



**RAMNIRANJAN JHUNJHUNWALA COLLEGE  
GHATKOPAR (W), MUMBAI - 400 086**

**DEPARTMENT OF INFORMATION TECHNOLOGY  
2021 - 2022**

**M.Sc. (I.T.) SEM I  
CLOUD COMPUTING**

**NAME: KARAN CHAVARE  
ROLL NO: 37**



Hindi Vidya Prachar Samiti's  
**RAMNIRANJAN**  
**JHUNJHUNWALA COLLEGE**  
**(AUTONOMOUS)**



Opposite Ghatkopar Railway Station, Ghatkopar West, Mumbai-400086

### **CERTIFICATE**

This is to certify that Mr./Miss/Mrs. **KARAN BAJRANG CHAVARE** with **Seat No: 37** has successfully completed the necessary course of experiments in the subject of **Cloud Computing** during the academic year 2021 – 2022 complying with the requirements of RAMNIRANJAN JHUNJHUNWALA COLLEGE OF ARTS, SCIENCE AND COMMERCE, for the course of M.Sc. (IT) Semester -I.

---

Internal Examiner

---

Date

---

Head Of Department

---

College Seal

---

External Examiner

# INDEX

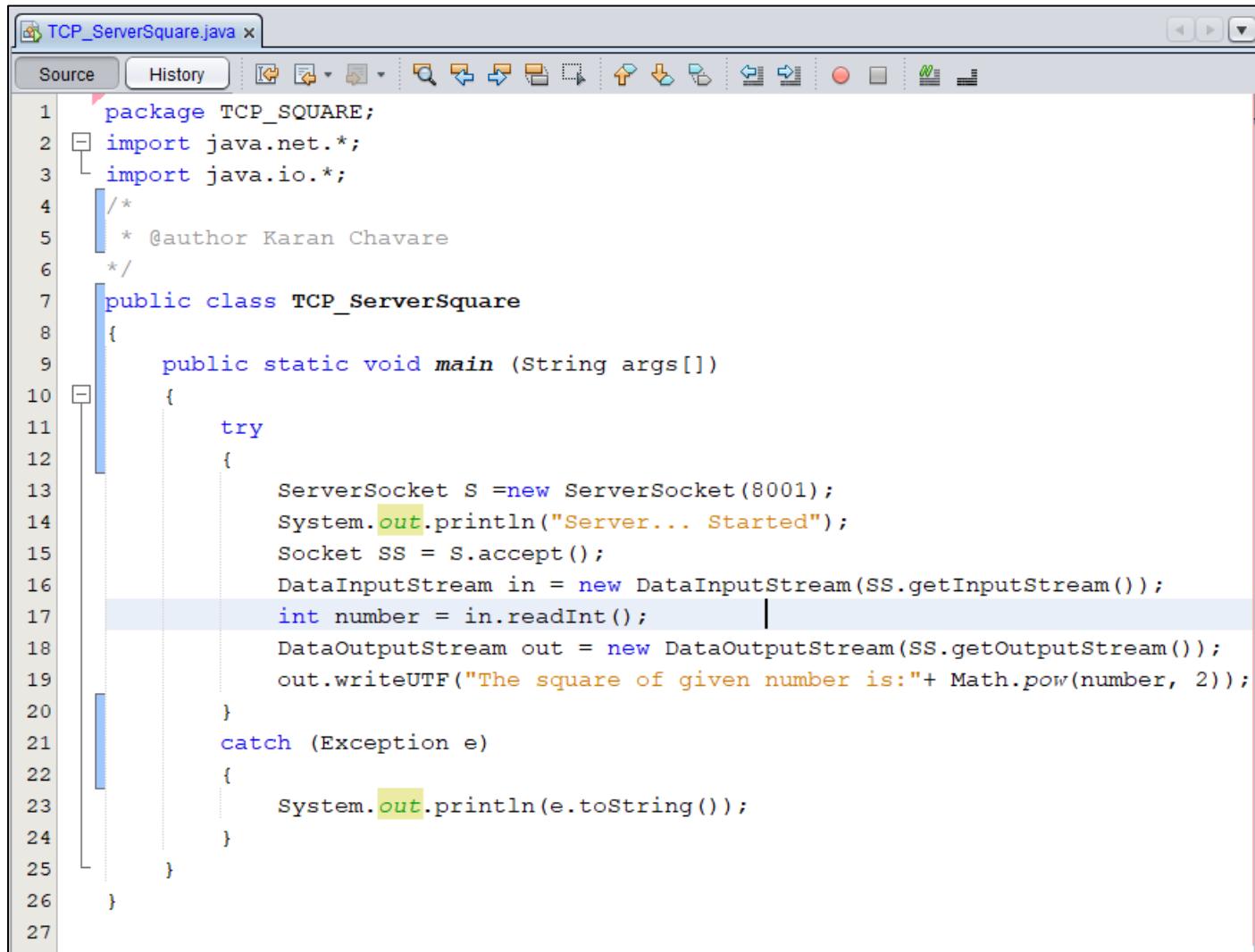
Sr No	Practical	Date
1	Implement Client Server Communication model using TCP.	28/08/2021
2	Implement Client Server Communication model using UDP.	28/08/2021
3	Implementation of Web Services.	04/09/2021
4	Develop Application for Google App Engine.	18/09/2021
5	Implement Virtualization Using VMWare ESXI Server and managing with vSphere Client.	18/09/2021
6	Implement Virtualization Using Hyper-V	09/10/2021
7	Develop a Cloud Application for Microsoft Azure.	16/10/2021

# PRACTICAL - 1

## ❖ **Implement Client Server Communication Model Using TCP.**

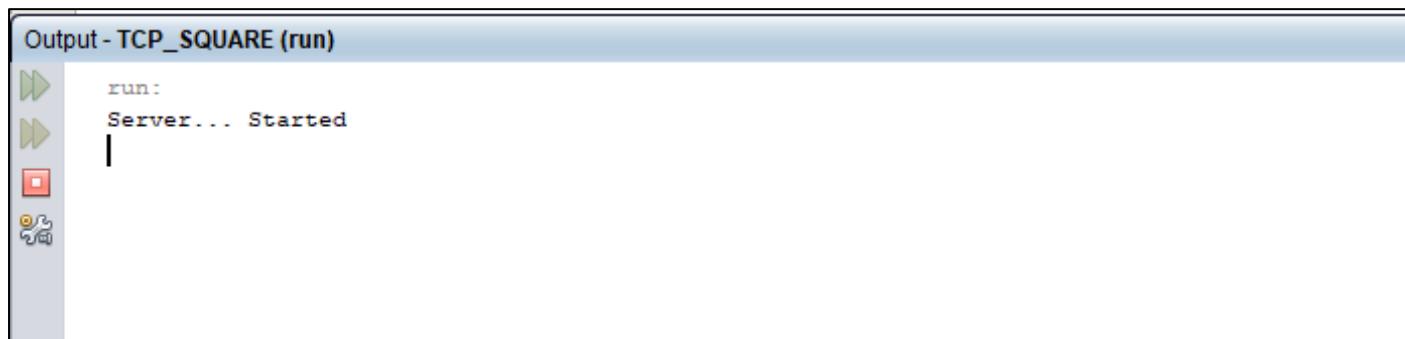
- Write a client program to enter the number and server program to calculate the square of the entered number using TCP Communication.

### TCP\_ClientSquare.java



The screenshot shows a Java code editor window titled "TCP\_ServerSquare.java". The code implements a TCP server that listens on port 8001, reads an integer from the client, calculates its square using Math.pow(number, 2), and then sends the result back to the client via DataOutputStream. The code uses ServerSocket, Socket, DataInputStream, and DataOutputStream classes.

```
1 package TCP_SQUARE;
2 import java.net.*;
3 import java.io.*;
4 /*
5  * @author Karan Chavare
6  */
7 public class TCP_ServerSquare
8 {
9     public static void main (String args[])
10    {
11        try
12        {
13            ServerSocket S =new ServerSocket(8001);
14            System.out.println("Server... Started");
15            Socket SS = S.accept();
16            DataInputStream in = new DataInputStream(SS.getInputStream());
17            int number = in.readInt();
18            DataOutputStream out = new DataOutputStream(SS.getOutputStream());
19            out.writeUTF("The square of given number is:"+ Math.pow(number, 2));
20        }
21        catch (Exception e)
22        {
23            System.out.println(e.toString());
24        }
25    }
26 }
27
```



The screenshot shows the "Output - TCP\_SQUARE (run)" window. It displays the text "run:" followed by "Server... Started", indicating that the server has started listening on port 8001.

Output - TCP\_SQUARE (run)

```
run:
Server... Started
```

## TCP ServerSquare.java

The screenshot shows the source code for `TCP_ClientSquare.java`. The code implements a client that connects to a server at `localhost` port `8001`. It reads a number from the user, sends it to the server, and receives the square of the number back. The code uses `InputStreamReader`, `BufferedReader`, `DataOutputStream`, and `DataInputStream` for communication.

```
1 package TCP_SQUARE;
2 import java.net.*;
3 import java.io.*;
4 /*
5 * @author Karan Chavare
6 */
7 public class TCP_Clientsquare
8 {
9     public static void main(String args[])
10    {
11        try
12        {
13            Socket S = new Socket("LocalHost", 8001);
14            InputStreamReader Ir = new InputStreamReader(System.in);
15            BufferedReader br =new BufferedReader(Ir);
16            System.out.println("Enter a number");
17            int number = Integer.parseInt(br.readLine());
18            DataOutputStream out =new DataOutputStream(S.getOutputStream());
19            out.writeInt(number);
20            DataInputStream in = new DataInputStream(S.getInputStream());
21            System.out.println(in.readUTF());
22            S.close();
23        }
24        catch (Exception e)
25        {
26            System.out.println(e.toString());
27        }
28    }
29 }
30
```

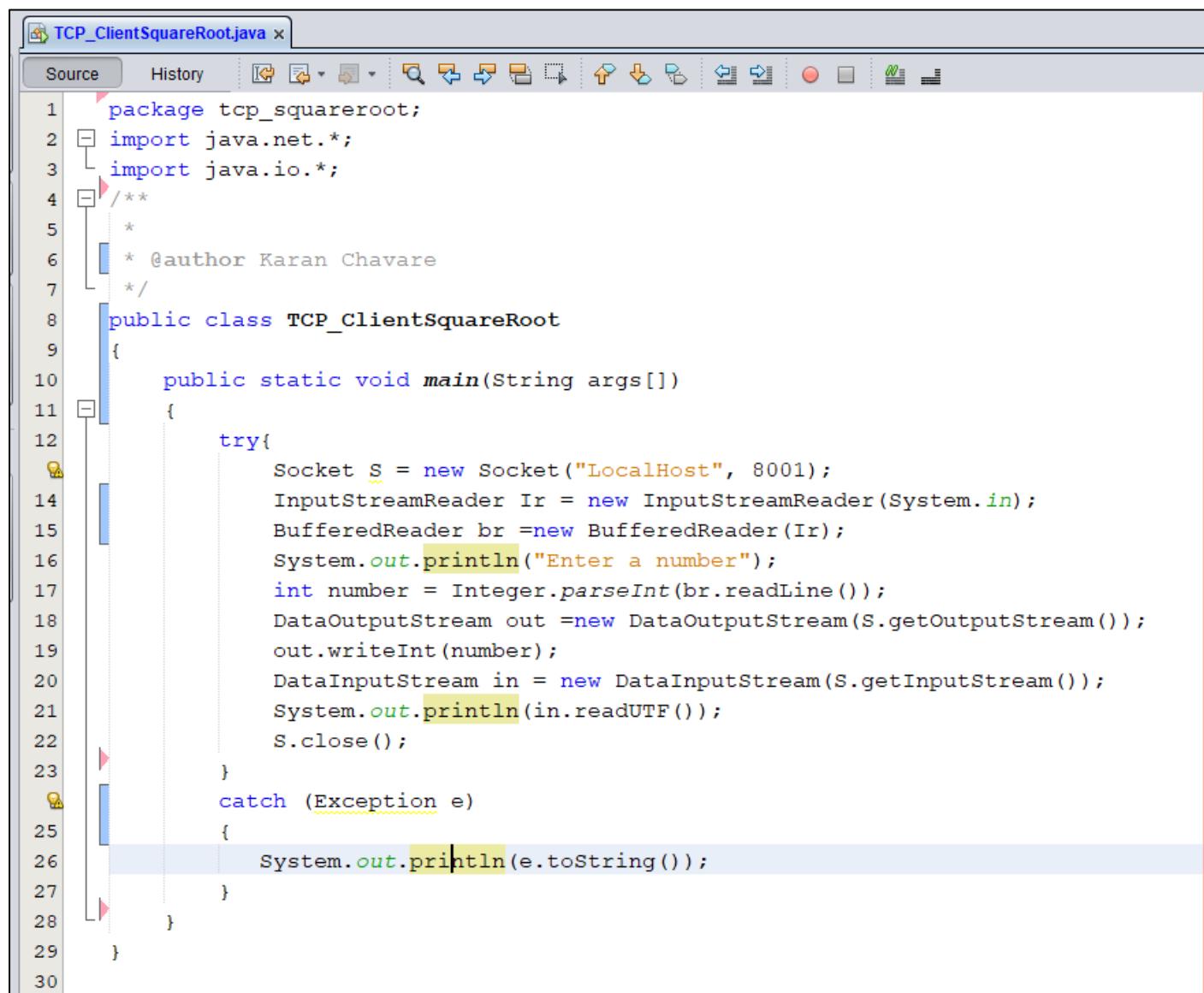
The screenshot shows the output window of the IDE. It displays the run log for the `TCP_SQUARE` project. The application runs successfully, prompting the user to enter a number, receiving the square of the number (25.0), and then exiting.

```
Output
TCP_SQUARE (run) × TCP_SQUARE (run) #2 ×

run:
Enter a number
5
The square of given number is:25.0
BUILD SUCCESSFUL (total time: 7 seconds)
```

- Write a client program to enter the number and server program to calculate the square root of the entered number using TCP Communication.

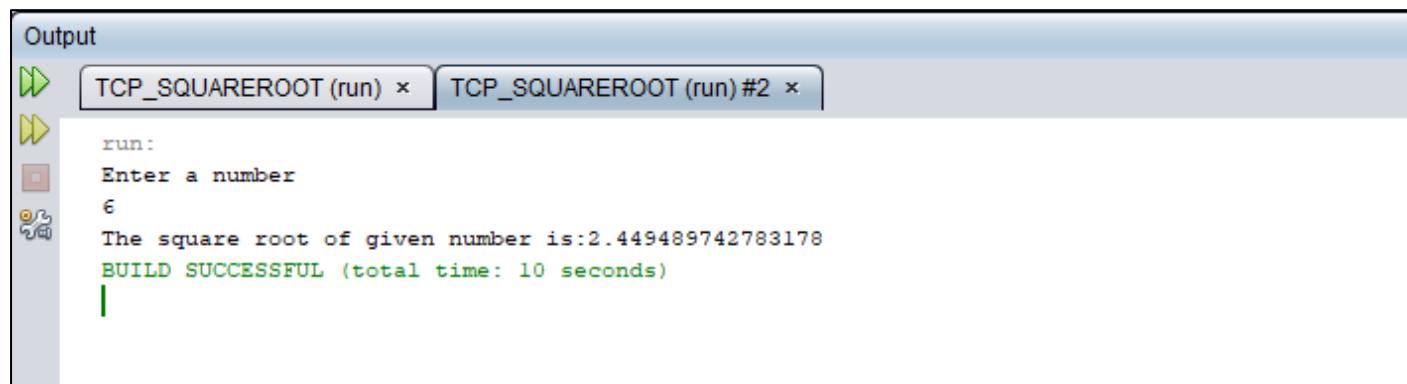
### TCP\_ClientSquareRoot.java



```

1 package tcp_squareroot;
2 import java.net.*;
3 import java.io.*;
4 /**
5 *
6 * @author Karan Chavare
7 */
8 public class TCP_ClientSquareRoot
9 {
10     public static void main(String args[])
11     {
12         try{
13             Socket S = new Socket("LocalHost", 8001);
14             InputStreamReader Ir = new InputStreamReader(System.in);
15             BufferedReader br =new BufferedReader(Ir);
16             System.out.println("Enter a number");
17             int number = Integer.parseInt(br.readLine());
18             DataOutputStream out =new DataOutputStream(S.getOutputStream());
19             out.writeInt(number);
20             DataInputStream in = new DataInputStream(S.getInputStream());
21             System.out.println(in.readUTF());
22             S.close();
23         }
24         catch (Exception e)
25         {
26             System.out.println(e.toString());
27         }
28     }
29 }
30

```



```

Output
TCP_SQUAREROOT (run) x TCP_SQUAREROOT (run) #2 x
run:
Enter a number
6
The square root of given number is:2.449489742783178
BUILD SUCCESSFUL (total time: 10 seconds)
|
```

## TCP ServerSquareRoot.java

The screenshot shows the code editor window of an IDE. The title bar says "TCP\_ServerSquareRoot.java". The menu bar includes "Source", "History", and various tool icons. The code itself is a Java program:

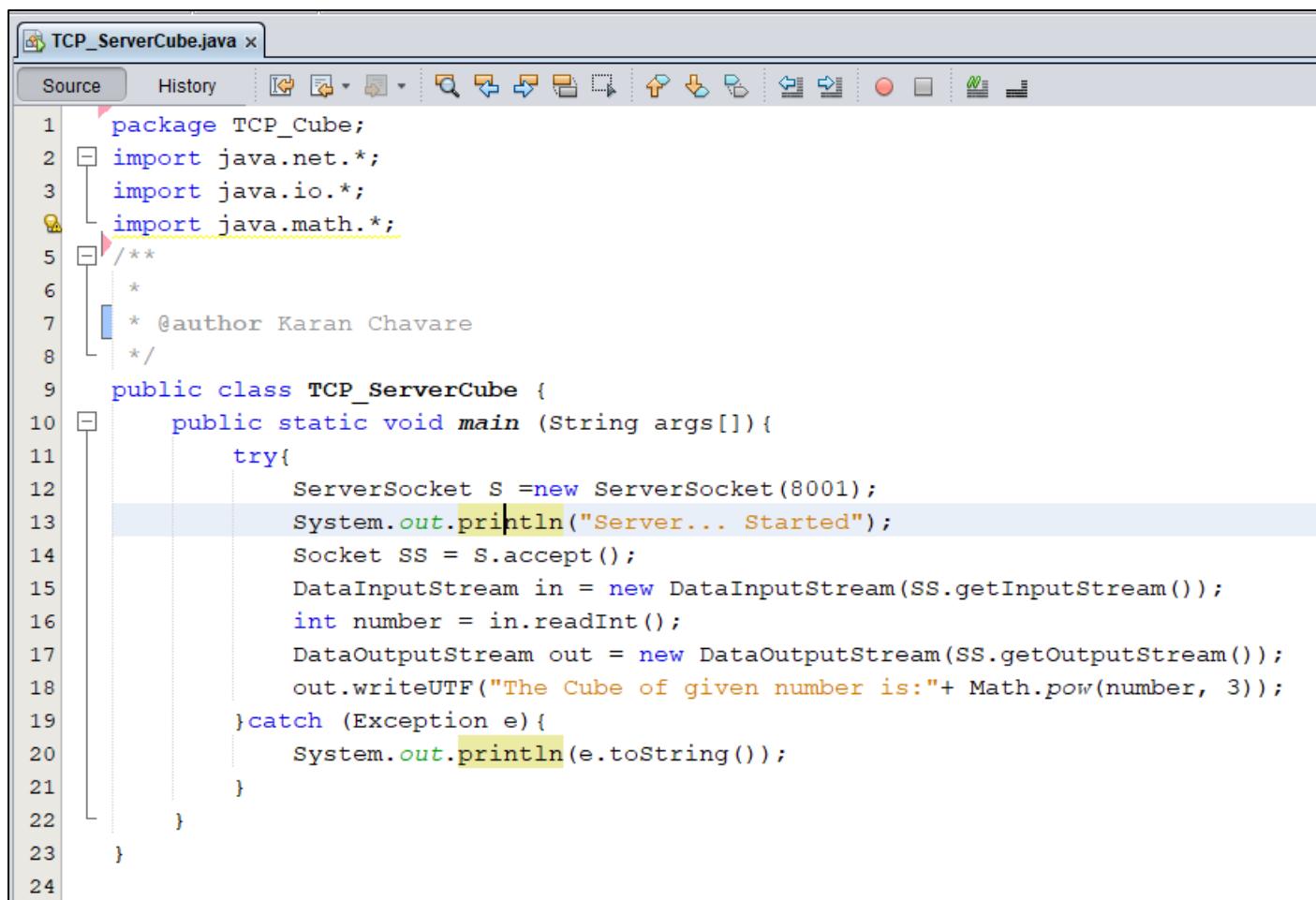
```
1 package tcp_squareroot;
2 import java.net.*;
3 import java.io.*;
4 /**
5 * 
6 * @author Karan Chavare
7 */
8 public class TCP_ServerSquareRoot
9 {
10     public static void main (String args[])
11     {
12         try
13         {
14             ServerSocket S =new ServerSocket(8001);
15             System.out.println("Server... Started");
16             Socket SS = S.accept();
17             DataInputStream in = new DataInputStream(SS.getInputStream());
18             int number = in.readInt();
19             DataOutputStream out = new DataOutputStream(SS.getOutputStream());
20             out.writeUTF("The square root of given number is:"
21                         + Math.sqrt(number));
22         }
23         catch (Exception e)
24         {
25             System.out.println(e.toString());
26         }
27     }
28 }
29
```

The screenshot shows the output window titled "Output - TCP\_SQUAREROOT (run)". It displays the output of the program's execution:

```
run:
Server... Started
```

- Write a client program to enter the number and server program to calculate the Cube of the entered number using TCP Communication.

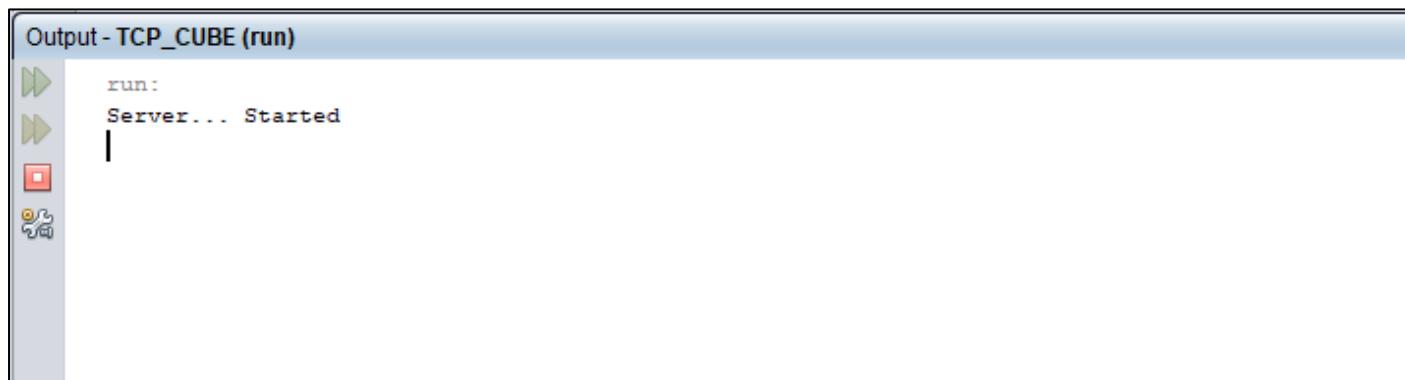
### TCP\_ServerCube.java



```

TCP_ServerCube.java
Source History
1 package TCP_Cube;
2 import java.net.*;
3 import java.io.*;
4 import java.math.*;
5 /**
6 *
7 * @author Karan Chavare
8 */
9 public class TCP_ServerCube {
10     public static void main (String args[]){
11         try{
12             ServerSocket S =new ServerSocket(8001);
13             System.out.println("Server... Started");
14             Socket SS = S.accept();
15             DataInputStream in = new DataInputStream(SS.getInputStream());
16             int number = in.readInt();
17             DataOutputStream out = new DataOutputStream(SS.getOutputStream());
18             out.writeUTF("The Cube of given number is:"+ Math.pow(number, 3));
19         }catch (Exception e){
20             System.out.println(e.toString());
21         }
22     }
23 }
24

```



Output - TCP\_CUBE (run)

```

run:
Server... Started
|_

```

## TCP ClientCube.java

The screenshot shows the source code for `TCP_ClientCube.java` in an IDE. The code implements a TCP client to calculate the cube of a number entered by the user. It uses `InputStreamReader` and `BufferedReader` to read the input, and `DataOutputStream` to write the output.

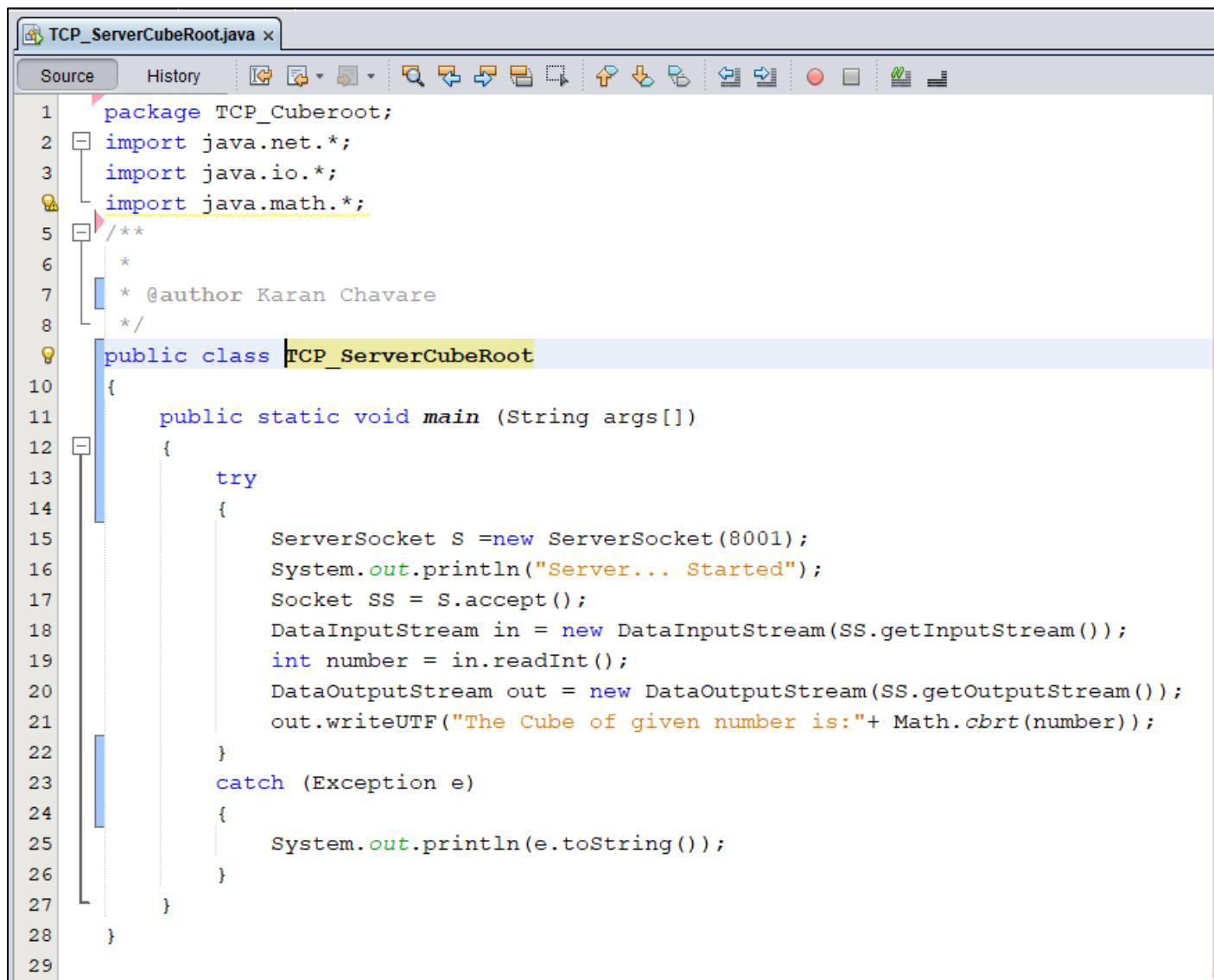
```
1 package TCP_Cube;
2 import java.net.*;
3 import java.io.*;
4 /**
5 *
6 * @author Karan Chavare
7 */
8 public class TCP_ClientCube
9 {
10     public static void main(String args[])
11     {
12         try
13         {
14             Socket S = new Socket("LocalHost", 8001);
15             InputStreamReader Ir = new InputStreamReader(System.in);
16             BufferedReader br =new BufferedReader(Ir);
17             System.out.println("Enter a number");
18             int number = Integer.parseInt(br.readLine());
19             DataOutputStream out =new DataOutputStream(S.getOutputStream());
20             out.writeInt(number);
21             DataInputStream in = new DataInputStream(S.getInputStream());
22             System.out.println(in.readUTF());
23             S.close();
24
25         }
26         catch (Exception e)
27         {
28             System.out.println(e.toString());
29         }
30     }
31 }
32 }
```

The screenshot shows the output window of the IDE during the execution of the `TCP_CLIENT` application. The console displays the interaction between the client and the server, including the user input and the calculated result.

```
debug:
Enter a number
9
The Cube of given number is:729.0
BUILD SUCCESSFUL (total time: 4 seconds)
```

- Write a client program to enter the number and server program to calculate the Cube Root of the entered number using TCP Communication.

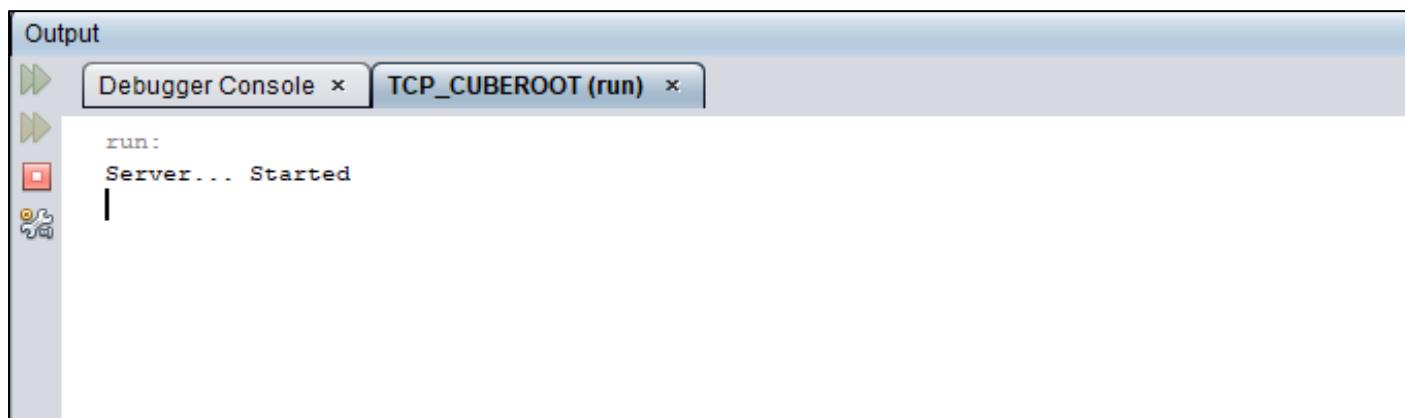
### TCP\_ServerCubeRoot.java



```

1 package TCP_Cuberoott;
2 import java.net.*;
3 import java.io.*;
4 import java.math.*;
5 /**
6 *
7 * @author Karan Chavare
8 */
9 public class TCP_ServerCubeRoot
10 {
11     public static void main (String args[])
12     {
13         try
14         {
15             ServerSocket S = new ServerSocket(8001);
16             System.out.println("Server... Started");
17             Socket SS = S.accept();
18             DataInputStream in = new DataInputStream(SS.getInputStream());
19             int number = in.readInt();
20             DataOutputStream out = new DataOutputStream(SS.getOutputStream());
21             out.writeUTF("The Cube of given number is:"+ Math.cbrt(number));
22         }
23         catch (Exception e)
24         {
25             System.out.println(e.toString());
26         }
27     }
28 }
29

```



```

Output
Debugger Console x TCP_CUBEROOT (run) x
run:
Server... Started
|
```

## TCP\_ClientCubeRoot.java

The screenshot shows a Java code editor window titled "TCP\_ClientCubeRoot.java". The code implements a client application using sockets to calculate the cube root of a number. The code uses BufferedReader and DataOutputStream to handle input and output streams.

```
1 package TCP_Cuberoot;
2 import java.net.*;
3 import java.io.*;
4 /**
5  * 
6  * @author Karan Chavare
7  */
8 public class TCP_ClientCubeRoot
9 {
10     public static void main(String args[])
11     {
12         try
13         {
14             Socket S = new Socket("LocalHost", 8001);
15             InputStreamReader Ir = new InputStreamReader(System.in);
16             BufferedReader br =new BufferedReader(Ir);
17             System.out.println("Enter a number");
18             int number = Integer.parseInt(br.readLine());
19             DataOutputStream out =new DataOutputStream(S.getOutputStream());
20             out.writeInt(number);
21             DataInputStream in = new DataInputStream(S.getInputStream());
22             System.out.println(in.readUTF());
23             S.close();
24         }
25         catch (Exception e)
26         {
27             System.out.println(e.toString());
28         }
29     }
30 }
31
```

The screenshot shows the "Output" window of the IDE. It displays the run log, which includes the command "run:", user input "Enter a number", the number "8", the program's response "The Cube Root of given number is:2.0", and the build status "BUILD SUCCESSFUL (total time: 2 seconds)".

```
run:
Enter a number
8
The Cube Root of given number is:2.0
BUILD SUCCESSFUL (total time: 2 seconds)
```

## PRACTICAL - 2

### ❖ **Implement Client Server Communication Model Using UDP.**

- Write a client program to enter the number and server program to calculate the factorial of the entered number using UDP Communication.

#### UDP\_ServerFactorial.java

```
1 package udp_factorial;
2 import java.io.*;
3 import java.net.*;
4
5 public class UDP_ServerFactorial {
6     public static void main(String args[]){
7         try
8         {
9             int factnum;
10            DatagramSocket ds = new DatagramSocket(2000);
11            byte b[] = new byte[1024];
12            DatagramPacket dp = new DatagramPacket(b,b.length);
13            ds.receive(dp);
14            String str = new String(dp.getData(),0,dp.getLength());
15            System.out.println(str);
16            int a = Integer.parseInt(str);
17            factnum = fact(a);
18            String s1 = "The factorial of a given number is " + factnum;
19            byte b1[] = new byte[1024];
20            b1 = s1.getBytes();
21            DatagramPacket dp1;
22            dp1= new DatagramPacket(b1,b1.length,
23                InetAddress.getLocalHost(),1000);
24            ds.send(dp1);
25        }
26        catch(Exception e)
27        {
28            e.printStackTrace();
29        }
30    }
31
32    public static int fact(int arg1){
33        int fact=1,i=1;
34        for (i=fact;i<=arg1;i++){
35            fact=fact*i;
36        }
37        return fact;
38    }
39}
```

## UDP ClientFactorial.java

The screenshot shows a Java code editor window with the file `UDP_ClientFactorial.java` open. The code implements a UDP client to calculate the factorial of a given number. It uses `DatagramSocket` for sending and receiving data over UDP.

```
1 package udp_factorial;
2 import java.net.*;
3 import java.util.*;
4 import java.io.*;
5 /**
6 *
7 * @author Karan Chavare
8 */
9 public class UDP_ClientFactorial
10 {
11     public static void main(String arg[])
12     throws Exception
13     {
14         DatagramSocket Ds = new DatagramSocket(1000);
15         InputStreamReader Ir = new InputStreamReader(System.in);
16         BufferedReader Bf = new BufferedReader(Ir);
17
18         System.out.println("Enter the Number:");
19         String num = Bf.readLine();
20
21         byte b[] = new byte[1024];
22         b = num.getBytes();
23
24         DatagramPacket Dp;
25         Dp = new DatagramPacket(b, b.length, InetAddress.getLocalHost(), 2000);
26         Ds.send(Dp);
27
28         byte b1[] = new byte[1024];
29         DatagramPacket dp1= new DatagramPacket(b1, b1.length);
30         Ds.receive(dp1);
31
32         String str = new String(dp1.getData(), 0, dp1.getLength());
33         System.out.println(str);
34     }
35
36 }
37
```

The screenshot shows the output window of the IDE displaying the execution results of the `UDP_ClientFactorial` application. The application prompts the user to enter a number, receives the input, calculates the factorial using a UDP socket, and then prints the result back to the console.

