

CLOUD COMPUTING

INDEX

Practical No.	Topic	Signature
1.	Implementing Client Server communication model using TCP	
a	A client server based program using TCP to find if the number entered is prime	
b	A client server TCP based chatting application	
2.	Implementing Client Server communication model using UDP.	
a	A client server based program using UDP to find if the number entered is even or odd.	
b	A client server based program using UDP to find the factorial of the entered number	
C	A program to implement simple calculator operations like addition, Subtraction, multiplication and division.	
d	A program that finds the square, square root, cube and cube root of the entered number.	
3.	Implementing Multicast Socket	
4.	Object communication using RMI	
a	A RMI based application program to display current date and time	
b	A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three.	
5.	Implementation of web services	
a	Implementing “Big” Web Service	
b	Implementing Web Service that connects to MySQL database	
6.	Implement Xen virtualization and manage with Xen Center	
7.	Implement virtualization using VMWare ESXi Server and managing with vCenter	
8.	Implement Windows Hyper V virtualization	
9.	Develop application for Microsoft Azure.	
10.	Develop application for Google App Engine	

Practical No: 01

Aim: Write a program for implementing Client Server communication model using TCP.
Practical 1A: A client server based program using TCP to find if the number entered is prime.

Code:

1. tcpServerPrime.java

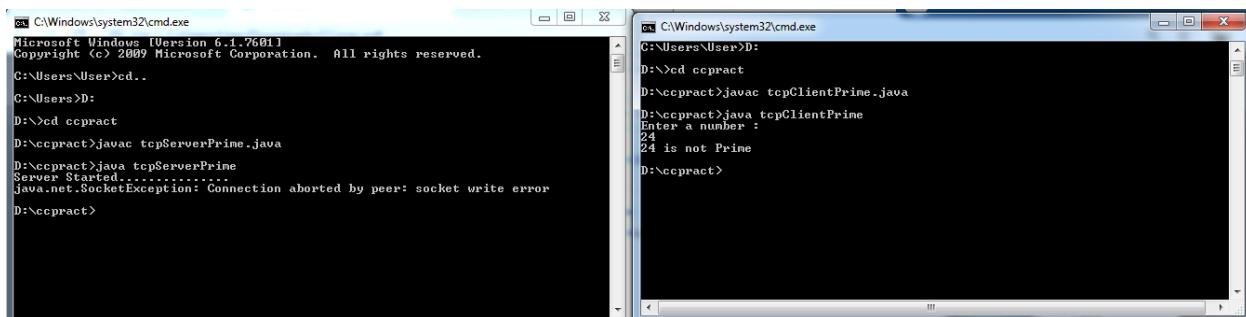
```
import java.net.*;
import java.io.*;
class tcpServerPrime
{
public static void main(String args[])
{
try
{
ServerSocket ss = new ServerSocket(8001);
System.out.println("Server Started.....");
Socket s = ss.accept();
DataInputStream in = new
DataInputStream(s.getInputStream()); int x= in.readInt();
DataOutputStream otc = new
DataOutputStream(s.getOutputStream()); int y = x/2;
if(x ==1 || x ==2 || x ==3)
{
otc.writeUTF(x + "is Prime");
System.exit(0);
}
for(int i=2; i<=y; i++)
{
if(x%i != 0)
{
otc.writeUTF(x + " is Prime");
}
else
{
otc.writeUTF(x + " is not Prime");
}
}
}
catch(Exception e)
{
System.out.println(e.toString());
}
}
}
```

2. tcpClientPrime.java

```
import java.net.*;
import java.io.*;
class tcpClientPrime
{
public static void main(String args[])
{
Try {
```

```
Socket cs = new Socket("LocalHost",8001);
BufferedReader infu = new BufferedReader(new
InputStreamReader(System.in));
System.out.println("Enter a number : ");
int a = Integer.parseInt(infu.readLine());
DataOutputStream out = new
DataOutputStream(cs.getOutputStream());
out.writeInt(a);
DataInputStream in = new
DataInputStream(cs.getInputStream());
System.out.println(in.readUTF()); cs.close();
}
catch(Exception e)
{
System.out.println(e.toString());
}
}
}
```

Output:-



Practical 1B: A client server TCP based chatting application.

Code:

1. ChatServer.java

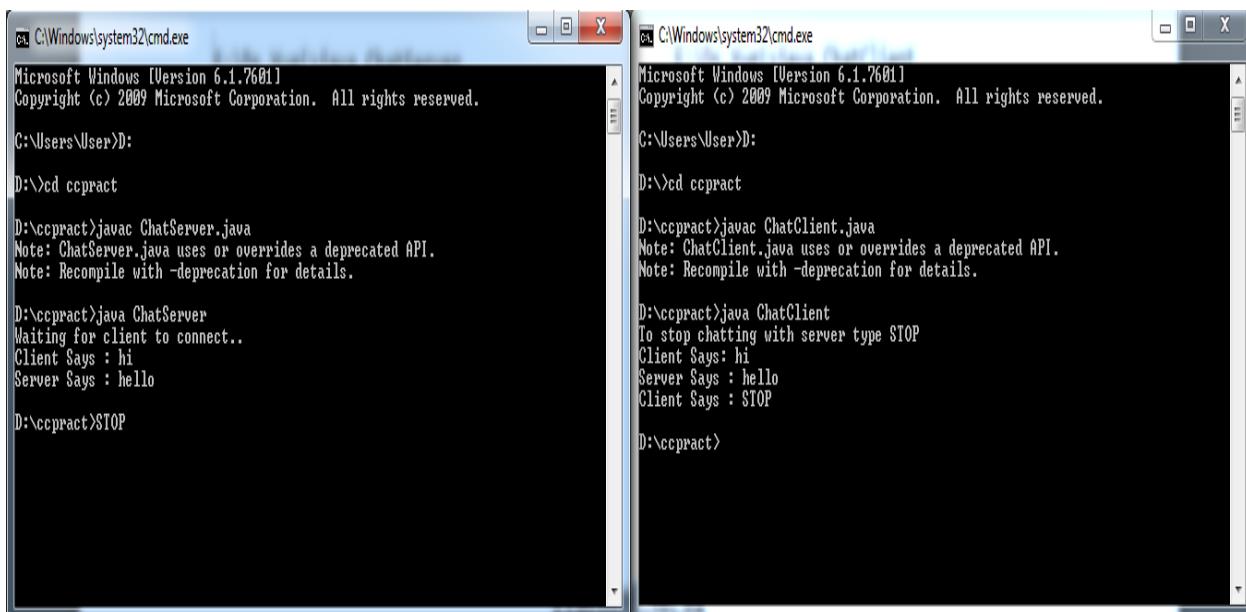
```
import java.net.*;
import java.io.*;
class ChatServer
{
public static void main(String args[])
{
try
{
ServerSocket ss = new ServerSocket(8000);
System.out.println("Waiting for client to
connect.."); Socket s = ss.accept();
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
DataOutputStream out = new
DataOutputStream(s.getOutputStream()); DataInputStream in = new
DataInputStream(s.getInputStream()); String receive, send;
while((receive = in.readLine()) != null)
{
if(receive.equals("STOP"))
break;
System.out.println("Client Says : "+receive);
System.out.print("Server Says : ");
send = br.readLine();
out.writeBytes(send+"\n");
}
br.close();
in.close();
out.close();
s.close();
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

2. ChatClient.java

```
import java.net.*;
import java.io.*;
class ChatClient
{
public static void main(String args[])
{
try
{
Socket s = new Socket("Localhost",8000);
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
DataOutputStream out = new
```

```
DataOutputStream(s.getOutputStream()); DataInputStream in = new
DataInputStream(s.getInputStream()); String msg;
System.out.println("To stop chatting with server type
STOP"); System.out.print("Client Says: "); while((msg =
br.readLine()) != null)
{
out.writeBytes(msg+"\n");
if(msg.equals("STOP"))
break;
System.out.println("Server Says : "+in.readLine());
System.out.print("Client Says : ");
}
br.close();
in.close();
out.close();
s.close();
}
catch(Exception e)
{
e.printStackTrace();
}
}
```

Output:



The image shows two Microsoft Windows command-line windows (cmd.exe) running side-by-side. Both windows display the standard Windows command prompt interface with a blue title bar and a black body.

Left Window (Chat Server):

- Shows the command: `D:\ccpract>javac ChatServer.java`
- Includes a note: `Note: ChatServer.java uses or overrides a deprecated API.`
- Includes a note: `Note: Recompile with -deprecation for details.`
- Shows the command: `D:\ccpract>java ChatServer`
- Shows the message: `Waiting for client to connect..`
- Shows the message: `Client Says : hi`
- Shows the message: `Server Says : hello`
- Shows the command: `D:\ccpract>STOP`

Right Window (Chat Client):

- Shows the command: `D:\ccpract>javac ChatClient.java`
- Includes a note: `Note: ChatClient.java uses or overrides a deprecated API.`
- Includes a note: `Note: Recompile with -deprecation for details.`
- Shows the command: `D:\ccpract>java ChatClient`
- Shows the message: `To stop chatting with server type STOP`
- Shows the message: `Client Says: hi`
- Shows the message: `Server Says : hello`
- Shows the message: `Client Says : STOP`
- Shows the command: `D:\ccpract>`

Practical No: 02

Aim: Write a program for implementing Client Server communication model using UDP.

Practical 2A: A client server based program using UDP to find if the number entered is even or odd.

Code:

1. udpServerEO.java

Program which finds entered number is even or odd

```
import java.io.*;
import java.net.*;
CLOUD COMPUTING
Teacher Reference Manual 2019-20
public class udpServerEO
{
public static void main(String args[])
{
try
{
DatagramSocket ds = new DatagramSocket(2000);
byte b[] = new byte[1024];
DatagramPacket dp = new DatagramPacket(b,b.length);
ds.receive(dp);
String str = new String(dp.getData(),0,dp.getLength());
System.out.println(str);
int a= Integer.parseInt(str);
String s= new String();
if (a%2 == 0)
s = "Number is even";
else
s = "Number is odd";
byte b1[] = new byte[1024];
b1 = s.getBytes();
DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
ds.send(dp1);
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

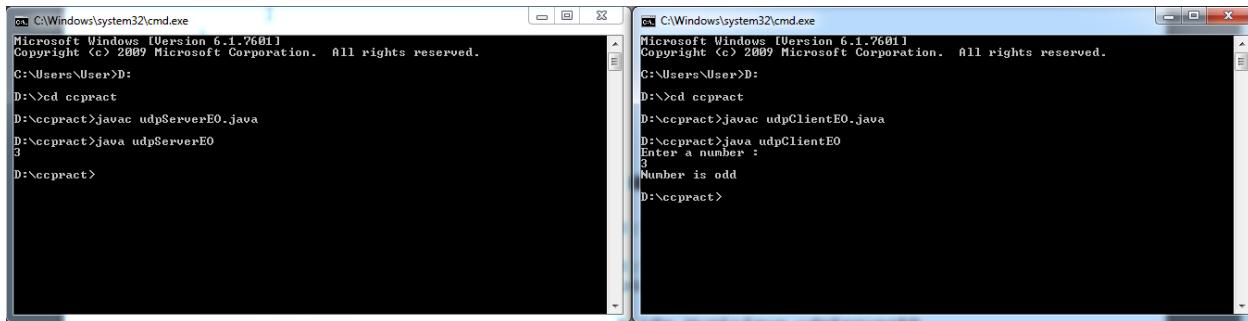
2. udpClientEO.java

Program which finds entered number is even or odd

```
import java.io.*;
import java.net.*;
public class udpClientEO
{
public static void main(String args[])
{
try
{
```

```
DatagramSocket ds = new DatagramSocket(1000);
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
System.out.println("Enter a number : ");
String num = br.readLine();
byte b[] = new byte[1024];
b=num.getBytes();
DatagramPacket dp = new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);
ds.send(dp);
byte b1[] = new byte[1024];
DatagramPacket dp1 = new
DatagramPacket(b1,b1.length); ds.receive(dp1);
String str = new
String(dp1.getData(),0,dp1.getLength());
System.out.println(str);
}
catch(Exception e)
{
e.printStackTrace();
}
}
}
```

Output:-



Practical 2B: A client server based program using UDP to find the factorial of the entered number.

Code:

1. udpServerFact.java

Program which calculate factorial of a number

```
import java.io.*;
import java.net.*;
public class udpServerFact
{
    public static void main(String args[])
    {
        try
        {
            DatagramSocket ds = new DatagramSocket(2000);
            byte b[] = new byte[1024];
            DatagramPacket dp = new DatagramPacket(b,b.length);
            ds.receive(dp);
            String str = new
String(dp.getData(),0,dp.getLength());
            System.out.println(str);
            int a= Integer.parseInt(str);
            int f = 1, i;
            String s= new String();
            for(i=1;i<=a;i++)
            {
                f=f*i;
            }
            s=Integer.toString(f);
            String str1 = "The Factorial of " + str + " is : " +
f;
            byte b1[] = new byte[1024];
            b1 =
str1.getBytes();
            DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(),1000);
            ds.send(dp1);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

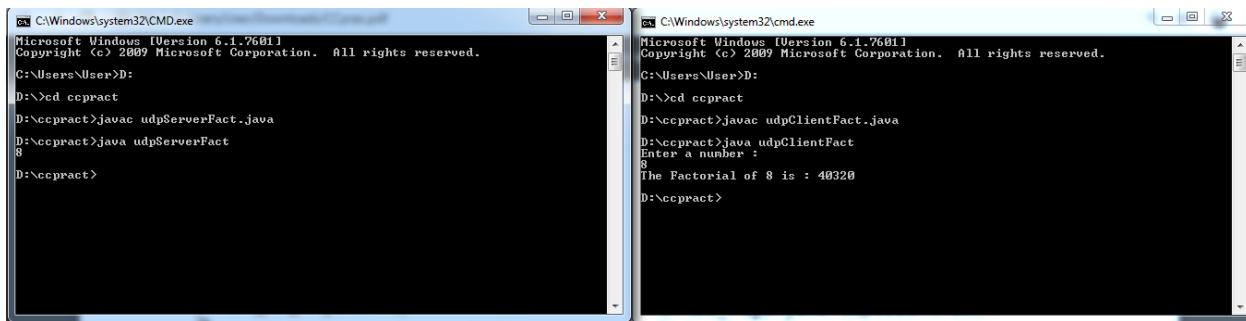
2. udpClientFact.java

Program which calculate factorial of a number

```
import java.io.*;
import java.net.*;
public class udpClientFact
{
    public static void main(String args[])
    {
```

```
try
{
DatagramSocket ds = new DatagramSocket(1000);
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
System.out.println("Enter a number : ");
String num = br.readLine();
byte b[] = new byte[1024];
b= num.getBytes();
DatagramPacket dp = new
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);
ds.send(dp);
byte b1[] = new byte[1024];
DatagramPacket dp1 = new DatagramPacket(b1,b1.length);
ds.receive(dp1);
String str = new
String(dp1.getData(),0,dp1.getLength());
System.out.println(str);
}
catch(Exception e)
{
e.printStackTrace();
}
}
```

Output:



Practical 2C: A program to implement simple calculator operations like addition,Subtraction, multiplication and division.

Code:

1. RPCServer.java

```
import java.util.*;
import java.net.*;
class RPCServer
{
DatagramSocket ds;
DatagramPacket dp;
String str,methodName,result;
int val1,val2;
RPCServer()
{
try
{
ds=new DatagramSocket(1200);
byte b[]=new byte[4096];
while(true)
{
dp=new DatagramPacket(b,b.length);
ds.receive(dp);
str=new String(dp.getData(),0,dp.getLength());
if(str.equalsIgnoreCase("q"))
{
System.exit(1);
}
else
{
StringTokenizer st = new StringTokenizer(str,"");
int i=0;
while(st.hasMoreTokens())
{
String token=st.nextToken();
methodName=token;
val1 = Integer.parseInt(st.nextToken());
val2 = Integer.parseInt(st.nextToken());
}
}
System.out.println(str);
InetAddress ia = InetAddress.getLocalHost();
if(methodName.equalsIgnoreCase("add"))
{
result= "" + add(val1,val2);
}
else if(methodName.equalsIgnoreCase("sub"))
{
result= "" + sub(val1,val2);
}
else if(methodName.equalsIgnoreCase("mul"))
{
result= "" + mul(val1,val2);
}
```

```

}
else if(methodName.equalsIgnoreCase("div"))
{
result= "" + div(val1,val2);
}
byte b1[]=result.getBytes();
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
System.out.println("result :" +result +"\n");
ds1.send(dp1);
}
}
catch (Exception e)
{
e.printStackTrace();
}
}
public int add(int val1, int val2)
{
return val1+val2;
}
public int sub(int val3, int val4)
{
return val3-val4;
}
public int mul(int val3, int val4)
{
return val3*val4;
}
public int div(int val3, int val4)
{
return val3/val4;
}
public static void main(String[] args)
{
new RPCServer();
}
}

```

2. RPCClient.java

```

Import java.io.*;
import java.net.*;
class RPCClient
{
RPCClient()
{
try
{
InetAddress ia = InetAddress.getLocalHost();
DatagramSocket ds = new DatagramSocket();
DatagramSocket ds1 = new DatagramSocket(1300);
System.out.println("\nRPC Client\n");

```

```
System.out.println("Enter method name and parameter like add 34\n");
while (true)
{
BufferedReader br = new
BufferedReader(new InputStreamReader(System.in));
String str = br.readLine();
byte b[] = str.getBytes();
DatagramPacket dp = new
DatagramPacket(b,b.length,ia,1200);
ds.send(dp);
dp = new DatagramPacket(b,b.length);
ds1.receive(dp);
String s = new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult = " + s + "\n");
}
}
catch (Exception e)
{
e.printStackTrace();
}
}
public static void main(String[] args)
{
new RPCClient();
}
}
```

Output:

The image displays two separate command-line windows from a Windows operating system. Both windows are titled 'C:\Windows\system32\cmd.exe'.

Left Window (RPCServer.java):

```
C:\Windows\system32\cmd.exe - java RPCServer
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>D:
D:>cd mscit
D:\mscit>cd cc
D:\mscit\cc>javac RPCServer.java
D:\mscit\cc>java RPCServer
add 5 2
result :7
```

Right Window (RPCClient.java):

```
C:\Windows\system32\cmd.exe - java RPCClient
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>D:
D:>cd mscit
D:\mscit>cd cc
D:\mscit\cc>javac RPCClient.java
D:\mscit\cc>java RPCClient
RPC Client
Enter method name and parameter like add 34
add 5 2
Result = 7
```

Practical 2D: A program that finds the square, square root, cube and cube root of the entered number.

Code:

1. RPCNumServer.java

```
import java.util.*;
import java.net.*;
import java.io.*;
class RPCNumServer
{
DatagramSocket ds;
DatagramPacket dp;
String str,methodName,result;
int val;
RPCNumServer()
{
try
{
ds=new DatagramSocket(1200);
byte b[]=new byte[4096];
while(true)
{
dp=new DatagramPacket(b,b.length);
ds.receive(dp);
str=new
String(dp.getData(),0,dp.getLength());
if(str.equalsIgnoreCase("q"))
{
System.exit(1);
}
else
{
StringTokenizer st = new StringTokenizer(str, " ");
int i=0;
while(st.hasMoreTokens())
{
String token=st.nextToken();
methodName=token;
val = Integer.parseInt(st.nextToken());
}
}
System.out.println(str);
InetAddress ia = InetAddress.getLocalHost();
if(methodName.equalsIgnoreCase("square"))
{
result= "" + square(val);
}
else if(methodName.equalsIgnoreCase("squareroot"))
{
result= "" + squareroot(val);
}
else if(methodName.equalsIgnoreCase("cube"))
{
}
```

```

result= "" + cube(val);
}
else if(methodName.equalsIgnoreCase("cuberoot"))
{
result= "" + cuberoot(val);
}
byte b1[]=result.getBytes();
DatagramSocket ds1 = new DatagramSocket();
DatagramPacket dp1 = new
DatagramPacket(b1,b1.length,InetAddress.getLocalHost(), 1300);
System.out.println("result :
"+result+"\n"); ds1.send(dp1);
}
}
catch (Exception e)
{
e.printStackTrace();
}
}
public double square(int a) throws Exception
{
double ans;
ans = a*a;
return ans;
}
public double squareroot(int a) throws Exception
{
double ans;
ans = Math.sqrt(a);
return ans;
}
public double cube(int a) throws Exception
{
double ans;
ans = a*a*a;
return ans;
}
public double cuberoot(int a) throws Exception
{
double ans;
ans = Math.cbrt(a);
return ans;
}
public static void main(String[] args)
{
new RPCNumServer();
}
}

```

2. RPCNumClient.java

```

import java.io.*;
import java.net.*;

```

```
class RPCNumClient
{
RPCNumClient()
{
try
{
InetAddress ia = InetAddress.getLocalHost();
DatagramSocket ds = new DatagramSocket();
DatagramSocket ds1 = new DatagramSocket(1300);
System.out.println("\nRPC Client\n");
System.out.println("1. Square of the number - square\n2. Square root
of the number - squareroot\n3. Cube of the number - cube\n4. Cube root of the number -
cuberoott");
System.out.println("Enter method name and the number\n");
while (true)
{
BufferedReader br = new
BufferedReader(new InputStreamReader(System.in));
String str = br.readLine();
byte b[] = str.getBytes();
DatagramPacket dp = new
DatagramPacket(b,b.length,ia,1200);
ds.send(dp);
dp = new DatagramPacket(b,b.length);
ds1.receive(dp);
String s = new String(dp.getData(),0,dp.getLength());
System.out.println("\nResult = " + s + "\n");
}
}
catch (Exception e)
{
e.printStackTrace();
}
}
public static void main(String[] args)
{
new RPCNumClient();
}
```

Output:

The screenshot shows two windows side-by-side. The left window is titled 'C:\Windows\system32\cmd.exe - java RPCNumServer' and contains the following text:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>cd Desktop
C:\Users\User\Desktop>javac RPCNumServer.java
C:\Users\User\Desktop>javac RPCNumClient.java
C:\Users\User\Desktop>java RPCNumServer
square ?
result : 49.0
squareroot 25
result : 5.0
cube 3
result : 27.0
cuberoot 27
result : 3.0
```

The right window is titled 'C:\Windows\system32\cmd.exe - java RPCNumClient' and contains the following text:

```
RPC Client
1. Square of the number - square
2. Square root of the number - squareroot
3. Cube of the number - cube
4. Cube root of the number - cuberoot
Enter method name and the number
square 7
Result = 49.0
squareroot 25
Result = 5.0
cube 3
Result = 27.0
cuberoot 27
Result = 3.0
```

Practical No: 03

Aim: A multicast Socket example.

Code:

1. BroadcastServer.java

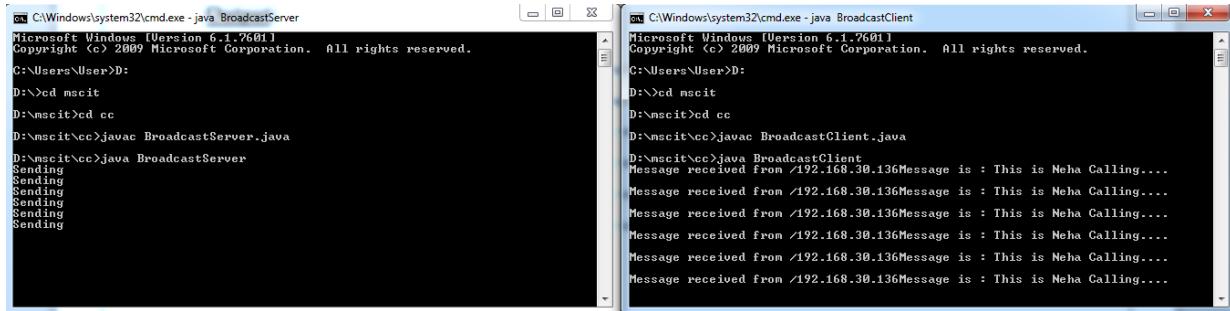
```
import java.net.*;
import java.io.*;
import java.util.*;
public class BroadcastServer
{
    public static final int PORT = 1234;
    public static void main(String args[])throws Exception {
        MulticastSocket socket;
        DatagramPacket packet;
        InetAddress address;
        address = InetAddress.getByName("239.1.2.3");
        socket = new MulticastSocket();
        socket.joinGroup(address);
        byte[] data = null; for(;;)
        {
            Thread.sleep(10000);
            System.out.println("Sending "); String
            str = ("This is Neha Calling...."); data
            = str.getBytes();
            packet = new DatagramPacket(data, str.length(),address,PORT);
            socket.send(packet);
        }
    }
}
```

2. BroadcastClient.java

```
import java.net.*;
import java.io.*;
public class BroadcastClient
{
    public static final int PORT = 1234;
    public static void main(String args[])throws Exception {
        MulticastSocket socket;
        DatagramPacket packet;
        InetAddress address;
        address = InetAddress.getByName("239.1.2.3");
        socket = new MulticastSocket(PORT);
        socket.joinGroup(address); byte[]
        data = new byte[100];
        packet = new DatagramPacket(data,data.length());
        for(;;)
        {
            socket.receive(packet);
            String str = new String(packet.getData()); System.out.println("Message
            received from "+ packet.getAddress() +
            "Message is : "+str);
        }
    }
}
```

}

Output



The image shows two separate command-line windows running on Microsoft Windows 6.1.7601. Both windows have a title bar 'C:\Windows\system32\cmd.exe - java BroadcastServer' or 'BroadcastClient'. The left window displays the server's broadcast loop, showing multiple 'Sending' messages. The right window displays the client receiving multiple broadcast messages from the server.

Left Window (Server):

```
C:\Windows\system32\cmd.exe - java BroadcastServer
Microsoft Windows [Version 6.1.7601]
Copyright <>> 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>D:
D:>cd mscit
D:>cd cc
D:\mscit\cc>javac BroadcastServer.java
D:\mscit\cc>java BroadcastServer
Sending
Sending
Sending
Sending
Sending
Sending
```

Right Window (Client):

```
C:\Windows\system32\cmd.exe - java BroadcastClient
Microsoft Windows [Version 6.1.7601]
Copyright <>> 2009 Microsoft Corporation. All rights reserved.

C:\Users\User>D:
D:>cd mscit
D:>cd cc
D:\mscit\cc>javac BroadcastClient.java
D:\mscit\cc>java BroadcastClient
Message received from /192.168.30.136Message is : This is Neha Calling....
Message received from /192.168.30.136Message is : This is Neha Calling....
Message received from /192.168.30.136Message is : This is Neha Calling....
Message received from /192.168.30.136Message is : This is Neha Calling....
Message received from /192.168.30.136Message is : This is Neha Calling....
```

Practical No: 04

Aim: Write a program to show the object communication using RMI.

Practical 4A: A RMI based application program to display current date and time.

Code:

1. InterDate.java

```
import java.rmi.*;  
public interface InterDate extends Remote  
{  
    public String display() throws Exception;  
}
```

2. ServerDate.java

```
import java.rmi.*;  
import java.rmi.server.*;  
import java.util.*;  
public class ServerDate extends UnicastRemoteObject implements  
InterDate {  
    public ServerDate() throws Exception  
    {  
    }  
    public String display() throws Exception  
    {  
        String str = "";  
        Date d = new Date();  
        str = d.toString();  
        return str;  
    }  
    public static void main(String args[]) throws  
Exception {  
    ServerDate s1 = new ServerDate();  
    Naming.bind("DS",s1);  
    System.out.println("Object registered.....");  
    }  
}
```

3. ClientDate.java

```
import java.rmi.*;  
import java.io.*;  
public class ClientDate  
{  
    public static void main(String args[]) throws  
Exception {  
    String s1;  
    InterDate h1 = (InterDate)Naming.lookup("DS");  
    s1 = h1.display();  
    System.out.println(s1);  
    }  
}
```

Output:

The image displays three separate windows of a Windows Command Prompt (cmd.exe) running on Microsoft Windows Version 6.1.7601. Each window shows a sequence of commands being entered and their corresponding outputs.

- Top Left Window:** Shows the command `rmiregistry` being run, which creates a registry object on the server.
- Top Right Window:** Shows the command `java ServerDate` being run, which registers the `ServerDate` object with the registry.
- Bottom Window:** Shows the command `javac ClientDate.java` being run, followed by the output of the Java application, which prints the current date and time: `Fri Jan 03 02:32:25 IST 2020`.

Practical 4B: A RMI based application program that converts digits to words, e.g. 123 will be converted to one two three.

Code:

1. InterConvert.java

```
import java.rmi.*;  
public interface InterConvert extends Remote  
{  
    public String convertDigit(String no) throws Exception;  
}
```

2. ServerConvert.java

```
import java.rmi.*;  
import java.rmi.server.*;  
public class ServerConvert extends UnicastRemoteObject implements  
InterConvert {  
    public ServerConvert() throws Exception  
    {  
    }  
    public String convertDigit(String no) throws Exception  
    {  
        String str = "";  
        for(int i = 0; i < no.length(); i++)  
        {  
            int p = no.charAt(i);  
            if( p == 48)  
            {  
                str += "zero ";  
            }  
            if( p == 49)  
            {  
                str += "one ";  
            }  
            if( p == 50)  
            {  
                str += "two ";  
            }  
            if( p == 51)  
            {  
                str += "three ";  
            }  
            if( p == 52)  
            {  
                str += "four ";  
            }  
            if( p == 53)  
            {  
                str += "five ";  
            }  
            if( p == 54)  
            {  
                str += "six ";  
            }  
            if( p == 55)
```

```
{  
str += "seven ";  
}  
if( p == 56)  
{  
str += "eight ";  
}  
if( p == 57)  
{  
str += "nine ";  
}  
}  
return str;  
}  
public static void main(String args[]) throws  
Exception {  
ServerConvert s1 = new ServerConvert();  
Naming.bind("Wrd",s1);  
System.out.println("Object registered....");  
}
```

3. ClientConvert.java

```
import java.rmi.*;  
import java.io.*;  
public class ClientConvert  
{  
public static void main(String args[]) throws  
Exception {  
InterConvert h1 =  
(InterConvert)Naming.lookup("Wrd"); BufferedReader  
br = new BufferedReader(new  
InputStreamReader(System.in));  
System.out.println("Enter a number :\t"); String no = br.readLine();  
String ans = h1.convertDigit(no);  
System.out.println("The word representation of the entered digit is : " +ans);  
}  
}
```

Output:

The image displays three separate Windows command-line windows (cmd.exe) arranged vertically. The top-left window shows the compilation of a Java file named ServerConvert.java using the javac command. It also shows the creation of a class named ServerConvert using the rmiregistry command. The top-right window shows the compilation of ClientConvert.java using the javac command. The bottom window shows the execution of the Java application using the java command, resulting in the output "Object registered....".

