sudo apt-get install openssh-client

sudo apt-get install openssh-server

ssh localhost

ssh-keygen -t rsa -P '' -f ~/.ssh/id\_rsa

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

chmod 0600 ~/.ssh/authorized\_keys

cd hadoop-3.3.1/

sbin/start-dfs.sh

jps

sbin/stop-dfs.sh

jps

bin/hdfs dfs -mkdir /tuesday

#This is to create a folder

bin/hdfs dfs -put amazon-meta.txt /tuesday

here we are uploading amazon-meta.txt to tuesday folder .this is called file write to hdfs.and the files will be created into blocks and t takes no of replications from default as we have given 2 . (in hadoop each block is 128 mb from hadoop secong version and in 1st version it was 64 mb ).it will be dived into blocks 932.22(c in link to get file size)/128=7.28~~8 so it ll divide into 8 block,1st 7 blocks will be 128 mb so 128\*7 is 896 so the left over will be last block size ie 932.22-896 ~~36 so it ll be 36 mb.

#to compile the wordcount program

javac -classpath "share/hadoop/common/hadoop-common-3.3.1.jar:share/hadoop/common/lib/hadoop-annotations-3.3.1.jar:share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.3.1.jar:Share/hadoop/common/lib/commons-cli-1.2.jar" -d wordcount WordCount.java

cd wordcount/

ls

jar cvf wordCount.jar \*.class

# this is to make java achive file ,this jar file is having all the classes from the wordcount

then paste the jar file here----------> /home/sai/hadoop-3.3.1/share/hadoop/mapreduce

cd ..

we should run yarn to get resource manager and node manager

sbin/start-yarn.sh

jps

# so all 5 process are running now

http://127.0.0.1:8088 ------------>to check how many mappers are there we use this ip adress

bin/yarn jar /home/parallels/hadoop-3.3.1/share/hadoop/mapreduce/wordCount.jar WordCount /tuesday /tuesday\_output #this is to run the program taking input

head -100 xaa

(xaa is the input file ie amazon-meta.txt in our program ) #100 lines of input data

wc -l part-r-oooo #this gives us how many lines are there in the file and part-r-oooo is the input file.

bin/hdfs dfs -D dfs-blocksize=64000000 -put xaa /thrusday ---------------->64000000 it split the data file taking soppose 300 is total size and 300/64 creates 5 blocks

