Tribhuvan University
Institute of Science and Technology
Patan Multiple Campus

LAB REPORT ON
CLOUD COMPUTING (BIT408)

A partial fullfilment of the requirements for Bachelors in Information Technology(BIT)

Submitted To:
Sachita Nand Mishra
Lecturer, Patan Multiple Campus

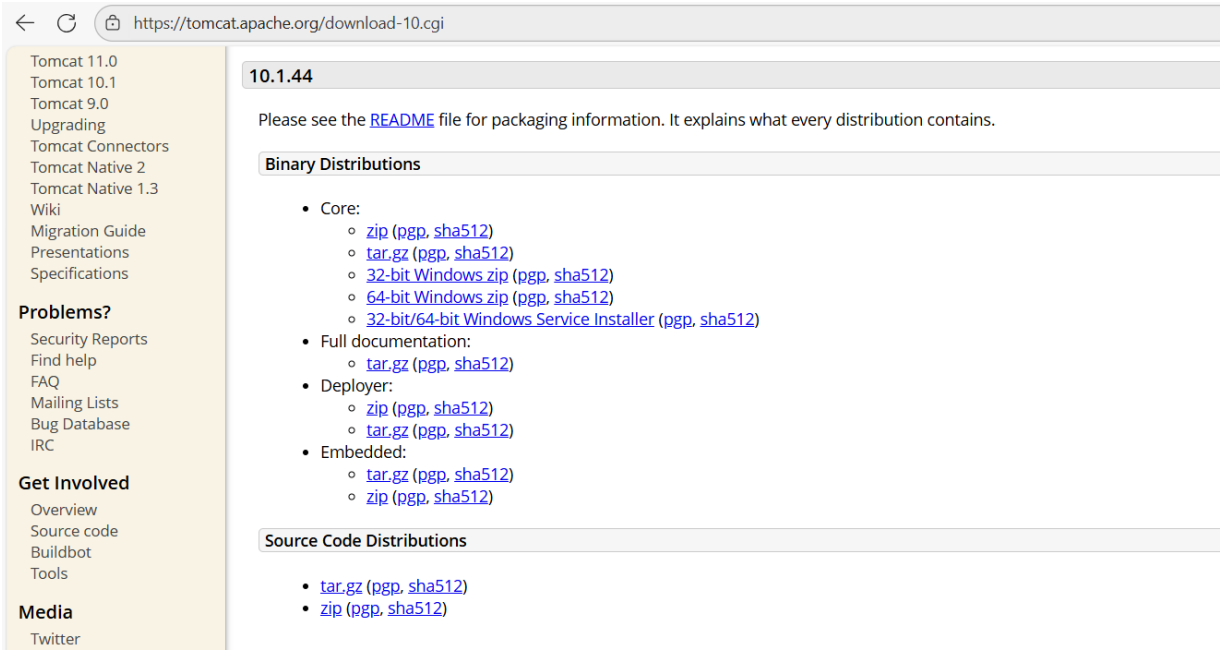
Submitted By:
Suprabha Aryal
BIT 7th Semester
Exam Roll No. : 604/078
Class Roll No. :01/078

List of Lab Works

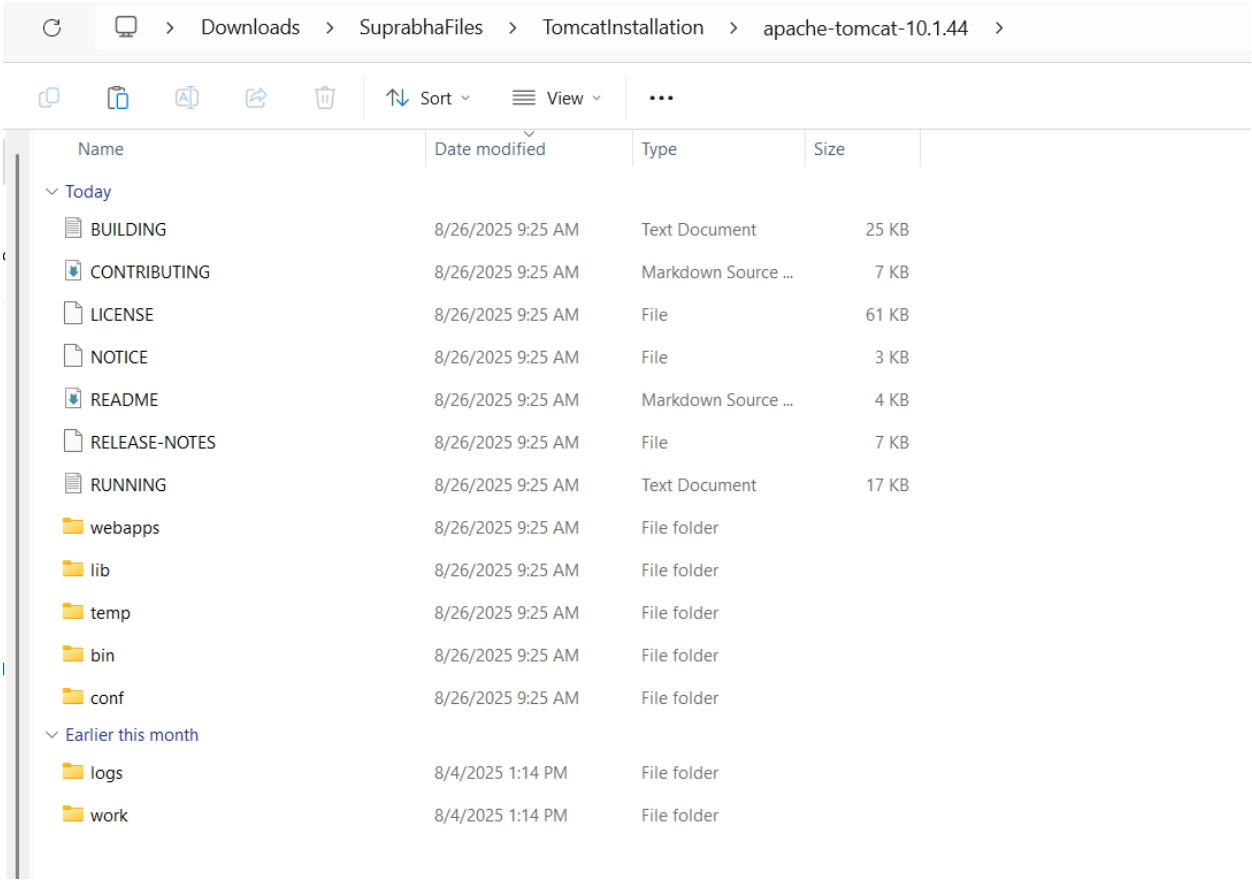
S.N.	Title	Date	Signature
1.	Installation and Testing of tomcat Server on Java IDE		
2.	SOAP Web Service in Java (JAX-WS)		
3.	Consuming SOAP Web services in java		
4.	Implement Windows Hyper V virtualization		
5.	Implement Virtualization using VirtualBox		
6.	Simulate a cloud scenario using CloudSim		
7.	Installation and testing of Hadoop single node cluster on windows		
8.	Map reduce word count program using java		
9.	Develop and application for Google App Engine		

Lab-1 Installation and testing of Apache Tomcat in Java IDE

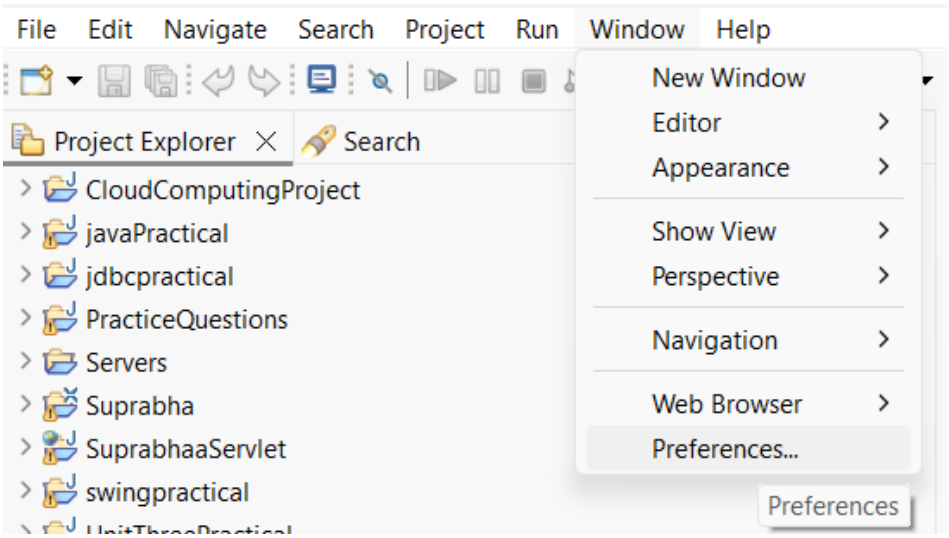
Step 1: Download Tomcat from https://tomcat.apache.org/download-10.cgi :



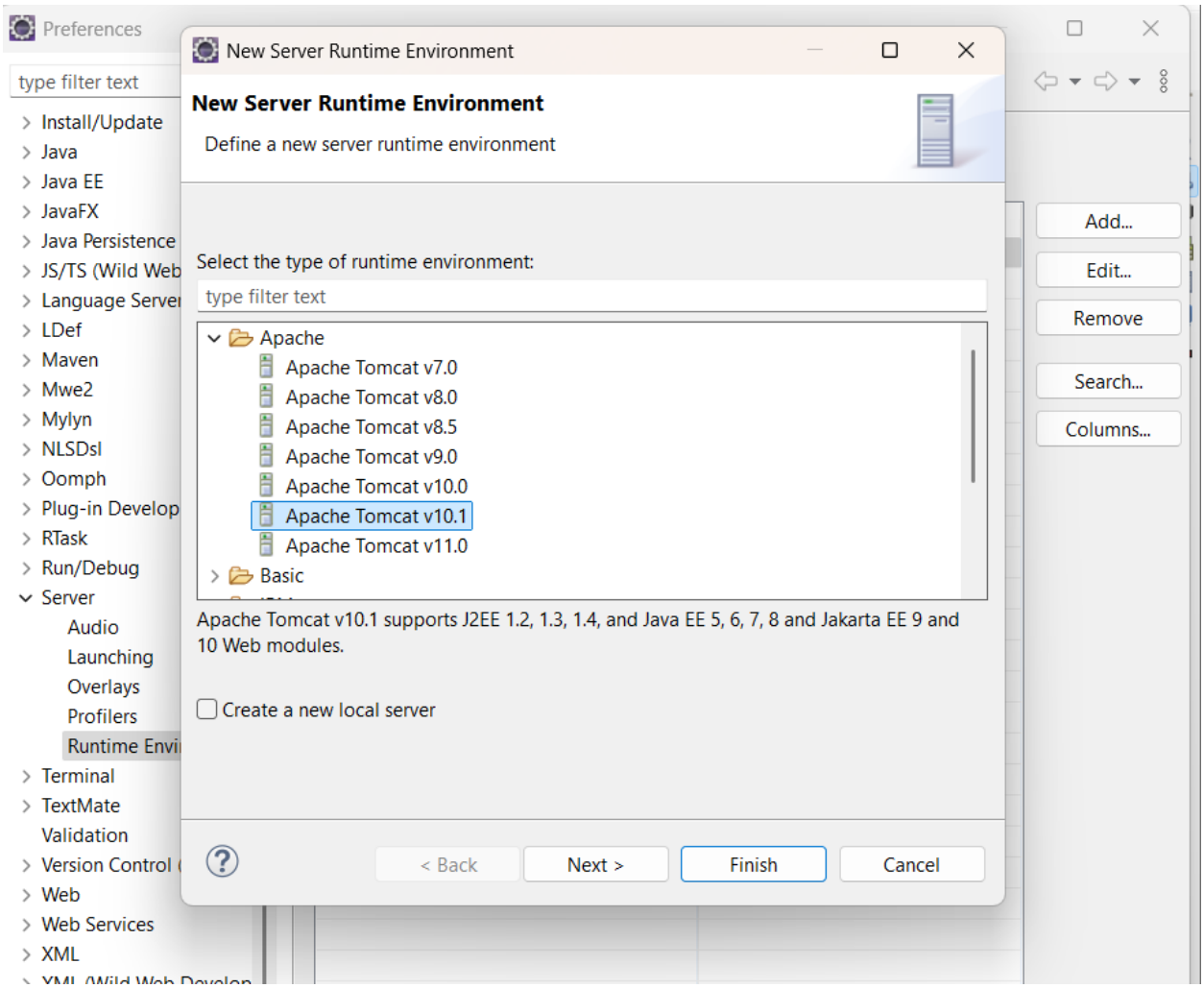
Step: Extract the downloaded tomcat in your preferred folder:



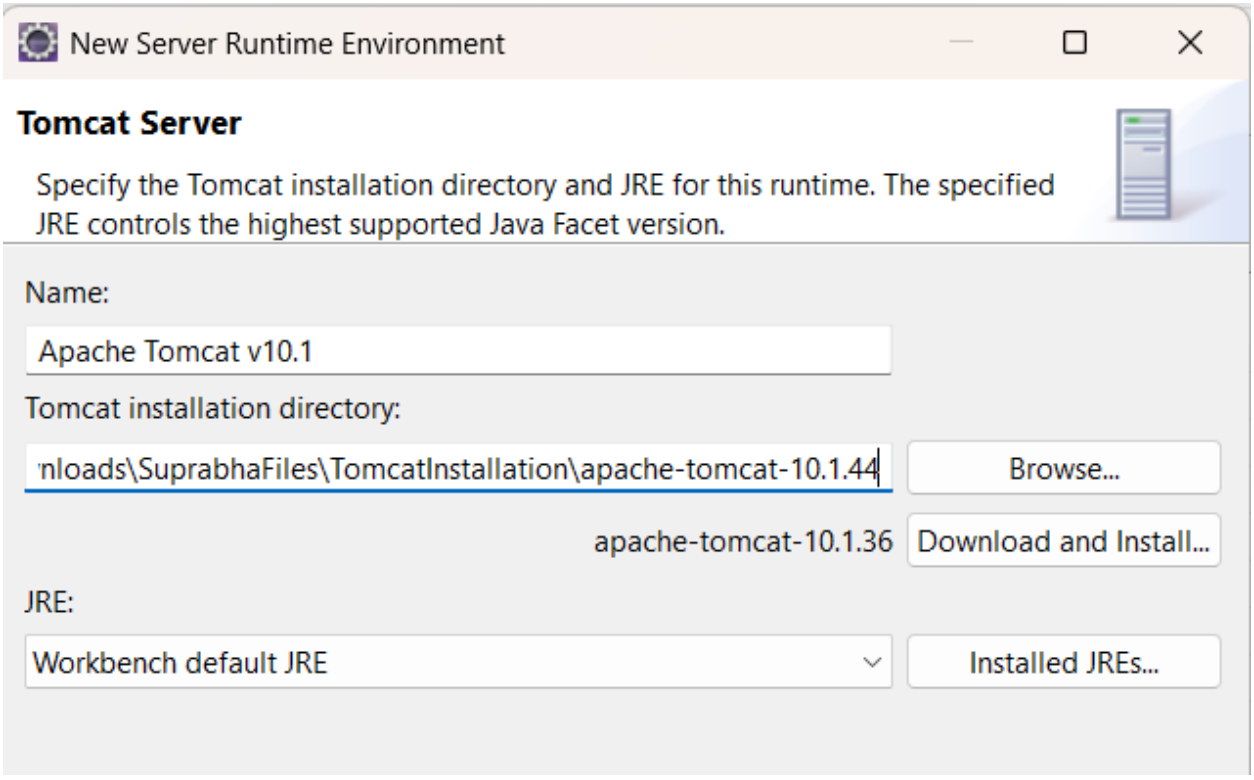
Step 3: Add Tomcat to the IDE. Here, I'm using Eclipse . First, Open the Window> Preferences:



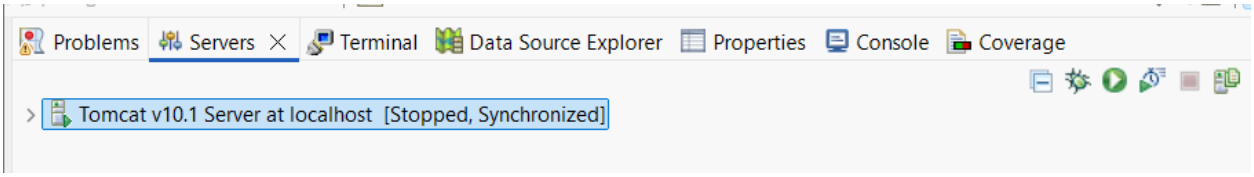
Step 4: After going to Preferences, Select the type of runtime environment:



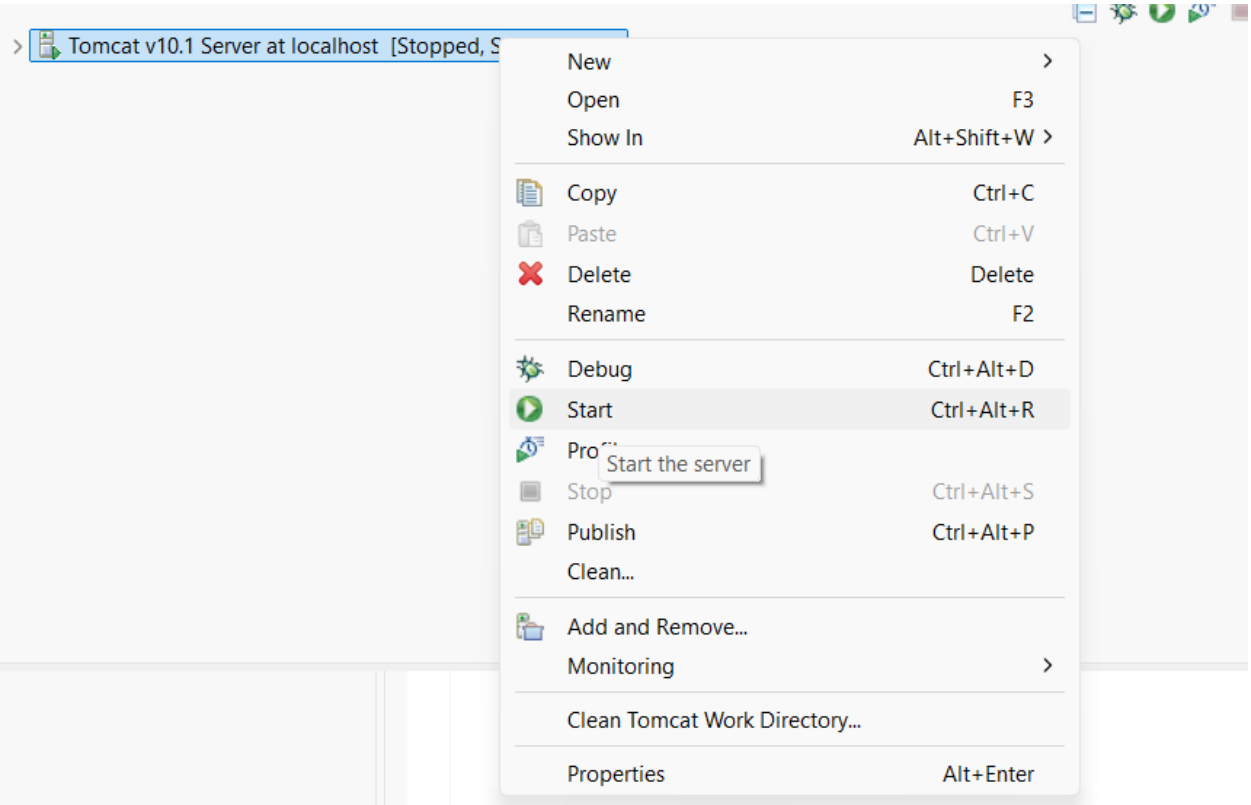
Step 5: And then click next and browse to the directory where you have extracted the apache tomcat and click finish.

