

**Tribhuvan University  
Institute of Science and Technology  
Patan Multiple Campus**



**LAB REPORT ON  
CLOUD COMPUTING (BIT408)**

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

**Submitted To:**  
Sachita Nand Mishra  
Lecturer, Patan Multiple Campus

**Submitted By:**  
Suprabha Aryal  
BIT 7<sup>th</sup> 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>:

The screenshot shows the Apache Tomcat 10.1.44 download page. The browser address bar displays <https://tomcat.apache.org/download-10.cgi>. On the left, a sidebar contains links for various Tomcat versions (11.0, 10.1, 9.0), documentation (Upgrading, Connectors, Native 2, Native 1.3, Wiki, Migration Guide, Presentations, Specifications), problem-solving (Problems?, Security Reports, Find help, FAQ, Mailing Lists, Bug Database, IRC), getting involved (Overview, Source code, Buildbot, Tools), and media (Twitter). The main content area is titled '10.1.44' and includes a link to the [README](#) file. It features two sections: 'Binary Distributions' and 'Source Code Distributions'. The 'Binary Distributions' section lists links for Core (zip, tar.gz, 32-bit Windows zip, 64-bit Windows zip, 32-bit/64-bit Windows Service Installer), Full documentation (tar.gz), Deployer (zip, tar.gz), and Embedded (tar.gz, zip). The 'Source Code Distributions' section lists links for tar.gz and zip.

Tomcat 11.0  
Tomcat 10.1  
Tomcat 9.0  
Upgrading  
Tomcat Connectors  
Tomcat Native 2  
Tomcat Native 1.3  
Wiki  
Migration Guide  
Presentations  
Specifications

**Problems?**  
Security Reports  
Find help  
FAQ  
Mailing Lists  
Bug Database  
IRC

**Get Involved**  
Overview  
Source code  
Buildbot  
Tools

**Media**  
Twitter

### 10.1.44

Please see the [README](#) file for packaging information. It explains what every distribution contains.

#### Binary Distributions

- Core:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
  - [32-bit Windows zip \(pgp, sha512\)](#)
  - [64-bit Windows zip \(pgp, sha512\)](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
  - [tar.gz \(pgp, sha512\)](#)
- Deployer:
  - [zip \(pgp, sha512\)](#)
  - [tar.gz \(pgp, sha512\)](#)
- Embedded:
  - [tar.gz \(pgp, sha512\)](#)
  - [zip \(pgp, sha512\)](#)

#### Source Code Distributions

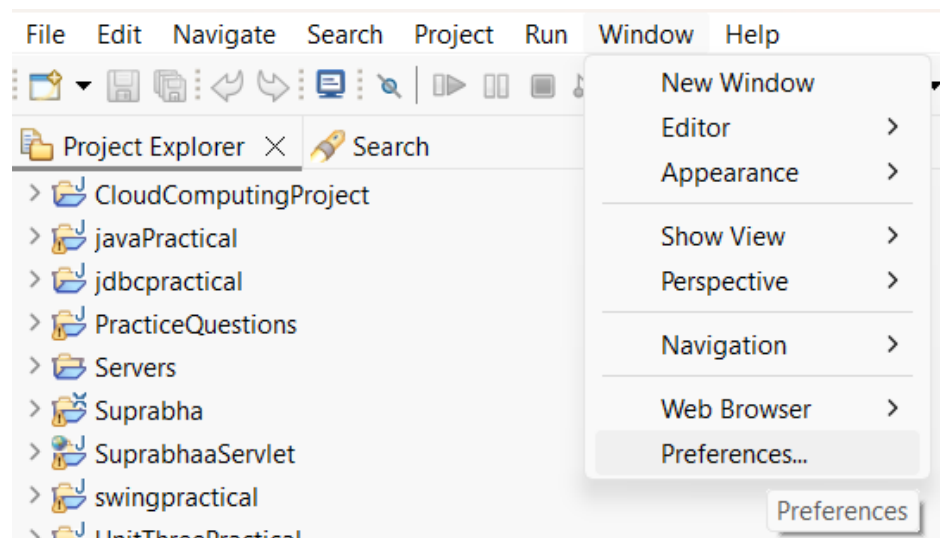
- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

Step: Extract the downloaded tomcat in your preferred folder:

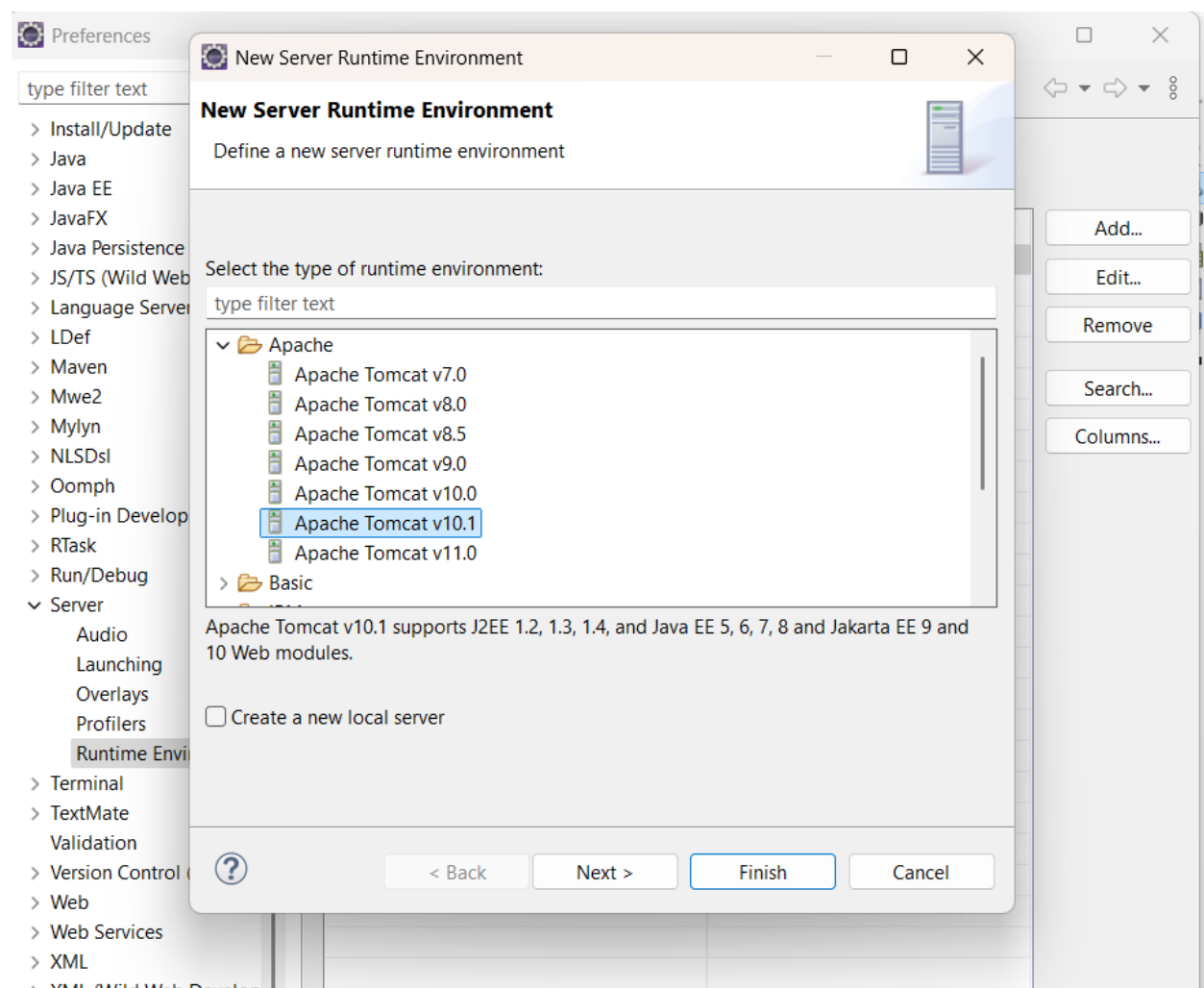
The screenshot shows a file explorer window with the path: Downloads > SuprabhaFiles > TomcatInstallation > apache-tomcat-10.1.44. The file list includes:

Name	Date modified	Type	Size
▼ Today			
BUILDING	8/26/2025 9:25 AM	Text Document	25 KB
CONTRIBUTING	8/26/2025 9:25 AM	Markdown Source ...	7 KB
LICENSE	8/26/2025 9:25 AM	File	61 KB
NOTICE	8/26/2025 9:25 AM	File	3 KB
README	8/26/2025 9:25 AM	Markdown Source ...	4 KB
RELEASE-NOTES	8/26/2025 9:25 AM	File	7 KB
RUNNING	8/26/2025 9:25 AM	Text Document	17 KB
webapps	8/26/2025 9:25 AM	File folder	
lib	8/26/2025 9:25 AM	File folder	
temp	8/26/2025 9:25 AM	File folder	
bin	8/26/2025 9:25 AM	File folder	
conf	8/26/2025 9:25 AM	File folder	
▼ Earlier this month			
logs	8/4/2025 1:14 PM	File folder	
work	8/4/2025 1:14 PM	File folder	

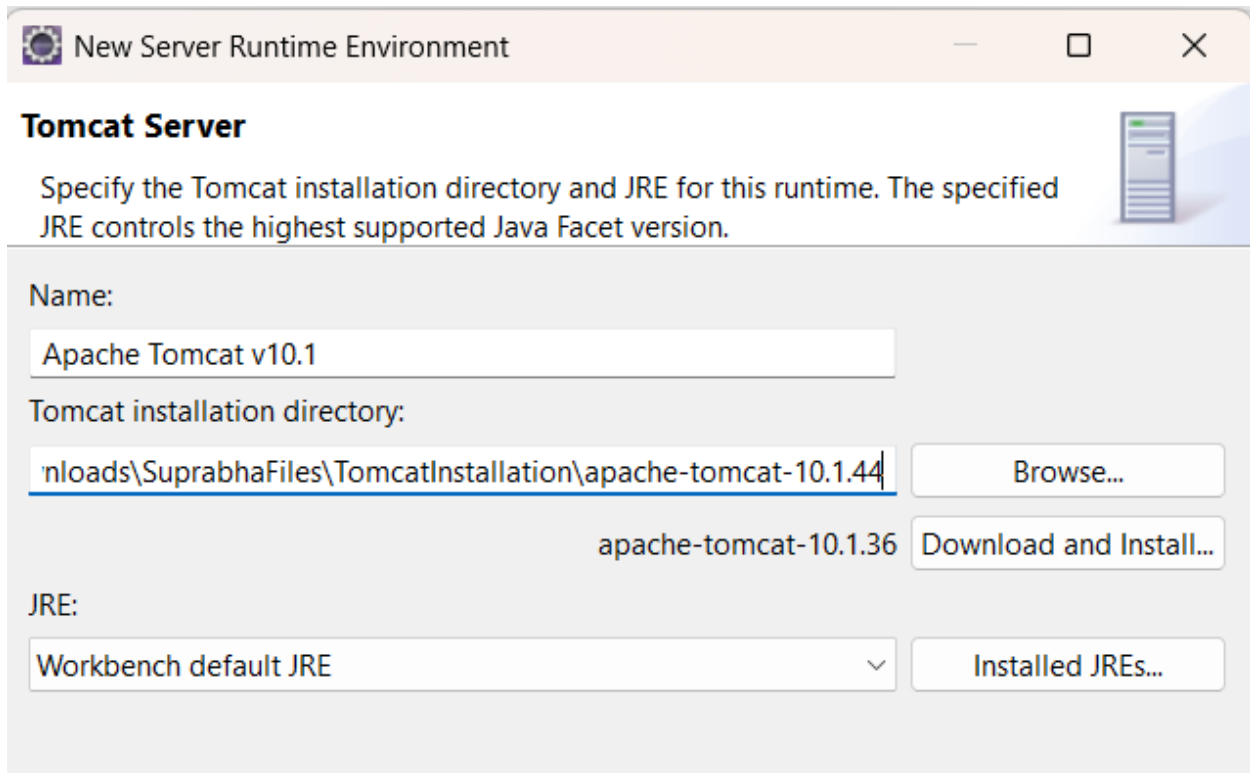
Step 3: Add Tomcat to the IDE. Here, I'm using Eclipse . First, Open the Window> Preferences:



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.



**New Server Runtime Environment**

### Tomcat Server

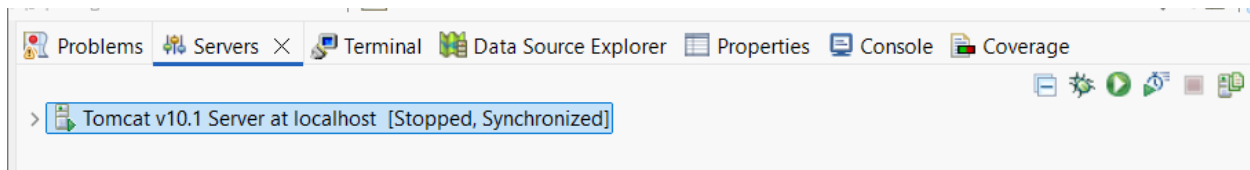
Specify the Tomcat installation directory and JRE for this runtime. The specified JRE controls the highest supported Java Facet version.