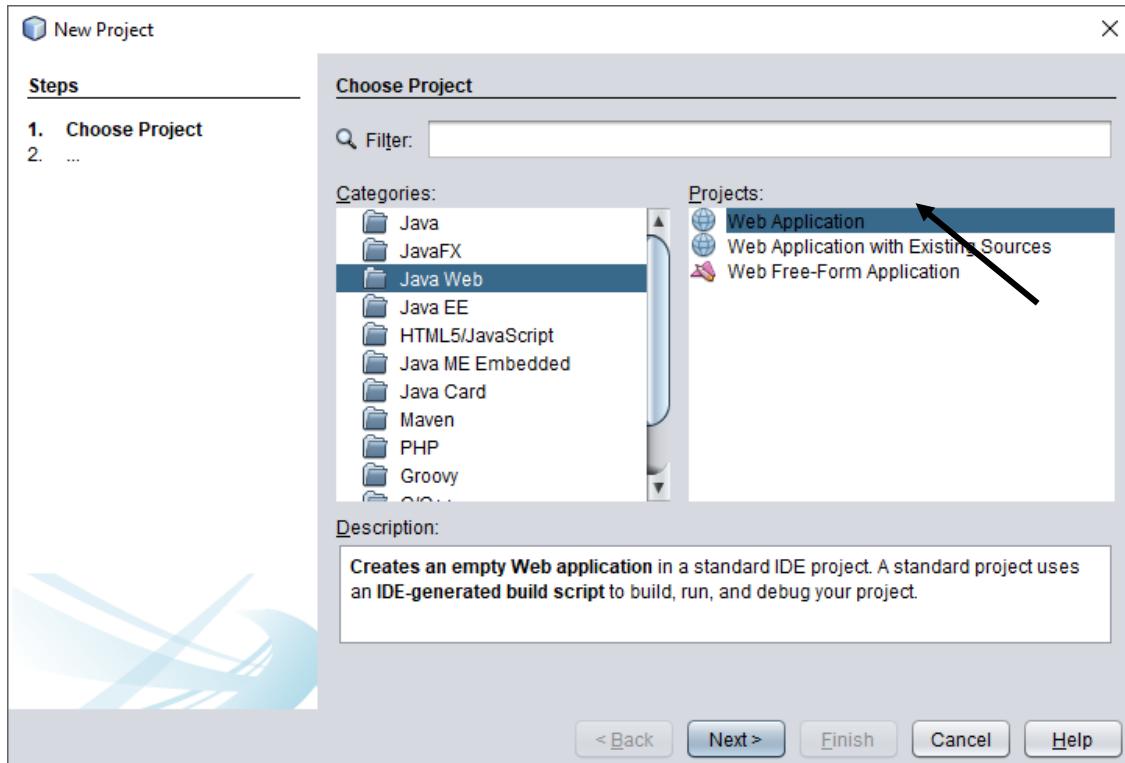
```
run:
Enter the Number:
3
The factorial of a given number is 6
BUILD SUCCESSFUL (total time: 4 seconds)
```

## PRACTICAL - 3

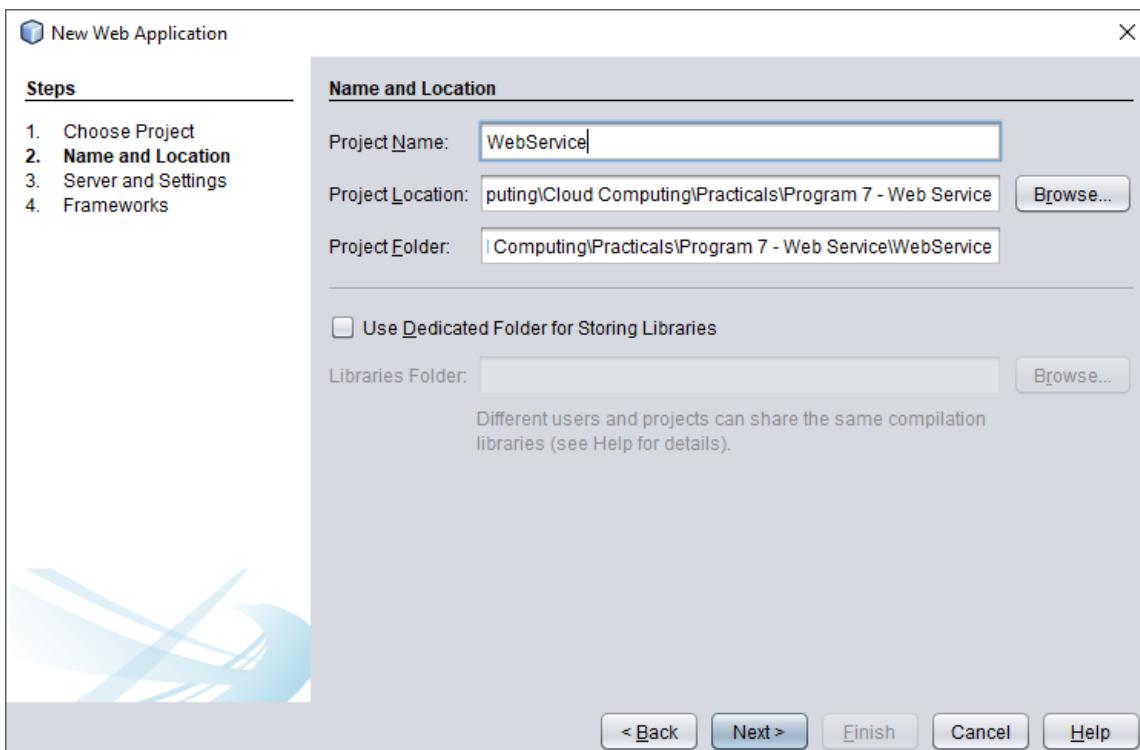
### ❖ Implementation of Web Services.

- **Creating a Webservice**

- Choose File > New Project.
- Select Web Application from the Java Web & Click Next.

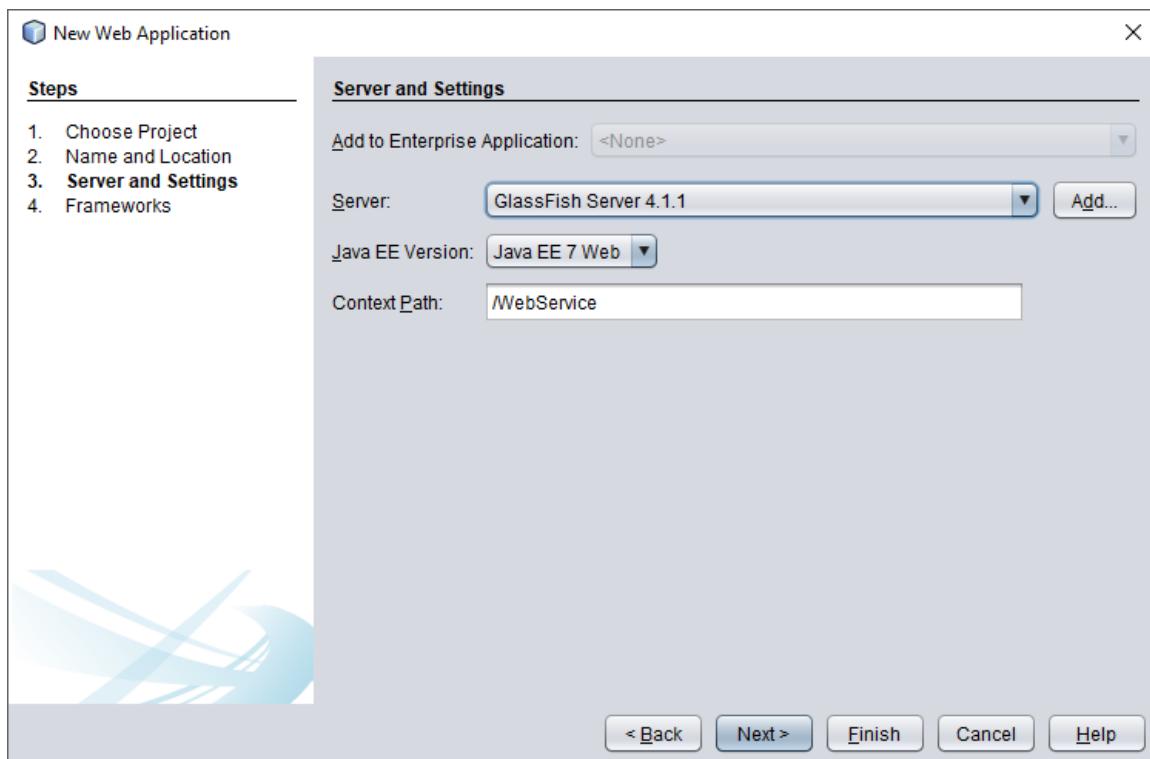


- Name the project WebService. Select a location for the project. Click Next.



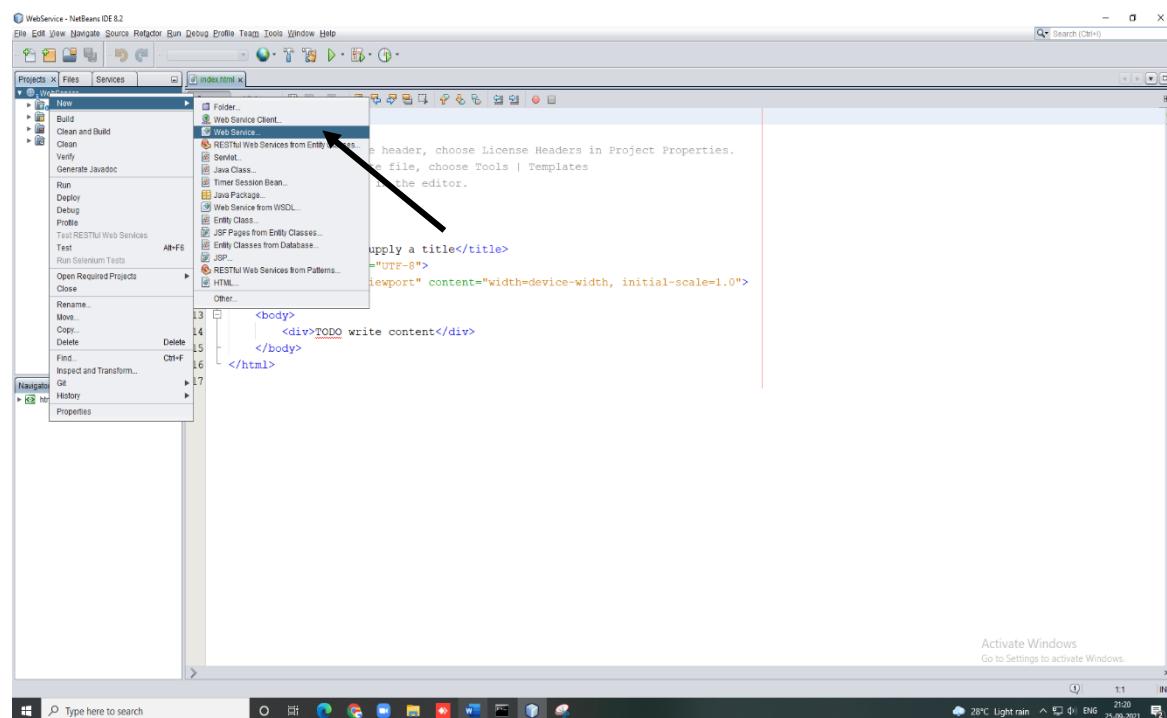
## PRACTICAL - 3

- Select your server and Java EE version and click Finish.



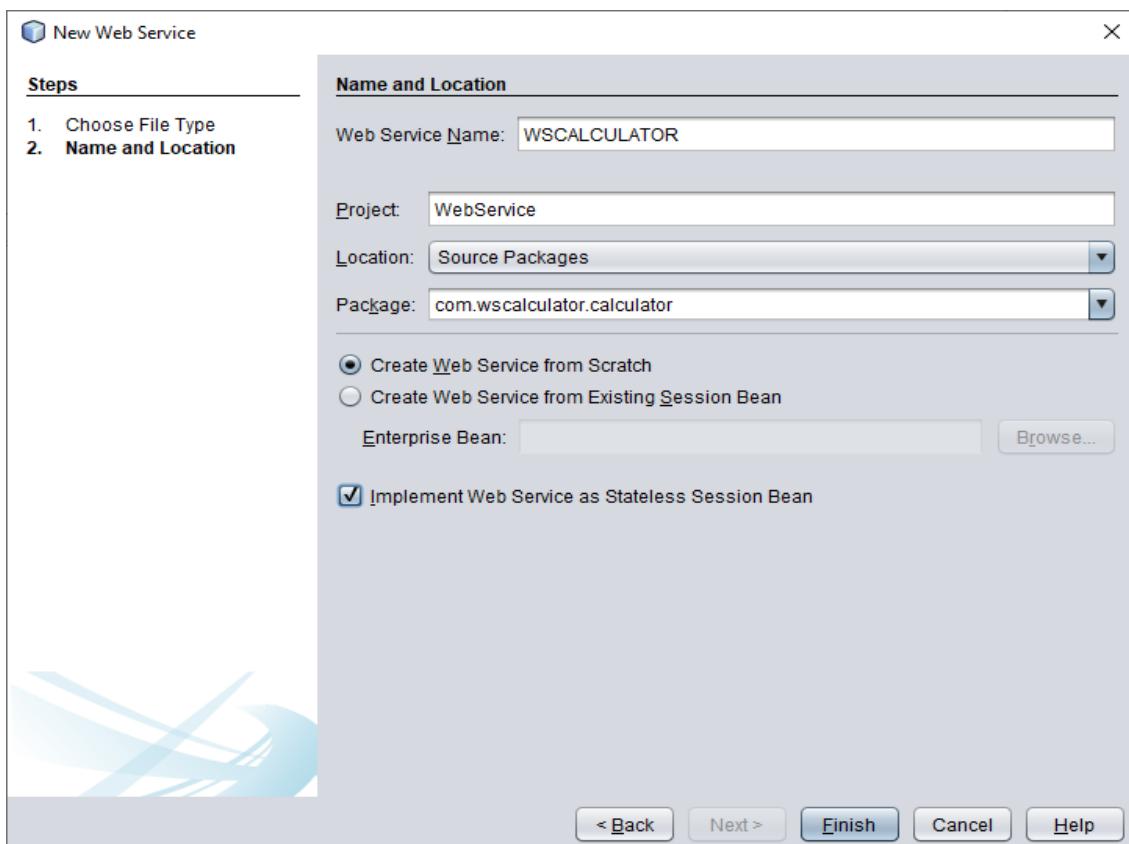
- **Creating a Web Service from a Java Class**

- Right-click the WebService node in project explorer window and choose New > Web Service.



## PRACTICAL - 3

- Name the web service as WSCALCULATOR and type com.wscalculator.calculator in Package.
- Leave create web service from scratch selected. If you are creating a Java EE 6 project on GlassFish,
- Select Implement Web Service as a Stateless Session Bean.
- Click Finish. The projects window displays the structure of the new web service and the source code is shown in the editor area.



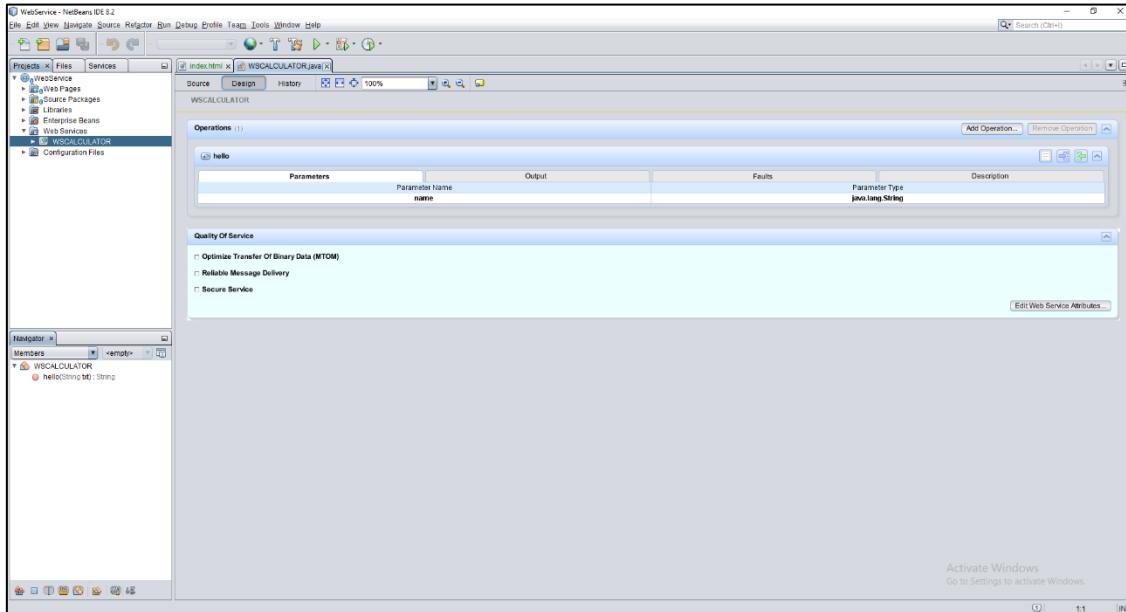
- **Adding an Operation to the Web Service**

The goal of this exercise is to add to the web service an operation that adds two numbers received from a client. The NetBeans IDE provides a dialog for adding an operation to a web service. You can open this dialog either in the web service visual designer or in the web service context menu.

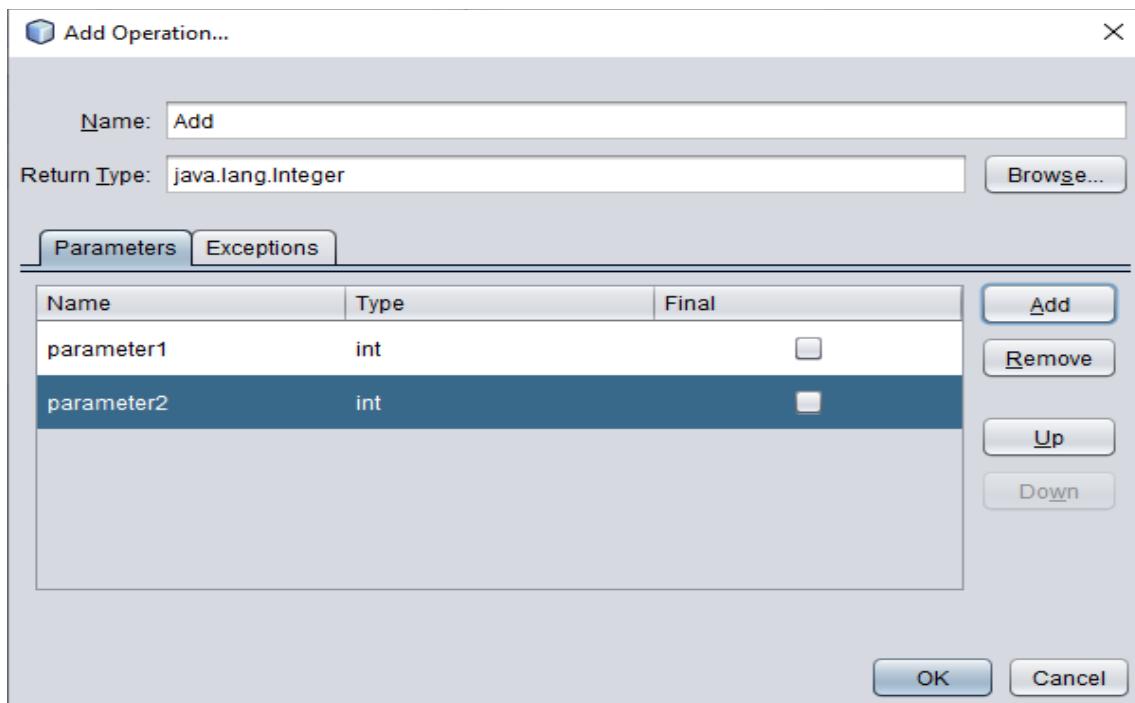
# PRACTICAL - 3

## To add an operation to the web service:

- Change to the Design view in the editor.

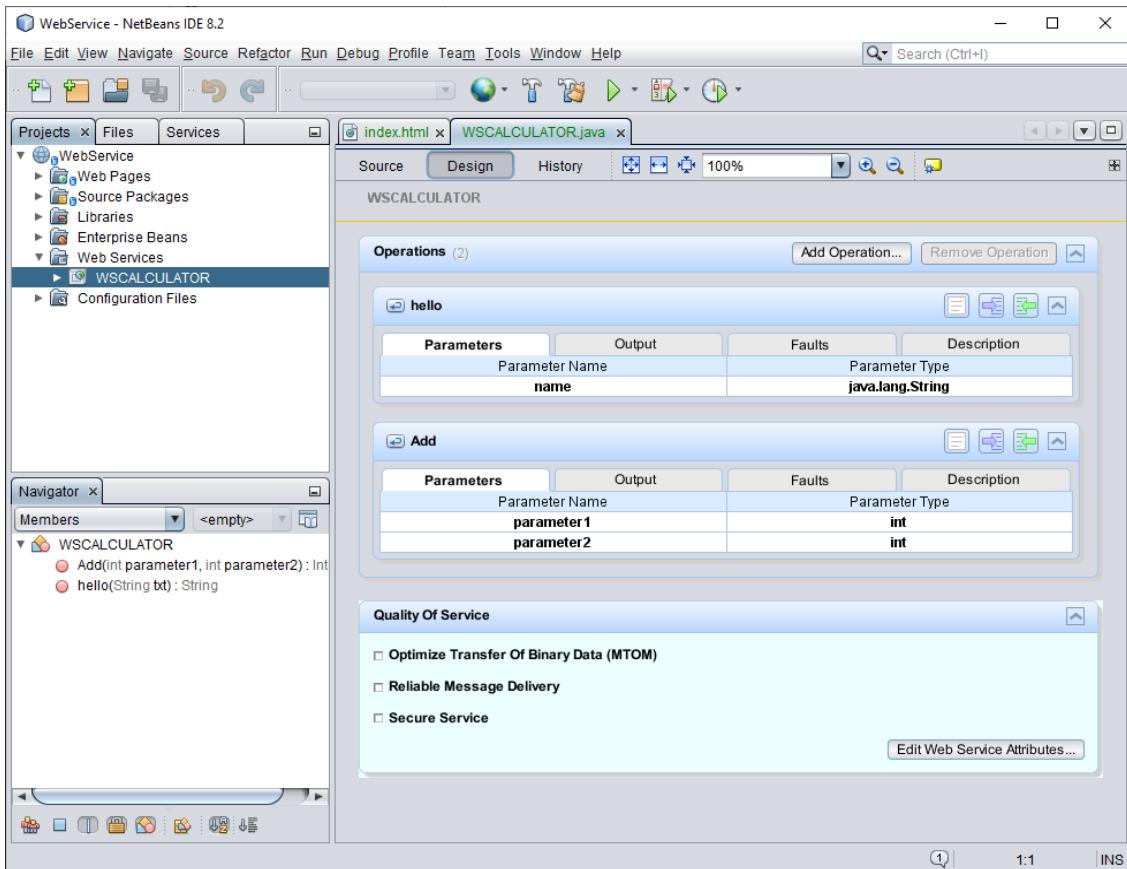


- Click Add Operation in either the visual designer or the context menu. The Add Operation dialog opens.
- In the upper part of the Add Operation dialog box, type add in Name and type int in the Return Type drop-down list.
- In the lower part of the Add Operation dialog box, click Add and create a parameter of type int named parameter1.
- Click Add again and create a parameter of type int called parameter2. You now see the following:



## PRACTICAL - 3

- Click OK at the bottom of the Add Operation dialog box. You return to the editor.
- The visual designer now displays the following:



- Click Source. And code the following.

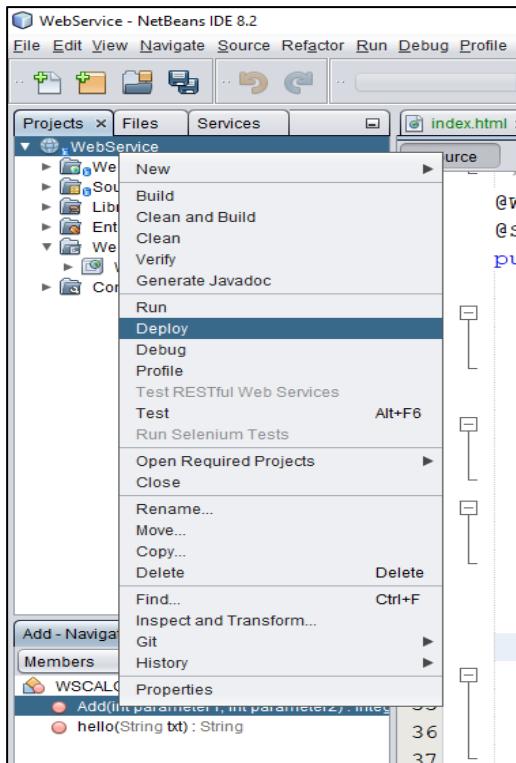
```
16  /*
17  * To change this template, choose Tools | Templates
18  * and open the template in the editor.
19  */
20  package com.wso2.sample;
21
22  /**
23  * This is a sample web service operation
24  */
25  @WebService(serviceName = "WSCALULATOR")
26  @Stateless()
27  public class WSCALULATOR {
28
29      /**
30      * Web service operation
31      */
32      @WebMethod(operationName = "hello")
33      public String hello(@WebParam(name = "name") String txt) {
34          return "Hello " + txt + " !";
35      }
36
37      /**
38      * Web service operation
39      */
40      @WebMethod(operationName = "Add")
41      public Integer Add(@WebParam(name = "parameter1") int parameter1,
42                         @WebParam(name = "parameter2") int parameter2)
43      {
44          //TODO write your implementation code here:
45          return parameter1 + parameter2;
46      }
47  }
```

## PRACTICAL - 3

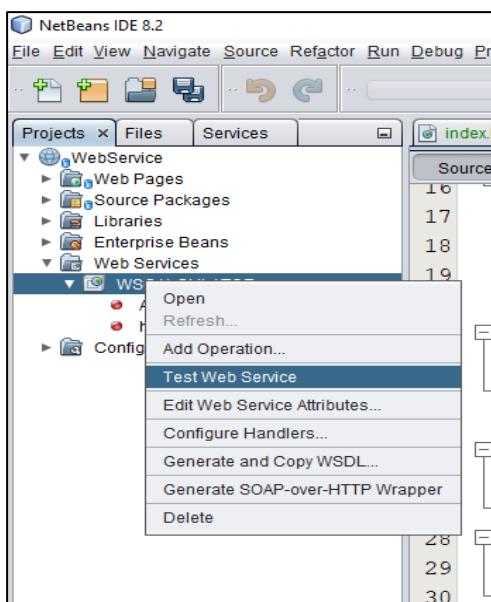
- **Deploying and Testing the Web Service**

After you deploy a web service to a server, you can use the IDE to open the server's test client, if the server has a test client. The GlassFish servers provide test clients.

- To test successful deployment to a GlassFish server:
- Right-click the project and choose Deploy. The IDE starts the application server, builds the application, and deploys the application to the server.



- In the IDE's Projects tab, expand the Web Services node of the WebService project.
- Right-click the WSCALCULATOR node, and choose Test Web Service



## PRACTICAL - 3

- The IDE opens the tester page in your browser if you deployed a web application to the GlassFish server.

The screenshot shows a web browser window titled "WSCALCULATOR Web Service Tester". The URL in the address bar is "http://localhost:8080/WSCALCULATOR/WSCALCUL...". The page content includes a heading "WSCALCULATOR Web Service Tester", a sub-heading "This form will allow you to test your web service implementation ([WSDL File](#))", and instructions: "To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name." Below these are two sections for methods:

**Methods :**

```
public abstract java.lang.Integer com.wscalculator.calculatorWSCALCULATOR.add(int,int)
add ( [ ] , [ ] )
```

```
public abstract java.lang.String com.wscalculator.calculatorWSCALCULATOR.hello(java.lang.String)
hello ( [ ] )
```

- If you deployed to the GlassFish server, type two numbers in the tester page, as shown below:
- The sum of the two numbers is displayed:

The screenshot shows a web browser window titled "Method invocation trace". The URL in the address bar is "http://localhost:8080/WSCALCULATOR/WSCALCUL...". The page content includes a section for "Method parameter(s)" with a table:

Type	Value
int	5
int	6

Below this is a section for "Method returned" showing the result: "java.lang.Integer : "11"".

At the bottom, there is a "SOAP Request" section containing the XML code for the SOAP message:

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Header>
<S:Body>
<ns2:Add xmlns:ns2="http://calculator.wscalculator.com/">
<parameter1>5</parameter1>
```

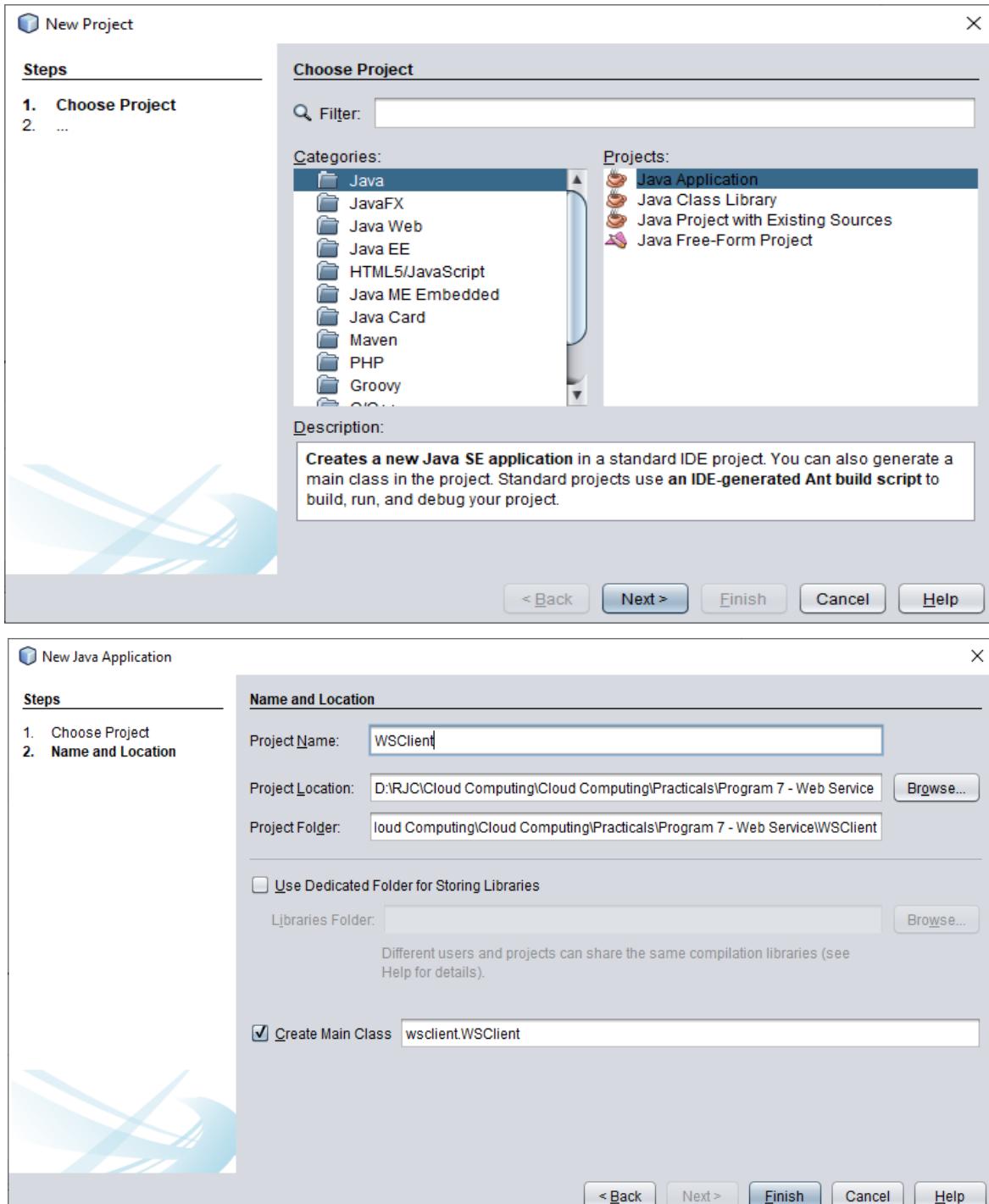
# PRACTICAL - 3

- **Consuming the Web Service**

➤ Now that you have deployed the web service, you need to create a client to make use of the web service's Add method.

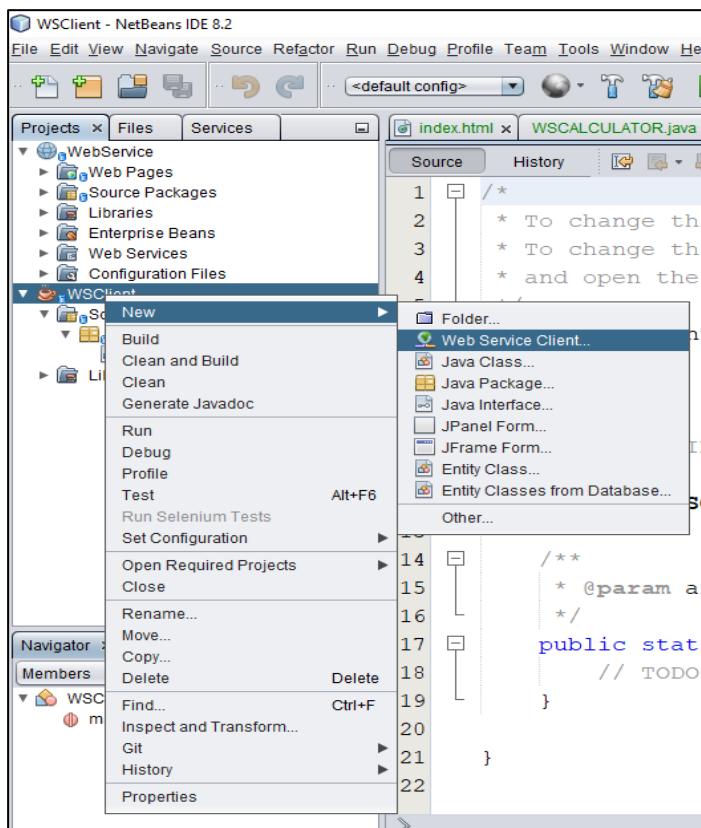
- **Client1: Java Class in Java SE Application**

- Choose File > New Project. Select Java Application from the Java category.
- Name the project WSClient.
- Leave Create Main Class selected and accept all other default settings. Click Finish.

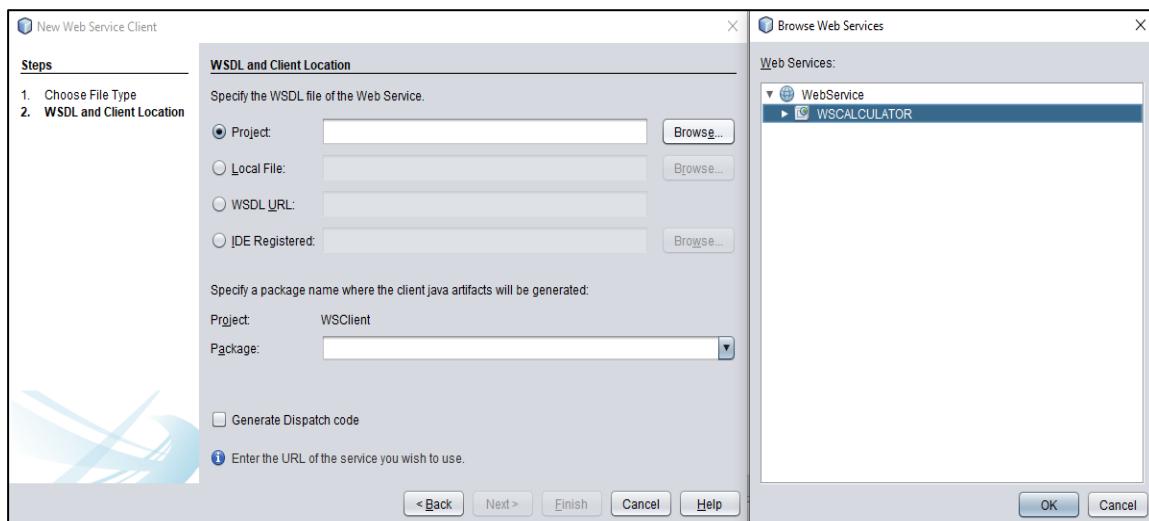


## PRACTICAL - 3

- Right-click the WSClient node and choose New > Web Service Client. The New Web Service Client wizard opens.

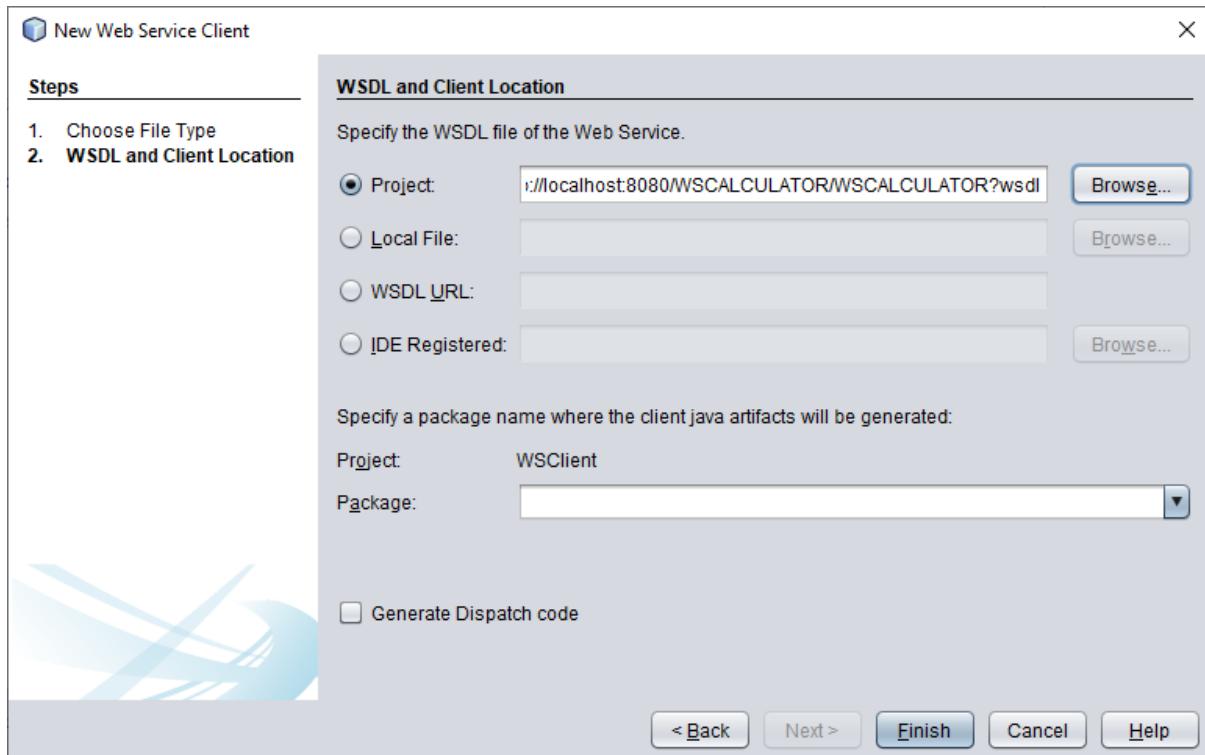


- Select Project as the WSDL source. Click Browse. Browse to the WSCALCULATOR web service in the WebService project. When you have selected the web service, click OK.

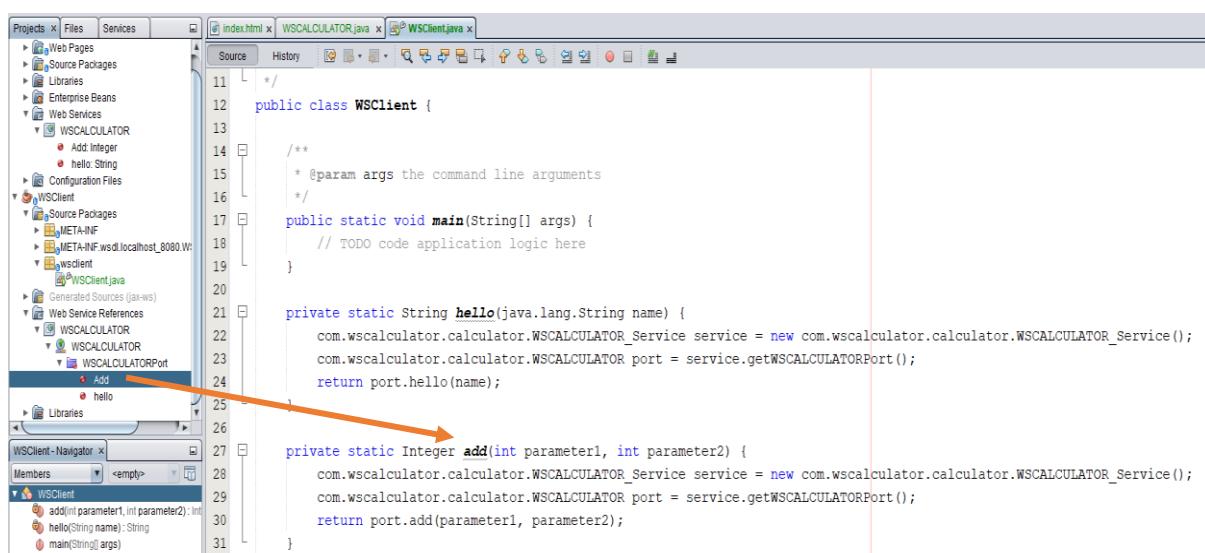


## PRACTICAL - 3

- Do not select a package name. Leave this field empty.

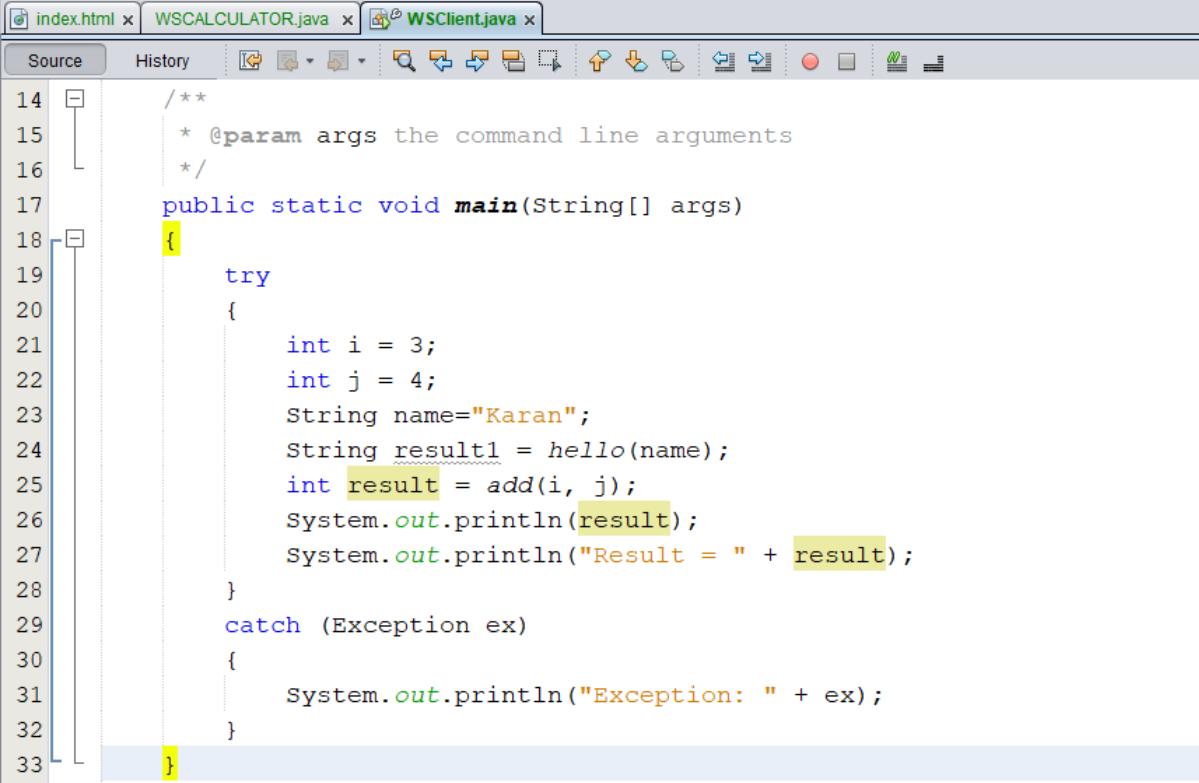


- Leave the other settings at default and click Finish. The Projects window displays the new web service client, with a node for the add method that you created:  
➤ Double-click your main class so that it opens in the Source Editor. Drag the add node below the main() method (or ) you can right-click in the editor and then choose Insert Code > Call Web Service Operation.



## PRACTICAL - 3

- In the main() method body, replace the TODO comment with code that initializes values for i and j, calls add(), and prints the result.



The screenshot shows an IDE interface with three tabs at the top: index.html, WSCALCULATOR.java, and WSClient.java. The WSClient.java tab is active, displaying the following Java code:

```
14  /**
15   * @param args the command line arguments
16   */
17  public static void main(String[] args)
18  {
19      try
20      {
21          int i = 3;
22          int j = 4;
23          String name="Karan";
24          String result1 = hello(name);
25          int result = add(i, j);
26          System.out.println(result);
27          System.out.println("Result = " + result);
28      }
29      catch (Exception ex)
30      {
31          System.out.println("Exception: " + ex);
32      }
33 }
```

- Right-click the project node and choose Run.
- The Output window now shows the sum:

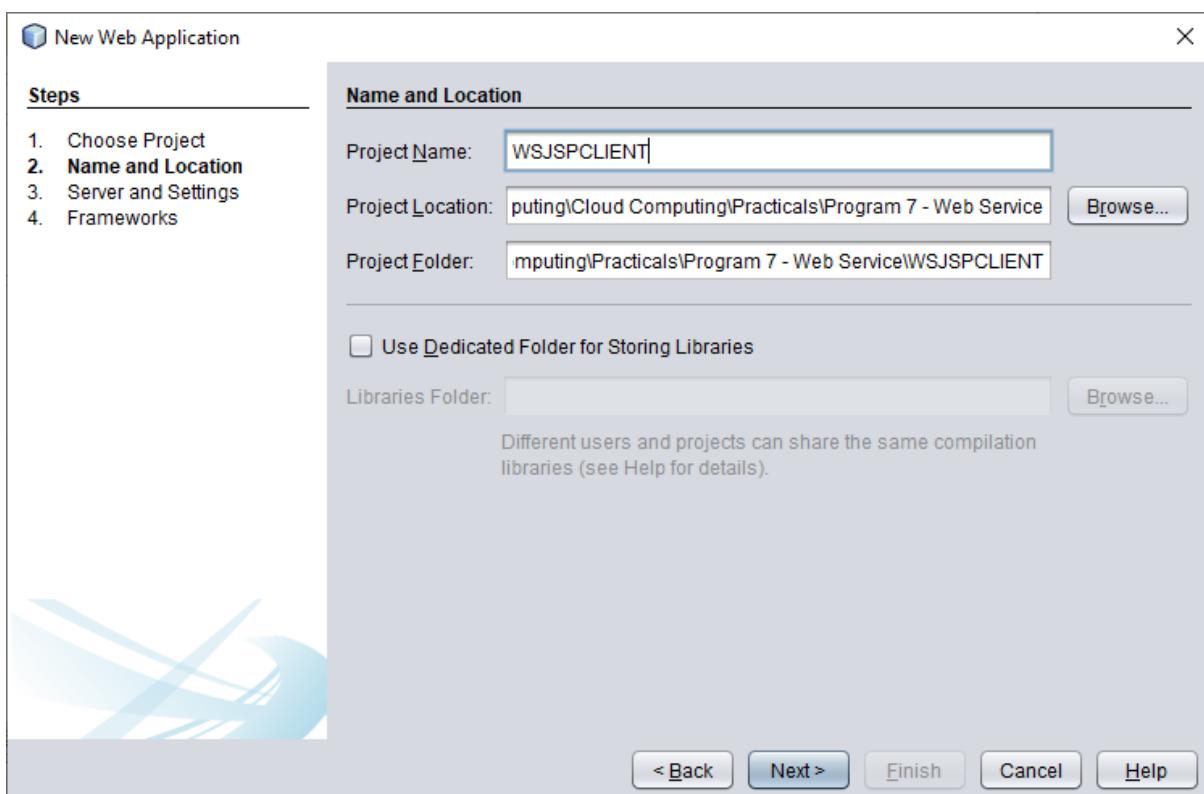
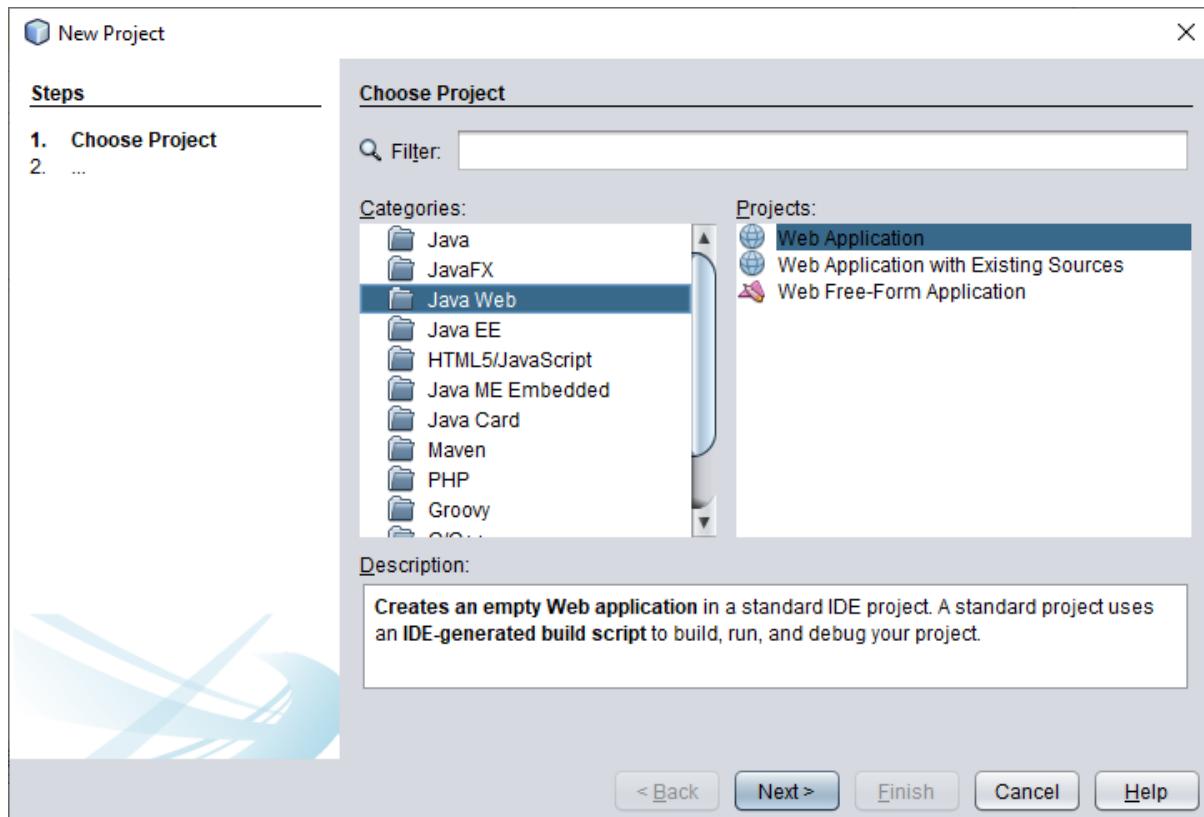


The screenshot shows the Output window of the IDE, which displays the following build logs:

```
init:
Deleting: D:\RJC\Cloud Computing\Cloud Computing\Practicals\Program 7 - Web Service\WSClient.jar
deps-jar:
Updating property file: D:\RJC\Cloud Computing\Cloud Computing\Practicals\Program 7 - Web Service\WSClient.jar
wsimport-init:
wsimport-client-WSCALCULATOR:
files are up to date
wsimport-client-generate:
compile:
run:
Hello Karan !
Result = 7
BUILD SUCCESSFUL (total time: 0 seconds)
```

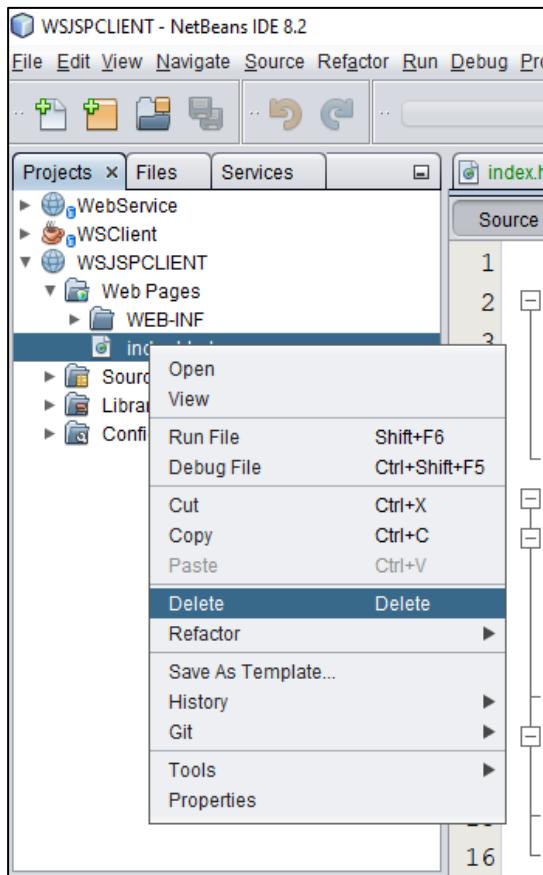
## PRACTICAL - 3

- Client2: JSP page in web application.
- Choose File > New Project.
- Select Web Application from the Java Web category. Name the project WSJSPCLIENT.

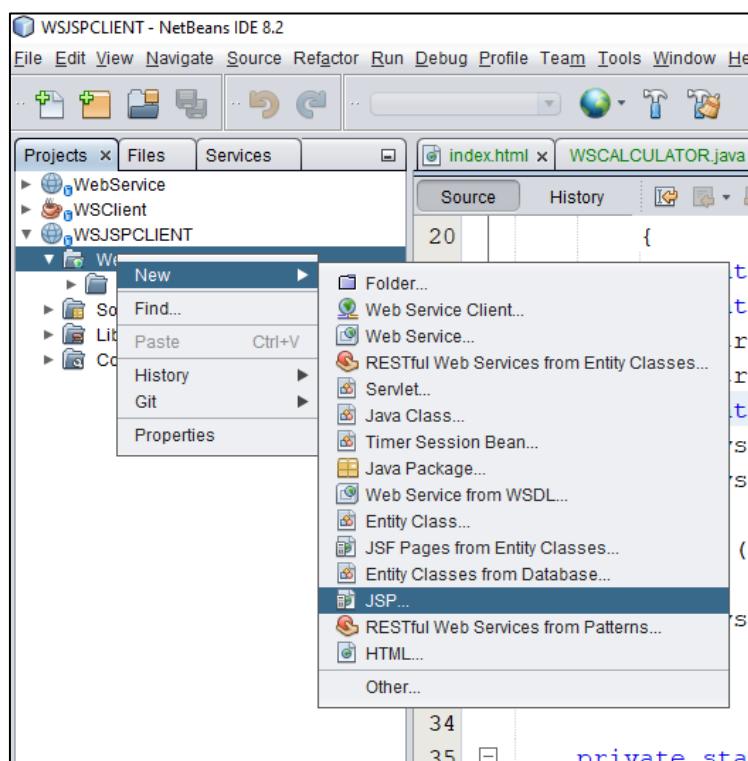


## PRACTICAL - 3

- Click Next and then click Finish.
- Expand the Web Pages node under the project node and delete index.html.

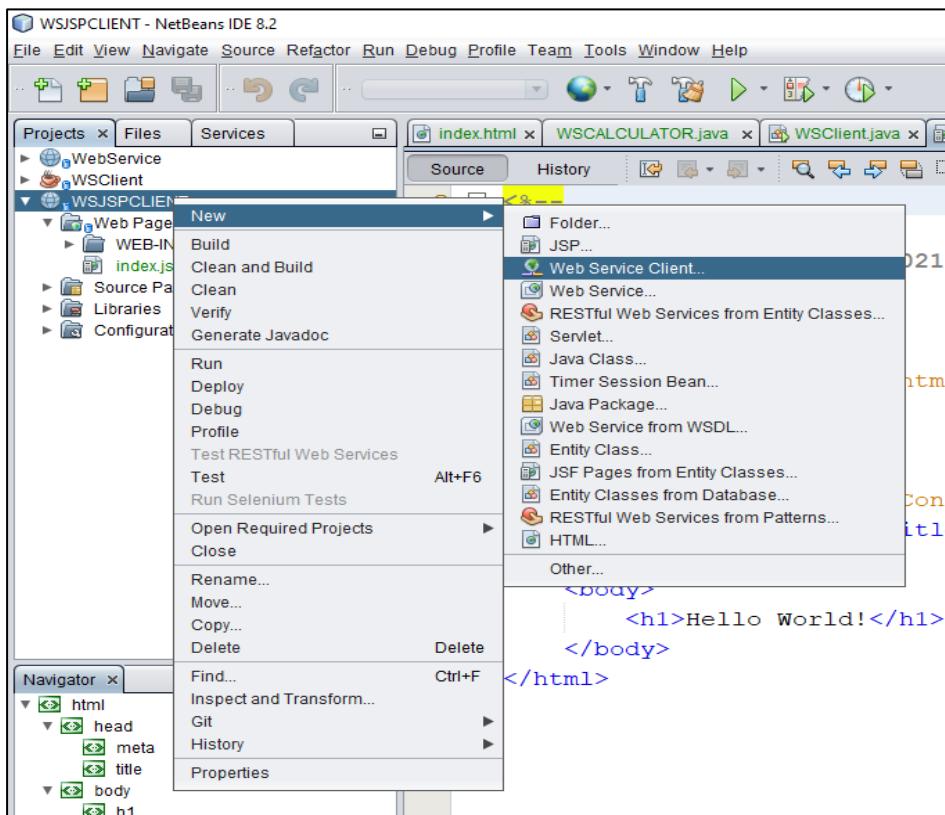


- Right-click the Web Pages node and choose New > JSP in the popup menu.

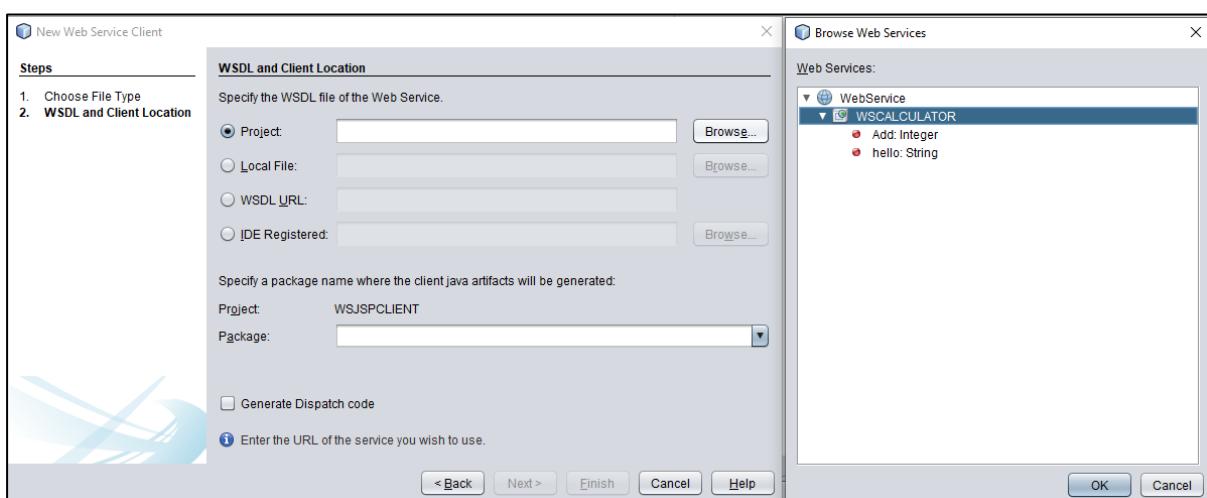


## PRACTICAL - 3

- If JSP is not available in the popup menu, choose New > Other and select JSP in the Web category of the New File wizard.
- Type index for the name of the JSP file in the New File wizard. Click Finish.
- Right-click the WSJSPCLIENT node and choose New > Web Service Client.

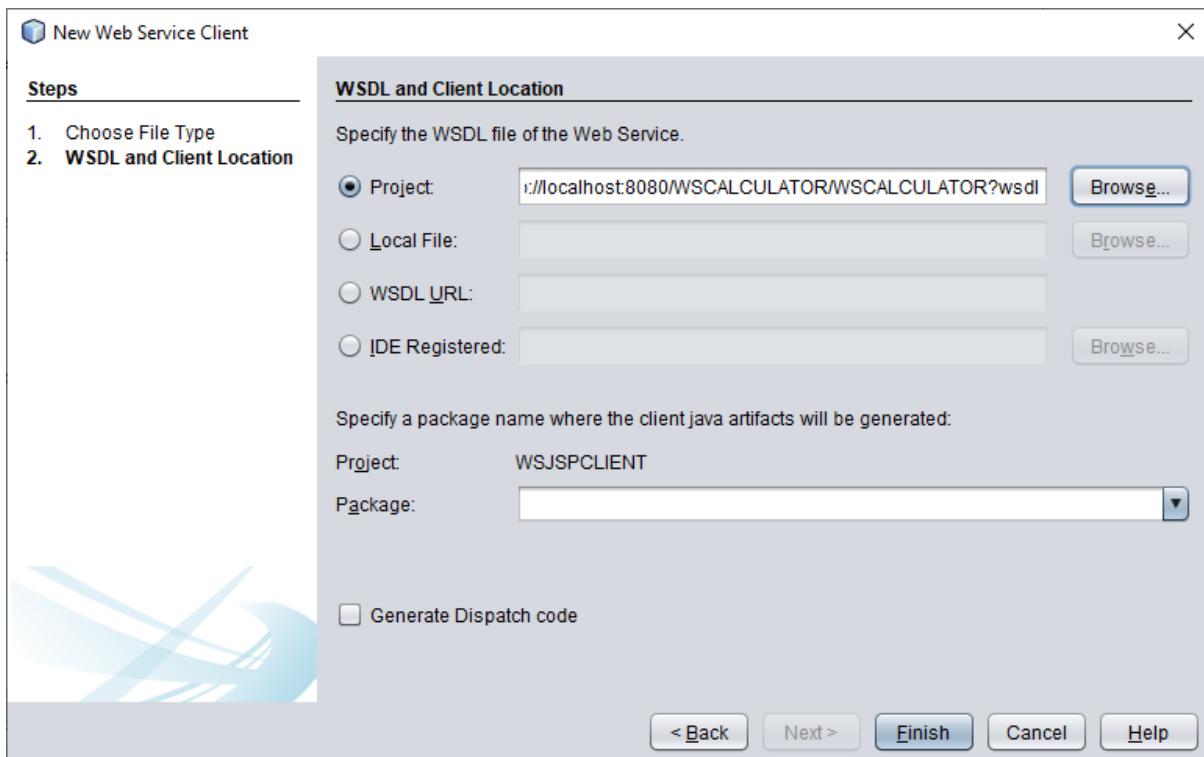


- Select Project as the WSDL source.
- Click Browse. Browse to the WSCALCULATOR web service in the WebService project.
- When you have selected the web service, click OK.

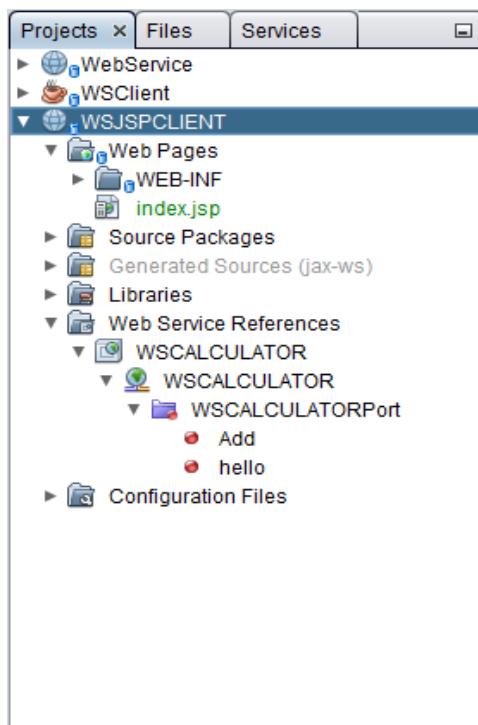


## PRACTICAL - 3

- Do not select a package name. Leave this field empty.
- Leave the other settings at default and click Finish.

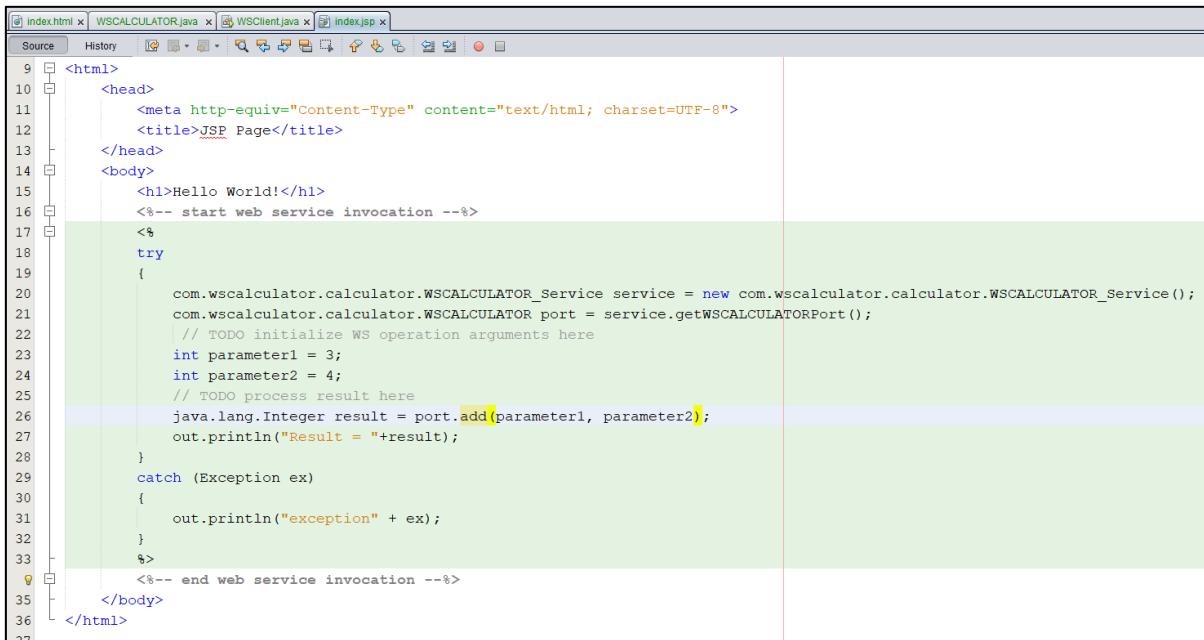


- The Projects window displays the new web service client, as shown below:



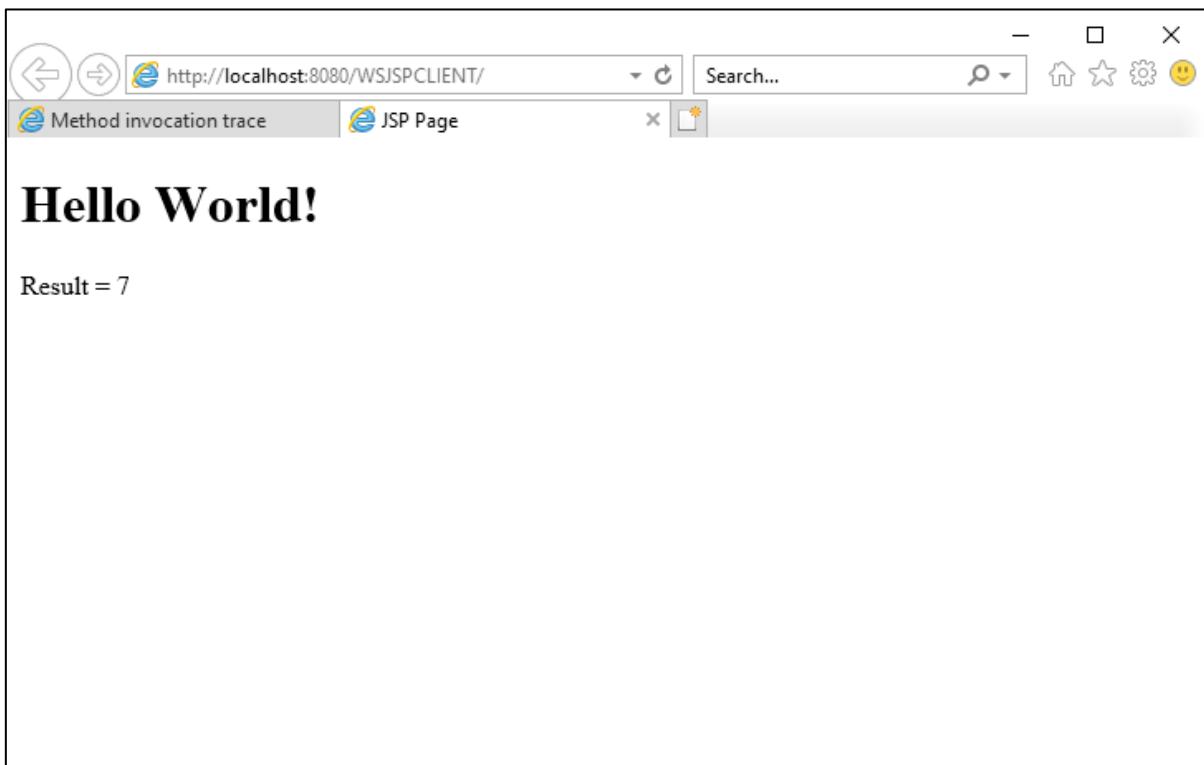
## PRACTICAL - 3

- In the Web Service References node, expand the node that represents the web service.
- The add operation, which you will invoke from the client, is now exposed.
- Drag the add operation to the client's index.jsp page and drop it below the H1 tags.



```
9 <html>
10 <head>
11     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12     <title>JSP Page</title>
13 </head>
14 <body>
15     <h1>Hello World!</h1>
16     <%-- start web service invocation --%>
17     <%
18     try
19     {
20         com.wscalculator.calculator.WSCALCULATOR_Service service = new com.wscalculator.calculator.WSCALCULATOR_Service();
21         com.wscalculator.calculator.WSCALCULATOR port = service.getWSCALCULATORPort();
22         // TODO initialize WS operation arguments here
23         int parameter1 = 3;
24         int parameter2 = 4;
25         // TODO process result here
26         java.lang.Integer result = port.add(parameter1, parameter2);
27         out.println("Result = "+result);
28     }
29     catch (Exception ex)
30     {
31         out.println("exception" + ex);
32     }
33     <%-- end web service invocation --%>
34     </body>
35 </html>
```

- Right-click the project node and choose Run.
- The server starts if it wasn't running already. The application is built and deployed, and the browser opens, displaying the calculation result:

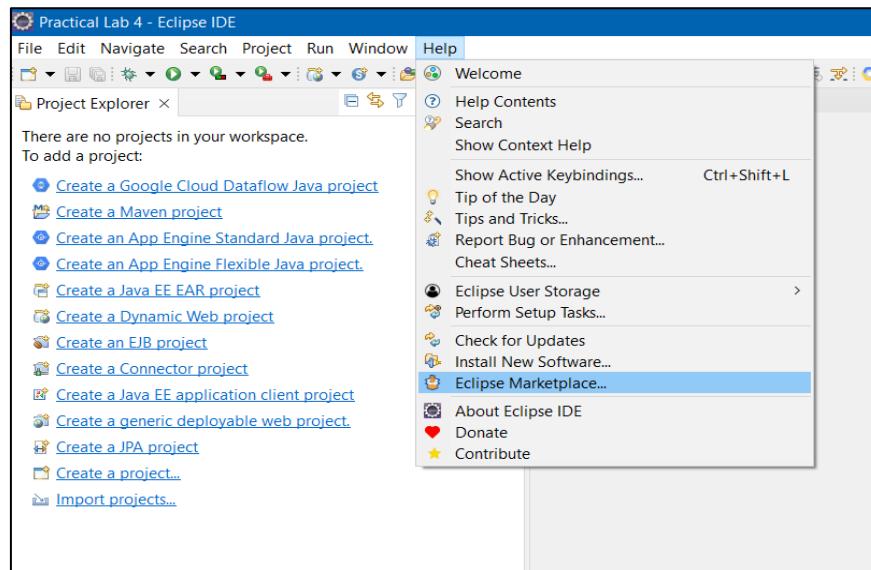


# PRACTICAL NO - 4

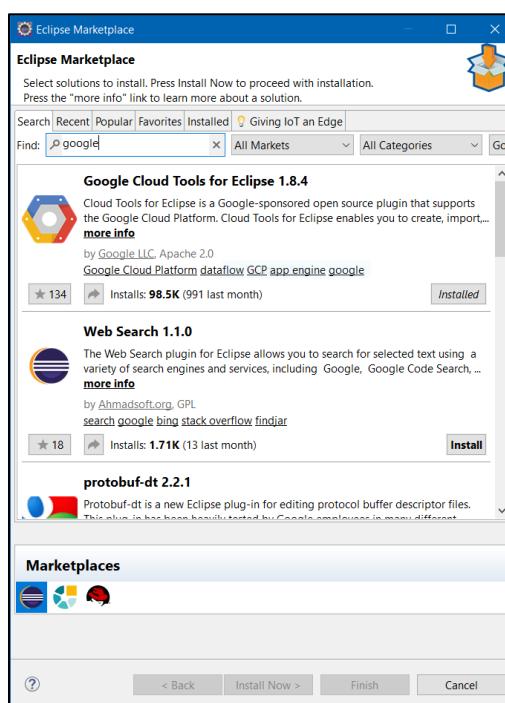
## ❖ Develop Application for Google App Engine.

### ● Steps:

- Download Eclipse IDE for Enterprise Java Developers and Install.
- Open Eclipse and Choose your workspace and click on it.
- Click on >>Help>>**Eclipse Marketplace**.

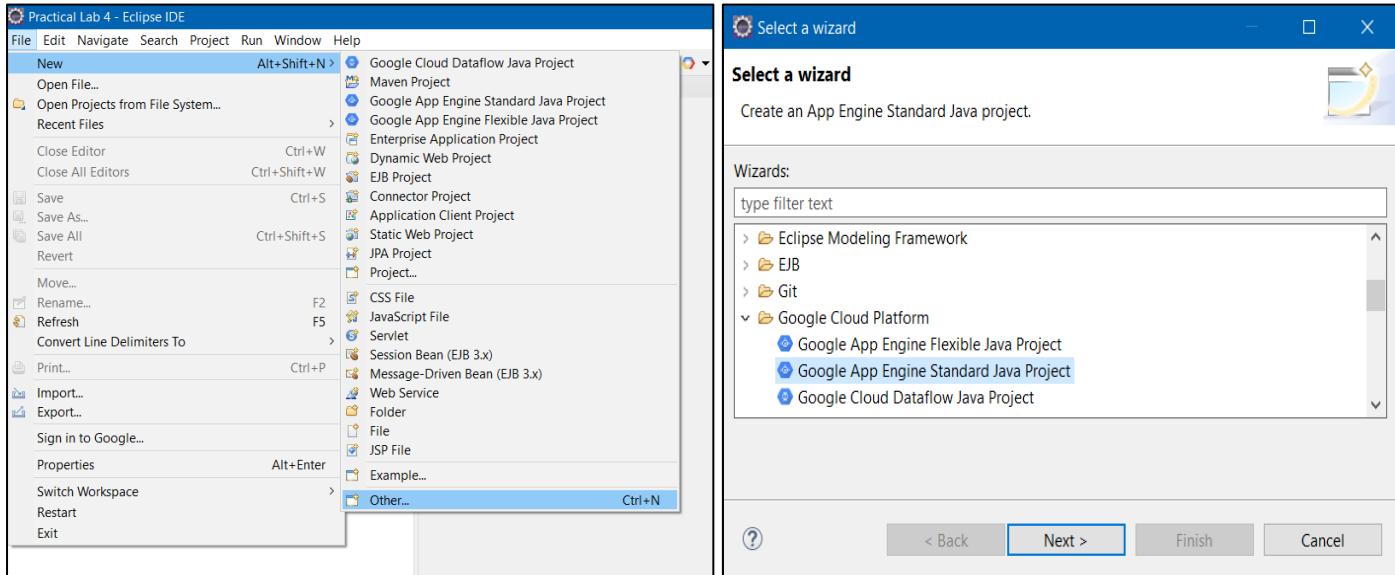


- In Find Search Box: Search for “Google cloud tools for eclipse” and click on the Install button. After complete installation of the plugin restart the Eclipse.

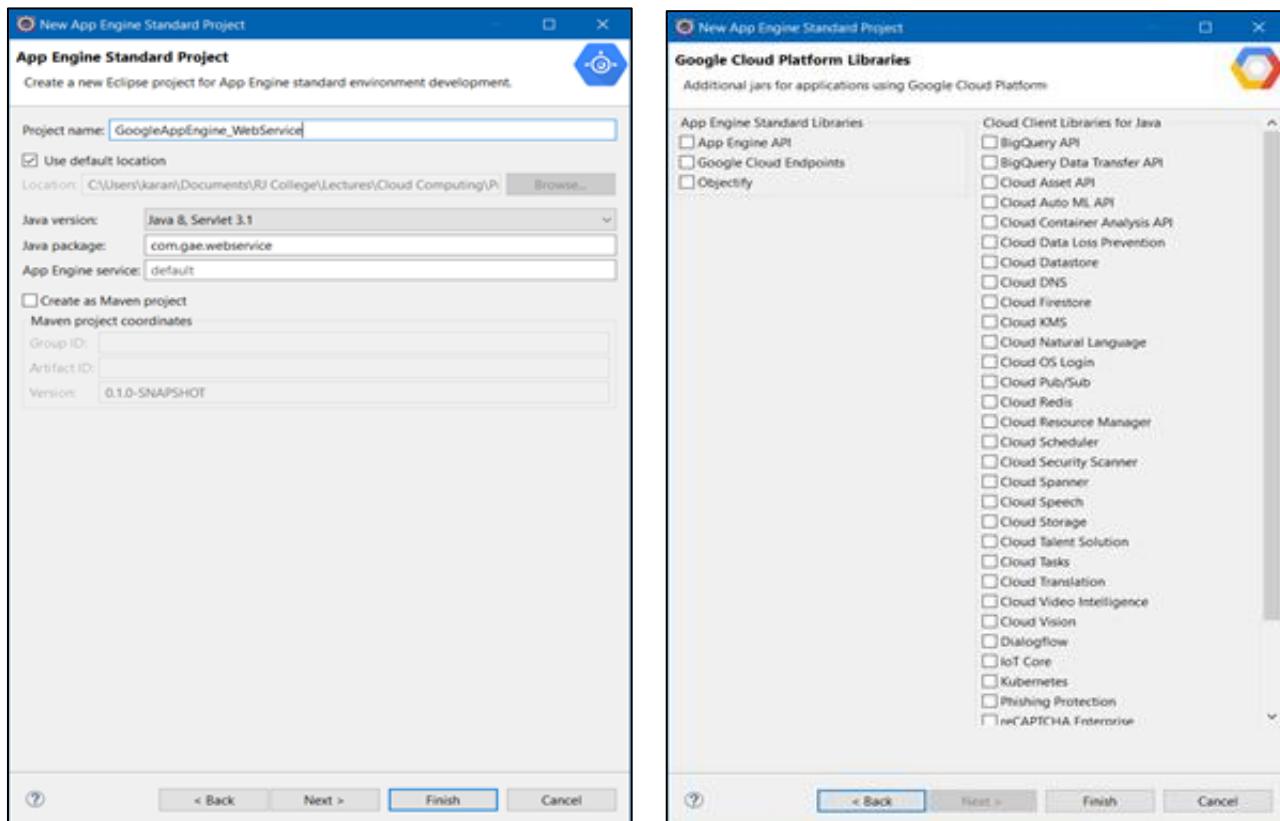


## PRACTICAL NO - 4

- Click on the ‘File’ menu and select >>New>>Other and search with the “Google” keyword and select Google App Engine Standard Java Project under Google Cloud Platform and click on Next



- Enter Project Name “**GoogleAppEngine\_WebService**” and Package Name “**com.gae.webservice**” and then Click on >>**Next**>>**Finish**.



## PRACTICAL NO - 4

- Open project hierarchy.
- Edit the file as required (Unedited file too can be used. Here the editing is done to “what should be displayed” on the browser). Save the file.

### index.html

