**VENNELA G**

**20BDS0146**

**PRINCIPLES OF CLOUD COMPUTING**

**LAB ASSESSMENT-2**

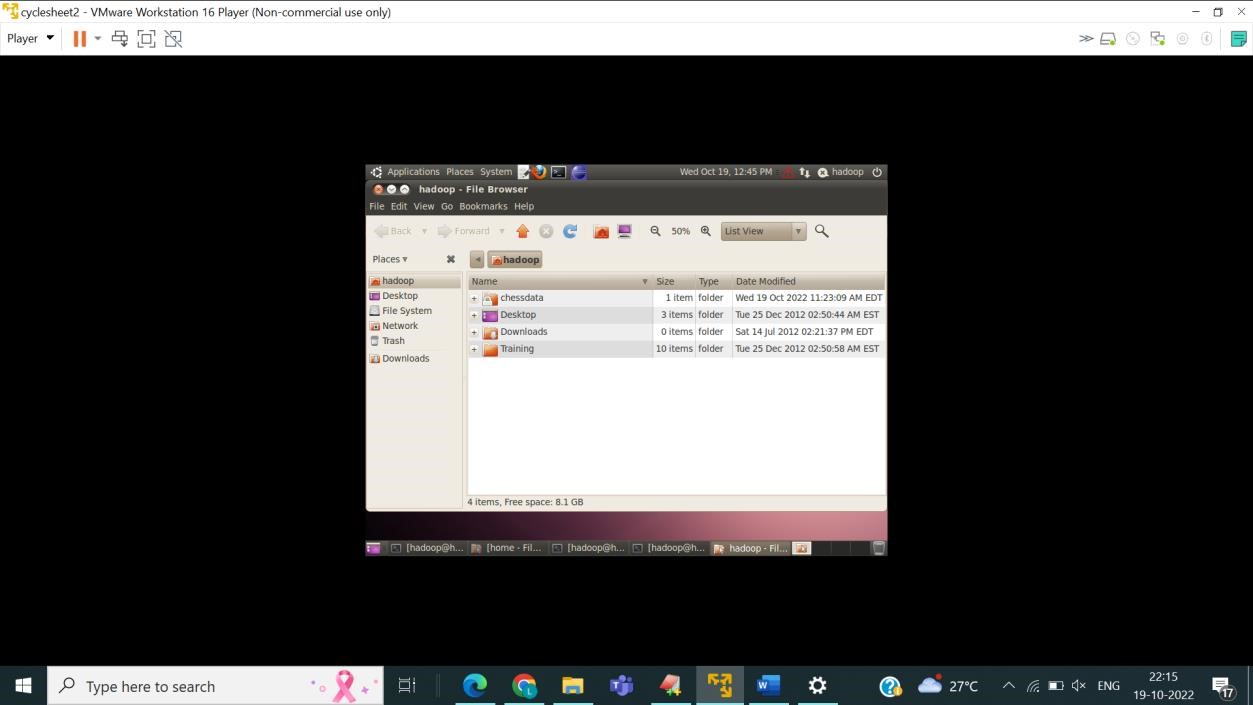
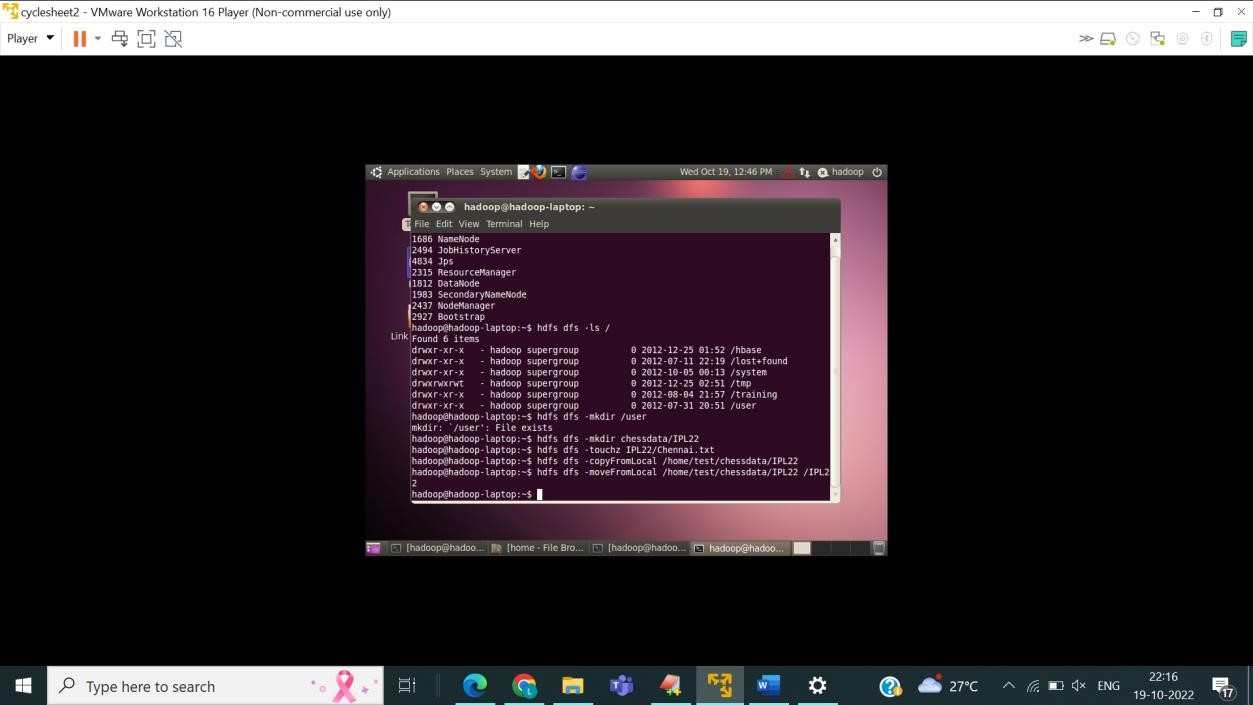
**QUESTION 1:**

* Start and check hadoop

Start-all.sh

Jps

* LS command-to see directories hdfs dfs -ls /
* Making Directory- hdfs dfs -mkdir chessdata/IPL22
* Create new file- hdfs dfs -touchz IPL22/Chennai.txt
* Copy from Local hdfs dfs -copyFromLocal chessdata/IPL22
* This command will move file from local to hdfs- hdfs dfs -moveFromLocal chessdata/IPL22/Chennai.txt /IPL22



* 1. Creating Directories and Subdirectories-
* Command- cd /home sudo Mkdir test

