Installation of Single Node Hadoop and Executing Word Count Program

1. Update all the softwares in Ubuntu

```
]$ sudo apt-get update
```

1. Create new user in system.

```
]$ sudo adduser hduser
```

2. set the correct password. Adding hduser to sudoers list.

```
]$ sudo usermod -aG sudo hduser
```

Verify User Belongs to Sudo Group

```
]$ groups hduser
```

Logout and Login as hduser

3. Install Java 8 and verify that it is working.

```
]$ java -version
]$ sudo apt-get install default-jdk
]$ sudo apt-get install default-jre
```

OR

Manual installation of Java suing JDK 1.8

Download the JDK1.8 tar.gz file from the following URL
http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

Download: Linux x64 Compressed Archive (jdk-8u261-linux-x64.tar.gz)

If you download software in your physical machine, use shared folder option provided by Virtual box guest additions and share the downloaded software folder to your virtual machine. Downloaded software will be available in the VM now.

Untar the file using the following command,

```
]$ sudo tar xfz jdk-8u261-linux-x64.tar.gz
```

Move the java to /usr/local/java

```
]$ sudo mv jdk1.8.0_45 /usr/local/java
```

Please use the correction version in the above command

Set the Path and ClassPath Variables in ~/.bashrc

```
]$ sudo gedit ~/.bashrc
 OR
 ]$ sudo nano ~/.bashrc
```

Add the following line in the editor, save the file and exit.

```
 # JAVA PATH VARIABLES
export JAVA_HOME=/usr/local/java
 export PATH=$JAVA_HOME/bin:{$PATH}
 export CLASSPATH=$JAVA_HOME/lib
 # END OF JAVA PATH VARIABLES
```

Write: Ctrl + O, give enter to write to the same file and Ctrl+X for **exit**

Now, save the file and Exit.

```
]$ . ~/.bashrc
OR
]$ source ~/.bashrc
```

4. Verify JAVA is Installed or not using this Command

```
]$ java -version
```

Make sure that you can see the JAVA version that you have installed.

5. Installing Secure Shell

```
]$ sudo apt-get install ssh
]$ sudo apt-get install rsync
]$ sudo apt-get install openssh-server
]$ sudo apt-get install openssh-client
```

Generate key pair and add the public key to the authorized_keys.

```
]$ ssh-keygen -t rsa -P '' -f ~/.ssh/id_rsa
]$ sudo cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
```

6. Verify SSH,

```
]$ ssh localhost
```

Make sure that you are able to connect localhost through ssh without a password.

7. Download latest version of Apache Hadoop package
   https://hadoop.apache.org/releases.html
   https://hadoop.apache.org/old/releases.html
   Download Binary package
   https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.1.4/hadoop-3.1.4.tar.gz

   If you download software in your physical machine, use shared folder option provided by Virtual box guest additions and share the downloaded software folder to your virtual machine. Downloaded software will be available in the VM now.

Installing Hadoop 3.1.4

Extract and move the hadoop to an installation directory

```
]$ sudo tar xfz hadoop-3.1.4.tar.gz

]$  sudo mv hadoop-3.1.4 /usr/local/hadoop
```

(Installed Directory)

Add the JAVA_HOME to hadoop-env.sh file

```
]$ sudo gedit /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

OR

```
]$ sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh
```

Locate java_home and set

```
JAVA_HOME=/usr/local/java
```

Now, save and Exit the file. **Write:** Ctrl + O, give enter to write to the same file and Ctrl+X for **exit**

Add the following lines to ~/.bashrc

```
]$  sudo gedit ~/.bashrc
```

OR

```
]$ sudo nano ~/.bashrc
```

```
#HADOOP VARIABLES START

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
 export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_INSTALL/lib/native
 export HADOOP_OPTS="-Djava.library.path=$HADOOP_INSTALL/lib"
 export HADOOP_CLASSPATH=$JAVA_HOME/lib/tools.jar
 #HADOOP VARIABLES END
```

OR

```
#HADOOP Variables
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$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 PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin export
HADOOP_CLASSPATH=$
{JAVA_HOME}/lib/tools.jar
```

Write: Ctrl + O, give enter to write to the same file and Ctrl+X for **exit**

Refresh the bashrc file so that our environment variables can be accessed.

]$. ~/.bashrc

OR

]$ **source ~/.bashrc**

Check the hadoop version

hadoop version

- Add the following <property> tag to core-site.xml.
- Add the below property tag inside <configuration> </configuration> tag.

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

OR

```
]$ sudo nano /usr/local/hadoop/etc/hadoop/core-site.xml
```

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

Now, save the file and Exit. Ctrl +O, enter and Ctrl+X

- Add the following <property> tags to yarn-site.xml
- Add the below property tag inside <configuration> </configuration> tag.

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

OR