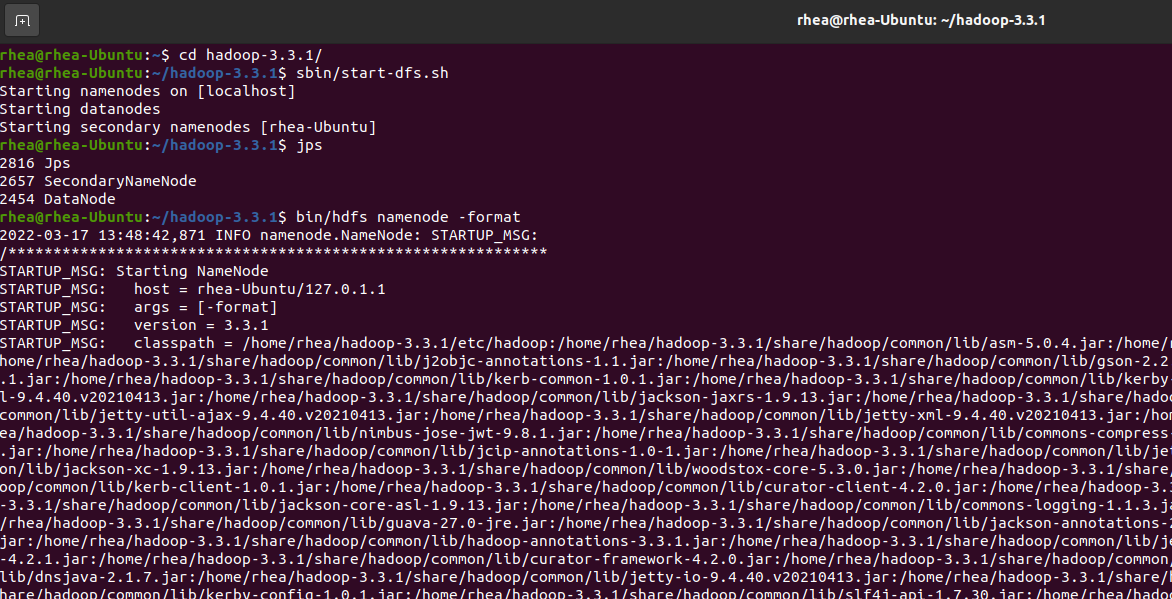
cd hadoop-3.3.1/

sbin/start-dfs.sh

jps

sbin/stop-dfs.sh

bin/hdfs namenode –format



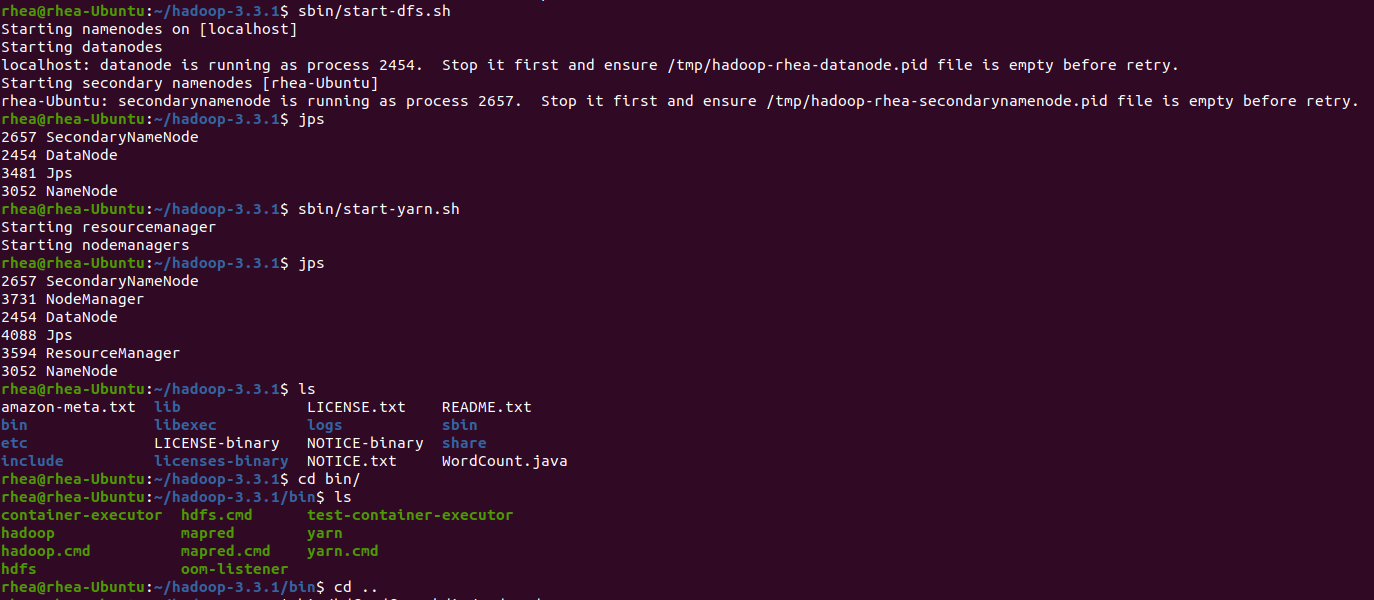
sbin/start-dfs.sh

jps

sbin/start-yarn.sh

sbin/start-dfs.sh

jps



bin/hdfs dfs -mkdir /Tuesday

bin/hdfs dfs -put amazon-meta.txt /Tuesday

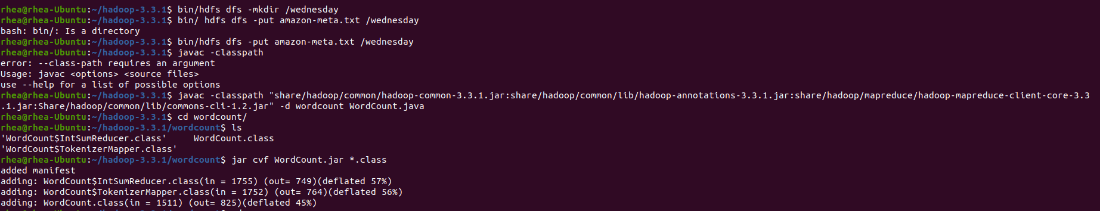
javac -classpath "share/hadoop/common/hadoop-common-3.3.1.jar:share/hadoop/common/lib/hadoop-annotations-3.3.1.jar:share/hadoop/mapreduce/hadoop-mapreduce-client-core-3.3.1.jar:Share/hadoop/common/lib/commons-cli-1.2.jar" -d wordcount WordCount.java