Cd test

sudo Mkdir chessdata Cd chessdata sudo Mkdir IPL22 Cd IPL22

* 1. Creating a file using gedit editor
* Command-

Gedit Chennai.txt

1. Creating the directory in DFS folder-

• Command- Cd /home/Hadoop sudo Mkdir regno Cd regno

sudo Mkdir chessdata Cd chessdata sudo Mkdir ipl22 Cd ipl22

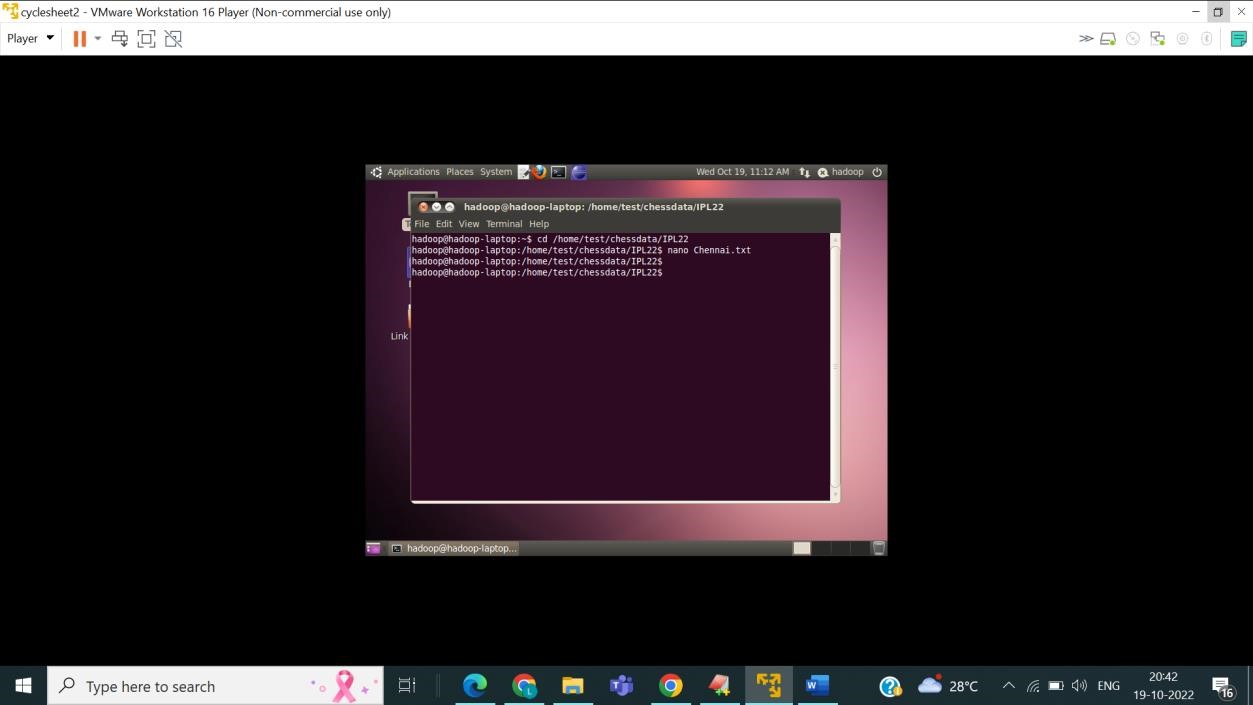
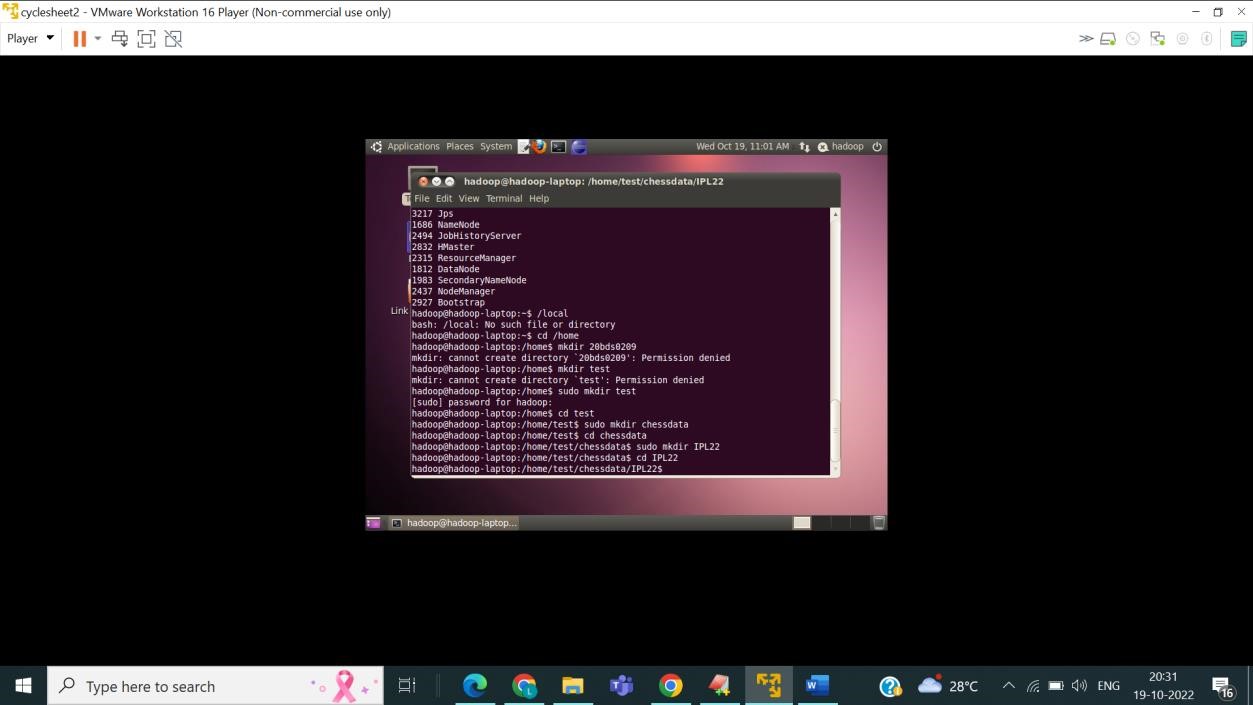
1. Transfer the file from local system to hdfs folder-

• Commands-This is the command to copy the file

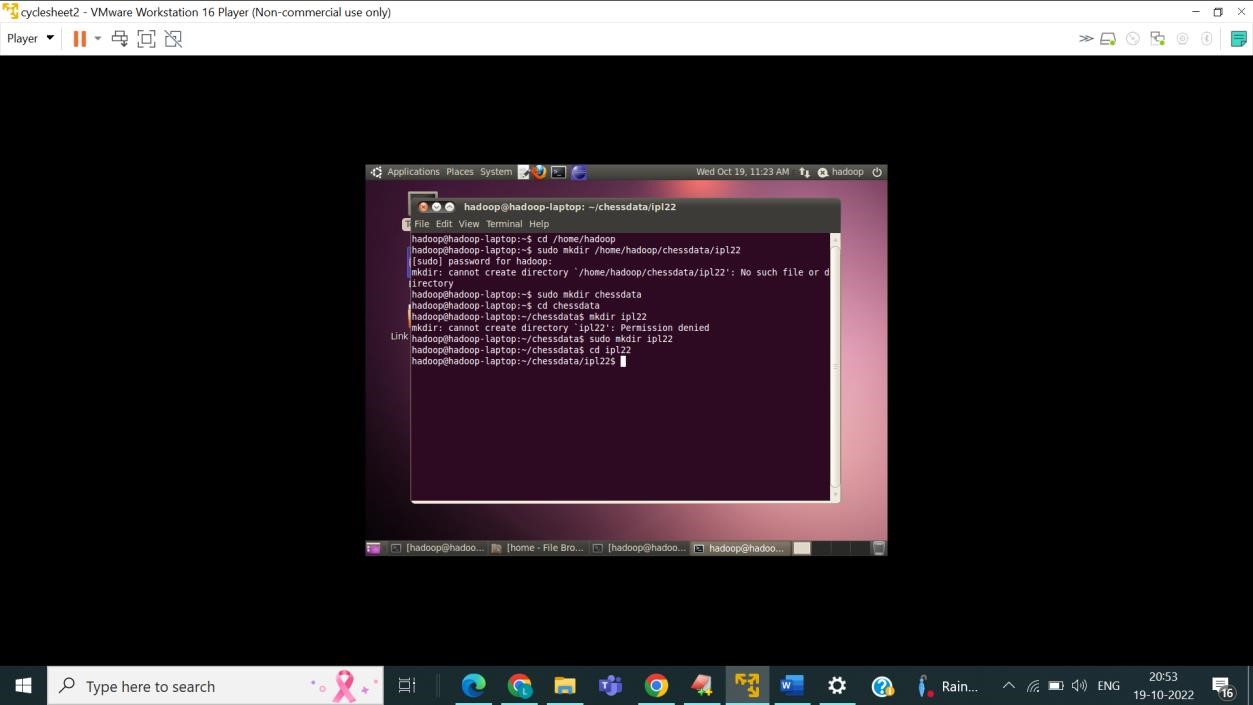
Hdfs dfs -cp /home/test/chessdata/IPL22/Chennai.txt