```
1 <!DOCTYPE html>
2 <html xmlns="http://www.w3.org/1999/xhtml" lang="en">
3   <head>
4     <meta http-equiv="content-type" content="application/xhtml+xml; charset=UTF-8" />
5     <title>First Web Service</title>
6   </head>
7   <body>
8     <h1>First Web Service</h1>
9     <form action="add_me">
10       <table>
11         <tr>
12           <td>
13             <label>First number </label>
14             </td>
15             <td>
16               <input type="text" name="num1"/> <br/>
17             </td>
18           </tr>
19           <tr>
20             <td>
21               <label>Second number </label>
22             </td>
23             <td>
24               <input type="text" name="num2"/> <br/>
25             </td>
26           </tr>
27           <tr>
28             <td><button type="submit" name="calculate">Product and Sum </button><br/></td>
29           </tr>
30         </table>
31     </form>
32   </body>
33 </html>
```

### GAE\_WebService.java

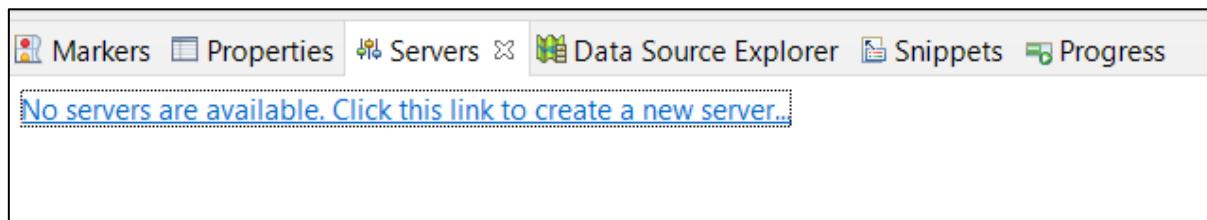
```
1 package com.gae.webservice;
2 import java.io.IOException;
3
4 @WebServlet("/GAE_WebService")
5
6 public class GAE_WebService extends HttpServlet
7 {
8     private static final long serialVersionUID = 1L;
9     public GAE_WebService()
10    {
11        super();
12    }
13    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
14    {
15        // declare variables
16        PrintWriter pw = null;
17        Integer FNO = 0;
18        Integer SNO = 0;
19
20        // get PrintWriter object
21        pw = response.getWriter();
22
23        // get form data (from request parameter)
24        FNO = Integer.parseInt(request.getParameter("num1"));
25        SNO = Integer.parseInt(request.getParameter("num2"));
26        int sum = FNO + SNO;
27        int product = FNO * SNO;
28        pw = response.getWriter();
29        pw.println("The Answer :" + sum + "\n The product :" + product);
30    }
31    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
32    {
33        doGet(request, response);
34    }
35}
```

# PRACTICAL NO - 4

## Web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.jsp</welcome-file>
    </welcome-file-list>
    <servlet>
        <servlet-name>Add</servlet-name>
        <servlet-class>com.gae.webservice.GAE_WebService</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>Add</servlet-name>
        <url-pattern>/add_me</url-pattern>
    </servlet-mapping>
</web-app>
```

- Configuring “App Engine Standard” Server. Click on the link to create a new server under Server tab.



The screenshot shows two dialog boxes side-by-side:

**Left Dialog (Define a New Server):**

- Header: "Define a New Server" with a "New Server" icon.
- Text: "Choose the type of server to create".
- Section: "Select the server type:" with a "type filter text" input field.
- Tree View: A tree view showing categories like Apache, Basic, Google, IBM, and App Engine Standard. "App Engine Standard" is selected.
- Text: "Publishes and runs projects on a local App Engine server".
- Form Fields:
  - Server's host name: "localhost"
  - Server name: "App Engine Standard at localhost"
  - Server runtime environment: "App Engine Standard Runtime" with an "Add..." button and a "Configure runtime environments..." link.
  - Server port: "8080"
- Buttons: "?", "< Back", "Next >", "Finish", "Cancel".

**Right Dialog (Add and Remove):**

- Header: "Add and Remove" with a "New Server" icon.
- Text: "Modify the resources that are configured on the server".
- Section: "Move resources to the right to configure them on the server".
- Section: "Available" (left): An empty list.
- Section: "Configured" (right): A list containing "GoogleAppEngine\_WebServ".
- Buttons: "Add >", "< Remove", "Add All >>", "<< Remove All".
- Buttons: "?", "< Back", "Next >", "Finish", "Cancel".

- Start the “App Engine Standard” server.

## PRACTICAL NO - 4

### Output:

The screenshot shows a web browser window with the title bar "App Engine Standard at localhost". The address bar displays "http://localhost:8080/". The main content area is titled "First Web Service". It contains two input fields: "First number" and "Second number", each with a corresponding empty text box. Below these fields is a button labeled "Product and Sum".

The screenshot shows a web browser window with the title bar "App Engine Standard at localhost". The address bar displays "http://localhost:8080/add\_me?num1=12&num2=10&calculate=". The main content area displays the text "The Answer :22 The product :120".

# PRACTICAL NO - 5

## ➤ **Implement Virtualization Using VMWare ESXi Server and managing with vSphere Client.**

VMware ESXi is the bare-metal hypervisor in the VMware vSphere virtualization platform. As a bare-metal hypervisor for creating and running virtual machines (VMs), VMware ESXi runs on top and accesses the hardware directly without the need to install an operating system. This direct access to hardware allows it to perform better, run faster and be more scalable than other types of hypervisors. This makes VMware ESXi ideal for use in a large-scale virtual desktop infrastructure (VDI), in conjunction with the other components in the VMware vSphere platform.

1. Press the Enter key to boot from the installation media.

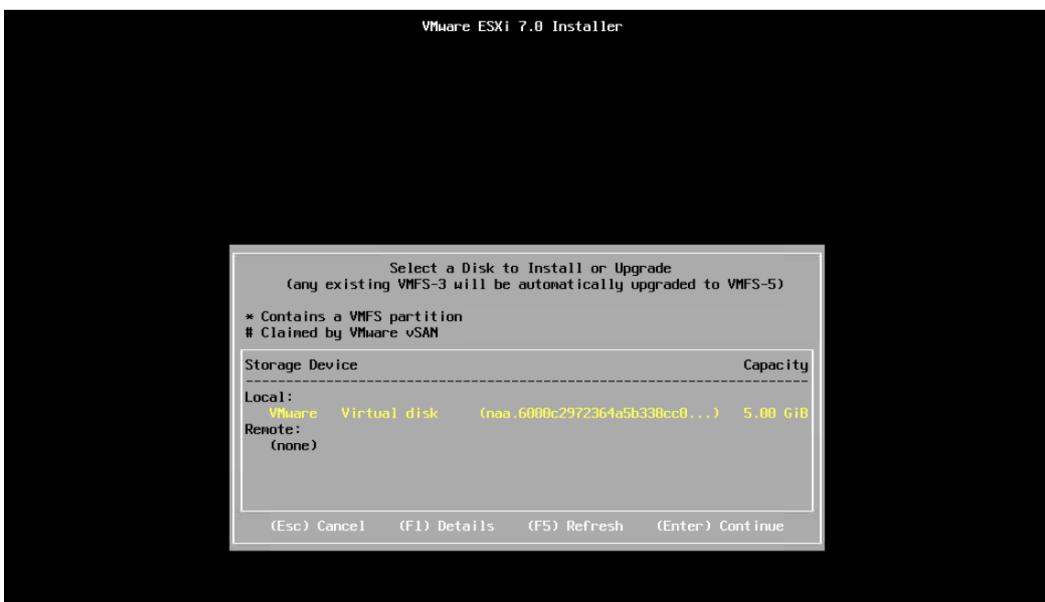


2. Press the Enter key to start the installation.

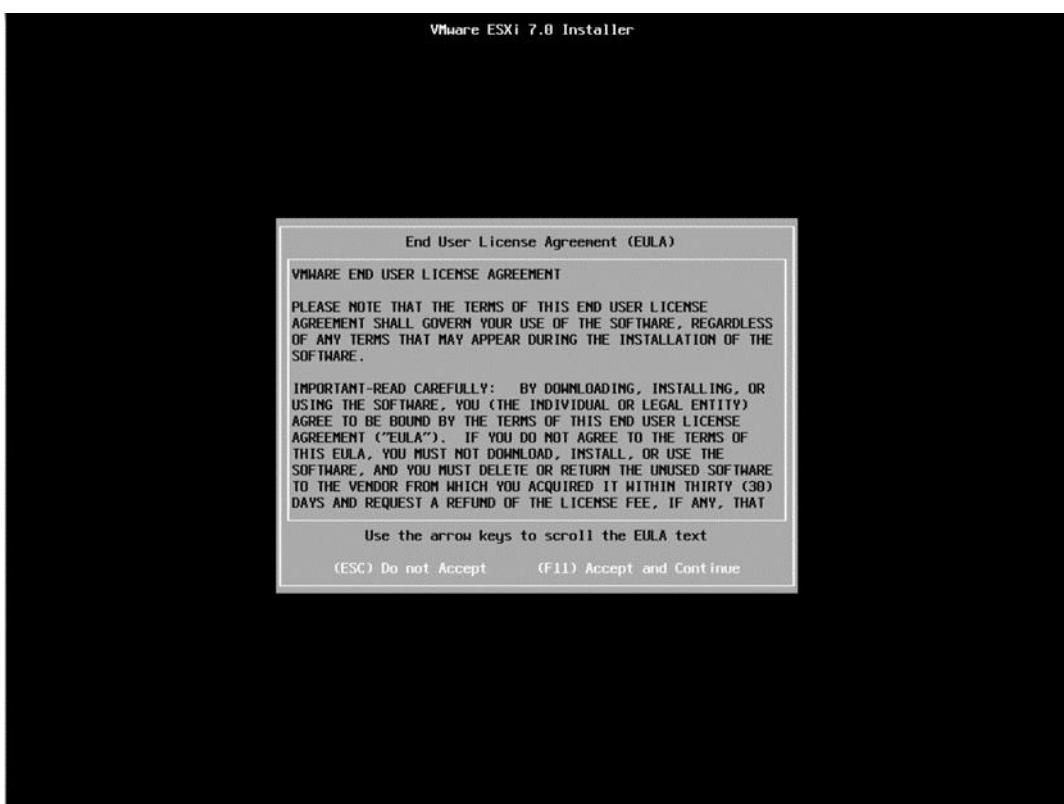


# PRACTICAL NO - 5

Press the Enter key to select the drive to install ESXi on.

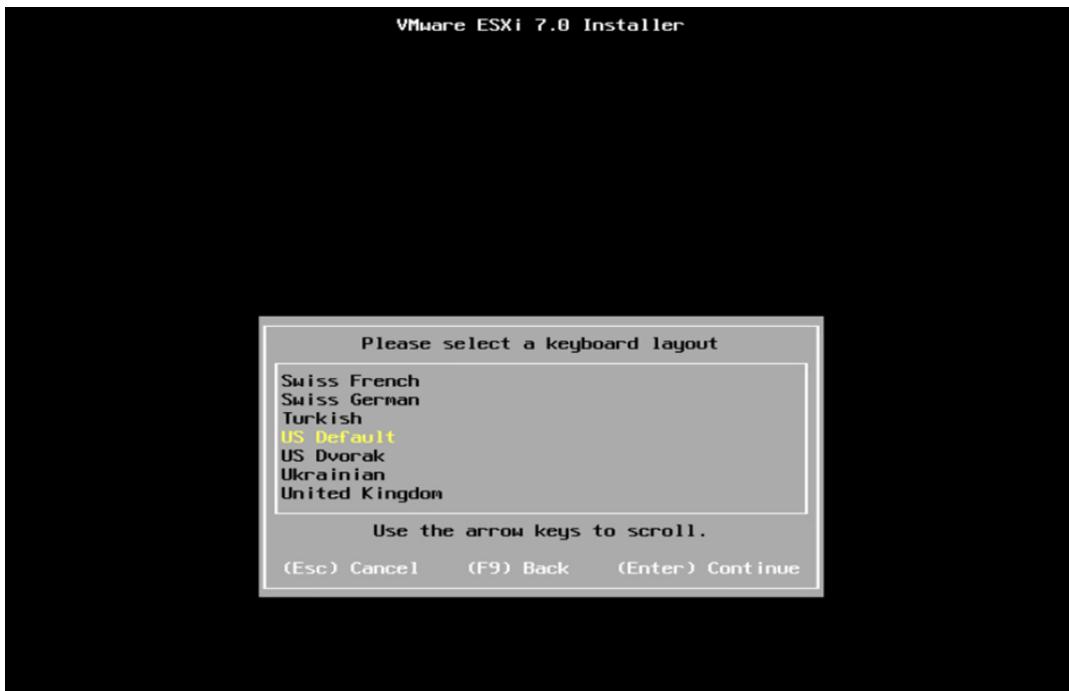


3. Press the Tab key to Accept the EULA and continue.



## PRACTICAL NO - 5

4. Press the Enter key to select the US Default keyboard



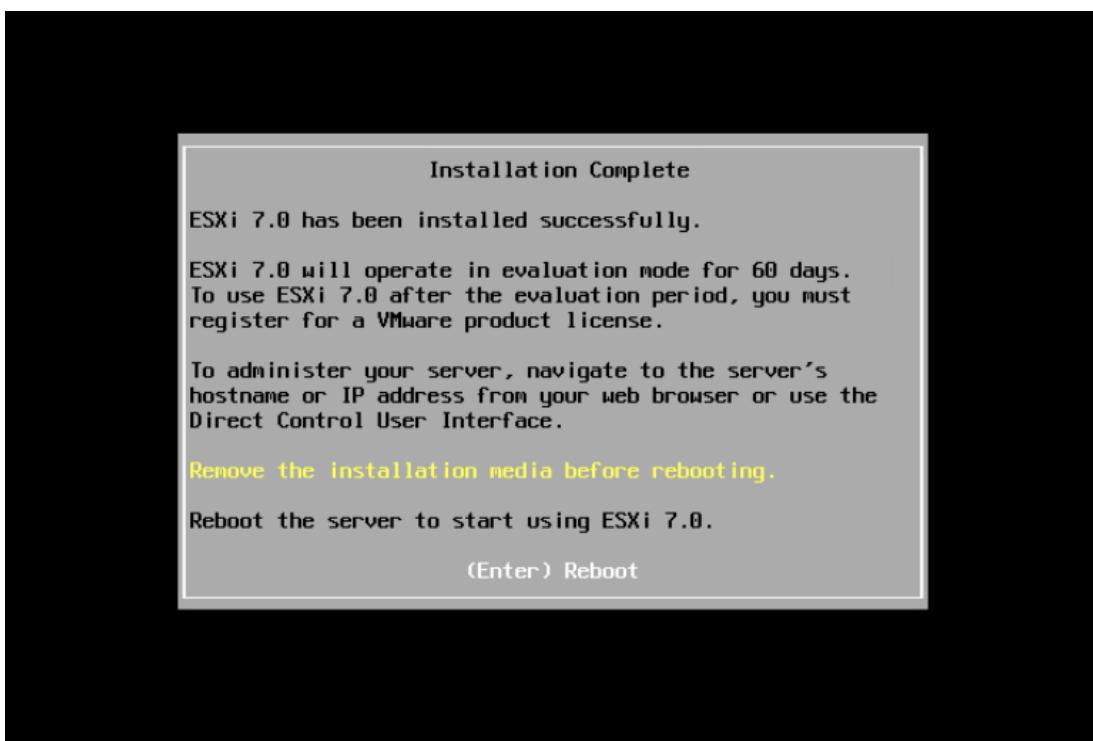
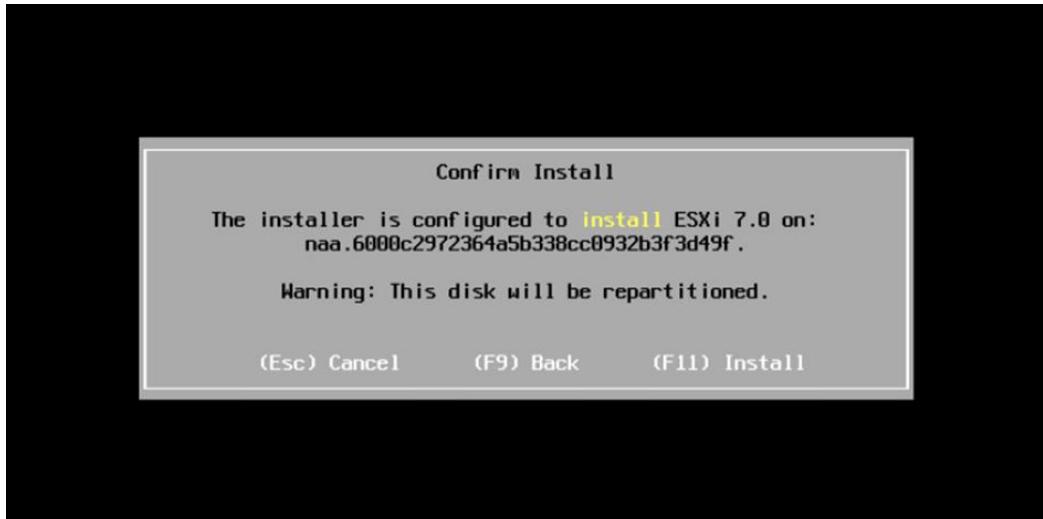
5. For the Root password, type VMware1!
6. Press the down-arrow Key.
7. Confirm the password by typing VMware1! again.



8. Press the Enter key.

## PRACTICAL NO - 5

9. Press the Tab key to start the installation.



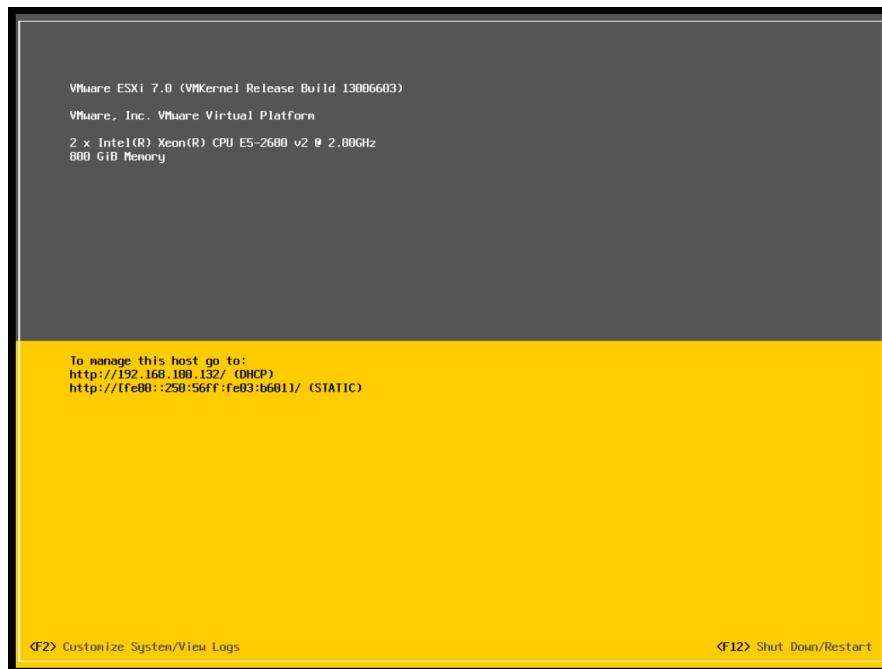
10. Press the Enter key to reboot the server.

# PRACTICAL NO - 5

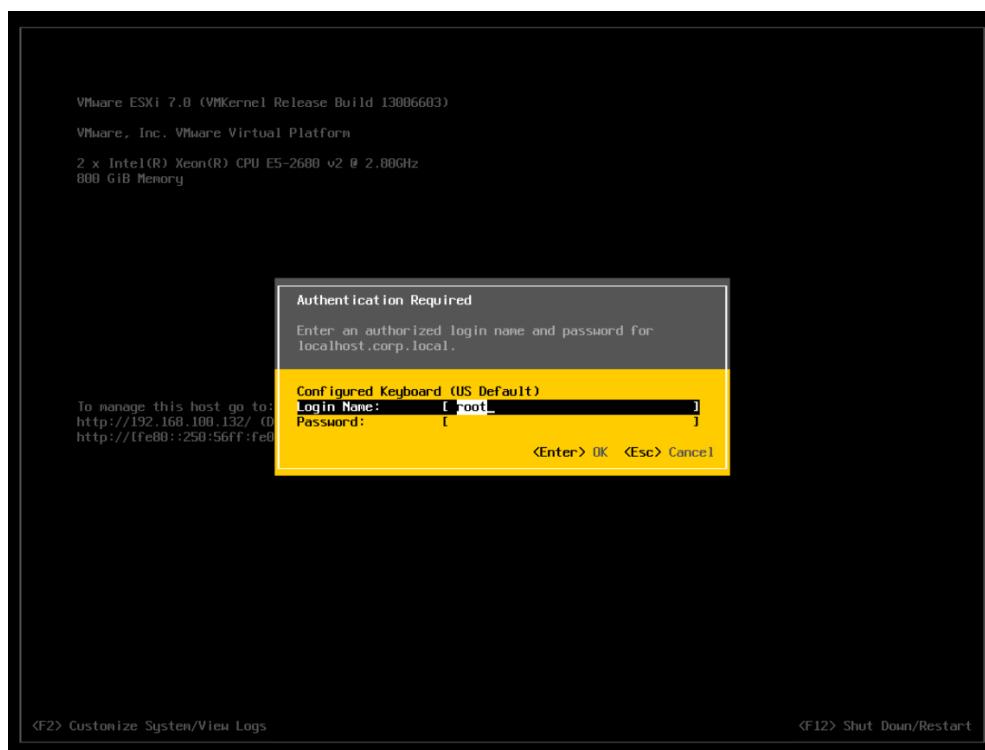
**NOTE:** At this point, you would want to remove the installation media prior to restarting.

Now that the server is back up and running, we will need to configure the settings for the Management Network. This will allow us to connect the ESXi host to the network and at add the host to the vCenter Server.

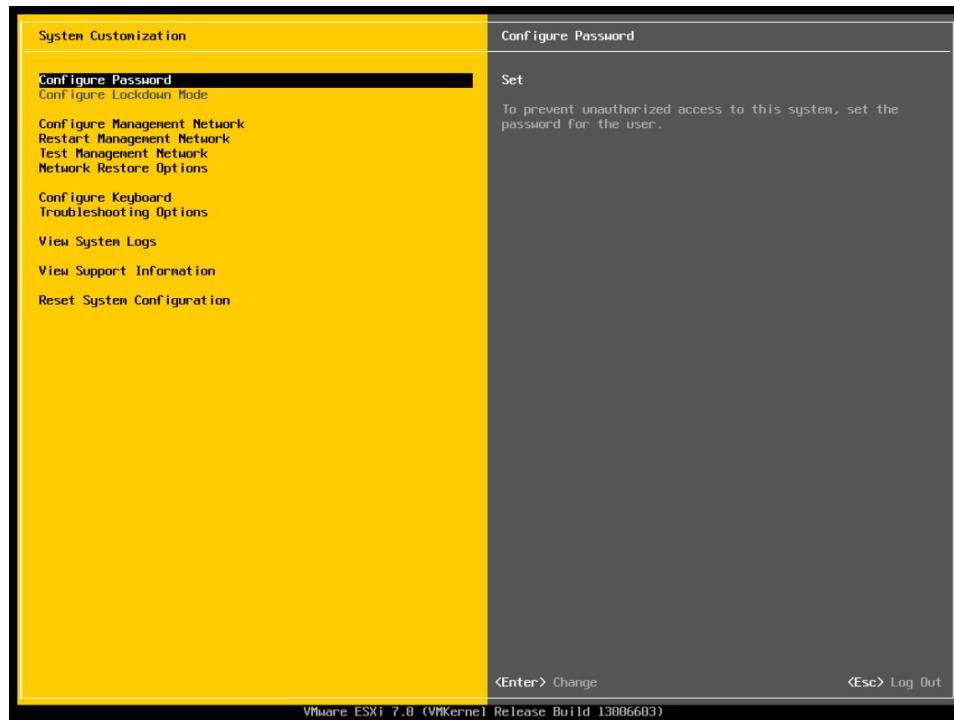
1. Press the F2 key to customize the system,



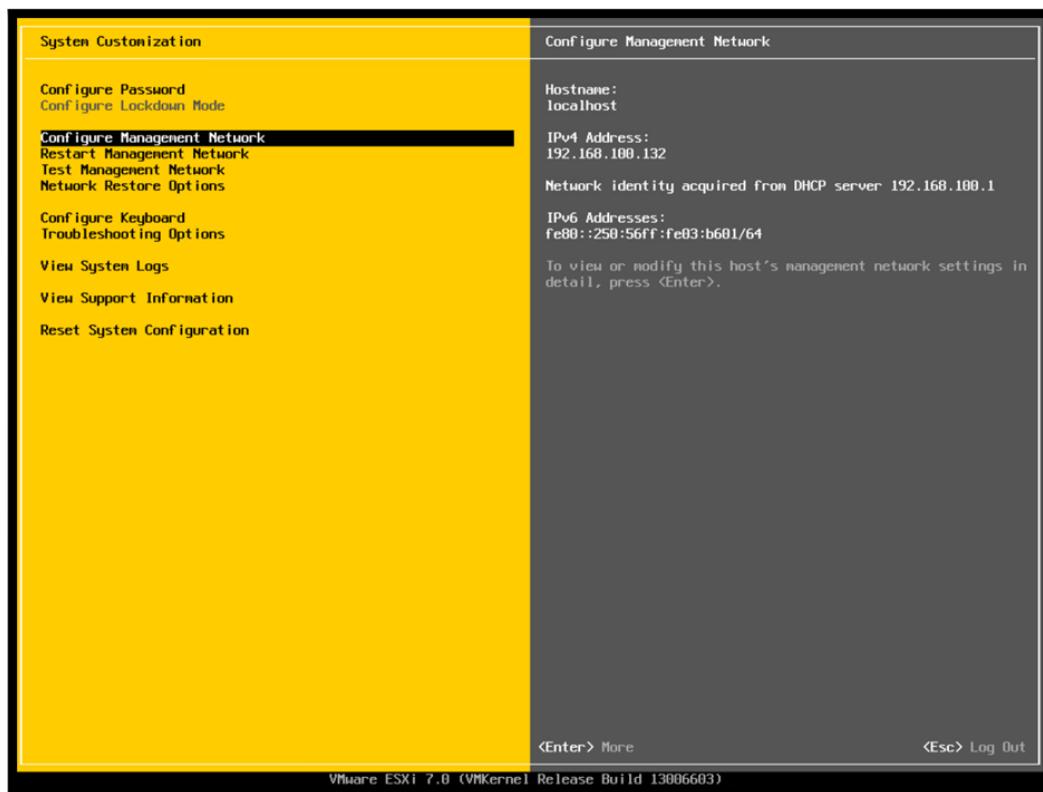
2. Press the down-arrow key to move to the password field.
3. Type VMware1! for the password.
4. Press the Enter key.



# PRACTICAL NO - 5

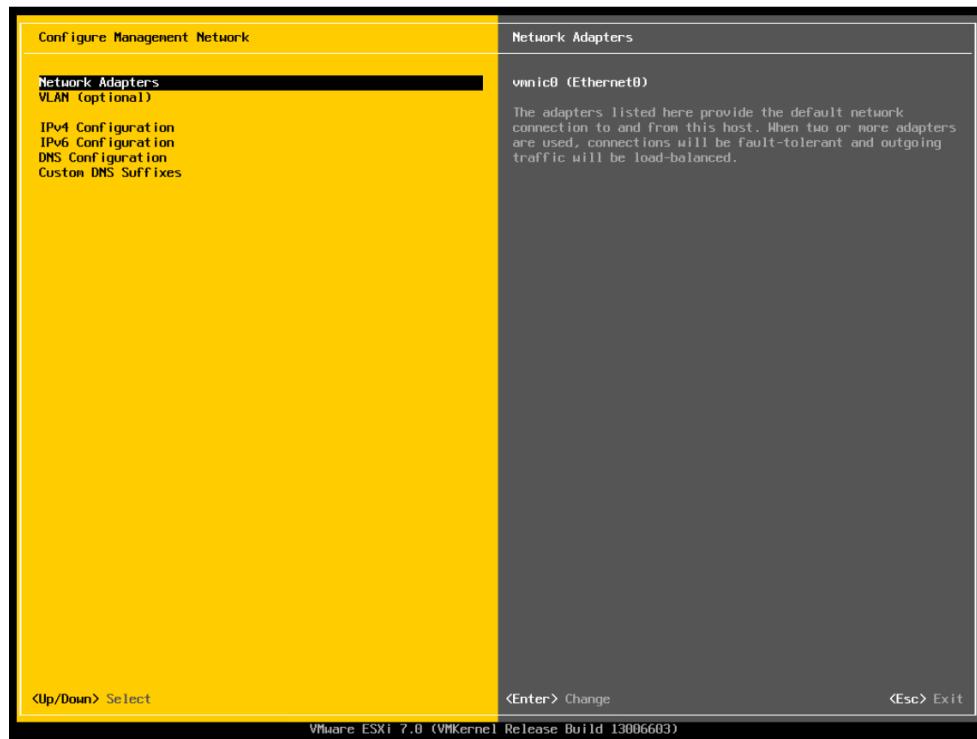


5. Use the down arrow key to select Configure Management Network.
6. Press the Enter key.

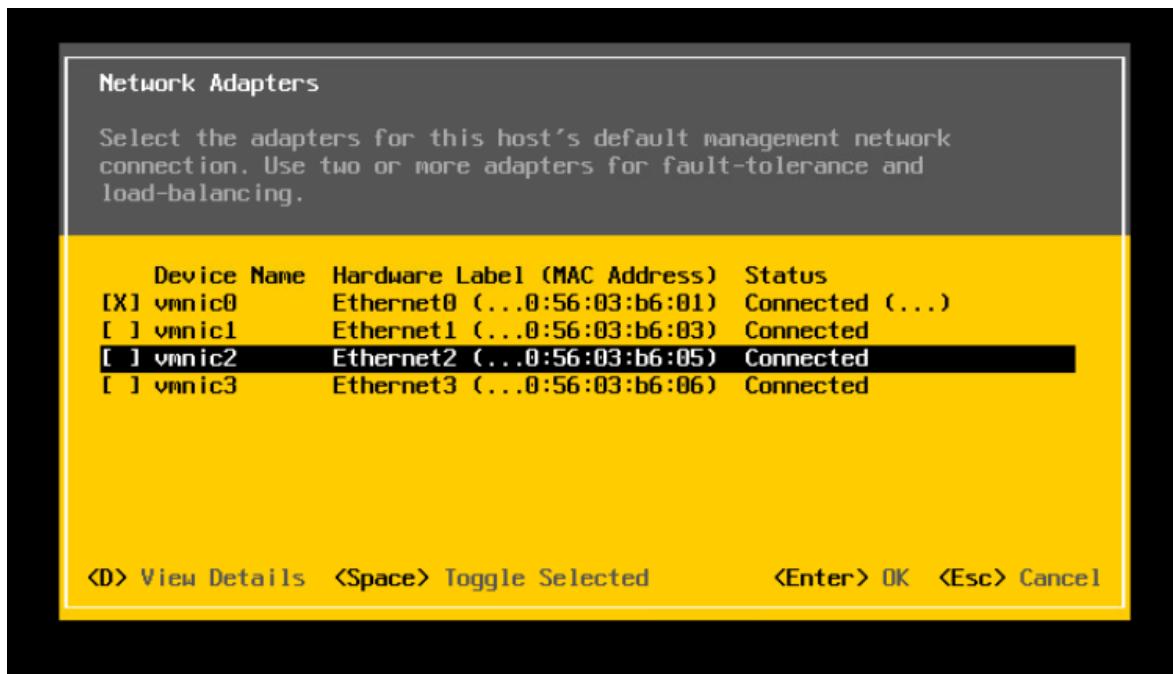


# PRACTICAL NO - 5

7. Press the Enter key again to select Network Adapters.



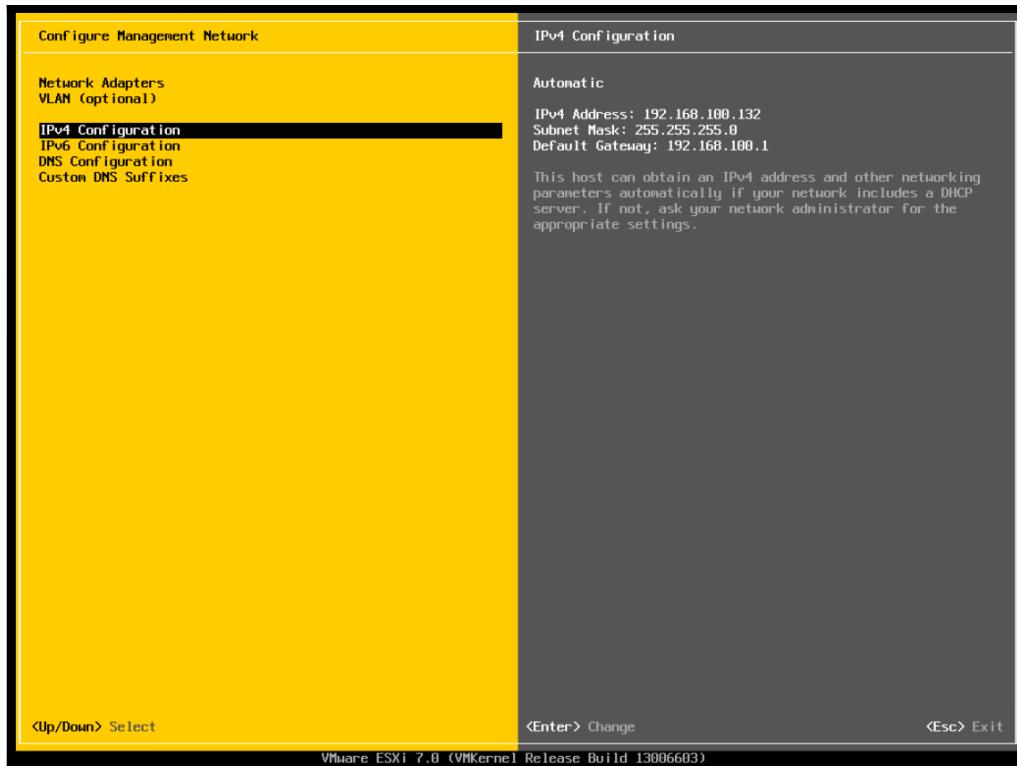
8. Use the down-arrow key to highlight vmnic2.
9. Press the Spacebar to select vmnic2.
10. This will let us use two connections for load balancing and fault tolerance.



