HADOOP 2.7.3 INSTALLING ON UBUNTU 16.04 (SINGLE-NODE CLUSTER)

Installing Java

sudo apt-get update

sudo apt-get install default-jdk

java -version

Adding a dedicated Hadoop user

sudo addgroup Hadoop

sudo adduser --ingroup hadoop hduser

We can check if we create the **hadoop** group and **hduser** user:

groups hduser

Installing SSH

**ssh** has two main components:

1. **ssh** : The command we use to connect to remote machines - the client.
2. **sshd** : The daemon that is running on the server and allows clients to connect to the server.

The **ssh** is pre-enabled on Linux, but in order to start **sshd** daemon, we need to install **ssh**first. Use this command to do that :

sudo apt-get install ssh

This will install ssh on our machine. If we get something similar to the following, we can think it is setup properly:

which ssh

which sshd

Create and Setup SSH Certificates

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus our local machine. For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost.

So, we need to have SSH up and running on our machine and configured it to allow SSH public key authentication.

Hadoop uses SSH (to access its nodes) which would normally require the user to enter a password. However, this requirement can be eliminated by creating and setting up SSH certificates using the following commands. If asked for a filename just leave it blank and press the enter key to continue.

su hduser

ssh-keygen

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

The second command adds the newly created key to the list of authorized keys so that Hadoop can use ssh without prompting for a password.

We can check if ssh works:

ssh localhost

Install Hadoop

wget http://mirrors.sonic.net/apache/hadoop/common/hadoop-2.6.5/hadoop-2.6.5.tar.gz

tar xvzf hadoop-2.6.5.tar.gz

We want to move the Hadoop installation to the **/usr/local/hadoop** directory. So, we should create the directory first:

sudo mkdir -p /usr/local/hadoop

We can check again if **hduser** is not in **sudo** group:

sudo -v

This can be resolved by logging in as a root user, and then add **hduser** to **sudo** group:

su k

sudo adduser hduser sudo

Now, the **hduser** has root priviledge, we can move the Hadoop installation to the **/usr/local/hadoop** directory without any problem:

sudo su hduser

sudo mv \* /usr/local/hadoop

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

Setup Configuration Files

The following files should to be modified to complete the Hadoop setup:

1. ~/.bashrc
2. /usr/local/hadoop/etc/hadoop/hadoop-env.sh
3. /usr/local/hadoop/etc/hadoop/core-site.xml
4. /usr/local/hadoop/etc/hadoop/mapred-site.xml.template
5. /usr/local/hadoop/etc/hadoop/hdfs-site.xml

**1. ~/.bashrc**:

Before editing the **.bashrc** file in **hduser**'s home directory, we need to find the path where Java has been installed to set the **JAVA\_HOME** environment variable using the following command:

update-alternatives --config java

Now we can append the following to the end of **~/.bashrc**:

vi ~/.bashrc

#HADOOP VARIABLES START

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

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"

#HADOOP VARIABLES END

source ~/.bashrc

Note that the JAVA\_HOME should be set as the path just before the '.../bin/':

javac –version

which javac

readlink -f /usr/bin/javac

**2. /usr/local/hadoop/etc/hadoop/hadoop-env.sh**

We need to set **JAVA\_HOME** by modifying **hadoop-env.sh** file.

vi /usr/local/hadoop/etc/hadoop/hadoop-env.sh

export JAVA\_HOME=/usr/lib/jvm/java-8-openjdk-amd64

Adding the above statement in the **hadoop-env.sh** file ensures that the value of JAVA\_HOME variable will be available to Hadoop whenever it is started up.