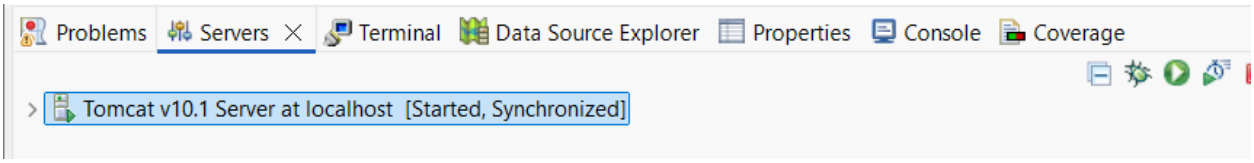


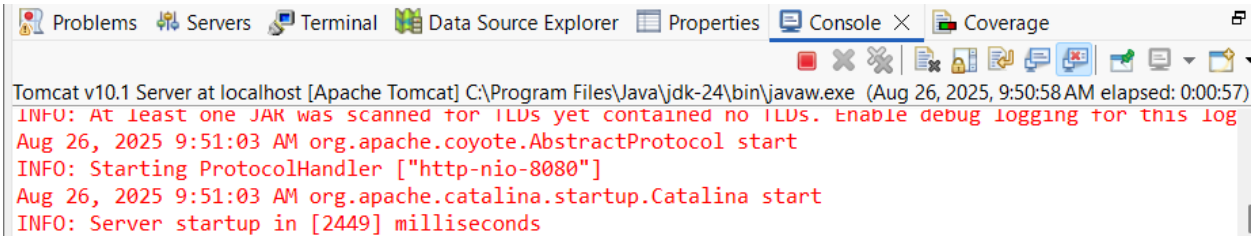
The server is stopped,



Step 6: Start the Server:



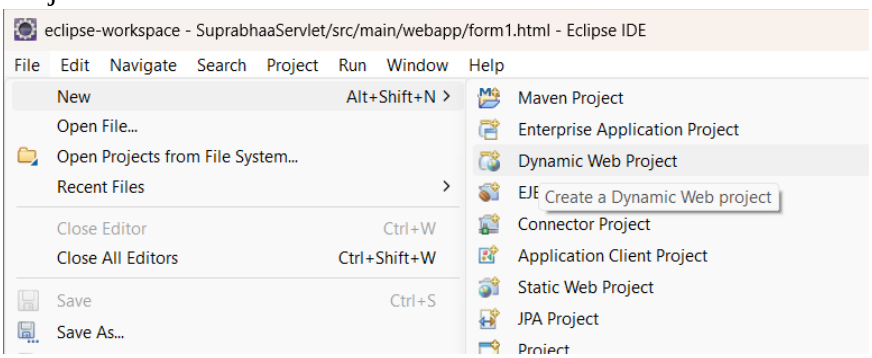




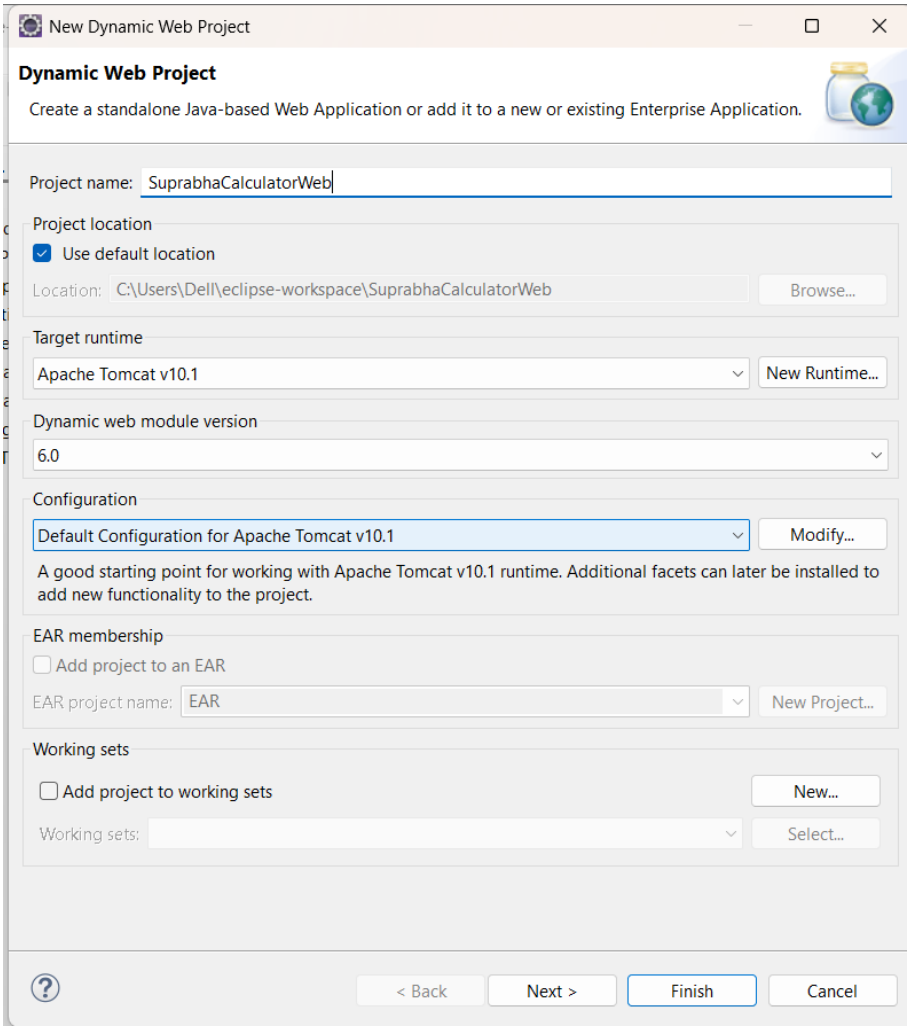
Lab-2 SOAP Web Service in Java (JAX-WS)

Develop a simple SOAP based Web Service in Java using JAX-WS, called as "CalculatorService".

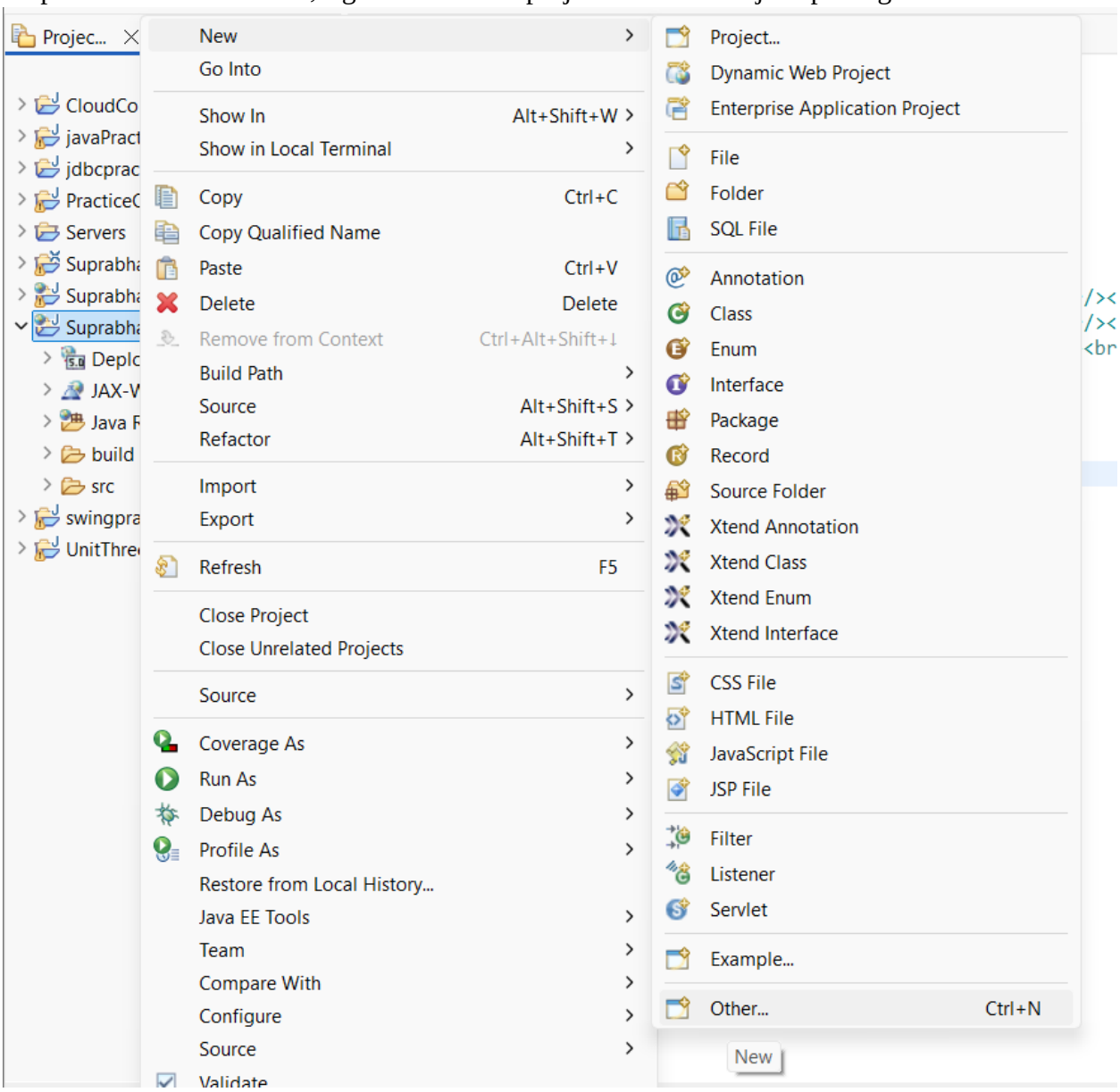
Step 1: Create a new Java Web application project. Go to, File > New > Dynamic Web Project:

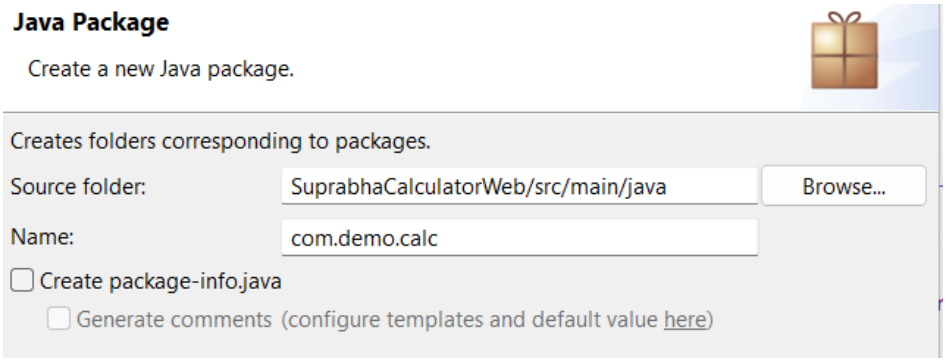


Enter the project name and then click finish:



Step 2: Add a web service , right click on the project and create a java package:



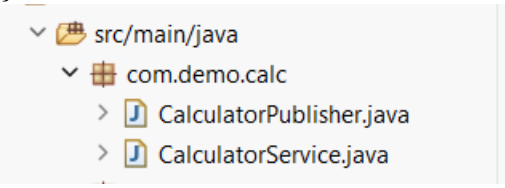


Create a CalculatorService.java class with

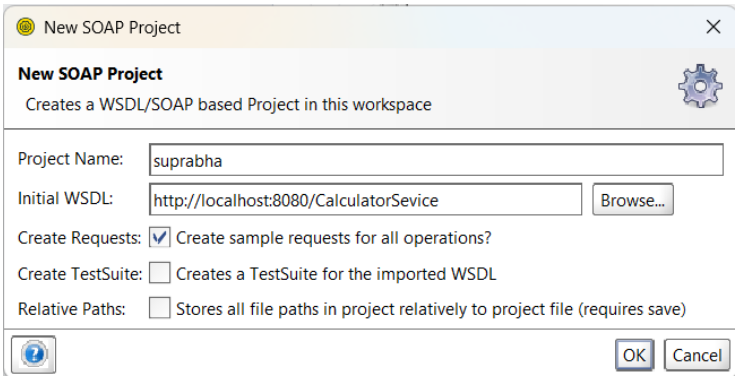
```
package com.demo.calc;

import javax.jws.WebMethod;
import javax.jws.WebService;

@WebService
public class CalculatorService {

    @WebMethod
    public int add(int a, int b) {
        return a + b;
    }

    @WebMethod
    public int subtract(int a, int b) {
        return a - b;
    }

    @WebMethod
    public int multiply(int a, int b) {
        return a * b;
    }

    @WebMethod
    public double divide(int a, int b) {
        if (b == 0) throw new IllegalArgumentException("Division by zero!");
        return (double) a / b;
    }
}
```

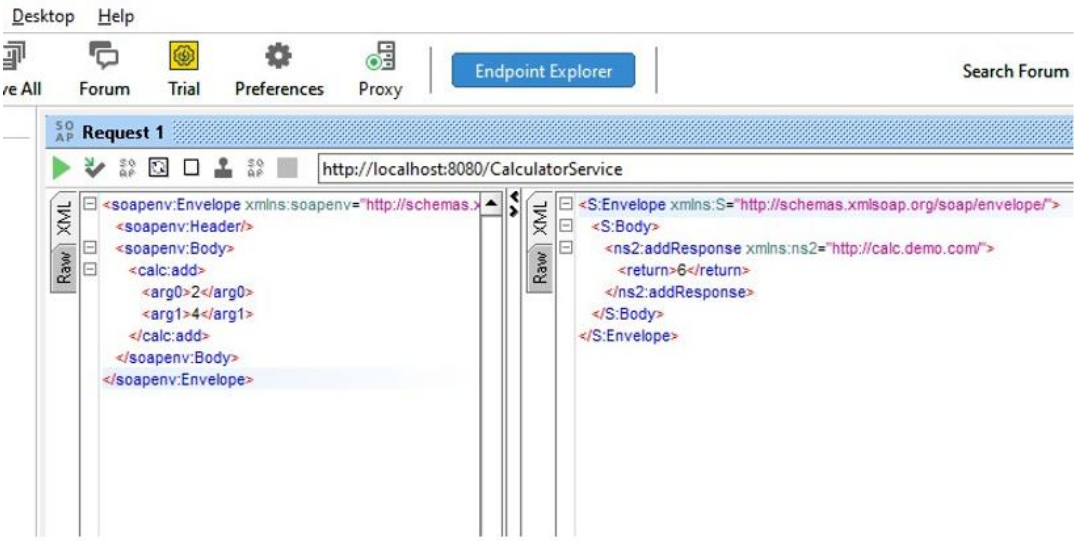
Create CaclulatorPublisher.java class with

```
package com.demo.calc;

import javax.xml.ws.Endpoint;

public class CalculatorPublisher {
   public static void main(String[] args) {
      String url = "http://localhost:8080/CalculatorService";
      Endpoint.publish(url, new CalculatorService());
      System.out.println("Service published at: " + url + "?wsdl");
   }
}
```



Using SOAP UI to test this file:

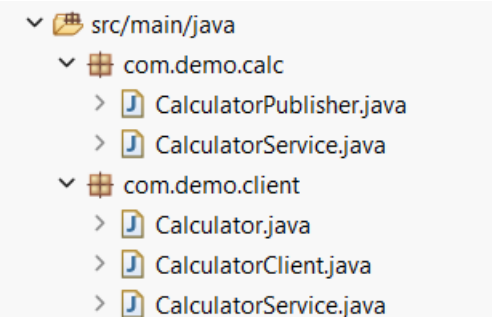




Lab-3 Consuming Soap Web Services in java

Client to make user of web services add method

Step 1: create a new package named client and create following interfaces and classes:



With the code as follows

```
package com.demo.client;
public class CalculatorClient
{

   public static void main(String[] args) {

      CalculatorService service = new CalculatorService(null);

      Calculator port = service.getCalculatorPort();


      int result = port.add(7, 5);

      System.out.println("Result from CalculatorService: " + result);

   }
}
```

Create Calculator interface with:

```
package com.demo.client;

import javax.jws.WebMethod;
import javax.jws.WebService;
importjavax.jws.soap.SOAPBin
ding;

@WebService(targetNamespace = "http://calc.demo.com/")

@SOAPBinding(style = SOAPBinding.Style.DOCUMENT, use =
SOAPBinding.Use.LITERAL) public interface Calculator {

   @WebMethod

   int add(int a, int b);
```

}

And, finally create a calculatorservice class extending service as such: package com.demo.client;

import javax.xml.namespace.QName; import javax.xml.ws.Service; import java.net.URL; public class CalculatorService extends Service {

private static final QName SERVICE_NAME = new QName("http://calc.demo.com/",

"CalculatorService");
public CalculatorService(URL wsdlDocumentLocation) {

super(wsdlDocumentLocation, SERVICE_NAME);