11. Press the Enter key.

# PRACTICAL NO - 5

12. Use the down-arrow key to highlight IPv4 Configuration.
13. Press the Enter key.



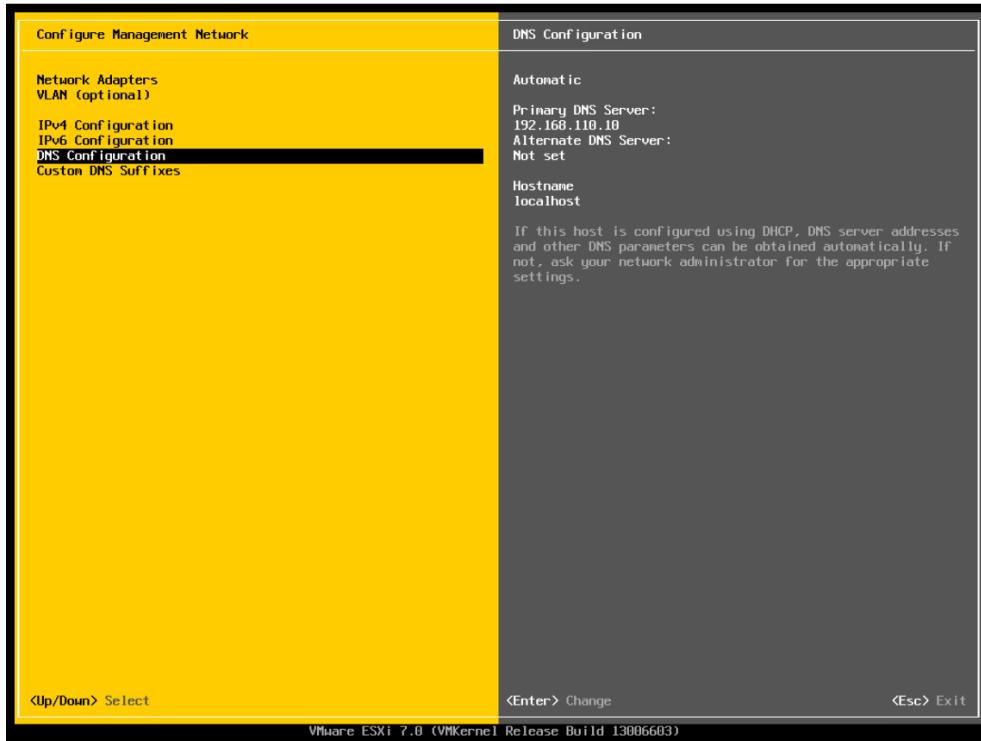
14. Use the down-arrow key to select Set static IPv4 address and network configuration.
15. Press the Spacebar to select it.



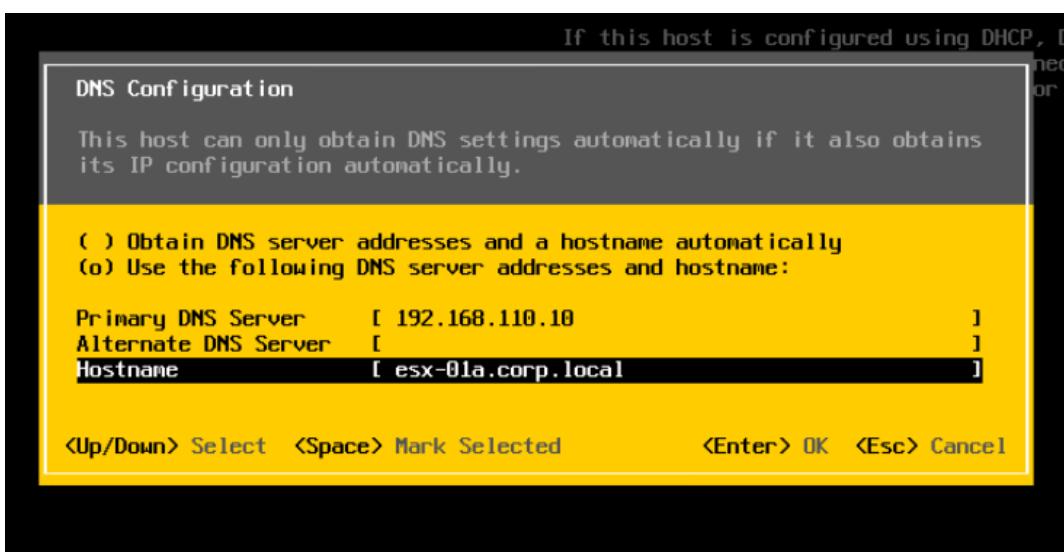
16. Use the down-arrow key to move to the IPv4 Address field.
17. Type 192.168.110.51 for the IP Address.
18. Use the down-arrow key to move to the Default Gateway field.
19. Type 192.168.110.10 for the Default Gateway.
20. Press the Enter key.

# PRACTICAL NO - 5

21. Use the down-arrow key to highlight DNS Configuration.
22. Press the Enter key.

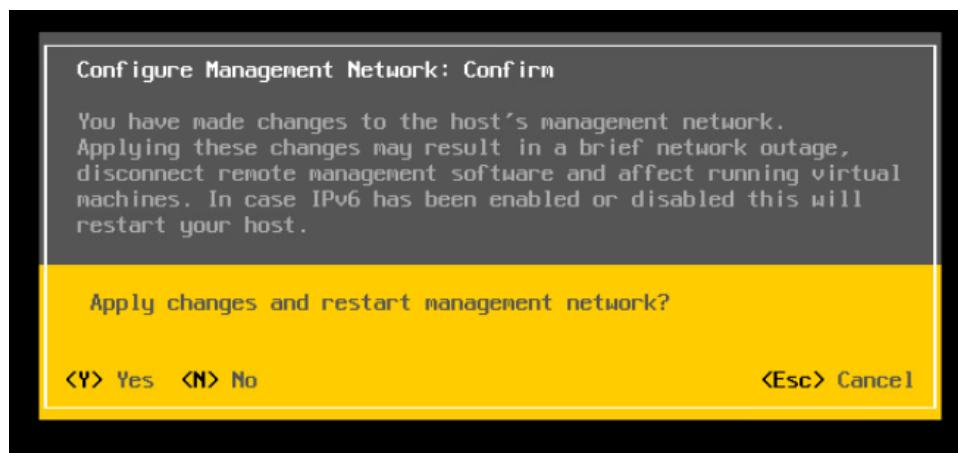


23. Use the down-arrow key to highlight Use the following DNS server addresses and hostname.
24. Press the spacebar to select it.
25. Use the down-arrow key to move to the Hostname field.
26. Type esx-01a-corp.local for the hostname.
27. Press the Enter key.
28. Press the Tab key to apply these changes.

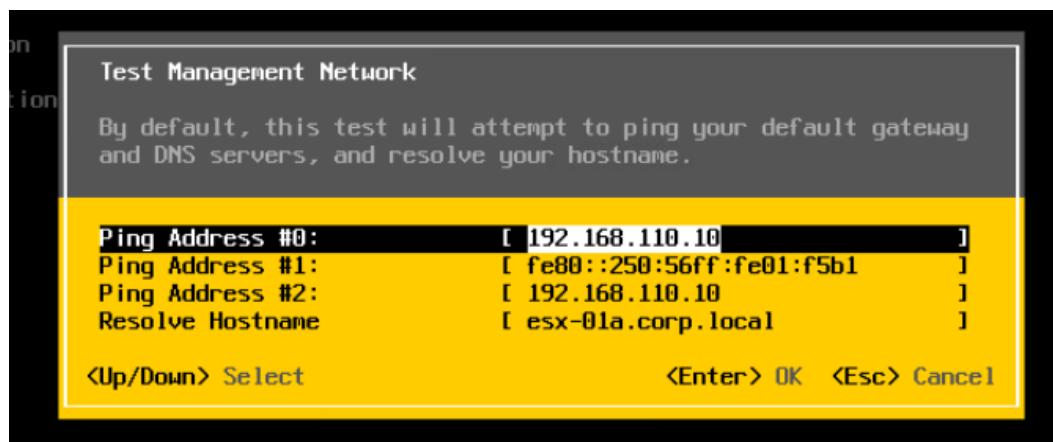
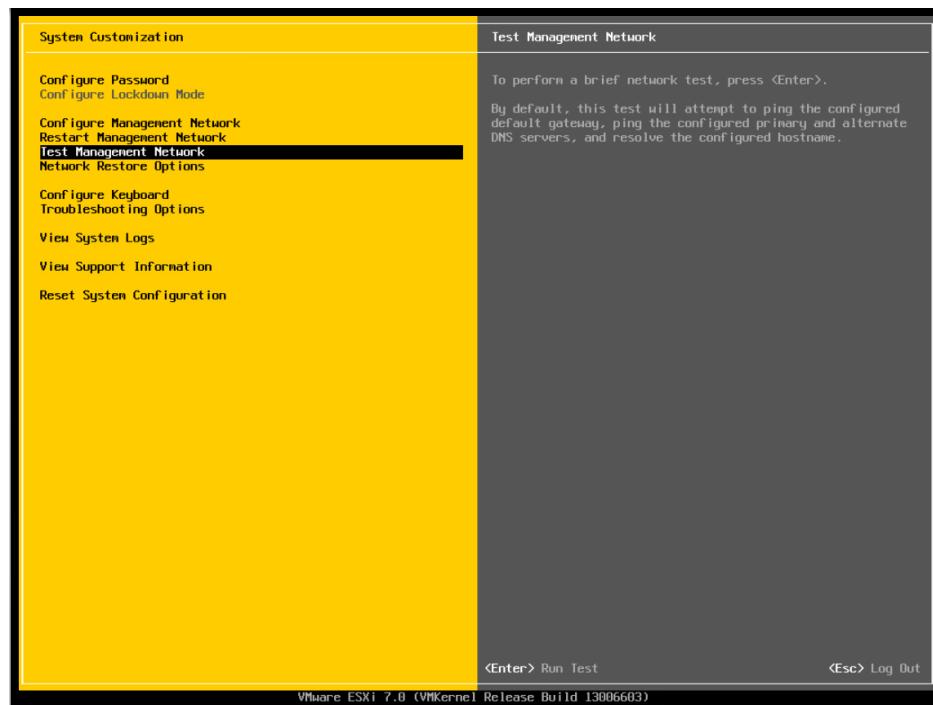


# PRACTICAL NO - 5

29. Press the Y key to apply the changes and restart the management network.

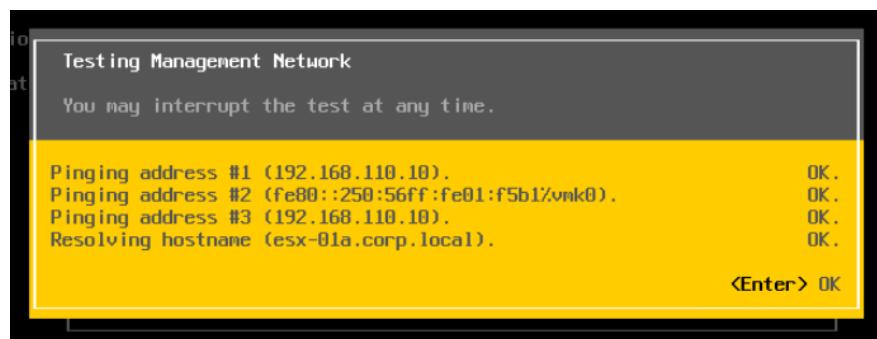


30. Use the down-arrow key to highlight Test Management Network.  
31. Press the Enter key.

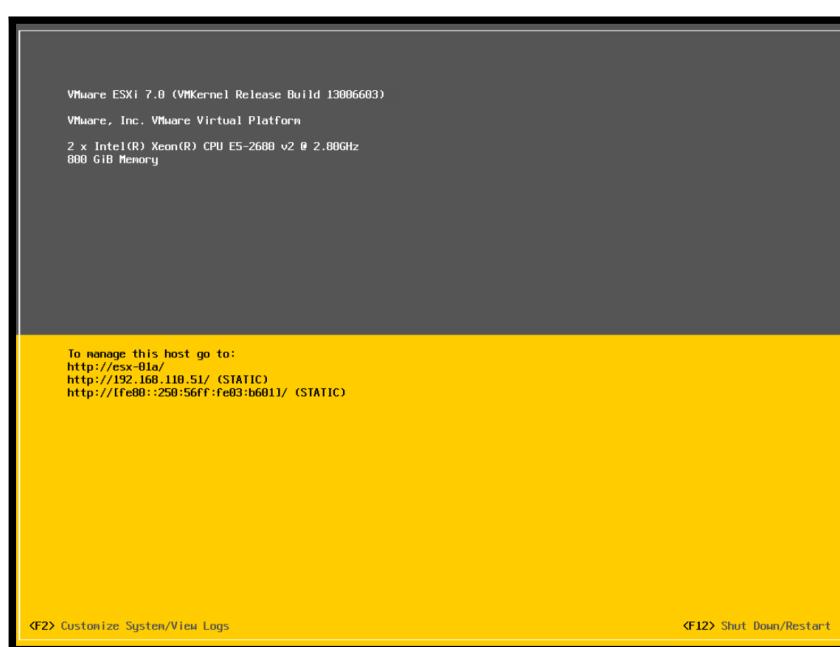
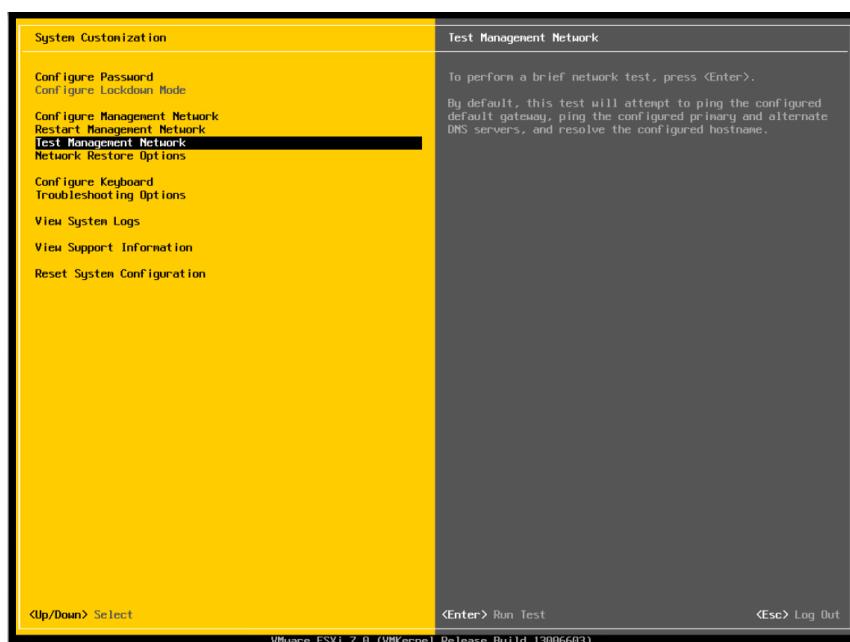


# PRACTICAL NO - 5

32. Press the Enter key again to start the test.



33. Press the Enter key once the test has completed successfully.  
34. Press the Tab key to log out of the console.



# PRACTICAL NO - 5

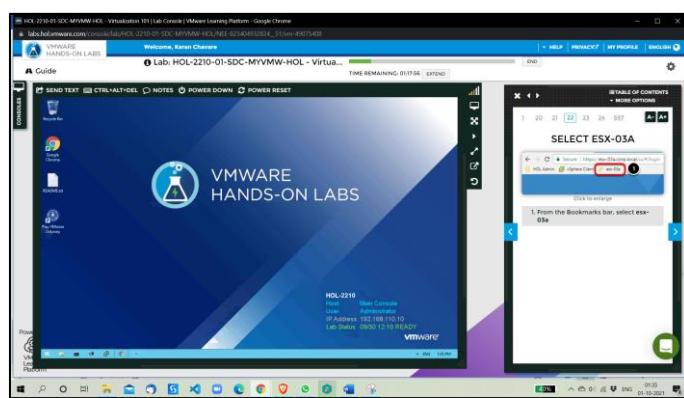
## ESXI HOST CLIENT

The VMware Host Client is an HTML5-based client that is used to connect to and manage single ESXi hosts.

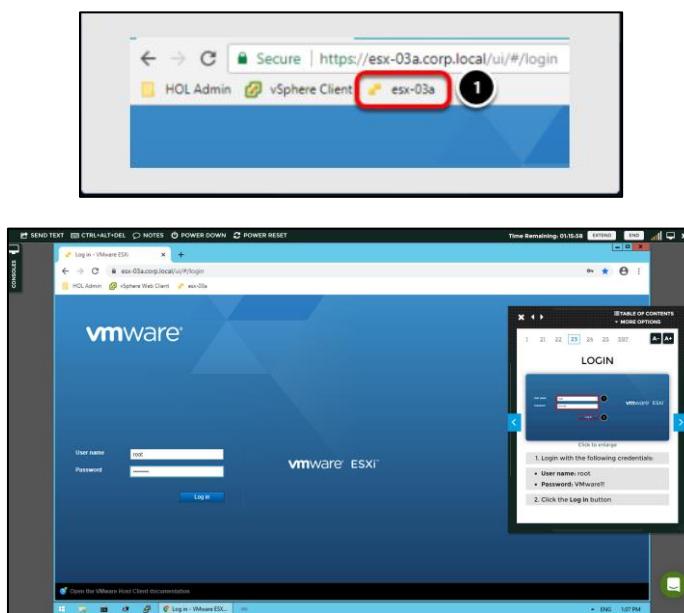
You can use the VMware Host Client to perform administrative and basic troubleshooting tasks, as well as advanced administrative tasks on your target ESXi host. You can also use the VMware Host Client to conduct emergency management when vCenter Server is not available.

It is important to know that the VMware Host Client is different from the vSphere Web Client, regardless of their similar user interfaces. You use the vSphere Web Client to connect to vCenter Server and manage multiple ESXi hosts, whereas you use the VMware Host Client to manage a single ESXi host.

1. Click on the **Chrome Icon** on the Windows Quick Launch Task Bar



2. From the Bookmarks bar, select **esx-03a**



3. Login with the following credentials:
4. Username: **root**
5. Password: **VMware1!**
6. Click the **Log in** button

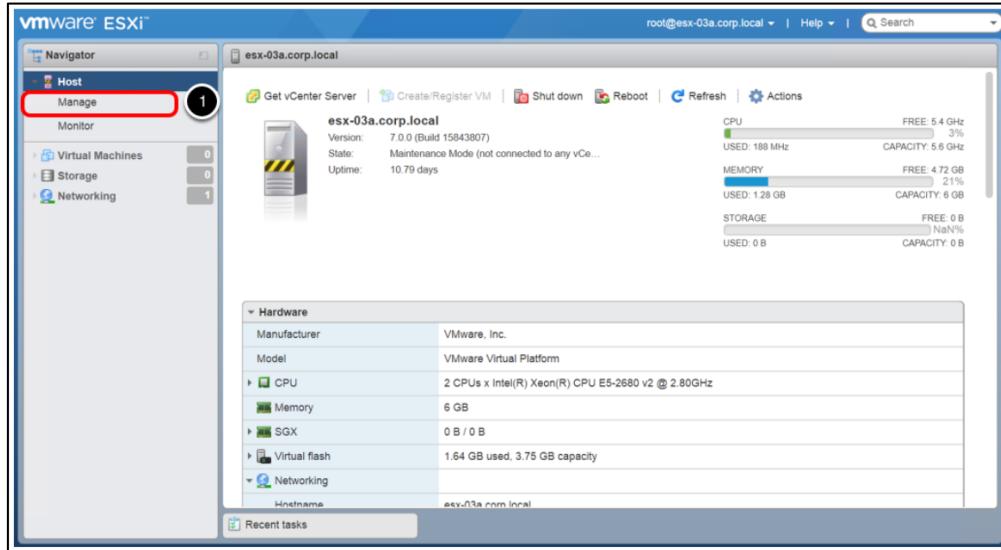
# PRACTICAL NO - 5

The ESXi Host, in this case, **esx-03a**, can now be directly managed. This can be useful in test/dev environments where a vCenter Server is not present or in a production environment where the vCenter Server is not reachable.

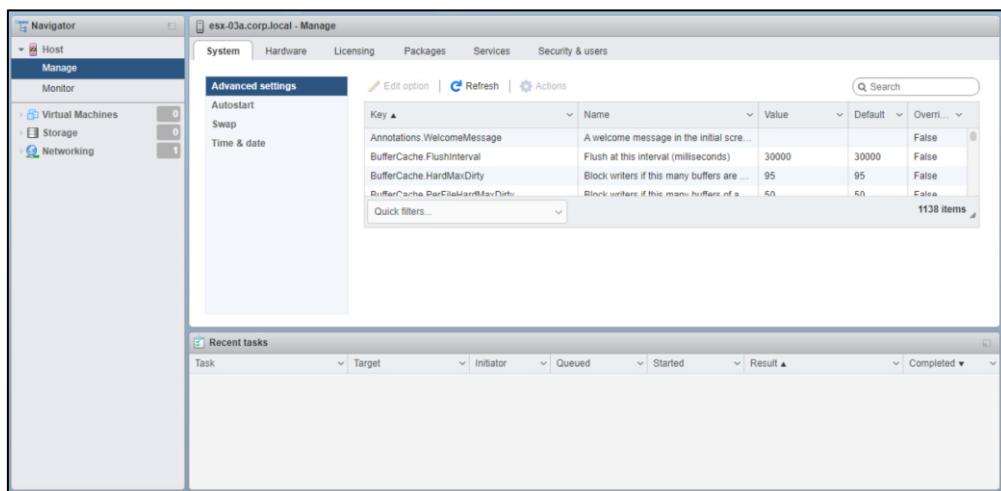
The initial screen shows high-level details and recent tasks. There are also various power options for the host and an Actions menu for the most common tasks. Note that the server is currently in Maintenance Mode, which will be discussed in a future lesson. Click to minimize the Recent tasks interface to gain more room.

- Click on **Manage**

- Click on **System tab**



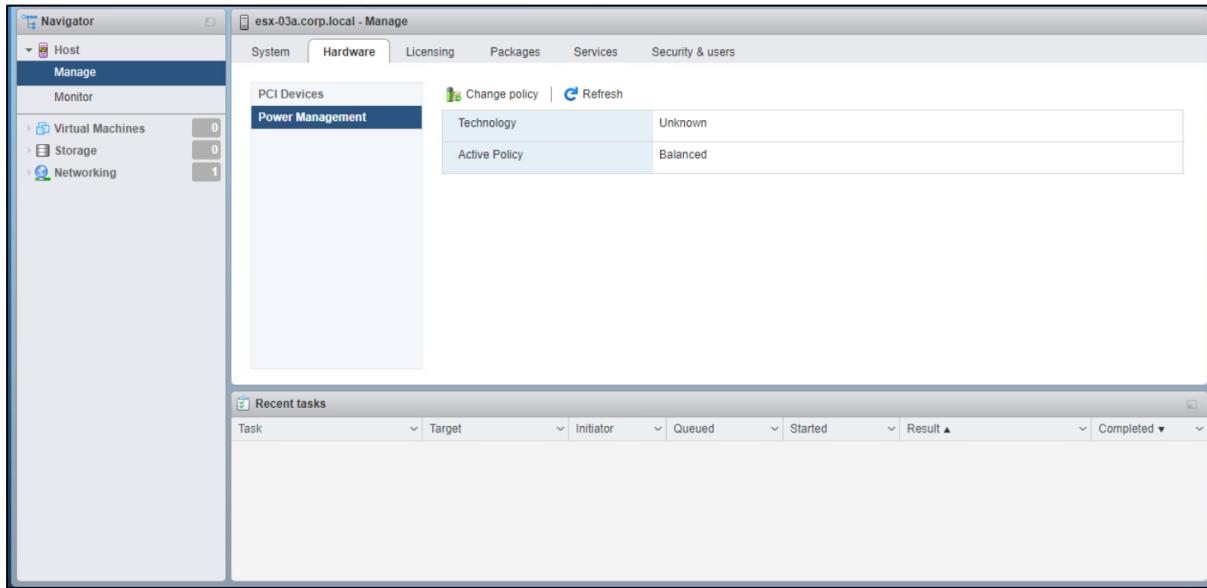
On the **System** tab, the most common options set here are the date and time for the host. It can be set and synchronized with an NTP server or set manually. In addition, AutoStart settings for the host can be configured here as well.



# PRACTICAL NO - 5

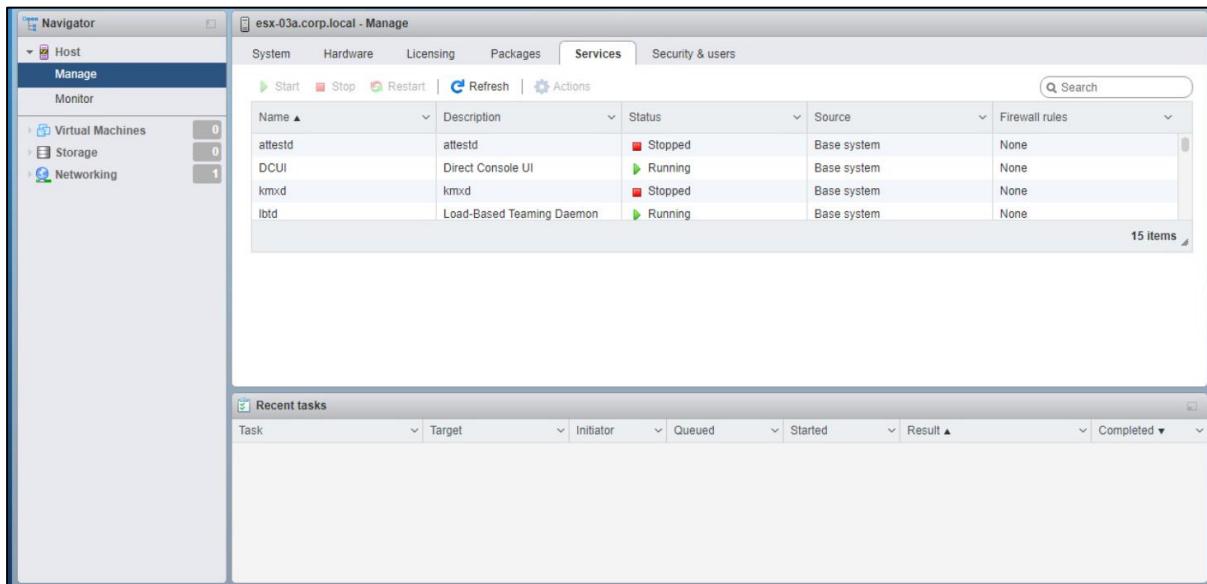
- Click on the **Hardware** tab
- 1. Click **Power Management**

This is where power management policies can be set for the host.



- Click the **Services** tab

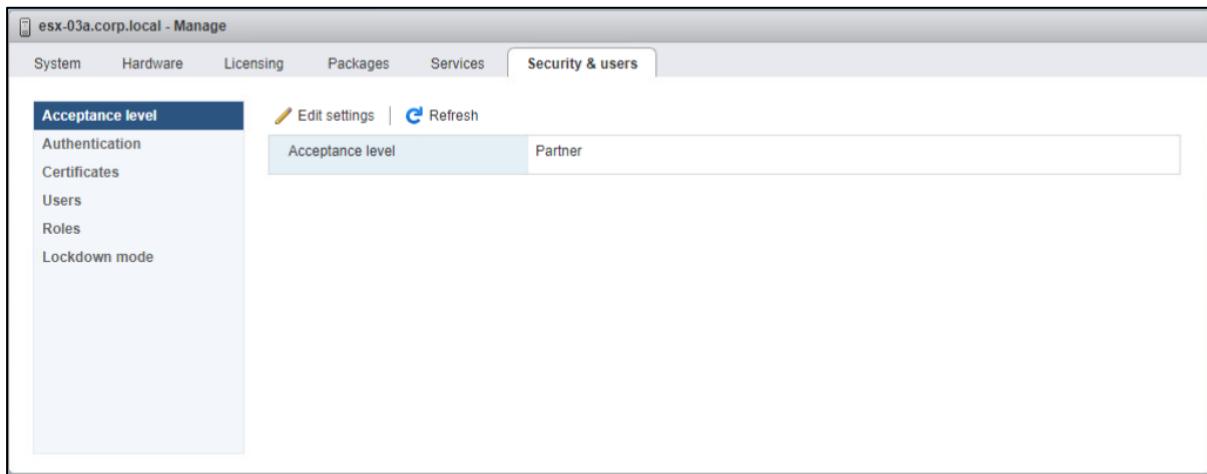
Services like SSH access and the Direct Console UI can be stopped and started from this screen.



# PRACTICAL NO - 5

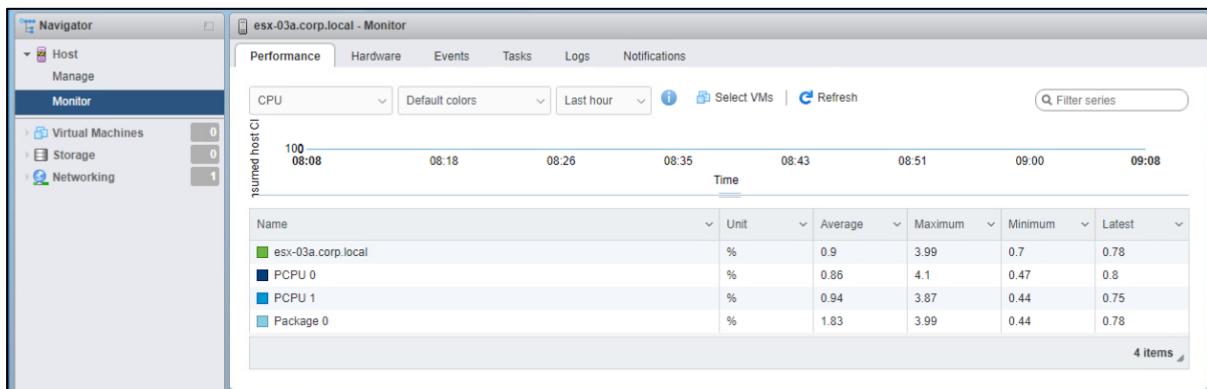
- Click on **Security & users**

On the Security & Users tab, security options such as authentication to Active Directory and Certificates can be set here. There is also the ability to create additional roles and user accounts for the host itself. This option uses accounts that are local only to the host and not shared with any other hosts or vCenter Server. vCenter Server is set up to use single sign-on which makes account management much easier. This will be reviewed in the lessons that follow.



- Click on **Monitor**

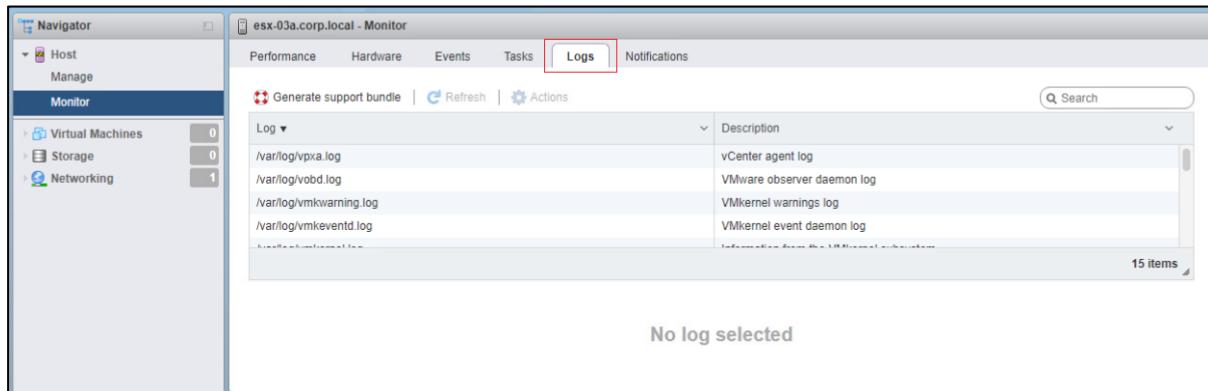
The Monitor section includes Performance Charts, Hardware monitoring, an event log and other useful monitoring information.



# PRACTICAL NO - 5

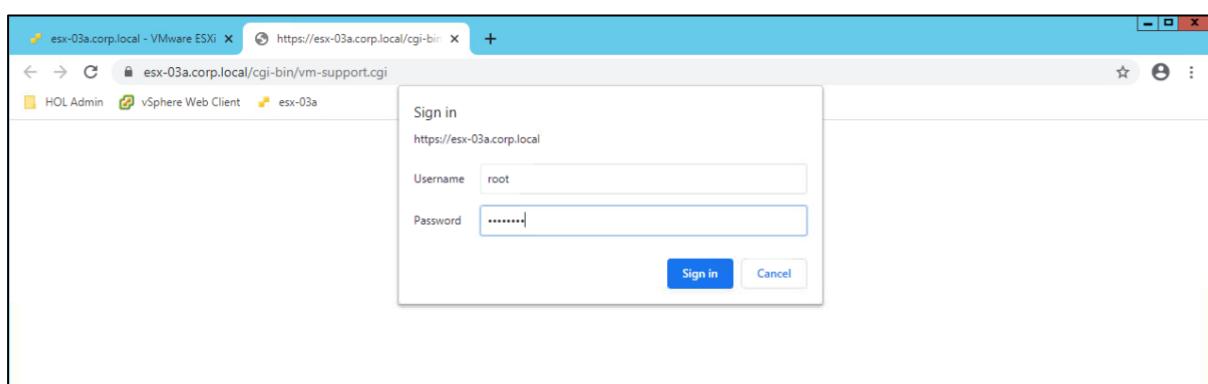
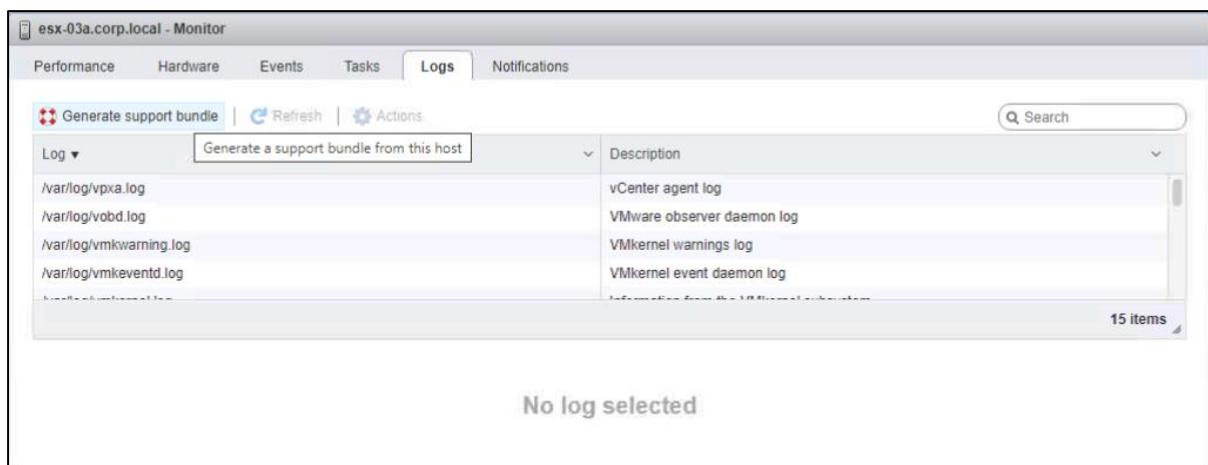
- Click the **Logs** tab

On the Logs tab, a support bundle can be created that includes log files and system information that can be helpful in troubleshooting issues.



