

Cloud Computing
&
Soft Computing Techniques
Certified Journal

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

MASTER OF SCIENCE
(INFORMATION TECHNOLOGY)

By
Uday Valmik Lanke



DEPARTMENT OF INFORMATION TECHNOLOGY
KERALEYA SAMAJAM (REGD.) DOMBIVLI'S
MODEL COLLEGE (AUTONOMOUS)
Re-Accredited 'A' Grade by NAAC

(Affiliated to University of Mumbai)

FOR THE YEAR
(2022-23)



Keraleeya Samajam(Regd.) Dombivli's
MODEL COLLEGE
Re-Accredited Grade "A" by NAAC

Kanchan Goan Village, Khambalpada, Thakurli East – 421201
Contact No – 7045682157, 7045682158. www.model-college.edu.in

**DEPARTMENT OF INFORMATION TECHNOLOGY
AND COMPUTER SCIENCE**

CERTIFICATE

This is to certify that Mr. /Miss _____

Studying in Class _____ Seat No. _____

Has completed the prescribed practicals in the subject _____

During the academic year _____

Date : _____

External Examiner

Internal Examiner
M.Sc. Information Technology

INDEX

Sr No	Practical name	Date	Sign
1	<p>Write a program for implementing Client Server communication model using TCP.</p> <p>1A: A client server-based program using TCP to find if the number entered is prime.</p> <p>1B: A client server TCP based chatting application</p>	<p>8th October 2022</p> <p>15th October 2022</p>	
2	<p>Write a program for implementing Client Server communication model using UDP.</p> <p>2A: A client server-based program using UDP to find if the number entered is even or odd.</p> <p>2B. A client server-based program using UDP to find the factorial of the entered number.</p> <p>2C. A program to implement simple calculator operations like addition, subtraction, multiplication and division.</p> <p>2D. A program that finds the square, square root, cube and cube root of the entered number.</p>	16 th October 2022	
3	A multicast Socket example	22 nd October 2022	
4	<p>Write a program to show the object communication using RMI.</p> <p>4A: A RMI based application program to display current date and time.</p> <p>4B: A RMI based application program that converts digits to words, e.g., 123 will be converted to one two three.</p>	9 th November 2022	
5	Implement Xen virtualization and manage with Xen Center	3 rd December 2022	
6	Implement virtualization using VMWare ESXi Server and managing with vCenter	10 th December 2022	
7	Implement windows Hyper V virtualization	13 th December 2023	

CLOUD COMPUTING

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:

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());
        }
    }
}
```

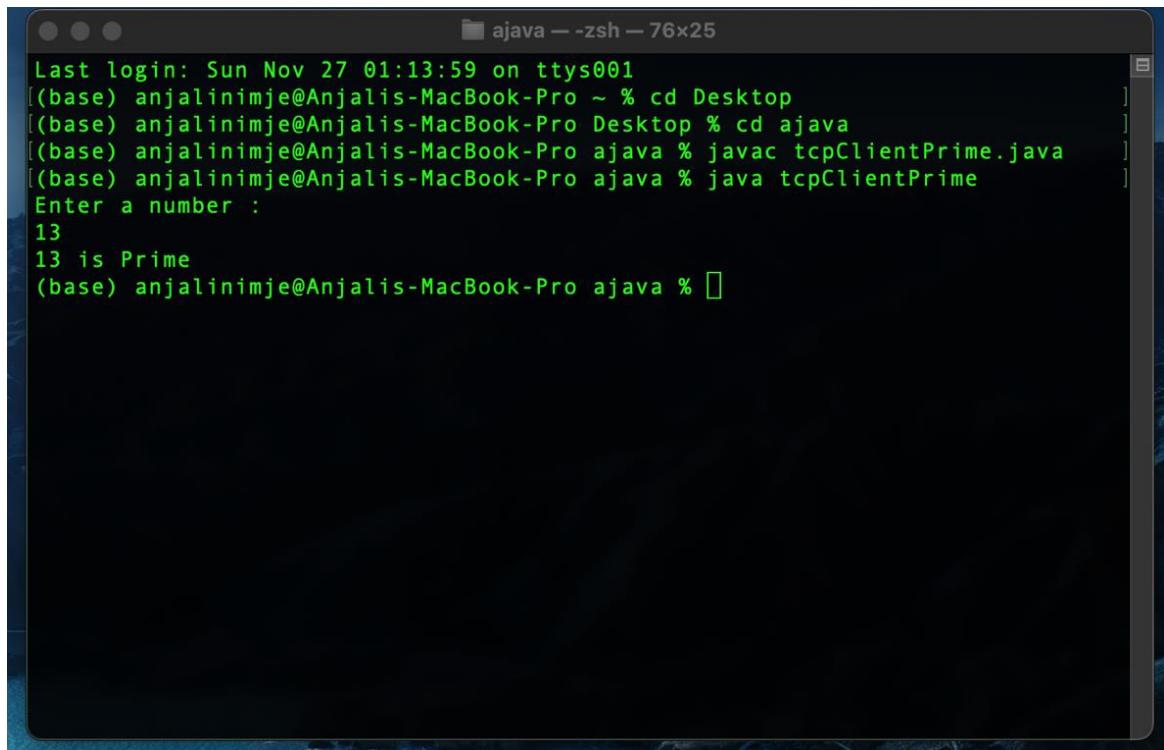
2. tcpClientPrime.java

```
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 : ");

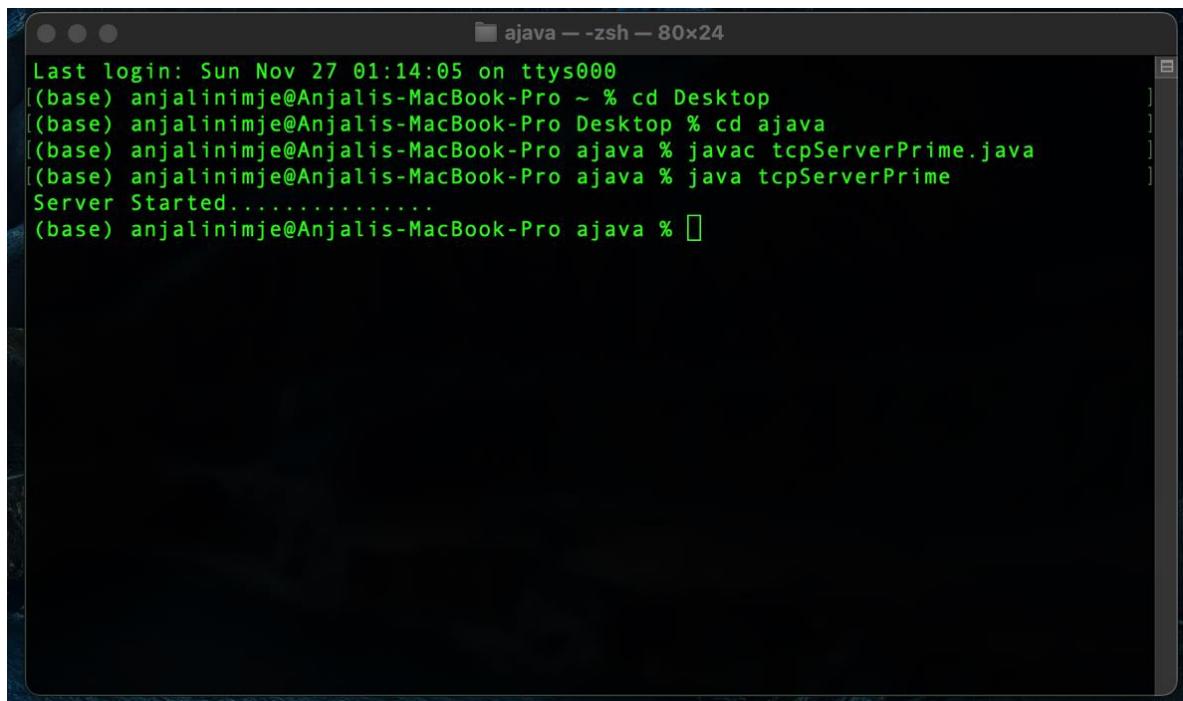
            int a = Integer.parseInt(infu.readLine());
            DataOutputStream out = new DataOutputStream(cs.getOutputStream());
            out.writeInt(a);

            DataInputStream in = new DataInputStream(cs.getInputStream());
            System.out.println(in.readUTF()); cs.close();
        }
        catch(Exception e)
        {
            System.out.println(e.toString());
        }
    }
}
```

OUTPUT:



```
Last login: Sun Nov 27 01:13:59 on ttys001
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac tcpClientPrime.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java tcpClientPrime
Enter a number :
13
13 is Prime
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```



```
Last login: Sun Nov 27 01:14:05 on ttys000
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac tcpServerPrime.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java tcpServerPrime
Server Started.....
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```

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:

```
ajava --zsh-- 80x24
Last login: Sun Nov 27 01:14:10 on ttys001
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac ChatServer.java
Note: ChatServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac ChatServer.java -Xlint
ChatServer.java:26: warning: [deprecation] readLine() in DataInputStream has bee
n deprecated
        while((receive = in.readLine()) != null)
                           ^
1 warning
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java ChatServer
Waiting for client to connect..
Client Says : hi anjali.
Server Says : hi anushka
Client Says : bye
Server Says : bye anjali client
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```

```
ajava --zsh-- 80x24
Last login: Sun Nov 27 01:17:46 on ttys002
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac ChatClient.java
Note: ChatClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java ChatClient
To stop chatting with server type STOP
Client Says: hi anjali.
Server Says : hi anushka
Client Says : bye
Server Says : bye anjali client
Client Says : STOP
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```

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.*;
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));

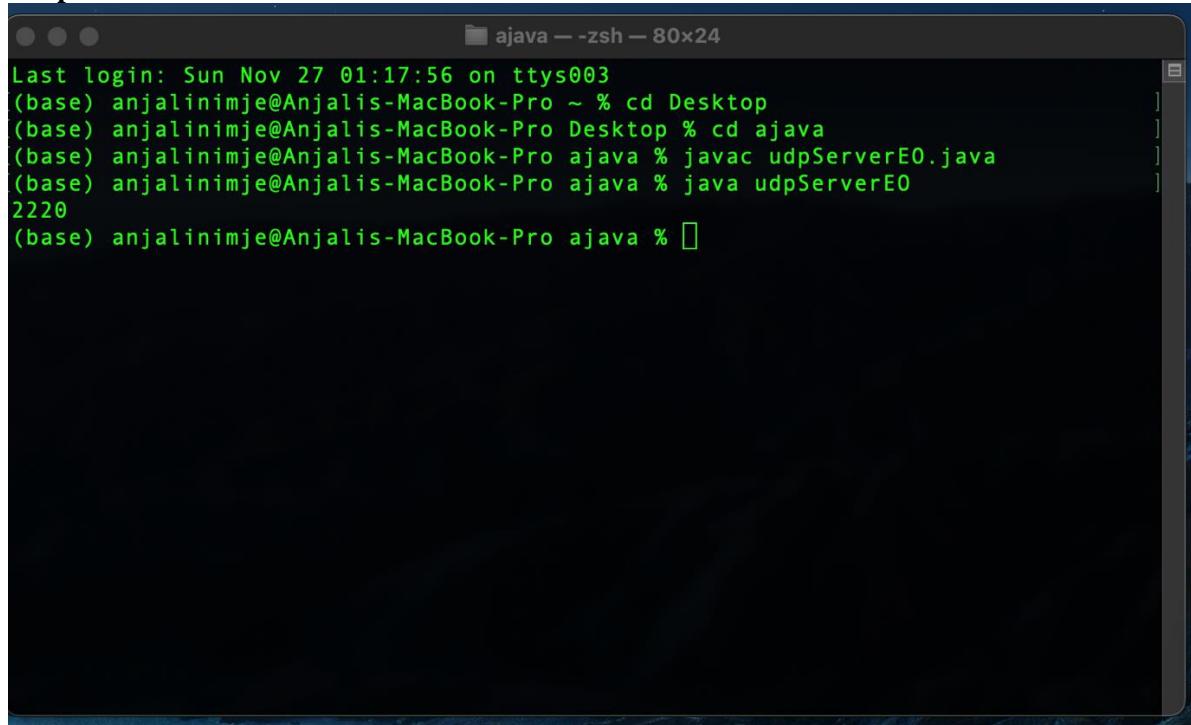
            System.out.println("Enter a number : ");
            String num = br.readLine();
            byte b[] = new byte[1024];
            b=num.getBytes();
            DatagramPacket dp = new

DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);

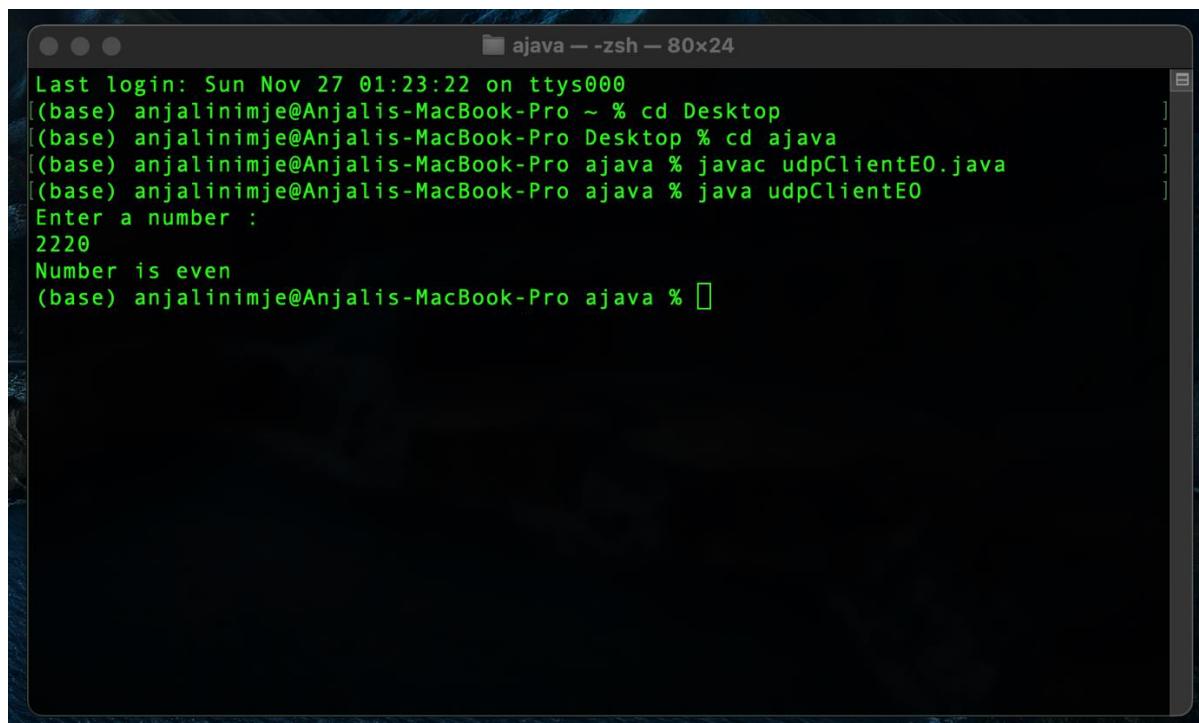
            ds.send(dp);
            byte b1[] = new byte[1024];
            DatagramPacket dp1 = new
            DatagramPacket(b1,b1.length); ds.receive(dp1);

            String str = new
            String(dp1.getData(),0,dp1.getLength());
            System.out.println(str);
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

Output:



```
Last login: Sun Nov 27 01:17:56 on ttys003
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac udpServerE0.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java udpServerE0
2220
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```



```
Last login: Sun Nov 27 01:23:22 on ttys000
[(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
[(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
[(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac udpClientE0.java
[(base) anjalinimje@Anjalis-MacBook-Pro ajava % java udpClientE0
Enter a number :
2220
Number is even
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```

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

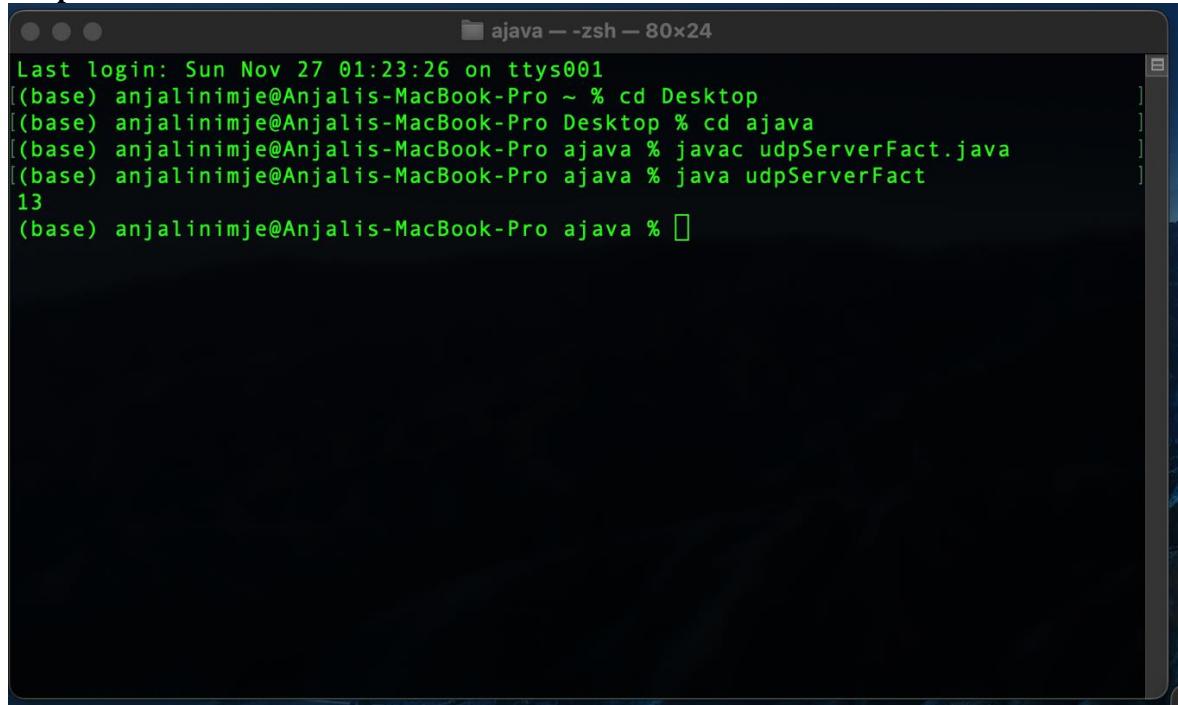
```
/*Program which calculate factorial of a number*/
import java.io.*;
import java.net.*;
public class udpClientFact
{
    public static void main(String args[])
    {
        try
        {
            DatagramSocket ds = new DatagramSocket(1000);
            BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));

            System.out.println("Enter a number : ");
            String num = br.readLine();
            byte b[] = new byte[1024];
            b=num.getBytes();
            DatagramPacket dp = new