cd wordcount/

ls

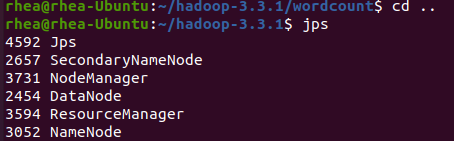
jar cvf WordCount.jar \*.class

(then copy the jar file from home/NAME/hadoop-3.3.1/wordcount and paste the jar file here----------> /home/NAME/hadoop-3.3.1/share/hadoop/mapreduce)



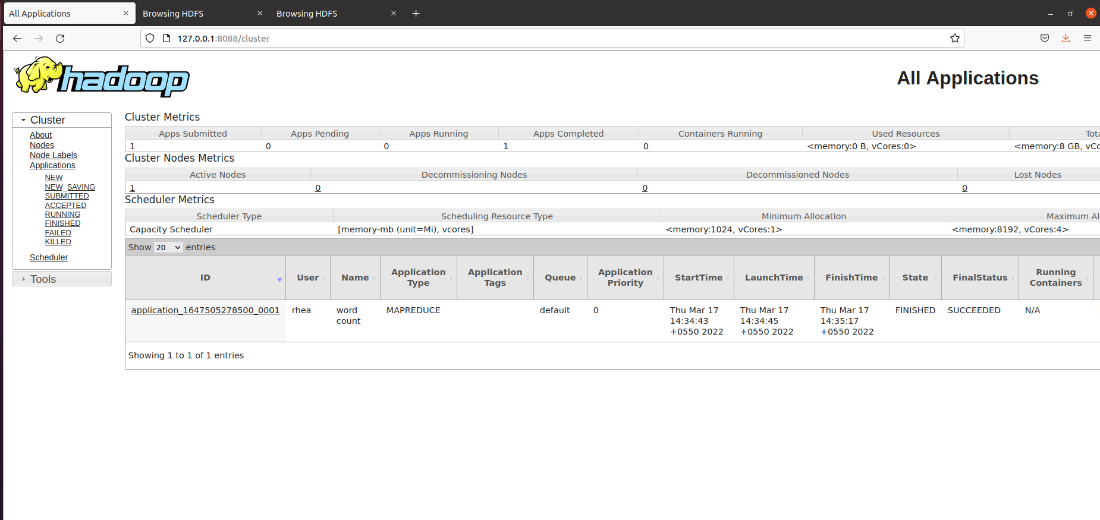
cd ..

Jps



now go to browser and search for url <http://127.0.0.1:8088>

(check number of mappers running)



Come back to terminal

bin/yarn jar /home/NAME/hadoop-3.3.1/share/hadoop/mapreduce/WordCount.jar WordCount /tuesday /tuesday\_output

(in terminal output for this line, check if INFO mapreduce.Job: map 100% reduce 100%)