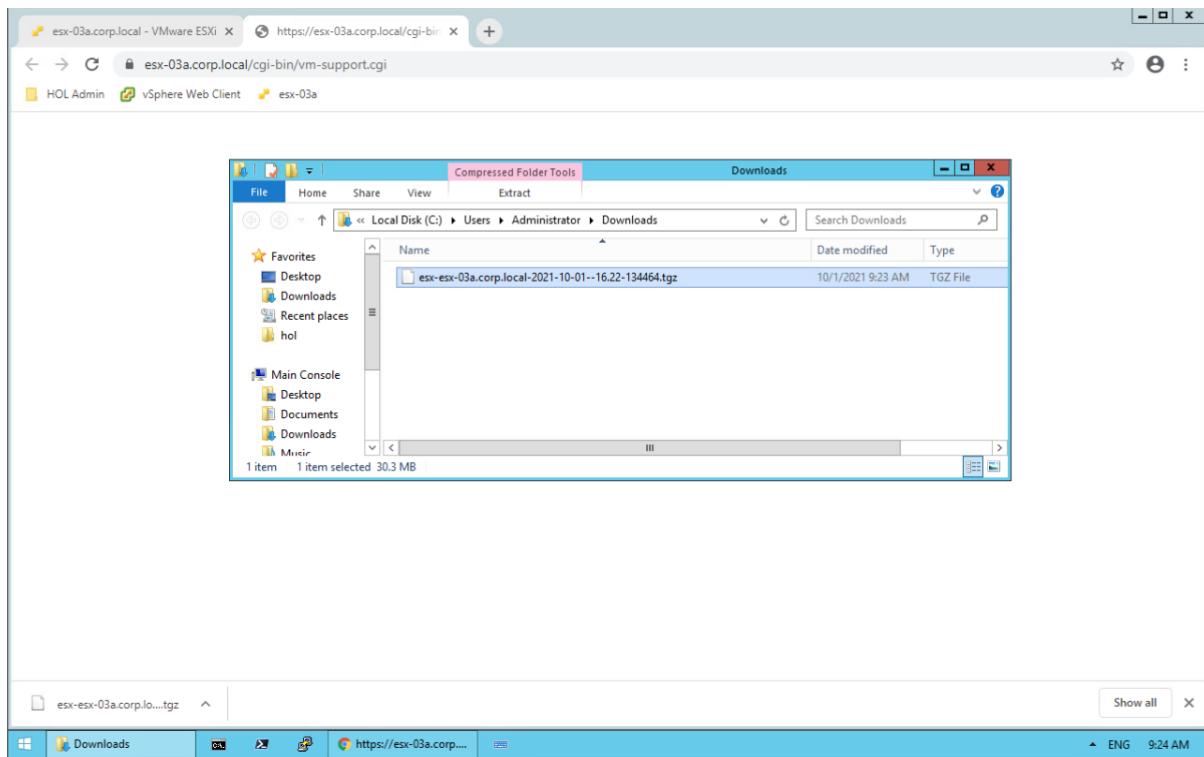
- Click the **Generate Support Bundle** button.

This operation will automatically download the support file. It will take a couple of minutes. You may be asked to provide credentials. Use the same information you used to log in and click Sign In button.



## PRACTICAL NO - 5

- a. Click on the **arrow on the downloaded file** and select **Show in folder**.
- b. A **pop-up window** will appear with the downloaded support file. Review file if needed.
- c. **Close window** when finished.



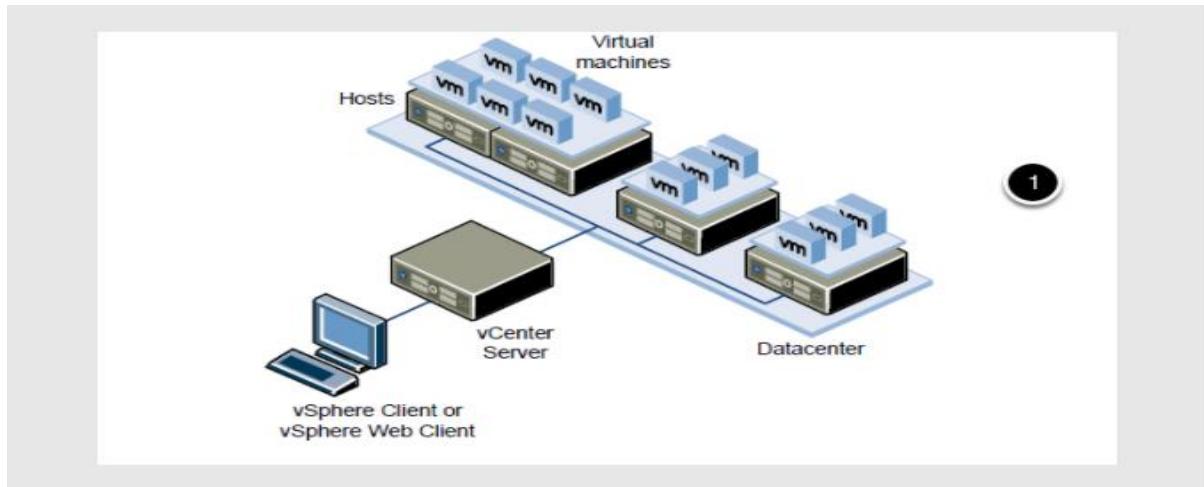
In addition to managing and monitoring the host, Virtual Machines can be created, Storage and Networking can be configured at the host level.

The ESXi Host Client can be very useful in situations where a vCenter Server is not present to manage the host. However, when a vCenter Server is present, it is the preferred option and provides better tools to manage your infrastructure.

# PRACTICAL NO - 5

## VCENTER 7 OVERVIEW

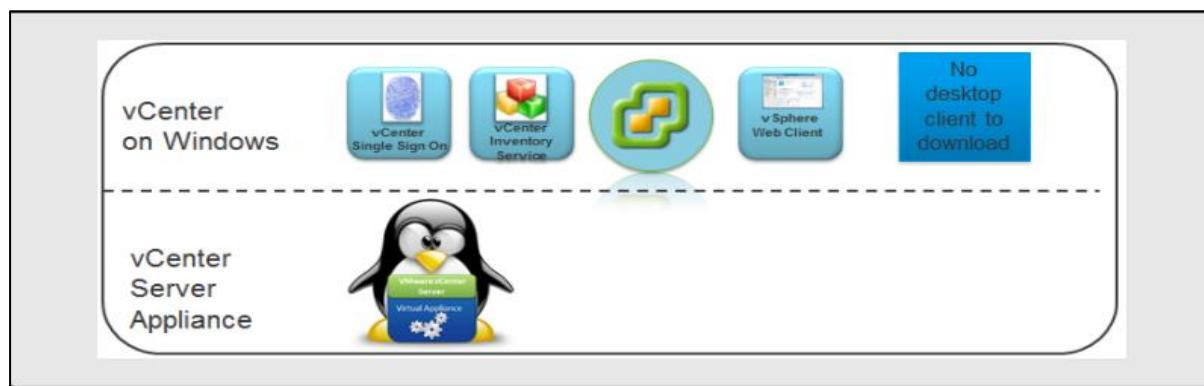
vCenter Server unifies resources from individual hosts so that those resources can be shared among virtual machines in the entire datacentre. It accomplishes this by managing the assignment of virtual machines to the hosts and the assignment of resources to the virtual machines within a given host based on the policies that the system administrator sets.



The above diagram shows how vCenter fits in the vSphere stack. With vCenter installed, you have a central point of management. vCenter Server allows the use of advanced vSphere features such as vSphere Distributed Resource Scheduler (DRS), vSphere High Availability (HA), vSphere vMotion, and vSphere Storage vMotion.

The other component is the vSphere Web Client. The vSphere Web Client is the interface to vCenter Server and multi-host environments. It also provides console access to virtual machines. The vSphere Web Client lets you perform all administrative tasks by using an in-browser interface.

- **VCENTER 7 COMPONENTS**

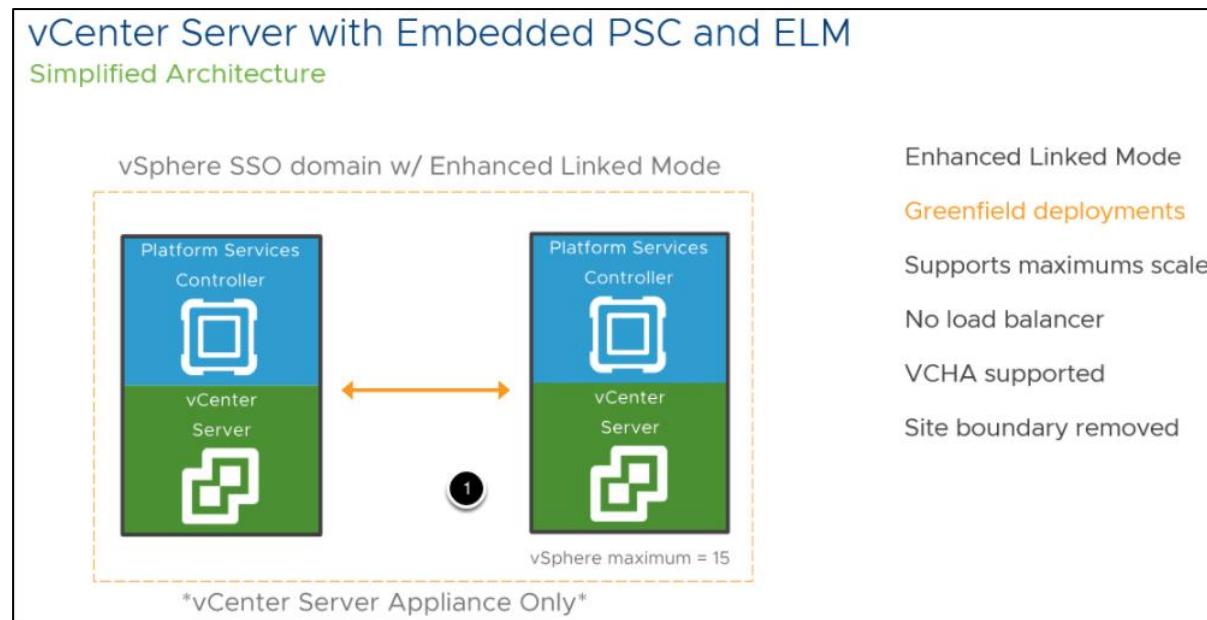


Starting with vSphere 5.1 there are two methods to deploy vCenter. The first method is a Windows installation. With the Windows method, you can install vCenter Single Sign-On, Inventory Service, and vCenter Server on the same host machine (as with vCenter Simple Install) or on different virtual machines.

The vCenter Server Appliance (vCSA) is a single preconfigured Linux-based virtual machine optimized for running vCenter Server and associated services.

# PRACTICAL NO - 5

- PLATFORM SERVICES CONTROLLER (PSC)



The Platform Services Controller (PSC) includes common services that are used across the suite. These include Single Sign-On (SSO), Licensing, and the VMware Certificate Authority (VMCA). You will learn more about SSO and the VMCA in the following pages.

The PSC is the first piece that is either installed or upgraded. When upgrading an SSO instance becomes a PSC. There are two models of deployment, embedded and centralized.

Embedded means the PSC and vCenter Server are installed on a single virtual machine. – Embedded is recommended for sites with a single SSO solution such as a single vCenter.

Centralized means the PSC and vCenter Server are installed on different virtual machines. – Centralized is recommended for sites with two or more SSO solutions such as multiple vCenter Servers, vRealize Automation, etc. When deploying in the centralized model it is recommended to make the PSC highly available as to not have a single point of failure, in addition to utilizing vSphere HA a load balancer can be placed in front of two or more PSC's to create a highly available PSC architecture.

The PSC and vCenter servers can be mixed and matched, meaning you can deploy Appliance PSC's along with Windows PSC's with Windows and appliance-based vCenter Servers. Any combination uses the PSC's built-in replication.

## Use Case:

- The PSC removes services from vCenter and makes them centralized across the vCloud Suite.
- This gives customers a single point to manage all their vSphere roles and permissions along with licensing.
- Reducing vCenter Server installation complexity allows customers to install or upgrade to vSphere 7 faster.
- There are only two installs' options:

# PRACTICAL NO - 5

- Embedded PSC which installs all components on a single virtual machine.
- Centralized, the customer must install the PSC and vCenter Server separately.
- In either installation model, all vCenter Server services are installed on the vCenter Server reducing the complexity of planning and installing vCenter Server.

- **VCENTER SINGLE SIGN ON**

vSphere 5.1 introduced vCenter Single Sign On (SSO) as part of the vCenter Server management infrastructure. This change affects the vCenter Server installation, upgrading, and operation. Authentication by vCenter Single Sign On makes the VMware cloud infrastructure platform more secure by allowing the vSphere software components to communicate with each other through a secure token exchange mechanism, instead of requiring each component to authenticate a user separately with a directory service like Active Directory.

- **VCENTER SINGLE SIGN ON - TYPICAL DEPLOYMENT**



Starting with version 5.1, vSphere includes a vCenter Single Sign-On service as part of the vCenter Server management infrastructure.

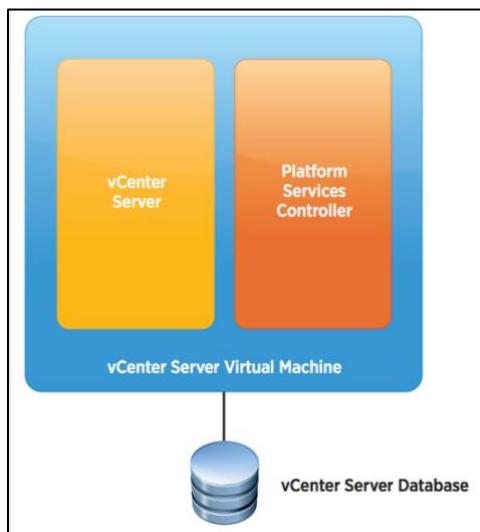
Authentication with vCenter Single Sign-On makes vSphere more secure because the vSphere software components communicate with each other by using a secure token exchange mechanism, and all other users also authenticate with vCenter Single Sign-On.

Starting with vSphere 6.0, vCenter Single Sign-On is either included in an embedded deployment or part of the Platform Services Controller. The Platform Services Controller contains all of the services that are necessary for the communication between vSphere components including vCenter Single Sign-On, VMware Certificate

## PRACTICAL NO - 5

Authority, VMware Lookup Service, and the licensing service. For example, in the image above, SSO resides within the Platform Services Controller as part of this multi-vCenter topology.

- **VCENTER SINGLE SIGN ON - SINGLE VCENTER**



In a single vCenter topology, the PSC (along with all of its associated services) can run on a single machine, also called the embedded deployment. This single machine could be a physical Windows server, a Windows VM, or the vCSA.

While vCenter Server requires a database, as shown above, SSO itself does not have such a requirement.

### VCENTER SERVER AND CREATING A VIRTUAL MACHINE

The previous lesson reviewed the ESXi Host Client, which can be used to manage one ESXi host at a time. This lesson will introduce the vSphere Client which is used to connect to vCenter Server to manage your collective infrastructure. In addition, the process of creating a virtual machine will also be covered.

The vSphere Client is the primary method for system administrators and end-users to interact with the virtual data center environment created by VMware vSphere. vSphere manages a collection of objects that make up the virtual data center, including hosts, clusters, virtual machines, data storage, and networking resources.

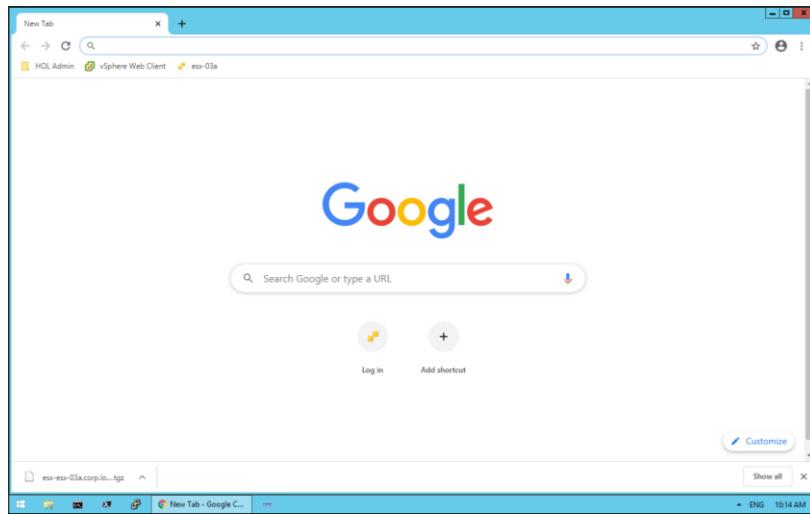
The vSphere Client is a Web browser-based application that you can use to manage, monitor, and administer the objects that make up your virtualized data center. You can use the vSphere Client to observe and modify the vSphere environment in the following ways.

- Viewing health, status, and performance information on vSphere objects
- Issuing management and administration commands to vSphere objects
- Creating, configuring, provisioning, or deleting vSphere objects

## PRACTICAL NO - 5

You can extend vSphere in different ways to create a solution for your unique IT infrastructure. You can extend the vSphere Client with additional GUI features to support these new capabilities, with which you can manage and monitor your unique vSphere environment.

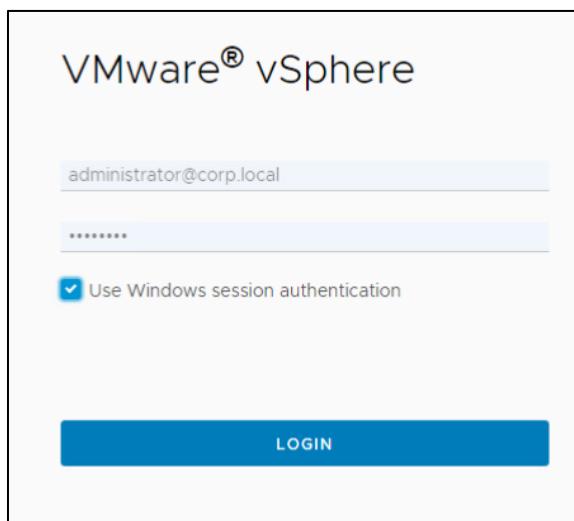
If you are not already in Chrome, double click on **Google Chrome** on your desktop. If you are already in Google Chrome, open a new tab.



Click the **vSphere Web Client** bookmark.

Log in using the following method:

- Click the "**Use Windows session authentication**" check box.
- Click the "**Login**" button.



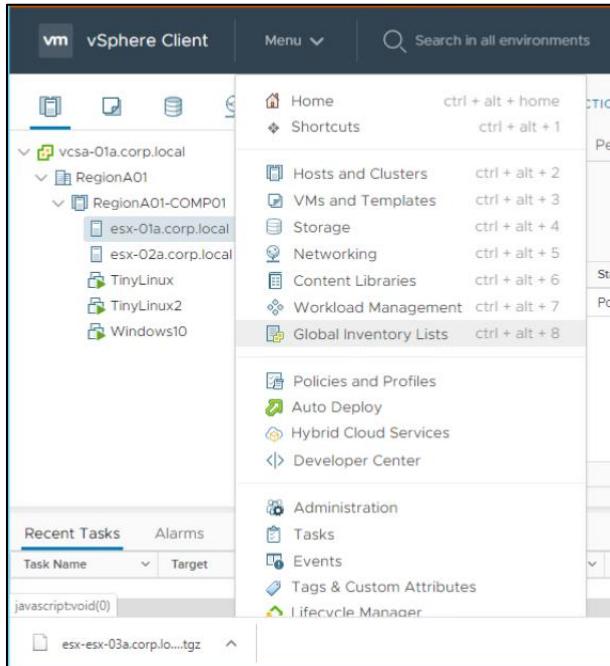
By default, you are brought to a view that shows the Hosts and Clusters attached to vCenter. Get a more complete look by viewing the Global Inventory Lists.

# PRACTICAL NO - 5

- **VCENTER INVENTORY**

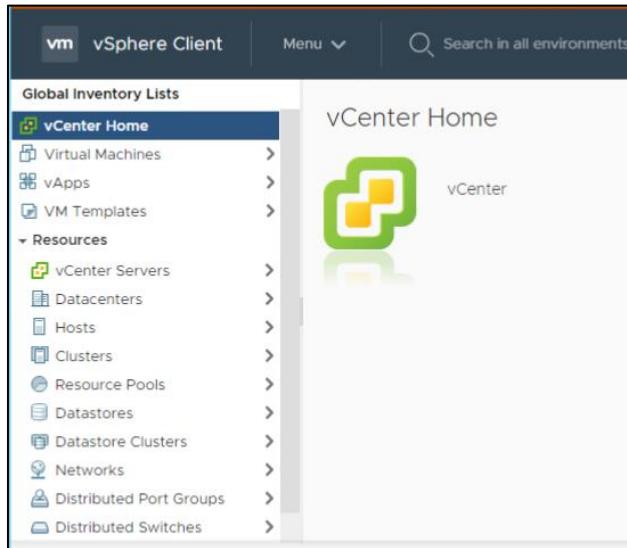
Click on the Menu drop-down list and select Global Inventory Lists.

Clicking Global Inventory Lists will take you to the inventory page where you find all the objects associated with vCenter Server systems such as data centers, hosts, clusters, networking, storage, and virtual machines.



- **CHILD OBJECTS, DATA CENTERS, AND HOSTS**

Click the "Virtual Machines" inventory item. By selecting this inventory item, you are presented with a list of the VMs which are located in this environment.



- **VIRTUAL MACHINE SUMMARY**

Here are all the virtual machines associated with this vCenter instance.

# PRACTICAL NO - 5

The screenshot shows two windows of the vSphere Client.

**Top Window:** A list of virtual machines. The table shows the following data:

Name	State	Status	Provisioned Space	Used Space	Host CPU	Host Mem
TinyLinux	Powered On	Normal	436.84 MB	371.84 MB	0 Hz	176 MB
TinyLinux2	Powered On	Normal	436.83 MB	371.83 MB	0 Hz	177 MB
Windows10	Powered On	Normal	27.08 GB	18.15 GB	41 MHz	2.03 GB

**Bottom Window:** A detailed view of the Windows10 VM. The Summary tab is selected. Key details shown include:

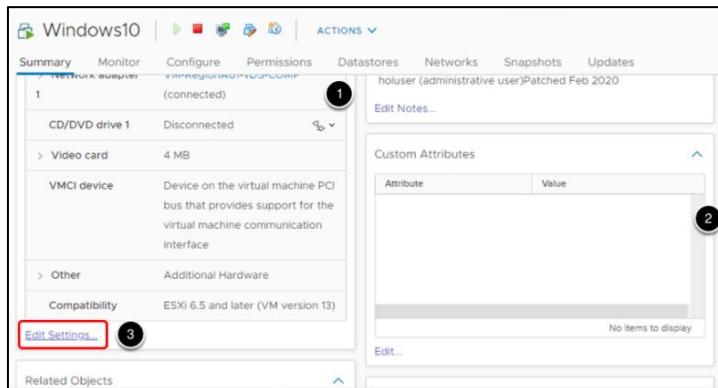
- Guest OS: Microsoft Windows 10 (64-bit)
- Compatibility: ESXi 6.5 and later (VM version 13)
- VMware Tools: Running, version:11297 (Current)
- DNS Name: Windows10.corp.local
- IP Addresses: 192.168.120.53
- Host: esx-02a.corp.local

On the right, resource usage statistics are displayed:

- CPU USAGE: 41 MHz
- MEMORY USAGE: 204 MB
- STORAGE USAGE: 18.15 GB

- Click the "Windows10" virtual machine.
- Click the "Summary" Tab for that virtual machine. On this page, you can see all the details regarding the virtual machine. There is an "Edit Settings" link as well to modify the settings of the virtual machine.
- Expand the VM Hardware section.
- EDIT THE SETTINGS OF A VIRTUAL MACHINE.**

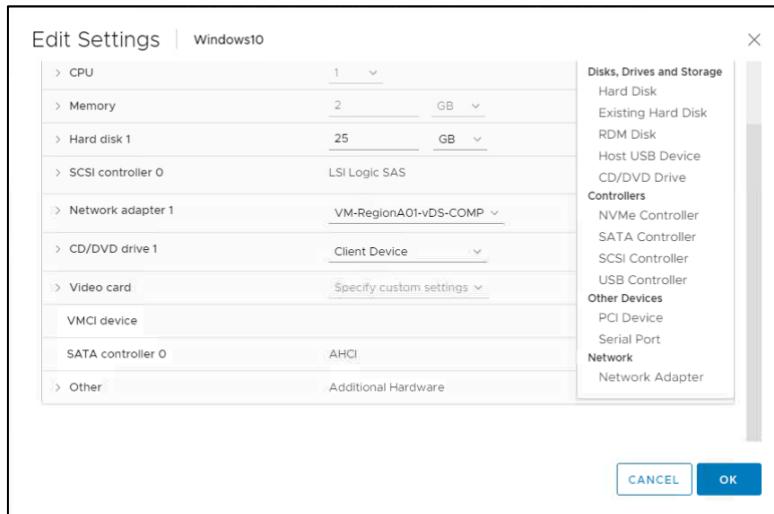
  - Review the VM Hardware for the windows10 virtual machine. Note that there is currently only one network adapter.
  - Use the scroll bar to move to the bottom of the VM Hardware section.
  - Click "Edit Settings" so a second network adapter can be added to the virtual machine.



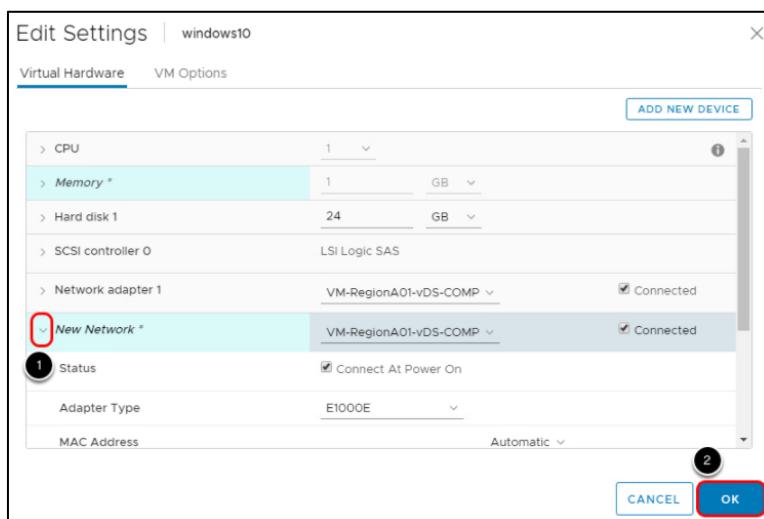
# PRACTICAL NO - 5

- ADD A SECOND NETWORK ADAPTER

1. In the Edit Setting window, click the Add New Device button.
2. Select Network Adapter from the drop-down list.



3. Click the arrow next to the New Network card to expand and view its settings. Notice that the MAC address is blank at this point. A new MAC address will be generated once this NIC is added, or we are able to specify (with some rules) our own MAC address.
4. Click "OK" to add the device to the VM. When you select "OK" a new task is created.



- RECENT TASKS LIST

Click on on the Arrows to view the **Recent Tasks** to watch the task's progress.



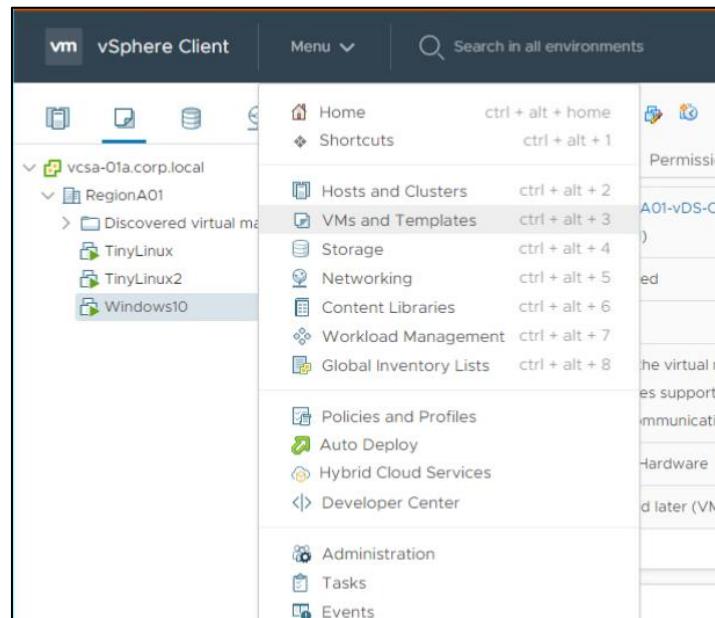
Review the "Recent Tasks" list. Once the task is complete, a second Network Adapter should be shown in the "VM Hardware" section. Note the networks are in a disconnected state because the VM is powered off. Once you are done viewing the Recent Tasks list, click the down-arrows to minimize it.

# PRACTICAL NO - 5

- **CREATE A VIRTUAL MACHINE**

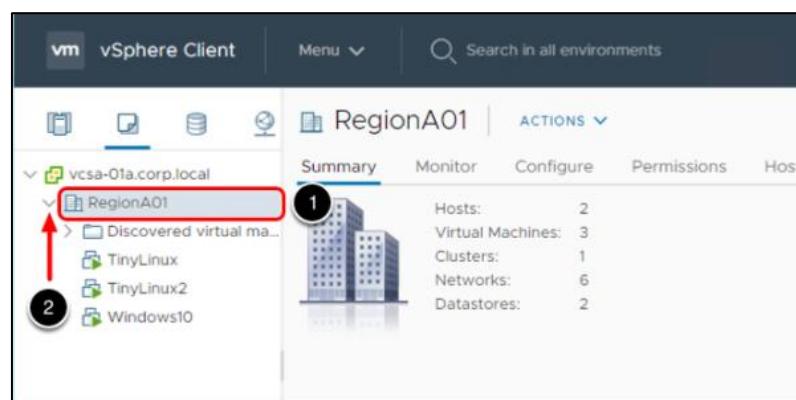
In the next steps, we will create a virtual machine and then, install an operating system.

1. To return to the VMs and Templates view, click on **Menu**.
2. Select **VMs and Templates**.



- **SELECT AND EXPAND DATACENTER**

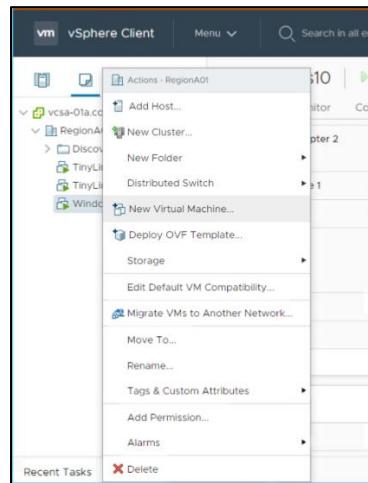
1. Click on RegionA01 Datacenter.
2. Expand RegionA01 Datacenter so the virtual machines under it can be seen.



# PRACTICAL NO - 5

- **START THE NEW VIRTUAL MACHINE WIZARD**

1. Right-click on RegionA01 Datacenter.
2. Click New Virtual Machine to start the new virtual machine wizard.



This wizard is used to create a new Virtual Machine and place it in the vSphere inventory.

- **VIRTUAL MACHINE WIZARD**

The image contains two side-by-side screenshots of the 'New Virtual Machine' wizard in the vSphere Client.  
Left Screenshot: The title bar says 'New Virtual Machine'. On the left, a vertical list of steps is shown: 1 Select a creation type (highlighted), 2 Select a name and folder, 3 Select a compute resource, 4 Select storage, 5 Select compatibility, 6 Select a guest OS, 7 Customize hardware, 8 Ready to complete. The main pane shows a 'Create a new virtual machine' section with a description: 'This option guides you through creating a new virtual machine. You will be able to customize processors, memory, network connections, and storage. You will need to install a guest operating system after creation.' At the bottom are 'CANCEL', 'BACK', and a red-bordered 'NEXT' button.  
Right Screenshot: The title bar says 'New Virtual Machine'. It shows the same step list. The main pane has sections for 'Select a name and folder' (with 'Virtual machine name: web-serv01') and 'Select a location for the virtual machine' (with 'RegionA01' selected). At the bottom are 'CANCEL', 'BACK', and a red-bordered 'NEXT' button.  
Callouts: A circled number '1' points to the 'NEXT' button in the left screenshot. A circled number '2' points to the 'NEXT' button in the right screenshot.

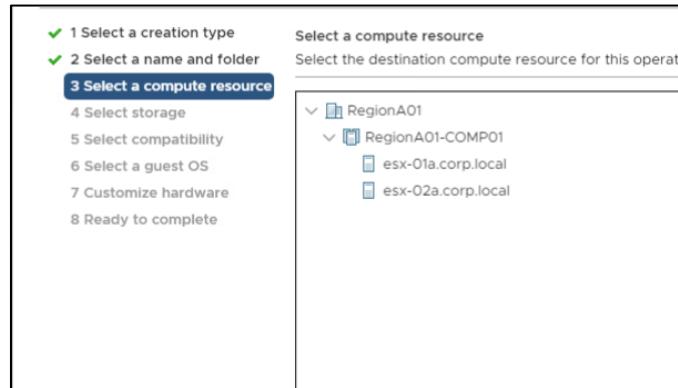
1. Since the Create a new virtual machine wizard is highlighted, just click Next.
2. Enter web-serv01 for the name of the new virtual machine.
3. Click Next.

# PRACTICAL NO - 5

- **VIRTUAL MACHINE PLACEMENT**

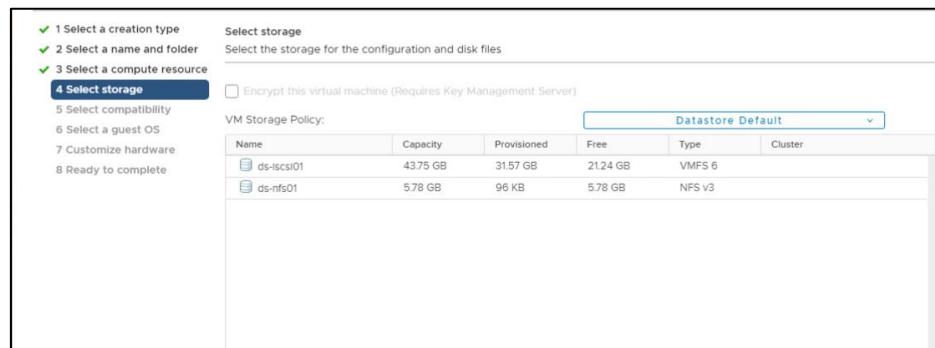
Because Distributed Resource Scheduler (DRS) is not enabled, you just have to select a host to use for the VM. More details on DRS will be covered later in this module.

1. Click esx-01a.corp.local.
2. Click Next.



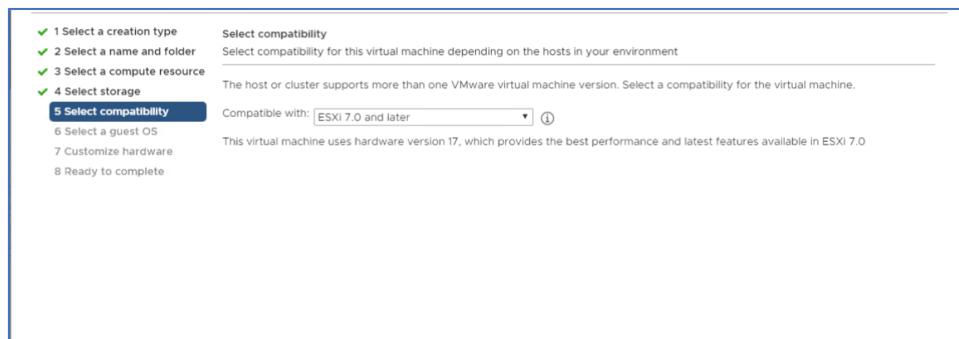
- **SELECT STORAGE**

1. Ensure the ds-iscsi01 datastore is selected.
2. Click Next.



- **COMPATIBILITY**

1. Select ESXi 7.0 and later.
2. Click Next to accept.



# PRACTICAL NO - 5

In this step, we will be selecting what operating system we will be installing. When we select the operating system, the supported virtual hardware and recommended configuration is used to create the virtual machine. Keep in mind this does not create a virtual machine with the operating system installed, but rather creates a virtual machine that is tuned appropriately for the operating system you have selected.

1. For the Guest OS Family, select Linux from the drop-down menu.
2. For the Guest OS Version, select VMware Photon OS (64-bit).
3. Click Next to continue.

The screenshot shows the 'Select a guest OS' step of a VM creation wizard. On the left, a sidebar lists steps 1 through 6, with '6 Select a guest OS' highlighted in blue. Step 7, 'Customize hardware', is also visible. On the right, the 'Select a guest OS' panel has a sub-header 'Choose the guest OS that will be installed on the virtual machine'. It includes a note about identifying the guest OS for appropriate defaults. The 'Guest OS Family' dropdown is set to 'Linux', and the 'Guest OS Version' dropdown is set to 'VMware Photon OS (64-bit)'. A note at the bottom says 'Ready to complete'.

- **CHANGE VIRTUAL DISK SIZE.**

The recommended virtual hardware settings are shown as the default. These can be modified if needed.

1. Leave the default settings and click Next.

The screenshot shows the 'Customize hardware' step of the VM creation wizard. On the left, a sidebar lists steps 1 through 7, with '7 Customize hardware' highlighted in blue. Step 8, 'Ready to complete', is also visible. The main area is titled 'Virtual Hardware' and shows configuration options for CPU (1), Memory (2 GB), New Hard disk (16 GB), New SCSI controller (VMware Paravirtual), New Network (VM Network), New CD/DVD Drive (Client Device), Video card (Specify custom settings), Security Devices (Not Configured), VMCI device, New SATA Controller (New SATA Controller), and Other (Additional Hardware). An 'ADD NEW DEVICE' button is at the top right. A note at the bottom says 'Compatibility: ESXi 7.0 and later (VM version 17)'.

The settings for the virtual machine can be verified prior to it being created.