}

public Calculator getCalculatorPort() {

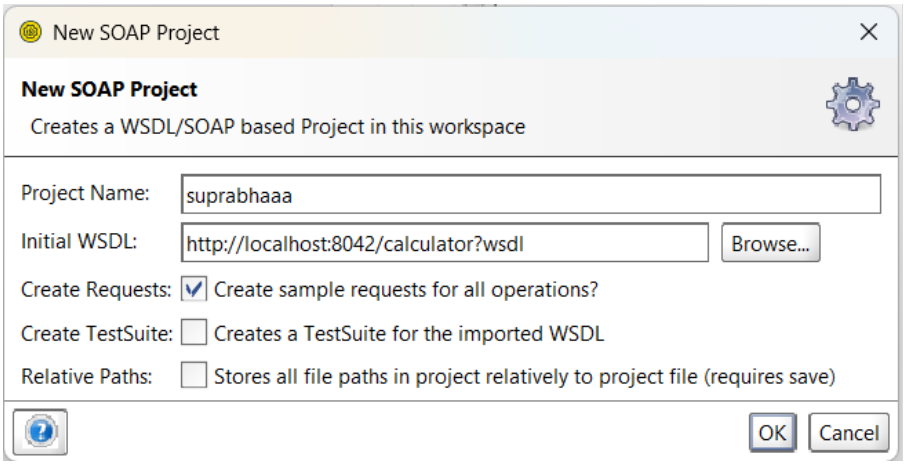
return super.getPort(new QName("http://calc.demo.com/", "CalculatorServicePort"),

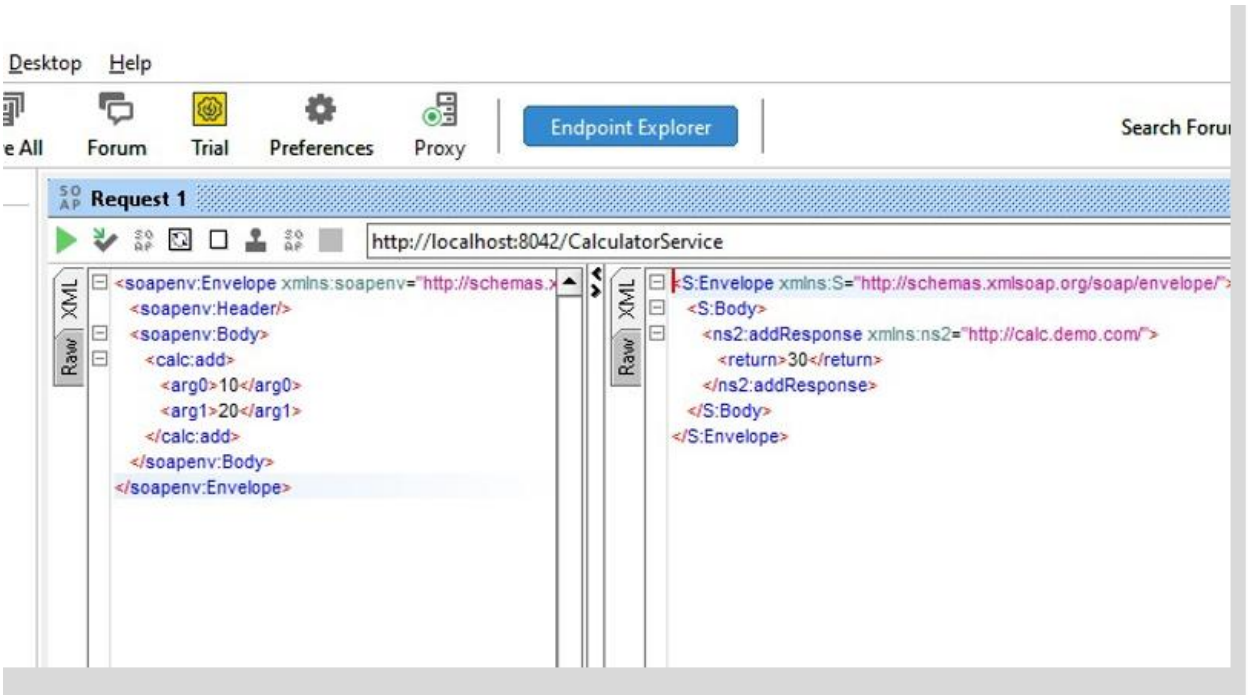
Calculator.class);

}

}

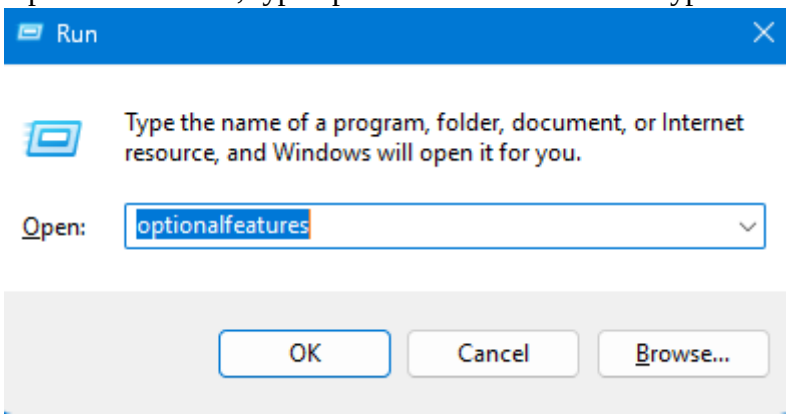
Compile the client and test it using SOAP UI



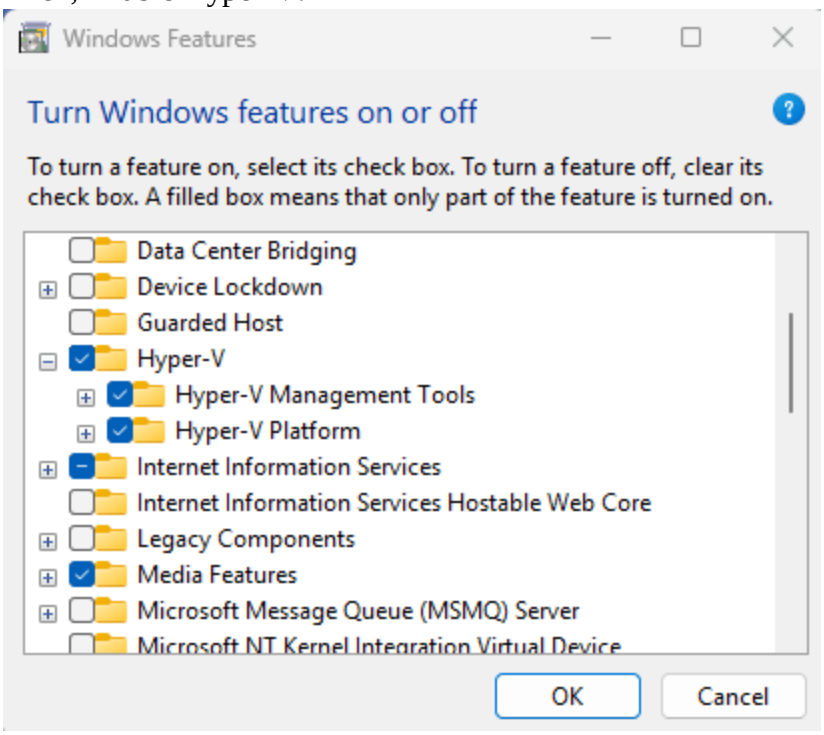

Desktop
Help
e All
Forum
Trial
Preferences
Proxy
Endpoint Explorer
Search Foru
Request 1
http://localhost:8042/CalculatorService
XML
Raw
<soapenv:Envelope xmlns:soapenv="http://schemas.x
<soapenv:Header/>
<soapenv:Body>
<calc:add>
<arg0>10</arg0>
<arg1>20</arg1>
</calc:add>
</soapenv:Body>
</soapenv:Envelope>
XML
Raw
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:addResponse xmlns:ns2="http://calc.demo.com/">
<return>30</return>
</ns2:addResponse>
</S:Body>
</S:Envelope>

Lab-4 Implement Windows Hyper V virtualization

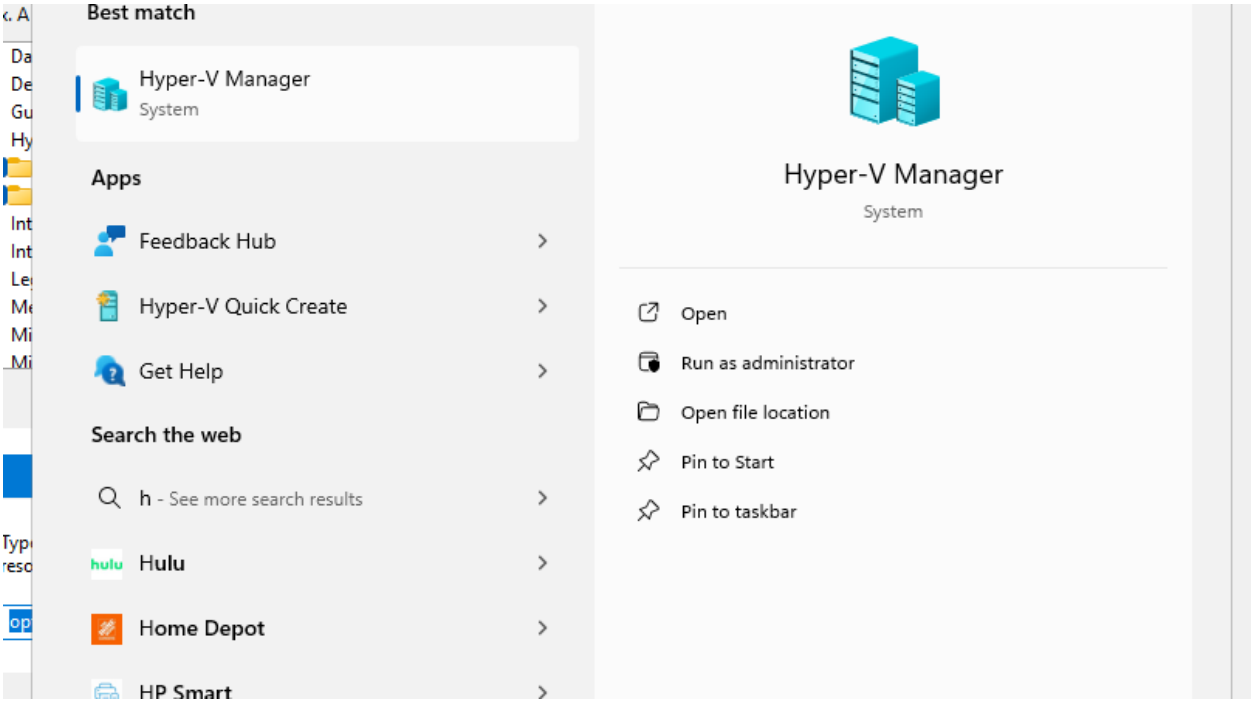
Open windows+R, type optinal features to enable Hyper-V:

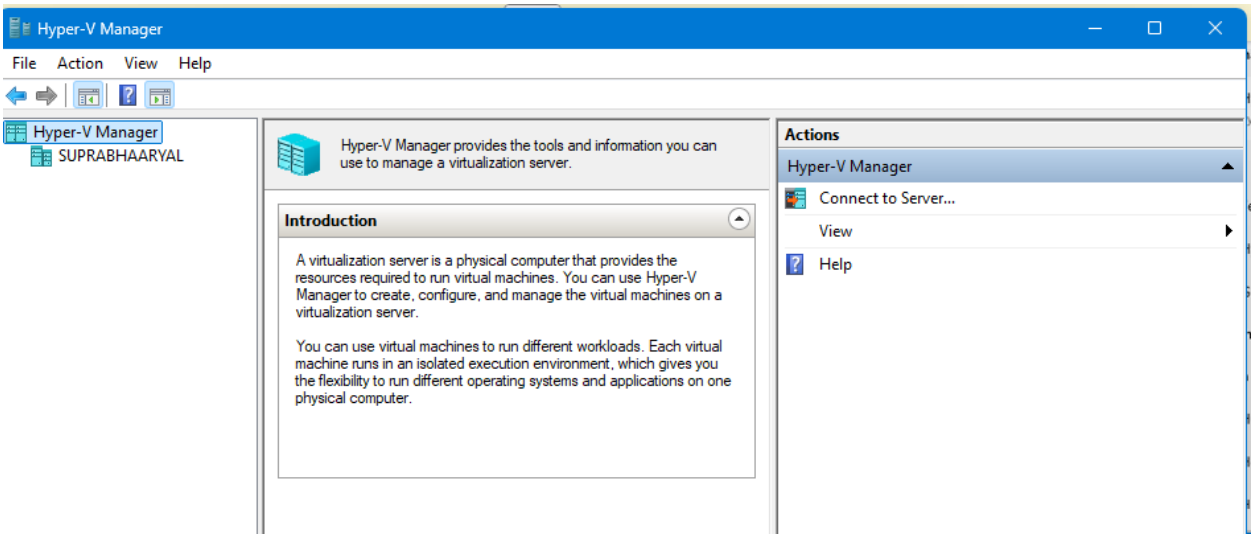


Then, Enable Hyper-V:

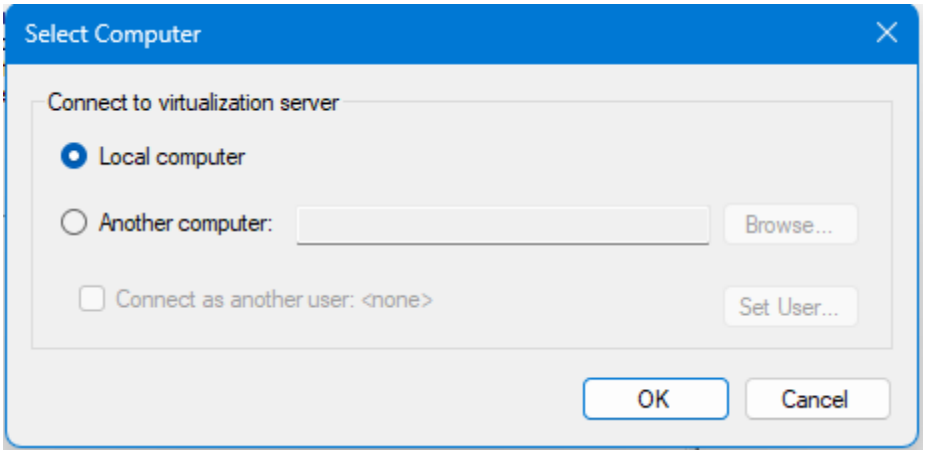


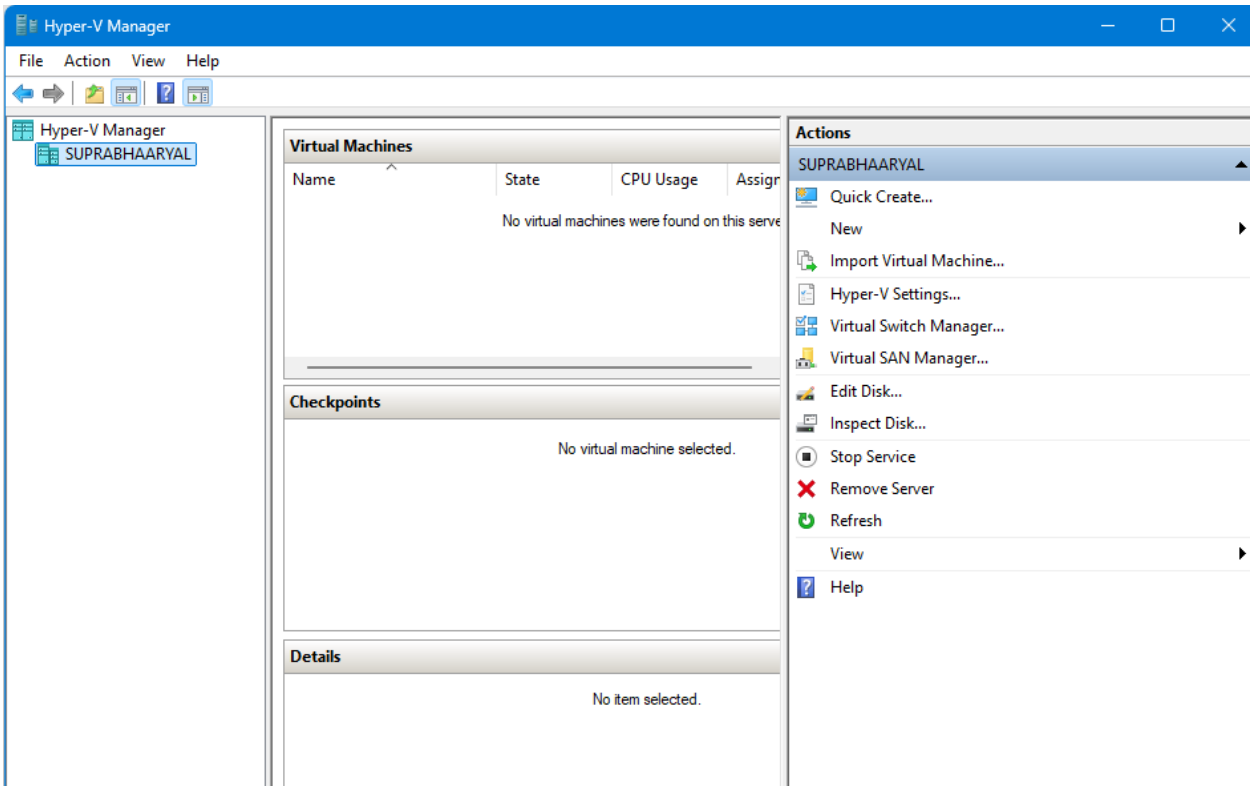
Launch Hyper-V Manager:



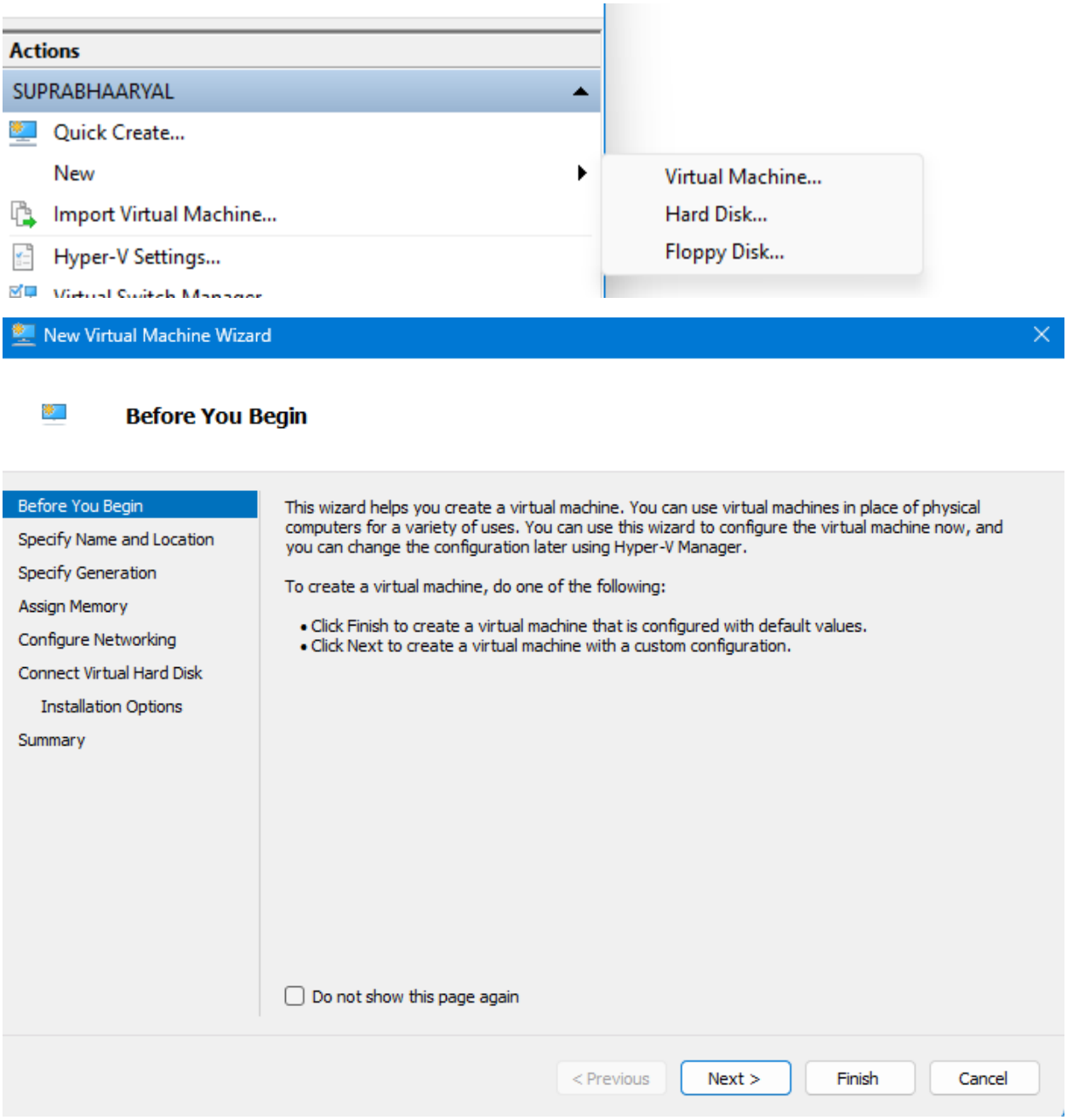


Connect to virtualization server as a Local computer:

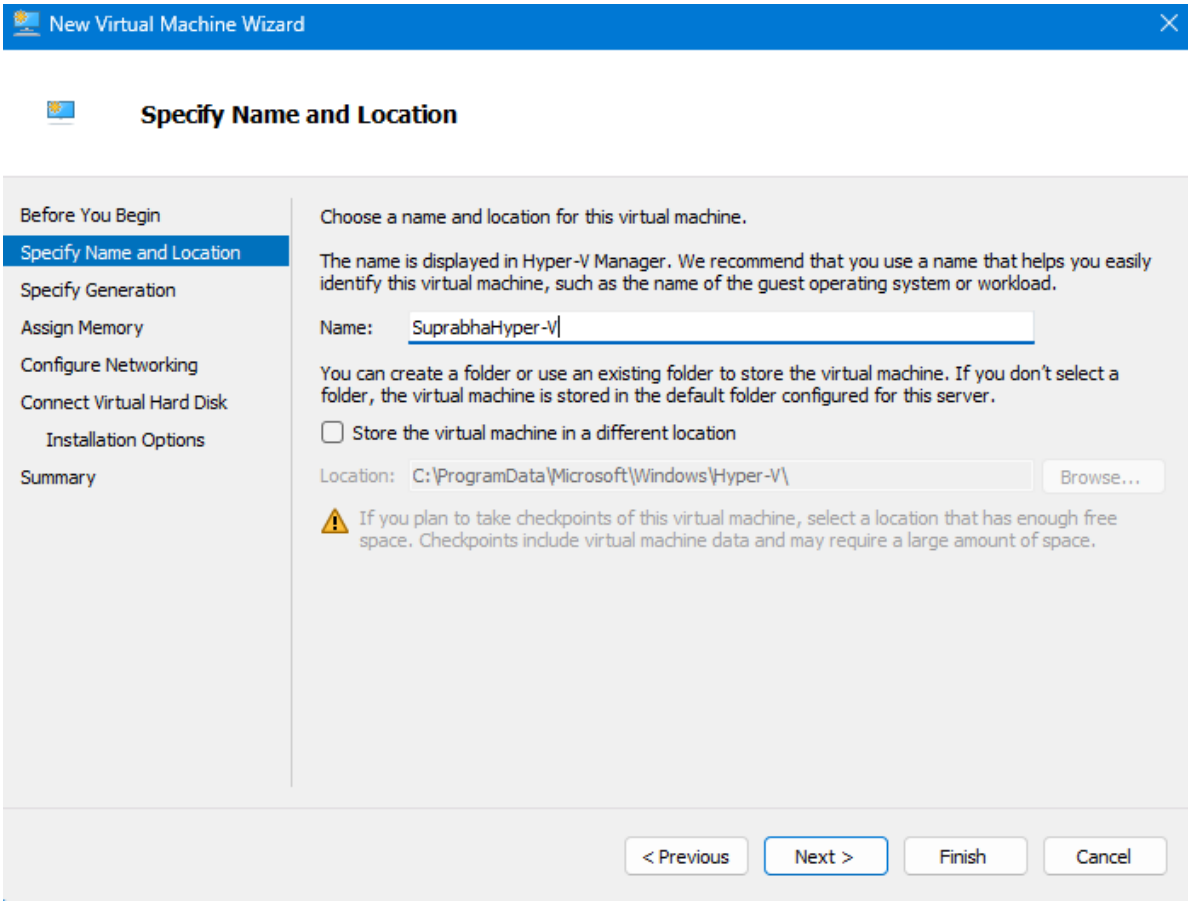




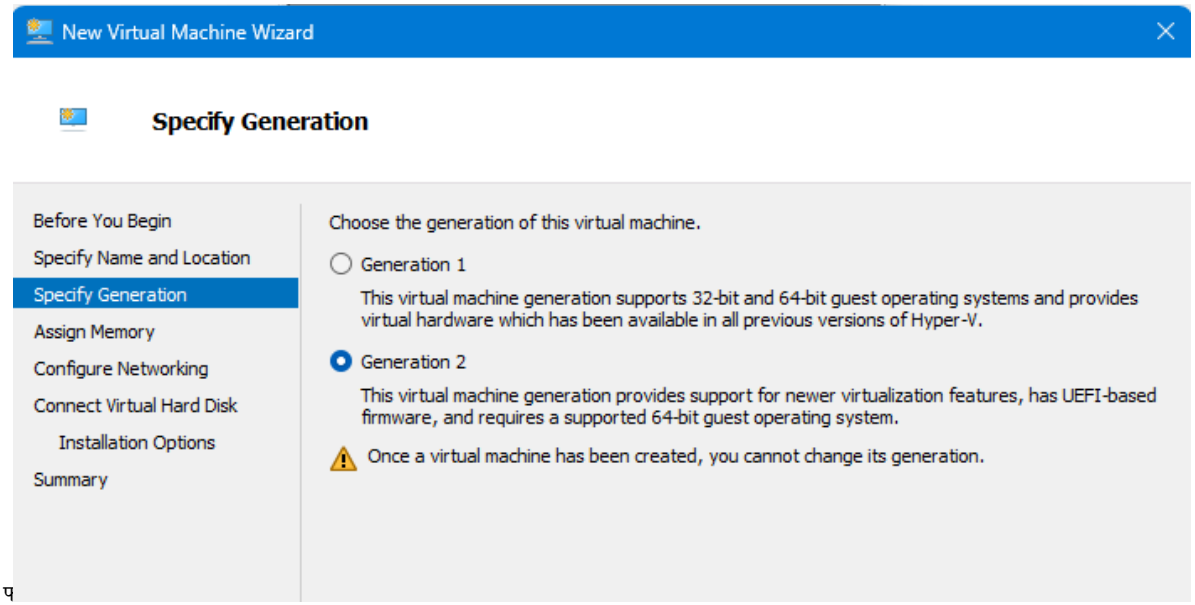
Create a new Vitual Machine:



Specify Name and Location:

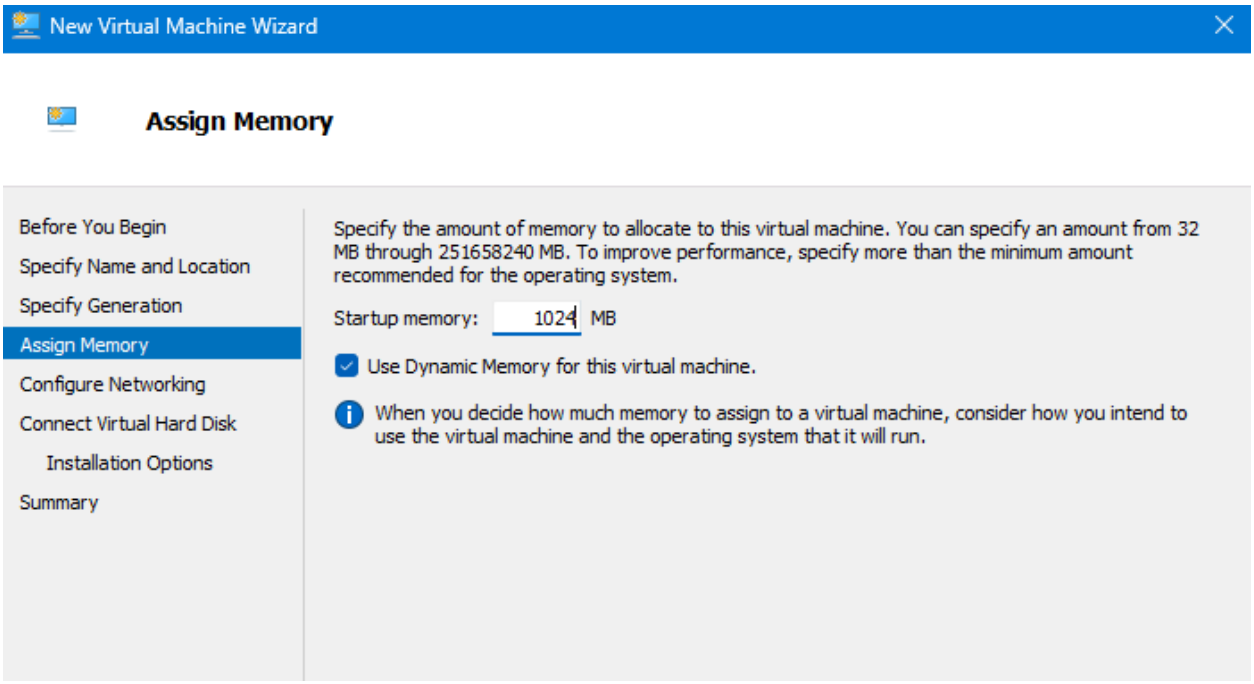


Specify Generation:

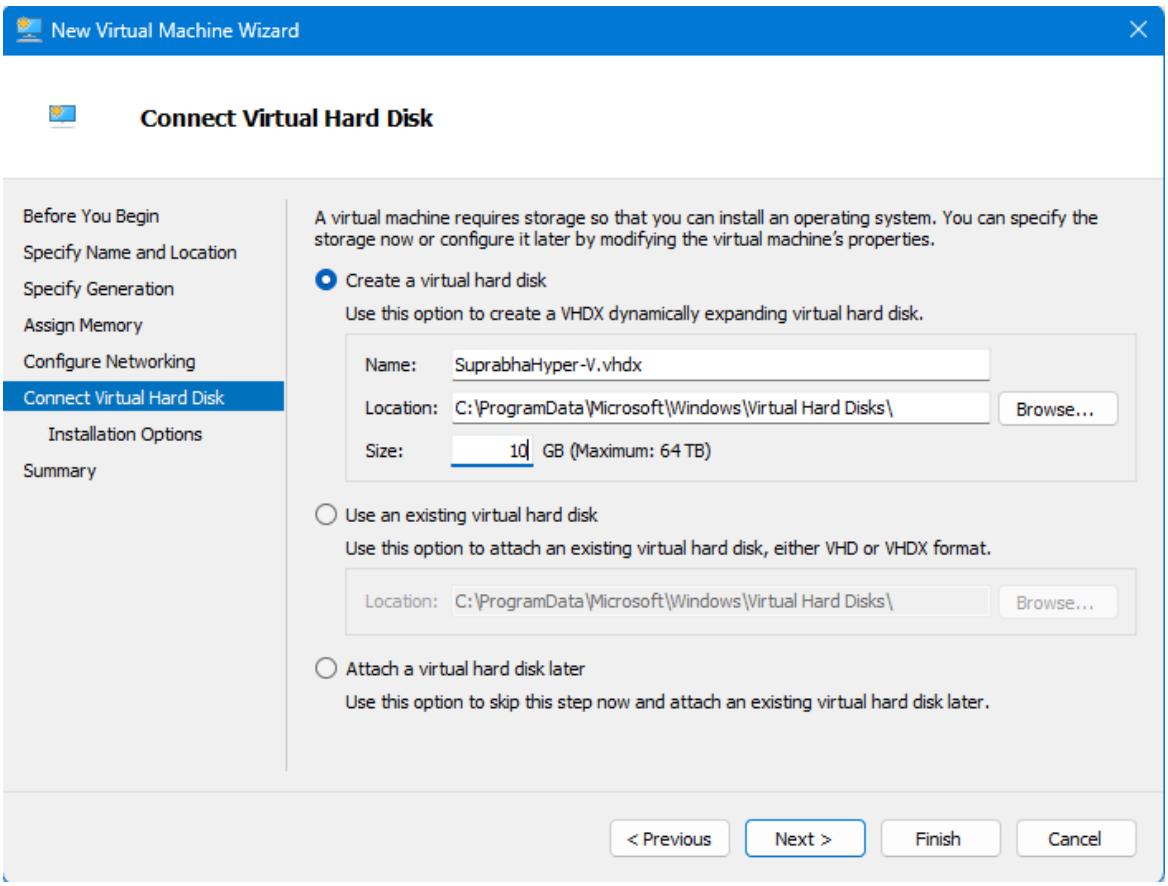


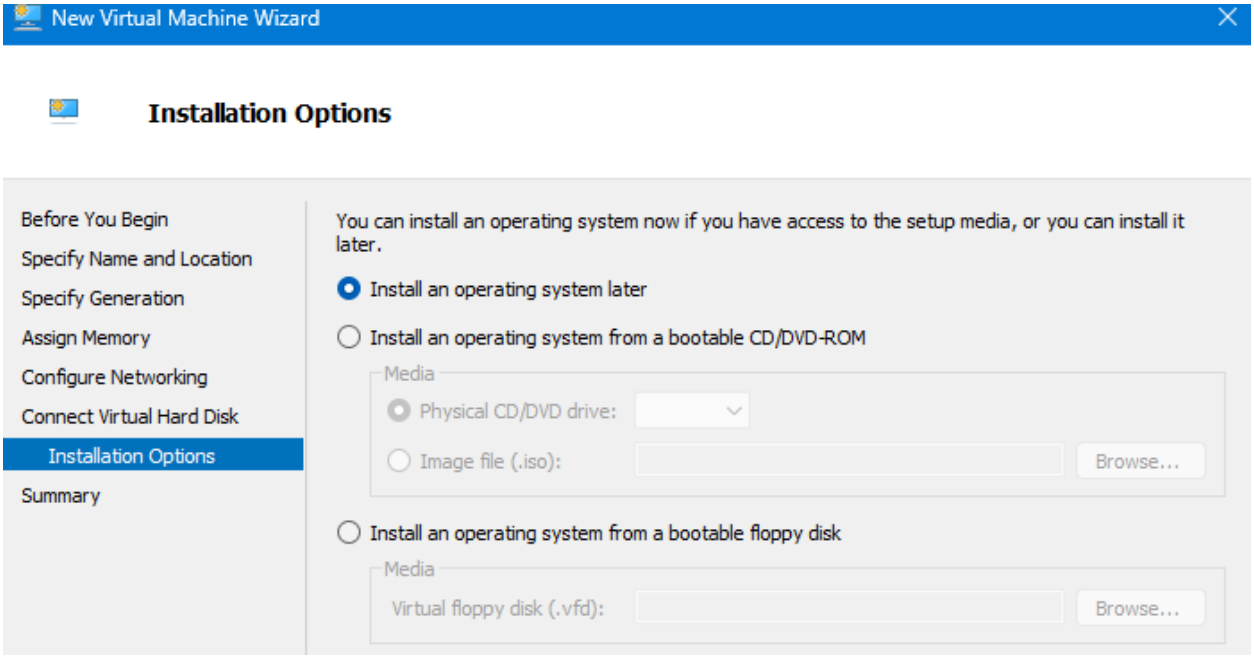
ף

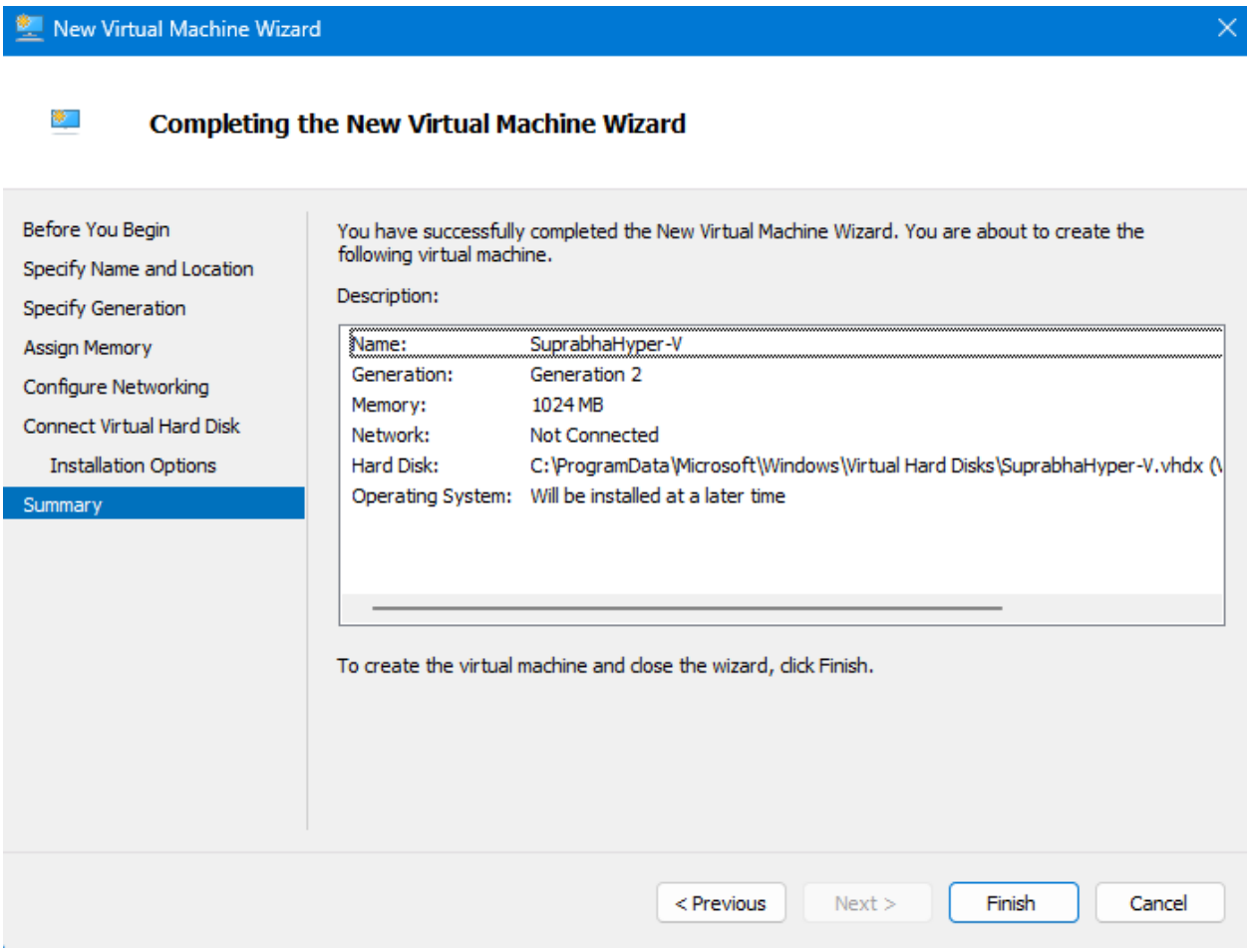
Assign memory :



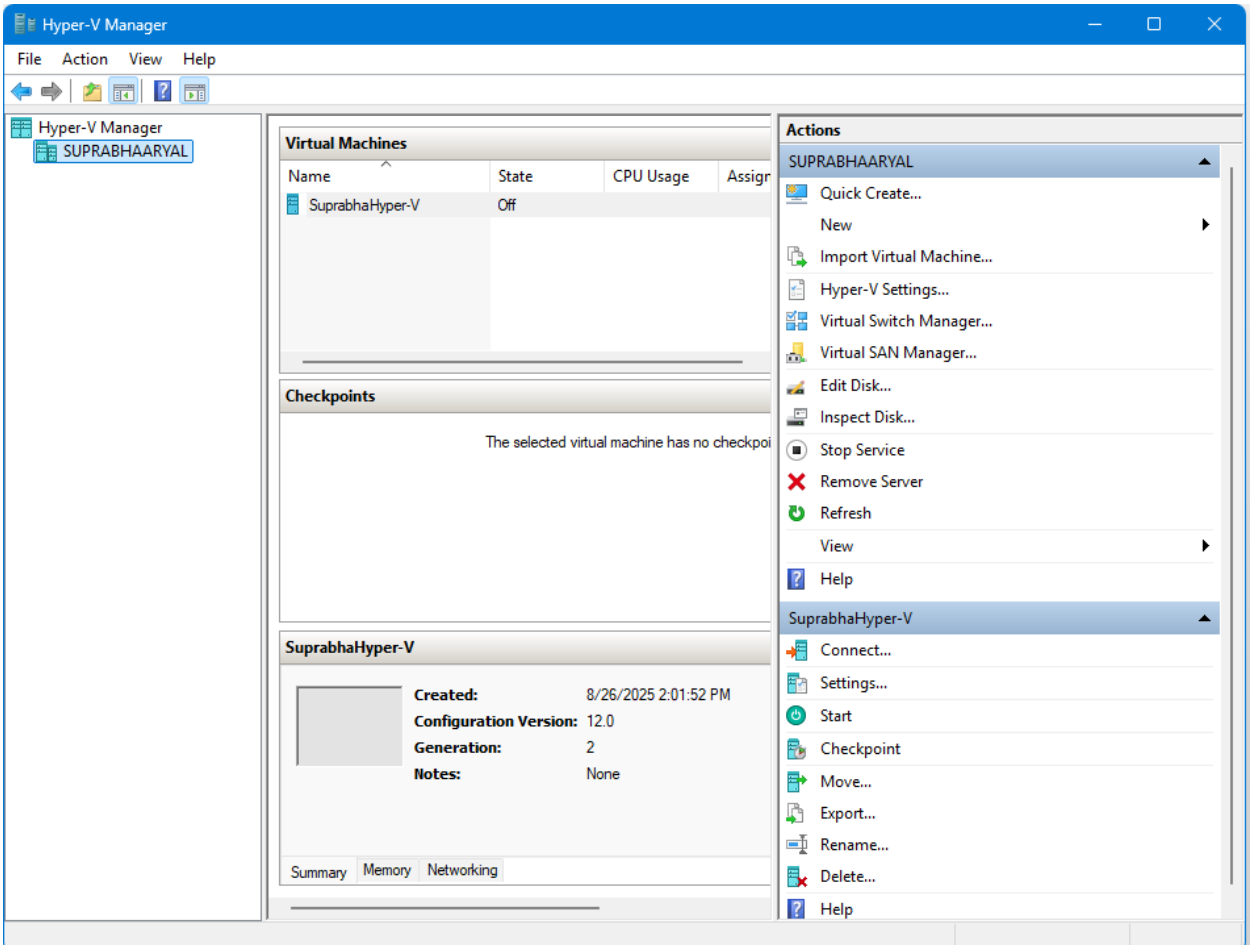
Setup harddisk:




New Virtual Machine Wizard
Installation Options
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
You can install an operating system now if you have access to the setup media, or you can install it later.
Install an operating system later
Install an operating system from a bootable CD/DVD-ROM
Media
Physical CD/DVD drive:
Image file (.iso):
Browse...
Install an operating system from a bootable floppy disk
Media
Virtual floppy disk (.vfd):
Browse...

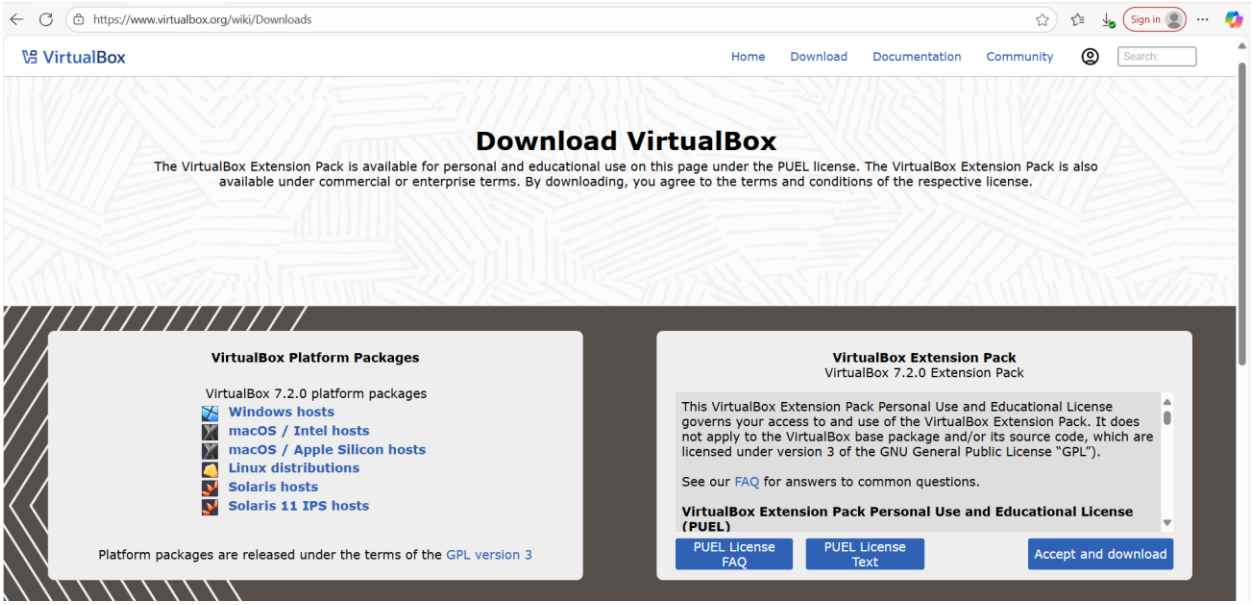

New Virtual Machine Wizard
Completing the New Virtual Machine Wizard
Before You Begin
Specify Name and Location
Specify Generation
Assign Memory
Configure Networking
Connect Virtual Hard Disk
Installation Options
Summary
You have successfully completed the New Virtual Machine Wizard. You are about to create the following virtual machine.
Description:
Name: SuprabhaHyper-V
Generation: Generation 2
Memory: 1024 MB
Network: Not Connected
Hard Disk: C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\SuprabhaHyper-V.vhdx (\
Operating System: Will be installed at a later time
To create the virtual machine and close the wizard, click Finish.
< Previous
Next >
Finish
Cancel

The Setup is completed , Now you can connect to the Virtual Machine:

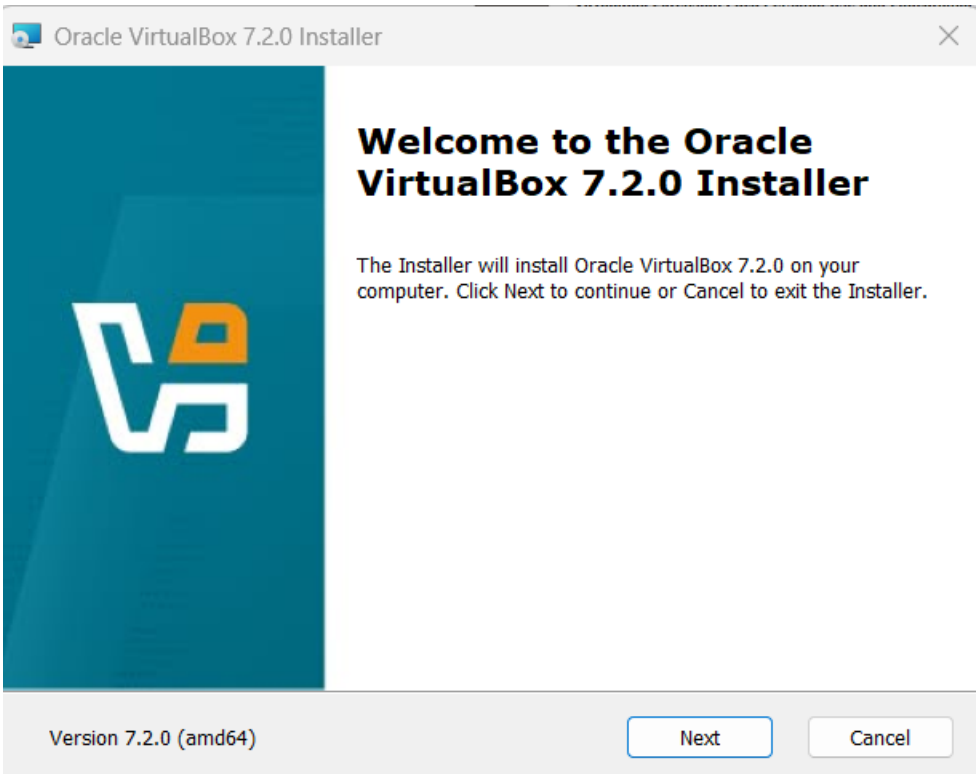


Lab-5 Implement Virtualization using VirtualBox

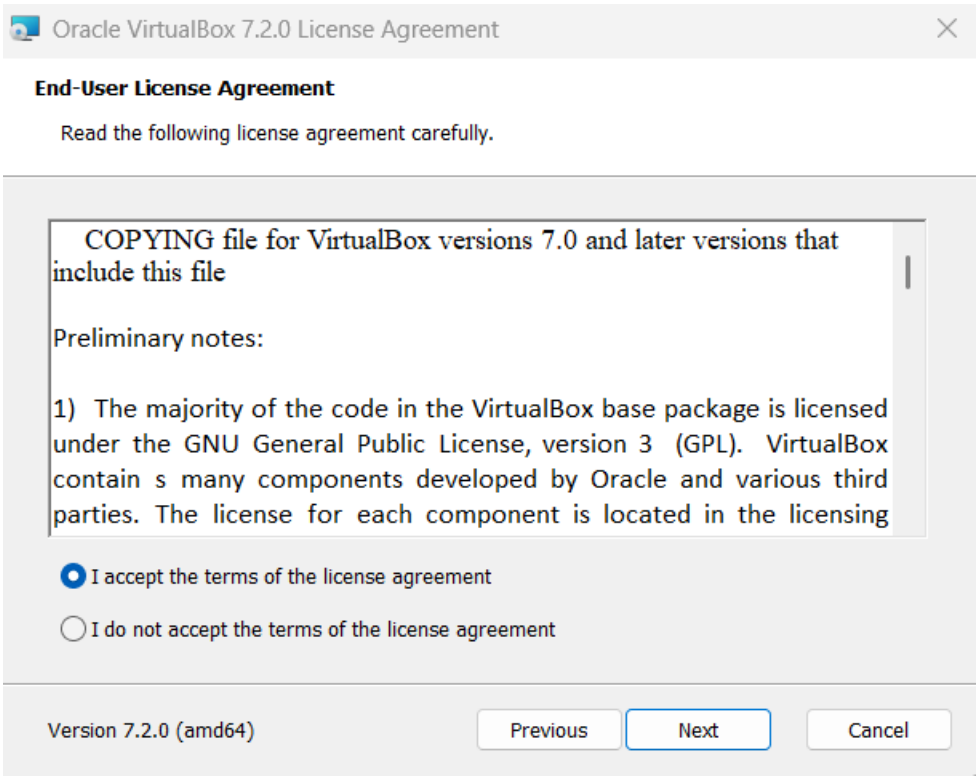
Download Virtualbox exe from https://www.virtualbox.org/wiki/Downloads :



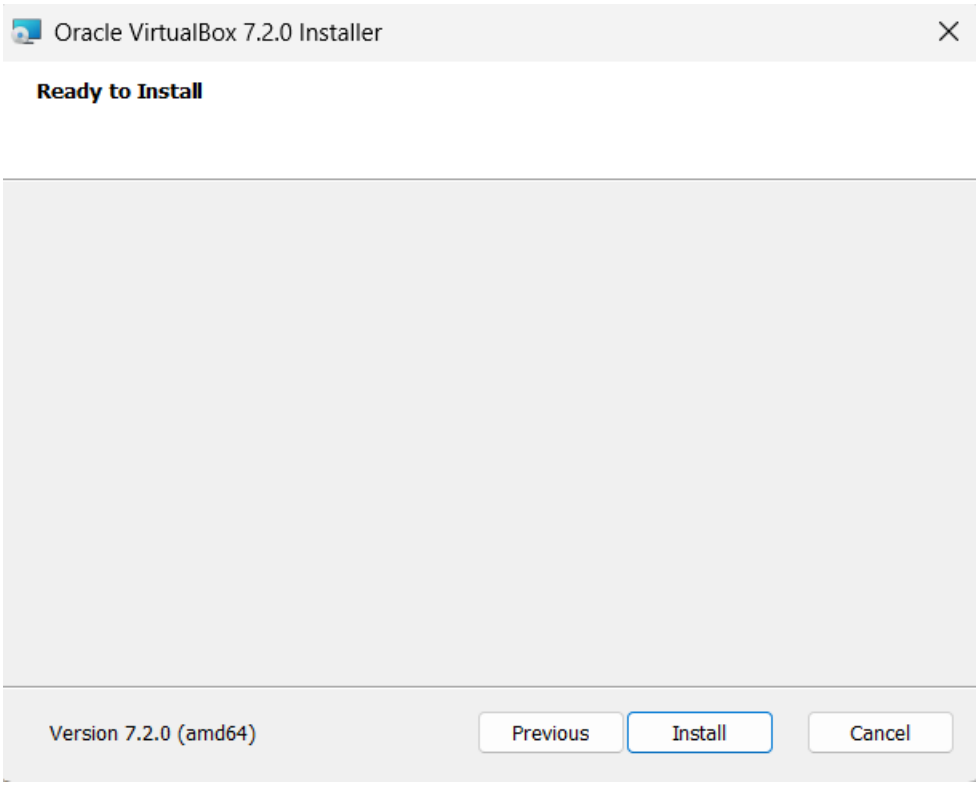
Follow the installation process:

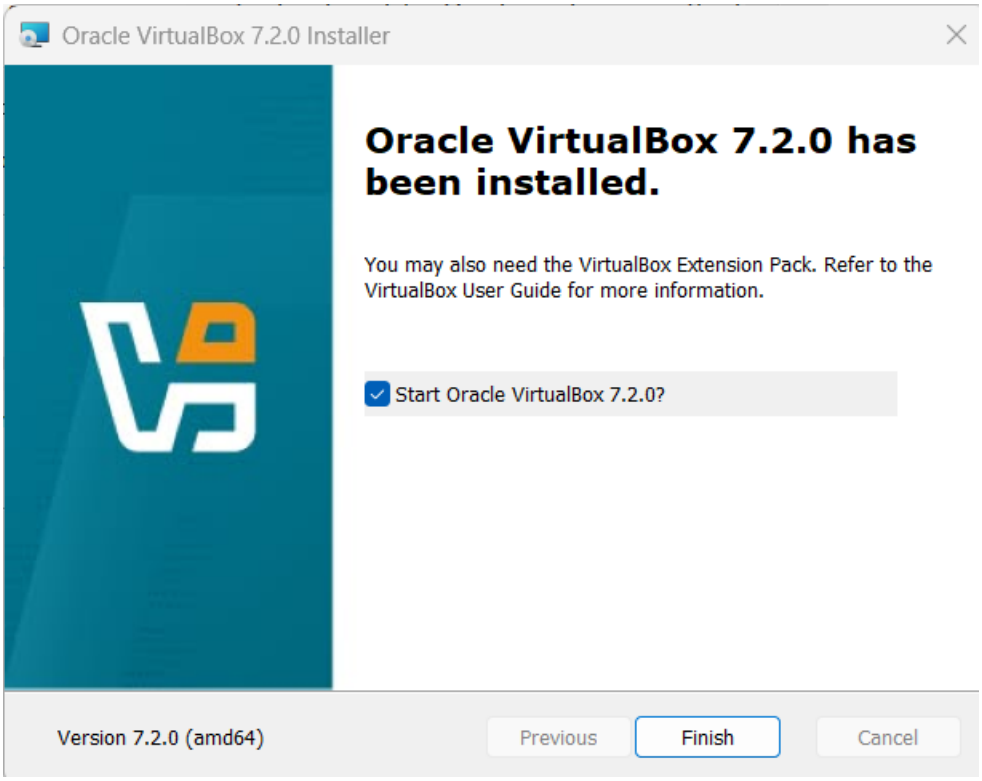


Accept the terms:

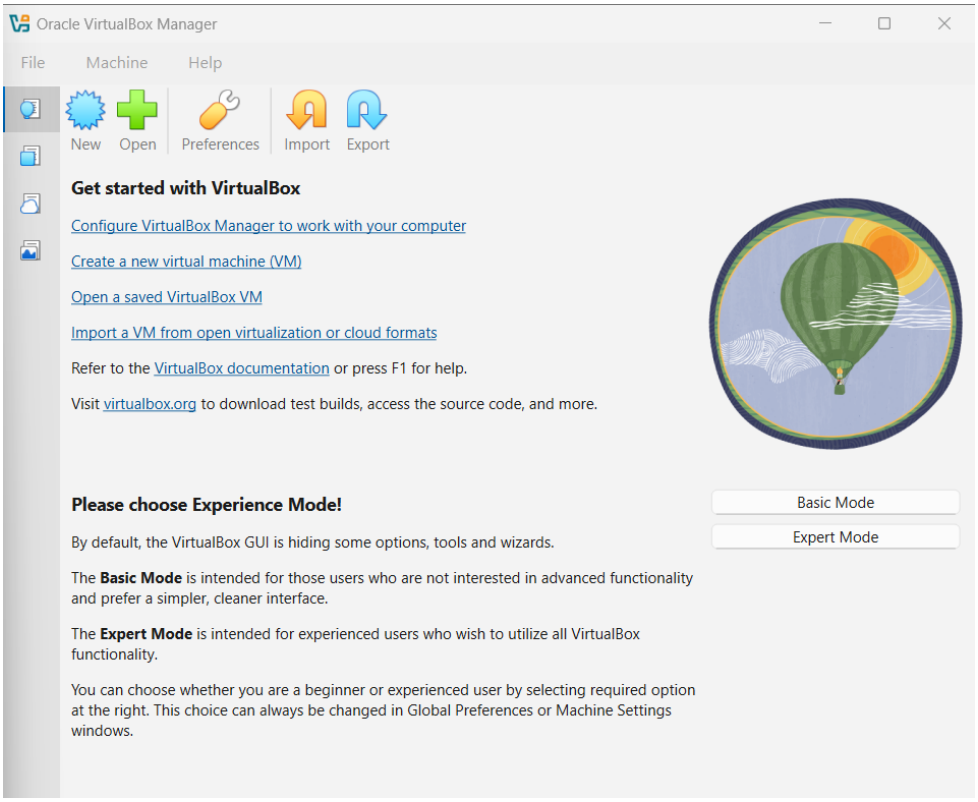


After following the necessary steps, install the application:

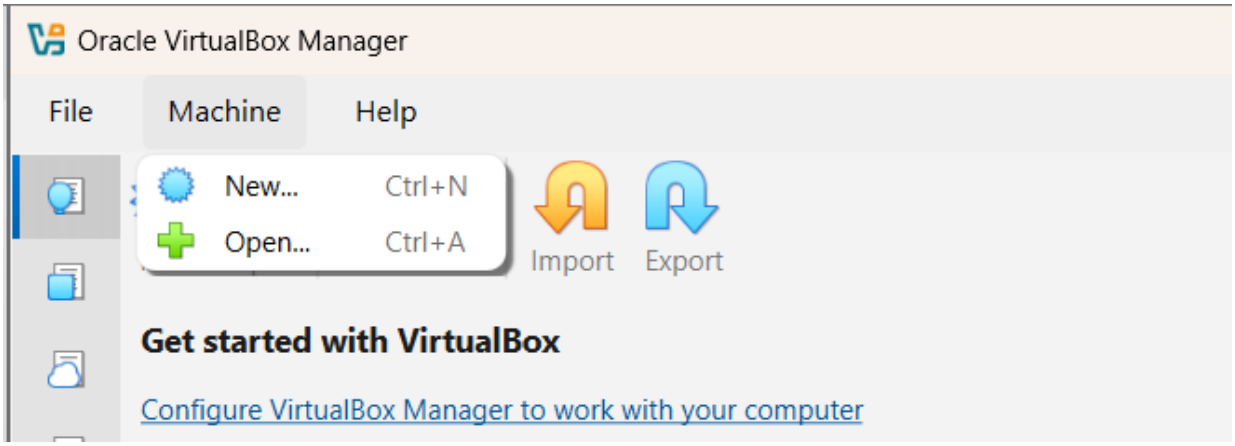


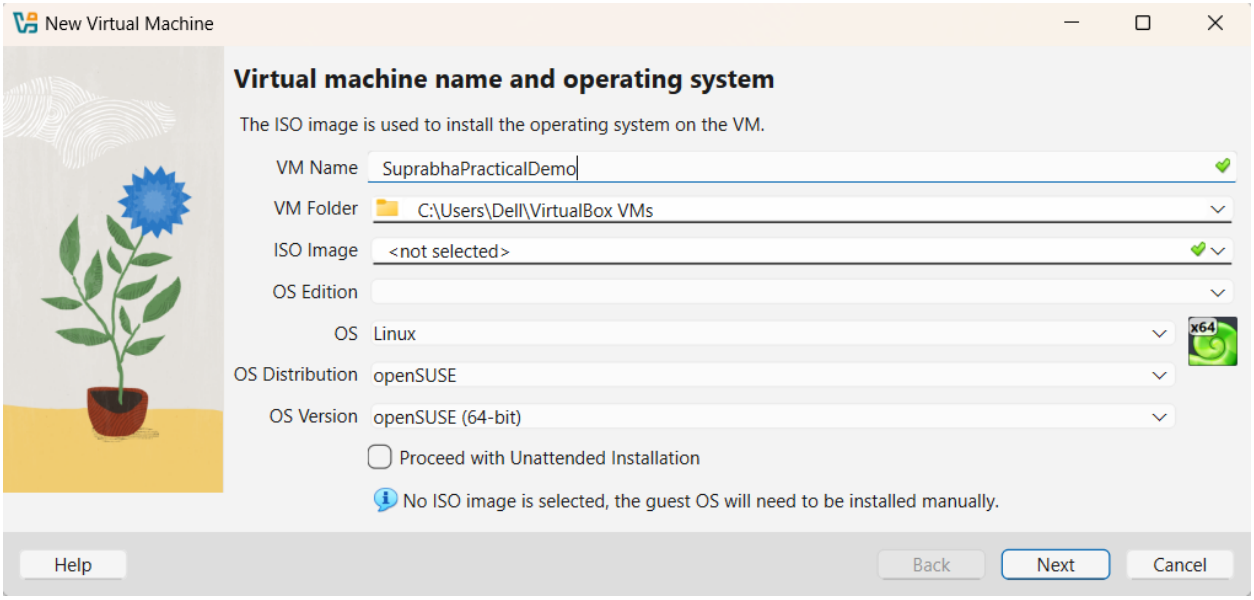


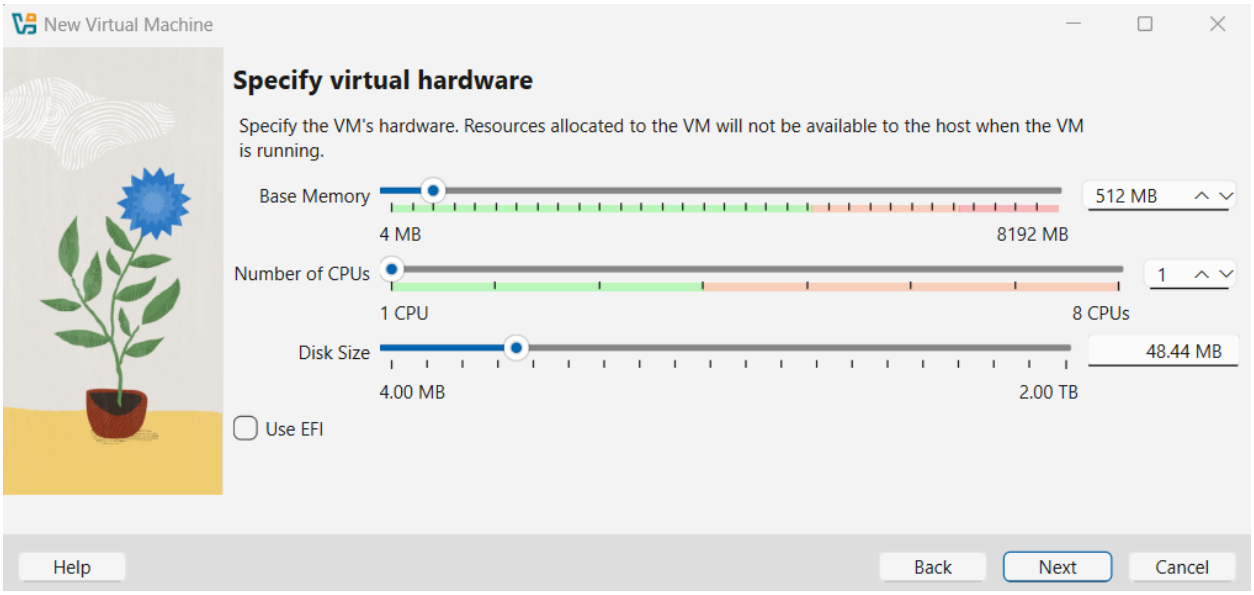
Launch the Application:

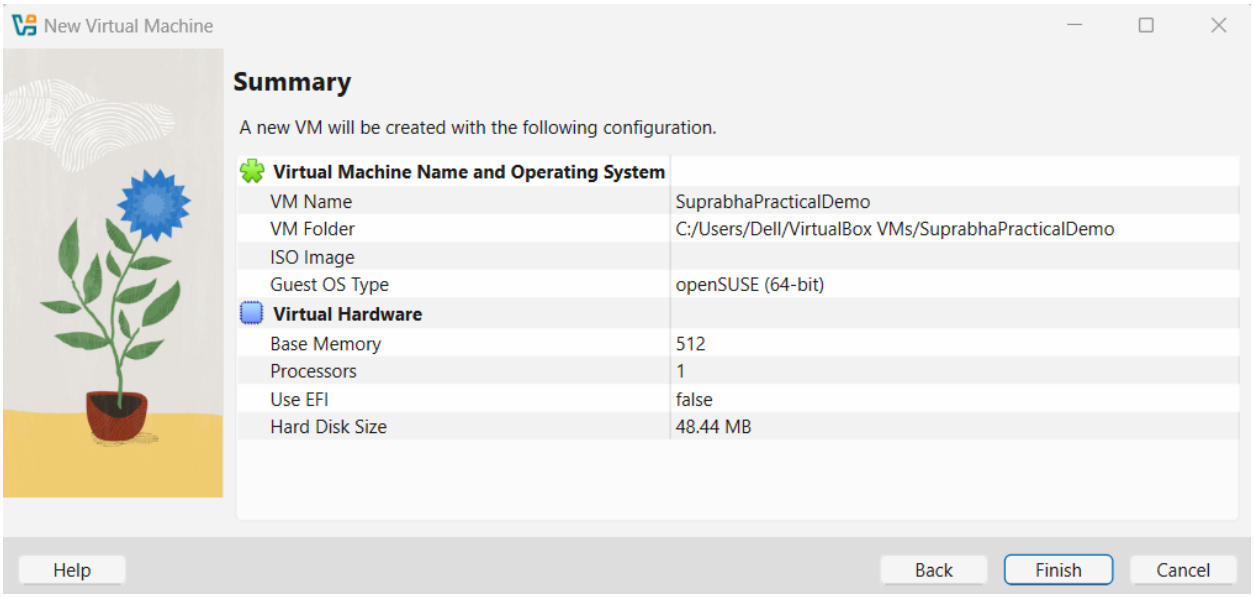


And then create a new vitual machine:

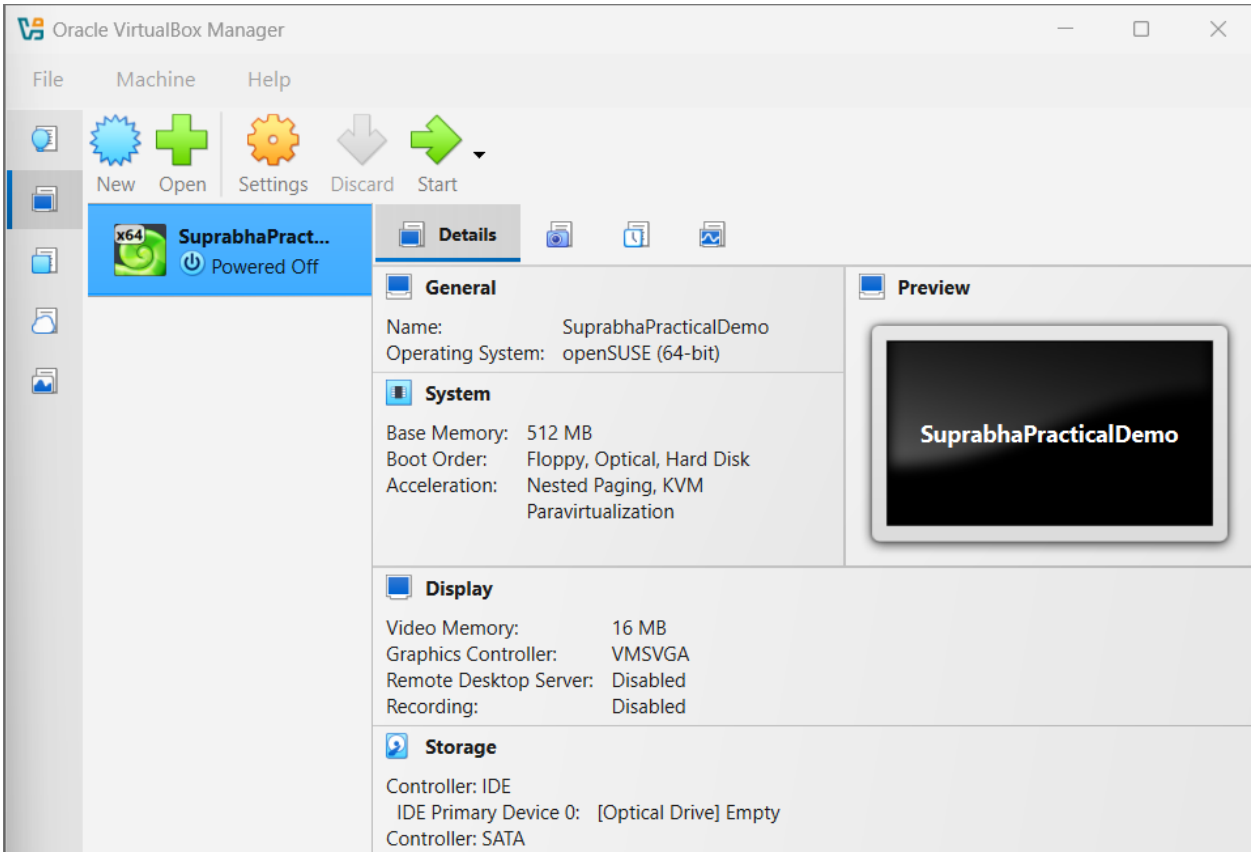


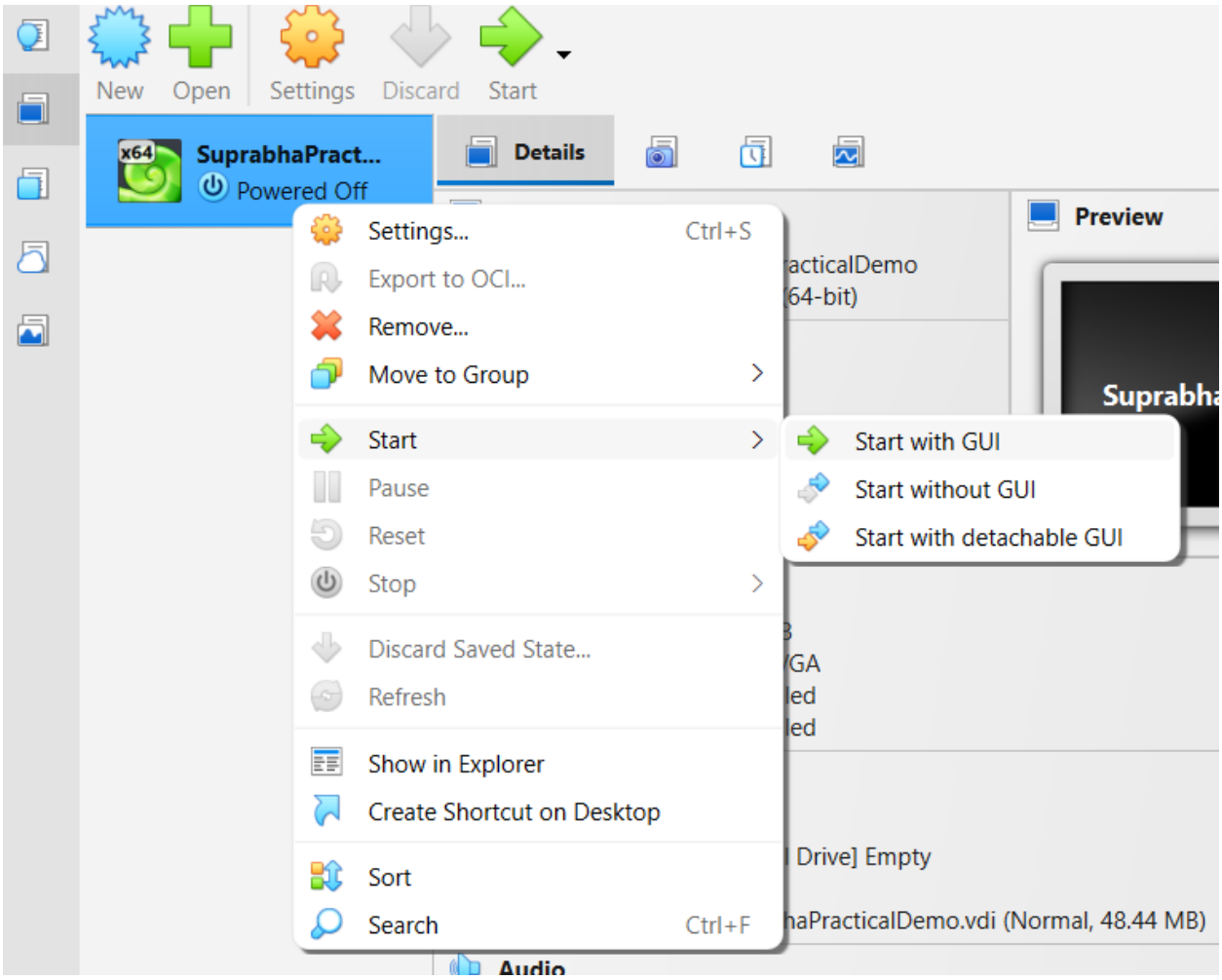


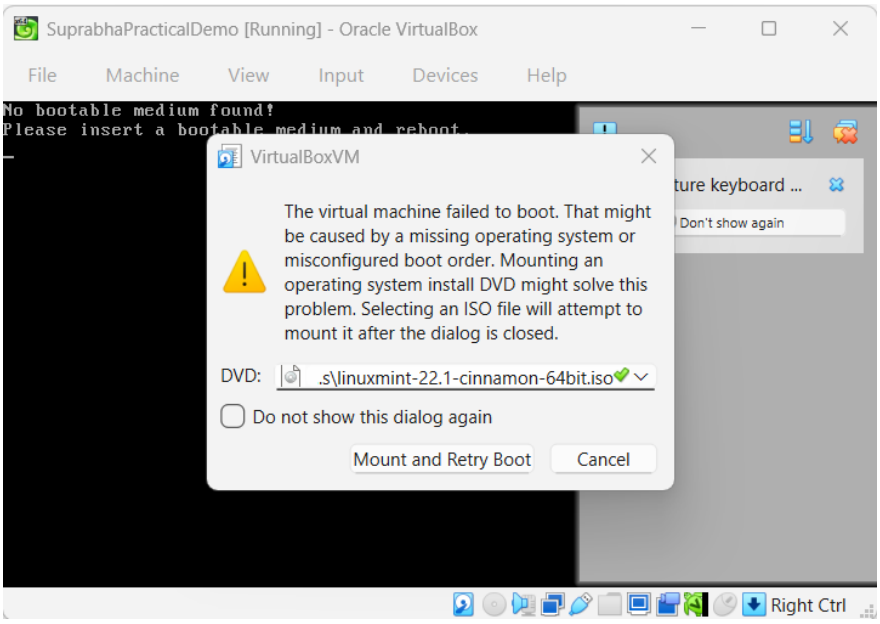


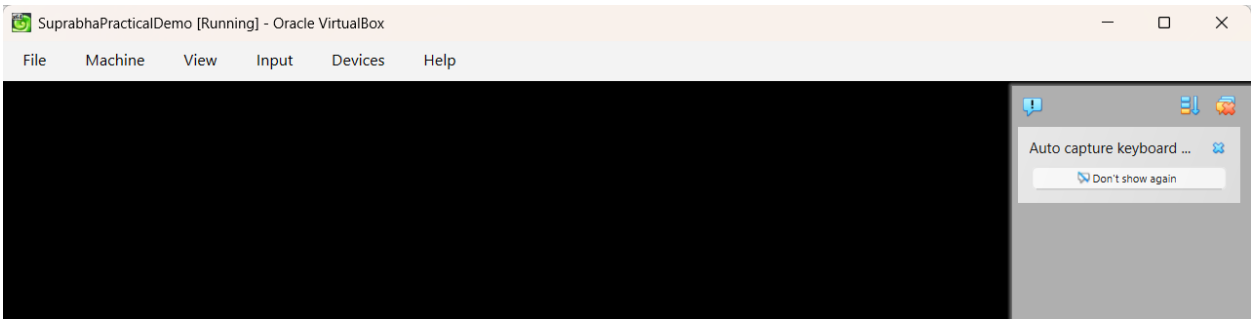

New Virtual Machine
Summary
A new VM will be created with the following configuration.
Virtual Machine Name and Operating System
VM Name
SuprabhaPracticalDemo
VM Folder
C:/Users/Dell/VirtualBox VMs/SuprabhaPracticalDemo
ISO Image
Guest OS Type
openSUSE (64-bit)
Virtual Hardware
Base Memory
512
Processors
1
Use EFI
false
Hard Disk Size
48.44 MB
Help
Back
Finish
Cancel

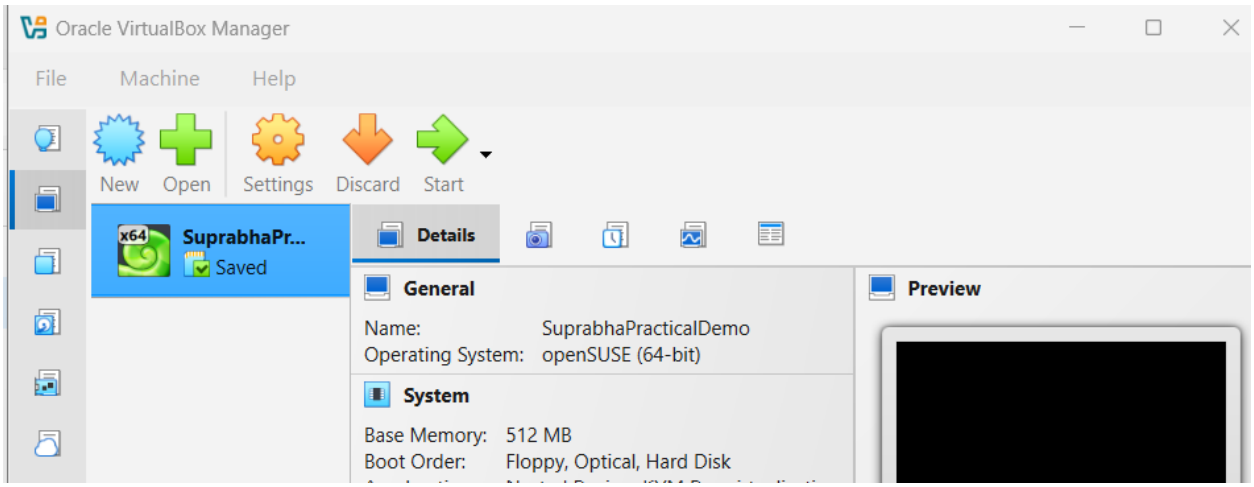
Start it up:


Oracle VirtualBox Manager
File
Machine
Help
New
Open
Settings
Discard
Start
SuprabhaPract...
Powered Off
Details
General
Name: SuprabhaPracticalDemo
Operating System: openSUSE (64-bit)
System
Base Memory: 512 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM
Paravirtualization
Preview
SuprabhaPracticalDemo
Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled
Storage
Controller: IDE
IDE Primary Device 0: [Optical Drive] Empty
Controller: SATA


New
Open
Settings
Discard
Start
SuprabhaPract...
Powered Off
Details
Preview
Settings... Ctrl+S
Export to OCI...
Remove...
Move to Group
Start
Pause
Reset
Stop
Discard Saved State...
Refresh
Show in Explorer
Create Shortcut on Desktop
Sort
Search Ctrl+F
Start with GUI
Start without GUI
Start with detachable GUI
Audio


SuprabhaPracticalDemo [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
No bootable medium found!
Please insert a bootable medium and reboot.
VirtualBoxVM
The virtual machine failed to boot. That might be caused by a missing operating system or misconfigured boot order. Mounting an operating system install DVD might solve this problem. Selecting an ISO file will attempt to mount it after the dialog is closed.
DVD: .s\linuxmint-22.1-cinnamon-64bit.iso
Do not show this dialog again
Mount and Retry Boot
Cancel
Right Ctrl


SuprabhaPracticalDemo [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Auto capture keyboard ...
Don't show again


Oracle VirtualBox Manager
File
Machine
Help
New
Open
Settings
Discard
Start
SuprabhaPr...
Saved
Details
General
Name: SuprabhaPracticalDemo
Operating System: openSUSE (64-bit)
System
Base Memory: 512 MB
Boot Order: Floppy, Optical, Hard Disk
Preview

Lab-6 Simulate a cloud scenario using CloudSim

Import cloudsim and the necessary dependencies required ;

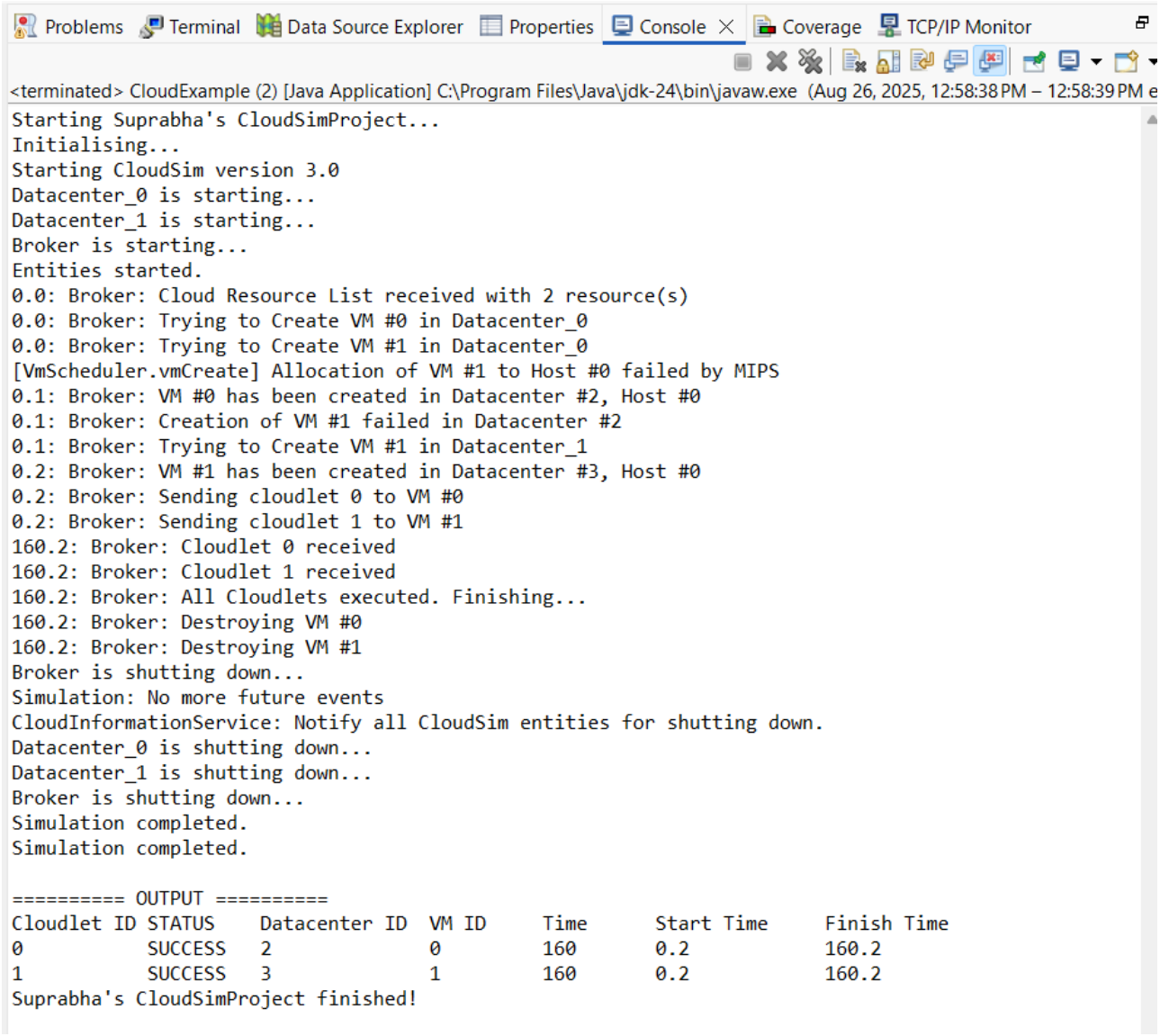
And execute the following code in CloudExample.java:

```
package CloudSimProject;

import java.text.DecimalFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.LinkedList;
import java.util.List;

import org.cloudbus.cloudsim.*;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.*;

public class CloudExample {

    private static List<Cloudlet> cloudletList;
    private static List<Vm> vmlist;

    public static void main(String[] args) {

        Log.printLine("Starting Suprabha's CloudSimProject...");

        try {
            // ---------- 1. INIT ----------
            int numUsers = 1;
            Calendar calendar = Calendar.getInstance();
            boolean traceFlag = false;
            CloudSim.init(numUsers, calendar, traceFlag);

            // ---------- 2. DATACENTERS ----------
            createDatacenter("Datacenter_0");
```

```
createDatacenter("Datacenter_1");

// ---------- 3. BROKER ----------
DatacenterBroker broker = createBroker();
int brokerId = broker.getId();

// ---------- 4. VMs ----------
vmlist = new ArrayList<>();
int mips = 250;
long size = 10_000;   // image size (MB)
int ram  = 512;       // VM memory (MB)
long bw   = 1_000;    // bandwidth
int pes   = 1;        // number of vCPUs
String vmm = "Xen";

vmlist.add(new Vm(0, brokerId, mips, pes, ram, bw, size, vmm,
            new CloudletSchedulerTimeShared()));
vmlist.add(new Vm(1, brokerId, mips, pes, ram, bw, size, vmm,
            new CloudletSchedulerTimeShared()));
broker.submitVmList(vmlist);

// ---------- 5. CLOUDLETS ----------
cloudletList = new ArrayList<>();
long length      = 40_000;
long fileSize    = 300;
long outputSize  = 300;
UtilizationModel util = new UtilizationModelFull();

Cloudlet c1 = new Cloudlet(0, length, pes, fileSize, outputSize,
                    util, util, util);
c1.setUserId(brokerId);
Cloudlet c2 = new Cloudlet(1, length, pes, fileSize, outputSize,
                    util, util, util);
c2.setUserId(brokerId);

cloudletList.add(c1);
cloudletList.add(c2);
broker.submitCloudletList(cloudletList);

// pin each cloudlet to a different VM
broker.bindCloudletToVm(c1.getCloudletId(), vmlist.get(0).getId());
broker.bindCloudletToVm(c2.getCloudletId(), vmlist.get(1).getId());

// ---------- 6. RUN SIM ----------
CloudSim.startSimulation();
List<Cloudlet> newList = broker.getCloudletReceivedList();
```

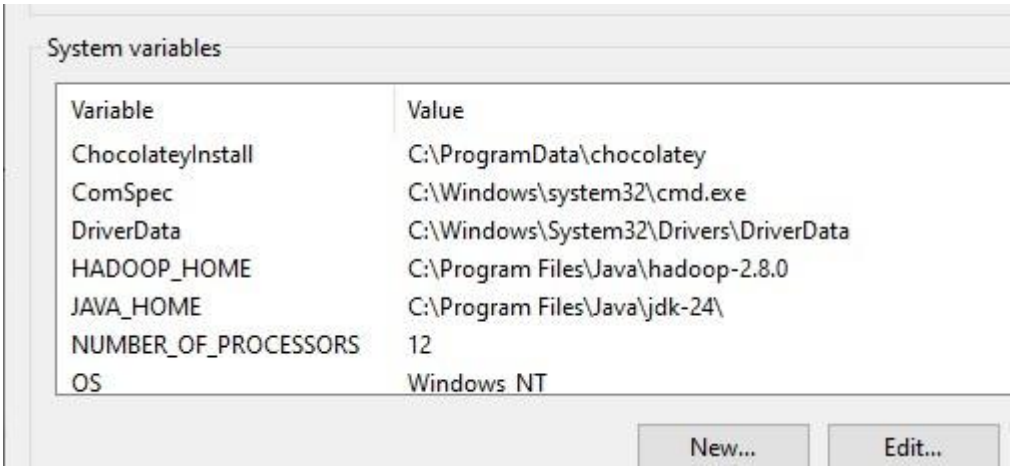
```
            CloudSim.stopSimulation();

            printCloudletList(newList);
            Log.printLine("Suprabha's CloudSimProject finished!");

        } catch (Exception e) {
            e.printStackTrace();
            Log.printLine("The simulation has been terminated due to an unexpected error.");
        }
    }

    /* ---------- helper methods ---------- */

    private static Datacenter createDatacenter(String name) {
        List<Host> hostList = new ArrayList<>();
        List<Pe>   peList   = new ArrayList<>();

        int mips = 1000;
        peList.add(new Pe(0, new PeProvisionerSimple(mips)));

        int hostId  = 0;
        int ram     = 2048;      // MB
        long storage = 1_000_000; // MB
        int bw      = 10_000;

        hostList.add(new Host(
                hostId,
                new RamProvisionerSimple(ram),
                new BwProvisionerSimple(bw),
                storage,
                peList,
                new VmSchedulerSpaceShared(peList)
        ));

        String arch           = "x86";
        String os             = "Linux";
        String vmm            = "Xen";
        double timeZone       = 10.0;
        double cost           = 3.0;
        double costPerMem     = 0.05;
        double costPerStorage = 0.001;
        double costPerBw      = 0.0;

        DatacenterCharacteristics characteristics = new DatacenterCharacteristics(
                arch, os, vmm, hostList, timeZone,
                cost, costPerMem, costPerStorage, costPerBw);
```

```
        try {
            return new Datacenter(name, characteristics,
                    new VmAllocationPolicySimple(hostList),
                    new LinkedList<Storage>(), 0);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    private static DatacenterBroker createBroker() {
        try {
            return new DatacenterBroker("Broker");
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }

    /* ---------- the bit that was broken ---------- */
    private static void printCloudletList(List<Cloudlet> list) {
        DecimalFormat dft = new DecimalFormat("###.##");

        Log.printLine("\n========== OUTPUT ==========");
        Log.printLine(String.format(
                "%-12s%-10s%-15s%-10s%-10s%-15s%-15s",
                "Cloudlet ID", "STATUS", "Datacenter ID", "VM ID",
                "Time", "Start Time", "Finish Time"));

        for (Cloudlet cloudlet : list) {
            if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {
                Log.printLine(String.format(
                        "%-12d%-10s%-15d%-10d%-10s%-15s%-15s",
                        cloudlet.getCloudletId(), "SUCCESS",
                        cloudlet.getResourceId(), cloudlet.getVmId(),
                        dft.format(cloudlet.getActualCPUTime()),
                        dft.format(cloudlet.getExecStartTime()),
                        dft.format(cloudlet.getFinishTime())));   // **only TWO ) before ;**
            }
        }
    }
}
```

