INSTALLATION OF HADOOP IN UBUNTU 14.04 LTS

MULTI NODE CONFIGURATION

Step 1. Follow the single node configuration.

Configuration done in the master and slave computers. (Assumption: Single Master and Single Slave.) There can be more than 1 slave.

Step 2. Note the IP address of master machine and slave machine

| \$ifconfig |
|------------|
|------------|

Step 3. In the /etc/hostname file of master add the name of the name-node system.

| sudo gedit /etc/hostname | |
|--------------------------|--|
| master | |

In the /etc/hostname file of slave add the name of the data-node system.

| sla | ave | | | |
|-----|-----|--|--|--|
| | | | | |

Step 4. In the /etc/hosts file add the name-node(ip-address, name) and data-nodes(ip-address, name). Name-node is the master and data-node is the slave.

| \$sudo gedit /etc/hosts | |
|-------------------------|--|
| 192.168.1.2 master | |
| 192.168.1.3 slave | |

Restart the system for changes to take place.

Step 5. Configuration of the hadoop files: core-site.xml, mapred-site.xml, hdfs-site.xml and yarn-site.xml

```
$cd $HADOOP_PREFIX/etc/hadoop
```

verify the path: /usr/local/hadoop-2.5.1/etc/hadoop

5.1. Configuration of the core-site.xml file

set the value of the property fs.default.name

5.2.Configuration of themapred-site.xml

```
$sudo gedit mapred-site.xml
```

set the value of the property mapred.job.tracker

```
<name>mapred.job.tracker</name>
<value>master:53212</value>
```

5.3. Configuration of thehdfs-site.xml

```
$sudo gedit hdfs-site.xml
```

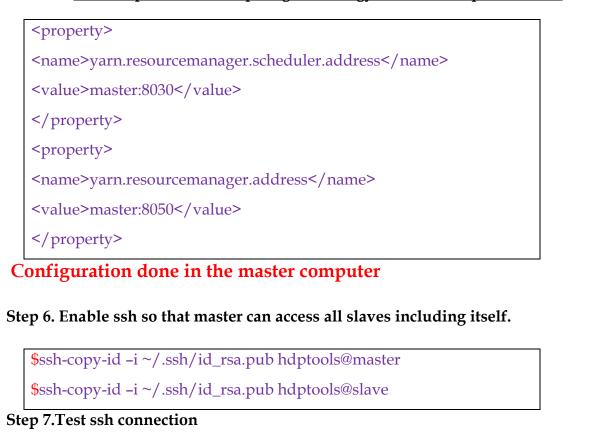
set the value of the properties dfs.replication and dfs.data.dir as

```
<property>
<name>dfs.replication</name>
<value>2</value>
</property>
<property>
<name>dfs.data.dir</name>
<value>/usr/local/hadoop/hdfs</value>
</property>
</property>
```

set the value of the properties yarn.nodemanager.auxservices,yarn.nodemanager.auxservices.

mapreduce_shuffle.class, yarn.resourcemanager.resource-tracker.address, yarn.resourcemanager.scheduler.address,yarn.resourcemanager.address

5.4. Configuration of theyarn-site.xml



\$ssh master

After successful ssh connection.

\$exit

Check Slave connection

\$ssh slave \$exit

Step 8. In \$HADOOP_PREFIX/etc/hadoop/masters, set the name-node(master) name.

\$sudo gedit masters

master

| 7.3. In \$HADOOP_PREFIX/etc/hado | oop/slaves, set the data-node(slave) names.(master |
|---------------------------------------|--|
| can act also as one of the data-node) | |

| \$sudo gedit slaves | |
|---------------------|--|
| | |
| master | |
| slave | |

8. Format the Hadoop File system implemented on top of the local file system using

```
$cd bin
```

Verify the path: /usr/local/hadoop-2.5.1/bin

```
$hadoopnamenode –format
$cd ..
```

9. In the master, Start Hadoop using

```
$cd sbin
```

 $Verify\ the\ path: /usr/local/hadoop-2.5.1/sbin$

```
$./start-all.sh
$jps
```

10. Accessing Hadoop on Browser

The default port number to access Hadoop is 50070. Use the following url to get Hadoop services on browser.

http://localhost:50070/

11. Verify All Applications for Cluster

The default port number to access all applications of cluster is 8088. Use the following url to visit this service.

http://localhost:8088/

COMPILATION AND EXECUTION OF MAP REDUCE PROGRAM

1. Write a Map Reduce program and save it in the \$HOME Path.

Eg.:WordCount.java

2. Create a empty directory "WordCount" in \$HOME path. The following command is to create a directory to store the compiled java classes.

cd \$HOME

mkdir WordCount

3. Download Hadoop-core-1.2.1.jar, which is used to compile and execute the Map Reduce program.

Visit the following link:

http://mvnrepository.com/artifact/org.apache.hadoop/hadoop-core/1.2.1

to download the jar. Let us assume the downloaded folder is

/usr/local/hadoop-2.5.1/

4. Compile the java program by specifying the class path:

```
javac -classpath /usr/local/hadoop-2.5.1/hadoop-core-
1.2.1.jar -d WordCount WordCount.java
```

5. Create jar files for the program

```
jar -cvf WordCount.jar -C WordCount/
```

- **6.** Create an input file "fsample.txt" contains some text in \$HOME path.
- 7. Change directory to Hadoop

```
cd $HADOOP PREFIX
```

8. The following command is used to create an input directory in HDFS.

bin/hdfs dfs -mkdir input_dir

9. The following command is used to copy the input file named **fsample.txt** in the input directory of HDFS.

```
bin/hdfs dfs -put $HOME/fsample.txt /input_dir
```

10. The following command is used to verify the files in the input directory.

```
bin/hdfs dfs -ls input_dir/
```

11. The following command is used to run the application by taking the input files from the input directory.

```
bin/hadoop jar $HOME/WordCount.jar hadoop.WordCount /input dir /output dir
```

12. The following command is used to verify the resultant files in the output folder.

```
bin/hdfs dfs -ls output dir/
```

13. The following command is used to see the output in **Part-00000** file. This file is generated by HDFS.

14. Copy the output files from the distributed file system to the local file system and examine them:

```
/usr/local/hadoop-2.5.1$ bin/hdfs dfs -get output/*
/home/huser/Downloads/output/
```

/usr/local/hadoop-2.5.1\$ cat output/*