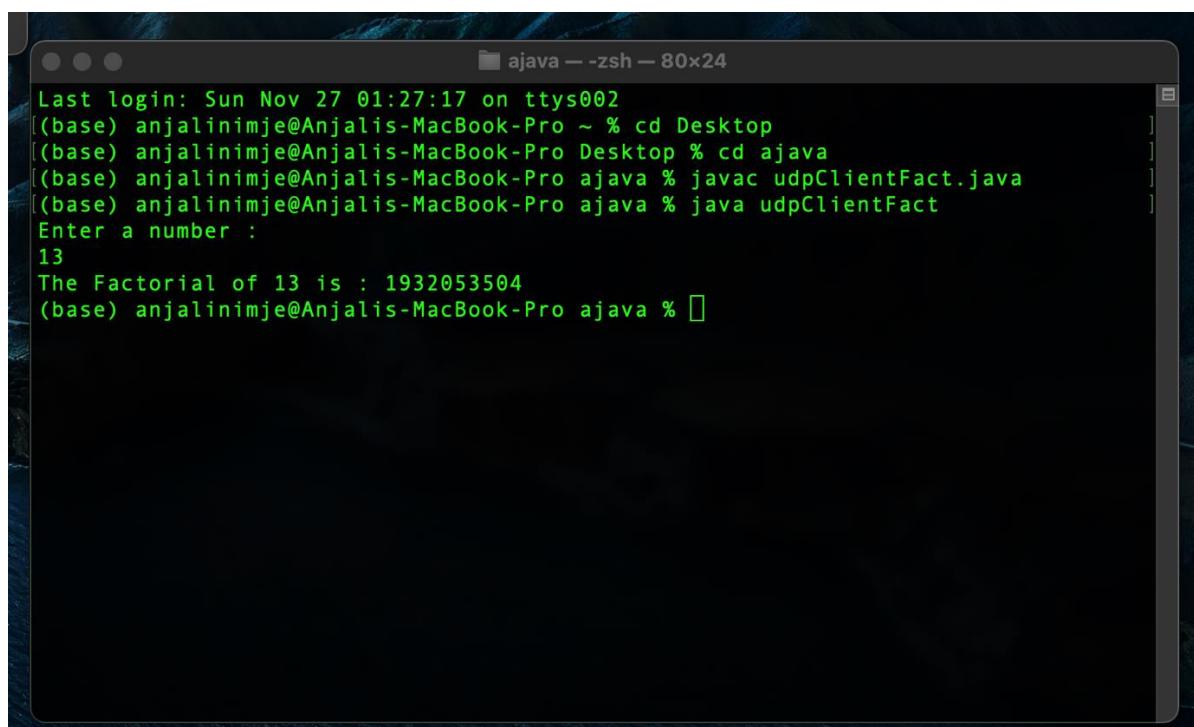
DatagramPacket(b,b.length,InetAddress.getLocalHost(),2000);

            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:



```
ajava -- zsh -- 80x24
Last login: Sun Nov 27 01:23:26 on ttys001
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac udpServerFact.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java udpServerFact
13
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```



```
ajava -- zsh -- 80x24
Last login: Sun Nov 27 01:27:17 on ttys002
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac udpClientFact.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java udpClientFact
Enter a number :
13
The Factorial of 13 is : 1932053504
(base) anjalinimje@Anjalis-MacBook-Pro ajava %
```

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 3 4\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:

```
Last login: Sun Nov 27 01:43:58 on ttys003
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac RPCServer.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java RPCServer
add 22 31
result : 53

sub 31 22
result : 9

mul 20 22
result : 440

div 40 4
result : 10

□
```

```
Last login: Sun Nov 27 01:47:41 on ttys000
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac RPCClient.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java RPCClient

RPC Client

Enter method name and parameter like add 3 4

add 22 31

Result = 53

sub 31 22

Result = 9

mul 20 22

Result = 440

div 40 4

Result = 10

□
```

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 :

```
ajava — java RPCNumServer — 80x24
Last login: Sun Nov 27 01:50:14 on ttys003
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac RPCNumServer.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java RPCNumServer
square 5
result :25.0

squareroot 25
result :5.0

cube 6
result :216.0

cuberoot 27
result :3.0

[]
```

```
ajava — java RPCNumClient — 80x31
Last login: Sun Nov 27 01:55:57 on ttys000
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac RPCNumClient.java
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java RPCNumClient

RPC Client

1. Square of the number - square
2. Square root of the number - square root
3. Cube of the number - cube
4. Cube root of the number cube root
Enter method name and the number

square 5

Result = 25.0

squareroot 25

Result = 5.0

cube 6

Result = 216.0

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;

        // set the multicast address to your local subnet
        address = InetAddress.getByName("239.1.2.3");
        socket = new MulticastSocket();

        // join a Multicast group and send the group
        messages socket.joinGroup(address);
        byte[] data =
        null; for(;;)
        {
            Thread.sleep(10000);
            System.out.println("Sending ");
            String str = ("This is Anjali and Anushka Calling....");
            Data = str.getBytes();
            packet = new DatagramPacket(data, str.length(),address,PORT);
            // Sends the packet
            socket.send(packet);
        } // end for
    } // end main
} // end class BroadcastServer
```

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;

        // set the mulitcast address to your local subnet
        address = InetAddress.getByName("239.1.2.3");
        socket = new MulticastSocket(PORT);

        //join a Multicast group and wait for a
        message socket.joinGroup(address); byte[]
        data = new byte[100];
        packet = new DatagramPacket(data,data.length);
        for(;;)
        {
            // receive the packets
            socket.receive(packet);
            String str = new String(packet.getData()); System.out.println("Message
            received from "+ packet.getAddress() + "Message is : "+str);
        } // for
    } // main
} // end BroadcastClient
```

Output

```
■ ajava — java BroadcastServer — 80x24
Last login: Sun Nov 27 02:10:05 on ttys000
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac BroadcastServer.java
Note: BroadcastServer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac BroadcastServer.java -Xlint

BroadcastServer.java:24: warning: [deprecation] joinGroup(InetAddress) in Multic
astSocket has been deprecated
    socket.joinGroup(address);
          ^
1 warning
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java BroadcastServer
Sending
Sending
Sending
[
```

```
■ ajava — java BroadcastClient — 88x18
Last login: Sun Nov 27 02:10:09 on ttys001
(base) anjalinimje@Anjalis-MacBook-Pro ~ % cd Desktop
(base) anjalinimje@Anjalis-MacBook-Pro Desktop % cd ajava
(base) anjalinimje@Anjalis-MacBook-Pro ajava % javac BroadcastClient.java
Note: BroadcastClient.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
(base) anjalinimje@Anjalis-MacBook-Pro ajava % java BroadcastClient
Message received from /192.168.1.100 Message is : This is ANJALI and ANUSHKA Calling....
Message received from /192.168.1.100 Message is : This is ANJALI and ANUSHKA Calling....
Message received from /192.168.1.100 Message is : This is ANJALI and ANUSHKA Calling....
Message received from /192.168.1.100 Message is : This is ANJALI and ANUSHKA Calling....
Message received from /192.168.1.100 Message is : This is ANJALI and ANUSHKA Calling....
Message received from /192.168.1.100 Message is : This is ANJALI and ANUSHKA 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

```
Command Prompt - cmd.exe -rmiregistry - rmiregistry
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Student00>d:

D:\>cd "ANJALI MSC"

D:\ANJALI MSC>cmd.exe -rmiregistry
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:\ANJALI MSC>set path="C:\Program Files\Java\jdk1.8.0_121\bin"

D:\ANJALI MSC>javac ServerDate.java

D:\ANJALI MSC>javac ClientDate.java

D:\ANJALI MSC>rmic ServerDate
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

D:\ANJALI MSC>rmiregistry
```

```
Command Prompt - java ServerDate
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Student00>d:

D:\>cd "ANJALI MSC"

D:\ANJALI MSC>set path="C:\Program Files\Java\jdk1.8.0_121\bin"

D:\ANJALI MSC>java ServerDate
Object registered.....
```

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Student00>d:

D:\>cd "ANJALI MSC"

D:\ANJALI MSC>set path="C:\Program Files\Java\jdk1.8.0_121\bin"

D:\ANJALI MSC>java ClientDate
Sat Dec 10 12:06:08 IST 2022

D:\ANJALI MSC>
```

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:

```
Command Prompt - cmd.exe -rmiregistry - rmiregistry
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Student00>d:

D:\>cd "ANJALI MSC"

D:\ANJALI MSC>cmd.exe -rmiregistry
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:\ANJALI MSC>set path="C:\Program Files\Java\jdk1.8.0_121\bin"
D:\ANJALI MSC>javac ServerConvert.java
D:\ANJALI MSC>javac ClientConvert.java

D:\ANJALI MSC>rmic ServerConvert
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

D:\ANJALI MSC>rmiregistry
```

```
Command Prompt - java ServerConvert
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Student00>d:

D:\>cd "ANJALI MSC"

D:\ANJALI MSC>java ServerConvert
Object registered....
```

```
Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

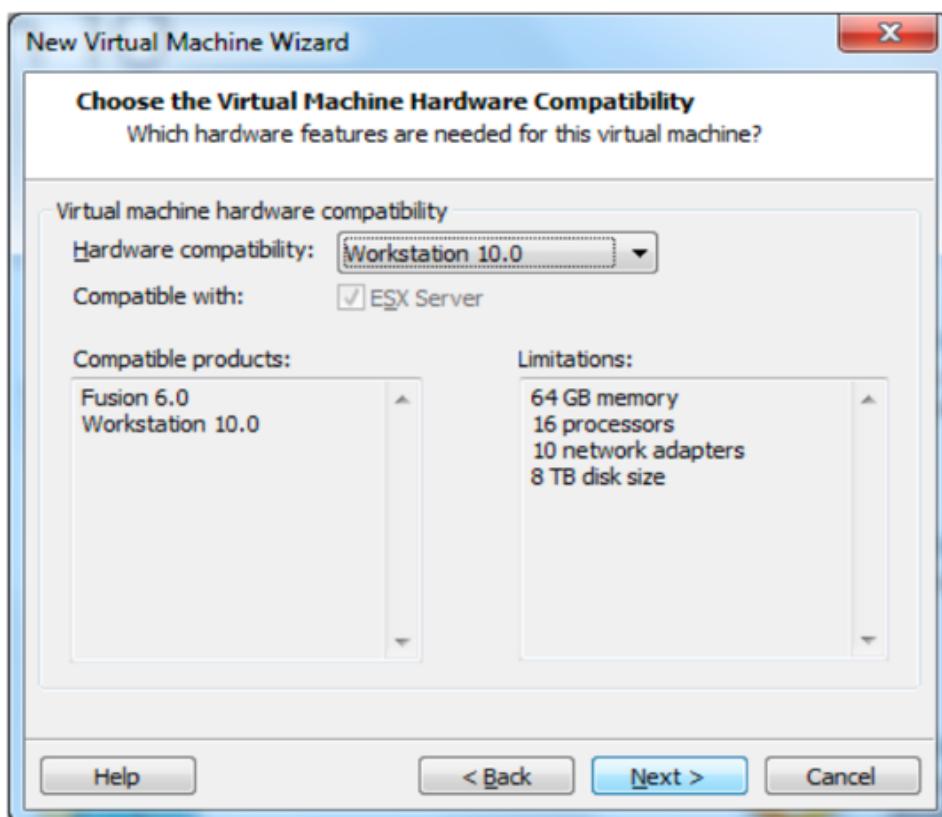
C:\Users\Student00>d:

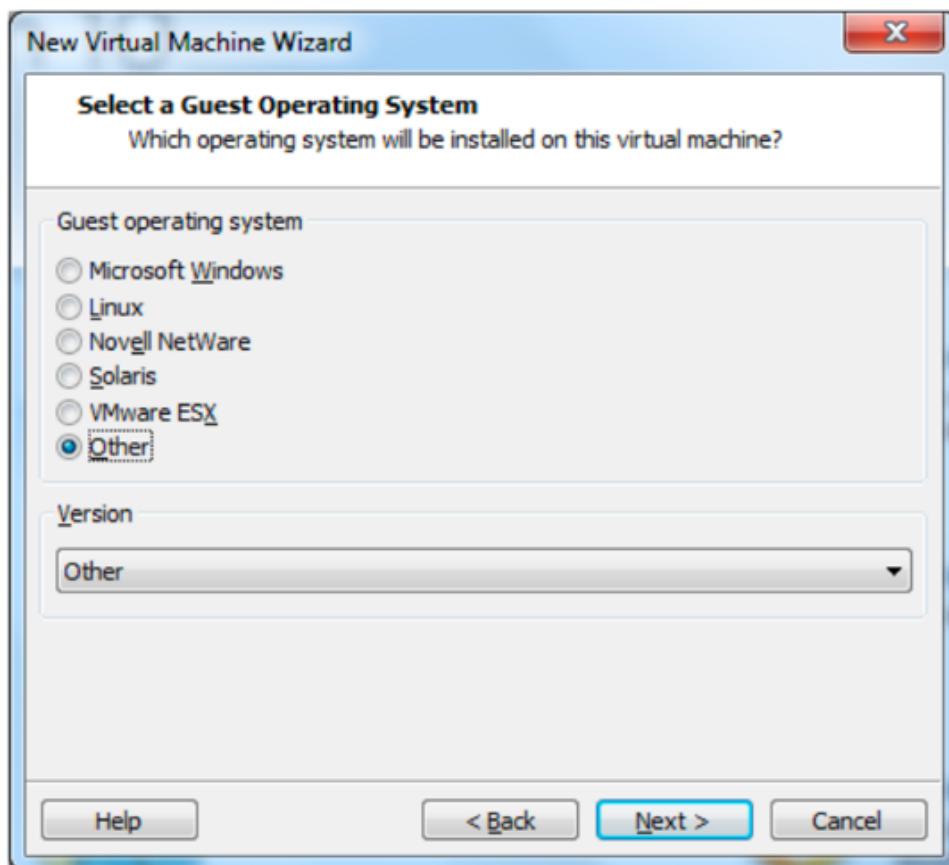
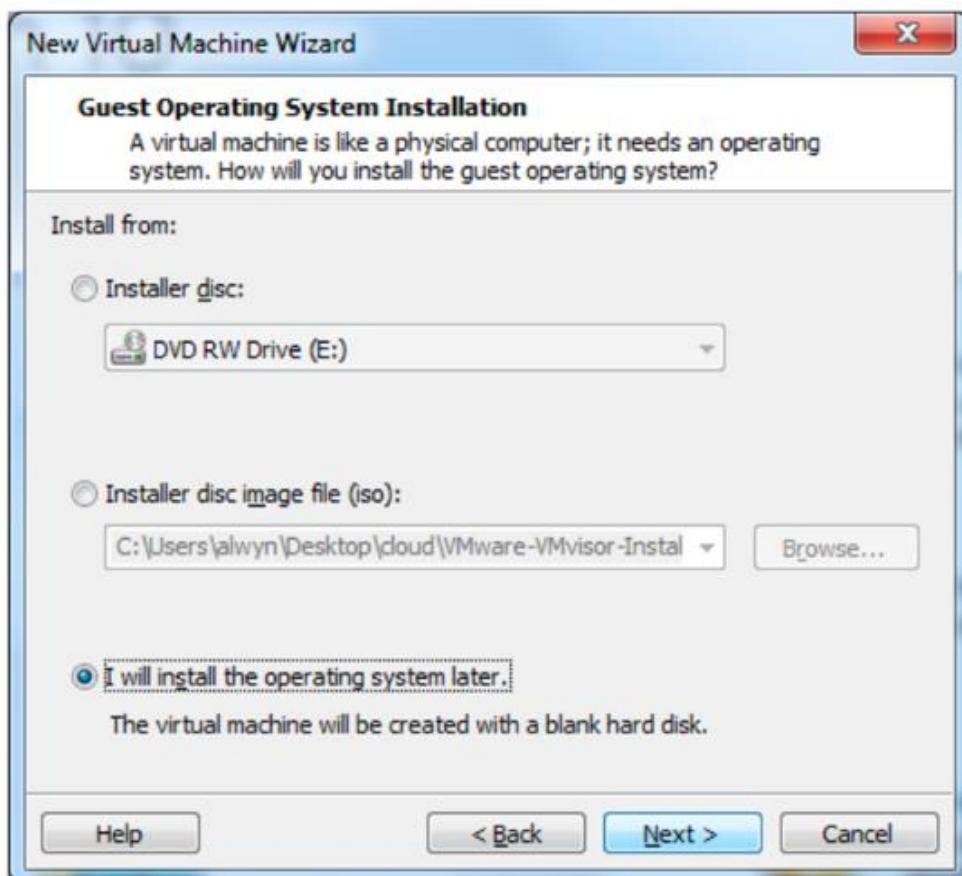
D:\>cd "ANJALI MSC"

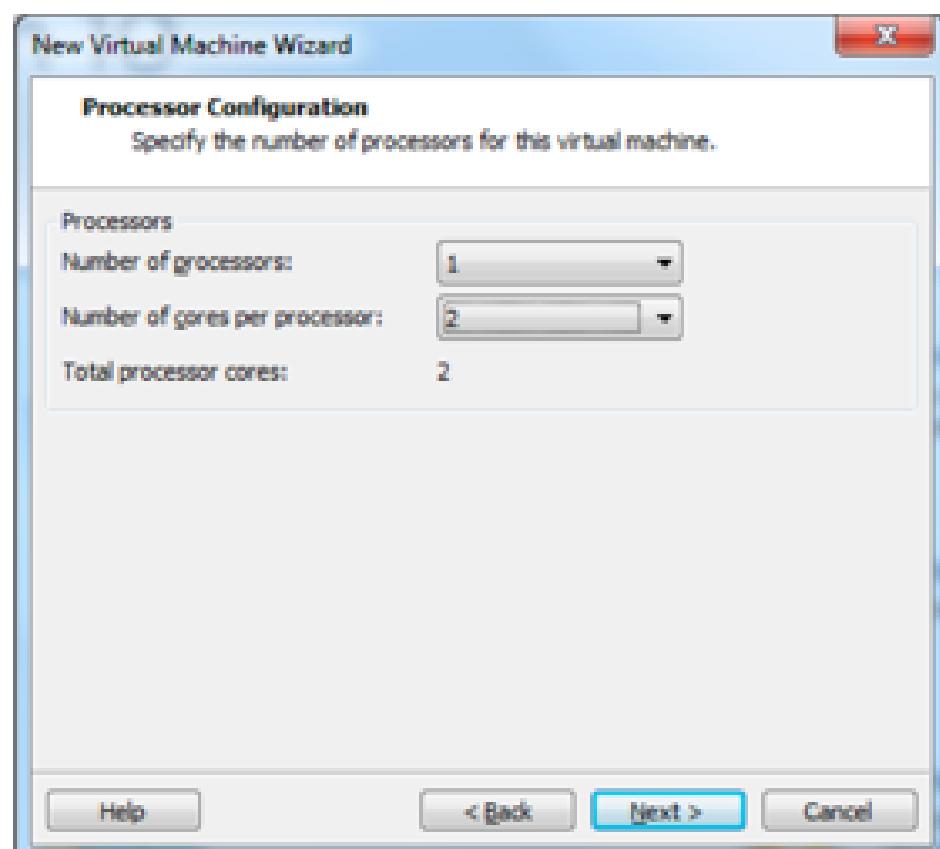
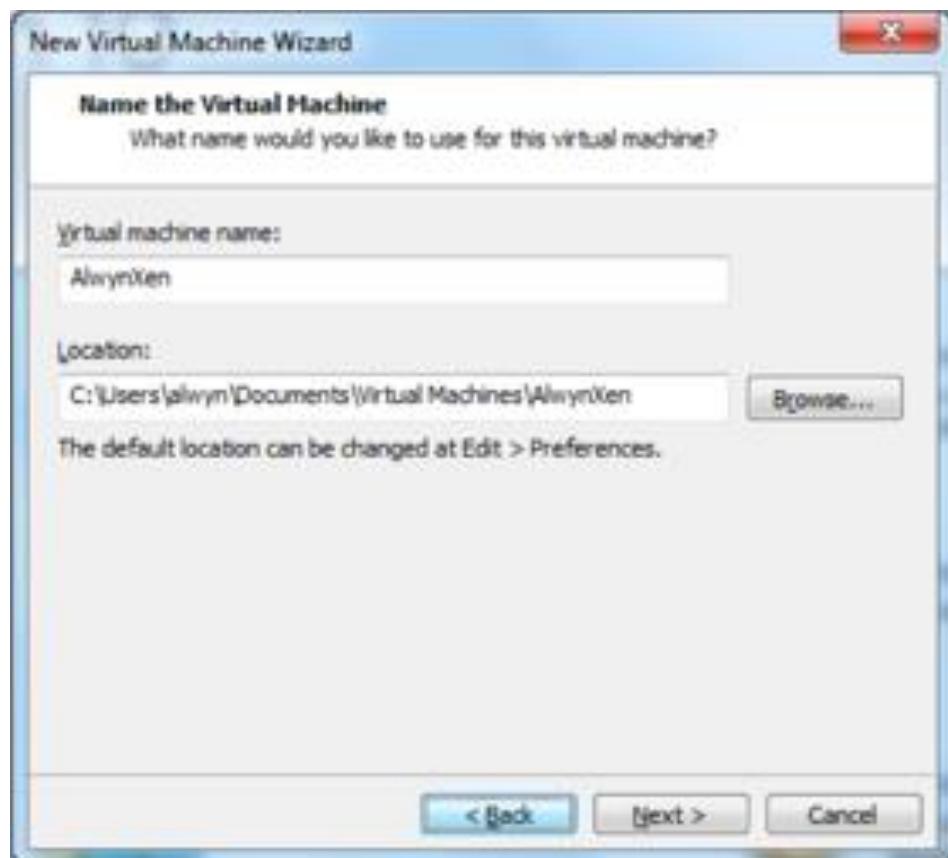
D:\ANJALI MSC>java ClientConvert
Enter a number :
22
The word representation of the entered digit is : two two
D:\ANJALI MSC>
```

