(Single Node Cluster)

Update

 Update repositories– after installation of Ubuntu 15.0, update repositories by this command

sudo apt-get update

 Java installation–install java package

sudo apt-get install openjdk-7-jdk

2.Hadoop wget http://apache.mirrors.lucidnetworks.net/hadoop/common/stable2/hadoop- 2.7.1.tar.gz

 ssh installation –

• install ssh

$sudo apt-get install ssh

• Generate ssh keys

$ssh-keygen –t rsa –P “”

• Now just enable ssh access

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

• Test ssh access

$ssh localhost

After the installation of ssh,

• Download the tar ball file of Hadoop 2.7.1,it would be downloaded in the download folder.

• Copy the tar file in Home folder.

• Extract the contents of tar file, you will see the Hadoop-2.7.1 folder after extracting it.

Hadoop Installation steps– in Single Node cluster in Pseudo distributed mode

 Update “.bashrc” file

For locating .bashrc file, change to home directory ( $ cd)and list out .bashrc file by typing

command ( $ls –al .b\*) and Edit the file ( $ sudo gedit .bashrc)

.bashrc file will open up and set Hadoop-related environment variables as follows, at the end

of the .bashrc file and save the file–

export HADOOP\_HOME=$HOME/hadoop-2.7.1

export HADOOP\_CONF\_DIR=$HOME/hadoop-2.7.1/etc/hadoop

export HADOOP\_MAPRED\_HOME=$HOME/hadoop-2.7.1

export HADOOP\_COMMON\_HOME=$HOME/hadoop-2.7.1

export HADOOP\_HDFS\_HOME=$HOME/hadoop-2.7.1

export YARN\_HOME=$HOME/hadoop-2.7.1

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

export PATH=$PATH:$HOME/hadoop-2.7.1/bin

 Change dir to Hadoop-2.7.1 ($cd Hadoop-2.7.1), and list its contents ($ls). Bin, include, lib, etc

Directory and other directories would be listed.

 Change dir to etc ($cd etc), list its contents. Only 1 dir would be there named Hadoop

 Now, change dir to Hadoop directory ($cd Hadoop) and list its contents.It would list out various

configuration files.

Various configuration files of our interest are – Hadoop-env.sh, core-site.xml

 Configure JAVA\_HOME in ‘hadoop-env.sh’.

This file specifies environment variables that affect the JDK used by Apache Hadoop 2.7.1

daemons started by the Hadoop start-up scripts.

$sudo gedit hadoop-env.sh

This opens up hadoop-env.sh file.

Update the JAVA\_HOME to: (the JAVA\_HOME already exists, just comment it and update with

following path) and save the file.

export JAVA\_HOME=/usr/lib/jvm/java-7-openjdk-amd64

 Configuring core-site.xml

$sudo gedit core-site.xml

File opens up, configuration tag are already present , add property tag as follows-

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>

 Configuring hdfs-site.xml

Before configuring the file, we need to make 2 directories – Namenode and Datanode to store

HDFS data.

$ mkdir -p $HOME/hadoop2\_data/hdfs/namenode

$ mkdir -p $HOME/hadoop2\_data/hdfs/datanode

Open hdfs-site.xml by $sudo gedit hdfs-site.xml and add the following configurations.

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

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

<value>file:/home/lab140/hadoop-2.7.1/hadoop2\_data/hdfs/namenode</value>

</property>

<property>

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

<value>file:/home/lab140/hadoop-2.7.1/hadoop2\_data/hdfs/datanode</value>

</property>

</configuration>

 Configuring mapred-site.xml

We need to copy the mapred-site.xml template by following command since

mapred-site.xml is a blank file.

$cp mapred-site.xml.template mapred-site.xml

Now,open mapred-site.xml file by $sudo gedit mapred-site.xmland add following configurations.

<configuration>

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>

 Configuring yarn-site.xml

Open yarn-site.xml fileby $sudo gedit yarn-site.xmland add following configurations.

<configuration>

<!-- Site specific YARN configuration properties -->

<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>

</configuration>

 Start the DFS services:

The first step in starting up Hadoopinstallation is formatting the Hadoopfile-system, which is

implemented on top of the local file-systems of your cluster. This is required on the first time

Hadoopinstallation.Do not format a running Hadoopfile-system, this will cause all your

data to be erased.

To format the file-system, run the command:

$hadoop namenode –format

You are now all set to start the HDFS services i.e. Name Node, Resource Manager, Node

Manager and Data Nodes on your Apache HadoopCluster!

 Starting the Services:

Change the directory to Hadoop-2.7.1->sbin

And type following command.

$./hadoop-daemon.sh start datanode

Datanode should start after the command.We can check that by typing jps which shows

process id and daemons which are running.

$./hadoop-daemon.sh start namenode

Namenode

shouldshould start after the command.

 Starting the YARN Daemons:

Start the YARN Daemons i.e. Resource Manager and Node Manager. Cross check the service

start-up using JPS (Java Process Monitoring Tool).

In the same sbin folder, type the following commands on the terminal.

$./yarn-daemon.sh start resourcemanger

$./yarn-daemon.sh start nodemanger

 Check the NameNodestatus:

http://localhost: 50070/dfshealth.jsp

 Verify all applications for clusters

http://localhost: 8088

Multi Node Cluster

1. Configuring core-site.xml

$sudo gedit core-site.xml

File opens up, configuration tag are already present , add property tag as follows-

<configuration>

<property>

<name>fs.default.name</name>

<value>hdfs://hadoopmaster: 9000</value>

</property>

</configuration>

Open hdfs-site.xml by

$sudo gedit hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<property>

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

<value>file:/home/laksh/hadoop-2.7.1/hadoop2\_data/hdfs/datanode</value>

</property>

</configuration>

Configuring yarn-site.xml

Open yarn-site.xml file by

$sudo gedit yarn-site.xml and add following configurations.

<property>

<name>yarn.resourcemanager.resource-tracker.address</name>

<value>HadoopMaster:8025</value>

</property>

<property>

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

<value>HadoopMaster:8035</value>

</property>

<property>

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

<value>HadoopMaster:8050</value>

</property>

Update Mapred-site.xml

Update this file by updating and adding following properties,

$ sudo gedit mapred-site.xml

## Paste/Update these lines into <configuration> tag

<property>

<name>mapreduce.job.tracker</name>

<value>HadoopMaster:5431</value>

</property>

<property>

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

<value>yarn</value>

</property>

Update masters

Update the directory of master nodes of Hadoop cluster

## To edit file, fire the below given command

$ sudo gedit masters

## Add name of master nodes

HadoopMaster

Update slaves

Update the directory of slave nodes of Hadoop cluster

## To edit file, fire the below given command

$ sudo gedit slaves

## Add name of slave nodes

HadoopSlave1

HadoopSlave2

Copying/Sharing/Distributing Hadoop config files to rest all nodes – master/slaves

Use rsync for distributing configured Hadoop source among rest of nodes via network.

# In HadoopSlave1 machine

sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave1:/usr/local/hadoop/

# In HadoopSlave2 machine

sudo rsync -avxP /usr/local/hadoop/ hduser@HadoopSlave2:/usr/local/hadoop/