Name:  
Apache Tomcat v10.1

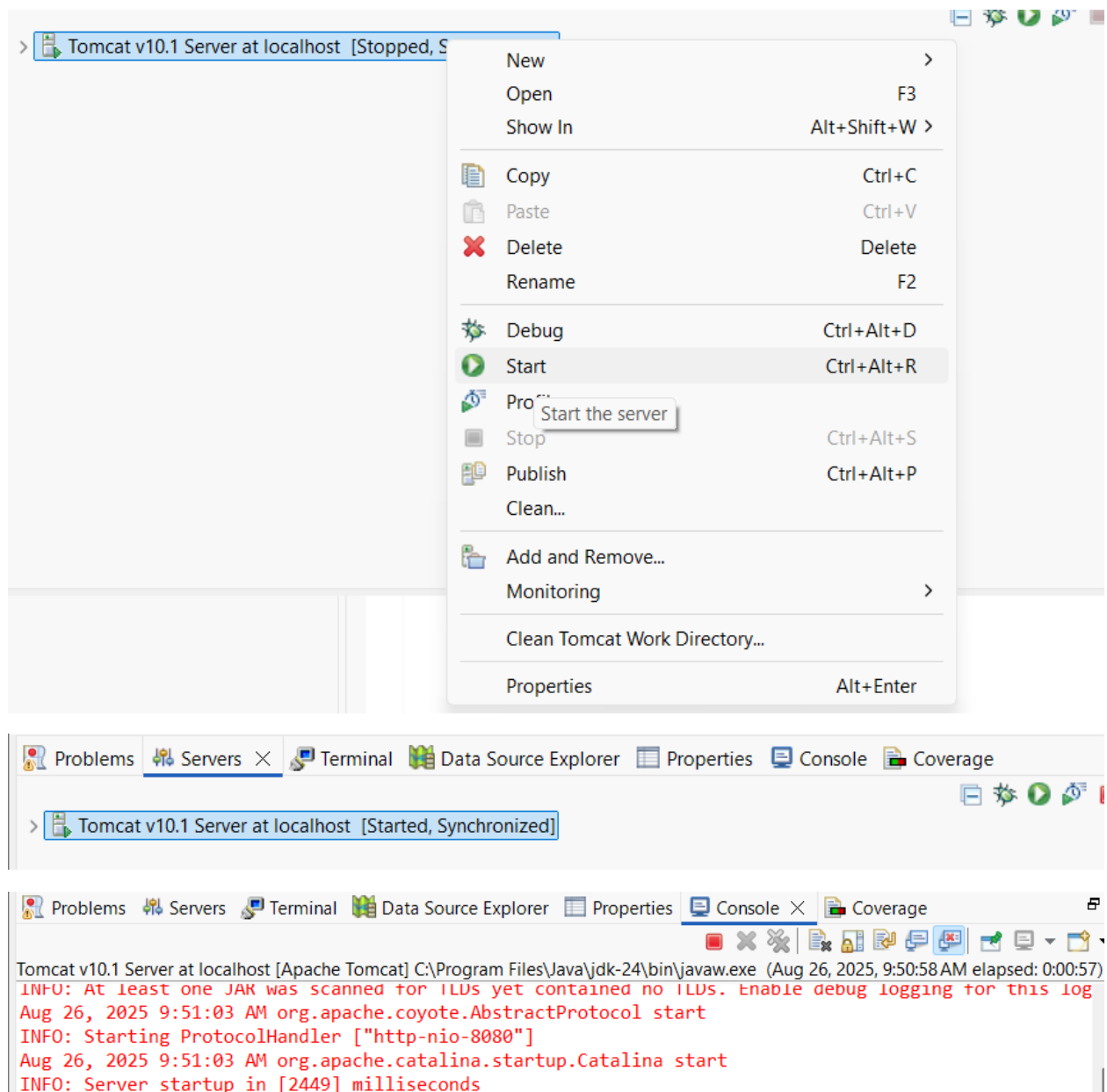
Tomcat installation directory:  
nloads\SuprabhaFiles\TomcatInstallation\apache-tomcat-10.1.44   
apache-tomcat-10.1.36

JRE:  
Workbench default JRE

The server is stopped,



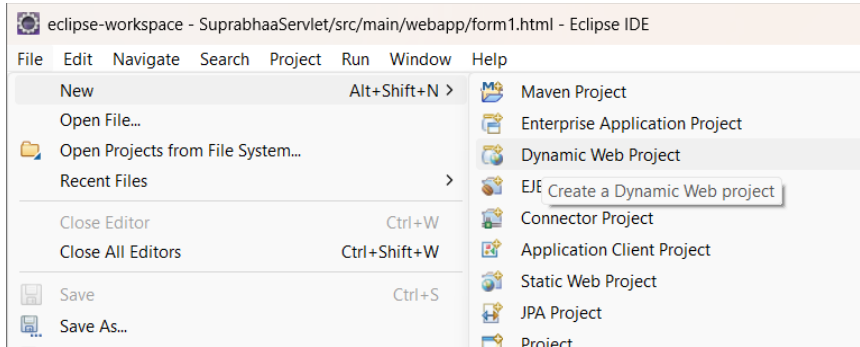
## Step 6: Start the Server:



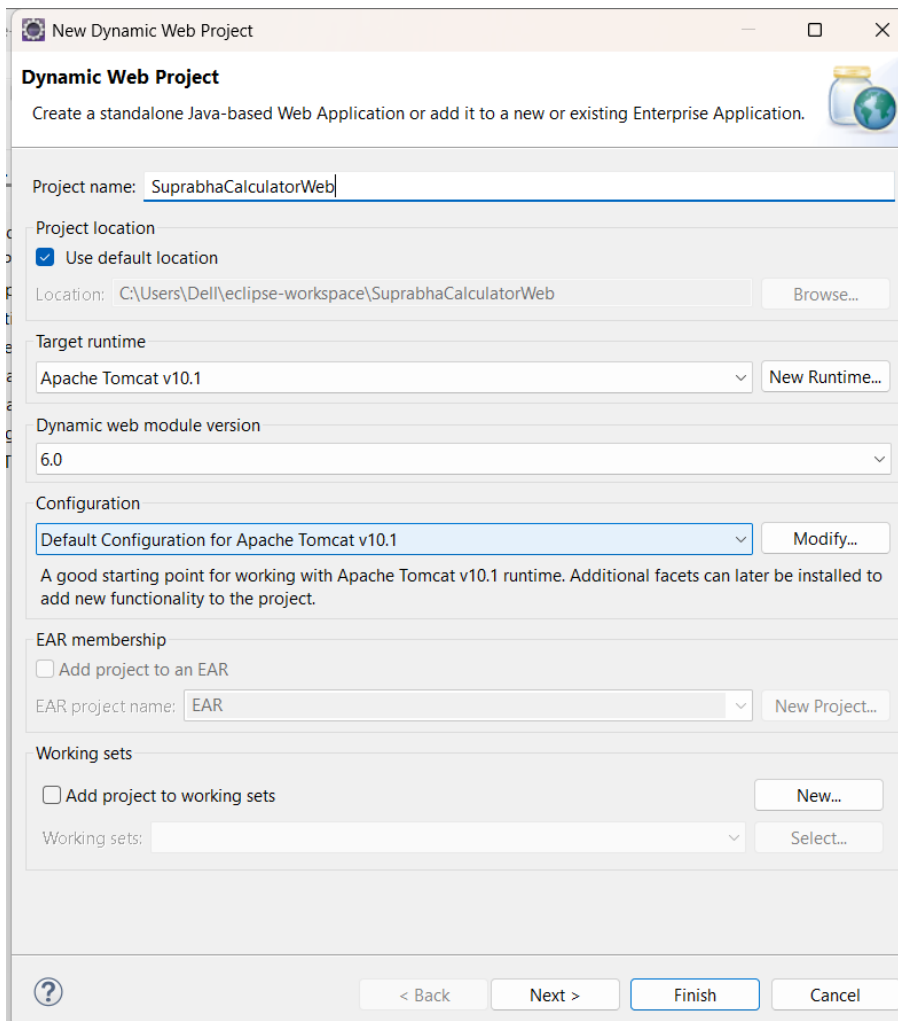
## Lab-2 SOAP Web Service in Java (JAX-WS)

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

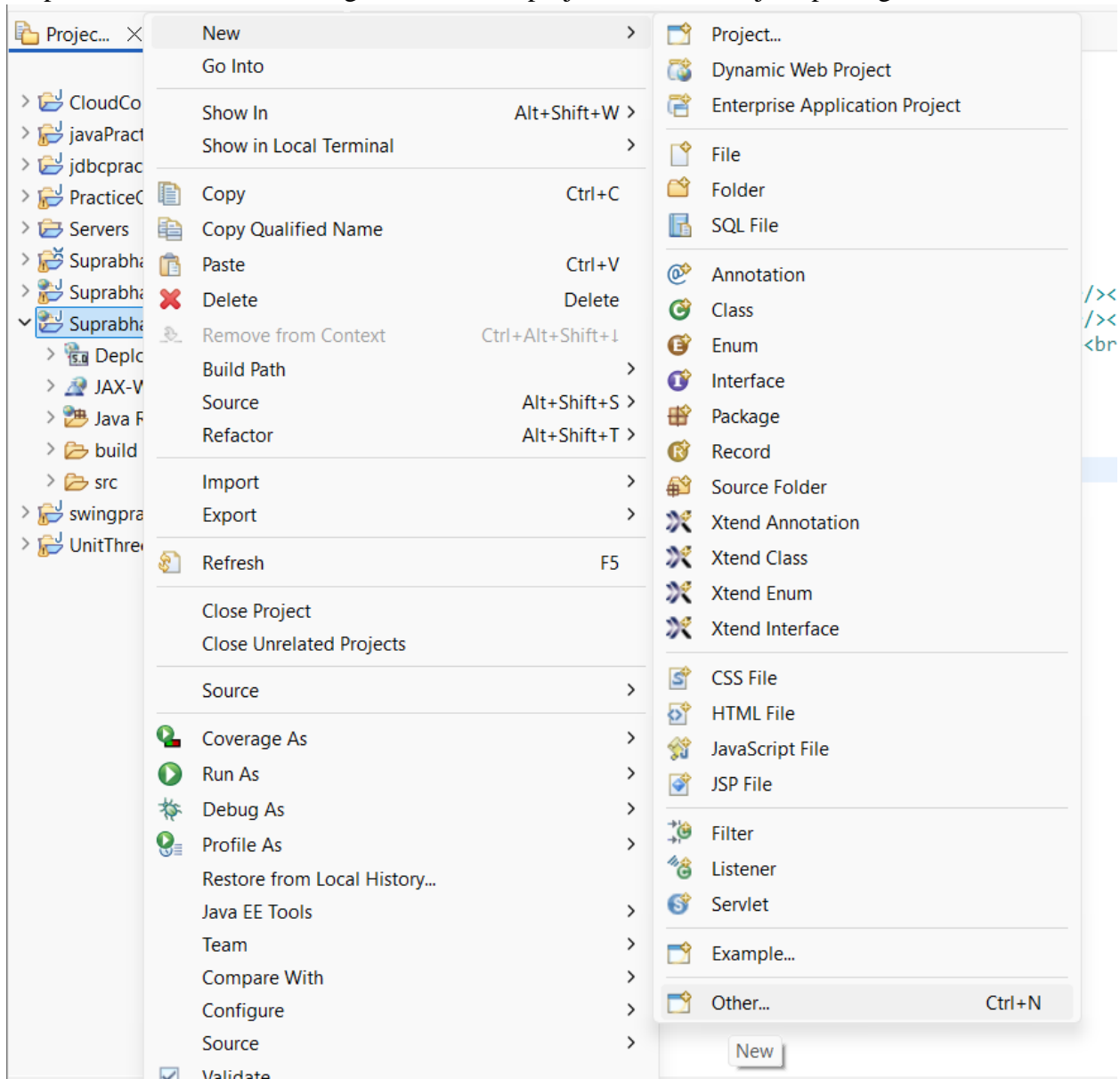
Step 1: Create a new Java Web application project. Go to, File > New > Dynamic Web Project:



Enter the project name and then click finish:



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



## Select a wizard

Create a Java package





## Java Package

Create a new Java package.



Creates folders corresponding to packages.

Source folder:

[Browse...](#)

Name:

☐ Create package-info.java

☐ Generate comments (configure templates and default value [here](#))

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;  
    }
```

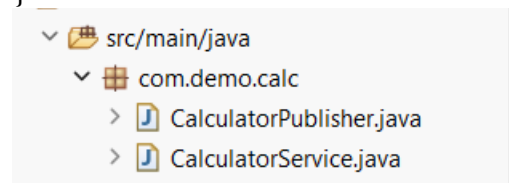
```
    @WebMethod
```

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

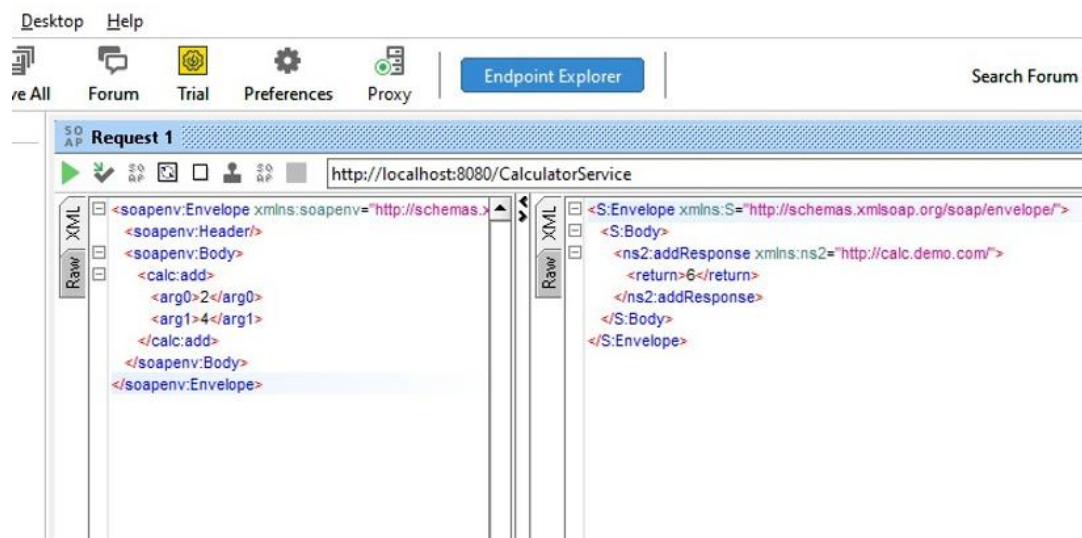
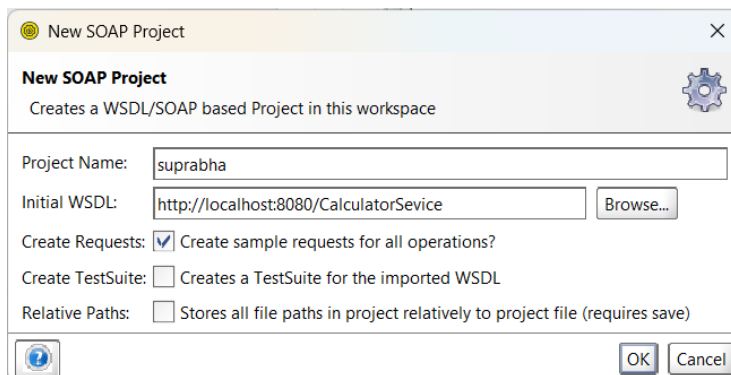
Create CalculatorPublisher.java class with package com.demo.calc;