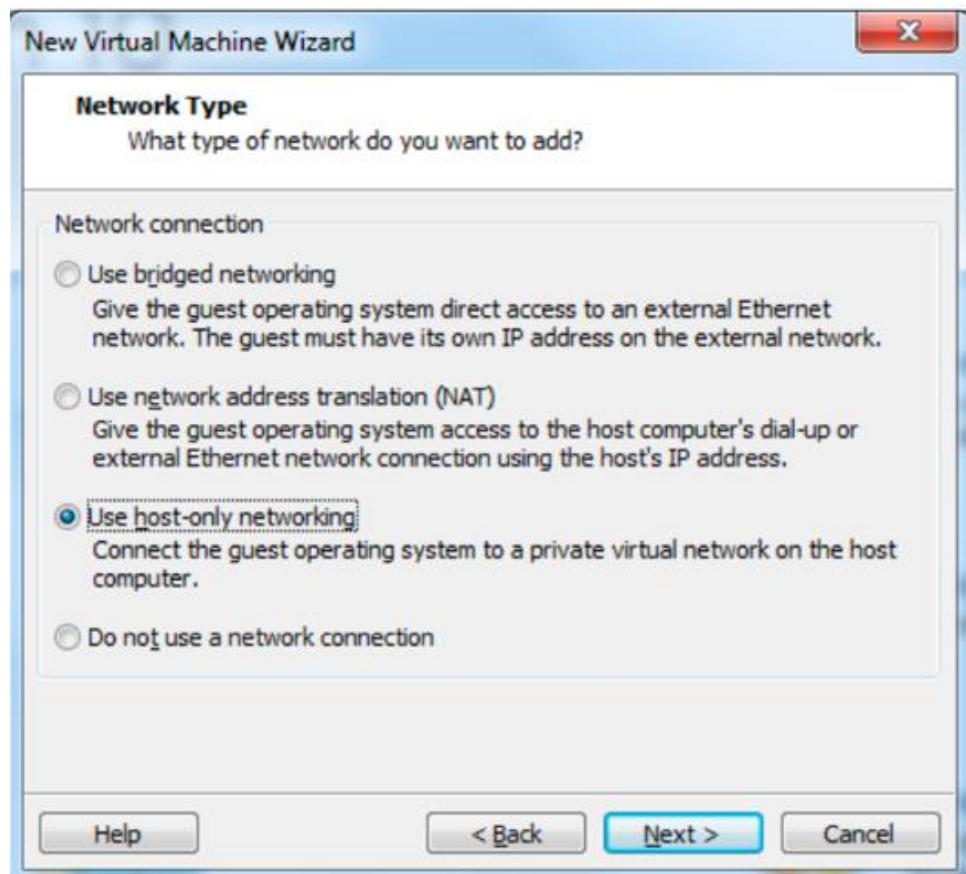
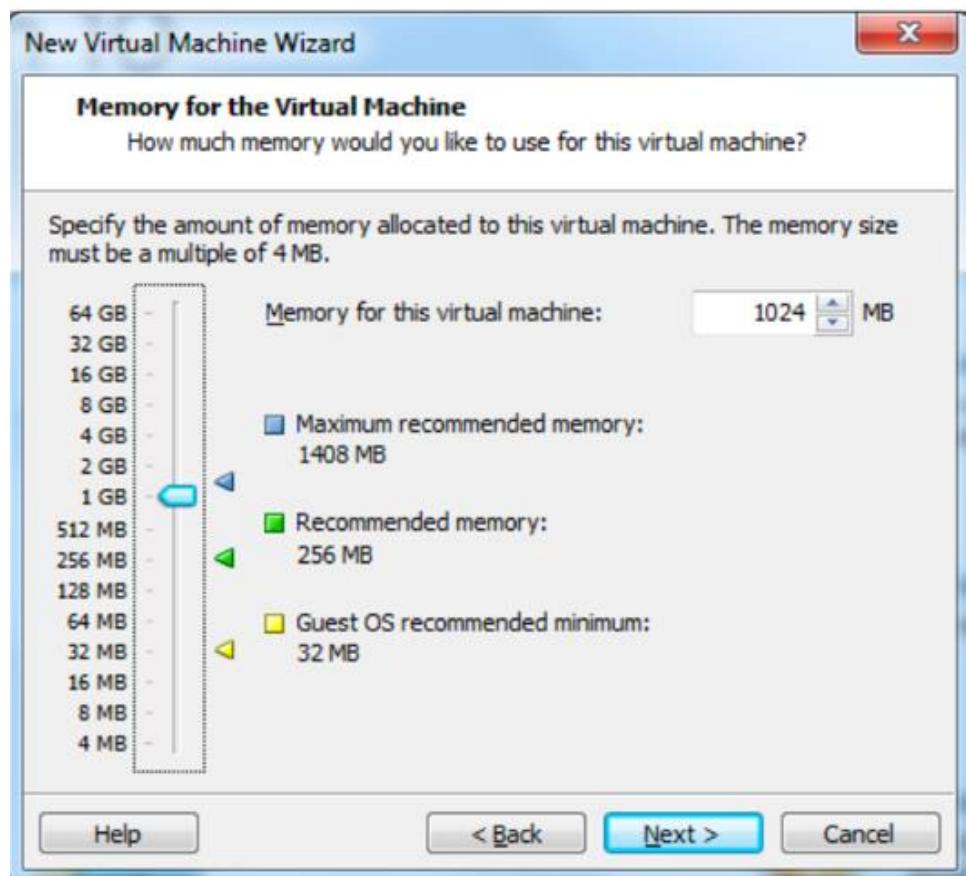
Practical: 05

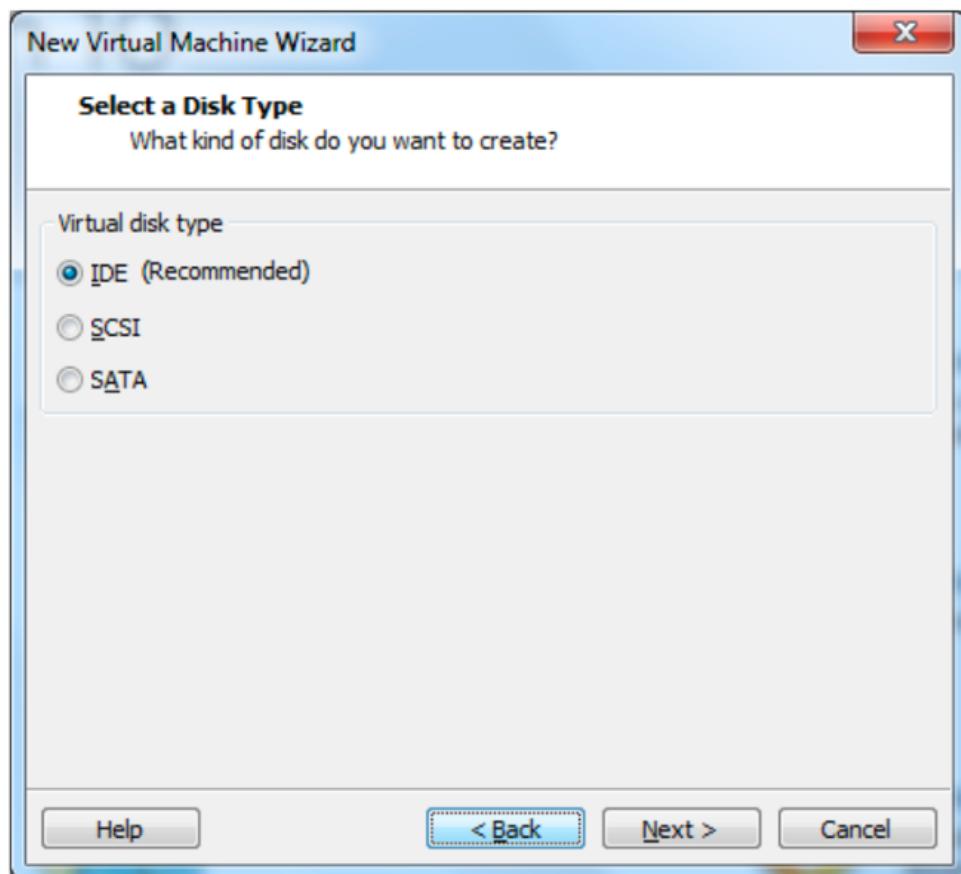
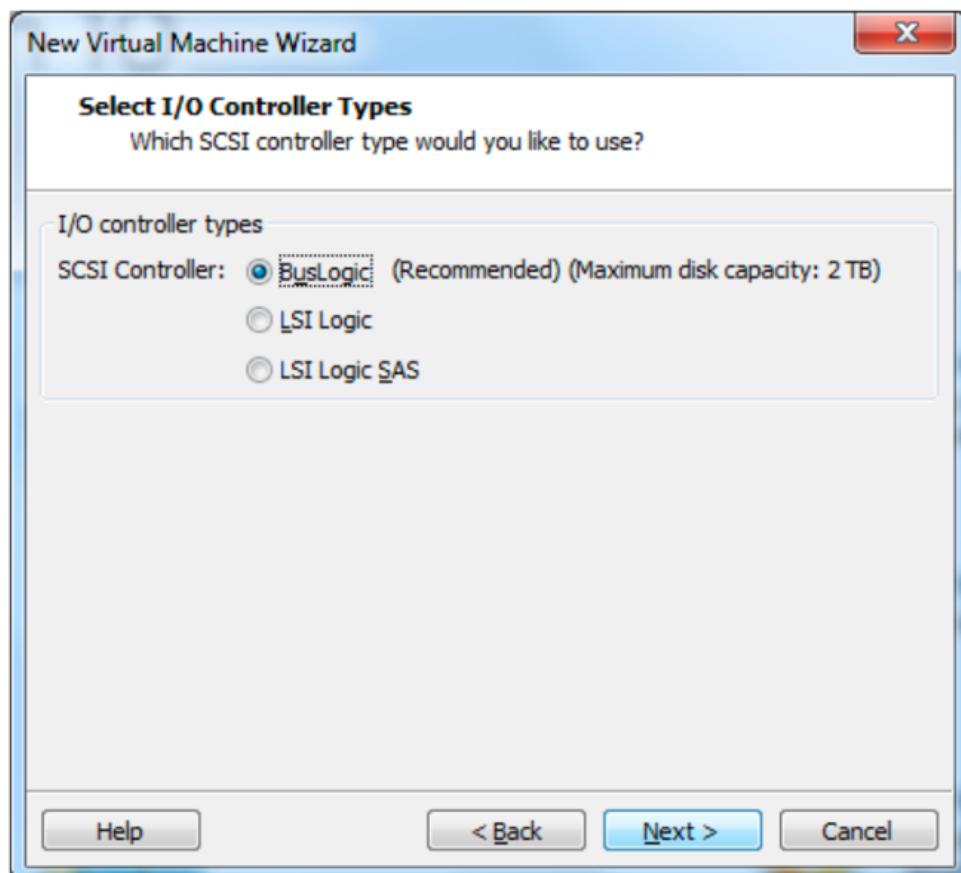
Aim: Implement Xen virtualization and manage with Xen Center

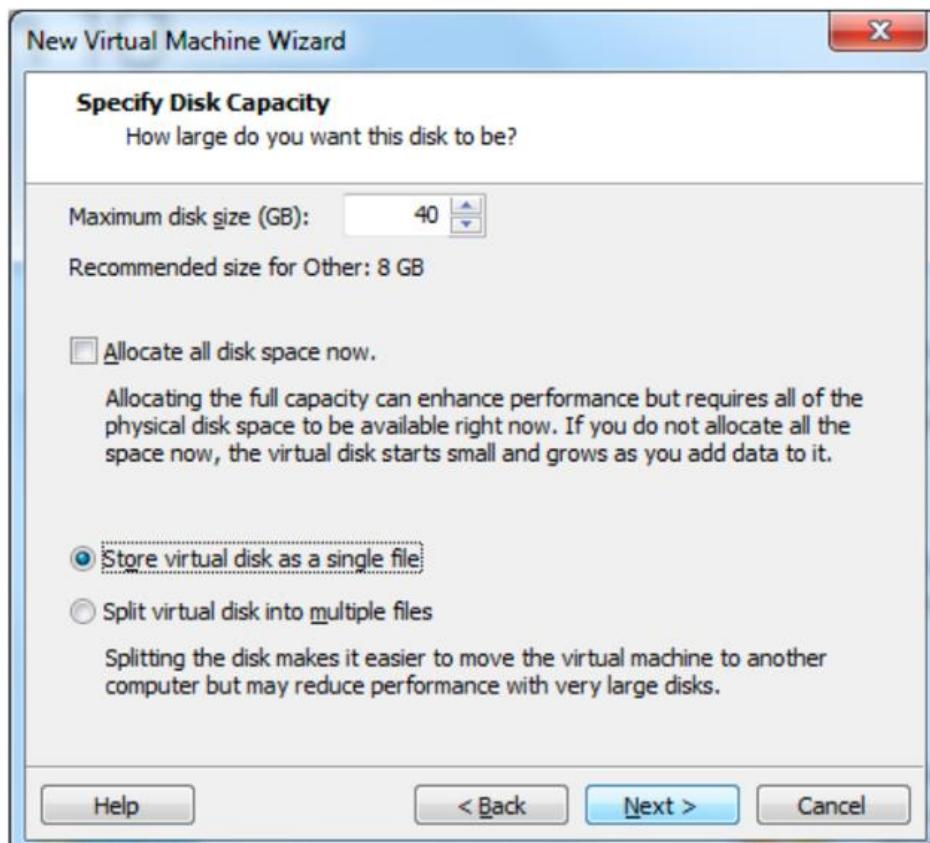
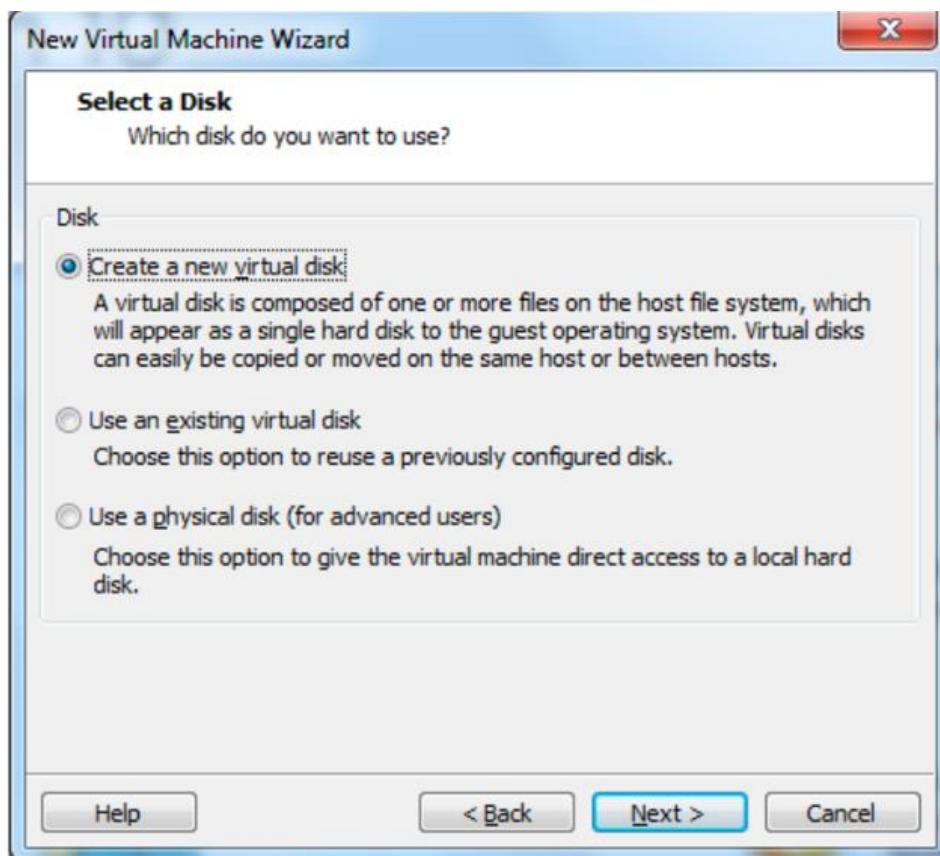