This is the command to put the file from local system to HDFS folder

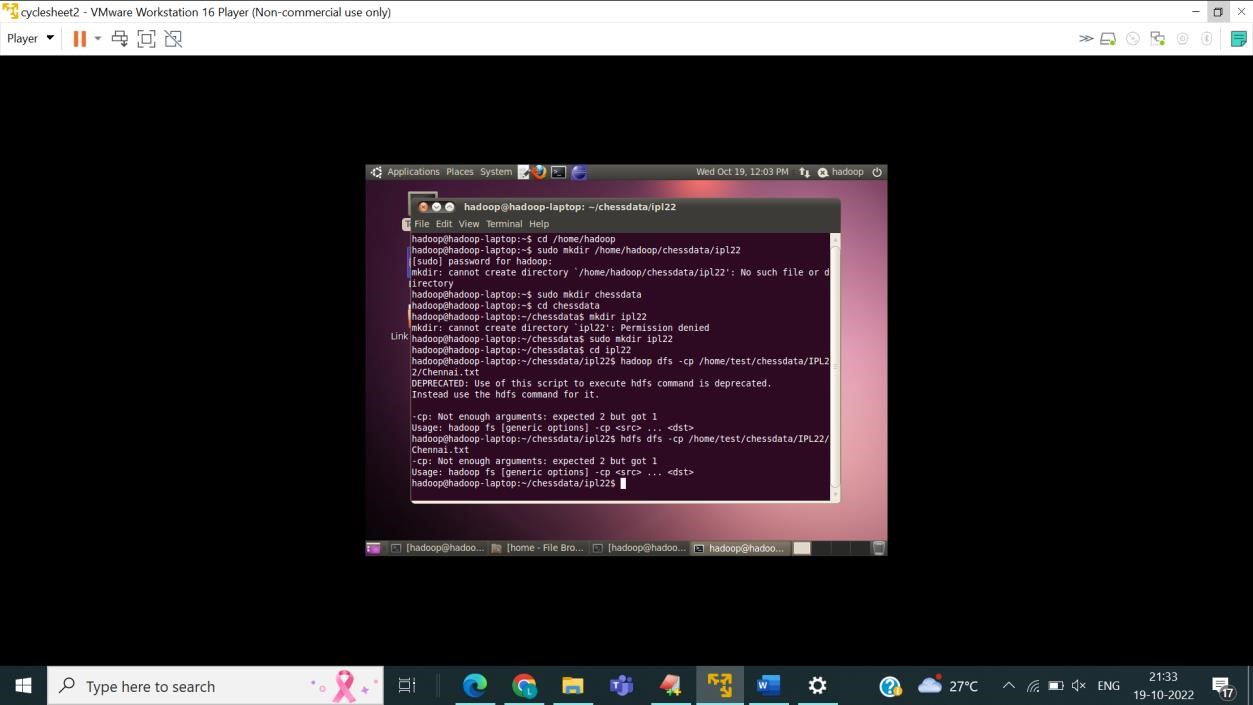
Hdfs dfs -put /home/Hadoop/regno/chessdata/ipl22

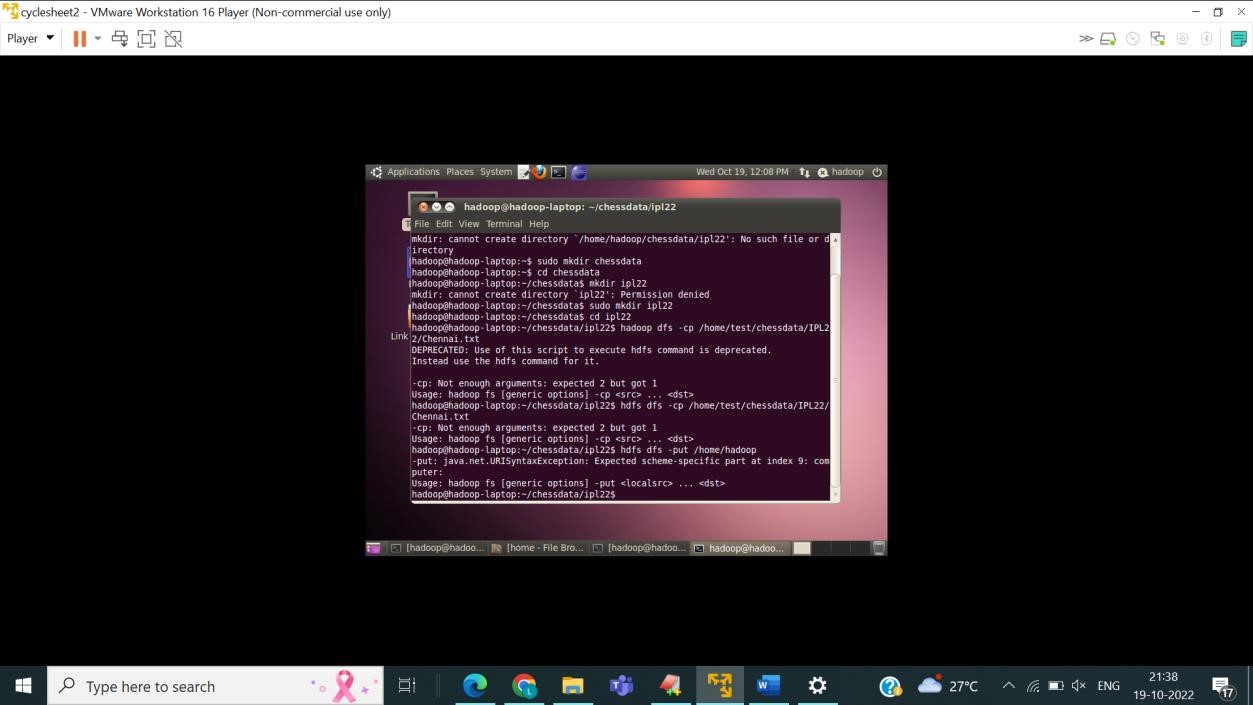


DFS FOLDER-



Transferring File from Local system to HDFS folder





* 1. Creating Directories and Subdirectories-
     + Command-

cd

Mkdir test

Cd test

Mkdir chessdata

Cd chessdata

Mkdir IPL22

Cd IPL22

* 1. Creating a file using gedit editor
     + Command-

Type nul > Chennai.txt

* 1. Creating the directory in DFS folder-
     + Command- Cd /home/Hadoop

Mkdir regno

Cd regno

Mkdir chessdata

Cd chessdata

Mkdir ipl22

Cd ipl22

* 1. Transfer the file from local system to hdfs folder-
     + Commands-This is the command to copy the file

Hdfs dfs -cp /Users/Vennelag/chessdata/IPL22/Chennai.txt

This is the command to put the file from local system to HDFS folder

Hdfs dfs -put /Users/Vennelag/chessdata/ipl22

**QUESTION 2:**

import java.io.IOException;

import java.util.StringTokenizer; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.fs.Path; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.mapreduce.Reducer; import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class pcc\_cyclesheet2\_2

{

public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>

{

private Text word=new Text(); private final static IntWritable one=new IntWritable(1); public void map (Object key, Text value, Context context )throws IOException,

InterruptedException{

StringTokenizer itr=new StringTokenizer(value.toString());

while(itr.hasMoreTokens())

{

word.set(itr.nextToken());

context.write(word,one);

}

}

}

public static class IntSumReducer extends Reducer<Text,IntWritable,Text,IntWritable>

{

private IntWritable result=new IntWritable();

public void reduce(Text key,Iterable<IntWritable>values, Context context) throws IOException,InterruptedException

{

int sum=0;

for(IntWritable val:values)

{

sum+=val.get();

}

results.set(sum); context.write(key,result);

}

}

public static void main(String[] args)