**3. /usr/local/hadoop/etc/hadoop/core-site.xml**:

The **/usr/local/hadoop/etc/hadoop/core-site.xml** file contains configuration properties that Hadoop uses when starting up.   
This file can be used to override the default settings that Hadoop starts with.

sudo mkdir -p /app/hadoop/tmp

sudo chown hduser:hadoop /app/hadoop/tmp

Open the file and enter the following in between the <configuration></configuration> tag:

vi /usr/local/hadoop/etc/hadoop/core-site.xml

<configuration>

<property>

<name>hadoop.tmp.dir</name>

<value>/app/hadoop/tmp</value>

<description>A base for other temporary directories.</description>

</property>

<property>

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

<value>hdfs://localhost:54310</value>

<description>The name of the default file system. A URI whose

scheme and authority determine the FileSystem implementation. The

uri's scheme determines the config property (fs.SCHEME.impl) naming

the FileSystem implementation class. The uri's authority is used to

determine the host, port, etc. for a filesystem.</description>

</property>

</configuration>

**4. /usr/local/hadoop/etc/hadoop/mapred-site.xml**

By default, the **/usr/local/hadoop/etc/hadoop/** folder contains   
**/usr/local/hadoop/etc/hadoop/mapred-site.xml.template**   
file which has to be renamed/copied with the name **mapred-site.xml**:

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

The **/usr/local/hadoop/etc/hadoop/mapred-site.xml** file is used to specify which framework is being used for **MapReduce**.  
We need to enter the following content in between the <configuration></configuration> tag:

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>localhost:54311</value>

<description>The host and port that the MapReduce job tracker runs

at. If "local", then jobs are run in-process as a single map

and reduce task.

</description>

</property>

</configuration>

**5. /usr/local/hadoop/etc/hadoop/hdfs-site.xml**

The **/usr/local/hadoop/etc/hadoop/hdfs-site.xml** file needs to be configured for each host in the cluster that is being used.   
It specifies the directories which will be used as the **namenode** and the **datanode** on that host.

Before editing this file, we need to create two directories which will contain the **namenode**and the **datanode** for this Hadoop installation.

This can be done using the following commands:

sudo mkdir -p /usr/local/hadoop\_store/hdfs/namenode

sudo mkdir -p /usr/local/hadoop\_store/hdfs/datanode

sudo chown -R hduser:hadoop /usr/local/hadoop\_store

Open the file and enter the following content in between the <configuration></configuration> tag:

vi /usr/local/hadoop/etc/hadoop/hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

<description>Default block replication.

The actual number of replications can be specified when the file is created.

The default is used if replication is not specified in create time.

</description>

</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>

</configuration>

Format the New Hadoop Filesystem

Now, the Hadoop file system needs to be formatted so that we can start to use it. The **format**command should be issued with write permission since it creates **current** directory   
under **/usr/local/hadoop\_store/hdfs/namenode** folder:

hadoop namenode -format

Note that **hadoop namenode -format** command should be executed once before we start using Hadoop.   
If this command is executed again after Hadoop has been used, it'll destroy all the data on the Hadoop file system.

Starting Hadoop

Now it's time to start the newly installed single node cluster.   
We can use **start-all.sh** or (**start-dfs.sh** and **start-yarn.sh**)

cd /usr/local/hadoop/sbin

ls

sudo su hduser

Start NameNode daemon and DataNode daemon:

start-dfs.sh

Browse the web interface for the NameNode; by default it is available at:

NameNode - http://localhost:50070/

Start ResourceManager daemon and NodeManager daemon:

start-yarn.sh

We can check if it's really up and running:

hduser@laptop:/usr/local/hadoop/sbin$ jps

Stopping Hadoop

In order to stop all the daemons running on our machine, we can run **stop-all.sh** or (**stop-dfs.sh** and **stop-yarn.sh**) :

Hadoop Web Interfaces

Let's start the Hadoop again and see its Web UI:

start-dfs.sh

start-yarn.sh

Type **http://localhost:50070/** into our browser, then we'll see the web UI of the NameNode daemon:

Type in **http://localhost:50090/status.jsp** as url, we get **SecondaryNameNode**:

The default port number to access all the applications of cluster is 8088. Use the following url to visit **Resource Manager**:

http://localhost:8088/

We may need the following configurations set properly.

**/usr/local/hadoop/etc/hadoop/yarn-site.xml**:

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

</configuration>

**/usr/local/hadoop/etc/hadoop/mapred-site.xml**:

<configuration>

<property>

<name>mapred.job.tracker</name>

<value>localhost:54311</value>

</property>

<property>

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

<value>yarn</value>

</property>

</configuration>