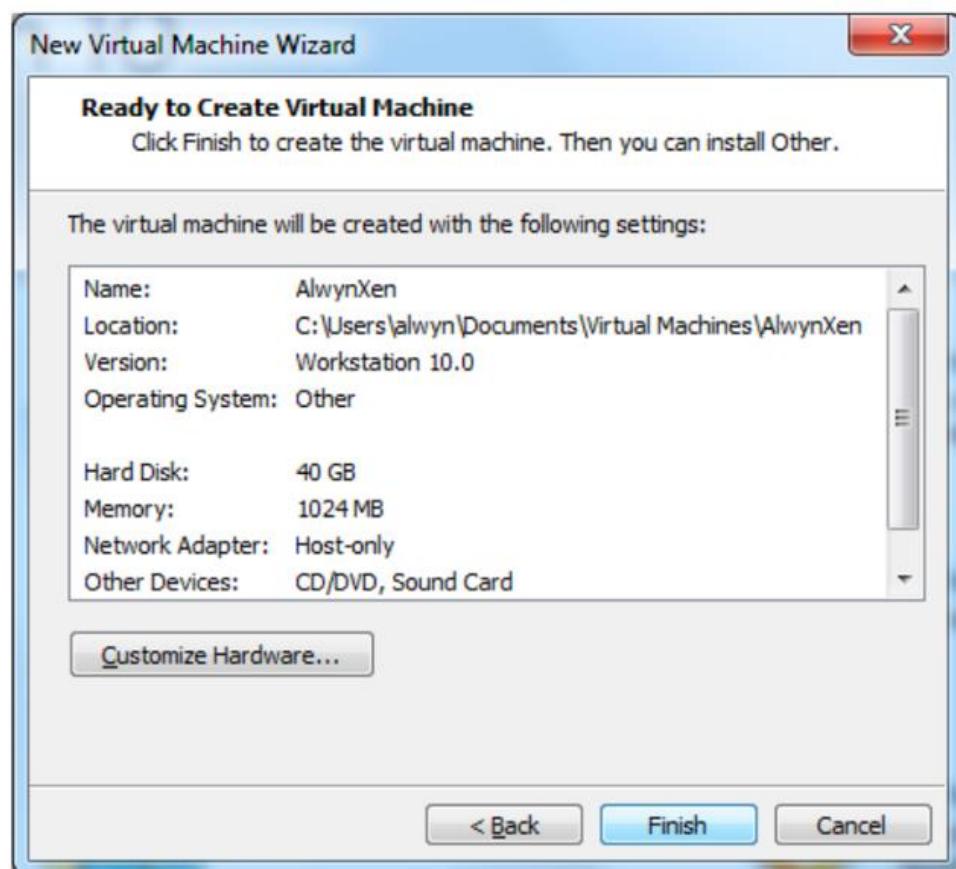
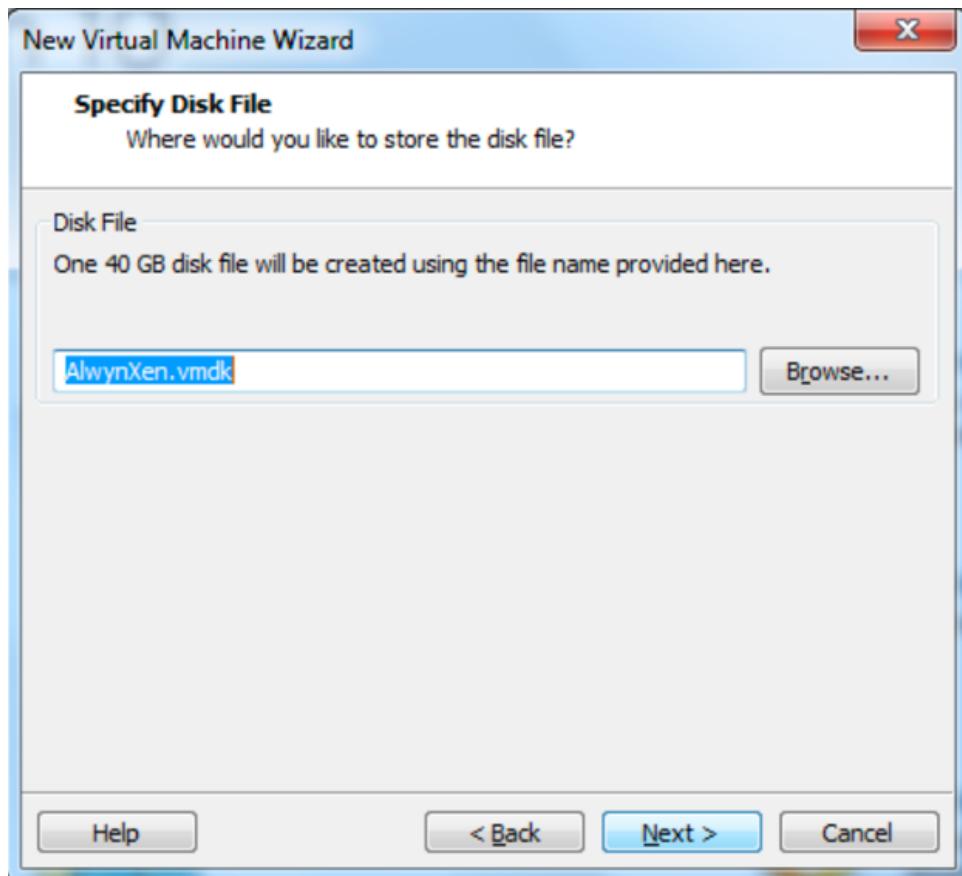


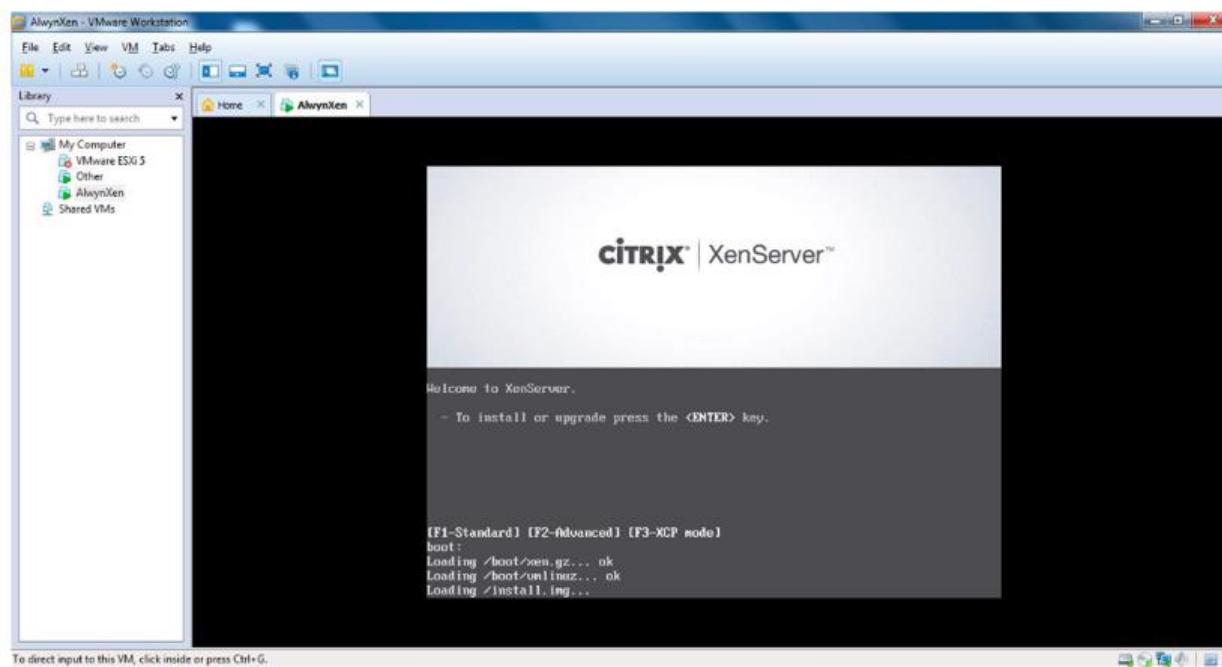
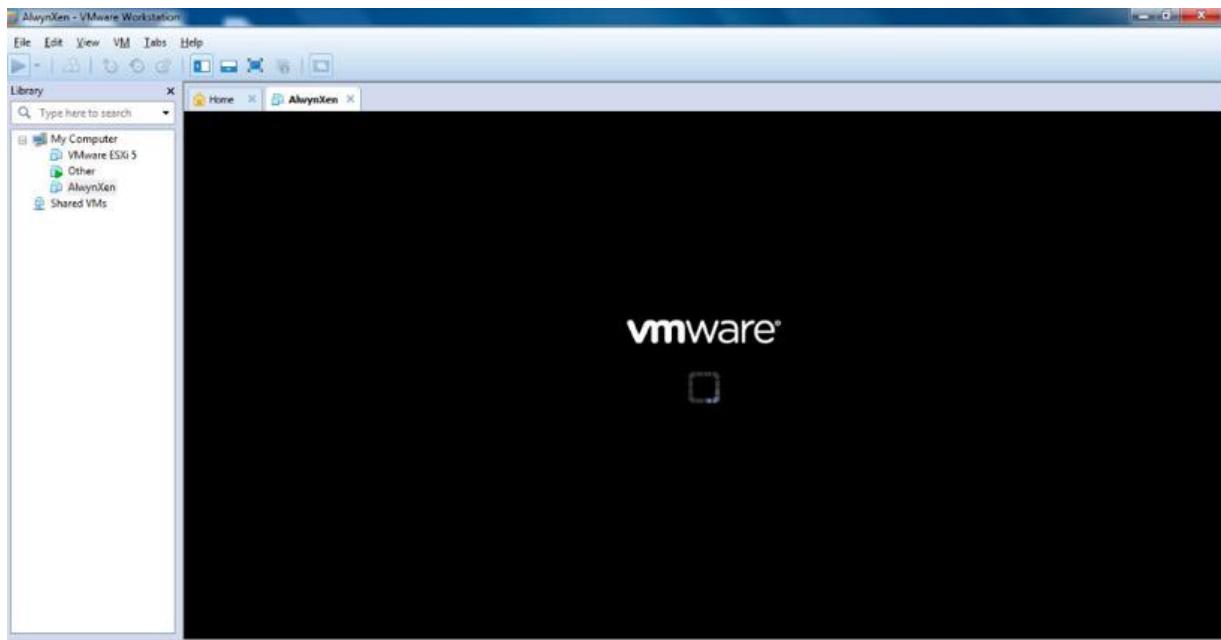


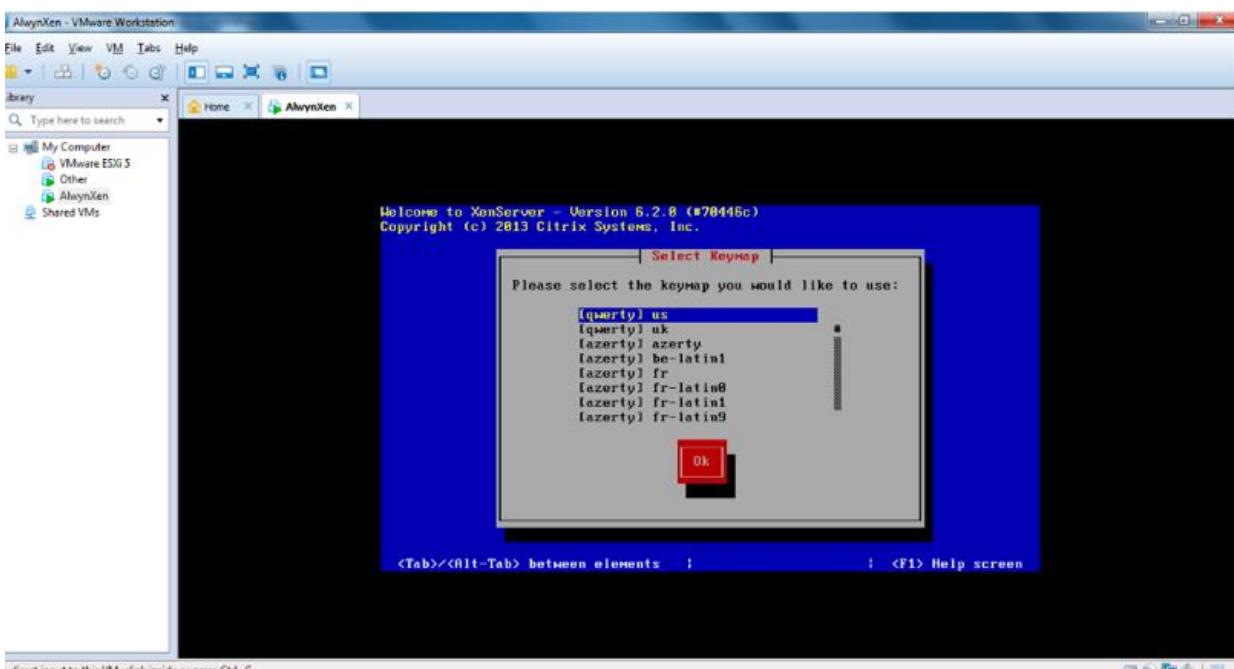
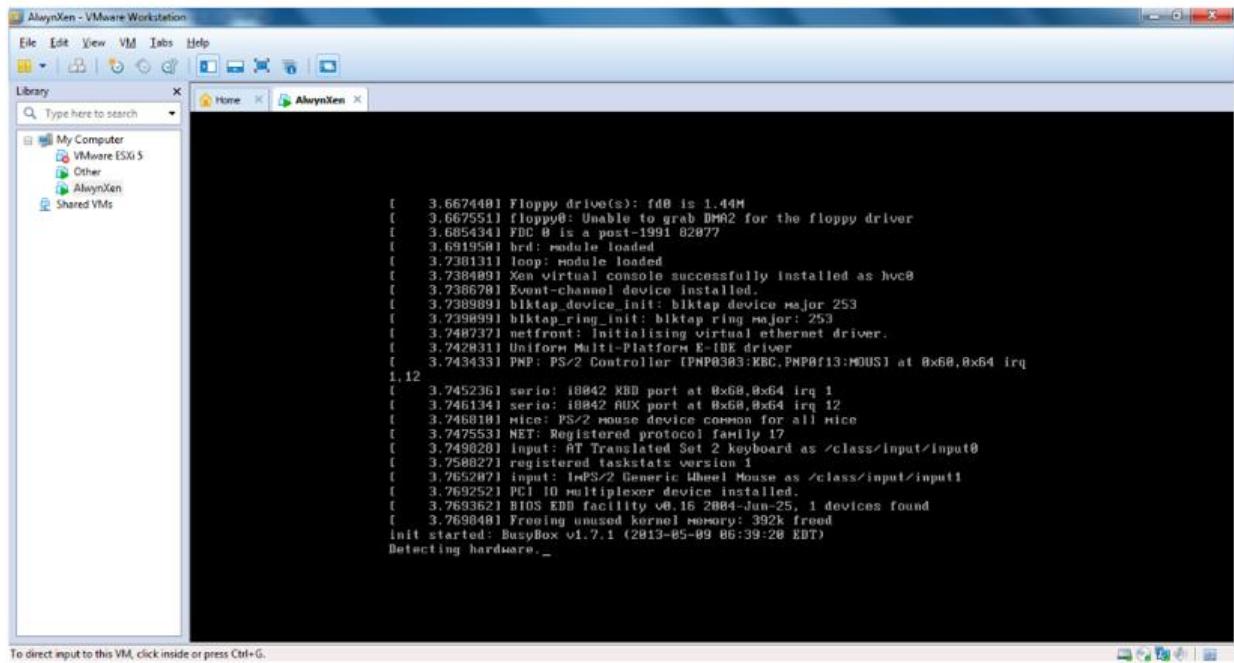












Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



<Tab>/<Alt-Tab> between elements : <F9> load driver : <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

End User License Agreement

CITRIX(R) LICENSE AGREEMENT

This is a legal agreement ("AGREEMENT") between you, the Licensed User, and Citrix Systems, Inc., Citrix Systems International GmbH, or Citrix Systems Asia Pacific Pty Ltd. Your location of receipt of this product or feature release (both hereinafter "PRODUCT") or technical support (hereinafter "SUPPORT") determines the providing entity hereunder (the applicable entity is hereinafter referred to as "CITRIX"). Citrix Systems, Inc., a Delaware corporation, licenses this PRODUCT in the Americas and Japan and

Accept EULA

Back

<Tab>/<Alt-Tab> between elements : <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

| System Hardware |

The following problem(s) were found with your hardware:

Hardware virtualization assist support is not available on this system. Either it is not present, or is disabled in the system's BIOS. This capability is required to start Windows virtual machines.

You may continue with the installation, though XenServer might have limited functionality until you have addressed these problems.

Ok

Back

<Tab>/<Alt-Tab> between elements :

: <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

| Virtual Machine Storage |

Which disks would you like to use for Virtual Machine storage? One storage repository will be created that spans the selected disks. You can choose not to prepare any storage if you wish to create an advanced configuration after installation.