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>javac ServerConvert.java
error: cannot read: ServerConvert.java
1 error
D:\nscit\cc>javac ServerConvert.java
D:\nscit\cc>rmiregistry

C:\Windows\system32\cmd.exe - rmiregistry

C:\Windows\system32\cmd.exe
C:\Users\user>D:
D:\>not recognized as an internal or external command,
operable program or batch file.
C:\Users\user>D:
D:\>ed nscit
D:\nscit>
nscit is not recognized as an internal or external command,
operable program or batch file.
D:\nscit>d cc
D:\nscit>javac ClientConvert.java
D:\nscit>javac ClientConvert.java
D:\nscit>java ClientConvert
Enter a number : 5
The word representation of the entered digit is : five
D:\nscit>

C:\Windows\system32\cmd.exe - java ServerConvert

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\user>:
D:\>ed nscit
D:\nscit>d cc
D:\nscit>java ServerConvert
Object registered....
```

Practical No: 05

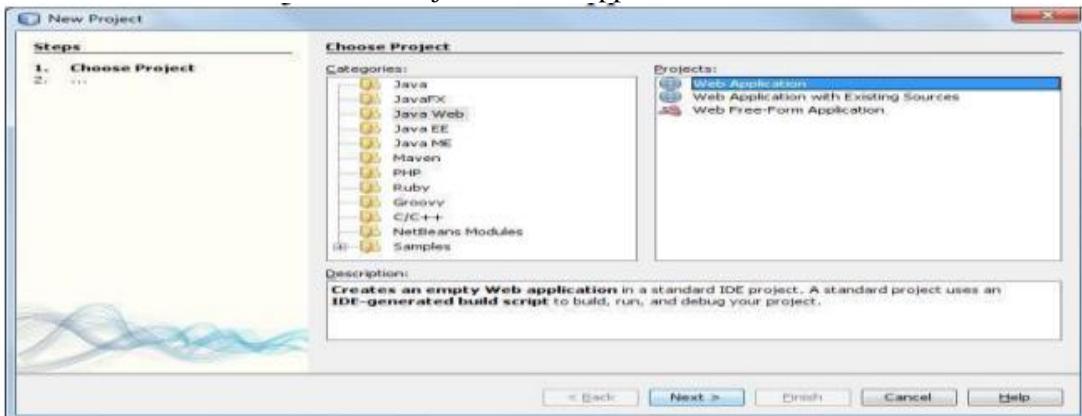
Aim: Show the implementation of web services.

Practical 5A: Implementing “Big” Web Service.

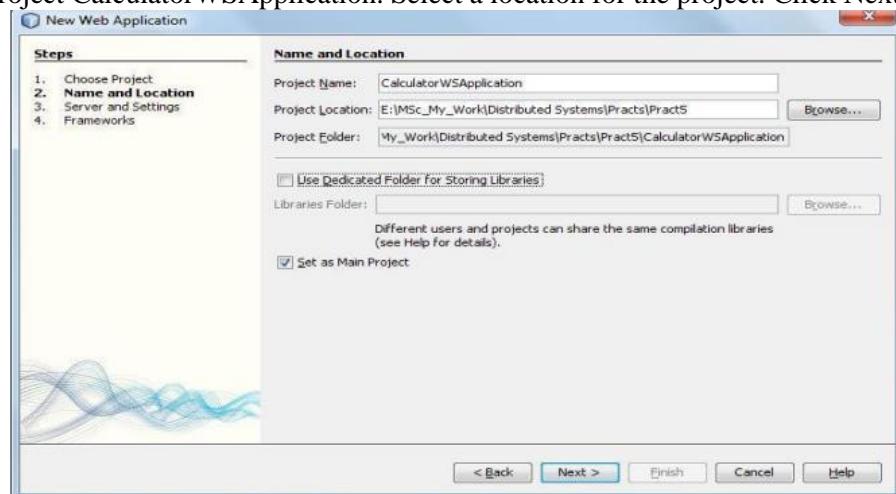
1) Creating a Web Service

A. Choosing a Container:

1. Choose File > New Project. Select Application from the Java Web.



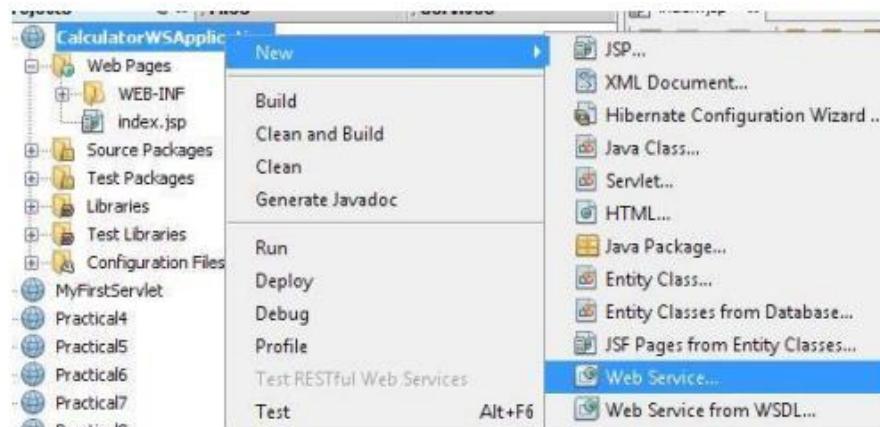
2. Name the project CalculatorWSApplication. Select a location for the project. Click Next.



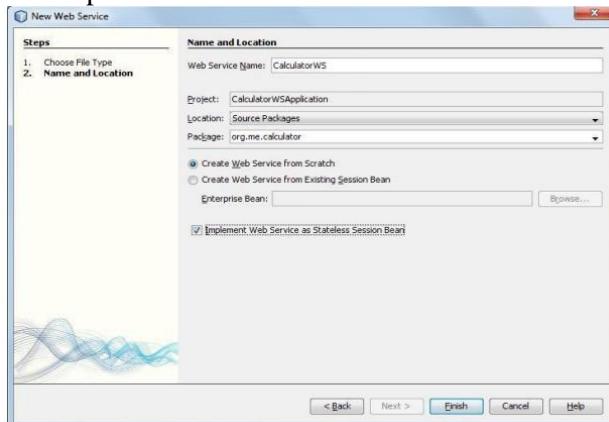
3. Select your server and Java EE version and click Finish.

B. Creating a Web Service from a Java Class

1. Right-click the CalculatorWSApplication node and choose New > Web Service.



2. Name the web service CalculatorWS and type org.me.calculator in Package. Leave Create Web Service from Scratch selected. If you are creating a Java EE 6 project on GlassFish or WebLogic, select Implement Web Service as a Stateless Session Bean.



3. Click Finish. The Projects window displays the structure of the new web service and the source code is shown in the editor area.

2) 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.

A. To add an operation to the web service:

1. Change to the Design view in the editor.



2. Click Add Operation in either the visual designer or the context menu. The Add Operation dialog opens.

3. In the upper part of the Add Operation dialog box, type add in Name and type int in

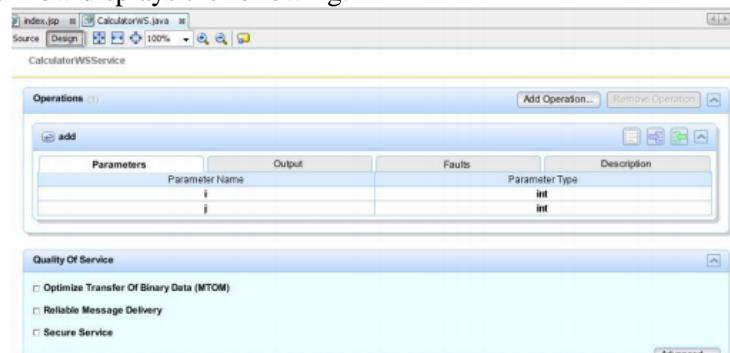
the Return Type drop-down list.

4. In the lower part of the Add Operation dialog box, click Add and create a parameter of type int named i.
5. Click Add again and create a parameter of type int called j. You now see the following:



6. Click OK at the bottom of the Add Operation dialog box. You return to the editor.

7. The visual designer now displays the following:



8. Click Source. And code the following.

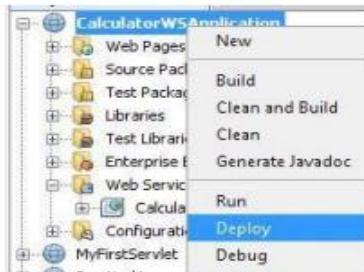
```
@WebMethod(operationName = "add")
public int add(@WebParam(name = "i") int i, @WebParam(name = "j") int j)
{
    int k = i + j;
    return k;
}
```

3) Deploying and Testing the Web Service

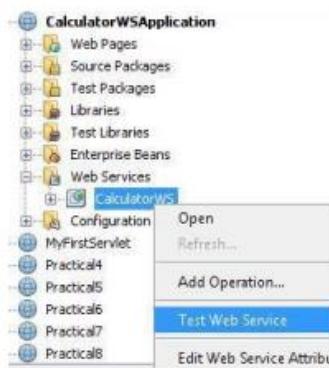
After you deploy a web service to a server, you can use the IDE to open the server's test client, if the server has a test client. The GlassFish and WebLogic servers provide test clients.

A. To test successful deployment to a GlassFish or WebLogic server:

1. Right-click the project and choose Deploy. The IDE starts the application server, builds the application, and deploys the application to the server



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



3. The IDE opens the tester page in your browser, if you deployed a web application to the GlassFish server.
4. If you deployed to the GlassFish server, type two numbers in the tester page, as shown below:
5. The sum of the two numbers is displayed:

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

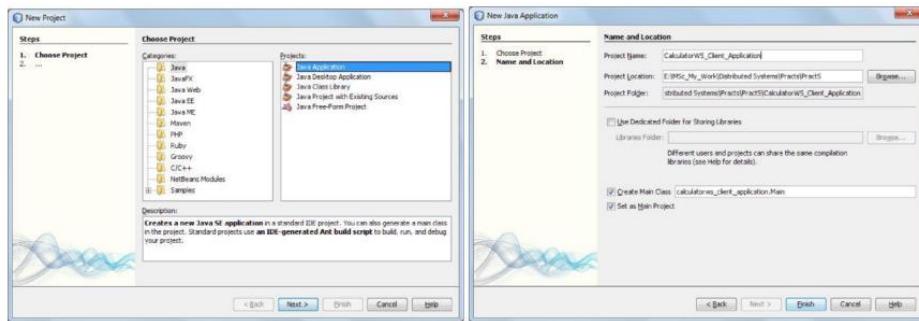
```
public abstract int org.me.calculator.CalculatorWS.add(int,int)
add (2 ,3 )
```

4) Consuming the Web Service

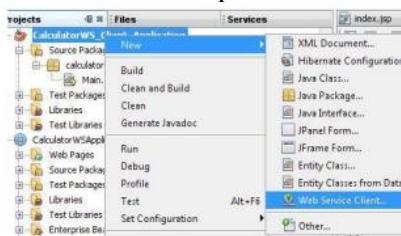
Now that you have deployed the web service, you need to create a client to make use of the web service's add method.

Client: Java Class in Java SE Application

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



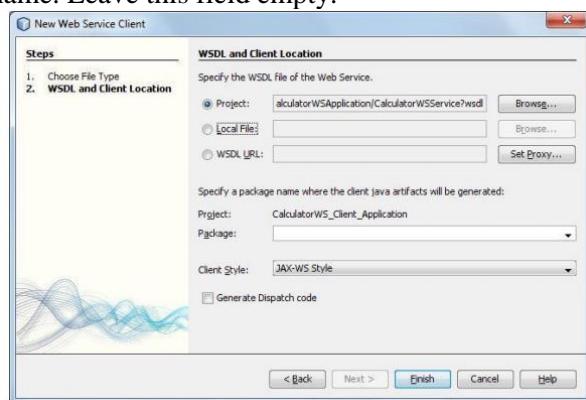
2. Right-click the CalculatorWS_Client_Application node and choose New > Web Service Client. The New Web Service Client wizard opens.



3. Select Project as the WSDL source. Click Browse. Browse to the CalculatorWS web service in the CalculatorWSApplication project. When you have selected the web service, click OK.



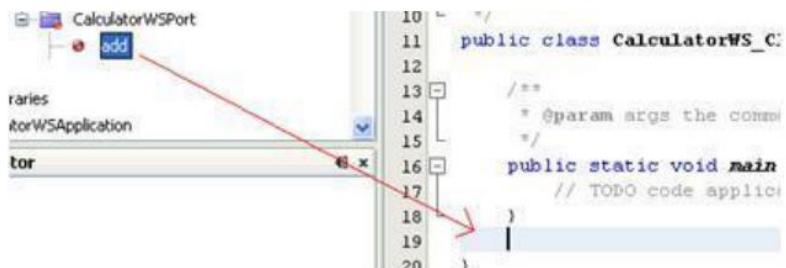
4. Do not select a package name. Leave this field empty.



5. 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.



You now see the following:

```

public static void main(String[] args)
{
// TODO code application logic here
}
private static int add(int i, int j)
{
org.me.calculator.CalculatorWS_Service service = new
org.me.calculator.CalculatorWS_Service();
org.me.calculator.CalculatorWS port =
service.getCalculatorWSPort(); return port.add(i, j);
}

```

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

```

public static void main(String[] args)
{
int i = 3;
int j = 4;
int result = add(i, j);
System.out.println("Result = " + result);
}

```

- Surround the main() method code with a try/catch block that prints an exception.

```

public static void main(String[] args)
{
try
{
int i = 3;
int j = 4;

```

```
int result = add(i, j);
System.out.println("Result = " + result);
} catch (Exception ex) {
System.out.println("Exception: " + ex);
}
```

9. Right-click the project node and choose Run.

The Output window now shows the sum:

compile:

run:

Result = 7

BUILD SUCCESSFUL (total time: 1 second)

Practical 5B: Implementing Web Service that connects to MySQL database.

Building Web Service:-

JAX-WS is an important part of the Java EE platform.

JAX-WS simplifies the task of developing Web services using Java technology.

It provides support for multiple protocols such as SOAP, XML and by providing a facility for supporting additional protocols along with HTTP.

With its support for annotations, JAX-WS simplifies Web service development and reduces the size of runtime files.

Here basic demonstration of using IDE to develop a JAX-WS Web Service is given.

After creating the web service, create web service clients that use the Web service over a network which is called consuming a web service.

The client is a servlet in a web application.

Let's build a Web Service that returns the book name along with its cost for a particular ISBN.

To begin building this service, create the data store. The server will access the data stored in a MySQL table to serve the client.

1) Creating MySQL DB Table

create database bookshop;

use bookshop;

Create a table named Books that will store valid books information

create table books(isbn varchar(20) primary key, bookname varchar(100), bookprice varchar(10));

Insert valid records in the Books table

insert into books values("111-222-333","Learn My SQL","250");

insert into books values("111-222-444","Java EE 6 for Beginners","850");

insert into books values("111-222-555","Programming with Android","500");

insert into books values("111-222-666","Oracle Database for you","400");

insert into books values("111-222-777","Asp.Net for advanced programmers","1250");

2) Creating a web service

i. Choosing a container

Web service can be either deployed in a Web container or in an EJB container.

If a Java EE 6 application is created, use a Web container because EJBs can be placed directly in a Web application.

ii. Creating a web application

To create a Web application, select File - New Project.

New Project dialog box appears. Select Java Web available under the Categories section and Web Application available under the Projects section. Click Next.

New Web Application dialog box appears. Enter BookWS as the project name in the Project Name textbox and select the option Use Dedicated Folder for Storing Libraries.

Click Next. Server and Settings section of the New Web Application dialog box appears.

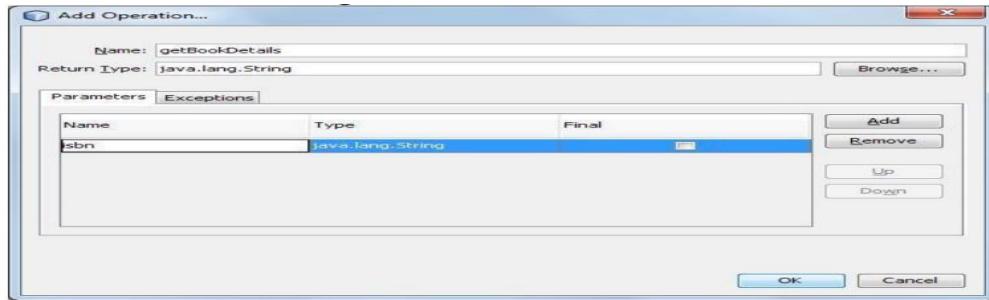
Choose the default i.e. GlassFish v3 Domain as the Web server, the Java EE 6 Web as the Java EE version and the Context Path.

Click –Finish

The Web application named BookWS is created.

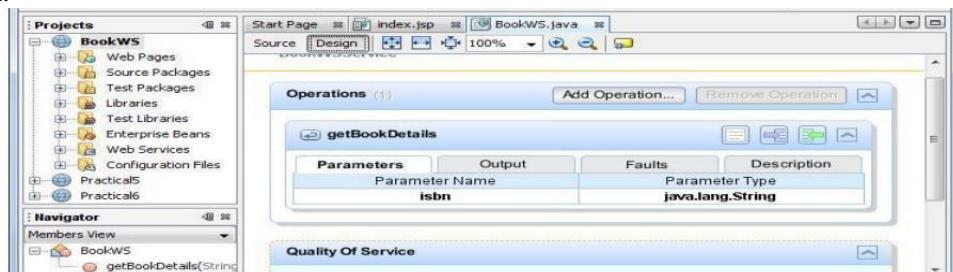
iii. Creating a web service

Right-click the BookWS project and select New -> Web Service as shown in diagram.



New Web Service dialog box appears. Enter the name BookWS in the Web Service Name textbox, webservice in the Package textbox, select the option Create Web Service from scratch and also select the option implement web service as a stateless session bean as shown in the diagram.

Click Finish.



The web service in the form of java class is ready.

3) Designing the web service

Now add an operation which will accept the ISBN number from the client to the web service.

i. Adding an operation to the web service

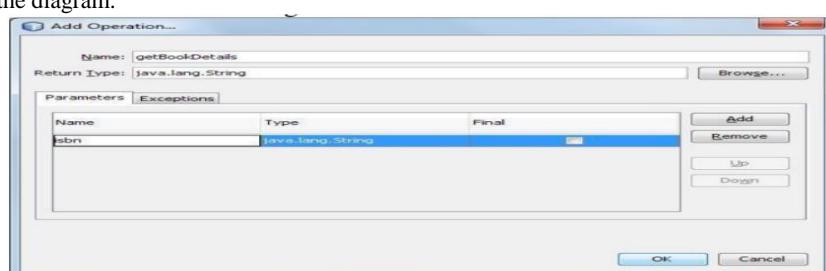
Change the source view of the BookWS.java to design view by clicking Design available just below the name of the BookWS.java tab.

The window changes as shown in the diagram.

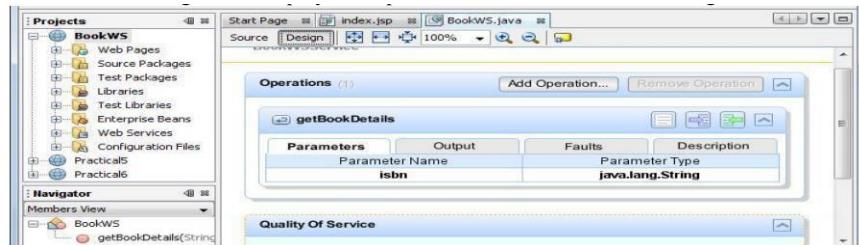
Click Add Operation available in the design view of the web service.

Add Operation dialog appears. Enter the name getBookDetails in the Name textbox and java.lang.String in the Return Type textbox as shown in the diagram.

In Add Operation dialog box, click Add and create a parameter of the type String named as shown in the diagram.



Click Ok. The design view displays the operation added as shown in the diagram.



Click Source. The code spec expands due to the operation added to the web service as shown in the diagram.



Modify the code spec of the web service BookWS.java.

Code Spec

```

package webservice;
import java.sql.*;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
import javax.ejb.Stateless;
@WebService()
@Stateless()
public class BookWS {
    /**
     * Web service operation
     */
    @WebMethod(operationName = "getBookDetails")
    public String getBookDetails(@WebParam(name = "isbn") String isbn) {
        //TODO write your implementation code here:
        Connection dbcon = null;
        Statement stmt = null;
        ResultSet rs = null;
        String query = null;
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            dbcon =
                DriverManager.getConnection("jdbc:mysql://localhost/bookshop","root","123");
            stmt = dbcon.createStatement();
            query = "select * from books where isbn = " +isbn+
                ""; rs = stmt.executeQuery(query); rs.next();
        }
    }
}

```

```

String bookDetails = "<h1>The name of the book is <b>" + rs.getString("bookname") + "</b> and its cost is <b>" + rs.getString("bookprice") + "</b></h1>..";
return bookDetails;
}
catch(Exception e)
{
System.out.println("Sorry failed to connect to the database.." + e.getMessage());
}
return null;
}
}

```

Explanation

In the above code number entered by returned.

spec, a database connection is established.

Based on the ISBN the user, the associated book name and price is retrieved and

CLOUD COMPUTING

Teacher Reference Manual 2019-20

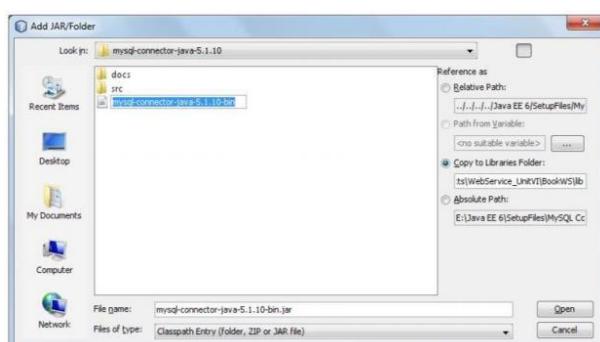
4) Adding the MySQL connector

We need to add a reference of MySQL connector to our web service. It is via this connector that our web service will be able to communicate with the database.

Right click on the libraries and select Add JAR/Folder as shown in the diagram.



Choose the location where mysql-coonector-java-5.1.10-bin is located, select it and click on open as shown.



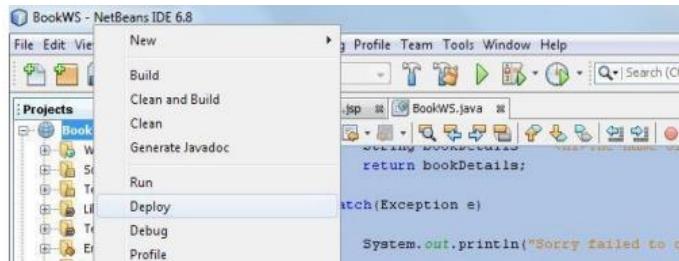
5) Deploying and testing the web service

When a web service is deployed to a web container, the IDE allows testing the web service to see if it functions as expected.

The tester application provided by GlassFish, is integrated into the IDE for this purpose as it allows the developer to enter values and test them.

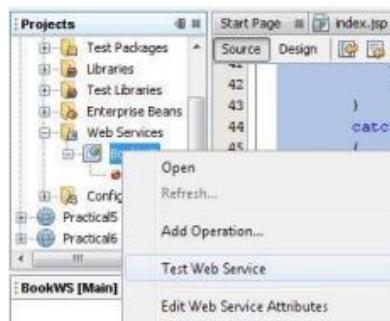
No facility for testing whether an EJB module is deployed successfully is currently available.

To test the BookWS application, right click the BookWS project and select Deploy as shown in the diagram.



The IDE starts the server, builds the application and deploys the application to the server. Follow the progress of these operations in the BookWS (run-deploy) and GlassFish v3 Domain tabs in the Output view.

Now expand the web services directory of the BookWS project, right-click the BookWS Web service and select Test web service as shown in the diagram.



The IDE opens the tester page in the web browser, if the web application is deployed using GlassFish server as shown in the figure.



Enter the ISBN number as shown in the diagram.

Click getBookDetails. The book name and its cost are displayed as shown in the diagram.



Method parameter(s)

Type	Value
java.lang.String	111-222-333

6) Consuming the web service

Once the web service is deployed, the next most logical step is to create a client to make use of the web service's getBookDetails() method.

i. Creating a web application

To create a web application, select File -> New Project.

New project dialog box appears, select java web available under the categories section and web application available under the projects section. Click Finish.

New web application dialog box appears. Enter BookWSServletClient as the project name in the Project Name textbox and select the option Use Dedicated Folder for Storing Libraries.

Click Next. Server and settings section of the new web application, dialog box appears.

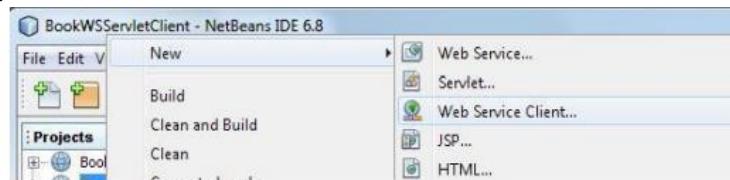
Choose the default i.e. GlassFish v3 Domain as the web server, the Java EE 6 web as the Java EE version and the context path.

Click Finish.

The web application named BookWSServletClient is created.

ii. Adding the web service to the client application

Right-click the BookWSServletClient project and select New -> Web Service Client as shown in the diagram.



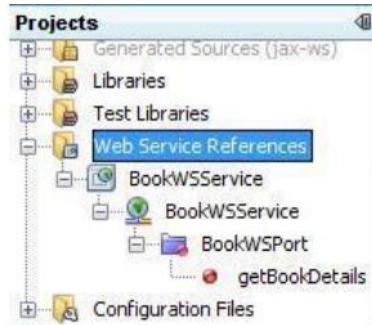
New Web Service Client dialog box appears. In the Project section, click Browse and browse through the web service which needs to be consumed. Click ok. The name of the web service appears in the New Web Service Client as shown in the diagram.

Leave the other settings as it is.

Click Finish.



The Web Service Reference directory is added to the BookWSServletClient application as shown in the diagram. It displays the structure of the newly created client including the getBookDetails() method created earlier.



iii. Creating a servlet

Create retreiveBookDetails.java using NetBeans IDE.

Right click source package directory, select New -> Servlet.

New Servlet dialog box appears. Enter retreiveBookDetails in the Class Name textbox and enter servlet in the package textbox.

Click Next. Configure Servlet Deployment section of the New Servlet dialog box appears. Keep the defaults.

Click Finish.

This creates the servlet named retreiveBookDetails.java in the servlet package.

retreiveBookDetails.java is available with the default skeleton created by the NetBeans IDE which needs to be modified to hold the application logic.

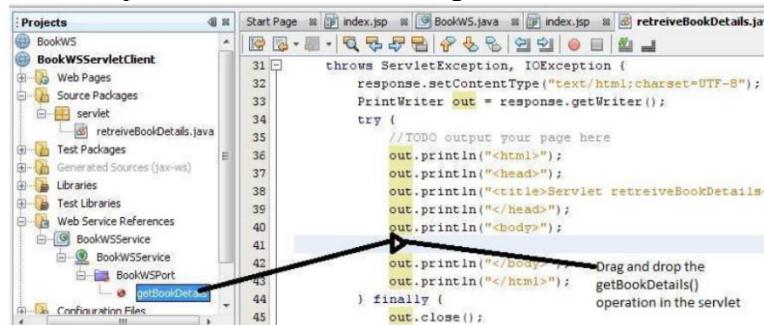
In the retreiveBookDetails.java source code, remove the following comments available in the body of the processRequest() method.

```
/*TODO output your page here*/
```

Replace the following code spec:

```
out.println("<h1>Servlet retreiveBookDetails at " + request.getContextPath () + "</h1>");
```

With the code spec of the getBookDetails() operation of the web service by dragging and dropping the getBookDetails operation as shown in the diagram.



The Servlet code spec changes as shown in the diagram

```

40 // Content-type: text/html
41 out.println("<html>");
42 out.println("  <head>");
43 out.println("    <tittle>BookWSServlet : retrieveBookDetails</tittle>");
44 out.println("  </head>");
45 out.println("  <body>");
46
47     try { // Call Web-Service Operation
48         webservice.BookWS port = service.getBookWSPort();
49         // TODO initialize WS operation arguments here
50         java.lang.String isbn = "";
51         // TODO process result here
52         java.lang.String result = port.getBookDetails(isbn);
53         out.println("Result = "+result);
54     } catch (Exception ex) {
55         // TODO handle custom exceptions here
56     }
57 }

```

The web service is instantiated by the @WebServiceRef annotation.

Now change the following code spec:

java.lang.String isbn = "“”;

to

java.lang.String isbn = request.getParameter("isbn");

iv. Creating an HTML form

Once the web service is added and the servlet is created, the form to accept ISBN from the user needs to be coded.

Since NetBeans IDE by default [as a part of Web Application creation] makes available index.jsp file. Modify it to hold the following code spec.

