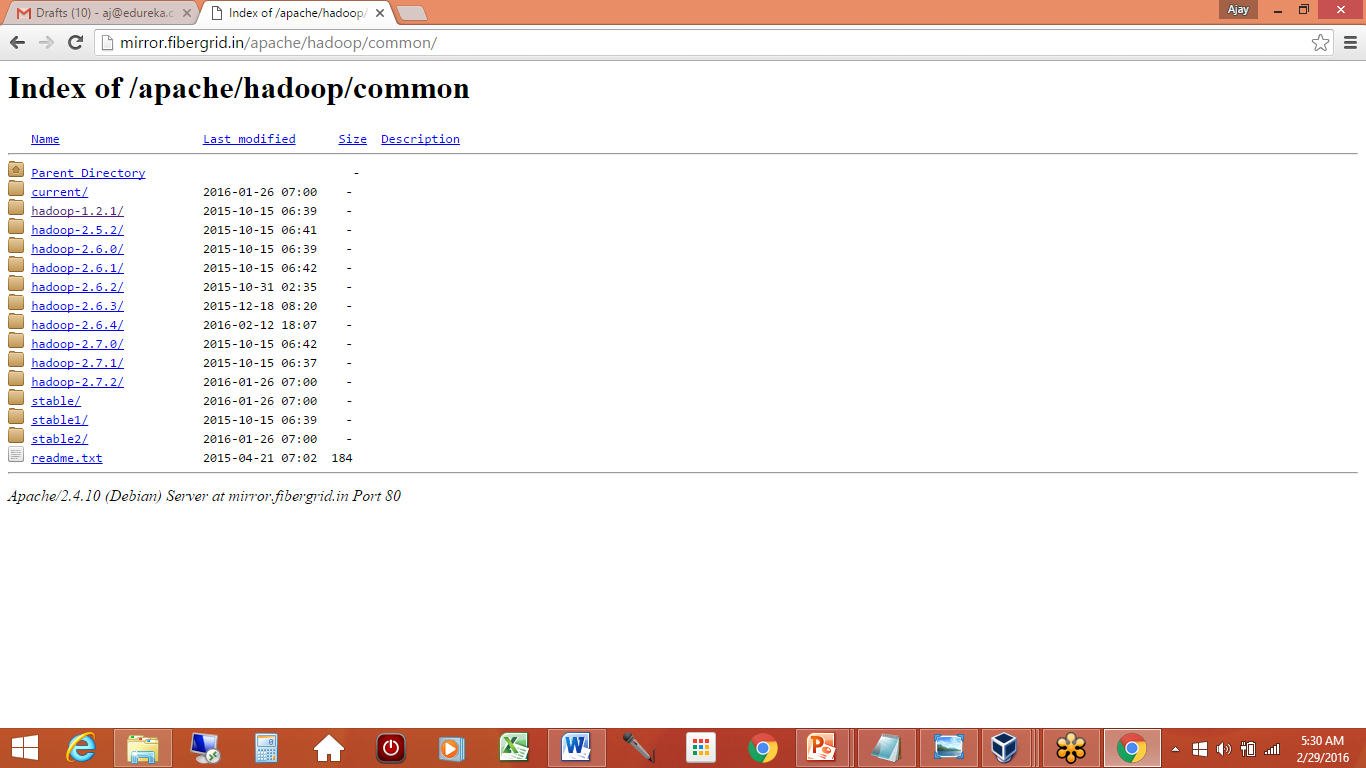
* Before setting Bridged adaptor as network settings, you can use NAT as adaptor and download Hadoop package from internet. ( Just search –Apache Hadoop download releases)

<http://hadoop.apache.org/releases.html>

Scroll Down and find

Download the release hadoop-X.Y.Z-src.tar.gz from a [mirror site](http://www.apache.org/dyn/closer.cgi/hadoop/common).

Click on mirror site, click on link and then you might see



Click on Hadoop 2.5.2 and download and save the tar file.

[hadoop-2.5.2.tar.gz](http://mirror.fibergrid.in/apache/hadoop/common/hadoop-2.5.2/hadoop-2.5.2.tar.gz)

**Pseudo mode means, all daemons will run on same machine.**

1. Do a SUDO IFCONFIG and check your IP address.
2. Make a note of IP address, broad cast address and mask address. (eth0---will show this)
3. See if local loopback ie (lo) shows 127.0.0.1
4. For Pseudo distributed we don’t need ip addresses as of now, we just need loopback address.
5. Update /etc/hosts ---

127.0.0.1 hostname ( you can give hostname command at $ and get your hostname).

Even if IFCONFIG doesn’t show ip address now, don’t worry, just proceed.

( Using these three ways you can make sure to acquire IP address for your machine.

The location of scripts/files might change depending on linux distribution

1. Sudo /etc/init.d/networking restart
2. Sudo ifdown eth0 and Sudo ifup eth0
3. Sudo invoke-rc.d networking stop and invoke-rc.d networking start)
4. You can also details in /etc/network/interfaces to see if you are using DHCP or static ipaddresses. Dynamic is fine as of now.
5. Now create a dedicated group and dedicated user which we will use to install Hadoop and make the admin for our HDFS cluster.
6. (Ubuntu commands are as follows).

Sudo Addgroup Hadoop

Sudo adduser hduser –ingroup Hadoop

Centos commands are as follows:

Groupadd Hadoop

Useradd –g Hadoop hduser

Passwd hduser

1. You would have downloaded Hadoop package by default in $HOME/Downloads.

Hadoop-1.2.1.tar.gz or Hadoop-2.5.2.tar.gz

Untar this tar file, in a location of your choice.Like I did in /usr/local.

