

**University Of Mumbai
Institute of Distance & Open Learning**



**PRACTICAL JOURNAL IN
BIG DATA ANALYTICS**

SUBMITTED BY

**APPLICATION ID:
SEAT NO :**

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY PART-I
SEMESTER II**

**ACADEMIC YEAR
2022-2023**

**INSTITUTE OF DISTANCE AND OPEN LEARNING
IDOL BUILDING, VIDYANAGARI,
SANTACRUZ (EAST), MUMBAI-400 098**

**CONDUCTED AT
PARLE TILAK VIDYALAYA ASSOCIATION'S
MULUND COLLEGE OF COMMERCE(AUTONOMOUS)
MULUND (WEST), MUMBAI – 400080**

**University of Mumbai
Institute of Distance & Open Learning**



Dr. Shankar Dayal Sharama Bhavan, Kalina,
Vidanagari, Santacruz (E), Mumbai-400 098.

Certificate

This is to certify that Mr..... Application ID:_____ from Institute of Distance and Open Learning, University of Mumbai , Santacruz East, Mumbai-400098 , PCP Centre: Parle Tilak Vidyalaya Association's Mulund College of Commerce (Autonomous) has successfully completed all the practical of Paper II titled **BIG DATA ANALYTICS** for M.Sc. (I.T.) Part I Semester – II during the academic year **2022-2023**.

Centre Coordinator

External Examiner

Date:

Seal

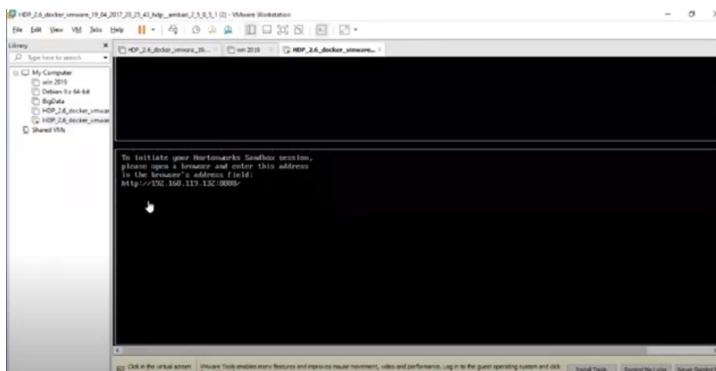
Sr. No	Practical	Date	Sign
1	Install, configure and run Hadoop and HDFS and explore HDFS.		
2	Implement word count / frequency programs using MapReduce.		
3	Implement an MapReduce program that processes a weather dataset.		
4	Implement the program using Pig.		
5	Implement the application in Hive.		
6	Implement an application that stores big data in Hbase/ Python.		
7	Implement Decision tree classification techniques.		
8	Implement SVM classification techniques.		

Practical No. 1

Aim:- Install, configure and run Hadoop and HDFS ad explore HDFS.

Download Virtual machine setup ie VMWare setup (in which Hadoop is configured).

Step 1: Load the server on VM ware workstation



Step 2: To enable admin login open shell and reset root login.

Open Terminal 192.168.119.132:4200

In Sandbox login enter root

And Password is Hadoop

And reset the password

```
root@sandbox.hortonworks.com's password:  
You are required to change your password immediately (root enforced)  
Last login: Wed Jun 30 14:50:19 2021 from 172.17  
.0.2  
Changing password for root.  
(current) UNIX password:  
New password:  
Retype new password:  
[root@sandbox ~]#
```

Windows linux system and Hadoop system are different

When we type **ls** command it is executed in local system

```
(current) UNIX password:  
New password:  
Retype new password:  
[root@sandbox ~]# ls  
anaconda-ks.cfg  install.log.syslog  
blueprint.json   sandbox.info  
build.out        start_ambari.sh  
hdp              start_hbase.sh  
install.log      [root@sandbox ~]#
```

When we type **hdfs dfs -ls** it will execute in Hadoop system directory

```
New password:  
Retype new password:  
[root@sandbox ~]# ls  
anaconda-ks.cfg  install.log.syslog  
blueprint.json   sandbox.info  
build.out        start_ambari.sh  
hdp             start_hbase.sh  
install.log  
[root@sandbox ~]# hdfs dfs -ls /
```

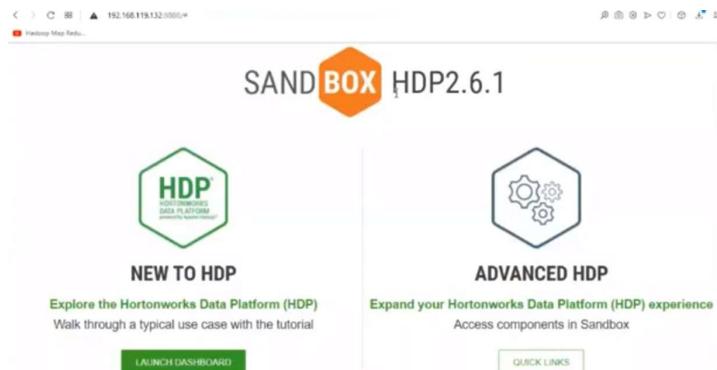
Step 3: Reset Admin account Password

```
[root@sandbox ~]# ambari-admin-password-reset  
Please set the password for admin:  
Please retype the password for admin:  
  
The admin password has been set.  
Restarting ambari-server to make the password change effective...  
  
Using python /usr/bin/python  
Restarting ambari-server  
Waiting for server stop...  
Ambari Server stopped  
Ambari Server running with administrator privileges.  
Organizing resource files at /var/lib/ambari-server/resources...  
Ambari database consistency check started...  
Server PID at: /var/run/ambari-server/ambari-server.pid  
Server out at: /var/log/ambari-server/ambari-server.out  
Server log at: /var/log/ambari-server/ambari-server.log  
Waiting for server start.....
```

Server listening on 8080 and shell login is complete.

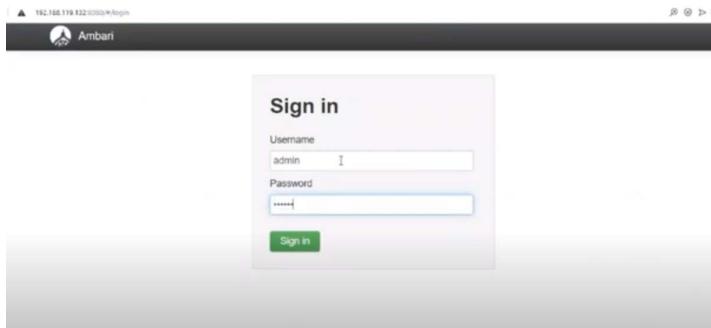
```
The admin password has been set.  
Restarting ambari-server to make the password change effective...  
  
Using python /usr/bin/python  
Restarting ambari-server  
Waiting for server stop...  
Ambari Server stopped  
Ambari Server running with administrator privileges.  
Organizing resource files at /var/lib/ambari-server/resources...  
Ambari database consistency check started...  
Server PID at: /var/run/ambari-server/ambari-server.pid  
Server out at: /var/log/ambari-server/ambari-server.out  
Server log at: /var/log/ambari-server/ambari-server.log  
Waiting for server start.....  
Server started listening on 8080  
  
DB configs consistency check: no errors and warnings were found.  
[root@sandbox ~]#
```

To use graphical user interface login to 192.168.119.132:4200

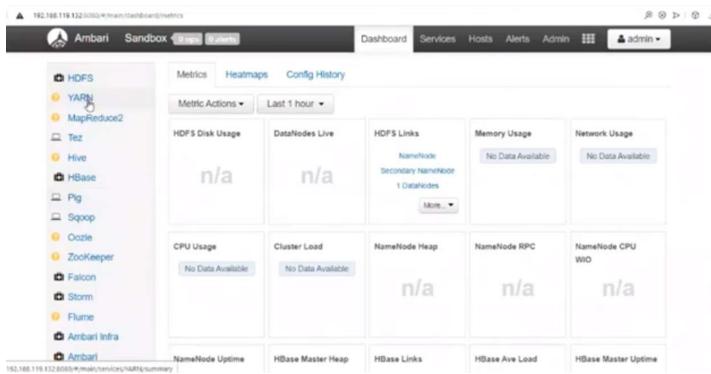


Click on Launch Dashboard

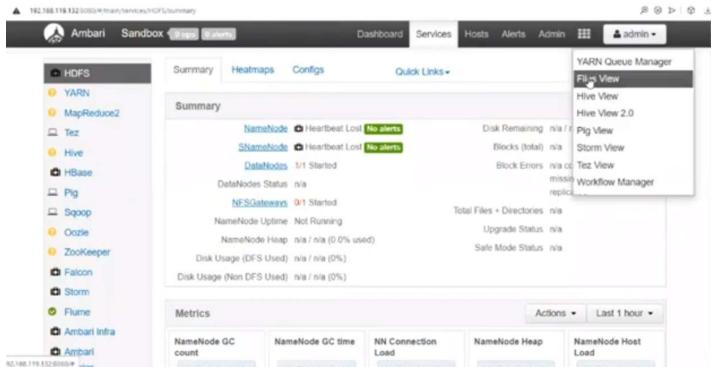
Enter the username and password for admin login.



Below is the Hadoop server.



To view file in HDFS click on HDFS and click on File view.



Commands:

- 1) To view root folder file from terminal use command `hdfs dfs -ls /user` and press enter.
It will display all the files in the root user that we see in UI(Screenshot 2).
ls: This command is used to list all the files

```
DB configs consistency check: no errors and warnings were found.
[root@sandbox ~]# hdfs dfs -ls /user
Found 13 items
drwxr-xr-x  - admin      hdfs      0 2017-04-19 19:09 /user/admin
drwxrwx---  - ambari-qa  hdfs      0 2017-04-19 18:48 /user/ambari-qa
drwxr-xr-x  - amy_ds     hdfs      0 2017-04-19 19:04 /user/amy_ds
drwxr-xr-x  - hbase      hdfs      0 2017-04-19 18:48 /user/hbase
drwxr-xr-x  - hcat       hdfs      0 2017-04-19 18:51 /user/hcat
drwxr-xr-x  - hive        hdfs      0 2017-04-19 19:08 /user/hive
drwxr-xr-x  - holger_gov hdfs      0 2017-04-19 19:05 /user/holger_gov
drwxrwxr-x  - livy        hdfs      0 2017-04-19 18:49 /user/livy
drwxr-xr-x  - maria_dev   hdfs      0 2017-04-19 18:58 /user/maria_dev
drwxrwxr-x  - oozie       hdfs      0 2017-04-19 18:52 /user/oozie
drwxr-xr-x  - raj_ops    hdfs      0 2017-04-19 19:06 /user/raj_ops
drwxrwxr-x  - spark       hdfs      0 2017-04-19 18:49 /user/spark
drwxr-xr-x  - zeppelin   hdfs      0 2017-04-19 18:49 /user/zeppelin
[root@sandbox ~]#
```

Name >	Size >	Last Modified >	Owner >
admin	--	2017-04-20 00:39	admin
ambari-qa	--	2017-04-20 00:18	ambari-qa
amy_ds	--	2017-04-20 00:34	amy_ds

2) mkdir: To create a directory.

Create a folder in Hadoop directory. Type command `hdfs dfs -mkdir /bigdatatest` and enter. After it execute the command, we will see whether it is created folder in UI.

```
[root@sandbox bigdata]# hdfs dfs -mkdir /bigdatatest
[root@sandbox bigdata]#
```

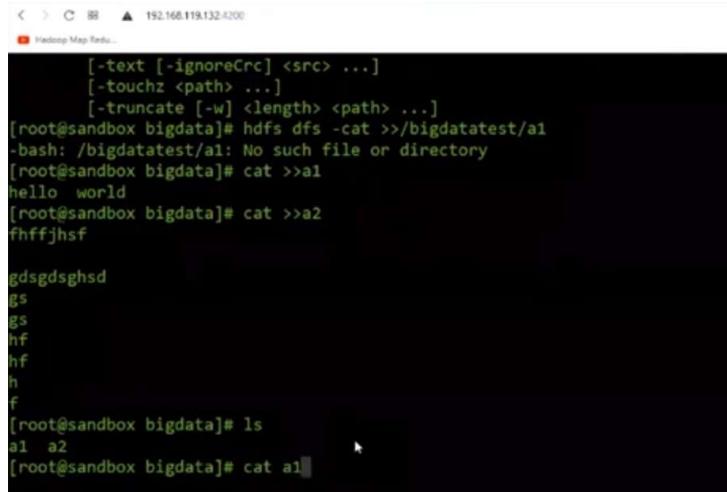
Name >	Size >	Last Modified >	Owner >
app-logs	--	2017-04-20 00:38	yarn
apps	--	2017-04-20 00:25	hdfs
ats	--	2017-04-20 00:18	yarn
bigdatatest	--	2021-06-30 20:31	root
demo	--	2017-04-20 00:33	hdfs
hdp	--	2017-04-20 00:18	hdfs

3) Create a file in local directory

Cat: Create a file.

Cat>>

To terminate press ctrl+d



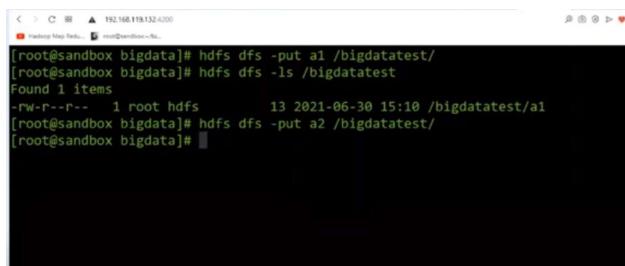
```
< > C B 192.168.119.132:2200
HDFS Map Redu...
[-text [-ignoreCrc] <src> ...]
[-touchz <path> ...]
[-truncate [-w] <length> <path> ...]
[root@sandbox bigdata]# hdfs dfs -cat >/bigdatatest/a1
-bash: /bigdatatest/a1: No such file or directory
[root@sandbox bigdata]# cat >>a1
hello world
[root@sandbox bigdata]# cat >>a2
fhhhjhsf

gdsgdsgs
gs
gs
hf
hf
h
f
[root@sandbox bigdata]# ls
a1 a2
[root@sandbox bigdata]# cat a1
```

4) To upload files/directory to from local to HDFS

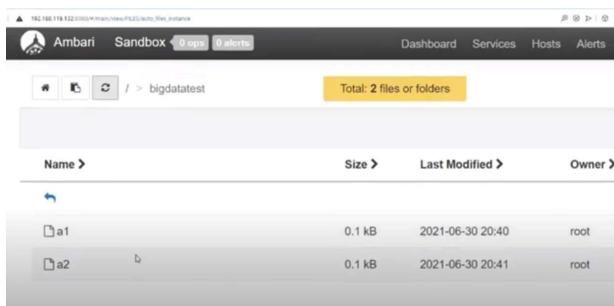
Put: to move a local file or directories into the distributed file system

Command: hdfs dfs -put a1 /bigdatatest/ and hdfs dfs -put a2 /bigdatatest/ will upload both the files.



```
< > C B 192.168.119.132:2200
HDFS Map Redu...
[root@sandbox bigdata]# hdfs dfs -put a1 /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -ls /bigdatatest
Found 1 items
-rw-r--r-- 1 root hdfs 13 2021-06-30 15:10 /bigdatatest/a1
[root@sandbox bigdata]# hdfs dfs -put a2 /bigdatatest/
[root@sandbox bigdata]#
```

Refresh the user interface we can see both the files.



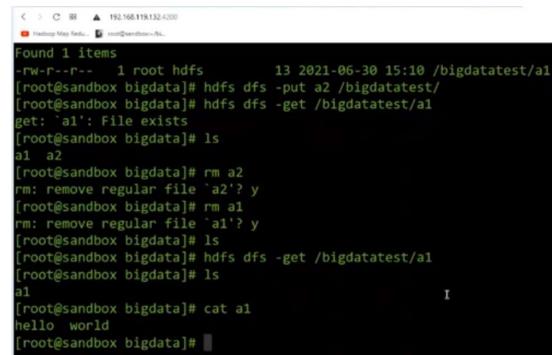
The screenshot shows the Ambari UI with the "File Browser" tab selected. The path is set to "/bigdatatest". A message box indicates "Total: 2 files or folders". Below is a table listing the contents:

Name	Size	Last Modified	Owner
a1	0.1 kB	2021-06-30 20:40	root
a2	0.1 kB	2021-06-30 20:41	root

5) To download files/directories to from hdfs to local

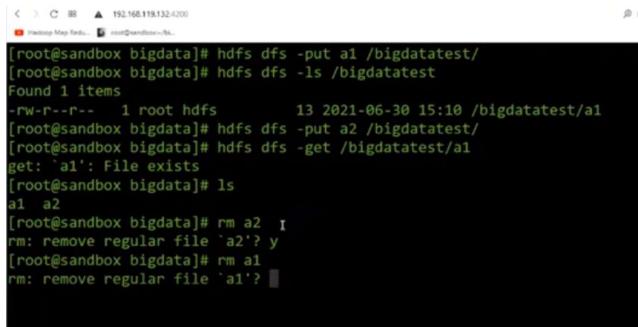
Get: To copy files/folders from hdfs store to local file system.

Command: hdfs dfs -get /bigdatatest/a1 and hdfs dfs -get /bigdatatest/ a2 will upload both the files.



```
< > C 192.168.119.132:4200
Hadoop Map Reduce... root@ sandbox ~%.
Found 1 items
-rw-r--r-- 1 root hdfs 13 2021-06-30 15:10 /bigdatatest/a1
[root@ sandbox bigdata]# hdfs dfs -put a2 /bigdatatest/
[root@ sandbox bigdata]# hdfs dfs -get /bigdatatest/a1
get: 'a1': File exists
[root@ sandbox bigdata]# ls
a1 a2
[root@ sandbox bigdata]# rm a2
rm: remove regular file 'a2'? y
[root@ sandbox bigdata]# rm a1
rm: remove regular file 'a1'? y
[root@ sandbox bigdata]# ls
[root@ sandbox bigdata]# hdfs dfs -get /bigdatatest/a1
[root@ sandbox bigdata]# ls
a1
[root@ sandbox bigdata]# cat a1
Hello World
[root@ sandbox bigdata]#
```

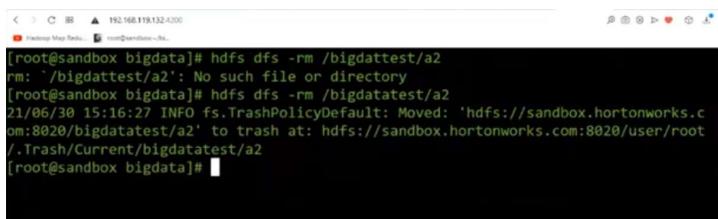
- 6) To remove file from local use rm command



```
< > C 192.168.119.132:4200
Hadoop Map Reduce... root@ sandbox ~%.
[root@ sandbox bigdata]# hdfs dfs -put a1 /bigdatatest/
[root@ sandbox bigdata]# hdfs dfs -ls /bigdatatest
Found 1 items
-rw-r--r-- 1 root hdfs 13 2021-06-30 15:10 /bigdatatest/a1
[root@ sandbox bigdata]# hdfs dfs -put a2 /bigdatatest/
[root@ sandbox bigdata]# hdfs dfs -get /bigdatatest/a1
get: 'a1': File exists
[root@ sandbox bigdata]# ls
a1 a2
[root@ sandbox bigdata]# rm a2
rm: remove regular file 'a2'? y
[root@ sandbox bigdata]# rm a1
rm: remove regular file 'a1'? y
[root@ sandbox bigdata]#
```

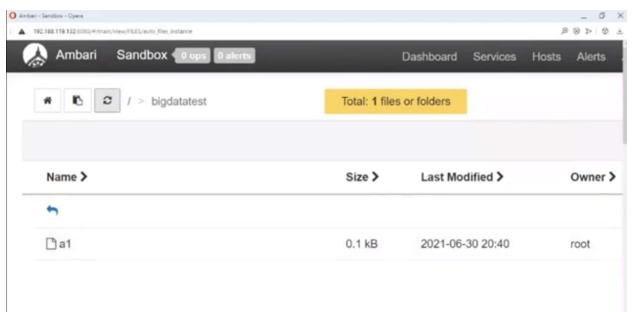
- 7) To remove file from Hadoop directory

Command: hdfs dfs -rm a2 /gibdatatest/a2



```
< > C 192.168.119.132:4200
Hadoop Map Reduce... root@ sandbox ~%.
[root@ sandbox bigdata]# hdfs dfs -rm /bigdatatest/a2
rm: '/bigdatatest/a2': No such file or directory
[root@ sandbox bigdata]# hdfs dfs -rm /bigdatatest/a2
21/06/30 15:16:27 INFO fs.TrashPolicyDefault: Moved: 'hdfs://sandbox.hortonworks.com:8020/bigdatatest/a2' to trash at: hdfs://sandbox.hortonworks.com:8020/user/root/.Trash/Current/bigdatatest/a2
[root@ sandbox bigdata]#
```

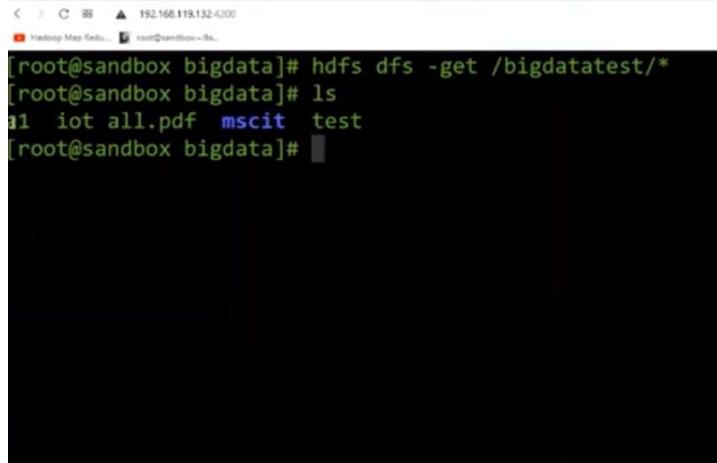
Now refresh the UI and the file will be deleted.



Name	Size	Last Modified	Owner
a1	0.1 kB	2021-06-30 20:40	root

8) To download all the files from hdfs to local

Command: hdfs dfs -get /bigdatatest/*

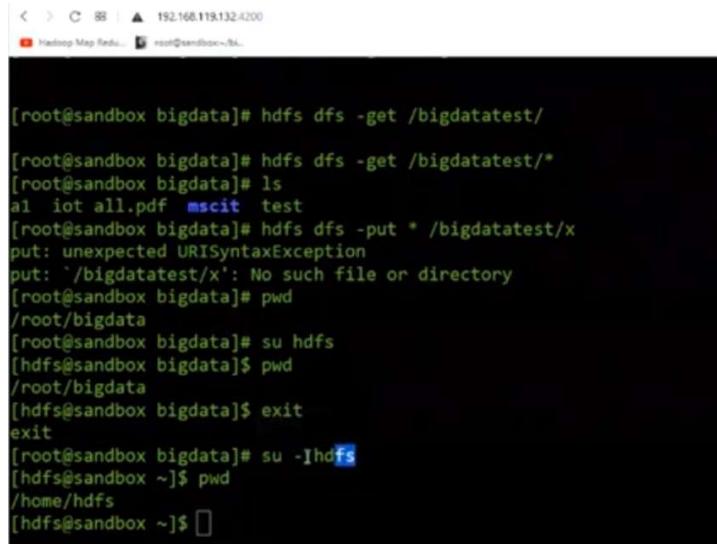


```
< > C 88 |▲ 192.168.119.132:4200
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/*
[root@sandbox bigdata]# ls
a1 iot all.pdf mscit test
[root@sandbox bigdata]#
```

9) Change user and directory and change user only

Command: su – hdfs (Change user and directory)

Su hdfs (change user only)



```
< > C 88 |▲ 192.168.119.132:4200
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/
[root@sandbox bigdata]# hdfs dfs -get /bigdatatest/*
[root@sandbox bigdata]# ls
a1 iot all.pdf mscit test
[root@sandbox bigdata]# hdfs dfs -put * /bigdatatest/x
put: unexpected URISyntaxException
put: '/bigdatatest/x': No such file or directory
[root@sandbox bigdata]# pwd
/root/bigdata
[root@sandbox bigdata]# su hdfs
[hdfs@sandbox bigdata]$ pwd
/root/bigdata
[hdfs@sandbox bigdata]$ exit
exit
[root@sandbox bigdata]# su -Ihdfs
[hdfs@sandbox ~]$ pwd
/home/hdfs
[hdfs@sandbox ~]$
```

Practical No. 2

Aim:- Implement word count / frequency programs using MapReduce

Map Reduce as two component Map and Reduce.

Java program:

write program save as WordCount.java

```
///////////
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 WordCount {

    public static class TokenizerMapper extends Mapper<Object, Text, Text, IntWritable>{
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();
        public void map(Object key, Text value, Context context
        ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {://"This is the output is the"
                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
        {//is,3
        int sum = 0;
        for (IntWritable val : values) {
        sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
        }
    }
}
```

```

}

public static void main(String[] args) throws Exception {
    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);
}
}
///////////

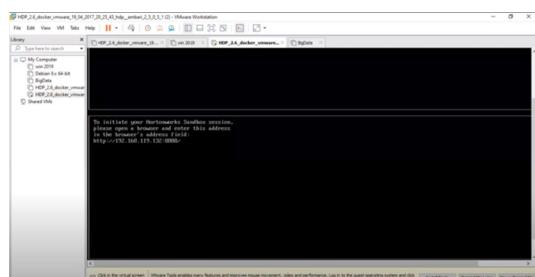
```

Text File:

Hello World

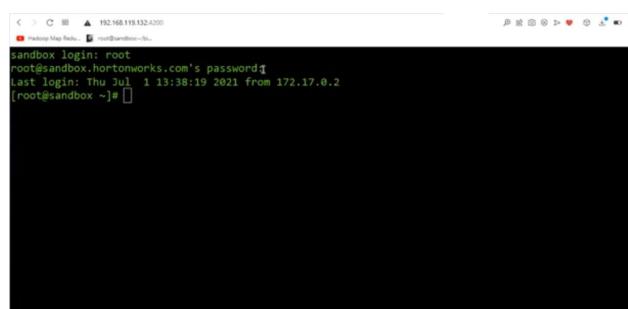
This is the output is the

Start the server (Horton Sandbox)



Open the terminal with 192.168.119.132/4200

Enter the login: root and the password and enter



Create a folder in local directory.

Command: mkdir mscitp2

Change the directory cd mscitp2

```
 sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]#
```

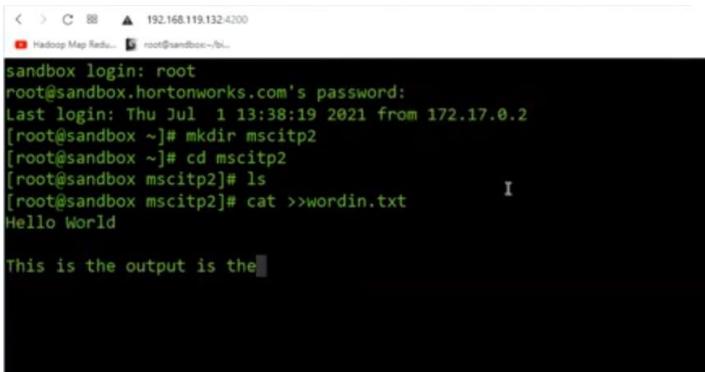
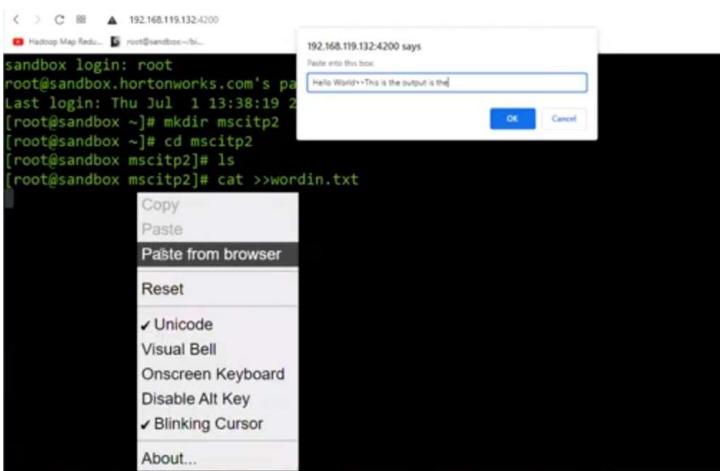
Now create input file

Command: cat >> wordin.txt

Paste the text by right clicking on terminal

Hello World

This is the output is the



To remove the extra space type command

```
vi wordin.txt
```

After removing the extra space check the content of the file

```
cat wordin.txt
```

A terminal window titled 'Hadoop Map Reduce' with the IP address '192.168.119.132:4200'. The session shows the following commands and output:

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Thu Jul  1 13:38:19 2021 from 172.17.0.2
[root@sandbox ~]# mkdir mscitp2
[root@sandbox ~]# cd mscitp2
[root@sandbox mscitp2]# ls
[root@sandbox mscitp2]# cat >>wordin.txt
Hello World

This is the output is the
[root@sandbox mscitp2]# vi wordin.txt
[root@sandbox mscitp2]# cat wordin.txt
Hello World
This is the output is the
[root@sandbox mscitp2]#
```

Create another file wordcount.java

A terminal window titled 'Hadoop Map Reduce' with the IP address '192.168.119.132:4200'. The session shows the following commands and output:

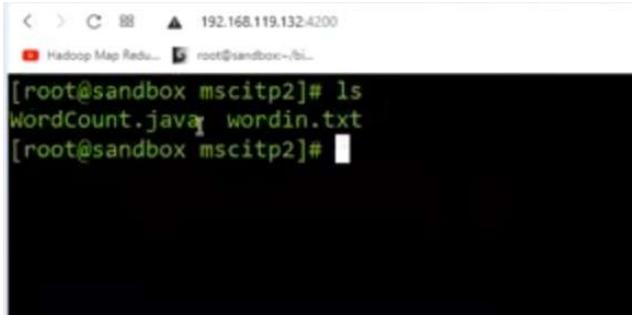
```
This is the output is the
[root@sandbox mscitp2]# vi wordin.txt
[root@sandbox mscitp2]# cat wordin.txt
Hello World
This is the output is the
[root@sandbox mscitp2]# cat >>WordCount.java
```

Paste the java code.

A terminal window titled 'Hadoop Map Reduce' with the IP address '192.168.119.132:4200'. A context menu is open over the Java code, showing options like 'Copy', 'Paste', and 'Paste from browser'. The 'Paste from browser' option is highlighted.

Press control d to save the file

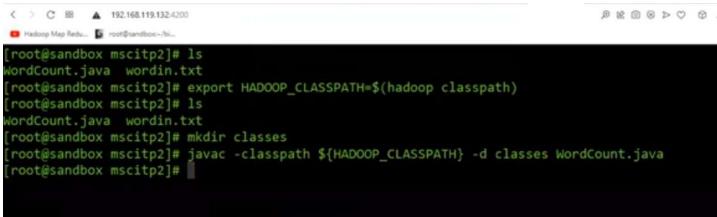
Check both the files create with command ls



```
[root@sandbox mscitp2]# ls
WordCount.java wordin.txt
[root@sandbox mscitp2]#
```

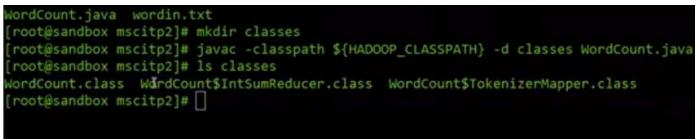
Now, to compile the java file

```
export HADOOP_CLASSPATH=$(hadoop classpath)
mkdir classes (To keep the compile files)
javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java
```



```
[root@sandbox mscitp2]# ls
WordCount.java wordin.txt
[root@sandbox mscitp2]# export HADOOP_CLASSPATH=$(hadoop classpath)
[root@sandbox mscitp2]# ls
WordCount.java wordin.txt
[root@sandbox mscitp2]# mkdir classes
[root@sandbox mscitp2]# javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java
[root@sandbox mscitp2]#
```

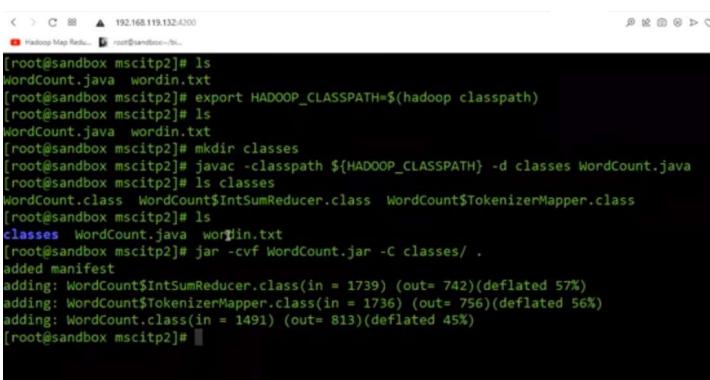
Check class files are created with command ls classes



```
WordCount.java wordin.txt
[root@sandbox mscitp2]# mkdir classes
[root@sandbox mscitp2]# javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java
[root@sandbox mscitp2]# ls classes
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class
[root@sandbox mscitp2]#
```

Now we have to bind all the class into single jar file with below command

```
jar -cvf WordCount.jar -C classes/ .
```



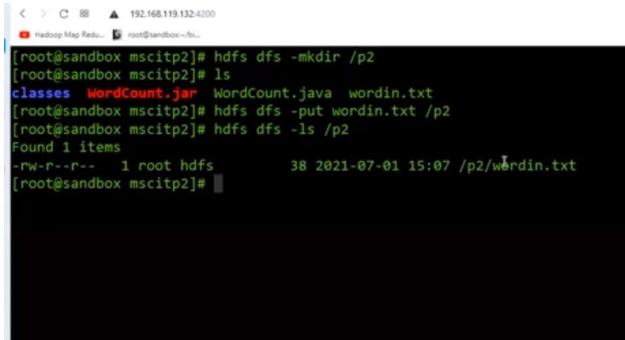
```
[root@sandbox mscitp2]# ls
WordCount.java wordin.txt
[root@sandbox mscitp2]# export HADOOP_CLASSPATH=$(hadoop classpath)
[root@sandbox mscitp2]# ls
WordCount.java wordin.txt
[root@sandbox mscitp2]# mkdir classes
[root@sandbox mscitp2]# javac -classpath ${HADOOP_CLASSPATH} -d classes WordCount.java
[root@sandbox mscitp2]# ls classes
WordCount.class WordCount$IntSumReducer.class WordCount$TokenizerMapper.class
[root@sandbox mscitp2]# ls
classes WordCount.java wordin.txt
[root@sandbox mscitp2]# jar -cvf WordCount.jar -C classes/ .
added manifest
adding: WordCount$IntSumReducer.class(in = 1739) (out= 742)(deflated 57%)
adding: WordCount$TokenizerMapper.class(in = 1736) (out= 756)(deflated 56%)
adding: WordCount.class(in = 1491) (out= 813)(deflated 45%)
[root@sandbox mscitp2]#
```

Run ls command we can see jar file is created.



```
[root@sandbox mscitp2]# ls
classes WordCount.jar WordCount.java wordin.txt
[root@sandbox mscitp2]#
```

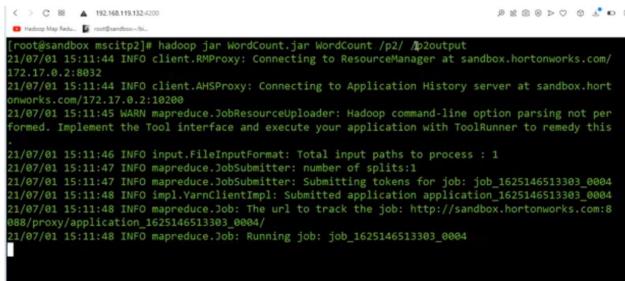
wordin.txt should be present in word directory of hdfs. So we need to upload wordin.txt file.



```
[root@sandbox mscitp2]# hdfs dfs -mkdir /p2
[root@sandbox mscitp2]# ls
classes WordCount.jar WordCount.java wordin.txt
[root@sandbox mscitp2]# hdfs dfs -put wordin.txt /p2
[root@sandbox mscitp2]# hdfs dfs -ls /p2
Found 1 items
-rw-r--r-- 1 root hdfs      38 2021-07-01 15:07 /p2/wordin.txt
[root@sandbox mscitp2]#
```

We need to put the final output p2output.

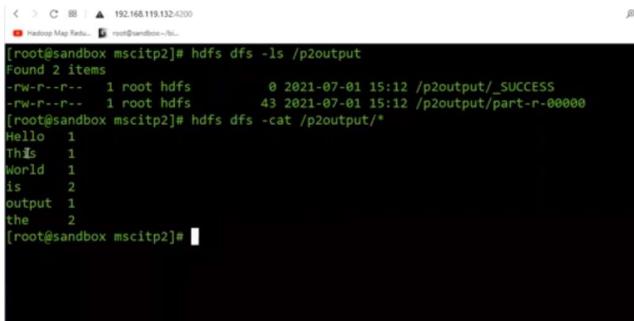
hadoop jar WordCount.jar WordCount /p2/ /p2output



```
[root@sandbox mscitp2]# hadoop jar WordCount.jar WordCount /p2/ /p2output
21/07/01 15:11:44 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8082
21/07/01 15:11:44 INFO client.AHSProxy: Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
21/07/01 15:11:45 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the ToolRunner to remedy this
21/07/01 15:11:46 INFO input.FileInputFormat: Total input paths to process : 1
21/07/01 15:11:47 INFO mapreduce.JobSubmitter: number of splits:1
21/07/01 15:11:47 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1625146513303_0004
21/07/01 15:11:48 INFO impl.YarnClientImpl: Submitted application application_1625146513303_0004
21/07/01 15:11:48 INFO mapreduce.Job: The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1625146513303_0004/
21/07/01 15:11:48 INFO mapreduce.Job: Running job: job_1625146513303_0004
21/07/01 15:11:48 INFO mapreduce.Job: Job: job_1625146513303_0004
```

Print the content of the output file

Command: hdfs dfs -cat /p2output/*



```
[root@sandbox mscitp2]# hdfs dfs -ls /p2output
Found 2 items
-rw-r--r-- 1 root hdfs      0 2021-07-01 15:12 /p2output/_SUCCESS
-rw-r--r-- 1 root hdfs     43 2021-07-01 15:12 /p2output/part-r-00000
[root@sandbox mscitp2]# hdfs dfs -cat /p2output/*
Hello 1
This 1
World 1
is 2
output 1
the 2
[root@sandbox mscitp2]#
```

Ctrl + l to clear the screen.

vi filename.txt= this command will create/ open filename.txt

two modes of vi editor

- 1) Insert mode – i (press i key)
- 2) Command mode – esc key

:wq is to save and exit

Practical No. 3

Aim:- Implement an MapReduce program that processes a weather dataset.

Java program:

MyMaxMin.java

```
//////////  
// importing Libraries  
import java.io.IOException;  
import java.util.Iterator;  
import org.apache.hadoop.fs.Path;  
import org.apache.hadoop.io.LongWritable;  
import org.apache.hadoop.io.Text;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;  
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;  
import org.apache.hadoop.mapreduce.Job;  
import org.apache.hadoop.mapreduce.Mapper;  
import org.apache.hadoop.mapreduce.Reducer;  
import org.apache.hadoop.conf.Configuration;  
  
public class MyMaxMin {  
    // Mapper  
    /*MaxTemperatureMapper class is static  
     * and extends Mapper abstract class  
     * having four Hadoop generics type  
     * LongWritable, Text, Text, Text.  
     */  
    public static class MaxTemperatureMapper extends  
        Mapper<LongWritable, Text, Text, Text> {
```

```
/**  
 * @method map  
 * This method takes the input as a text data type.  
 * Now leaving the first five tokens, it takes  
 * 6th token is taken as temp_max and  
 * 7th token is taken as temp_min. Now  
 * temp_max > 30 and temp_min < 15 are  
 * passed to the reducer.  
 */
```

```
// the data in our data set with  
// this value is inconsistent data  
public static final int MISSING = 9999;
```

```
@Override  
public void map(LongWritable arg0, Text Value, Context context)  
throws IOException, InterruptedException {  
  
    // Convert the single row(Record) to  
    // String and store it in String  
    // variable name line  
  
    String line = Value.toString();  
  
    // Check for the empty line  
    if (!(line.length() == 0)) {  
  
        // from character 6 to 14 we have  
        // the date in our dataset  
        String date = line.substring(6, 14);
```

```

        // similarly we have taken the maximum
        // temperature from 39 to 45 characters
        float temp_Max = Float.parseFloat(line.substring(39,
45).trim());

        // similarly we have taken the minimum
        // temperature from 47 to 53 characters

        float temp_Min = Float.parseFloat(line.substring(47,
53).trim());

        // if maximum temperature is
        // greater than 30, it is a hot day
        if (temp_Max > 30.0) {

            // Hot day
            context.write(new Text("The Day is Hot Day :" + date),
new
Text(String.valueOf(temp_Max)));
        }

        // if the minimum temperature is
        // less than 15, it is a cold day
        if (temp_Min < 15) {

            // Cold day
            context.write(new Text("The Day is Cold Day :" +
date),
new Text(String.valueOf(temp_Min)));
        }
    }
}

```

```

        }

    }

// Reducer

/*MaxTemperatureReducer class is static
and extends Reducer abstract class
having four Hadoop generics type
Text, Text, Text, Text.

*/
//The Day is Cold Day :20150101 ,-21.8
public static class MaxTemperatureReducer extends
    Reducer<Text, Text, Text, Text> {

    /**
     * @method reduce
     * This method takes the input as key and
     * list of values pair from the mapper,
     * it does aggregation based on keys and
     * produces the final context.
    */

    public void reduce(Text Key, Iterator<Text> Values, Context context)
        throws IOException, InterruptedException {
        // putting all the values in
        // temperature variable of type String
        String temperature = Values.next().toString();
        context.write(Key, new Text(temperature));
    }

}

```

```
/**  
 * @method main  
 * This method is used for setting  
 * all the configuration properties.  
 * It acts as a driver for map-reduce  
 * code.  
 */  
  
public static void main(String[] args) throws Exception {  
  
    // reads the default configuration of the  
    // cluster from the configuration XML files  
    Configuration conf = new Configuration();  
  
    // Initializing the job with the  
    // default configuration of the cluster  
    Job job = new Job(conf, "weather example");  
  
    // Assigning the driver class name  
    job.setJarByClass(MyMaxMin.class);  
  
    // Key type coming out of mapper  
    job.setMapOutputKeyClass(Text.class);  
  
    // value type coming out of mapper  
    job.setMapOutputValueClass(Text.class);  
  
    // Defining the mapper class name  
    job.setMapperClass(MaxTemperatureMapper.class);  
  
    // Defining the reducer class name
```

```
job.setReducerClass(MaxTemperatureReducer.class);

// Defining input Format class which is
// responsible to parse the dataset
// into a key value pair
job.setInputFormatClass(TextInputFormat.class);

// Defining output Format class which is
// responsible to parse the dataset
// into a key value pair
job.setOutputFormatClass(TextOutputFormat.class);

// setting the second argument
// as a path in a path variable
Path outputPath = new Path(args[1]);

// Configuring the input path
// from the filesystem into the job
FileInputFormat.addInputPath(job, new Path(args[0]));

// Configuring the output path from
// the filesystem into the job
FileOutputFormat.setOutputPath(job, new Path(args[1]));

// deleting the context path automatically
// from hdfs so that we don't have
// to delete it explicitly
OutputPath.getFileSystem(conf).delete(OutputPath);

// exiting the job only if the
```

```

        // flag value becomes false
        System.exit(job.waitForCompletion(true) ? 0 : 1);

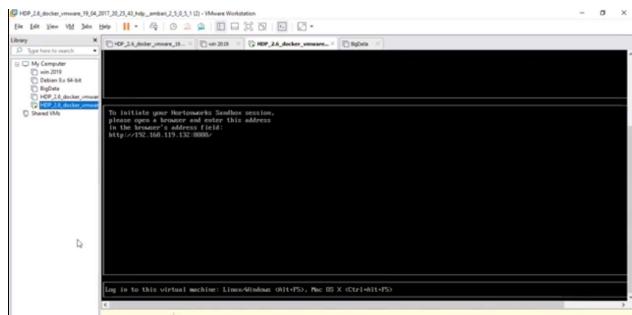
    }

}

///////////

```

Start the server



Open the terminal with 192.168.119.132/4200

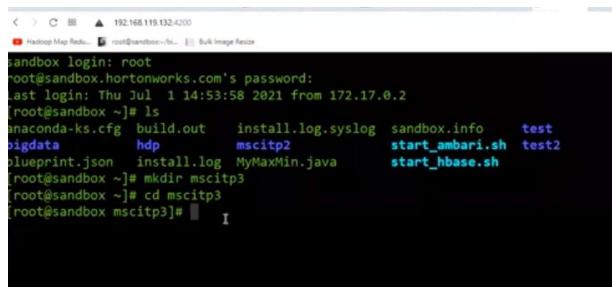
Enter the login: root and the password and enter



Create a folder in local directory.

Command: mkdir mscitp3

Change the directory cd mscitp3



Now create input file

Command: cat >> weatherin2.txt

Paste the weather dataset by right clicking on terminal

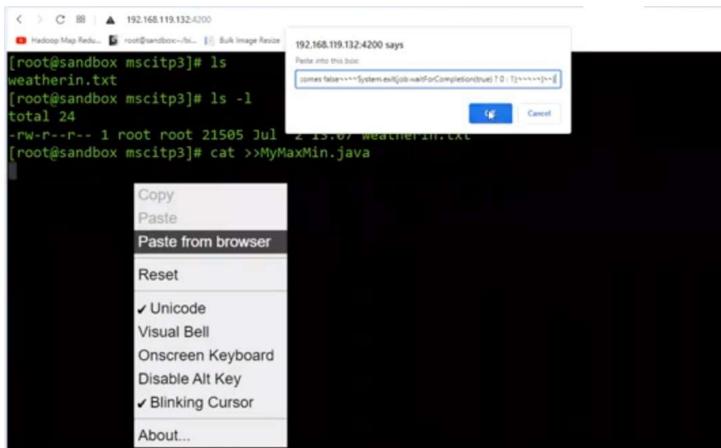
Ctrl d will save the file

Run command ls to see the file.

Create java file

Command: cat >>MyMaxMin.java

Paste the java code and ctrl d to save the file



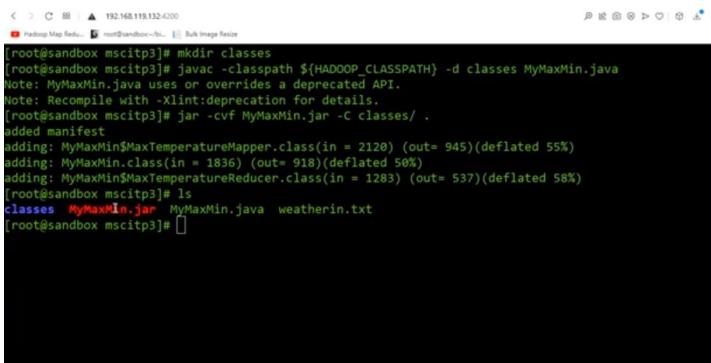
export HADOOP_CLASSPATH=\$(hadoop classpath) //compile and to create jar file

mkdir classes

javac -classpath \${HADOOP_CLASSPATH} -d classes MyMaxMin.java

After compile need to create a jar file

jar -cvf MyMaxMin.jar -C classes/ .



Now, put weatherin.txt in hdfs

Before that create a folder

Command: hdfs dfs -mkdir /p3input123

Then run command: hdfs dfs -put weatherin2.txt /p3input123

hadoop jar MyMaxMin.jar MyMaxMin /p3inputw /output123

```
< > C □ | ▲ 192.168.119.132:4200
Hadoop Map Reduce [root@sandbox mscitp3]# Bulk Image Resize
[root@sandbox mscitp3]# hdfs dfs -mkdir /p3input123
[root@sandbox mscitp3]# hdfs dfs -put weatherin2.txt /p3input123
[21/07/02 13:22:35 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8082
[21/07/02 13:22:35 INFO client.AHSProxy: Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
[21/07/02 13:22:36 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
[21/07/02 13:22:36 INFO input.FileInputFormat: Total input paths to process : 1
[21/07/02 13:22:37 INFO mapreduce.JobSubmitter: number of splits:1
```

Check outfile is created

Command: hdfs dfs -ls /output123

```
< > C □ | ▲ 192.168.119.132:4200
Hadoop Map Reduce [root@sandbox mscitp3]# Bulk Image Resize
[root@sandbox mscitp3]# hdfs dfs -ls /output123
ls: '/output123': No such file or directory
[root@sandbox mscitp3]# hdfs dfs -ls /output123
Found 2 items
-rw-r--r-- 1 root hdfs          0 2021-07-02 13:22 /output123/_SUCCESS
-rw-r--r-- 1 root hdfs    1970 2021-07-02 13:22 /output123/part-r-00000
[root@sandbox mscitp3]# 
```

hdfs dfs -cat /output123/*