```

<%@page contentType="text/html" pageEncoding="UTF-8"%><!DOCTYPE
HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>SOAP Cleint - Get Book Details</title>
</head>
<body bgcolor="pink">
<form name="frmgetBookDetails" method="post" action="retreiveBookDetails">
<h1>
ISBN : <input type="text" name="isbn"/><br><br>
</h1>
<input type="submit" value="Submit"/>
</form>
</body>
</html>

```

v. Building the Web Application

Build the web application.

Right click BookWSServletClient project and select Build.

Once the Build menu item is clicked the details about the compilation and building of the BookWSServletClient Web application appears in the output – BookWSServletClient (dist) window.

vi. Running the Application

Once the compilation and building of the web application is done run the application.

Right click the BookWSServerCleint project and select run.

Once the run processing completes in NetBeans IDE a web browser is automatically launched and the BookWSServletCleint application is executed as shown in the diagram.

Enter the ISBN as shown in the diagram

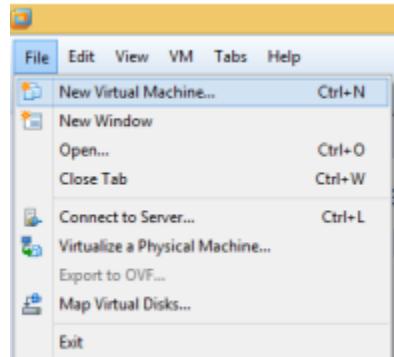


Click Submit. The book name and its cost are displayed as shown in the diagram.

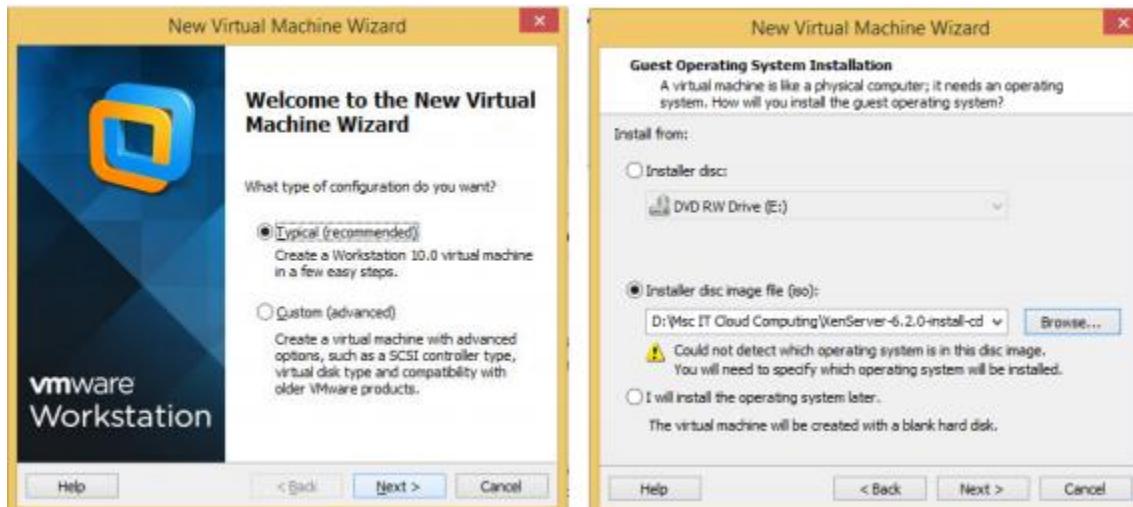


Practical no:-6

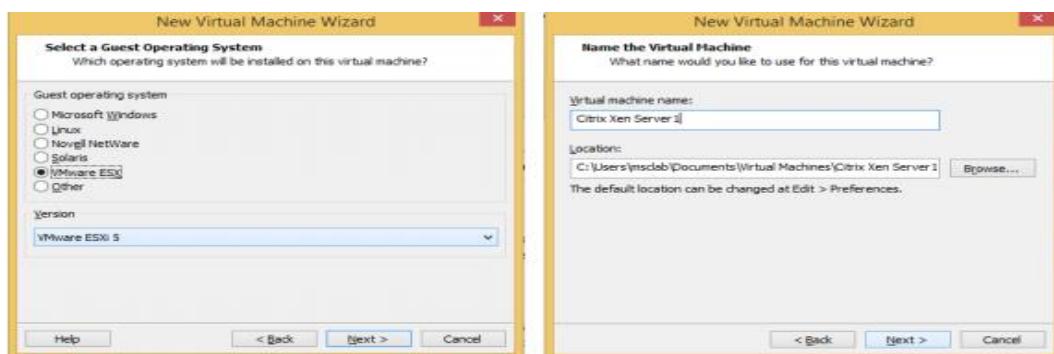
Create a new Virtual Machine in VMware Workstation FileNew Virtual Machine...



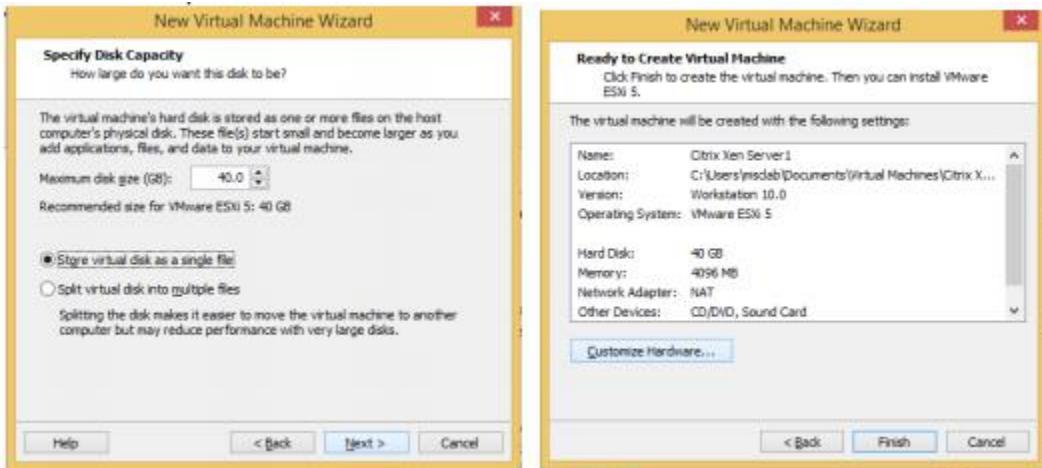
In the New Virtual Machine Wizard select Typical (recommended) and click on “Next” button. In the next step to select the iso file click on the Browse... button and select “XenServer-6.2.0- install-cd” file. Then click on the “Next” button



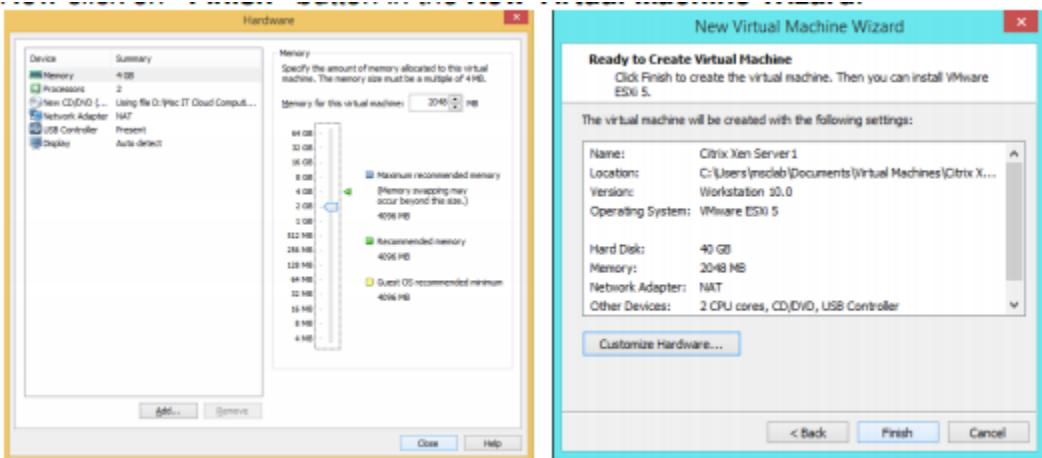
Select Guest OS as “VMware ESXi” and Version as “VMware ESXi 5”. Give a name to the Virtual Machine.



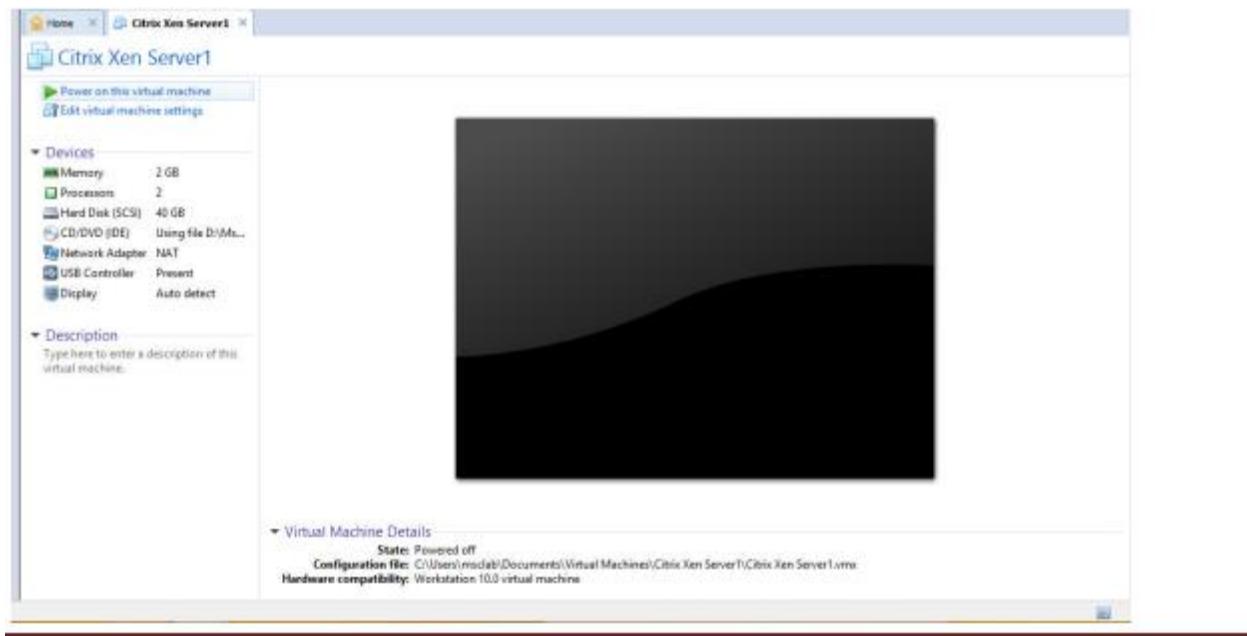
In the next step select “Store virtual disk as a single file” and click on “Next” button. At the final step click on the “Customize Hardware...” button.



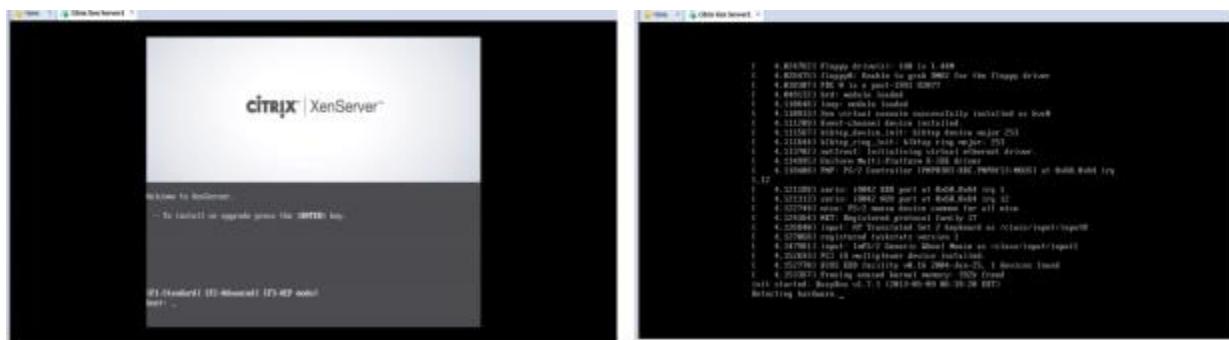
At the Hardware window select Memory size as 2GB and Close the window. Now click on “Finish” button in the New Virtual Machine Wizard.



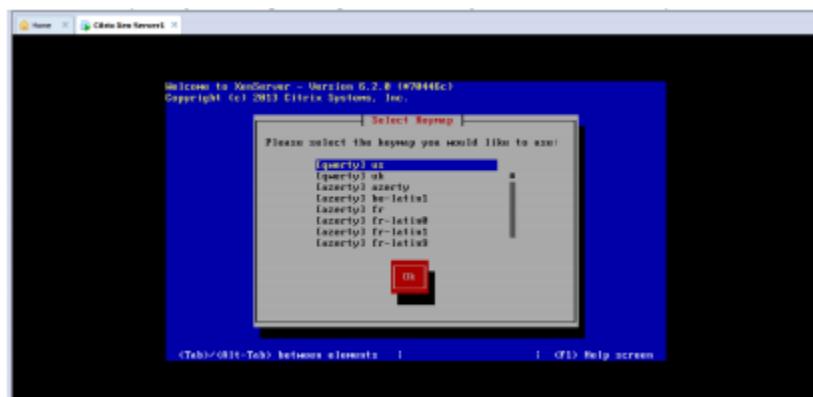
Power On the Citrix Xen Server1 virtual machine. Power On the Citrix Xen Server1 virtual machine.



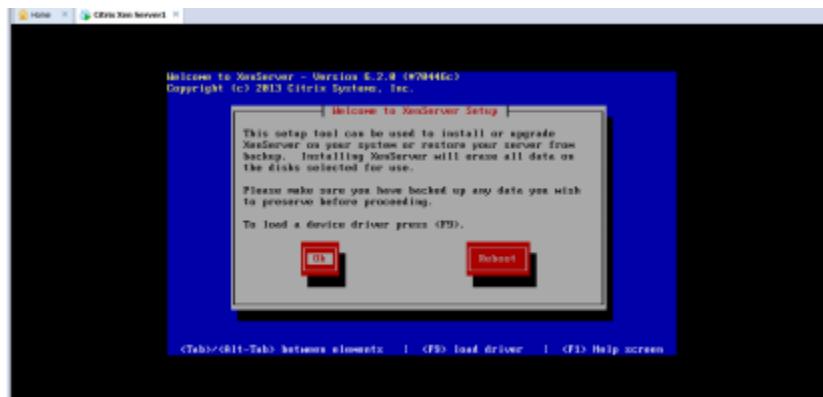
The Xen Server is starting...



Select Keypad as [qwerty] us and press Enter key.



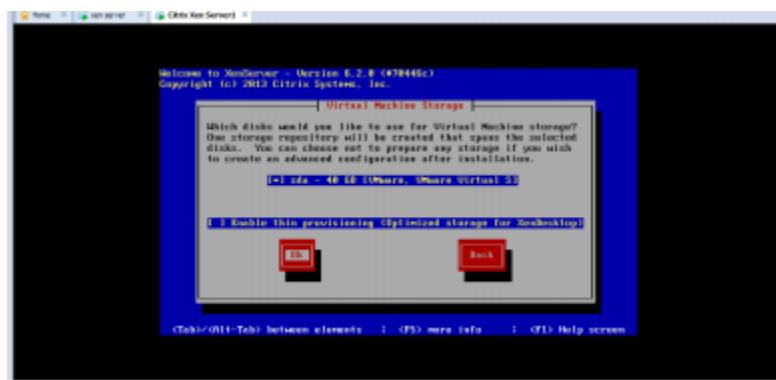
In the Welcome to XenServer Setup screen press Enter key to choose Ok button.



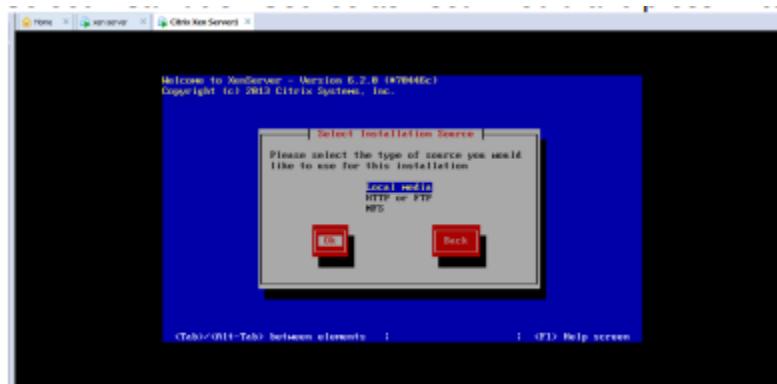
For License Agreement select Back and press Enter.



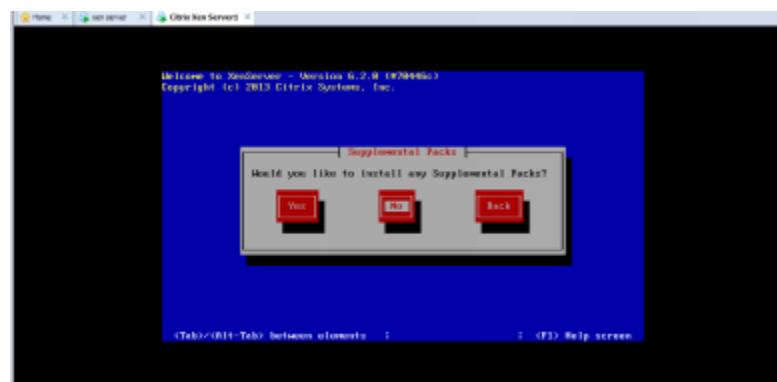
Leave the VM Storage default and click on Ok (press Enter)



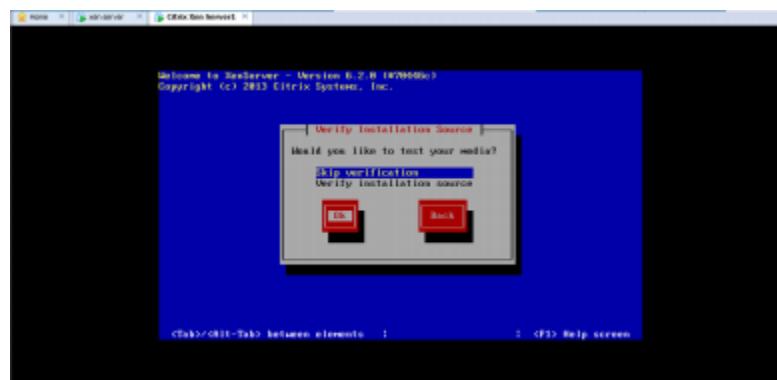
Select Installation Source as Local media and press Enter key



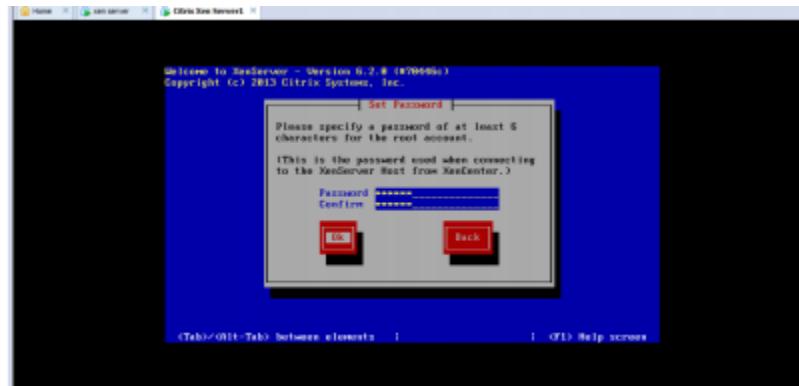
For the installation of Supplemental Packs select No button and press Enter key