{

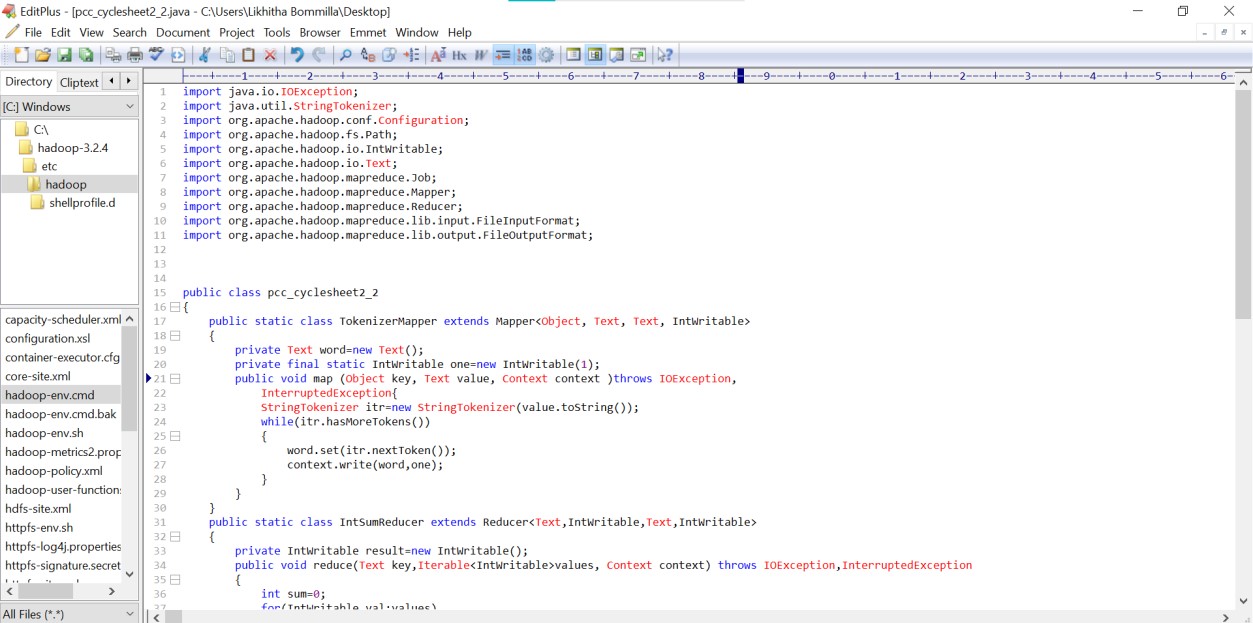
Configuration conf=new Configuration(); Job job=Job.getInstance(conf,"word count"); job.setJarByClass(WordCount.class); job.setMapperClass(TokenizerMapper.class); job.setCombinerClass(IntSumReducer.class); job.setReducerClass(IntSumReducer.class); job.setOutputKeyClass(Text.class); job.setOutputValueClass(IntWritable.class);

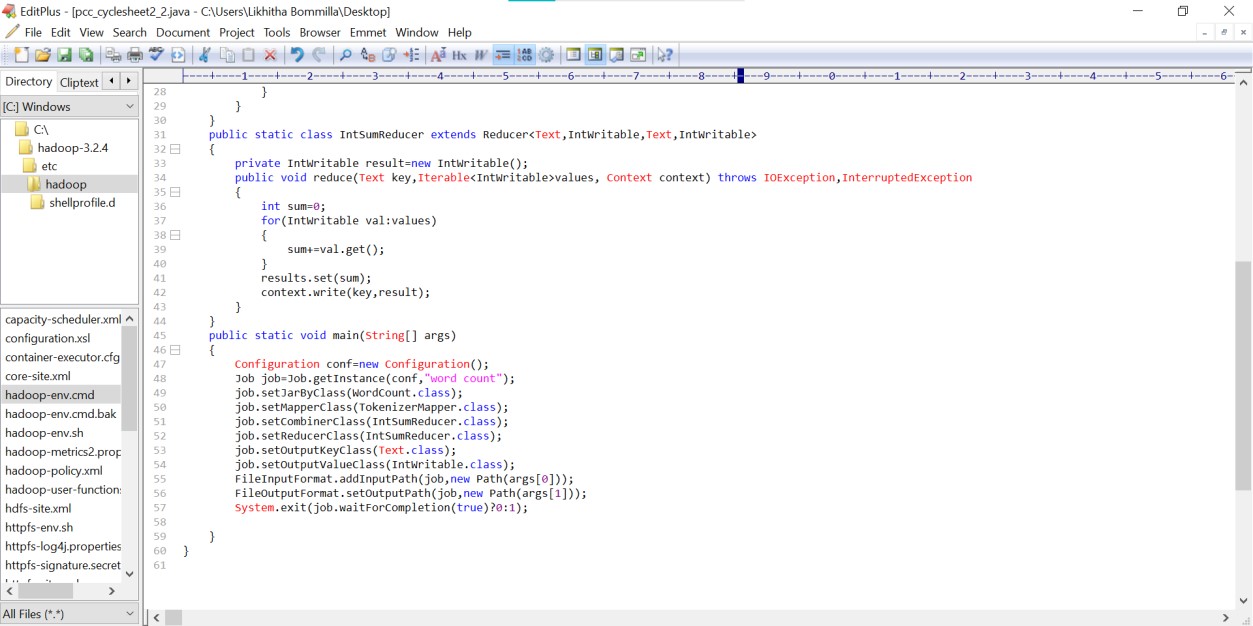
FileInputFormat.addInputPath(job,new Path(args[0])); FileOutputFormat.setOutputPath(job,new Path(args[1]));

System.exit(job.waitForCompletion(true)?0:1);

}

}





HADOOP-

1. Making a local WordCount\_classes directory- Mkdir wordcount\_classes
2. Compile the pcc\_cyclesheet2\_2 , use the Hadoop classpath

Javac -cp Hadoop classpath -d wordcount\_classes pcc\_cyclesheet2\_2.java

1. Create a java archive for distribution

Jar -cvf wordcount.jar -C wordcount-classes/ .

1. Create a directory and move the file into HDFS

Hdfs dfs -mkdir Hometown

Hdfs dfs -put Hometown.txt

1. Run work count, but first

Hadoop jar Hometown.jar Count hometown

1. Check for output from Hadoop job
2. Move it back to working directory

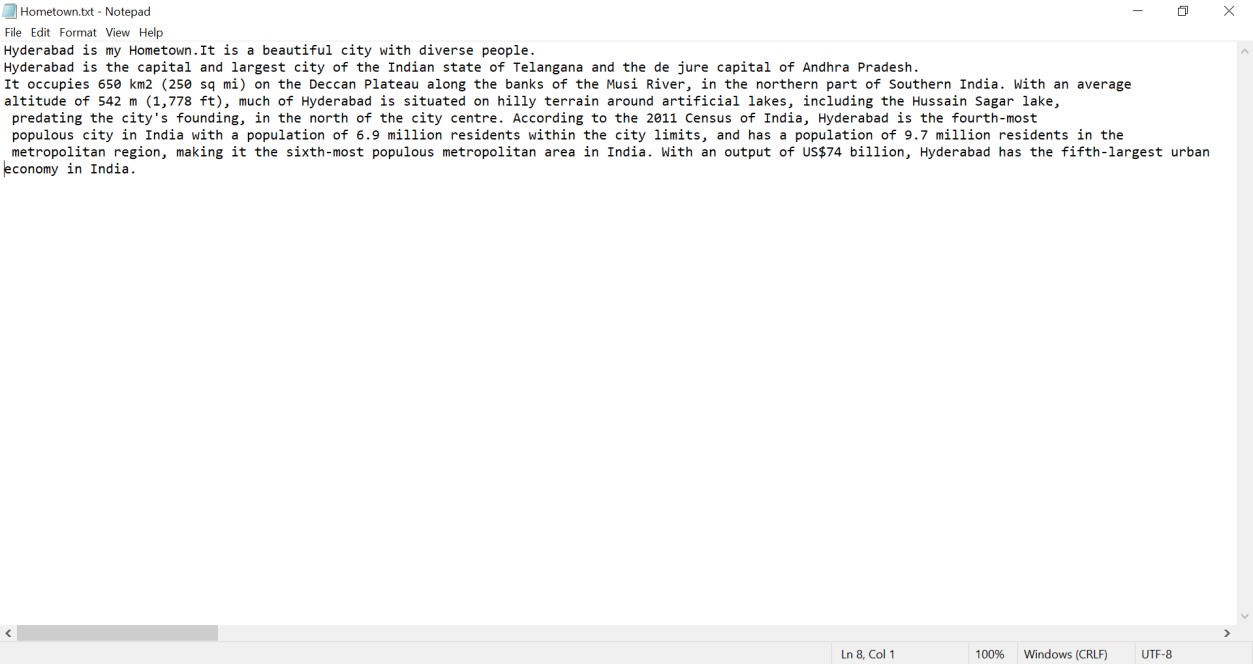
C:\Users\Vennelag>cd c:\hadoop c:\hadoop>sbin\start-dfs c:\hadoop>sbin\start-yarn starting yarn daemons