```
< > C □ | ▲ 192.168.119.132:4200
Hadoop Map Reduce [root@sandbox mscitp3]# Bulk Image Resize
[root@sandbox mscitp3]# hdfs dfs -ls /output123
ls: '/output123': No such file or directory
[root@sandbox mscitp3]# hdfs dfs -ls /output123
Found 2 items
-rw-r--r-- 1 root hdfs          0 2021-07-02 13:22 /output123/_SUCCESS
-rw-r--r-- 1 root hdfs    1970 2021-07-02 13:22 /output123/part-r-00000
[root@sandbox mscitp3]# hdfs dfs -cat /output123/*
The Day is Cold Day :20170101 -4.7
The Day is Cold Day :20170102 -9.2
The Day is Cold Day :20170103 -10.7
The Day is Cold Day :20170104 -10.1
The Day is Cold Day :20170105 -20.0
The Day is Cold Day :20170106 -23.7
The Day is Cold Day :20170107 -22.1
The Day is Cold Day :20170108 -21.2
The Day is Cold Day :20170109 -17.7
The Day is Cold Day :20170110 -21.0
The Day is Cold Day :20170111 -22.7
The Day is Cold Day :20170112 -20.2
The Day is Cold Day :20170113 -19.0
The Day is Cold Day :20170114 -25.5
```

Practical No. 4

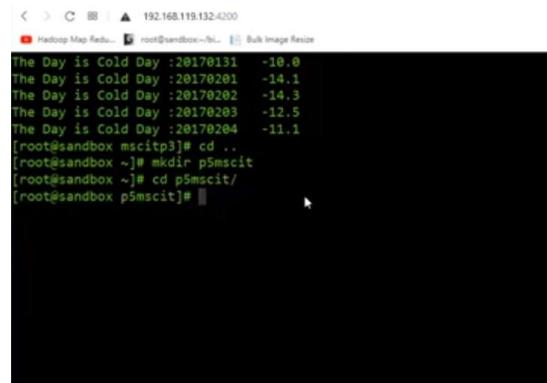
Aim:- Implement the program using Pig.

Dataset:

```
001,Rajiv,Reddy,21,9848022337,Hyderabad  
002,siddarth,Battacharya,22,9848022338,Kolkata  
003,Rajesh,Khanna,22,9848022339,Delhi  
004,Preethi,Agarwal,21,9848022330,Pune  
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar  
006,Archana,Mishra,23,9848022335,Chennai  
007,Komal,Nayak,24,9848022334,trivendram  
008,Bharathi,Nambiyar,24,9848022333,Chennai  
#student.txt
```

create a directory and get into that directory

Command: mkdir p5mscit

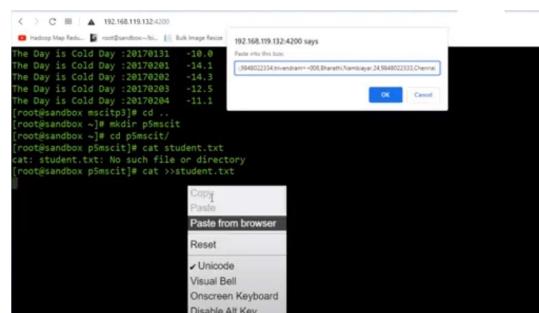


```
The Day is Cold Day :20170131 -10.0  
The Day is Cold Day :20170201 -14.1  
The Day is Cold Day :20170202 -14.3  
The Day is Cold Day :20170203 -12.5  
The Day is Cold Day :20170204 -11.1  
[root@sandbox mscitp3]# cd ..  
[root@sandbox ~]# mkdir p5mscit  
[root@sandbox ~]# cd p5mscit/  
[root@sandbox p5mscit]#
```

Create a file

Command: cat >>student.txt

Right click and paste the text



```

< > C □ ▲ 192.168.119.132:4200
Hadoop Map Reduce root@ sandbox ~ % Bulk Image Resize
The Day is Cold Day :20170201 -14.1
The Day is Cold Day :20170202 -14.3
The Day is Cold Day :20170203 -12.5
The Day is Cold Day :20170204 -11.1
[root@ sandbox msctip3]# cd ..
[root@ sandbox ~]# mkdir p5msc1t
[root@ sandbox ~]# cd p5msc1t/
[root@ sandbox p5msc1t]# cat student.txt
cat: student.txt: No such file or directory
[root@ sandbox p5msc1t]# cat >>student.txt
001,Rajiv,Reddy,21,98488022337,Hyderabad
002,Siddarth,Battacharya,22,98488022338,Kolkata
003,Rajesh,Khanna,22,98488022339,Delhi
004,Preethi,Agarwal,21,98488022330,Pune
005,Trupti,Mohanthy,23,98488022336,Bhubaneshwar
006,Archana,Mishra,23,98488022335,Chennai
007,Komal,Nayak,24,98488022334,trivendram
008,Bharathi,Nambiayar,24,98488022333,Chennai

```

Remove the space with vi editor

Command: vi student.txt and press i for insert mode

After editing: wq and enter

Print the content and see the text

```

< > C □ ▲ 192.168.119.132:4200
Hadoop Map Reduce root@ sandbox ~ % Bulk Image Resize
[root@ sandbox p5msc1t]# cat student.txt
001,Rajiv,Reddy,21,98488022337,Hyderabad
002,siddarth,Battacharya,22,98488022338,Kolkata
003,Rajesh,Khanna,22,98488022339,Delhi
004,Preethi,Agarwal,21,98488022330,Pune
005,Trupti,Mohanthy,23,98488022336,Bhubaneshwar
006,Archana,Mishra,23,98488022335,Chennai
007,Komal,Nayak,24,98488022334,trivendram
008,Bharathi,Nambiayar,24,98488022333,Chennai
[root@ sandbox p5msc1t]#

```

Create a program file

```
//////////////////script start
student = LOAD 'student.txt' USING PigStorage(',')
as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray);
```

```
student_order = ORDER student BY age DESC;
student_limit = LIMIT student_order 4;
```

```
Dump student_limit;
/////////script end
```

```

< > C □ ▲ 192.168.119.132:4200
Hadoop Map Reduce root@ sandbox ~ % Bulk Image Resize
[root@ sandbox p5msc1t]# cat student.txt
001,Rajiv,Reddy,21,98488022337,Hyderabad
002,siddarth,Battacharya,22,98488022338,Kolkata
003,Rajesh,Khanna,22,98488022339,Delhi
004,Preethi,Agarwal,21,98488022330,Pune
005,Trupti,Mohanthy,23,98488022336,Bhubaneshwar
006,Archana,Mishra,23,98488022335,Chennai
007,Komal,Nayak,24,98488022334,trivendram
008,Bharathi,Nambiayar,24,98488022333,Chennai
[root@ sandbox p5msc1t]# cat >>program.pig
student = LOAD 'student.txt' USING PigStorage(',');
as (id:int, firstname:chararray, lastname:chararray, age:int, phone:chararray, city:chararray);

student_order = ORDER student BY age DESC;

student_limit = LIMIT student_order 4;

Dump student_limit;
[root@ sandbox p5msc1t]#

```

Upload student on hdfs

Command: hdfs dfs -put student.txt /user/root/

Run the pig program

```
[root@sandbox p5mscit]# hdfs dfs -put student.txt /user/root/
[root@sandbox p5mscit]# pig program.pig
```

Output:

```
< > C ▲ 192.168.119.132:4200
[root@sandbox p5mscit]# Bulk Image Resize
student.txt

Output(s):
Successfully stored 4 records (211 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp81689
1614/tmp500376273"

2021-07-02 13:51:34,079 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1
2021-07-02 13:51:34,079 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapReduceUtil - Total input paths to process : 1
(7,Komal,Nayak,24,9848022334,trivendram)
(8,Bharathi,Nambiyar,24,9848022333,Chennai)
(5,Trupthi,Mohanty,23,9848022336,Bhuwaneshwar)
(6,Archana,Mishra,23,9848022335,Chennai)
2021-07-02 13:51:34,563 [main] INFO org.apache.pig.Main - Pig script completed in 39 seconds and 726 milliseconds (39726 ms)
2021-07-02 13:51:34,575 [main] INFO org.apache.pig.backend.hadoop.executionengine.tez.TezLauncher - Shutting down thread pool
2021-07-02 13:51:34,688 [pool-1-thread-1] INFO org.apache.pig.backend.hadoop.executionengine.tez.TezSessionManager - Shutting down Tez session org.apache.tez.client.TezClient@132aba44
2021-07-02 13:51:34 Shutting down Tez session , sessionName=Piglatin:program.pig, applicationId=application_1625229346715_0004
```

Practical No. 5

Aim:- Implement the application in Hive.

Dataset:

```
001,Rajiv,Reddy,21,9848022337,Hyderabad  
002,siddarth,Battacharya,22,9848022338,Kolkata  
003,Rajesh,Khanna,22,9848022339,Delhi  
004,Preethi,Agarwal,21,9848022330,Pune  
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar  
006,Archana,Mishra,23,9848022335,Chennai  
007,Komal,Nayak,24,9848022334,trivendram  
008,Bharathi,Nambiayar,24,9848022333,Chennai  
#student.txt
```

create a directory and get into that directory

Command: mkdir p6mscit

Create a file

Command: cat >>data.txt

Right click and paste the text

```
[root@sandbox ~]# mkdir p6mscit  
mkdir: cannot create directory 'p6mscit': File exists  
[root@sandbox ~]# cd p6mscit  
[root@sandbox p6mscit]# cat >>data.txt  
001,Rajiv,Reddy,21,9848022337,Hyderabad  
002,siddarth,Battacharya,22,9848022338,Kolkata  
003,Rajesh,Khanna,22,9848022339,Delhi  
004,Preethi,Agarwal,21,9848022330,Pune  
005,Trupthi,Mohanthy,23,9848022336,Bhuwaneshwar  
006,Archana,Mishra,23,9848022335,Chennai  
007,Komal,Nayak,24,9848022334,trivendram  
008,Bharathi,Nambiayar,24,9848022333,Chennai
```

Remove the space with vi editor

Command: vi student.txt and press i for insert mode

After editing: wq and enter

Print the content and see the text

Now start the hive terminal

Command: hive

```

[root@sandbox p6mscit]# vi data.txt
[root@sandbox p6mscit]# hive
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.0.3-8/0/hive-log4j.properties
hive> 

```

Copy paste below command on hive and enter

create table

```

CREATE TABLE IF NOT EXISTS employee ( eid int, fname String,
lname String, age int, contact String, city String)
COMMENT 'Employee details'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;

```

```

[root@sandbox p6mscit]# vi data.txt
[root@sandbox p6mscit]# hive
log4j:WARN No such property [maxFileSize] in org.apache.log4j.DailyRollingFileAppender.

Logging initialized using configuration in file:/etc/hive/2.6.0.3-8/0/hive-log4j.properties
hive> CREATE TABLE IF NOT EXISTS employee ( eid int, fname String,
> lname String, age int, contact String, city String)
> COMMENT 'Employee details'
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> LINES TERMINATED BY '\n'
> STORED AS TEXTFILE;

```

Run command: LOAD DATA LOCAL INPATH 'data.txt' OVERWRITE INTO TABLE employee;

```

hive> LOAD DATA LOCAL INPATH 'data.txt' OVERWRITE INTO TABLE employee;
Loading data to table default.employee
Table default.employee stats: [numFiles=1, numRows=0, totalSize=339, rawDataSize=0]
OK
Time taken: 1.296 seconds
hive> ; 

```

Run the command like select * from employee;

```

Loading data to table default.employee
Table default.employee stats: [numFiles=1, numRows=0, totalSize=339, rawDataSize=0]
OK
Time taken: 1.296 seconds
hive> select * from employee;
OK
1    Rajiv Reddy  21    9848022337    Hyderabad
2    siddarth    Battacharya 22    9848022338    Kolkata
3    Rajesh Khanna 22    9848022339    Delhi
4    Preethi Agarwal 21    9848022330    Pune
5    Trupti Mohanty 23    9848022336    Bhawaneshwar
6    Archana Mishra 23    9848022335    Chennai
7    Komal Nayak   24    9848022334    trivendram
8    Bharathi Nambiyar 24    9848022333    Chennai
Time taken: 0.188 seconds, Fetched: 8 row(s)
hive> select * from employee where age > 23;
OK
7    Komal Nayak   24    9848022334    trivendram
8    Bharathi Nambiyar 24    9848022333    Chennai
Time taken: 0.551 seconds, Fetched: 2 row(s)
hive> 

```

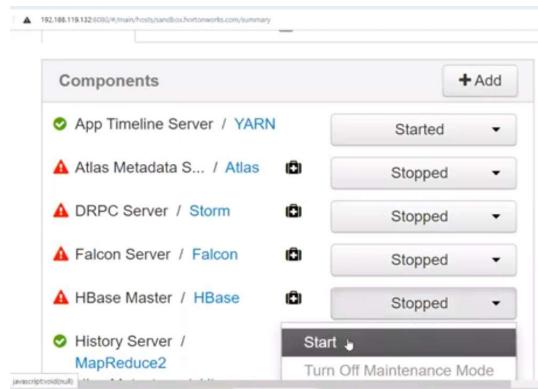
Practical No. 6

Aim:- Implement an application that stores big data in Hbase/ Python

What is HBase?

HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable. It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System.

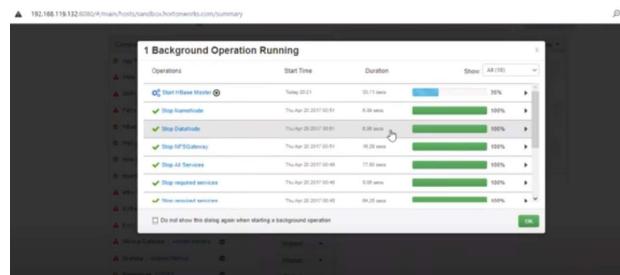
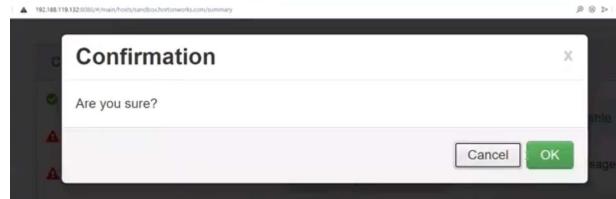
Go to GUI page and start the hbase service.



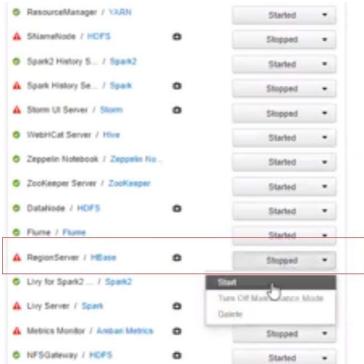
The screenshot shows the Ambari Summary page with a list of components and their statuses:

Component	Status
App Timeline Server / YARN	Started
Atlas Metadata S... / Atlas	Stopped
DRPC Server / Storm	Stopped
Falcon Server / Falcon	Stopped
HBase Master / HBase	Stopped
History Server / MapReduce2	Started

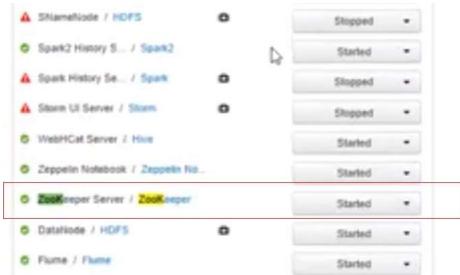
Click on OK to start the service.



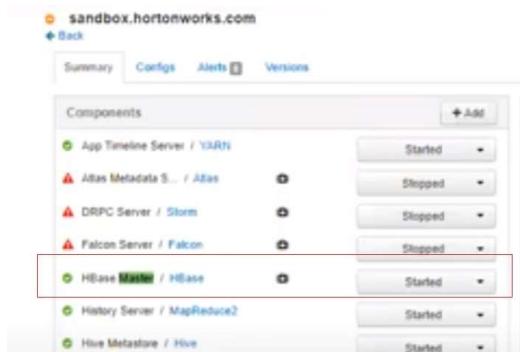
Now we must start region server.



Check zookeeper server is started.



Check hbase and region server are started.





Command: **which application-name** gives directory in which application-name is installed.

Open the shell

192.168.119.132:4200

```
< > C BB | ▲ 192.168.119.132:4200
[root@sandbox ~]# Hadoop Map Reduce
[root@sandbox ~]# root@sandbox:~$ Bulk Image Resize
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Fri Jul  2 13:05:32 2021 from 172.17.0.2
[root@sandbox ~]#
```

Command: hbase shell

It will start the server

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Mon Jul  5 14:46:09 2021 from 172.17.0.2
[root@sandbox ~]# hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.0.3-8, r3307790b5a22cf93100cad0951760718dee5dec7, Sat
Apr  1 21:41:47 UTC 2017
```

Enter the command create 'test', 'cf' and it will create the table

```
sandbox login: root
root@sandbox.hortonworks.com's password:
Last login: Mon Jul  5 14:46:09 2021 from 172.17.0.2
[root@sandbox ~]# hbase shell
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 1.1.2.2.6.0.3-8, r3307790b5a22cf93100cad0951760718dee5dec7, Sat
Apr  1 21:41:47 UTC 2017

hbase(main):001:0> create 'test', 'cf'
0 row(s) in 1.7330 seconds
      I
=> Hbase::Table - test
hbase(main):002:0>
```

Check the table is created with command

List- It will list all the tables created.

```

< > C ⌂ ▲ 192.168.119.132:6200
Hbase Map Reduce... wordCountWordCount... Bulk Image Reader...
Apr 1 21:41:47 UTC 2017

hbase(main):001:0> create 'test', 'cf'
0 row(s) in 1.730 seconds

=> Hbase::Table - test
hbase(main):002:0> list
TABLE          I
ATLAS_ENTITY_AUDIT_EVENTS
atlas_titan
iemployee
test
4 row(s) in 0.0740 seconds

=> ["ATLAS_ENTITY_AUDIT_EVENTS", "atlas_titan", "iemployee", "test"]
hbase(main):003:0>

```

If we want to see column description of a table.

Command- describe tablename

```

hbase(main):003:0> describe 'test'
Table test is ENABLED
test
COLUMN FAMILIES DESCRIPTION
{NAME => 'cf', BLOOMFILTER => 'ROW', VERSIONS => '1', IN_MEMORY => 'False',
KEEP_DELETED_CELLS => 'FALSE', DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', COMPRESSION => 'NONE', MIN_VERSIONS => '0', BLOCKCACHE => 'true',
BLOCKSIZE => '65536', REPLICATION_SCOPE => '0'}
1 row(s) in 0.1950 seconds

hbase(main):004:0>

```

Now, we have to put the values in table

Values:

put ‘test’, ‘row1’, ‘cf:a’, ‘value1’

put ‘test’, ‘row2’, ‘cf:b’, ‘value2’

put ‘test’, ‘row3’, ‘cf:c’, ‘value3’

copy paste the data in shell.

```

< > C ⌂ ▲ 192.168.119.132:6200
Hbase Map Reduce... wordCountWordCount... Bulk Image Reader...
[REDACTED]
[REDACTED]
[REDACTED]

hbase(main):004:0> put 'test', 'row1', 'cf:a', 'value1'
0 row(s) in 0.1950 seconds

hbase(main):004:0>
hbase(main):005:0> put 'test', 'row2', 'cf:b', 'value2'
0 row(s) in 0.0140 seconds

hbase(main):007:0>
hbase(main):008:0> put 'test', 'row3', 'cf:c', 'value3'
0 row(s) in 0.0340 seconds

```

We to display the records of table

Command: scan ‘test’

```

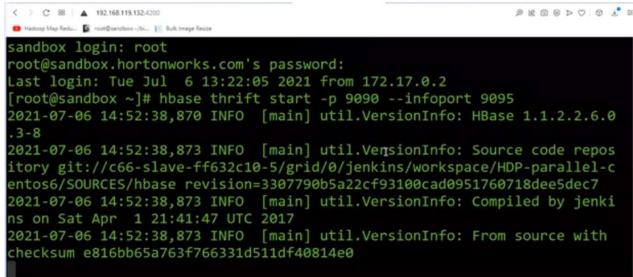
hbase(main):009:0> scan 'test'
ROW           COLUMN+CELL
row1          column=cf:a, timestamp=1625496989589, value=value1
row2          column=cf:b, timestamp=1625496989697, value=value2
row3          column=cf:c, timestamp=1625496993087, value=value3
3 row(s) in 0.0620 seconds

```

Python: storage/retrieval

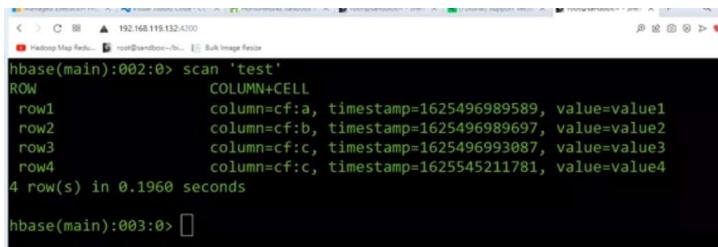
Start the service with command

Hbase thrift start -p 9090 –inforport 9095



```
root@sandbox.hortonworks.com's password:
Last login: Tue Jul  6 13:22:05 2021 from 172.17.0.2
[root@sandbox ~]# hbase thrift start -p 9090 --inforport 9095
2021-07-06 14:52:38,870 INFO [main] util.VersionInfo: HBase 1.1.2.2.6.0
2021-07-06 14:52:38,873 INFO [main] util.VersionInfo: Source code repository git://c66-slave-ff632c10-5/grid/0/jenkins/workspace/HDP-parallel-c
2021-07-06 14:52:38,873 INFO [main] util.VersionInfo: entos6/SOURCES/hbase revision=3307790b5a22cf93100cad0951760718dee5dec7
2021-07-06 14:52:38,873 INFO [main] util.VersionInfo: Compiled by Jenkins on Sat Apr  1 21:41:47 UTC 2017
2021-07-06 14:52:38,873 INFO [main] util.VersionInfo: From source with
checksum e816bb65a763f766331d511df40814e0
```

Create the table the way we did it in hbase and see the records using scan command



```
hbase(main):002:0> scan 'test'
ROW                                COLUMN+CELL
row1      column=cf:a, timestamp=1625496989589, value=value1
row2      column=cf:b, timestamp=1625496989697, value=value2
row3      column=cf:c, timestamp=1625496993087, value=value3
row4      column=cf:c, timestamp=1625545211781, value=value4
4 row(s) in 0.1960 seconds

hbase(main):003:0>
```

Create a program file

Import happybase as hb

```
conn=hb.connection('192.168.119.132', 9090)

print(conn.table('test').row('row1'))
print(conn.table('test').row('row2'))
print(conn.table('test').row('row3'))
print(conn.table('test').row('row4'))

table = conn.table('test')

table.put(b'row5', {b'cf:r': b'value5'})

print(conn.table('test').row('row5'))
```

```

Mains.py - C:\Users\Ganesh\Desktop\all class Big data\All code and steps\bbase2.py (1.8)
File Edit Format Run Options Window Help
import happybase as hb
conn=hb.Connection('192.168.119.132', 9
print(conn.table('test').row('row1'))
print(conn.table('test').row('row2'))
print(conn.table('test').row('row3'))
print(conn.table('test').row('row4'))
table = conn.table('test')
table.put(b'row5', {b'cf:r': b'value5'})
print(conn.table('test').row('row5'))

IDLE Shell 1.8.5
File Edit Shell Debug Options Window Help
all class\Big data\All code and steps\bbase2.py >>>
===== RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\bbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5']
>>>
===== RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\bbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5']
>>> |
```

Run a scan command on shell to display the values

```

hbase(main):004:0> scan 'test'
ROW          COLUMN+CELL
row1        column=cf:a, timestamp=1625496989589, value=value1
row2        column=cf:b, timestamp=1625496989697, value=value2
row3        column=cf:c, timestamp=1625496993087, value=value3
row4        column=cf:c, timestamp=1625545211781, value=value4
row5        column=cf:r, timestamp=1625583481042, value=value5
5 row(s) in 0.0320 seconds
```

Now, try with duplicate value at row 5 say value t

```

Mains.py - C:\Users\Ganesh\Desktop\all class Big data\All code and steps\bbase2.py (1.8)
File Edit Format Run Options Window Help
import happybase as hb
conn=hb.Connection('192.168.119.132', 9
print(conn.table('test').row('row1'))
print(conn.table('test').row('row2'))
print(conn.table('test').row('row3'))
print(conn.table('test').row('row4'))
table = conn.table('test')
table.put(b'row5', {b'cf:t': b'value5'})
print(conn.table('test').row('row5'))

IDLE Shell 1.8.5
File Edit Shell Debug Options Window Help
all class\Big data\All code and steps\bbase2.py >>>
===== RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\bbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:c': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5']
[b'cf:t': b'value5']
>>>
===== RESTART: C:\Users\Ganesh\Desktop\Practice\Big Data\bbaseprogram.py =====
[b'cf:a': b'value1']
[b'cf:b': b'value2']
[b'cf:b': b'value3']
[b'cf:c': b'value4']
[b'cf:r': b'value5', b'cf:t': b'value5']
>>> |
```

Run a scan command on shell to display the values

When there is unique value, it will create a record. If duplicate value it will not create a record

```

hbase(main):005:0> scan 'test'
ROW          COLUMN+CELL
row1        column=cf:a, timestamp=1625496989589, value=value1
row2        column=cf:b, timestamp=1625496989697, value=value2
row3        column=cf:c, timestamp=1625496993087, value=value3
row4        column=cf:c, timestamp=1625545211781, value=value4
row5        column=cf:r, timestamp=1625583481042, value=value5
row5        column=cf:t, timestamp=1625583505297, value=value5
5 row(s) in 0.1320 seconds
```

Practical No. 7

Aim:- Implement Decision tree classification techniques

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Using the Iris dataset, we can construct a tree as follows:

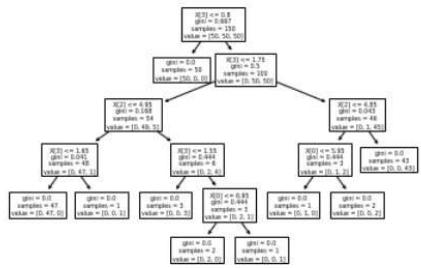
+ Code + Text

```
▶ from sklearn.datasets import load_iris
  from sklearn import tree
  iris = load_iris()
  X, y = iris.data, iris.target
  clf = tree.DecisionTreeClassifier()
  clf = clf.fit(X, y)
```

Once trained, we can plot the tree with the `plot_tree` function:

```
[4] tree.plot_tree(clf)

[Text(167.4, 199.32, 'X[3] <= 0.8\ngini = 0.667\nsamples = 150\nvalue = [50, 50, 50]'),
Text(141.64615384615385, 163.07999999999998, 'gini = 0.0\nsamples = 50\nvalue = [50, 0, 0]'),
Text(193.15384615384616, 163.07999999999998, 'X[3] <= 1.75\ngini = 0.5\nsamples = 100\nvalue = [0, 50, 50]'),
Text(103.01538461538462, 126.83999999999999, 'X[2] <= 4.95\ngini = 0.168\nsamples = 54\nvalue = [0, 49, 5]'),
Text(51.50769230769231, 90.6, 'X[3] <= 1.65\ngini = 0.041\nsamples = 48\nvalue = [0, 47, 1]'),
Text(25.753846153846155, 54.35999999999985, 'gini = 0.0\nsamples = 47\nvalue = [0, 47, 0]'),
Text(77.26153846153846, 54.35999999999985, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(154.52307692307693, 90.6, 'X[3] <= 1.55\ngini = 0.444\nsamples = 6\nvalue = [0, 2, 4]'),
Text(128.76923076923077, 54.35999999999985, 'gini = 0.0\nsamples = 3\nvalue = [0, 0, 3]'),
Text(180.27692307692308, 54.35999999999985, 'X[0] <= 6.95\ngini = 0.444\nsamples = 3\nvalue = [0, 2, 1]'),
Text(154.52307692307693, 18.11999999999976, 'gini = 0.0\nsamples = 2\nvalue = [0, 2, 0]'),
Text(206.03076923076924, 18.11999999999976, 'gini = 0.0\nsamples = 1\nvalue = [0, 0, 1]'),
Text(283.2923076923077, 126.83999999999999, 'X[2] <= 4.85\ngini = 0.043\nsamples = 46\nvalue = [0, 1, 45]'),
Text(257.53846153846155, 90.6, 'X[0] <= 5.95\ngini = 0.444\nsamples = 3\nvalue = [0, 1, 2]'),
Text(231.7846153846154, 54.35999999999985, 'gini = 0.0\nsamples = 1\nvalue = [0, 1, 0]'),
Text(283.2923076923077, 54.35999999999985, 'gini = 0.0\nsamples = 2\nvalue = [0, 0, 2]'),
Text(309.04615384615386, 90.6, 'gini = 0.0\nsamples = 43\nvalue = [0, 0, 43]')
```



Practical No. 8

Aim:- Implement SVM classification techniques

Support Vector Machines

Generally, Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes.

Loading data:

```
[1] #Import scikit-learn dataset library
    from sklearn import datasets

    #Load dataset
    cancer = datasets.load_breast_cancer()
```

Exploring data:

```
▶ # print the names of the 13 features
print("Features: ", cancer.feature_names)

# print the label type of cancer('malignant' 'benign')
print("Labels: ", cancer.target_names)

▷ Features: ['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
Labels: ['malignant' 'benign']
```

Check the shape of the dataset using shape.

```
▶ # print data(feature)shape
cancer.data.shape

▷ (569, 30)
```

Check top 5 records of the feature set.

```
[4] # print the cancer data features (top 5 records)
print(cancer.data[0:5])

[[1.799e+01 1.038e+01 1.228e+02 1.001e+03 1.184e-01 2.776e-01 3.001e-01
 1.471e-01 2.419e-01 7.871e-02 1.095e+00 9.053e-01 8.589e+00 1.534e+02
 6.399e-03 4.904e-02 5.373e-02 1.587e-02 3.003e-02 6.193e-03 2.538e+01
 1.733e+01 1.846e+02 2.019e+03 1.622e-01 6.656e-01 7.119e-01 2.654e-01
 4.601e-01 1.189e-01]
[2.057e+01 1.777e+01 1.329e+02 1.326e+03 8.474e-02 7.864e-02 8.690e-02
 7.017e-02 1.812e-01 5.667e-02 5.435e-01 7.339e-01 3.398e+00 7.408e+01
 5.225e-03 1.308e-02 1.860e-02 1.340e-02 1.389e-02 3.532e-03 2.499e-01
 2.341e+01 1.588e+02 1.956e+03 1.238e-01 1.866e-01 2.416e-01 1.866e-01
 2.750e-01 8.902e-02]
[1.969e+01 2.125e+01 1.300e+02 1.203e+03 1.096e-01 1.599e-01 1.974e-01
 1.279e-01 2.069e-01 5.999e-02 7.456e-01 7.869e-01 4.585e+00 9.403e+01
 6.150e-03 4.006e-02 3.832e-02 2.058e-02 2.250e-02 4.571e-03 2.357e+01
 2.553e+01 1.525e+02 1.709e+03 1.444e-01 4.245e-01 4.504e-01 2.430e-01
 3.613e-01 8.758e-02]
[1.142e+01 2.038e+01 7.758e+01 3.861e+02 1.425e-01 2.839e-01 2.414e-01
 1.052e-01 2.597e-01 9.744e-02 4.956e-01 1.156e+00 3.445e+00 2.723e+01
 9.110e-03 7.458e-02 5.661e-02 1.867e-02 5.963e-02 9.208e-03 1.491e+01
 2.650e+01 9.887e+01 5.677e+02 2.098e-01 8.663e-01 6.869e-01 2.575e-01
 6.638e-01 1.730e-01]
[2.029e+01 1.434e+01 1.351e+02 1.297e+03 1.003e-01 1.328e-01 1.980e-01
 1.043e-01 1.809e-01 5.883e-02 7.572e-01 7.813e-01 5.438e+00 9.444e+01
 1.149e-02 2.461e-02 5.688e-02 1.885e-02 1.756e-02 5.115e-03 2.254e+01
 1.667e+01 1.522e+02 1.575e+03 1.374e-01 2.050e-01 4.000e-01 1.625e-01
 2.364e-01 7.678e-02]]
```

Target set:

Splitting Data:

To understand model performance, dividing the dataset into a training set and a test set is a good strategy.

Split the dataset by using the function `train_test_split()`. you need to pass 3 parameters features, target, and `test_size`. Additionally, you can use `random_state` to select records randomly.

```
[7] # Import train_test_split function
    from sklearn.model_selection import train_test_split

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target, test_size=0.3,random_state=109) # 70% training and 30% test
```

Generate Model:

Let's build support vector machine model. First, import the SVM module and create support vector classifier object by passing argument kernel as the linear kernel in SVC() function.

Then, fit your model on train set using `fit()` and perform prediction on the test set using `predict()`.

```
[8] #Import svm model
from sklearn import svm

#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

#Train the model using the training sets
clf.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

Evaluating the Model:

Let's estimate how accurately the classifier or model can predict the breast cancer of patients. Accuracy can be computed by comparing actual test set values and predicted values.

```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics

# Model Accuracy: how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.9649122807017544
```

**University Of Mumbai
Institute of Distance & Open Learning**



**PRACTICAL JOURNAL IN
MODERN NETWORKING
SUBMITTED BY
APPLICATION ID:
SEAT NO :**

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY PART-I
SEMESTER II**

**ACADEMIC YEAR
2022-2023**

**INSTITUTE OF DISTANCE AND OPEN LEARNING
IDOL BUILDING, VIDYANAGARI,
SANTACRUZ (EAST), MUMBAI-400 098**

**CONDUCTED AT
PARLE TILAK VIDYALAYA ASSOCIATION'S
MULUND COLLEGE OF COMMERCE(AUTONOMOUS)
MULUND (WEST), MUMBAI – 400080**

University of Mumbai
Institute of Distance & Open Learning



Dr. Shankar Dayal Sharama Bhavan, Kalina,
Vidanagari, Santacruz (E), Mumbai-400 098.

Certificate

This is to certify that Mr. Application ID:from Institute of Distance and Open Learning, University of Mumbai , Santacruz East, Mumbai-400098 , PCP Centre: Parle Tilak Vidyalaya Association's Mulund College of Commerce (Autonomous) has successfully completed all the practical of Paper II titled **MODERN NETWORKING** for M.Sc. (I.T.) Part I Semester – II during the academic year **2022-2023**.

Centre Coordinator

External Examiner

Date:

Seal

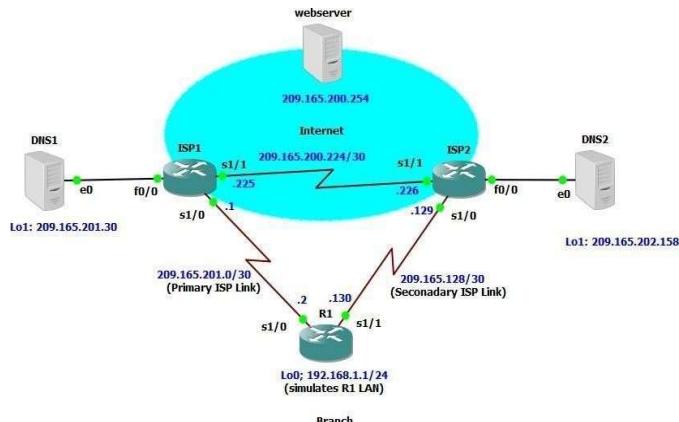
INDEX

Sr. No.	Practical	Date	Sign
1	Configure IP SLA Tracking and Path Control Topology		
2	Using the AS_PATH Attribute		
3	Configuring IBGP and EBGP Sessions, Local Preference, and MED		
4	Secure the Management Plane		
5	Configure and Verify Path Control Using PBRs		
6	Simulating MPLS environment		

Practical No. 1

Aim: Configure IP SLA Tracking and Path Control Topology

Topology :



Objectives

- Configure and verify the IP SLA feature.
- Test the IP SLA tracking feature.
- Verify the configuration and operation using show and debug commands.

Step 1: Prepare the routers and configure the router hostname and interface addresses.

Router R1

```
interface Loopback 0
```

```
ip address 192.168.1.1 255.255.255.0
```

```
interface Serial0/0
```

```
ip address 209.165.201.2 255.255.255.252
```

```
no shutdown interface Serial0/0/1
```

```
ip address 209.165.202.130 255.255.255.252
```

```
no shutdown
```

```
R1(config)#interface Loopback 0
```

```
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#
R1(config-if)#int s1/0
R1(config-if)#ip address 209.165.201.2 255.255.255.252
R1(config-if)#no shutdown
```

```
R1(config-if)#int s1/1
```

```
R1(config-if)#ip address 209.165.202.130 255.255.255.252
R1(config-if)#no shutdown
```

Router ISP1 (R2)

interface Loopback0

ip address 209.165.200.254 255.255.255.255

interface Loopback1

ip address 209.165.201.30 255.255.255.255

interface Serial0/0/0

ip address 209.165.201.1 255.255.255.252

no shutdown interface

Serial0/0/1

ip address 209.165.200.225 255.255.255.252

no shutdown

```
ISP1(config)#interface Loopback0
ISP1(config-if)#
"May 18 15:24:24.315: %LINEPROTO-5-UPDOWN: Line protocol on interface Loopback0 changed state to up
ISP1(config-if)#ip address 209.165.200.254 255.255.255.255
ISP1(config-if)#interface Loopback1
ISP1(config-if)#
"May 18 15:24:36.915: %LINEPROTO-5-UPDOWN: Line protocol on interface Loopback1 changed state to up
ISP1(config-if)#ip address 209.165.201.30 255.255.255.255
ISP1(config-if)#int s1/0
ISP1(config-if)#ip address 209.165.201.1 255.255.255.252
ISP1(config-if)#no shutdown
ISP1(config-if)#
"May 18 15:25:03.695: %LINK-3-UPDOWN: Interface Serial1/0, Line protocol up
ISP1(config-if)#
ISP1(config-if)#
"May 18 15:25:04.699: %LINEPROTO-5-UPDOWN: Line protocol on interface Serial1/0 changed state to up
ISP1(config-if)#int s1/1
ISP1(config-if)#ip address 209.165.200.225 255.255.255.252
ISP1(config-if)#no shutdown
```

Router ISP2 (R3)

interface Loopback0

ip address 209.165.200.254 255.255.255.255

interface Loopback1

ip address 209.165.202.158 255.255.255.255

interface Serial0/0/0 description

ISP2 --> R1

ip address 209.165.202.129 255.255.255.252

no shutdown interface

Serial0/0/1

ip address 209.165.200.226 255.255.255.252

no shutdown

```
ISP2(config)#interface Loopback0
ISP2(config-if)#
*May 18 15:25:22.219: %LINEPROTO-5-UPDOWN: Line protocol on
ISP2(config-if)#ip address 209.165.200.254 255.255.255.255
ISP2(config-if)#interface Loopback1
ISP2(config-if)#
*May 18 15:25:34.595: %LINEPROTO-5-UPDOWN: Line protocol on
ISP2(config-if)#ip address 209.165.202.158 255.255.255.255
ISP2(config-if)#int s1/0
ISP2(config-if)#ip address 209.165.202.129 255.255.255.252
ISP2(config-if)#no shutdown
ISP2(config-if)#
ISP2(config-if)#
*May 18 15:26:01.299: %LINK-3-UPDOWN: Interface Serial1/0,
ISP2(config-if)#
*May 18 15:26:02.303: %LINEPROTO-5-UPDOWN: Line protocol on
ISP2(config-if)#int s1/1
ISP2(config-if)#ip address 209.165.200.226 255.255.255.252
ISP2(config-if)#no shutdown
```

b. Verify the configuration by using the show interfaces description command. The output from router R1 is shown here as an example.

R1# show interfaces description

Interface	Status	Protocol Description
Fa0/0	admin down	down
Se1/0	up	up
Se1/1	up	up
Se1/2	admin down	down
Se1/3	admin down	down
Lo0	up	up

c. The current routing policy in the topology is as follows:

- Router R1 establishes connectivity to the Internet through ISP1 using a default static route.
- ISP1 and ISP2 have dynamic routing enabled between them, advertising their respective public address pools.
- ISP1 and ISP2 both have static routes back to the ISP LAN.

Router R1

ip route 0.0.0.0 0.0.0.0 209.165.201.1

```
R1(config)#  
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1
```

Router ISP1 (R2)

router eigrp

network 209.165.200.224 0.0.0.3

network 209.165.201.0 0.0.0.31

no auto-summary

ip route 192.168.1.0 255.255.255.0 209.165.201.2

```
ISP1(config)#router eigrp 1  
ISP1(config-router)#network 209.165.200.224 0.0.0.3  
ISP1(config-router)#network 209.165.201.0 0.0.0.31  
ISP1(config-router)#no auto-summary  
ISP1(config-router)#ip route 192.168.1.0 255.255.255.0 209.165.201.2
```

Router ISP2 (R3)

router eigrp 1

network 209.165.200.224 0.0.0.3

network 209.165.202.128 0.0.0.31

no auto-summary

ip route 192.168.1.0 255.255.255.0 209.165.202.130

```

ISP2(config)#router eigrp 1
ISP2(config-router)#network 209.165.200.224 0.0.0.3
ISP2(config-router)#
*May 18 15:30:14.515: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 209.1
ISP2(config-router)#network 209.165.202.128 0.0.0.31
ISP2(config-router)#
ISP2(config-router)#
*May 18 15:30:28.971: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 209.1
ISP2(config-router)#ip route 192.168.1.0 255.255.255.0 209.165.202.130

```

Step 2: Verify server reachability.

- Before implementing the Cisco IOS SLA feature, you must verify reachability to the Internet servers. From router R1, ping the web server, ISP1 DNS server, and ISP2 DNS server to verify connectivity. You can copy the following Tcl script and paste it into R1.

```

R1(tcl)# foreach address {
+>(tcl)# 209.165.200.254
+>(tcl)# 209.165.201.30
+>(tcl)# 209.165.202.158
+>(tcl)# } { ping $address source 192.168.1.1 }

```

```

R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#209.165.200.254
+>(tcl)#209.165.201.30
+>(tcl)#209.165.202.158
+>(tcl)#} {
+>(tcl)#ping $address source 192.168.1.1
+>(tcl)#

```

Type escape sequence to abort.
 Sending 5, 100-byte ICMP Echos to 209.165.200.254, timeout is 2 seconds:
 Packet sent with a source address of 192.168.1.1
 !!!!!
 Success rate is 100 percent (5/5), round-trip min/avg/max = 28/78/96 ms
 Type escape sequence to abort.
 Sending 5, 100-byte ICMP Echos to 209.165.201.30, timeout is 2 seconds:
 Packet sent with a source address of 192.168.1.1
 !!!!!
 Success rate is 100 percent (5/5), round-trip min/avg/max = 20/31/48 ms
 Type escape sequence to abort.
 Sending 5, 100-byte ICMP Echos to 209.165.202.158, timeout is 2 seconds:
 Packet sent with a source address of 192.168.1.1
 !!!!!
 Success rate is 100 percent (5/5), round-trip min/avg/max = 16/37/60 ms

b. Trace the path taken to the web server, ISP1 DNS server, and ISP2 DNS server. You can copy the following Tcl script and paste it into R1.

```
R1(tcl)# foreach address {  
+>(tcl)# 209.165.200.254  
+>(tcl)# 209.165.201.30  
+>(tcl)# 209.165.202.158  
  
R1(tcl)#foreach address {  
+>(tcl)#209.165.200.254  
+>(tcl)#209.165.201.30  
+>(tcl)#209.165.202.158  
+>(tcl)#} {  
+>(tcl)#trace $address source 192.168.1.1  
+>(tcl)#}  
+>(tcl)# } { trace $address source 192.168.1.1 }
```

```
Type escape sequence to abort.  
Tracing the route to 209.165.200.254  
  
 1 209.165.201.1 24 msec 24 msec 16 msec  
Type escape sequence to abort.  
Tracing the route to 209.165.201.30  
  
 1 209.165.201.1 16 msec 24 msec 24 msec  
Type escape sequence to abort.  
Tracing the route to 209.165.202.158  
  
 1 209.165.201.1 24 msec 24 msec 32 msec  
 2 209.165.200.226 28 msec 44 msec 72 msec
```

Step 3: Configure IP SLA probes.

- a. Create an ICMP echo probe on R1 to the primary DNS server on ISP1 using the ip sla command. the previous ip sla monitor command. In addition, the icmp-echo command has replaced the type echo protocol ipIcmpEcho command.

```
R1(config)# ip sla 11
```

```
R1(config-ip-sla)# icmp-echo 209.165.201.30
```

```
R1(config-ip-sla-echo)# frequency 10
```

```
R1(config-ip-sla-echo)# exit
```

```
R1(config)# ip sla schedule 11 life forever start-time now
```

```
R1(config)#ip sla 11  
R1(config-ip-sla)#icmp-echo 209.165.201.30  
R1(config-ip-sla-echo)#frequency 10  
R1(config-ip-sla-echo)#exit  
R1(config)#ip sla schedule 11 life forever start-time now  
R1(config)#exit
```

- b. Verify the IP SLAs configuration of operation 11 using the show ip sla configuration 11 command.

```
R1# show ip sla configuration 11
```

```
R1(tcl)#show ip sla configuration 11
IP SLAs, Infrastructure Engine-II.
Entry number: 11
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.201.30/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
    Operation frequency (seconds): 10 (not considered if
        Next Scheduled Start Time: Start Time already passed
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): Forever
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): Active
```

- c. Issue the show ip sla statistics command to display the number of successes, failures, and results of the latest operations.

R1# show ip sla statistics

```
R1#show ip sla statistics 22
IPSLAs Latest Operation Statistics

IPSLA operation id: 22
Type of operation: icmp-echo
    Latest RTT: 64 milliseconds
Latest operation start time: *15:40:40.823 UTC Tue May 18 2021
Latest operation return code: OK
Number of successes: 6
Number of failures: 0
Operation time to live: Forever
```

- d. Although not actually required because IP SLA session 11 alone could provide the desired fault tolerance, create a second probe, 22, to test connectivity to the second DNS server located on router ISP2. You can copy and paste the following commands on R1.

```
R1(config)#
R1(config)#ip sla 22
R1(config-ip-sla)#icmp-echo 209.165.202.158
R1(config-ip-sla-echo)#frequency 10
R1(config-ip-sla-echo)#exit
R1(config)#ip sla schedule 22 life forever start-time now
```

- e. Verify the new probe using the show ip sla configuration and show ip sla statistics commands.

R1# show ip sla configuration 22

```
R1#show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.
Entry number: 22
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source address: 209.165.202.158/0.0.0.0
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
```

R1# show ip sla statistics 22

```
R1#show ip sla statistics 22
IPSLAs Latest Operation Statistics

IPSLA operation id: 22
Type of operation: icmp-echo
    Latest RTT: 64 milliseconds
Latest operation start time: *15:40:40.823 UTC Tue May 18 2021
Latest operation return code: OK
Number of successes: 6
Number of failures: 0
Operation time to live: Forever
```

Step 4: Configure tracking options.

- Remove the current default route on R1, and replace it with a floating static route having an administrative distance of 5.

R1(config)# no ip route 0.0.0.0 0.0.0.0 209.165.201.1

R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 5

R1(config)# exit

```
R1(config)#
R1(config)#no ip route 0.0.0.0 0.0.0.0 209.165.201.1
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.201.1 5
R1(config)#exit
```

- Verify the routing table. R1#

show ip route

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0
```

- c. Use the track 1 ip sla 11 reachability command to enter the config-track sub-configuration mode.

```
R1(config)# track 1 ip sla 11 reachability
```

```
R1(config-track)#
```

```
R1(config)#  
R1(config)#track 1 ip sla 11 reachability  
R1(config-track)#delay down 10 up 1  
R1(config-track)#exit
```

- d. Configure the floating static route that will be implemented when tracking object 1 is active. To view routing table changes as they happen, first enable the debug ip routing command. Next, use the ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1 command to create a floating static default route via 209.165.201.1 (ISP1). Notice that this command references the tracking object number 1, which in turn references IP SLA operation number 11.

```
R1# debug ip routing
```

```
R1#debug ip routing  
IP routing debugging is on
```

```
R1(config)# ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1
```

```
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1  
R1(config)#  
*May 18 15:43:00.035: RT: closer admin distance for 0.0.0.0, flushing 1 routes  
*May 18 15:43:00.035: RT: NET-RED 0.0.0.0/0  
*May 18 15:43:00.035: RT: add 0.0.0.0/0 via 209.165.201.1, static metric [2/0]  
*May 18 15:43:00.039: RT: NET-RED 0.0.0.0/0  
*May 18 15:43:00.039: RT: default path is now 0.0.0.0 via 209.165.201.1  
*May 18 15:43:00.039: RT: new default network 0.0.0.0  
*May 18 15:43:00.043: RT: NET-RED 0.0.0.0/0
```

- e. Repeat the steps for operation 22, track number 2, and assign the static route an admin distance higher than track 1 and lower than 5. On R1, copy the following configuration, which sets an admin distance of 3. track 2 ip sla 22 reachability delay down 10 up 1 exit

```
ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2
```

```
R1(config)#track 1 ip sla 22 reachability  
R1(config-track)#delay down 10 up 1  
R1(config-track)#exit  
*May 18 15:43:56.339: RT: NET-RED 0.0.0.0/0  
R1(config-track)#exit  
R1(config)##ip route 0.0.0.0 0.0.0.0 209.165.201.1 2 track 1  
R1(config)#ip route 0.0.0.0 0.0.0.0 209.165.202.129 3 track 2  
R1(config)##
```

- f. Verify the routing table again

```
R1# show ip route
```

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF i
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA extern
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2
      ia - IS-IS inter area, * - candidate default, U - per
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0.0

  209.165.201.0/30 is subnetted, 1 subnets
C    209.165.201.0 is directly connected, Serial1/0
  209.165.202.0/30 is subnetted, 1 subnets
C    209.165.202.128 is directly connected, Serial1/1
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

Step 5: Verify IP SLA operation.