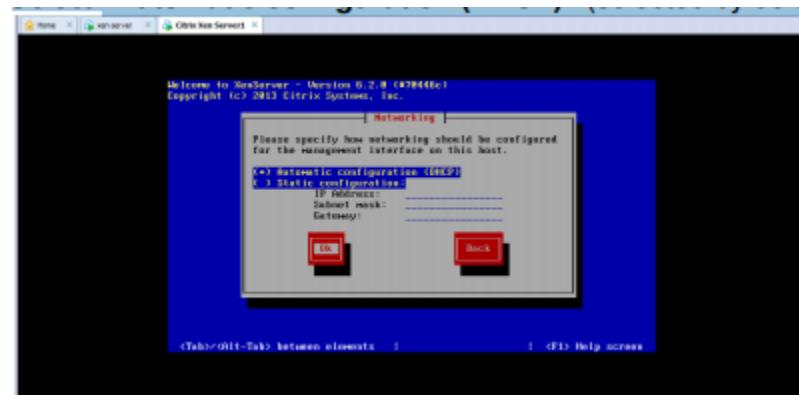
Select Skip Verification and click on Ok button (press Enter key).



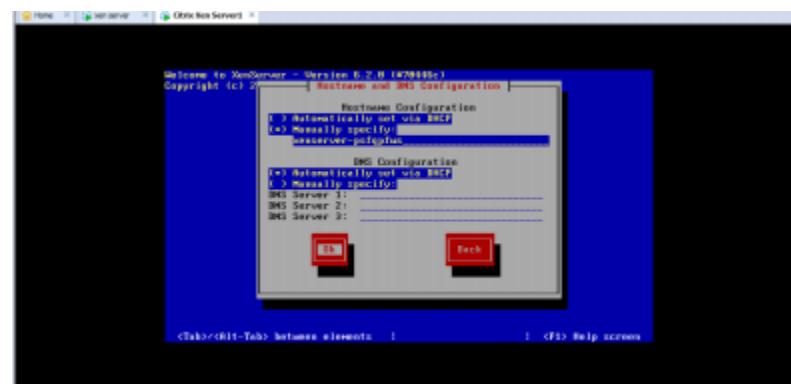
Enter password. (This password will be used from XenCenter to connect to the XenServer Host)



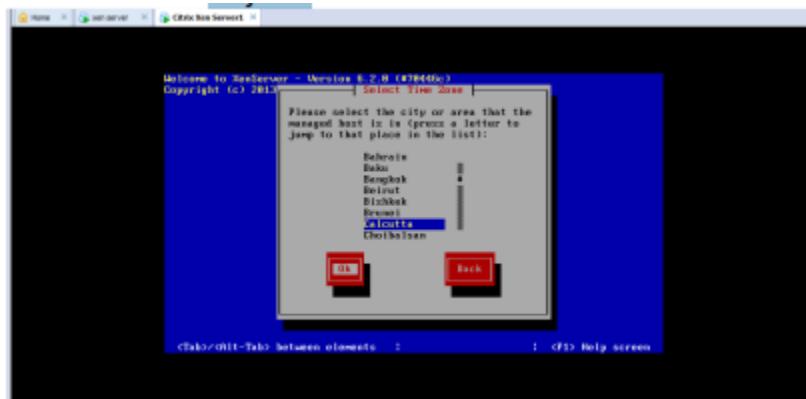
Select “Automatic configuration (DHCP)” (selected by default) and press Enter key



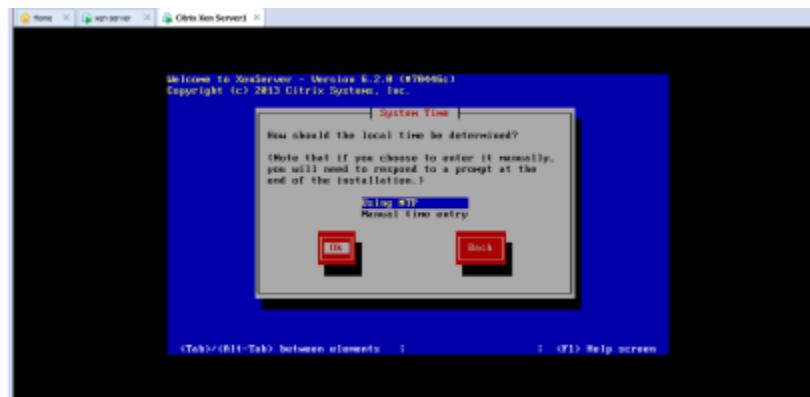
For Hostname and DNS Configuration too keep the default configuration and press Enter key.



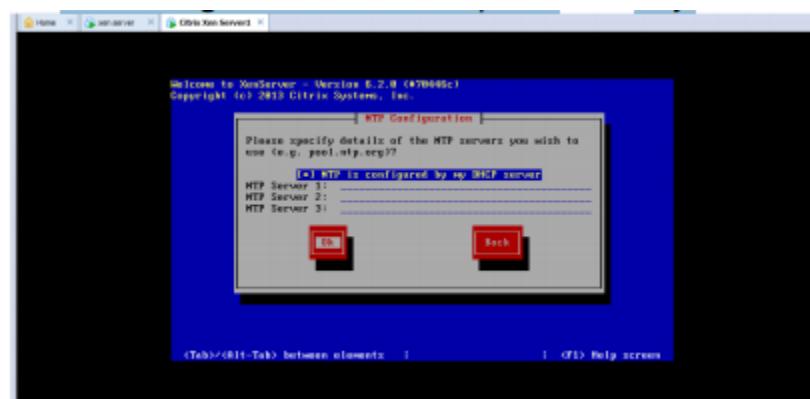
Select Calcutta city for Time Zone.



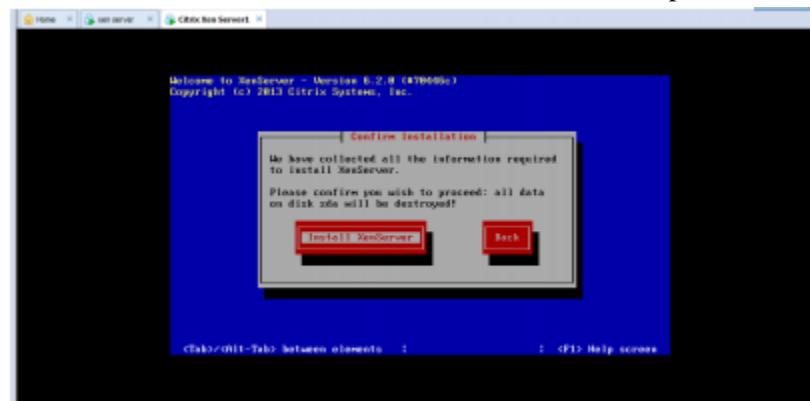
Select System Time as Using NTP and press Enter key.



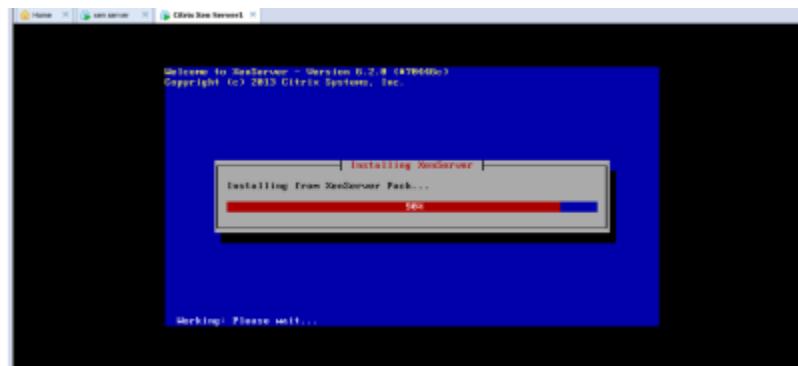
For NTP Configuration default and press Enter key.



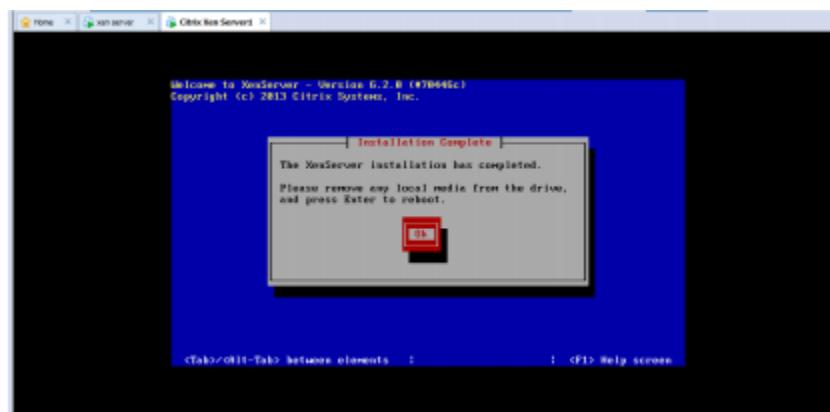
At Confirm Installation screen select Install XenServer and press Enter key.



XenServer is been installed...



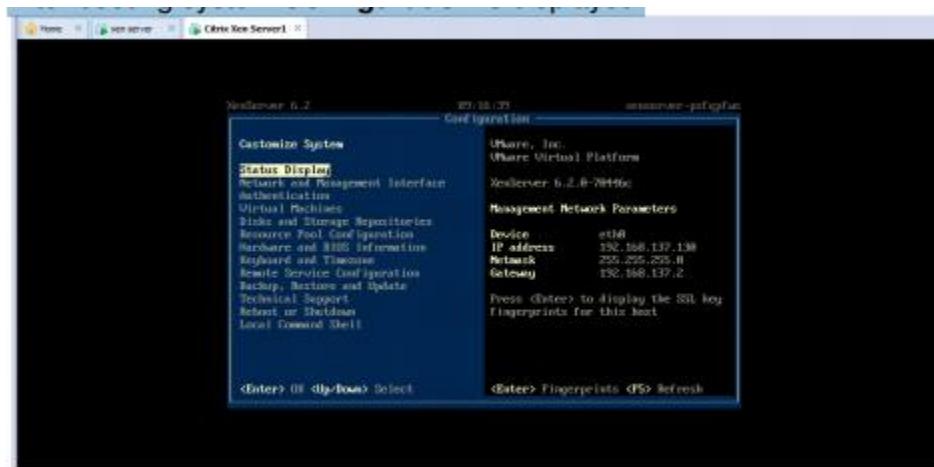
At the Installation Complete screen press Enter key



The XenServer is booting...



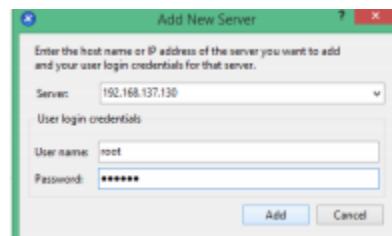
After booting system Configuration is displayed.



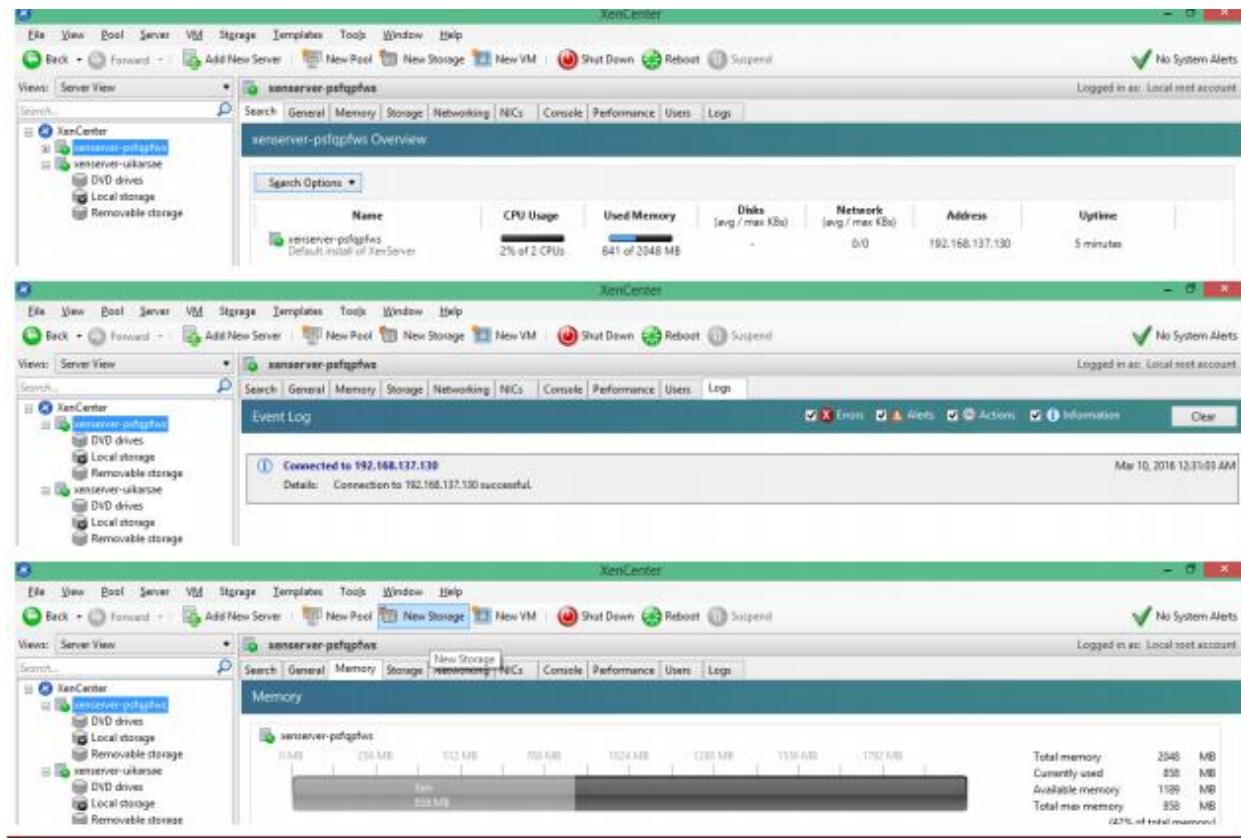
Install XenCenter. Open XenCenter. At the Home screen click on ADD a server option.



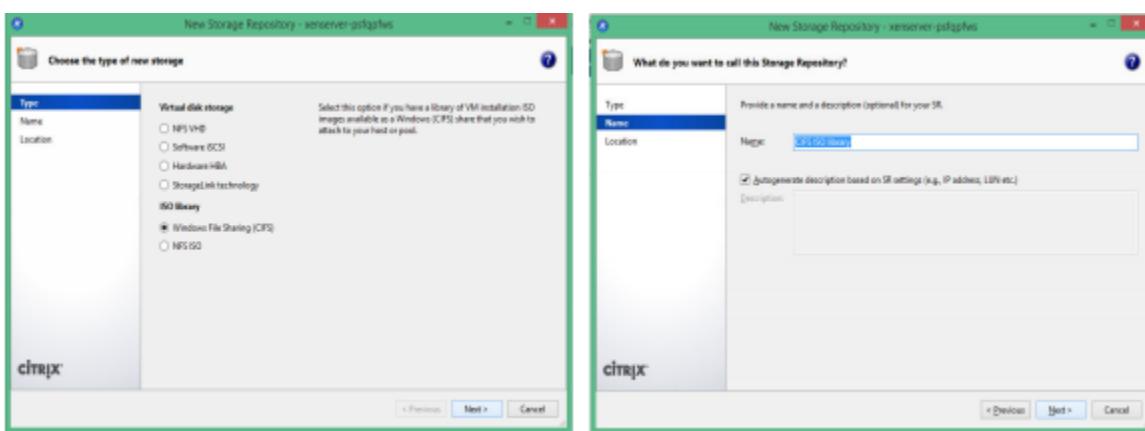
In the Add New Server enter the User name as “root” and password that was used during the XenServer installation. Click on “Add” button.



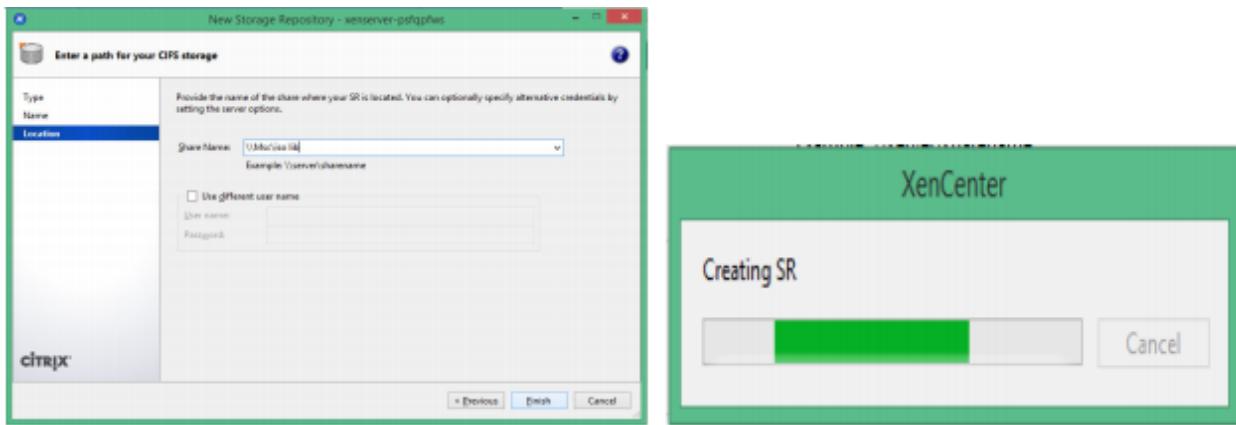
The added server can be seen in XenCenter Search, Logs and Memory tab. Click on New Storage in the Tools bar.



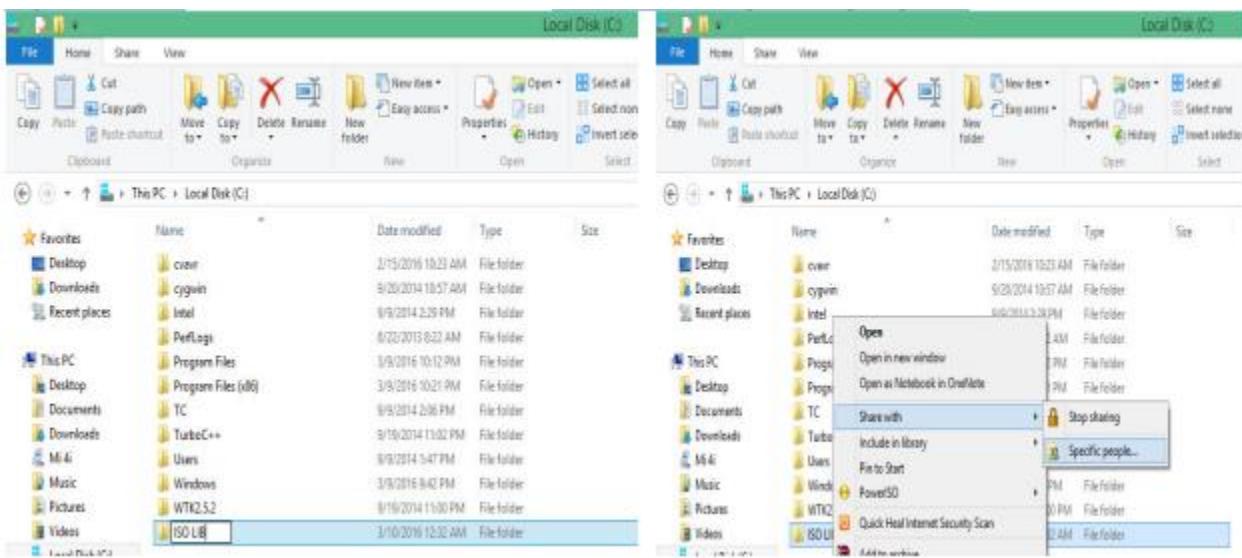
In the New Storage Repository window under ISO library select “Windows File Sharing (CIFS)”. Click on the “Next” button. In next screen give a new name to the new storage here it is “CIFS ISO library”. Click on the “Next” button.



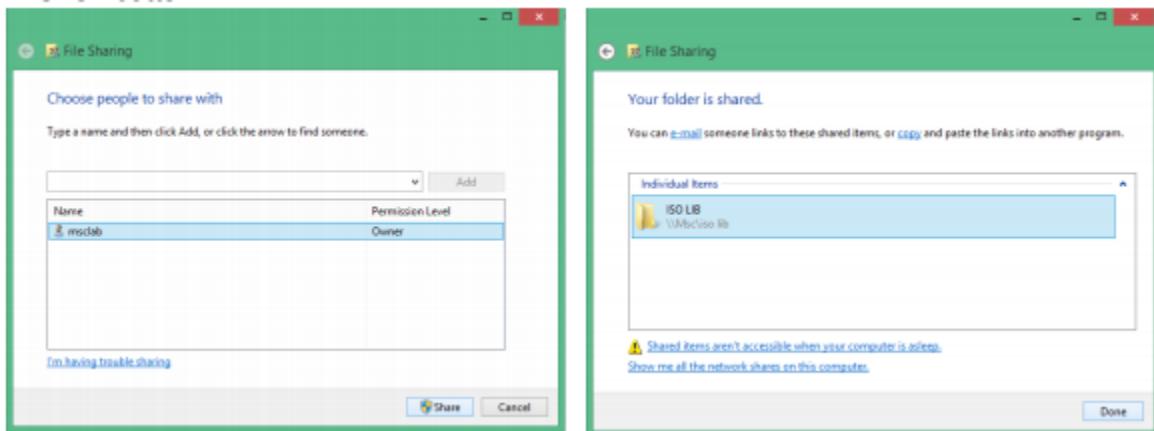
Enter the share name. Here it is “\\Msc\\iso lib”. Click on “Finish” button. It shows that the Storage Repository (SR) is being created...



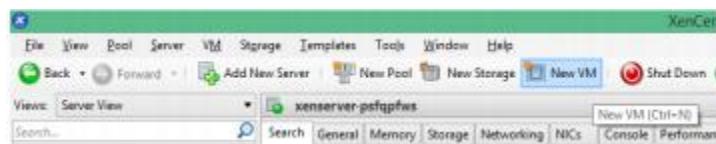
In C:\ create a new folder “ISO LIB”. Right click on “ISO LIB” folder and select Share with->Specific people...



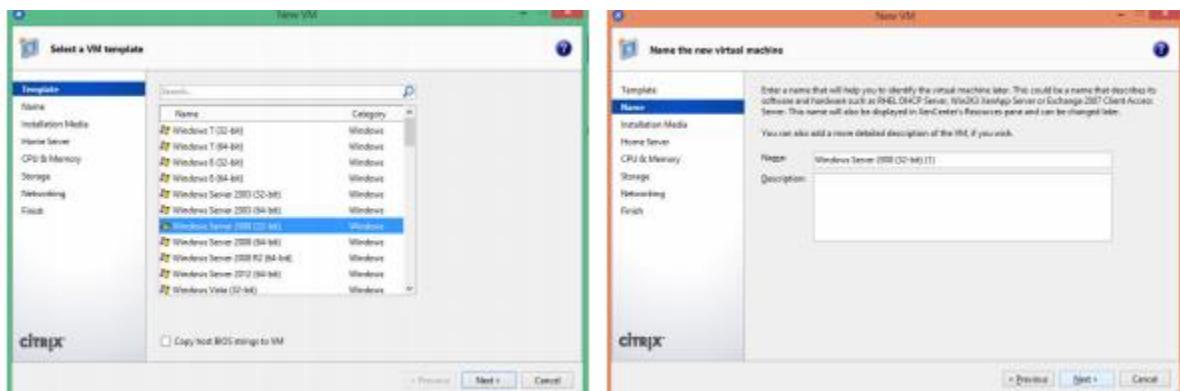
In the File Sharing window select the Host system name (here “msclab”). Click on “Share” At the final screen it shows that the ISO LIB folder is been shared with msclab. Click on “Done” button



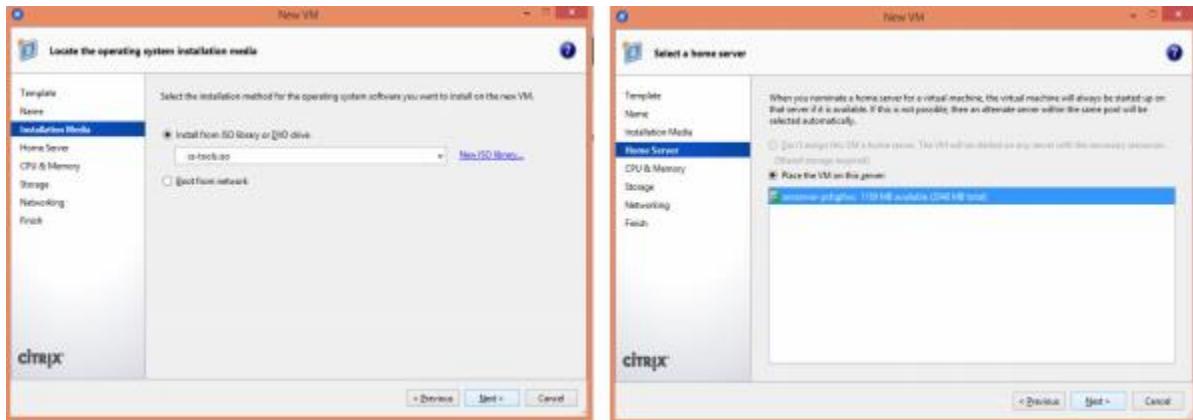
Now click on the New VM to create a new virtual machine.



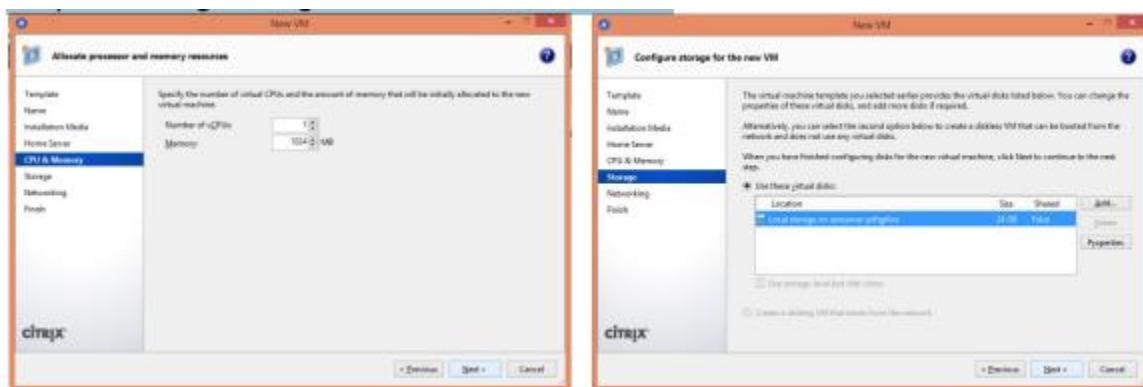
In New VM window select Template as “Windows Server 2008 (32-bit)”. Click on “Next” button. Give a name to the virtual machine or keep it default. Then click on “Next” button.



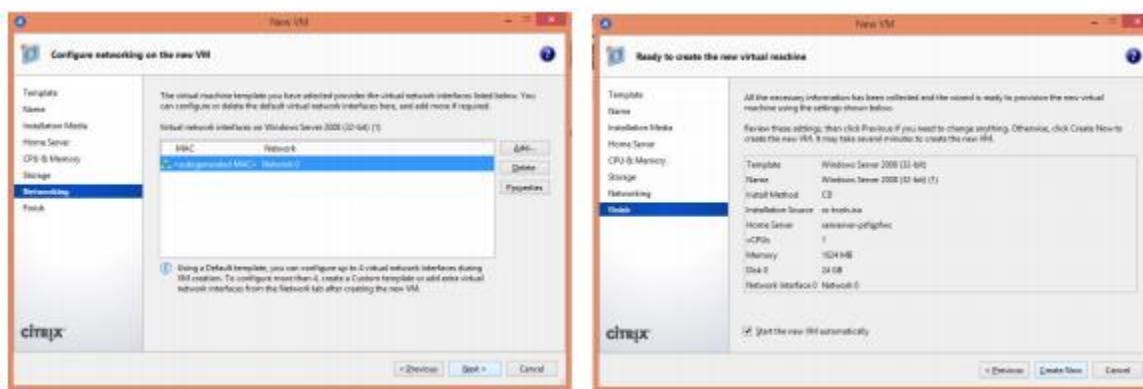
Select the “xs-tools.iso” file for installation. Click on “Next” button. In the next window select the “Place the VM on this server” and click on the “Next” button.