C:\Users\Vennelag>cd c:\hadoop

C:\hadoop>bin\hdfs dfs -mkdir input

C:\hadoop>bin\hdfs dfs -copyFromLocal c:/Hometown.txt input

Hometown.txt has:



C:\hadoop>hdfs dfs -ls input

Found 1 items

-rw-r--r-- 1 Vennelag supergroup 55 2014-02-03 13:19 input/Hometown.txt

C:\hadoop>bin\hdfs dfs -cat input/Hometown.txt

**Mapreduce:**

C:\hadoop>bin\yarn jar share/hadoop/mapreduce/hadoop-mapreduce-examples-2.2.0.jar wordcount input output

14/02/03 13:22:02 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

14/02/03 13:22:03 INFO input.FileInputFormat: Total input paths to process : 1 14/02/03 13:22:03 INFO mapreduce.JobSubmitter: number of splits:1

: :

14/02/03 13:22:04 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1391412385921\_0002

14/02/03 13:22:04 INFO impl.YarnClientImpl: Submitted application application\_1391412385921\_0002 to ResourceManager at /0.0.0.0:8032

14/02/03 13:22:04 INFO mapreduce.Job: The url to track the job: http://ABHIJITG:8088/proxy/application\_1391412385921\_0002/

14/02/03 13:22:04 INFO mapreduce.Job: Running job: job\_1391412385921\_0002

14/02/03 13:22:14 INFO mapreduce.Job: Job job\_1391412385921\_0002 running in uber mode : false

14/02/03 13:22:14 INFO mapreduce.Job: map 0% reduce 0%

14/02/03 13:22:22 INFO mapreduce.Job: map 100% reduce 0%

14/02/03 13:22:30 INFO mapreduce.Job: map 100% reduce 100%

14/02/03 13:22:30 INFO mapreduce.Job: Job job\_1391412385921\_0002 completed successfully