Text

Description automatically generated



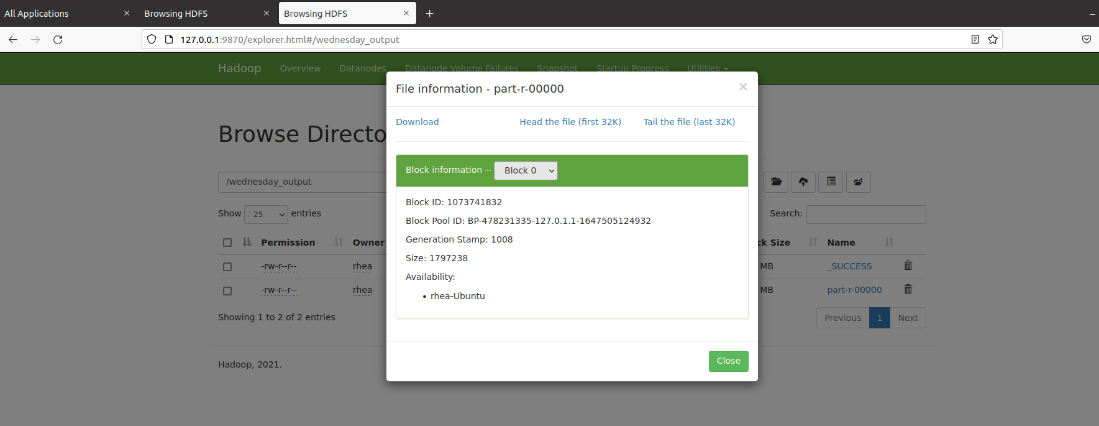
now go to browser and search for url <http://127.0.0.1:9870>

(click on Utilities dropdown on the top)

(click on Browse The file system)

(tuesday\_output file is the output file, click on it)

(click on part-r-0000 and download it)



22-03-2022

NEW TERMINAL

Cd Hadoop 3.2.2

Sbin/start-dfs.sh

Jps

Sbin/start-yarn.sh

jps

Cd Hadoop-2.7.2S/ (check directory before doing this)

Cd hive/

Ls

Cd bin

./hive –server hiveserver

NEW TERMINAL

Inside ~/hive/bin$

./hive

(hive shell opens)

Show databases;

Create database AI20Batch (use àchecks if the database exists, if yes then it becomes the current database else it creates one)

Use AI20Batch (to use a database that has been created)

Show tables; (list the tables inside the database created)

Two types of tables can be created (table created is in HDFS)

* Internal table / managed table (we will create this)
* External table

Create external table if not exists TABLENAME(rollno INT, name STRNG, gpa, FLOAT) row format delimited fields terminated by ‘\t’ ;

Show tables;

LOAD DATA LOCAL INPATH ‘FILENAME.tsv’ OVERWRITE INTO TABLE TABLENAME; (FILENAME.tsv -> separated by tab space)

Select \* from TABLENAME; (star à all the columns are extracted)

(otherwise we can write   
 Select ColumnName(separated column by ,) from TABLENAME CONDITION

EG: FOR GPA>5

Select \* from student where gpa>5; )

select name, rollno from student (only name and rollno columns are extracted)

(when we write a query, the hive query is translated into map reduce job)

Hive à introduced by facebook (for SQL professionals working with structured data, couldn’t utilize the benefits of Hadoop cluster)

CODE (22-03-2022)

Cd Hadoop 3.2.2

Sbin/start-dfs.sh

Jps

Sbin/start-yarn.sh

jps

Cd Hadoop-2.7.2S/ (check directory before doing this)

Cd hive/

Ls

Cd bin

./hive –server hiveserver

NEW TERMINAL

Inside ~/hive/bin$

./hive

(hive shell opens)

Show databases;

Create database AI20Batch

Use AI20Batch

Show tables;

CREATE TABLE IF NOT EXISTS STUDENT(rollno INT, name STRING, gpa FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘\t’

LOAD DATA LOCAL INPATH ‘student.tsv’ OVERWRITE INTO TABLE student;

Select \* from student;

Select \* from student where gpa>5;

(only map task was done for this, not reduce task à because there is not aggregation / reducing stage for this task)

Hdfs àuser à hive à warehouse à AI20batch.dbà student à student1.tsv

24-03-2022

Cd Hadoop

bin/start-dfs.shjps

NEW TERMINAL

Cd

Cd pig-0.10.1/

Cd bin

./pig -x local

Grunt> A = load ‘Pig\_scripts\_dataset/student1.tsv’ (rollno:int, name:chararray, gpa:float);

Grunt>

**Q. Consider the following text file which contains 4 lines of text.sample.txt  
we welcome you to school today  
school today school today  
we welcome you to school today  
please come in and play  
Explain map and reduce algorithm to count how many words start with w, y, t, s, p, c, i, and  
a. Also, identify what would be the <key, value> from the mapper's and reducer's output.  
Draw the MapReduce workflow diagram for the same.  
The output of this mapreduce algorithm should be as follows.  
word\_startwith\_a 1  
word\_startwith\_w 4  
word\_startwith\_y 2  
word\_startwith\_t 6  
word\_startwith\_s 4  
word\_startwith\_p 2  
word\_startwith\_c 1  
word\_startwith\_i 1**

Changes:

While(itr.hasMoreTokens())

{

String temp = itr.nextToken();

If(temp.charAt(0)==’a’)

Context.write(new text(“word\_startwith\_a”),one);

Else If(temp.charAt(0)==’w’)

Context.write(new text(“word\_startwith\_w”),one);

…….