```
or ]$ sudo nano /usr/local/hadoop/etc/hadoop/yarn-site.xml
```

```
<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>
```

Now, save the file and Exit. Ctrl +O, enter and Ctrl+X

- Add the following <property> tag to core-site.xml
- Add the below property tag inside <configuration> </configuration> tag.

- Copy the MapRed-site.xml.template file to MapRed-site.xml
-]$ **cp /usr/local/hadoop/etc/hadoop/mapred-site.xml.template usr/local/hadoop/etc/hadoop/mapred-site.xml**

  ]$ **sudo gedit /usr/local/hadoop/etc/hadoop/mapred-site.xml**

  OR

  ]$ **sudo nano /usr/local/hadoop/etc/hadoop/mapred-site.xml**

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

Now, save the file and Exit. Ctrl +O, enter and Ctrl+X

- Create Namenode and Datanode directories
-]$ sudo **mkdir -p /usr/local/hadoop_store/hdfs**
- Add the following <property> tags to hdfs-site.xml
- Add the below property tag inside <configuration> </configuration> tag.

  ]$ **gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

  OR

  ]$ **sudo nano gedit /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

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

```
        <name>dfs.datanode.data.dir</name>
        <value>file:/usr/local/hadoop_store/hdfs/datanode</value>
</property>
```

Now, save the file and Exit. Ctrl +O, enter and Ctrl+X

- Change owner of hadoop_store

```
]$ sudo chown hduser:hduser -R /usr/local/hadoop
]$ sudo chown hduser:hduser -R /usr/local/hadoop_store
```

also give the folder the full permission

```
]$  sudo chmod -R 777 /usr/local/hadoop
]$  sudo chmod -R 777 /usr/local/hadoop_store
```

- Format your HDFS, make sure you have logged in as hadoop/hduser user.

```
]$ hdfs namenode -format
```

- Start/Stop the Hadoop Cluster.

```
]$ start-all.sh or stop-all.sh
```

- Access the User Inerfaces
- ResourceManager @- http://localhost:8088/
- NameNode @- http://localhost:50070/

5. Start the hadoop dfs and yarn process

```
]$ start-dfs.sh start-yarn.sh
```

6. Check the hadoop deamons by running the following command

```
]$ jps
// Output
<PID> ResourceManager
<PID> SecondaryNameNode
<PID> DataNode
<PID> NameNode
<PID> NodeManager
```

<PID> Jps

this will show the list of Hadoop components which are active and running and also the java process (jps) running.

7. Access the Hadoop DFS and Hadoop YARN Web UI,

NameNode UI

http://localhost:9870/

YARN UI

http://localhost:8042/

##Compiling and Running the Hadoop Programs

Compiling the hadoop program,

```
]$ hadoop com.sun.tools.javac.Main MainClass.java
```

Running the hadoop program

```
]$ hadoop jar <<jar file>> <<MainClass>> <<Command Line Arguments List>>
```

Basic Hadoop Commands

1. List the hadoop directories

```
]$ hadoop fs ls /
```

2. Create a new directory in hadoop

```
]$ hadoop fs mkdir <<dirname>>
```

3. Copy files to HDFS

```
]$ hadoop fs put <<file>> <<hdfsdirectory>>
```

4. Copy files from HDFS

```
]$ hadoop fs get <<filepathinhdfs>> <<localdirectory>>
```

I. Managing files in HDFS. Follow below link.

Create an input file with few sentences in it. Upload input file into HDFS using put command / copyFromLocal command
put *command to store file in HDFS,* ***get*** *command to read / retrieve file from HDFS.*

```
hduser ]$ /usr/local/hadoop/bin/hadoop dfs –copyFromLocal /tmp/MapReduceInput /user/hduser/MapReduceInput

hduser ]$ /usr/local/hadoop/bin/hadoop dfs –ls /tmp/MapReduceInput /user/hduser/MapReduceInput
```

https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

II. Write word count program in Java using Map and Reduce functions. Create word count program into a jar file using Eclipse.
Follow link in reference section. Either download wordcount.jar file and execute it or use Eclipse IDE to export java program into a .jar file.

https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v2.0

http://hortonworks.com/hadoop-tutorial/using-commandline-manage-files-hdfs/

III. Execute the word count program’s jar file using below command in hduser.

```
hduser]$ ls  /usr/local/hadoop/
```

wordcount.jar file will be shared to you if necessary.

```
hduser ]$ /usr/local/hadoop/bin/hadoop jar /usr/local/Hadoop/Hadoop-examples-2.7.1.jar wordcount /user/hduser/MapReduceInput /user/hduser/MapReduce.output
```

OR

```
hadoop jar wordcount.jar   /usr/local/hadoop/input   /usr/local/hadoop/output

hduser ]$ /usr/local/hadoop/bin/hadoop dfs –ls  /user/hduser
hduser ]$ /usr/local/hadoop/bin/hadoop dfs –ls /user/hduser/MapReduce.output
hduser ]$ /usr/local/hadoop/bin/hadoop dfs –cat /user/hduser/MapReduce.output/part-r-00000
```

Finally stop all Hadoop services.
]$ stop-all.sh
]$ stop-dfs.sh
]$ stop-yarn.sh

Exit from localhost using **]$exit** command