```
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");  
    }  
}
```



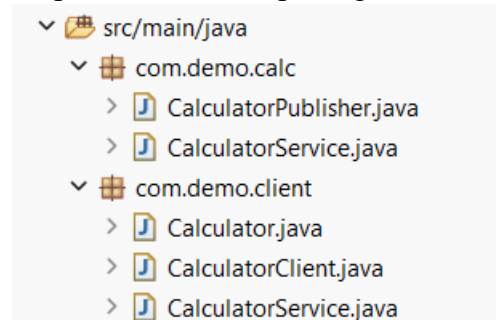
Using SOAP UI to test this file:



### 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;
import javax.jws.soap.SOAPBinding;

@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/",

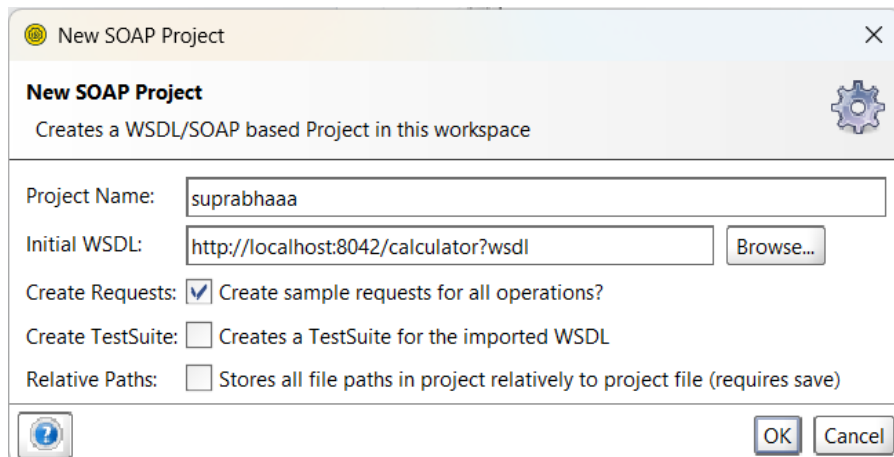
"CalculatorService");
    public CalculatorService(URL wsdlDocumentLocation) {
        super(wsdlDocumentLocation, SERVICE_NAME);
    }

    public Calculator getCalculatorPort() {

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

    }
}
```

Compile the client and test it using SOAP UI



Request 1

http://localhost:8042/CalculatorService

XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <calc:add>
      <arg0>10</arg0>
      <arg1>20</arg1>
    </calc:add>
  </soapenv:Body>
</soapenv:Envelope>
```

Raw

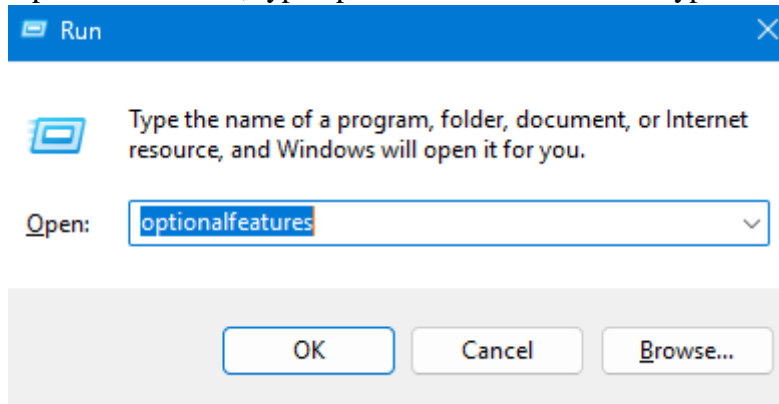
XML