The following summarizes the process:

- Disable the DNS loopback interface on ISP1 (R2).
- Observe the output of the debug command on R1.
- Verify the static route entries in the routing table and the IP SLA statistics of R1.
- Re-enable the loopback interface on ISP1 (R2) and again observe the operation of the IP SLA tracking feature.

```
ISP1(config)# interface loopback 1
```

```
ISP1(config-if)# shutdown
```

```
ISP1(config)#
ISP1(config)#interface loopback 1
ISP1(config-if)#shutdown
```

b. Verify the routing table. R1#

```
show ip rout
```

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ex-
      E1 - OSPF external type 1, E2 - OSPF external typ-
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1
      ia - IS-IS inter area, * - candidate default, U -
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 209.165.201.1 to network 0.0.0

  209.165.201.0/30 is subnetted, 1 subnets
C    209.165.201.0 is directly connected, Serial1/0
  209.165.202.0/30 is subnetted, 1 subnets
C    209.165.202.128 is directly connected, Serial1/1
C    192.168.1.0/24 is directly connected, Loopback0
5*   0.0.0.0/0 [2/0] via 209.165.201.1
```

c. Verify the IP SLA statistics.

```
R1# show ip sla statistics
```

```
R1#show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11
Type of operation: icmp-echo
  Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: *15:47:41.887 UTC Tue May 18 2021
Latest operation return code: No connection
Number of successes: 51
Number of failures: 13
Operation time to live: Forever
```

d. Initiate a trace to the web server from the internal LAN IP address.

```
R1# trace 209.165.200.254 source 192.168.1.1
```

```
R1#trace 209.165.200.254 source 192.168.1.1

Type escape sequence to abort.
Tracing the route to 209.165.200.254

  1 209.165.201.1 4 msec 32 msec 32 msec
```

f. Again examine the IP SLA statistics. R1# show
ip sla statistics

```
R1#show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 11
Type of operation: icmp-echo
    Latest RTT: 57 milliseconds
Latest operation start time: *15:50:01.887 UTC Tue May 18 2021
Latest operation return code: OK
Number of successes: 61
Number of failures: 17
Operation time to live: Forever
```

g. Verify the routing table.

```
R1# show ip route
```

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2
      ia - IS-IS inter area, * - candidate default, U - per-
      o - ODR, P - periodic downloaded static route

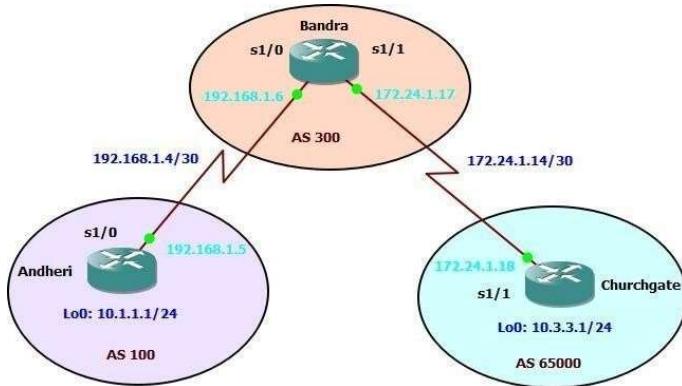
Gateway of last resort is 209.165.201.1 to network 0.0.0.0

  209.165.201.0/30 is subnetted, 1 subnets
C    209.165.201.0 is directly connected, Serial1/0
  209.165.202.0/30 is subnetted, 1 subnets
C    209.165.202.128 is directly connected, Serial1/1
C    192.168.1.0/24 is directly connected, Loopback0
S*   0.0.0.0/0 [2/0] via 209.165.201.1
```

Practical No - 2

Aim: Using the AS_PATH Attribute

Topology :



Objective :

- Use BGP commands to prevent private AS numbers from being advertised to the outside world.
- Use the AS_PATH attribute to filter BGP routes based on their source AS number

Step 1 : Prepare the routers for the lab.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations.

Step 2 : Configure the hostname and interface addresses.

a. You can copy and paste the following configurations into your routers to begin.

Router R1 (hostname Andheri)

```
Andheri(config)# interface Loopback0 Andheri(config-if)# ip  
address 10.1.1.1 255.255.255.0 Andheri(config-if)# exit  
Andheri(config)# interface Serial0/0/0  
Andheri(config-if)# ip address 192.168.1.5 255.255.255.252  
Andheri(config-if)# no shutdown  
Andheri(config-if)# end  
Andheri#
```

```
R1#  
R1#conf t  
Enter configuration commands, one per line. End with CNTL/Z.  
R1(config)#hostname Andheri  
Andheri(config)#int loopback 0  
Andheri(config-if)#  
*May 7 09:30:42.867: %LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0  
Andheri(config-if)#ip address 10.1.1.1 255.255.255.0  
Andheri(config-if)#exit  
Andheri(config)#int s1/0  
Andheri(config-if)#ip address 192.168.1.5 255.255.255.252  
Andheri(config-if)#no shutdown  
Andheri(config-if)#  
*May 7 09:31:41.315: %LINK-3-UPDOWN: Interface Serial1/0, changed state to up
```

Router R2 (hostname Bandra)

```
Bandra(config)# interface Loopback0  
Bandra(config-if)# ip address 10.2.2.1 255.255.255.0  
Bandra(config-if)# interface Serial0/0/0  
Bandra(config-if)# ip address 192.168.1.6 255.255.255.252  
Bandra(config-if)# no shutdown  
Bandra(config-if)# exit Bandra(config)#  
interface Serial0/0/1  
Bandra(config-if)# ip address 172.24.1.17 255.255.255.252  
Bandra(config-if)# no shutdown  
Bandra(config-if)# end  
Bandra#
```

```
R2#
R2#conf t
Enter configuration commands, one per line. End with CNTL/Z
R2(config)#hostname Bandra
Bandra(config)#int loopback 0
Bandra(config-if)#ip addr
*May  7 09:31:30.407: %LINEPROTO-5-UPDOWN: Line protocol
Bandra(config-if)#ip address 10.2.2.1 255.255.255.0
Bandra(config-if)#exit
Bandra(config)#int s1/0
Bandra(config-if)#ip address 192.168.1.6 255.255.255.252
Bandra(config-if)#no shutdown
Bandra(config-if)#exit
```

Router R3 (hostname ChurchGate)

```
Churchgate(config)# interface Loopback0

Churchgate(config-if)# ip address 10.3.3.1 255.255.255.0

Churchgate(config-if)# exit Churchgate(config)#
interface Serial0/0/1

Churchgate(config-if)# ip address 172.24.1.18 255.255.255.252

Churchgate(config-if)# no shutdown

Churchgate(config-if)# end

Churchgate#
```

```
Bandra(config)#int s1/1
Bandra(config-if)#ip address 172.24.1.17 255.255.255.252
Bandra(config-if)#no shutdown
Bandra(config-if)#
*May  7 09:33:39.591: %LINK-3-UPDOWN: Interface Serial1/1,
```

```
R3#
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z
R3(config)#hostname Churchgate
Churchgate(config)#int loopback 0
Churchgate(config-if)#ip[
*May  7 09:33:31.243: %LINEPROTO-5-UPDOWN: Line protocol on I
Churchgate(config-if)#ip address 10.3.3.1 255.255.255.0
Churchgate(config-if)#exit
Churchgate(config)#int s1/1
Churchgate(config-if)#ip address 172.24.1.18 255.255.255.252
Churchgate(config-if)#no shutdown
Churchgate(config-if)#
*May  7 09:34:39.795: %LINK-3-UPDOWN: Interface Serial1/1, ch
```

b. Use ping to test the connectivity between the directly connected routers.

```
Bandra#ping 192.168.1.5
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/32/40 ms
Bandra#
Bandra#
Bandra#ping 172.24.1.18
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/36/48 ms
Bandra#
```

Step 3 : Configure BGP.

a. Configure BGP for normal operation. Enter the appropriate BGP commands on each router so that they identify their BGP neighbors and advertise their loopback networks.

```
Andheri(config)# router bgp 100
```

```
Andheri(config-router)# neighbor 192.168.1.6 remote-as 300
```

```
Andheri(config-router)# network 10.1.1.0 mask 255.255.255.0
```

```
Andheri#conf t
Enter configuration commands, one per line.  End with CNTL/DEL.
Andheri(config)#router bgp 100
Andheri(config-router)#neighbor 192.168.1.6 remote-as 300
Andheri(config-router)#network 10.1.1.0 mask 255.255.255.0
```

```
Bandra(config)# router bgp 300
```

```
Bandra(config-router)# neighbor 192.168.1.5 remote-as 100
```

```
Bandra(config-router)# neighbor 172.24.1.18 remote-as 65000
```

```
Bandra(config-router)# network 10.2.2.0 mask 255.255.255.0
```

```
Bandra#conf t
Enter configuration commands, one per line.  End with CNTL/DEL.
Bandra(config)#router bgp 300
Bandra(config-router)#neighbor 192.168.1.5 remote-as 100
Bandra(config-router)#
*May 7 10:04:59.051: %BGP-5-ADJCHANGE: neighbor 192.168.1.5
Bandra(config-router)#neighbor 172.24.1.18 remote-as 65000
Bandra(config-router)#network 10.2.2.0 mask 255.255.255.0
```

```
Churchgate(config)# router bgp 65000
```

```
Churchgate(config-router)# neighbor 172.24.1.17 remote-as 300
```

```
Churchgate(config-router)# network 10.3.3.0 mask 255.255.255.0
```

```
Churchgate#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Churchgate(config)#router bgp 65000
Churchgate(config-router)#neighbor 172.24.1.17 remote-as 300
Churchgate(config-router)#
*May  7 10:04:44.195: %BGP-5-ADJCHANGE: neighbor 172.24.1.17
Churchgate(config-router)#network 10.3.3.0 mask 255.255.255.0
```

- b. Verify that these routers have established the appropriate neighbor relationships by issuing the show ip bgp neighbors command on each router.

```
Bandra# show ip bgp neighbors
```

```
Bandra#show ip bgp neighbors
BGP neighbor is 172.24.1.18,  remote AS 65000,  external link
  BGP version 4,  remote router ID 10.3.3.1
  BGP state = Established, up for 00:01:30
  Last read 00:00:13, last write 00:00:44, hold time is 180, keep
  seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    New ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

              Sent      Rcvd
  Opens:          1          1
  Notifications: 0          0
  Updates:        3          1
  Keepalives:     2          3
  Route Refresh: 0          0
  Total:         6          5
Default minimum time between advertisement runs is 30 seconds
```

Step 4 : Remove the private AS.

- a. DBandralay the Andheri routing table using the show ip route command. Andheri should have a route to both 10.2.2.0 and 10.3.3.0. Troubleshoot if necessary.

```
Andheri#show ip route
```

```

Andheri#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, V - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  10.0.0.0/24 is subnetted, 3 subnets
B    10.3.3.0 [20/0] via 192.168.1.6, 00:04:51
B    10.2.2.0 [20/0] via 192.168.1.6, 00:05:22
C    10.1.1.0 is directly connected, Loopback0
  192.168.1.0/30 is subnetted, 1 subnets
C      192.168.1.4 is directly connected, Serial1/0

```

b . Ping again, this time as an extended ping, sourcing from the Loopback0 interface address. **ping 10.3.3.1 source 10.1.1.1 or ping 10.3.3.1 source Lo0**

```

Andheri#ping 10.3.3.1 source 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/52/64 ms
Andheri#
Andheri#show ip bgp
BGP table version is 4, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - in
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* > 10.1.1.0/24      0.0.0.0              0        32768 i
* > 10.2.2.0/24      192.168.1.6          0        0 300 i
* > 10.3.3.0/24      192.168.1.6          0        0 300 65000 i

```

c. Now check the BGP table on Andheri. The AS_ PATH to the 10.3.3.0 network should be AS 300. It no longer has the private AS in the path.

```
Andheri# show ip bgp
```

```

Andheri#ping 10.3.3.1 source 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/60/9
Andheri#show ip bgp
BGP table version is 5, local router ID is 10.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*  10.1.1.0/24       0.0.0.0              0        32768 i
*  10.2.2.0/24     192.168.1.6           0        300 i
*  10.3.3.0/24     192.168.1.6           0        300 i

```

Step 5 : Use the AS_PATH attribute to filter routes.

- Configure a special kind of access list to match BGP routes with an AS_PATH attribute that both begins and ends with the number 100. Enter the following commands on Bandra.

```
Bandra(config)# ip as-path access-list 1 deny ^100$
```

```
Bandra(config)# ip as-path access-list 1 permit .*
```

```

Bandra#
Bandra#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Bandra(config)#ip as-path access-list 1 deny ^100$
Bandra(config)#ip as-path access-list 1 permit .*

```

- Apply the configured access list using the neighbor command with the filter-list option.

```
Bandra(config)# router bgp 300
```

```
Bandra (config-router)# neighbor 192.168.1.5 remove-private-as
```

```

Bandra(config)#router bgp 300
Bandra(config-router)#neighbor 172.24.1.18 filter-list 1 out
Bandra(config-router)#exit

```

- Use the clear ip bgp * command to reset the routing information. Wait several seconds and then check the routing table for BANDRA. The route to 10.1.1.0 should be in the routing table.

```
Andheri# show ip route
```

```

Bandra#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  172.24.0.0/30 is subnetted, 1 subnets
C        172.24.1.16 is directly connected, Serial1/1
  10.0.0.0/24 is subnetted, 3 subnets
B          10.3.3.0 [20/0] via 172.24.1.18, 00:13:19
C        10.2.2.0 is directly connected, Loopback0
B          10.1.1.0 [20/0] via 192.168.1.5, 00:13:20
  192.168.1.0/30 is subnetted, 1 subnets
C        192.168.1.4 is directly connected, Serial1/0

```

d. Return to BANDRA and verify that the filter is working as intended.

```
Bandra# show ip bgp regexp ^100$
```

```

Bandra#show ip bgp regexp ^100$
BGP table version is 4, local router ID is 10.2.2.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop           Metric LocPrf Weight Path
* > 10.1.1.0/24    192.168.1.5        0          0 100 i

```

e. Run the following Tcl script on all routers to verify whether there is connectivity. All pings from BANDRA should be successful. Andheri should not be able to ping the Churchgate loopback 10.3.3.1 or the WAN link 172.24.1.16/30. Churchgate should not be able to ping the Andheri loopback 10.1.1.1 or the WAN link 192.168.1.4/30.

```

Bandra#tclsh
Bandra(tcl)#foreach address {
+>10.1.1.1
+>10.2.2.1
+>10.3.3.1
+>192.168.1.5
+>192.168.1.6
+>172.24.1.17
+>172.24.1.18
+>} { ping $address }

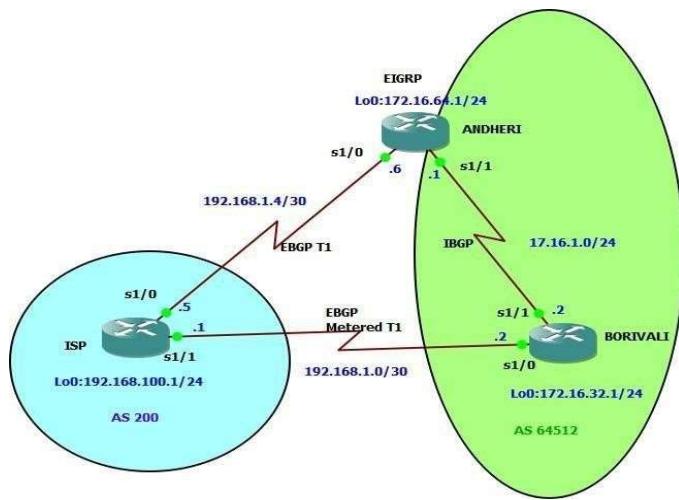
```

```
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/40/64 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/32/48 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.1.5, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/28/40 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 192.168.1.6, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/59/64 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.24.1.17, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/56/68 ms  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 172.24.1.18, timeout is 2 seconds:  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/31/48 ms
```

Practical No - 3

Aim: Configuring IBGP and EBGP Sessions, Local Preference, and MED

Topolgy :



Objectives

- For IBGP peers to correctly exchange routing information, use the **next-hop-self** command with the **Local-Preference** and **MED** attributes.
- Ensure that the flat-rate, unlimited-use T1 link is used for sending and receiving data to and from the AS 200 on ISP and that the metered T1 only be used in the event that the primary T1 link has failed.

Step 1: Configure interface addresses.

Router R1 (hostname ISP)

```
ISP(config)# interface Loopback0
```

```
ISP(config-if)# ip address 192.168.100.1 255.255.255.0
```

```
ISP(config-if)# exit
```

```
ISP(config)# interface Serial0/0/0
ISP(config-if)# ip address 192.168.1.5 255.255.255.252
ISP(config-if)# no shutdown
ISP(config-if)# exit
ISP(config)# interface Serial0/0/1
ISP(config-if)# ip address 192.168.1.1 255.255.255.252
ISP(config-if)# no shutdown
ISP(config-if)# end
ISP(config-if)#interface Loopback0
ISP(config-if)#ip address 192.168.100.1 255.255.255.0
ISP(config-if)#exit
ISP(config)#
ISP(config)#int s1/0
ISP(config-if)#ip address 192.168.1.5 255.255.255.252
ISP(config-if)#no shutdown
ISP(config-if)#exit
ISP(config)#
*May 18 17:42:51.491: %LINK-3-UPDOWN: Interface Serial1
ISP(config)#
*May 18 17:42:52.495: %LINEPROTO-5-UPDOWN: Line protocol
ISP(config)#
ISP(config)#int s1/1
ISP(config-if)#ip address 192.168.1.1 255.255.255.252
ISP(config-if)#no shutdown
```

Router R2 (hostname SanJose1)

```
SanJose1(config)# interface Loopback0
SanJose1(config-if)# ip address 172.16.64.1 255.255.255.0
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/0
SanJose1(config-if)# ip address 192.168.1.6 255.255.255.252
SanJose1(config-if)# no shutdown
SanJose1(config-if)# exit
SanJose1(config)# interface Serial0/0/1
SanJose1(config-if)# ip address 172.16.1.1 255.255.255.0
```

```
SanJose1(config-if)# no shutdown
```

```
SanJose1(config-if)# end
```

```
ANDHERI(config)#interface Loopback0
ANDHERI(config-if)#
*May 18 17:42:40.167: %LINEPROTO-5-UPDOWN: Line protocol o
ANDHERI(config-if)#ip address 172.16.64.1 255.255.255.0
ANDHERI(config-if)#exit
ANDHERI(config)#
ANDHERI(config)#int s1/0
ANDHERI(config-if)#ip address 192.168.1.6 255.255.255.252
ANDHERI(config-if)#no shutdown
ANDHERI(config-if)#exit
ANDHERI(config)#
*May 18 17:43:11.899: %LINK-3-UPDOWN: Interface Serial1/0
ANDHERI(config)#
*May 18 17:43:12.903: %LINEPROTO-5-UPDOWN: Line protocol o
ANDHERI(config)#
ANDHERI(config)#int s1/1
ANDHERI(config-if)#ip address 172.16.1.1 255.255.255.0
ANDHERI(config-if)#no shutdown
```

Router R3 (hostname SanJose2)

```
SanJose2(config)# interface Loopback0
```

```
SanJose2(config-if)# ip address 172.16.32.1 255.255.255.0
```

```
SanJose2(config-if)# exit
```

```
SanJose2(config)# interface Serial0/0/0
```

```
SanJose2(config-if)# ip address 192.168.1.2 255.255.255.252
```

```
SanJose2(config-if)# no shutdown
```

```
SanJose2(config-if)# exit
```

```
SanJose2(config)# interface Serial0/0/1
```

```
SanJose2(config-if)# ip address 172.16.1.2 255.255.255.0
```

```
SanJose2(config-if)# no shutdown
```

```
SanJose2(config-if)# end
```

```
BORIVALI(config)#interface Loopback0
BORIVALI(config-if)#
*May 18 17:43:25.783: %LINEPROTO-5-UPDOWN: Line protocol o
BORIVALI(config-if)#ip address 172.16.32.1 255.255.255.0
BORIVALI(config-if)#exit
BORIVALI(config)#
BORIVALI(config)#int s1/0
BORIVALI(config-if)#ip address 192.168.1.2 255.255.252
BORIVALI(config-if)#no shutdown
BORIVALI(config-if)#exit
BORIVALI(config)#
*May 18 17:43:54.311: %LINK-3-UPDOWN: Interface Serial1/0,
BORIVALI(config)#
BORIVALI(config)#
*May 18 17:43:55.315: %LINEPROTO-5-UPDOWN: Line protocol o
BORIVALI(config)#int s1/1
BORIVALI(config-if)#ip address 172.16.1.2 255.255.255.0
BORIVALI(config-if)#exit
```

Step 2: Configure EIGRP.

Configure EIGRP between the SanJose1 and SanJose2 routers. (Note: If using an IOS prior to 15.0, use the no auto-summary router configuration command to disable automatic summarization. This command is the default beginning with IOS 15.)

```
SanJose1(config)# router eigrp 1
```

```
SanJose1(config-router)# network 172.16.0.0
```

```
ANDHERI(config)# router eigrp 1
ANDHERI(config-router)#network 172.16.0.0
```

```
SanJose2(config)# router eigrp 1
```

```
SanJose2(config-router)# network 172.16.0.0
```

```
BORIVALI(config)#router eigrp 1
BORIVALI(config-router)#network 172.16.0.0
```

Step 3: Configure IBGP and verify BGP neighbors.

- Configure IBGP between the SanJose1 and SanJose2 routers. On the SanJose1 router, enter the following configuration.

```
SanJose1(config)# router bgp 64512
```

```
SanJose1(config-router)# neighbor 172.16.32.1 remote-as 64512
```

```
SanJose1(config-router)# neighbor 172.16.32.1 update-source lo0
```

```
ANDHERI(config)#  
ANDHERI(config)#router bgp 64512  
ANDHERI(config-router)#neighbor 172.16.32.1 remote-as 64512  
ANDHERI(config-router)#neighbor 172.16.32.1 update-source lo0
```

If multiple pathways to the BGP neighbor exist, the router can use multiple IP interfaces to communicate with the neighbor. The source IP address therefore depends on the outgoing interface. The **update-source lo0** command instructs the router to use the IP address of the interface Loopback0 as the source IP address for all BGP messages sent to that neighbor.

- b. Complete the IBGP configuration on SanJose2 using the following commands.

```
SanJose2(config)# router bgp 64512
```

```
SanJose2(config-router)# neighbor 172.16.64.1 remote-as 64512
```

```
SanJose2(config-router)# neighbor 172.16.64.1 update-source lo0
```

```
BORIVALI(config)#router bgp 64512  
BORIVALI(config-router)#neighbor 172.16.64.1 remote-as 64512  
BORIVALI(config-router)#neighbor 172.16.64.1 update-source lo0
```

- c. Verify that SanJose1 and SanJose2 become BGP neighbors by issuing the **show ip bgp neighbors** command on SanJose1. View the following partial output. If the BGP state is not established, troubleshoot the connection.

```
SanJose2# show ip bgp neighbors
```

```
BORIVALI#show ip bgp neighbors  
BGP neighbor is 172.16.64.1, remote AS 64512,  
  BGP version 4, remote router ID 172.16.64.1  
  BGP state = Established, up for 00:00:47  
  Last read 00:00:47, last write 00:00:47, hol  
  Neighbor capabilities:  
    Route refresh: advertised and received(new  
    New ASN Capability: advertised and receive  
    Address family IPv4 Unicast: advertised an  
  Message statistics:  
    InQ depth is 0  
    OutQ depth is 0  
  
          Sent      Rcvd  
  Opens:        1        1  
  Notifications: 0        0  
  Updates:      0        0
```

Step 4: Configure EBGP and verify BGP neighbors.

d. Configure ISP to run EBGP with SanJose1 and SanJose2. Enter the following commands on ISP.

```
ISP(config)# router bgp 200
ISP(config-router)# neighbor 192.168.1.6 remote-as 64512
ISP(config-router)# neighbor 192.168.1.2 remote-as 64512
ISP(config-router)# network 192.168.100.0
```

```
ISP(config)#router bgp 200
ISP(config-router)#neighbor 192.168.1.6 remote-as 64512
ISP(config-router)#neighbor 192.168.1.2 remote-as 64512
ISP(config-router)#network 192.168.100.0
```

- e. Configure a discard static route for the 172.16.0.0/16 network. Any packets that do not have a more specific match (longer match) for a 172.16.0.0 subnet will be dropped instead of sent to the ISP. Later in this lab we will configure a default route to the ISP.

```
SanJose1(config)# ip route 172.16.0.0 255.255.0.0 null0
```

```
ANDHERI(config)#ip route 172.16.0.0 255.255.0.0 null0
ANDHERI(config)#
```

- f. Configure SanJose1 as an EBGP peer to ISP.

```
SanJose1(config)# router bgp 64512
```

```
SanJose1(config-router)# neighbor 192.168.1.5 remote-as 200
```

```
SanJose1(config-router)# network 172.16.0.0
```

```
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 192.168.1.5 remote-as 200
ANDHERI(config-router)#network 172.16.0.0
ANDHERI(config-router)#exit
```

- g. Use the **show ip bgp neighbors** command to verify that SanJose1 and ISP have reached the established state. Troubleshoot if necessary.

```
SanJose1# show ip bgp neighbors
```

```
ANDHERI#show ip bgp neighbors
BGP neighbor is 172.16.32.1, remote AS 64512, internal link
  BGP version 4, remote router ID 172.16.32.1
  BGP state = Established, up for 00:02:49
  Last read 00:00:56, last write 00:00:21, hold time is 180
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    New ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
```

Configure a discard static route for 172.16.0.0/16 on SanJose2 and as an EBGP peer to ISP.

```
SanJose2(config)# ip route 172.16.0.0 255.255.0.0 null0
```

```
SanJose2(config)# router bgp 64512
```

```
SanJose2(config-router)# neighbor 192.168.1.1 remote-as 200
```

```
SanJose2(config-router)# network 172.16.0.0
```

```
BORIVALI(config)#ip route 172.16.0.0 255.255.0.0 null0
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#neighbor 192.168.1.1 remote-as 200
BORIVALI(config-router)#
*May 18 18:00:01.031: %BGP-5-ADJCHANGE: neighbor 192.168.1.1 Up
BORIVALI(config-router)#network 172.16.0.0
```

Step 5: View BGP summary output.

In Step 4, the **show ip bgp neighbors** command was used to verify that SanJose1 and ISP had reached the established state. A useful alternative command is **show ip bgp summary**. The output should be similar to the following.

```
SanJose2# show ip bgp summary
```

```
BORIVALI# show ip bgp summary
BGP router identifier 172.16.32.1, local AS number 64512
BGP table version is 5, main routing table version 5
2 network entries using 264 bytes of memory
4 path entries using 208 bytes of memory
5/2 BGP path/bestpath attribute entries using 840 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
Bitfield cache entries: current 2 (at peak 2) using 64 bytes of memory
BGP using 1400 total bytes of memory
BGP activity 2/0 prefixes, 4/0 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/Pfx
172.16.64.1    4      64512     7     8       5     0     0 00:04:18      2
192.168.1.1    4      200      5     4       3     0     0 00:00:26      1
```

Step 6: Verify which path the traffic takes.

f. Clear the IP BGP conversation with the **clear ip bgp *** command on ISP. Wait for the conversations to reestablish with each SanJose router.

```
ISP# clear ip bgp *
```

```
*May 18 18:02:18.022
ISP#clear ip bgp *
ISP#
```

g. Test whether ISP can ping the loopback 0 address of 172.16.64.1 on SanJose1 and the serial link between SanJose1 and SanJose2, 172.16.1.1.

```
ISP# ping 172.16.64.1
```

```
ISP#ping 172.16.64.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
*May 18 18:02:42.575: %BGP-5-ADJCHANGE: neighbor 192.168.1.6 Up .....
Success rate is 0 percent (0/5)
```

ISP# **ping 172.16.1.1**

```
ISP#ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

- h. Now ping from ISP to the loopback 0 address of 172.16.32.1 on SanJose2 and the serial link between SanJose1 and SanJose2, 172.16.1.2.

ISP# **ping 172.16.32.1**

```
ISP# ping 172.16.32.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/31/48 ms
```

ISP# **ping 172.16.1.2**

```
ISP#ping 172.16.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/24/32 ms
```

- I. Issue the **show ip bgp** command on ISP to verify BGP routes and metrics.

ISP# **show ip bgp**

```
ISP#show ip bgp
BGP table version is 3, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 172.16.0.0        192.168.1.6          0        0 64512 i
*,> 192.168.1.2      192.168.1.2          0        0 64512 i
*> 192.168.100.0    0.0.0.0            0       32768 i
5500
```

- i. At this point, the ISP router should be able to get to each network connected to SanJose1 and SanJose2 from the loopback address 192.168.100.1. Use the extended **ping** command and specify the source address of ISP Lo0 to test.

ISP# ping 172.16.1.1 source 192.168.100.1

```
ISP#ping 172.16.1.1 source 192.168.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/38/48 ms
```

ISP# ping 172.16.32.1 source 192.168.100.1

```
ISP# ping 172.16.32.1 source 192.168.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.32.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/32/48 ms
ISP# ping 172.16.1.2 source 192.168.100.1
```

```
ISP#ping 172.16.1.2 source 192.168.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/28/48 ms
```

ISP# ping 172.16.64.1 source 192.168.100.1

```
ISP#ping 172.16.64.1 source 192.168.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.64.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.100.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/35/56 ms
```

Step 7: Configure the BGP next-hop-self feature.

j. Issue the following commands on the ISP

router. ISP(config)# **router bgp 200**

ISP(config-router)# **network 192.168.1.0 mask 255.255.255.252**

ISP(config-router)# **network 192.168.1.4 mask 255.255.255.252**

```
ISP(config)#router bgp 200
ISP(config-router)#network 192.168.1.0 mask 255.255.255.252
ISP(config-router)#network 192.168.1.4 mask 255.255.255.252
ISP(config-router)#exit
```

k. Issue the **show ip bgp** command to verify that the ISP is correctly injecting its own WAN links into BGP.

ISP# show ip bgp

```
ISP#show ip bgp
BGP table version is 5, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0        192.168.1.6          0          0 64512 i
*  192.168.1.2        192.168.1.2          0          0 64512 i
*> 192.168.1.0/30    0.0.0.0            0          32768 i
*> 192.168.1.4/30    0.0.0.0            0          32768 i
*> 192.168.100.0     0.0.0.0            0          32768 i
```

- l. Verify on SanJose1 and SanJose2 that the opposite WAN link is included in the routing table. The output from SanJose2 is as follows.

SanJose2# show ip route

```
BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
```

- m. To better understand the **next-hop-self** command we will remove ISP advertising its two WAN links and shutdown the WAN link between ISP and SanJose2. The only possible path from SanJose2 to ISP's 192.168.100.0/24 is through SanJose1.

ISP(config)# router bgp 200

ISP(config-router)# no network 192.168.1.0 mask 255.255.255.252

ISP(config-router)# no network 192.168.1.4 mask 255.255.255.252

ISP(config-router)# exit ISP(config)#

interface serial 0/0/1 ISP(config-if)#

shutdown

```
ISP(config)#router bgp 200
ISP(config-router)#no network 192.168.1.0 mask 255.255.255.252
ISP(config-router)#no network 192.168.1.4 mask 255.255.255.252
ISP(config-router)#exit
ISP(config)#int s1/1
ISP(config-if)#shutdown
```

- n. Display SanJose2's BGP table using the **show ip bgp** command and the IPv4 routing table with **show ip route**.

SanJose2# show ip bgp

```
BORIVALI#show ip bgp
BGP table version is 14, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*-> 172.16.0.0        0.0.0.0              0       32768 i
* i                 172.16.64.1           0       100      0 i
* i192.168.100.0    192.168.1.5           0       100      0 200 i
*>                  192.168.1.1           0       0 200 i
```

SanJose2# show ip route

```
BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C        172.16.32.0/24 is directly connected, Loopback0
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial1/1
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:11:47, Serial1/1
```

SanJose1(config)# router bgp 64512

SanJose1(config-router)# neighbor 172.16.32.1 next-hop-self

```
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 172.16.32.1 next-hop-self
ANDHERI(config-router)#exit
```

SanJose2(config)# router bgp 64512

SanJose2(config-router)# neighbor 172.16.64.1 next-hop-self

```
BORIVALI(config-router)#router bgp 64512
BORIVALI(config-router)#neighbor 172.16.64.1 next-hop-self
```

o. Reset BGP operation on either router with the **clear ip bgp ***

command. SanJose1# **clear ip bgp ***

```
ANDHERI#clear ip bgp *
```

SanJose2# **clear ip bgp ***

```
BORIVALI#clear ip bgp *
```

p. After the routers have returned to established BGP speakers, issue the **show ip bgp** command on

SanJose2 and notice that the next hop is now SanJose1 instead of ISP.

SanJose2# **show ip bgp**

```
BORIVALI#show ip bgp
BGP table version is 1, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop        Metric LocPrf Weight Path
* i172.16.0.0      172.16.64.1      0    100      0 i
* i192.168.100.0   172.16.64.1      0    100      0 200 i
```

- q. The **show ip route** command on SanJose2 now displays the 192.168.100.0/24 network because SanJose1 is the next hop, 172.16.64.1, which is reachable from SanJose2.

SanJose2# **show ip route**

```
BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level
       ia - IS-IS inter area, * - candidate default, U - per-user static r
       o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
```

- r. Before configuring the next BGP attribute, restore the WAN link between ISP and SanJose3. This will change the BGP table and routing table on both routers. For example, SanJose2's routing table shows 192.168.100.0/24 will now have a better path through ISP.

ISP(config)# **interface serial 0/0/1**

ISP(config-if)# **no shutdown**

```
ISP(config)#int s1/1
ISP(config-if)#no shutdown
```

SanJose2# **show ip route**

```

BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF i
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA exten
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2
      ia - IS-IS inter area, * - candidate default, U - per
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

    172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C      172.16.32.0/24 is directly connected, Loopback0
S      172.16.0.0/16 is directly connected, Null0
C      172.16.1.0/24 is directly connected, Serial1/1
D      172.16.64.0/24 [90/2297856] via 172.16.1.1, 00:15:05
    192.168.1.0/30 is subnetted, 1 subnets
C          192.168.1.0 is directly connected, Serial1/0
B      192.168.100.0/24 [20/0] via 192.168.1.1, 00:00:18

```

Step 8: Set BGP local preference.

- s. Because the local preference value is shared between IBGP neighbors, configure a simple route map that references the local preference value on SanJose1 and SanJose2. This policy adjusts outbound traffic to prefer the link off the SanJose1 router instead of the metered T1 off SanJose2.

```
SanJose1(config)# route-map PRIMARY_T1_IN
```

```
permit 10 SanJose1(config-route-map)# set
```

```
local-preference 150 SanJose1(config-route-map)# exit
```

```
SanJose1(config)# router bgp 64512
```

```
SanJose1(config-router)# neighbor 192.168.1.5 route-map PRIMARY_T1_IN in
```

```

ANDHERI(config)#route-map PRIMARY_T1_IN permit 10
ANDHERI(config-route-map)#set local-preference 150
ANDHERI(config-route-map)#exit
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_IN in

```

```
SanJose2(config)# route-map SECONDARY_T1_IN
```

```
permit 10 SanJose2(config-route-map)# set
```

```
local-preference 125 SanJose1(config-route-map)# exit
```

```
SanJose2(config)# router bgp 64512
```

```
SanJose2(config-router)# neighbor 192.168.1.1 route-map SECONDARY_T1_IN in
```

```

BORIVALI(config)#route-map SECONDARY_T1_IN permit 10
BORIVALI(config-route-map)#set local-preference 125
BORIVALI(config-route-map)#exit
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_IN in

```

- t. Use the **clear ip bgp * soft** command after configuring this new policy. When the conversations have been reestablished, issue the **show ip bgp** command on SanJose1 and SanJose2.

SanJose1# **clear ip bgp * soft**

```
ANDHERI#clear ip bgp * soft
```

SanJose2# **clear ip bgp * soft**

```
BORIVALI#clear ip bgp * soft
```

SanJose1# **show ip bgp**

```
ANDHERI#show ip bgp
BGP table version is 6, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0        172.16.32.1        0     100      0 i
*> 0.0.0.0           0.0.0.0             0         32768 i
*> 192.168.100.0    192.168.1.5        0     150      0 200 i
```

SanJose2# **show ip bgp**

```
BORIVALI#show ip bgp
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0        0.0.0.0             0         32768 i
* i                 172.16.64.1        0     100      0 i
* 192.168.100.0     192.168.1.1        0         0 200 i
*>i                172.16.64.1        0     150      0 200 i
```

Step 9: Set BGP MED.

- u. In the previous step we saw that SanJose1 and SanJose2 will route traffic for 192.168.100.0/24 using the link between SanJose1 and ISP. Examine what the return path ISP takes to reach AS 64512. Notice that the return path is different from the original path. This is known as asymmetric routing and is not necessarily an unwanted trait.

ISP# **show ip bgp**

```

ISP#show ip bgp
BGP table version is 11, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*  172.16.0.0        192.168.1.2          0          0 64512 i
*-> 192.168.1.6      192.168.1.6          0          0 64512 i
*> 192.168.100.0    0.0.0.0             0          32768 i

```

ISP# show ip route

```

ISP#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPFv2
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ex-
      E1 - OSPF external type 1, E2 - OSPF external typ-
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1
      ia - IS-IS inter area, * - candidate default, U -
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

B  172.16.0.0/16 [20/0] via 192.168.1.6, 00:04:56
  192.168.1.0/30 is subnetted, 2 subnets
C    192.168.1.0 is directly connected, Serial1/1
C    192.168.1.4 is directly connected, Serial1/0
C  192.168.100.0/24 is directly connected, Loopback0

```

- a. Use an extended **ping** command to verify this situation. Specify the **record** option and compare your output to the following. Notice the return path using the exit interface 192.168.1.1 to SanJose2

```
BORIVALI#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort,
Sending 5, 100-byte ICMP Echoes to 192.168.100.1, timeout is 2 seconds:
Packet sent with a source address of 172.16.32.1
Packet has IP options: Total option bytes= 39, padded length=40
Record route: <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
```

```
Reply to request 4 (8 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.100.1)
(192.168.1.5)
(172.16.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/47/68 ms
```

If you are unfamiliar with the **record** option, the important thing to note is that each IP address in brackets is an outgoing interface. The output can be interpreted as follows:

1. A ping that is sourced from 172.16.32.1 exits SanJose2 through s0/0/1, 172.16.1.2. It then arrives at the s0/0/1 interface for SanJose1.
2. SanJose1 S0/0/0, 192.168.1.6, routes the packet out to arrive at the S0/0/0 interface of ISP. 3. The target of 192.168.100.1 is reached: 192.168.100.1.
4. The packet is next forwarded out the S0/0/1, 192.168.1.1 interface for ISP and arrives at the S0/0/0 interface for SanJose2.
5. SanJose2 then forwards the packet out the last interface, loopback 0, 172.16.32.1.

Although the unlimited use of the T1 from SanJose1 is preferred here, ISP currently takes the link from SanJose2 for all return traffic.

- b. Create a new policy to force the ISP router to return all traffic via SanJose1. Create a second route map utilizing the MED (metric) that is shared between EBGP neighbors.

```
SanJose1(config)#route-map PRIMARY_T1_MED_OUT permit 10
```

```
SanJose1(config-route-map)#set Metric 50
```

```
SanJose1(config-route-map)#exit
```

```
SanJose1(config)#router bgp 64512
```

```
SanJose1(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out
```

```
ANDHERI(config)#route-map PRIMARY_T1_MED_OUT permit 10
ANDHERI(config-route-map)#set Metric 50
ANDHERI(config-route-map)#exit
ANDHERI(config)#router bgp 64512
ANDHERI(config-router)#neighbor 192.168.1.5 route-map PRIMARY_T1_MED_OUT out
```

```
SanJose2(config)#route-map SECONDARY_T1_MED_OUT permit 10
```

```
SanJose2(config-route-map)#set Metric 75
```

```
SanJose2(config-route-map)#exit
```

```
SanJose2(config)#router bgp 64512
```

```
SanJose2(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
```

```
BORIVALI(config)#route-map SECONDARY_T1_MED_OUT permit 10
BORIVALI(config-route-map)#set Metric 75
BORIVALI(config-route-map)#exit
BORIVALI(config)#router bgp 64512
BORIVALI(config-router)#neighbor 192.168.1.1 route-map SECONDARY_T1_MED_OUT out
```

- v. Use the **clear ip bgp * soft** command after issuing this new policy. Issuing the **show ip bgp** command as follows on SanJose1 or SanJose2 does not indicate anything about this newly defined policy.

```
SanJose1# clear ip bgp * soft
```

```
ANDHERI#clear ip bgp * soft
```

```
SanJose2# clear ip bgp * soft
```

```
BORIVALI#clear ip bgp * soft
```

```
SanJose1# show ip bgp
```

```

ANDHERII#show ip bgp
BGP table version is 6, local router ID is 172.16.64.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* i172.16.0.0        172.16.32.1        0     100      0 i
*> 0.0.0.0           0.0.0.0           0      32768 i
**> 192.168.100.0   192.168.1.5       0     150      0 200 i

```

SanJose2# show ip bgp

```

BORIVALI#show ip bgp
BGP table version is 5, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0        0.0.0.0           0      32768 i
* i                 172.16.64.1        0     100      0 i
* 192.168.100.0    192.168.1.1       0     125      0 200 i
*>i                172.16.64.1        0     150      0 200 i

```

Reissue an extended **ping** command with the **record** command. Notice the change in return path using the exit interface 192.168.1.5 to SanJose1.

```

BORIVALI#ping
Protocol [ip]:
Target IP address: 192.168.100.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.16.32.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]: record
Number of hops [ 9 ]:
Loose, Strict, Record, Timestamp, Verbose[RV]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.100.1, timeout is 2 sec
Packet sent with a source address of 172.16.32.1
Packet has IP options: Total option bytes= 39, padded length=40

```

```

Reply to request 3 (60 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.100.1)
(192.168.1.5)
(172.16.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list

Reply to request 4 (52 ms). Received packet has options
Total option bytes= 40, padded length=40
Record route:
(172.16.1.2)
(192.168.1.6)
(192.168.100.1)
(192.168.1.5)
(172.16.1.1)
(172.16.32.1) <*>
(0.0.0.0)
(0.0.0.0)
(0.0.0.0)
End of list

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/52/60 ms

```

ISP# show ip bgp

```

ISP#show ip bgp
BGP table version is 11, local router ID is 192.168.100.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* 172.16.0.0        192.168.1.2          0        0 64512 i
*-> 192.168.100.0   0.0.0.0            0        0 32768 i

```

Step 10: Establish a default route.

The final step is to establish a default route that uses a policy statement that adjusts to changes in the network.

- Configure ISP to inject a default route to both SanJose1 and SanJose2 using BGP using the **default-originate** command. This command does not require the presence of 0.0.0.0 in the ISP router. Configure the 10.0.0.0/8 network which will not be advertised using BGP. This network will be used to test the default route on SanJose1 and SanJose2.

ISP(config)# **router bgp 200**

ISP(config-router)# **neighbor 192.168.1.6 default-originate**

```
ISP(config-router)# neighbor 192.168.1.2 default-originat
```

```
ISP(config-router)# exit
```

```
ISP(config)# interface loopback 10
```

```
ISP(config-if)# ip address 10.0.0.1 255.255.255.0
```

```
ISP(config)#router bgp 200
ISP(config-router)#neighbor 192.168.1.6 default-originat
ISP(config-router)#neighbor 192.168.1.2 default-originat
ISP(config-router)#exit
```

```
ISP(config-if)#ip address 10.0.0.1 255.255.255.0
ISP(config-if)#exit
```

- b. Verify that both routers have received the default route by examining the routing tables on SanJose1 and SanJose2. Notice that both routers prefer the route between SanJose1 and ISP.

```
SanJose1# show ip route
```

```
ANDHERI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ext
      E1 - OSPF external type 1, E2 - OSPF external type
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1,
      ia - IS-IS inter area, * - candidate default, U -
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 m
D        172.16.32.0/24 [90/2297856] via 172.16.1.2, 02:14
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial1/1
C        172.16.64.0/24 is directly connected, Loopback0
B        192.168.100.0/24 [200/0] via 172.16.32.1, 01:44:43
B*       0.0.0.0/0 [200/0] via 172.16.32.1, 01:44:43
```

```
SanJose2# show ip route
```

```
BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OS
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ex
      E1 - OSPF external type 1, E2 - OSPF external typ
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1
      ia - IS-IS inter area, * - candidate default, U -
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 m
C        172.16.32.0/24 is directly connected, Loopback0
S        172.16.0.0/16 is directly connected, Null0
C        172.16.1.0/24 is directly connected, Serial1/1
D        172.16.64.0/24 [90/2297856] via 172.16.1.1, 02:1
      192.168.1.0/30 is subnetted, 1 subnets
C          192.168.1.0 is directly connected, Serial1/0
B        192.168.100.0/24 [20/0] via 192.168.1.1, 01:44:59
B*       0.0.0.0/0 [20/0] via 192.168.1.1, 01:45:00
```

- c. The preferred default route is by way of SanJose1 because of the higher local preference attribute configured on SanJose1 earlier.

SanJose2# **show ip bgp**

```
BORIVALI# show ip bgp
BGP table version is 9, local router ID is 172.16.32.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
* > 0.0.0.0          192.168.1.1        0     125      0 200 i
* > 172.16.0.0        0.0.0.0           0          32768 i
* i     172.16.64.1       172.16.64.1      0     100      0 i
* > 192.168.100.0     192.168.1.1        0     125      0 200 i
```

- d. Using the traceroute command verify that packets to 10.0.0.1 is using the default route through SanJose1.

SanJose2# **traceroute 10.0.0.1**

```
BORIVALI#traceroute 10.0.0.1
Type escape sequence to abort.
Tracing the route to 10.0.0.1

 1 192.168.1.1 [AS 200] 28 msec 32 msec 32 msec
 2 192.168.1.1 [AS 200] !H !H !H
```

- e. Next, test how BGP adapts to using a different default route when the path between SanJose1 and ISP goes down.

ISP(config)# **interface serial 0/0/0**

ISP(config-if)# **shutdown**

```
ISP(config)#int s1/0
ISP(config-if)#shutdown
```

- f. Verify that both routers are modified their routing tables with the default route using the path between SanJose2 and ISP.

SanJose1# **show ip route**

```
ANDHERI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA ext
      E1 - OSPF external type 1, E2 - OSPF external type
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1,
      ia - IS-IS inter area, * - candidate default, U -
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.16.32.1 to network 0.0.0.0

      172.16.0.0/16 is variably subnetted, 4 subnets, 2 ma
D  172.16.32.0/24 [00/2297856] via 172.16.1.2, 02:15
S  172.16.0.0/16 is directly connected, Null0
C  172.16.1.0/24 is directly connected, Serial1/1
C  172.16.64.0/24 is directly connected, Loopback0
B  192.168.100.0/24 [200/0] via 172.16.32.1, 01:45:58
B*  0.0.0.0/0 [200/0] via 172.16.32.1, 01:45:58
```

SanJose2# show ip route

```
BORIVALI#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B -
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA exte
      E1 - OSPF external type 1, E2 - OSPF external type
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1,
      ia - IS-IS inter area, * - candidate default, U - p
      o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

  172.16.0.0/16 is variably subnetted, 4 subnets, 2 mas
C    172.16.32.0/24 is directly connected, Loopback0
S    172.16.0.0/16 is directly connected, Null0
C    172.16.1.0/24 is directly connected, Serial1/1
D    172.16.64.0/24 [90/2297856] via 172.16.1.1, 02:15:
      192.168.1.0/30 is subnetted, 1 subnets
C      192.168.1.0 is directly connected, Serial1/0
B*   192.168.100.0/24 [20/0] via 192.168.1.1, 01:46:10
B*   0.0.0.0/0 [20/0] via 192.168.1.1, 01:46:10
```

- g. Verify the new path using the traceroute command to 10.0.0.1 from SanJose1. Notice the default route is now through SanJose2.

SanJose1# trace 10.0.0.1

```
ANDHERI#trace 10.0.0.1

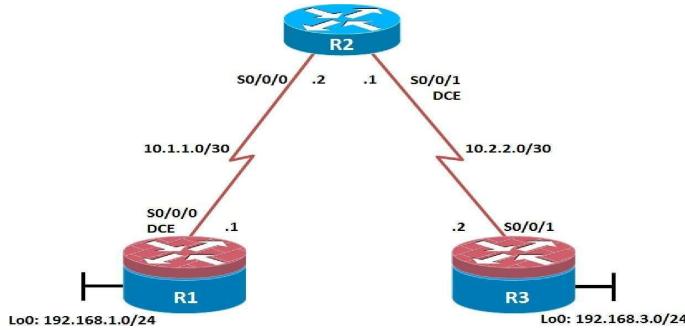
Type escape sequence to abort.
Tracing the route to 10.0.0.1

  1 172.16.1.2 64 msec 32 msec 32 msec
  2 192.168.1.1 [AS 200] 56 msec 28 msec 64 msec
  3 192.168.1.1 [AS 200] !H !H !H
ANDHERI#
```

Practical No - 4

Aim: Secure the Management Plane

Topology:



Objectives :

- Secure management access.
- Configure enhanced username password security.
- Enable AAA RADIUS authentication.
- Enable secure remote management.

Step 1: Configure loopbacks and assign addresses.

Cable the network as shown in the topology diagram. Erase the startup configuration and reload each router to clear previous configurations. Using the addressing scheme in the diagram, apply the IP addresses to the interfaces on the R1, R2, and R3 routers. You can copy and paste the following configurations into your routers to begin.

Router R1

```
interface Loopback 0
```

```
ip address 192.168.1.1 255.255.255.0  
exit
```

```
interface Serial0/0/0  
ip address 10.1.1.1 255.255.255.252  
no shutdown exit  
end
```

Router R2

```
interface Serial0/0/0  
ip address 10.1.1.2 255.255.255.252  
no shutdown exit  
interface Serial0/0/1  
ip address 10.2.2.1 255.255.255.252  
no shutdown exit  
end
```

Router R3

```
interface Loopback0  
ip address 192.168.3.1 255.255.255.0  
exit  
interface Serial0/0/1  
ip address 10.2.2.2 255.255.255.252  
no shutdown exit  
end
```

Step 2: Configure static routes.