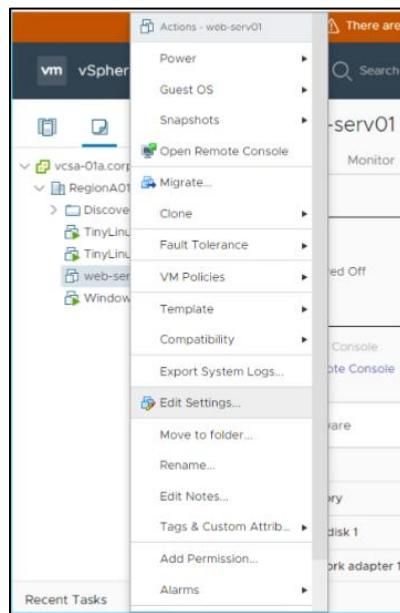
1. Click Finish to create the virtual machine.
2. Congratulations on creating your first virtual machine web-serv01!
3. In the next steps, Photon OS will be installed on the virtual machine.

# PRACTICAL NO - 5

- ATTACHING AN ISO TO A VIRTUAL MACHINE

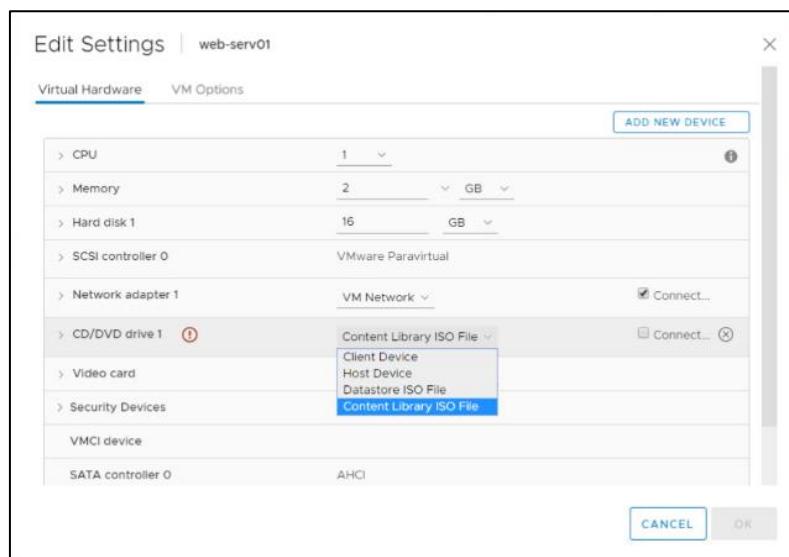
To make it easier to install operating systems on virtual machines, ISO images can be used. These can be kept in the same storage used for virtual machines. In addition, vCenter offers a Content Library as a repository. Content Libraries can then be synchronized to ensure every location is using the same versions.

1. To attach an ISO image to the virtual machine we just created, make sure web-serv01 is selected.
2. Right-click on web-serv01 and select Edit Settings...



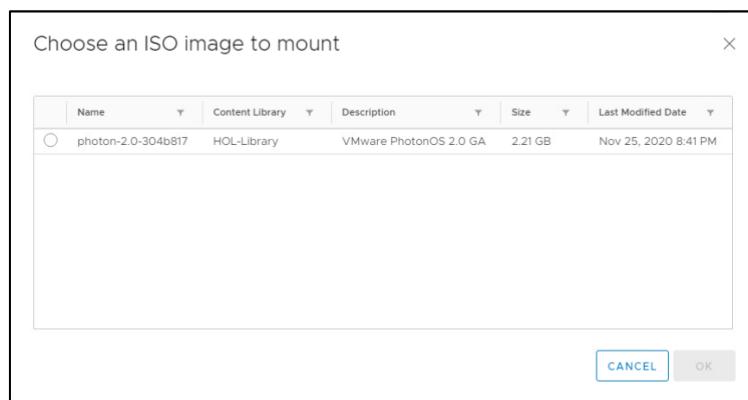
- CONTENT LIBRARY ISO FILE

From the CD/DVD drive 1 drop-down menu, select Content Library ISO File. This will open a file explorer to select that file.



# PRACTICAL NO - 5

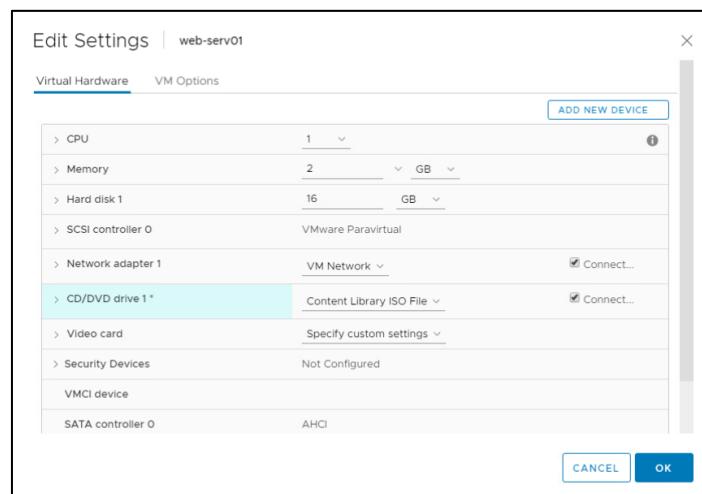
- **SELECT PHOTON**



1. Click the radio button next to photon-2.0-304b817.
2. Click OK.

- **CONNECT THE DRIVE**

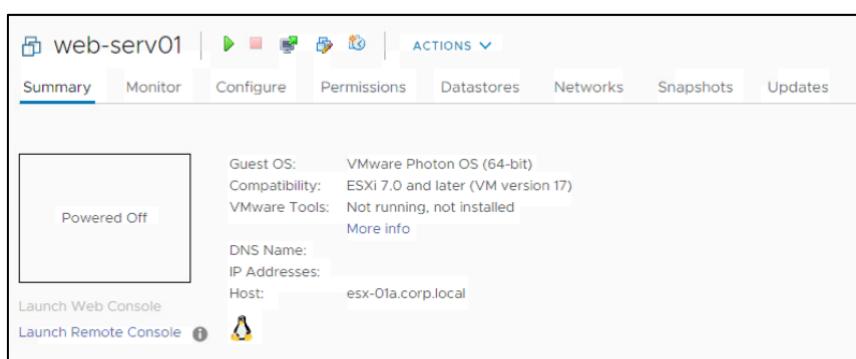
Finally, we want to attach or connect the ISO image to the virtual machine.



1. Click the Connected check box next to CD/DVD drive 1.
2. Click OK.

- **POWER ON WEB-SERV01**

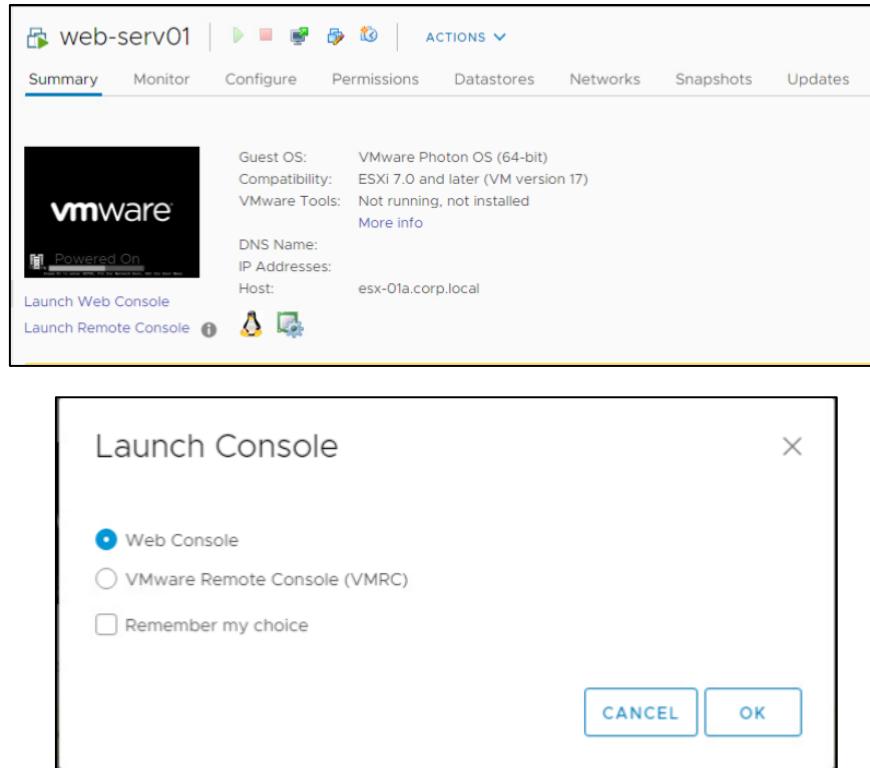
Click the green play button to power on the virtual machine and start the installation.



# PRACTICAL NO - 5

- **LAUNCH CONSOLE**

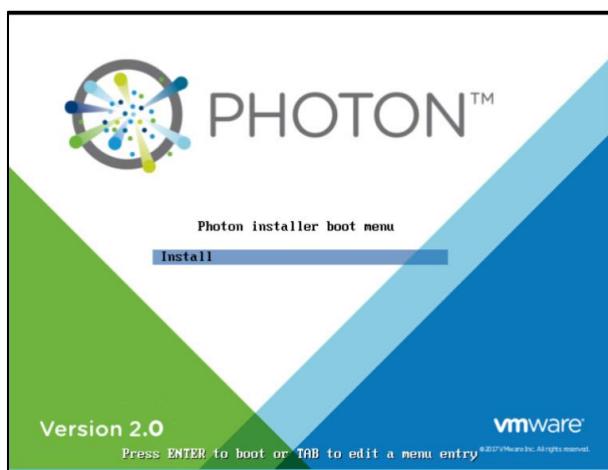
To launch the console window, click anywhere in the console window screen.



1. Select the Web Console.
2. Click OK.

Note you also have the option of using the VMware Remote Console (VMRC). This is a separate application that needs to be installed on your local device as opposed to the Web Console which will launch in a new browser tab. The VMRC can be useful in certain situations when you need more capabilities, like attaching devices or power cycling options.

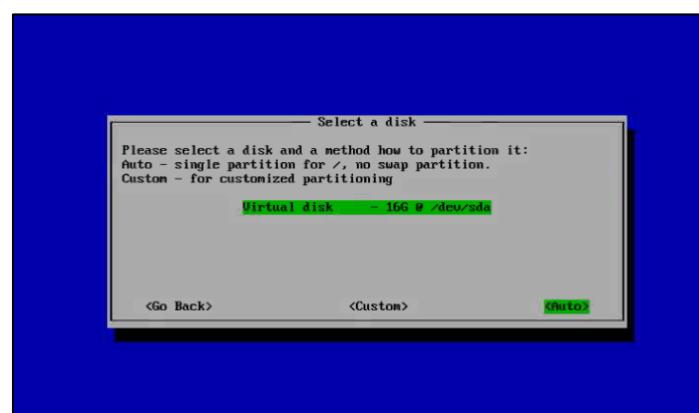
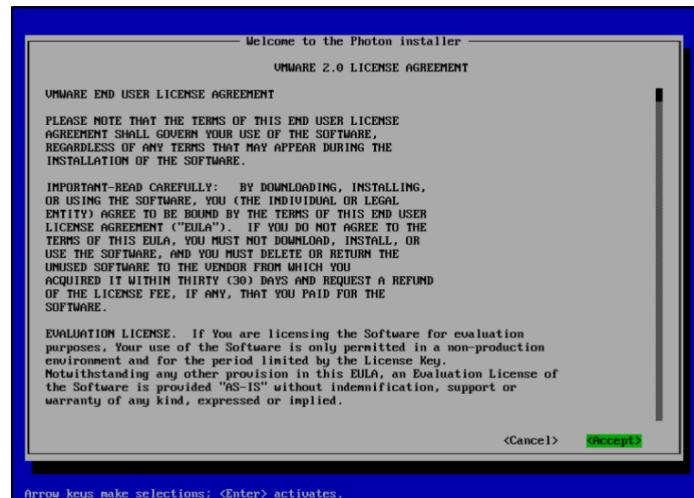
- **PHOTON BOOT SCREEN**



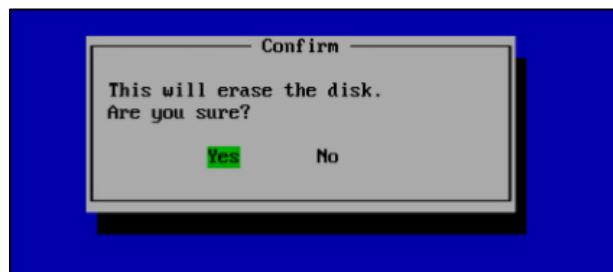
A new tab will open, and you will be presented with the Photon OS boot screen.

1. Press the Enter key to start the installation process.
2. After the boot process is complete, you will be presented with a license agreement.
3. Press Enter to accept.

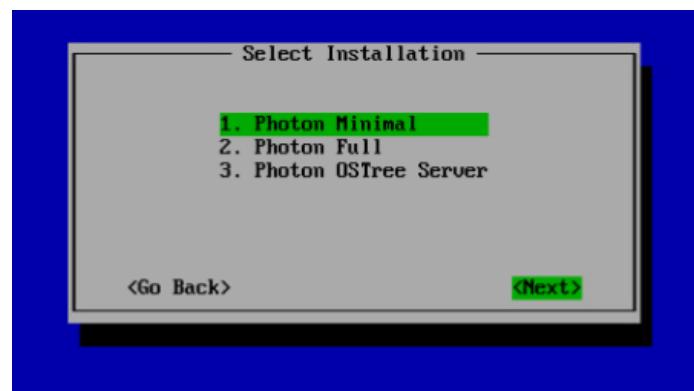
# PRACTICAL NO - 5



4. Press Enter to accept the selected disk and use the auto partitioning option.

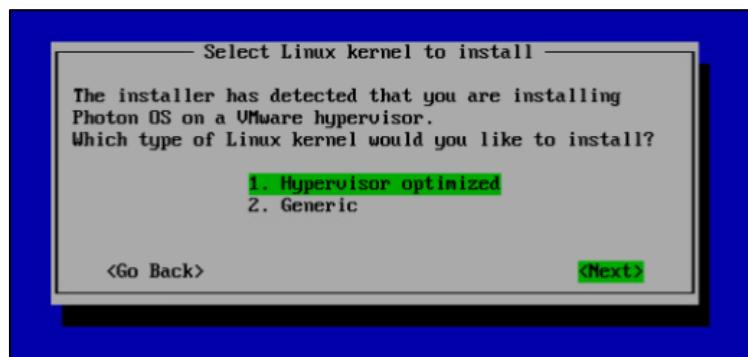


5. Press Enter to confirm the disk should be erased.

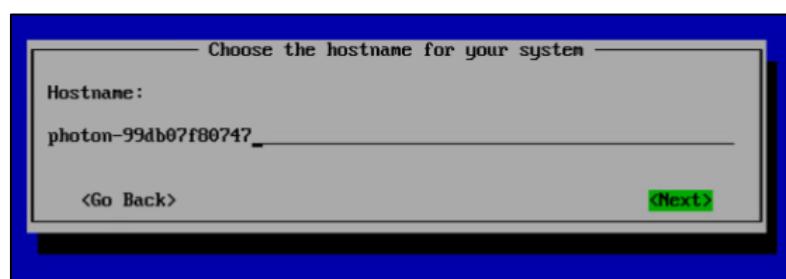


## PRACTICAL NO - 5

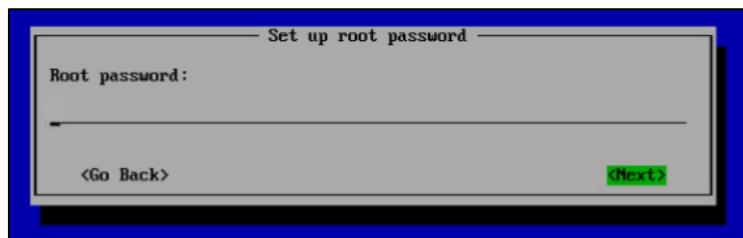
6. At the Select Installation screen, make sure the default option of 1. Photon Minimal is selected.
7. Press the Enter key.



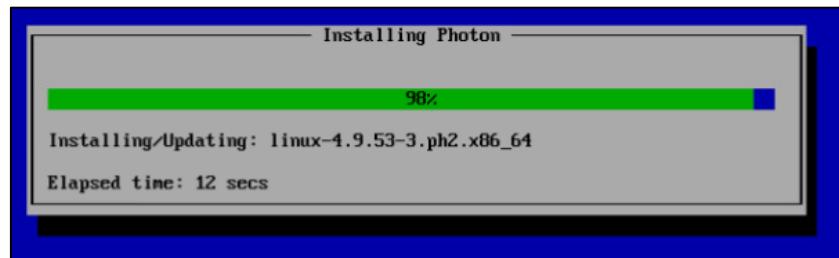
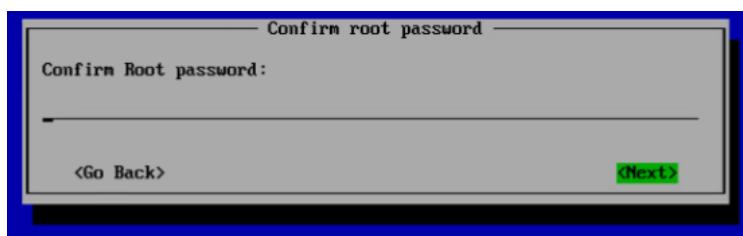
8. Use the arrow key to select 2. Generic.
9. Press the Enter key.



10. Use the Backspace key to remove the default hostname.
11. Type web-serv01.
12. Press the Enter key.

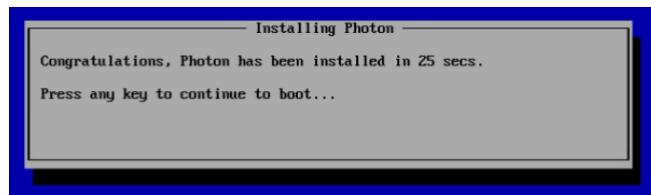


13. For the password, use VMware1!VMware1!
14. Type VMware1!VMware1! again to confirm the password.
15. Press the Enter key.



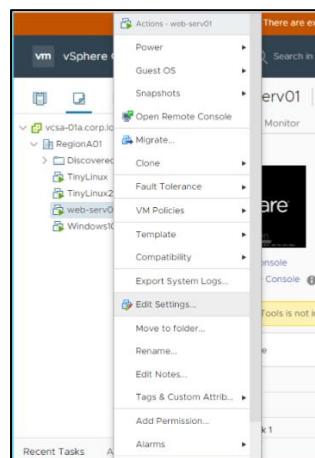
# PRACTICAL NO - 5

After a minute or two, the installation will be complete. Press a key to reboot the virtual machine. After a minute or two, the system should boot the login prompt.



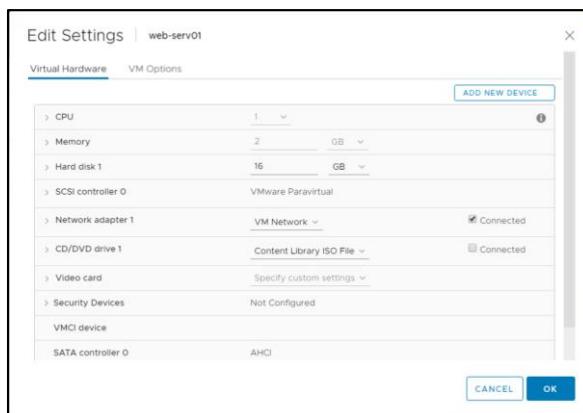
Now that the operating system has been installed and is up and running, the ISO image needs to be disconnected from the virtual machine.

Select the **vSphere- web-serv01** tab.

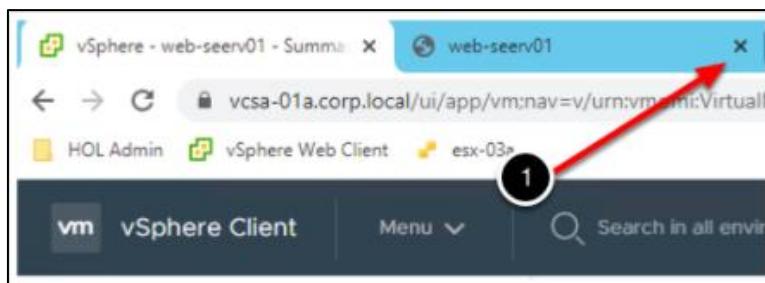


Make sure web-serv01 is still highlighted.

1. Right-click on web-serv01.
2. Select Edit Settings...
3. Uncheck the Connected box next to CD/DVD drive 1.



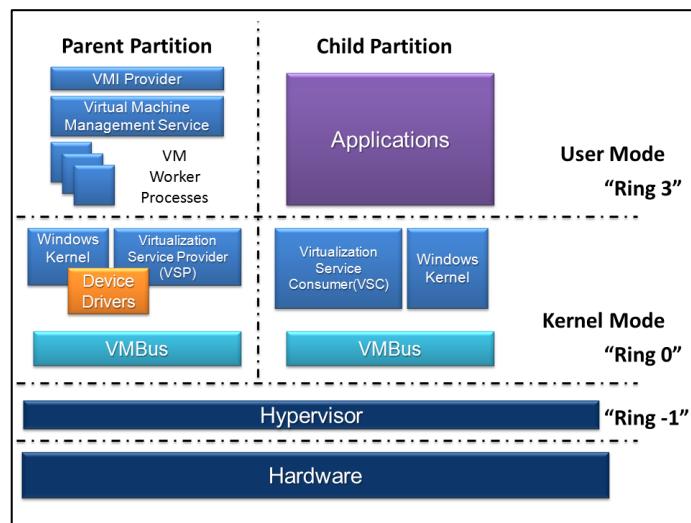
Click the 'X' to close the console window for web-serv01.



## PRACTICAL NO - 6

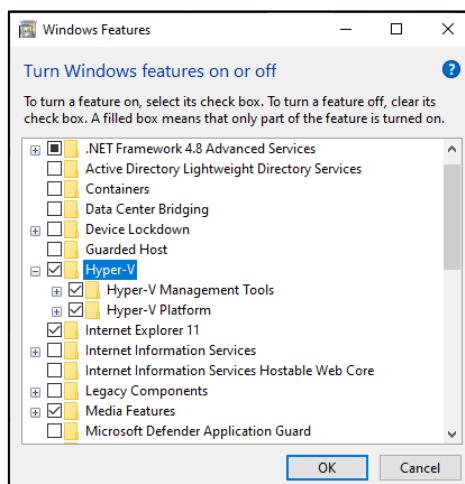
### ❖ Implement Virtualization Using Hyper-V:

- **Hyper-V** is Microsoft's hardware virtualization product.
- It lets you create and run a software version of a computer, called a *virtual machine*.
- Each virtual machine acts like a complete computer, running an operating system and programs.
- When you need computing resources, virtual machines give you more flexibility, help save time and money, and are a more efficient way to use hardware than just running one operating system on physical hardware.
- **Hyper-V** runs each virtual machine in its own isolated space, which means you can run more than one virtual machine on the same hardware at the same time.
- **Hyper-V** features a **Type 1** hypervisor-based architecture.
- The hypervisor virtualizes processors and memory and provides mechanisms for the virtualization stack in the root partition to manage child partitions (virtual machines) and expose services such as I/O devices to the virtual machines.



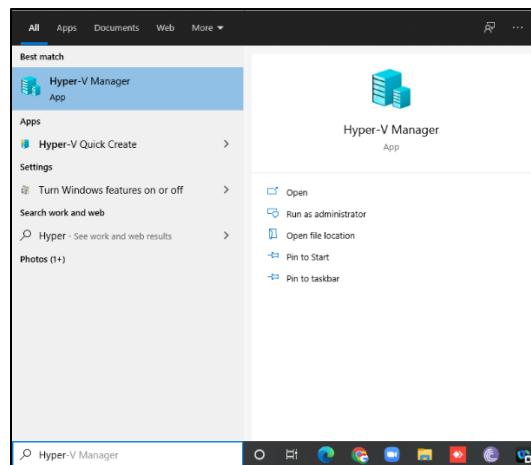
### ❖ How to Create and Run Virtual Machines with Hyper-V.

- First, we must uninstall VMware software if already installed on computer because the VMware Workstation installer does not support running on a Hyper-V virtual machine.
- After uninstalling VMware, we can proceed to next step: -
- Tap the Windows key, type “Windows features” to perform a search, and then click the “Turn Windows features on or off” shortcut. Check the Hyper-V checkbox in the list and click OK to install it. Restart your computer when prompted.

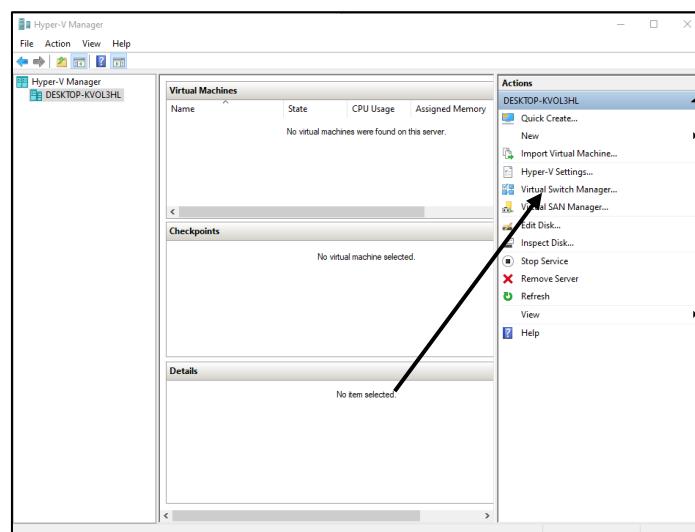


## PRACTICAL NO - 6

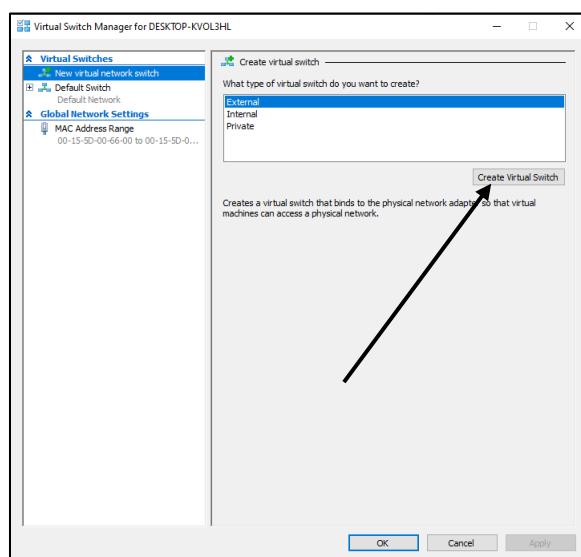
- After Restart Search for Hyper-V manager in search box and click on that.



- For creating virtual machine, first we must create virtual switch
- Click on **virtual switch manager** option

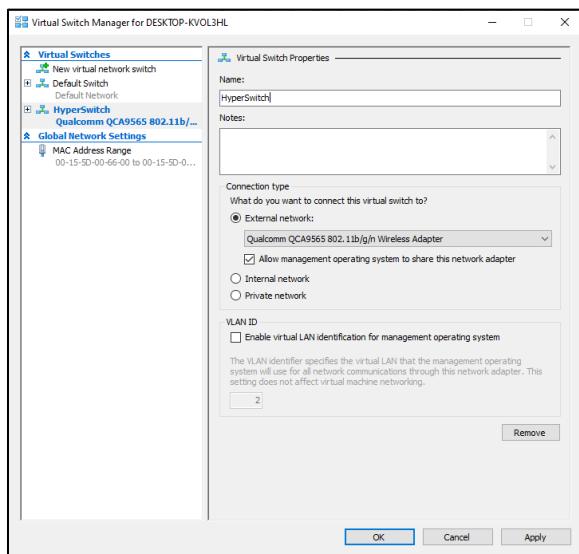


- Select External as a connection type and then click on **create virtual switch**

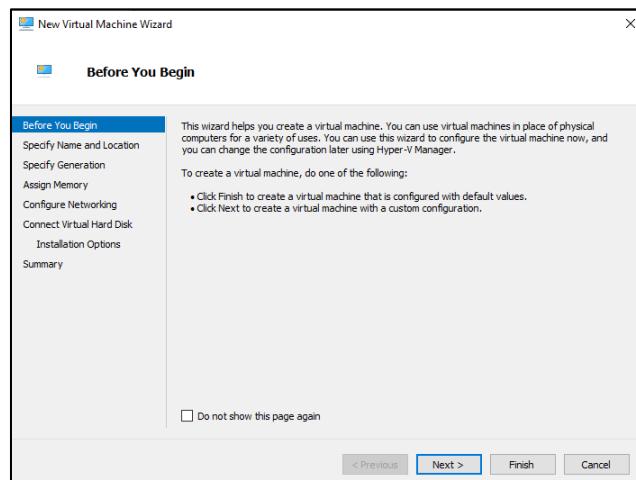
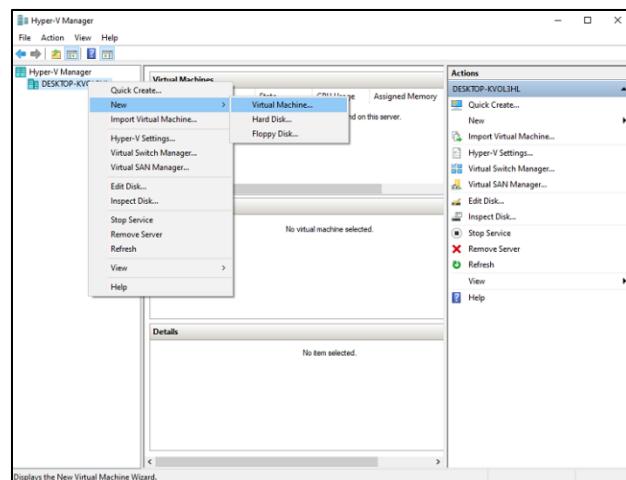


## PRACTICAL NO - 6

- Give name to your virtual switch then click on **apply** button.



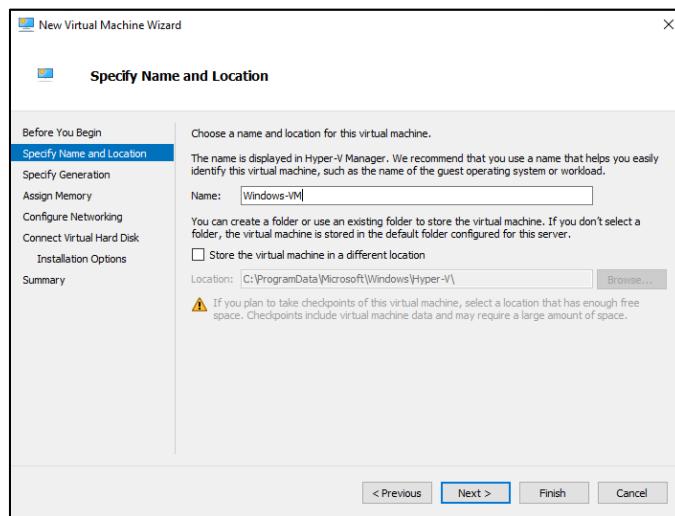
- Now right click on server and select new **virtual machine**.



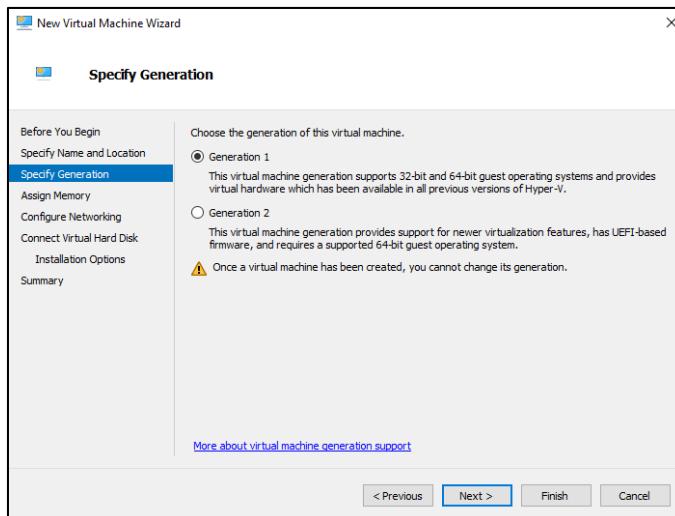
- Click on **Next**

## PRACTICAL NO - 6

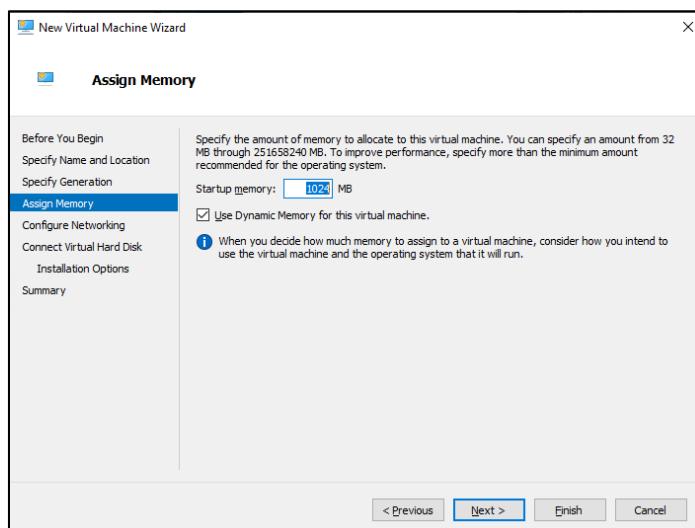
- Provide name to virtual machine then click on **Next** button.



- Specify generation: **Generation 1**

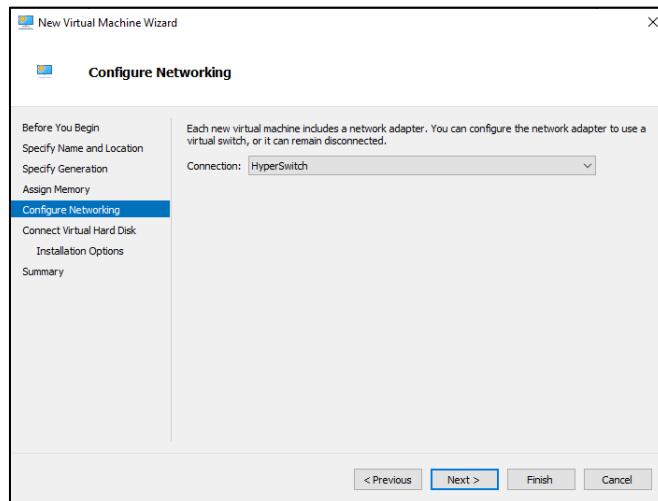


- Tick on use **Dynamic Memory** for this virtual machine.

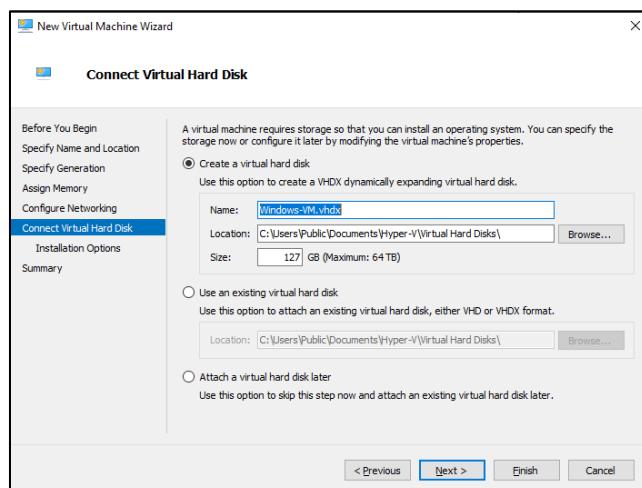


## PRACTICAL NO - 6

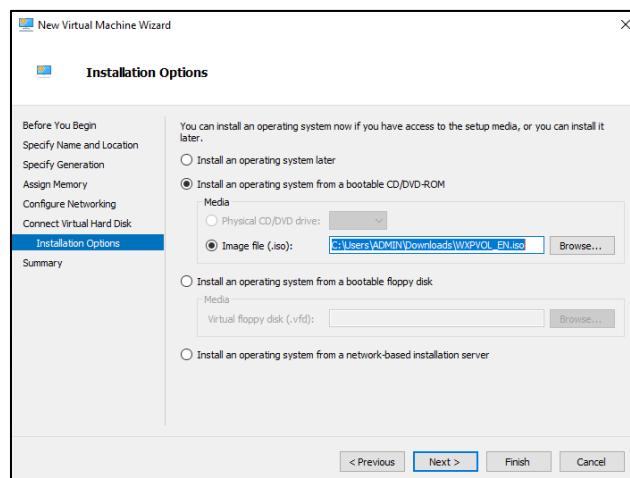
- Select switch which we created earlier for our virtual machine from drop-down list and then click on **next**



- Description of virtual machine and location where it will store virtual machine related files and size require for this machine click on **next**

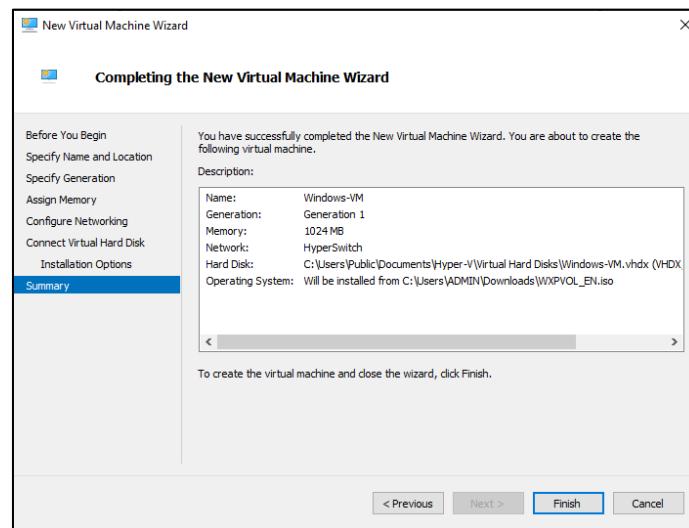


- New virtual machine wizard panel will appear, where we will choose operating system which we want to install on virtual machine.
- Select install an operating system from boot CD/DVD-ROM and then select Image file(.iso) and browse your OS iso file then click on **next** button.