```
<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>
```

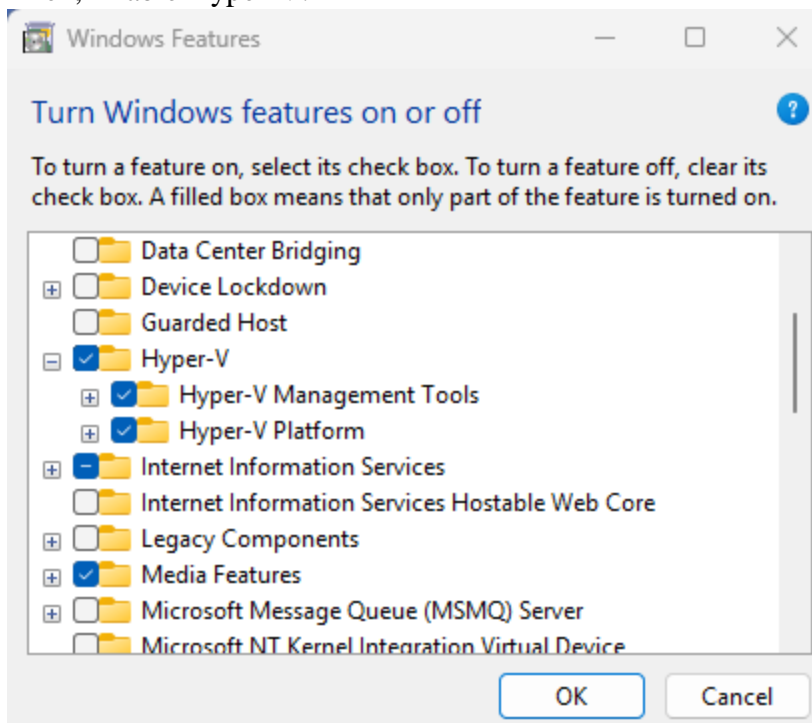
Raw

#### Lab-4 Implement Windows Hyper V virtualization

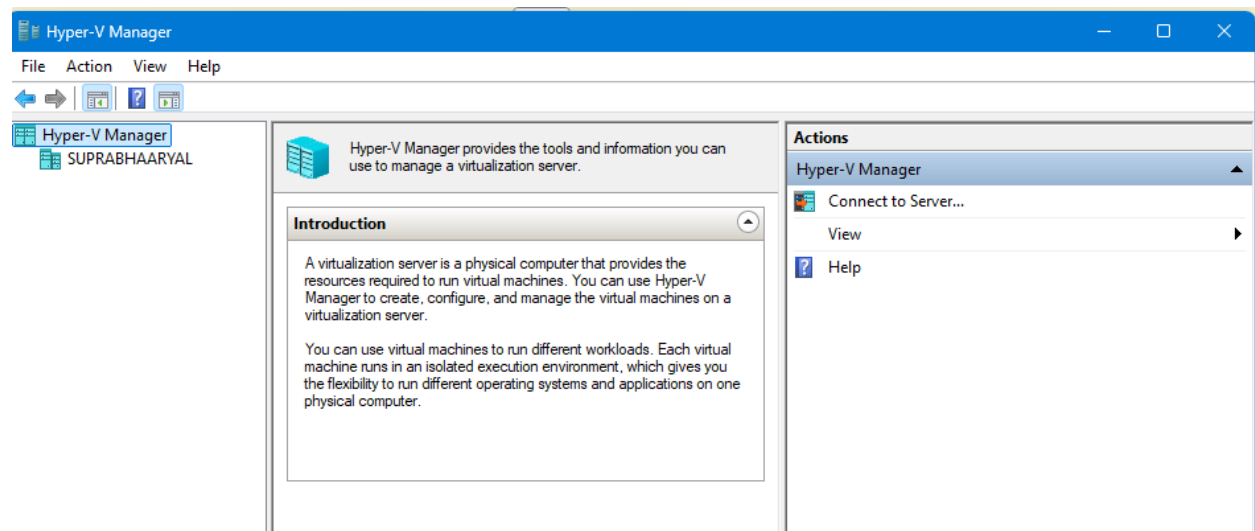
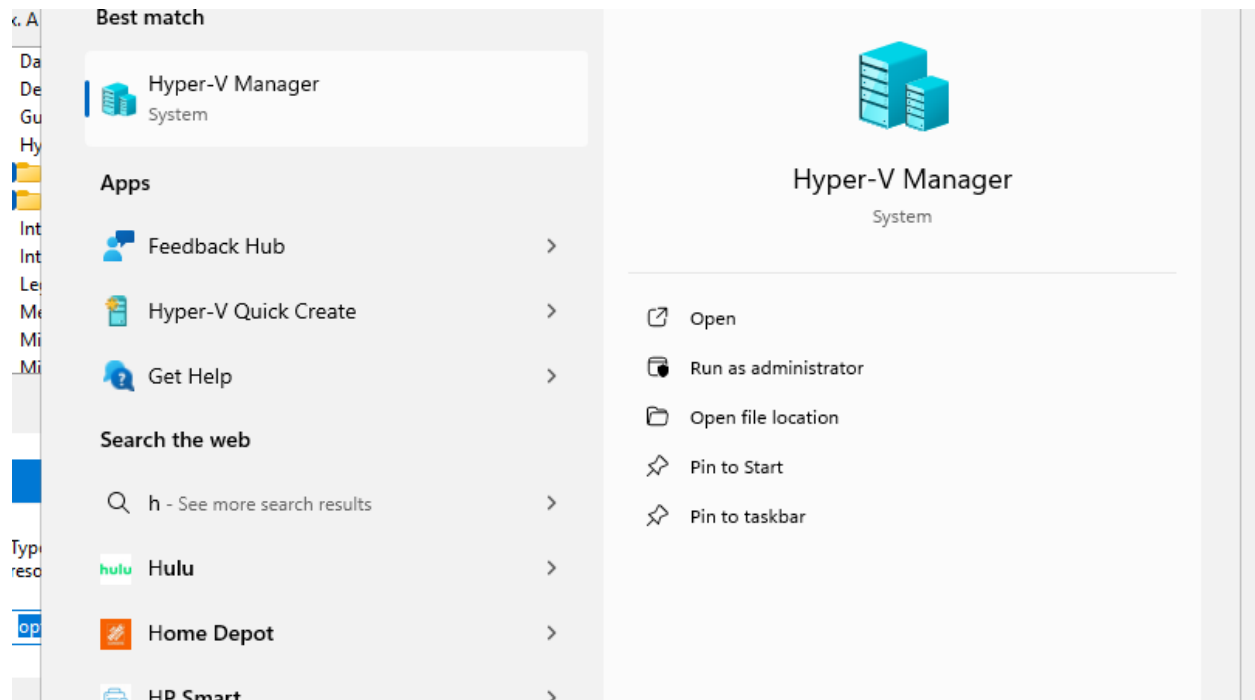
Open windows+R, type optional features to enable Hyper-V:



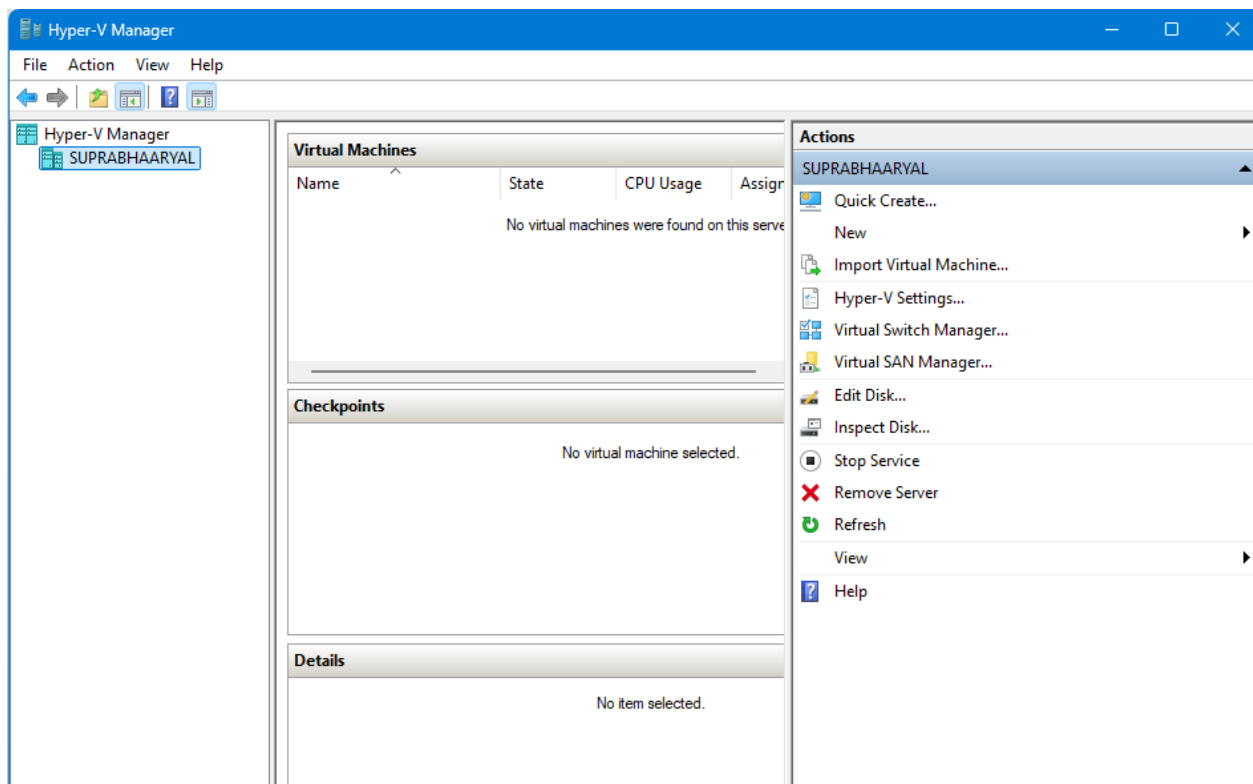
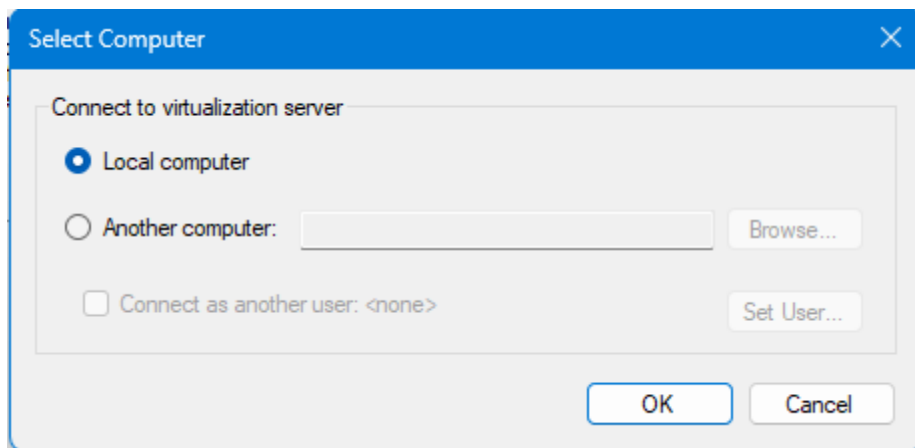
Then, Enable Hyper-V:



## Launch Hyper-V Manager:

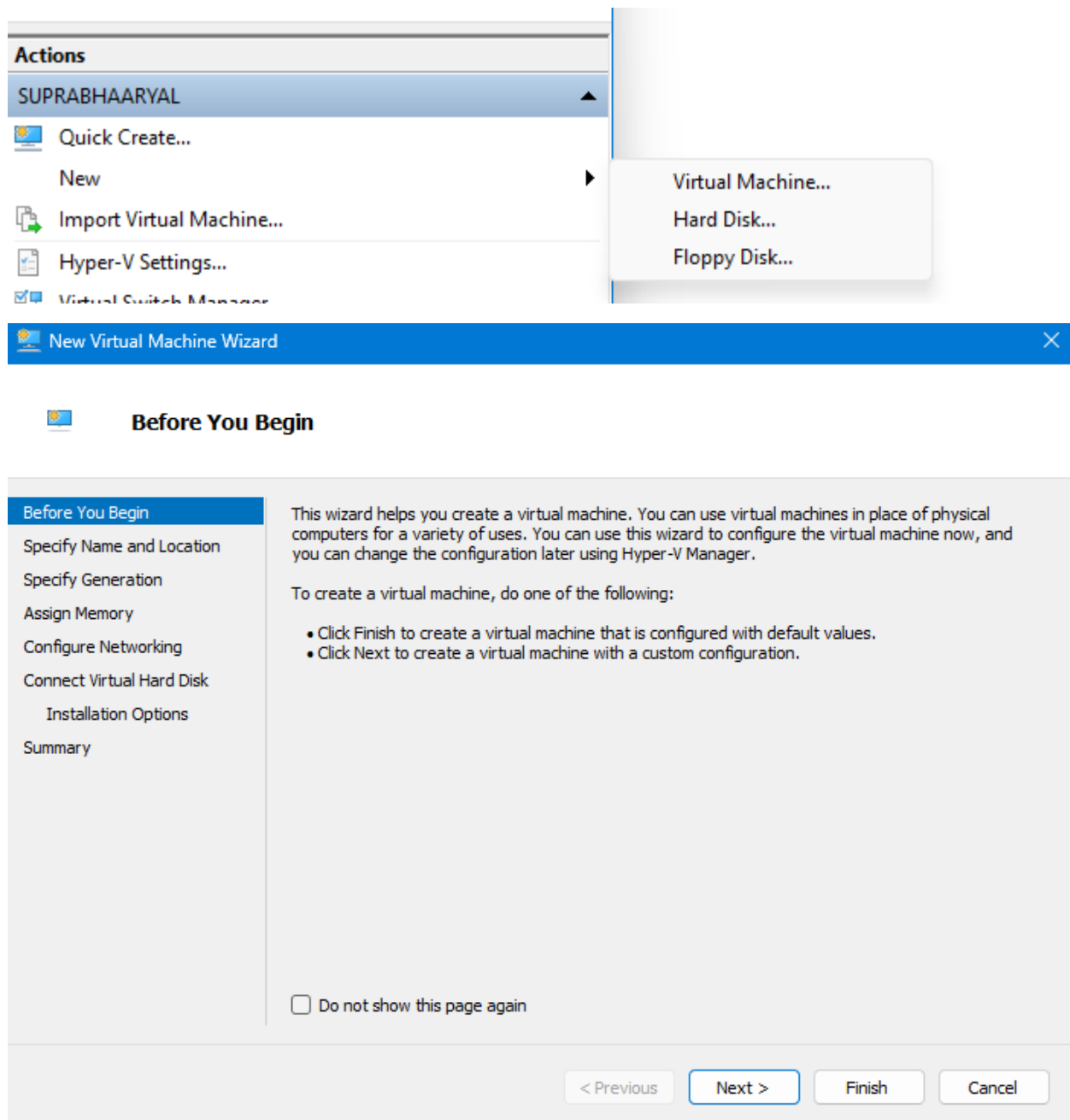


Connect to virtualization server as a Local computer:





Create a new Virtual Machine:



## Specify Name and Location:

The screenshot shows the 'Specify Name and Location' step of the 'New Virtual Machine Wizard'. The left sidebar contains a list of steps: 'Before You Begin', 'Specify Name and Location' (highlighted), 'Specify Generation', 'Assign Memory', 'Configure Networking', 'Connect Virtual Hard Disk', 'Installation Options', and 'Summary'. The main area has the title 'Specify Name and Location' and a sub-header 'Choose a name and location for this virtual machine.' Below this, a text box explains: 'The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.' A text input field for 'Name' contains 'SuprabhaHyper-V'. Another text box explains: 'You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.' There is a checkbox labeled 'Store the virtual machine in a different location' which is currently unchecked. Below it, a text input field for 'Location' contains 'C:\ProgramData\Microsoft\Windows\Hyper-V\' and a 'Browse...' button. A warning icon and text state: 'If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.' At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

New Virtual Machine Wizard

**Specify Name and Location**

Before You Begin  
Specify Name and Location  
Specify Generation  
Assign Memory  
Configure Networking  
Connect Virtual Hard Disk  
Installation Options  
Summary

Choose a name and location for this virtual machine.

The name is displayed in Hyper-V Manager. We recommend that you use a name that helps you easily identify this virtual machine, such as the name of the guest operating system or workload.

Name:

You can create a folder or use an existing folder to store the virtual machine. If you don't select a folder, the virtual machine is stored in the default folder configured for this server.

☐ Store the virtual machine in a different location

Location:  If you plan to take checkpoints of this virtual machine, select a location that has enough free space. Checkpoints include virtual machine data and may require a large amount of space.

< Previous Next > Finish Cancel

## Specify Generation:

The screenshot shows the 'Specify Generation' step of the 'New Virtual Machine Wizard'. The left sidebar contains a list of steps: 'Before You Begin', 'Specify Name and Location', 'Specify Generation' (highlighted), 'Assign Memory', 'Configure Networking', 'Connect Virtual Hard Disk', 'Installation Options', and 'Summary'. The main area has the title 'Specify Generation' and a sub-header 'Choose the generation of this virtual machine.' Below this, there are two radio button options: 'Generation 1' and 'Generation 2'. 'Generation 1' is described as: 'This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.' 'Generation 2' is described as: 'This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.' A warning icon and text state: 'Once a virtual machine has been created, you cannot change its generation.' At the bottom, there are four buttons: '< Previous', 'Next >', 'Finish', and 'Cancel'.

