Install Ubuntu 18.04

Download ISO image from following link and install on your (Virtual) machine:

<https://ubuntu.com/download/desktop>

use machine name mpstme, username: mpstme, password: mpstme

(if you do not know how to install contact me)

Download Java 8 from: (I guess 8.221)

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Download Hadoop 3.2.1 from:

<https://www.apache.org/dyn/closer.cgi/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz>

<http://apachemirror.wuchna.com/hadoop/common/hadoop-3.1.2/hadoop-3.1.2.tar.gz>

go to Download directory and cp \*.gz to /usr/local (in your Ubuntu virtual machine)

1. Login as Root:

**sudo su**

**whoami**  -- should give root

1. Adding a dedicated Hadoop system user called “hduser”

We will use a dedicated Hadoop user account for running Hadoop. While that’s not required it is recommended because it helps to separate the Hadoop installation from other software applications and user accounts running on the same machines (this: security, permission and backups etc.)

1. Create a group called hadoop

**sudo addgroup hadoop**

1. Create User “hduser”

**sudo adduser hduser**

It might ask to enter password 2 times followed by some information, just press enter and Yes. We have given password ha

1. Add hduser to hadoop group

**sudo adduser hduser hadoop**

**One line command for 4 and 5: sudo adduser –ingroup hadoop hduser**

1. Add the “hduser” to sudoers list s
2. o that hduser can do admitasks

**sudo visudo**

Add a line under ##Allow member of group sudo to execute any command anywhere in the format.

**hduser ALL=(ALL:ALL) ALL**

Control+X yes and enter to save file

Logout and login as hduser

INSTALL JAVA:

You are ~/Downloads directory

Step 2: sudo cp jdk-8u221-linux-x64.tar.gz /usr/local

(coping tar file from Downloads directory to /usr/local/.)

cd /usr/local

Step5:- sudo tar -xvzf jdk-8u221-linux-x64.tar.gz

Step6:- sudo rm jdk-8u221-linux-x64.tar.gz

Step7:- sudo ln -s jdk1.8.0\_221 java (rm -r java if symbolic link error)

sudo chmod 777 jdk1.8.0\_221

sudo chmod 777 java

Step8:- sudo update-alternatives --install ''/usr/bin/java" "java" "/usr/local/java/bin/java" 1 (please mind two hyphens – and also double quotes “”)

Step9:- sudo update-alternatives --install ''/usr/bin/javac" "javac" "/usr/local/java/bin/javac" 1

Step10:- sudo update-alternatives --install ''/usr/bin/javaws" "javaws" "/usr/local/java/bin/javaws" 1

Step11:-

sudo update-alternatives –set java /usr/local/java/bin/java

sudo update-alternatives –set javac /usr/local/java/bin/javac

sudo update-alternatives –set javaws /usr/local/java/bin/javaws

add the following into your ~/.bashrc file

export JAVA\_HOME=/usr/local/java

export JRE\_HOME=$JAVA\_HOME/jre

export PATH=$PATH:$JAVA\_HOME/bin:$JRE\_HOME/bin

$source ~/.bashrc

$java -version

It should give java 1.8.0\_221

Configure SSH

Hadoop require SSH access to manage its nodes, i.e. remote machines plus your local machine if you want to use Hadoop on it (which is what we want to do in this class). For our single node setup of Hadoop, we therefore need to configure SSH access to localhost for the hduse user we created in the previous section.

1. ##install ssh server on your machine

**sudo apt-get install openssh-server**

If this did not work, then install openssh-server using Ubuntu software center by searching for openssh-server

1. Generate SSH key for communication

**ssh-keygen**

Just press enter for whatever is asked.

Generate public/private rsa key pair

Enter file to save the key (/home/hduser/.ssh/id\_rsa):

Created directory ‘/home/hduser/.ssh’

Your identification has been saved in /home/hduser/.ssh/id\_rsa

Your public key has been saved in /home/hduser/.ssh/id\_rsa.pub

The key fingerprint is SHA256:yN0RoEI/gcLxVr99FGJqzINWBD0J1MJnjs++NKaIjcU

The key’s randomart image is:

+---[RSA 2048]----+

| ....+\*\*+o+ . |

| oo.ooOBo o . |

| .+ \*\*B.. . |

| . +.=.= o |

| ooS o . |

| . o . |

| E .+ |

| = . +.. |

| o o . .. |

+----[SHA256]------+

OR some thing like that

1. Copy Public Key to Authorized\_key file & edit the permission

#now copy the public key to the authorized\_keys file, so that ssh shpuld not require password every time

**cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys**

Change permission of the authorized\_keys file to have all permission for hduser

**chmod 700 ~/.ssh/authorized\_keys**

1. Start SSH

If ssh is not running, then run it by giving the below command

**sudo /etc/init.d/ssh restart**

1. Test your SSH connectivity

**ssh localhost**

Type yes, when asked for. You should be able to connect without password. If you are asked to enter password here, then something went wrong. Please check you previous steps.