```
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.2
```

```
R3(config)# ip route 0.0.0.0 0.0.0.0 10.2.2.1
```

```
(config)# ip route 192.168.1.0 255.255.255.0 10.1.1.1
```

```
R2(config)# ip route 192.168.3.0 255.255.255.0 10.2.2.2
foreach address {

192.168.1.1
10.1.1.1
10.1.1.2
10.2.2.1
10.2.2.2
192.168.3.1
} { ping $address }

R1# tclsh

R1(tcl)#foreach address {
+>(tcl)#192.168.1.1
+>(tcl)#10.1.1.1
+>(tcl)#10.1.1.2
+>(tcl)#10.2.2.1
+>(tcl)#10.2.2.2
+>(tcl)#192.168.3.1
+>(tcl)#{ { ping $address } Type
escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms Type
escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms Type
escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms Type
escape sequence to abort.
```

Sending 5, 100-byte ICMP Echos to 10.2.2.1, timeout is 2 seconds:

!!!!

Step 3: Secure management access.

1. On R1, use the **security passwords** command to set a minimum password length of 10 characters. R1(config)# **security passwords min-length 10**

2. Configure the enable secret encrypted password on both routers. R1(config)# **enable secret class12345**

3. Configure a console password and enable login for routers. For additional security, the **exec-timeout** command causes the line to log out after 5 minutes of inactivity. The **logging synchronous** command prevents console messages from interrupting command entry.

R1(config)# **line console 0**

R1(config-line)# **password ciscoconpass**

R1(config-line)# **exec-timeout 5 0**

R1(config-line)# **login**

R1(config-line)# **logging synchronous**

R1(config-line)# **exit**

Configure the password on the vty lines for router R1.

R1(config)# **line vty 0 4**

R1(config-line)# **password ciscovtypass**

R1(config-line)# **exec-timeout 5 0**

R1(config-line)# **login**

R1(config-line)# **exit**

4. The aux port is a legacy port used to manage a router remotely using a modem and is hardly ever used. Therefore, disable the aux port.

R1(config)# **line aux 0**

R1(config-line)# **no exec**

R1(config-line)# **end**

5. Enter privileged EXEC mode and issue the **show run** command. Can you read the enable secret password? Why or why not?

R1(config)

```
# service password-encryption  
R1(config)#
```

6. Configure a warning to unauthorized users with a message-of-the-day (MOTD) banner using the **banner motd** command. When a user connects to one of the routers, the MOTD banner appears before the login prompt. In this example, the dollar sign (\$) is used to start and end the message.

```
R1(config)# banner motd $Unauthorized access strictly prohibited!$
```

```
R1(config)# exit
```

Step 4: Configure enhanced username password security.

1. To create local database entry encrypted to level 4 (SHA256), use the **username name secret password** global configuration command. In global configuration mode, enter the following command:

```
R1(config)# username JR-ADMIN secret class12345
```

```
R1(config)# username ADMIN secret class54321
```

2. Set the console line to use the locally defined login

accounts. R1(config)# **line console 0**

```
R1(config-line)# login local
```

```
R1(config-line)# exit
```

3. Set the vty lines to use the locally defined login

accounts. R1(config)# **line vty 0 4**

```
R1(config-line)# login local
```

```
R1(config-line)# end
```

4. Repeat the steps 4a to 4c on R3.

1. To verify the configuration, telnet to R3 from R1 and login using the ADMIN local database account.

```
R1# telnet 10.2.2.2
```

Trying 10.2.2.2 ... Open

Unauthorized access strictly prohibited! User

Access Verification

Username: **ADMIN**

Password:

Step 5: Enabling AAA RADIUS Authentication with Local User for Backup.
Configure the specifics for the first RADIUS server located at 192.168.1.101. Use **RADIUS-1-pa55w0rd** as the server password.

```
R1(config)# radius server RADIUS-1  
R1(config-radius-server)# address ipv4 192.168.1.101  
R1(config-radius-server)# key RADIUS-1-pa55w0rd  
R1(config-radius-server)# exit
```

1. Configure the specifics for the second RADIUS server located at 192.168.1.102. Use **RADIUS-2-pa55w0rd** as the server password.

```
R1(config)# radius server RADIUS-2  
R1(config-radius-server)# address ipv4 192.168.1.102  
R1(config-radius-server)# key RADIUS-2-pa55w0rd  
R1(config-radius-server)# exit
```

2. Assign both RADIUS servers to a server group.

```
R1(config)# aaa group server radius RADIUS-GROUP  
R1(config-sg-radius)# server name RADIUS-1  
R1(config-sg-radius)# server name RADIUS-2  
R1(config-sg-radius)# exit
```

3. Enable the default AAA authentication login to attempt to validate against the server group. If they are not available, then authentication should be validated against the local database..

```
R1(config)# aaa authentication login default group RADIUS-GROUP local
```

4. Enable the default AAA authentication Telnet login to attempt to validate against the server group. If they are not available, then authentication should be validated against a case sensitive local database.

```
R1(config)# aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case  
Alter the VTY lines to use the TELNET-LOGIN AAA authentication method. R1(config)# line vty 0 4  
R1(config-line)# login authentication TELNET-LOGIN
```

```
R1(config-line)# exit
```

Repeat the steps 5a to 5g on R3.

5. To verify the configuration, telnet to R3 from R1 and login using the ADMIN local database account.

```
R1# telnet 10.2.2.2
```

Trying 10.2.2.2 ... Open

Unauthorized access strictly prohibited! User

Access Verification

Username: **admin**

Password:

Authentication failed

Username: **ADMIN**

Password:

Step 6: Enabling secure remote management using SSH.

1. SSH requires that a device name and a domain name be configured. Since the router already has a name assigned, configure the domain name.

```
R1(config)# ip domain-name ccnasecurity.com
```

2. The router uses the RSA key pair for authentication and encryption of transmitted SSH data. Although optional it may be wise to erase any existing key pairs on the router.

```
R1(config)# crypto key zeroize rsa
```

3. Generate the RSA encryption key pair for the router. Configure the RSA keys with **1024** for the number of modulus bits. The default is 512, and the range is from 360 to 2048.

```
R1(config)# crypto key generate rsa general-keys modulus 1024
```

The name for the keys will be: R1.ccnasecurity.com

% The key modulus size is 1024 bits

% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]

```
R1(config)#
```

```
Jan 10 13:44:44.711: %SSH-5-ENABLED: SSH 1.99 has been enabled
```

4. Cisco routers support two versions of SSH:
 - **SSH version 1 (SSHv1)**: Original version but has known vulnerabilities.

- **SSH version 2 (SSHv2):** Provides better security using the Diffie-Hellman key exchange and the strong integrity-checking message authentication code(MAC).

Configure SSH version 2 on R1.

```
R1(config)# ip ssh version 2
```

```
R1(config)#
```

5. Configure the vty lines to use only SSH connections.

```
R1(config)# line vty 04
```

```
R1(config-line)# transport input ssh
```

```
R1(config-line)# end
```

6. Verify the SSH configuration using the **show ip ssh**

command. R1# **show ip ssh**

SSH Enabled - version 2.0

Authentication timeout: 120 secs; Authentication retries: 3

Minimum expected Diffie Hellman key size : 1024 bits IOS

Keys in SECSH format(ssh-rsa, base64 encoded):

ssh-rsa

```
AAAAB3NzaC1yc2EAAAQABAAgQC3Lehh7ReYlgyDzls6wq+mFzxqzoaZFr9XGx+
Q/ yio
```

```
dFYw00hQo80tZy1W1Ff3Pz6q7Qi0y00urwddHZ0kBZceZK9EzJ6wZ+9a87KKDETWrGSLi6c8lE/y4K+
Z/oVrMMZk7bpTM1MFdP41YgkTf35utYv+TcqbsYo++KJiYk+xw==
```

7. Repeat the steps 6a to 6f on R3.
8. Although a user can SSH from a host using the SSH option of TeraTerm or PuTTY, a router can also SSH to another SSH enabled device. SSH to R3 from R1.

```
R1# ssh -l ADMIN 10.2.2.2
```

Password:

Unauthorized access strictly prohibited! R3>

R3> **en** Password:

R3

Device Configurations

Router R1

service password-encryption

hostname R1

security passwords min-length 10

enable secret 5 \$1\$t6eK\$FZ.JdmMLj8QSgNkpChyZz.

aaa new-model

aaa group server radius RADIUS-GROUP server

name RADIUS-1

server name RADIUS-2

aaa authentication login default group RADIUS-GROUP local

aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case

ip domain name ccnasecurity.com

username JR-ADMIN secret 5

\$1\$0u0q\$lwimCZIAuQtV4C1ezXL1S0 username ADMIN secret 5

\$1\$NSVD\$/YjzB7Auyes1sAt4qMfpd.

ip ssh version 2 interface

Loopback0 description R1

LAN

ip address 192.168.1.1 255.255.255.0

interface Serial0/0/0

description R1 --> R2

ip address 10.1.1.1 255.255.255.252

no fair-queue

ip route 0.0.0.0 0.0.0.0 10.1.1.2

radius server RADIUS-1

address ipv4 192.168.1.101 auth-port 1645 acct-port 1646

key 7 107C283D2C2221465D493A2A717D24653017

radius server RADIUS-2

address ipv4 192.168.1.102 auth-port 1645 acct-port 1646

key 7 03367A2F2F3A12011C44090442471C5C162E

banner motd ^CUnauthorized access strictly prohibited!^C

```
line con 0
exec-timeout 5 0
password 7 070C285F4D061A0A19020A1F17
logging synchronous line
aux 0
no exec
password 7 060506324F411F0D1C0713181F
login authentication TELNET-LOGIN
transport input ssh
end
Router R2
hostname R2
enable secret 5 $1$DJS7$xvJDW87zLs8pSJDFUlCPB1
interface Serial0/0/0
ip address 10.1.1.2 255.255.255.252
no fair-queue
interface Serial0/0/1
ip address 10.2.2.1 255.255.255.252
clock rate 128000

ip route 192.168.1.0 255.255.255.0 10.1.1.1
ip route 192.168.3.0 255.255.255.0 10.2.2.2

line con 0
exec-timeout 0 0 logging
synchronous

line vty 0 4 password
cisco login
```

```
end
Router R3
service password-encryption
hostname R3
security passwords min-length 10
enable secret 5 $1$5OY4$4J6VFlvGNKjwQ8XtajgUk1
aaa new-model
aaa group server radius RADIUS-GROUP server
name RADIUS-1
server name RADIUS-2
aaa authentication login default group RADIUS-GROUP local
aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case
ip domain name ccnasecurity.com
```

```
username JR-ADMIN secret 5 $1$b4m1$RVmjL9S3gxKh1xr8qzNqr/
username ADMIN secret 5 $1$zGV7$pVgSEbinvXQ7f7uyxeKBjip
ssh version 2
```

```
interface Loopback0
description R3 LAN
ip address 192.168.3.1 255.255.255.0
```

```
interface Serial0/0/1
description R3 --> R2
ip address 10.2.2.2 255.255.255.252
```

```
ip route 0.0.0.0 0.0.0.0 10.2.2.1

radius server RADIUS-1
address ipv4 192.168.1.101 auth-port 1645 acct-port 1646
key 7 01212720723E354270015E084C5000421908
```

```
radius server RADIUS-2
```

```
address ipv4 192.168.1.102 auth-port 1645 acct-port 1646  
key 7 003632222D6E384B5D6C5C4F5C4C1247000F  
banner motd ^CUnauthorized access strictly prohibited!^C
```

```
line con 0
```

```
exec-timeout 5 0
```

```
password 7 104D000A0618110402142B3837
```

```
logging synchronous
```

```
line aux 0 no
```

```
exec
```

```
line vty 0 4
```

```
exec-timeout 5 0
```

```
password 7 070C285F4D060F110E020A1F17
```

```
login authentication TELNET-LOGIN
```

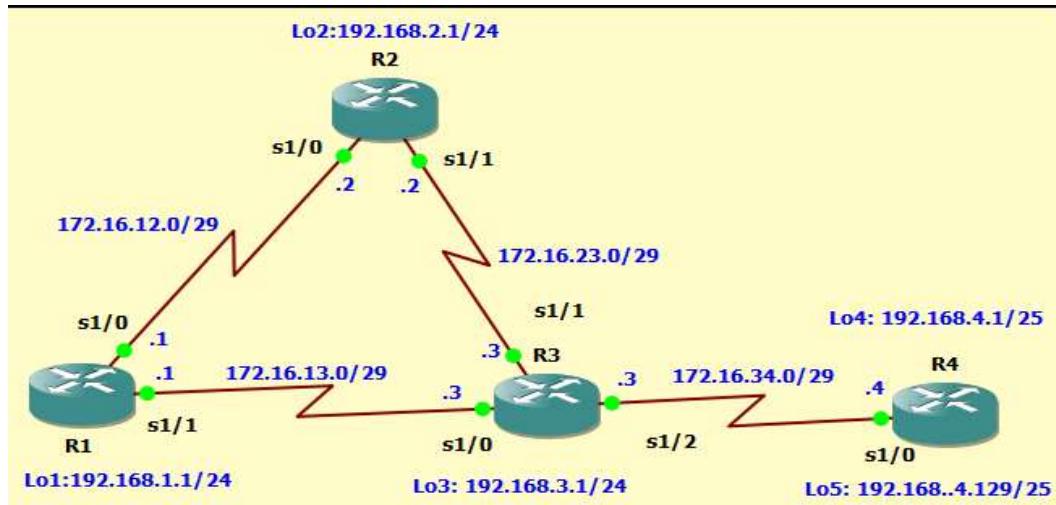
```
transport input ssh
```

```
end
```

Practical No - 5

Aim : Configure and Verify Path Control Using PBR

Topology :



Objectives

- Configure and verify policy-based routing.
- Select the required tools and commands to configure policy-based routing operations.
- Verify the configuration and operation by using the proper show and debug commands.

Step 1: Configure loopbacks and assign addresses.

- Cable the network as shown in the topology diagram. Erase the startup configuration, and reload each router to clear previous configurations.
- Using the addressing scheme in the diagram, create the loopback interfaces and apply IP addresses to these and the serial interfaces on R1, R2, R3, and R4. On the serial interfaces connecting R1 to R3 and R3 to R4, specify the bandwidth as 64 Kb/s and set a clock rate on the DCE using the **clock rate 64000** command. On the serial interfaces connecting R1 to R2 and R2 to R3, specify the bandwidth as 128 Kb/s and set a clock rate on the DCE using the **clock rate 128000** command.

You can copy and paste the following configurations into your routers to begin.

Note: Depending on the router model, interfaces might be numbered differently than those listed. You might need to alter them accordingly.

Router R1

```
interface Lo1
ip address 192.168.1.1 255.255.255.0
interface Serial0/0/0
ip address 172.16.12.1 255.255.255.248
```

no shutdown interface

Serial0/0/1

ip address 172.16.13.1 255.255.255.248

no shutdown End

```
R1(config)#int Lo1
```

```
R1(config-if)#ip address 192.168.1.1 255.255.255.0
R1(config-if)#int s1/0
R1(config-if)#ip address 172.16.12.1 255.255.255.248
R1(config-if)#no shutdown
```

```
R1(config-if)#int s1/1
R1(config-if)#ip address 172.16.13.1 255.255.255.248
R1(config-if)#no shutdown
*May 19 23:06:21.987: %LINEPROTO-5-UPDOWN: Line protocol
R1(config-if)#no shutdown
*May 19 23:06:21.987: %LINEPROTO-5-UPDOWN: Line protocol
```

Router R2

interface Lo2

ip address 192.168.2.1 255.255.255.0

interface Serial0/0/0

ip address 172.16.12.2 255.255.255.248

no shutdown

interface Serial0/0/1

ip address 172.16.23.2 255.255.255.248

no shutdown End

```
R2(config)#int Lo2
R2(config-if)#
*May 19 23:06:13.083: %LINEPROTO-5-UPDOWN: Line protocol
R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#int s1/0
R2(config-if)#ip address 172.16.12.2 255.255.255.248
R2(config-if)#no shutdown
```

```
R2(config)#int s1/1
R2(config-if)#ip address 172.16.23.2 255.255.255.248
R2(config-if)#no shutdown
```

Router R3

```
interface Lo3
  ip address 192.168.3.1 255.255.255.0
interface Serial0/0/0
  ip address 172.16.13.3 255.255.255.248
  no shutdown interface
Serial0/0/1
  ip address 172.16.23.3 255.255.255.248
  no shutdown interface
Serial0/1/0
  ip address 172.16.34.3 255.255.255.248
  no shutdown End
```

```
R3(config)#int Lo3
R3(config-if)#
*May 19 23:07:08.351: %LINEPROTO-5-UPDOWN: Line protocol on interface Lo3 transitioned from up to down
R3(config-if)#ip address 192.168.3.1 255.255.255.0
R3(config-if)#int s1/0
R3(config-if)#ip address 172.16.13.3 255.255.255.248
R3(config-if)#no shutdown

R3(config-if)#int s1/1
R3(config-if)#ip address 172.16.23.3 255.255.255.248
R3(config-if)#no shutdown

R3(config-if)#int s1/2
R3(config-if)#ip address 172.16.34.3 255.255.255.248
R3(config-if)#no shutdown
R3(config-if)#exit
```

Router R4

```
interface Lo4
  ip address 192.168.4.1 255.255.255.128
interface Lo5
  ip address 192.168.4.129 255.255.255.128
interface Serial0/0/0
  ip address 172.16.34.4 255.255.255.248
  no shutdown End
```

```
R4(config)#int lo4
R4(config-if)#
*May 19 23:08:16.239: %LINEPROTO-5-UPDOWN: Line protocol on interface Lo4 changed state to up
R4(config-if)#ip address 192.168.4.1 255.255.255.128
R4(config-if)#interface Lo5
R4(config-if)#
*May 19 23:08:32.527: %LINEPROTO-5-UPDOWN: Line protocol on interface Lo5 changed state to up
R4(config-if)#ip address 192.168.4.129 255.255.255.128
R4(config-if)#int s1/0
R4(config-if)#ip address 172.16.34.4 255.255.255.248
R4(config-if)#no shutdown
```

3. Verify the configuration with the **show ip interface brief**, **show protocols**, and **show interfaces description** commands. The output from router R3 is shown here as an example.

R3# show ip interface brief

```
R3#show ip interface brief
Interface          IP-Address      OK? Method Status       Protocol
FastEthernet0/0    unassigned     YES unset administratively down down
Serial1/0          172.16.13.3   YES manual up        up
Serial1/1          172.16.23.3   YES manual up        up
Serial1/2          172.16.34.3   YES manual up        up
Serial1/3          unassigned     YES unset administratively down down
Loopback3          192.168.3.1   YES manual up        up
```

R3# show protocols

```
R3#show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is administratively down, line protocol is down
  Internet address is 172.16.13.3/29
Serial1/0 is up, line protocol is up
  Internet address is 172.16.13.3/29
Serial1/1 is up, line protocol is up
  Internet address is 172.16.23.3/29
Serial1/2 is up, line protocol is up
  Internet address is 172.16.34.3/29
Serial1/3 is administratively down, line protocol is down
Loopback3 is up, line protocol is up
  Internet address is 192.168.3.1/24
```

R3# show interfaces description

```
R3#show interfaces description
Interface          Status      Protocol Description
Fa0/0              admin down  down
Se1/0              up         up
Se1/1              up         up
Se1/2              up         up
Se1/3              admin down  down
```

Step 3: Configure basic EIGRP.

- Implement EIGRP AS 1 over the serial and loopback interfaces as you have configured it for the other EIGRP labs.
- Advertise networks 172.16.12.0/29, 172.16.13.0/29, 172.16.23.0/29, 172.16.34.0/29, 192.168.1.0/24,

192.168.2.0/24, 192.168.3.0/24, and 192.168.4.0/24 from their respective routers.
You can copy and paste the following configurations into your routers.

Router R1

```
router eigrp 1  
network 192.168.1.0  
network 172.16.12.0.0.0.7  
network 172.16.13.0.0.0.7  
no auto-summary
```

```
R1(config)#router eigrp 1  
R1(config-router)#network 192.168.1.0  
R1(config-router)#network 172.16.12.0 0.0.0.7  
R1(config-router)#network 172.16.13.0 0.0.0.7  
R1(config-router)#no auto-summary
```

Router R2

```
router eigrp 1  
network 192.168.2.0  
network 172.16.12.0.0.0.7  
network 172.16.23.0.0.0.7  
no auto-summary
```

```
R2(config)#router eigrp 1  
R2(config-router)#network 192.168.2.0  
R2(config-router)#network 172.16.12.0 0.0.0.7  
R2(config-router)#network 172.16.23.0 0.0.0.7  
R2(config-router)#no auto-summary
```

Router R3

```
router eigrp 1  
network 192.168.3.0  
network 172.16.13.0.0.0.7  
network 172.16.23.0.0.0.7  
network 172.16.34.0.0.0.7  
no auto-summary
```

```
R3(config)#router eigrp 1  
R3(config-router)#network 192.168.3.0  
R3(config-router)#network 172.16.13.0 0.0.0.7  
R3(config-router)#network 172.16.23.0 0.0.0.7  
R3(config-router)#network 172.16.34.0 0.0.0.7  
R3(config-router)#no auto-summary
```

Router R4 router eigrp

1

network 192.168.4.0

network 172.16.34.0 0.0.0.7

no auto-summary

```
R4(config)#router eigrp 1
R4(config-router)#network 192.168.4.0
R4(config-router)#network 172.16.34.0 0.0.0.7
R4(config-router)#no auto-summary
```

You should see EIGRP neighbor relationship messages being generated.

Step 4: Verify EIGRP connectivity.

cc. Verify the configuration by using the **show ip eigrp neighbors** command to check which routers have EIGRP adjacencies.

R1# show ip eigrp neighbors

```
R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H  Address          Interface      Hold Uptime    SRTT    RTO   Q  Seq
   (sec)           (ms)          Cnt Num
1  172.16.13.3      Se1/1        11 00:00:31   26    200   0  18
0  172.16.12.2      Se1/0        12 00:00:44   37    222   0  13
R1#
```

R2# show ip eigrp neighbors

```
R2#show ip eigrp neighbors
*May 19 23:13:42.783: %SYS-5-CONFIG_I: Configured from console by console
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H  Address          Interface      Hold Uptime    SRTT    RTO   Q  Seq
   (sec)           (ms)          Cnt Num
1  172.16.23.3      Se1/1        11 00:00:50   41    246   0  20
0  172.16.12.1      Se1/0        11 00:01:04   30    200   0  18
R2#
```

R3# show ip eigrp neighbors

```
R3#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H  Address          Interface      Hold Uptime    SRTT    RTO   Q  Seq
   (sec)           (ms)          Cnt Num
2  172.16.34.4      Se1/2        12 00:00:44   48    288   0  6
1  172.16.23.2      Se1/1        11 00:00:58   26    200   0  19
0  172.16.13.1      Se1/0        12 00:00:58   281   1686  0  20
R3#
```

R4# show ip eigrp neighbors

```
R4#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H  Address          Interface      Hold Uptime    SRTT    RTO   Q  Seq
   (sec)           (ms)          Cnt Num
0  172.16.34.3      Se1/0        10 00:00:55   23    200   0  26
R4#
```

Run the following Tcl script on all routers to verify full connectivity.

R1# tclsh

```
R1#tclsh
R1(tcl)#foreach address {
+>(tcl)#172.16.12.1
+>(tcl)#172.16.12.2
+>(tcl)#172.16.13.1
+>(tcl)#172.16.13.3
+>(tcl)#172.16.23.2
+>(tcl)#172.16.23.3
+>(tcl)#172.16.34.3
+>(tcl)#172.16.34.4
+>(tcl)#192.168.1.1
+>(tcl)#192.168.2.1
+>(tcl)#192.168.3.1
+>(tcl)#192.168.4.1
+>(tcl)#} { ping $address }

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/61/76 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/27/40 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 40/58/80 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.13.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/31/44 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/32 ms
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/40/48 ms
Type escape sequence to abort.
```

Step 5: Verify the current path.

Before you configure PBR, verify the routing table on R1.

On R1, use the **show ip route** command. Notice the next-hop IP address for all networks discovered by EIGRP.

R1# show ip route

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS le
      ia - IS-IS inter area, * - candidate default, U - per-user stati
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  172.16.0.0/29 is subnetted, 4 subnets
D    172.16.34.0 [90/2681856] via 172.16.13.3, 00:01:50, Serial1/1
D    172.16.23.0 [90/2681856] via 172.16.13.3, 00:01:50, Serial1/1
      [90/2681856] via 172.16.12.2, 00:01:50, Serial1/0
C    172.16.12.0 is directly connected, Serial1/0
C    172.16.13.0 is directly connected, Serial1/1
D    192.168.4.0/24 [90/2809856] via 172.16.13.3, 00:01:38, Serial1/1
C    192.168.1.0/24 is directly connected, Loopback1
D    192.168.2.0/24 [90/2297856] via 172.16.12.2, 00:01:50, Serial1/0
D    192.168.3.0/24 [90/2297856] via 172.16.13.3, 00:01:50, Serial1/1
```

R4# traceroute 192.168.1.1 source 192.168.4.1

```
R4#traceroute 192.168.1.1 source 192.168.4.1

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 36 msec 32 msec 32 msec
  2 172.16.13.1 28 msec 56 msec 84 msec
```

R4# traceroute 192.168.1.1 source 192.168.4.129

```
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

  1 172.16.34.3 44 msec 28 msec 28 msec
  2 172.16.13.1 64 msec 28 msec 64 msec
```

On R3, use the **show ip route** command and note that the preferred route from R3 to R1 LAN 192.168.1.0/24 is via R2 using the R3 exit interface S0/0/1.

R3# show ip route

```
R3#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS le
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set
```

```
172.16.0.0/29 is subnetted, 4 subnets
C    172.16.34.0 is directly connected, Serial1/2
C    172.16.23.0 is directly connected, Serial1/1
D    172.16.12.0 [90/2681856] via 172.16.23.2, 00:02:29, Serial1/1
      [90/2681856] via 172.16.13.1, 00:02:29, Serial1/0
C    172.16.13.0 is directly connected, Serial1/0
D    192.168.4.0/24 [90/2297856] via 172.16.34.4, 00:02:17, Serial1/2
D    192.168.1.0/24 [90/2297856] via 172.16.13.1, 00:02:29, Serial1/0
D    192.168.2.0/24 [90/2297856] via 172.16.23.2, 00:02:29, Serial1/1
C    192.168.3.0/24 is directly connected, Loopback3
```

R3#

On R3, use the **show interfaces serial 0/0/0** and **show interfaces s0/0/1** commands.

R3# show interfaces serial0/0/0

```
R3#show int s1/0
Serial1/0 is up, line protocol is up
  Hardware is M4T
  Internet address is 172.16.13.3/29
  MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
```

```
Routing Descriptor Blocks:
172.16.13.1 (Serial1/0), from 172.16.13.1, Send flag is 0x0
  Composite metric is (2297856/128256), Route is Internal
  Vector metric:
    Minimum bandwidth is 1544 Kbit
    Total delay is 25000 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
172.16.23.2 (Serial1/1), from 172.16.23.2, Send flag is 0x0
```

Confirm that R3 has a valid route to reach R1 from its serial 0/0/0 interface using the **show ip eigrp topology 192.168.1.0** command.

R3# show ip eigrp topology 192.168.1.0

```
R3#show ip eigrp topology 192.168.1.0
IP-EIGRP (AS 1): Topology entry for 192.168.1.0/24
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is
  Routing Descriptor Blocks:
  172.16.13.1 (Serial1/0), from 172.16.13.1, Send flag is 0x0
    Composite metric is (2297856/128256), Route is Internal
    Vector metric:
      Minimum bandwidth is 1544 Kbit
      Total delay is 25000 microseconds
      Reliability is 255/255
      Load is 1/255
      Minimum MTU is 1500
      Hop count is 1
  172.16.23.2 (Serial1/1), from 172.16.23.2, Send flag is 0x0
```

Step 6: Configure PBR to provide path control.

The steps required to implement path control include the following:

- Choose the path control tool to use. Path control tools manipulate or bypass the IP routing table.
For PBR, **route-map** commands are used.
- Implement the traffic-matching configuration, specifying which traffic will be manipulated. The **match** commands are used within route maps.
- Define the action for the matched traffic using **set** commands within route maps.
- Apply the route map to incoming traffic.

As a test, you will configure the following policy on router R3:

- All traffic sourced from R4 LAN A must take the R3 --> R2 --> R1 path.
 - All traffic sourced from R4 LAN B must take the R3 --> R1 path.
- hh. On router R3, create a standard access list called **PBR-ACL** to identify the R4 LAN B network.

```
R3(config)# ip access-list standard PBR-ACL
```

```
R3(config-std-nacl)# remark ACL matches R4 LAN B
```

```
traffic R3(config-std-nacl)# permit 192.168.4.128 0.0.0.127
```

```
R3(config- std-nacl)# exit
```

```
R3(config)#ip access-list standard PBR-ACL
R3(config-std-nacl)#remark ACL matches R4 LAN B traffic
R3(config-std-nacl)#permit 192.168.4.128 0.0.0.127
R3(config-std-nacl)#exit
```

```
R3(config)#
```

- ii. Create a route map called **R3-to-R1** that matches PBR-ACL and sets the next-hop interface to the R1 serial 0/0/1 interface.

```
R3(config)# route-map R3-to-R1 permit
```

```
R3(config-route-map)# description RM to forward LAN B traffic to R1
```

```
R3(config-route-map)# match ip address PBR-ACL
```

```
R3(config-route-map)# set ip next-hop 172.16.13.1
```

```
R3(config-route-map)# exit
```

```
R3(config)#route-map R3-to-R1 permit
R3(config-route-map)#match ip address PBR-ACL
R3(config-route-map)#set ip next-hop 172.16.13.1
R3(config-route-map)#exit
```

- jj. Apply the R3-to-R1 route map to the serial interface on R3 that receives the traffic from R4. Use the **ip policy route-map** command on interface S0/1/0.

```
R3(config)# interface s0/1/0
```

```
R3(config-if)# ip policy route-map R3-to-R1
```

```
R3(config-if)# end
```

```
R3(config)#int s1/2
R3(config-if)#ip policy route-map R3-to-R1
R3(config-if)#end
```

kk. On R3, display the policy and matches using the **show route-map** command.

```
R3# show route-map
```

```
R3#show route-map
route-map R3-to-R1, permit, sequence 10
  Match clauses:
    ip address (access-lists): PBR-ACL
  Set clauses:
    ip next-hop 172.16.13.1
  Policy routing matches: 0 packets, 0 bytes
R3#
```

Step 7: Test the policy.

ll. On R3, create a standard ACL which identifies all of the R4 LANs.

```
R3# conf t
```

Enter configuration commands, one per line. End with CNTL/Z.

```
R3(config)# access-list 1 permit 192.168.4.0 0.0.0.255 R3(config)#
exit
```

```
R3#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R3(config)#access-list 1 permit 192.168.4.0 0.0.0.255
R3(config)#exit
```

mm. Enable PBR debugging only for traffic that matches the R4 LANs.

```
R3# debug ip policy ?
```

```
R3#debug ip policy ?
<1-199>  Access list
dynamic  dynamic PBR
<cr>
```

```
R3# debug ip policy 1
```

```
R3#debug ip policy 1
Policy routing debugging is on for access list 1
```

nn. Test the policy from R4 with the **traceroute** command, using R4 LAN A as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.1
```

```
R4#traceroute 192.168.1.1 source 192.168.4.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1

 1 172.16.34.3 40 msec 12 msec 32 msec
 2 172.16.13.1 60 msec 48 msec 88 msec
```

```
R3#
*May 19 23:17:36.819: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1,
*May 19 23:17:36.851: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1,
*May 19 23:17:36.879: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1,
*May 19 23:17:36.915: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1,
g
*May 19 23:17:36.971: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1,
g
*May 19 23:17:37.031: IP: s=192.168.4.1 (Serial1/2), d=192.168.1.1
R3#, len 28, FIB policy rejected(no match) - normal forwarding
R3#
```

oo. Test the policy from R4 with the **traceroute** command, using R4 LAN B as the source network.

```
R4# traceroute 192.168.1.1 source 192.168.4.129
```

```
R4#traceroute 192.168.1.1 source 192.168.4.129

Type escape sequence to abort.
Tracing the route to 192.168.1.1

 1 172.16.34.3 40 msec 28 msec 32 msec
 2 172.16.13.1 60 msec 64 msec 32 msec
```

```
R3# 168.4.129 (Serial1/2), d=192.168.1.1, len 28, FIB policy match
*May 19 23:17:55.763: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.763: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.823: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.823: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.827: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.883: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.883: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
*May 19 23:17:55.887: IP: s=192.168.4.129 (Serial1/2), d=192.168.1.1,
```

pp. On R3, display the policy and matches using the **show route-map** command.

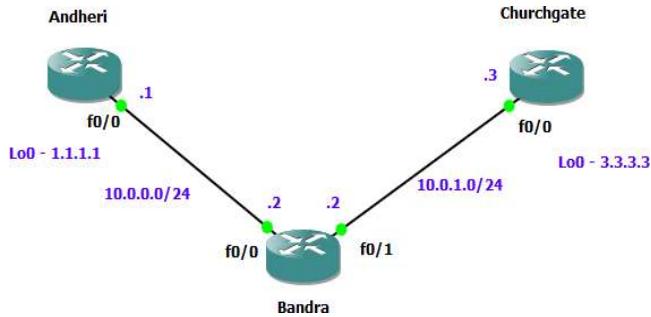
```
R3#show route-map
route-map R3-to-R1, permit, sequence 10
 Match clauses:
   ip address (access-lists): PBR-ACL
 Set clauses:
   ip next-hop 172.16.13.1
 Policy routing matches: 6 packets, 192 bytes
```

```
R3# show route-map
```

Practical No - 6

Aim: Cisco MPLS Configuration

Topology :



Step 1 – IP addressing of MPLS Core and OSPF

First bring 3 routers into your topology R1, R2, R3 position them as below. We are going to address the routers and configure ospf to ensure loopback to loopback connectivity between R1 and R

```
Andheri(config)#int lo0
Andheri(config-if)#ip add 1.1.1.1 255.255.255.255
Andheri(config-if)#ip ospf 1 area 0
Andheri(config-if)#
Andheri(config-if)#int f0/0
Andheri(config-if)#ip add 10.0.0.1 255.255.255.0
Andheri(config-if)#no shut
Andheri(config-if)#ip ospf 1 area 0

Bandra(config)#int lo0
Bandra(config-if)#
Bandra(config-if)#ip add 2.2.2.2 255.255.255.255
Bandra(config-if)#ip ospf 1 area 0
Bandra(config-if)#
Bandra(config-if)#int f0/0
Bandra(config-if)#ip add 10.0.0.2 255.255.255.0
Bandra(config-if)#no shut
Bandra(config-if)#ip ospf 1 area 0
Bandra(config-if)#
Bandra(config-if)#int f0/1
Bandra(config-if)#ip add 10.0.1.2 255.255.255.0
Bandra(config-if)#no shut
Bandra(config-if)#ip ospf 1 area 0
```

```
Churchgate(config)#int lo0
Churchgate(config-if)#ip add 3.3.3.3 255.255.255.255
Churchgate(config-if)#ip ospf 1 area 0
Churchgate(config-if)#
Churchgate(config-if)#int f0/0
Churchgate(config-if)#ip add 10.0.1.3 255.255.255.0
Churchgate(config-if)#no shut
Churchgate(config-if)#ip ospf 1 area 0
```

You should now have full ip connectivity between R1, R2, R3 to verify this we need to see if we can ping between the loopbacks of R1 and R3

```
Andheri#ping 3.3.3.3 source lo0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/52/64 ms
```

Step 2 – Configure LDP on all the interfaces in the MPLS Core

In order to run MPLS you need to enable it, there are two ways to do this.

At each interface enter the mpls ip command

Under the ospf process use the mpls ldp autoconfig command

```
Andheri(config)#router ospf 1
Andheri(config-router)#mpls ldp autoconfig
```

```
Bandra(config)#router ospf 1
Bandra(config-router)#mpls ldp autoconfig
```

```
Churchgate(config)#router ospf 1
Churchgate(config-router)#mpls ldp autoconfig
```

You should see log messages coming up showing the LDP neighbors are up.

```

Bandra#
*May 29 17:03:09.559: %SYS-5-CONFIG_I: Configured from console by console
Bandra#
*May 29 17:03:28.631: %LDP-5-NBRCHG: LDP Neighbor 3.3.3.3:0 (2) is UP

```

To verify the mpls interfaces the command is very simple – sh mpls interface

This is done on R2 and you can see that both interfaces are running mpls and using LDP

```

Bandra#sh mpls int
Interface          IP           Tunnel   BGP Static Operational
FastEthernet0/0    Yes (ldp)    No        No  No      Yes
FastEthernet0/1    Yes (ldp)    No        No  No      Yes
Bandra#

```

You can also verify the LDP neighbors with the sh mpls ldp neighbors command.

```

Bandra#sh mpls ldp neigh
Peer LDP Ident: 1.1.1.1:0; Local LDP Ident 2.2.2.2:0
  TCP connection: 1.1.1.1.646 - 2.2.2.2.25712
  State: Oper; Msgs sent/rcvd: 9/9; Downstream
  Up time: 00:01:23
  LDP discovery sources:
    FastEthernet0/0, Src IP addr: 10.0.0.1
    Addresses bound to peer LDP Ident:
      10.0.0.1      1.1.1.1
Peer LDP Ident: 3.3.3.3:0; Local LDP Ident 2.2.2.2:0
  TCP connection: 3.3.3.3.50470 - 2.2.2.2.646
  State: Oper; Msgs sent/rcvd: 8/8; Downstream
  Up time: 00:00:54
  LDP discovery sources:
    FastEthernet0/1, Src IP addr: 10.0.1.3
    Addresses bound to peer LDP Ident:
      10.0.1.3      3.3.3.3

```

One more verification to confirm LDP is running ok is to do a trace between R1 and R3 and verify if you get MPLS Labels show up in the trace.

```

Andheri#trace 3.3.3.3
Type escape sequence to abort.
Tracing the route to 3.3.3.3

 1 10.0.0.2 [MPLS: Label 17 Exp 0] 20 msec 60 msec 60 msec
 2 10.0.1.3 60 msec 60 msec 60 msec

```

Step 3 – MPLS BGP Configuration between R1 and R3

We need to establish a Multi Protocol BGP session between R1 and R3 this is done by configuring the vpnv4 address family as below

```

Andheri(config)#router bgp 1
Andheri(config-router)#neighbor 3.3.3.3 remote-as 1
Andheri(config-router)#neighbor 3.3.3.3 update-source Loopback0
Andheri(config-router)#no auto-summary
Andheri(config-router)#!
Andheri(config-router)#address-family vpnv4
Andheri(config-router-af)#neighbor 3.3.3.3 activate

```

```

Churchgate(config)#router bgp 1
Churchgate(config-router)#neighbor 1.1.1.1 remote-as 1
Churchgate(config-router)#neighbor 1.1.1.1
*May 29 17:06:19.459: %BGP-5-ADJCHANGE: neighbor 1.1.1.1 Up
Churchgate(config-router)#neighbor 1.1.1.1 update-source loopback 0
Churchgate(config-router)#no auto-summary
Churchgate(config-router)#address-family vpng4
Churchgate(config-router-af)#neighbor 1.1.1.1 activate

```

To verify the BGP session between R1 and R3 issue the command sh bgp vpng4 unicast all summary

```

Andheri#sh bgp vpng4 unicast all summary
BGP router identifier 1.1.1.1, local AS number 1
BGP table version is 1, main routing table version 1

Neighbor          V      AS MsgRcvd  MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
3.3.3.3           4      1      5       6       1       0     0 00:00:30      0

```

Step 4 – Add two more routers, create VRFs

We will add two more routers into the topology so it now looks like the final topology

```

Borivali(config)#int lo0
Borivali(config-if)#ip ad
*May 29 17:13:47.223: %LINEPROTO-5-UPDOWN: Line protocol o
Borivali(config-if)#ip address 4.4.4.4 255.255.255.255
Borivali(config-if)#ip ospf 2 area 2
Borivali(config-if)#int f0/0
Borivali(config-if)#ip addresss 192.168.1.4 255.255.255.0
^
% Invalid input detected at '^' marker.

Borivali(config-if)#ip address 192.168.1.4 255.255.255.0
Borivali(config-if)#ip ospf 2 area 2
Borivali(config-if)#no shut

```

```

Andheri(config)#int f0/1
Andheri(config-if)#no shut
Andheri(config-if)#ip address
*May 29 17:14:16.199: %LINK-3-UPDOWN: Interface FastEther
*May 29 17:14:17.199: %LINEPROTO-5-UPDOWN: Line protocol
Andheri(config-if)#ip address 192.168.1.1 255.255.255.0

```

```

Andheri(config-if)#ip vrf RED
Andheri(config-vrf)#rd 4:4
Andheri(config-vrf)#route-target both 4:4

```

```

Andheri(config-vrf)#int f0/1
Andheri(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.1.1 removed due to enabling VRF RED

```

```

Andheri#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
  ip vrf forwarding RED
  ip address 192.168.1.1 255.255.255.0
  duplex auto
  speed auto
end

```

If you issue the command sh ip route this shows the routes in the global table and you will notice that you do not see 192.168.1.0/24

```

Andheri#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user subnet route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  1.0.0.0/32 is subnetted, 1 subnets
C    1.1.1.1 is directly connected, Loopback0
  2.0.0.0/32 is subnetted, 1 subnets
O    2.2.2.2 [110/2] via 10.0.0.2, 00:19:39, FastEthernet0/0
  3.0.0.0/32 is subnetted, 1 subnets
O    3.3.3.3 [110/3] via 10.0.0.2, 00:18:35, FastEthernet0/0
  10.0.0.0/24 is subnetted, 2 subnets
C      10.0.0.0 is directly connected, FastEthernet0/0
O      10.0.1.0 [110/2] via 10.0.0.2, 00:18:45, FastEthernet0/0

```

We just need to enable OSPF on this interface and get the loopback address for R4 in the VRF RED

```

Andheri#sh ip route vrf RED
Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user subnet route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/1

```

routing table before proceeding.

```

Andheri(config)#int f0/1
Andheri(config-if)#ip ospf 2 area 2

```

If we now check the routes in the VRF RED routing table you should see 4.4.4.4 in there as well.

```
Andheri#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/2] via 192.168.1.4, 00:00:11, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/1
```

```
Andheri#sh ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 10.0.0.2, 00:28:18, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/3] via 10.0.0.2, 00:27:14, FastEthernet0/0
      10.0.0.0/24 is subnetted, 2 subnets
C        10.0.0.0 is directly connected, FastEthernet0/0
O        10.0.1.0 [110/2] via 10.0.0.2, 00:27:24, FastEthernet0/0
```

```
Andheri#sh ip route vrf RED

Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/2] via 192.168.1.4, 00:07:42, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/1
```

We now need to repeat this process for R3 & R6 Router 6 will peer OSPF using process number 2 to a VRF configured on R3. It will use the local site addressing to 192.168.2.0/24

```
Mahim(config)#int lo0
Mahim(config-if)#
*May 29 17:18:58.903: %LINEPROTO-5-UPDOWN: Line protocol
Mahim(config-if)#ip add 6.6.6.6 255.255.255.255
Mahim(config-if)#ip ospf 2 area 2
Mahim(config-if)#int f0/0
Mahim(config-if)#ip add 192.168.2.6 255.255.255.0
Mahim(config-if)#ip ospf 2 area 2
Mahim(config-if)#no shut
```

```
Churchgate(config)#int f0/1
Churchgate(config-if)#no shut
Churchgate(config-if)#ip add
*May 29 17:23:19.111: %LINK-3-UPDOWN: Interface FastEthernet0/1 is up
*May 29 17:23:20.111: %LINEPROTO-5-UPDOWN: Line protocol
Churchgate(config-if)#ip add 192.168.2.3 255.255.255.0
```

We also need to configure a VRF onto R3 as well.

```
Churchgate(config-if)#ip vrf RED
Churchgate(config-vrf)#rd 4:4
Churchgate(config-vrf)#route-target both 4:4

Churchgate(config-vrf)#int f0/1
Churchgate(config-if)#ip vrf forwarding RED
% Interface FastEthernet0/1 IP address 192.168.2.3 removed due to enabling VRF RED
Churchgate(config-if)#int f0/1
Churchgate(config-if)#ip add 192.168.2.1 255.255.255.0

Churchgate#sh run int f0/1
Building configuration...