New Virtual Machine Wizard

**Specify Generation**

Before You Begin  
Specify Name and Location  
Specify Generation  
Assign Memory  
Configure Networking  
Connect Virtual Hard Disk  
Installation Options  
Summary

Choose the generation of this virtual machine.

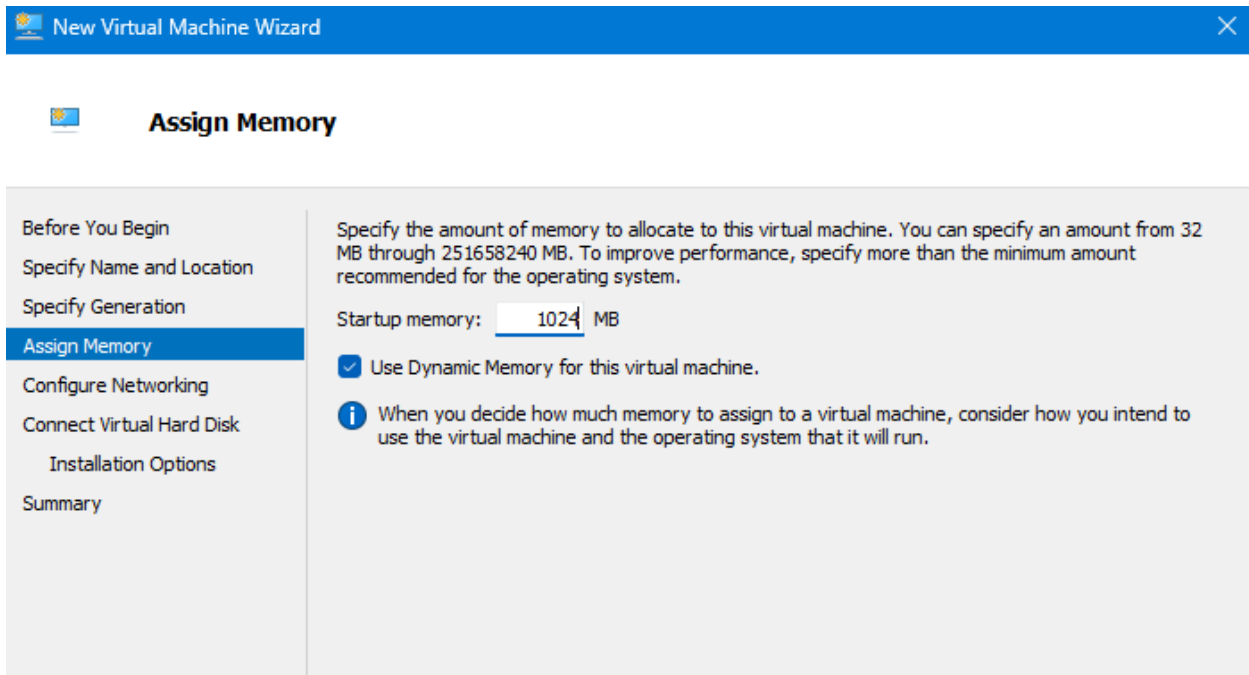
☐ Generation 1  
This virtual machine generation supports 32-bit and 64-bit guest operating systems and provides virtual hardware which has been available in all previous versions of Hyper-V.

☒ Generation 2  
This virtual machine generation provides support for newer virtualization features, has UEFI-based firmware, and requires a supported 64-bit guest operating system.

Once a virtual machine has been created, you cannot change its generation.

< Previous Next > Finish Cancel

Assign memory :



The screenshot shows the 'Assign Memory' step of the 'New Virtual Machine Wizard'. The left sidebar contains a list of steps: 'Before You Begin', 'Specify Name and Location', 'Specify Generation', 'Assign Memory' (highlighted), 'Configure Networking', 'Connect Virtual Hard Disk', 'Installation Options', and 'Summary'. The main area has a title bar 'New Virtual Machine Wizard' with a close button. Below the title bar is a small icon and the title 'Assign Memory'. The main content area explains that the user should specify the amount of memory to allocate, ranging from 32 MB to 251658240 MB. It includes a 'Startup memory' field set to '1024 MB' and a checkbox for 'Use Dynamic Memory for this virtual machine.' which is checked. An information icon and text advise considering the intended use of the virtual machine and the operating system.

**Assign Memory**

Before You Begin  
Specify Name and Location  
Specify Generation  
**Assign Memory**  
Configure Networking  
Connect Virtual Hard Disk  
Installation Options  
Summary

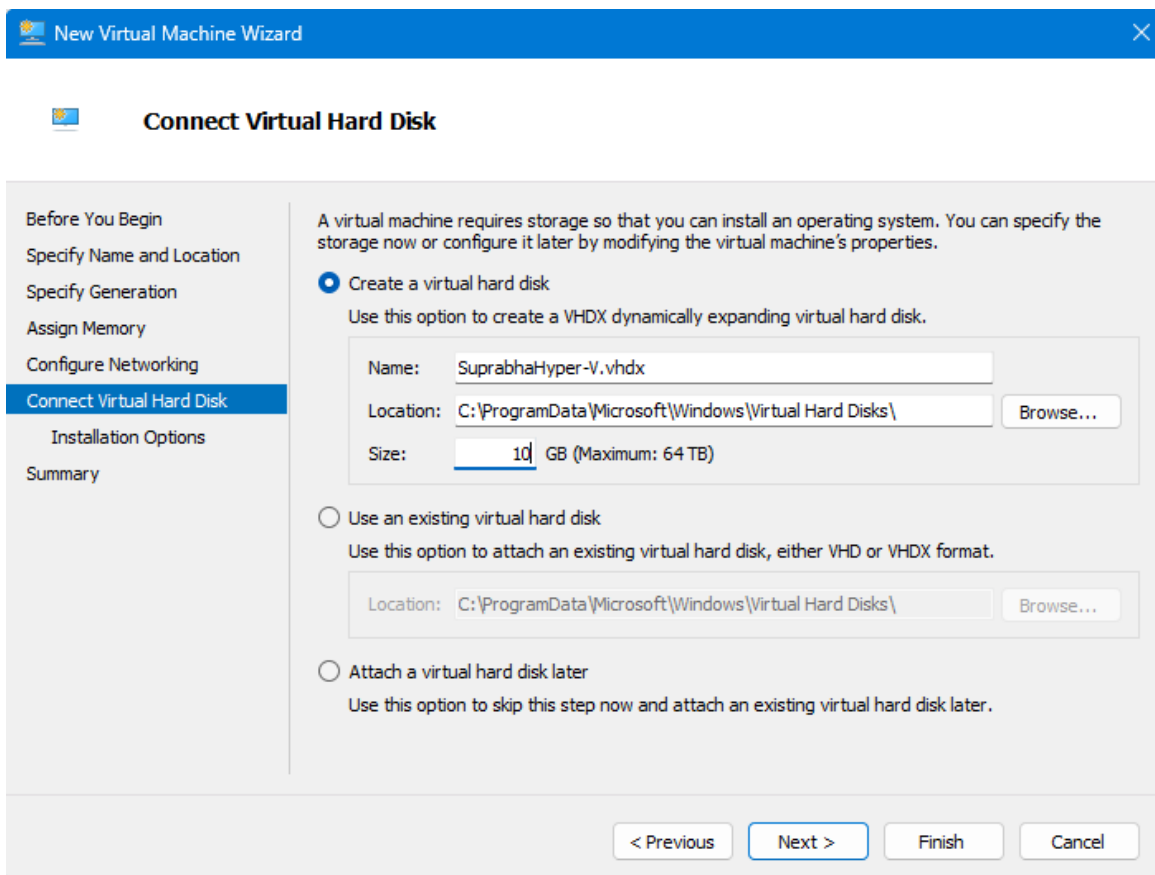
Specify the amount of memory to allocate to this virtual machine. You can specify an amount from 32 MB through 251658240 MB. To improve performance, specify more than the minimum amount recommended for the operating system.

Startup memory:  MB

☒ Use Dynamic Memory for this virtual machine.

**i** When you decide how much memory to assign to a virtual machine, consider how you intend to use the virtual machine and the operating system that it will run.

Setup harddisk:



The screenshot shows the 'Connect Virtual Hard Disk' step of the 'New Virtual Machine Wizard'. The left sidebar contains a list of steps: 'Before You Begin', 'Specify Name and Location', 'Specify Generation', 'Assign Memory', 'Configure Networking', 'Connect Virtual Hard Disk' (highlighted), 'Installation Options', and 'Summary'. The main area has a title bar 'New Virtual Machine Wizard' with a close button. Below the title bar is a small icon and the title 'Connect Virtual Hard Disk'. The main content area explains that a virtual machine requires