…….

Word.set(itr.nextToken());

…….

}

No need to change reduce function (no modification required)

Modify main and reduce function for lab exam

Sudo apt install default -jdk

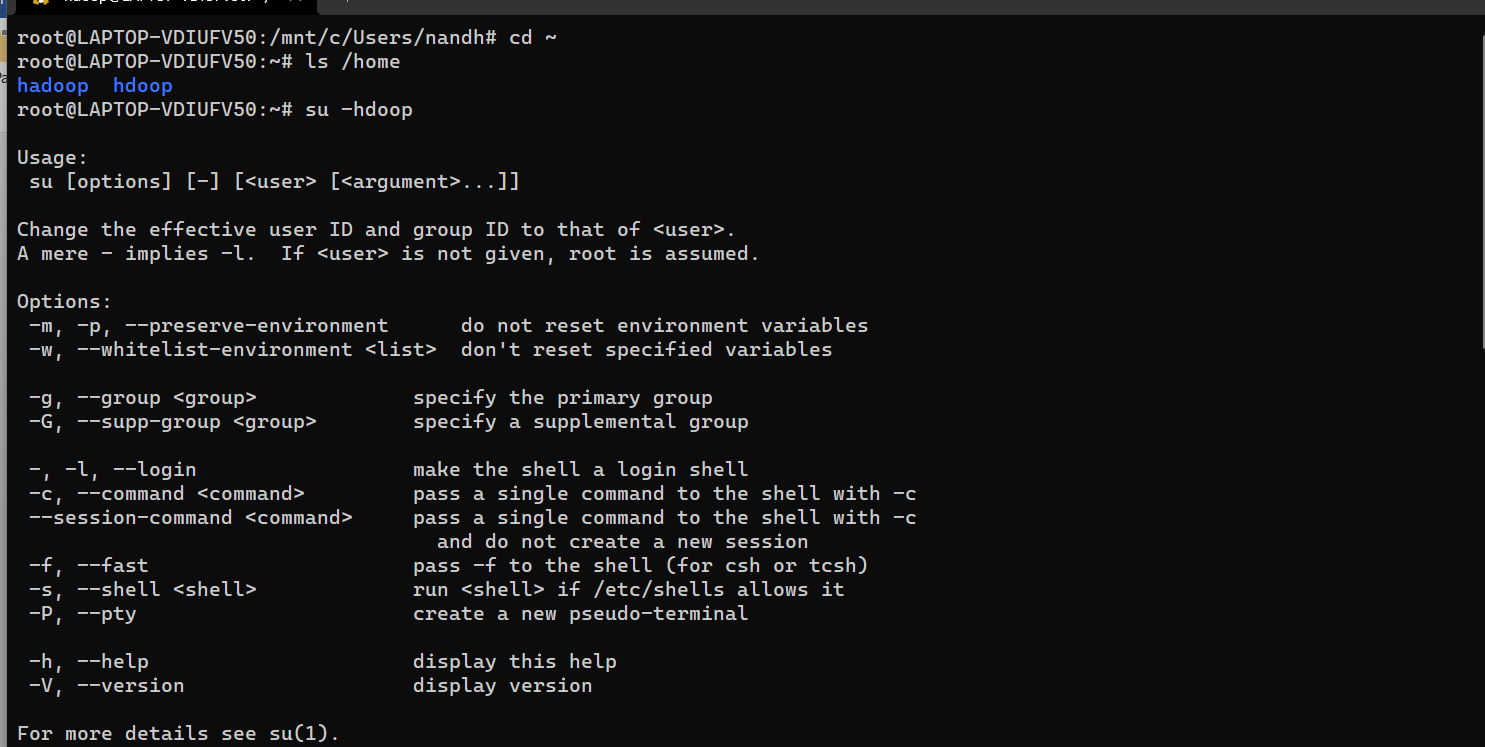
Java –version javac –version 11.