14/02/03 13:22:31 INFO mapreduce.Job: Counters: 43

File System Counters

FILE: Number of bytes read=89

FILE: Number of bytes written=160142

FILE: Number of read operations=0

FILE: Number of large read operations=0

FILE: Number of write operations=0

HDFS: Number of bytes read=171

HDFS: Number of bytes written=59

HDFS: Number of read operations=6

HDFS: Number of large read operations=0

HDFS: Number of write operations=2

Job Counters

Launched map tasks=1

Launched reduce tasks=1

Data-local map tasks=1

Total time spent by all maps in occupied slots (ms)=5657

Total time spent by all reduces in occupied slots (ms)=6128

Map-Reduce Framework

Map input records=2

Map output records=7

Map output bytes=82

Map output materialized bytes=89

Input split bytes=116

Combine input records=7

Combine output records=6

Reduce input groups=6

Reduce shuffle bytes=89

Reduce input records=6

Reduce output records=6

Spilled Records=12

Shuffled Maps =1

Failed Shuffles=0

Merged Map outputs=1

GC time elapsed (ms)=145

CPU time spent (ms)=1418

Physical memory (bytes) snapshot=368246784

Virtual memory (bytes) snapshot=513716224

Total committed heap usage (bytes)=307757056

Shuffle Errors

BAD\_ID=0

CONNECTION=0

IO\_ERROR=0

WRONG\_LENGTH=0 WRONG\_MAP=0

WRONG\_REDUCE=0

File Input Format Counters

Bytes Read=55

File Output Format Counters

Bytes Written=59

C:\hadoop>bin\hdfs dfs -cat output/\*

Example 1

Hadoop 2

Install 1

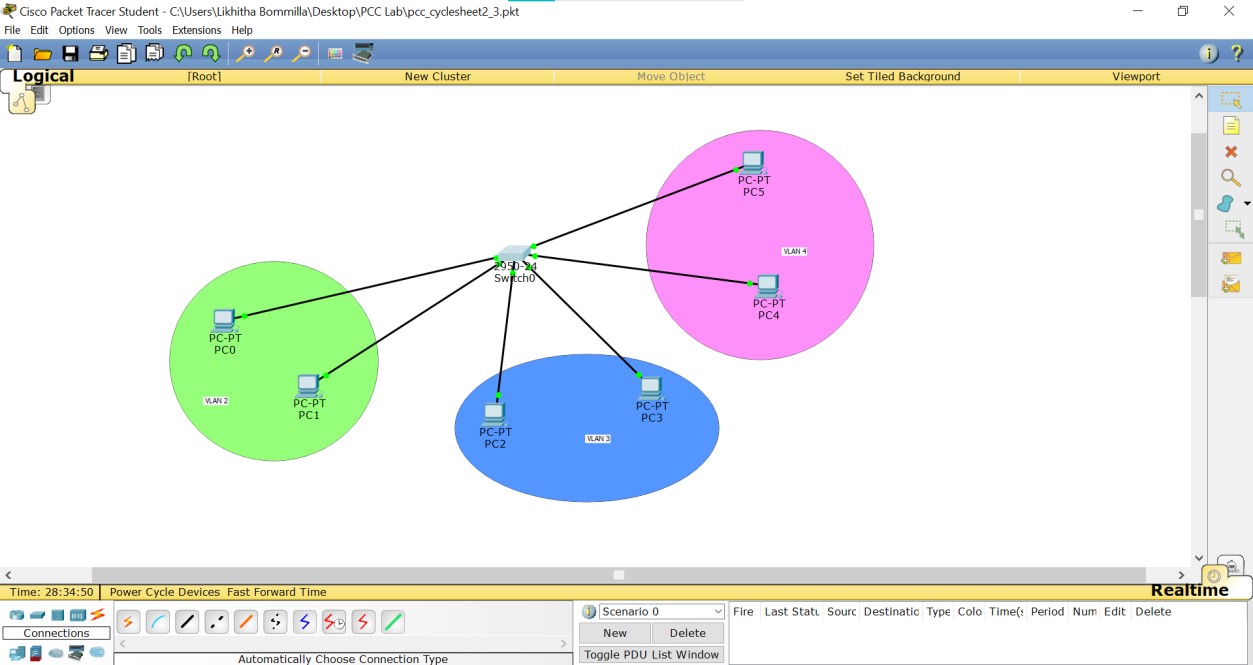
Mapreduce 1

Run 1

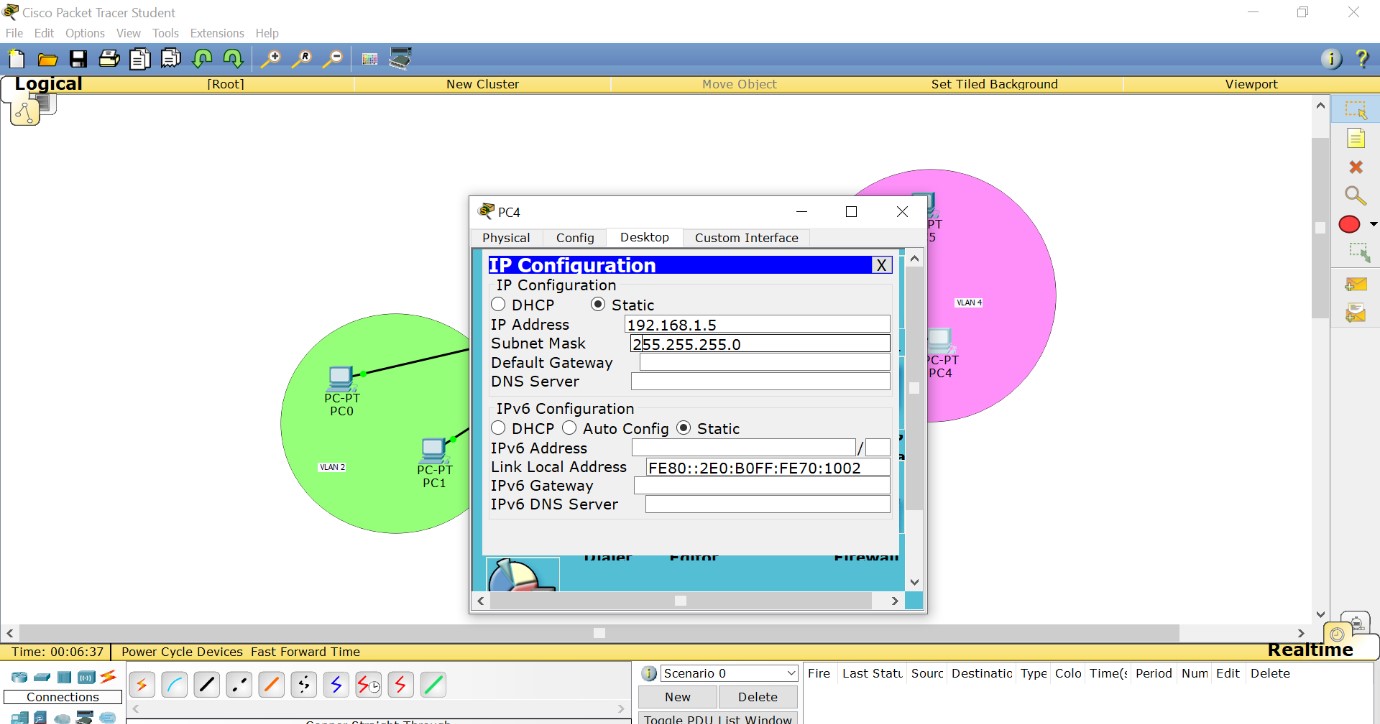
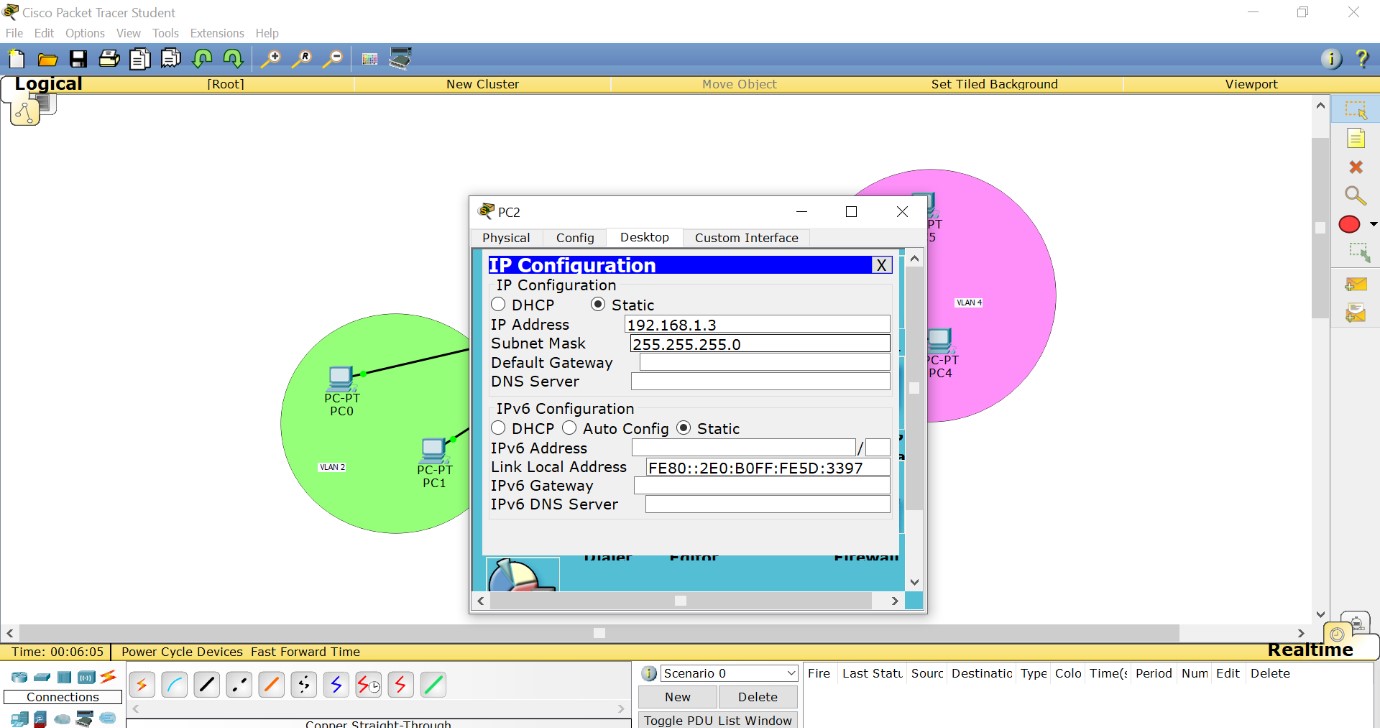
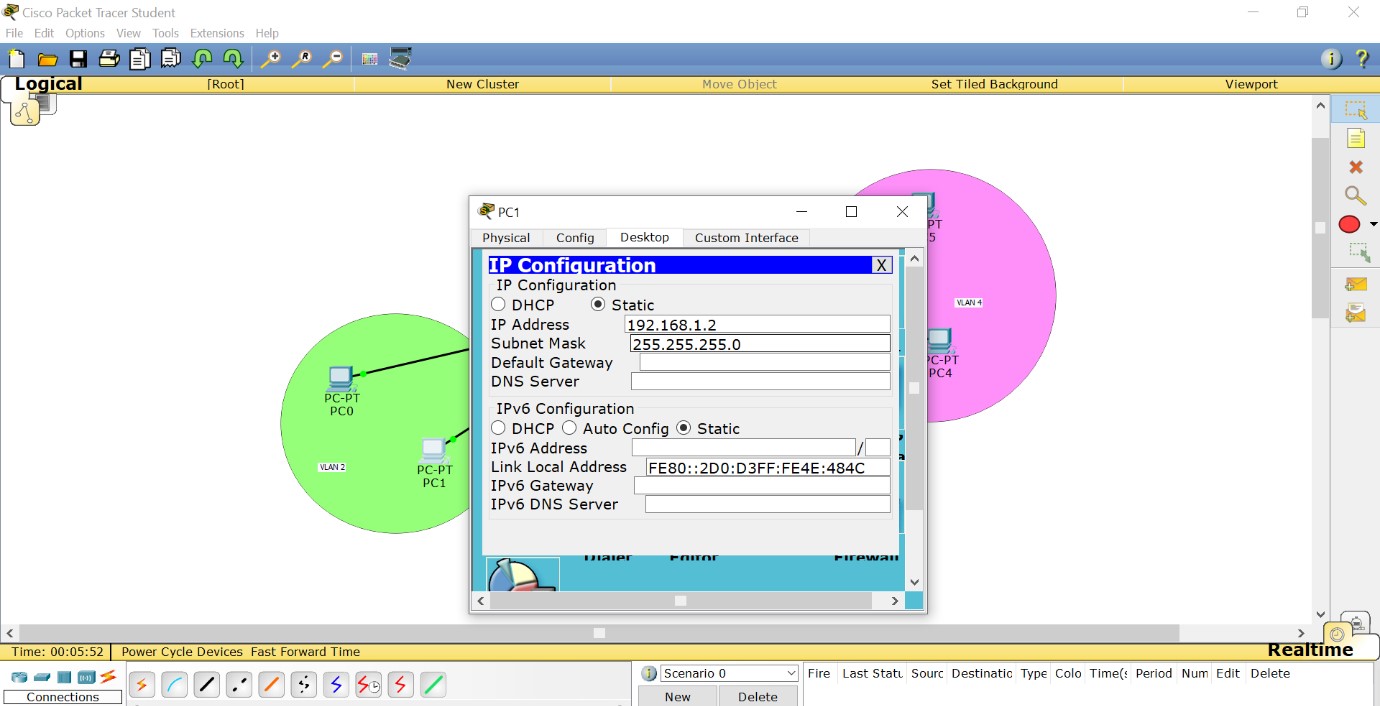
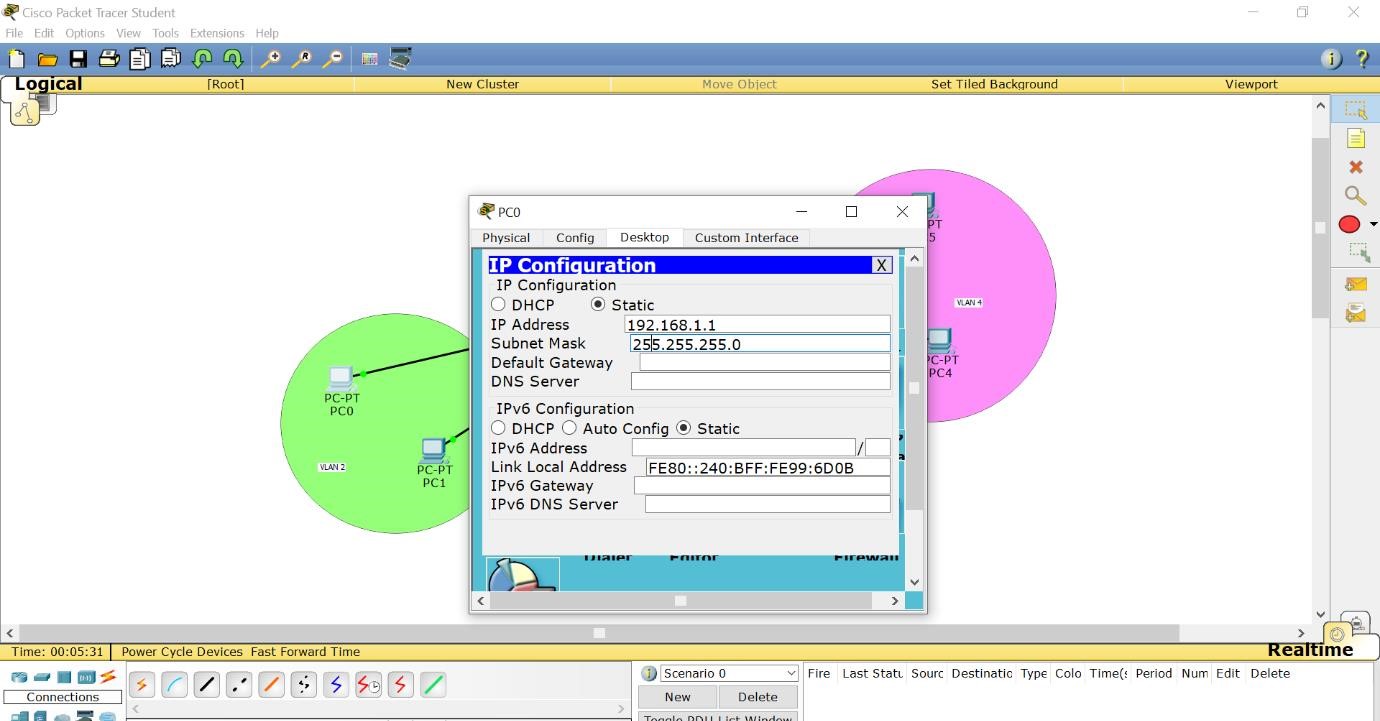
Wordcount 1