storage and offers three options: 'Create a virtual hard disk' (selected), 'Use an existing virtual hard disk', and 'Attach a virtual hard disk later'. The 'Create a virtual hard disk' option includes fields for 'Name' (SuprabhaHyper-V.vhdx), 'Location' (C:\ProgramData\Microsoft\Windows\Virtual Hard Disks\), and 'Size' (10 GB, Maximum: 64 TB). The 'Use an existing virtual hard disk' option includes a 'Location' field. The 'Attach a virtual hard disk later' option is also present. At the bottom, there are buttons for '< Previous', 'Next >', 'Finish', and 'Cancel'.

**Connect Virtual Hard Disk**

Before You Begin  
Specify Name and Location  
Specify Generation  
Assign Memory  
Configure Networking  
**Connect Virtual Hard Disk**  
Installation Options  
Summary

A virtual machine requires storage so that you can install an operating system. You can specify the storage now or configure it later by modifying the virtual machine's properties.

☒ Create a virtual hard disk  
Use this option to create a VHDX dynamically expanding virtual hard disk.

Name:

Location:

Size:  GB (Maximum: 64 TB)

☐ Use an existing virtual hard disk  
Use this option to attach an existing virtual hard disk, either VHD or VHDX format.

Location:

☐ Attach a virtual hard disk later  
Use this option to skip this step now and attach an existing virtual hard disk later.



## 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):

- ☐ Install an operating system from a bootable floppy disk

Media

Virtual floppy disk (.vfd):



## 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.

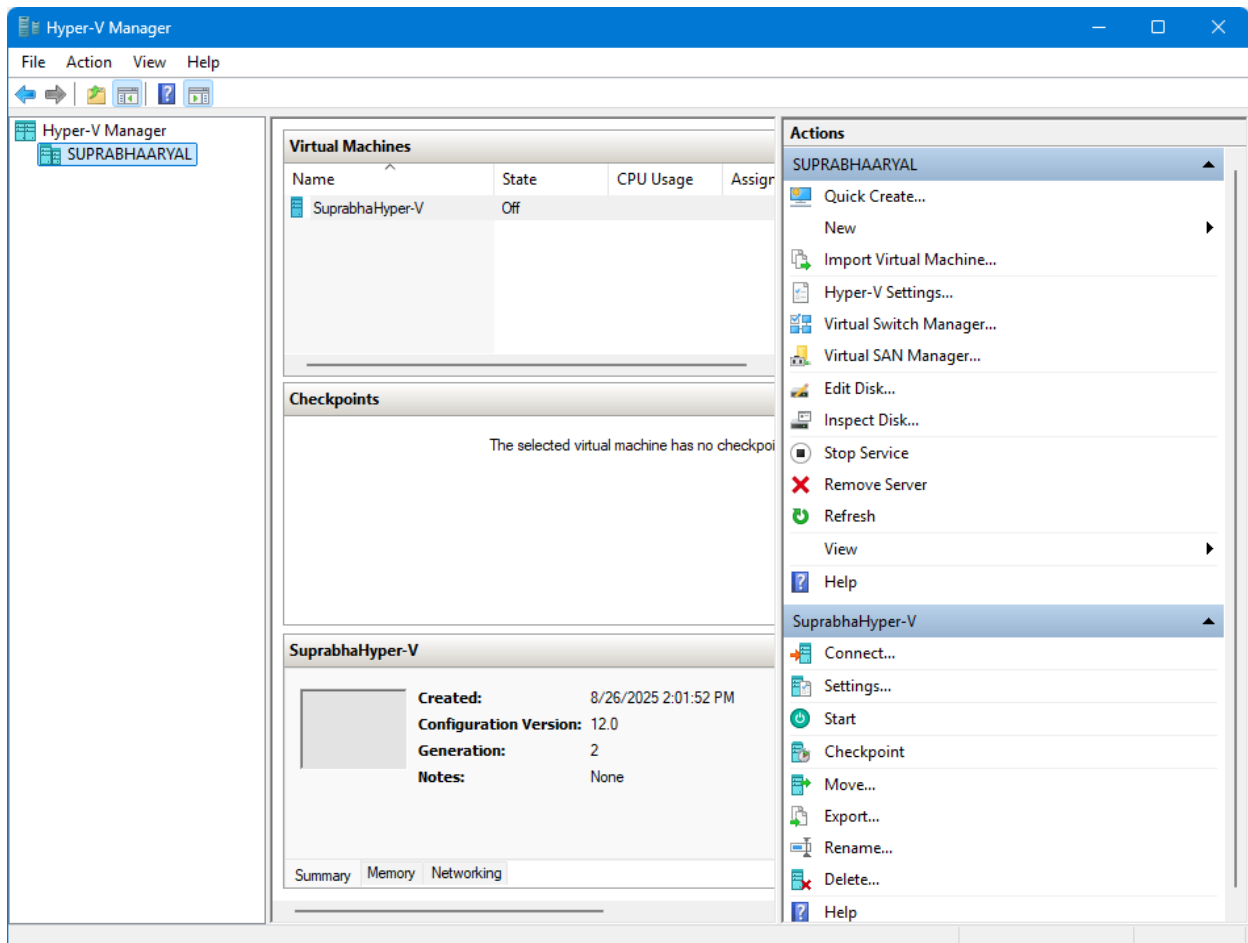
< Previous

Next >

Finish

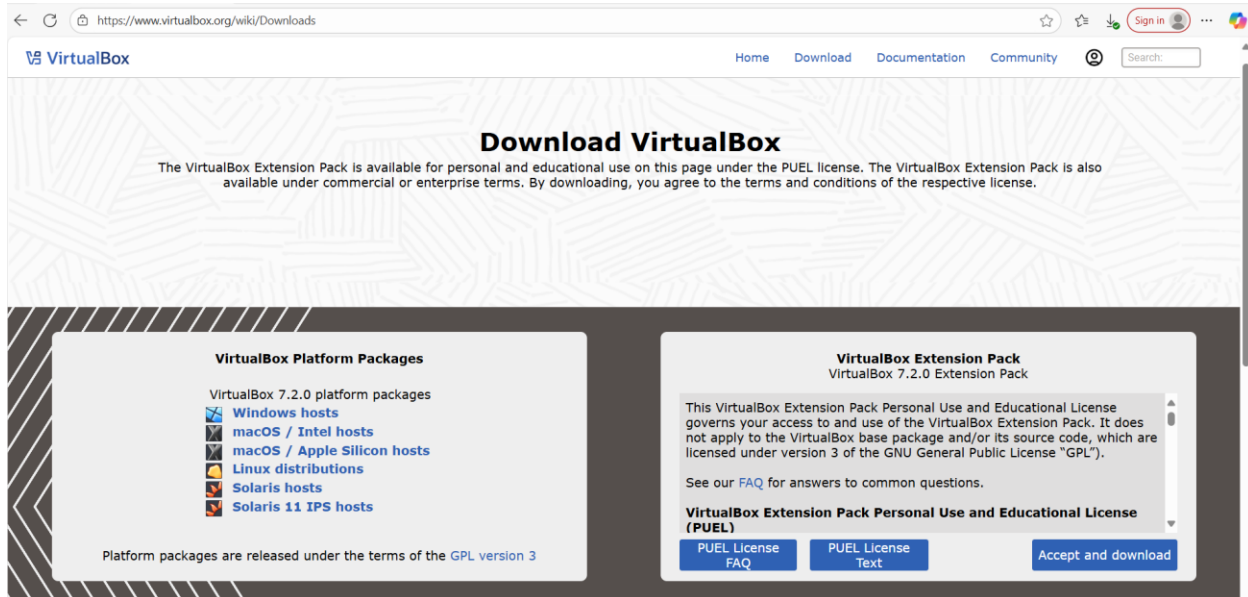
Cancel

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

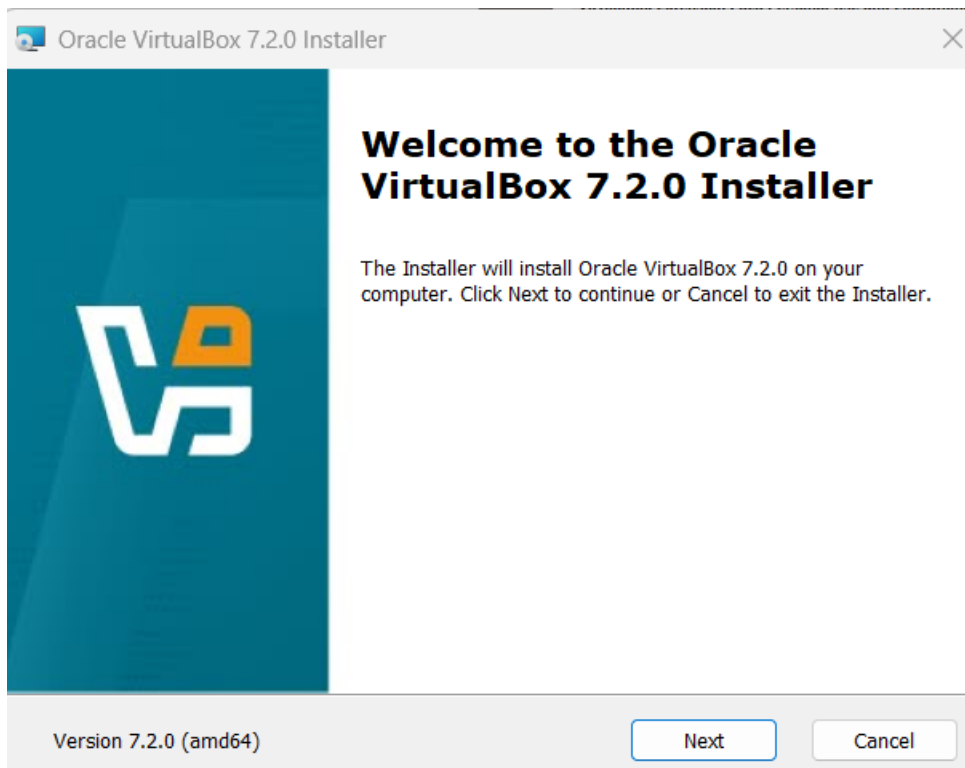


## Lab-5 Implement Virtualization using VirtualBox

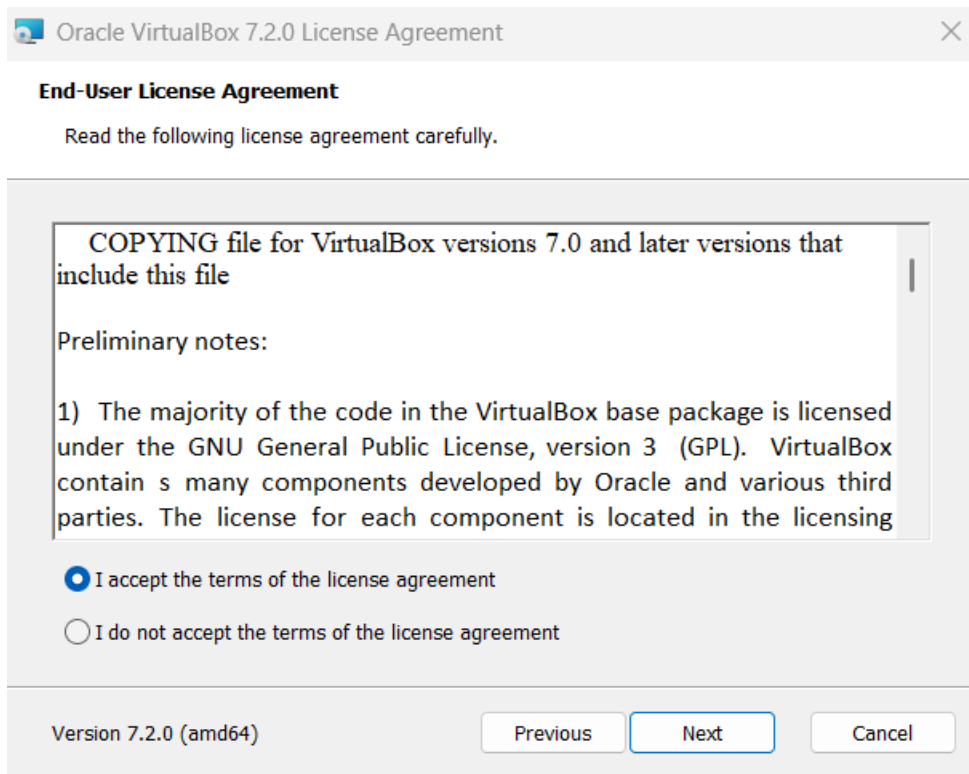
Download Virtualbox exe from <https://www.virtualbox.org/wiki/Downloads> :



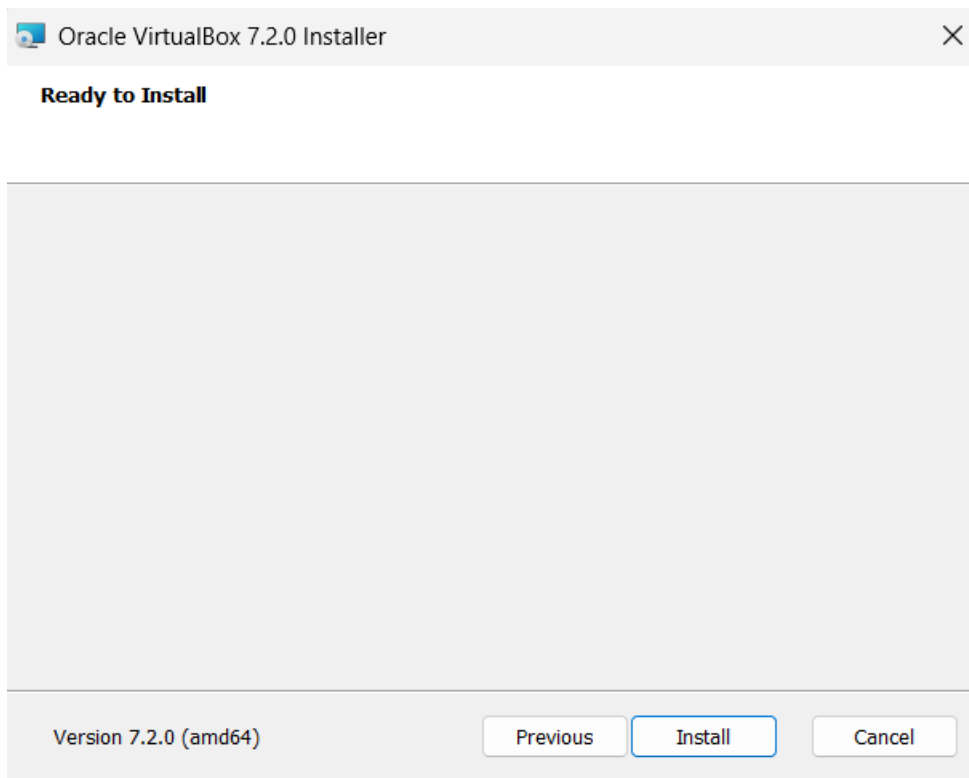
Follow the installation process:

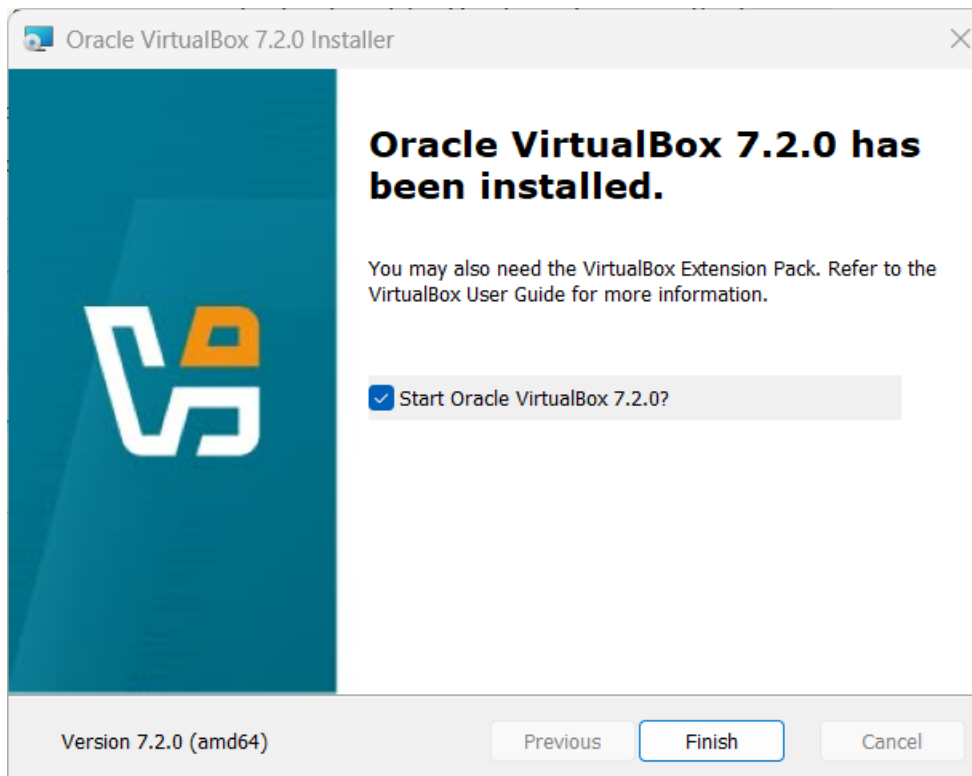


Accept the terms:

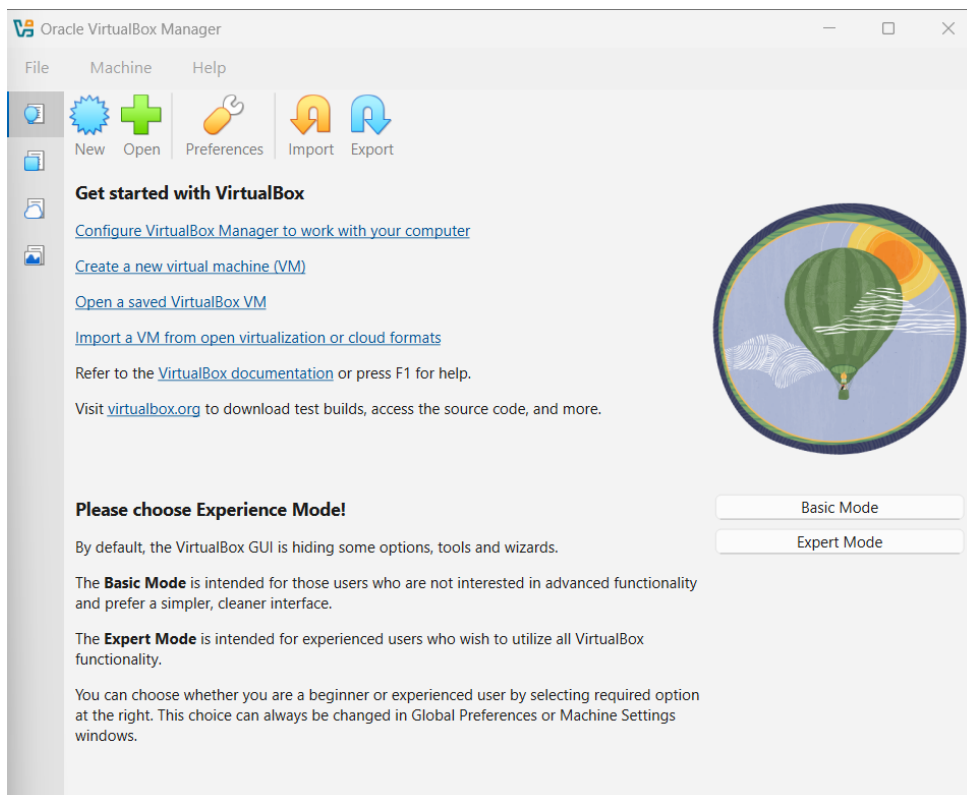


After following the necessary steps, install the application:



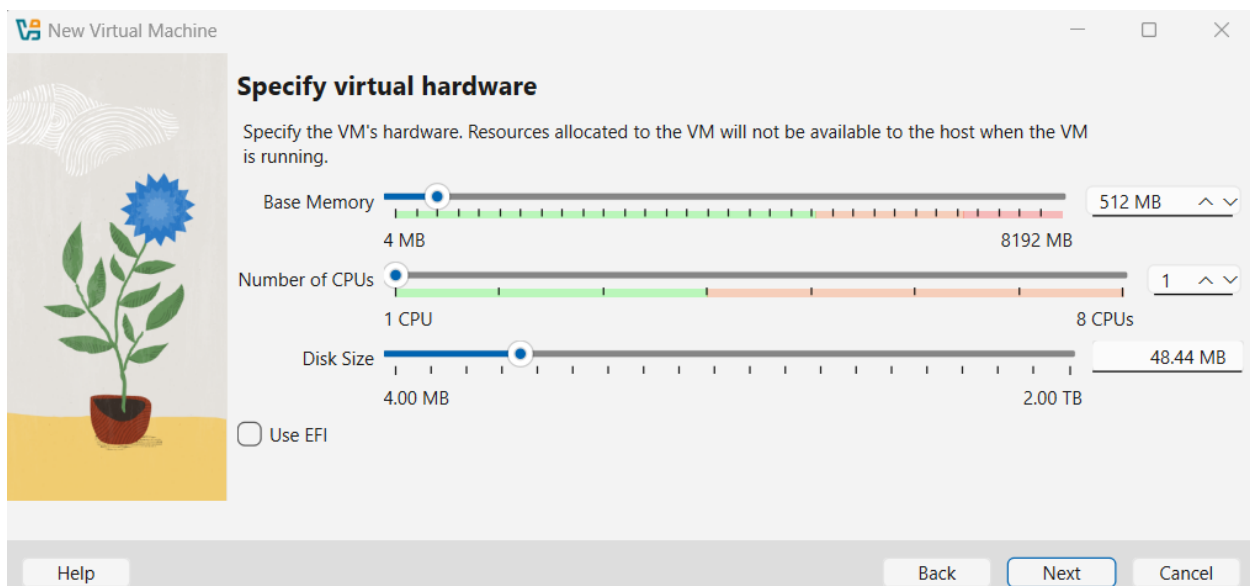
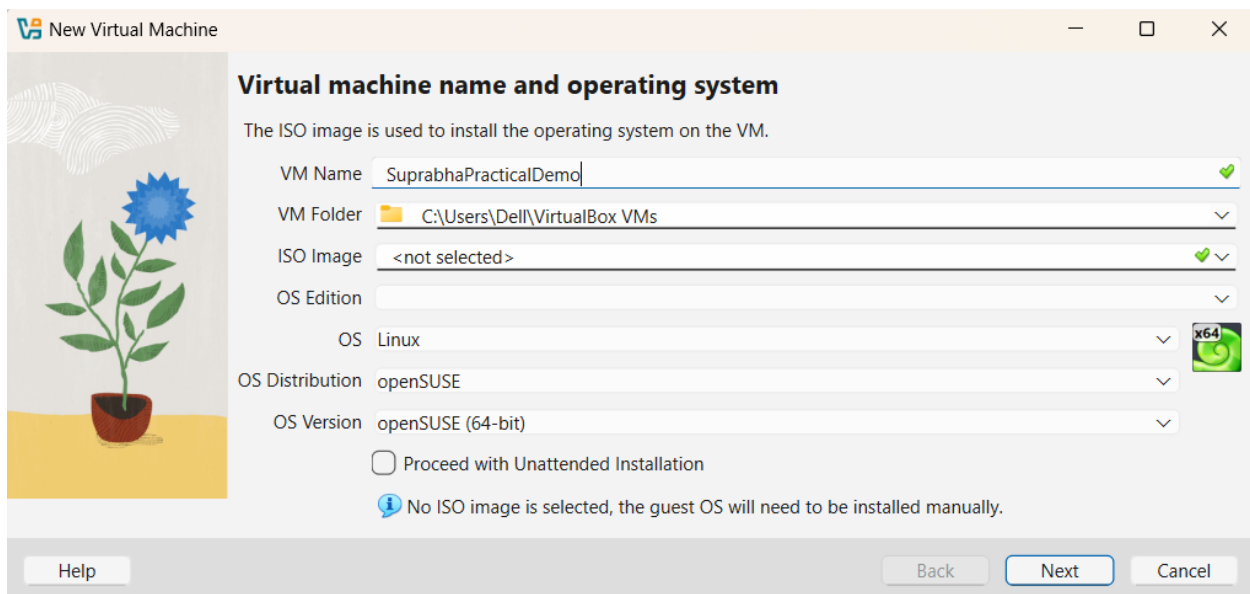
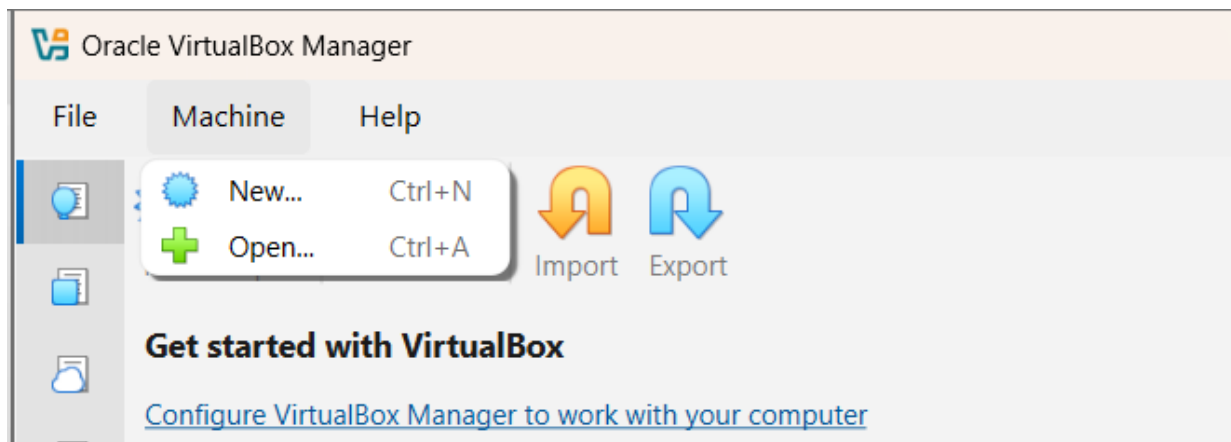


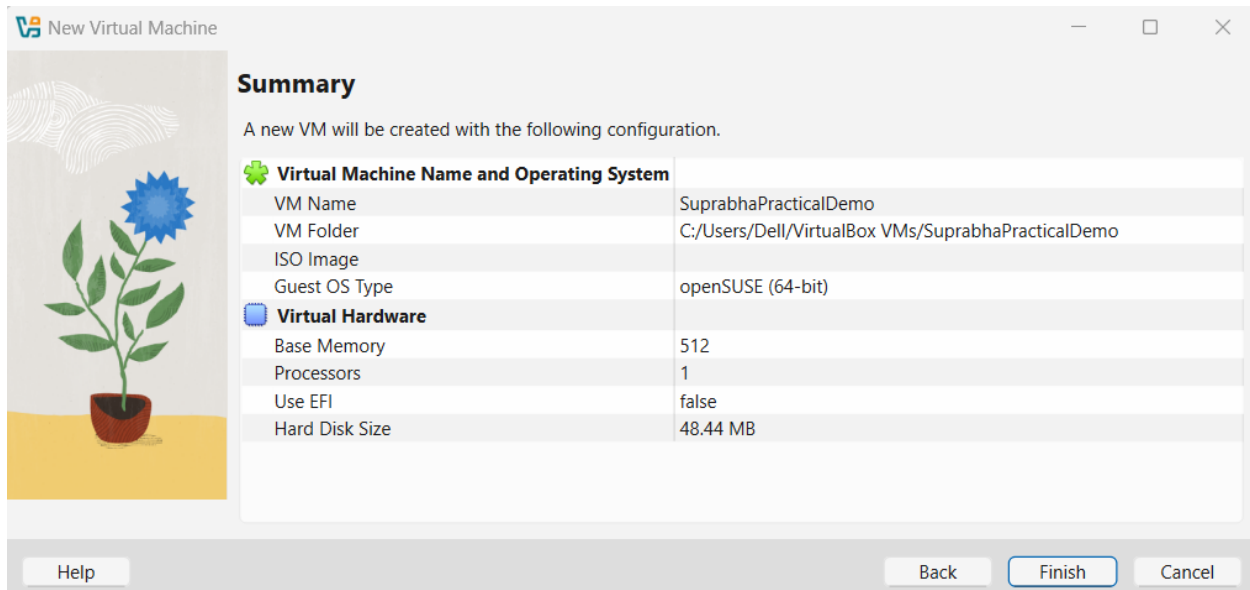
Launch the Application:



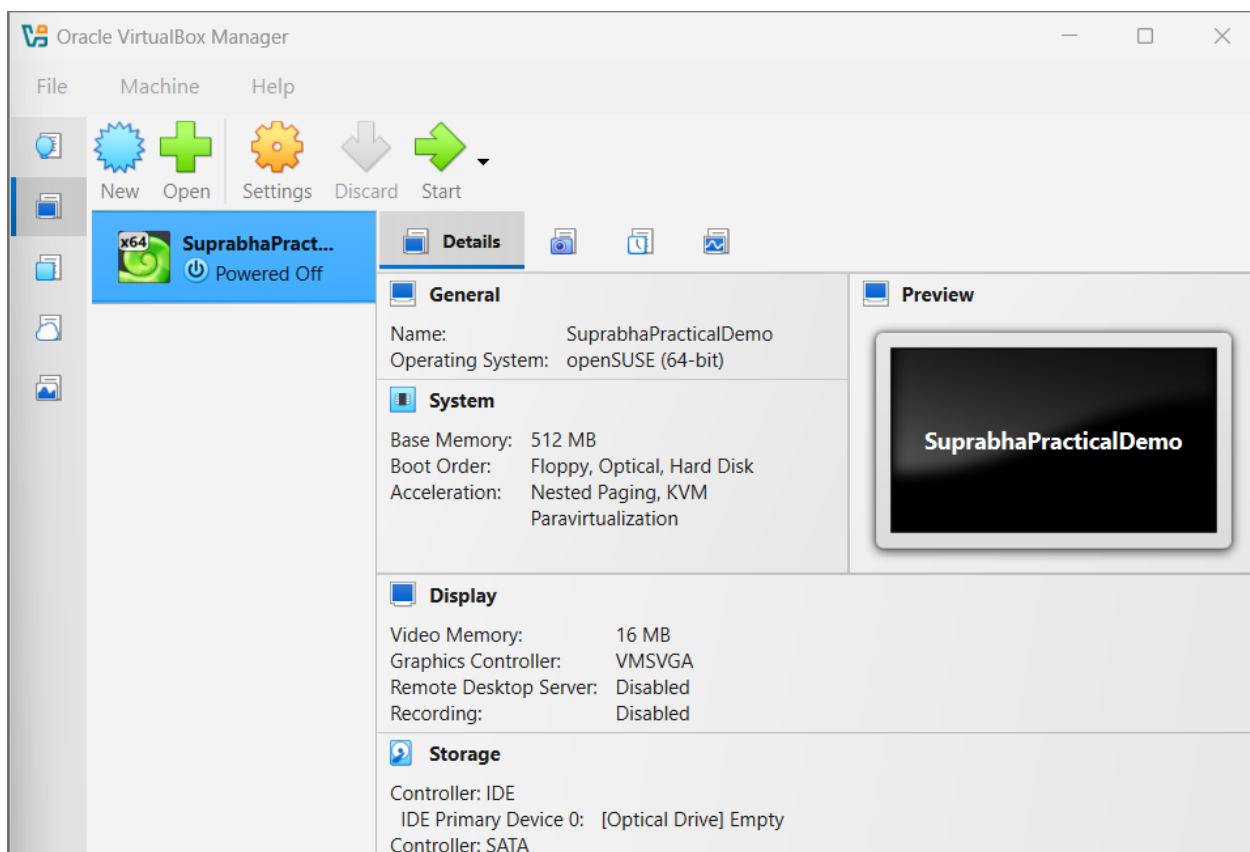


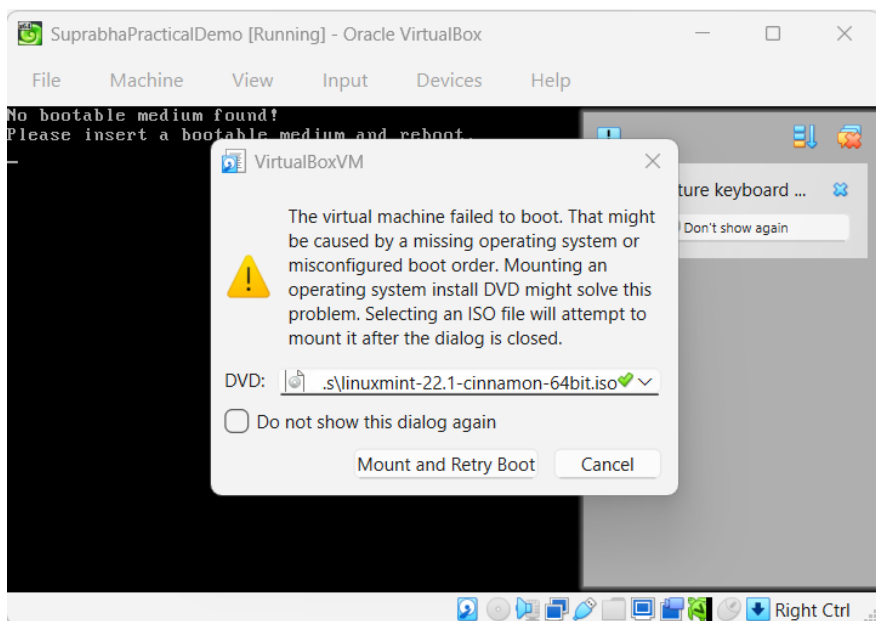
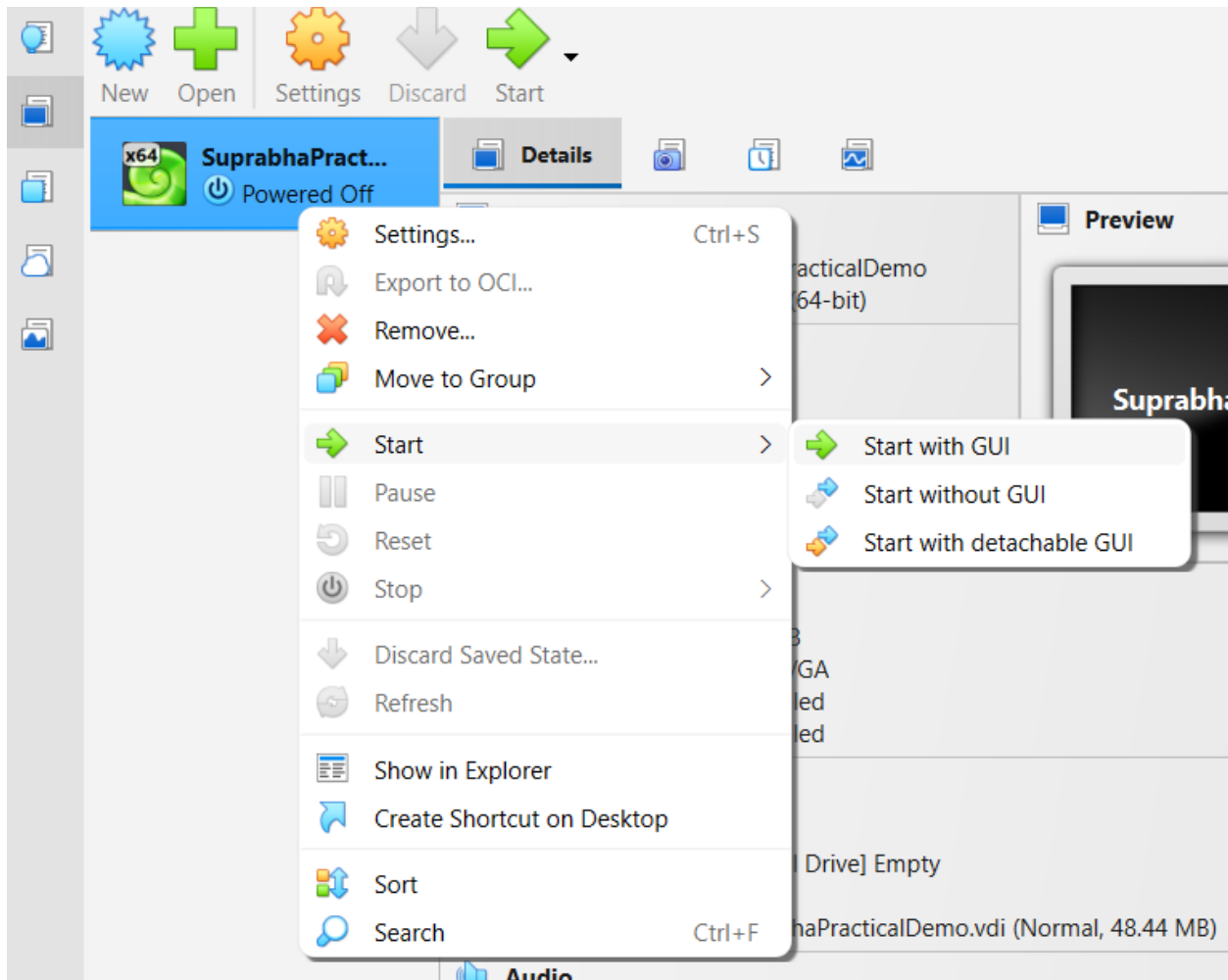
And then create a new virtual machine:

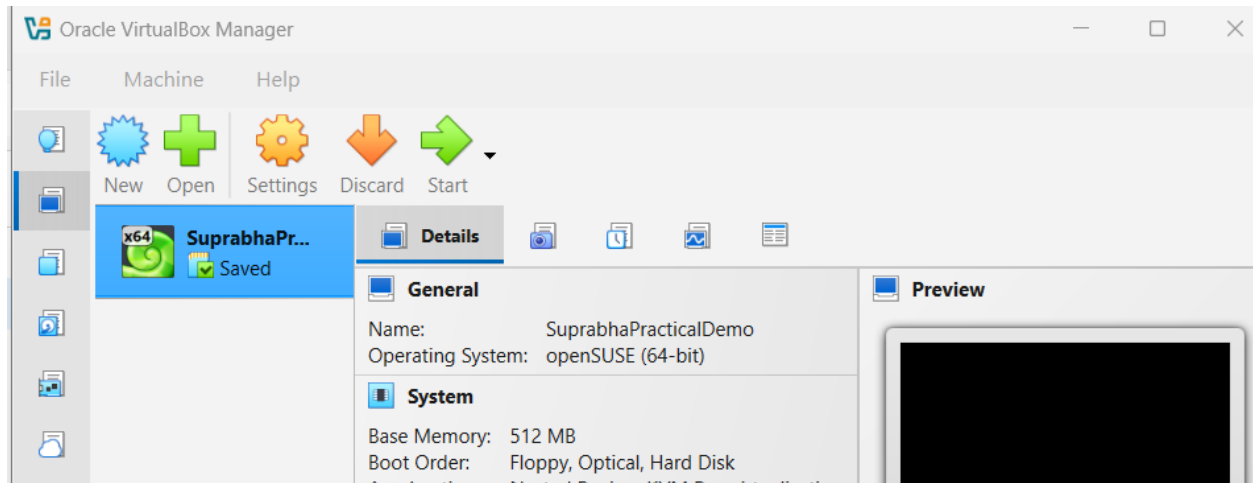
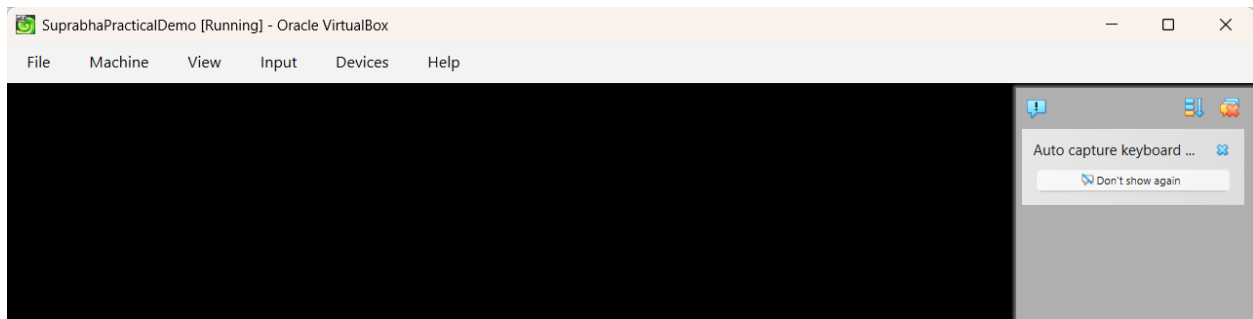




## Start it up:







## Lab-6 Simulate a cloud scenario using CloudSim

Import cloudsims 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.println("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.println("Suprabha's CloudSimProject finished!");

    } catch (Exception e) {
        e.printStackTrace();
        Log.println("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.println("\n===== OUTPUT =====");
    Log.println(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.println(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 ;**
        }
    }
}
}