[*] sda - 40 GB [ATA VMware Virtual I]

[] Enable thin provisioning (Optimized storage for XenDesktop)

Ok

Back

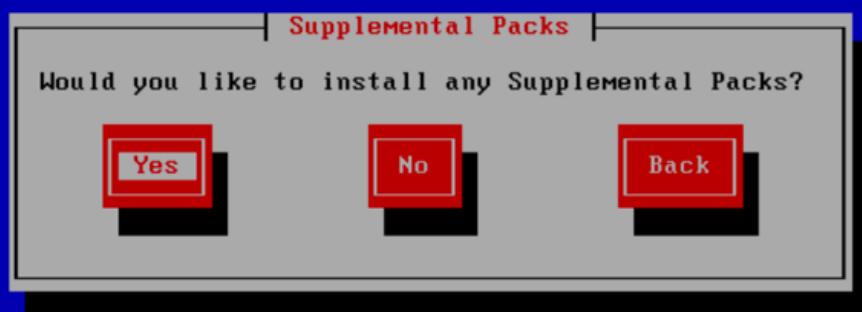
<Tab>/<Alt-Tab> between elements : <F5> More info : <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

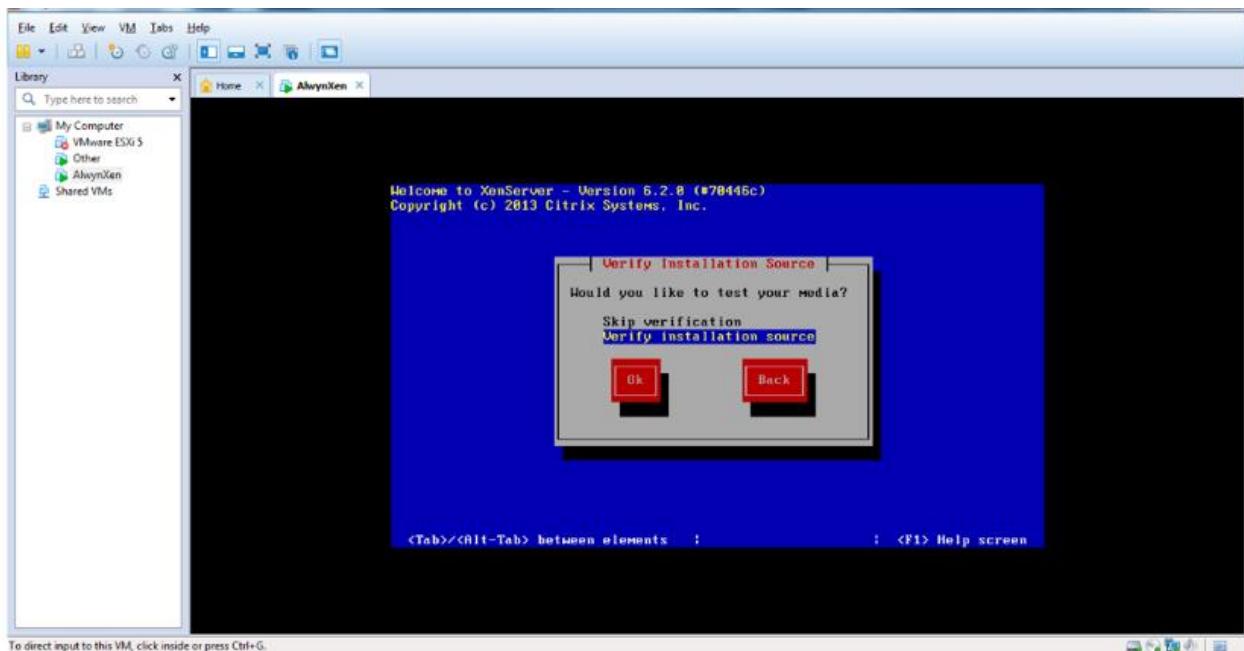


<Tab>/<Alt-Tab> between elements : ; <F1> Help screen

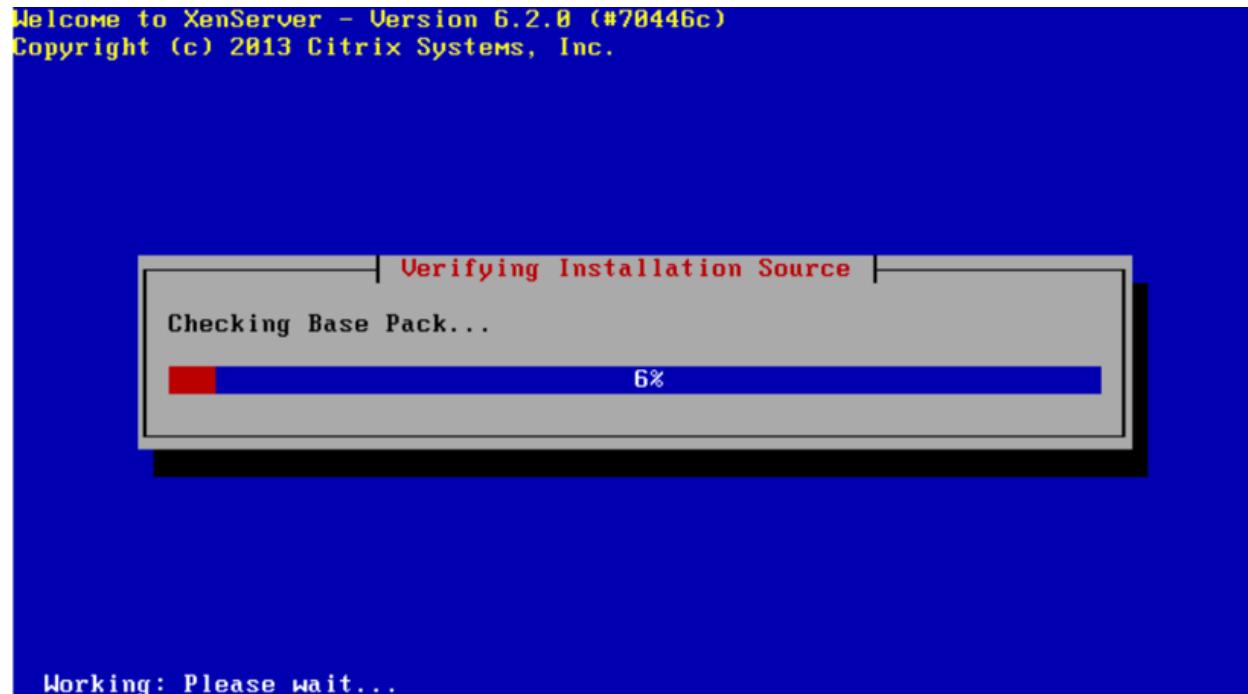
Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



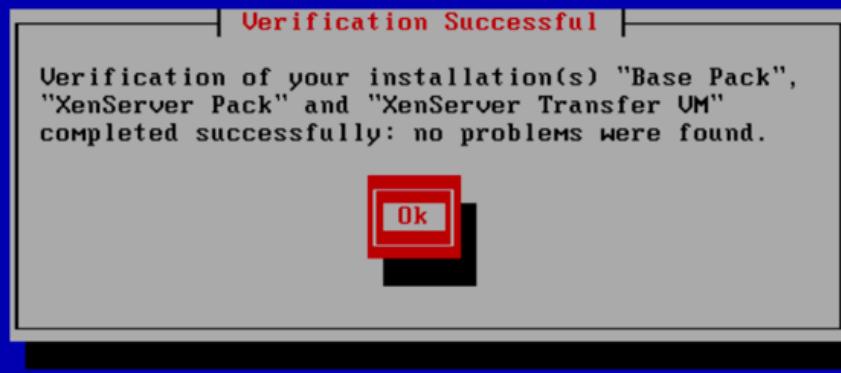
<Tab>/<Alt-Tab> between elements : ; <F1> Help screen



Welcome to XenServer – Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



<Tab>/<Alt-Tab> between elements

! <F1> Help screen

Copyright (c) 2013 Citrix Systems, Inc.

Networking

Please specify how networking should be configured for the management interface on this host.

(*) Automatic configuration (DHCP)
() Static configuration:

IP Address: _____
Subnet Mask: _____
Gateway: _____

Ok **Back**

<Tab>/<Alt-Tab> between elements : :

: <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)

Copyright (c) 2012 Citrix Systems, Inc.

Hostname and DNS Configuration

Hostname Configuration

(*) Automatically set via DHCP
() Manually specify:
xenserver-cwbyhqd1 _____

DNS Configuration

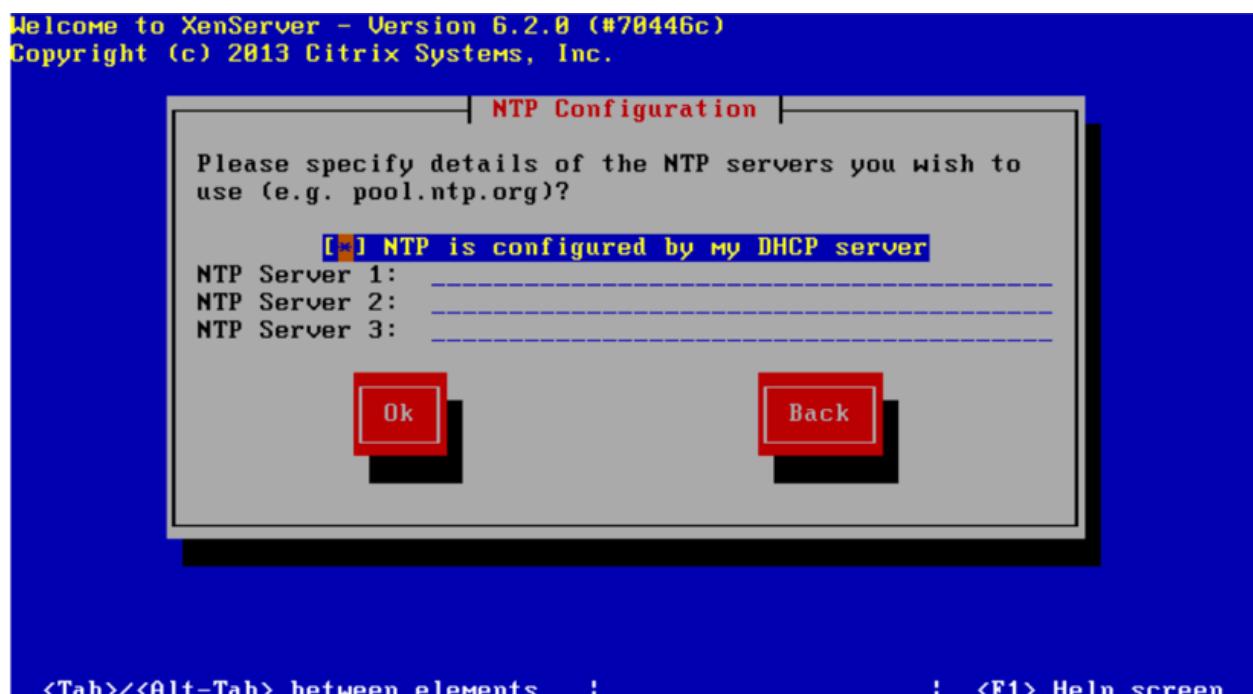
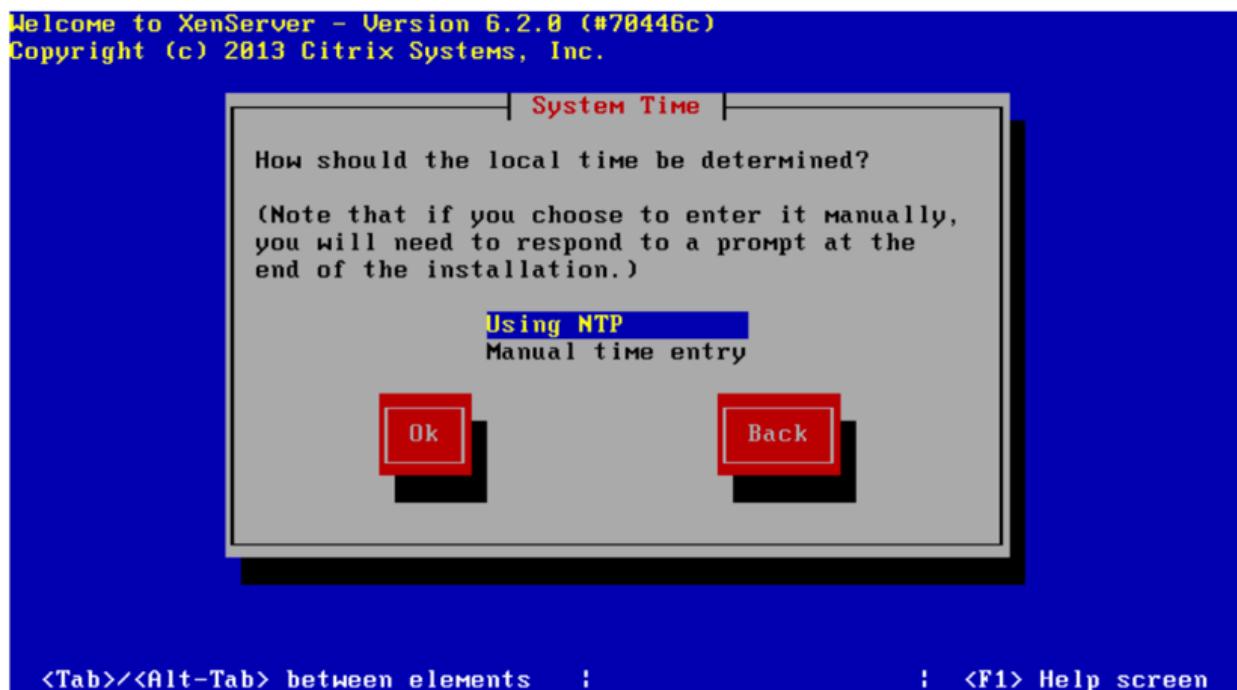
(*) Automatically set via DHCP
() Manually specify:
DNS Server 1: _____
DNS Server 2: _____
DNS Server 3: _____

Ok **Back**

<Tab>/<Alt-Tab> between elements : :

: <F1> Help screen





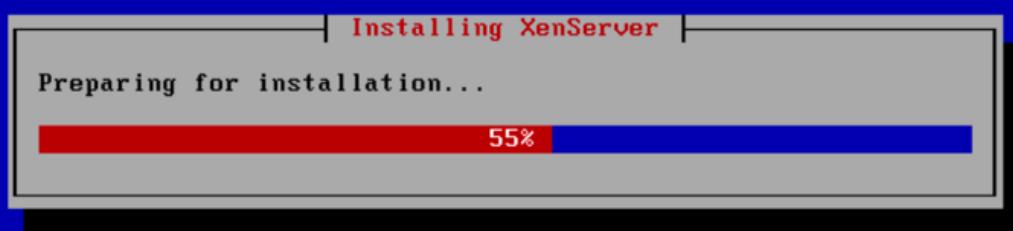
Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



<Tab>/<Alt-Tab> between elements

: <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.



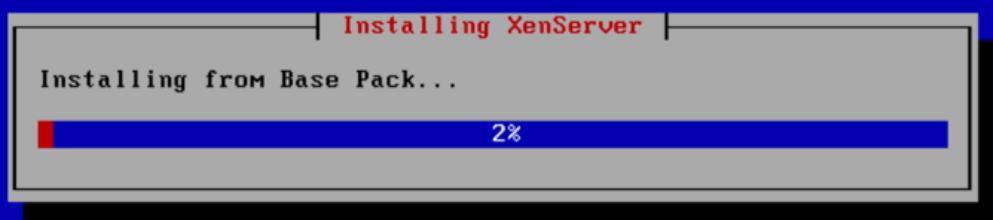
Working: Please wait...

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

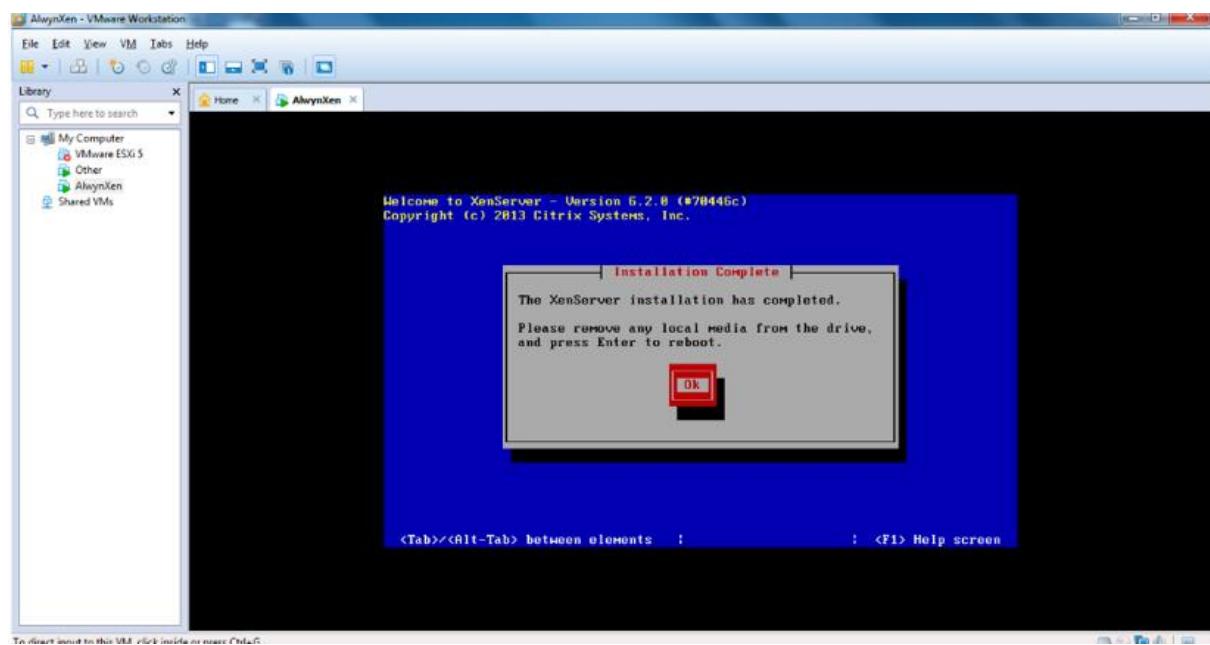
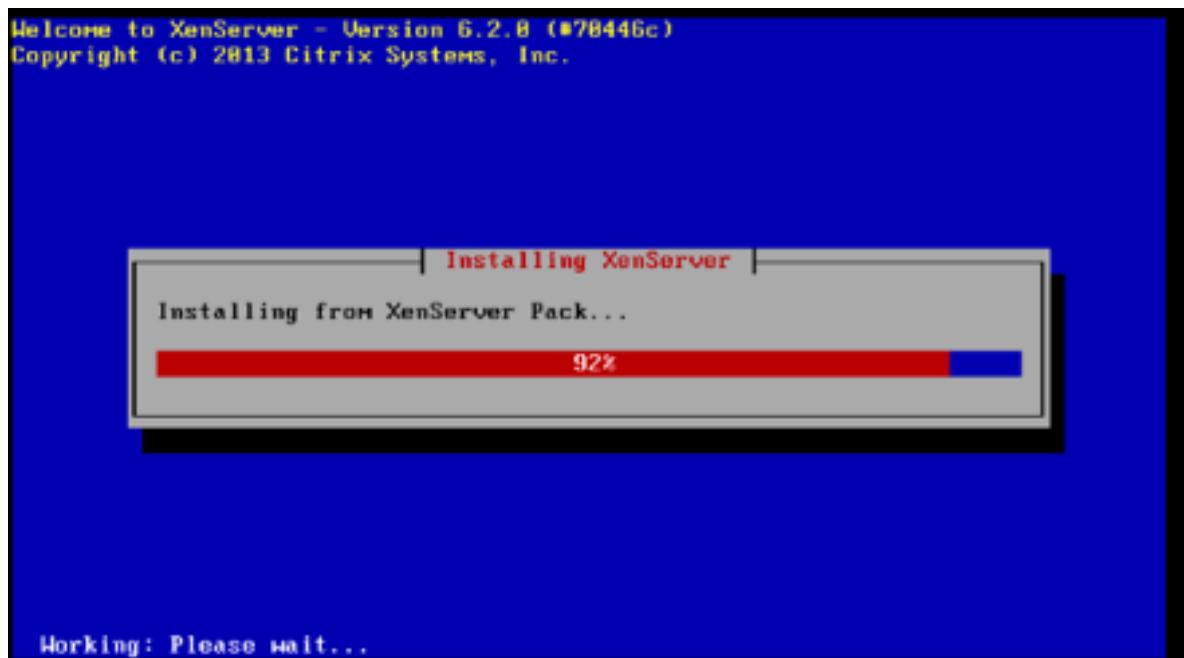


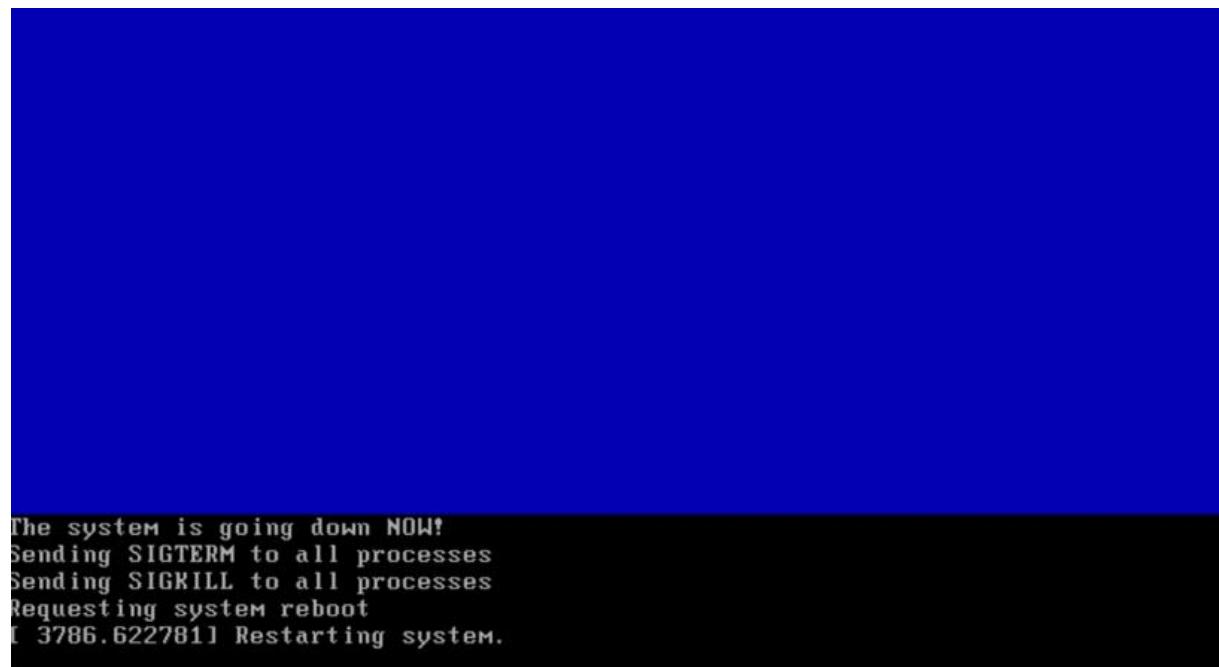
<Tab>/<Shift-Tab> between elements : | <F1> Help screen

Welcome to XenServer - Version 6.2.0 (#70446c)
Copyright (c) 2013 Citrix Systems, Inc.