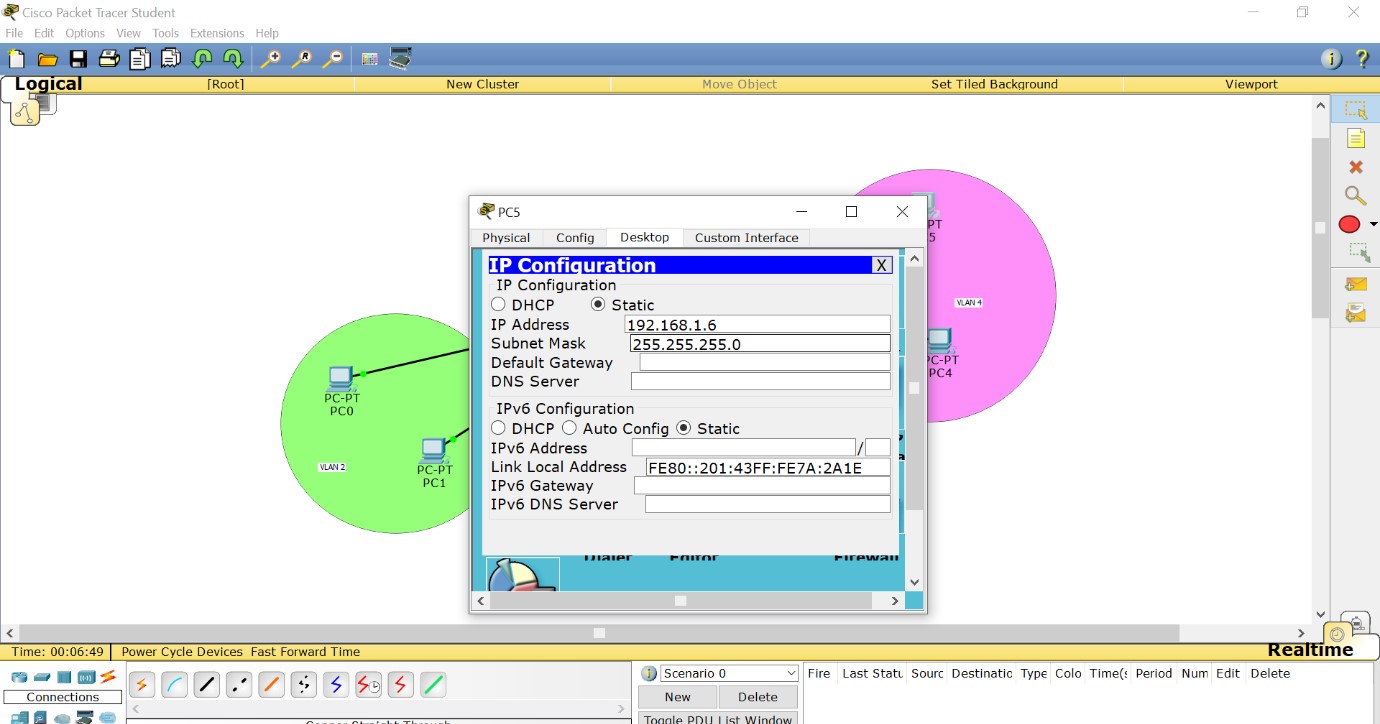
**QUESTION 3**

BUILDING THE CONNECTION

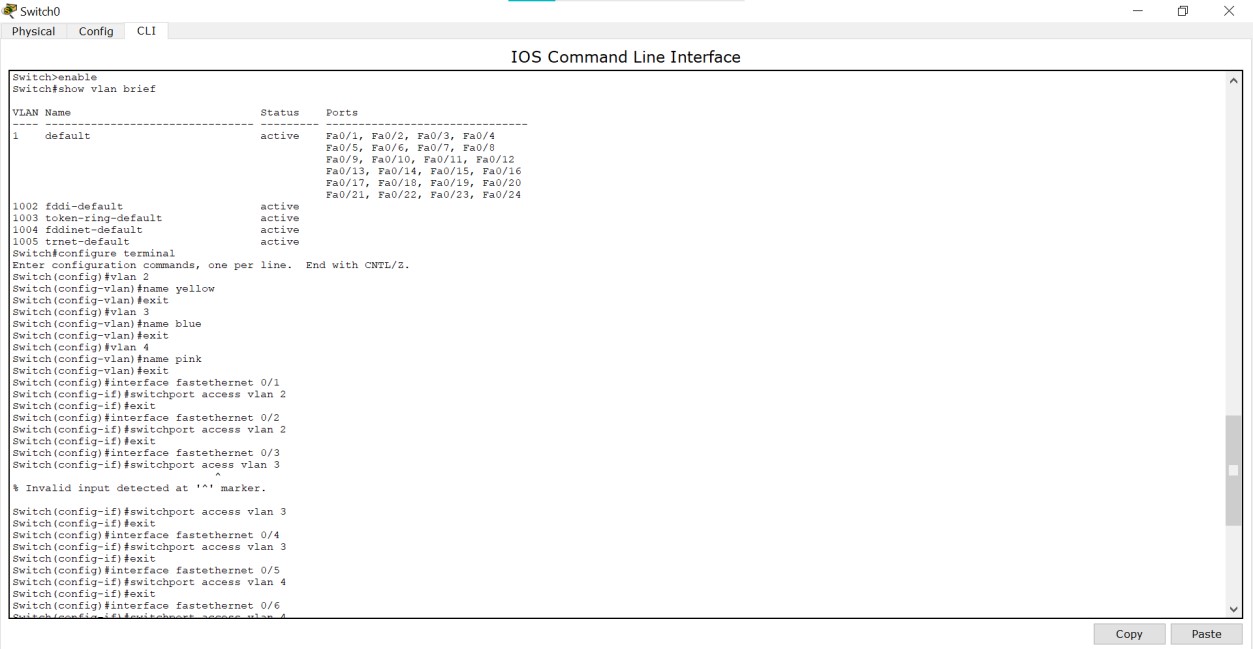


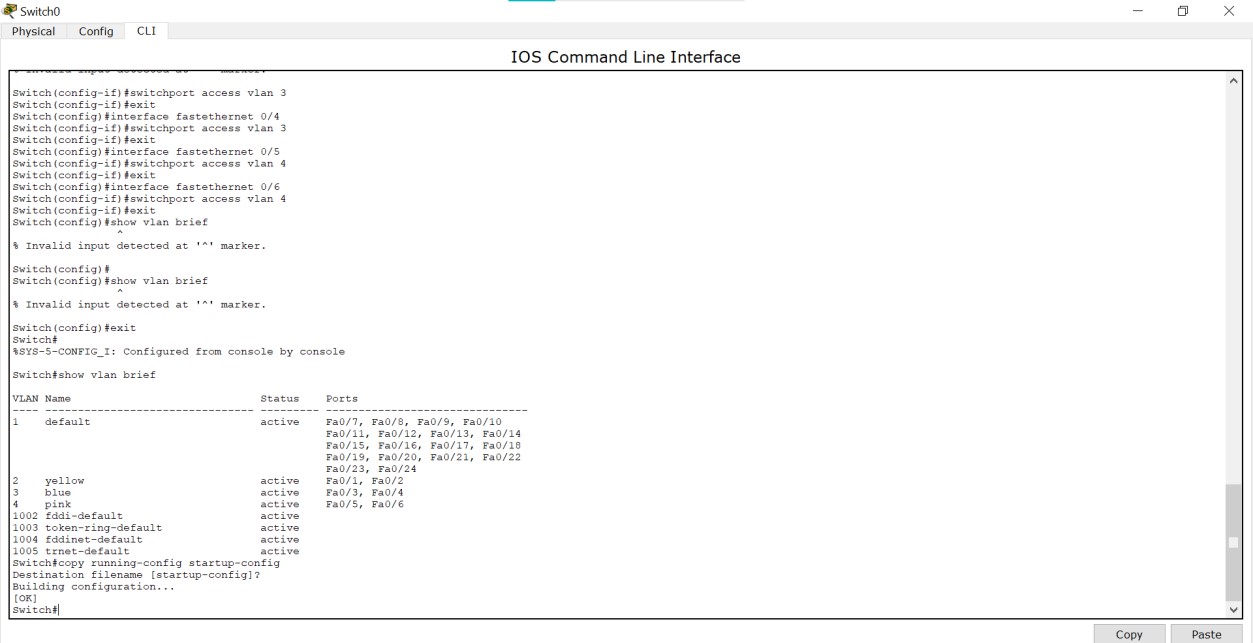
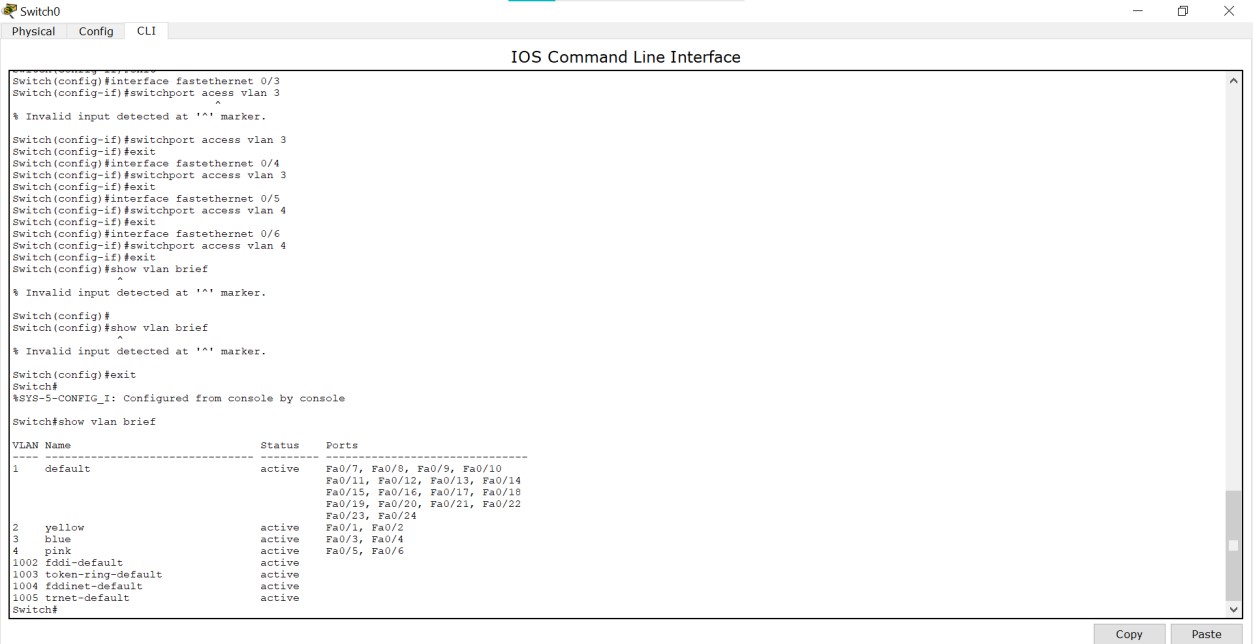
CONFIGURING THE PCS(ADDING IP ADDRESSES)





COMMANDS to switch li





AFTER SENDING THE MESSAGES TO PCS