Leave the Memory settings default and click on the “Next” button. Keep the storage configuration default and click on “Next” button.



Leave the Networking settings default and click on “Next” button. Click on “Create Now” button.

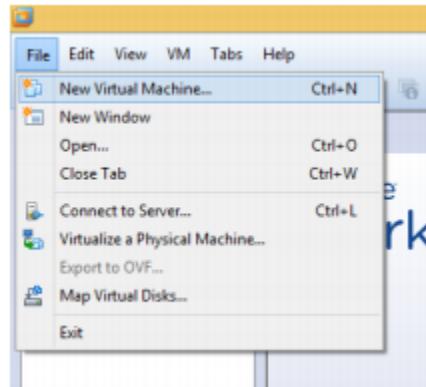


The created virtual machine can be seen in the XenCenter.

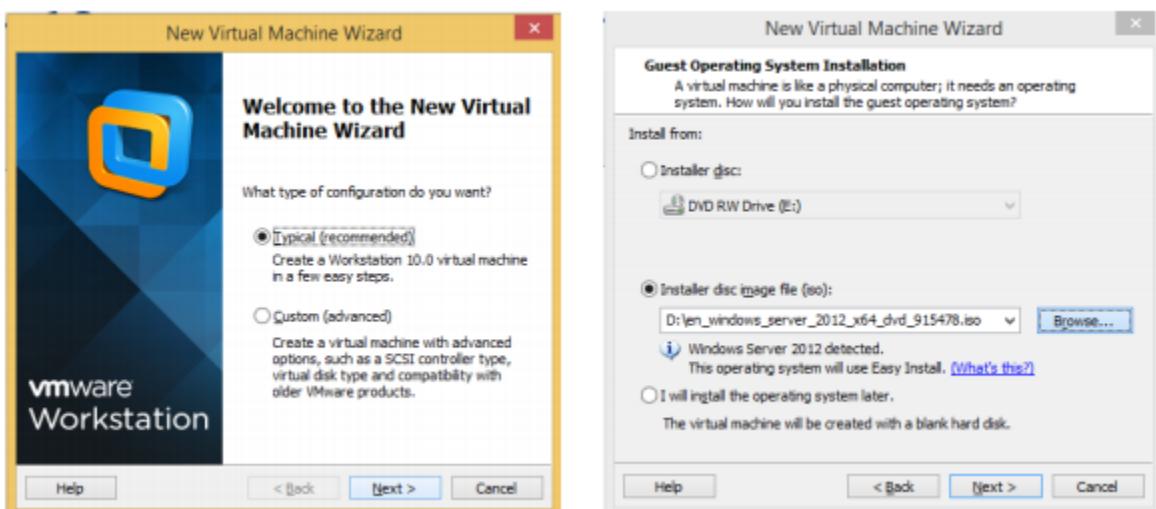


Practical 7

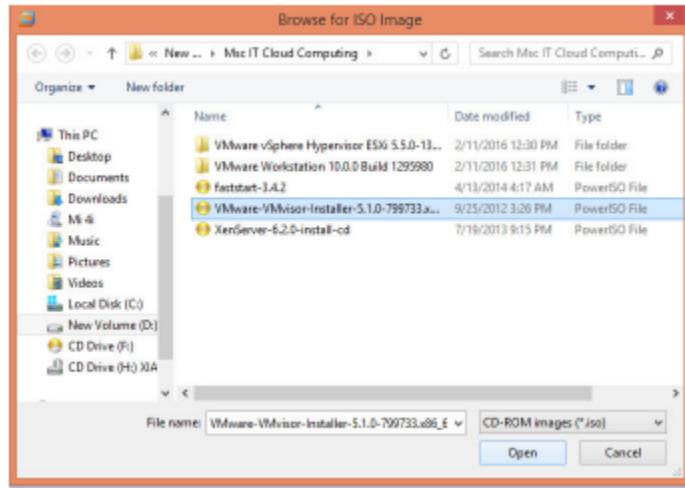
IMPLEMENT ESXi SERVER Create a new virtual machine in VMWare Workstation. File>New Virtual Machine...



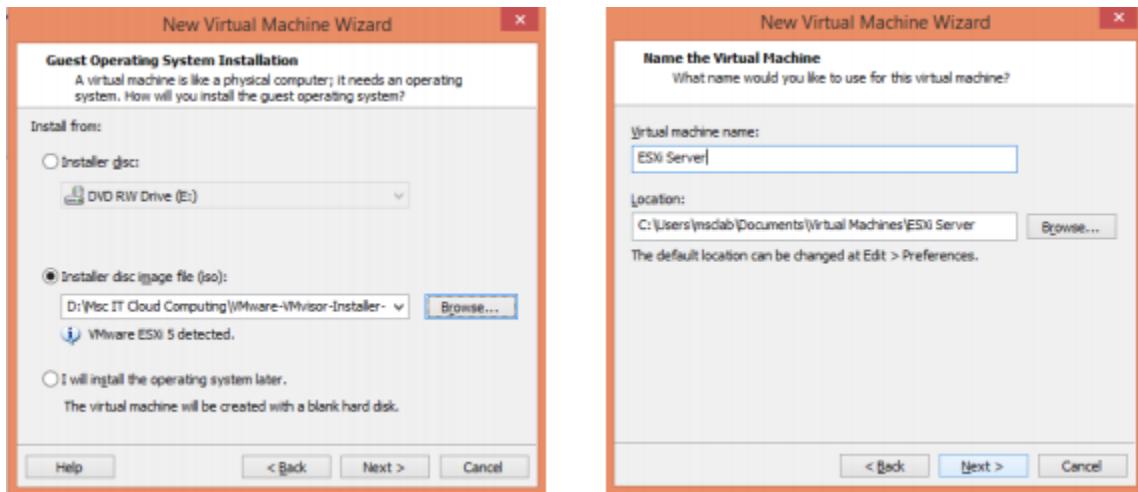
In the New Virtual Machine Wizard select the “Typical” option and click on “Next”. In the next screen click on “Browse...” to select the iso file for the VM.



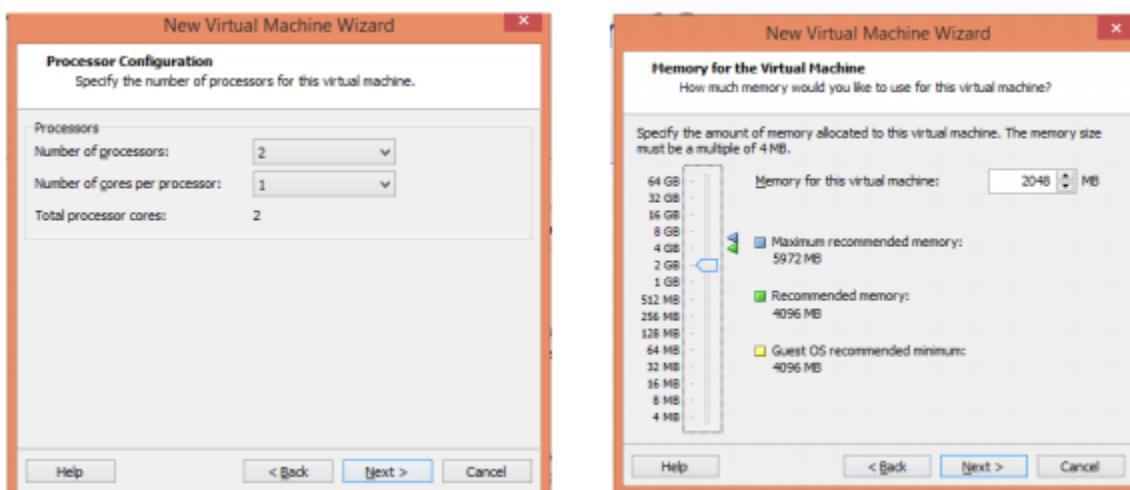
Select the “VMware-VMvisor-installer-5.1.0.....iso” file. Click on “Open”.



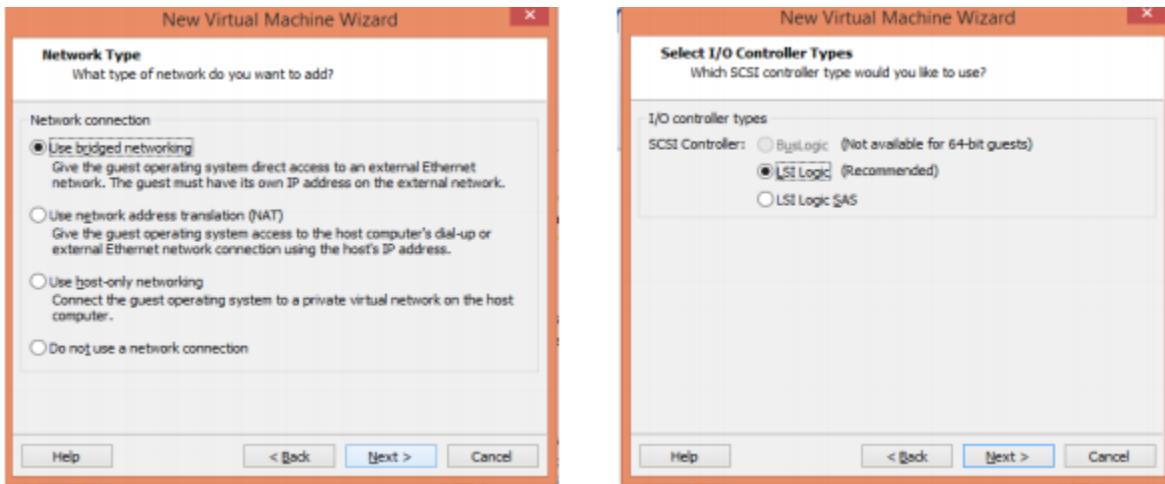
Click on “Next” button. In the next screen name the VM as “ESXi Server”. Click on “Next” button.



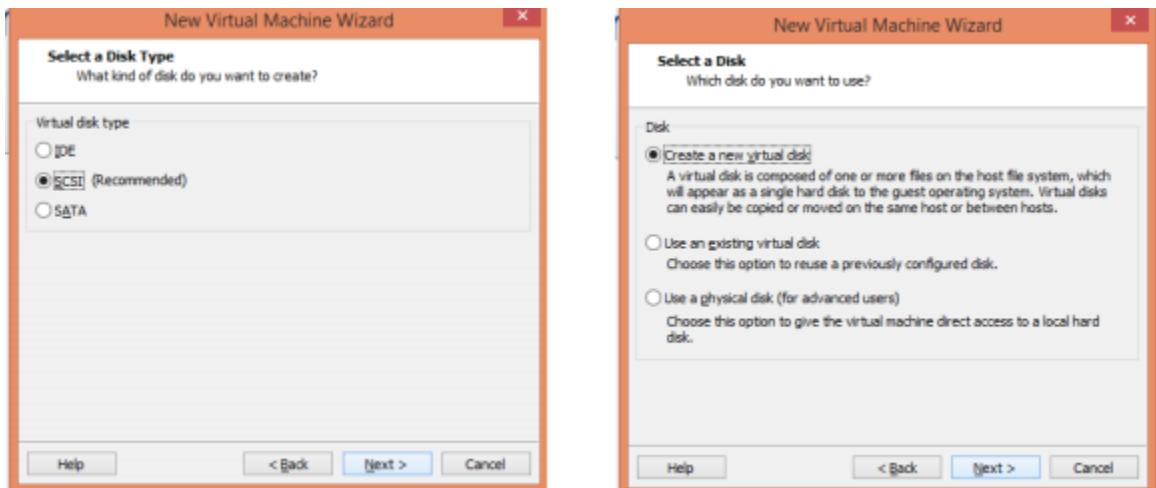
Keep the Processor Configuration default and click on “Next”. In the next screen select 2GB memory and click on “Next” button



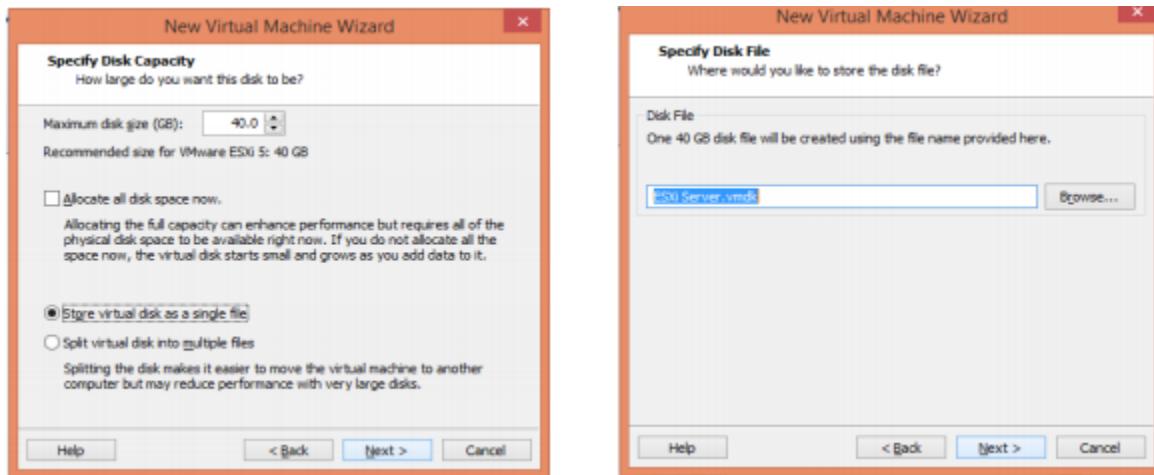
Select Network Type as “Use bridged networking” and click on “Next”. In the next screen select the Recommended I/O Controller “LSI Logic”. Click on the “Next” button.



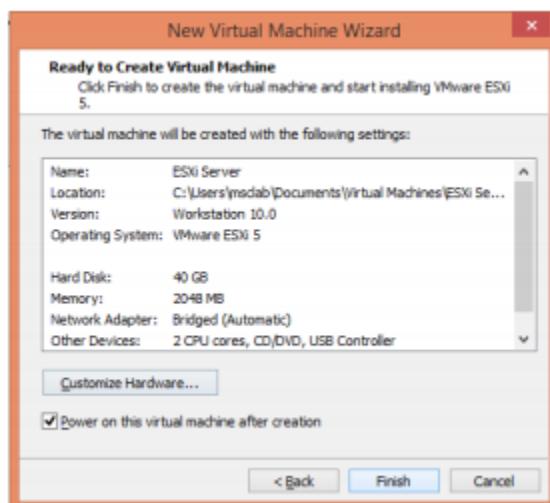
Select the Disk Type as “SCSI (Recommended)” and click on “Next”. In the next screen select “Create a new virtual disk” option, then click on “Next” button.



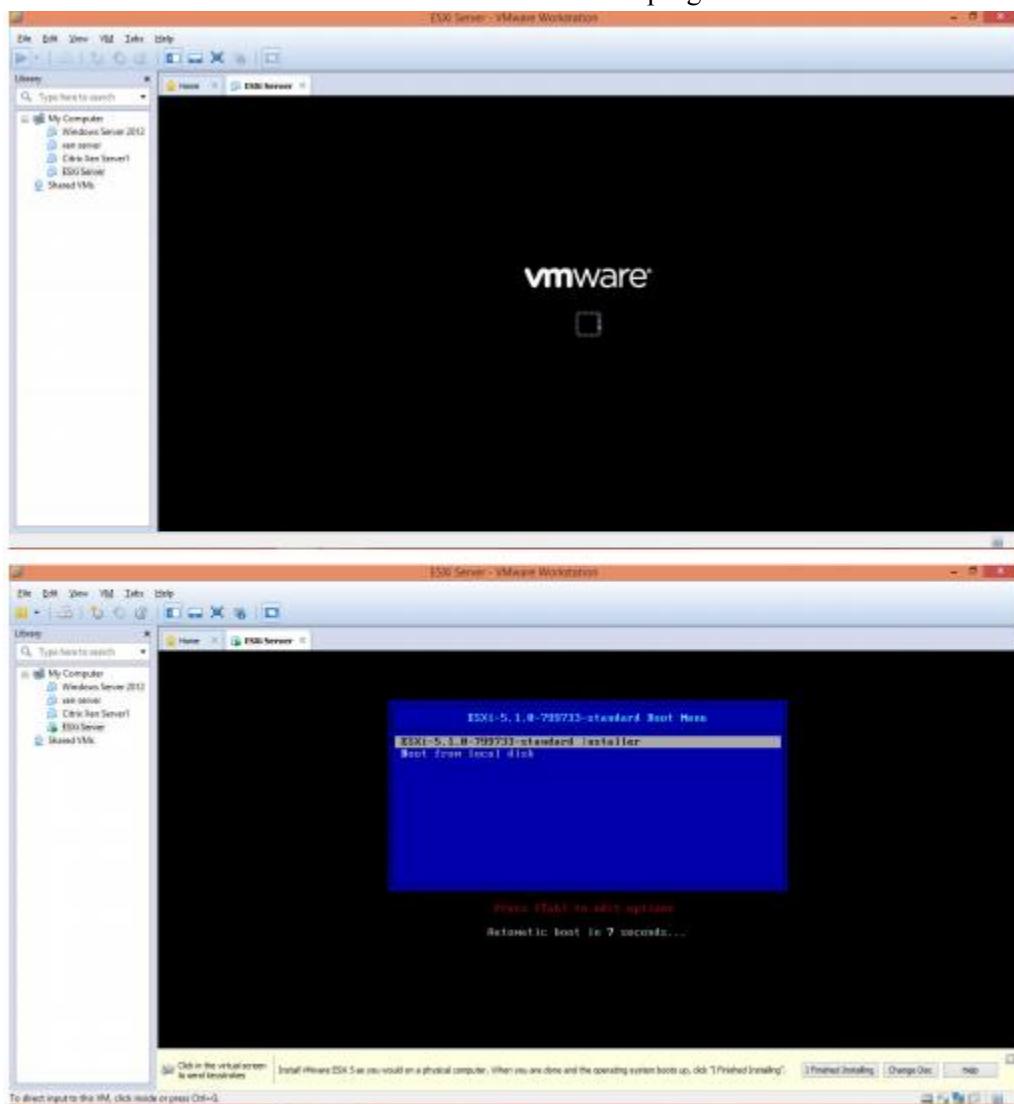
Select Disk Capacity as “Store virtual disk as a single file”. Click on “Next”. In next window leave the Disk file name default and click on “Next”

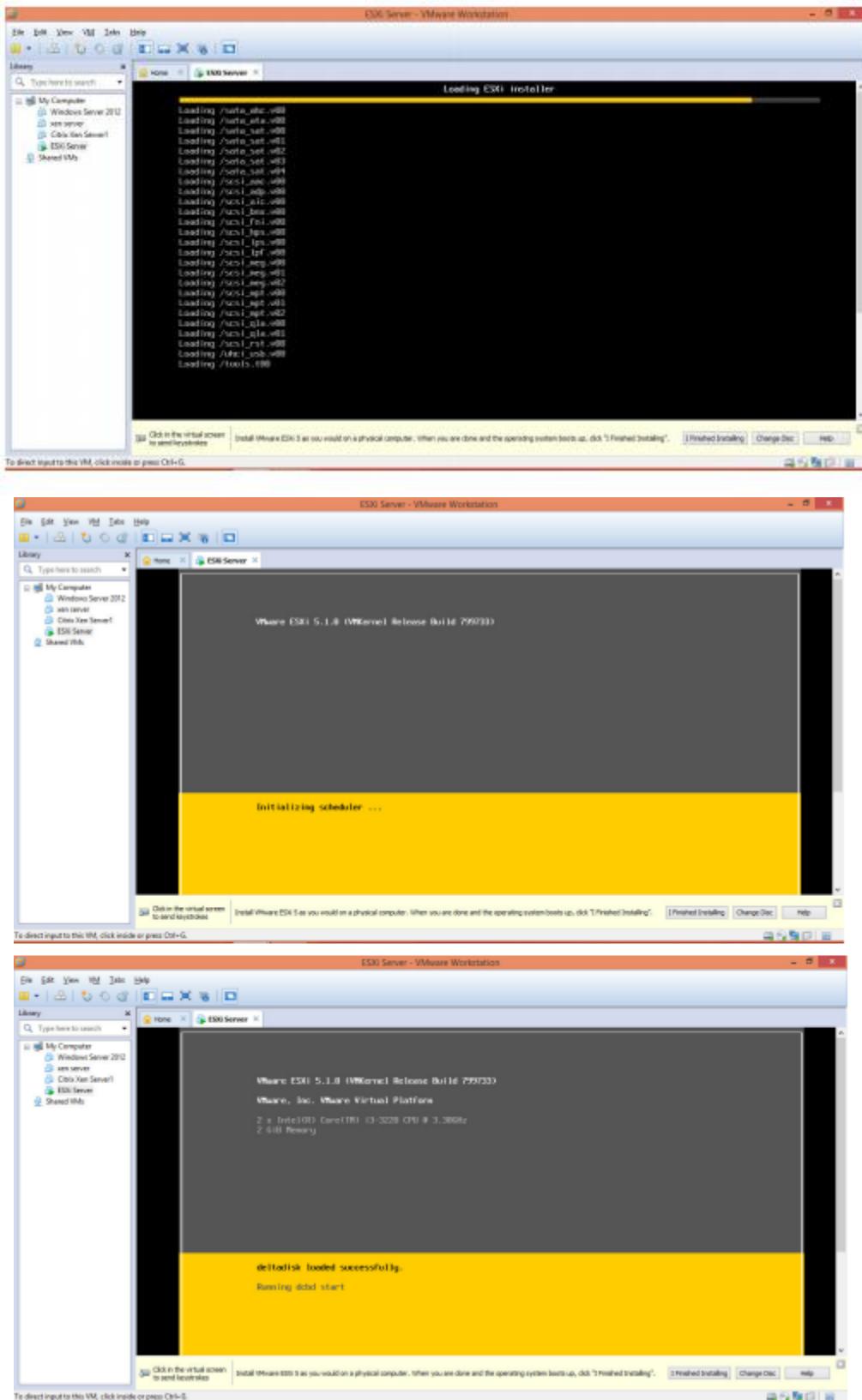


At the final window click on “Finish” button

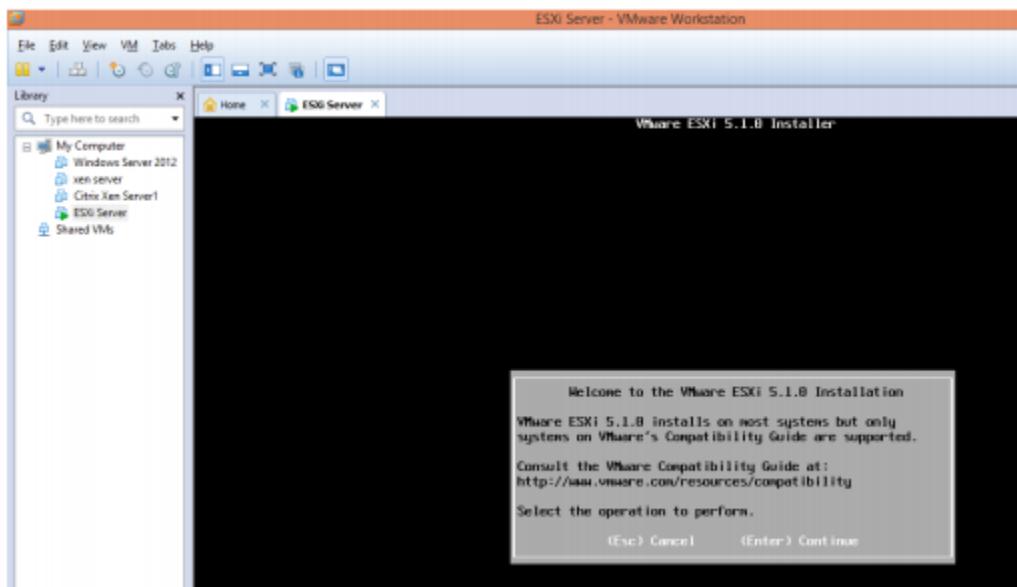


Creation of Virtual Machine is in progress...

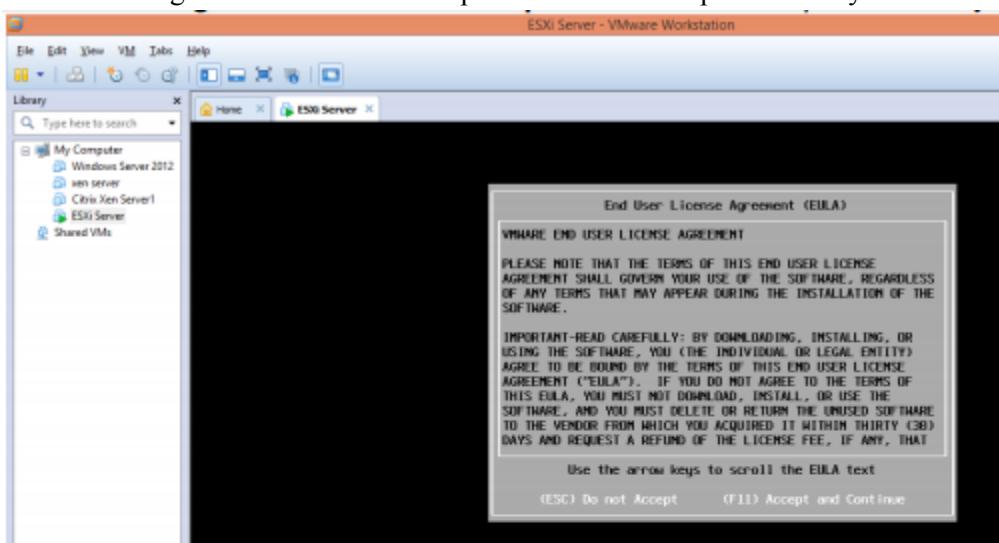




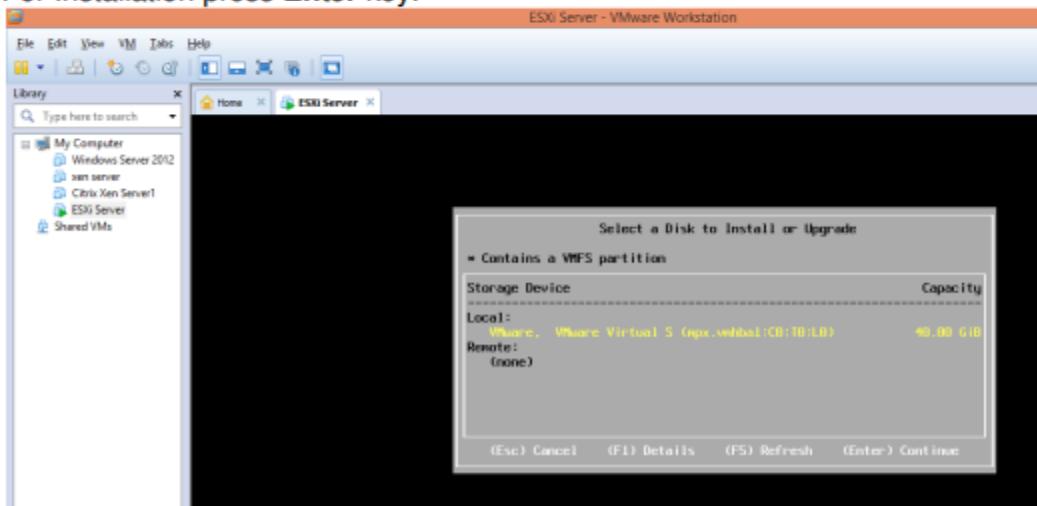
In the following screen click on Continue or press Enter key.



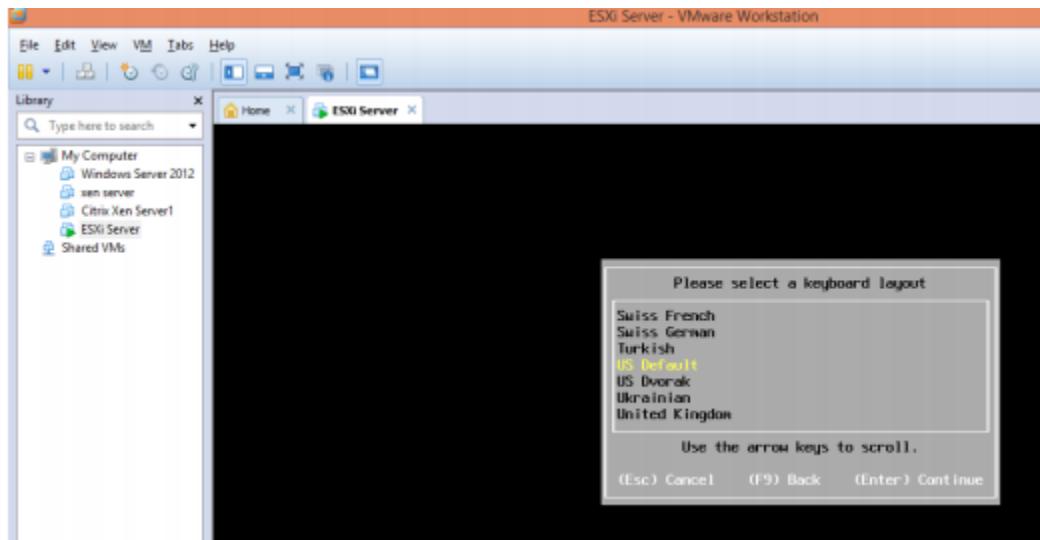
For License Agreement click on “Accept and Continue” or press F11 key.



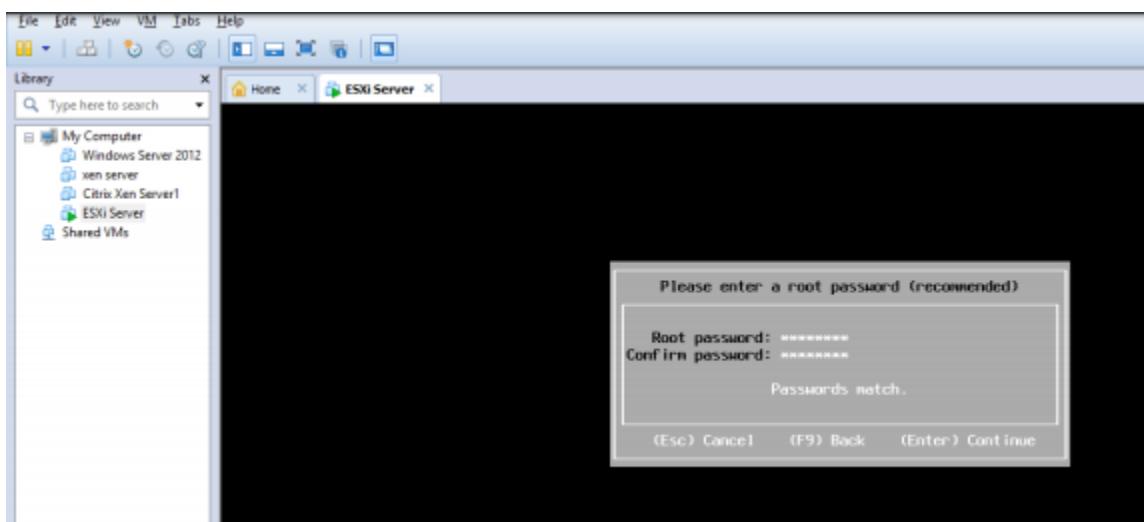
For Installation press Enter key.



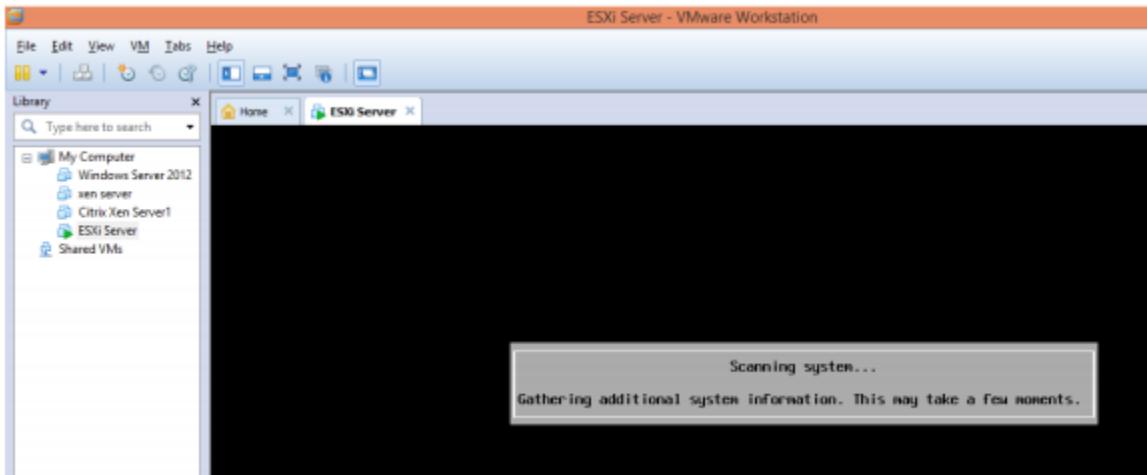
Keep the default settings for keyboard settings and press Enter key.



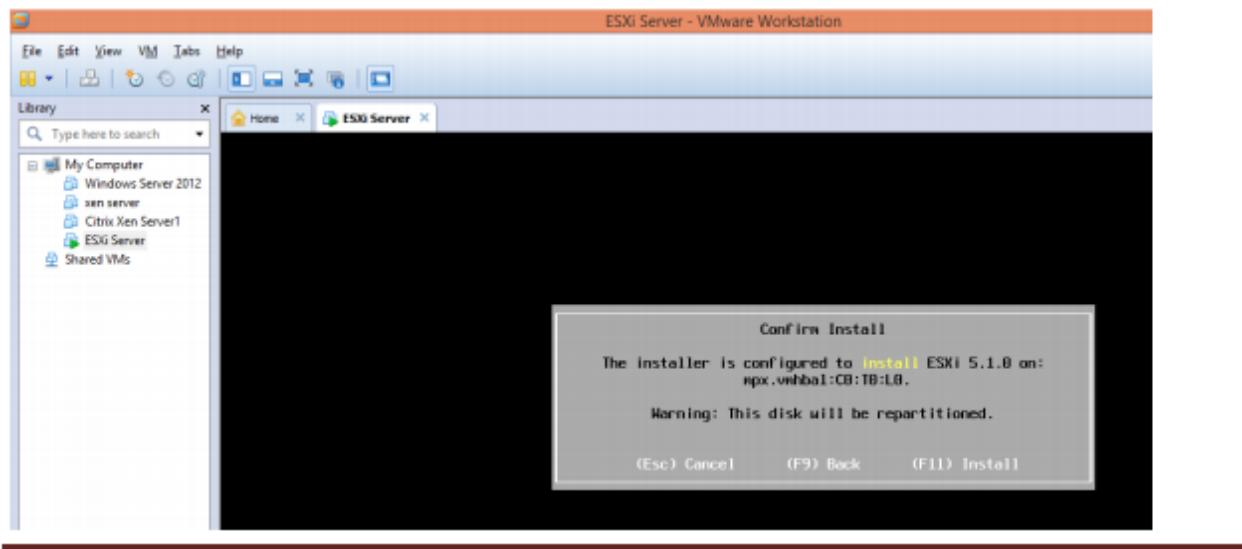
Enter the root and confirmation password and click on Continue or press Enter key



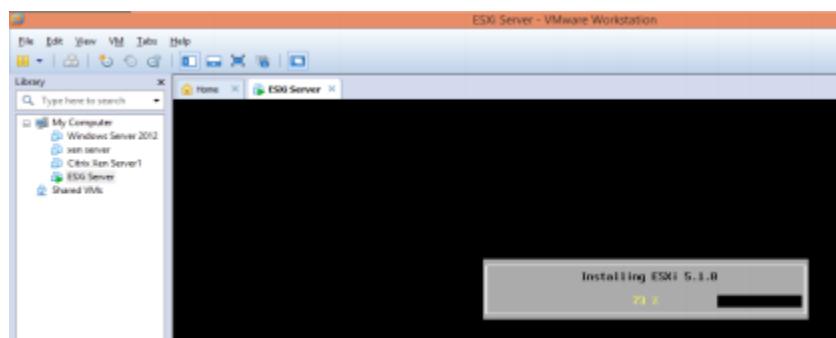
The Installation is in progress...



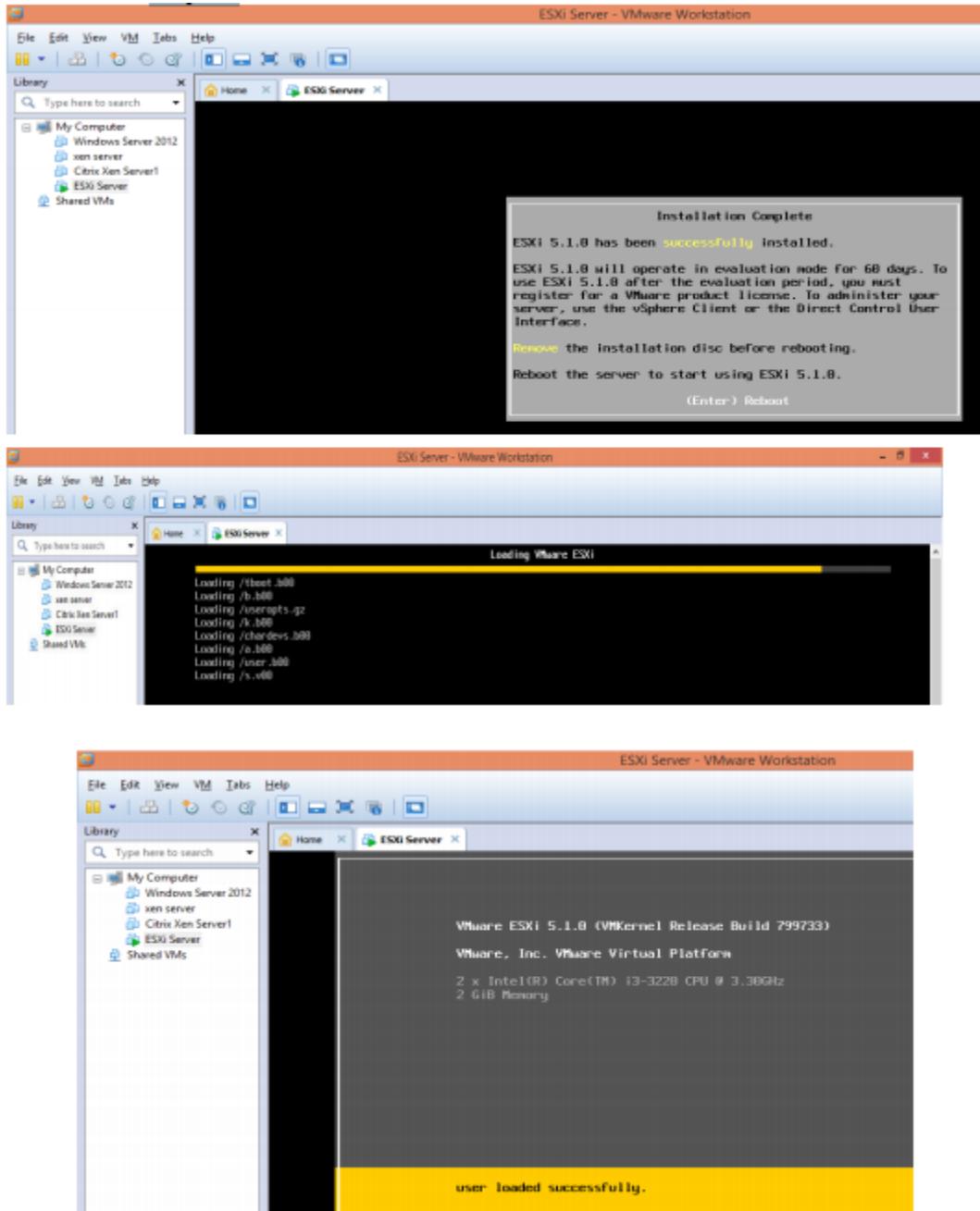
Press F11 key to Install



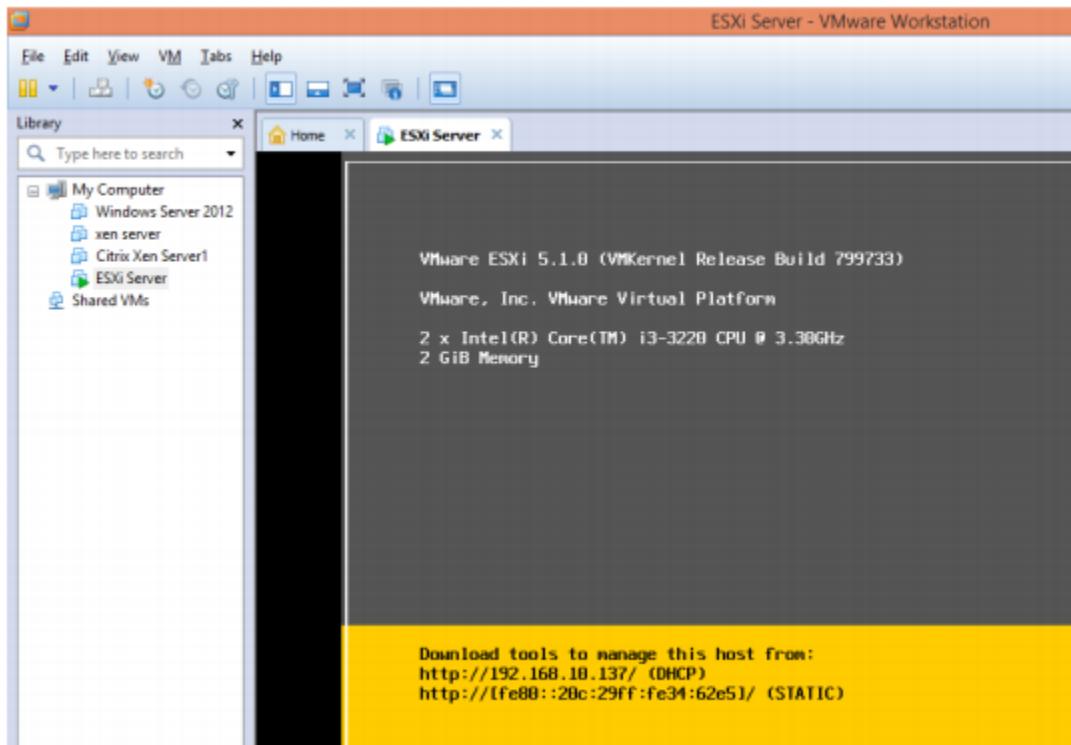
Installing ESXi 5.1.0...



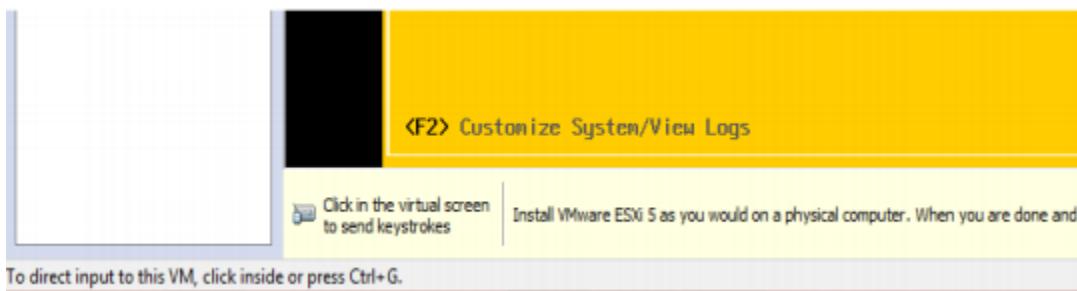
Press Enter key to Reboot



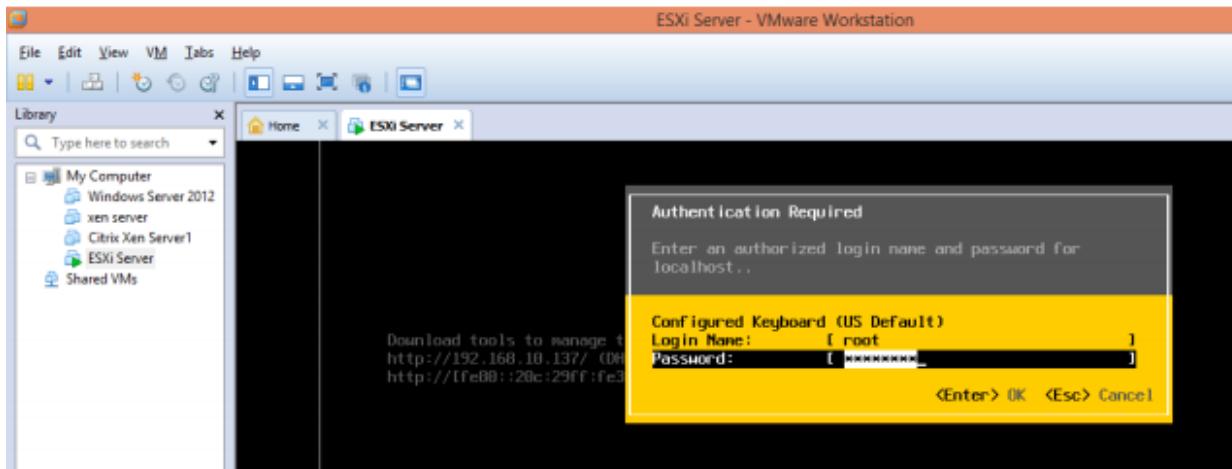
Note the DHCP IP Address. Here it is 192.168.10.137



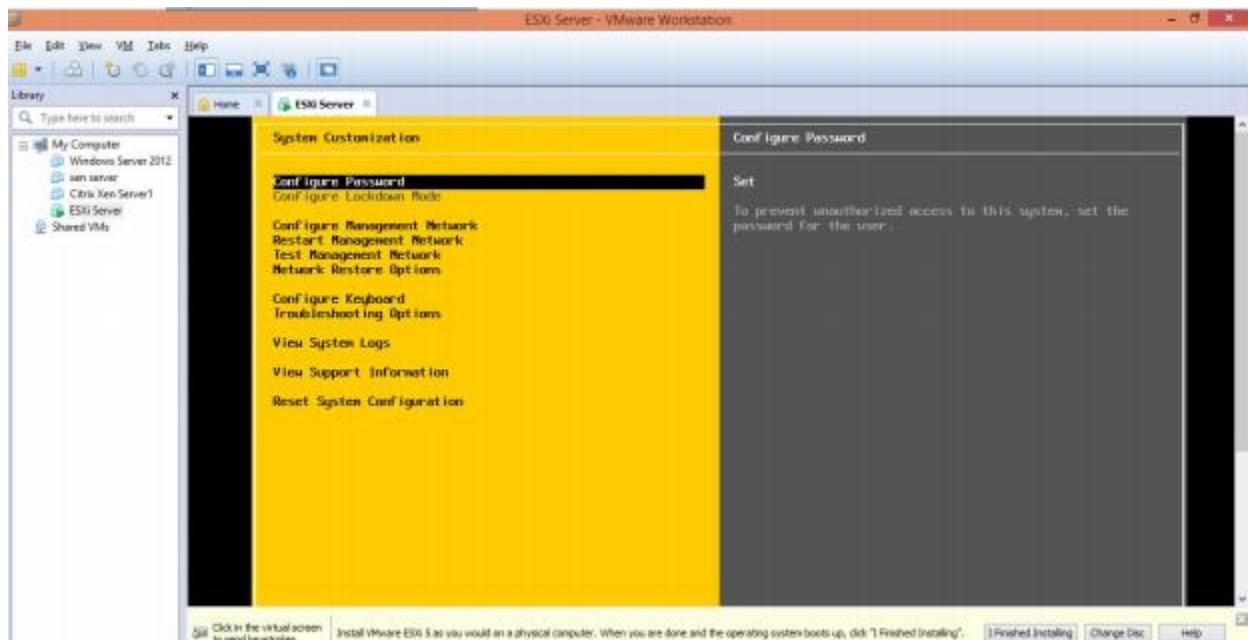
Press the F2 key for customizing system as it is shown at the bottom of the VM.