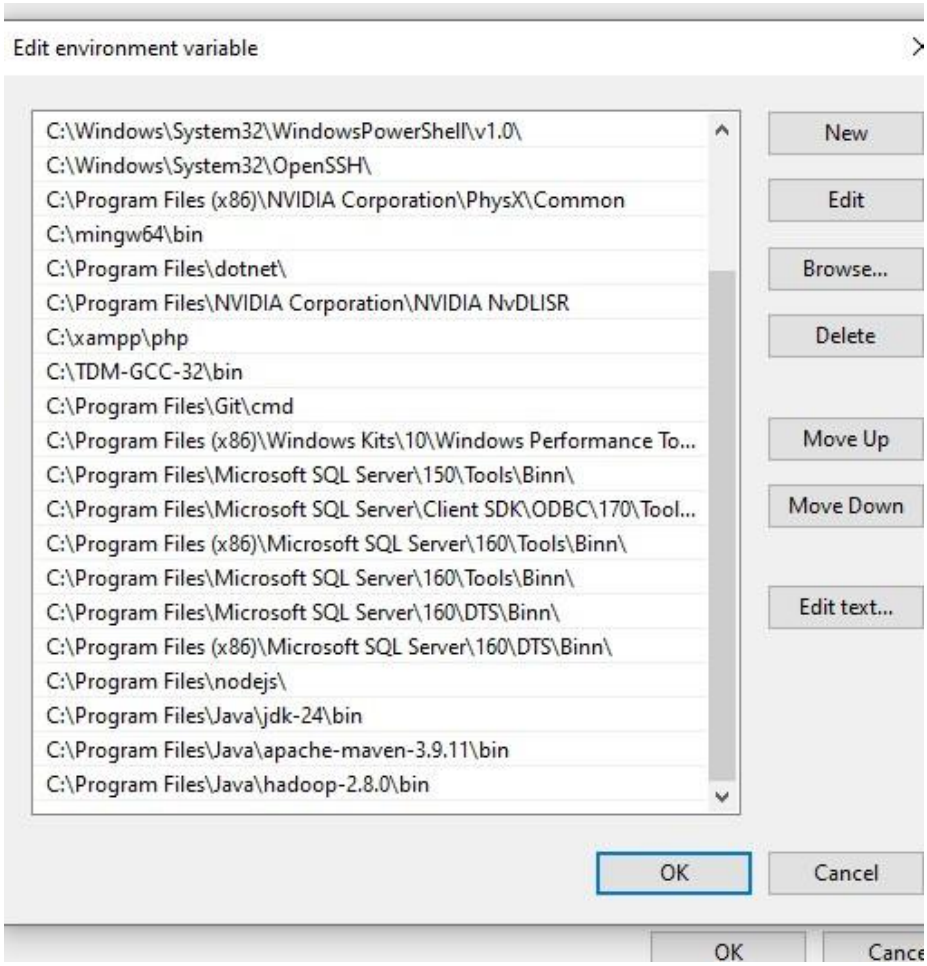

Problems
Terminal
Data Source Explorer
Properties
Console
Coverage
TCP/IP Monitor
<terminated> CloudExample (2) [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (Aug 26, 2025, 12:58:38 PM – 12:58:39 PM e
Starting Suprabha's CloudSimProject...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
========== OUTPUT ==========
Cloudlet ID STATUS Datacenter ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 160 0.2 160.2
1 SUCCESS 3 1 160 0.2 160.2
Suprabha's CloudSimProject finished!

Lab-7 Installation and testing of Hadoop single node cluster on windows

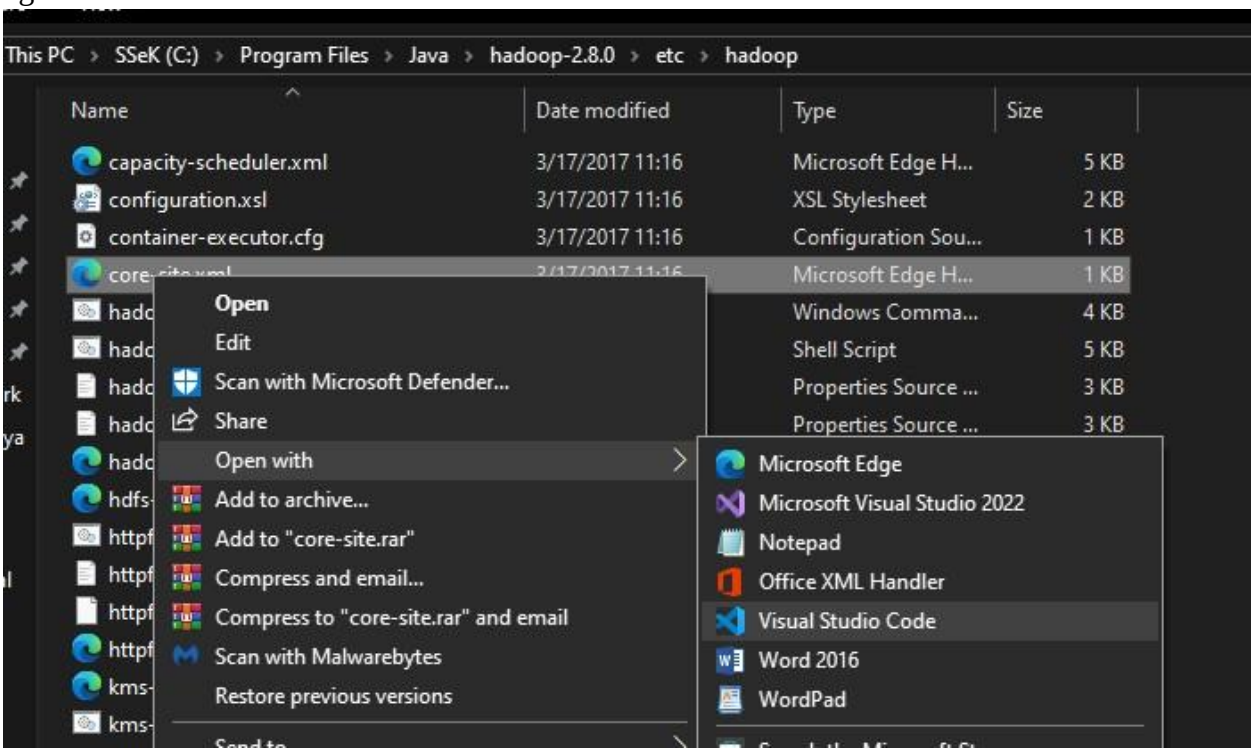
Download Hadoop from http://archive.apache.org/dist/hadoop/core//hadoop-2.8.0/hadoop-2.8.0.tar.gz and then, install and setup JDK and Hadoop set up system variables for both:



Then go to the system environment variables and add this to the path:



Go to Hadoop folder hadoop-2.8.0\etc\hadoop, edit core-site.xml with a rich text editor, eg Visual Studio Code



Modify
<configuration>

</configuration>

To
<configuration>

<property>

<name>fs.defaultFS</name>

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

</property>

</configuration>

Find mapred-site.xml.template and modify it to mapred-site.xml and add :

<property>

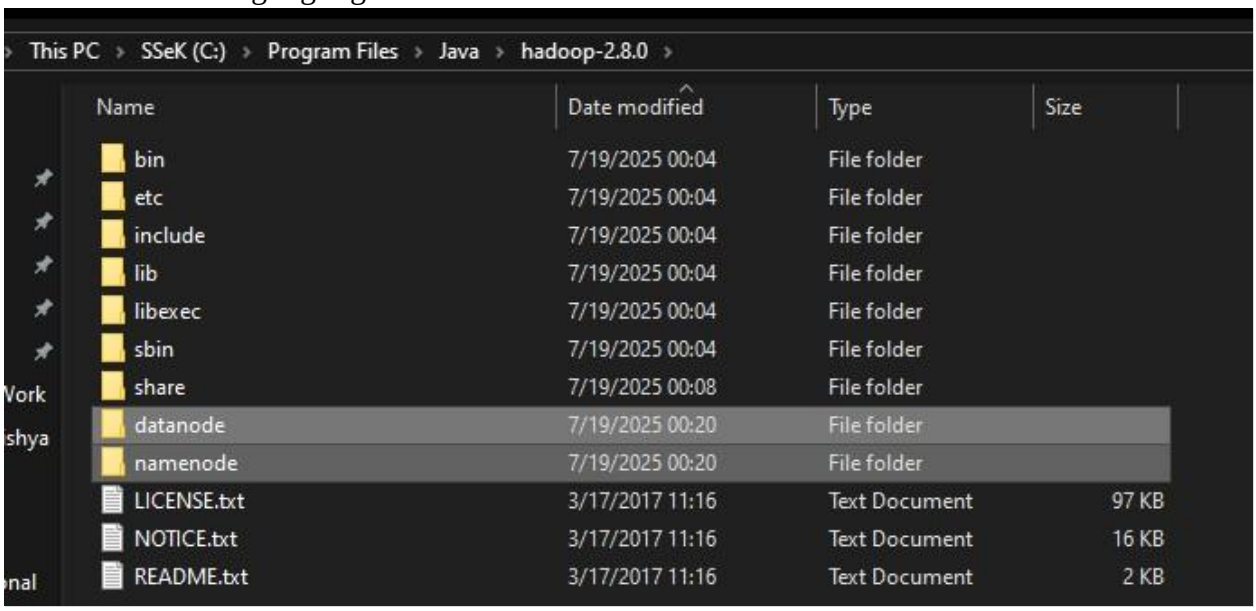
<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

To the configuration tags

Create the following highlighted folders



Edit Hadoop-2.8.0\etc\hadoop/hdfs-site.xml with

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>C:\hadoop-2.8.0\data\namenode</value>

</property>

<property>

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

<value>C:\hadoop-2.8.0\data\datanode</value>

</property>

And Hadoop-2.8.0\etc\hadoop\yarn-site.xml with

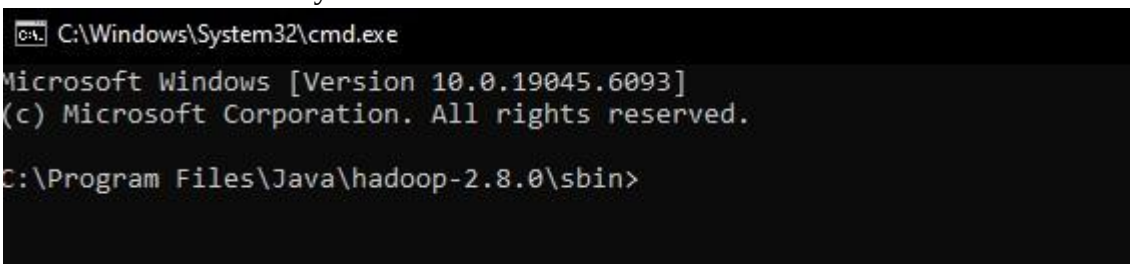
```
 <property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce_shuffle</value>

  </property>

  <property>

<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

  </property>
```

Edit the file Hadoop-2.8.0\etc/hadoop\hadoop-env.cmd and write @rem in front of "set JAVA_HOME=%JAVA_HOME%". Write set JAVA_HOME={JDK directory} at the next row. This is C:\Program Files\Java\jdk-24 for me

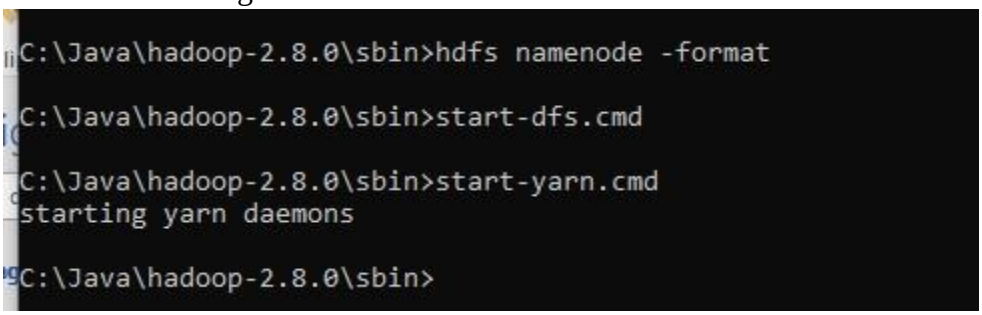
Download Hadoop Configuration.zip from https://github.com/MuhammadBilalYar/HADOOP-INSTALLATION-ON-WINDOW10/blob/master/Hadoop%20Configuration.zip

And replace the bin from Hadoop-2.8.0 with bin from configuration

Run cmd at this directory



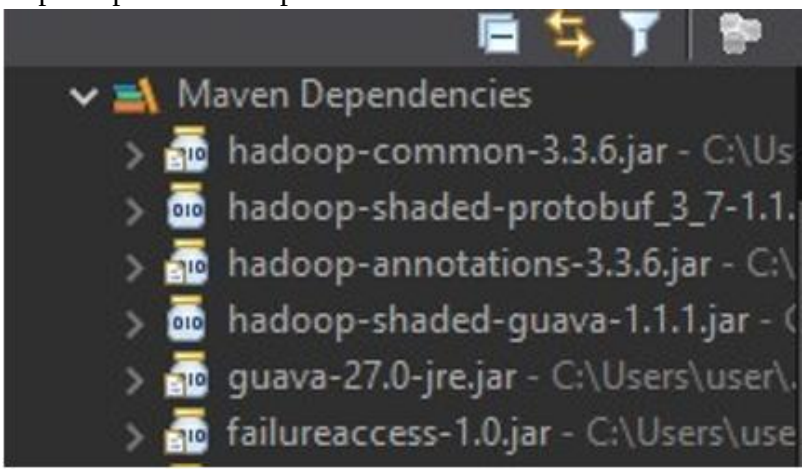
Runn the following commands



In browser go to localhost:8088

Lab-8 Mapreduce wordcount program using java

Import apache Hadoop 3.3.6 :

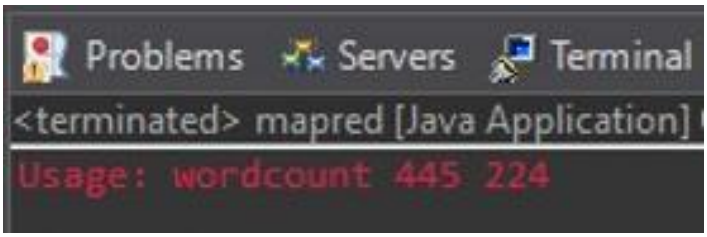


Create a word count reducer class with following code

```
package mapred;

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;

import com.ctc.wstx.util.WordSet;

public class mapred {

  public static class TokenizerMapper

       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);

    private Text word = new Text();
```

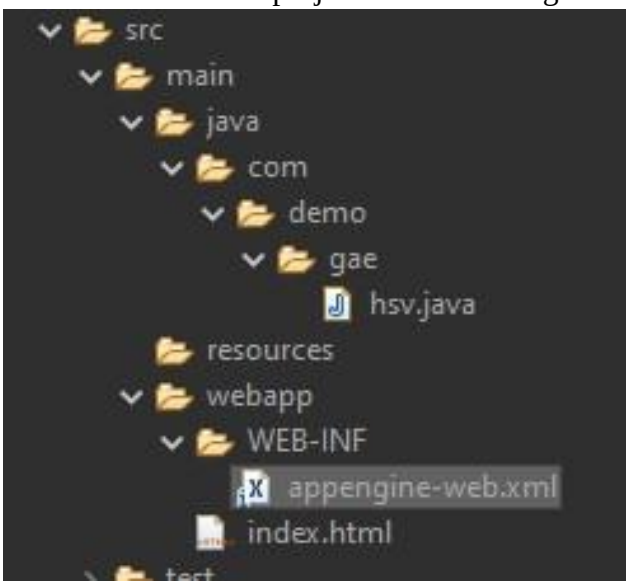
```
  public void map(Object key, Text value, Context
context                    ) throws IOException,
InterruptedException {

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

     while
(itr.hasMoreTokens())        {
word.set(itr.nextToken());
context.write(word, one);
     }

   }

  }

   public class SumReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

    @Override

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

 int sum = 0;

 for (IntWritable val : values) {

            sum += val.get();

       }

       context.write(key, new IntWritable(sum));

   }

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

    Configuration conf = new Configuration();
Job job = Job.getInstance(conf, "word
count");
job.setJarByClass(WordSet.class);
job.setMapperClass(TokenizerMapper.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);
```

```
  }
}
```


Problems
Servers
Terminal
<terminated> mapred [Java Application]
Usage: wordcount 445 224

Lab-9 Develop and application for Google App Engine

Create a new maven project with following structure



With each containing the following
code Hsv.java

```
package com.demo.gae;

import java.io.IOException;

import jakarta.servlet.http.*;

@SuppressWarnings("serial")
public class hsv extends
HttpServlet {

   @Override     public void doGet(HttpServletRequest req,
HttpServletResponse resp)        throws IOException {
resp.setContentType("text/plain");
resp.getWriter().println("Hello, world from Java 17!");

   }

}
```

Appengine-web.xml

```
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">

  <runtime>java17</runtime>

  <threadsafe>true</threadsafe>

</appengine-web-app>
```

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello GAE</title>
  </head>
  <body>
    <h1>Hello from Java 17 on App Engine!</h1>
    <p><a href="/hello">Go to Servlet</a></p>
  </body>
</html>
```

Run the application by
mvn appengine:run
gcloud app deploy

Hello App Engine!

Available Servlets:
GoogleAppEngine