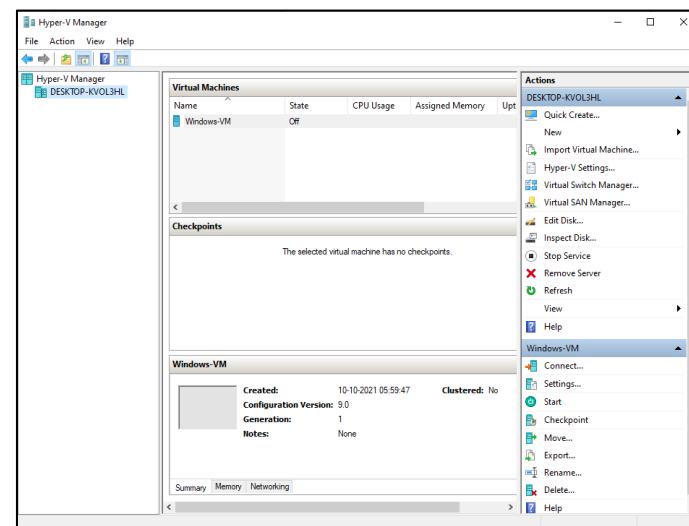


## PRACTICAL NO - 6

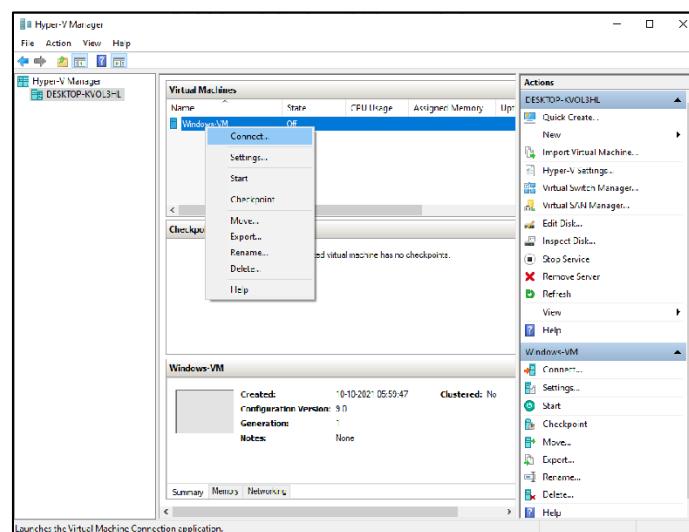
- Summary report will be generated about virtual machine then click on **Finish** button.



- In virtual machine panel your virtual machine will appear which has off state.

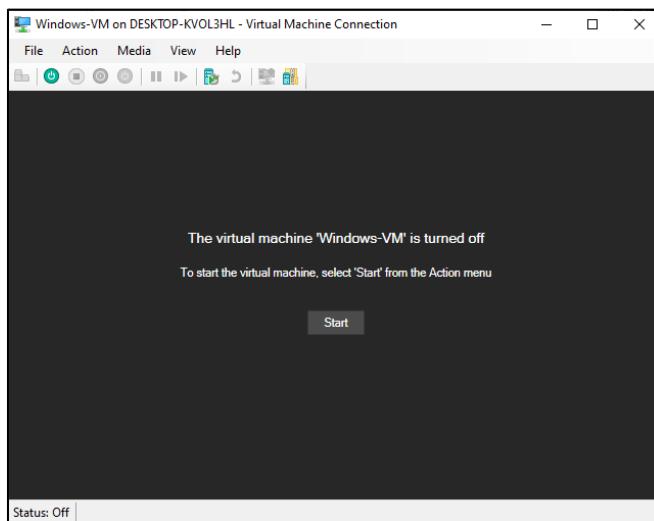


- Right click on virtual machine and click on **connect** option.

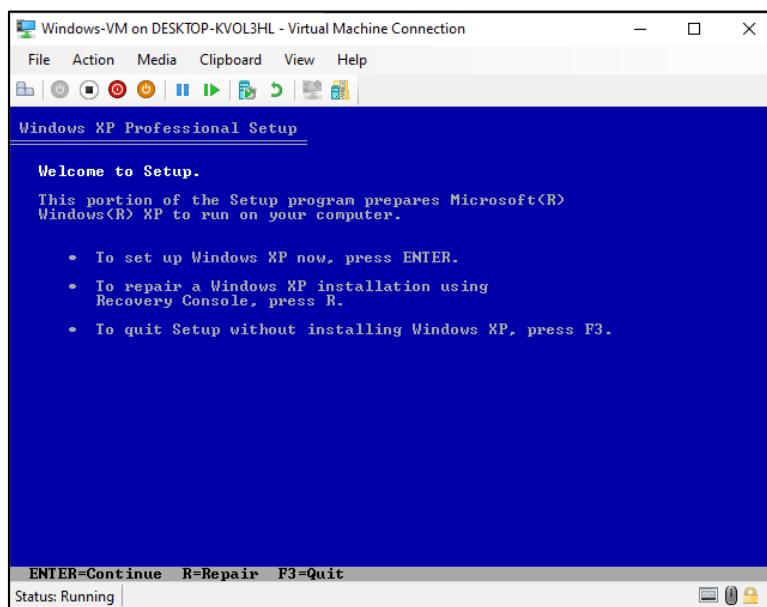


## PRACTICAL NO - 6

- Now turn on virtual machine.



- Virtual machine will start with below screen.



# PRACTICAL NO - 7

## ❖ Develop a Cloud Application for Microsoft Azure.

- Azure VMs are an on-demand scalable cloud-computing resource.
- They're like virtual machines that are hosted in Windows Hyper-V.
- They include processor, memory, storage, and networking resources.
- We can start and stop virtual machines at will, just like with Hyper-V, and manage them from the Azure portal or with the Azure CLI.
- We can also use a Remote Desktop Protocol (RDP) client to connect directly to the Windows desktop user interface (UI) and use the VM as if you were signed into a local Windows computer.

### Step 1: Activating Sandbox

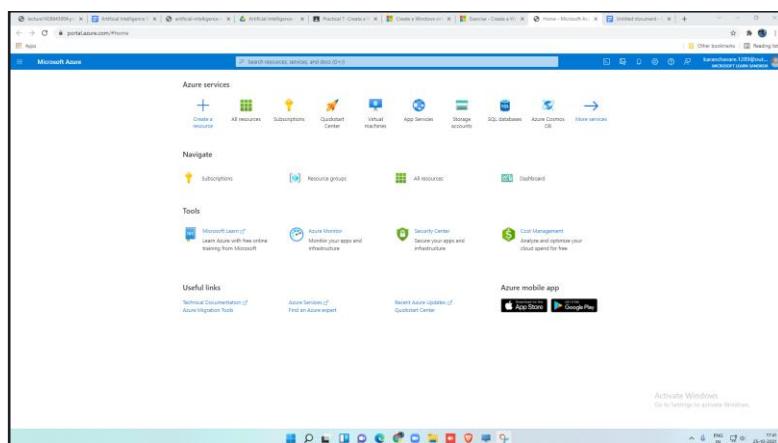
In this practical, we are going to use the built-in sandbox by logging in with a Microsoft account and activating the same for further process.



### Step 2: Creating a new Windows virtual machine

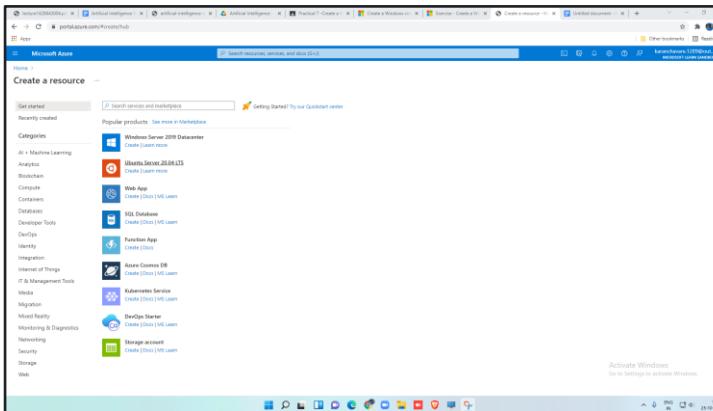
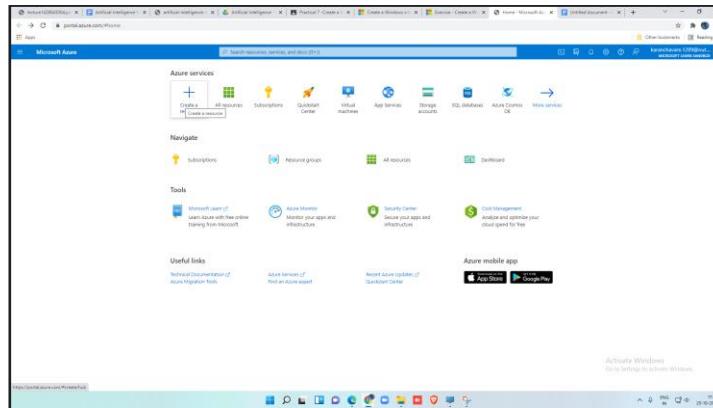
You can create Windows VMs with the Azure portal, Azure CLI, or Azure PowerShell. The best approach is to use the portal because the Create a virtual machine wizard collects all the required information and provides hints and validation messages throughout the process.

1. Sign in to the Azure portal using the same account you used to activate the sandbox.

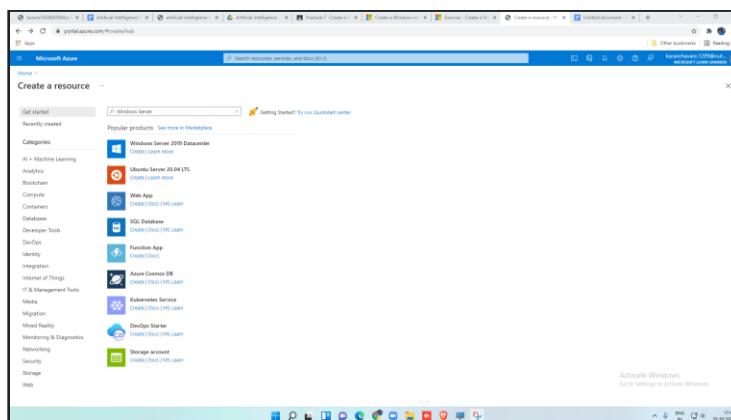


## PRACTICAL NO - 7

2. On the Azure portal, under Azure services, **select Create a resource**. The Create a resource pane appears.

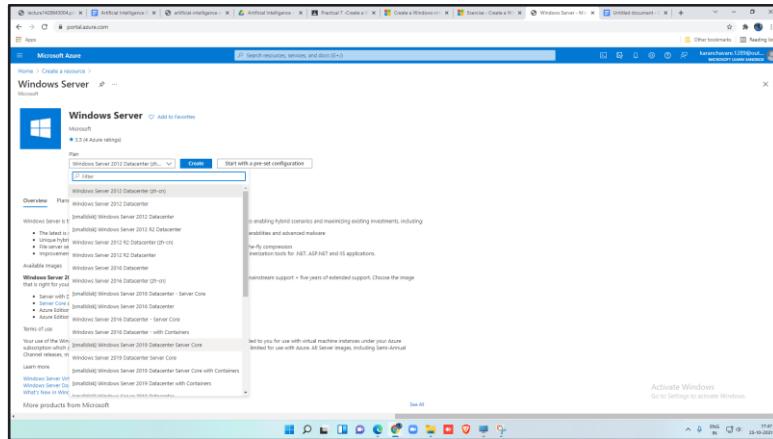


3. In the Search services and marketplace search box, **search for and select Windows Server**, and press **Enter**. The Windows Server pane appears.

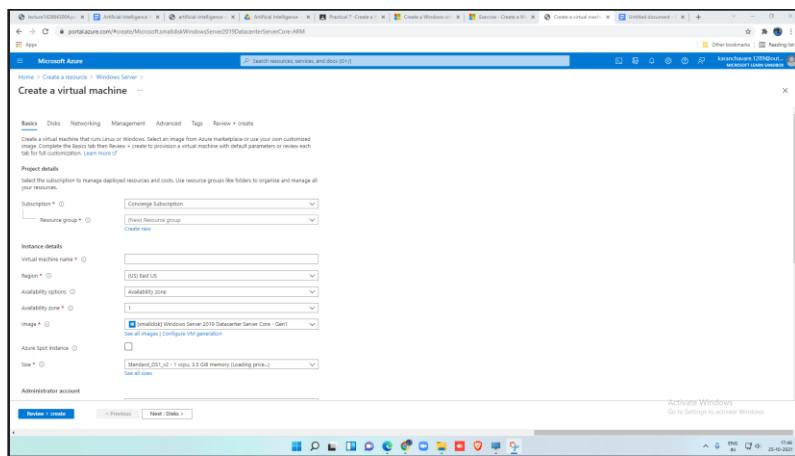


## PRACTICAL NO - 7

4. There are several Windows Server options to choose from to create your VM. In the Plan dropdown list, scroll down, and select [smalldisk] Windows Server 2019 Datacenter.



5. Select **Create**. The Create a virtual machine pane appears.



# PRACTICAL NO - 7

## Step 3: Configure the VM settings.

Azure presents a wizard as a series of tabs to walk you through all the configuration details for creating the VM. The first tab is Basics. You can select Next or Previous to move from one tab to another, or you can select any tab in the horizontal menu to move to a customizable configuration section.

Home > Create a resource > Windows Server >

### Create a virtual machine ...

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

**Project details**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ Concierge Subscription

Resource group \* ⓘ (New) Resource group Create new

**Instance details**

Virtual machine name \* ⓘ

Region \* ⓘ (US) East US

Availability options ⓘ Availability zone

Availability zone \* ⓘ 1

Image \* ⓘ [smalldisk] Windows Server 2019 Datacenter Server Core - Gen1 See all images | Configure VM generation

Azure Spot instance ⓘ

Size \* ⓘ Standard\_DS1\_v2 - 1 vcpu, 3.5 GiB memory (\$91.98/month) See all sizes

**Administrator account**

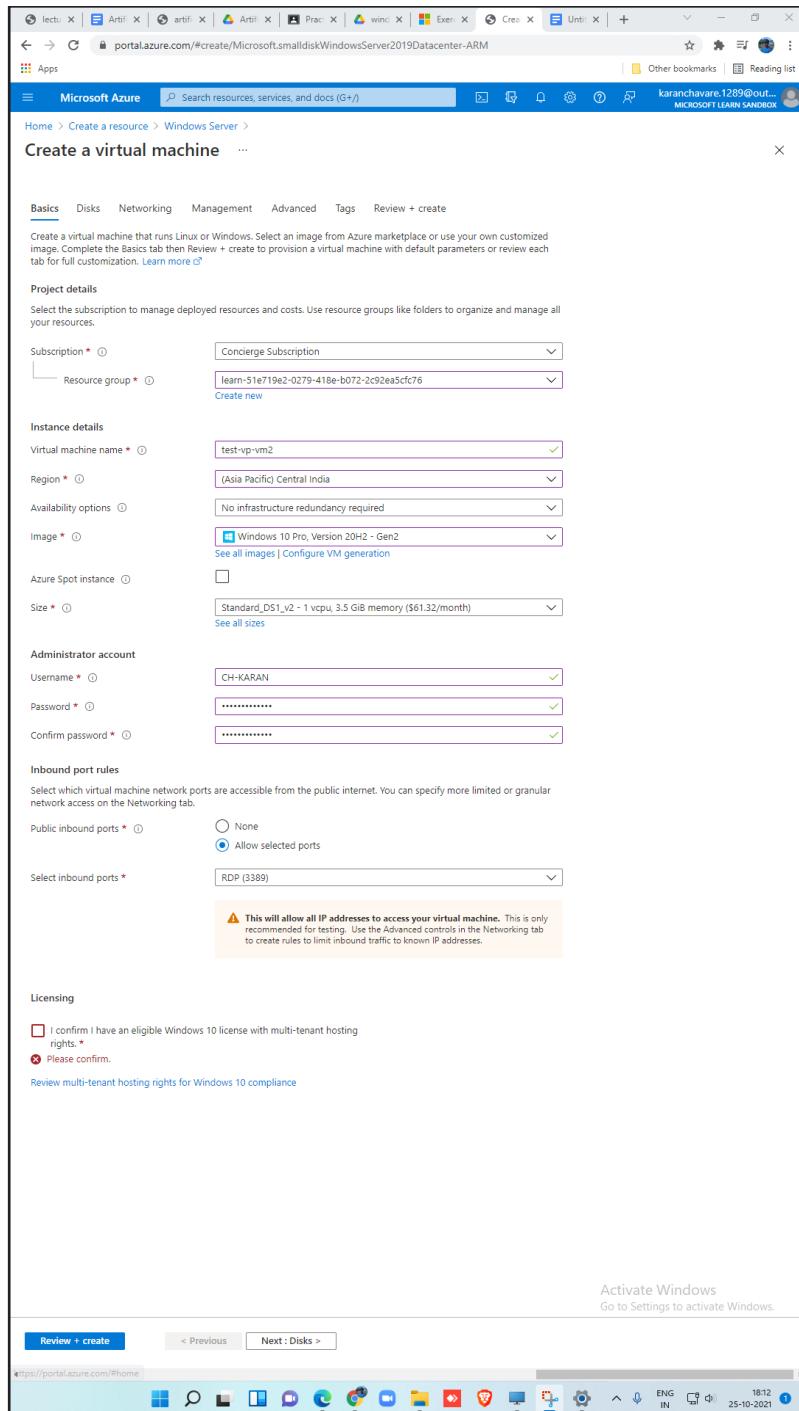
**Review + create** < Previous Next : Disks >

The screenshot shows the 'Create a virtual machine' wizard on the 'Basics' tab. It includes fields for Project details (Subscription: Concierge Subscription, Resource group: (New) Resource group), Instance details (Virtual machine name, Region: (US) East US, Availability options: Availability zone, Availability zone: 1, Image: Windows Server 2019 Datacenter Server Core - Gen1), and Administrator account (Size: Standard\_DS1\_v2). Navigation buttons at the bottom include 'Review + create', '< Previous', and 'Next : Disks >'.

- **Configure basic VM settings.**

1. On the Basics tab, enter the following values for each setting and select **Next**.

# PRACTICAL NO - 7



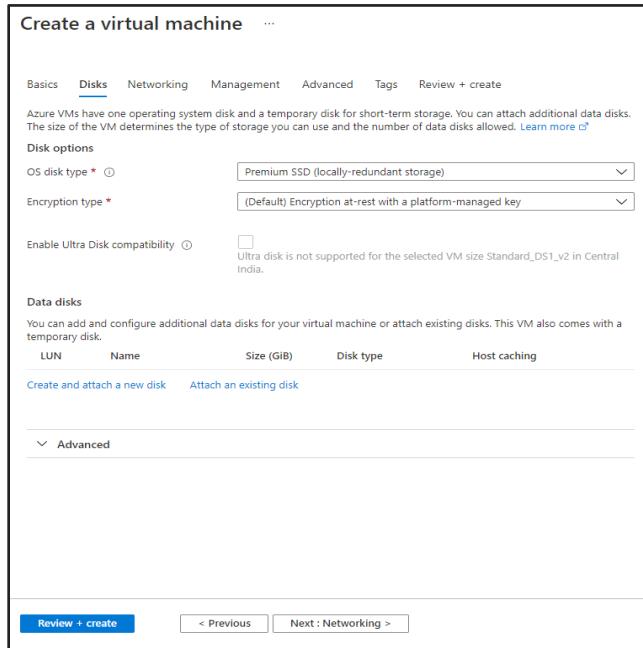
The free sandbox allows you to create resources in a subset of the Azure global regions. Select a region from the following list when you create resources:

- West US 2
- South Central US
- Central US
- East US
- West Europe
- Southeast Asia
- Japan East
- Brazil South
- Australia Southeast
- Central India

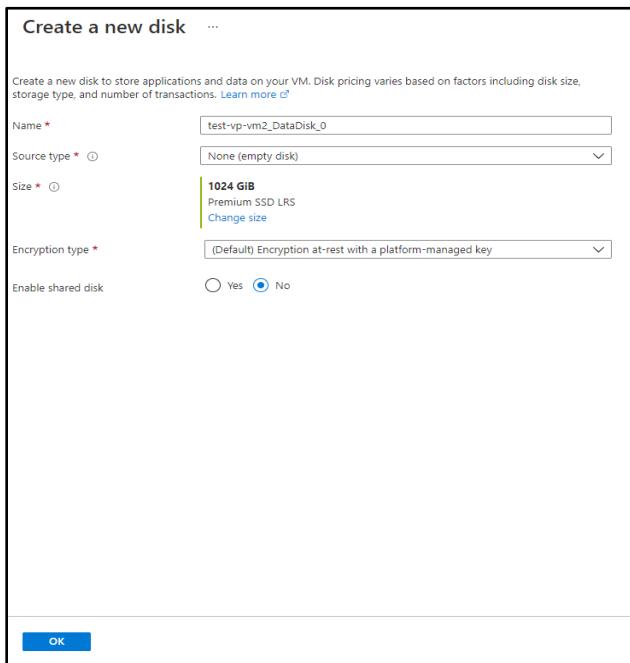
# PRACTICAL NO - 7

- Configure disks for the VM.

- On the Disks tab, enter or select the following values for each setting.



- Select Create and attach a new disk link. The Create a new disk pane appears.



- Select OK to save the settings and close the pane.

## PRACTICAL NO - 7

- On the Create a virtual machine pane Disks tab, under Data disks, there should now be a new row showing the newly configured disk. Then click on **Next: Networking**.

LUN	Name	Size (GiB)	Disk type	Host caching
0	test-vp-vm2_DataDisk...	1024	Premium SSD LRS	None

Create and attach a new disk    Attach an existing disk

Advanced

- Configure the network.

- In a production system, where other components are already in use, it would be important to use an existing virtual network so that the VM can communicate with the other cloud services in the production solution. If no virtual network has been defined in this location, create it here and configure the:
  - Subnet: The first subnet to subdivide the address space - must fit within the defined address space. After the VNet is created, you can add more subnets.
  - Public IP: Overall IPv4 space available to this network.
- On the **Networking** tab, let's change some of the settings. Under the input field for Virtual network, select **Create new**. The Create virtual network pane appears.

Setting	Value
Address space	
Address range	Select the checkbox in the row below the heading, and enter <code>172.16.0.0/16</code> to give the address space a full range of addresses. If another address range row exists, select it to delete it.
Subnets	
Subnet name	Select the checkbox in the row below the heading, and enter <code>default</code> in the first input field. If another row exists, select it to delete it.
Address range	In the empty input field, enter <code>172.16.1.0/24</code> to give the subnet 256 IP addresses of space.

# PRACTICAL NO - 7

Create virtual network

The Microsoft Azure Virtual Network service enables Azure resources to securely communicate with each other in a virtual network which is a logical isolation of the Azure cloud dedicated to your subscription. You can connect virtual networks to other virtual networks, or your on-premises network. [Learn more](#)

Name \* learn-51e719e2-0279-418e-b072-2c92ea5fc76-vnet

**Address space**  
The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

<input type="checkbox"/> Address range	Addresses	Overlap		...
<input type="checkbox"/> 172.16.0.0/16	172.16.0.0 - 172.16.255.255 (65536 addresses)	None		...
	(0 Addresses)	None		

**Subnets**  
The subnet's address range in CIDR notation. It must be contained by the address space of the virtual network.

<input type="checkbox"/> Subnet name	Address range	Addresses		...
<input type="checkbox"/> default	172.16.1.0/24	172.16.1.0 - 172.16.1.255 (256 addresses)		...
		(0 Addresses)		

Activate Windows  
Go to Settings to activate Windows.

**OK** **Discard**

3. Select **OK** to save your settings and return to the Create a virtual machine pane.

## Step 4: Finish configuring the VM and create the image.

On the Create a virtual machine pane, the rest of the tabs have reasonable defaults and there's no need to change any of them. You can explore the other tabs if you like. Each field has an (i) icon next to it which, if selected, will show a detailed definition of that configuration setting. Reviewing field descriptions is a great way to learn about the settings you can use to configure the VM.

- Select **Review + create**. The system will validate your options and display details about the VM being created.

# PRACTICAL NO - 7

Create a virtual machine ...

Validation passed

Basics Disks Networking Management Advanced Tags Review + create

PRODUCT DETAILS

Standard DS1 v2 by Microsoft Subscription credits apply 0.1260 USD/hr Terms of use | Privacy policy Pricing for other VM sizes

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

**⚠ You have set RDP port(s) open to the internet.** This is only recommended for testing. If you want to change this setting, go back to Basics tab.

Basics

Subscription	Concierge Subscription
Resource group	learn-3c071f87-a2ce-43a9-9e64-8d7e1c09c01e
Virtual machine name	test-vp-vm2
Region	West US
Availability options	No infrastructure redundancy required
Image	[smalldisk] Windows Server 2019 Datacenter - Gen1

Create < Previous Next > Download a template for automation

- Select **Create** to deploy the VM. The Azure dashboard will show the name VM that's being deployed and details about your deployment. Deployment may take several minutes.

CreateVm-MicrosoftWindowsServer.WindowsServer-201-20211025203256 | Overview

Deployment

Search (Ctrl+ /) < Delete Cancel Redeploy Refresh

We'd love your feedback! →

Deployment is in progress

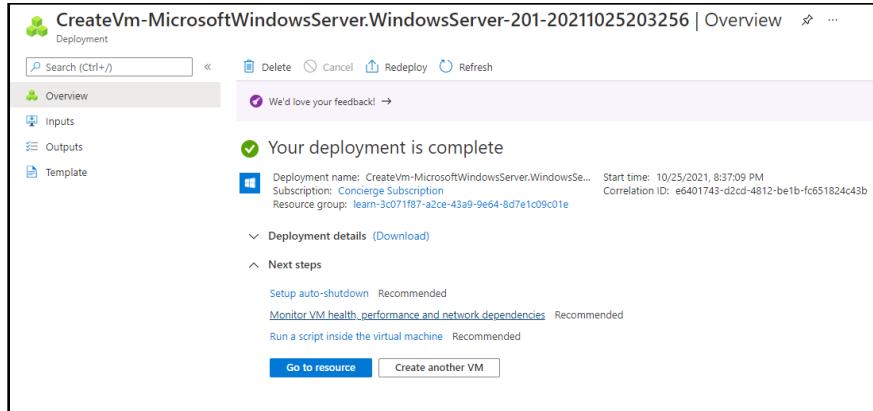
Deployment name: CreateVm-MicrosoftWindowsServer.WindowsSe.. Start time: 10/25/2021, 8:37:09 PM Correlation ID: e6401743-d2cd-4812-be1b-fc651824c43b

Subscription: Concierge Subscription Resource group: learn-3c071f87-a2ce-43a9-9e64-8d7e1c09c01e

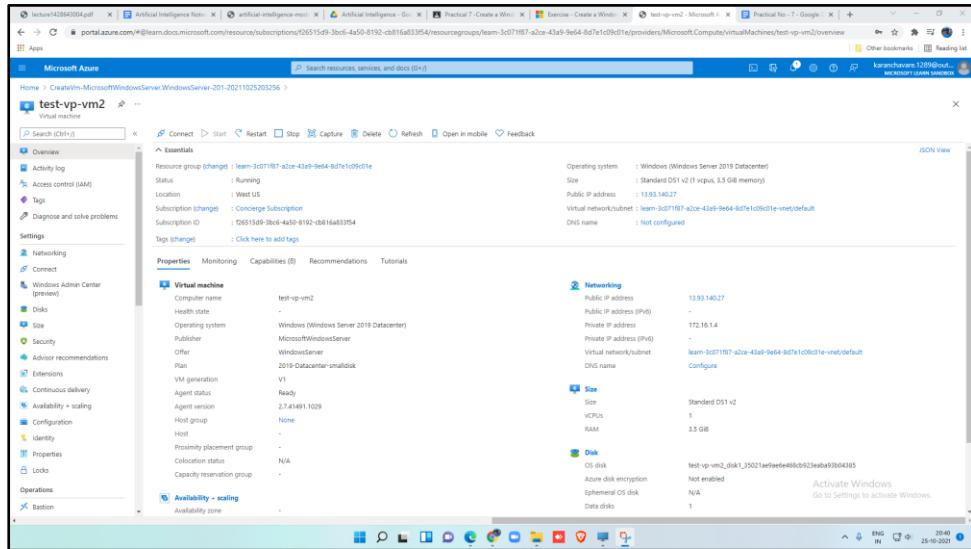
Deployment details (Download)

Resource	Type	Status	Operation details
test-vp-vm2750	Microsoft.Network/networkInterfaces	Created	Operation details
learn-3c071f87-a2ce-43a9-9e64-8d7e1c09c01e-vnet	Microsoft.Network/virtualNetworks	OK	Operation details
test-vp-vm2-ip	Microsoft.Network/publicIPAddresses	OK	Operation details
test-vp-vm2-nsg	Microsoft.Network/networkSecurityGroups	OK	Operation details
test-vp-vm2_DataDisk_0	Microsoft.Compute/disks	OK	Operation details

# PRACTICAL NO - 7



- After deployment completes, select **Go to the resource**. Your virtual machine pane appears.
- Now, let's look at what we can do with this VM.



## Step 5: Use RDP to connect to Windows Azure virtual machines.

Now that we have a Windows VM in Azure, the next thing you'll do is put your applications and data on those VMs to process our traffic videos.

However, unless you've set up a site-to-site VPN to Azure, your Azure VMs won't be accessible from your local network. If you're just getting started with Azure, it's unlikely that you have a working site-to-site VPN. So how can you transfer files to Azure VMs? One easy way is to use Azure's Remote Desktop Connections feature to share your local drives with your new Azure VMs.

Now that we have a new Windows virtual machine, we need to install our custom software onto it. There are several options to choose from:

## PRACTICAL NO - 7

- Remote Desktop Protocol (RDP)
- Custom scripts
- Custom VM images (with the software preinstalled)

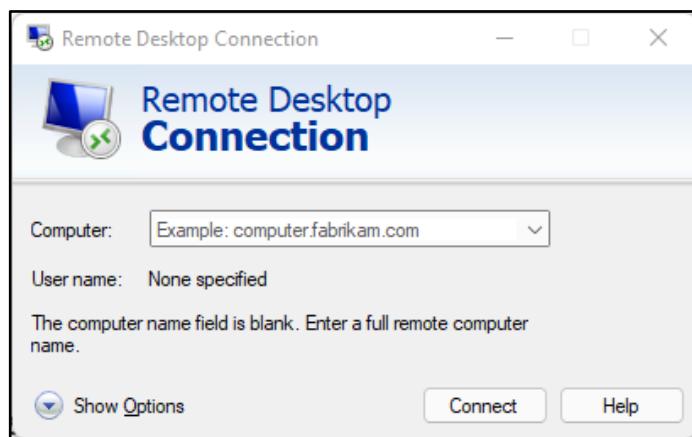
### What is the Remote Desktop Protocol?

Remote Desktop (RDP) provides remote connectivity to the UI of Windows-based computers. RDP enables you to sign in to a remote physical or virtual Windows computer and control that computer as if you were seated at the console. An RDP connection enables you to carry out the vast majority of operations that you can do from the console of a physical computer, with the exception of some power and hardware-related functions.

An RDP connection requires an RDP client. Microsoft provides RDP clients for the following operating systems:

- Windows (built-in)
- macOS
- iOS
- Android

The following screenshot displays the Remote Desktop Protocol client in Windows 11.



### Step 6: Connecting to an Azure VM.

Azure VMs communicate on a virtual network. They can also have an optional public IP address assigned to them. With a public IP, we can communicate with the VM over the Internet. Alternatively, we can set up a virtual private network (VPN) that connects our on-premises network to Azure - letting us securely connect to the VM without exposing a public IP. This approach is covered in another module and is fully documented if you are interested in exploring that option.

## PRACTICAL NO - 7

One thing to be aware of with public IP addresses in Azure is they're often dynamically allocated. That means the IP address can change over time - for VMs this happens when the VM is restarted. You can pay more to assign static addresses if you want to connect directly to an IP address instead of a name and need to ensure that the IP address won't change.

→ **Connecting to a VM in Azure using RDP is a simple process.**

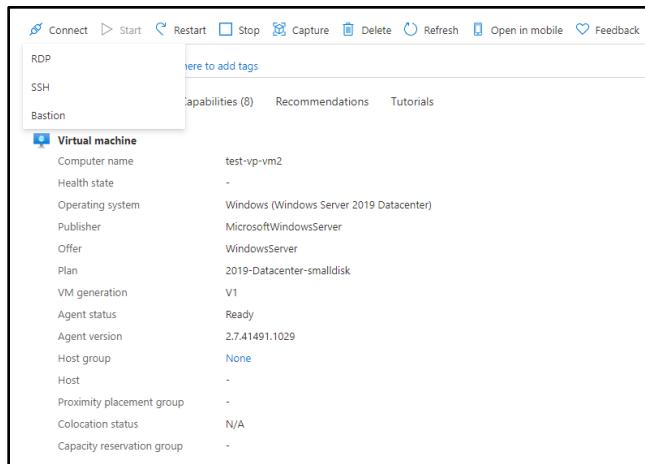
To connect to an Azure VM with an RDP client, you will need:

- Public IP address of the VM (or private if the VM is configured to connect to your network)
- Port number

You can enter this information into the RDP client or download a preconfigured RDP file.

→ **Download the RDP file.**

1. In the Azure portal, you go to the properties of your VM, and at the top, click **Connect**.

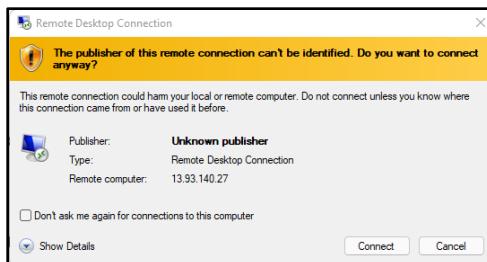


2. This will show you the IP addresses assigned to the VM and give you the option to download a preconfigured.rdp file that Windows then opens in the RDP client. You can choose to connect over the public IP address of the VM in the RDP file. Instead, if you're connecting over VPN or ExpressRoute, you can select the internal IP address. You can also select the port number for the connection.

# PRACTICAL NO - 7

The screenshot shows a web-based interface for connecting to a virtual machine via RDP. At the top, there are tabs for RDP, SSH, and BASTION, with RDP being the active tab. Below the tabs, a section titled "Connect with RDP" contains instructions: "To connect to your virtual machine via RDP, select an IP address, optionally change the port number, and download the RDP file." There are two input fields: "IP address \*" with "Public IP address (13.93.140.27)" selected, and "Port number \*" with "3389" entered. A blue button labeled "Download RDP File" is below these fields. Further down, there are links for troubleshooting: "Can't connect?", "Test your connection", "Troubleshoot RDP connectivity issues", and "Provide feedback" with a link "Tell us about your connection experience".

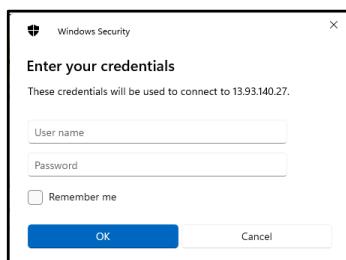
3. If you're using a static public IP address for the VM, you can save the .rdp file to your desktop. If you're using dynamic IP addressing, the .rdp file only remains valid while the VM is running. If you stop and restart the VM, you must download another .rdp file.
4. When you connect, you'll typically receive two warnings. These are:
  - Publisher warning - caused by the .rdp file not being publicly signed.
  - Certificate warning - caused by the machine certificate not being trusted.



5. In test environments, these warnings can be ignored. In production environments, the .rdp file can be signed using RDPSIGN.EXE and the machine certificate placed in the client's Trusted Root Certification Authorities store.

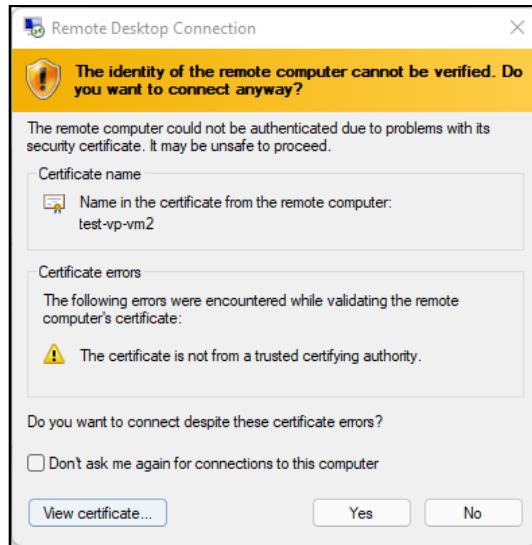
## → Connect to the Windows VM through RDP Client.

1. On the Remote Desktop Connection dialog box, note the security warning and the remote computer IP address, and then select Connect to start the connection to the VM.
2. In the Windows Security dialog box, enter the username and password that you used in creating Virtual Machine.



## PRACTICAL NO - 7

3. In the second Remote Desktop Connection dialog box, note the certificate errors, and then select Yes.



4. Here we are successfully connected to the virtual machine on the Azure cloud through RDP.