Enter the username as root and the root password (which was used earlier).



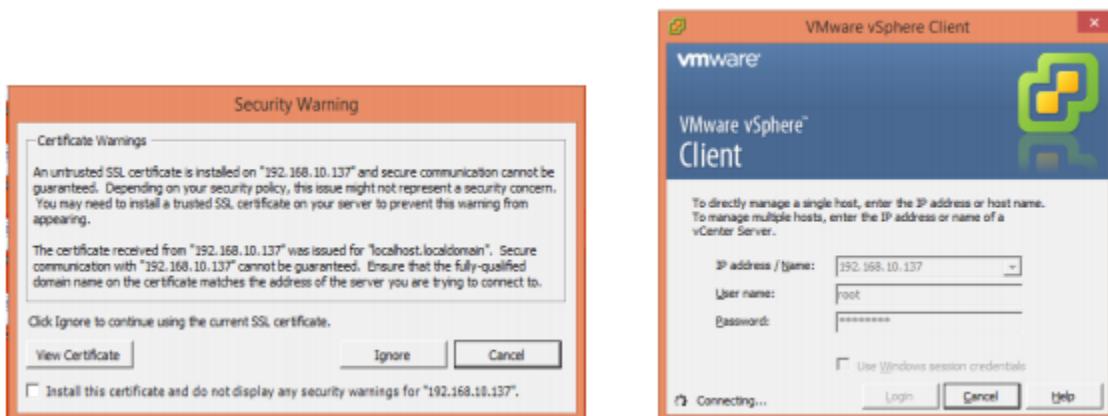
It shows the System Customization screen.



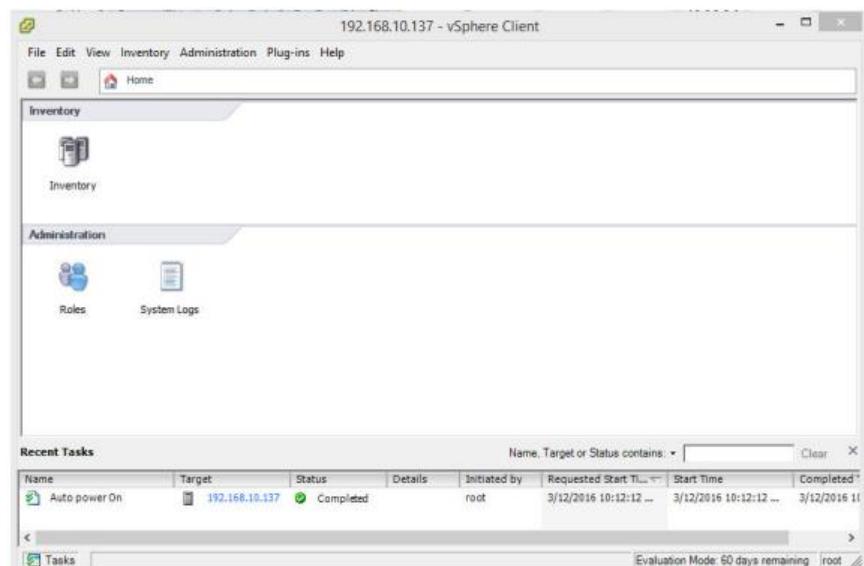
Now start the VMware vSphere Client. Enter the IP address (DHCP IP address of ESXi Server), user name as root and the same password as the ESXi System. Click on “Login” button.



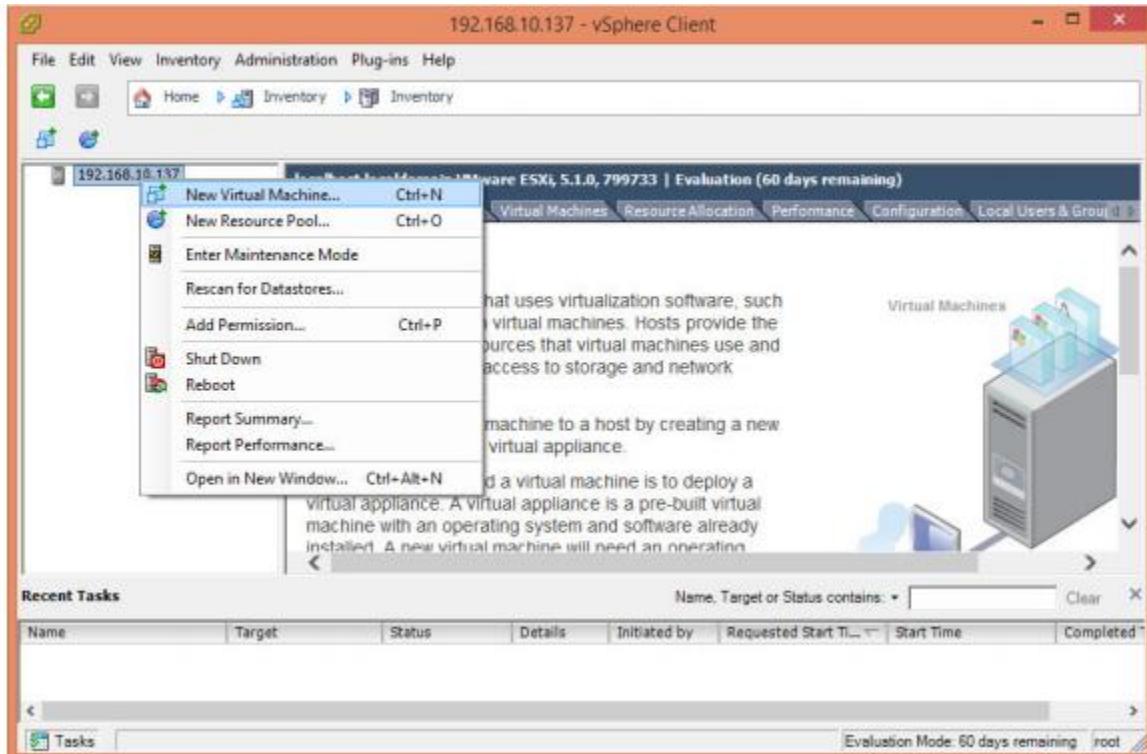
Click on the Ignore for the Security Warning



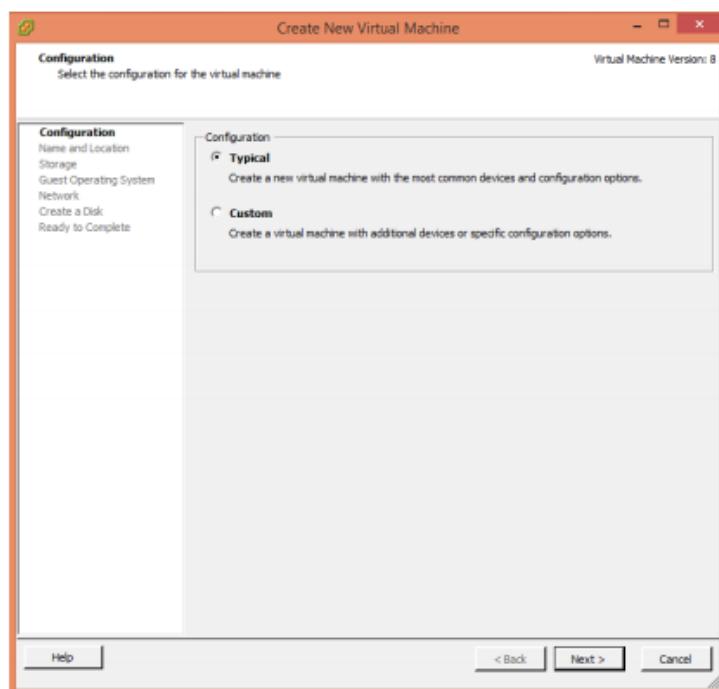
After login the VMware vSphere Client looks like the following image. At the bottom of this screen the connectivity of VMware vSphere Client where the target IP address is the ESXi Server's DHCP IP address. Click on the Inventory.



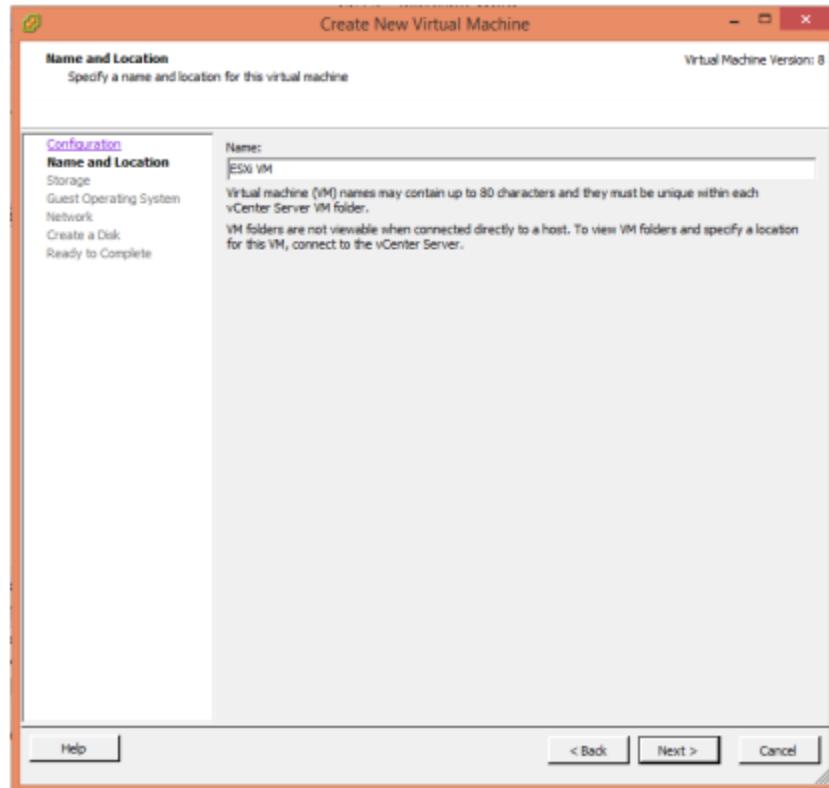
It shows the IP Address (192.168.10.137) listed on the left pane. Right click on the IP address and select the option “New Virtual Machine...”



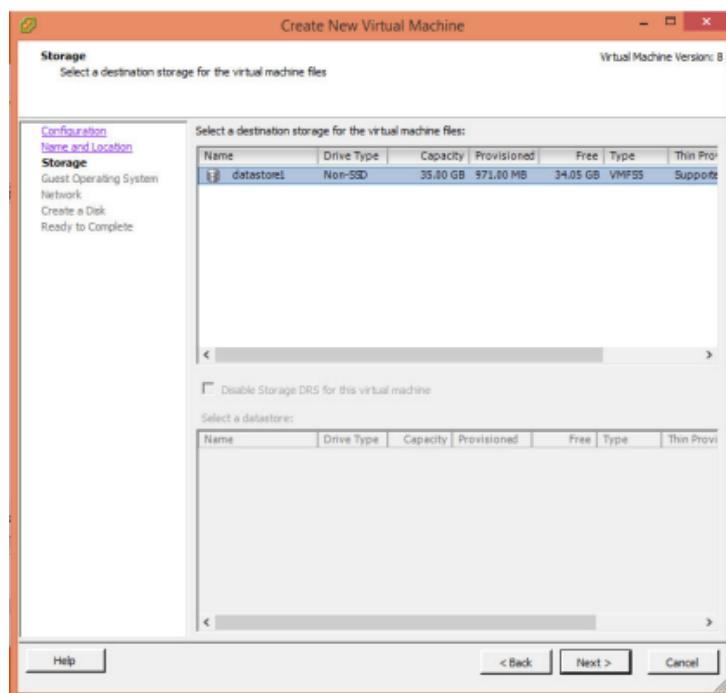
In the Create New Virtual Machine window select the Typical option and click on “Next”.



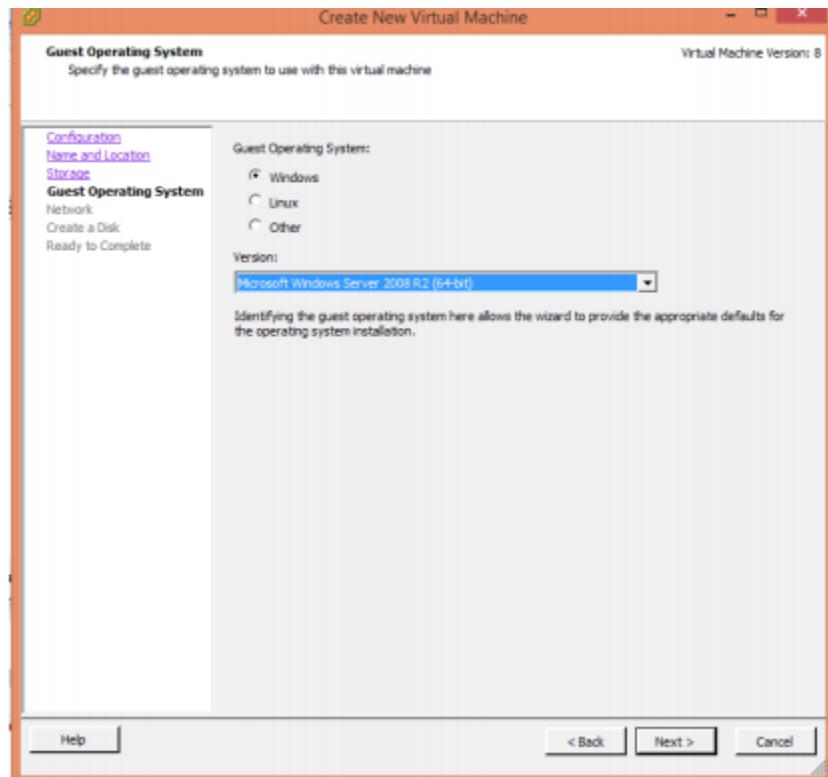
Give a name to the Virtual Machine. Here it is given as ESXi VM. Click on the “Next” button



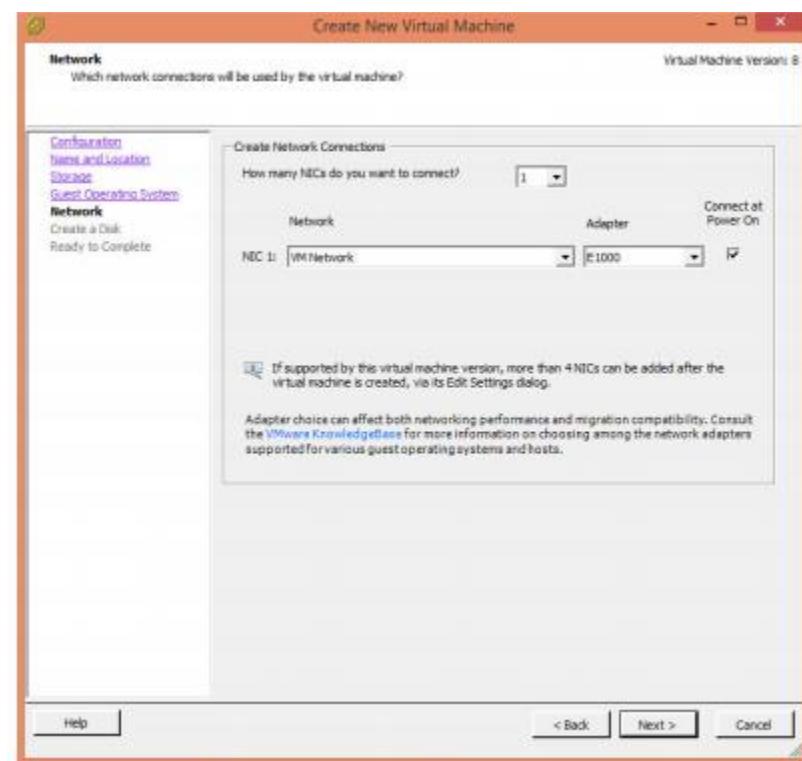
In the next screen keep the settings default for Storage and click on “Next”.



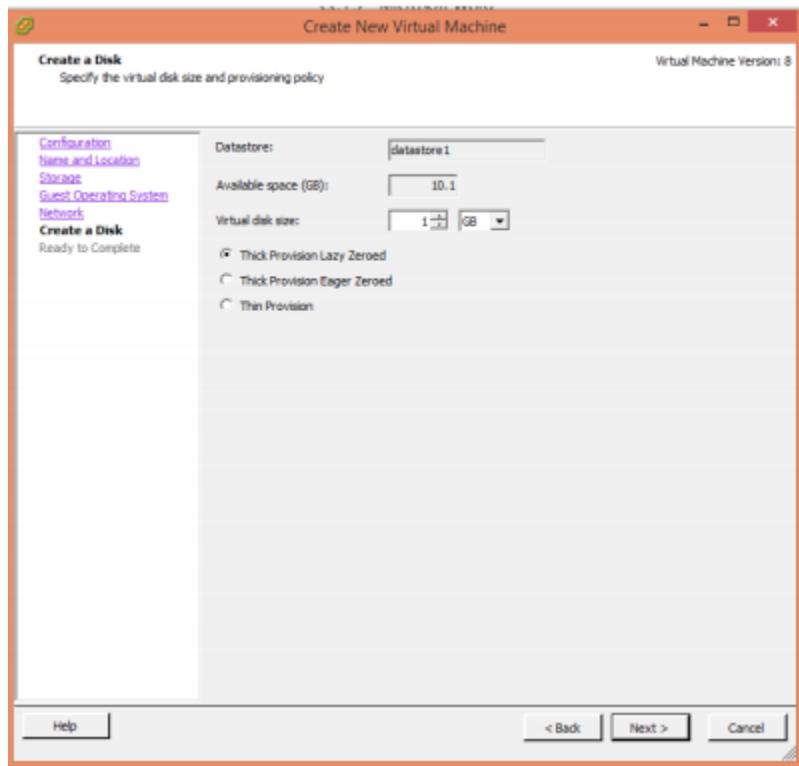
Select the Guest OS as Windows and Version as “Microsoft Windows Server 2008 R2 (32-bit)



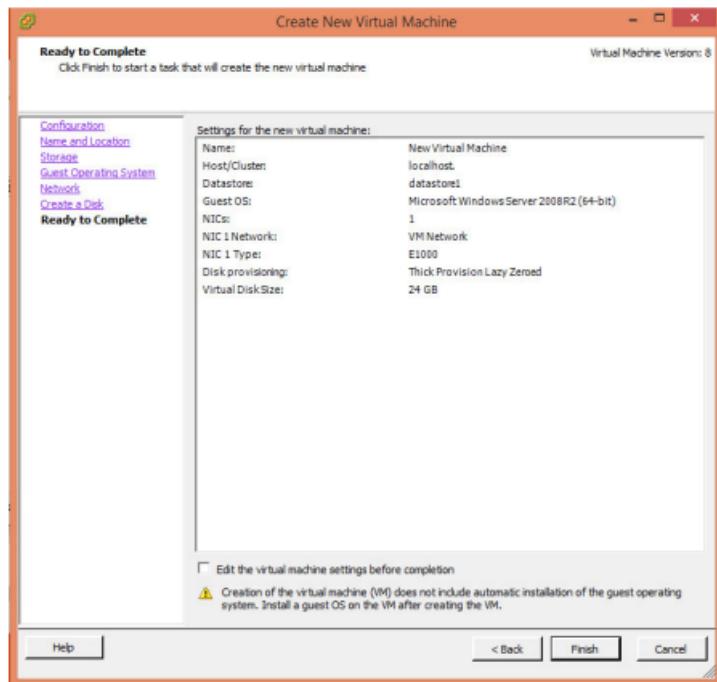
Leave the Network settings default and click on “Next”



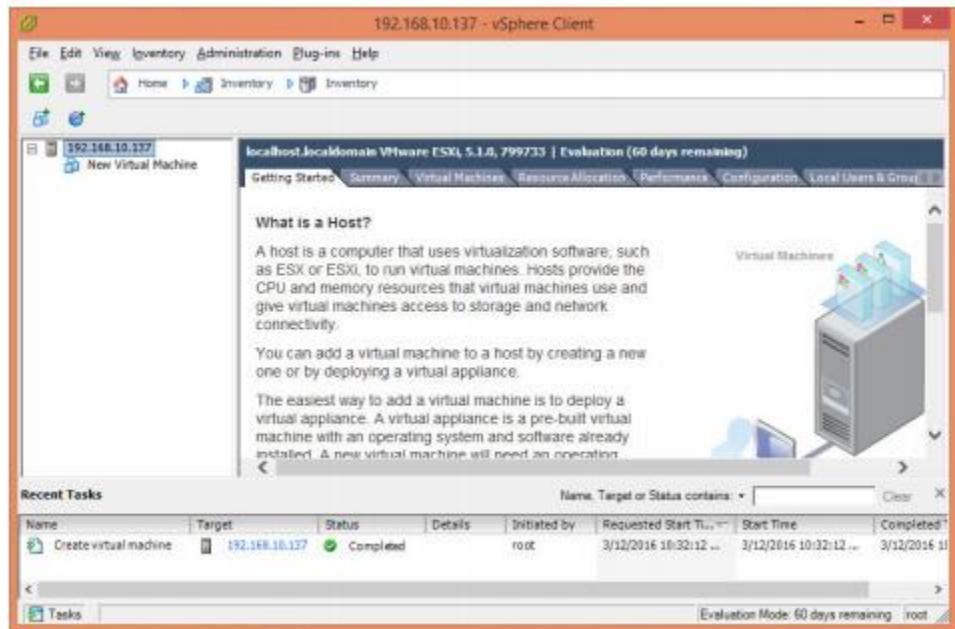
Choose the Disk space (depending on the available space of user's system. Here it is 1GB) of the VM in GB. Click on the "Next" button



At the final screen click on "Finish" button.

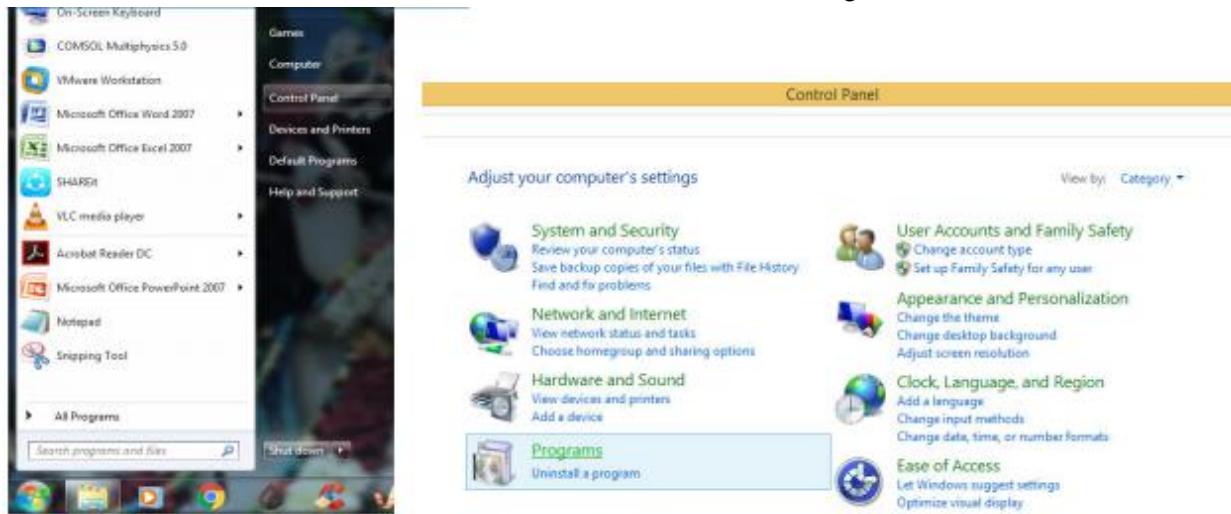


The new created Virtual Machine is listed under the IP Address.

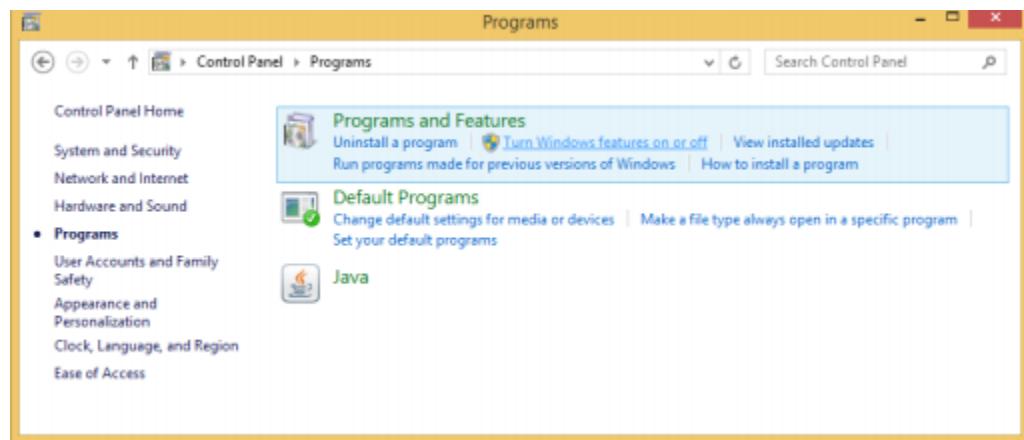


Practical 8:-

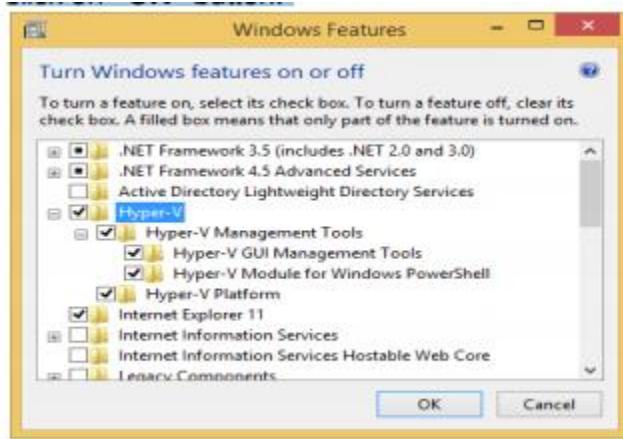
NATIVE VIRTUALIZATION USING HYPER V To turn ON the Hyper V feature of Windows go to Control Panel Start button → Control Panel. In Control Panel click on Programs.



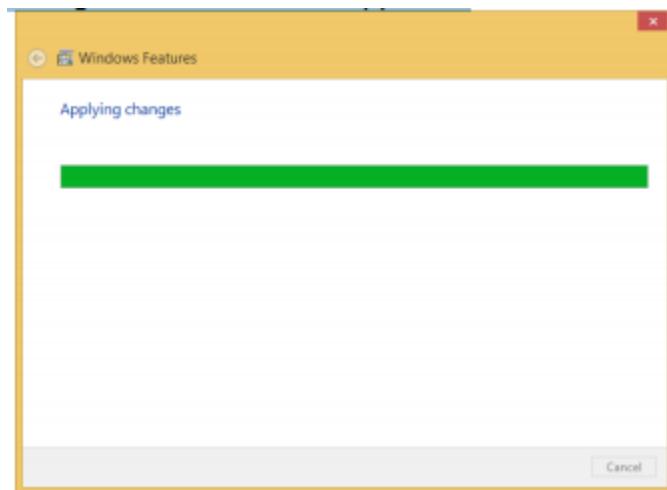
In Programs window click on Turn Windows features on or off



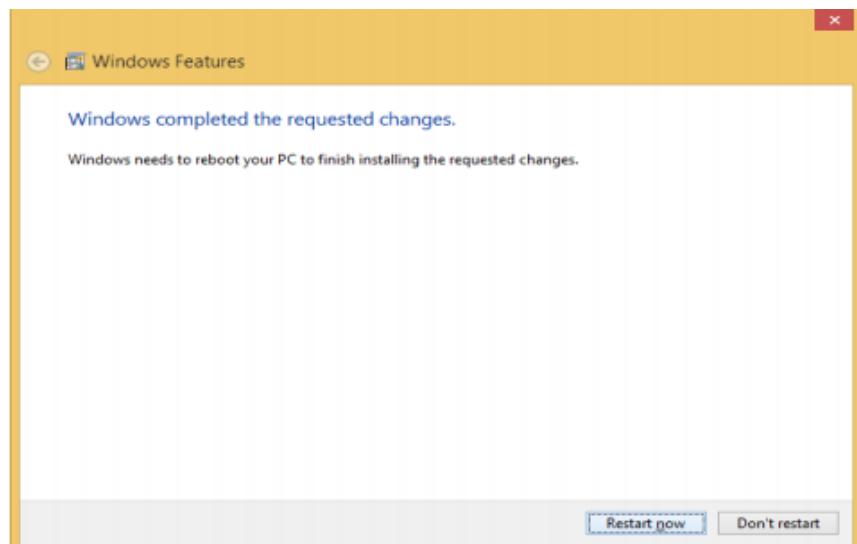
In the “Windows Features” window check (select) the Hyper V and its sub options. Then click on “OK” button.



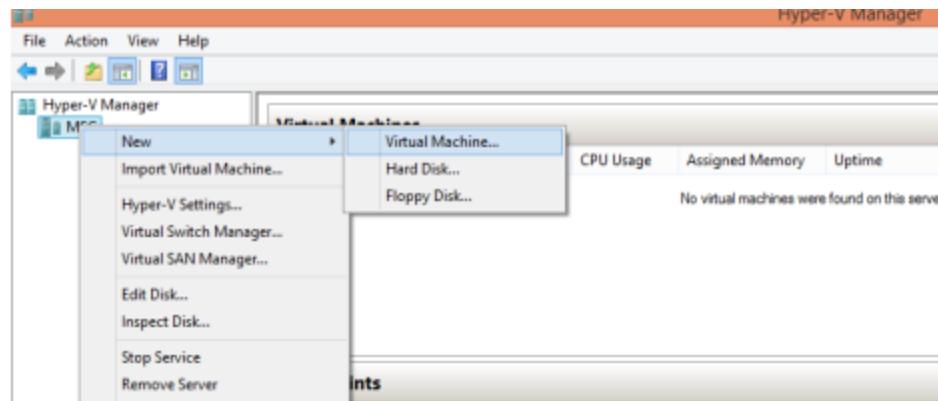
Changes on Windows are applied...



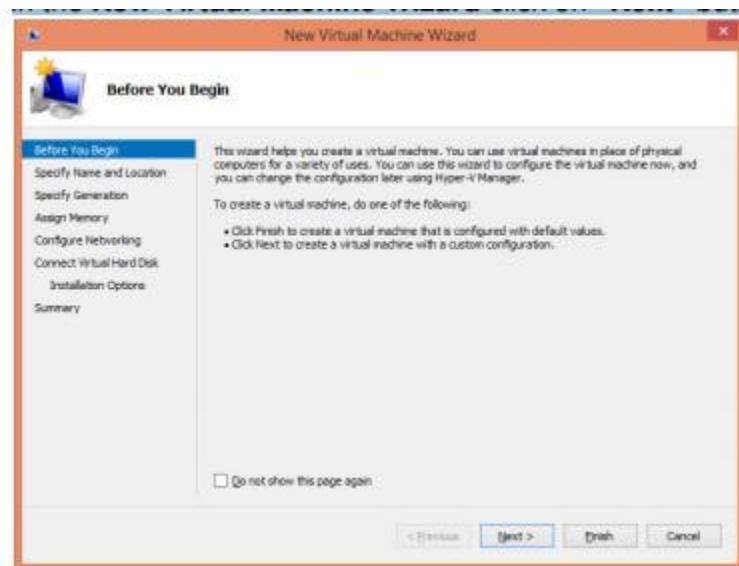
To apply the changes on the Windows OS Restart the System by click on the “Restart now” button.



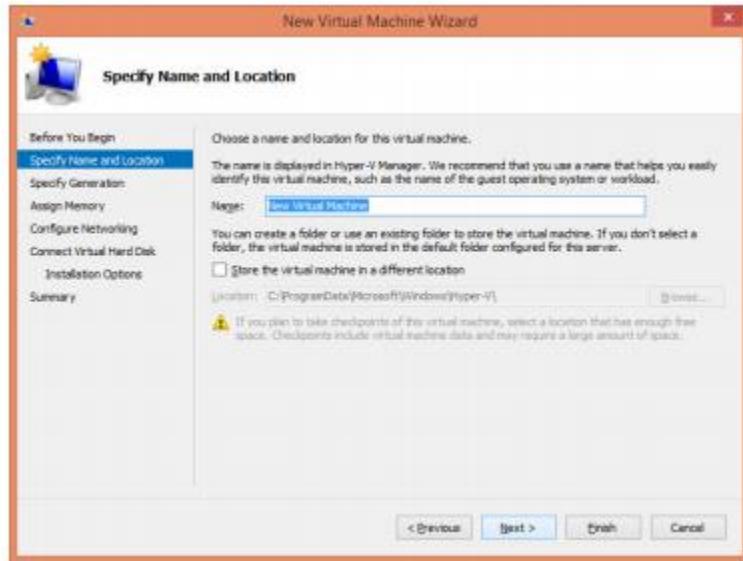
Now open Hyper-V Manager. Create a new VM under Hyper-V ManagerMSC (System User). Right click on MSC select NewVirtual Machine...



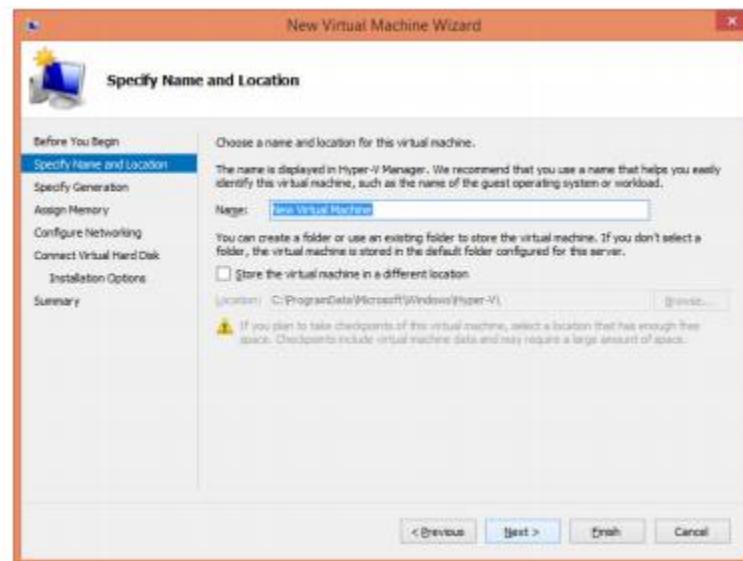
In the New Virtual Machine Wizard click on “Next” button.



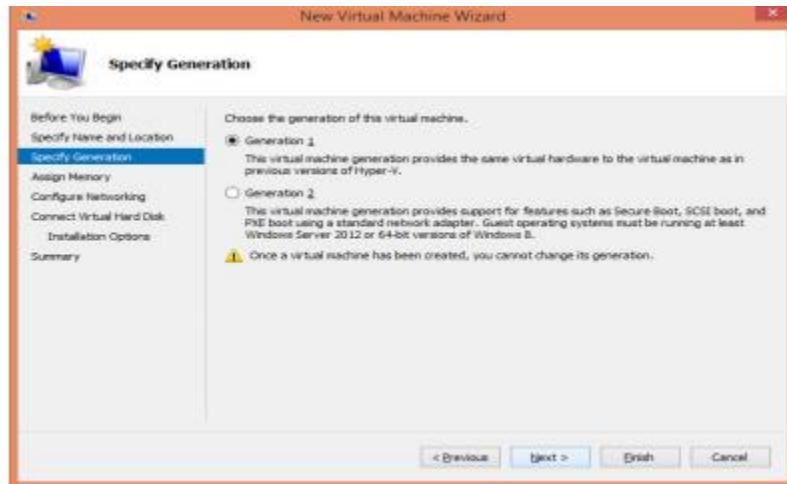
Give a name to the VM or leave it default. Click on the “Next” button