1. Disable IPV6

Hadoop and IPV6 do not agree on the meaning of 0.0.0.0 address, thus it is advisable to disable IPV6 adding following lines at the end of /etc/sysctl.conf

**sudo nano /etc/sysctl.conf**

#disable ipv6

net.ipv6.conf.all.disable\_ipv6 = 1

net.ipv6.conf.default.disable\_ipv6 = 1

net.ipv6.conf.io.disable\_ipv6 = 1

1. Check if IPV6 is disable or not.

**cat /proc/sys/net/ipv6/conf/all/disable\_ipv6**

it should show you 0

after reboot it should show you 1

INSTALL HADOOP

1. Download Hadoop

Let us hadoop 2.7.3 (hadoop-2.7.3.tar.gz). Download hadoop-2.7.3.tar.gz from apache site and save it in hduser/Desktop

1. Move the zip file to /usr/local/

**sudo mv ~/Desktop/hadoop-2.7.3.tar.gz /usr/local/**

**cd /usr/local**

**sudo tar –xvzf hadoop-2.7.3.tar.gz**

**sudo rm hadoop-2.7.3.tar.gz**

**sudo ln –s hadoop-2.7.3 hadoop**

**sudo chown –R hduser:hadoop hadoop-2.7.3**

**sudo chown –R hduser:hadoop hadoop**

**sudo chmod 777 hadoop-2.7.3**

1. Edit hadoop-env.sh and configure Java

Add the following to /usr/local/hadoop/etc/hadoop/hadoop-env.sh by removing

Export JAVA\_HOME=${JAVA\_HOME}

sudo nano /usr/local/hadoop/etc/hadoop/hadoop-env.sh

export HADOOP\_OPTS=-Djava.net.preferIPv4Stack=true

export HADOOP\_HOME\_WARN\_SUPPRESS=”TRUE”

export JAVA\_HOME=/usr/local/java

1. Update ~/.bashrc

#Set Hadoop-related environment 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

$source ~/.bashrc

Test Hadoop installation

$ hadoop version

Hadoop latest version should shown

1. Create a temporary directory which will be used as base location for DFS

sudo mkdir –p /app/hadoop/tmp

sudo chown –R hduser:hadoop /app/hadoop/tmp

sudo chmod –R 777 /app/hadoop/tmp

1. Update yarn-site.xml

**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>**

1. Update core-site.xml

**nano /usr/local/hadoop/etc/hadoop/core-site.xml**

**<property>**

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

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

**<description>A base for other temporary directory.</description>**

**</property>**

**<property>**

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

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

**<description>The name of the default file system.**

**A URI whose scheme and authority determine the FileSystem implementation. The uri’s scheme determine 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>**

1. Create mapred-site.xml from mapred-site.xml.template

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

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

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

**<property>**

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

**<value>yarn</value>**

**</property>**

1. **Create some directory**

**sudo mkdir –p /usr/local/hadoop/yarn\_data/hdfs/namenode**

**sudo mkdir –p /usr/local/hadoop/yarn\_data/hdfs/datanode**

**sudo chmod 777 /usr/local/hadoop/yarn\_data/hdfs/namenode**

**sudo chmod 777 /usr/local/hadoop/yarn\_data/hdfs/datanode**

**sudo chown –R hduser:hadoop /usr/local/hadoop/yarn\_data/hdfs/namenode**

**sudo chown –R hduser:hadoop /usr/local/hadoopd/yarn\_data/hdfs/datanode**

1. Update hdfs-site.html

**nano /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/yarn\_data/hdfs/namenode </value>**

**</property>**

**<property>**

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

**<value>file:/usr/local/hadoop/yarn\_data/hdfs/datanode </value>**

**</property>**

1. Format your namenode

**hadoop namenode –format**

1. Start your single node cluster

start-dfs.sh

start-yarn.sh

jps all jobs like resource manager secondary namenode, datanode

**// NameNode:** [**http://localhost:50070**](http://localhost:50070)

**Resource Manager** [**http://localhost:8088**](http://localhost:8088)

**// MapReduceJobHostory Service** [**http://localhost:19888**](http://localhost:19888)