```

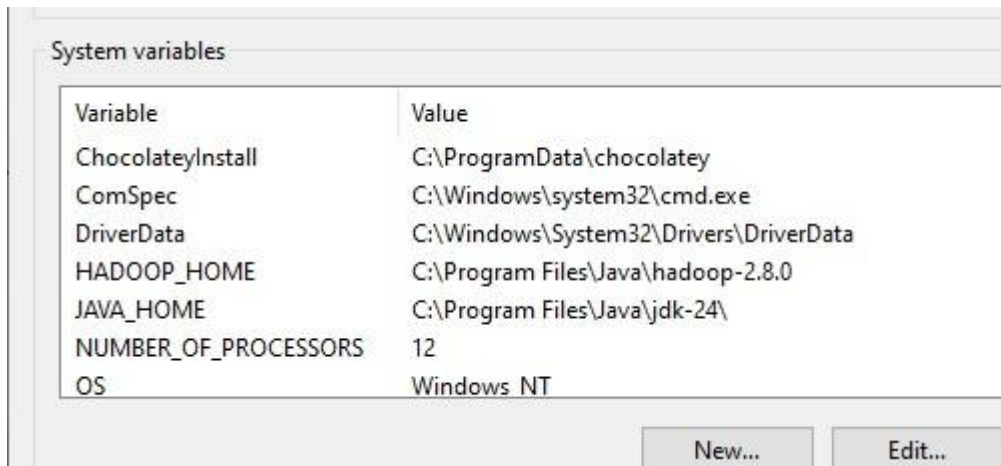


```
Problems Terminal Data Source Explorer Properties Console Coverage TCP/IP Monitor
<terminated> CloudExample (2) [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (Aug 26, 2025, 12:58:38 PM – 12:58:39 PM e
Starting Suprabha's CloudSimProject...
Initialising...
Starting CloudSim version 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.

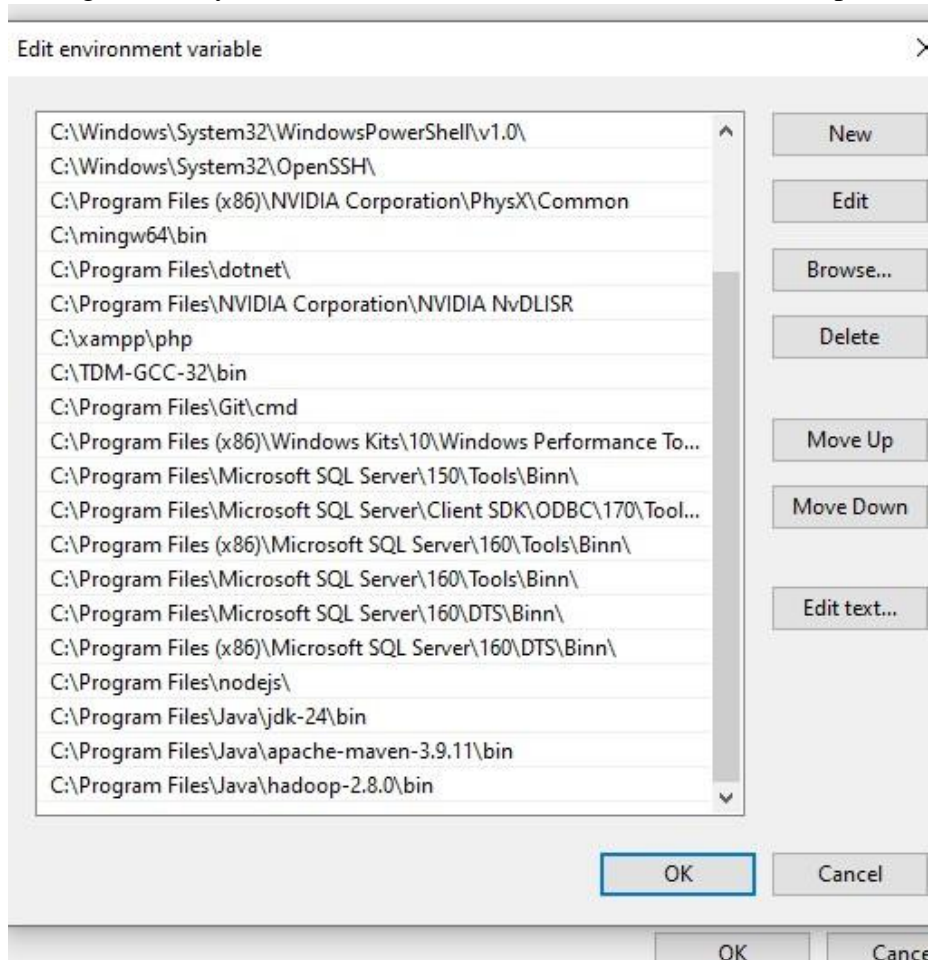
===== OUTPUT =====
Cloudlet ID STATUS Datacenter ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 160 0.2 160.2
1 SUCCESS 3 1 160 0.2 160.2
Suprabha's CloudSimProject finished!
```