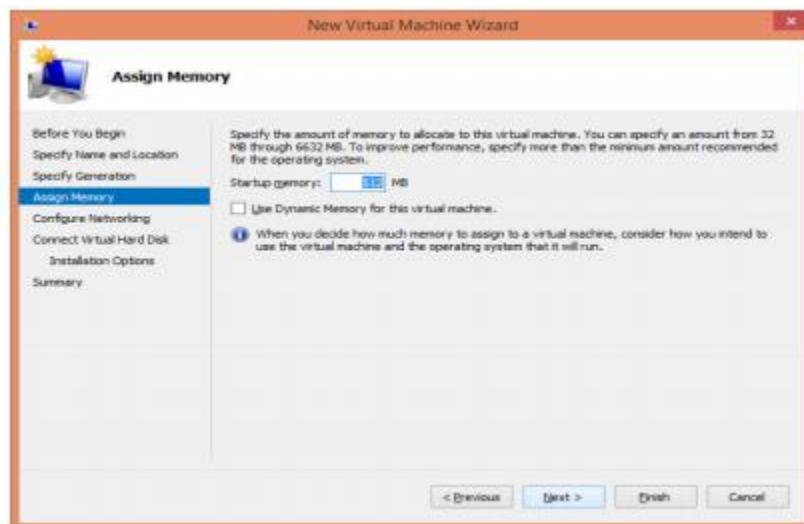
Give a name to the VM or leave it default. Click on the “Next” button.



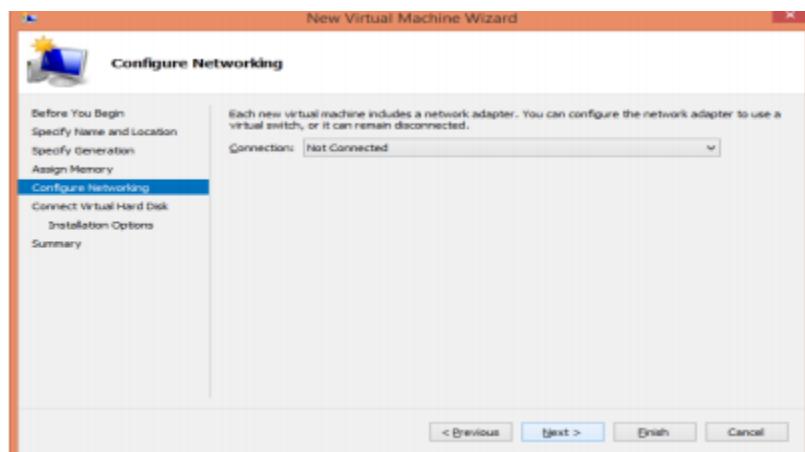
Select “Generation 1” option in the next window and click on “Next” button



In the Assign Memory window keep the default memory settings as 512 MB and click on “Next” button.

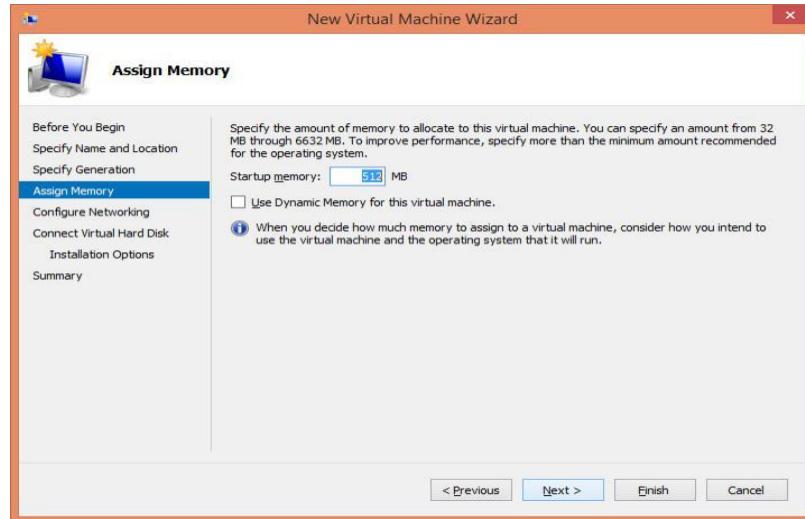


In the Configure Networking window select connection as “Not Connected”. Click on the “Next” button.

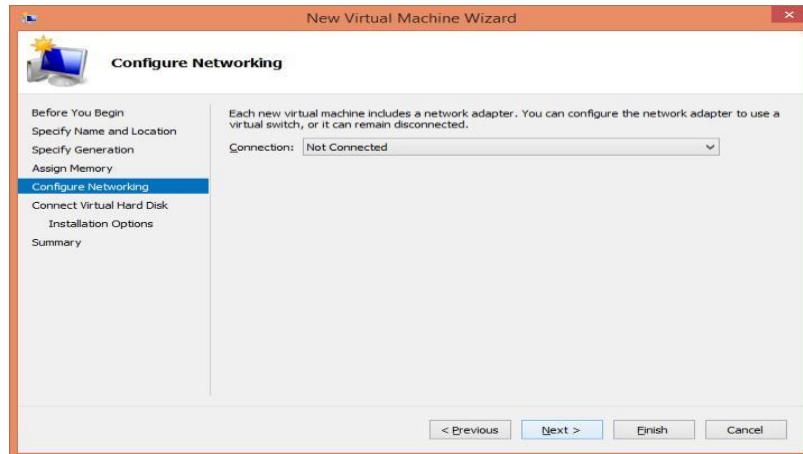


In the next window where it asks about virtual hard disk size; leave the default settings as it is and click on “Next” button

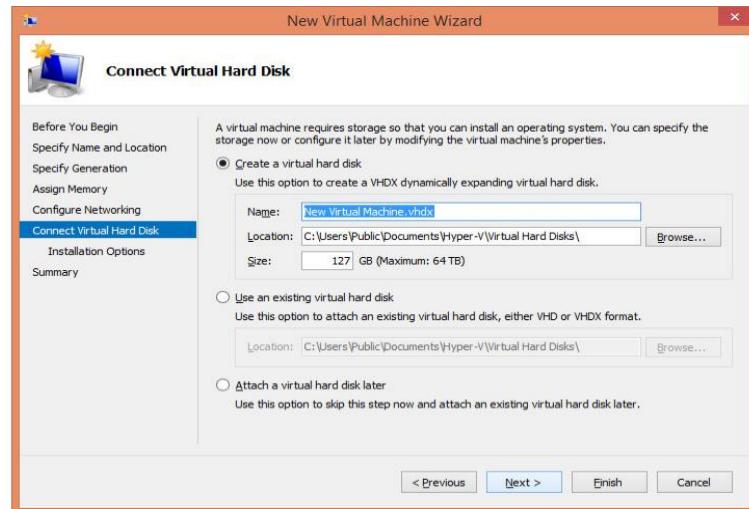
In the **Assign Memory** window keep the **default** memory settings as **512 MB** and click on “Next” button.



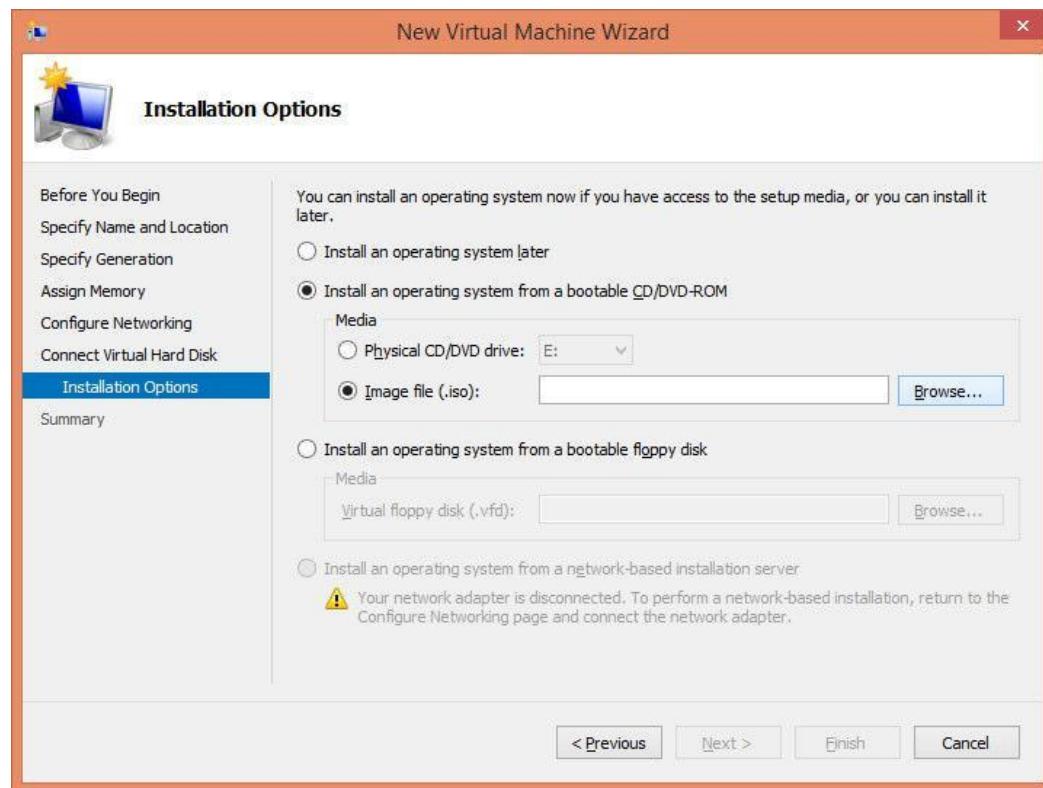
In the **Configure Networking** window select connection as “**Not Connected**”. Click on the “Next” button.



In the next window where it asks about **virtual hard disk size**; leave the default settings as it is and click on “Next” button

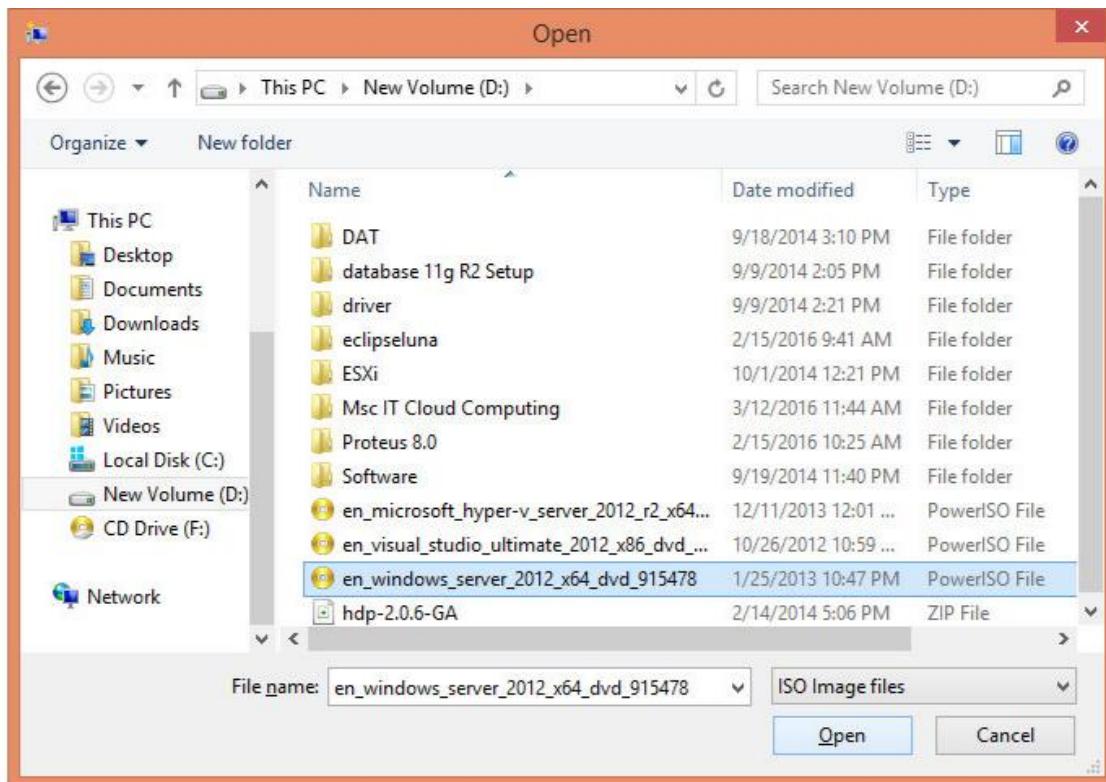


In the **Installation Options** window to select the “.iso” file for the VM click on the “Browse...” button

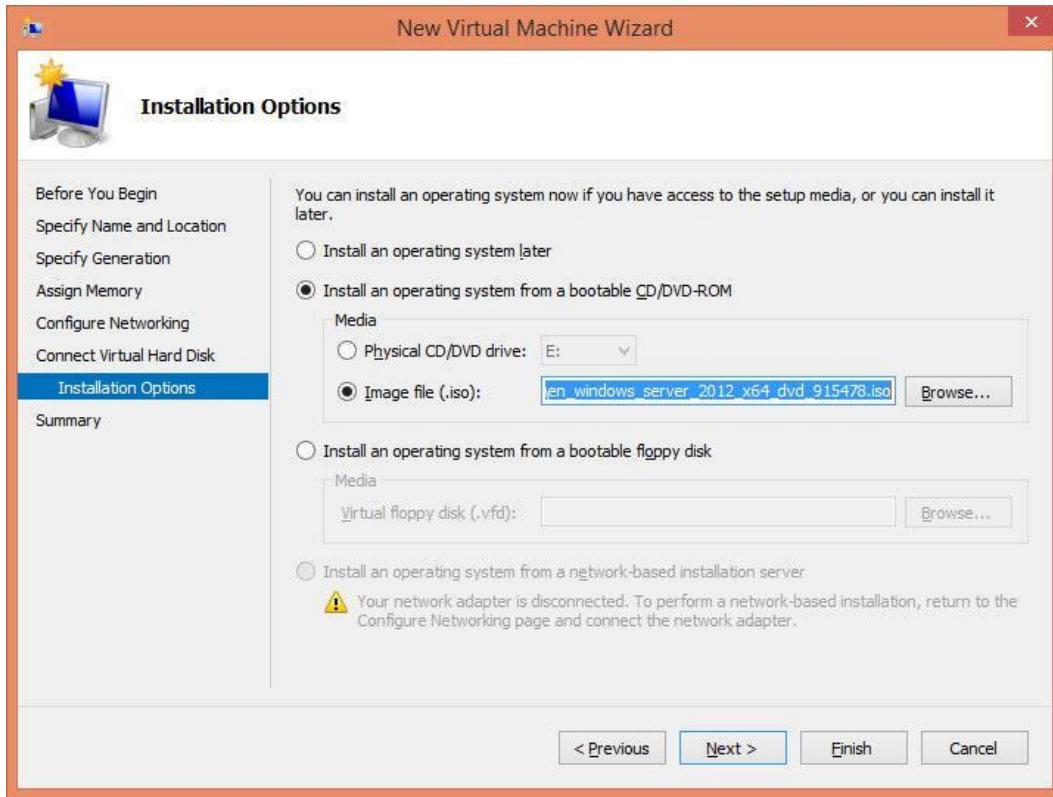


Select the **Windows Server 2012 “.iso”** (here it is,

“en_windows_server_2012_x64_dvd_915478”) file and click on “Open”.

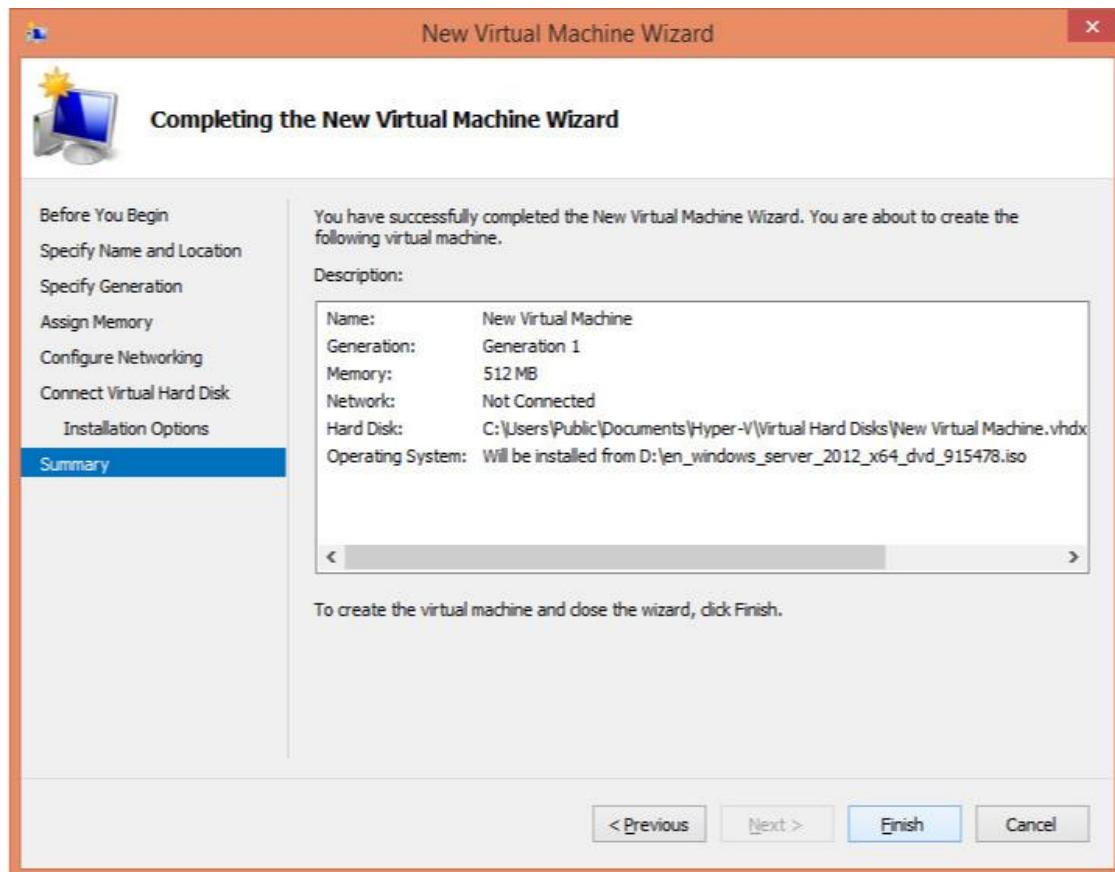


In the **Installation Options** window click on “**Next**” button.



In the final window it shows the **summary** of the selected options for the VM (to be created).

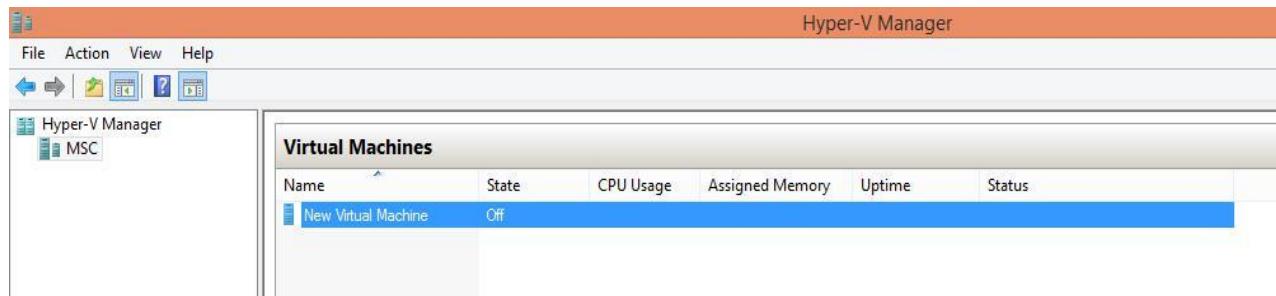
Click on “**Finish**” button.



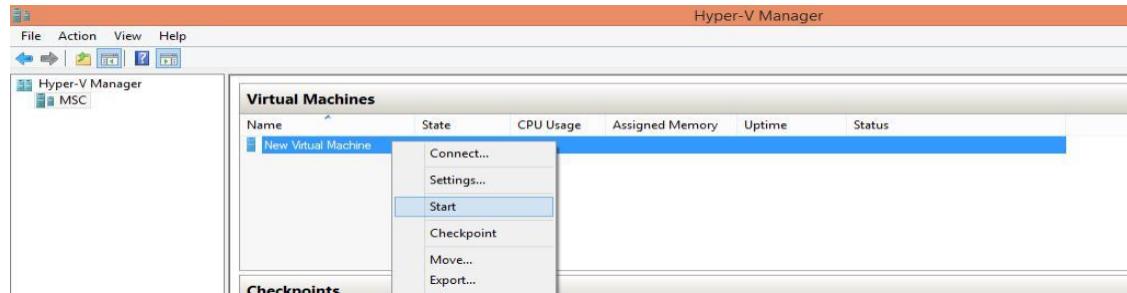
Creation of the VM is in progress...



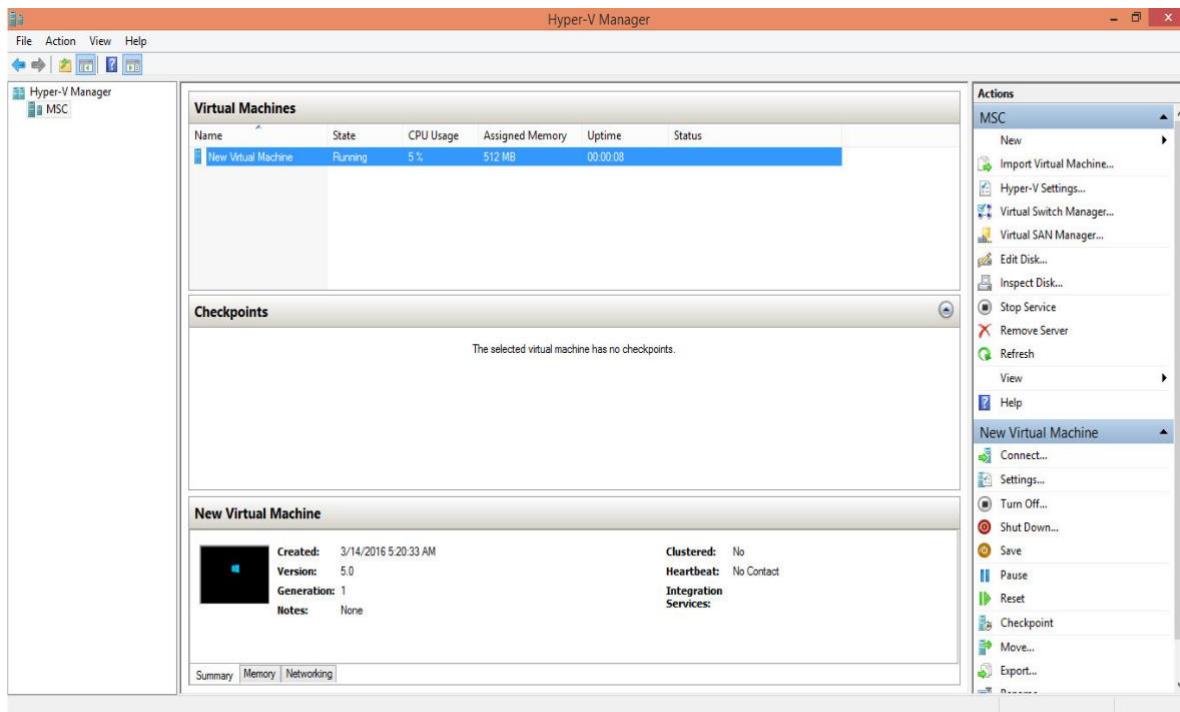
Under the **Virtual Machines** section the newly created VM “**New Virtual Machine**” is listed.



Right click on the “**New Virtual Machine**” and click on “**Start**” option.



The “New Virtual Machine” is started and the status of this VM can be seen under **Virtual Machine** section. The **details** about this **VM** can be seen at the bottom section of the Hyper-V Manager.



Practical No.9

DEVELOPING APPLICATION FOR WINDOWS AZURE.

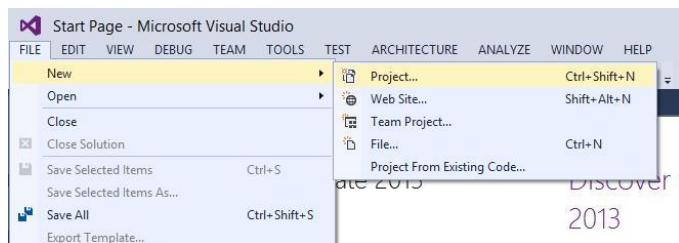
To develop an application for Windows Azure

DEVELOPING APPLICATION FOR WINDOWS AZURE.

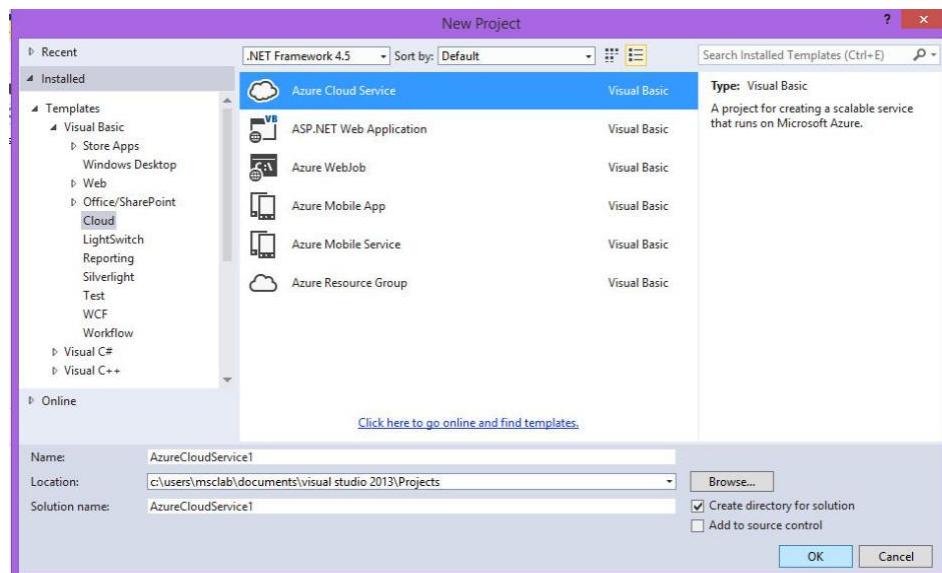
on Visual Studio install the “Microsoft Azure

SDK for .NET (VS 2013) – 2.8.2.1”

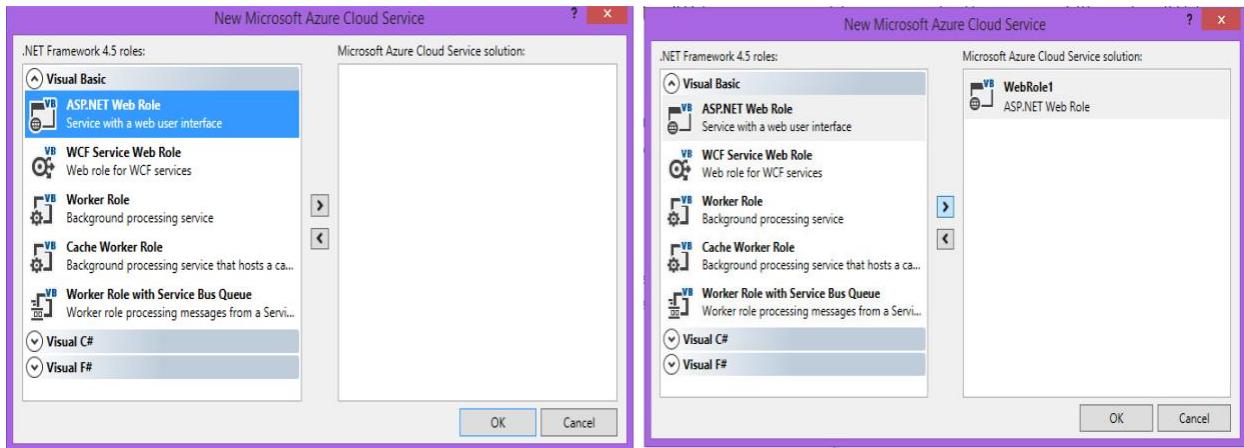
Start Visual Studio 2013. To create a new project go to **File menu** **New Project...**



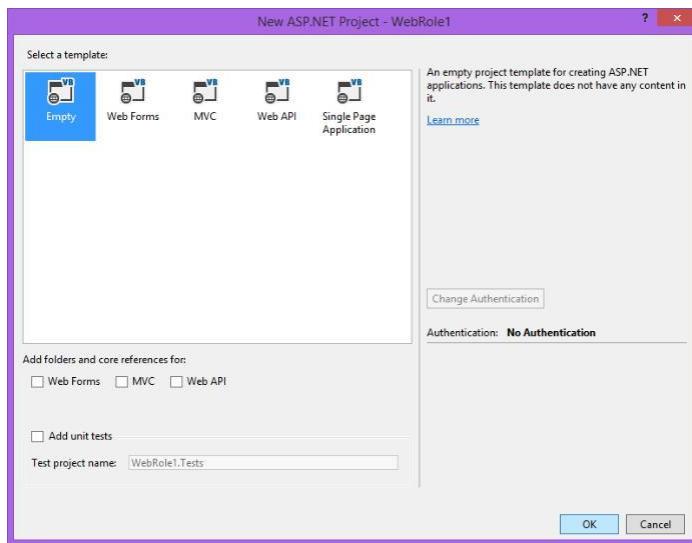
In the **New Project** window select **Azure Cloud Service**. Give a name to the project or leave it default and then click on “**OK**” button.



In the **New Microsoft Azure Cloud Service** window select “**ASP.NET Web Role**” and click on the right arrow (button). Then click on “**OK**” button.

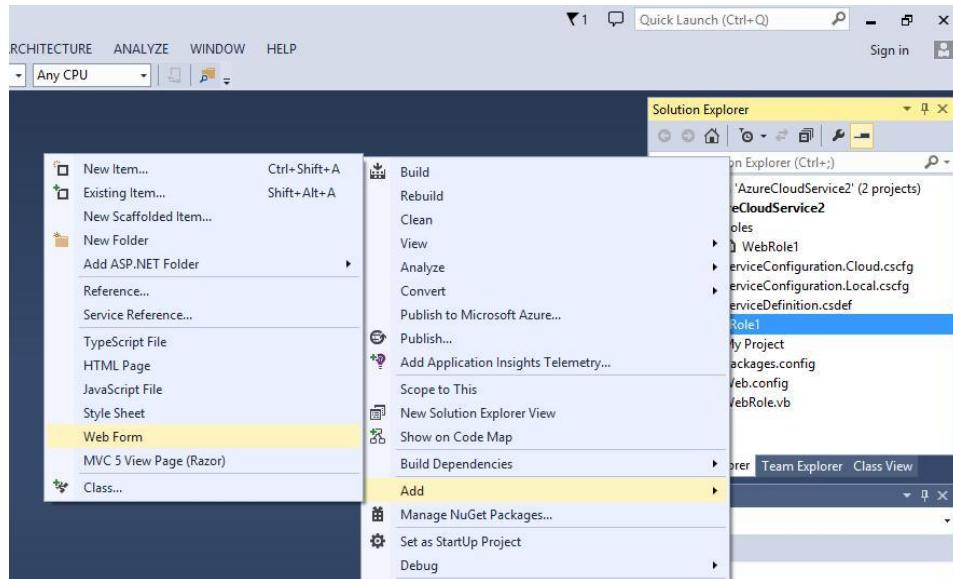


