1.sudo apt update

2.To install python

sudo apt install python3 python3-pip -y

python3 –version

3.To install Java

sudo apt install openjdk-8-jdk -y

To check java installed path :

readlink -f $(which java)

the above should return something like:

/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java

Copy that and use in bashrc editing

4.To download Hadoop

sudo wget https://downloads.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz

extract

sudo tar -xvzf hadoop-3.3.6.tar.gz

rename the folder

sudo mv hadoop-3.3.6 hadoop

change the ownership

sudo chown -R $USER:$USER hadoop

5. Edit your ~/.bashrc

nano ~/.bashrc

Hadoop Environment Variables

export HADOOP_HOME=$HOME/hadoop

export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin

export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64

export PATH=$PATH:$JAVA_HOME/bin

to apply changes

source ~/.bashrc

to check

echo $HADOOP_HOME

You should see:

/home/your-username/Hadoop

echo $JAVA_HOME

Should show:

/usr/lib/jvm/java-8-openjdk-amd64

Final check

hadoop version

java -version

6. SSH server

sudo apt install openssh-server -y

Then check SSH status:

sudo systemctl status ssh

If it’s not active, start it:

sudo systemctl start ssh

And enable it at boot:

sudo systemctl enable ssh

7.Set up passwordless SSH

just enter for all do not type anything for the cmd

ssh-keygen -t rsa

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

chmod 600 ~/.ssh/authorized_keys

Test it:

ssh localhost

It should log in without asking for a password.
Type exit to return.

8.configuration

nano $HADOOP_HOME/etc/hadoop/core-site.xml

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

nano $HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
<configuration>
 <property>
  <name>dfs.replication</name>
  <value>1</value>
 </property>
 <property>
  <name>dfs.namenode.name.dir</name>
  <value>file:///home/your-username/hadoopdata/hdfs/namenode</value>
 </property>
 <property>
  <name>dfs.datanode.data.dir</name>
  <value>file:///home/your-username/hadoopdata/hdfs/datanode</value>
 </property>
</configuration>
```

For the above change the username

nano $HADOOP_HOME/etc/hadoop/yarn-site.xml

```
<configuration>
```

```
  <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>
```

nano $HADOOP_HOME/etc/hadoop/mapred-site.xml

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

  <property>
    <name>yarn.app.mapreduce.am.env</name>
    <value>HADOOP_MAPRED_HOME=/home/username/hadoop</value>
  </property>

  <property>
    <name>mapreduce.map.env</name>
    <value>HADOOP_MAPRED_HOME=/home/username/hadoop</value>
  </property>

  <property>
    <name>mapreduce.reduce.env</name>
```

<value>HADOOP_MAPRED_HOME=/home/username/hadoop</value>

</property>

</configuration>

Replace the username

9.Create HDFS directories

mkdir -p ~/hadoopdata/hdfs/namenode

mkdir -p ~/hadoopdata/hdfs/datanode

sudo chown -R $USER:$USER ~/hadoopdata

10.Start Hadoop Services

hdfs namenode -format

start-dfs.sh

start-yarn.sh

You should now see the NameNode, DataNode, ResourceManager, and NodeManager start successfully.

11.Final verification

jps

You should see something like:

NameNode

DataNode

SecondaryNameNode

ResourceManager

NodeManager

Finally Hadoop done

12.Program

hdfs dfs -ls /

It should show an empty root directory

hdfs dfs -mkdir /user

hdfs dfs -mkdir /user/username

Replace username with the actual name

echo "Hello Hadoop!" > test.txt

hdfs dfs -put test.txt /user/username/

Replace username with the actual name

You should see:

Hello Hadoop!

13.Map-reduce program

hdfs dfs -mkdir -p /input

Create a test file locally:

echo -e "Hello Hadoop\nHadoop is fun\nHello World\nHello Hadoop" > sample.txt

Upload it to HDFS:

hdfs dfs -put sample.txt /input/

Check the upload:

hdfs dfs -ls /input

You should see sample.txt listed.

If /output already exists, delete it first:

hdfs dfs -rm -r /output

To Run the WordCount MapReduce example

hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordcount /input /output

Check the output

List output directory:

hdfs dfs -ls /output

Display the results:

hdfs dfs -cat /output/part-r-00000

You should see a word count of all words in sample.txt, e.g.:

Hadoop 3

Hello 3

World 1

is 1

fun 1