Current configuration : 119 bytes
!
interface FastEthernet0/1
  ip vrf forwarding RED
  ip address 192.168.2.1 255.255.255.0
  duplex auto
  speed auto
end
```

Check the router in vrf RED

```

Churchgate#sh ip route vrf RED

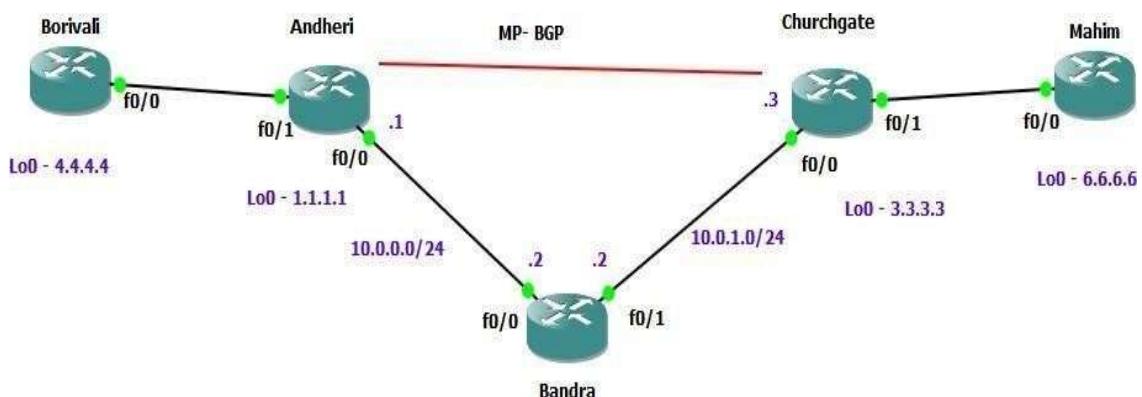
Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user subnet route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      6.0.0.0/32 is subnetted, 1 subnets
O        6.6.6.6 [110/2] via 192.168.2.6, 00:01:10, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/1

```

Ok so we have come a long way now let's review the current situation. We now have this setup



```

Borivali#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user subnet route
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      4.0.0.0/32 is subnetted, 1 subnets
C        4.4.4.4 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, FastEthernet0/0

```

As expected we have the local interface and the loopback address. When we are done we want to see 6.6.6.6 in there so we can ping across the MPLS Check the routes on R1

```
Andheri#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external ty
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS
      ia - IS-IS inter area, * - candidate default, U - per-user
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
O        2.2.2.2 [110/2] via 10.0.0.2, 00:28:18, FastEthernet0/0
      3.0.0.0/32 is subnetted, 1 subnets
O        3.3.3.3 [110/3] via 10.0.0.2, 00:27:14, FastEthernet0/0
      10.0.0.0/24 is subnetted, 2 subnets
C          10.0.0.0 is directly connected, FastEthernet0/0
O          10.0.1.0 [110/2] via 10.0.0.2, 00:27:24, FastEthernet0/0
```

```
Andheri#sh ip route vrf RED
Routing Table: RED
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
      ia - IS-IS inter area, * - candidate default, U - per-user sta
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

      4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/2] via 192.168.1.4, 00:07:42, FastEthernet0/1
C        192.168.1.0/24 is directly connected, FastEthernet0/1
```

```
Andheri(config)#router bgp 1
Andheri(config-router)#address-family ipv4 vrf RED
Andheri(config-router-af)#redistribute ospf 2
Andheri(config-router-af)#exit
Andheri(config-router)#end
```

```
Churchgate(config)#router bgp 1
Churchgate(config-router)#address-family ipv4 vrf RED
Churchgate(config-router-af)#redistribute ospf 2
Churchgate(config-router-af)#end
```

```

Andheri#sh ip bgp vpng4 vrf RED
BGP table version is 9, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
* 4.4.4.4/32        192.168.1.4          2       32768 ?
* 6.6.6.6/32        3.3.3.3             2      100      0 ?
* 192.168.1.0       0.0.0.0             0       32768 ?
* 192.168.2.0       3.3.3.3             0      100      0 ?

```

```

Churchgate#sh ip bgp vpng4 vrf RED
BGP table version is 9, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

      Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 4:4 (default for vrf RED)
*i 4.4.4.4/32       1.1.1.1             2      100      0 ?
* 6.6.6.6/32        192.168.2.6          2       32768 ?
*i 192.168.1.0      1.1.1.1             0      100      0 ?
* 192.168.2.0       0.0.0.0             0       32768 ?

```

Which it is! 6.6.6.6 is now in the BGP table in VRF RED on R3 with a next hop of 192.168.2.6 (R6) and also 4.4.4.4 is in there as well with a next hop of 1.1.1.1 (which is the loopback of R1 – showing that it is going over the MPLS and R2 is not in the picture)

```

Andheri(config)#router ospf 2
Andheri(config-router)#redistribute bgp 1 subnets

Churchgate(config)#router ospf 2
Churchgate(config-router)#redistribute bgp 1 subnets

```

Before we do let's see what the routing table look like on R4

```

Borivali#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

  4.0.0.0/32 is subnetted, 1 subnets
C    4.4.4.4 is directly connected, Loopback0
  6.0.0.0/32 is subnetted, 1 subnets
O IA  6.6.6.6 [110/3] via 192.168.1.1, 00:00:50, FastEthernet0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
O IA 192.168.2.0/24 [110/2] via 192.168.1.1, 00:00:50, FastEthernet0/0

```

Do the same step of on R6

```

Mahim#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static
      o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

        4.0.0.0/32 is subnetted, 1 subnets
O IA    4.4.4.4 [110/3] via 192.168.2.1, 00:00:22, FastEthernet0/0
        6.0.0.0/32 is subnetted, 1 subnets
C     6.6.6.6 is directly connected, Loopback0
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 00:00:22, FastEthernet0/0
C   192.168.2.0/24 is directly connected, FastEthernet0/0

```

Lets check ping command

```

Borivali#ping 6.6.6.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 6.6.6.6, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 112/120/128 ms

```

Which we can – to prove this is going over the mpls and be label switched and not routed, lets do a trace

```

Borivali#trace 6.6.6.6

Type escape sequence to abort.
Tracing the route to 6.6.6.6

 1 192.168.1.1 20 msec 32 msec 24 msec
 2 10.0.0.2 [MPLS: Labels 17/19 Exp 0] 112 msec 136 msec 124 msec
 3 192.168.2.1 [MPLS: Label 19 Exp 0] 72 msec 92 msec 92 msec
 4 192.168.2.6 140 msec 124 msec 124 msec

```

**University Of Mumbai
Institute of Distance & Open Learning**



**PRACTICAL JOURNAL IN
MICROSERVICES ARCHITECTURE**

**SUBMITTED BY
APPLICATION ID:
SEAT NO :**

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY PART-I
SEMESTER II**

**ACADEMIC YEAR
2022-2023**

**INSTITUTE OF DISTANCE AND OPEN LEARNING
IDOL BUILDING, VIDYANAGARI,
SANTACRUZ (EAST), MUMBAI-400 098**

**CONDUCTED AT
PARLE TILAK VIDYALAYA ASSOCIATION'S
MULUND COLLEGE OF COMMERCE(AUTONOMOUS)
MULUND (WEST), MUMBAI – 400080**

**University of Mumbai
Institute of Distance & Open Learning**



Dr. Shankar Dayal Sharama Bhavan, Kalina,
Vidanagari, Santacruz (E), Mumbai-400 098.

Certificate

This is to certify that Mr. Application ID: from Institute of Distance and Open Learning, University of Mumbai , Santacruz East, Mumbai-400098 , PCP Centre: Parle Tilak Vidyalaya Association's Mulund College of Commerce (Autonomous) has successfully completed all the practical of Paper II titled **MICROSERVICES ARCHITECTURE** for M.Sc. (I.T.) Part I Semester – II during the academic year **2022-2023**.

Centre Coordinator

External Examiner

Date:

Seal

INDEX

Sr.No.	Practical	Date	Sign
1	Building APT.NET Core MVC Application.		
2	Building ASP.NET Core REST API.		
3	Working with Docker, Docker Commands, Docker Images, and Containers.		
4	Installing software packages on Docker, Working with Docker Volumes and Networks.		
5	Working with Docker Swarm.		
6	Working with Circle CI for continuous integration.		
7	Creating Microservice with ASP.NET Core.		
8	Working with Kubernetes.		

Practical No.1

Aim: Building APT.NET Core MVC Application.

1)Install .Net Core Sdk (Link:

<https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>)

2)create folder MyMVC folder in C: drive or any other drive

3)open command prompt and perform following operations

Command: to create mvc project

dotnet new mvc --auth none

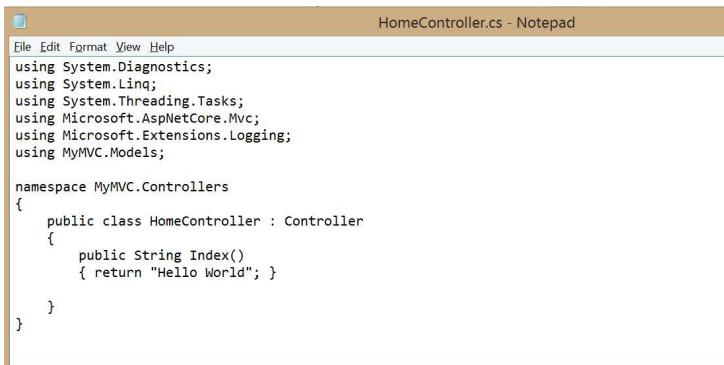
output:



```
C:\Users>cd..  
C:\>cd mymvc  
C:\MyMVC>dotnet new mvc --auth none  
Welcome to .NET 5.0!  
-----  
SDK Version: 5.0.301  
Telemetry  
-----  
The .NET tools collect usage data in order to help us improve your experience. It is collected by Microsoft and shared with the community. You can opt-out of telemetry by setting the DOTNET_CLI_TELEMETRY_OPTOUT environment variable to '1' or 'true' using your favorite shell.  
Read more about .NET CLI Tools telemetry: https://aka.ms/dotnet-cli-telemetry  
-----  
Installed an ASP.NET Core HTTPS development certificate.  
To trust the certificate run 'dotnet dev-certs https --trust' (Windows and macOS only)  
Learn about HTTPS: https://aka.ms/dotnet-https  
-----  
Write your first app: https://aka.ms/dotnet-hello-world  
Find out what's new: https://aka.ms/dotnet-whats-new  
Explore documentation: https://aka.ms/dotnet-docs  
Report issues and find source on GitHub: https://github.com/dotnet/core  
Use 'dotnet -help' to see available commands or visit: https://aka.ms/dotnet-cli  
-----  
Getting ready...  
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.  
This template contains technologies from parties other than Microsoft, see https://aka.ms/aspnetcore/5.0-third-party-notices for details.  
Processing post-creation actions...  
Running 'dotnet restore' on C:\MyMVC\MyMVC.csproj...  
Determining projects to restore...  
Restored C:\MyMVC\MyMVC.csproj (in 215 ms).  
Restore succeeded.  
C:\MyMVC>
```

4) Go to controllers folder and modify HomeController.cs file to match following:

Name	Date modified	Type	Size
Controllers	08-07-2021 09:46	File folder	
Models	08-07-2021 09:46	File folder	
obj	08-07-2021 09:46	File folder	
Properties	08-07-2021 09:46	File folder	
Views	08-07-2021 09:46	File folder	
wwwroot	08-07-2021 09:46	File folder	
appsettings.Development.json	08-07-2021 09:46	JSON File	1 KB
appsettings.json	08-07-2021 09:46	JSON File	1 KB
MyMVC	08-07-2021 09:46	CSPROJ File	1 KB
Program.cs	08-07-2021 09:46	C# Source File	1 KB
Startup.cs	08-07-2021 09:46	C# Source File	2 KB

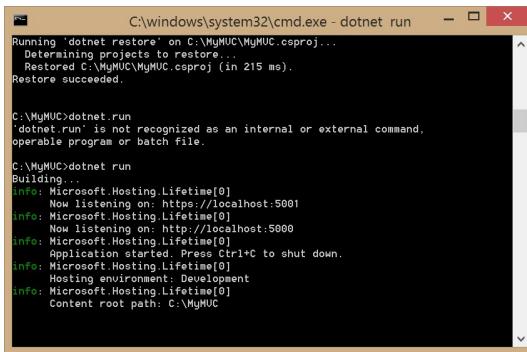


```
HomeController.cs - Notepad
File Edit Format View Help
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;

namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public String Index()
        { return "Hello World"; }

    }
}
```

Run the Project

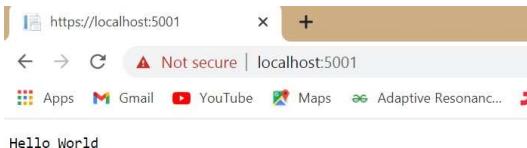


```
C:\windows\system32\cmd.exe - dotnet run
Running 'dotnet restore' on C:\MyMUC\MyMUC.csproj...
Determining projects to restore...
Restored C:\MyMUC\MyMUC.csproj (in 215 ms).
Restore succeeded.

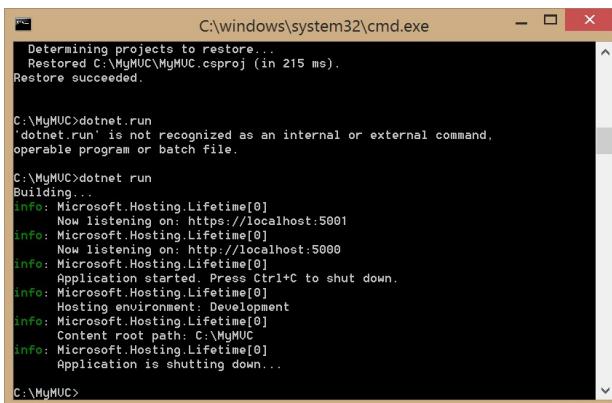
C:\MyMUC>dotnet run
'dotnet.run' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMUC>dotnet run
Building...
Info: Microsoft.Hosting.Lifetime[0]
  Now listening on: https://localhost:5001
Info: Microsoft.Hosting.Lifetime[0]
  Now listening on: http://localhost:5000
Info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Development
Info: Microsoft.Hosting.Lifetime[0]
  Content root path: C:\MyMUC
```

Now open browser and type URL: localhost:5000



Now go back to command prompt and stop running project using CTRL+C

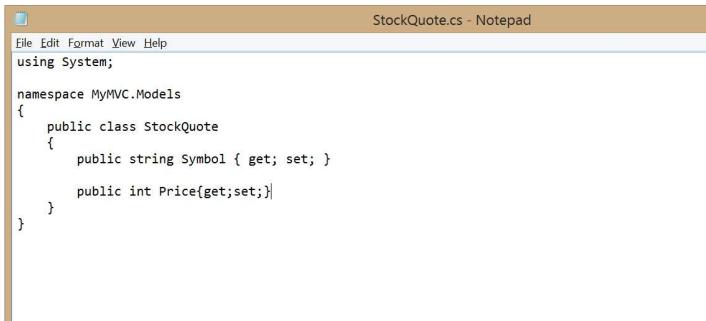


```
C:\windows\system32\cmd.exe
Determining projects to restore...
Restored C:\MyMUC\MyMUC.csproj (in 215 ms).
Restore succeeded.

C:\MyMUC>dotnet run
'dotnet.run' is not recognized as an internal or external command,
operable program or batch file.

C:\MyMUC>dotnet run
Building...
Info: Microsoft.Hosting.Lifetime[0]
  Now listening on: https://localhost:5001
Info: Microsoft.Hosting.Lifetime[0]
  Now listening on: http://localhost:5000
Info: Microsoft.Hosting.Lifetime[0]
  Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
  Hosting environment: Development
Info: Microsoft.Hosting.Lifetime[0]
  Content root path: C:\MyMUC
Info: Microsoft.Hosting.Lifetime[0]
  Application is shutting down...
```

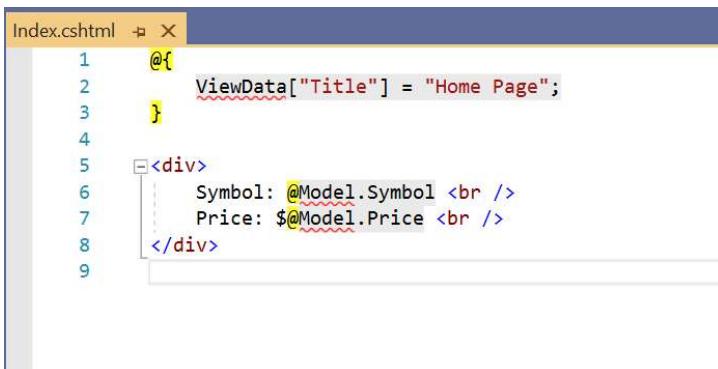
Go to models folder and add new file StockQuote.cs to it with following content



```
StockQuote.cs - Notepad
File Edit Format View Help
using System;

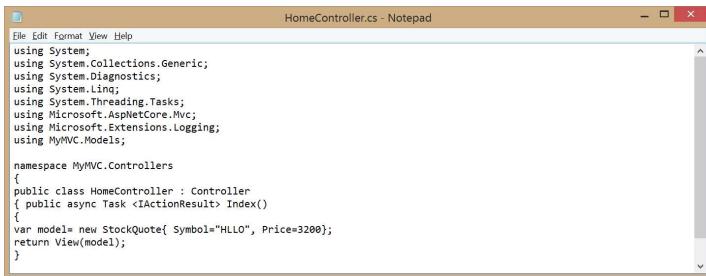
namespace MyMVC.Models
{
    public class StockQuote
    {
        public string Symbol { get; set; }
        public int Price { get; set; }
    }
}
```

Now Add View to folder then home folder in it and modify index.cshtml file to match following



```
Index.cshtml - p x
1  @{
2      ViewData["Title"] = "Home Page";
3  }
4
5  <div>
6      Symbol: @Model.Symbol <br />
7      Price: $@Model.Price <br />
8  </div>
9
```

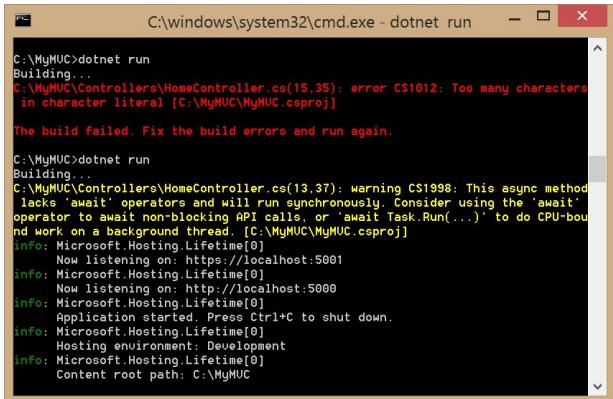
Now modify HomeController.cs file to match following:



```
HomeController.cs - Notepad
File Edit Format View Help
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Logging;
using MyMVC.Models;

namespace MyMVC.Controllers
{
    public class HomeController : Controller
    {
        public async Task<IActionResult> Index()
        {
            var model = new StockQuote { Symbol = "HLL0", Price = 3200 };
            return View(model);
        }
    }
}
```

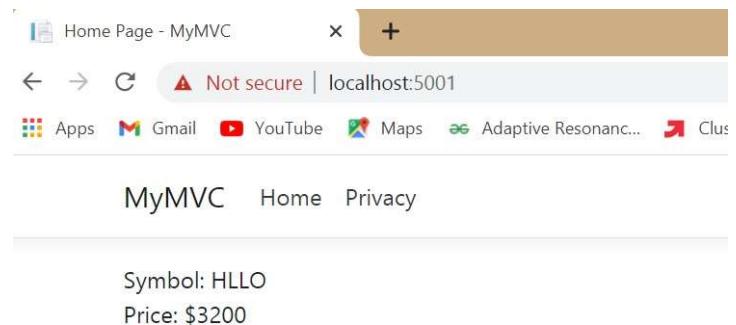
Now run the project using



```
C:\MyMUC>dotnet run
Building ...
C:\MyMUC\Controllers\HomeController.cs(15,35): error CS1012: Too many characters in character literal [C:\MyMUC\MyMUC.csproj]
The build failed. Fix the build errors and run again.

C:\MyMUC>dotnet run
Building ...
C:\MyMUC\Controllers\HomeController.cs(13,37): warning CS1998: This async method lacks 'await' operators and will run synchronously. Consider using the 'await' operator to await non-blocking API calls, or 'await Task.Run(...)' to do CPU-bound work on a background thread. [C:\MyMUC\MyMUC.csproj]
Info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
Info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
Info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
Info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\MyMUC
```

Now go back to browser and refresh to get modified view response



Practical NO. 2

Aim:Building ASP.NET Core REST API.

Software requirement:

1. Download and install

To start building .NET apps you just need to download and install the .NET SDK (Software Development Kit version 3.0 above).

Link:

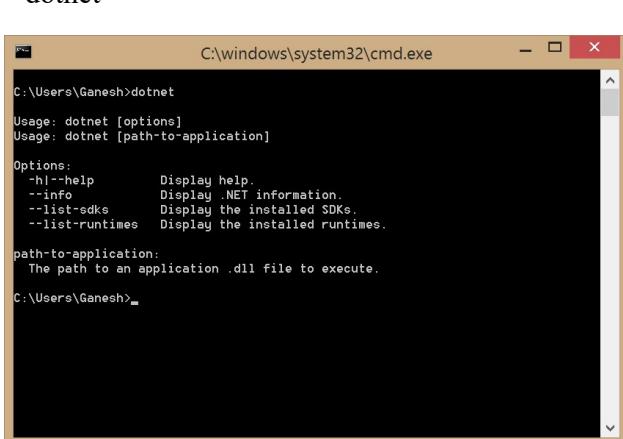
<https://dotnet.microsoft.com/learn/dotnet/hello-world-tutorial/install>

2. Check everything installed correctly

Once you've installed, open a new command prompt and run the following command:

Command prompt

> dotnet



```
C:\Windows\system32\cmd.exe
C:\Users\Ganesh>dotnet
Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help      Display help.
  --info         Display .NET information.
  --list-sdks    Display the installed SDKs.
  --list-runtimes Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\Ganesh>
```

Create your web API

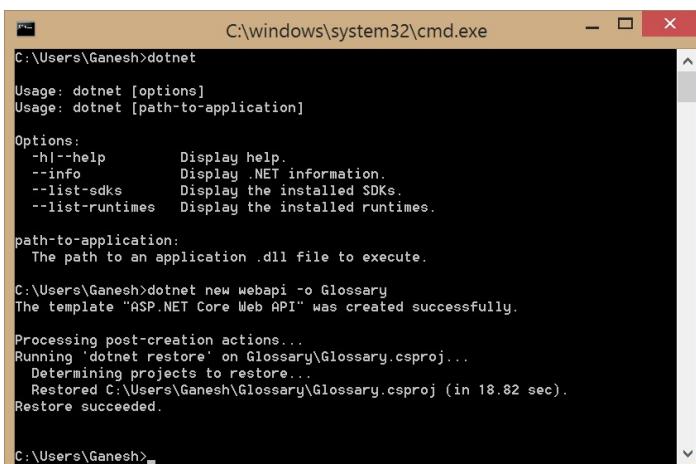
1. Open two command prompts

Command prompt 1:

Command:

dotnet new webapi -o Glossary

Output:



```
C:\Windows\system32\cmd.exe
C:\Users\Ganesh>dotnet
Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help      Display help.
  --info         Display .NET information.
  --list-sdks    Display the installed SDKs.
  --list-runtimes Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.

C:\Users\Ganesh>dotnet new webapi -o Glossary
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Running 'dotnet restore' on Glossary\Glossary.csproj...
  Determining projects to restore...
  Restored C:\Users\Ganesh\Glossary\Glossary.csproj (in 18.82 sec).
Restore succeeded.

C:\Users\Ganesh>
```

Command:

```
cd Glossary  
dotnet run
```

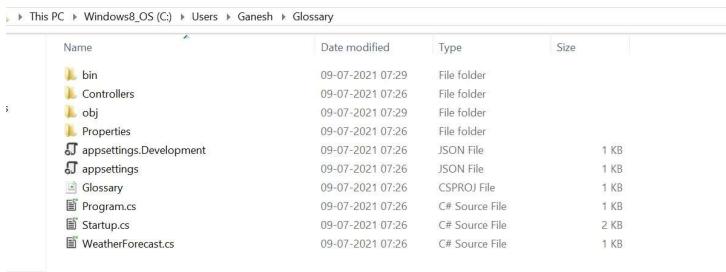
Output:

```
C:\windows\system32\cmd.exe - dotnet run

Processing post-create actions...
Running 'dotnet restore' on Glossary\Glossary.csproj...
  Determining projects to restore...
  Restored C:\Users\Ganesh\Glossary\Glossary.csproj (in 18.82 sec).
Restore succeeded.

C:\Users\Ganesh>cd Glossary

C:\Users\Ganesh\Glossary>dotnet run
Building...
[...]
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
      Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Ganesh\Glossary
```



Command Prompt 2: (try running readymade weatherforecast class for testing)

Command:

```
curl --insecure https://localhost:5001/weatherforecast
```

output:



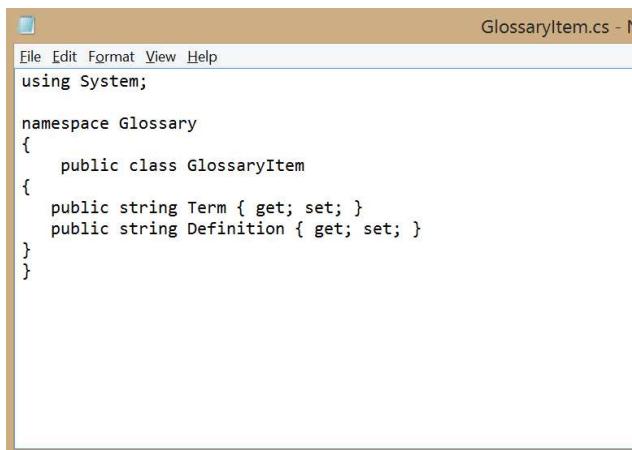
Now change the content:

To get started, remove the WeatherForecast.cs file from the root of the project and the WeatherForecastController.cs file from the Controllers folder.

Add Following two files

1) D:\Glossary\GlossaryItem.cs (type it in notepad and save as all files)

```
//GlossaryItem.cs
namespace Glossary
{
    public class GlossaryItem
    {
        public string Term { get; set; }
        public string Definition { get; set; }
    }
}
```



D:\Glossary\Controllers\ GlossaryController.cs (type it in notepad and save as all files)

```
//Controllers/GlossaryController.cs
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.IO;
namespace Glossary.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class GlossaryController: ControllerBase
    {
        private static List<GlossaryItem> Glossary = new List<GlossaryItem> {
            new GlossaryItem
            {
                Term= "HTML",
                Definition = "Hypertext Markup Language"
            },
            new GlossaryItem
        };
    }
}
```

```

{
Term= "MVC",
Definition = "Model View Controller"
},
new GlossaryItem
{
Term= "OpenID",
Definition = "An open standard for authentication"
}
};

[HttpGet]
public ActionResult<List<GlossaryItem>> Get()
{
    return Ok(Glossary);
}

[HttpGet]
[Route("{term}")]
public ActionResult<GlossaryItem> Get(string term)
{
    var glossaryItem = Glossary.Find(item =>
        item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));
    if (glossaryItem == null)
    {
        return NotFound();
    }
    else
    {
        return Ok(glossaryItem);
    }
}

[HttpPost]
public ActionResult Post(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
    if (existingGlossaryItem != null)
    {
        return Conflict("Cannot create the term because it already exists.");
    }
    else
    {
        Glossary.Add(glossaryItem);
        var resourceUrl = Path.Combine(Request.Path.ToString(),
            Uri.EscapeUriString(glossaryItem.Term));
        return Created(resourceUrl, glossaryItem);
    }
}

[HttpPut]
public ActionResult Put(GlossaryItem glossaryItem)
{
    var existingGlossaryItem = Glossary.Find(item =>
        item.Term.Equals(glossaryItem.Term, StringComparison.InvariantCultureIgnoreCase));
    if (existingGlossaryItem == null)

```

```
{  
    return BadRequest("Cannot update a nont existing term.");  
} else  
{  
    existingGlossaryItem.Definition = glossaryItem.Definition;  
    return Ok();  
}  
}  
}  
[HttpDelete]  
[Route("{term}")]  
public ActionResult Delete(string term)  
{  
    var glossaryItem = Glossary.Find(item =>  
        item.Term.Equals(term, StringComparison.InvariantCultureIgnoreCase));  
    if (glossaryItem == null)  
    {  
        return NotFound();  
    }  
    else  
    {  
        Glossary.Remove(glossaryItem);  
        return NoContent();  
    }  
}  
}  
}  
}
```

```
File Edit Format View Help
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.IO;
namespace Glossary.Controllers
{
[ApiController]
[Route("api/[controller]")]
public class GlossaryController: ControllerBase
{
private static List<GlossaryItem> Glossary = new List<GlossaryItem> {
new GlossaryItem
{
Term= "HTML",
Definition = "Hypertext Markup Language"
},
new GlossaryItem
```

Output:

Name	Date modified	Type	Size
bin	09-07-2021 07:29	File folder	
Controllers	09-07-2021 09:20	File folder	
obj	09-07-2021 07:29	File folder	
Properties	09-07-2021 07:26	File folder	
appsettings.Development	09-07-2021 07:26	JSON File	1 KB
appsettings	09-07-2021 07:26	JSON File	1 KB
Glossary	09-07-2021 07:26	CSPROJ File	1 KB
GlossaryItem.cs	09-07-2021 09:19	C# Source File	1 KB
Program.cs	09-07-2021 07:26	C# Source File	1 KB
Startup.cs	09-07-2021 07:26	C# Source File	2 KB

Name	Date modified	Type
GlossaryController.cs	09-07-2021 09:22	C# Source File

Now stop running previous dotnet run on command prompt 1 using Ctrl+C. and Run it again for new code.

On Command prompt1:

Command:

dotnet run

output:

```
C:\windows\system32\cmd.exe - dotnet run
Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\Ganesh\Glossary
info: Microsoft.Hosting.Lifetime[0]
Application is shutting down...

C:\Users\Ganesh\Glossary>dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: C:\Users\Ganesh\Glossary
```

On Command prompt2:

1) Getting a list of items:

Command:

```
curl --insecure https://localhost:5001/api/glossary
```

Getting a single item

Command:

```
curl --insecure https://localhost:5001/api/glossary/MVC
```

Output:

Creating an item

Command:

```
curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
```

```
C:\Windows\system32\cmd.exe
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}]
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary/MUC
{"term": "MUC", "definition": "Model View Controller"}
C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "\nAn authentication process.\n"}' -H "Content-type: application/json"
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "\nAn authentication process.\n"}' -H "Content-type: application/json"
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "\nAn authentication process.\n"}' -H "Content-type: application/json" https://localhost:5001/api/glossary
{"term": "MFA", "definition": "An authentication process."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "Hypertext Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>
```

Update Item

Command:

```
curl --insecure -X PUT -d "{\"term\": \"MVC\", \"definition\":\"Modified record of Model View\"}
```

```
Controller."}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
```

Output:

```
C:\Windows\system32\cmd.exe

curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "\n    An authentication process.\n"}' -H "Content-Type: application/json"
curl: no URL specified!
curl: try 'curl --help' or 'curl --manual' for more information

C:\Users\Ganesh>curl --insecure -X POST -d '{"term": "MFA", "definition": "\n    An authentication process.\n"}' -H "Content-Type: application/json" https://localhost:5001/api/glossary
{"term": "MFA", "definition": "An authentication process."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "HyperText Markup Language"}, {"term": "MUC", "definition": "Model View Controller"}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
C:\Users\Ganesh>curl --insecure -X PUT -d '{"term": "MUC", "definition": "\n    Modified record of Model View Controller.\n"}' -H "Content-Type: application/json" https://localhost:5001/api/glossary
{"term": "MUC", "definition": "Modified record of Model View Controller."}

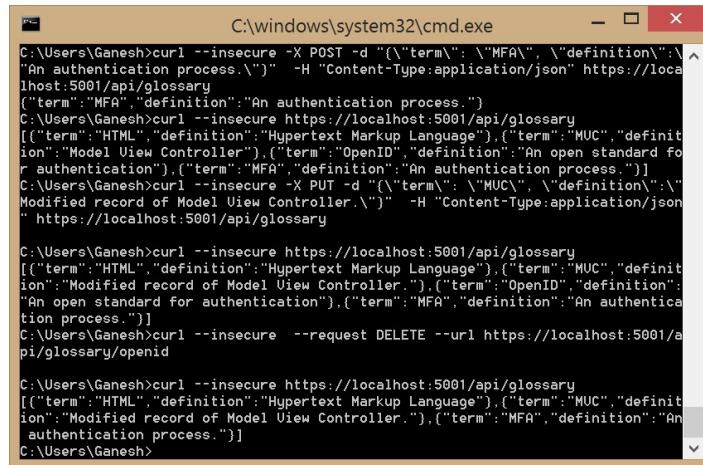
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term": "HTML", "definition": "HyperText Markup Language"}, {"term": "MUC", "definition": "Modified record of Model View Controller."}, {"term": "OpenID", "definition": "An open standard for authentication"}, {"term": "MFA", "definition": "An authentication process."}]
```

Delete Item

Command:

```
curl --insecure --request DELETE --url https://localhost:5001/api/glossary/openid
```

Output:



```
C:\Users\Ganesh>curl --insecure -X POST -d "{\"term\": \"MFA\", \"definition\": \"An authentication process.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
[{"term":"MFA","definition":"An authentication process."}
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"}, {"term":"MUC","definition":"Model View Controller"}, {"term":"OpenID","definition":"An open standard for authentication"}, {"term":"MFA","definition":"An authentication process."}]
C:\Users\Ganesh>curl --insecure -X PUT -d "{\"term\": \"MUC\", \"definition\": \"Modified record of Model View Controller.\"}" -H "Content-Type:application/json" https://localhost:5001/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"}, {"term":"MUC","definition":"Modified record of Model View Controller."}, {"term":"OpenID","definition":"An open standard for authentication"}, {"term":"MFA","definition":"An authentication process."}]
C:\Users\Ganesh>curl --insecure --request DELETE --url https://localhost:5001/api/glossary/openid
C:\Users\Ganesh>curl --insecure https://localhost:5001/api/glossary
[{"term":"HTML","definition":"Hypertext Markup Language"}, {"term":"MUC","definition":"Modified record of Model View Controller."}, {"term":"OpenID","definition":"An open standard for authentication"}, {"term":"MFA","definition":"An authentication process."}]
C:\Users\Ganesh>
```

Practical No. 3

Aim: Working with Docker, Docker Commands, Docker Images and Containers

After install ubuntu in vmware. Install docker

Command: sudo apt-get install docker.io



Install using the repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

Set up the repository

Update the apt package index and install packages to allow apt to use a repository over HTTPS:

1. \$ sudo apt-get update
2. \$ sudo apt-get install \apt-transport-https \
ca-certificates \
curl \
gnupg \
lsb-release

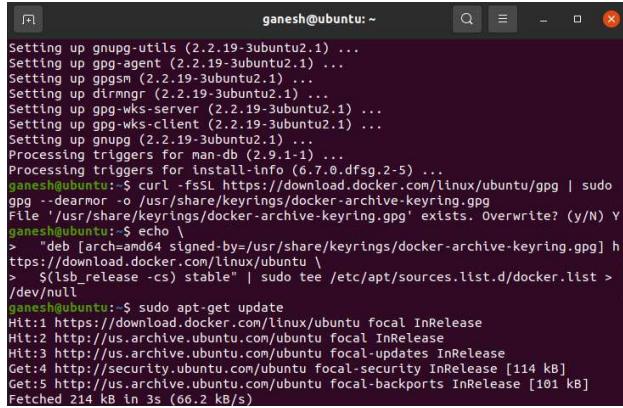
```
ganesh@ubuntu: ~
[GNOME Terminal]
Experimental: true
got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://<2>/var<3>/fdocker.sock/v1.24/version: dial unix /var/run/docker.sock: connect: permission denied
janesh@ubuntu: ~$ sudo apt-get update
[GNOME Terminal]
Get:1 http://archive.ubuntu.com/ubuntu focal InRelease [114 kB]
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [101 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease [101 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 214 kB in 4s (48.8 kB/s)
Reading package lists... Done
janesh@ubuntu: ~$ sudo apt-get install \
[GNOME Terminal]
    apt-transport-https \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20210110-20.04.1)
```

1. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

Use the following command to set up the **stable** repository

```
$ echo \
  "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```



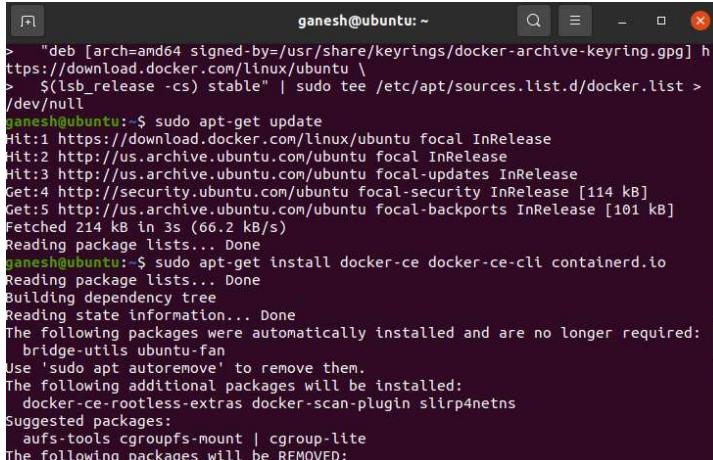
The terminal window shows the following sequence of commands:

```
Setting up gnupg-utils (2.2.19-3ubuntu2.1) ...
Setting up gpg-agent (2.2.19-3ubuntu2.1) ...
Setting up gpgsm (2.2.19-3ubuntu2.1) ...
Setting up dirmngr (2.2.19-3ubuntu2.1) ...
Setting up gpg-wks-server (2.2.19-3ubuntu2.1) ...
Setting up gpg-wks-client (2.2.19-3ubuntu2.1) ...
Setting up gnupg (2.2.19-3ubuntu2.1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for install-info (6.7.0.dfsg.2-5) ...
ganesh@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo
gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
File '/usr/share/keyrings/docker-archive-keyring.gpg' exists. Overwrite? (y/N) Y
ganesh@ubuntu:~$ echo \
> "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null
ganesh@ubuntu:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Fetched 214 kB in 3s (66.2 kB/s)
```

Install Docker Engine

Update the apt package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```



The terminal window shows the following sequence of commands:

```
> "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] h
https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null
ganesh@ubuntu:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Fetched 214 kB in 3s (66.2 kB/s)
Reading package lists... Done
ganesh@ubuntu:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  bridge-utils ubuntu-fan
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following packages will be REMOVED:
```

To install a *specific version* of Docker Engine, list the available versions in the repo, then select and install:

```
apt-cache madison docker-ce
```

```
ganesh@ubuntu:~$ apt-cache madison docker-ce
  installed docker-ce package post-installation script subprocess returned error
  exit status 1
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.4) ...
Errors were encountered while processing:
  docker-ce
E: Sub-process /usr/bin/dpkg returned an error code (1)
ganesh@ubuntu:~$ apt-cache madison docker-ce
  docker-ce | 5:20.10.7-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.6-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.5-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.4-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.3-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.2-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.1-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
  docker-ce | 5:20.10.0-3~0ubuntu-focal | https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
```

Docker Commands:

Docker –version

Docker version

```
ganesh@ubuntu:~$ docker --version
Docker version 20.10.2, build 20.10.2-0ubuntu1~20.04.2
ganesh@ubuntu:~$ docker version
Client:
  Version:          20.10.2
  API version:      1.41
  Go version:       go1.13.8
  Git commit:       20.10.2-0ubuntu1~20.04.2
  Built:            Tue Mar 30 21:24:57 2021
  OS/Arch:          linux/amd64
  Context:          default
  Experimental:    true
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://<2>Var%2Frun%2Fdocker.sock/v1.24/version: dial unix /var/run/docker.sock: connect: permission denied
ganesh@ubuntu:~$ docker images
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get http://<2>Var%2Frun%2Fdocker.sock/v1.24/images/json
```

Docker pull httpd

Pull an image or a repository from a registry

```
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

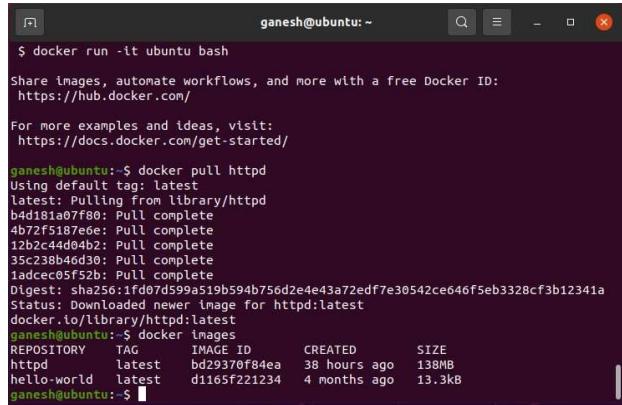
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

ganesh@ubuntu:~$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
b4d181a07f80: Pull complete
4b72f5187e6e: Pull complete
12b2c44d04b2: Pull complete
35c238b46d30: Pull complete
1addec05f52b: Pull complete
Digest: sha256:ffd07d599a519b594b75d2e4e43a72edf7e30542ce646f5eb3328cf3b12341a
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
ganesh@ubuntu:~$
```

Docker images

It lists all the images



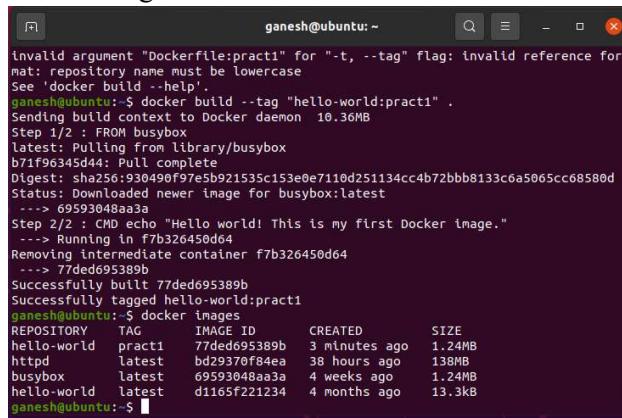
```
ganesh@ubuntu: ~
$ docker run -it ubuntu bash
Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
For more examples and ideas, visit:
https://docs.docker.com/get-started/
ganesh@ubuntu:~$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
b4d181a07f80: Pull complete
4b72f51b7e6: Pull complete
12b2c24d04b2: Pull complete
35c238b46d30: Pull complete
iadecce05f52b: Pull complete
Digest: sha256:1fd07d599a19b594b756d2e4e43a72edf7e30542ce646f5eb3328cf3b12341a
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
ganesh@ubuntu:~$ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
httpd           latest       bd29370f84ea   38 hours ago   138MB
hello-world     latest       d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$
```

```
#nano Dockerfile
FROM busybox
CMD echo "Hello world! This is my first Docker image."
```

//above two line we have to add into dockerfile
to save press **ctrl+o**(to write) then enter then **ctrl+x** (to exit)

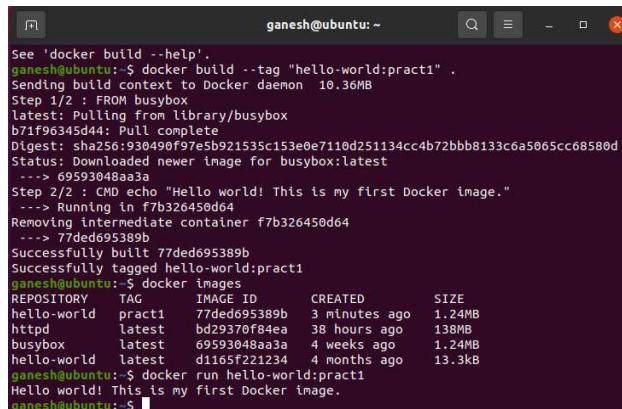
docker build --tag "hello-world:pract1" .

docker images



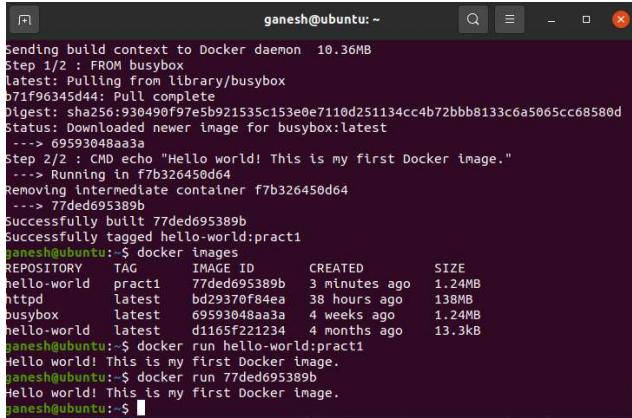
```
ganesh@ubuntu: ~
invalid argument "Dockerfile:pract1" for "-t, --tag" flag: invalid reference for
mat: repository name must be lowercase
See 'docker build --help'.
ganesh@ubuntu: $ docker build --tag "hello-world:pract1" .
Sending build context to Docker daemon 10.36MB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
b71f96345d44: Pull complete
Digest: sha256:930490f97e5b921535c153e0e7110d25i134cc4b72bbb8133c6a5065cc68580d
Status: Downloaded newer image for busybox:latest
--> 69593048aa3a
Step 2/2 : CMD echo "Hello world! This is my first Docker image."
--> Running in f7b326450d64
Removing intermediate container f7b326450d64
--> 77ded695389b
Successfully built 77ded695389b
Successfully tagged hello-world:pract1
ganesh@ubuntu: $ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
hello-world     pract1       77ded695389b   3 minutes ago  1.24MB
httpd           latest       bd29370f84ea   38 hours ago   138MB
busybox          latest       69593048aa3a   4 weeks ago   1.24MB
hello-world     latest       d1165f221234   4 months ago   13.3kB
ganesh@ubuntu:~$
```

docker run hello-world:pract1



```
ganesh@ubuntu: ~
See 'docker build --help'.
ganesh@ubuntu: $ docker run hello-world:pract1
Sending build context to Docker daemon 10.36MB
Step 1/2 : FROM busybox
latest: Pulling from library/busybox
b71f96345d44: Pull complete
Digest: sha256:930490f97e5b921535c153e0e7110d25i134cc4b72bbb8133c6a5065cc68580d
Status: Downloaded newer image for busybox:latest
--> 69593048aa3a
Step 2/2 : CMD echo "Hello world! This is my first Docker image."
--> Running in f7b326450d64
Removing intermediate container f7b326450d64
--> 77ded695389b
Successfully built 77ded695389b
Successfully tagged hello-world:pract1
ganesh@ubuntu: $ docker images
REPOSITORY      TAG          IMAGE ID      CREATED        SIZE
hello-world     pract1       77ded695389b   3 minutes ago  1.24MB
httpd           latest       bd29370f84ea   38 hours ago   138MB
busybox          latest       69593048aa3a   4 weeks ago   1.24MB
hello-world     latest       d1165f221234   4 months ago   13.3kB
ganesh@ubuntu: $ docker run hello-world:pract1
Hello world! This is my first Docker image.
ganesh@ubuntu:~$
```

```
docker run 77ded695389b
```



```
ganesh@ubuntu:~$ docker build -t hello-world .
Sending build context to Docker daemon 10.36MB
Step 1/2 : FROM busybox
Step 2/2 : CMD echo "Hello world! This is my first Docker image."
--> 77ded695389b
Successfully built 77ded695389b
Successfully tagged hello-world:pract1
ganesh@ubuntu:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world pract1 77ded695389b 3 minutes ago 1.24MB
httpd latest bd29370f84ea 38 hours ago 138MB
busybox latest 69593048aa3a 4 weeks ago 1.24MB
hello-world latest d1165f221234 4 months ago 13.3kB
ganesh@ubuntu:~$ docker run hello-world:pract1
Hello world! This is my first Docker image.
ganesh@ubuntu:~$ docker run 77ded695389b
Hello world! This is my first Docker image.
ganesh@ubuntu:~$
```

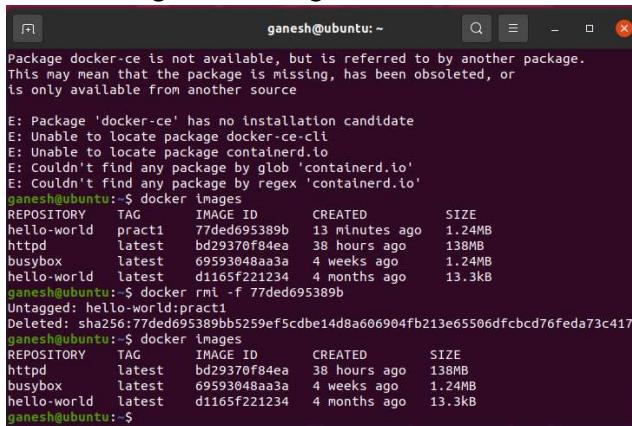
Docker rmi

Remove one or more images

```
docker rmi -f images-id
```

```
docker rmi -f 77ded695389b
```

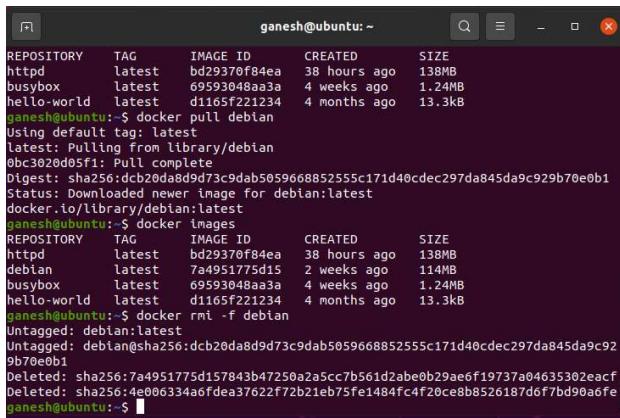
After running docker images we can see that 77ded695389b is deleted.



```
ganesh@ubuntu:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world pract1 77ded695389b 13 minutes ago 1.24MB
httpd latest bd29370f84ea 38 hours ago 138MB
busybox latest 69593048aa3a 4 weeks ago 1.24MB
hello-world latest d1165f221234 4 months ago 13.3kB
ganesh@ubuntu:~$ docker rmi -f 77ded695389b
Untagged: sha256:77ded695389b5259ef5cdbe14d8a606904fb213e65506dfbcd76feda73c417
ganesh@ubuntu:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest bd29370f84ea 38 hours ago 138MB
busybox latest 69593048aa3a 4 weeks ago 1.24MB
hello-world latest d1165f221234 4 months ago 13.3kB
ganesh@ubuntu:~$
```

```
docker rmi -f Respository-name
```

```
docker rmi -f Debian
```

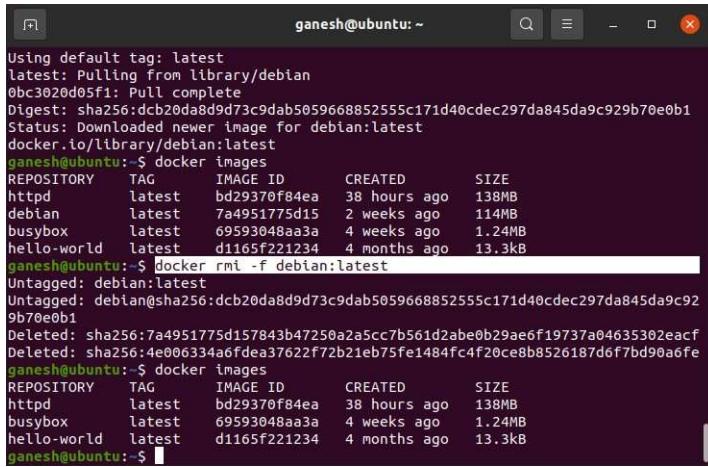


```
ganesh@ubuntu:~$ docker pull debian
Using default tag: latest
latest: Pulling from library/debian
0bc3020d05f1: Pull complete
Digest: sha256:dcbb0da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1
Status: Downloaded newer image for debian:latest
docker.io/library/debian:latest
ganesh@ubuntu:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
httpd latest bd29370f84ea 38 hours ago 138MB
debian latest 7a4951775d15 2 weeks ago 114MB
busybox latest 69593048aa3a 4 weeks ago 1.24MB
hello-world latest d1165f221234 4 months ago 13.3kB
ganesh@ubuntu:~$ docker rmi -f debian
Untagged: debian:latest
Untagged: debiangsha256:dcbb0da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1
Deleted: sha256:7a4951775d157843b47250a2a5cc7b561d2abe0b29ae6f19737a04635302eacf
Deleted: sha256:4e006334a6fdea37622f72b21eb75fe1484fc4f20ce8b8526187d6f7bd90a6fe
ganesh@ubuntu:~$
```

```
docker rmi -f Respository-name:tag
```

```
docker rmi -f debian:latest
```

After this debain image will be deleted



The screenshot shows a terminal window titled "ganesh@ubuntu: ~". The session starts with pulling the "debian" image from the Docker hub:

```
Using default tag: latest
latest: Pulling from library/debian
0bc3020d05f1: Pull complete
Digest: sha256:dcb20da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1
Status: Downloaded newer image for debian:latest
docker.io/library/debian:latest
```

Then, the user runs the command to delete the "debian" image:

```
ganesh@ubuntu:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
httpd          latest    bd29370f84ea  38 hours ago  138MB
debian          latest    7a4951775d19  2 weeks ago   114MB
busybox         latest    69593048aa3a  4 weeks ago   1.24MB
hello-world     latest    d1165f221234  4 months ago  13.3kB
ganesh@ubuntu:~$ docker rmi -f debian:latest
```

Output shows the image was untagged and then deleted:

```
Untagged: debian:latest
Untagged: debian@sha256:dcb20da8d9d73c9dab5059668852555c171d40cdec297da845da9c929b70e0b1
Deleted: sha256:7a4951775d157843b47250a2a5cc7b561d2abe0b29ae6f19737a04635302eacf
Deleted: sha256:4e006334a6fdea37622f72b21eb75fe1484fc4f20ce8bb526187d6f7bd90a6fe
```

Finally, the user runs the "docker images" command again to verify the image is gone:

```
ganesh@ubuntu:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
httpd          latest    bd29370f84ea  38 hours ago  138MB
busybox         latest    69593048aa3a  4 weeks ago   1.24MB
hello-world     latest    d1165f221234  4 months ago  13.3kB
ganesh@ubuntu:~$
```

Practical No. 4

Aim: Installing software packages on Docker, Working with Docker Volumes and Networks.