This creates a directory Hadoop-1.2.1 in the location you have chosen.

This creates a directory Hadoop-2.5.2 in the location you have chosen.

1. Make hduser as owner of this directory

/usr/local$ Sudo chown –R hduser:Hadoop Hadoop-1.2.1

Or

/usr/local$ Sudo chown –R hduser:Hadoop Hadoop-2.5.2

There are two options now,

-You can rename/move this Hadoop-1.2.1 in a directory Hadoop

-You can create a link to this directory Hadoop-1.2.1

ln –s Hadoop-1.2.1 hadoop

ln –s Hadoop-2.5.2 hadoop

Creating a link is good as it will help us in future when we want to change Hadoop version.

1. Now for your user’ hduser’, update the .BASHRC file

This should have java and Hadoop path updated.

Export JAVA\_HOME=/usr/lib/jvm/xxxxxx

Export HADOOP\_HOME=Hadoop path ( this will have for ex:

/usr/local/**linkname**

Thus everytime you change Hadoop version, you need not update bashrc file. Just unlinking the link from its current directory and pointing to new directory will be enough.

1. Before we proceed any further, we need to make sure if SSH Access is enabled.
2. Issue ssh-keygen –t rsa –P “”

This generates SSH public and private keys,w hich get stored in /home/hduser/.ssh

For SSH to localmachine is hostname, copy the public key .

Cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys

1. Now test ssh and ping loopback address.

Ping 127.0.0.1

Ping hostname

SSH hostname

SSh 127.0.0.1

Once All this is done, we are ready with our initial setup to sart editing Hadoop config files in

/usr/local/Hadoop/conf.

Editing Hadoop config files.

1. Edit Hadoop-env.sh ( java and Hadoop path)
2. Edit core-site.xml ( where namenode runs)
3. Edit hdfs-site.xml ( replication factor as 1 and path where metadata and data stores)
4. Edit mapred-site.xml (where job tracker runs)
5. Edit masters ( this should have 127.0.0.1 which means SNN would run on same machine)
6. Edit slaves ( this should have 127.0.0.1 which means DN would run on same machine)

Once this is done…

At /usr/local/Hadoop$bin/Hadoop namenode –format

$bin/start-dfs.sh

$bin/start-mapred.sh

Type jps and Check if all daemons are started.

**Steps to follow when you want to setup a MULTI NODE /FULLY DISTRIBUTED OF HADOOP CLUSTER**

In case of fully distributed mode of Hadoop cluster, we need to follow steps as above ,but with some changes.

1>/etc/hosts file on all systems now

This file would have now, addresses of each machine with hostname

127.0.0.1 localhost

Ip1 hostname1

Ip2 hostname2…. And so on.

2>SSH key needs to be copied not only to local machines, but also across machines.

distributing ssh public keys to other machines

to add the xxxx@master’s public SSH key (which should be in $HOME/.ssh/id\_rsa.pub) to the authorized\_keys file of xxxx@slave

hduser$machine1:-**ssh-copy-id -i $HOME/.ssh/id\_rsa.pub hduser@machine2**

3. Test SSH and ping across machines.

4. Now /usr/local/Hadoop/conf

In your namenode machine..

Masters --- will have ip address of machine where you want SNN to run

Slaves – will have ip address of all machines where you want Datanodes to run.

5. edit other config files same as first machine.

Note\*\* in HDFS-SITE.XML

* Increase replicaton to more than 1, depending on number of machines you have setup.
* Dfs.data.dir shows a path where data will get stored on each machine.

After editing all the config files on all systems, you can

At /usr/local/Hadoop$bin/Hadoop namenode –format

$bin/start-dfs.sh

$bin/start-mapred.sh

Check if all daemons are started.

**Hadoop v1 conf**

**Hadoop v2 conf in /usr/local/Hadoop/etc/hadoop**

**Machine 1:**

**Hadoop-env.sh**

#The Java implementation to use

export JAVA\_HOME=/usr/lib/jvm/xxxx

**Core-site.xml**

<property>

<name>fs.defaultFS</name>

<value>hdfs://machine1:9000</value>

</property>

**Hdfs-site.xml**

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/abc/name</value>

</property>

<property>

<name>dfs.datanode.name.dir</name>

<value>file:/abc/name</value>

</property>

Note\*\* please create abc directory in root i.e. / and give ownership hduser

**Mapred-site.xml.template**

<property>

<name>mapred.framework.name</name>

<value>yarn</value>

</property>

**Yarn-site.xml**

<property>

<name>yarn.resourcemanager.address</name>

<value>machine1:9001</value>

</property>

**Slaves**

Machine1

Machine2

**Machine 2:**

**Hadoop-env.sh**

#The Java implementation to use

export JAVA\_HOME=/usr/lib/jvm/xxxx

**Core-site.xml**

<property>

<name>fs.defaultFS</name>

<value>hdfs://machine1:9000</value>

</property>

**Hdfs-site.xml**

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.datanode.name.dir</name>

<value>file:/abc/data2</value>

</property>

Note\*\* please create abc directory in root i.e. / and give ownership hduser

**Mapred-site.xml.template**

<property>

<name>mapred.framework.name</name>

<value>yarn</value>

</property>

**Yarn-site.xml**

<property>

<name>yarn.resourcemanager.address</name>

<value>machine1:9001</value>

</property>

**Slaves**

You can keep this empty

Once all the files are edited..

Go to your machine 1--/usr/local/hadoop

bin/Hadoop namenode –format

sbin/start-all.sh

type jps on each machine and check if you the daemons

Machine 1:

Namenode

Resourcemanager

Nodemanager

Datanode

Machine 2:

Datanode

Nodemanager

Your cluster should be up