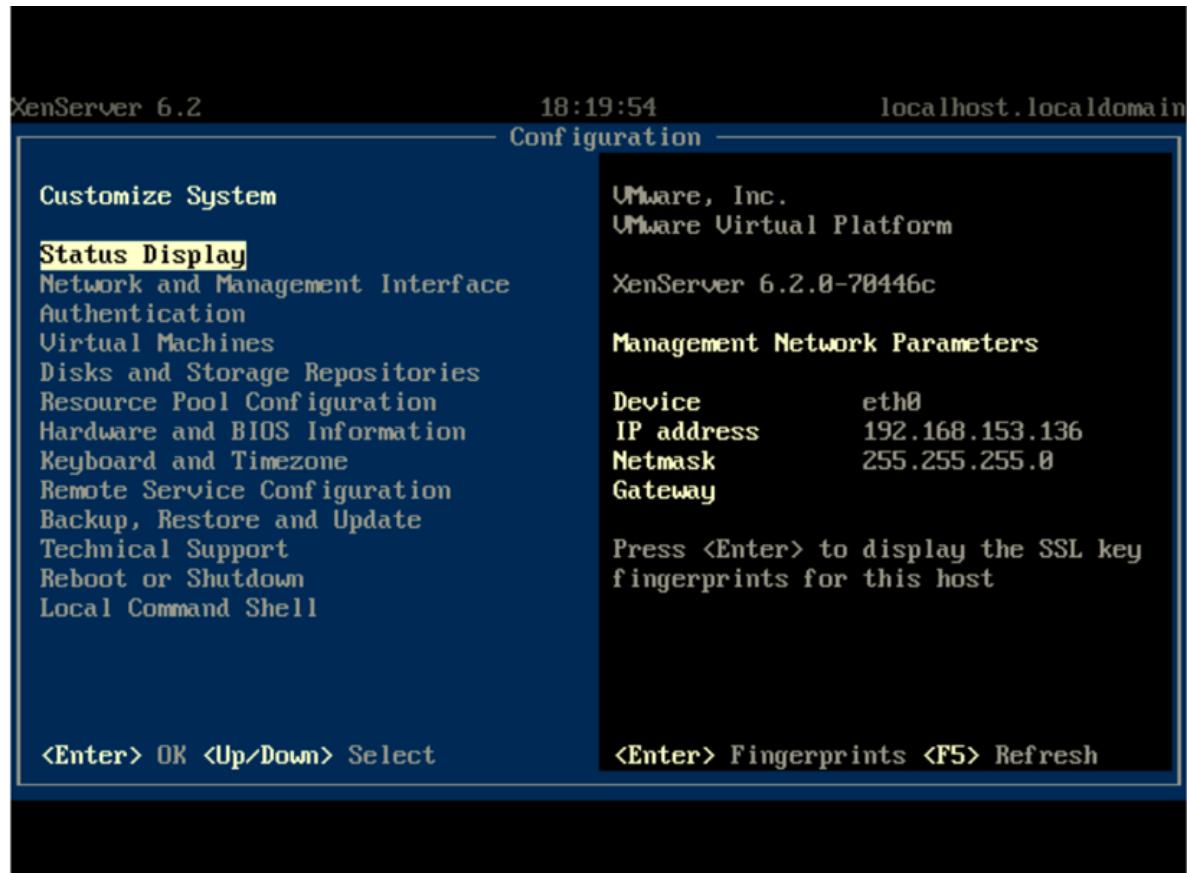


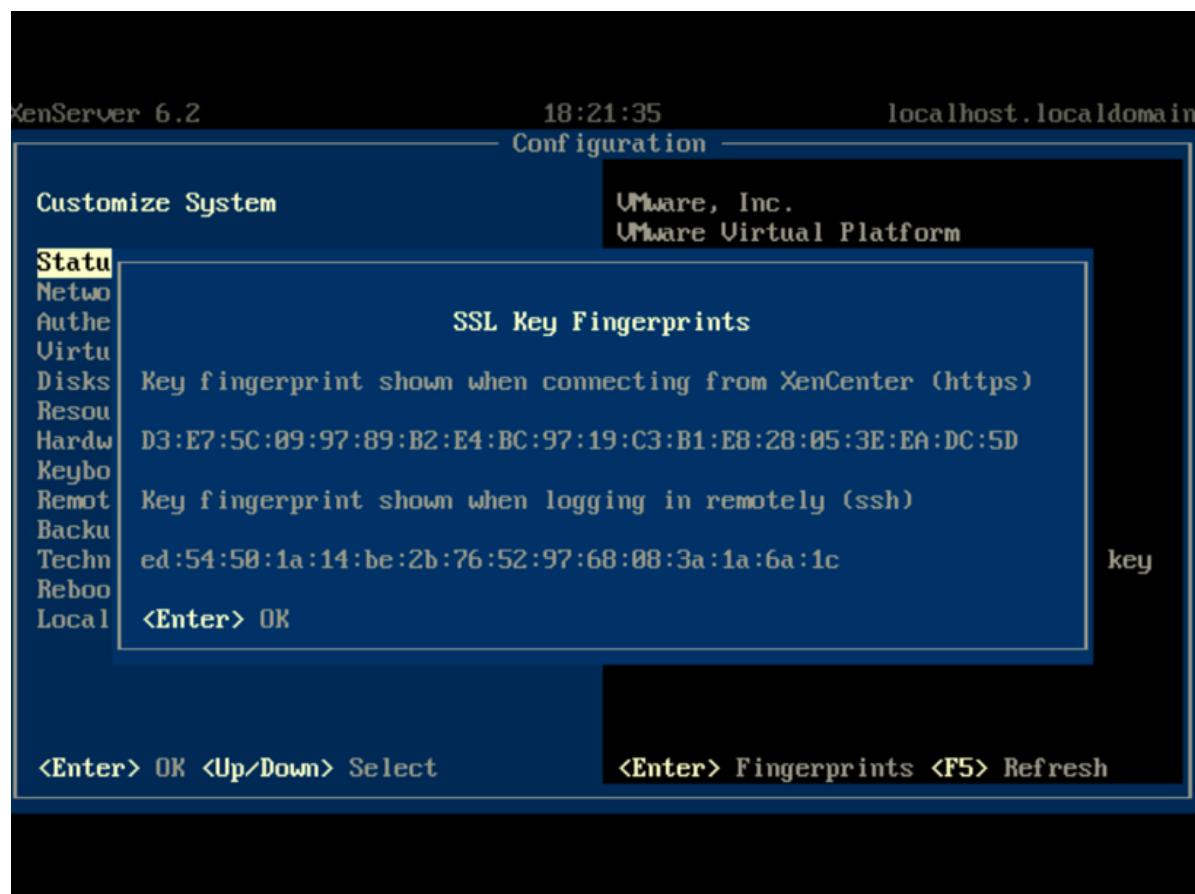
Working: Please wait...

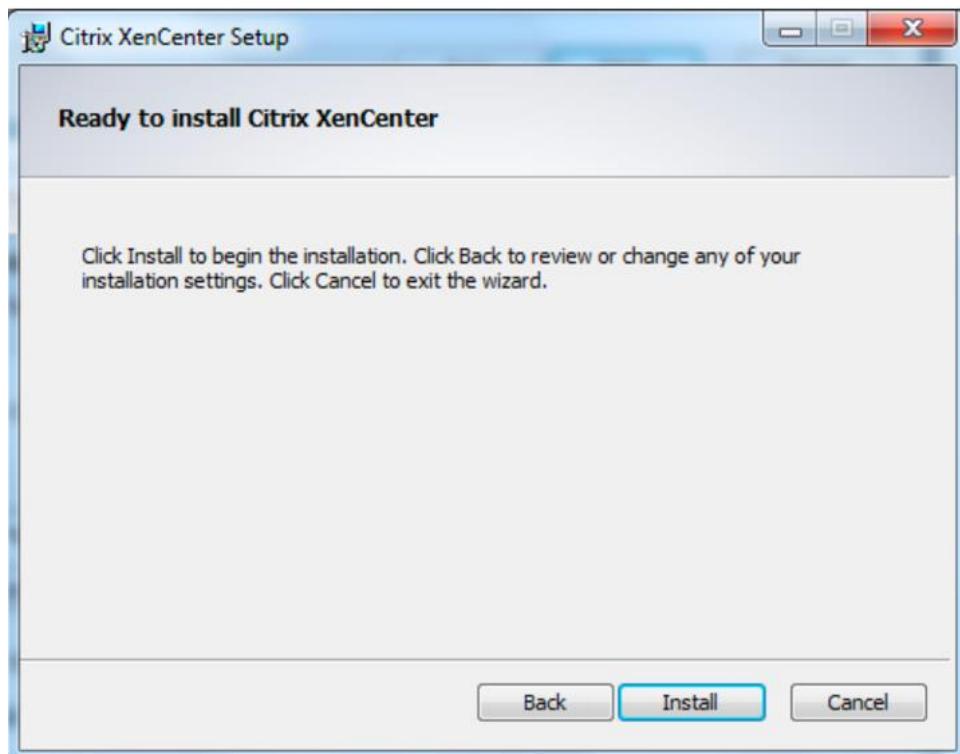
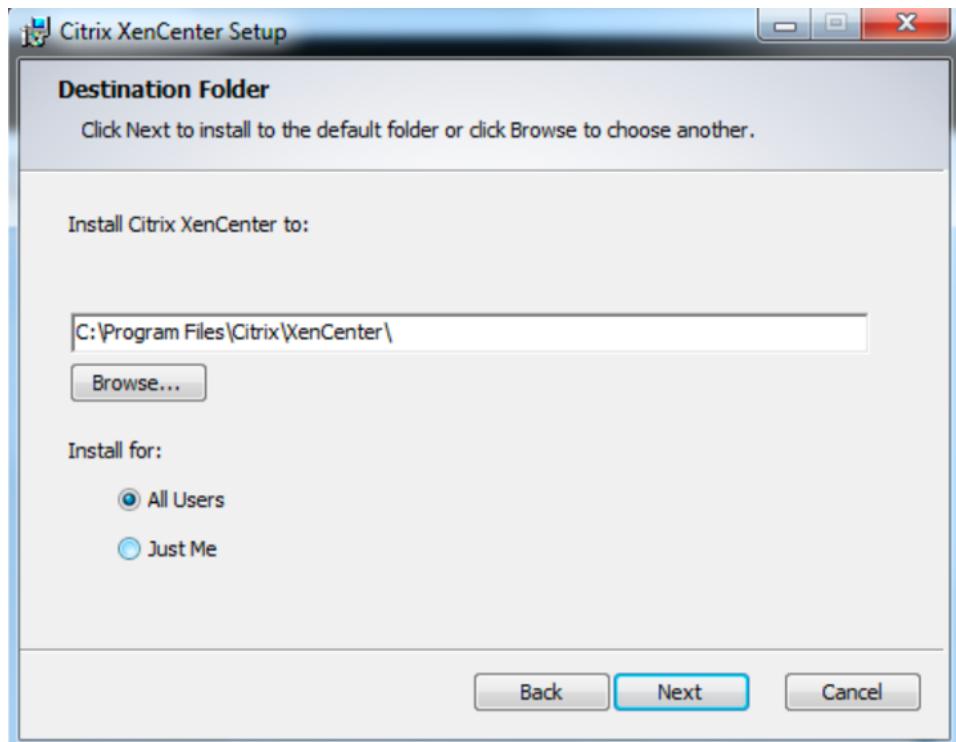


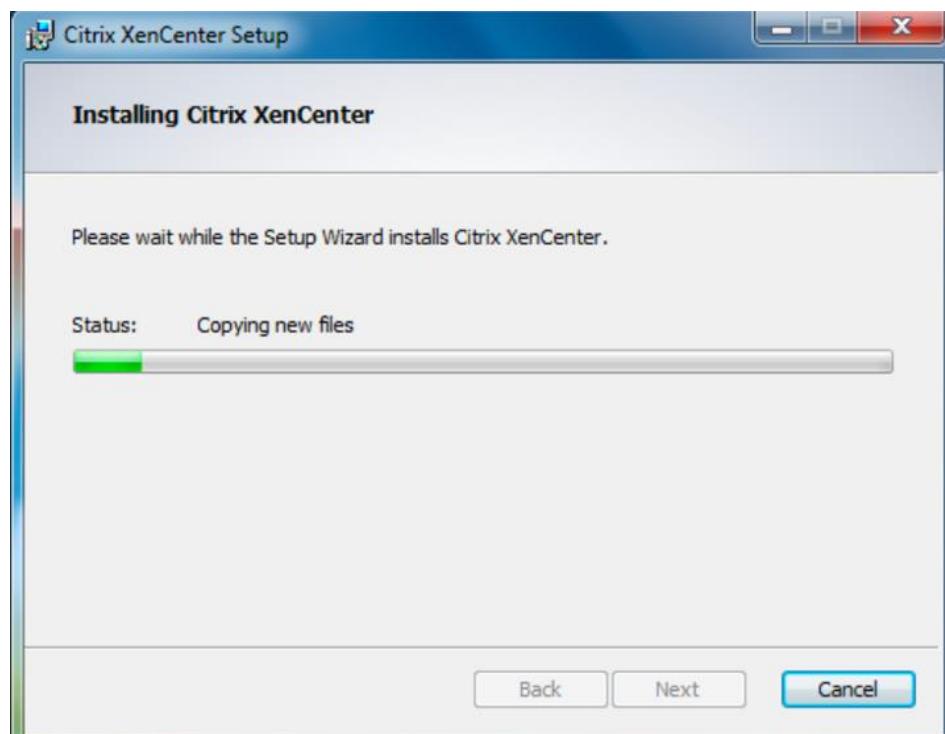
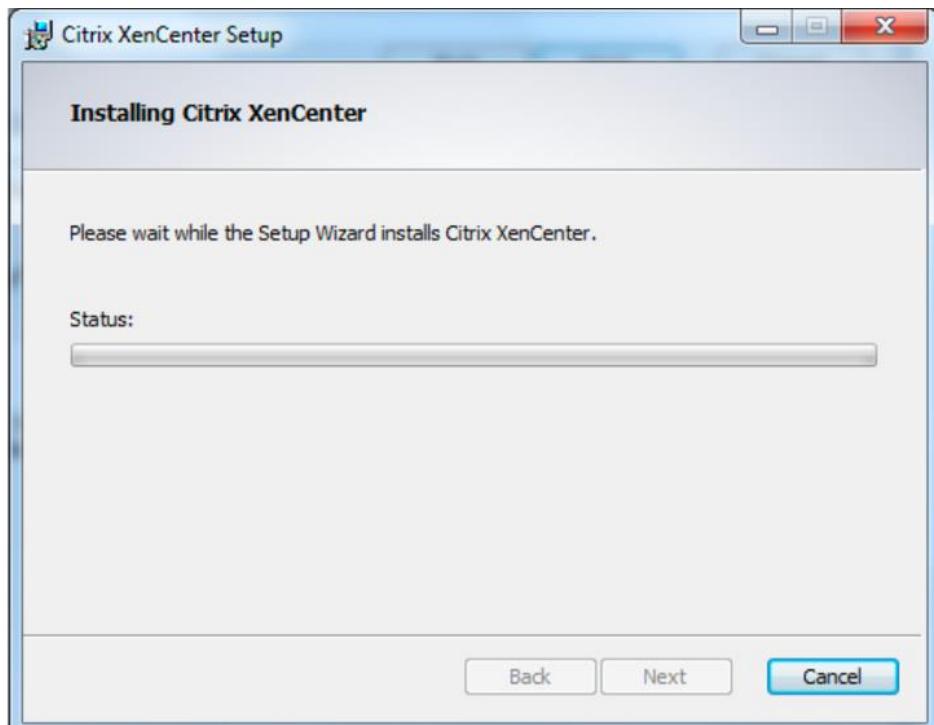