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

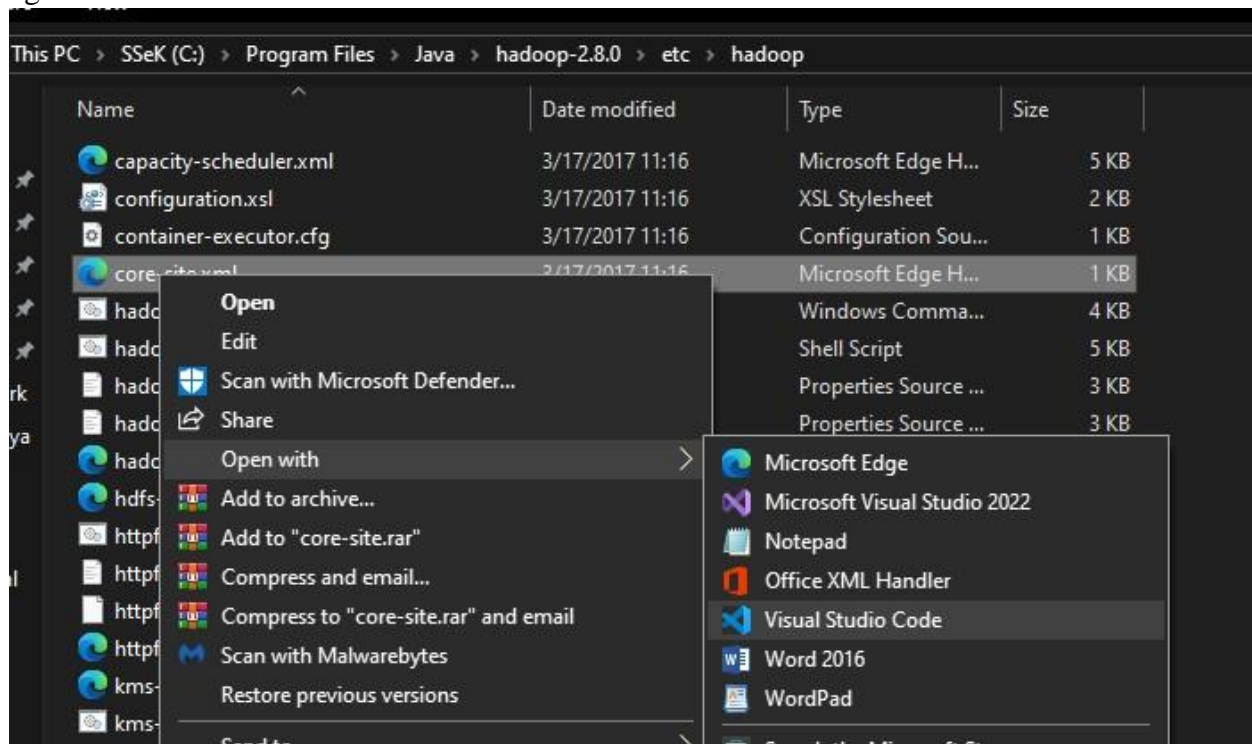
Download Hadoop from <http://archive.apache.org/dist/hadoop/core/hadoop-2.8.0/hadoop-2.8.0.tar.gz> and then, install and setup JDK and Hadoop set up system variables for both:



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>

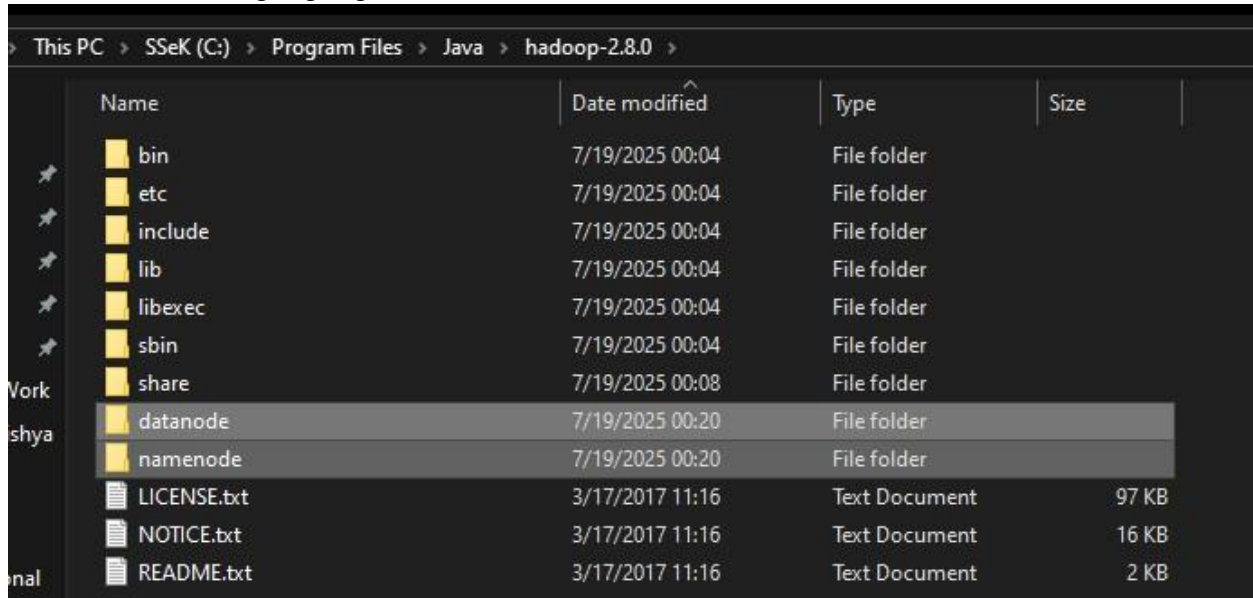
Find `mapred-site.xml.template` and modify it to `mapred-site.xml` and add :

<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>
```

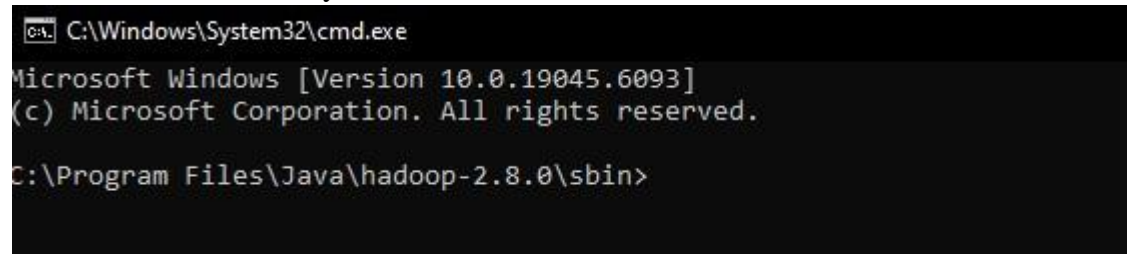
Edit the file Hadoop-2.8.0\etc\hadoop\hadoop-env.cmd and write @rem in front of “set JAVA\_HOME=%JAVA\_HOME%”. Write set JAVA\_HOME={JDK directory} at the next row. This is C:\Program Files\Java\jdk-24 for me

Download Hadoop Configuration.zip from

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

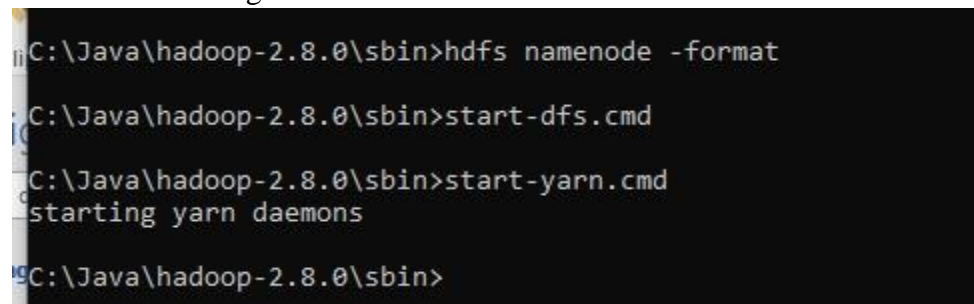
And replace the bin from Hadoop-2.8.0 with bin from configuration

Run cmd at this directory



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 10.0.19045.6093]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Program Files\Java\hadoop-2.8.0\sbin>
```

Run the following commands



```
C:\Java\hadoop-2.8.0\sbin>hdfs namenode -format  
  
C:\Java\hadoop-2.8.0\sbin>start-dfs.cmd  
  
C:\Java\hadoop-2.8.0\sbin>start-yarn.cmd  
starting yarn daemons  
  
C:\Java\hadoop-2.8.0\sbin>
```

In browser go to localhost:8088



# All Ap

▼ Cluster

[About](#)  
[Nodes](#)  
[Node Labels](#)  
[Applications](#)  
[NEW](#)  
[NEW SAVING](#)  
[SUBMITTED](#)  
[ACCEPTED](#)  
[RUNNING](#)

## Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running
0	0	0	0	0

## Cluster Nodes Metrics

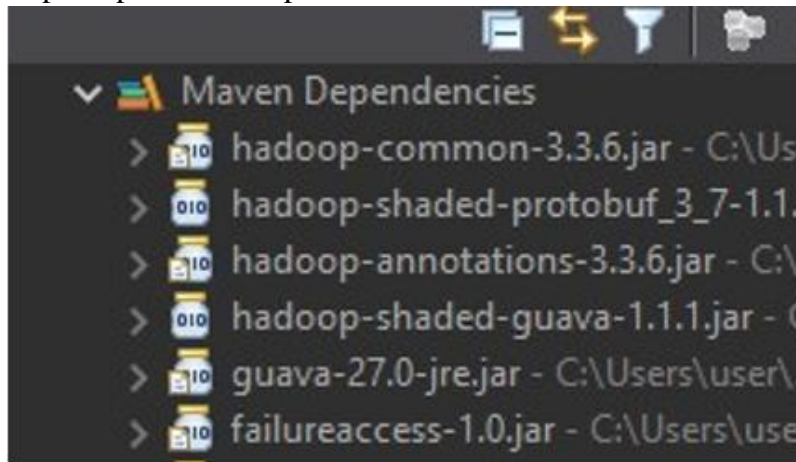
Active Nodes	Decommissioning Nodes	Decommissioned Nodes
0	0	0

## Scheduler Metrics

--	--	--	--	--

## 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]));

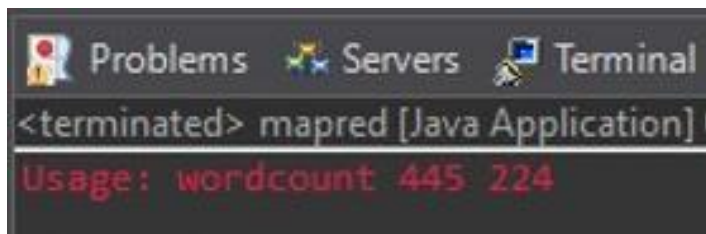
    System.exit(job.waitForCompletion(true) ? 0 : 1);
}

```



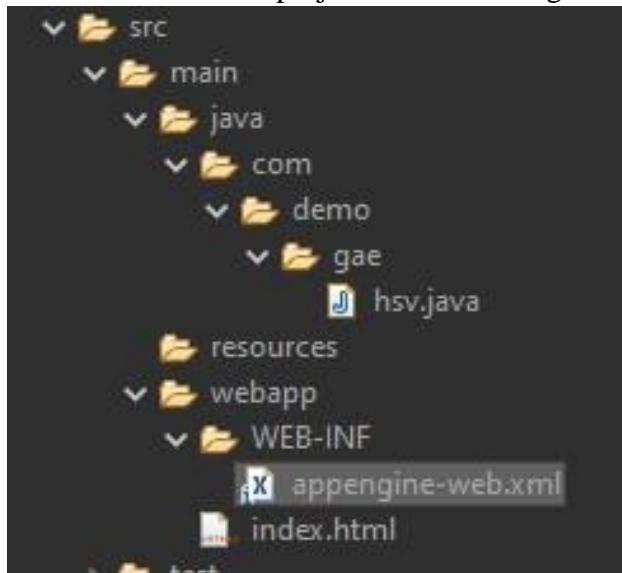
}

}



## 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>

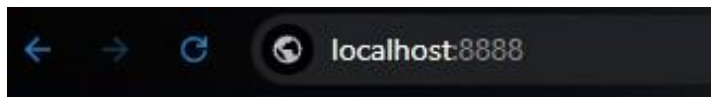
    <h1>Hello from Java 17 on App Engine!</h1>

    <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](#)