In the **New ASP .NET Project – WebRole1** select **Empty** template and click on “**OK**”button.

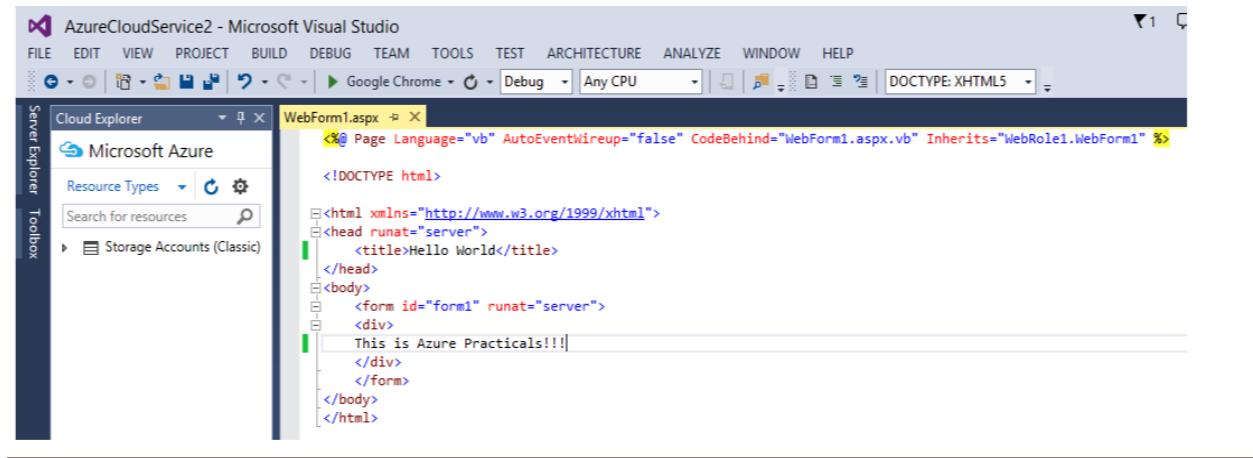


The WebRole1 is created.

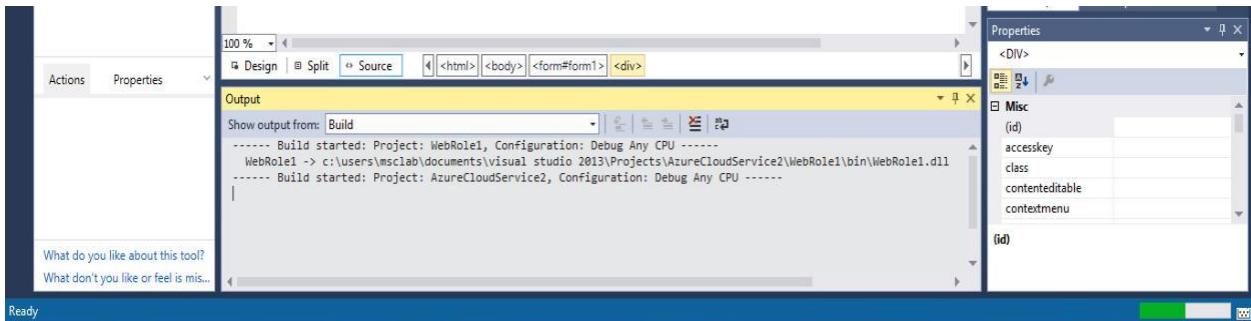
In the **Solution Explorer** right click on the **WebRole1** and select **Add → Web Form**.



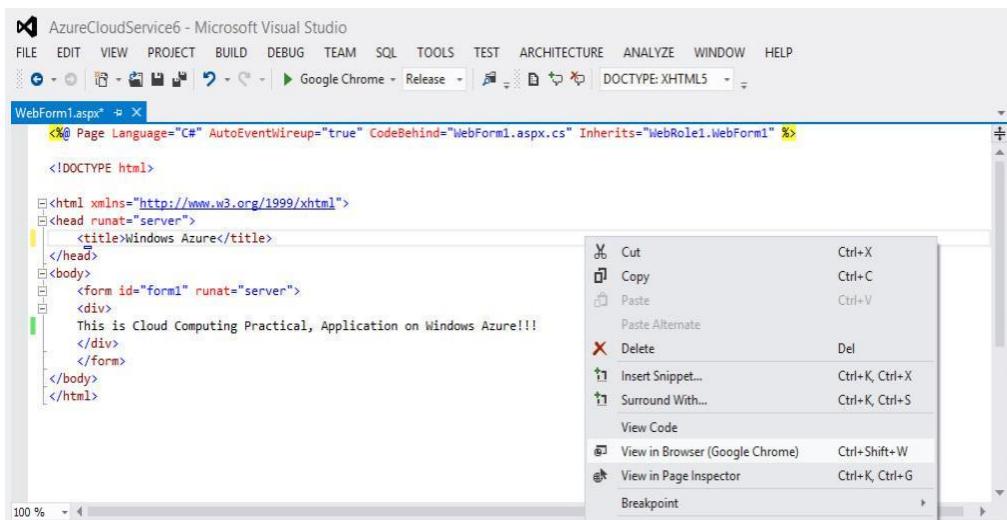
Do the required editing in the code...



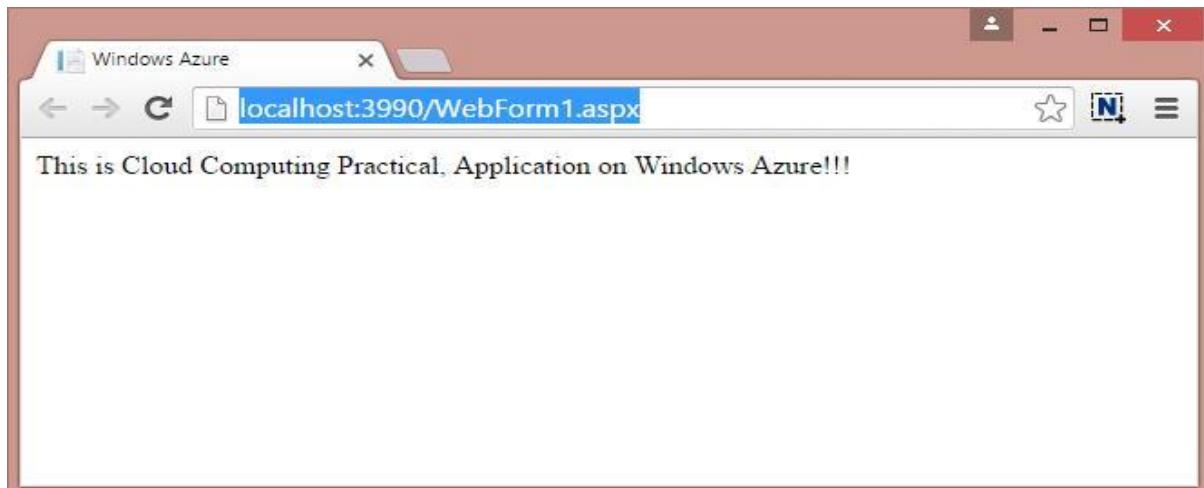
Now **build** (compile and run) the project. In the **Output** section the running of the code is shown as successful...



Then in the blank space in the WebForm1 itself right click and select “**View in Browser (Google Chrome)**” option.



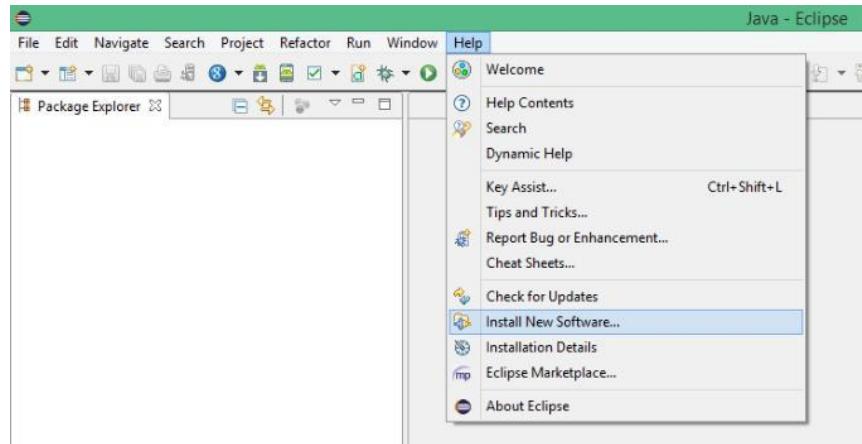
The browser here Google Chrome starts having the address as **localhost:3990/WebForm1.aspx** displaying the WebForm1's output.



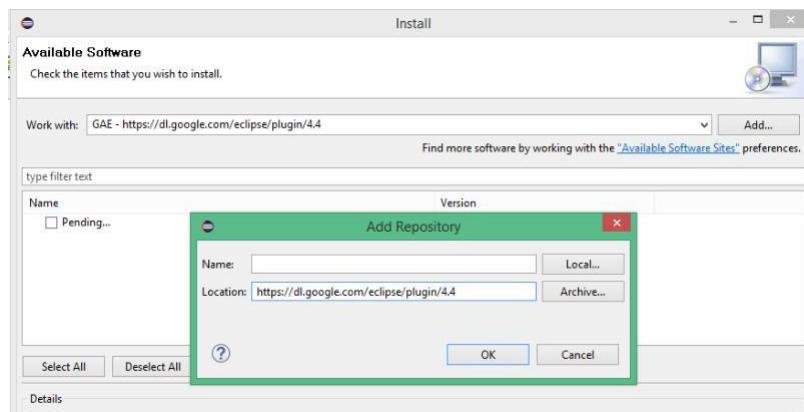
Practical No. 10

IMPLEMENT SEARCH ENGINE GOOGLE APP ENGINE (GAE)

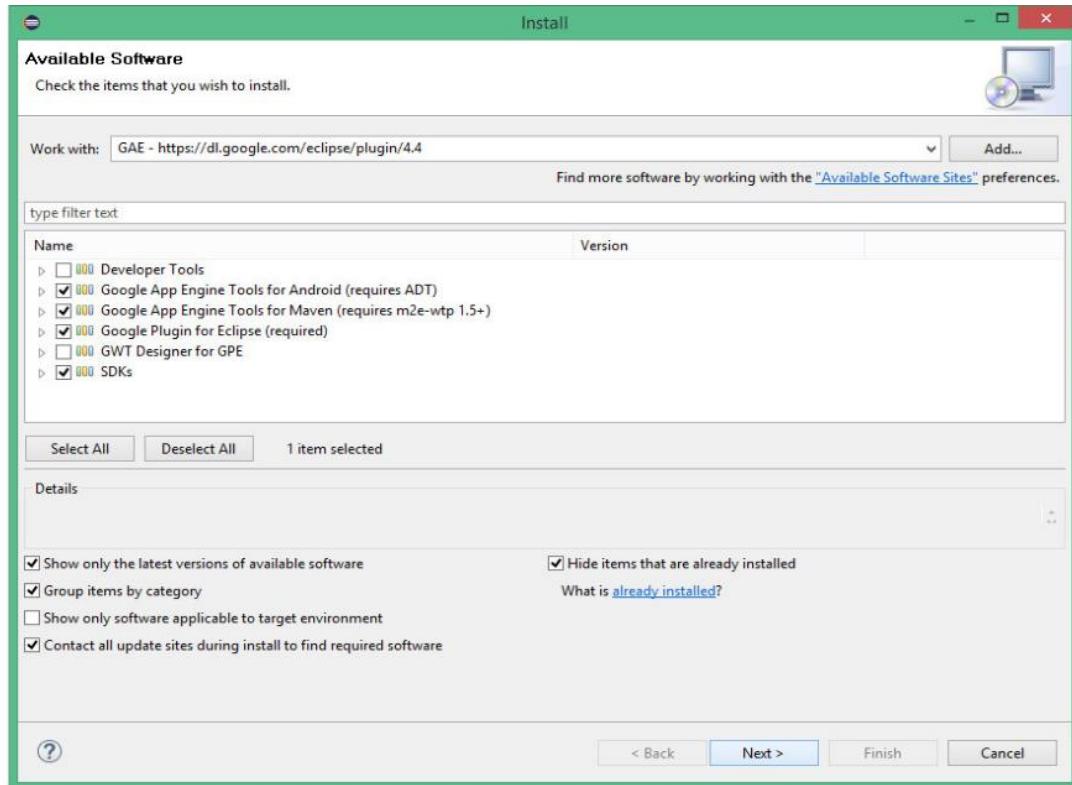
Open Eclipse Luna. Go to **Help Menu** **Install New Software...**



In **Install** window Click on the “Add” button besides the **Work with** textbox. **Add Repository** window appears. Enter the **Location** as “<https://dl.google.com/eclipse/plugin/4.4>” and click on “OK” button.

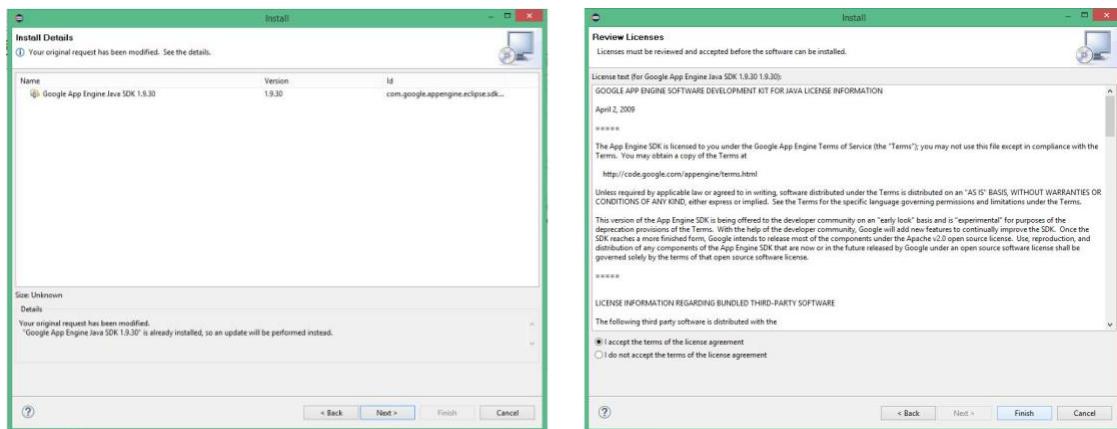


From the available softwares select the required softwares and tools as shown in the below image for the GAE. Then click on the “Next” button.

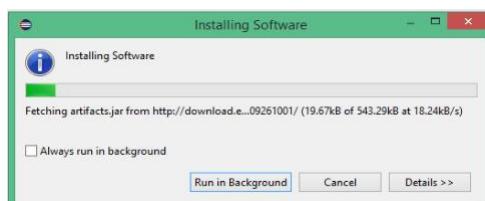


In the **Install Details** window click on “**Next**” button.

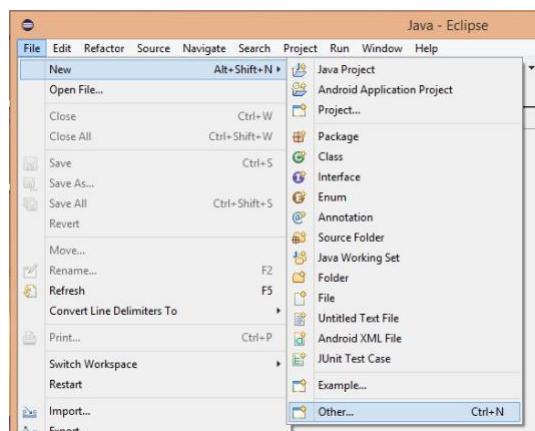
In the next window for Review Licenses select the option “**I accept.....**” and click on “**Finish**” button.



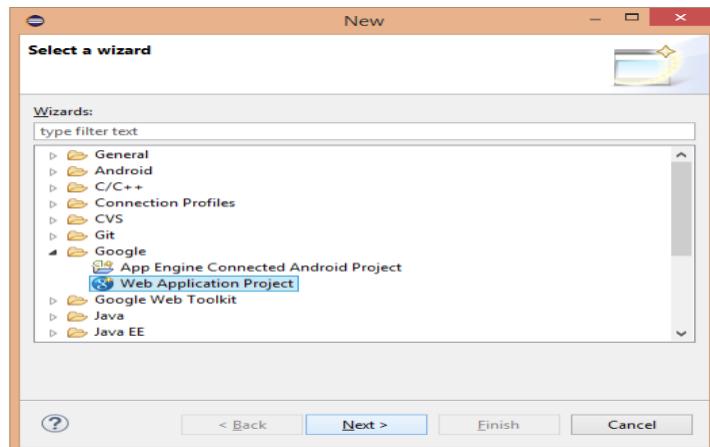
The installation is in progress...



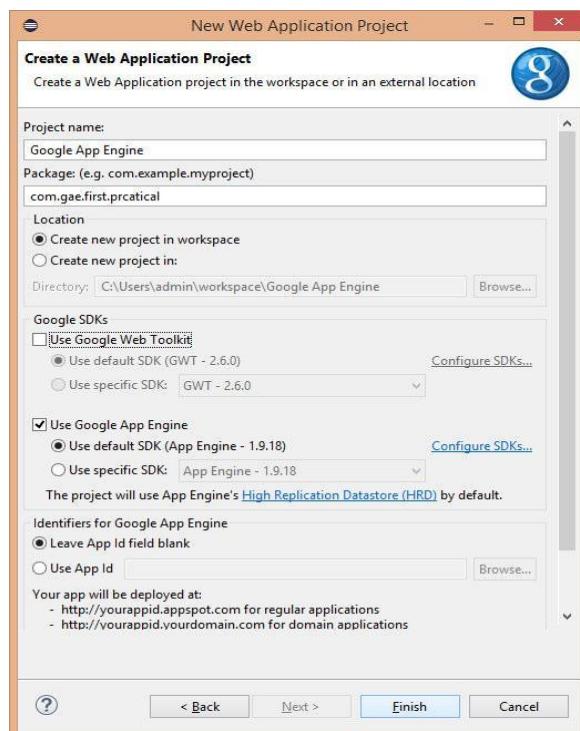
Now, go to **File Menu New Other**



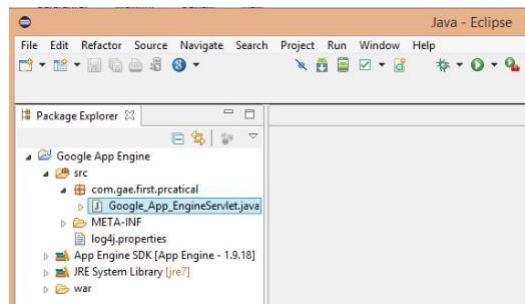
In the New window select **Google Web Application Project** and click on “Next” button.



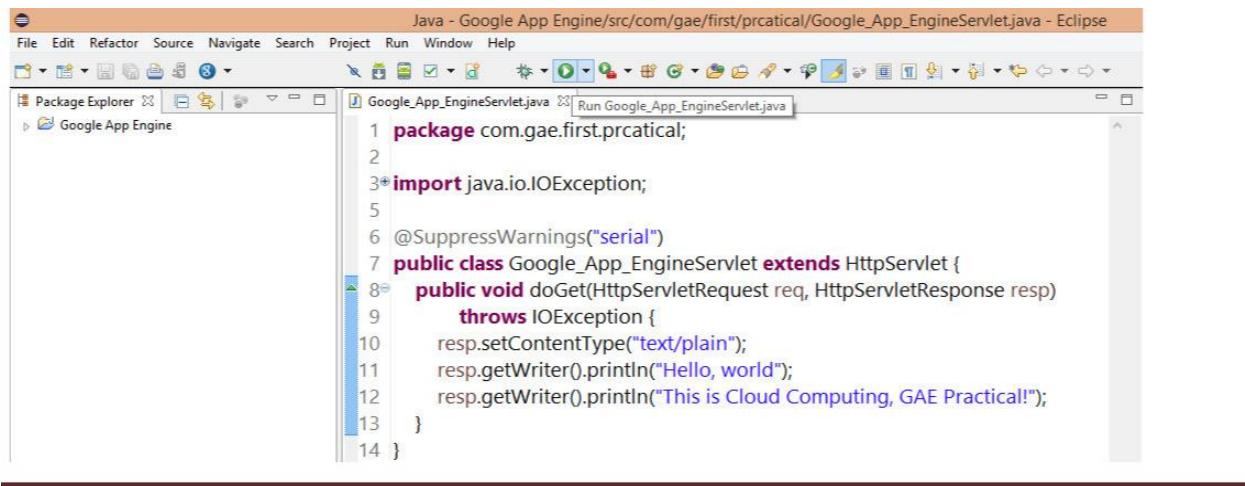
Enter the details for the new Web application project. Deselect the **Use Google Web Toolkit** option under the section **Google SDKs**. Click on the “Finish” button.



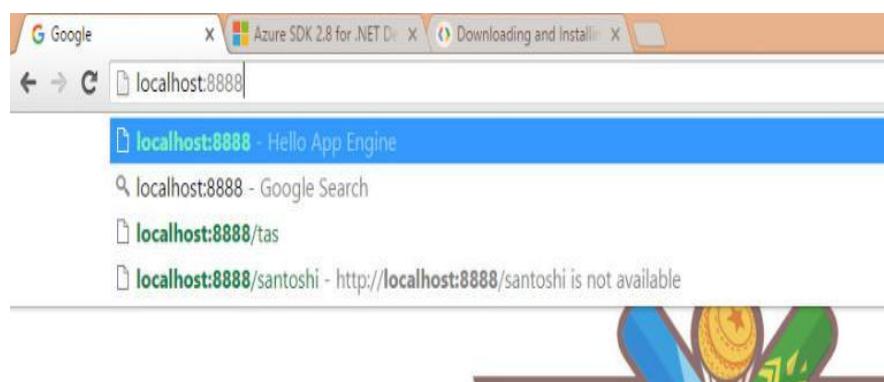
From the **Package Explorer** open the **.java** file (Here it is “**Google_App_EngineServlet.java**”).



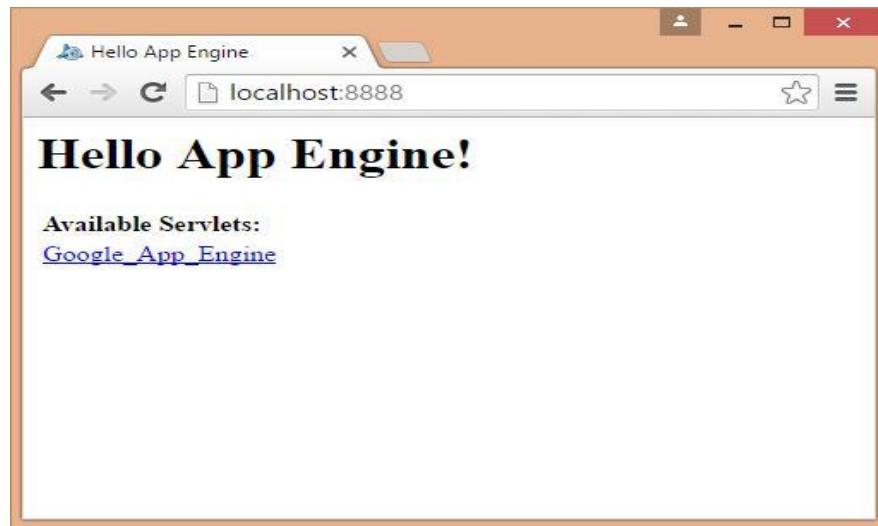
Edit the file as required (Unedited file too can be used. Here the editing is done to “what should be displayed” on the browser). **Save** the file. Click on the **Run** option available on the Tools bar.



In the browser (Here, Google Chrome) type the address as “**localhost:8888**”.



In **localhost:8888** the link to the **Google_App_EngineServlet.java** file as **Google_App_Engine** is displayed. Click on this link. It will direct you to “**localhost:8888/Google_App_Engine**”.



The **output text entered** in the **java program** is **displayed as the output** when clicked the link “**Google_App_Engine**”.