Volumes are the preferred mechanism for persisting data generated by and used by Docker containers. While bind mounts are dependent on the directory structure and OS of the host machine, volumes are completely managed by Docker.

List volumes created

Command: docker volume ls

To create volume.

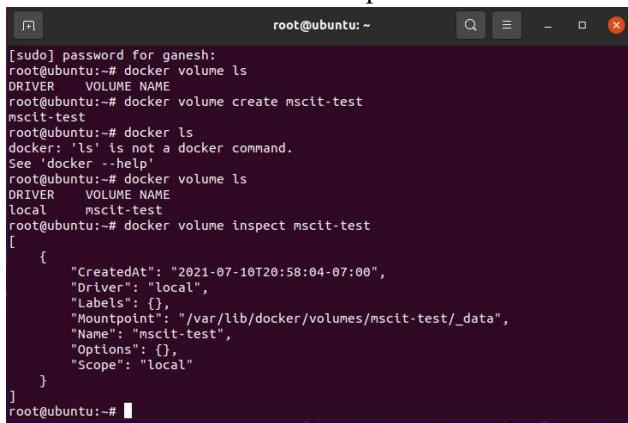
Command: docker volume create mscit-test



```
ganesh@ubuntu:~$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~$ docker volume ls
DRIVER      VOLUME NAME
ganesh@ubuntu:~$ sudo -i
[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
root@ubuntu:~# docker volume create mscit-test
mscit-test
root@ubuntu:~# docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local      mscit-test
root@ubuntu:~#
```

Return low-level information on Docker objects

Command: docker volume inspect mscit-test



```
[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
root@ubuntu:~# docker volume create mscit-test
mscit-test
root@ubuntu:~# docker ls
docker: 'ls' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local      mscit-test
root@ubuntu:~# docker volume inspect mscit-test
[
  {
    "CreatedAt": "2021-07-10T20:58:04-07:00",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/mscit-test/_data",
    "Name": "mscit-test",
    "Options": {},
    "Scope": "local"
  }
]
root@ubuntu:~#
```

Create a directory

mkdir mscit-volume

Now, change directory to mscit-volume

cd mscit-volume/

```
root@ubuntu:~# docker ls
docker: 'ls' is not a docker command.
See 'docker --help'.
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local        mscit-test
root@ubuntu:~# docker volume inspect mscit-test
[{"Name": "mscit-test", "Driver": "local", "Scope": "local", "Labels": {}, "Mountpoint": "/var/lib/docker/volumes/mscit-test/_data", "Options": {}, "CreatedAt": "2021-07-10T20:58:04-07:00", "Labels": {}}
root@ubuntu:~# ls
snap
root@ubuntu:~# mkdir mscit-volume
root@ubuntu:~# cd mscit-volume/
root@ubuntu:~/mscit-volume# nano Dockerfile
```

Create a file Nano Dockerfile

```
root@ubuntu:~/mscit-volume Dockerfile Modified
FROM ubuntu:trusty
CMD ["/bin/bash"]
WORKDIR /test_container
VOLUME ["/test_container"]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^L Replace ^U Paste Text ^T To Spell ^A Go To Line
```

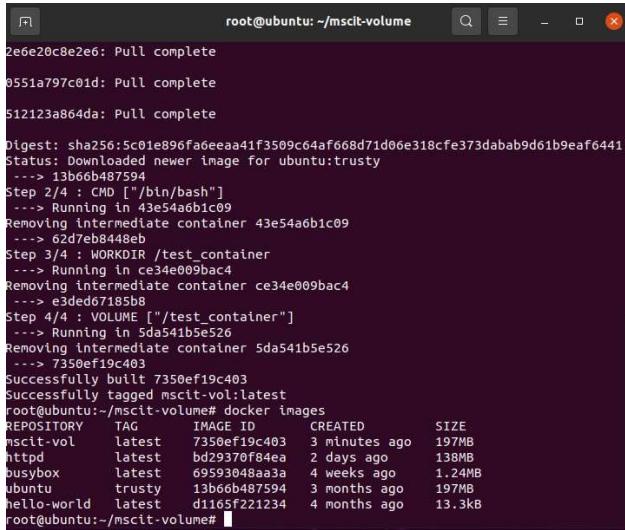
To create an image file
docker build --tag=mscit-vol:latest .

```
root@ubuntu:~# mkdir mscit-volume
root@ubuntu:~# cd mscit-volume/
root@ubuntu:~/mscit-volume# nano Dockerfile
root@ubuntu:~/mscit-volume# docker build --tag=mscit-vol:latest .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM ubuntu:trusty
trusty: Pulling from library/ubuntu
2e6620c8e2e6: Pull complete
0551a797c01d: Pull complete
512123a864da: Pull complete

Digest: sha256:5c01e896fa6eaa41f3509c64af668d71d06e318cf373dabab9d61b9eaf6441
Status: Downloaded newer image for ubuntu:trusty
--> 13b66b487594
Step 2/4 : CMD ["/bin/bash"]
--> Running in 43e54a6b1c09
Removing intermediate container 43e54a6b1c09
--> 62d7eb8448eb
Step 3/4 : WORKDIR /test_container
--> Running in ce34e009bac4
Removing intermediate container ce34e009bac4
--> e3ded67185b8
Step 4/4 : VOLUME ["/test_container"]
--> Running in 5da541b5e526
Removing intermediate container 5da541b5e526
--> 7350ef19c403
Successfully built 7350ef19c403
Successfully tagged mscit-vol:latest
root@ubuntu:~/mscit-volume#
```

Check the image create

Command: docker images

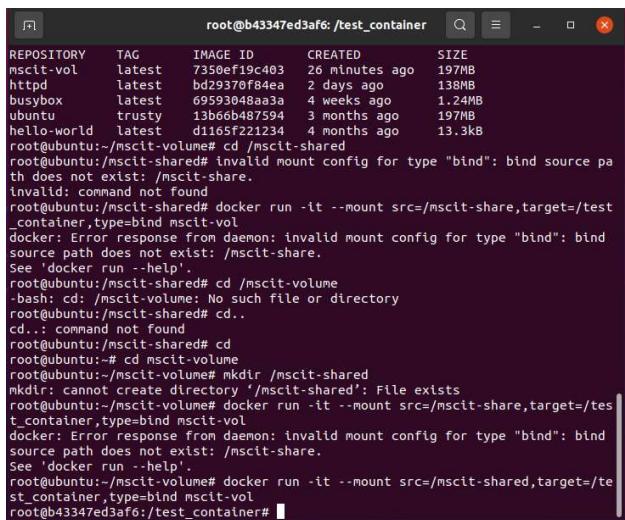


```
root@ubuntu:~/mscit-volume
2e6e20c8e2e6: Pull complete
9551a797c01d: Pull complete
512123a864da: Pull complete
Digest: sha256:5c01e896fa6eeaa41f3509c64af668d71d06e318cf373dabab9d61b9eaf6441
Status: Downloaded newer image for ubuntu:trusty
--> 13b66b487594
Step 2/4 : CMD ["/bin/bash"]
--> Running in 43e54a6b1c09
Removing intermediate container 43e54a6b1c09
--> 62d7eb8448e6
Step 3/4 : WORKDIR /test_container
--> Running in ce34e009bac4
Removing intermediate container ce34e009bac4
--> e3ded67185b8
Step 4/4 : VOLUME ["/test_container"]
--> Running in 5da541b5e526
Removing intermediate container 5da541b5e526
--> 7350ef19c403
Successfully built 7350ef19c403
Successfully tagged mscit-vol:latest
root@ubuntu:~/mscit-volume# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
mscit-vol latest 7350ef19c403 3 minutes ago 197MB
httpd latest bd29370f84ea 2 days ago 138MB
busybox latest 69593048aa3a 4 weeks ago 1.24MB
ubuntu trusty 13b66b487594 3 months ago 197MB
hello-world latest d1165f221234 4 months ago 13.3kB
root@ubuntu:~/mscit-volume#
```

Mounting the container

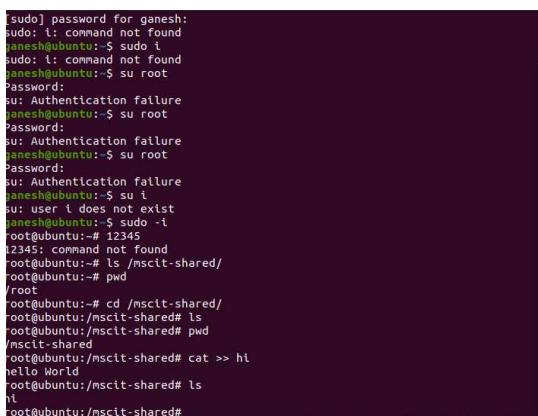
docker run -it --mount src=/mscit-shared,target=/test_container,type=bind mscit-vol

It will change in root and show test_Container



```
root@ubuntu:~/mscit-volume# cd /mscit-shared
root@ubuntu:/mscit-shared# Invalid mount config for type "bind": bind source path does not exist: /mscit-share.
invalid: command not found
root@ubuntu:/mscit-shared# docker run -it --mount src=/mscit-share,target=/test_container,type=bind mscit-vol
docker: Error response from daemon: invalid mount config for type "bind": bind source path does not exist: /mscit-share.
See 'docker run --help'.
root@ubuntu:/mscit-shared# cd /mscit-volume
-bash: cd: /mscit-volume: No such file or directory
root@ubuntu:/mscit-shared# cd..
cd: command not found
root@ubuntu:/mscit-shared# cd
root@ubuntu:# cd mscit-volume
root@ubuntu:~/mscit-volume# mkdir /mscit-shared
mkdir: cannot create directory '/mscit-shared': File exists
root@ubuntu:~/mscit-volume# docker run -it --mount src=/mscit-share,target=/tes
t_container,type=bind mscit-vol
docker: Error response from daemon: invalid mount config for type "bind": bind source path does not exist: /mscit-share.
See 'docker run --help'.
root@ubuntu:~/mscit-volume# docker run -it --mount src=/mscit-shared,target=/te
st_container,type=bind mscit-vol
root@b43347ed3af0:/test_container#
```

Now open other terminal and get into mscit-shared directory and create a file called hi



```
[sudo] password for gamesth
sudo: i: command not found
gamesth@ubuntu:~$ sudo i
sudo: i: command not found
gamesth@ubuntu:~$ su root
Password:
su: Authentication failure
gamesth@ubuntu:~$ su root
Password:
su: Authentication failure
gamesth@ubuntu:~$ su i
su: user i does not exist
gamesth@ubuntu:~$ sudo -i
root@ubuntu:~# 12345
12345: command not found
root@ubuntu:~# ls /mscit-shared/
root@ubuntu:~# pwd
/root
root@ubuntu:~# cd /mscit-shared/
root@ubuntu:/mscit-shared# ls
/mscit-shared
root@ubuntu:/mscit-shared# cat >> hi
Hello World
root@ubuntu:/mscit-shared# ls
hi
root@ubuntu:/mscit-shared#
```

Now check the file created in root is listed in test_Container and vice-versa.

```
root@ubuntu:/msc1-share# ls
hi
root@ubuntu:/msc1-share# ls
hi
root@ubuntu:/msc1-share# cat >> hi
hello World
root@ubuntu:/msc1-share# ls
hi
root@ubuntu:/msc1-share#
```

[sudo] password for ganesh:
sudo: i: command not found
ganesh@ubuntu:~\$ sudo l
sudo: i: command not found
ganesh@ubuntu:~\$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~\$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~\$ su root
Password:
su: Authentication failure
ganesh@ubuntu:~\$ su i
su: user i does not exist
ganesh@ubuntu:~\$ sudo -l
User i is not in the sudoers file.
This incident will be reported.
root@ubuntu:~\$ /msc1-shared/
root@ubuntu:~\$ pwd
/root
root@ubuntu:~# cd /msc1-shared/
root@ubuntu:/msc1-shared# ls
msc1-shared
root@ubuntu:/msc1-shared# pwd
/msc1-shared
root@ubuntu:/msc1-shared# cat >> hi
hello World
root@ubuntu:/msc1-shared# ls
hi
root@ubuntu:/msc1-shared#

[root@b43347ed3af6:/test_container]#

```
root@b43347ed3af6:/test_container# ls
hi
root@b43347ed3af6:/test_container# cat hi
hello World
root@b43347ed3af6:/test_container# ls
hi
root@b43347ed3af6:/test_container#
```

[root@b43347ed3af6:/test_container]#

```
root@ubuntu:/msc1-shared# ls
hi
root@ubuntu:/msc1-shared# ls
hi
root@ubuntu:/msc1-shared# cat >> hi
hello World
root@ubuntu:/msc1-shared# ls
hi
root@ubuntu:/msc1-shared#
```

[root@b43347ed3af6:/test_container]#

We can see that file location are mapped.

When below command is executed, it will delete the volume.

```
docker volume rm mscit-test
```

Network:

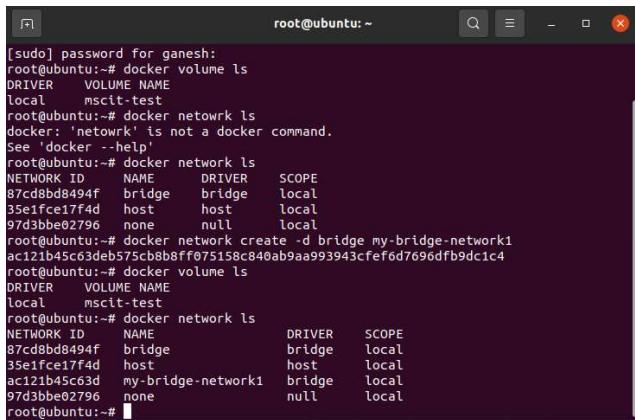
Create network with following command

```
docker network create -d bridge my-bridge-network1
```

```
ganesh@ubuntu:~$ docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
ganesh@ubuntu:~$ sudo -i
[sudo] password for ganesh:
root@ubuntu:~# docker volume ls
DRIVER      VOLUME NAME
local       mscit-test
root@ubuntu:~# docker netowrk ls
docker: 'netowrk' is not a docker command.
See 'docker --help'
root@ubuntu:~# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
87cd8bd8494f   bridge    bridge      local
35e1fce17f4d   host      host      local
97d3bbe02796   none      null      local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
ac121b45c63deb575cb88ff075158cb840ab9aa993943cfef6d7696dfb9dc1c4
root@ubuntu:~#
```

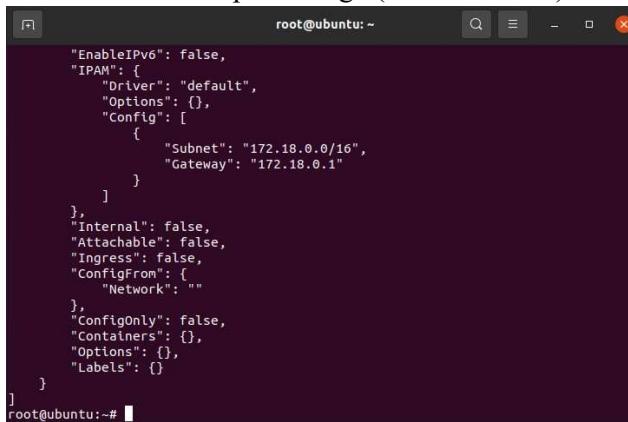
Check network is created with below command

```
Command: docker network ls
```



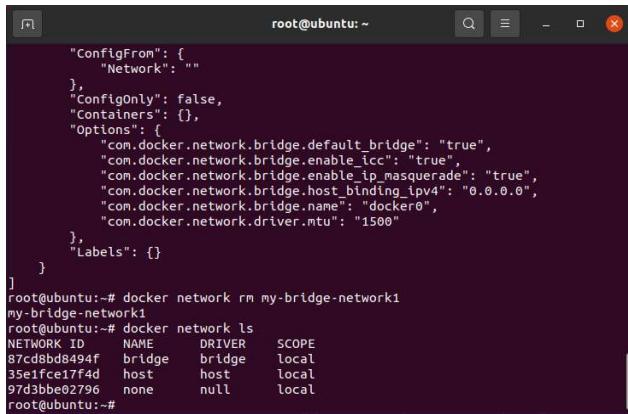
```
[sudo] password for ganesha:  
root@ubuntu:~# docker volume ls  
DRIVER      VOLUME NAME  
local        mscit-test  
root@ubuntu:~# docker netowrk ls  
docker: 'netowrk' is not a docker command.  
See 'docker --help'  
root@ubuntu:~# docker network ls  
NETWORK ID      NAME      DRIVER      SCOPE  
87cd8bd8494f    bridge    bridge    local  
35e1fce17f4d    host      host      local  
97d3bbe02796    none      null      local  
root@ubuntu:~# docker network create -d bridge my-bridge-network1  
ac121b45c63debb575cb8bfff075158c840ab9aa993943cfef6d7696dfb9dc1c4  
root@ubuntu:~# docker volume ls  
DRIVER      VOLUME NAME  
local        mscit-test  
root@ubuntu:~# docker network ls  
NETWORK ID      NAME      DRIVER      SCOPE  
87cd8bd8494f    bridge    bridge    local  
35e1fce17f4d    host      host      local  
ac121b45c63d    my-bridge-network1    bridge    local  
97d3bbe02796    none      null      local  
root@ubuntu:~#
```

We can inspect the created network with below command
docker network inspect bridge (network name)



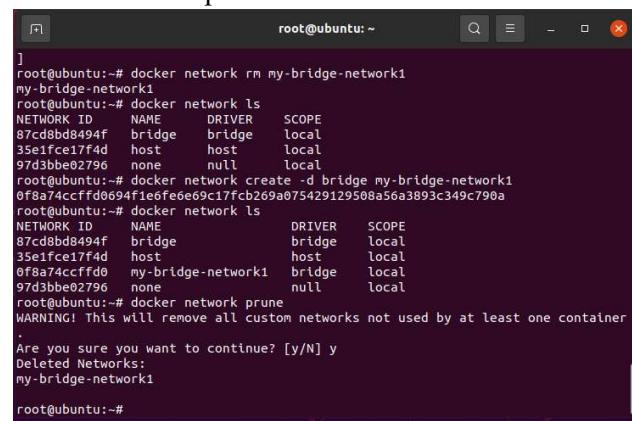
```
root@ubuntu:~#  
{"EnableIPv6": false,  
 "IPAM": [  
     {"Driver": "default",  
      "Options": {},  
      "Config": [  
          {  
              "Subnet": "172.18.0.0/16",  
              "Gateway": "172.18.0.1"  
          }  
      ]  
    },  
    {"Internal": false,  
     "Attachable": false,  
     "Ingress": false,  
     "ConfigFrom": {  
         "Network": ""  
     },  
     "ConfigOnly": false,  
     "Containers": {},  
     "Options": {},  
     "Labels": {}  
   }  
],  
"root@ubuntu:~#"
```

Now, lets remove the created network using below command.
docker network rm network-name
With docker network ls we can see the my-bridge-network1 is deleted.



```
root@ubuntu:~#  
{"ConfigFrom": {  
    "Network": ""  
},  
 "ConfigOnly": false,  
 "Containers": {},  
 "Options": {  
     "com.docker.network.bridge.default_bridge": "true",  
     "com.docker.network.bridge.enable_icc": "true",  
     "com.docker.network.bridge.enable_ip_masquerade": "true",  
     "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",  
     "com.docker.network.bridge.name": "docker0",  
     "com.docker.network.driver.mtu": "1500"  
},  
 "Labels": {}  
}  
root@ubuntu:~# docker network rm my-bridge-network1  
my-bridge-network1  
root@ubuntu:~# docker network ls  
NETWORK ID      NAME      DRIVER      SCOPE  
87cd8bd8494f    bridge    bridge    local  
35e1fce17f4d    host      host      local  
97d3bbe02796    none      null      local  
root@ubuntu:~#
```

With below command we can delete unused networks
docker network prune



A screenshot of a terminal window titled "root@ubuntu:~". The terminal shows the following sequence of commands:

```
[root@ubuntu:~# docker network rm my-bridge-network1
my-bridge-network1
root@ubuntu:~# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
87cd8bbd8494f   bridge    bridge      local
35e1fce17f4d   host      host      local
97d3bbe02796   none      null      local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
0f8a74ccffd0694f1e0fe6e69c17fc269a075429129508a56a3893c349c790a
root@ubuntu:~# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
87cd8bbd8494f   bridge    bridge      local
35e1fce17f4d   host      host      local
0f8a74ccffd0   my-bridge-network1  bridge      local
97d3bbe02796   none      null      local
root@ubuntu:~# docker network prune
WARNING! This will remove all custom networks not used by at least one container
.
Are you sure you want to continue? [y/N] y
Deleted Networks:
my-bridge-network1
root@ubuntu:~#
```

Practical No.5

Aim: Working with Docker Swarm.

(<https://docs.docker.com/get-started/swarm-deploy/>)

Here you will use same folder/project which we use in (Docker images and container of this file)

Make sure that Swarm is enabled on your Docker Desktop by typing docker system info, and looking for a message Swarm: active(you might have to scroll up a little).

If Swarm isn't running, simply type docker swarm init in a shell prompt to set it up.

Describe apps using stack files

Let's write a simple stack file to run and manage our bulletin board. Place the following in a file called bb-stack.yaml

version: '3.7'

services:bb-app:

image: bulletinboard:1.0ports:

- "8000:8080"

Deploy and check your application

1. Deploy your application to

docker stack deploy -c bb-stack.yaml demo

If all goes well, Swarm will report creating all your stack objects with no complaints:

```
F:\Microservices\getting-started-master\app>docker run -dp 3000:3000 getting-started
1e70bcde5379fcf48639f84aa3986fe0db475ff1539654aff48cd98bacbe99d7
```

```
F:\Microservices\getting-started-master\app>docker stack deploy -c bb-stack.yaml demo
Creating network demo_default
Creating service demo_bb-app
```

```
F:\Microservices\getting-started-master\app>
```

2. Make sure everything worked by listing your service

1. docker service ls

2) If all has gone well, your service will report with 1/1 of its replicas created:

```
F:\Microservices\getting-started-master\app>docker service ls
ID          NAME      MODE      REPLICAS  IMAGE
kdzmvo0odg9d  demo_bb-app  replicated  0/1       bulletinboard:1.0  *:8000->8080/tcp
```

1. Open a browser and visit your bulletin board at localhost:8000; you should see your bulletin board, the same as when we ran it as a stand-alone container in Docker images and container section of this file.

2. Once satisfied, tear down your application docker stack rm demo

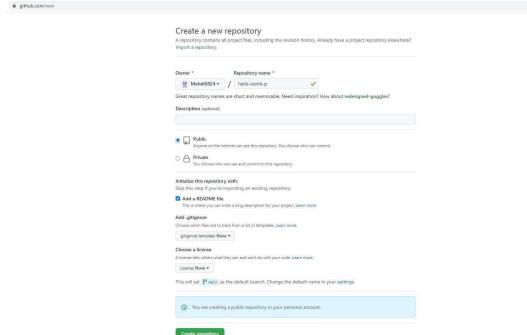
Practical No.6

Aim: Working with Circle CI for continuous integration.

Step 1 - Create a repository

1. Log in to GitHub and begin the process to create a new repository.
2. Enter a name for your repository (for example, hello-world).
3. Select the option to initialize the repository with a README file.
4. Finally, click Create repository.

There is no need to add any source code for now.



Login to Circle CI <https://app.circleci.com/> Using GitHub Login, Once logged in navigate to Projects.

A screenshot of the CircleCI dashboard under the 'Projects' tab. It shows two projects: 'hello-world' and 'hello-world-p'. Each project has a 'Set Up Project' button next to it. The dashboard also includes sections for 'Dashboard', 'Insights', 'Organization Settings', 'Plan', 'Notifications', 'Status', and 'Docs'.

Step 2 - Set up CircleCI

1. Navigate to the CircleCI Projects page. If you created your new repository under an organization, you will need to select the organization name.
2. You will be taken to the Projects dashboard. On the dashboard, select the project you want to set up (hello-world).

Select the option to commit a starter CI pipeline to a new branch, and click Set Up Project. This

will create a file .circleci/config.yml at the root of your repository on a new branch called circleci-project-setup

A screenshot of the CircleCI 'Select your config.yml file' dialog. It provides three options:

- Faster: Use the .circleci/config.yml in my repo
- Faster: Commit a starter CI pipeline to a new branch
- Fast: Take me to a config.yml template that I can edit

The second option is highlighted.

Step 3 - Your first pipeline

On your project's pipeline page, click the green Success button, which brings you to the workflow that ran (sayhelloworld). Within this workflow, the pipeline ran one job, called sayhello. Click say-hello to see the steps in this job:

- Spin up environment
- Preparing environment variables
- Checkout code
- Say hello

Now select the “say-hello-workflow” to the right of Success status column

The screenshot shows the CircleCI Pipeline interface for the 'hello-world' project. A single pipeline run is listed under 'hello-world'. The 'Workflow' column shows 'say-hello-workflow'. The 'Status' column shows a green success icon. The 'Branch / Commit' column shows 'circleci-project-setup' and the commit hash '3446468'. The 'Start' and 'Duration' columns show the run started 2m ago and took 4s. There are several action buttons like 'Edit Config', 'Trigger Pipeline', and 'Project Settings' at the top.

Select “say-hello” Job with a green tick

This screenshot shows the same CircleCI Pipeline interface, but the 'say-hello' job within the 'say-hello-workflow' has been selected. The 'Status' column for this specific job now shows a green checkmark icon. The rest of the pipeline run information remains the same as in the previous screenshot.

Select Branch and option circleci-project-setup

The screenshot shows the GitHub repository settings for 'Mehd5924/hello-world'. Under the 'Code' tab, the 'Branch' dropdown is set to 'main'. A modal window titled 'Switch branch/Tag' is open, showing a dropdown menu with 'circleci-project-setup' selected. Other options like 'main' and 'initial-commit' are also visible. The GitHub interface includes standard navigation bars like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'.

Step 4 - Break your build

In this section, you will edit the `.circleci/config.yml` file and see what happens if a build does not complete successfully. It is possible to edit files directly on GitHub

The screenshot shows the GitHub interface for the 'gkhrivastav/hello-world' repository. A yellow banner at the top indicates a 'circleci-project-setup' branch has been pushed 4 minutes ago. The repository has 1 main file, 2 branches, and 0 tags. It contains 1 commit from 'gkhrivastav' labeled 'Initial commit' made 13 days ago. The README.md file shows the content 'hello-world'. There are no releases or packages published.

Let's use the [Node orb](#). Replace the existing config by pasting the following code:

```

1 version: 2.1
2 orbs:
3   node: circleci/node@4.7.0
4 jobs:
5   build:
6     executor:
7       name: node/default
8       tag: '10.4'
9     steps:
10      - checkout
11      - node/with-cache:
12        steps:
13          - run: npm install
14      - run: npm run test

```

The GitHub file editor should look like this

The screenshot shows the GitHub file editor for the 'circleci/.config.yml' file. The code has been updated to include the Node orb configuration. The file now looks like this:

```

version: 2.1
orbs:
  node: circleci/node@4.7.0
jobs:
  build:
    executor:
      name: node/default
      tag: '10.4'
    steps:
      - checkout
      - node/with-cache:
        steps:
          - run: npm install
      - run: npm run test

```

At the bottom, there is a 'Commit changes' dialog box with the message 'we have updated to config.yml file' and two options: 'Commit directly to the .circleci-project-setup branch.' and 'Create a new branch for this commit and start a pull request. Learn more about pull requests.'

Scroll down and Commit your changes on GitHub

After committing your changes, then return to the Projects page in CircleCI. You should see a new pipeline running and it will fail! What's going on? The Node orb runs some common Node

tasks. Because you are working in an empty repository, running npm run test, a Node script, causes the configuration to fail. To fix this, you need to set up a Node project in your repository.

Step 5 – Use Workflows

You do not have to use orbs to use CircleCI. The following example details how to create a custom configuration that also uses the workflow feature of CircleCI.

Take a moment and read the comments in the code block below. Then, to see workflows in action, edit your .circleci/config.yml file and copy and paste the following text into it.

```
1 version: 2
2 jobs:
3   # we now have TWO jobs, so that a workflow can coordinate them!
4   one: # This is our first job.
5     docker: # it uses the docker executor
6       - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
7       auth:
8         username: mydockerhub-user
9         password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
10      steps:
11        - checkout # this pulls code down from GitHub
12        - run: echo "A first hello!" # This prints "A first hello" to stdout.
13        - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
14    two: # This is our second job.
15      docker: # it runs inside a docker image, the same as above.
16      - image: cimg/ruby:3.0.2
17      auth:
18        username: mydockerhub-user
19        password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
20      steps:
21        - checkout
22        - run: echo "A more familiar hi!" # We run a similar echo command to above.
23        - run: sleep 15 # and then sleep for 15 seconds.
24    # Under the workflows: map, we can coordinate our two jobs, defined above.
25 workflows:
26   version: 2
27   one_and_two: # this is the name of our workflow
28     jobs: # and here we list the jobs we are going to run.
29       - one
30       - two
```

You don't need to write the comments which are the text after #

2. Commit these changes to your repository and navigate back to the CircleCI Pipelines page. You should see your pipeline running.

```
1 version: 2
2 jobs:
3   # we now have TWO jobs, so that a workflow can coordinate them!
4   one: # This is our first job.
5     docker: # it uses the docker executor
6       - image: cimg/ruby:2.6.8 # specifically, a docker image with ruby 2.6.8
7       auth:
8         username: mydockerhub-user
9         password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
10      steps:
11        - checkout # this pulls code down from GitHub
12        - run: echo "A first hello!" # This prints "A first hello" to stdout.
13        - run: sleep 25 # a command telling the job to "sleep" for 25 seconds.
14    two: # This is our second job.
15      docker: # it runs inside a docker image, the same as above.
16      - image: cimg/ruby:3.0.2
17      auth:
18        username: mydockerhub-user
19        password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
20      steps:
21        - checkout
22        - run: echo "A more familiar hi!" # We run a similar echo command to above.
23        - run: sleep 15 # and then sleep for 15 seconds.
24    # Under the workflows: map, we can coordinate our two jobs, defined above.
25 workflows:
26   version: 2
27   one_and_two: # this is the name of our workflow
28     jobs: # and here we list the jobs we are going to run.
29       - one
30       - two
```

2. Click on the running pipeline to view the workflow you have created. You should see that two jobs ran (or are currently running!) concurrently.

The screenshot shows the CircleCI dashboard with three pipelines listed:

- hello-world:1**: Status: Success, Started: 20m ago, Duration: 0s.
- hello-world:2**: Status: Running, Started: 14m ago, Duration: 34s / 3s ago. It has two parallel jobs: "one" and "two".
- hello-world:3**: Status: Success, Started: 9m ago, Duration: 0s.

Step 5 – Add some changes to use workspaces

Each workflow has an associated workspace which can be used to transfer files to downstream

jobs as the workflow progress. You can use workspaces to pass along data that is unique to this run and which needed for down jobs. updating config.yml to the following

```

1  version: 2
2  jobs:
3    one:
4      docker:
5        - image: cimg/ruby:3.0.2
6        auth:
7          username: mydockerhub-user
8          password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
9      steps:
10        - checkout
11        - run: echo "A first hello"
12        - run: mkdir -p my_workspace
13        - run: echo "Trying out workspaces" > my_workspace/echo-output
14      persist_to_workspace:
15        # Must be an absolute path, or relative path from working_directory
16        root: my_workspace
17        # Must be relative path from root
18        paths:
19          - echo-output
20    two:
21      docker:
22        - image: cimg/ruby:3.0.2
23        auth:
24          username: mydockerhub-user
25          password: $DOCKERHUB_PASSWORD # context / project UI env-var reference
26      steps:
27        - checkout
28        - run: echo "A more familiar hi"
29        - attach_workspace:
30          # Must be absolute path or relative path from working_directory
31          at: my_workspace

```

```

32      - run: |
33          if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
34              echo "It worked!";
35          else
36              echo "Nope!";
37          fi
38      workflows:
39          version: 2
40          one_and_two:
41              jobs:
42                  - one
43                  - two
44              requires:
45                  - one
46

```

Updated config.yml in GitHub file editor should be updated like this

```

1 version: 2
2 jobs:
3     one:
4         docker:
5             - image: climg/ruby:3.0.2
6             - artifact:
7                 name: mydockerrhub-user
8                 password: $DOCKERRHUB_PASSWORD # context / project UI env-var reference
9         steps:
10             - checkout
11             - run: echo "I first hello"
12             - run: echo "Trying out workspaces" > my_workspace/echo-output
13             - persist_to_workspace:
14                 name: one
15                 path: my_workspace
16                 root: my_workspace
17                 # Must be relative path from root
18                 paths:
19                     - echo-output
20     two:
21         docker:
22             - image: climg/ruby:3.0.2
23             - artifact:
24                 name: mydockerrhub-user
25                 password: $DOCKERRHUB_PASSWORD # context / project UI env-var reference
26         steps:
27             - checkout
28             - run: echo "I more familiar hi"
29             - attach_workspace:
30                 # Must be absolute path or relative path from working_directory
31                 at: my_workspace
32             - run:
33                 if [[ $(cat my_workspace/echo-output) == "Trying out workspaces" ]]; then
34                     echo "It worked!";
35

```

Commit changes

Commit changes
Update config.yml
3rd Update

Commit directly to the `circleci-project-setup` branch.
 If I create a new branch for this commit and start a pull request, Learn more about pull requests.

Commit changes Cancel

Finally your workflow with the jobs running should look like this.

MehdiS824

Dashboard Project Branch Workflow

All Pipelines > hello-world > circleci-project-setup > **one_and_two**

Workflow: one_and_two

Duration: 15s Author: MehdiS824 Author & Message Update config.yml

one 15s two 35

... Insights Re-run

Practical No.7

Aim: Creating Microservice with ASP.NET Core.

Create your service

In your command prompt, run the following command to create your app:

Command: dotnet new webapi -o MyMicroservice --no-https -f net6.0

Then, navigate to the new directory created by the previous command:

Command : cd MyMicroservice

Run your service

In your command prompt, run the following command:

dotnet run

```
C:\Users\Sitlab\MyMicroservice>dotnet run
Building...
Info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5285
Info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
Info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
Info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\Sitlab\MyMicroservice\
```

Wait for the app to display that it's listening on http://localhost:<port number>, and then open a browser and navigate to http://localhost:<port number>/WeatherForecast.

In this example, it showed that it was listening on port 5020, so the following image shows the URL <http://localhost:5020/WeatherForecast>.

```
[{"date": "2022-05-20T11:25:08.090259+05:30", "temperatureC": 29, "temperatureF": 84, "summary": "Mild"}, {"date": "2022-05-21T11:25:08.0905164+05:30", "temperatureC": 39, "temperatureF": 102, "summary": "Mild"}, {"date": "2022-05-22T11:25:08.0905194+05:30", "temperatureC": 42, "temperatureF": 107, "summary": "Chilly"}, {"date": "2022-05-23T11:25:08.0905195+05:30", "temperatureC": 0, "temperatureF": 32, "summary": "Sweltering"}, {"date": "2022-05-24T11:25:08.0905197+05:30", "temperatureC": -2, "temperatureF": 29, "summary": "Bracing"}]
```

Download and install

You'll be asked to register for Docker Store before you can download the installer

Check that Docker is ready to use

Once you've installed, open a new command prompt and run the following command:

Type command:

docker –version

Add a DockerFile

Create a file called Dockerfile with this command:

Command : fsutil file createnew Dockerfile 0

You can then open it in your favorite text editor manually or with this command:

Command : start Dockerfile

Replace the content of the Dockerfile to the following in the text editor:

```
FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
```

```
WORKDIR /src
```

```
COPY MyMicroservice.csproj .
```

```
RUN dotnet restore
```

```
COPY ..
```

```
RUN dotnet publish -c release -o /app
```

```
FROM mcr.microsoft.com/dotnet/aspnet:6.0
```

```
WORKDIR /app
```

```
COPY --from=build /app .
```

```
ENTRYPOINT ["dotnet", "MyMicroservice.dll"]
```

Optional: Add a .dockerignore file

A .dockerignore file reduces the set of files that are used as part of 'docker build'. Fewer files will result in faster builds.

Create a file called .dockerignore file (this is similar to a .gitignore file if you're familiar

with those) with this command:

```
fsutil file createnew .dockerignore 0
```

You can then open it in your favorite text editor manually or with this command:

```
start .dockerignore
```

Replace the content of the .dockerignore to the following in the text editor:

Dockerfile

[b|B]in

[O|o]bj

Create Docker image

Run the following command:

Command prompt

```
docker build -t mymicroservice .
```

The docker build command uses the Dockerfile to build a Docker image.

Command prompt

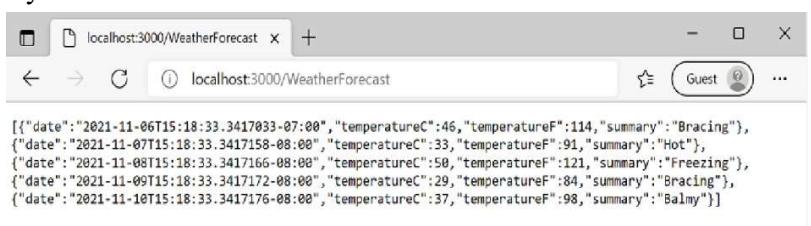
```
docker images
```

Run Docker image

You can run your app in a container using the following command:

Command prompt

```
# docker run -it --rm -p 3000:80 --name mymicroservicecontainer mymicroservice
```

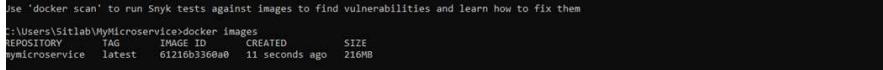


A screenshot of a web browser window titled "localhost:3000/WeatherForecast". The page displays a JSON array of weather data points. Each point includes a date, temperature in Celsius, temperature in Fahrenheit, and a summary. The data points are as follows:

```
[{"date": "2021-11-06T15:18:33.3417033-07:00", "temperatureC": 46, "temperatureF": 114, "summary": "Bracing"}, {"date": "2021-11-07T15:18:33.3417158-08:00", "temperatureC": 33, "temperatureF": 91, "summary": "Hot"}, {"date": "2021-11-08T15:18:33.3417166-08:00", "temperatureC": 58, "temperatureF": 121, "summary": "Freezing"}, {"date": "2021-11-09T15:18:33.3417172-08:00", "temperatureC": 29, "temperatureF": 84, "summary": "Bracing"}, {"date": "2021-11-10T15:18:33.3417176-08:00", "temperatureC": 37, "temperatureF": 98, "summary": "Balmy"}]
```

```
# docker ps
```

Optionally, you can view your container running in a separate command prompt using the following command:



```
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
C:\Users\Sitlab\MyMicroservice>docker images
REPOSITORY          TAG           IMAGE ID            CREATED             SIZE
mymicroservice      latest        61216b3360a0   11 seconds ago   216MB
```

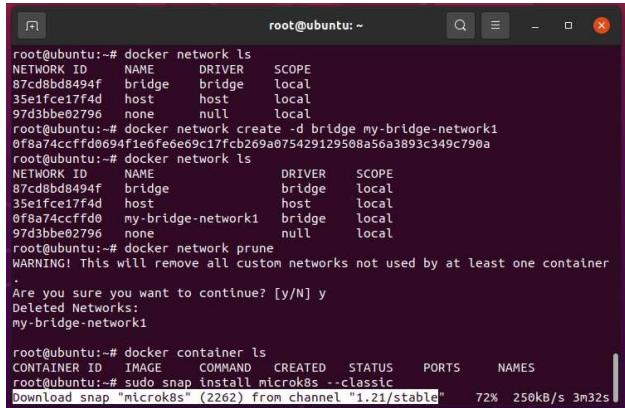
Practical No. 8

Aim: Working with Kubernetes.

Kubernetes, or k8s, is an open-source platform that automates Linux container operations. It eliminates many of the manual processes involved in deploying and scaling containerized applications. “In other words, you can cluster together groups of hosts running Linux containers, and Kubernetes helps you easily and efficiently manage those clusters.”