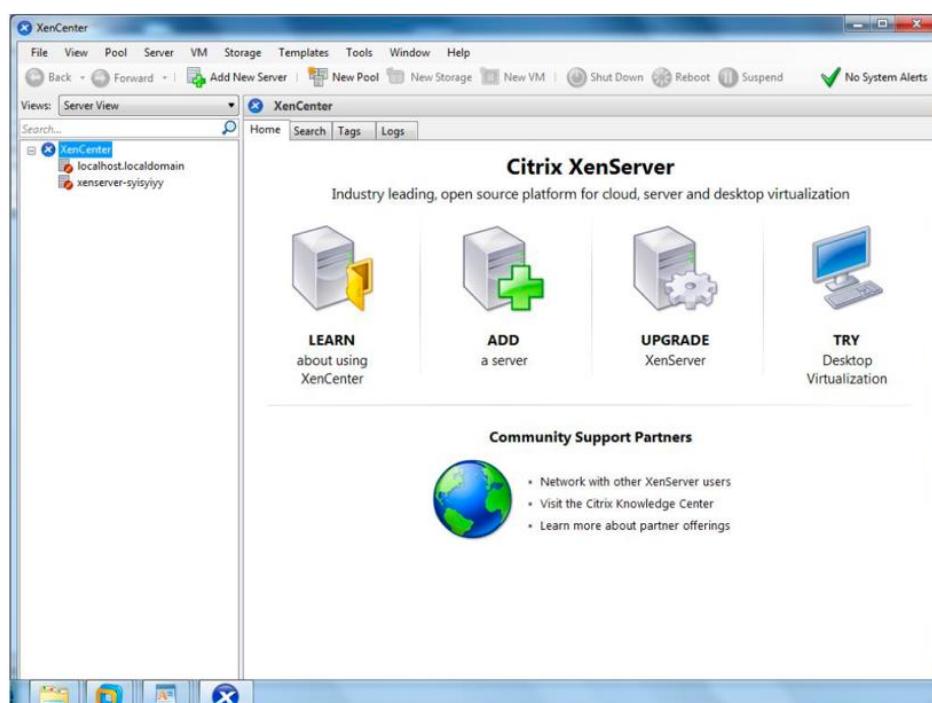
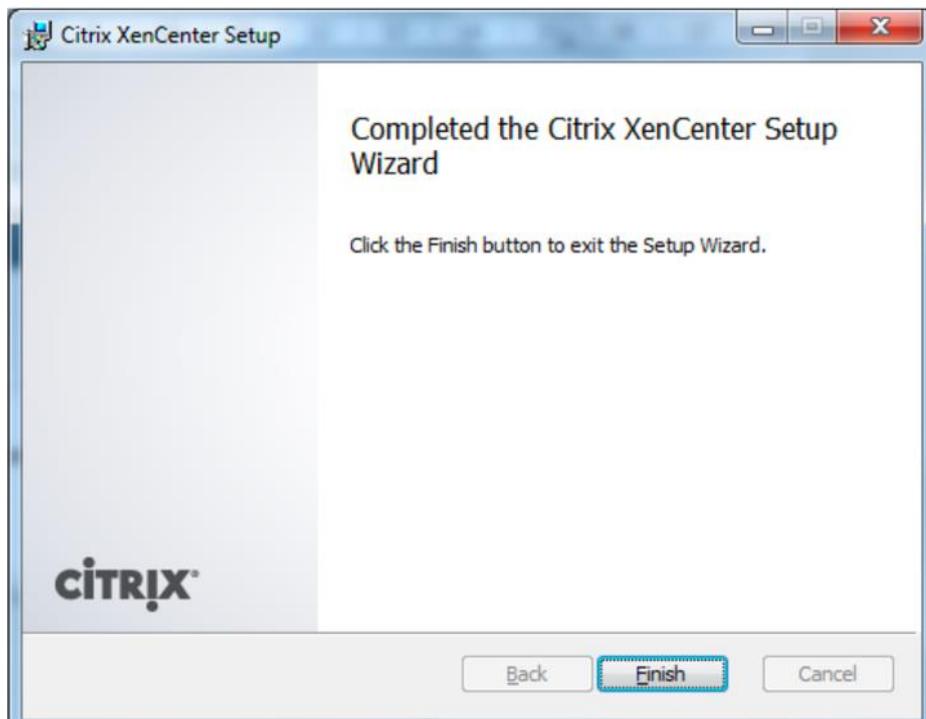
```
(XEN) Processor #0 7:7 APIC version 21
(XEN) Processor #1 7:7 APIC version 21
(XEN) IOAPIC[0]: apic_id 2, version 17, address 0xfec00000, GSI 0-23
(XEN) Enabling APIC mode: Flat. Using 1 I/O APICs
(XEN) Not enabling x2APIC: depends on iommu_supports_eim.
(XEN) Using scheduler: SMP Credit Scheduler (credit)
(XEN) Detected 2925.395 MHz processor.
(XEN) Initiating memory sharing.
(XEN) I/O virtualisation disabled
(XEN) ENABLING IO-APIC IRQs
(XEN) -> Using new ACK method
(XEN) Platform timer is 14.318MHz HPET.
(XEN) Allocated console ring of 64 KiB.
(XEN) Brought up 2 CPUs
(XEN) Testing NMI watchdog --- CPU#0 okay. CPU#1 okay.
(XEN) *** LOADING DOMAIN 0 ***
(XEN) Xen kernel: 64-bit, lsb, compat32
(XEN) Dom0 kernel: 32-bit, PAE, lsb, paddr 0x100000 -> 0x787000
(XEN) PHYSICAL MEMORY ARRANGEMENT:
(XEN) Dom0 alloc.: 000000003d000000->000000003e000000 (186355 pages to be allocated)
(XEN) Init. ramdisk: 000000003f1f3000->000000003f9ff600
(XEN) VIRTUAL MEMORY ARRANGEMENT:
(XEN) Loaded kernel: 00000000c0100000->000000000c0787000
(XEN) Init. ramdisk: 00000000c0787000->000000000c0f93600
(XEN) Phys-Mach map: 00000000c0f94000->000000000c1050000
(XEN) Start info: 00000000c1050000->000000000c10504b4
(XEN) Page tables: 00000000c1051000->000000000c1061000
(XEN) Boot stack: 00000000c1061000->000000000c1062000
(XEN) TOTAL: 00000000c0000000->000000000c1400000
(XEN) ENTRY ADDRESS: 00000000c0100000
(XEN) Dom0 has maximum 2 UCPUs
(XEN) Scrubbing Free RAM: .
```

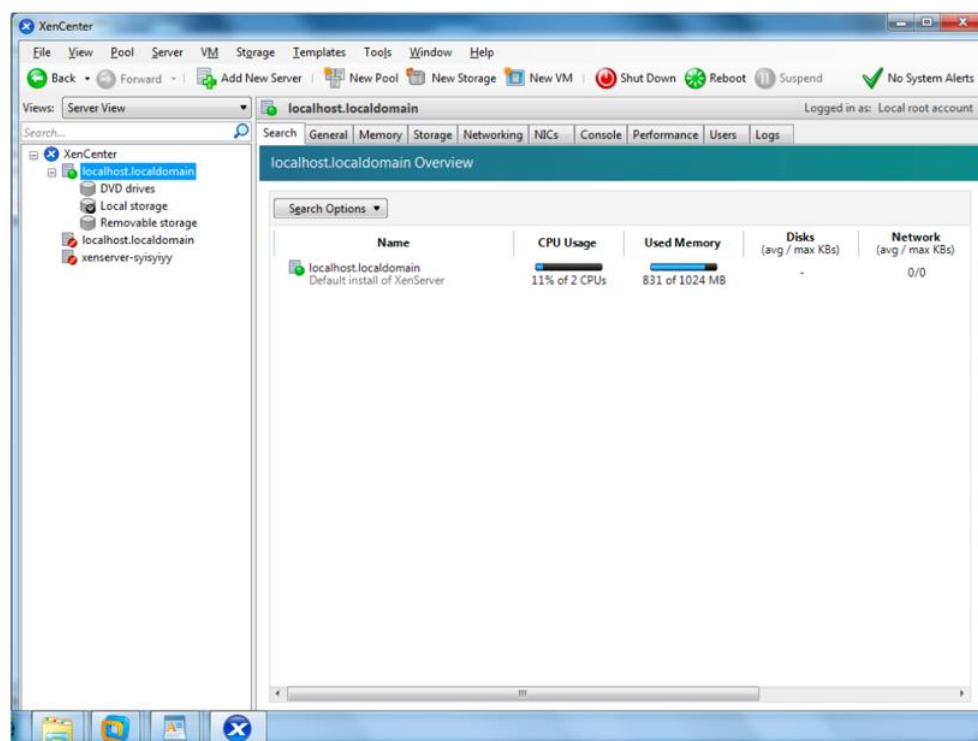
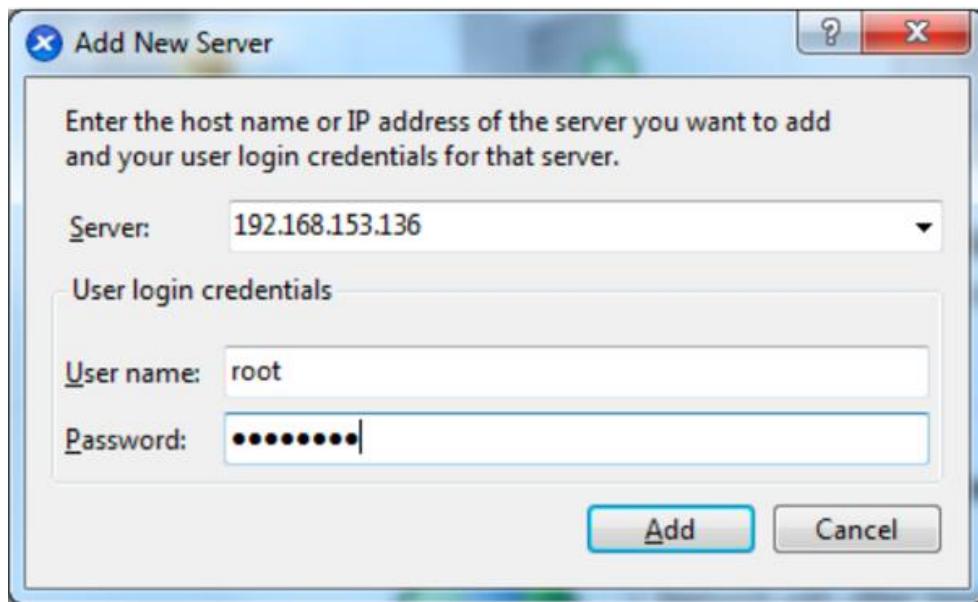






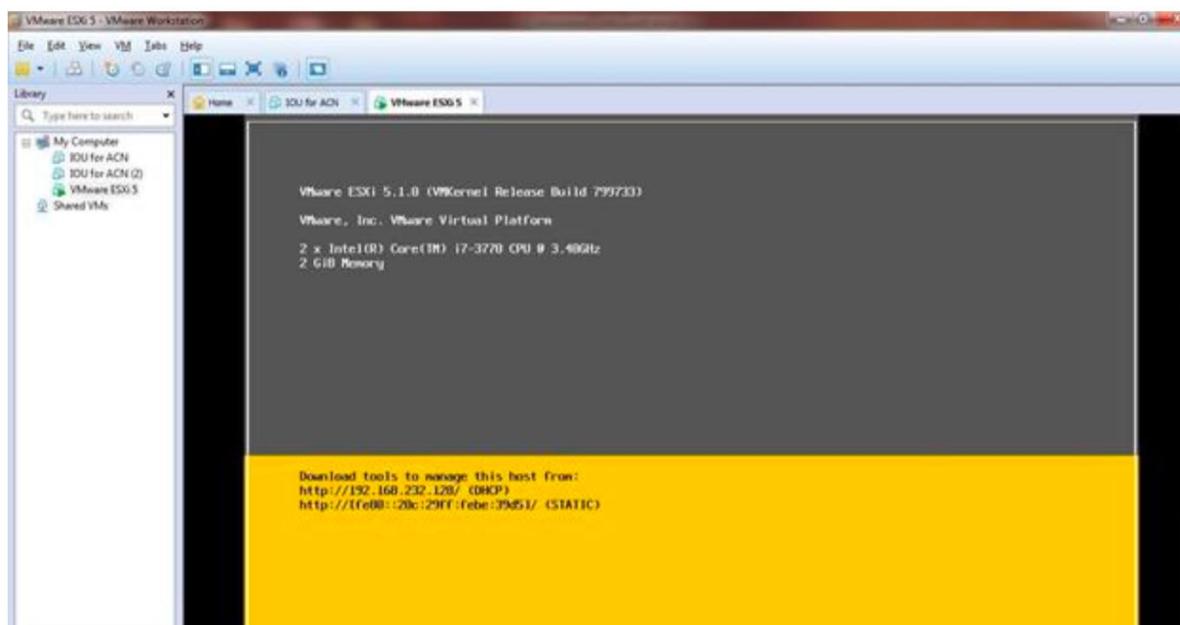






Practical: 06

1. Aim: Implement virtualization using VMWare ESXi Server and managing with vCenter



Windows XP Professional Setup

Welcome to Setup.

This portion of the Setup program prepares Microsoft® Windows® XP to run on your computer.

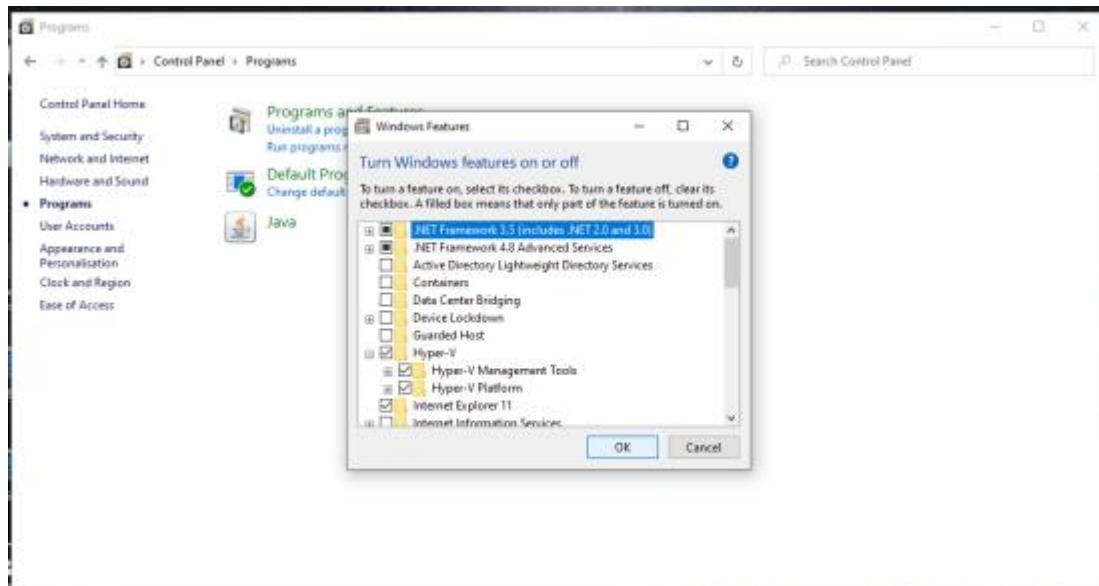
- To set up Windows XP now, press ENTER.
- To repair a Windows XP installation using Recovery Console, press R.
- To quit Setup without installing Windows XP, press F3.

Practical 07:

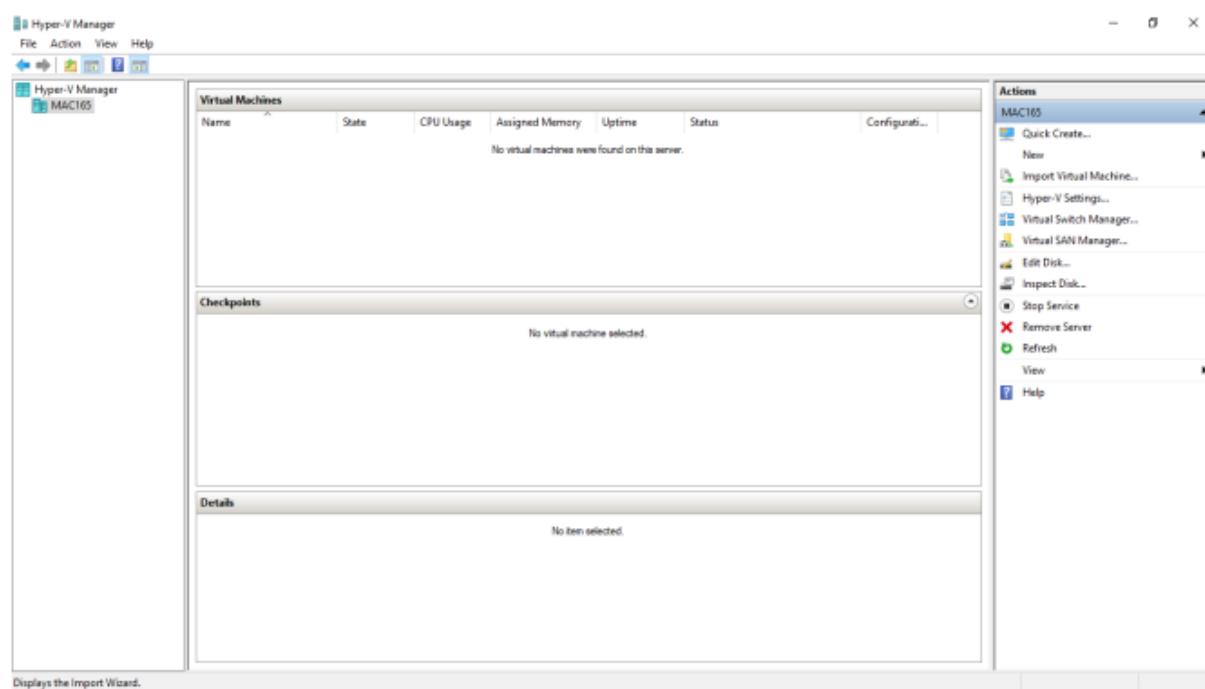
Aim: Implement windows Hyper V virtualization

Step 1 Uninstall Virtual Machine within the system

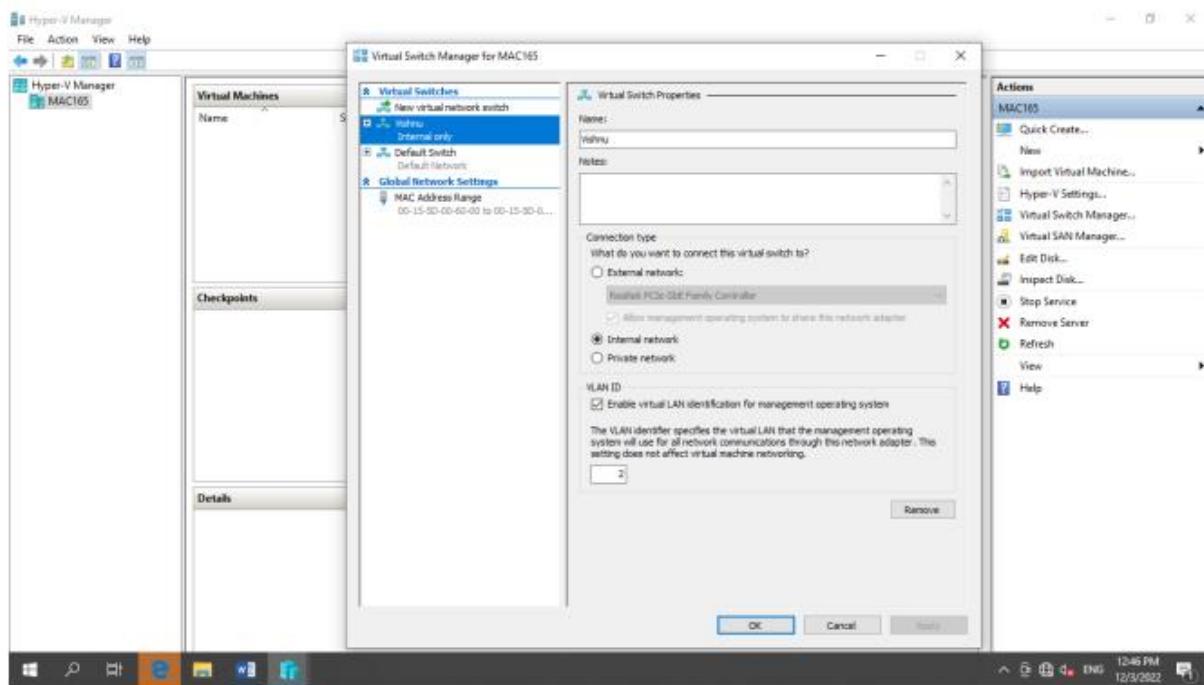
. Step 2 Enable Hyper V in the system by going in control panel.



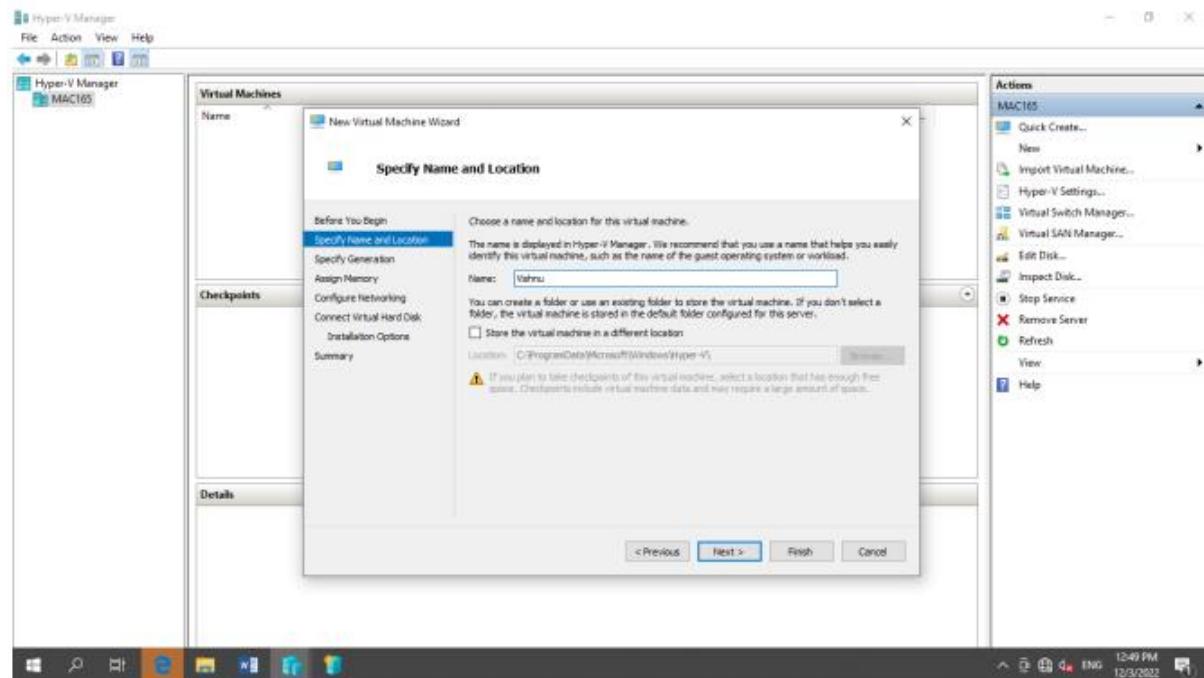
Step 3 Open Hyper V manager



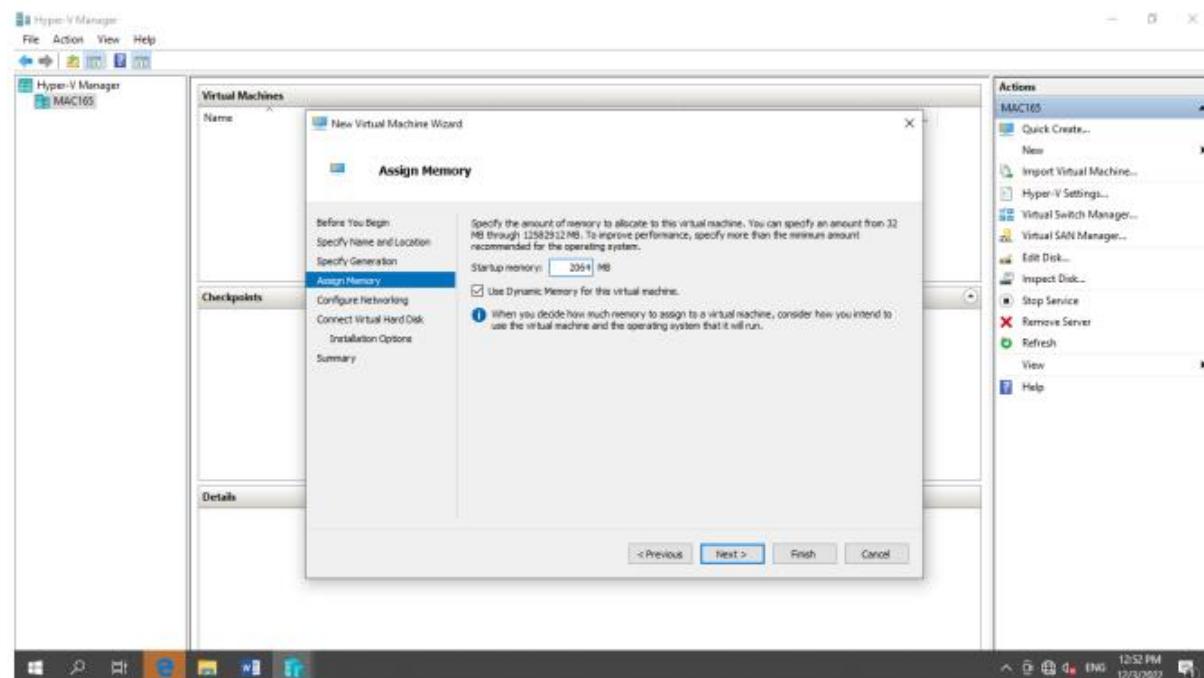
Step 4 Create a Virtual Switch by going in Virtual Switch Manager



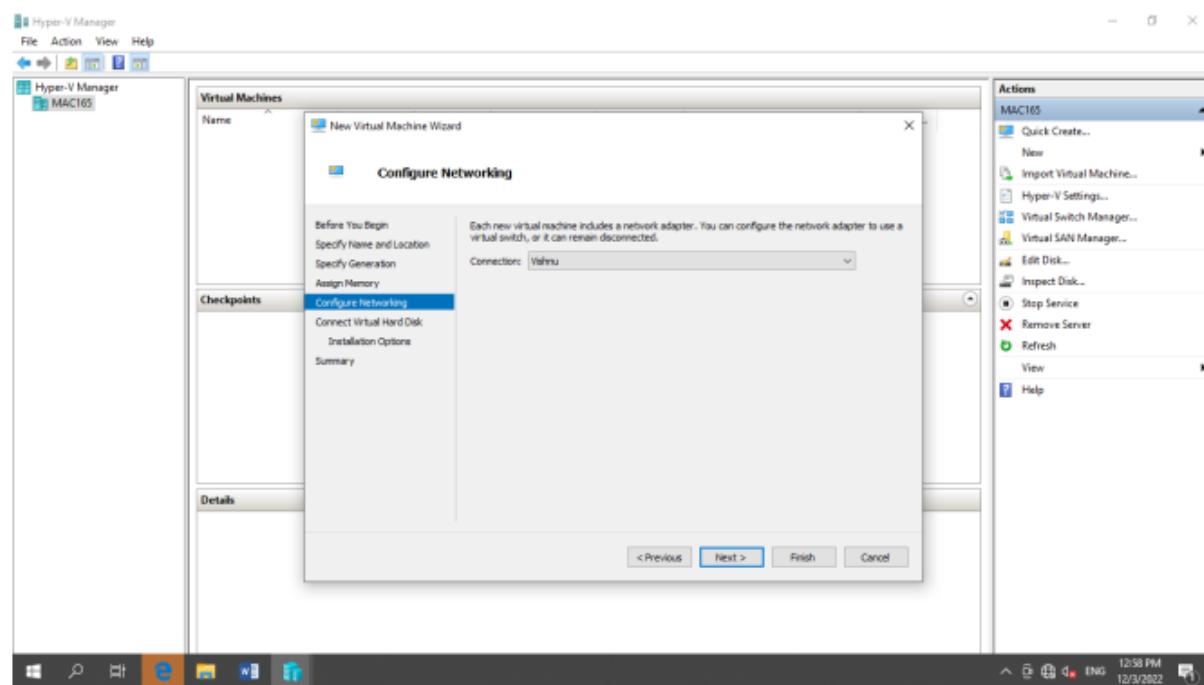
Step 5 Create Virtual Machine by clicking on new



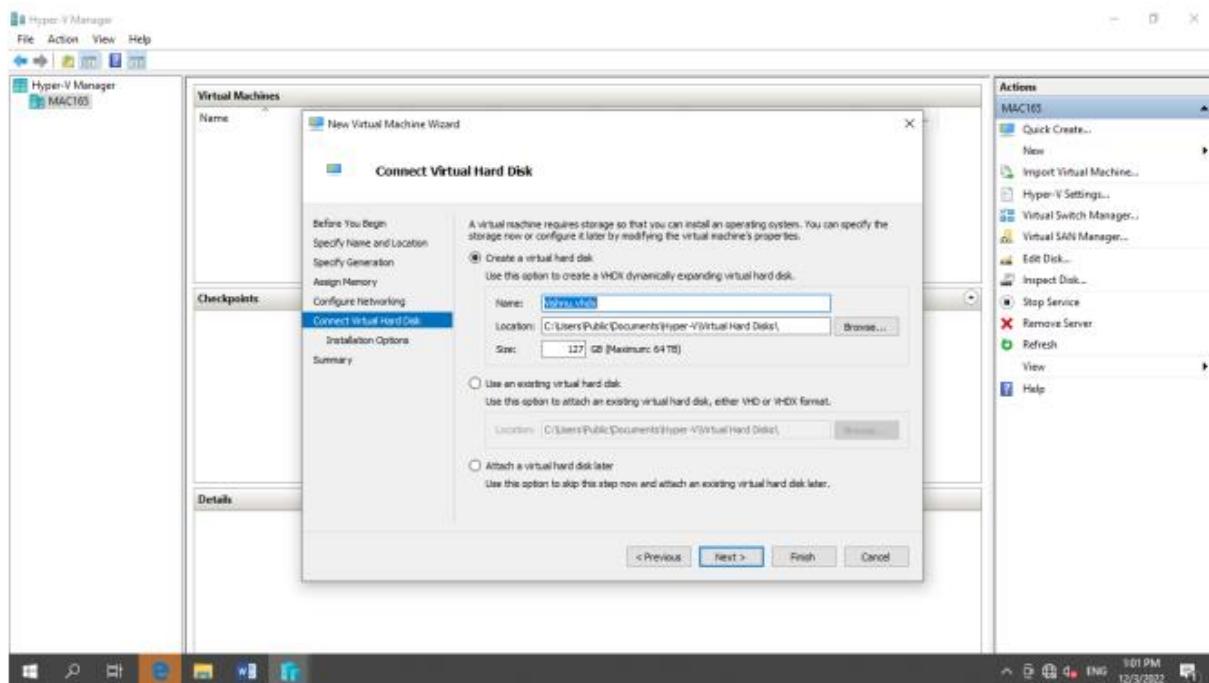
Step 6 Assign Memory



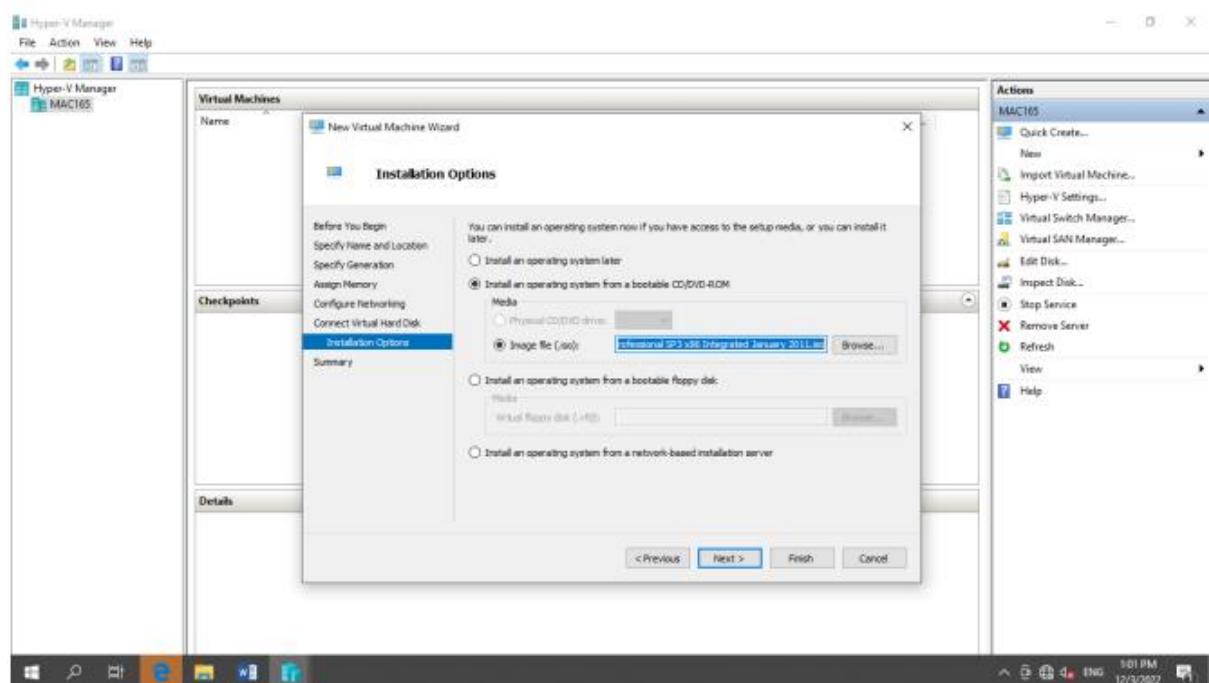
Step 7 Assign the Network using the Switch created



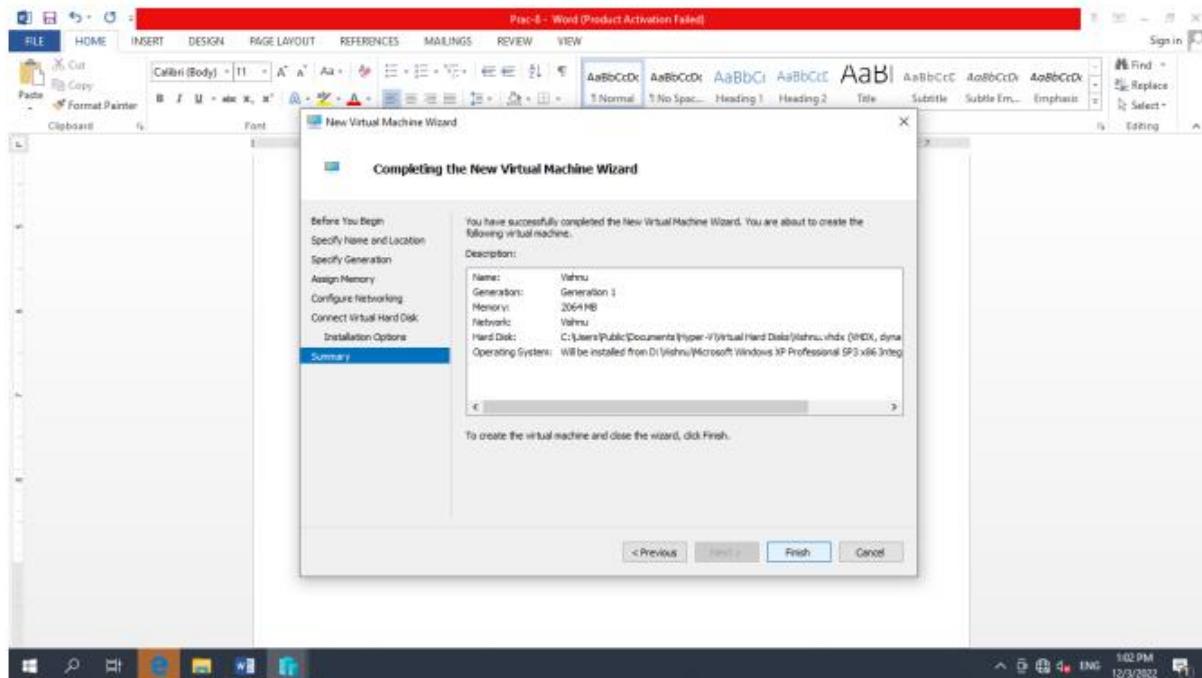
Step 8 Assign Disk Memory



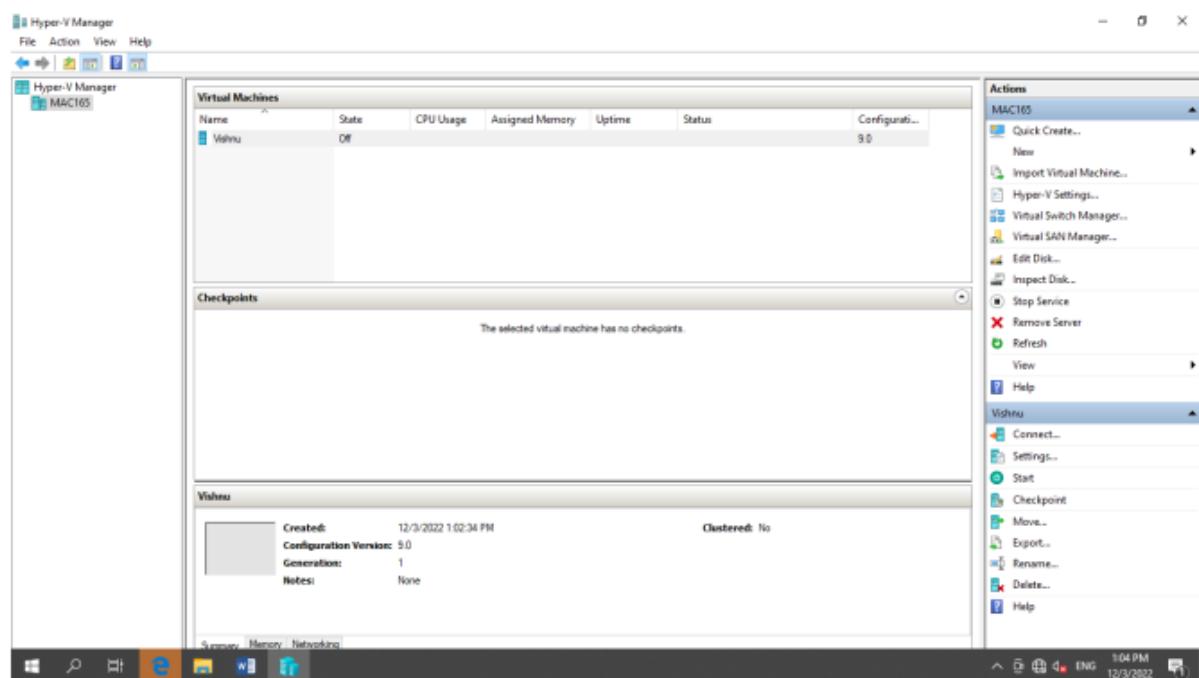
Step 9 Install ISO file of particular OS you want to install



Step 10 Click on Finish



Step 11 Virtual Machine is created



Step 12 Switch on the Virtual Machine