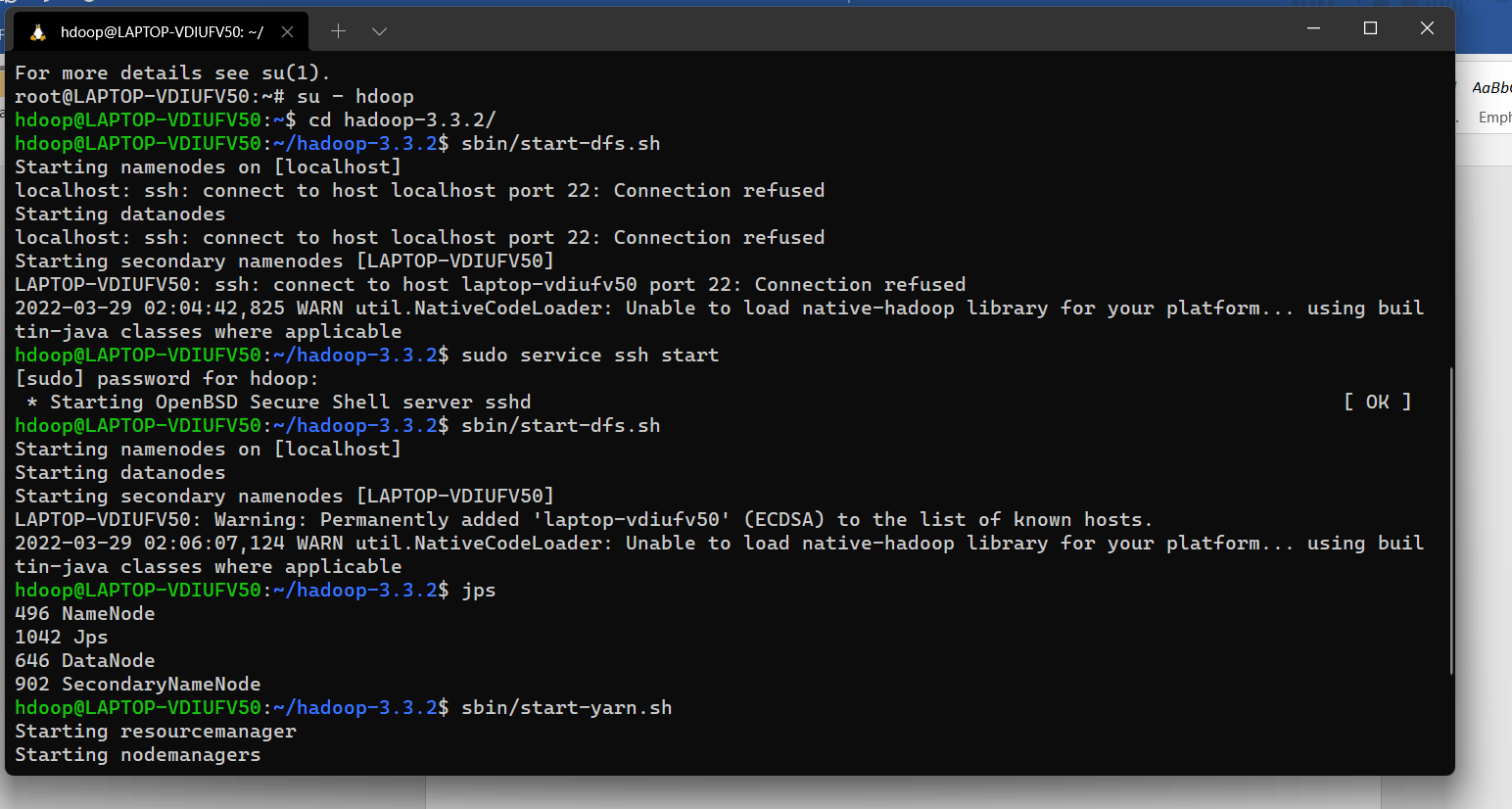
Sudo apt install defaulty-jdk

Java –version

Javac –version

Output -> 11.0.4





hdoop password: hdoop