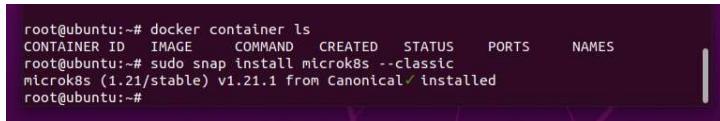
Install MicroK8s on Linux

```
sudo snap install microk8s --classic
```



```
root@ubuntu:~# docker network ls
NETWORK ID     NAME      DRIVER      SCOPE
87cd8bd8494f   bridge    bridge      local
35e1fce17fd4   host      host       local
97d3bbe02796   none      null      local
root@ubuntu:~# docker network create -d bridge my-bridge-network1
0f8a74ccff0d694f1fe0fe6e09c17fc269a075429129508a56a3893c349c790a
root@ubuntu:~# docker network ls
NETWORK ID     NAME      DRIVER      SCOPE
87cd8bd8494f   bridge    bridge      local
35e1fce17fd4   host      host       local
0f8a74ccff0d   my-bridge-network1  bridge      local
97d3bbe02796   none      null      local
root@ubuntu:~# docker network prune
WARNING! This will remove all custom networks not used by at least one container
.
Are you sure you want to continue? [y/N] y
Deleted Networks:
my-bridge-network1

root@ubuntu:~# docker container ls
CONTAINER ID IMAGE      COMMAND     CREATED      STATUS      PORTS      NAMES
root@ubuntu:~# sudo snap install microk8s --classic
Download snap "microk8s" (2262) from channel "1.21/stable"  72%  250kB/s 3m32s
```



```
root@ubuntu:~# docker container ls
CONTAINER ID IMAGE      COMMAND     CREATED      STATUS      PORTS      NAMES
root@ubuntu:~# sudo snap install microk8s --classic
microk8s (1.21/stable) v1.21.1 from Canonical✓ installed
root@ubuntu:~#
```

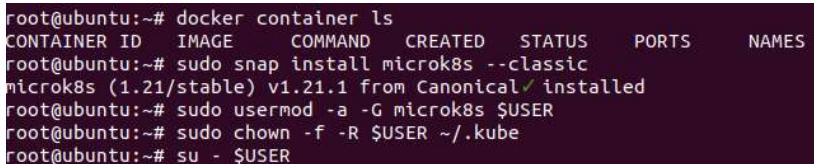
Add your user to the microk8s admin group

MicroK8s creates a group to enable seamless usage of commands which require admin privilege. Use the following commands to join the group:

```
sudo usermod -a -G microk8s $USER
```

```
sudo chown -f -R $USER ~/.kube
```

```
su - $USER
```



```
root@ubuntu:~# docker container ls
CONTAINER ID IMAGE      COMMAND     CREATED      STATUS      PORTS      NAMES
root@ubuntu:~# sudo snap install microk8s --classic
microk8s (1.21/stable) v1.21.1 from Canonical✓ installed
root@ubuntu:~# sudo usermod -a -G microk8s $USER
root@ubuntu:~# sudo chown -f -R $USER ~/.kube
root@ubuntu:~# su - $USER
```

Check the status while Kubernetes starts

```
microk8s status --wait-ready
```

```

root@ubuntu:~# microk8s status --wait-ready
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dashboard      # The Kubernetes dashboard
    ha-cluster     # Configure high availability on the current node
    metrics-server # K8s Metrics Server for API access to service metrics
  disabled:
    ambassador     # Ambassador API Gateway and Ingress
    cilium         # SDN, fast with full network policy
    dns            # CoreDNS
    fluentd        # Elasticsearch-Fluentd-Kibana logging and monitoring
    gpu            # Automatic enablement of Nvidia CUDA
    helm           # Helm 2 - the package manager for Kubernetes
    helm3          # Helm 3 - Kubernetes package manager
    host-access    # Allow Pods connecting to Host services smoothly
    ingress         # Ingress controller for external access
    istio          # Core Istio service mesh services
    jaeger         # Kubernetes Jaeger operator with its simple config
    keda           # Kubernetes-based Event Driven Autoscaling
    knative        # The Knative framework on Kubernetes.
    kubeflow       # KubeFlow for easy ML deployments
    linkerd        # Linkerd is a service mesh for Kubernetes and other
    frameworks
    metalLB        # Loadbalancer for your Kubernetes cluster
    multus         # Multus CNI enables attaching multiple network interfaces to pods
    openebs        # OpenEBS is the open-source storage solution for Kubernetes

```

Turn on the services you want
 microk8s enable dashboard dns ingress

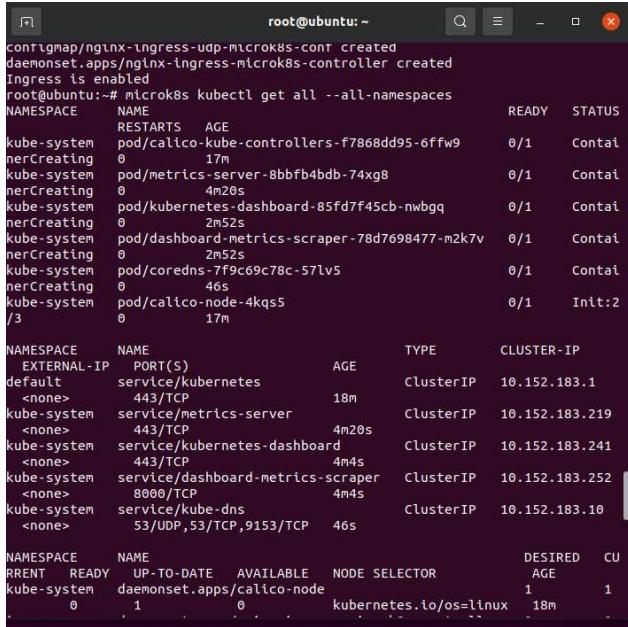
```

root@ubuntu:~# microk8s enable dashboard dns ingress
Addon dashboard is already enabled.
Enabling DNS
Applying manifest
serviceaccount/coredns created
configmap/coredns created
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
Enabling Ingress
ingressclass.networking.k8s.io/public created
namespace/ingress created
serviceaccount/nginx-ingress-microk8s-serviceaccount created
clusterrole.rbac.authorization.k8s.io/nginx-ingress-microk8s-clusterrole created
role.rbac.authorization.k8s.io/nginx-ingress-microk8s-role created
clusterrolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
rolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
configmap/nginx-load-balancer-microk8s-conf created
configmap/nginx-ingress-tcp-microk8s-conf created
configmap/nginx-ingress-udp-microk8s-conf created
daemonset.apps/nginx-ingress-microk8s-controller created
Ingress is enabled
root@ubuntu:~# microk8s kubectl get all --all-namespaces
NAMESPACE      NAME          READY   STATUS
NAMESPACE      NAME          RESTARTS   AGE
kube-system    pod/calico-kube-controllers-f7868dd95-6ffw9   0/1     Contai
nerCreating   0             17m
kube-system    pod/metrics-server-8bbfb4bdb-74xg8   0/1     Contai
nerCreating   0             4m20s
kube-system    pod/kubernetes-dashboard-85fd7f45cb-nwbqq   0/1     Contai
nerCreating   0             2m52s
kube-system    pod/dashboard-metrics-scraper-78d7698477-m2k7v   0/1     Contai

```

Start using Kubernetes

microk8s kubectl get all --all-namespaces



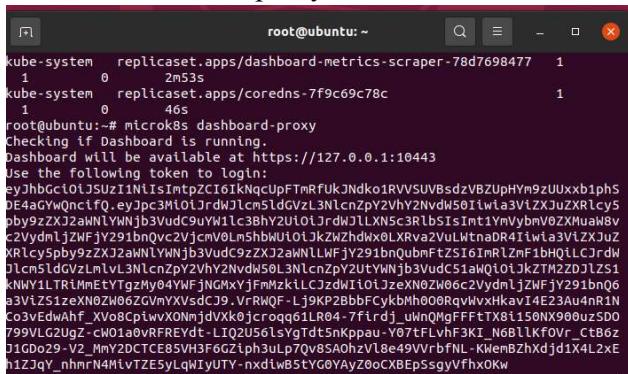
```
root@ubuntu:~ configmap/nginx-ingress-udp-microk8s-conf created
daemonset.apps/nginx-ingress-microk8s-controller created
Ingress is enabled
root@ubuntu:~# microk8s kubectl get all --all-namespaces
NAMESPACE      NAME          READY   STATUS
kube-system    RESTARS     AGE
pod/calico-kube-controllers-f7868dd95-6ffw9   0/1    Contai
nerCreating   0           17m
pod/metrics-server-8bbfb4bdb-74xg8   0/1    Contai
nerCreating   0           4m20s
pod/kubernetes-dashboard-85fd7f45cb-nwbqq   0/1    Contai
nerCreating   0           2m52s
pod/dashboard-metrics-scraper-78d7698477-m2k7v 0/1    Contai
nerCreating   0           2m52s
pod/coredns-7f9c69c78c-57lv5   0/1    Contai
nerCreating   0           46s
pod/calico-node-4kqs5   0/1    Init:2
/3

NAMESPACE      NAME          EXTERNAL-IP PORT(S)      AGE   TYPE   CLUSTER-IP
default        service/kubernetes <none>        443/TCP       18m  ClusterIP  10.152.183.1
kube-system    service/metrics-server <none>        443/TCP       4m20s ClusterIP  10.152.183.219
kube-system    service/kubernetes-dashboard <none>        443/TCP       4m4s  ClusterIP  10.152.183.241
kube-system    service/dashboard-metrics-scraper <none>        8000/TCP     4m4s  ClusterIP  10.152.183.252
kube-system    service/kube-dns <none>        53/UDP,53/TCP,9153/TCP 46s  ClusterIP  10.152.183.10

NAMESPACE      NAME          CURRENT-READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR          DESIRED   CU
kube-system    daemonset.apps/calico-node  0/1           1           0          kubernetes.io/os=linux  1          1
root@ubuntu:~#
```

Access the Kubernetes dashboard

microk8s dashboard-proxy

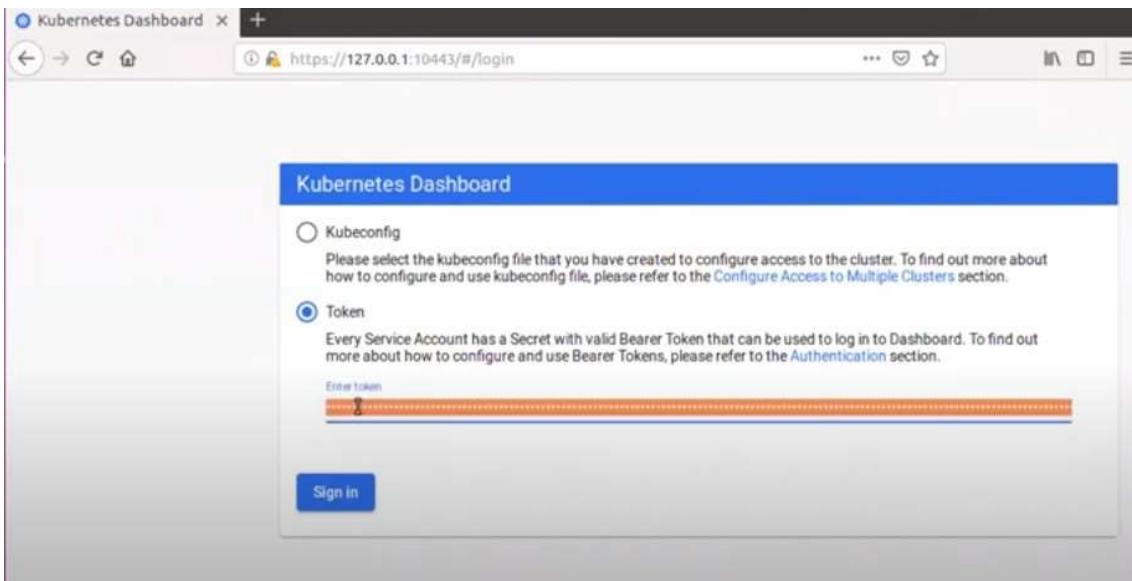


```
root@ubuntu:~ kube-system  replicaset.apps/dashboard-metrics-scraper-78d7698477  1
1  0           2m53s
kube-system  replicaset.apps/coredns-7f9c69c78c   1
1  0           46s
root@ubuntu:~# microk8s dashboard-proxy
Checking if Dashboard is running.
Dashboard will be available at https://127.0.0.1:10443
Use the following token to login:
eyJhbGciOiJSUzI1NiIsImtpZCI6IkNqcUpFTmRfUkJNdko1RVVSUVBsdzVBZUpHYm9zU
Uxxb1phSDE4aGYwQncifQeyJpc3MiOiJrdWJlcm5ldGVzL3NlcZpY2VhY2NvdW50Iiwi
a3ViZXJuZXRLcy5pb9zZXJ2aNWIYWNjb3VudC9uYW1lc3BhY2UiOjRdWJlXN5c3Rlb
SISImt1YmVybmV0ZXMuaw8vc2VydmljZWFlY291bnQvc2VjcnwLShbkU10LkzWZhdwLxLx
Rva2VulWtnadR4i1iwiLa3iZXJuZ
XRLcy5pb9zZXJ2aNWIYWNjb3VudC9zXJ2aNWIWFjY291bnQubmFtZSI6mRlZmF1bhQ1LCJrdW
Jlcm5ldGVzLmvl3NlcZpY2VhY2Nvdw50L3NlcnpY2utvNjbj3vudC51aWQ1o1jkZTM2ZDJLZS1
Jlcm5ldGVzLmvl3NlcZpY2VhY2Nvdw50L3NlcnpY2utvNjbj3vudC51aWQ1o1jkZTM2ZDJLZS1
ZSI6mRlZmF1bhQ1LCJrdWJlcm5ldGVzLmvl3NlcZpY2V
hY2Nvdw50L3NlcZpY2UtYWNjb3VudC51aWQ1o1jkZTM2ZDJLZS1kNWY1LTRiMmEt
YTgzMy04YWFjNGMxYjFmMzkiLCJzdWIiOjzeXN0ZW06c2VydmljZWFlY291bnQ6a3
ViZS1zeXN0ZW06ZGVmYXVsdcJ9.VrRWQF-Lj9KP2bbfCykbMh0O0RqvWvxHkavI4E23Au4nR1N
Co3vEdwAhf_XVo8CpiwXONnjdvXk0jcroqq61LR04-7firdj_uWnQMgFFFtTX8i150NX900uzSDO
799VLG2UgZ-cW01a0vRFREYdt-LIQ2U56lsyGtdtSnkppau-Y07tFLvhF3KI_N6BllKfOvR_Ctb6z
J1Gdo29-V2_MmY2DCTCE85VH3F6G2iph3uLp7Qv8SAohzVl8e49VVrbfNL-KWemBZhXjd1X4L2x
h1Z3qY_nhmRN4MivTZE5yLqWiyUTY-nxd1wB5tYGAyZoocXBEPsSsgyVfhxOKw
```

Token for login:

eyJhbGciOiJSUzI1NiIsImtpZCI6IkNqcUpFTmRfUkJNdko1RVVSUVBsdzVBZUpHYm9zU
Uxxb1phSDE4aGYwQncifQeyJpc3MiOiJrdWJlcm5ldGVzL3NlcZpY2VhY2NvdW50Iiwi
a3ViZXJuZXRLcy5pb9zZXJ2aNWIYWNjb3VudC9uYW1lc3BhY2UiOjRdWJlXN5c3Rlb
SISImt1YmVybmV0ZXMuaw8vc2VydmljZWFlY291bnQvc2VjcnwLShbkU10LkzWZhdwLxLx
Rva2VulWtnadR4i1iwiLa3iZXJuZ
XRLcy5pb9zZXJ2aNWIYWNjb3VudC9zXJ2aNWIWFjY291bnQubmFtZSI6mRlZmF1bhQ1LCJrdW
Jlcm5ldGVzLmvl3NlcZpY2VhY2Nvdw50L3NlcnpY2utvNjbj3vudC51aWQ1o1jkZTM2ZDJLZS1
Jlcm5ldGVzLmvl3NlcZpY2VhY2Nvdw50L3NlcnpY2utvNjbj3vudC51aWQ1o1jkZTM2ZDJLZS1
ZSI6mRlZmF1bhQ1LCJrdWJlcm5ldGVzLmvl3NlcZpY2V
hY2Nvdw50L3NlcZpY2UtYWNjb3VudC51aWQ1o1jkZTM2ZDJLZS1kNWY1LTRiMmEt
YTgzMy04YWFjNGMxYjFmMzkiLCJzdWIiOjzeXN0ZW06c2VydmljZWFlY291bnQ6a3
ViZS1zeXN0ZW06ZGVmYXVsdcJ9.VrRWQF-Lj9KP2bbfCykbMh0O0RqvWvxHkavI4E23Au4nR1N
Co3vEdwAhf_XVo8CpiwXONnjdvXk0jcroqq61LR04-7firdj_uWnQMgFFFtTX8i150NX900uzSDO
799VLG2UgZ-cW01a0vRFREYdt-LIQ2U56lsyGtdtSnkppau-Y07tFLvhF3KI_N6BllKfOvR_Ctb6z
J1Gdo29-V2_MmY2DCTCE85VH3F6G2iph3uLp7Qv8SAohzVl8e49VVrbfNL-KWemBZhXjd1X4L2x
h1Z3qY_nhmRN4MivTZE5yLqWiyUTY-nxd1wB5tYGAyZoocXBEPsSsgyVfhxOKw

Sign in with token:



The screenshot shows the Kubernetes Dashboard Cluster Roles page. At the top, there is a navigation bar with icons for back, forward, refresh, and search, along with a URL field showing <https://127.0.0.1:10443/#/clusterrole?namespace=default>. Below the navigation bar is a header bar with the text "kubernetes" and a search bar. The main content area has a blue header bar with the text "Cluster > Cluster Roles". On the left side, there is a sidebar with a "Cluster" section containing "Cluster Roles" (which is selected), "Namespaces", "Nodes", "Persistent Volumes", and "Storage Classes". Below that is a "Namespace" section with "default". Further down are sections for "Overview", "Workloads", "Cron Jobs", "Daemon Sets", "Deployments", "Jobs", and "Pods". The main content area displays a table titled "Cluster Roles". The table has columns for "Name", "Age", and three vertical ellipsis ("..."). The data in the table is as follows:

Name	Age	...
nginx-ingress-microk8s-clusterrole	2 hours	...
coredns	2 hours	...
kubernetes-dashboard	2 hours	...

At the bottom right of the table, there is a pagination indicator showing "1 - 3 of 3" and navigation arrows.

University Of Mumbai
Institute of Distance & Open Learning



**PRACTICAL JOURNAL IN
IMAGE PROCESSING**

**SUBMITTED BY
APPLICATION ID:
SEAT NO :**

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY PART-I
SEMESTER II**

**ACADEMIC YEAR
2022-2023**

**INSTITUTE OF DISTANCE AND OPEN LEARNING
IDOL BUILDING, VIDYANAGARI,
SANTACRUZ (EAST), MUMBAI-400 098**

**CONDUCTED AT
PARLE TILAK VIDYALAYA ASSOCIATION'S
MULUND COLLEGE OF COMMERCE(AUTONOMOUS)
MULUND (WEST), MUMBAI – 400080**

**University of Mumbai
Institute of Distance & Open Learning**



Dr. Shankar Dayal Sharama Bhavan, Kalina,
Vidanagari, Santacruz (E), Mumbai-400 098.

Certificate

This is to certify that Mr. Application ID:F from Institute of Distance and Open Learning, University of Mumbai , Santacruz East, Mumbai-400098 , PCP Centre: Parle Tilak Vidyalaya Association's Mulund College of Commerce (Autonomous) has successfully completed all the practical of Paper II titled **IMAGE PROCESSING** for M.Sc. (I.T.) Part I Semester – II during the academic year **2022-2023**.

Centre Coordinator

External Examiner

Date:

Seal

INDEX

Sr. No.	Practical Title	Date	Sign
1	<p>Basics</p> <ul style="list-style-type: none"> a. Program to calculate number of samples required for an image. b. Program to study the effects of reducing the spatial resolution of a digital image. c. Program to study the effects of varying the number of intensity levels in a digital image d. Program to perform image averaging (image addition) for noise reduction. e. Program to compare images using subtraction for enhancing the difference between images. 		
2	<p>Image Enhancement</p> <p>A Basic Intensity Transformation functions</p> <ul style="list-style-type: none"> i. Program to perform Image negation ii. Program to perform threshold on an image. iii. Program to perform Log transformation iv. Power-law transformations v. Piecewise linear transformations <p>a. Contrast Stretching</p> <p>b. Gray-level slicing with and without background.</p> <p>c. Bit-plane slicing</p> <p>B</p> <ul style="list-style-type: none"> 1. Program to plot the histogram of an image and categorise 2. Program to apply histogram equalization C. Write a program to perform convolution and correlation D. Write a program to apply smoothing and sharpening filters on grayscale and color images a) Low Pass b) High Pass 		
3	<p>Filtering in Frequency Domain</p> <ul style="list-style-type: none"> a) Program to apply Discrete Fourier Transform on an image b) Program to apply Low pass and High pass filters in frequency domain c) Program for butterworth and gaussian filter in frequency domain. d) Program for homomorphic filtering. 		
4	<p>Image Denoising</p> <ul style="list-style-type: none"> a) Program to denoise using spatial mean and median filtering b) Program for Image deblurring using Weiner filters. 		
5	Color Image Processing		

	<p>a) Program to read a color image and segment into RGB planes, histogram of color image. b) Program for converting from one color model to another model. c) Program to apply false colouring(pseudo) on a gray scale image.</p>		
6	<p>Fourier Related Transforms Program to compute Discrete Cosine Transforms.</p>		
7	<p>Morphological Image Processing</p> <ul style="list-style-type: none"> a. Program to apply erosion, dilation, opening, closing b. Program for detecting boundary of an image c. Program to apply Hit-or-Miss transform d. Program to apply morphological gradient on an image e. Program to apply Top-Hat/Bottom-hat Transformations 		
8	<p>Image Segmentation Program for Edge detection using Sobel, Prewitt and Canny.</p>		

Practical No. 1

Theory:

Image Processing:

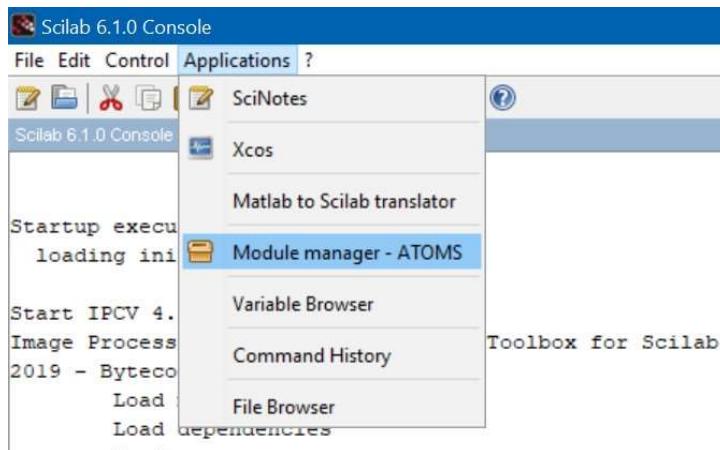
Digital image processing is the use of algorithms and mathematical models to process and analyze digital images. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image-based tasks.

Sampling and Quantization:

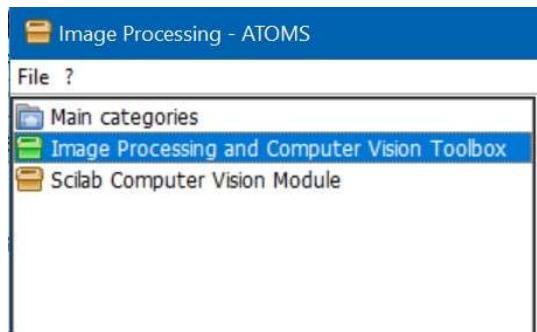
- To create a digital image, we need to convert the continuous sensed data into a digital format.
- This requires two processes: *sampling and quantization*.
- An image may be continuous with respect to the x- and y-coordinates, and also in amplitude.
- To digitize it, we have to sample the function in both coordinates and also in amplitude.
- Digitizing the coordinate values is called **sampling**.
- Digitizing the amplitude values is called **quantization**.

Initial Steps:

- Download SCILAB 6.1.0 and INSTALL it.
- Go to ‘Applications’ □ ‘Module Manage - ATOMSr’.

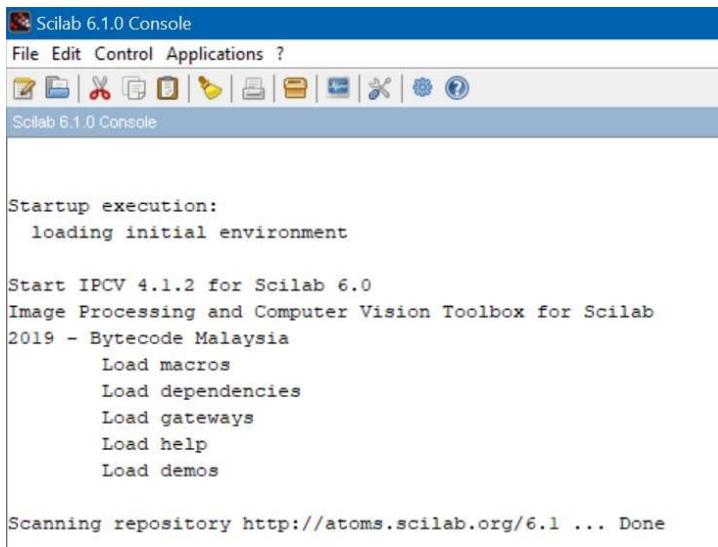


- Select ‘Image Processing’ □ Select ‘Image Processing and Computer Vision Toolbox’



- Click ‘Install’

- Restart scilab. Now opening screen will show following message:



```

Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console

Startup execution:
loading initial environment

Start IPCV 4.1.2 for Scilab 6.0
Image Processing and Computer Vision Toolbox for Scilab
2019 - Bytecode Malaysia
    Load macros
    Load dependencies
    Load gateways
    Load help
    Load demos

Scanning repository http://atoms.scilab.org/6.1 ... Done

```

Part A:

Aim: Program to calculate number of samples required for an image.

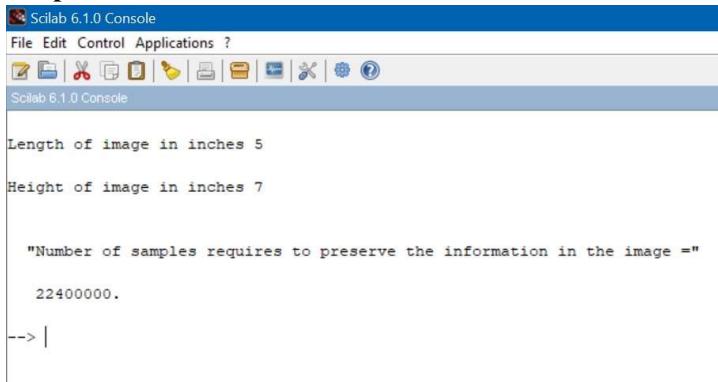
Code:

```

clc;
close;
m=input('Length of image in inches');
n= input('Height of image in inches');
N=400;
N2=2*N;
Fs=m*N2*n*N2;
disp('Number of samples requires to preserve the information in the image =', Fs);

```

Output:



```

Scilab 6.1.0 Console
File Edit Control Applications ?
Scilab 6.1.0 Console

Length of image in inches 5
Height of image in inches 7

"Number of samples requires to preserve the information in the image ="
22400000.

--> |

```

Part B:

Aim: Program to study the effects of reducing the spatial resolution of a digital image.

Code:

```
clc;
clear all;
Img1=imread('D:\MSc IT Image Processing \image\lena.jpeg');
Img = rgb2gray(Img1);
//512*512
subplot (2,3,1),imshow(Img),title('Original image 512*512');
//256*256
Samp=zeros(256);
m=1;
n=1;
for i=1:2:512
    for j=1:2:512
        Samp(m,n)=Img(i,j);
        n=n+1;
    end
    n=1;
    m=m+1;
end
SampImg256=mat2gray(Samp);
subplot(2,3,2);
imshow(SampImg256);
title('Sampled.Img256*256')

/////////
Samp=zeros(128);
m=1;
n=1;
for i=1:4:512
    for j=1:4:512
        Samp(m,n)=Img(i,j);
        n=n+1;
    end
    n=1;
    m=m+1;
end
SampImg128=mat2gray(Samp);
subplot(2,3,3),imshow(SampImg128),title('Sampled.Img128*128')

///////////////////
Samp=zeros(64);
m=1;
n=1;
for i=1:8:512
    for j=1:8:512
        Samp(m,n)=Img(i,j);
```

```

n=n+1;
end
n=1;
m=m+1;
end

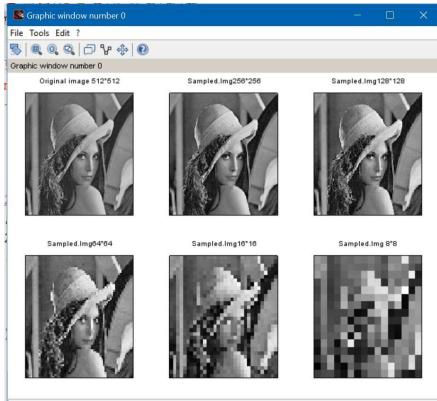
SampImg64=mat2gray(Samp);
subplot(2,3,4),imshow(SampImg64),title('Sampled.Img64*64')

Samp=zeros(16);
m=1;
n=1;
for i=1:16:512
    for j=1:16:512
        Samp(m,n)=Img(i,j);
        n=n+1;
    end
    n=1;
    m=m+1;
end
SampImg16=mat2gray(Samp);
subplot(2,3,5),imshow(SampImg16),title('Sampled.Img16*16')

Samp=zeros(8);
m=1;
n=1;
for i=1:32:512
    for j=1:32:512
        Samp(m,n)=Img(i,j);
        n=n+1;
    end
    n=1;
    m=m+1;
end
SampImg8=mat2gray(Samp);
subplot(2,3,6),imshow(SampImg8),title('Sampled.Img 8*8')

```

Output:



Part C:

Aim: Program to study the effects of varying the number of intensity levels in a digital image

Code:

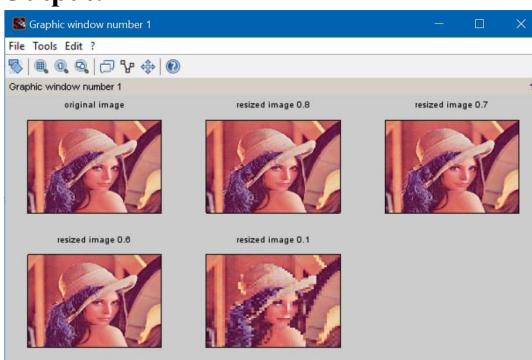
```
clc;
clear all;
figure(1)
subplot(3,3,1);
i=imread('D:\MSc IT Image Processing\image\lena.jpeg');
imshow(i);
title('original image');
subplot(3,3,2);
j1=imresize(i,0.8);
imshow(j1);
title('resized image 0.8');

subplot(3,3,3);
j2=imresize(i,0.7);
imshow(j2);
title('resized image 0.7');

subplot(3,3,4);
j3=imresize(i,0.6);
imshow(j3);
title('resized image 0.6');

subplot(3,3,5);
j4=imresize(i,0.1);
imshow(j4);
title('resized image 0.1');
```

Output:



Part D: and Part E:

Aim: Program to perform image averaging (image addition) for noise reduction.

Aim: Program to compare images using subtraction for enhancing the difference between images.

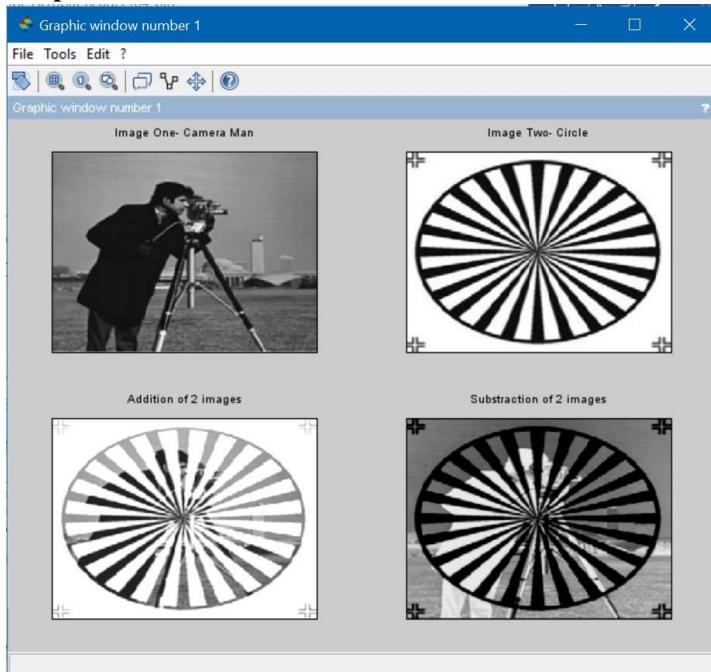
Code:

```

clc;
clear all;
A=imread('D:\MSc IT Image Processing\image\camera.png');
B=imread('D:\MSc IT Image Processing\image\circle.png');
A=rgb2gray(A);
B=rgb2gray(B);
C=imadd(B, A);
D=imsubtract(B,A);
figure(1);
subplot(2,2,1);
imshow(A);
title('Image One- Camera Man');
subplot(2,2,2);
imshow(B);
title('Image Two- Circle');
subplot(2,2,3);
imshow(C);
title('Addition of 2 images');
subplot(2,2,4);
imshow(D);
title('Subtraction of 2 images');

```

Output:



Practical No. 2

Image Enhancement

Theory:

Basic Intensity Transformation:

It is the simplest form of the transformation when the neighbourhood is of size 1×1 . In this case g depends only on the value of f at (x, y) and T becomes an intensity or gray level transformation function of the form $S=T(r)$ r -Denotes the gray level of $f(x, y)$ s -Denotes the gray level of $g(x, y)$ at any point (x, y) Because enhancement at any point in an image deepens only on the gray level at that point, technique in this category is referred to as point processing.

These are the simplest image enhancement techniques.

Image Negative – The negative of an image with gray level in the range $[0, L-1]$ is obtained by using the negative transformation. The expression of the transformation is $s= L-1-r$ Reverting the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is practically suited for enhancing white or gray details embedded in dark regions of an image especially when the black areas are dominant in size.

Log transformations- The general form of log transform is $S=c \log(1+R)$ Where $R \geq 0$ This transformation maps a narrow range of gray level values in the input image into a wider range of output gray levels. The opposite is true for higher values of input levels. These transformations are used to expand the values of dark pixels in an image while compressing the higher-level values. The opposite is true for inverse log transformation. The log transformation function has an important characteristic that it compresses the dynamic range of images with large variations in pixel values.

Power law transformation Power law transformation has the basic function $S= c r^\gamma$ Where c and γ are positive constants. Power law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input gray levels. We may get various curves by varying values of γ .

Piece wise Linear transformation functions- The principal advantage of piecewise linear functions is that these functions can be arbitrarily complex. But their specification requires considerably more user input

Contrast Stretching- It is the simplest piecewise linear transformation function. We may have various low contrast images and that might result due to various reasons such as lack of illumination, problem in imaging sensor or wrong setting of lens aperture during image acquisition. The idea behind contrast stretching is to increase the dynamic range of gray levels in the image being processed.

Bit Plane Slicing:

It is important to highlight the contribution made to the total image appearance by specific bits. Suppose that each pixel is represented by 8 bits.

Imagine that an image is composed of eight 1-bit planes ranging from bit plane 0 for the least significant bit to bit plane 7 for the most significant bit. In terms of 8-bit bytes, plane 0 contains all the lowest order bits in the image and plane 7 contains all the high order bits.

Histogram:

The histogram of a digital image with gray levels in the range [0, L-1] is a discrete function of the form $H(r_k)=n_k$ where r_k is the k th gray level and n_k is the number of pixels in the image having the level r_k . A normalized histogram is given by the equation $P(r_k)=n_k/n$ for $k=0, 1, 2, \dots, L-1$. $P(r_k)$ gives the estimate of the probability of occurrence of gray level r_k . The sum of all components of a normalized histogram is equal to 1. The histogram plots are simple plots of $H(r_k)=n_k$ versus r_k .

Histogram Equalization:

Histogram equalization is a common technique for enhancing the appearance of images. Suppose we have an image which is predominantly dark. Then its histogram would be skewed towards the lower end of the grey scale and all the image detail are compressed into the dark end of the histogram. If we could ‘stretch out’ the grey levels at the dark end to produce a more uniformly distributed histogram then the image would become much clearer.

Part A: Basic Intensity Transformation functions

Sub Part i:

Aim: Program to perform Image negation

Code:

```
//for gray image
clc;
clear all;
A = imread('D:\MSc IT Image Processing\image\negimg.jpg');
A=rgb2gray(A);
subplot(2,1,1);
imshow(A);
title('Original Image ');

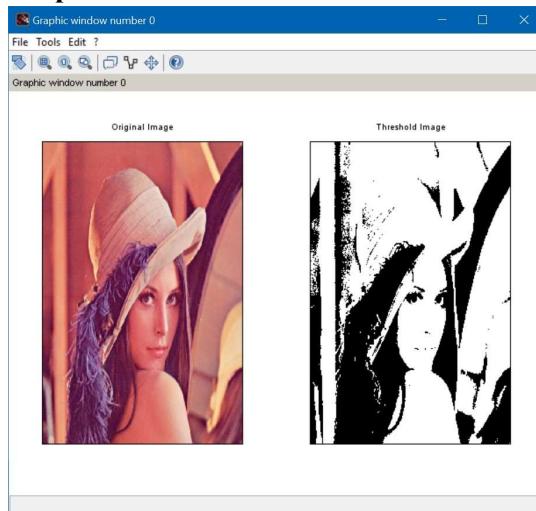
[row col]=size(A);
for x=1:row
    for y=1:col
        A(x,y)=255-A(x,y);
    end
end
subplot(2,1,2);
imshow(A);
title('Image after negation');
```

Output:



Sub Part ii:**Aim: Program to perform threshold on an image.****Code:**

```
clc;
clear all;
a= imread('D:\MSc IT Image Processing\image\lena.jpeg');
b=double(a);
subplot(1,2,1);
imshow(a);
title('Original Image');
t=175;
[m,n]=size(b);
for i = 1:m
    for j=1:n
        if(b(i,j)<t)
            c(i,j)=0;
        else
            c(i,j)=255;
        end
    end
end
subplot(1,2,2);
imshow(c);
title('Threshold Image');
```

Output:**Sub Part iii:****Aim: Program to perform Log transformation****Code:**

```
clc;
clear all;
a=imread('D:\MSc IT Image Processing\image\camera.png');
a=rgb2gray(a);
subplot(2,1,1);
```

```

imshow(a);
title('Original Image ');
c=1;
[r1 c1]=size(a);
for i=1:r1
    for j=1:c1
        b=double(a(i,j));
        s(i,j)=c*log10(1+b);
    end
end
disp('m ended');
new1=mat2gray(s);
//imshow(new1);
subplot(2,1,2);
imshow(new1);
title('Log Image ');

```

Output:



Sub Part iv:

Aim: Power-law transformations

Code:

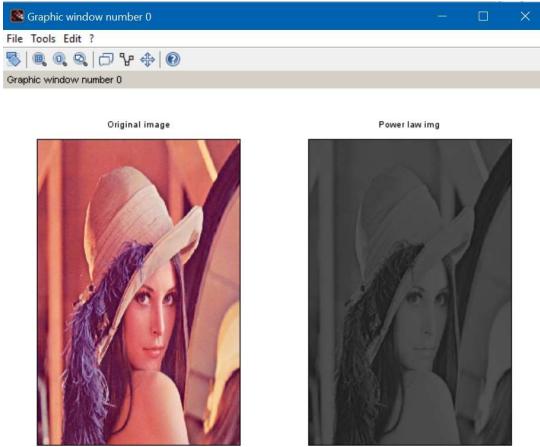
```

a = imread('D:\MSc IT Image Processing\image\lena.png');
b=double(a);
subplot(1,2,1);
imshow(a);
title("Original image");
k=1;
gamma=0.8;
[m,n]=size(b);
for i=1:m
    for j=1:n
        c(i,j)=k*(b(i,j)^gamma);
    end
end
subplot(1,2,2);
imshow(uint8(c));

```

```
title("Power law img");
```

Output:



Sub Part v: Piecewise linear transformations

Sub Sub Part a:

Aim: Contrast Stretching

Code:

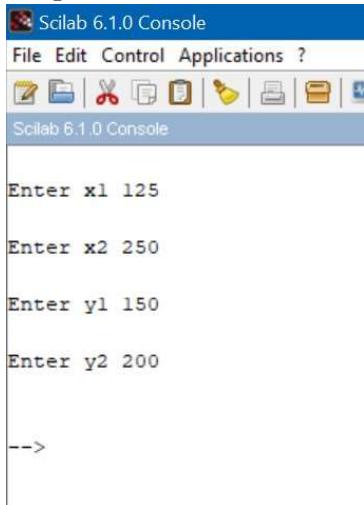
```
clc;
clear all;
a = imread('D:\MSc IT Image Processing\image\lenadark.jpeg');
b=double(a);
[m,n]=size(b); //to get digital image in the form of matrix m by n
x1=input('Enter x1');
x2=input('Enter x2');
y1=input('Enter y1');
y2=input('Enter y2');
slope1=y1/x1;
slope2=(y2-y1)/(x2-x1);
slope3=(255-y2)/(255-x2);
inter1=y1-slope2*x1;
inter2=y2-slope3*x1;
ics=zeros(m,n);
for i=1:m
    for j=1:n
        if(0<b(i,j)&&b(i,j)<x1)
            ics(i,j)=slope1*b(i,j);
        else if(x1< b(i,j)&&b(i,j)<x2)
            ics(i,j)=slope2*b(i,j)+inter1;
        else if(x2< b(i,j)&&b(i,j)<255)
            ics(i,j)=slope3*b(i,j)+inter2;
        end
    end
end
end
end
end
end
subplot(1,2,1);
```

```

imshow(a);
title('Original img');
subplot(1,2,2);
imshow(uint8(ics));
title('Contrast stretch image');

```

Output:



Scilab 6.1.0 Console

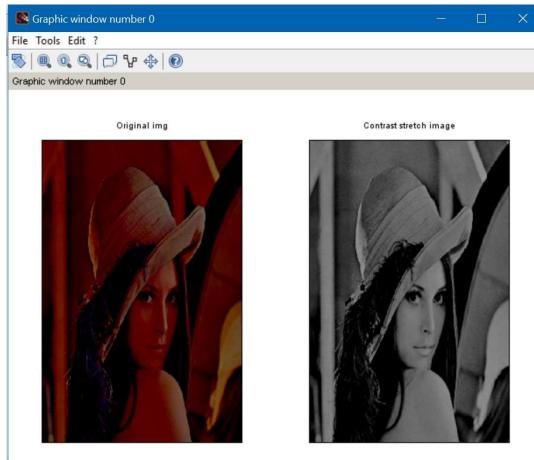
File Edit Control Applications ?

Scilab 6.1.0 Console

```

Enter x1 125
Enter x2 250
Enter y1 150
Enter y2 200
-->

```



Sub Sub Part b:

Aim: Gray-level slicing with and without background.

Code:

```

clc;
clear all;
a = imread('D:\MSc IT Image Processing\image\lenadark.jpeg');
b=double(a);
[m,n]=size(b);
x1=input("Enter x1");
x2=input("Enter x2");
c=zeros(m,n);
d=zeros(m,n);
for i=1:m
for j=1:n

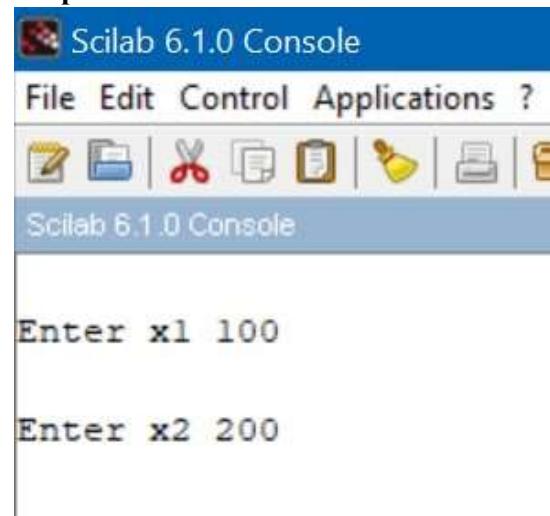
```

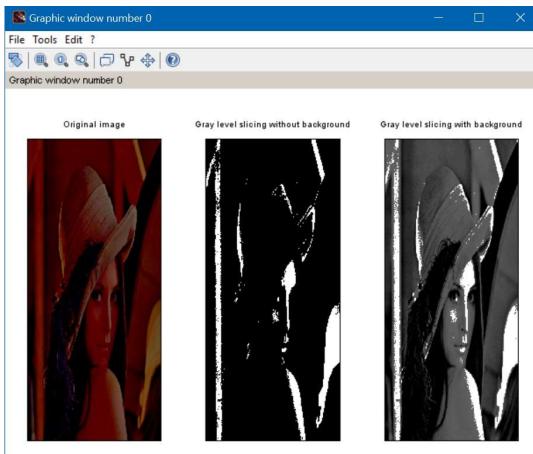
```

if(b(i,j)>=x1 && b(i,j)<=x2);
c(i,j)=255;
else
c(i,j)=0;
end
end
end
for i=1:m
for j=1:n
if(b(i,j)>=x1 && b(i,j)<=x2)
d(i,j)=255;
else
d(i,j)=b(i,j);
end
end
end
subplot(1,3,1);
imshow(a);
title("Original image");
subplot(1,3,2);
imshow(uint8(c));
title("Gray level slicing without background");
subplot(1,3,3);
imshow(uint8(d));
title("Gray level slicing with background");

```

Output:





Sub Sub Part c:

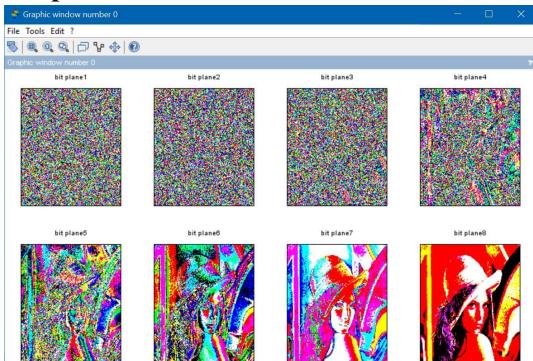
Aim: Bit-plane slicing

Code:

```
clc;
clear all;
a = imread('D:\MSc IT Image Processing\image\lena.png');
b=double(a);
```

```
for i=1:8
    f=bitget(b,i);
    subplot(2,4,i);
    imshow(f);
    title("bit plane" + string(i));
end
```

Output:



Part B:

Sub Part 1:

Aim: Program to plot the histogram of an image and categorise

Code:

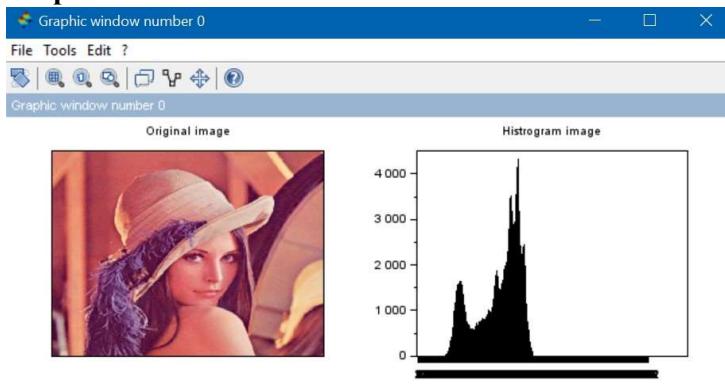
```
clc;
clear all;
Img= imread('D:\MSc IT Image Processing\image\circle.png');
Img1=double(Img);
[row col]=size(Img1);
```

```

h=zeros(row,col);
for n=1:1:row
for m=1:1:col;
if Img1(n,m)==0;
Img1(n,m)=1;
end
end
end
for n=1:1:row
for m=1:1:col
t=Img1(n,m);
h(t)=h(t)+1;
end
end
for n=1:row
for m=1:col
if(h(n,m)>0)
disp(h(n,m));
end
end
end
subplot(2,2,1),imshow(Img),title('Original image')
subplot(2,2,2),bar(h),title('Histogram image');

```

Output:



Sub Part 2:

Aim: Program to apply histogram equalization

Code:

```

clear all;
clc;
a =imread('D:\MSc IT Image Processing\image\lena.png');
a=rgb2gray(a);
b=size(a); // 2 -- b(1)r b(2) c
a=double(a);
freq_counts=zeros(1,256);
for i=1:b(1)
    for j=1:b(2)
        for k=0:255 // l=256 --- 0 to 255

```

```

if a(i,j)==k
    freq_counts(k+1)=freq_counts(k+1)+1;
end
end

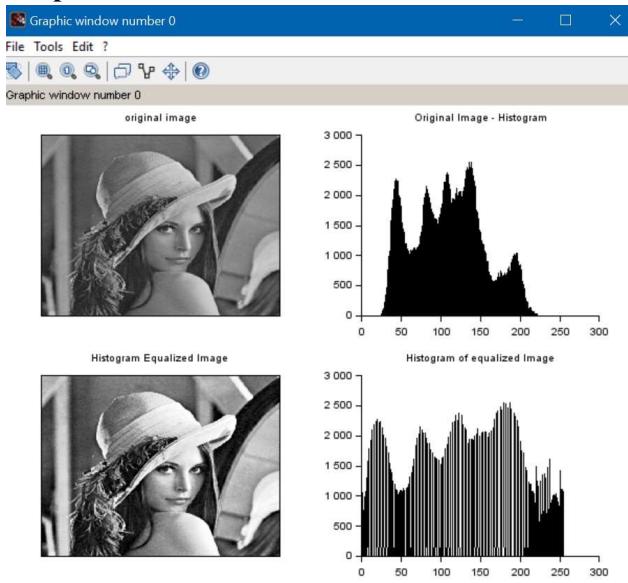
end
pdf=(1/(b(1)*b(2)))*freq_counts; // nk/n

cdf=zeros(1,256);
cdf(1)=pdf(1);
for i=2:256
    cdf(i)=cdf(i-1)+pdf(i);
end
cdf=round(255*cdf); // 256-1
ep=zeros(b); //256 x 256

for i=1:b(1)
    for j=1:b(2)
        t=(a(i,j)+1);
        ep(i,j)=cdf(t);
    end
end
hist2=zeros(1,256);
for i=1:b(1)
    for j=1:b(2)
        for k=0:255
            if ep(i,j)==k
                hist2(k+1)=hist2(k+1)+1;
            end
        end
    end
end
subplot(2,2,1);
// 1 2
// 3 4
imshow(uint8(a));
title('original image');
subplot(2,2,2);
plot2d3([0:255],freq_counts);
title('Original Image - Histogram');
subplot(2,2,3);
imshow(uint8(ep))
title('Histogram Equalized Image');
subplot(2,2,4);
plot2d3([0:255],hist2);
title('Histogram of equalized Image');

```

Output:



Part C:

Aim: Write a program to perform convolution and correlation

Code:

1. convolution

```
clc;
x=[4,5,6;7,8,9];
h=[1;1;1];
y=conv2(x,h);
disp(y);
```

Output:

```
4.      5.      6.
11.    13.    15.
11.    13.    15.
7.      8.      9.
```

--> |

2. Correlation.

Code:

```
clc;
x=input('Enter first sequence');
h=input('Enter second sequence');
y=xcorr(x,h);
disp(x);
disp(h);
```

```
disp(y);
```

Output-

```
Enter first sequence[1 2 3 4]
Enter second sequence[4 3 2 1]

1.    2.    3.    4.

4.    3.    2.    1.

1.    4.    10.   20.   25.   24.   16.

-->
```

Part D:

Aim: Write a program to apply smoothing and sharpening filters on grayscale and color images

a) Low Pass

Code:

```
//low pass filter for smoothing
clc;
clear all;
a1=imread('D:\MSc IT Image Processing\image\morphex.png');
a=double(a1);
[m,n]=size(a);
w=[1 1 1; 1 1 1; 1 1 1];
for i=2:m-1
for j=2:n-1
b(i,j)=(w(1)*a(i-1,j+1)+w(2)*a(i,j+1)+w(3)*a(i+1,j+1)+w(4)*a(i-1,j)+w(5)*a(i,j)+w(6)*a(i+1,j)+w(7)*a(i-1,j-1)+w(8)*a(i,j-1)+w(9)*a(i+1,j-1))
end
end
subplot(2,2,1), imshow(a1), title('Original Image');
subplot(2,2,2), imshow(uint8(b)), title('Low Pass Image');
```

Output:



b) High Pass

Code:

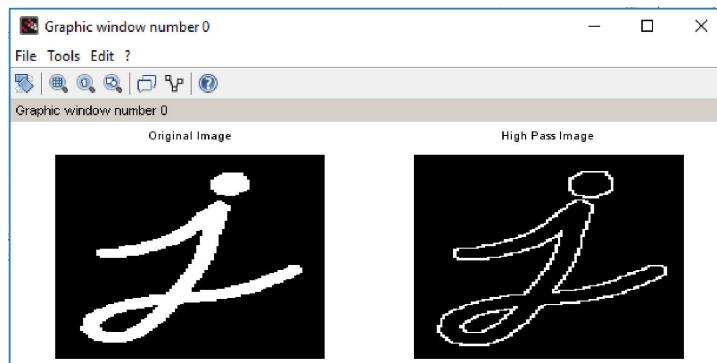
```
//high pass filter for smoothing
clc;
```

```

clear all;
a1=imread('D:\MSc IT Image Processing\image\morplex.png');
a=double(a1);
[m,n]=size(a);
w=[-1 -1 -1; -1 8 -1; -1 -1 -1];
for i=2:m-1
for j=2:n-1
b(i,j)=(w(1)*a(i-1,j+1)+w(2)*a(i,j+1)+w(3)*a(i+1,j+1)+w(4)*a(i-1,j)+w(5)*a(i,j)+w(6)*a(i+1,j)+w(7)*a(i-1,j-1)+w(8)*a(i,j-1)+w(9)*a(i+1,j-1))
end
end
subplot(2,2,1), imshow(a1),title('Original Image');
subplot(2,2,2), imshow(uint8(b)),title('High Pass Image');

```

Output:



Practical No. 3

Filtering in Frequency Domain

Fourier transform named after the French mathematician Jean Baptiste Joseph Fourier is one of the most prominent transforms used in Image processing. Any function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different coefficient, this sum is called Fourier series. Even the functions which are non-periodic but whose area under the curve if finite can also be represented in such form; this is now called Fourier transform.

A function represented in either of these forms and can be completely reconstructed via an inverse process with no loss of information.

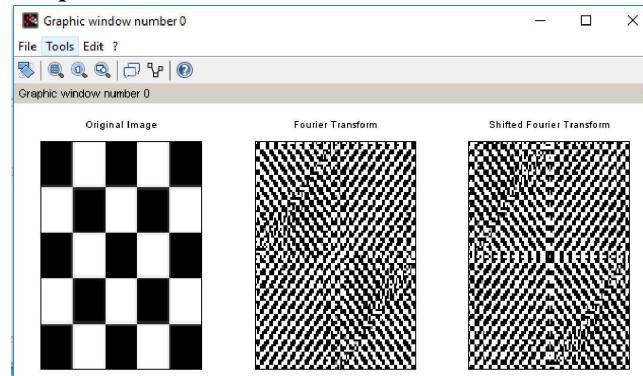
Part A:

Aim: Program to apply Discrete Fourier Transform on an image

Code:

```
clc;
clear;
close;
I=imread('D:\MSc IT Image Processing\image\checkbox.png');
subplot(1,3,1)
imshow(I)
title("Original Image")
I=double(I);
J=fft2(I);
subplot(1,3,2)
imshow(J)
title("Fourier Transform")
L=fftshift(real(J));
subplot(1,3,3)
imshow(L)
title("Shifted Fourier Transform")
```

Output:



Part B:

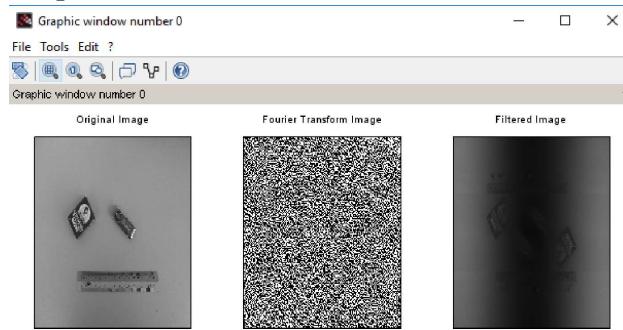
Aim: Program to apply Low pass and High pass filters in frequency domain

Low pass filter-

Code-

```
S=imread("D:\MSc IT Image Processing\image\measure_gray.jpg");
S2=fft2(im2double(S));
h=ffilt('lp',3,0.2,0.6);
img=imfilter(S2,h);
S4=real(ifft(img));
subplot(1,3,1);
imshow(S);
title("Original Image");
subplot(1,3,2);
imshow(S2);
title("Fourier Transform Image");
subplot(1,3,3);
imshow(S4);
title("Filtered Image");
```

Output:

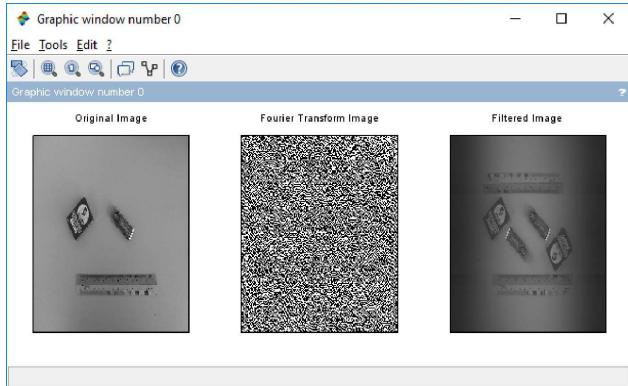


High pass filter-

Code-

```
S=imread("D:\MSc IT Image Processing\image\measure_gray.jpg");
S2=fft2(im2double(S));
h=ffilt('hp',3,0.2,0.6);
img=imfilter(S2,h);
S4=real(ifft(img));
subplot(1,3,1);
imshow(S);
title("Original Image");
subplot(1,3,2);
imshow(S2);
title("Fourier Transform Image");
subplot(1,3,3);
imshow(S4);
title("Filtered Image");
```

Output-



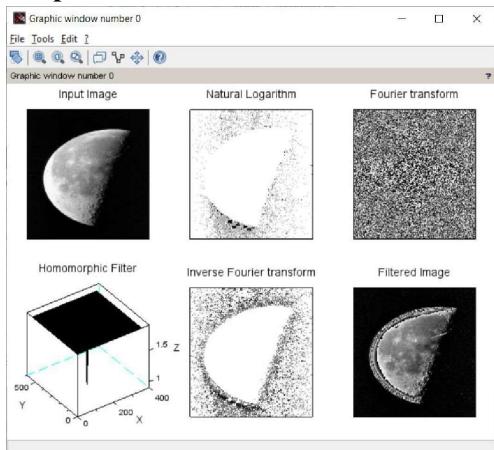
Part C:

Aim: Program for Homomorphic Filtering

Code:

```
1 clear;
2 clear;
3 a = imread('C:\Program Files\scilab-6.1.1\IRCV\images\moon.tif'); //Input image
4 subplot(3,1),imshow(a),title('Input Image');
5 a = double(a);
6 b = a;
7 //constant required for the filter
8 D0 = 50;
9 GL = 0.9;
10 GH= 1.8;
11 [m,n] =size(a); //size of input image
12
13 b=b+1; // add 1 to pixels 0 values which would result in undefined log values
14 log_b=log(b); //Taking log
15 subplot(3,1),imshow(log_b),title('Natural Logarithm');
16 c =fft2(log_b); //Taking fft
17 subplot(3,1),imshow(uint8(c)),title('Fourier transform');
18 dd =ifftshift(c);
19 for u=1:m
20     for v=1:n
21         H(u,v) = (GH,GL)^4*(1-exp(-1^(sqrt((u-m/2)^2+(v-n/2)^2)) ^2/D0)^2)+GL;
22     end
23 end
24 subplot(2,3,4)
25 mesh(H);
26 title('Homomorphic Filter');
27
28 x=dd.'H;
29 real_x=abs(ifft(x));
30 subplot(2,3,5),imshow(real_x),title('Inverse Fourier transform');
31
32 Final =exp(real_x);
33 subplot(2,3,6),imshow(uint8(Final)),title('Filtered Image');
34
```

Output:



Part D:

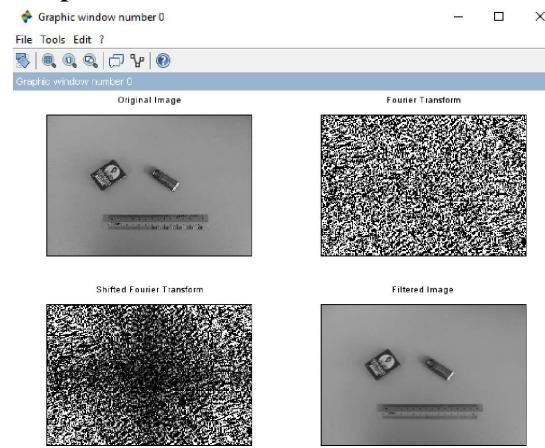
Aim: Program for butterworth and gaussian filter in frequency domain.

Butterworth-

Code:

```
S=imread('D:\MSc IT Image Processing\image\measure_gray.jpg');
h=mkfftfilter(S,'butterworth1',0.4);
S2= fft2(im2double(S));
S3 = S2.*fftshift(h);
S4 = real(ifft(S3));
subplot(2,2,1);
imshow(S);
title("Original Image");
subplot(2,2,2);
imshow(S2);
title("Fourier Transform");
subplot(2,2,3);
imshow(S3);
title("Shifted Fourier Transform");
subplot(2,2,4);
imshow(S4);
title("Filtered Image");
```

Output:



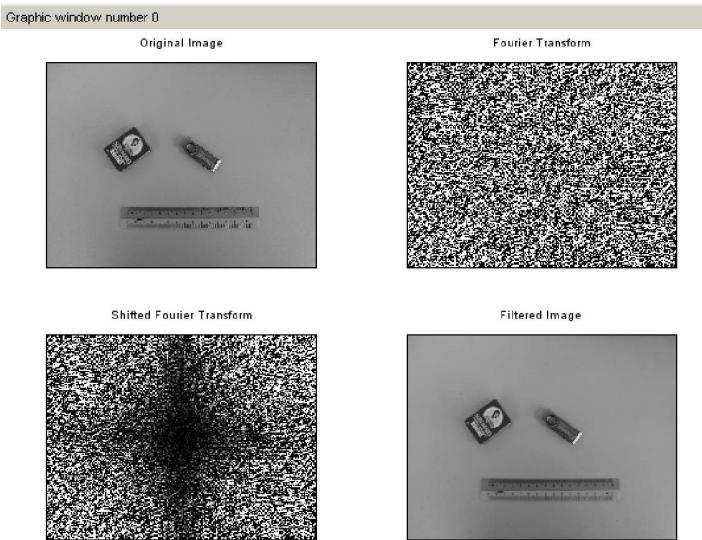
Gaussian filter-

Code:

```
S=imread('D:\MSc IT Image Processing\image\measure_gray.jpg');
h=mkfftfilter(S,'gauss',0.8);
S2= fft2(im2double(S));
S3 = S2.*fftshift(h);
S4 = real(ifft(S3));
subplot(2,2,1);
imshow(S);
title("Original Image");
subplot(2,2,2);
imshow(S2);
```

```
title("Fourier Transform");
subplot(2,2,3);
imshow(S3);
title("Shifted Fourier Transform");
subplot(2,2,4);
imshow(S4);
title("Filtered Image");
```

Output:



Practical No. 4

Image Denoising

Image Denoising is a computer vision task that involves removing noise from an image. Noise can be introduced into an image during acquisition or processing, and can reduce image quality and make it difficult to interpret. Image denoising techniques aim to restore an image to its original quality by reducing or removing the noise, while preserving the important features of the image.

Wiener Filtering is an approach that incorporates both the degradation function and statistical characteristics of noise into the restoration process. The method is founded on considering images and noise as random variables, and the objective is to find an estimate \hat{f} of the uncorrupted image f such that the mean square error between them is minimized.

Part A:

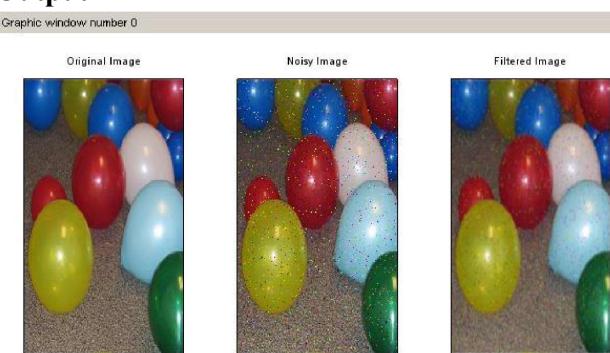
Aim: Program to denoise using spatial mean and median filtering

Mean-

Code-

```
Image=imread('D:\MSc IT Image Processing\image\balloons.png');
//Image=rgb2gray(Image);
NoisyImage=imnoise(Image,'salt & pepper', 0.02);
F1=fspecial('average',3);
FilterImage=imfilter(NoisyImage,F1);
subplot(1,3,1);
imshow(Image);
title('Original Image');
subplot(1,3,2);
imshow(NoisyImage);
title('Noisy Image');
subplot(1,3,3);
imshow(FilterImage);
title('Filtered Image');
```

Output-



Median- I

Code:

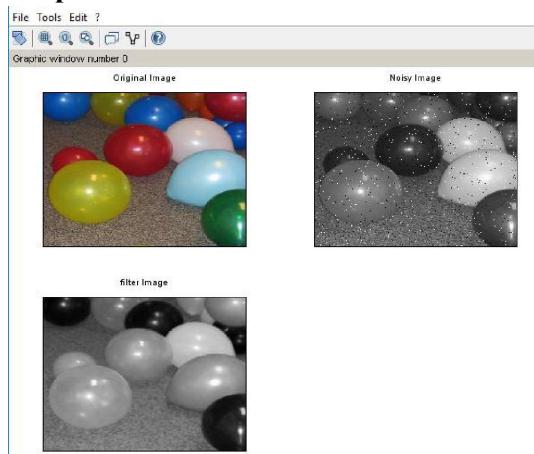
```
clc;
clear all;
```

```

a= imread('D:\MSc IT Image Processing\image\balloons.png');
image=rgb2gray(a);
NoisyImage = imnoise(image,'salt & pepper',0.02);
[m n]=size(a);
for i=2:m-1
for j=2:n-1
output(i,j)=median([a(i-1,j+1),a(i,j+1),a(i+1,j+1),a(i-1,j),a(i+1,j),a(i-1,j-1),a(i,j-1),a(i+1,j-1)]);
end
end
subplot(2,2,1);
imshow(a);
title('Original Image');
subplot(2,2,2);
imshow(NoisyImage);
title('Noisy Image');
subplot(2,2,3);
imshow(output);
title('filter Image');

```

Output:



Median II:

Code:

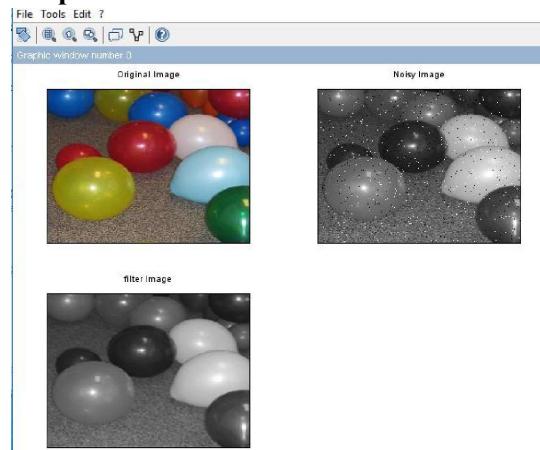
```

clc;
clear all;
a= imread('D:\MSc IT Image Processing\image\balloons.png');
image=rgb2gray(a);
NoisyImage = imnoise(image,'salt & pepper',0.02);
output=immedian(NoisyImage,[3,3]);
subplot(2,2,1);
imshow(a);
title('Original Image');
subplot(2,2,2);
imshow(NoisyImage);
title('Noisy Image');
subplot(2,2,3);

```

```
imshow(output);
title('filter Image');
```

Output:



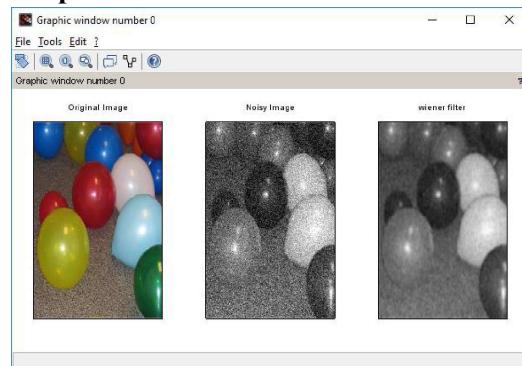
Part B:

Aim: Program for Image deblurring using Weiner filters.

Code:

```
clc;
clear all;
a= imread('D:\MSc IT Image Processing\image\balloons.png');
image=rgb2gray(a);
NoisyImage = imnoise(image,'gaussian',0.02);
wienerfilter=imwiener2(NoisyImage,[5,5],0.2);
subplot(1,3,1);
imshow(a);
title('Original Image');
subplot(1,3,2);
imshow(NoisyImage);
title('Noisy Image');
subplot(1,3,3);
imshow(wienerfilter);
title('wiener filter');
```

Output:



Practical No. 5

Color Image Processing

The utilization of color for image processing is inspired by two key factors. First, color is an important descriptor that also simplifies the identification and extraction of objects from the image. Color image processing is divided into two main parts: full color and pseudo-color processing. In the initial group, the images in question are usually taken with a full-color sensor, such as a color TV camera or a color scanner. In the second group, the issue is to assign a color to a specific monochrome intensity or set of intensities. Until recently, much of the digital color image processing was performed at the pseudo-color level. Even so, over the last decade, color sensors and color image processing hardware have become available at affordable prices. As a result, full-color image processing techniques are now used in a wide variety of applications, including publishing, visualization and the Internet.

Color Models:

By specifying a 3D coordinate system and a subspace that includes all constructible colors within a particular model, color patterns provide a standard way to specify a particular color. Every color that can be described by a model is a single point within the subspace it defines. The basic hardware (RGB, CMY, and YIQ) or image processing applications (HSI) are oriented to each color model.

Part A:

Aim: Program to read a color image and segment into RGB planes, histogram of color image.

Code:

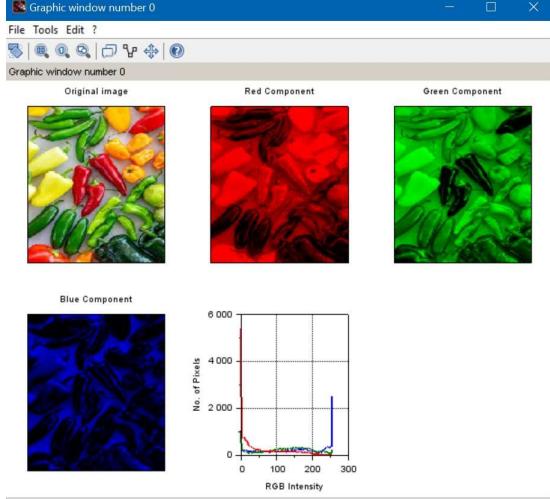
```
clc;
I = imread('D:\MSc IT Image Processing\image\peppers.jpg');
//rows and columns of image.
r=size(I,1);
c=size(I,2);
//creating zero matrices.
R=zeros(r,c,3);
G=zeros(r,c,3);
B=zeros(r,c,3);
//storing the corresponding color plane
//red plane
R(:,:,1) = I(:,:,1);
//green plane
G(:,:,2) = I(:,:,2);
//blue plane
B(:,:,3) = I(:,:,3);
subplot(2,3,1);
imshow(I);
title('Original image');
subplot(2,3,2);
imshow(uint8(R));
title('Red Component');
subplot(2,3,3);
```

```

imshow(uint8(G));
title('Green Component');
subplot(2,3,4);
imshow(uint8(B));
title('Blue Component');
R = I(:,:,1);
G = I(:,:,2);
B = I(:,:,3);
nBins = 256;
//Get histValues for each channel
[yR,x] = imhist(R,nBins);
[yG,x] = imhist(G,nBins);
[yB,x] = imhist(B,nBins);
//Plot them together in one plot
subplot(2,3,5)
plot(x,yR,x,yG,x,yB,"Linewidth",2);
xlabel("RGB Intensity");
ylabel("No. of Pixels");
set(gca(),"grid",[1,1]);

```

Output:



Part B:

Aim: Program for converting from one color model to another model.

Code 1:

```

clc;
rgb=imread('D:\MSc IT Image Processing\image\peppers.jpg');
hsv=rgb2hsv(rgb);
imwrite(hsv,"HSV.tif");
subplot(2,3,1);
imshow(rgb);
title('Original RGB Image');
subplot(2,3,2);
imshow(hsv);
title('HSV Image');

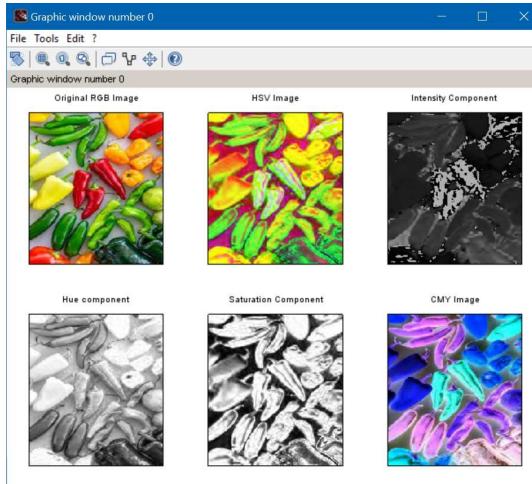
```

```

subplot(2,3,3);
imshow(hsv(:,:,3));
title('Intensity Component');
subplot(2,3,4);
imshow(hsv(:,:,1));
title('Hue component');
subplot(2,3,5);
imshow(hsv(:,:,2));
title('Saturation Component')
cmy=imcomplement(rgb);
imwrite(cmy,"CMY.tif");
subplot(2,3,6);
imshow(cmy);
title('CMY Image');

```

Output 1:



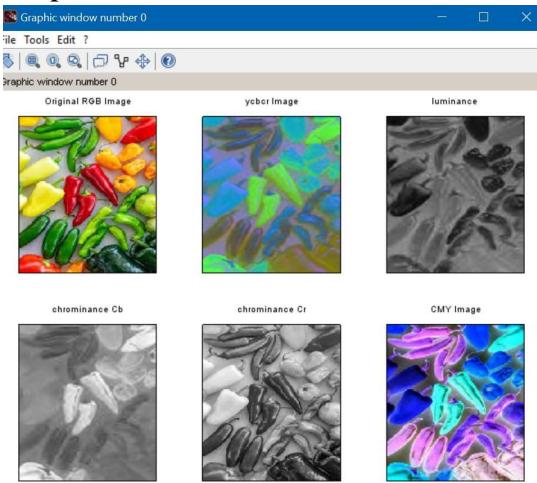
Code 2:

```

clc;
rgb = imread('D:\MSc IT Image Processing\image\peppers.jpg');
ycbcr=rgb2ycbcr(rgb);
subplot(2,3,1);
imshow(rgb);
title('Original RGB Image');
subplot(2,3,2);
imshow(ycbcr);
title('ycbcr Image');
subplot(2,3,3);
imshow(ycbcr(:,:,1));
title('luminance');
subplot(2,3,4);
imshow(ycbcr(:,:,2));
title('chrominance Cb');
subplot(2,3,5);
imshow(ycbcr(:,:,3));
title('chrominance Cr')

```

Output 2:



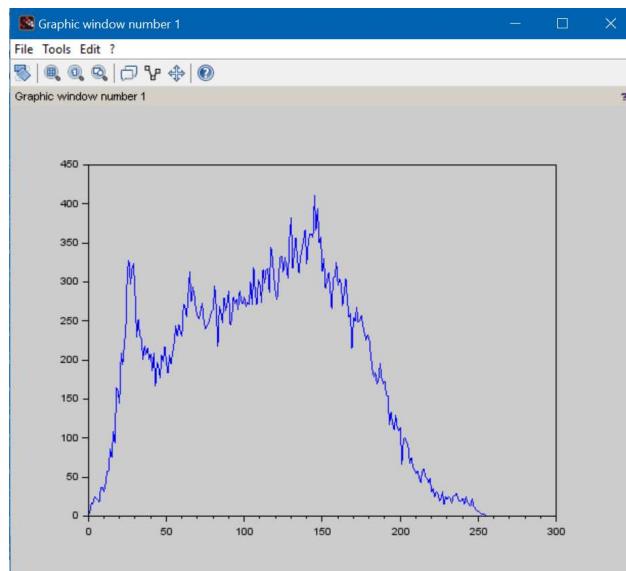
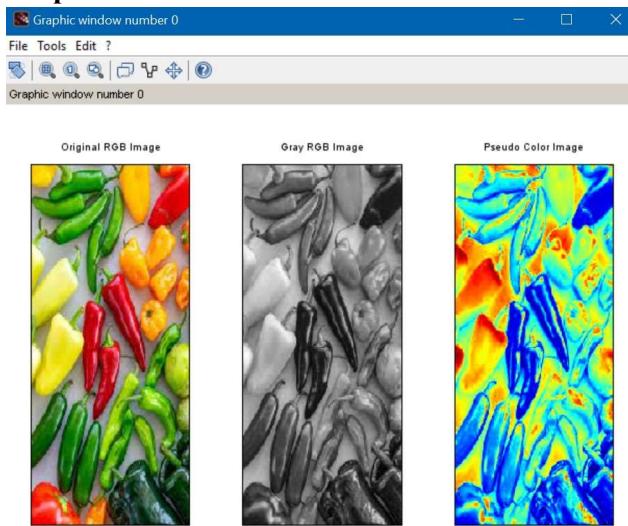
Part C:

Aim: Program to apply false colouring(pseudo) on a gray scale image.

Code:

```
clc;
rgb = imread('D:\MSc IT Image Processing\image\peppers.jpg');
rgb1=double(rgb);
Image = rgb2gray(rgb);
subplot(1,3,1);
imshow(rgb);
title('Original RGB Image');
subplot(1,3,2);
imshow(Image);
title('Gray RGB Image');
subplot(1,3,3);
imshow(Image,jetcolormap(256));
title('Pseudo Color Image');
Histogram = imhist(Image);
figure();
plot(0:255,Histogram');
```

Output:



Practical No. 6

Fourier Related Transforms

The Fourier real function transformation is highly complex. Three transforms associated to Fourier: the discrete Hartley transform, discrete cosine transform, and discrete sine transform. All three transforms handle unnecessary numbers' computational complexity and can be carried out via fast FFT-like algorithms.

Discrete Cosine Transform

The most frequently found form discrete cosine transform (DCT) is achieved by replacing the inverse transformations kernel. There are 8 standard DCT variations and various symmetry conditions are assumed. For example, even about a sample or about a point between two samples could be assumed to be the entry

$$s(x, u) = \alpha(u) \cos\left(\frac{(2x + 1)u\pi}{2N}\right)$$

Where

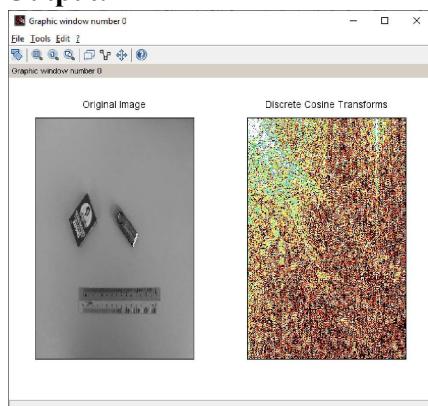
$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \dots, N - 1 \end{cases}$$

Aim: Program to compute Discrete Cosine Transforms.

Code:

```
clear all;
a=imread('D:\MSc IT Image Processing\image\measure_gray.jpg");
dctImg=imdct(a);
subplot(1,2,1);
imshow(a);
title('Original Image');
subplot(1,2,2);
imshow(dctImg, jetcolormap(256));
title('Discrete Cosine Transform');
```

Output:



Practical No. 7

Morphological Image Processing

Morphological image processing is a set of non-linear image processing relevant to the shape or morphology of the image features. As per Wikipedia, morphological operations only depend on the specific ordering of the values of the pixels, not the numerical values therein. Morphological operations may also be used in grayscale images such that the light transfer functions of the grayscale are uncertain and their absolute pixel values are either of no or minor concern.

Morphological techniques evaluate an image called a structural element with a small shape or template. The structuring element is placed anywhere in the image and compared with the corresponding pixel neighborhood. In some operations, the element is tested if it "fits" in the neighborhood, while others test if it "hits" or crosses it:

Erosion is the alternative to dilation. If dilation expands an image, erosion shrinks it. The structural element determines how the image is shrunk.

As the binary image is expanded from its original shape, dilation is defined as the technique of expanding a binary image from one shape to another. The structural element determines how the binary picture is expanded and how it is displayed.

Opening

The opening of image A by image B is denoted by \circ and is defined as the result of erosion and dilation by

$$A \circ B = (A \ominus B) \oplus B$$

Closing

Closing is a complementary procedure to opening, which is described as a dilation followed by an erosion. The closing of A by B is indicated by $A \cdot B$ and defined by

$$A \cdot B = (A \oplus B) \ominus B$$

Hit-and-miss transform

The hit-and-miss transform is a basic binary morphological operation that can be used to inspect an image for specific patterns of foreground and background pixels.

It is, in fact, the fundamental operation of binary morphology, as it is the source of practically all other binary morphological operators. As is the case with other binary morphological operators, it accepts a binary image and a structuring element as input and outputs another binary image.

The hit-or-miss transformation of an image A by B is denoted by $A \circledast B$. B is a structural element pair $B=(B_1, B_2)$. Instead of a single element, B1: the collection of B elements associated with a certain object B2: a collection of B elements that relate to the background

As follows is the definition of the hit-or-miss transform:

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

This transform is important for locating all pixel configurations that correspond to the B1 structure (i.e. a match) but not to the B2 structure (i.e. a miss). As a result, the hit-or-miss transform is used to detect shapes.

Boundary Extraction The boundary of a set A, indicated by $\beta(A)$, can be determined in the following manner: $\beta(A) = A - (A \circ B)$ where B denotes the structuring element.

Morphological Gradient

Dilation and erosion, in conjunction with image subtraction, can be used to obtain the morphological gradient, g , of a grayscale image f in the following manner:

$$g = (f \oplus b) - (f \ominus b)$$

Where b denotes an appropriate structuring element. The cumulative consequence of this equation is that dilation thickens and erosion shrinks regions in an image.

Their distinction highlights the divisions between regions. Because homogenous regions are unaffected (as long as the SE is not too large in relation to the image's resolution), the subtraction operation seeks to eliminate them. As a result, an image is created in which the edges are enhanced and the homogeneous areas are suppressed, creating a "derivative like" (gradient) effect.

Top-Hat and Bottom-Hat Transformations

Combining image subtraction with open and closed operations yields a top-hat transform and a bottom-hat transform. A grayscale image f 's top-hat transformation is defined as f minus its opening.

$$T_{hat}(f) = f - (f \cdot b)$$

Similarly, the bottom-hat transformation of f is defined as the closing of f minus f :

$$B_{hat}(f) = (f \cdot b) - f$$

These transformations have a variety of applications, one of which is the removal of objects from images using an element that opens or closes instead of resizing the object that was removed. After that, the difference operation produces an image that contains only the component that was deleted. In the case of bright objects on a dark background, the top-hat transform is used, whereas the bottom-hat transform is used in the case of the opposite. The white top hat transform and the black bottom-hat transform are two common names for these transformations.

Part A:

Aim: Program to apply erosion, dilation, opening, closing

Erosion:

Code:

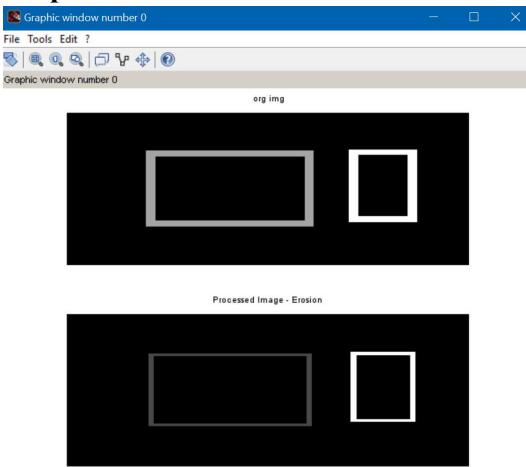
```
clc;
clear all;
a=imread('D:\MSc IT Image Processing\image\rectb.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img');
A1=a;
d=a;
[r,c]=size(d);
m=[2 1 2;1 2 1;2 1 2];
// m=ones(5,5);
for i=2:r-1
for j=2:c-1
```

```

new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1)) (m(5)*d(i,j))
(m(6)*d(i,j+1)) (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
A1(i,j)=min(new);
end
subplot(2,1,2);
title('org img');imshow(A1);title('Processed Image - Erosion');
end

```

Output:



Dilation:

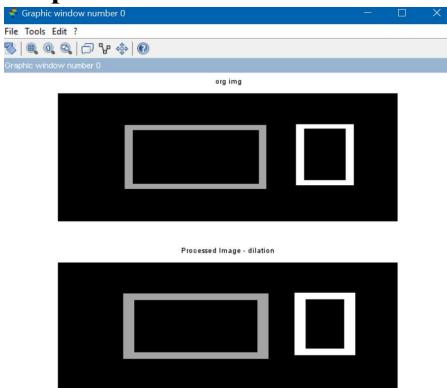
Code:

```

clc;
clear all;
a=imread('D:\MSc IT Image Processing\image\rectb.png');
a=rgb2gray(a);
d=a;
A1=a;
[r,c]=size(d);
subplot(2,1,1);
imshow(a);
title('org img');
m=[1 5 1;1 5 1;1 5 1]; // mask. ...3 x 3 , 5 x 5, 7 x 7
// m=ones(5,5);
for i=2:r-1
for j=2:c-1
new=[(m(1)*d(i-1,j-1)) (m(2)*d(i-1,j)) (m(3)*d(i-1,j+1)) (m(4)*d(i,j-1)) (m(5)*d(i,j))
(m(6)*d(i,j+1)) (m(7)*d(i+1,j-1)) (m(8)*d(i+1,j)) (m(9)*d(i+1,j+1))];
A1(i,j)=max(new);
end
subplot(2,1,2);
imshow(A1);title('Processed Image - dilation');
end

```

Output:

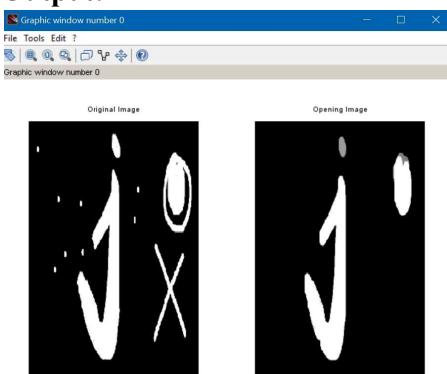


Opening:

Code:

```
a= imread('D:\MSc IT Image Processing\image\letter1.png');
se = imcreatese('ellipse',20,20); // rect, ellipse, cross, dim1, dim2
erosion = imerode(a,se); // image i or a, SE - Structural element
opening = imdilate(erosion,se)
subplot(1,2,1);
imshow(a);
title('Original Image');
subplot(1,2,2);
imshow(opening);
title('Opening Image');
```

Output:



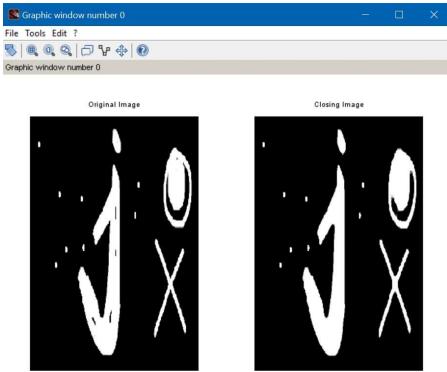
Closing:

Code:

```
clc;
clear all;
a= imread('D:\MSc IT Image Processing\image\letter2.png');
se = imcreatese('ellipse',10,10);
dilate = imdilate(a,se);
closing = imerode(dilate,se)
subplot(1,2,1);
imshow(a);
title('Original Image');
```

```
subplot(1,2,2);
imshow(closing);
title('Closing Image');
```

Output:



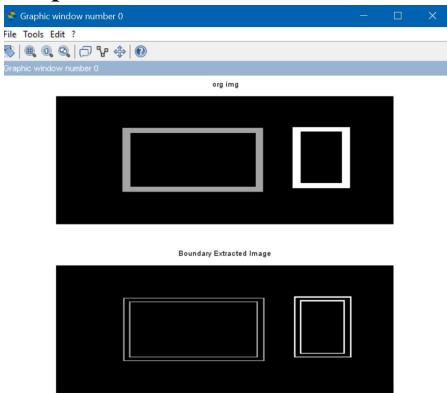
Part B:

Aim: Program for detecting boundary of an image

Code:

```
clc;
clear all;
a=imread('D:\MSc IT Image Processing\image\rectb.png');
a=rgb2gray(a);
subplot(2,1,1);
imshow(a);
title('org img');
z=a;
[r,c]=size(d);
m=[1 1 1;1 1 1;1 1 1];
for i=2:r-1
for j=2:c-1
new=[(m(1)*z(i-1,j-1)) (m(2)*z(i-1,j)) (m(3)*z(i-1,j+1))
(m(4)*z(i,j-1)) (m(5)*z(i,j)) (m(6)*z(i,j+1))
(m(7)*z(i+1,j-1)) (m(8)*z(i+1,j)) (m(9)*z(i+1,j+1))];
A2(i,j)=min(new);
aaa(i,j)=z(i,j)-A2(i,j);
end
end
subplot(2,1,2);
imshow(aaa);title('Boundary Extracted Image');
```

Output:



Part C:

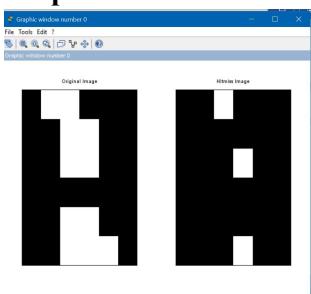
Aim: Program to apply Hit-or-Miss transform

Code:

```
clc;
clear all;
a=[0 1 1 0 0 0;
0 0 1 1 0 0;
0 0 1 1 0 0;
0 0 0 0 0 0;
0 0 1 1 0 0;
0 0 1 1 1 0];
a =im2bw(a,0.5); // image gray scale to image binary ---> below 0.5 --> black, above 0.5 --> white
se = [0 0 0 0;
0 1 1 0;
0 1 1 0;
0 0 0 0];
//se = imcreatese('ellipse',5,5);
S2 = imhitmiss(a,se);

subplot(1,2,1);
imshow(a);
title('Original Image')
subplot(1,2,2);
imshow(S2);
title('Hitmiss Image')
```

Output:



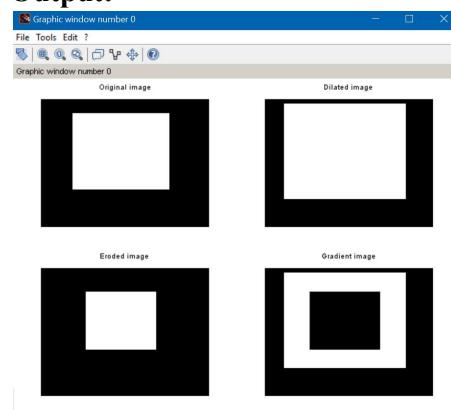
Part D:

Aim: Program to apply morphological gradient on an image

Code:

```
a=imread("D:\MSc IT Image Processing\image\square.png");
se=imcretese('rect',55,55);
dilation=imdilate(a,se);
erosion=imerode(a,se);
gradient=dilation-erosion
subplot(2,2,1);
imshow(a);
title("Original image");
subplot(2,2,2);
imshow(dilation);
title("Dilated image");
subplot(2,2,3);
imshow(erosion);
title("Eroded image");
subplot(2,2,4);
imshow(gradient);
title("Gradient image");
```

Output:



Part E:

Aim: Program to apply Top-Hat/Bottom-hat Transformations

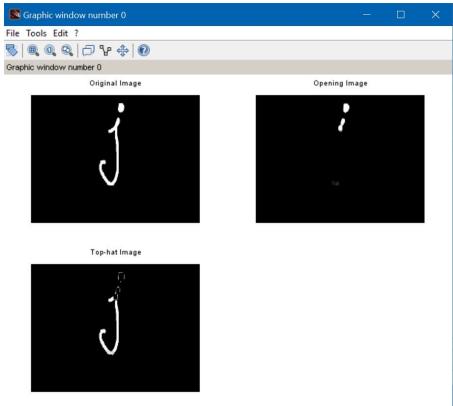
Top-Hat:

Code:

```
a= imread('D:\MSc IT Image Processing\image\letter.png');
se = imcretese('ellipse',10,10);
erosion = imerode(a,se);
opening = imdilate(erosion,se)
Top_Hat= a-opening
subplot(2,2,1);
imshow(a);
title('Original Image');
subplot(2,2,2);
imshow(opening);
title('Opening Image');
```

```
subplot(2,2,3);
imshow(Top_Hat);
title('Top-hat Image');
```

Output:

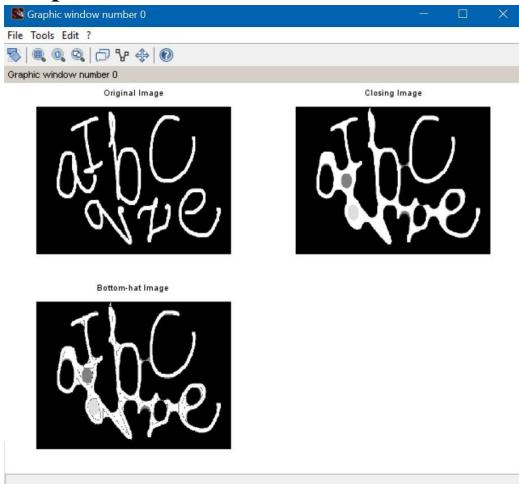


Bottom- Hat:

Code:

```
a= imread('D:\MSc IT Image Processing\image\letters.png');
se = imcreatese('ellipse',30,30);
dilate = imdilate(a,se);
closing = imerode(dilate,se)
bottom_hat=a+closing
subplot(2,2,1);
imshow(a);
title('Original Image');
subplot(2,2,2);
imshow(closing);
title('Closing Image');
subplot(2,2,3);
imshow(bottom_hat);
title('Bottom-hat Image');
```

Output:



Practical No. 8

Image Segmentation

Theory:

Prewitt Operator

The Prewitt operator was developed by Judith M. S. Prewitt. Prewitt operator is used for edge detection in an image. Prewitt operator detects both types of edges, these are:

Horizontal edges or along the x-axis, Vertical Edges or along the y-axis.

Wherever there is a sudden change in pixel intensities, an edge is detected by the mask. Since the edge is defined as the change in pixel intensities, it can be calculated by using differentiation. Prewitt mask is a first-order derivative mask. In the graph representation of Prewitt-mask's result, the edge is represented by the local maxima or local minima.

Sobel Operator

It is named after Irwin Sobel and Gary Feldman. Like the Prewitt operator Sobel operator is also used to detect two kinds of edges in an image: Vertical direction, Horizontal direction

The difference between Sobel and Prewitt Operator is that in Sobel operator the coefficients of masks are adjustable according to our requirement provided they follow all properties of derivative masks.

Canny Operator

Canny edge detector is probably the most commonly used and most effective method, it's much more complex edge detecting method then the ones described above.

Steps:

- Smooth the image with a Gaussian filter to reduce noise.
- Compute gradient of using any of the gradient operators Sobel or Prewitt.
- Extract edge points: Non-maximum suppression.
- Linking and thresholding: Hysteresis

Aim: Program for Edge detection using Sobel, Prewitt and Canny.

Edge Detection using Sobel-

Code-

```
//Marr-Hildreth(Laplassian of Gaussian)
clear all;
im= imread('D:\MSc IT Image Processing\image\balloons.png');
im=im2double(im);
//%smoothening the image with a filter
gfilter=[0 0 1 0 0;
          0 1 2 1 0;
          1 2 -16 2 1;
          0 1 2 1 0;
          0 0 1 0 0];
smim=conv2(im,gfilter)
```

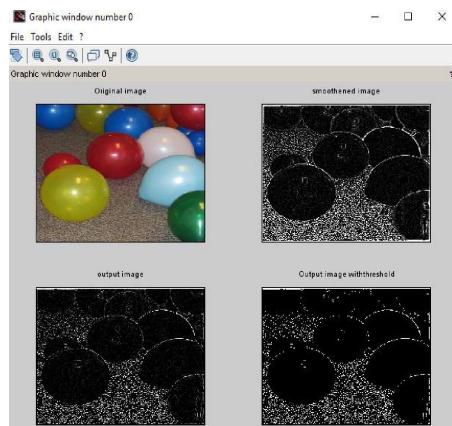
```

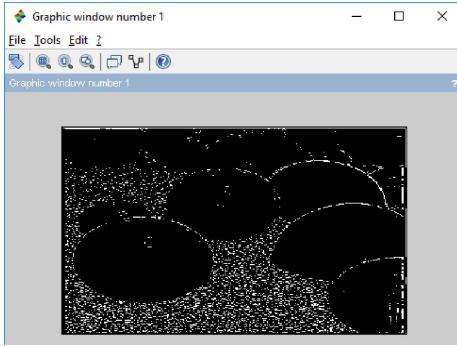
//%finding the zero crossings
[rr,cc]=size(smim);
zc=zeros([rr,cc]);
for i= 2:rr-1
for j=2:cc-1
if(smim(i,j)>0)
if(smim(i,j+1)>=0 && smim(i,j-1)<0) || (smim(i,j+1)<0 && smim(i,j-1)>=0)
zc(i,j)=smim(i,j+1);
elseif(smim(i+1,j)>=0 && smim(i-1,j)<0) || (smim(i+1,j)<0 && smim(i-1,j)>=0)
zc(i,j)=smim(i,j+1);
elseif(smim(i+1,j+1)>=0 && smim(i-1,j-1)<0) || (smim(i+1,j+1)<0 && smim(i-1,j-1)>=0)
zc(i,j)=smim(i,j+1);
elseif(smim(i-1,j+1)>=0 && smim(i+1,j-1)<0) || (smim(i-1,j+1)<0 && smim(i+1,j-1)>=0)
zc(i,j)=smim(i,j+1);
end
end
end
end
otpt=im2uint8(zc);
//%thresholding
otptth = otpt>105;
figure;
subplot(2,2,1);imshow(im);title('Original image');
subplot(2,2,2);imshow(smim);title('smoothened image');

subplot(2,2,3);imshow(otpt);title('output image');
subplot(2,2,4);imshow(otptth);title('Output image withthreshold');
//%final result
figure,imshow(otptth);

```

Output:





Edge Detection using Prewitt-

Code-

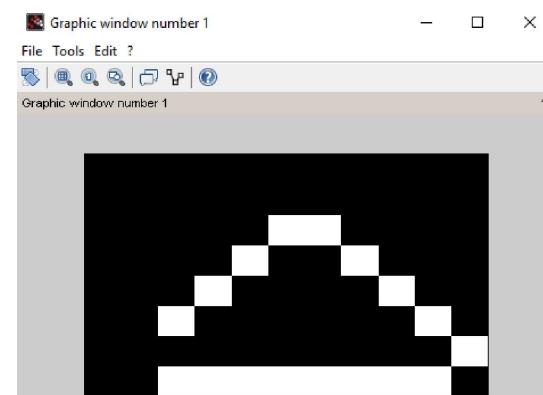
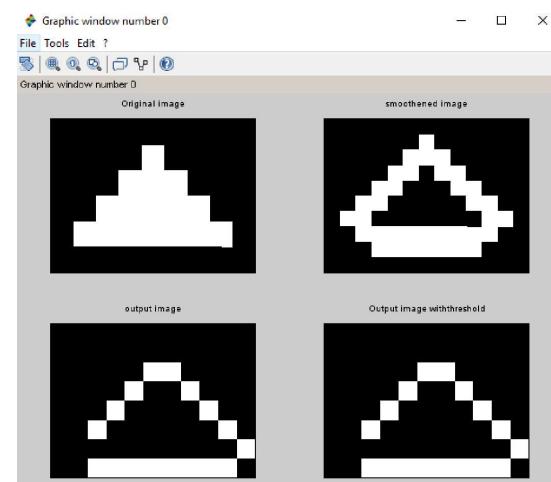
```
//Marr-Hildreth(Laplassian of Gaussian)
clc;
clear all;
//im= imread(' D:\MSc IT Image Processing\image\balloons.png');
im = [0 0 0 0 0 0 0 0;
0 0 0 1 0 0 0 0;
0 0 0 1 1 1 0 0 0;
0 0 1 1 1 1 1 0 0;
0 1 1 1 1 1 1 1 0;
0 0 0 0 0 0 0 0 0];
im=im2double(im);
//%smoothening the image with a filter
gfilter=[0 0 1 0 0;
0 1 2 1 0;
1 2 -16 2 1;
0 1 2 1 0;
0 0 1 0 0];
smim=conv2(im,gfilter)
disp(smim)
//%finding the zero crossings
[rr,cc]=size(smim);
zc=zeros([rr,cc]);
disp([rr,cc])
for i= 2:rr-1
for j=2:cc-1
if(smim(i,j)>0)
if(smim(i,j+1)>=0 && smim(i,j-1)<0) || (smim(i,j+1)<0 && smim(i,j-1)>=0)
zc(i,j)=smim(i,j+1);
elseif(smim(i+1,j)>=0 && smim(i-1,j)<0) || (smim(i+1,j)<0 && smim(i-1,j)>=0)
zc(i,j)=smim(i,j+1);
elseif(smim(i+1,j+1)>=0 && smim(i-1,j-1)<0) || (smim(i+1,j+1)<0 && smim(i-1,j-1)>=0)
zc(i,j)=smim(i,j+1);
elseif(smim(i-1,j+1)>=0 && smim(i+1,j-1)<0) || (smim(i-1,j+1)<0 && smim(i+1,j-1)>=0)
zc(i,j)=smim(i,j+1);
end
end
```

```

end
end
otpt=im2uint8(zc);
//%tresholding
otptth = otpt>105;
figure;
subplot(2,2,1);imshow(im);title('Original image');
subplot(2,2,2);imshow(smim);title('smoothened image');
subplot(2,2,3);imshow(otpt);title('output image');
subplot(2,2,4);imshow(otptth);title('Output image withthreshold');
//%final result
figure,imshow(otptth);

```

Output:



All 3 together:

Code:

```

1 clc;
2 image=imread("C:\Program Files\scilab-6.1.1\IPCV\images\checkerbox.png");
3 image = rgb2gray(image);
4 edge1 = edge(image,'sobel');
5 edge2 = edge(image,'prewitt');
6 edge3 = edge(image,'canny');
7 subplot(2,2,1);
8 imshow(image);
9 title('Original Image');
10 subplot(2,2,2);
11 imshow(edge1);
12 title('sobel Image');
13 subplot(2,2,3);
14 imshow(edge2);
15 title('prewitt Image');
16 subplot(2,2,4);
17 imshow(edge3);
18 title('canny Image');
19

```

Output:

