

**CLOUD COMPUTING LAB****B20EJ0604****6th Semester****B.Tech Computing and Information Technology****SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

Name	
SRN	
Branch	
Semester	
Section	
Academic Year	

Learning Objectives of the Course:

The objectives of this course are to:

1. Provide the knowledge of Map Reduce framework in solving problems related to big data.
2. Introduce Google Docs, Google Sheets, Google Slides and Google Forms.
3. Provide hands on experience on cloud environments.
4. Introduce IOT applications using Cloud.

Learning Outcomes of the Course:

Upon successful completion of the course, students should be able to:

1. Execute MapReduce programs on Hadoop and analyze the results.
2. Create documents, spreadsheets, slides and forms using Google Docs, Google Sheets, Google Slides and Google Forms.
3. Conduct some simple experiments on VMware products.
4. Carry out IOT applications using Cloud.

CONTENTS		
Sl. No.	Lab Sessions	Page No.
1	Evaluate the performance of MapReduce program on word count for different file size.	3-7
2	Design a document, excel sheet, presentation and form using Google Docs, Google sheets, Google Slides and Google Forms.	8-11
3	Create multiple VM's on a single physical machine. Provide a communication between them using and without using vClient.	12-22
4	Demonstrate the communication of VM's on different physical devices using ESXi.	23-24
5	Illustrate the cloning of VM's.	25-35
6	Illustrate the backup-restore scenario.	37-42
7	Add iSCSI adapter and modify the relative parameters. Create a new VMkernel using iSCSI port binding.	43-46
8	Performance Analysis of Virtual Machine vs Physical Machine.	47-50
9	Design a module to control an LED from Webserver using NodeMcu or Esp8266 programming with Arduino IDE.	51-57
Additional Sessions		
1	Create a data center using vCenter for multiple VM's.	58-61
2	Install and configure the vSphere Web client	62
3	Illustration of vMotion to move the VM's from one ESXi to another ESXi	63-64
4	Design and Demonstrate a program using CloudSim, showing how to create a datacenter with one host and run one cloudlet on it.	65-75
5	Design and Demonstrate using CloudSim, showing how to create two datacenters with one host each and run cloudlets of two users on them.	76-86
Appendix		
A	Install and Configure Hadoop.	87-95
B	Install and Configure VMware vSphere ESXi .	96-114

C	Installation of VMware vSphere Client.	115-119
	Assignment: Use MapReduce to implement K-Means Clustering for a real world big data clustering.	

Evaluation Sheet

Name of the Faculty In charge:

Program/ Experiment No	Title of the Program /Experiment	Date of Execution	Max Marks	Marks Obtained	Signature of the Faculty I/c
1					
2					
3					
4					
5					
Internal Assessment-1 (IA-1) Consolidated Marks					

Program/ Experiment No	Title of the Program /Experiment	Date of Execution	Max Marks	Marks Obtained	Signature of the Faculty I/c
6					
7					
8					
9					
Additional Session (1-5)					
Internal Assessment-2 (IA-2) Consolidated Marks					
		IA	Max Marks	Marks Obtained	Signature of the Faculty
		IA1			
		IA2			
		IA1+IA2			

Signature of the Faculty-in-charge with date

Semester End Lab Examination Evaluation Procedure 2023-24

Name of the Lab: Cloud Computing Lab		
Question	Parameters to be considered	Marks distribution
1-9	Write-up	
	Conduction & Results	
	Viva	
	Total	

Note:**(a) Lab course is conducted for a total of 50 Marks:**

- i. 20 Marks –Continuous Evaluation
- ii. 30 Marks Semester End Examination

(b) Passing Criterion:

- i. 08/30 in Semester End Examination
- ii. 20/50 overall

Session 1	
Problem Statement:	Evaluate the performance of MapReduce program on word count for different file size.
Student Learning Outcomes:	Execute MapReduce programs on Hadoop and analyze the results.
Theoretical Description:	<p>MapReduce: A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system.</p>
Requirements:	<ul style="list-style-type: none">Ubuntu Linux 10.04 LTS (deprecated: 8.10 LTS, 8.04, 7.10, 7.04)Hadoop 1.0.3, released May 2012Hadoop requires a working Java 1.5+ (aka Java 5) installation. However, using Java 1.6 (aka Java 6) is recommended for running Hadoop.
Procedure:	<ol style="list-style-type: none">Download example input data. We will use three ebooks from Project Gutenberg for this example:<ul style="list-style-type: none">The Outline of Science, Vol. 1 (of 4) by J. Arthur ThomsonThe Notebooks of Leonardo Da VinciUlysses by James JoyceDownload each ebook as text files in Plain Text UTF-8 encoding and store the files in a local temporary directory of choice, for example /tmp/gutenberg. <pre>1 hduser@ubuntu:~\$ ls -l /tmp/gutenberg/ 2 total 3604 3 -rw-r--r-- 1 hduser hadoop 674566 Feb 3 10:17 pg20417.txt 4 -rw-r--r-- 1 hduser hadoop 1573112 Feb 3 10:18 pg4300.txt 5 -rw-r--r-- 1 hduser hadoop 1423801 Feb 3 10:18 pg5000.txt 6</pre>Restart the Hadoop cluster. <pre>hduser@ubuntu:~\$ /usr/local/hadoop/bin/start-all.sh</pre>

3. Copy local example data to HDFS.

Before we run the actual MapReduce job, we first [have to copy](#) the files from our local file system to Hadoop's [HDFS](#).

```

1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -copyFromLocal /tmp/gutenberg /user/hduser/Gutenberg

2 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls /user/hduser

3 Found 1 items
4 drwxr-xr-x  - hduser supergroup          0 2010-05-08
17:40 /user/hduser/Gutenberg

5 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls /user/hduser/Gutenberg

6 Found 3 items
7 -rw-r--r--  3 hduser supergroup      674566 2011-03-10 11:38
/usr/hduser/gutenberg/pg20417.txt
8 -rw-r--r--  3 hduser supergroup      1573112 2011-03-10 11:38
/usr/hduser/gutenberg/pg4300.txt
9 -rw-r--r--  3 hduser supergroup      1423801 2011-03-10 11:38
/usr/hduser/gutenberg/pg5000.txt

10 hduser@ubuntu:/usr/local/hadoop$
```

4. Run the MapReduce job.

```
hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop-examples-1.0.3.jar wordcount /user/hduser/gutenberg /user/hduser/gutenberg-output
```

Output:

Example output of the previous command in the console:

```

1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop jar hadoop-examples*.jar
2 wordcount /user/hduser/gutenberg /user/hduser/gutenberg-output
3 10/05/08 17:43:00 INFO input.FileInputFormat: Total input paths to process : 4
4 3
5 10/05/08 17:43:01 INFO mapred.JobClient: Running job: job_201005081732_0001
6 10/05/08 17:43:02 INFO mapred.JobClient: map 0% reduce 0%
7 10/05/08 17:43:14 INFO mapred.JobClient: map 66% reduce 0%
8 10/05/08 17:43:17 INFO mapred.JobClient: map 100% reduce 0%
9 10/05/08 17:43:26 INFO mapred.JobClient: map 100% reduce 100%
10 10/05/08 17:43:28 INFO mapred.JobClient: Job complete: job_201005081732_0001
11 10/05/08 17:43:28 INFO mapred.JobClient: Counters: 17
12 10/05/08 17:43:28 INFO mapred.JobClient: Job Counters
13 10/05/08 17:43:28 INFO mapred.JobClient: Launched reduce tasks=1
14 10/05/08 17:43:28 INFO mapred.JobClient: Launched map tasks=3
15 10/05/08 17:43:28 INFO mapred.JobClient: Data-local map tasks=3
16 10/05/08 17:43:28 INFO mapred.JobClient: FileSystemCounters
17 10/05/08 17:43:28 INFO mapred.JobClient: FILE_BYTES_READ=2214026
18 10/05/08 17:43:28 INFO mapred.JobClient: HDFS_BYTES_READ=3639512
```

```

19 10/05/08 17:43:28 INFO mapred.JobClient: FILE_BYTES_WRITTEN=3687918
20 10/05/08 17:43:28 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=880330
21 10/05/08 17:43:28 INFO mapred.JobClient: Map-Reduce Framework
22 10/05/08 17:43:28 INFO mapred.JobClient: Reduce input groups=82290
23 10/05/08 17:43:28 INFO mapred.JobClient: Combine output records=102286
24 10/05/08 17:43:28 INFO mapred.JobClient: Map input records=77934
25 10/05/08 17:43:28 INFO mapred.JobClient: Reduce shuffle bytes=1473796
26 10/05/08 17:43:28 INFO mapred.JobClient: Reduce output records=82290
27 10/05/08 17:43:28 INFO mapred.JobClient: Spilled Records=255874
28 10/05/08 17:43:28 INFO mapred.JobClient: Map output bytes=6076267
29 10/05/08 17:43:28 INFO mapred.JobClient: Combine input records=629187
10/05/08 17:43:28 INFO mapred.JobClient: Map output records=629187
10/05/08 17:43:28 INFO mapred.JobClient: Reduce input records=102286

```

Check if the result is successfully stored in HDFS directory

/user/hduser/gutenberg-output:

```

1 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls
/usr/hduser
2 Found 2 items
3 drwxr-xr-x - hduser supergroup          0 2010-05-08 17:40
/usr/hduser/gutenberg
4 drwxr-xr-x - hduser supergroup          0 2010-05-08 17:43
/usr/hduser/gutenberg-output
5 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -ls
/usr/hduser/gutenberg-output
6 Found 2 items
7 drwxr-xr-x - hduser supergroup          0 2010-05-08 17:43
/usr/hduser/gutenberg-output/_logs
8 -rw-r--r-- 1 hduser supergroup     880802 2010-05-08 17:43
/usr/hduser/gutenberg-output/part-r-00000
9 hduser@ubuntu:/usr/local/hadoop$
```

Retrieve the job result from HDFS:

To inspect the file, you can copy it from HDFS to the local file system.
Alternatively, you can use the command

```
hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -cat
/usr/hduser/gutenberg-output/part-r-00000
```

to read the file directly from HDFS without copying it to the local file system.

Here, we will copy the results to the local file system.

```

1 hduser@ubuntu:/usr/local/hadoop$ mkdir /tmp/gutenberg-output
2 hduser@ubuntu:/usr/local/hadoop$ bin/hadoop dfs -getmerge
3 /user/hduser/gutenberg-output /tmp/gutenberg-output
4 hduser@ubuntu:/usr/local/hadoop$ head /tmp/gutenberg-
5 output/gutenberg-output
6 "(Lo)cra"      1
7 "1490"        1
8 "1498,"       1
9 "35"          1
10 "40,"         1
```

```

11 "A      2
12 "AS-IS".      1
13 "A_      1
14 "Absoluti      1
"Alack! 1
hduser@ubuntu:/usr/local/hadoop$
```

Hadoop Web Interfaces

Hadoop comes with several web interfaces which are by default (see `conf/hadoop-default.xml`) available at these locations:

- <http://localhost:50070/> – web UI of the NameNode daemon
- <http://localhost:50030/> – web UI of the JobTracker daemon
- <http://localhost:50060/> – web UI of the TaskTracker daemon

WordCount example reads text files and counts how often words occur. The input is text files and the output is text files, each line of which contains a word and the count of how often it occurred, separated by a tab.

Each mapper takes a line as input and breaks it into words. It then emits a key/value pair of the word and 1. Each reducer sums the counts for each word and emits a single key/value with the word and sum.

As an optimization, the reducer is also used as a combiner on the map outputs. This reduces the amount of data sent across the network by combining each word into a single record.

Program:

```

package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {

    public static class Map extends Mapper<LongWritable, Text,
    Text, IntWritable> {
        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(LongWritable key, Text value, Context
        context) throws IOException, InterruptedException {
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
```

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

    public static class Reduce extends Reducer<Text, IntWritable,
Text, IntWritable> {

        public void reduce(Text key, Iterable<IntWritable> values,
Context context)
            throws IOException, InterruptedException {
            int sum = 0;
            for (IntWritable val : values) {
                sum += val.get();
            }
            context.write(key, new IntWritable(sum));
        }
    }

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();

        Job job = new Job(conf, "wordcount");

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);

        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        job.waitForCompletion(true);
    }
}
```

Session 2A

Problem Statement:

Create your resume using Google Docs and send it to your teacher and get it reviewed by your teacher.

Student Learning Outcomes:

Create a document in consultation with others online using Google Docs.

Theoretical Description:

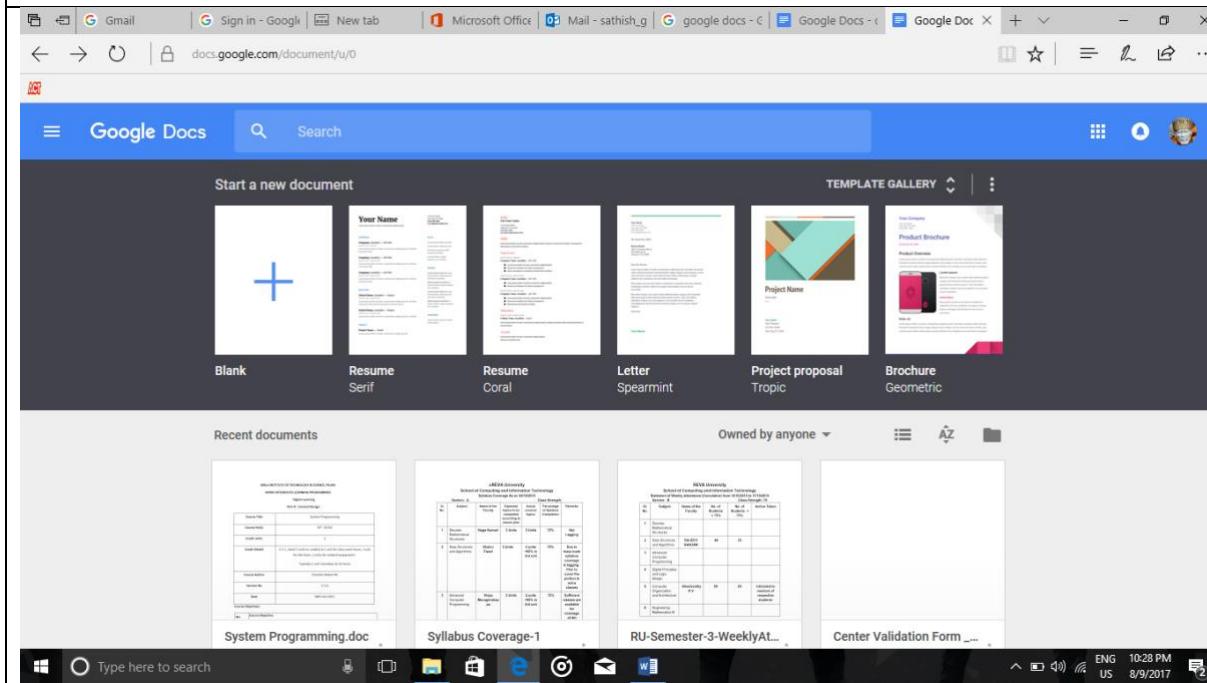
Google Docs brings documents to life with smart editing and styling tools to help easily format text and paragraphs. It helps access, create, and edit documents wherever you go — from your phone, tablet, or computer. The documents can be shared with friends, classmates, coworkers, family and get suggestions which may be accepted or rejected. It supports editing in real time. During document editing, chatting and commenting about the document is possible.

Requirements:

1. Browser
2. Google Account

Procedure:

1. Go to Google Docs and create a resume using a template.
2. Send it for suggestions to a teacher email id.
3. Open the teacher email and provide some suggestions in the resume.
4. Open the document in your login, and accept the suggestions.



Session 2B

Problem Statement:

Create a spreadsheet containing cab requirements and collect the details from various guest teachers.

Student Learning Outcomes:

Create a spreadsheet and collect data from a group of people.

Theoretical Description:

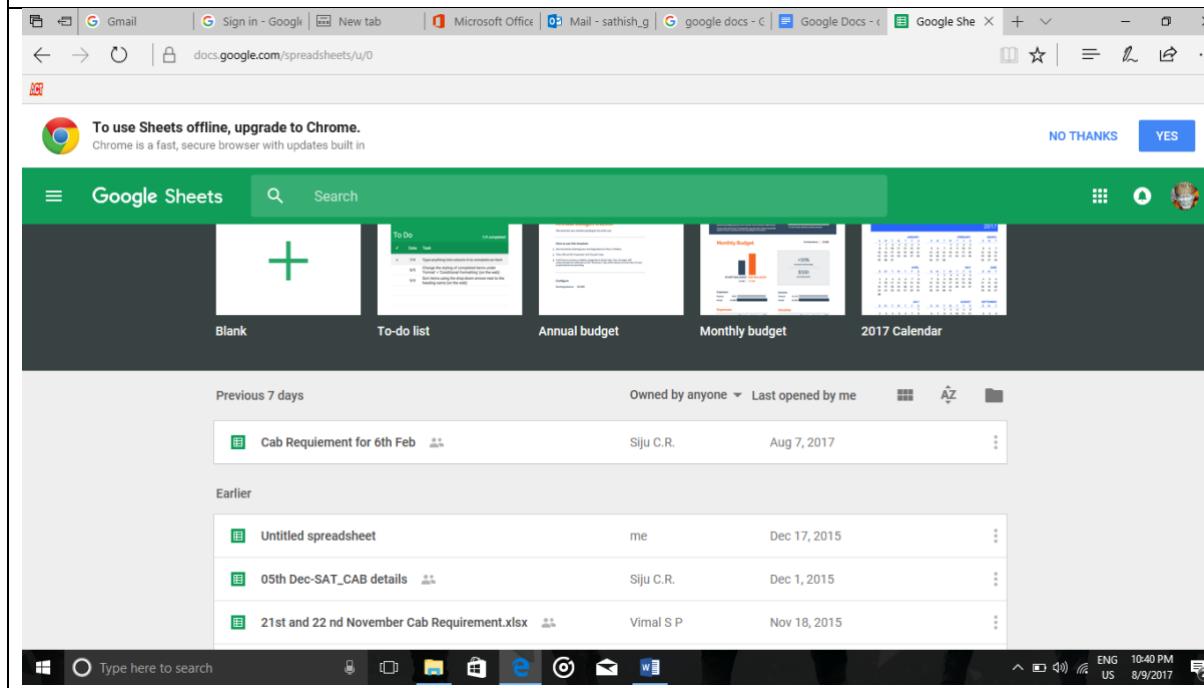
Google Sheets makes your data pop with colorful charts and graphs. Built-in formulas, pivot tables and conditional formatting options save time and simplify common spreadsheet tasks. It helps access, create, and edit spreadsheets wherever you go — from your phone, tablet, or computer. The spreadsheets can be shared with friends, classmates, coworkers, family and get suggestions which may be accepted or rejected. It supports editing in real time. During spreadsheet editing, chatting and commenting about the spreadsheet is possible.

Requirements:

1. Browser
2. Google Account

Procedure:

1. Go to Google Sheets and create a table containing schedule of classes.
2. Send it for filling their pickup address to the guest teachers.
3. Open the teacher email and provide pickup address.
4. Open the spreadsheet in your login, and send it to the transport department.



Session 2C

Problem Statement:

Create a presentation containing phase1 details of your project in collaboration with your project team members.

Student Learning Outcomes:

Create a presentation collectively using Google Slides.

Theoretical Description:

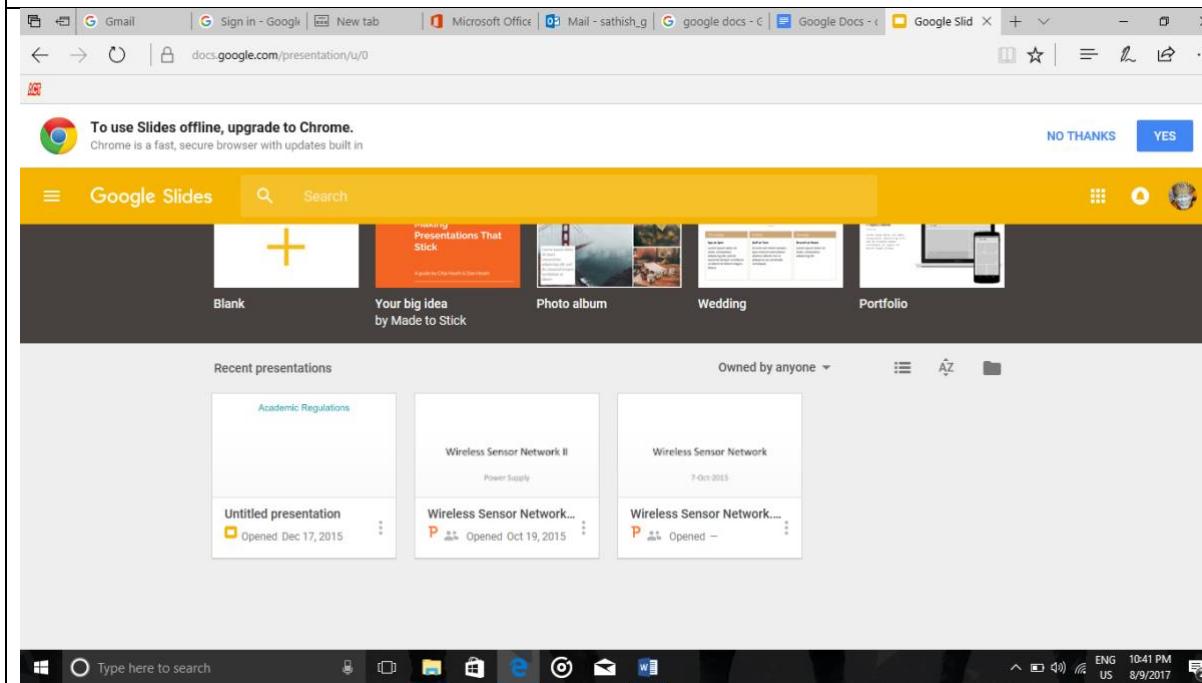
Google Slides makes your ideas shine with a variety of presentation themes, hundreds of fonts, embedded video, animations, and more. It helps access, create, and edit slides wherever you go — from your phone, tablet, or computer. The slides can be shared with friends, classmates, coworkers, family and get suggestions which may be accepted or rejected. It supports editing in real time. During slides editing, chatting and commenting about the slides is possible.

Requirements:

1. Browser
2. Google Account

Procedure:

1. Go to Google Slides and create a presentation to present the progress of your project.
2. Share it with your project team members and invite them to update the slides.
3. Open the project team member login and make some changes to the slides.
4. Open the spreadsheet in your login, and send it to the project coordinator.



Session 2D

Problem Statement:

Create a Form for event registration and get registrations from students.

Student Learning Outcomes:

Creating Forms using Google Forms.

Theoretical Description:

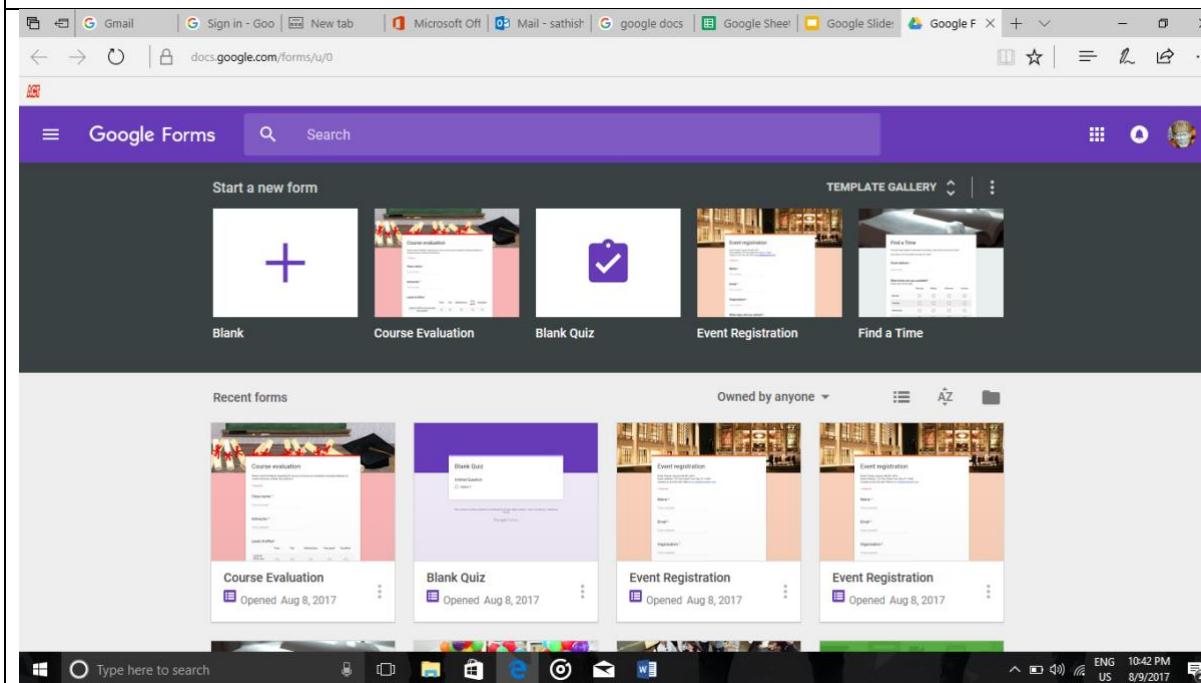
Google Forms helps plan your next camping trip, manage event registrations, whip up a quick poll, collect email addresses for a newsletter, create a pop quiz, and much more. Use your own photo or logo, and Forms will pick just the right colors to complete your own unique form, or choose from a set of curated themes to set the tone. Choose from a bunch of question options, from multiple choice to dropdowns to a linear scale. Add images and YouTube videos, or get fancy with page branching and question skip logic. Responses to your surveys are neatly and automatically collected in Forms, with real time response info and charts. Or, take your data further by viewing it all in Sheets.

Requirements:

1. Browser
2. Google Account

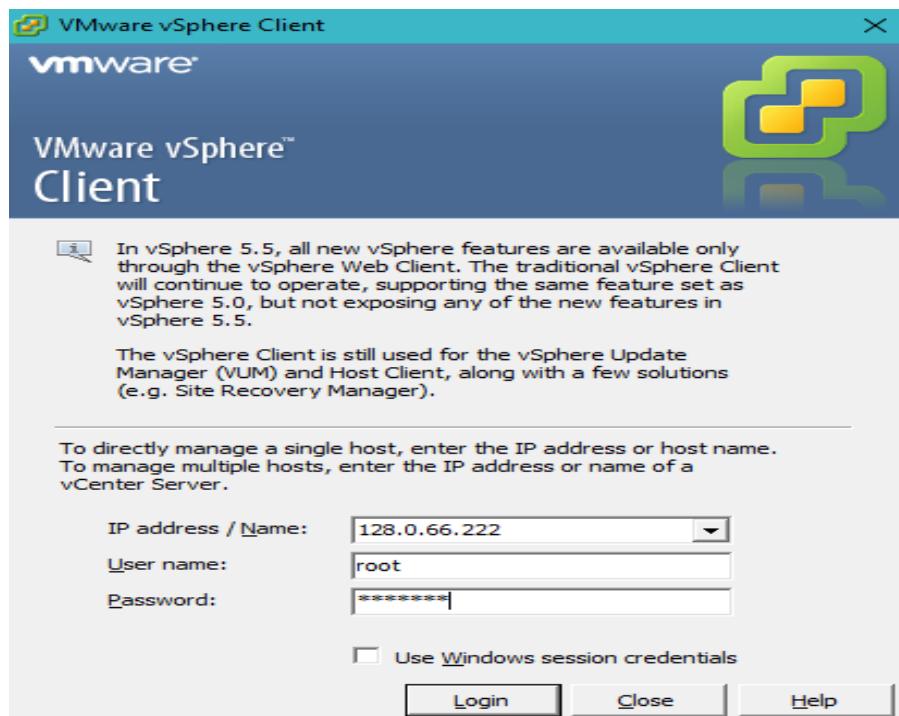
Procedure:

1. Go to Google Forms and create a event registration form.
2. Send it to students for registration.
3. Login as a student and register for an event.
4. Login in to your account and send the responses to the event coordinator.



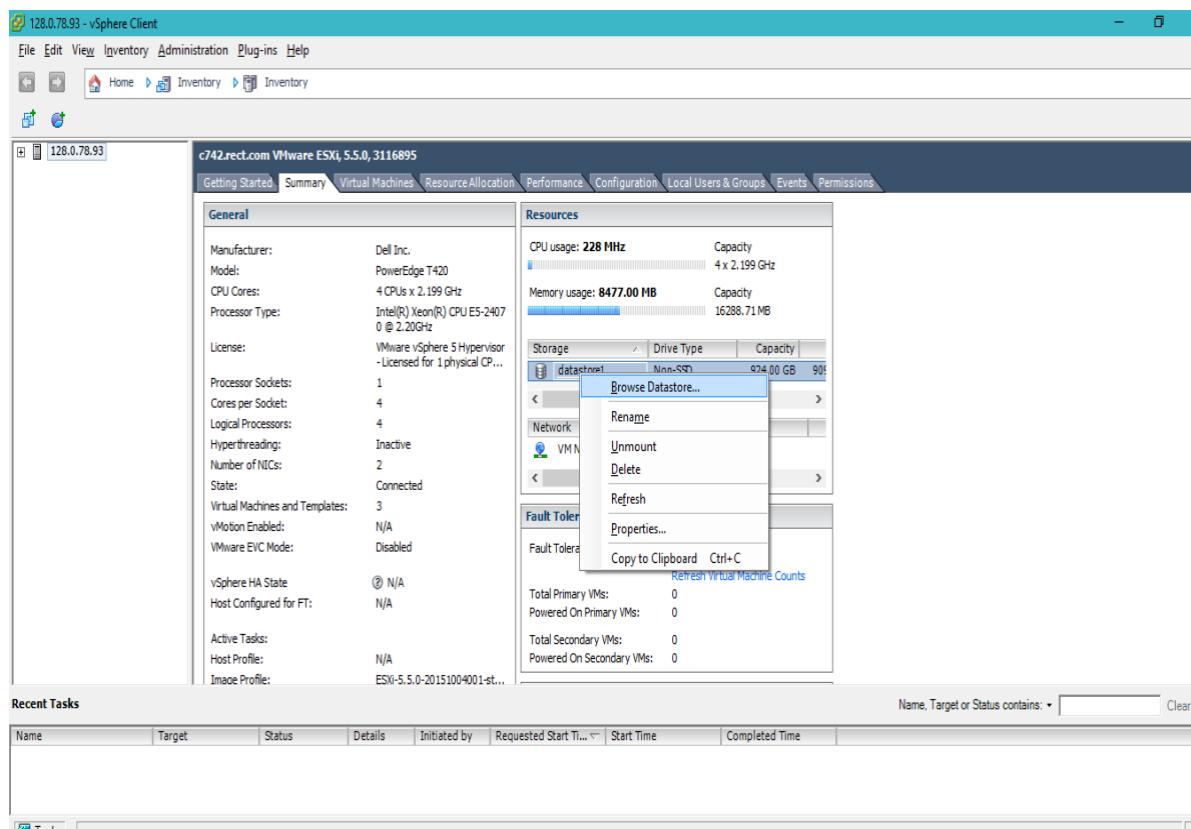
Session 3	
1	Problem Statement: Create multiple VM's on a single physical machine. Provide a communication between them using and without using vClient.
2	Student Learning Outcomes: To install and setup the Virtual Machine To check the connectivity between the VM's
3	Theoretical Description: Virtual Machine (VM): is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination. There are different kinds of virtual machines, each with different functions: <ul style="list-style-type: none"> • System virtual machines (also termed full virtualization VMs) provide a substitute for a real machine. They provide functionality needed to execute entire operating systems. A hypervisor uses native execution to share and manage hardware, allowing for multiple environments which are isolated from one another, yet exist on the same physical machine. Modern hypervisors use hardware-assisted virtualization, virtualization-specific hardware, primarily from the host CPUs. • Process virtual machines are designed to execute computer programs in a platform-independent environment.
4 Requirements	
The following list of requirements as a starting point. Like physical computers, the virtual machines running under VMware Workstation generally perform better if they have faster processors and more memory. <ul style="list-style-type: none"> ▪ PC Hardware - Standard x86-compatible personal computer, 400 MHz or faster CPU minimum (500 MHz recommended), Multiprocessor systems supported, 64-bit processor support for AMD64 Opteron, Athlon 64 and Intel IA-32e CPU (including "Nocona"). ▪ Memory - 128 MB minimum (256 MB recommended) ▪ Disk Drives - IDE and SCSI hard drives supported, up to 950GB capacity, At least 1GB free disk space recommended for each guest operating system and the application software used with it; if you use a default setup, the actual disk space needs are approximately the same as those for installing and running the guest operating system and applications on a physical computer. ▪ Local Area Networking (Optional) ▪ Host Operating System 	
5	Procedure
Access the ESXi through the vCleint, <ol style="list-style-type: none"> 1) Open vSphere Client and Enter the followings: <ol style="list-style-type: none"> a. IP of the server (eg: 128.0.78.222) 	

- b. Enter the User Name (User name of the server)
- c. Enter the password (password of the server)

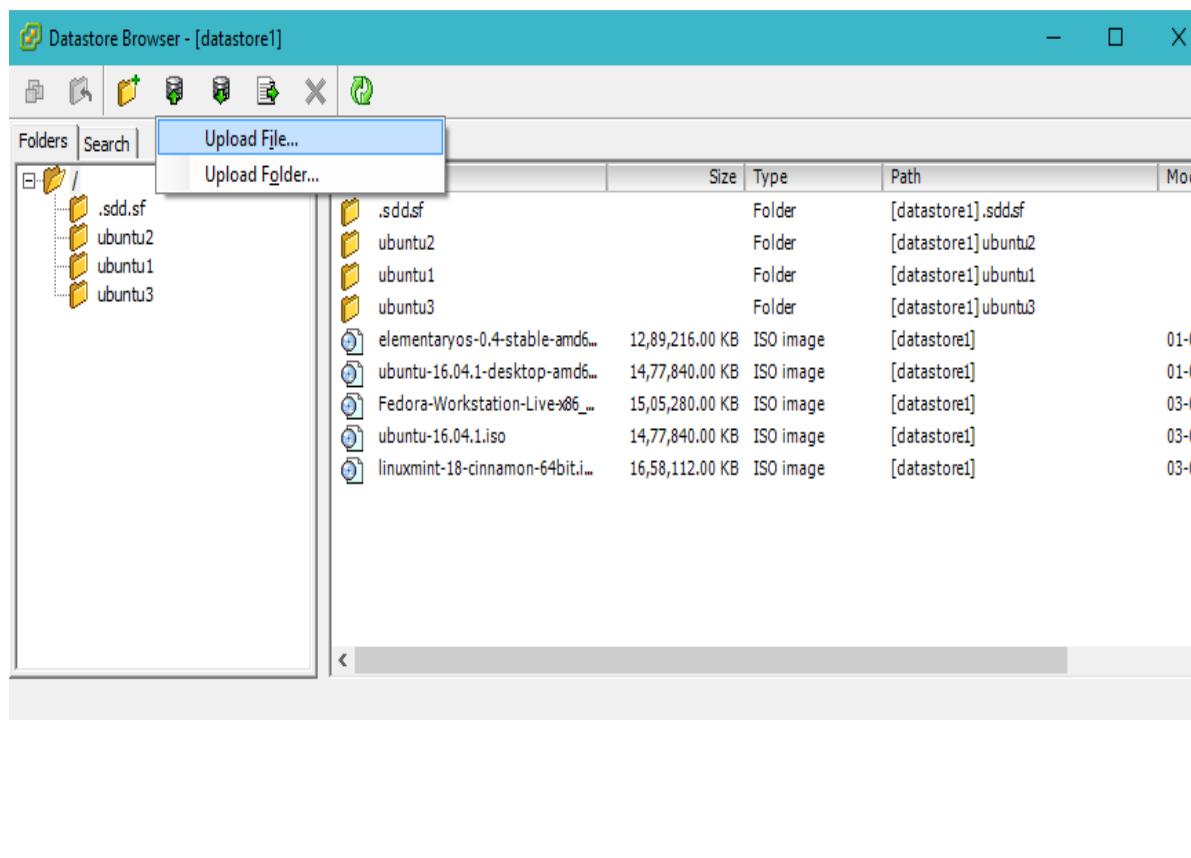


- 2) To upload to datastore, Open the summary tab from the interface

3) Right Click on datastore and browse datastore.

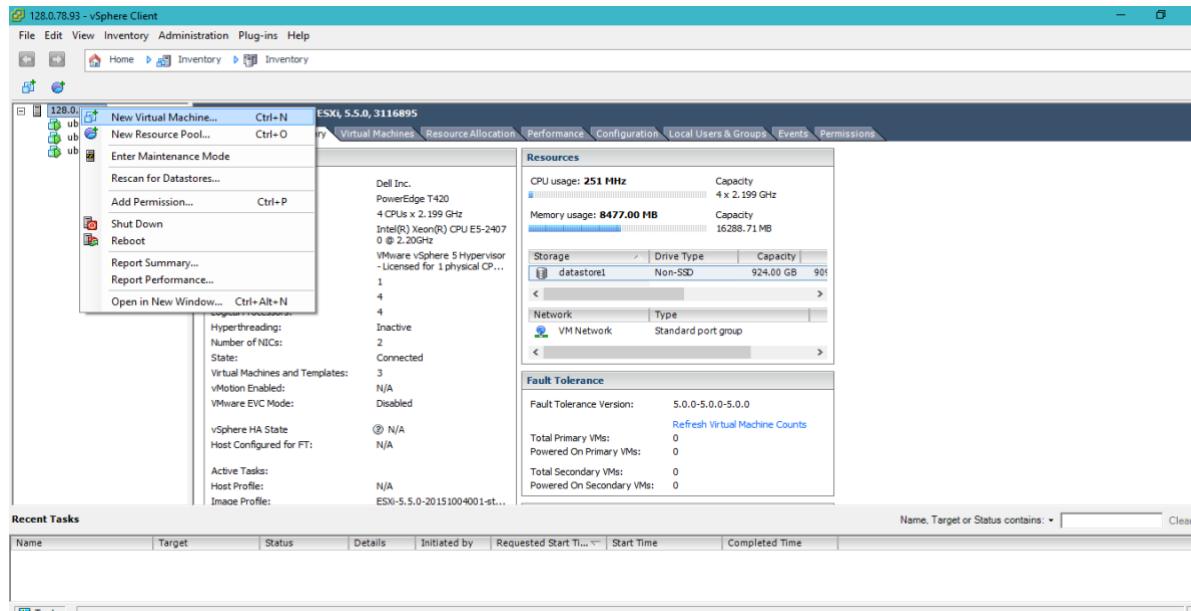


4) Click on upload icon, press upload file, browse the file and upload.

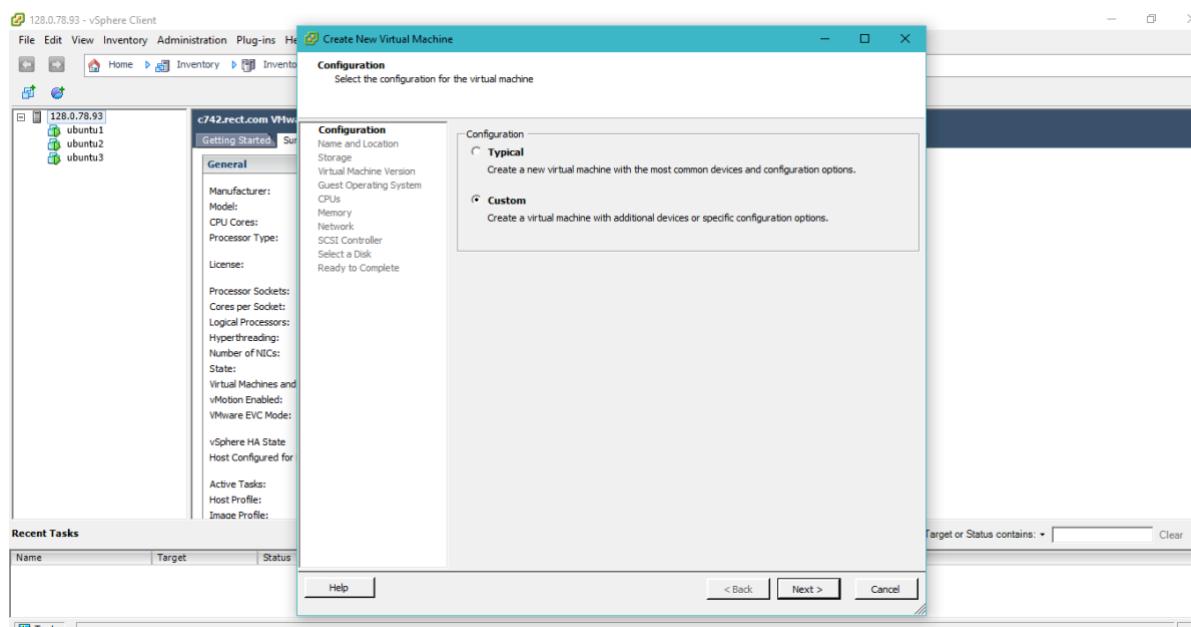


To create a VM, follow the procedure

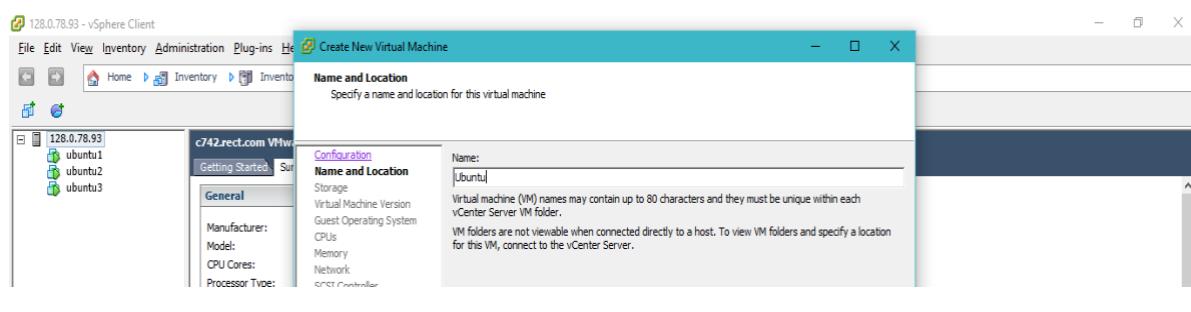
- 1) Right click on the server > New Virtual Machine



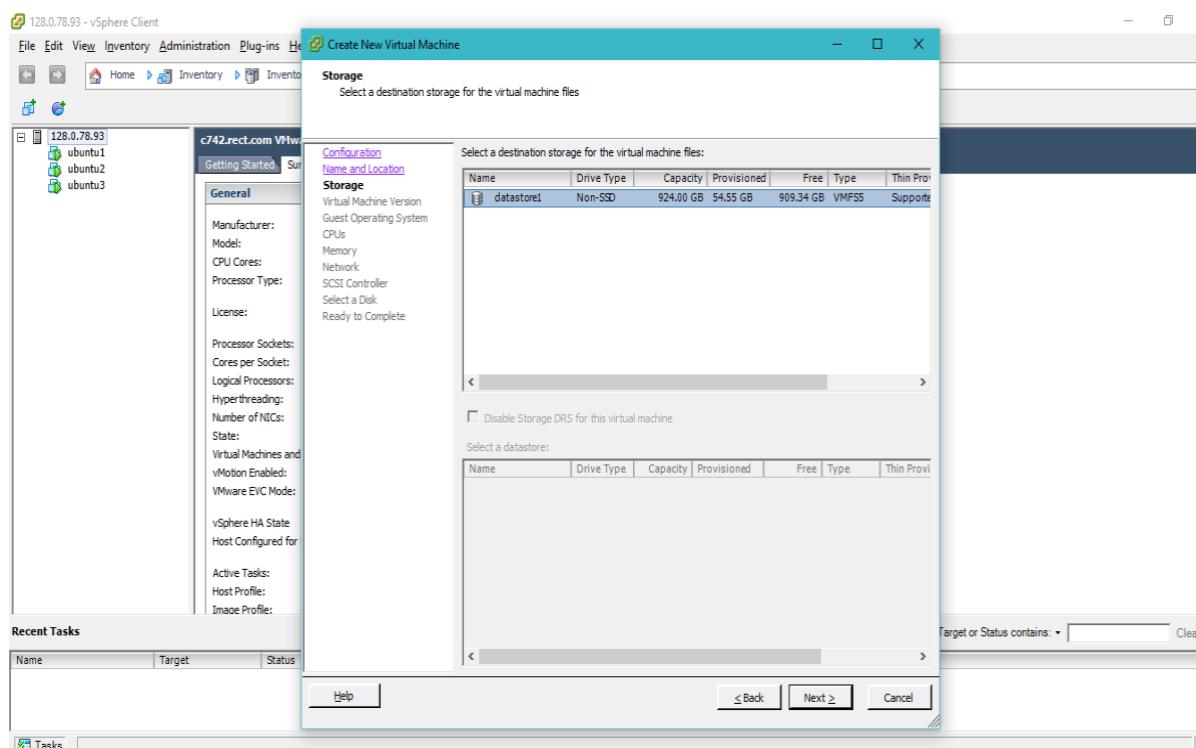
- 2) Select Custom and click next.



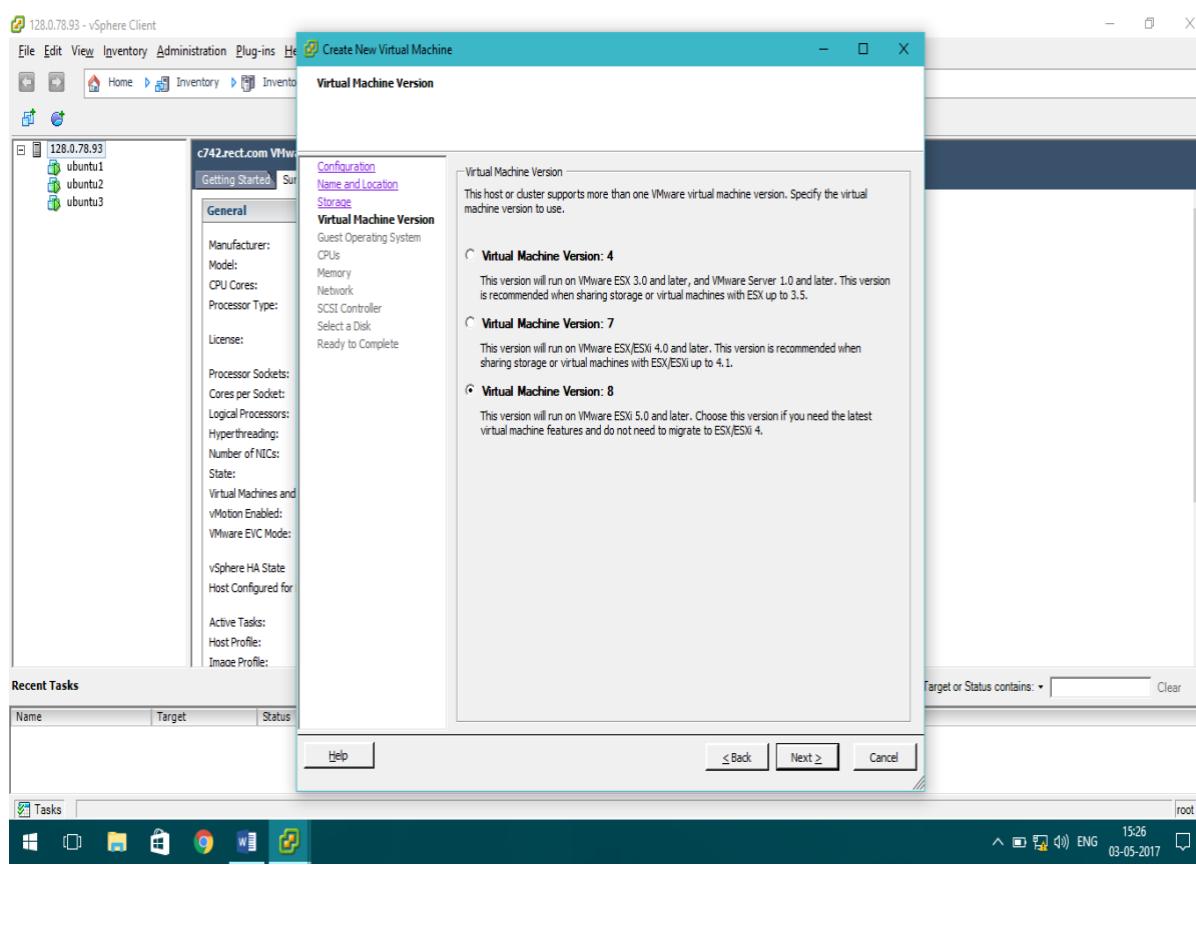
- 3) Give a name to the virtual machine and click next.



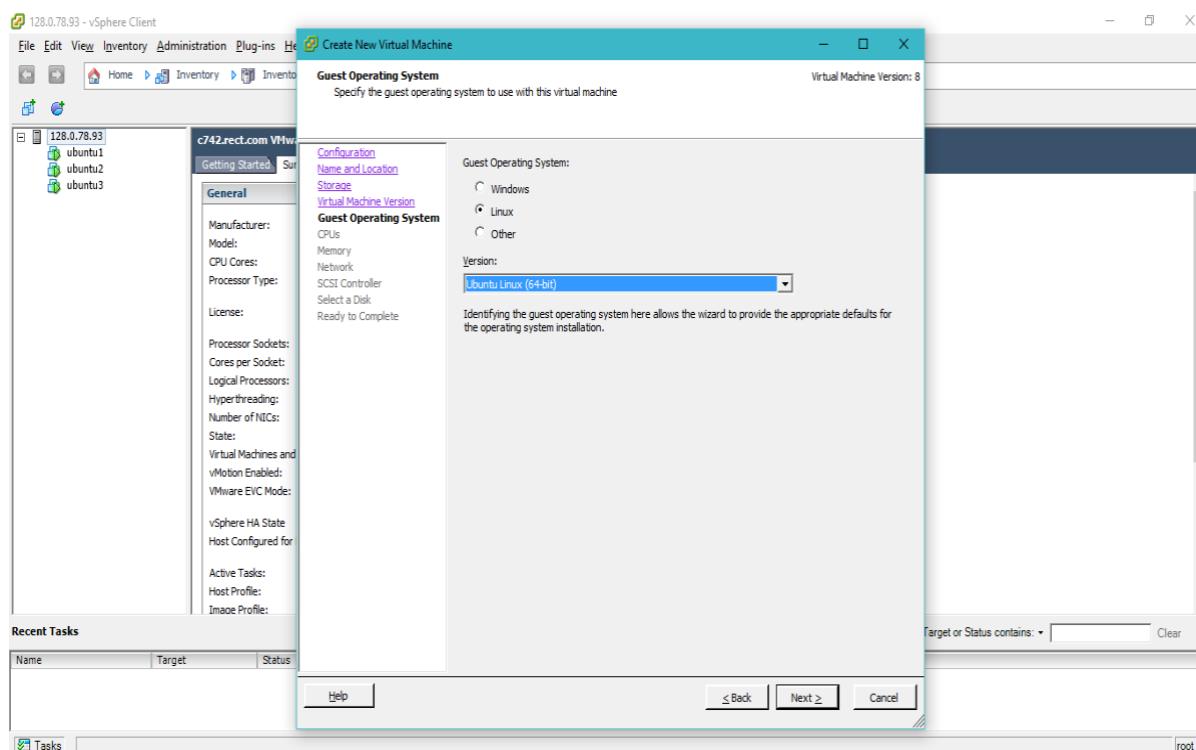
4) Select storage from the list and click next.



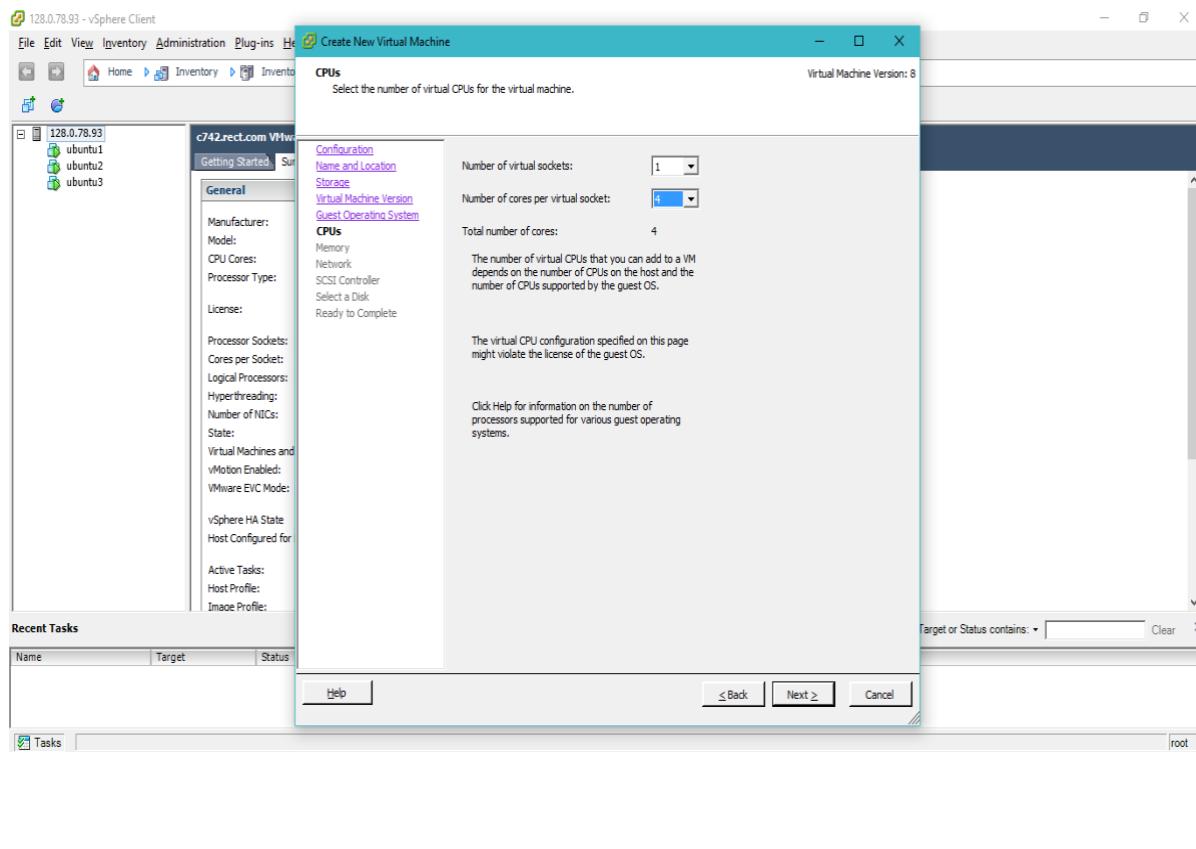
5) Select the virtual machine version (Select the latest version) and click next.



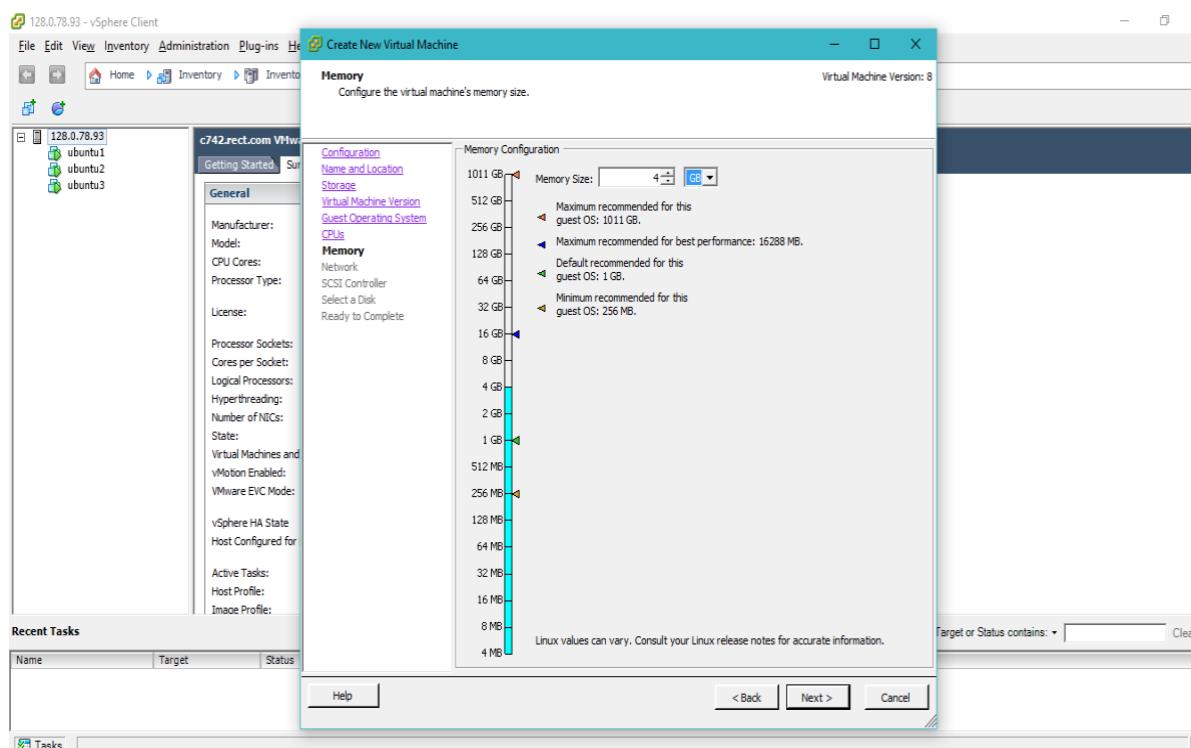
- 6) Select the type of guest Operating System and click next.



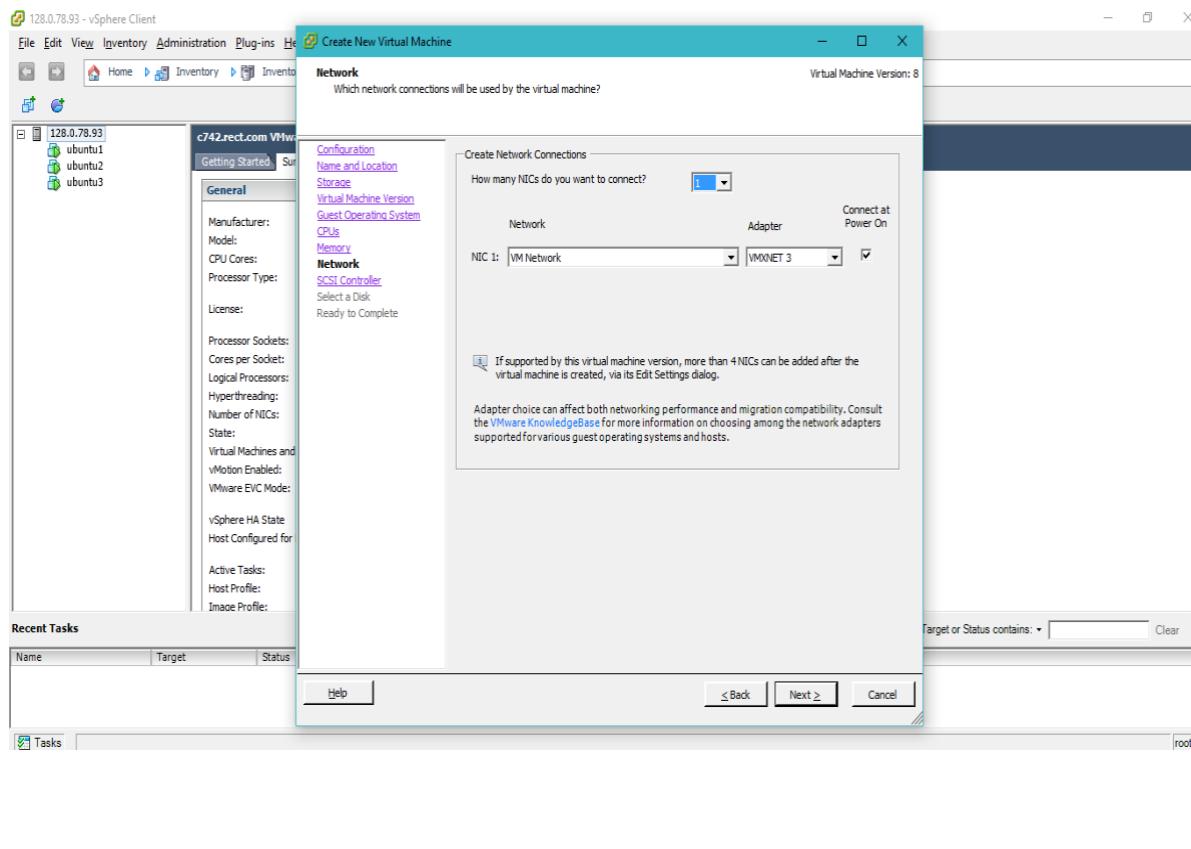
- 7) Select the number of virtual cores (Virtual sockets).



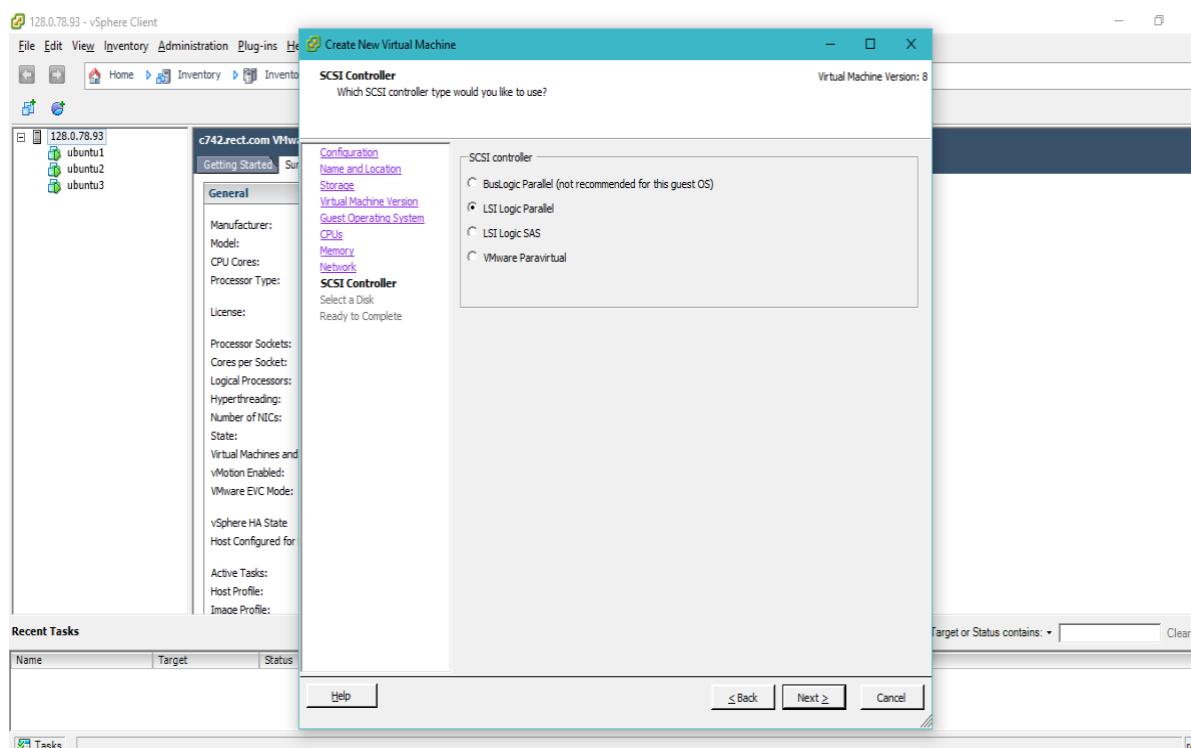
8) Select the amount of RAM and click next.



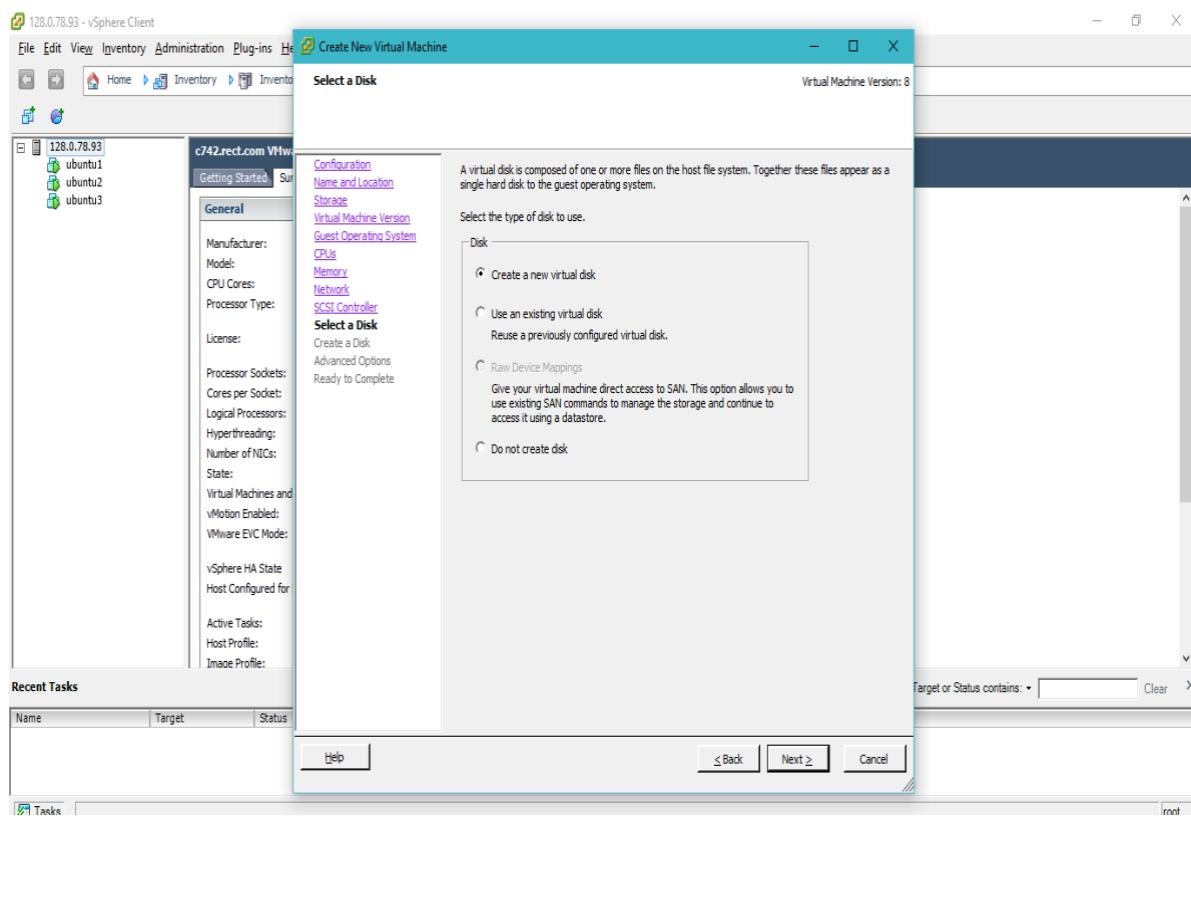
9) Don't change anything in the networks page and click next.



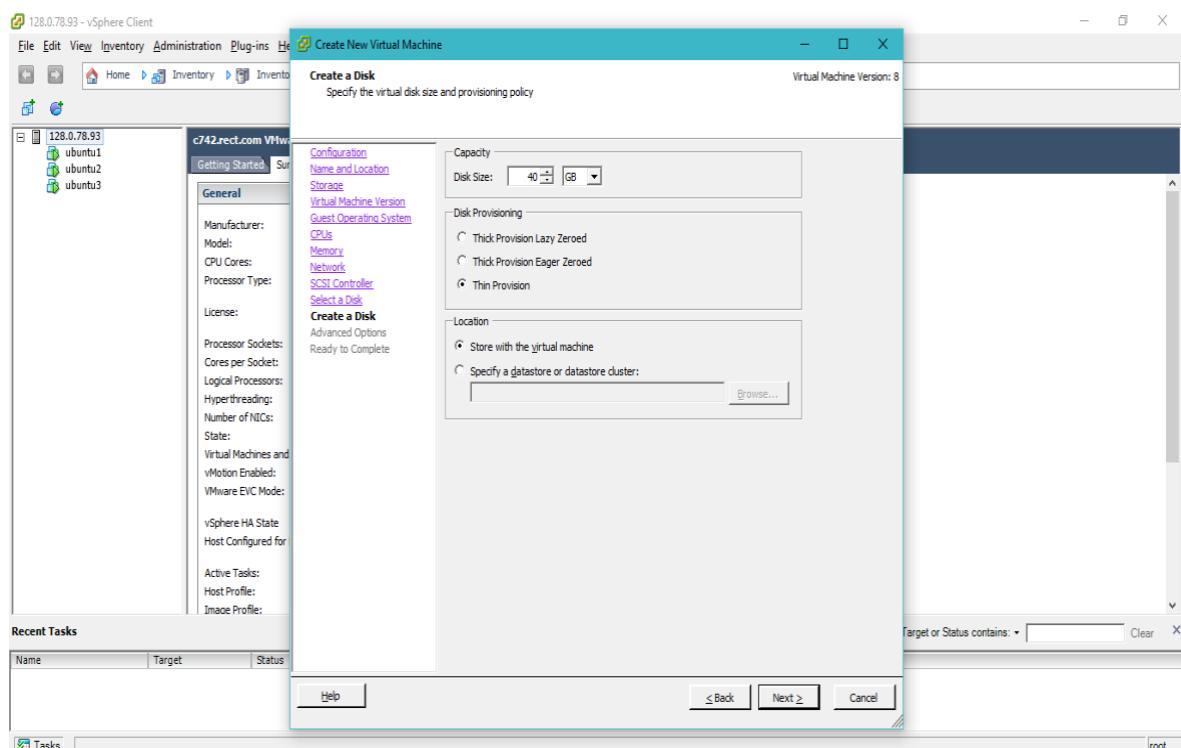
10) Select SCSI controller as LSI Logic parallel



11) Click on create a new disk.



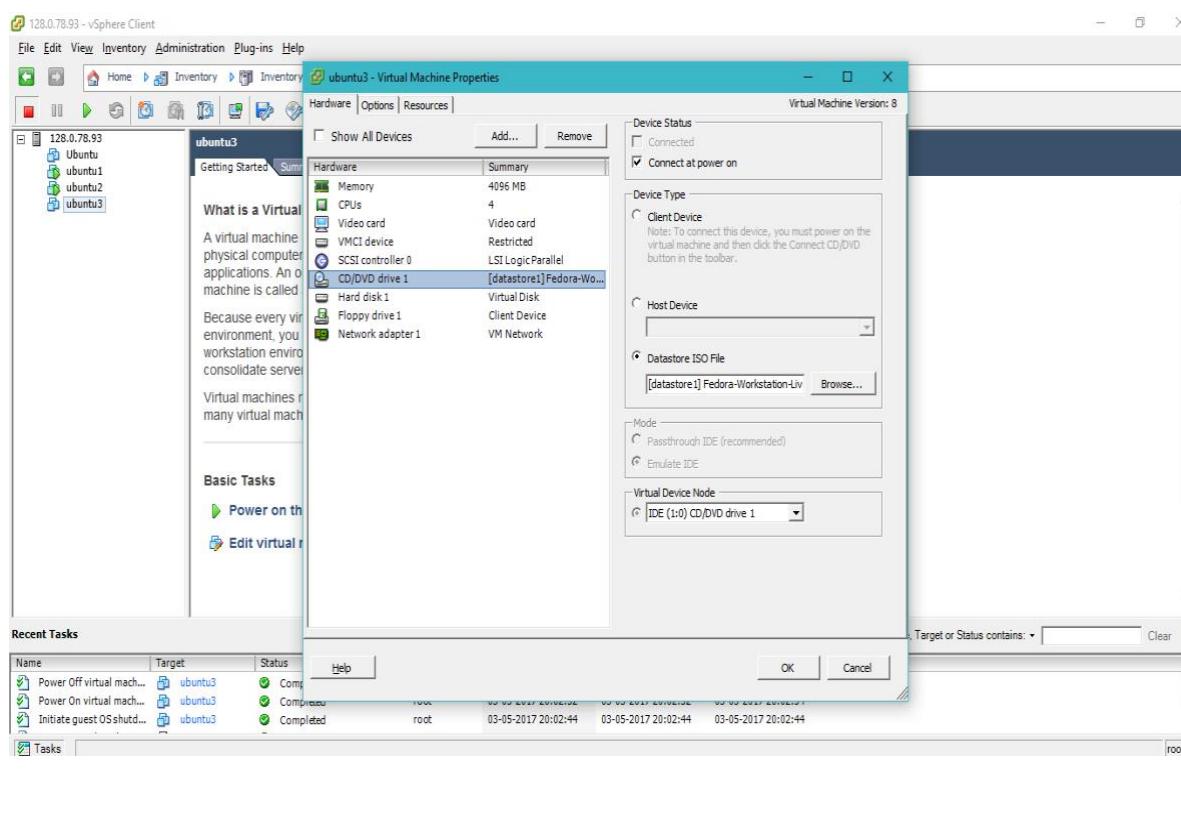
12) Enter the disk capacity (Atleast 30GB), select Thin provisioning and click next.



13) Review your configuration and finish.

14) Now select the VM and click on Edit Virtual machine settings.

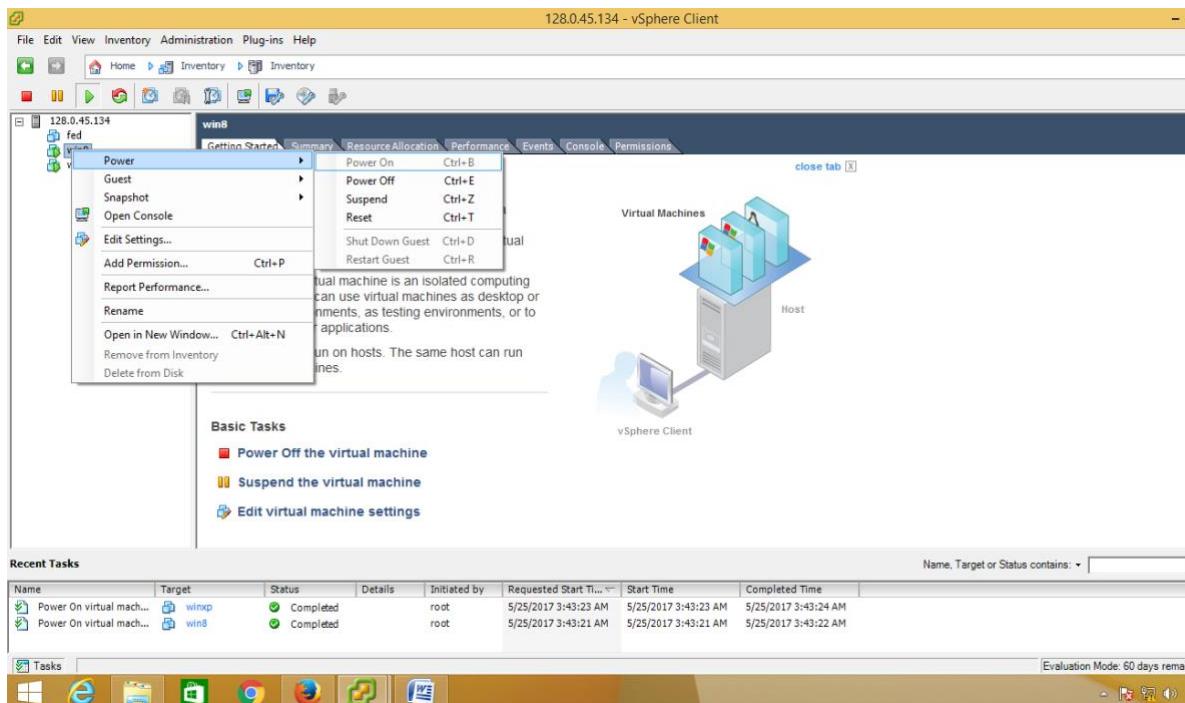
15) Click on CD/DVD drive 1, select datastore ISO file and locate the ISO on the datastore.



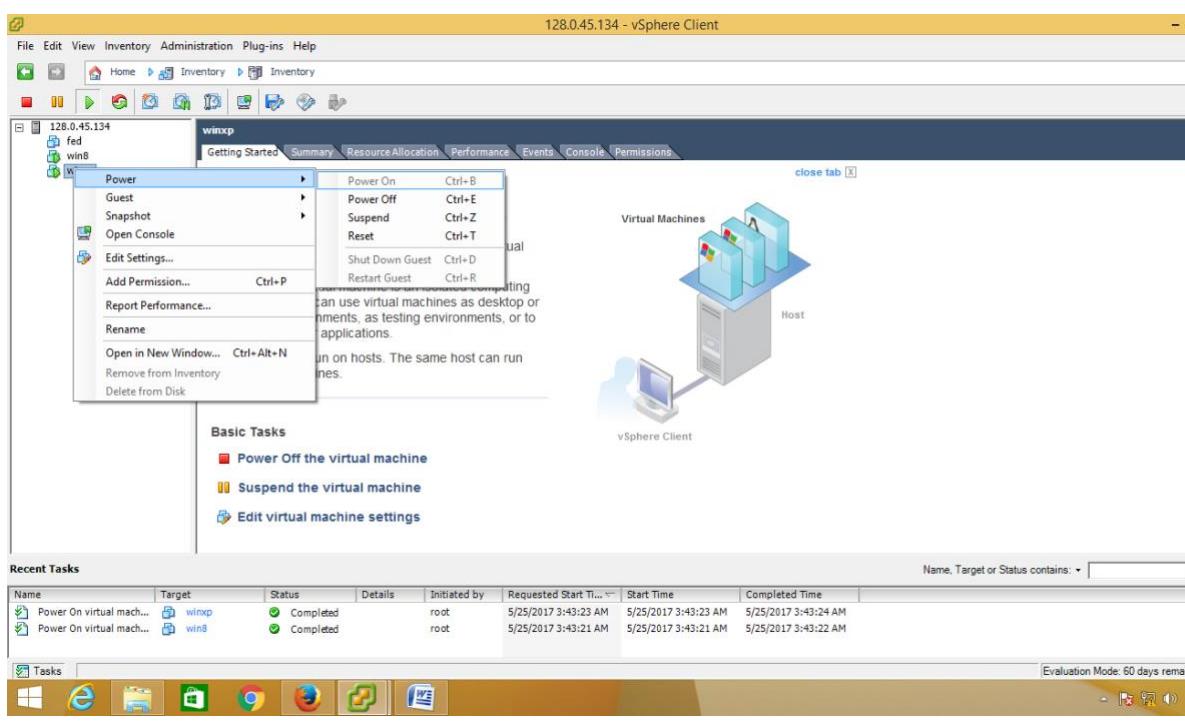
16) Turn on the virtual machine and proceed with normal installation of OS in to that VM.

Repeat the same to create one more VM with different name/IP on same ESXi, then later open them through different/same clients. Now PING between them using their IP address. As ping <ip of other vm>

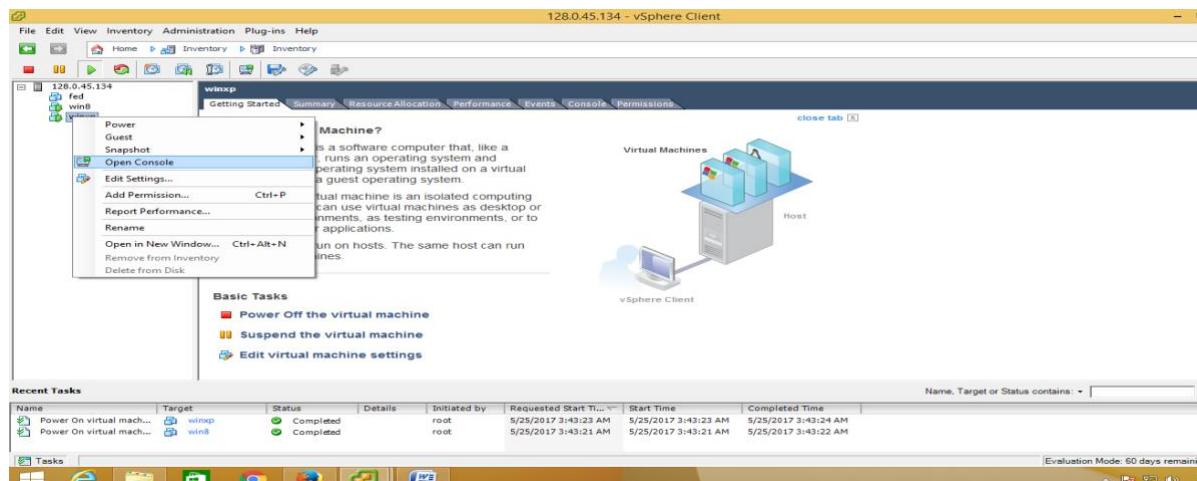
1. Power on the VM on ESXi



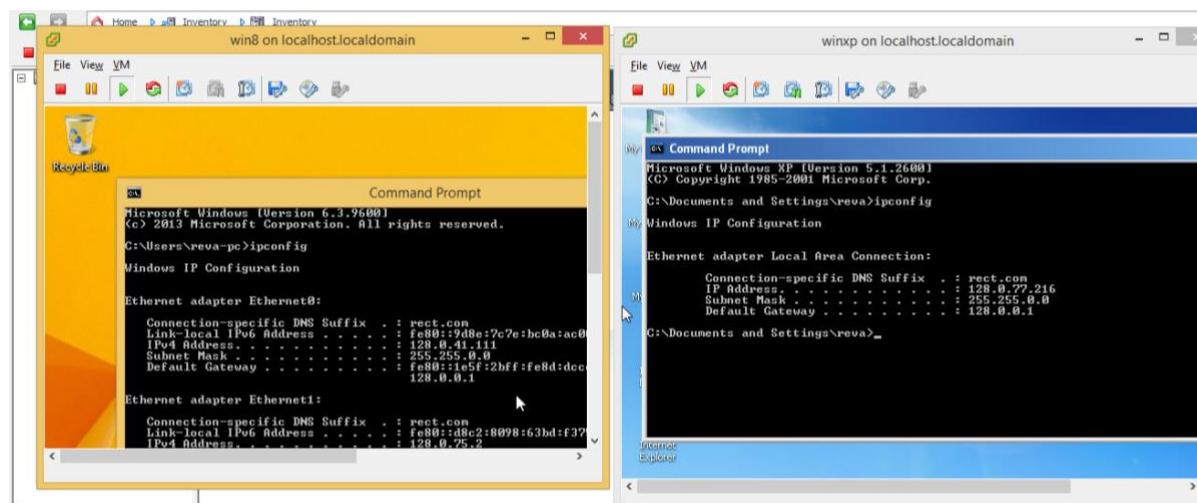
2. Power on the another VM on same ESXi



3. The console window can be displayed for those VM's separately.

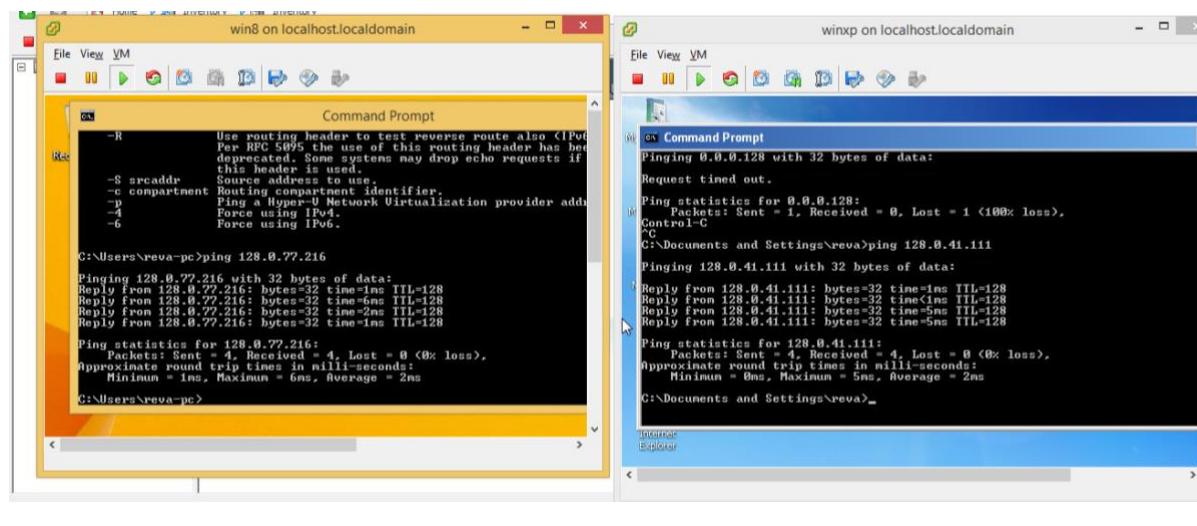


4. Find the IP address of those VM's using “ipconfig” command in command prompt



5. Check the connectivity for the communication between those VM's using PING command as

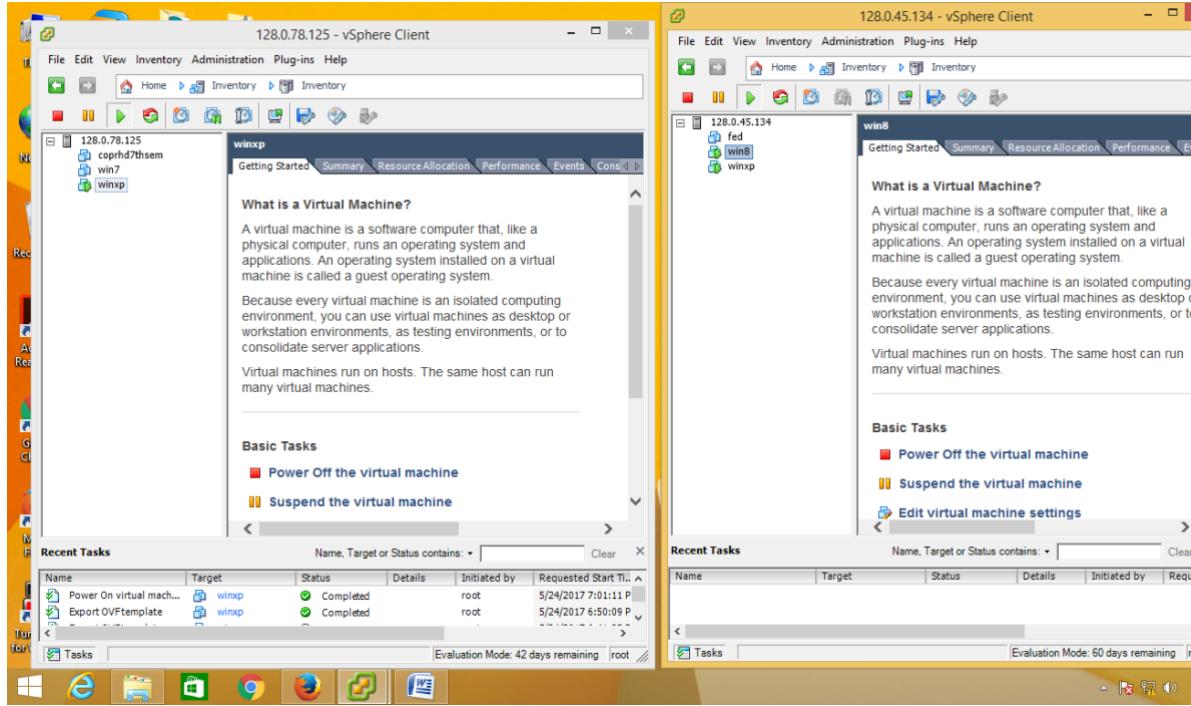
“ping <ip address of other VM>”



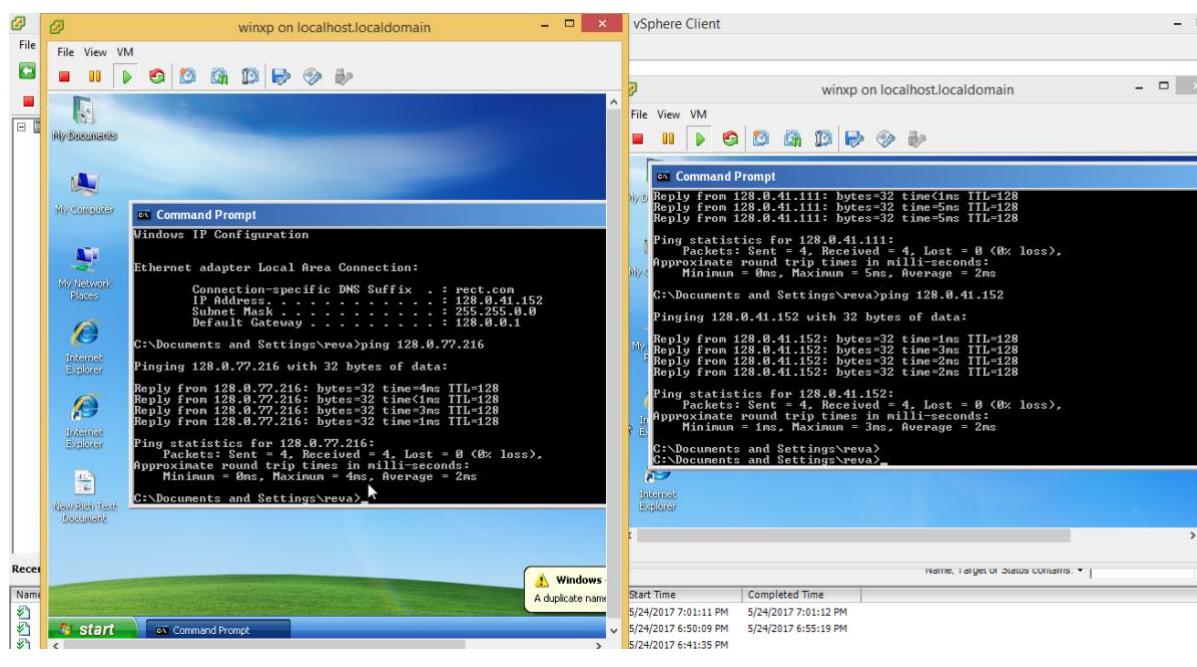
Session 4	
1	Problem Statement:
Demonstrate the communication of VM's on different physical devices using ESXi.	
2	Student Learning Outcomes:
To install and setup the Virtual Machine on same and different ESXi. To check the connectivity between the VM's	
3	Theoretical Description:
<p>Virtual Machine (VM): is an emulation of a computer system. Virtual machines are based on computer architectures and provide functionality of a physical computer. Their implementations may involve specialized hardware, software, or a combination.</p> <p>There are different kinds of virtual machines, each with different functions:</p> <ul style="list-style-type: none"> • System virtual machines (also termed full virtualization VMs) provide a substitute for a real machine. They provide functionality needed to execute entire operating systems. A hypervisor uses native execution to share and manage hardware, allowing for multiple environments which are isolated from one another, yet exist on the same physical machine. Modern hypervisors use hardware-assisted virtualization, virtualization-specific hardware, primarily from the host CPUs. • Process virtual machines are designed to execute computer programs in a platform-independent environment. 	
4	Requirements
The following list of requirements as a starting point. Like physical computers, the virtual machines running under VMware Workstation generally perform better if they have faster processors and more memory.	
<ul style="list-style-type: none"> ▪ PC Hardware - Standard x86-compatible personal computer, 400 MHz or faster CPU minimum (500 MHz recommended), Multiprocessor systems supported, 64-bit processor support for AMD64 Opteron, Athlon 64 and Intel IA-32e CPU (including "Nocona"). ▪ Memory - 128 MB minimum (256 MB recommended) ▪ Disk Drives - IDE and SCSI hard drives supported, up to 950GB capacity, At least 1GB free disk space recommended for each guest operating system and the application software used with it; if you use a default setup, the actual disk space needs are approximately the same as those for installing and running the guest operating system and applications on a physical computer. ▪ Local Area Networking (Optional) ▪ Host Operating System 	
5	Procedure
1) Open the already created VM's on same ESXi server using the different client browser's on different physical machines. Then now ping between those VM's of same ESXi. : Already demonstrated in the previous session.	

- 2) Provide the communication between the VM's created with different ESXi servers. So, create a new/ use the already created VM's with different ESXi's. Access the VM's on different ESXi's using the different client browser's on different physical machines. Then now ping between them by using their IP address.

- a. Login to two ESXi's using two different vClient terminal.



- b. Switch on the VM's on those different ESXi's,
 c. Find the IP address of them using ipconfig
 d. Check the connectivity for the communication using PING command

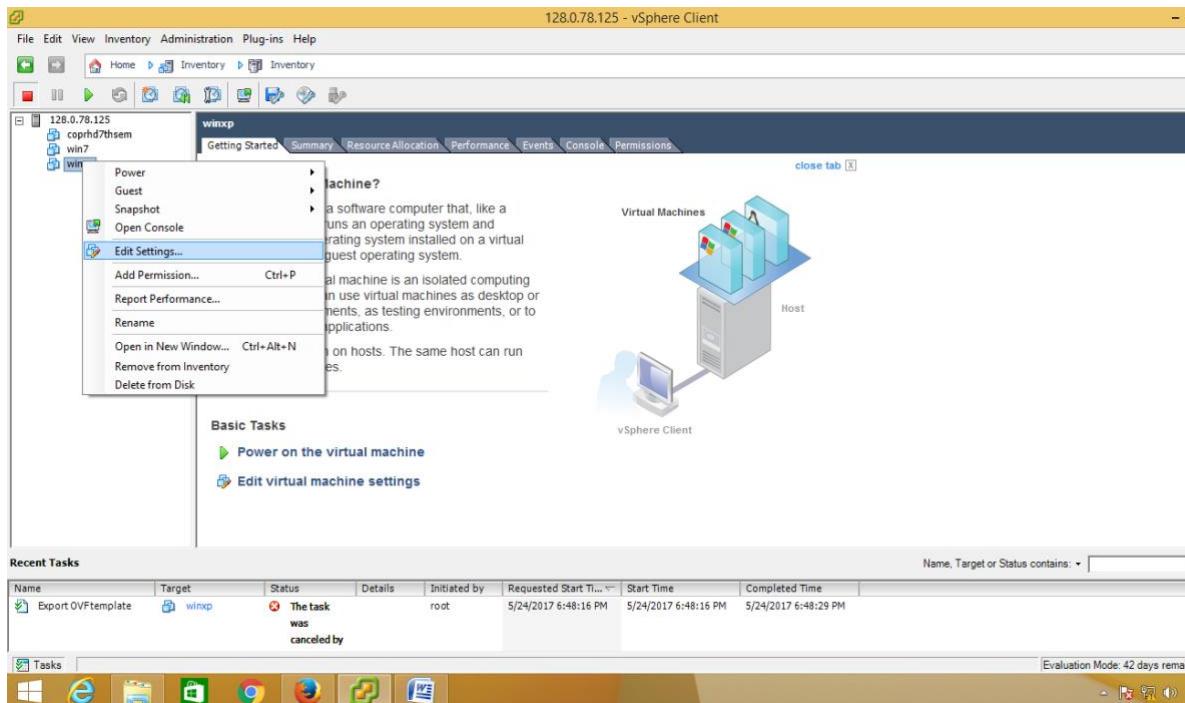


Session 5	
1	Problem Statement:
	Illustrate the cloning of VM's.
2	Student Learning Outcomes:
	To configure and maintain the VM's. To configure a multiple VM's on multiple ESXi's
3	Theoretical Description:
	<p>Cloning: Cloning a virtual machine creates a duplicate of the virtual machine with the same configuration and installed software as the original.</p> <p>Optionally, you can customize the guest operating system of the clone to change the virtual machine name, network settings, and other properties. This prevents conflicts that can occur if a virtual machine and a clone with identical guest operating system settings are deployed simultaneously.</p> <p>Installing a guest operating system and applications can be time consuming. With clones, you can make many copies of a virtual machine from a single installation and configuration process. Clones are useful when you must deploy many identical virtual machines to a group.</p> <p>With clones you can conveniently make complete copies of a virtual machine, without browsing a host file system or worrying if you have located all the configuration files.</p> <p>Open Virtualization Format (OVF) is an open standard for packaging and distributing virtual appliances or, more generally, software to be run in virtual machines.</p> <p>The standard describes an "open, secure, portable, efficient and extensible format for the packaging and distribution of software to be run in virtual machines". The OVF standard is not tied to any particular hypervisor or instruction set architecture. The unit of packaging and distribution is a so-called <i>OVF Package</i> which may contain one or more <i>virtual systems</i> each of which can be deployed to a virtual machine.</p>
4	Requirements
	<ul style="list-style-type: none"> ▪ You must be connected in order to clone a virtual machine. ▪ To customize the guest operating system of the virtual machine, check that your guest operating system meets the requirements for customization. See Guest Operating System Customization Requirements. ▪ To use a customization specification, you must first create or import the customization specification. ▪ To use a custom script to generate the host name or IP address for the new virtual machine, configure the script. See Configure a Script to Generate Computer Names and IP Addresses During Guest Operating System Customization
5	Procedure
	<p>There are two ways to be followed,</p> <p>I. Using OVF.</p>

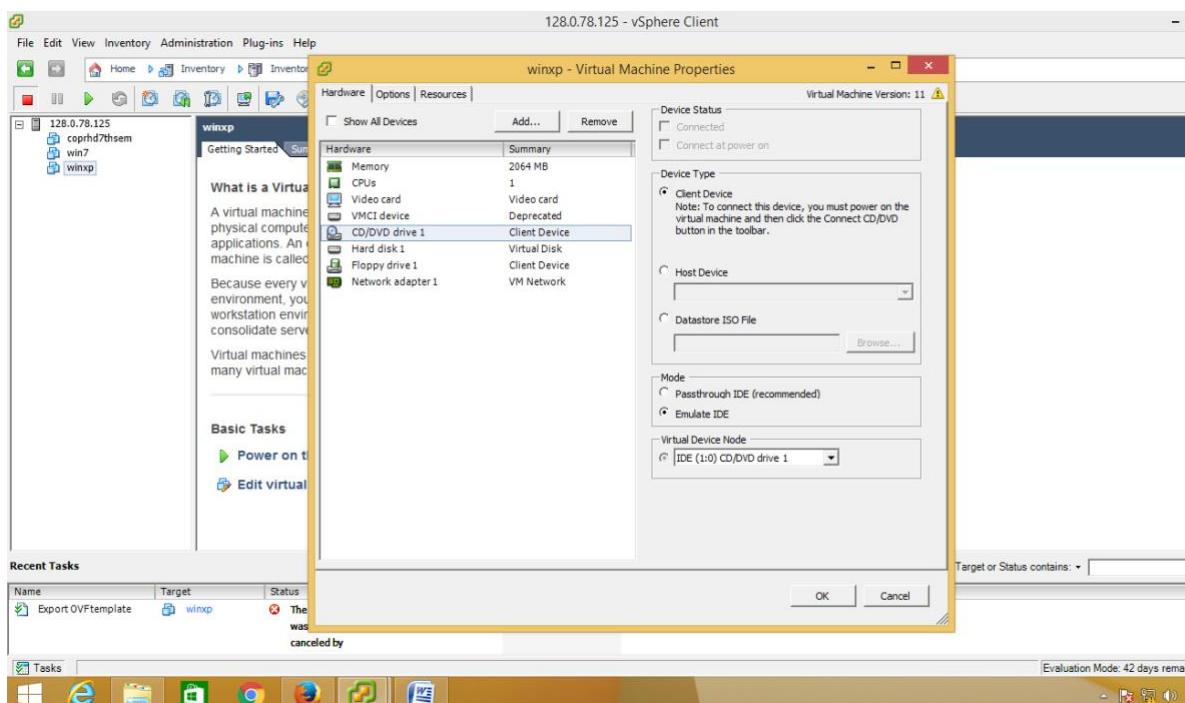
II. Download/Upload a Folder of VM

Method I:

1. Go to Edit Settings of the VM

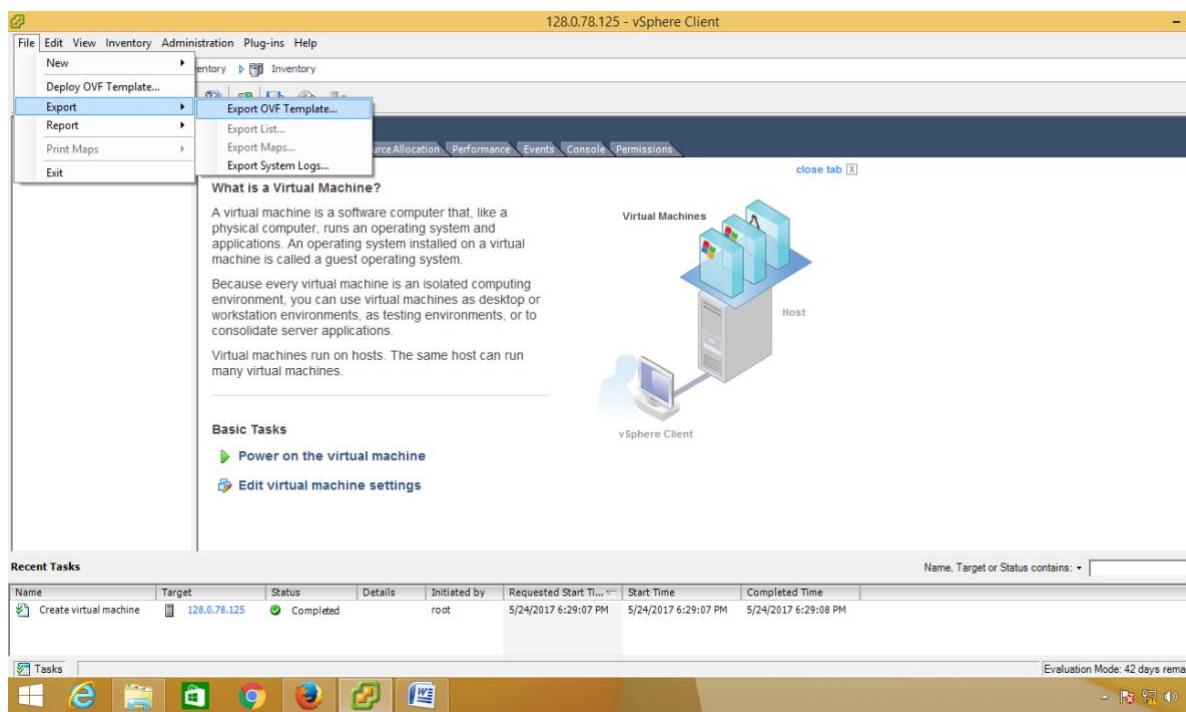


2. Change the CD/DCD drive from “datasore” to “Client Device”.

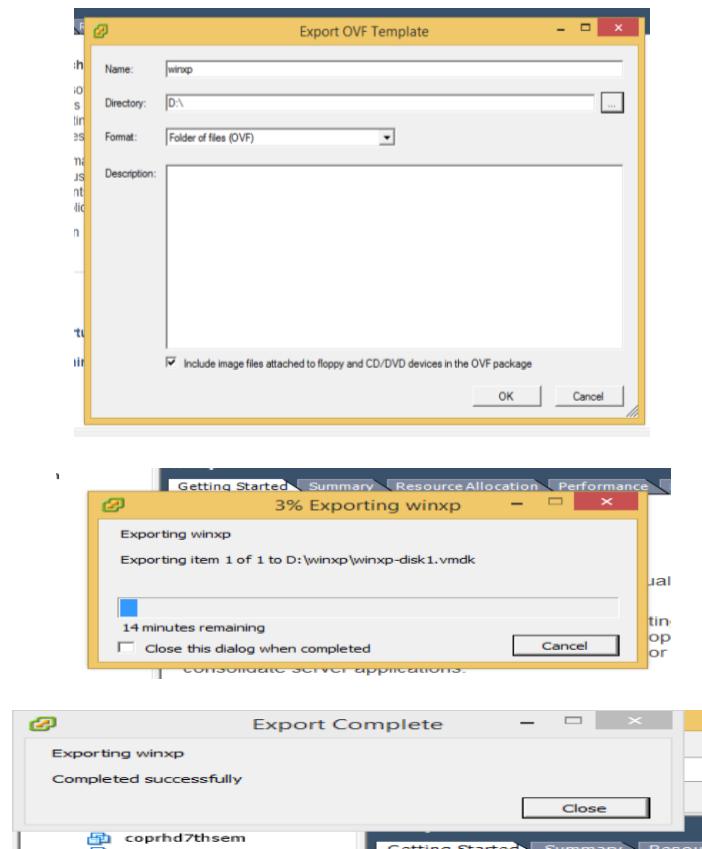


3. Make sure that the VM is turned OFF
4. Select the VM which you want to clone

5. Go to File -> Export -> Export OVF Template

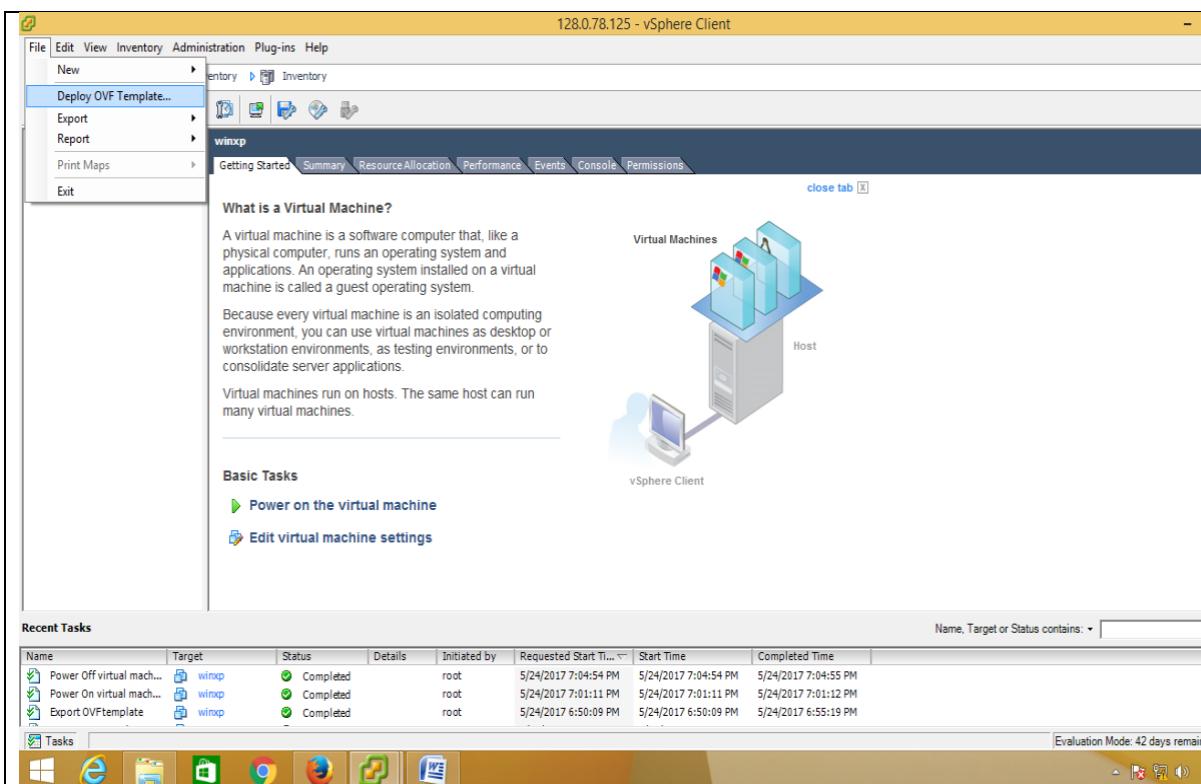


6. Select the destination where you wanted to store the copy of it

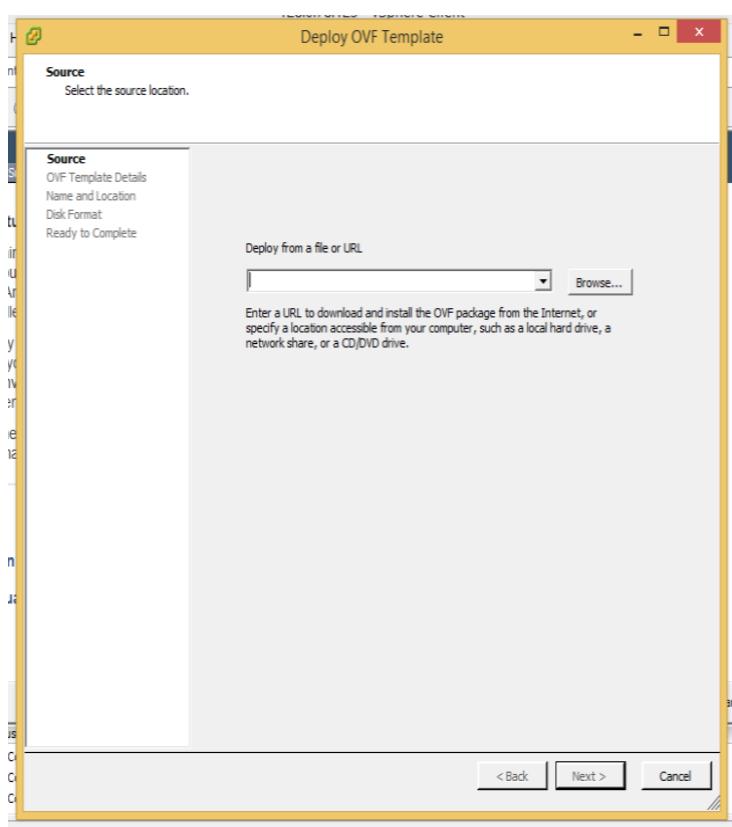


7. For importing a copy to same /different ESXi,

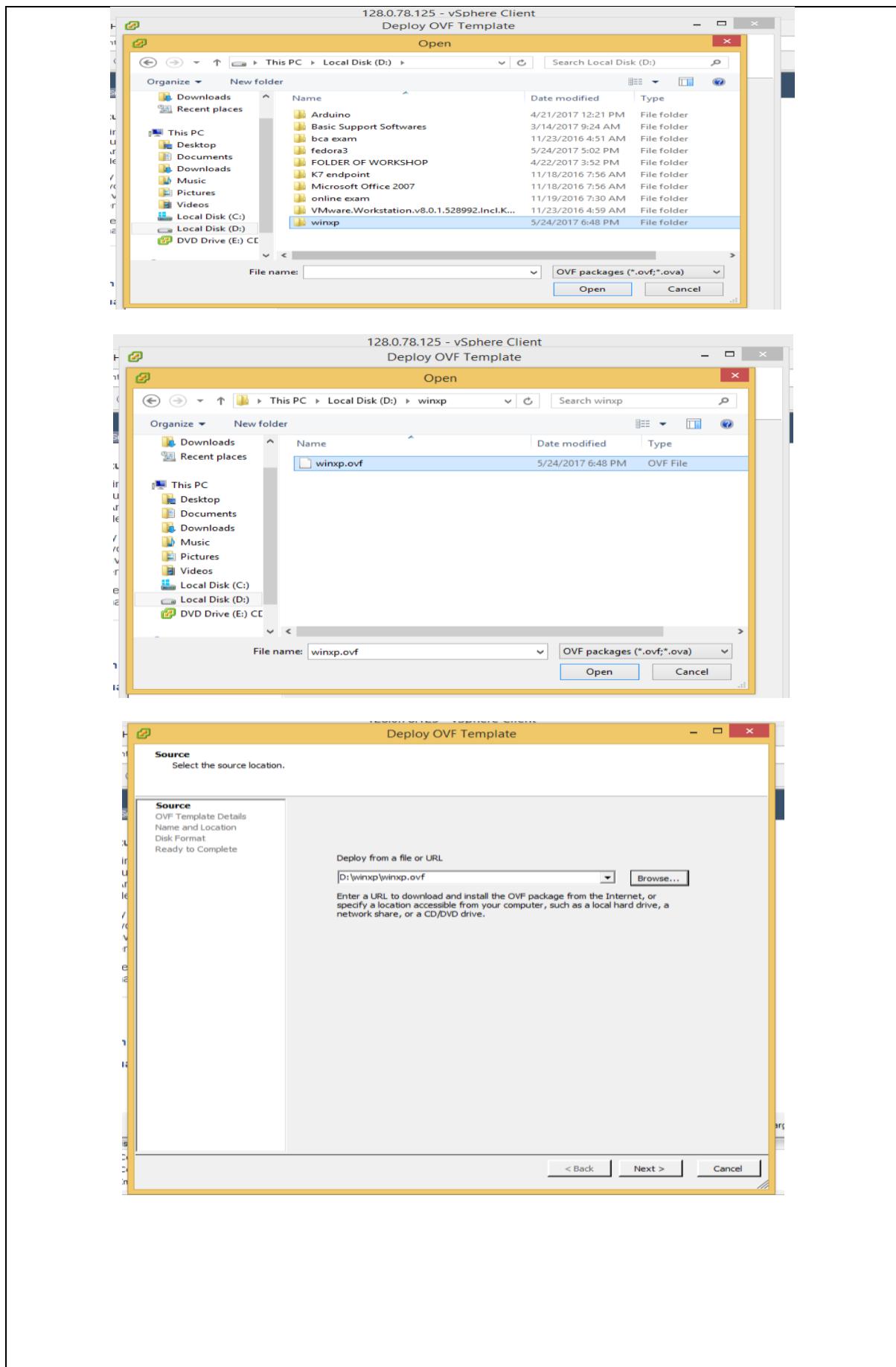
8. Go to file -> Deploy OVF Template

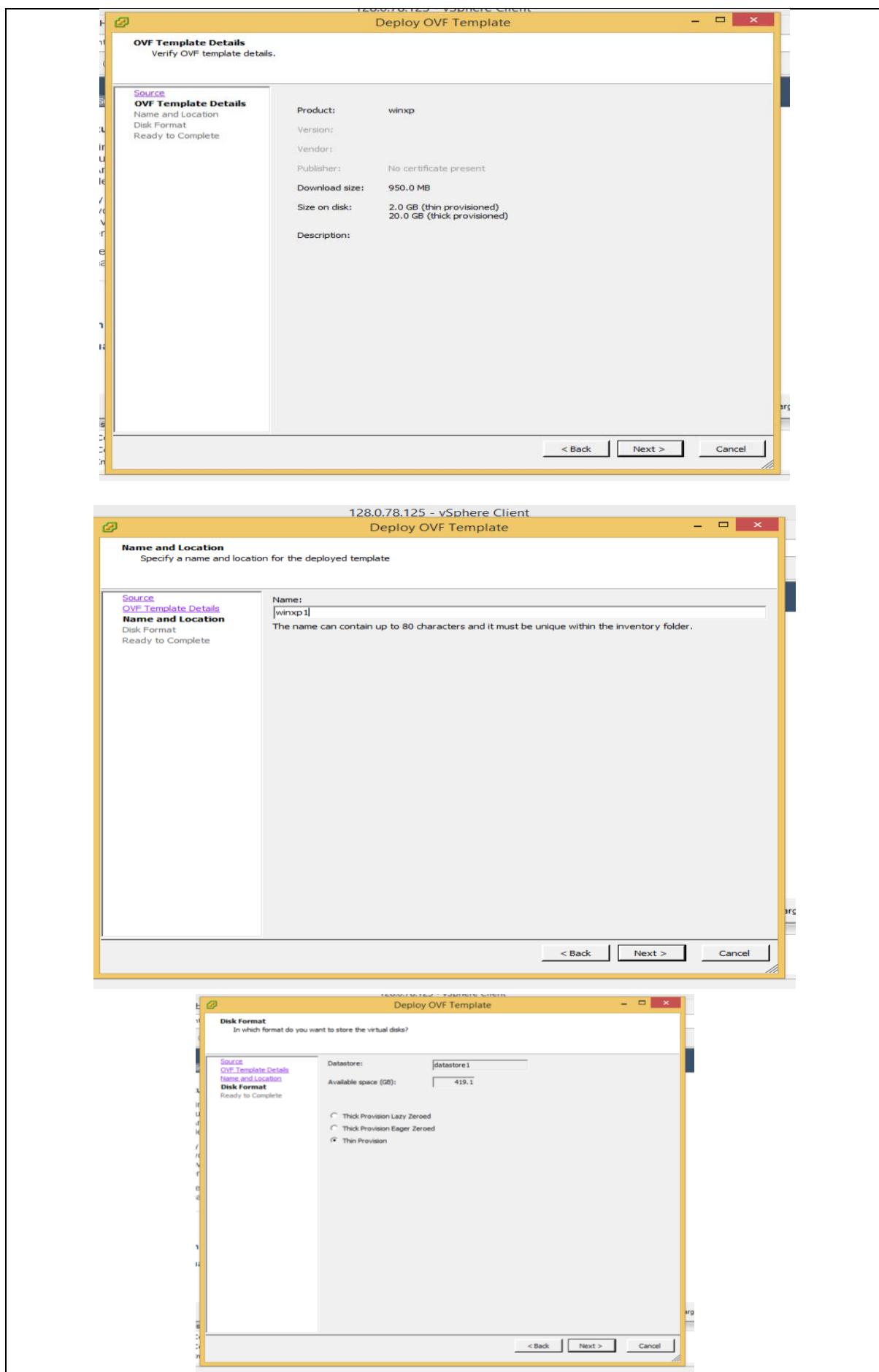


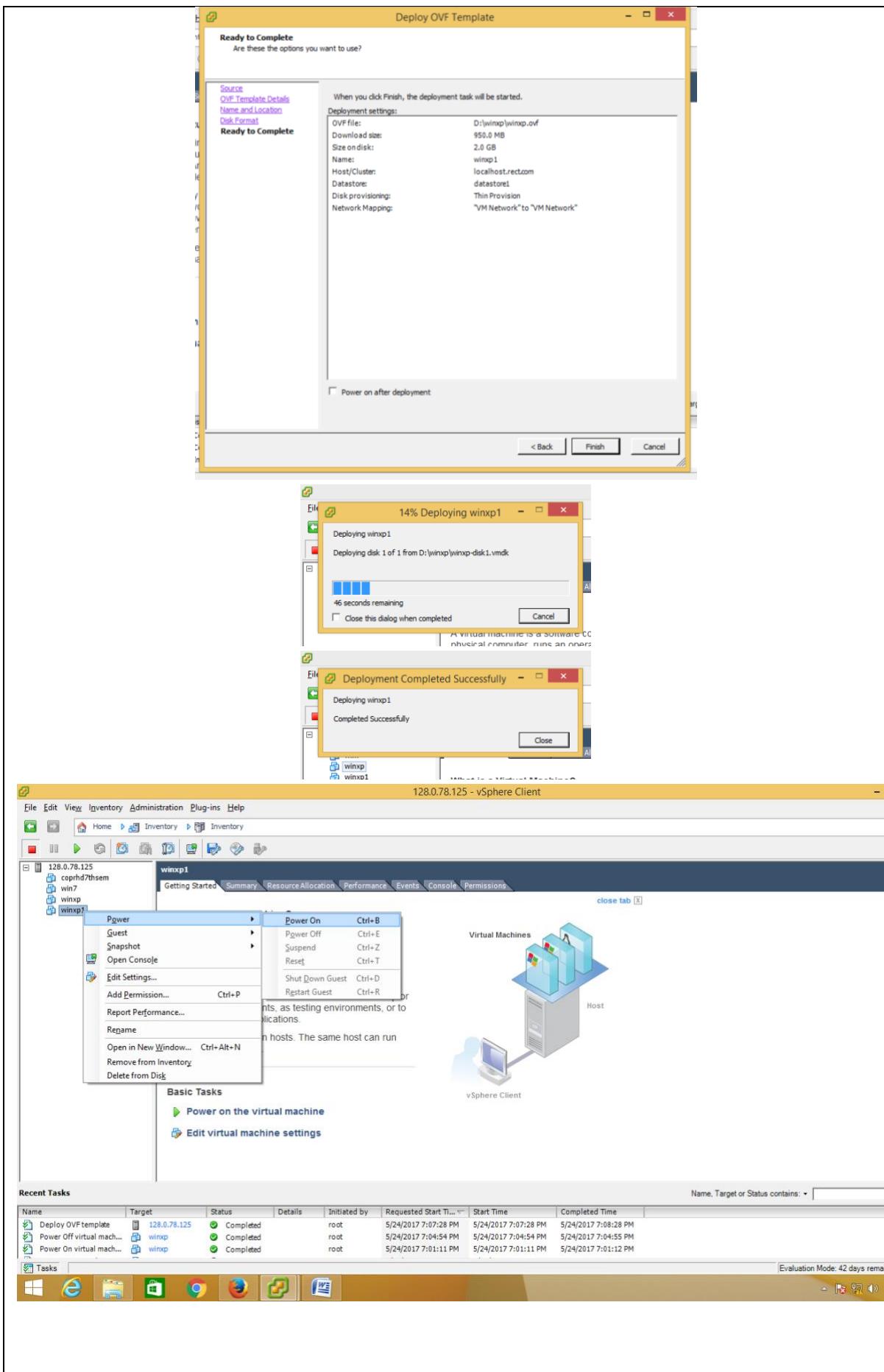
9. Select the source location from where the OVF file is to be deployed.



10. Select the exact copy of VM to be deployed among many files.

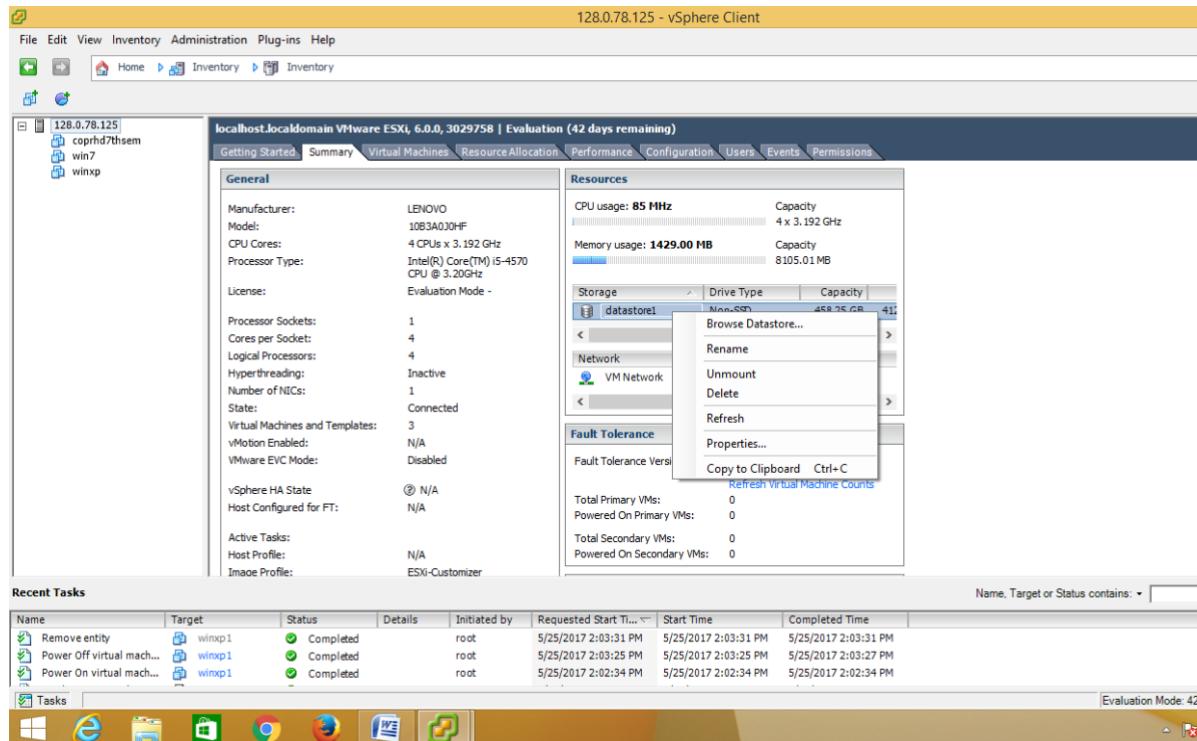




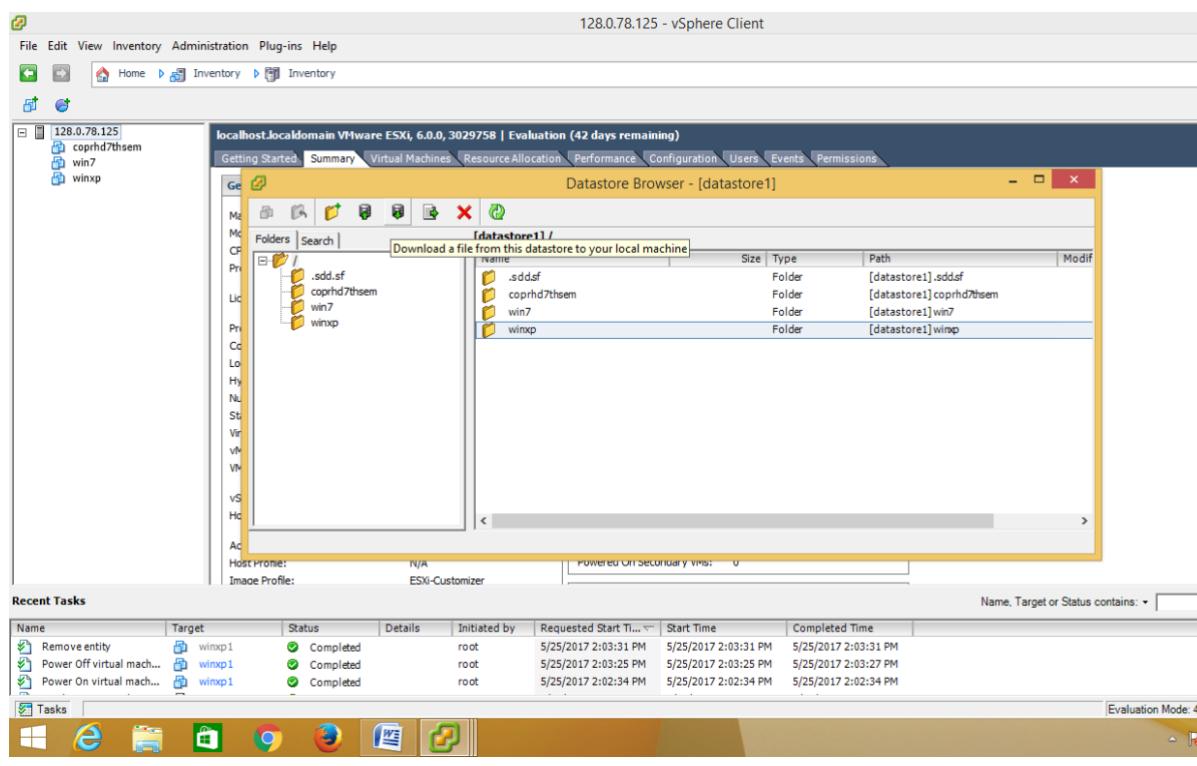


Method II:

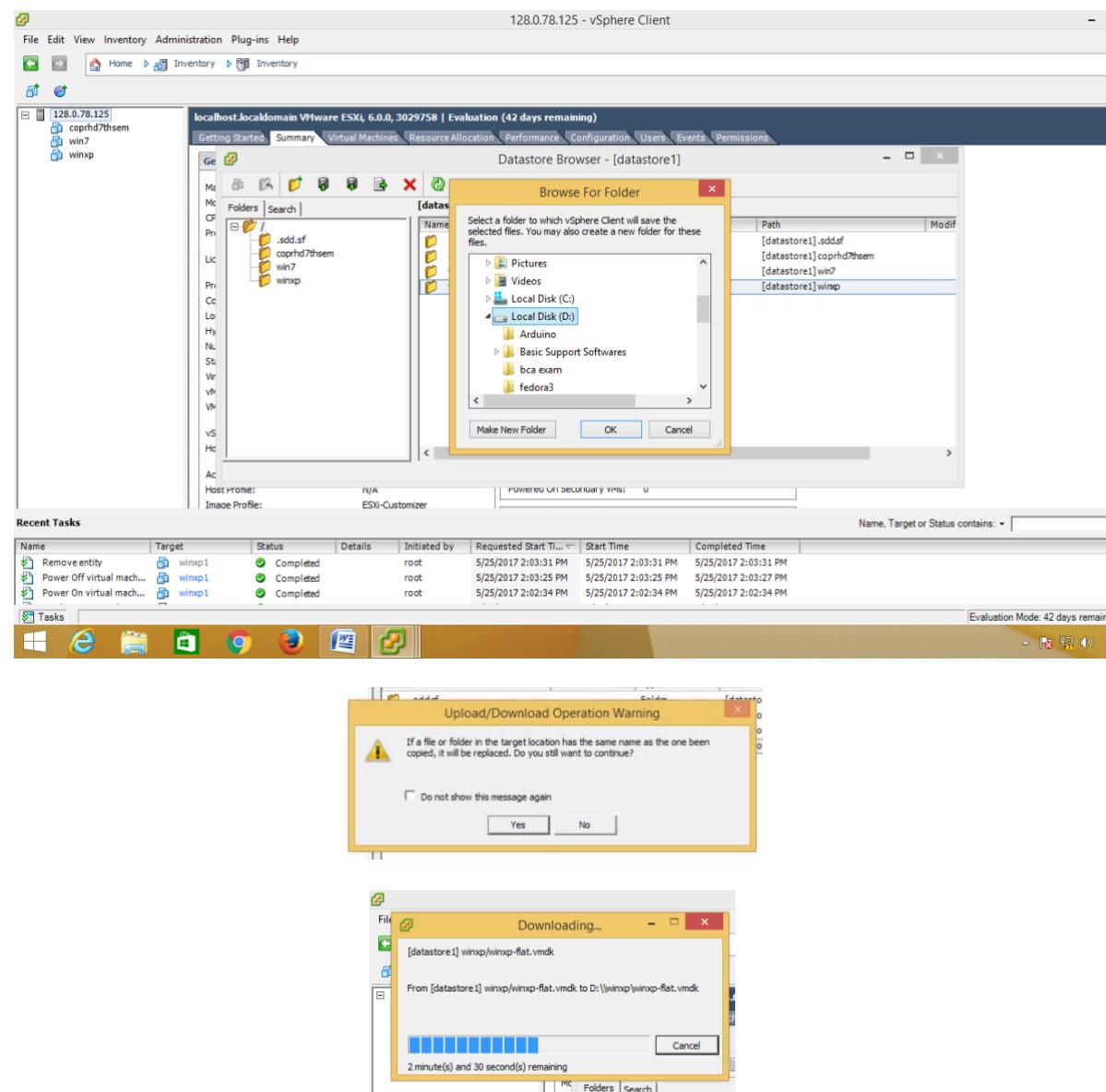
1. Select the ESXi, Right click on Datastore -> browse datastore.



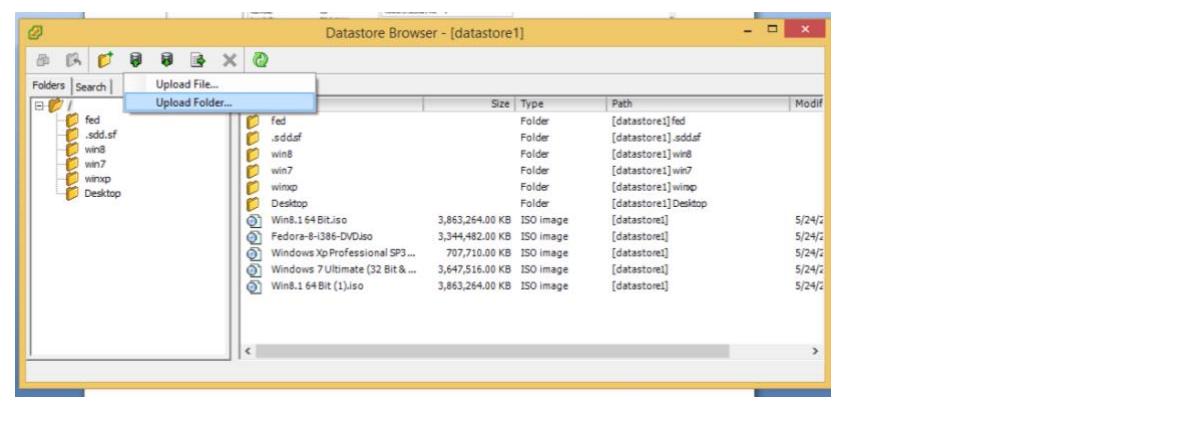
1. Select the folder of VM to be downloaded to physical machine
2. Select the option of “download a file from this data store to local machine”

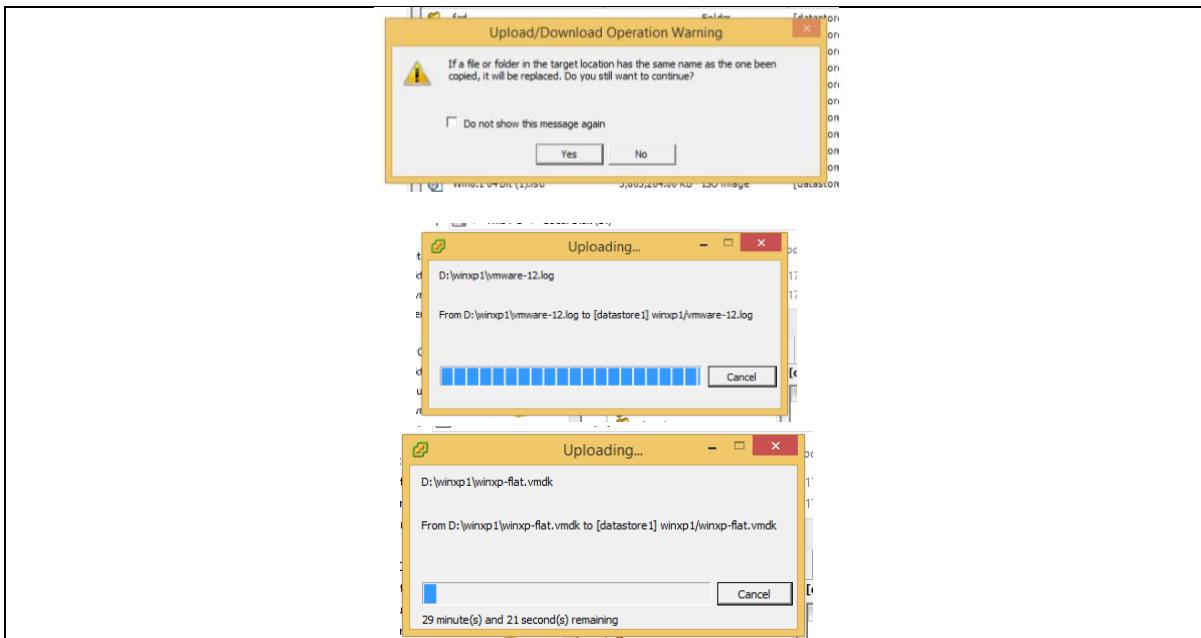


3. Select the storage location

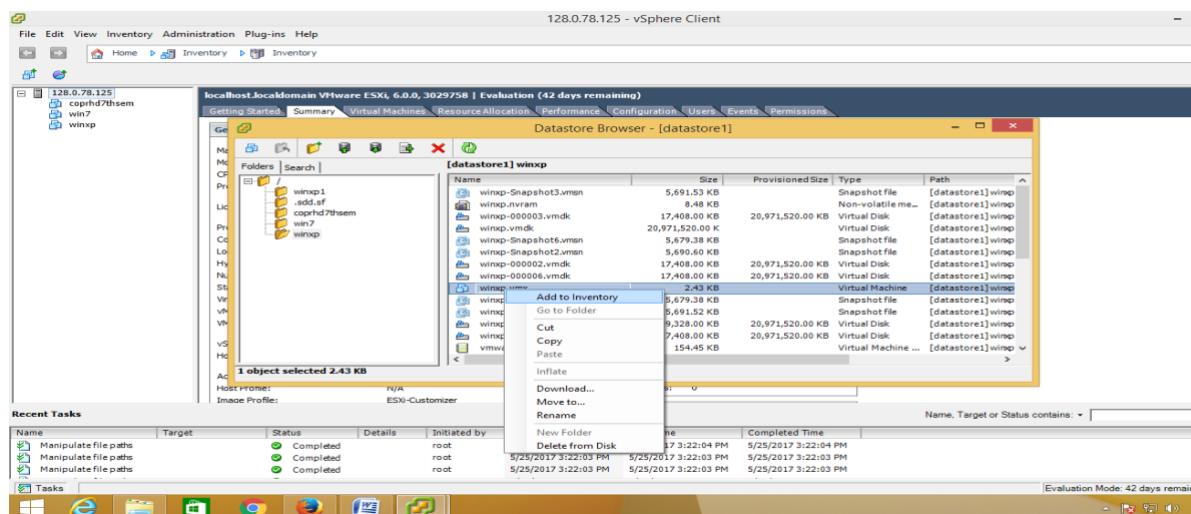


4. Go to vClient of any ESXi, Datastore -> Upload folder

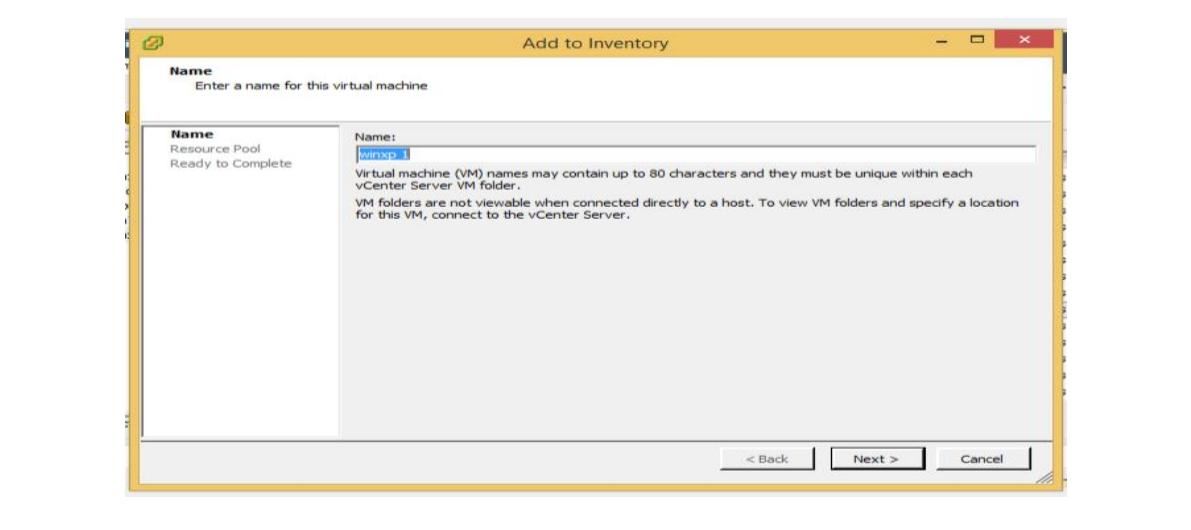




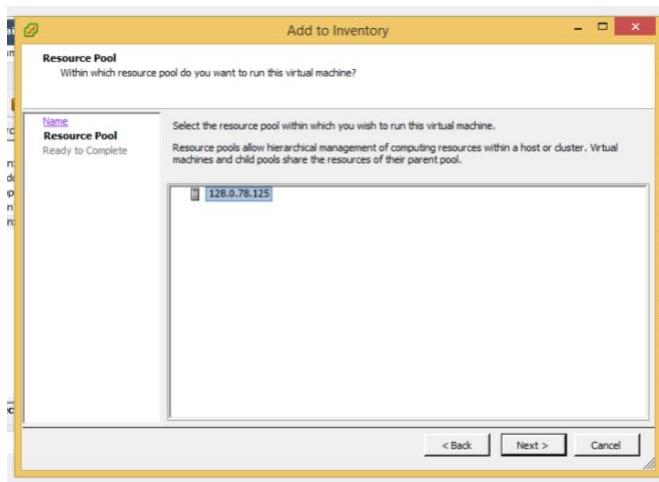
5. Once the folder of VM is uploaded successfully, then open the folder
6. Right click on the .vms file -> add to inventory



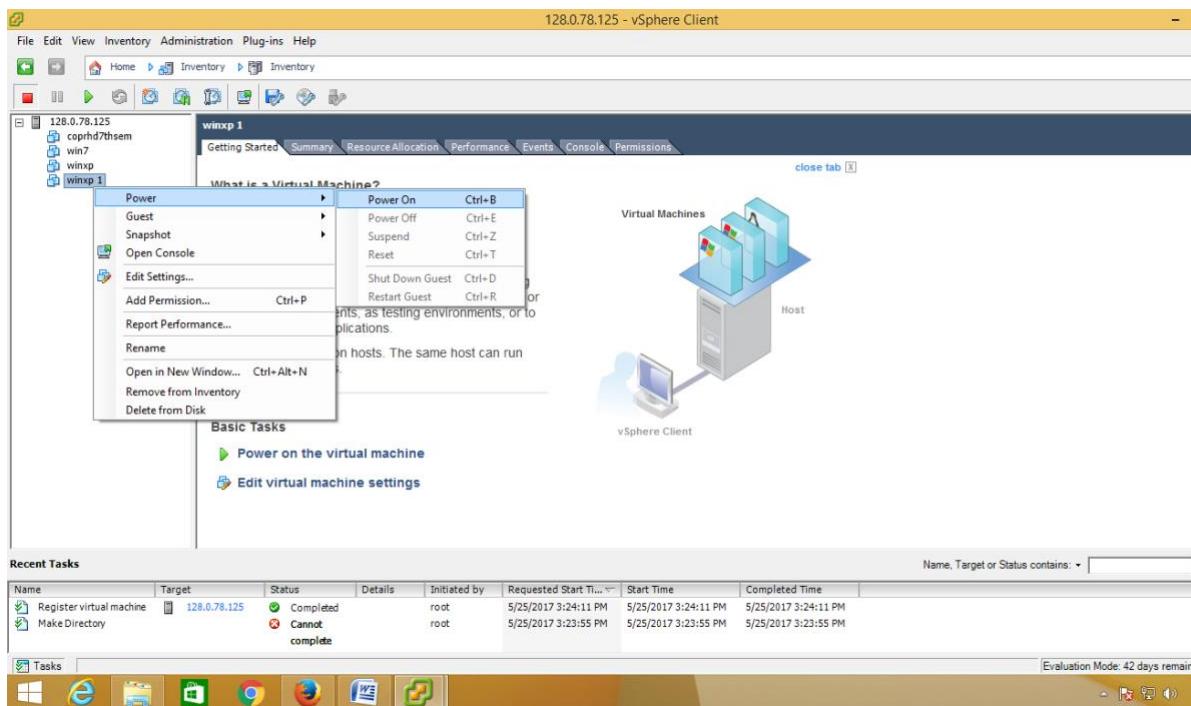
7. Provide the name for the new VM



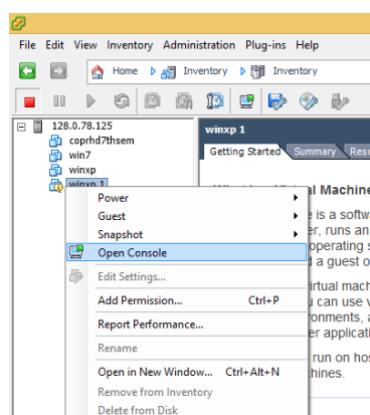
8. Select the destination ESXi



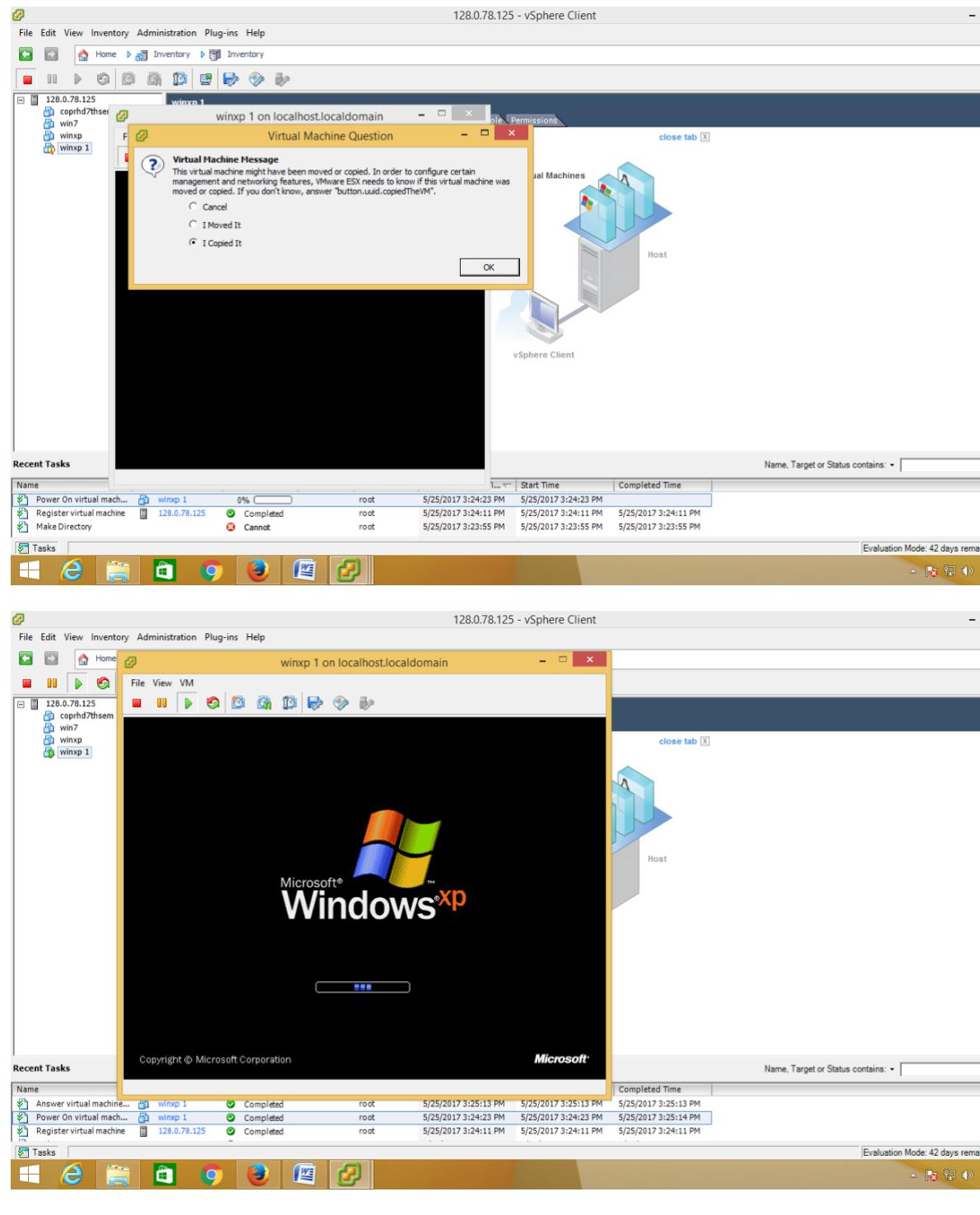
9. Once it is installed on to inventory then power ON the VM



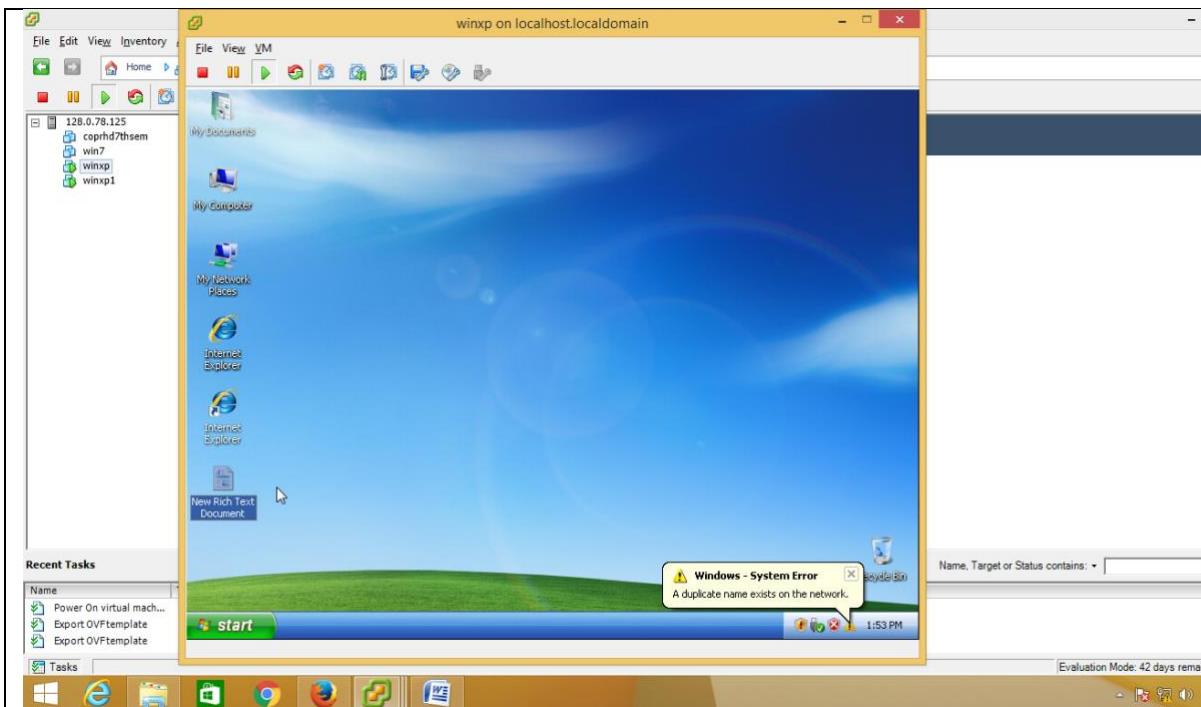
10. Open through the Console



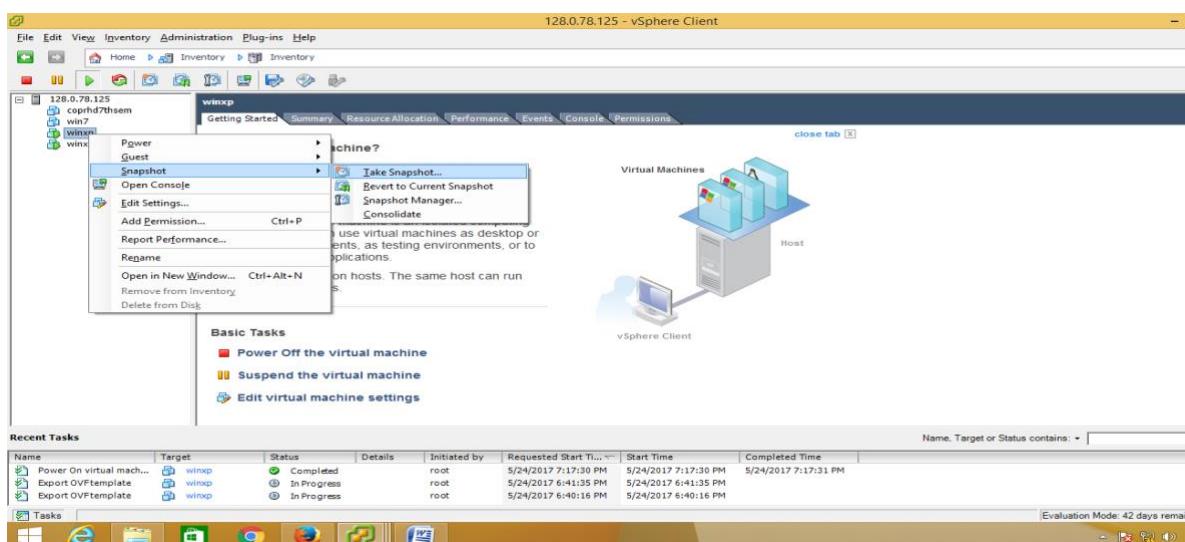
11. Select the options Moved – if you want to make the new copy replacing the old
Copied – if you wanted to create new by retaining the old.



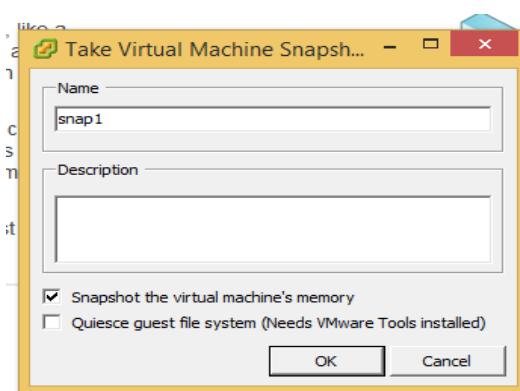
Session 6	
1	Problem Statement:
	Illustrate the backup-restore scenario.
2	Student Learning Outcomes:
	To figure out the different ways of backup and restore the sessions of VM
3	Theoretical Description:
	<p>A VMware snapshot: is a copy of the virtual machine's disk file (VMDK) at a given point in time. Snapshots provide a change log for the virtual disk and are used to restore a VM to a particular point in time when a failure or system error occurs. Snapshots alone do not provide backup.</p> <p>When you take a snapshot, you capture the state of the virtual machine settings and the virtual disk. If you are taking a memory snapshot, you also capture the memory state of the virtual machine. These states are saved to files that reside with the virtual machine's base files.</p> <p>A snapshot consists of files that are stored on a supported storage device. A Take Snapshot operation creates .vmdk, -delta.vmdk, .vmsd, and .vmsn files. By default, the first and all delta disks are stored with the base .vmdk file. The .vmsd and .vmsn files are stored in the virtual machine directory.</p>
4	Requirements
	<ul style="list-style-type: none"> ▪ If you are taking a memory snapshot of a virtual machine that has multiple disks in different disk modes, verify that the virtual machine is powered off. For example, if you have a special purpose configuration that requires you to use an independent disk, you must power off the virtual machine before taking a snapshot. ▪ To capture the memory state of the virtual machine, verify that the virtual machine is powered on. ▪ To quiesce the virtual machine files, verify that the virtual machine is powered on and that VMware Tools is installed ▪ Required privilege: Virtual machine.State. Create snapshot on the virtual machine.
5	Procedure
	<ol style="list-style-type: none"> 1. The VM, to which the backup is required should be in ON



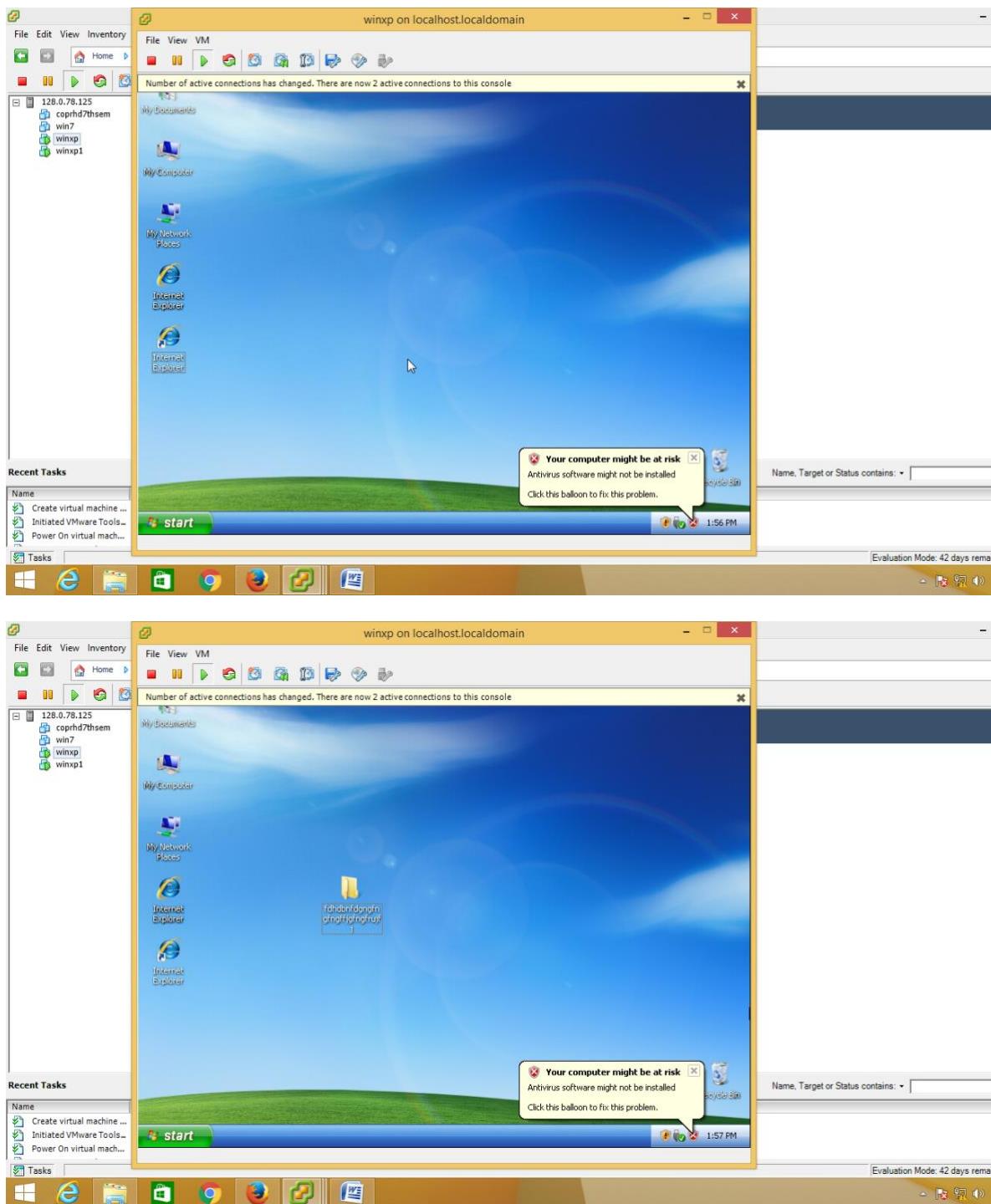
2. Right click on VM -> Snapshot -> Take Snapshot



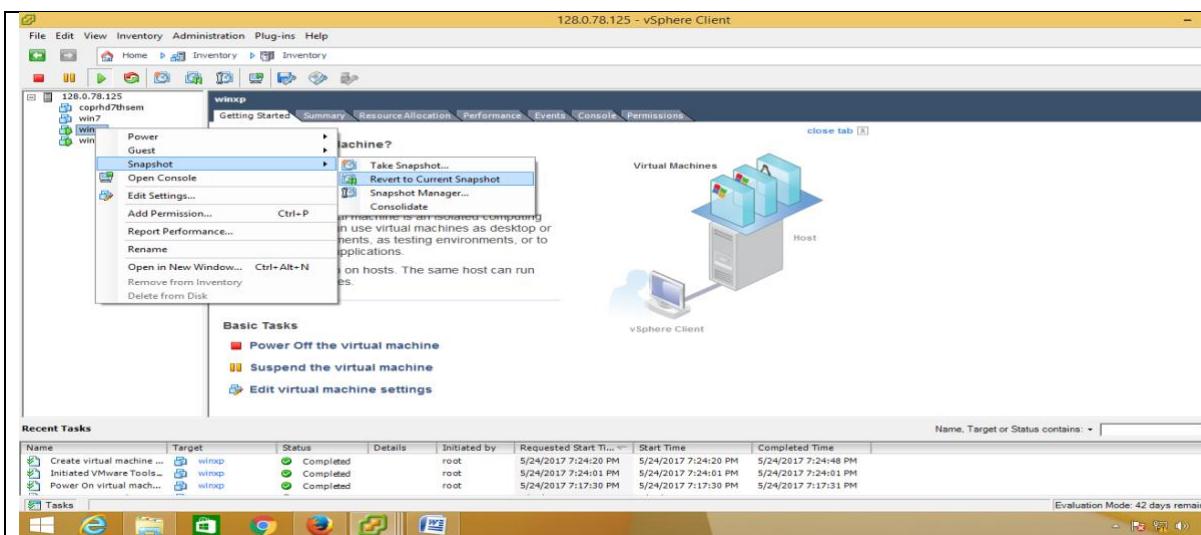
3. Provide the name for snapshot it may also be based on the time of backup.



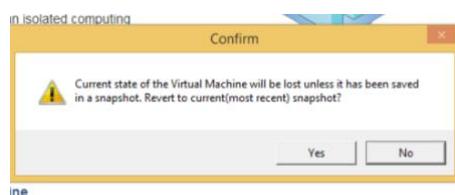
4. Then make changes/updates in the current VM



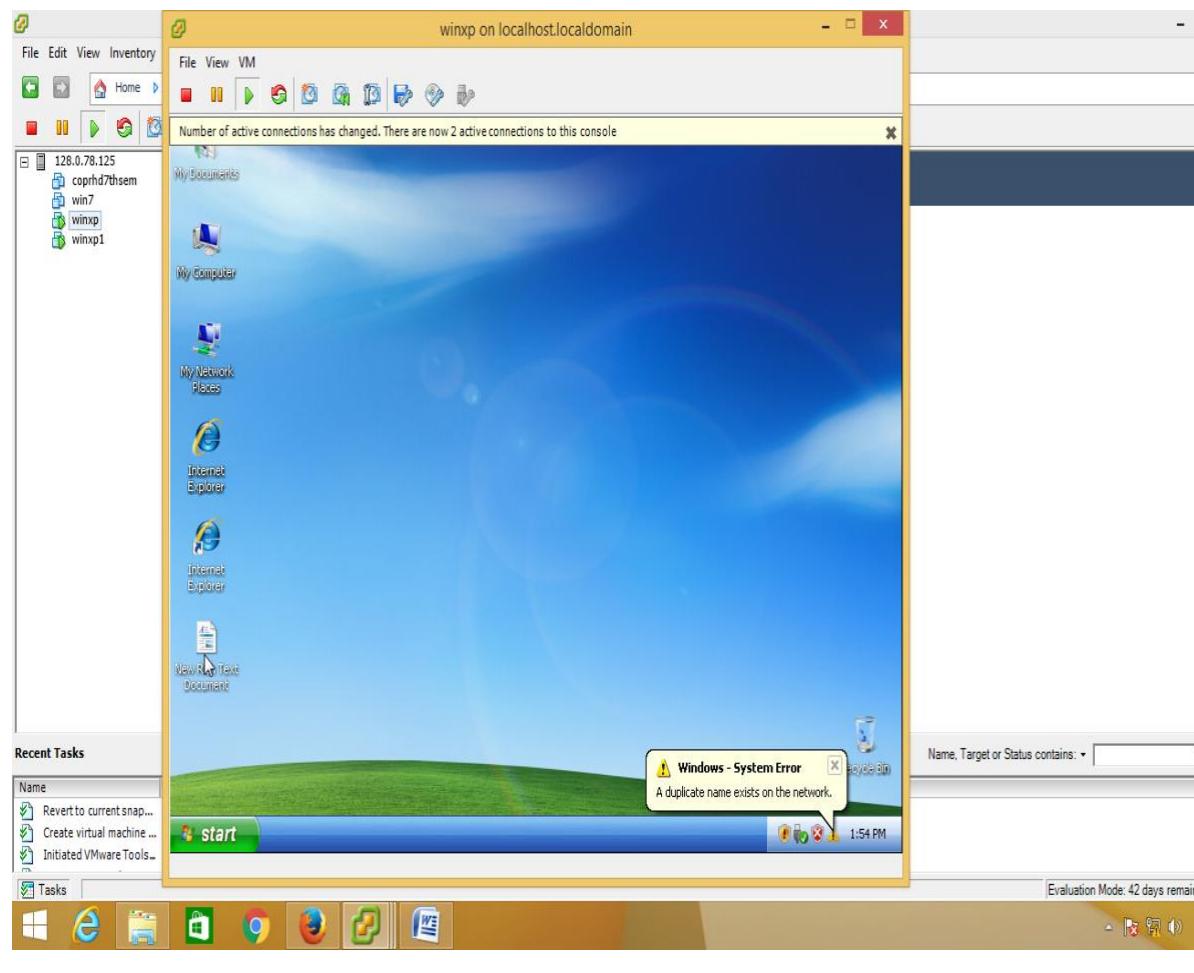
5. For retrieving / get back to initial state of VM,
6. Right click on VM -> Snapshot -> Revert to Current Snapshot



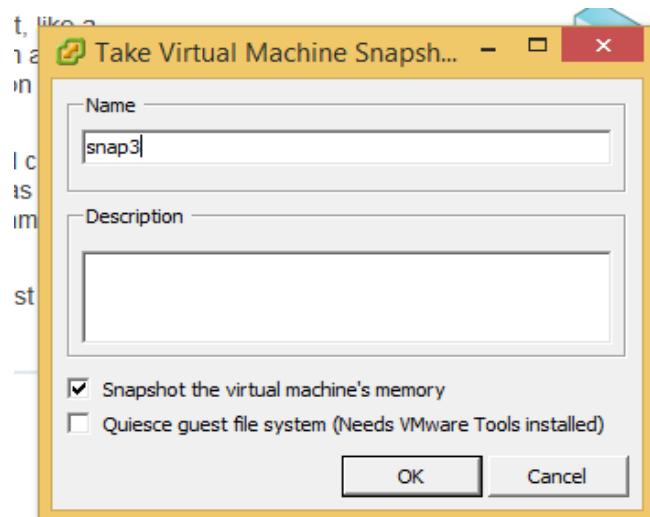
7. Accept for the current modification to be done in the VM



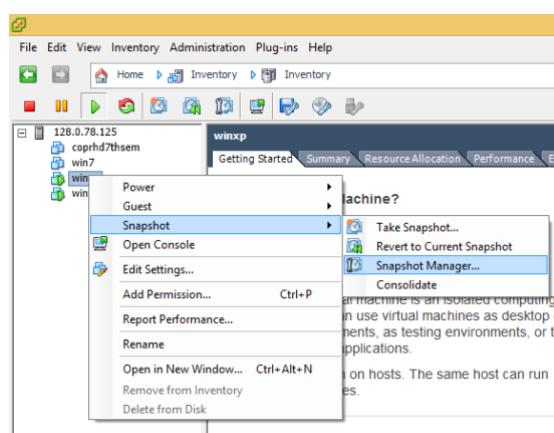
8. Get back to original state of VM



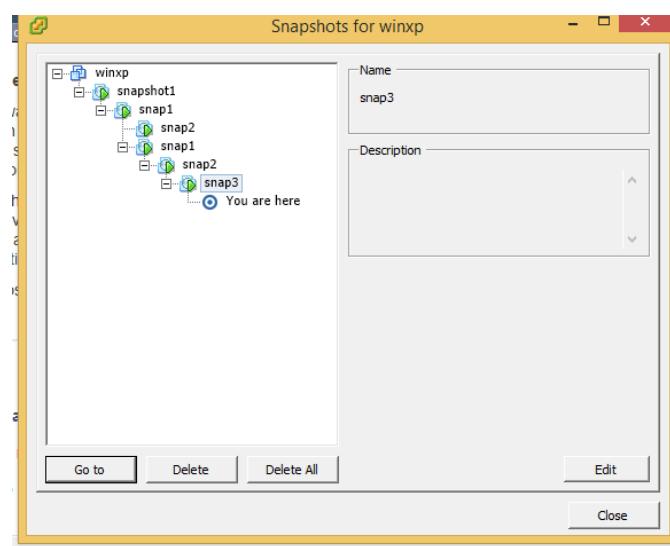
9. If you ever taken too many snapshots for the same VM, need to manage and revert to required state of VM



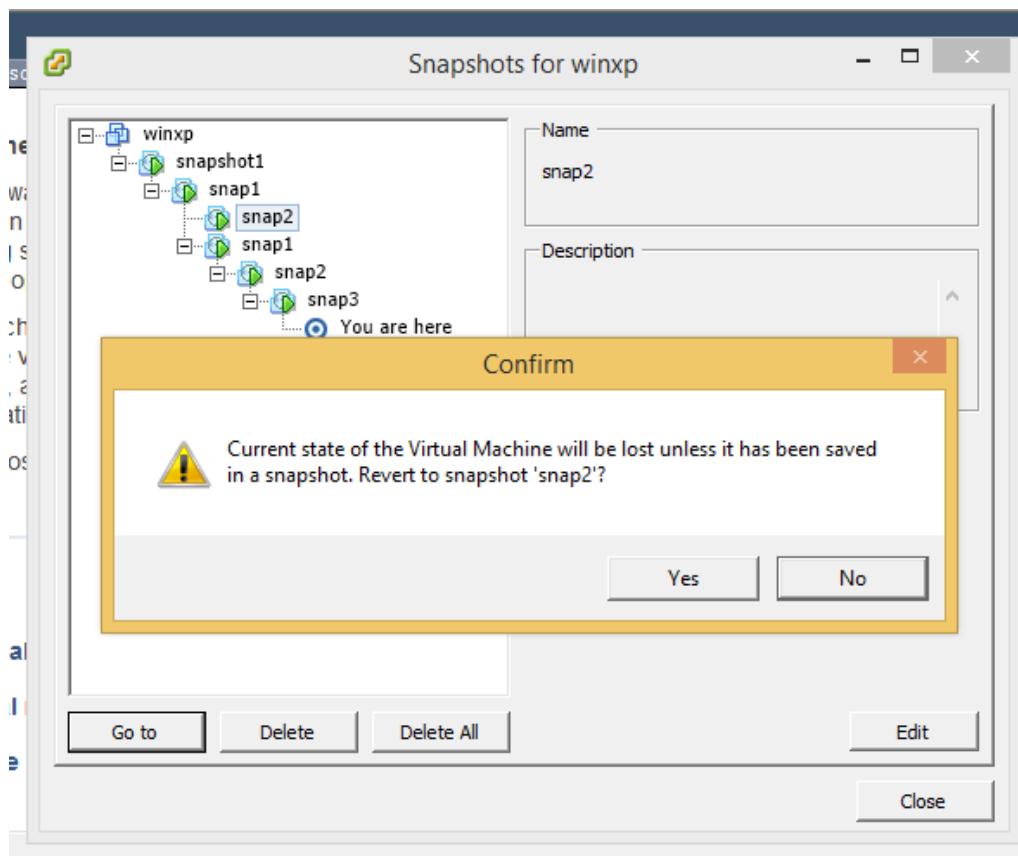
10. Right click on VM -> Snapshot -> Snapshot manager



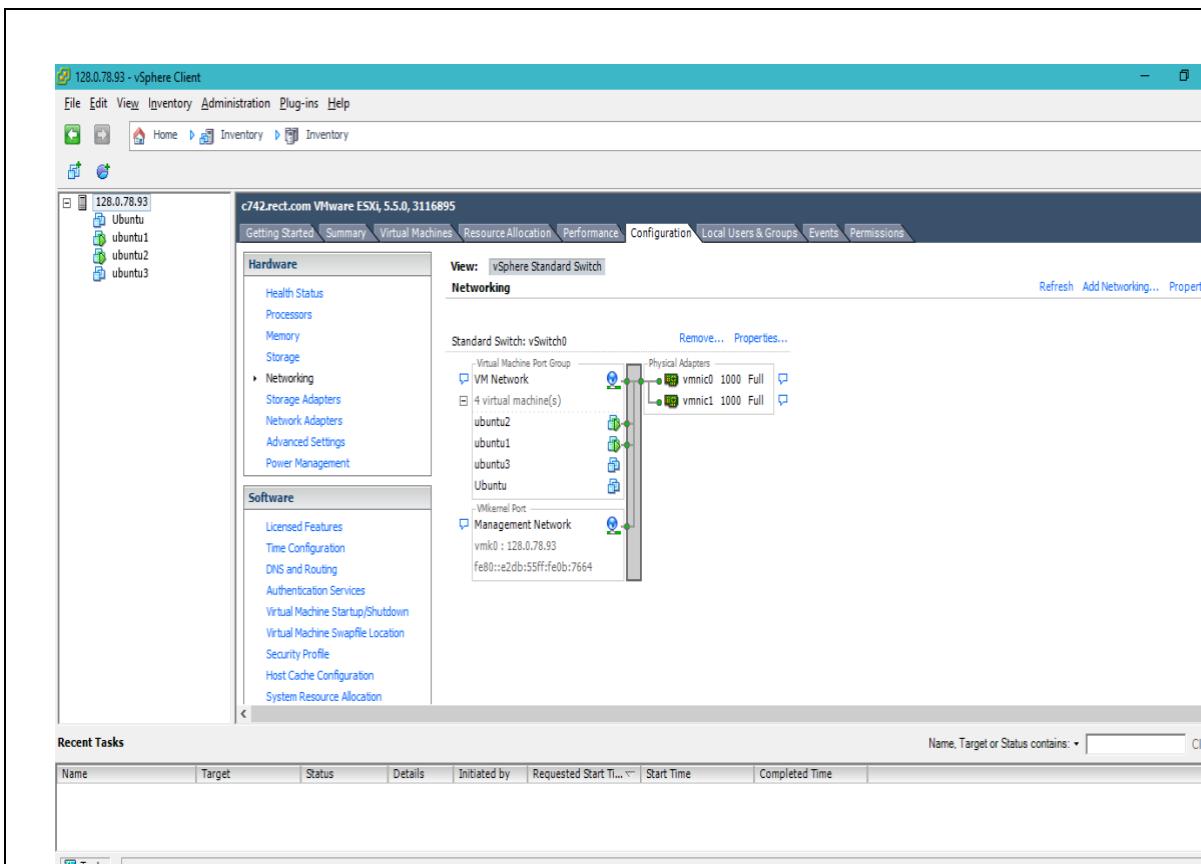
11. You can see the multiple snapshots taken for the VM and also current state of VM



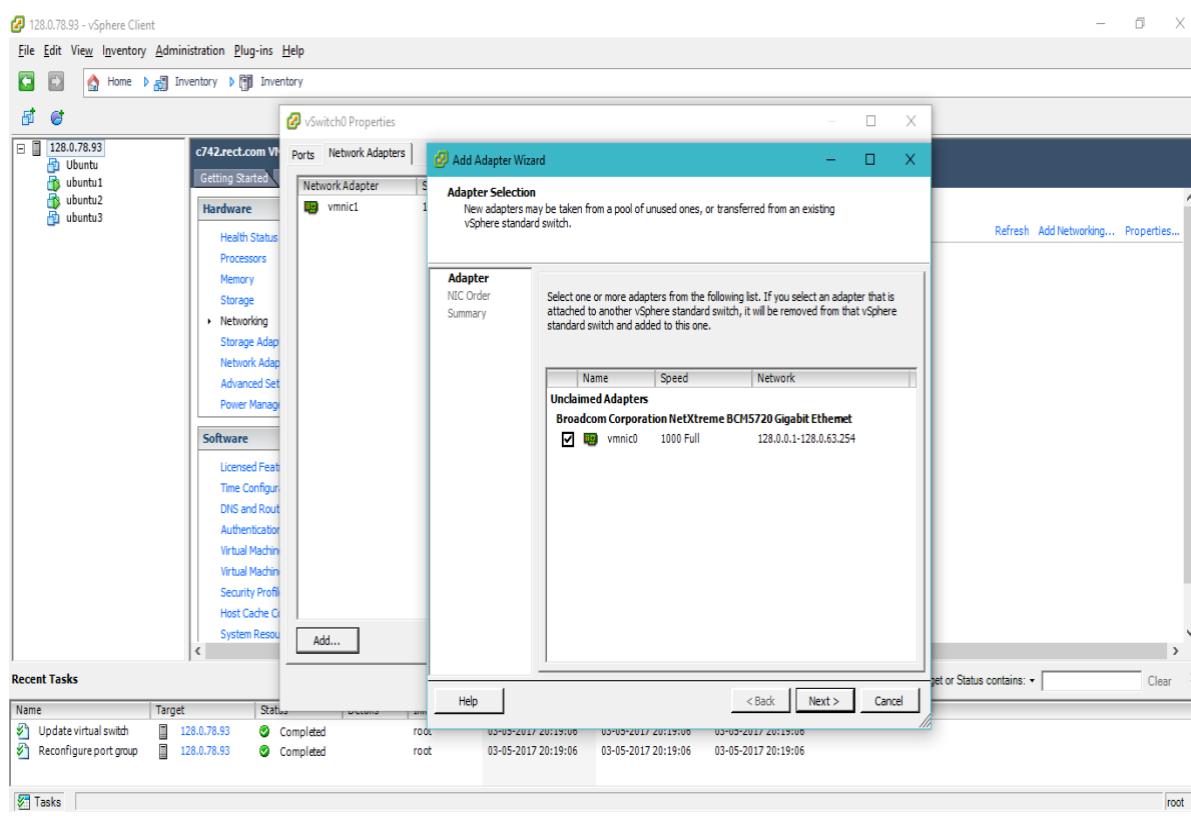
12. You can select any snapshot and revert to respective state of VM



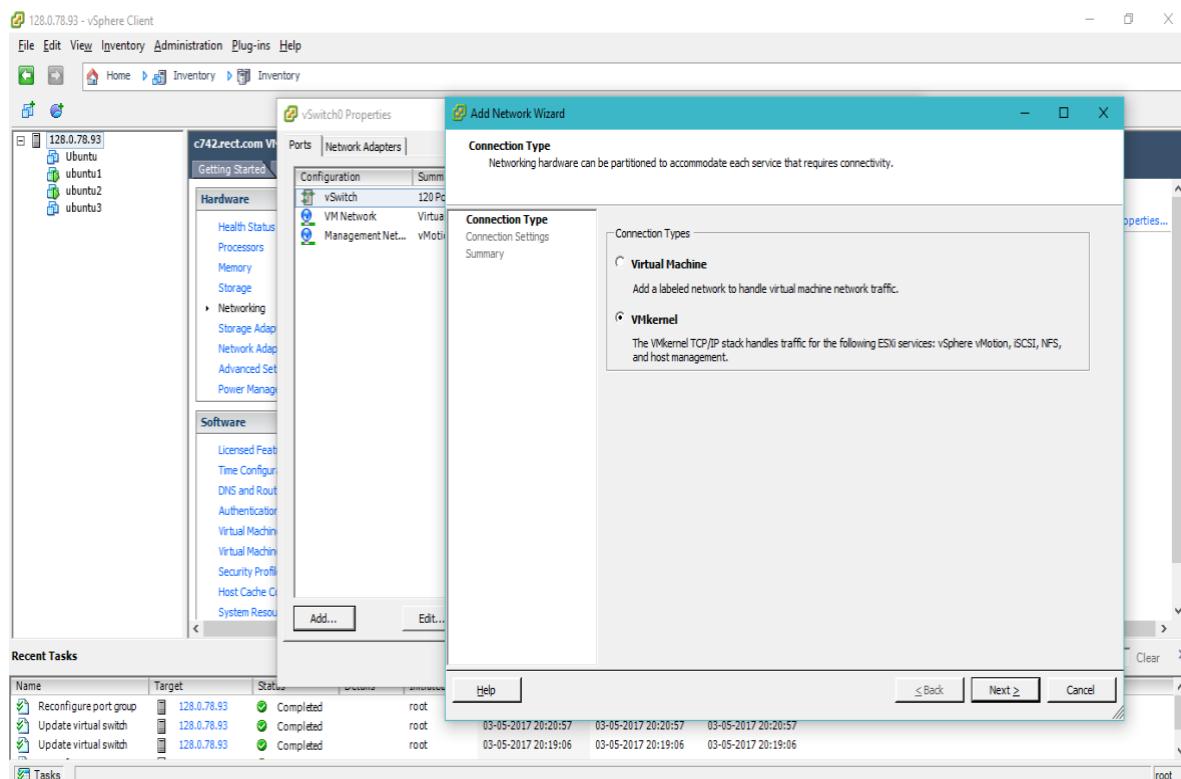
Session 7	
1	Problem Statement: Add iSCSI adapter and modify the relative parameters. Create a new VMkernel using iSCSI port binding.
2	Student Learning Outcomes: To provide the adapter for the already created VM's
3	Theoretical Description: <p>The VMkernel is the liaison between virtual machines (VMs) and the physical hardware that supports them. VMware calls VMkernel a microkernel because it runs on bare metal, directly on VMware ESX hosts. The VMkernel is responsible for allocating memory, scheduling CPUs and providing other hardware abstraction and operating system (OS) services.</p> <p>Virtual switch : Like a physical switch, a virtual switch lets you connect other networking components together. Virtual switches are created as needed by the VMware Workstation software, up to a total of nine switches. You can connect one or more virtual machines to a switch.</p> <p>By default, a few of the switches and the networks associated with them are used for special named configurations:</p> <ul style="list-style-type: none"> ▪ The bridged network uses VMnet0. ▪ The host-only network uses VMnet1. ▪ The NAT network uses VMnet8. <p>The other available networks are simply named VMnet2, VMnet3, VMnet4, and so on.</p> <p>To connect a virtual machine to a switch: In the virtual machine settings editor, select the virtual network adapter to connect, and then configure the adapter to use the desired virtual network.</p> <p>iSCSI SANs use Ethernet connections between computer systems, or host servers, and high performance storage subsystems. The SAN components include iSCSI host bus adapters (HBAs) or Network Interface Cards (NICs) in the host servers, switches and routers that transport the storage traffic, cables, storage processors (SPs), and storage disk systems.</p>
4	Requirements If your host has more than one physical network adapter for software and dependent hardware iSCSI, use the adapters for multipathing.
5	Procedure 1) Go to Configuration tab > click networking > properties



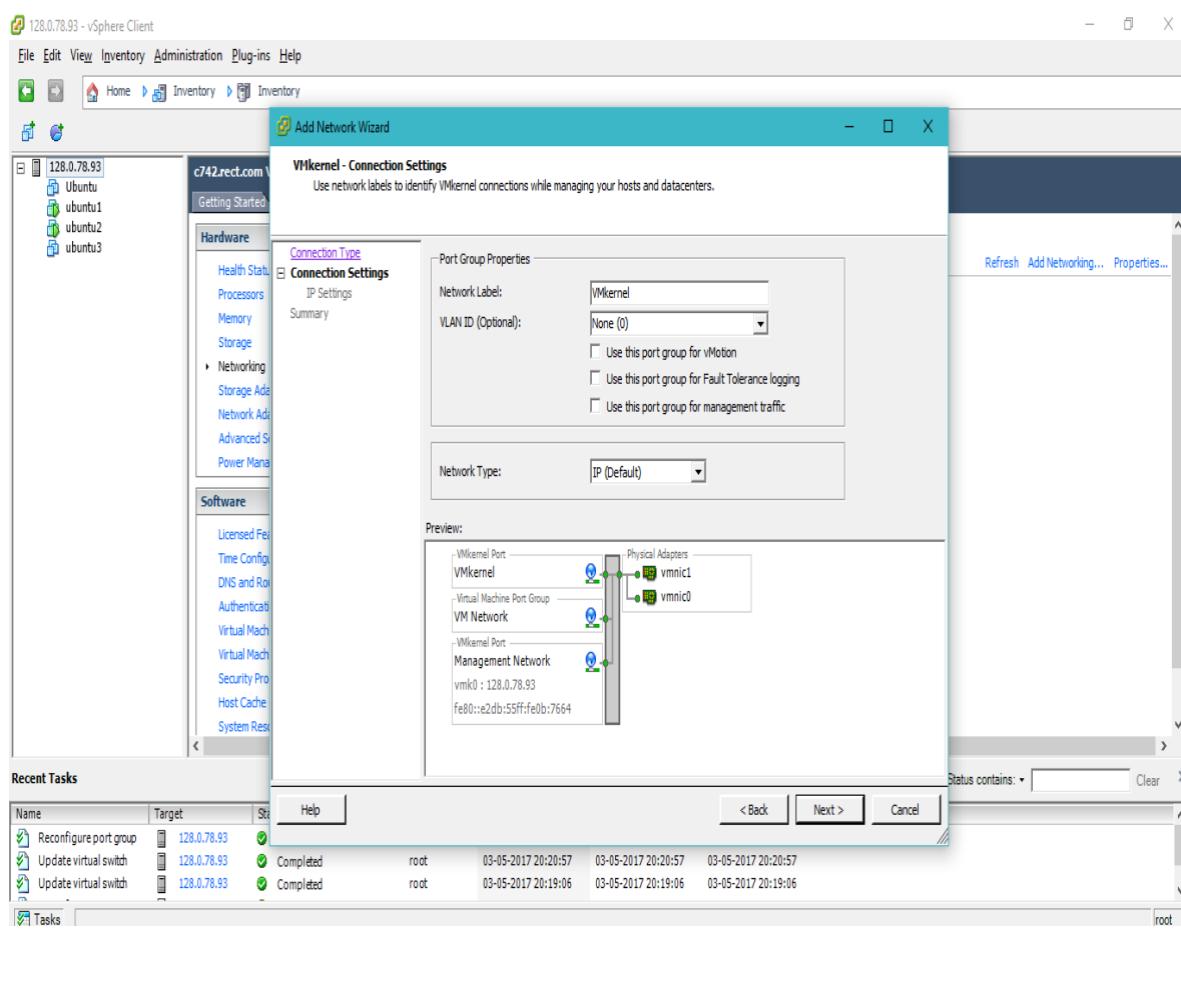
- 2) Select network adapter and click add.
- 3) Select the unclaimed adapter, click next and finish.



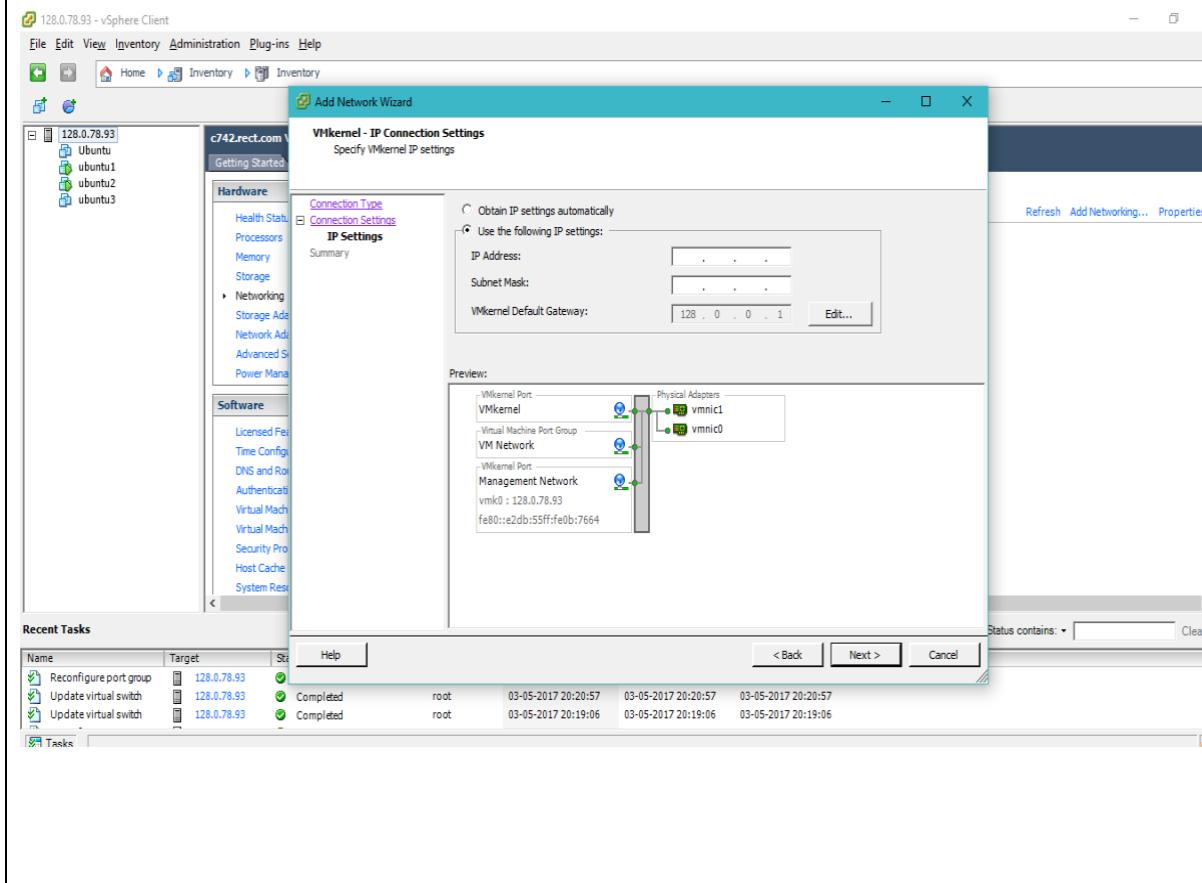
4) Now go to ports > click on add > select VMkernel and click next.



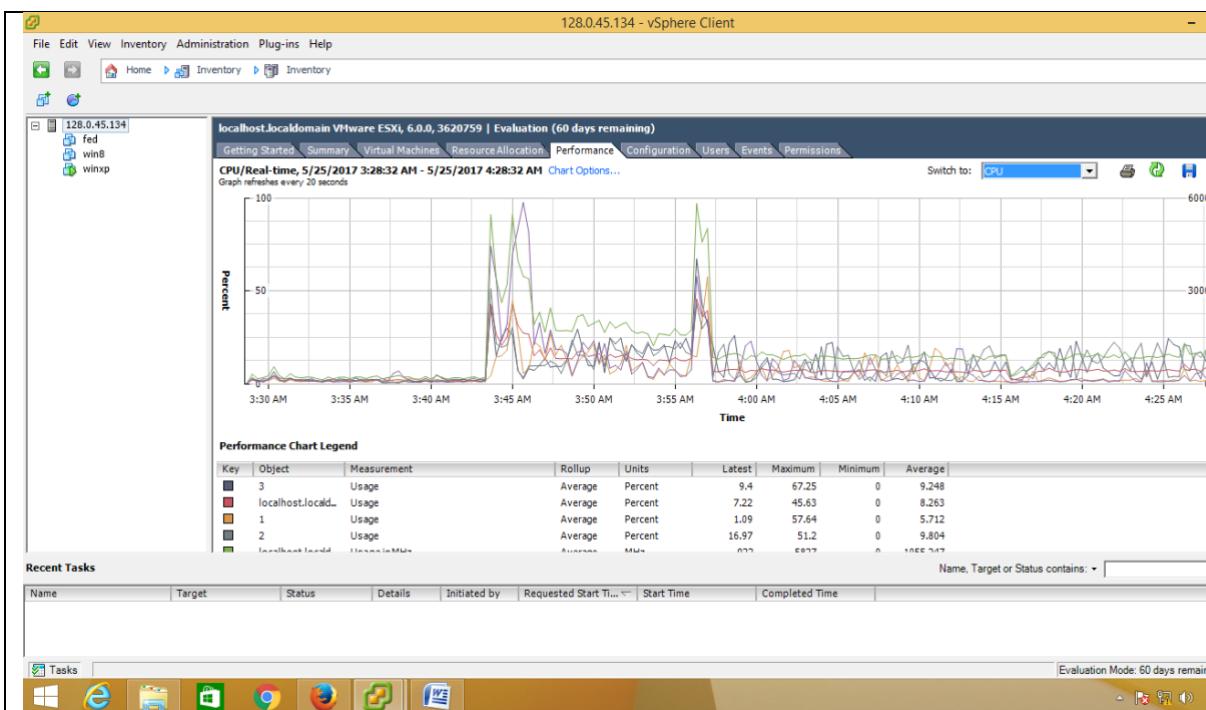
5) Select a label for the port and click next.



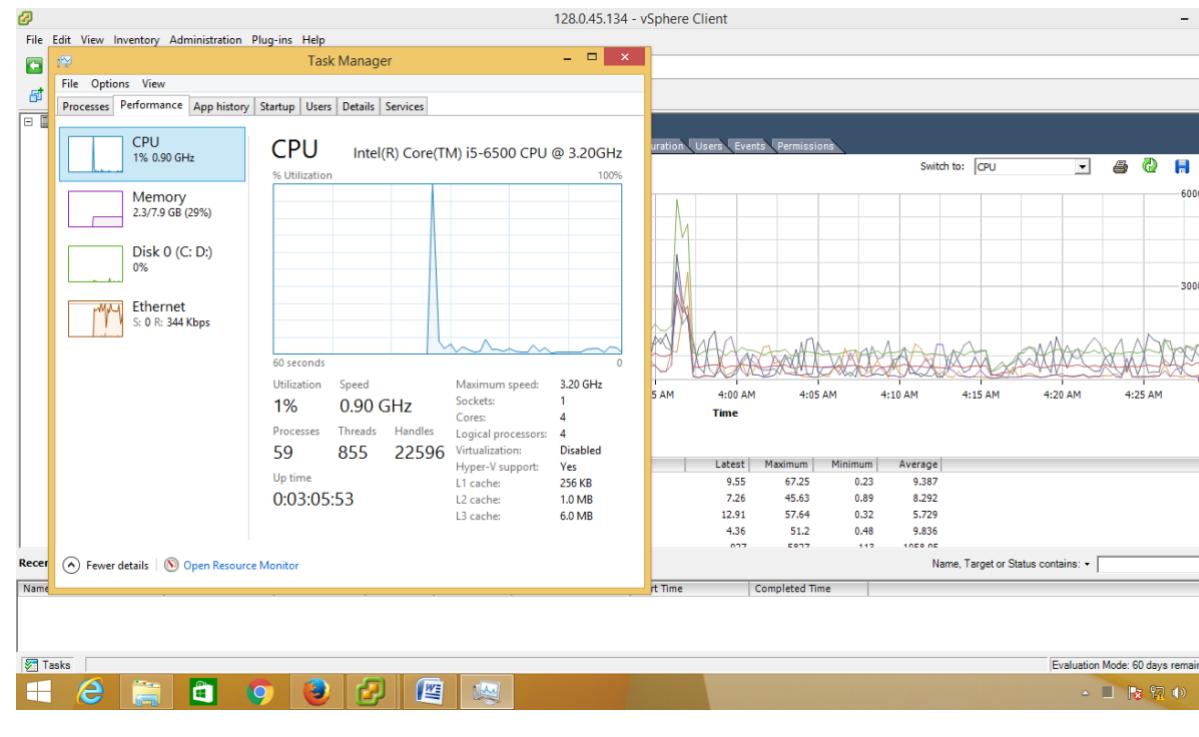
- 6) Enter an unused IP, Subnet mask click next and finish.

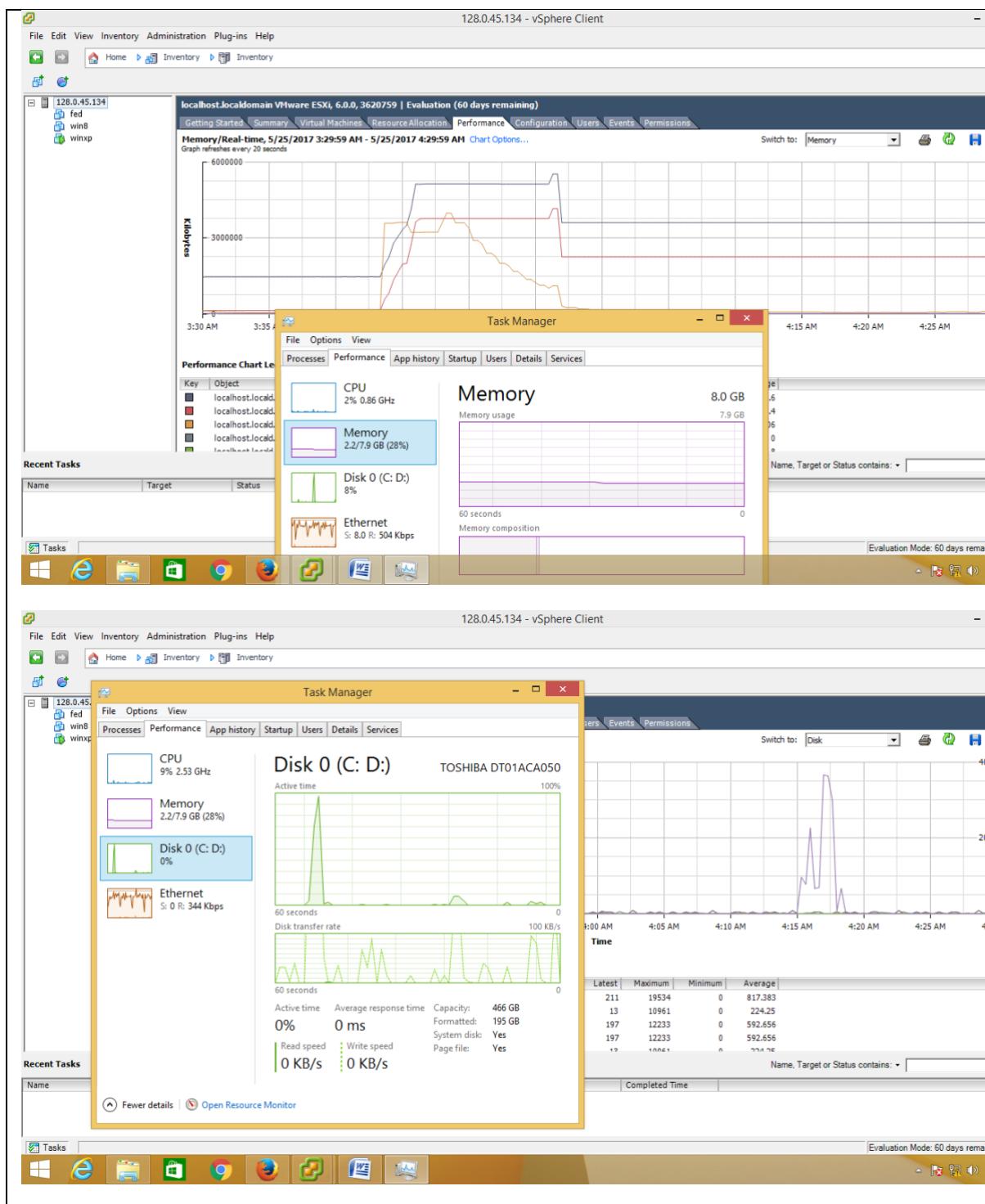


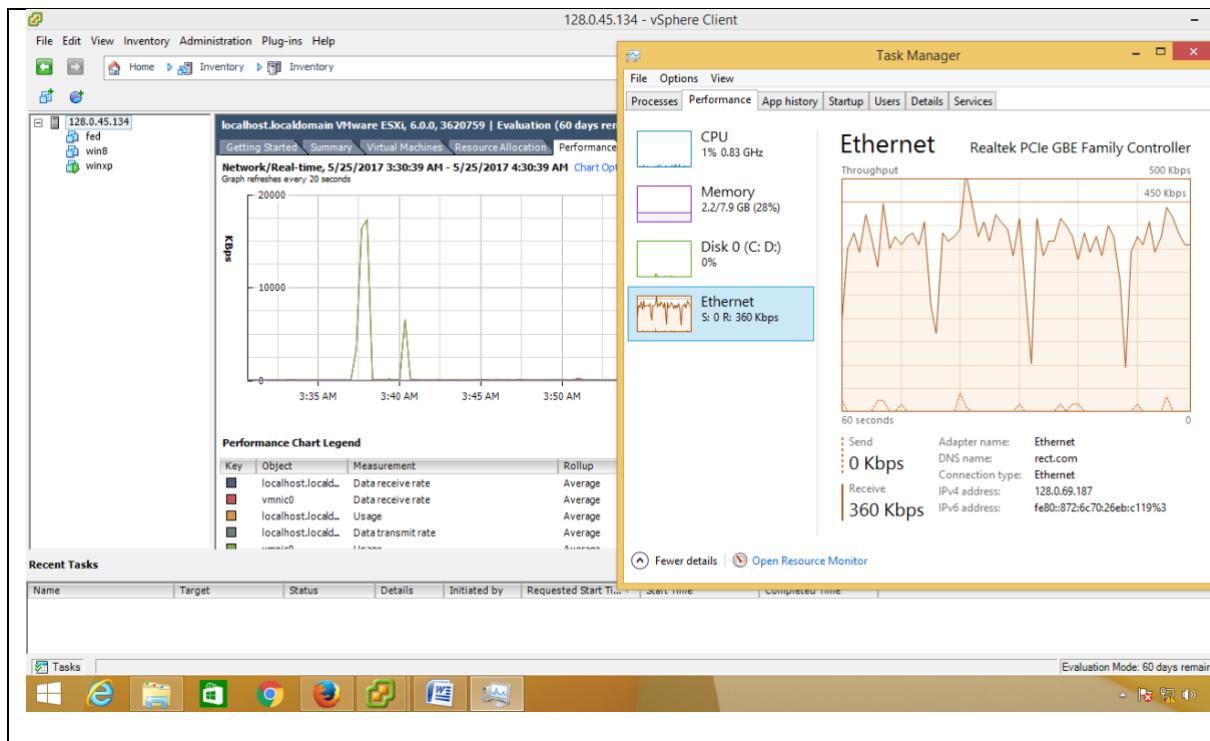
Session 8	
1	Problem Statement:
Performance Analysis of Virtual Machine vs Physical Machine	
2	Student Learning Outcomes:
To find the performance of VM's in ESXi when they are shut down and working state.	
3	Theoretical Description:
Performance seems to have two meanings: 1) The speed at which a computer operates, either theoretically (for example, using a formula for calculating Mtops - millions of theoretical instructions per second) or by counting operations or instructions performed (for example, (MIPS) - millions of instructions per second) during a benchmark test. The benchmark test usually involves some combination of work that attempts to imitate the kinds of work the computer does during actual use. Sometimes performance is expressed for each of several different benchmarks. 2) The total effectiveness of a computer system, including throughput , individual response time , and availability. Computer performance metrics (things to measure) include availability, response time, channel capacity, latency, completion time, service time, bandwidth, throughput, relative efficiency, scalability, performance per watt, compression ratio, instruction path length and speed up. CPU benchmarks are available	
4	Requirements
<ul style="list-style-type: none">▪ Physical machine on which the VM is hosted▪ Too many virtual machine hosted on physical machine.▪ Virtual machines hosted on bare metal ESXi	
5	Procedure
1. Select the ESXi -> Performance -> select CPU	



2. Go to task manager of physical machine -> select performance







Session 9

Problem Statement:

Control an LED from Cloud Service using Esp8266 (Node MCU) programming with Arduino IDE.

Student Learning Outcomes:

Create IOT application using Cloud.

Theoretical Description:

About Esp8266: An open-source firmware and development kit that helps to prototype our IOT products. It is interactive, programmable, low cost, simple, smart, and wi-fi enabled. it provides advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware. NodeMCU Dev Board is based on widely explored esp8266 System on Chip from Expressif. It combined features of WIFI access point and station + microcontroller and uses simple [LUA](#) based programming language.

Arduino is a physical computing platform based on a simple I/O board and a development environment that implements the Processing/Wiring language.

An Arduino board consists of an Atmel AVR microcontroller (ATmega168 in newer versions, ATmega8 in older versions) and complementary components to facilitate programming and incorporation into other circuits. Each board includes at least a 5-volt linear regulator and a 16MHz crystal oscillator. The microcontroller is pre-programmed with a boot loader so that an external programmer is not necessary.

The Arduino IDE is a cross-platform Java application that serves as a code editor and compiler and is also capable of transferring firmware serially to the board.

The programming language is derived from Wiring, a C-like language that provides similar functionality.

Requirements:

1. Esp8266 (NodeMCU)
2. Arduino IDE
3. LED
4. Breadboard
5. Jumper wires

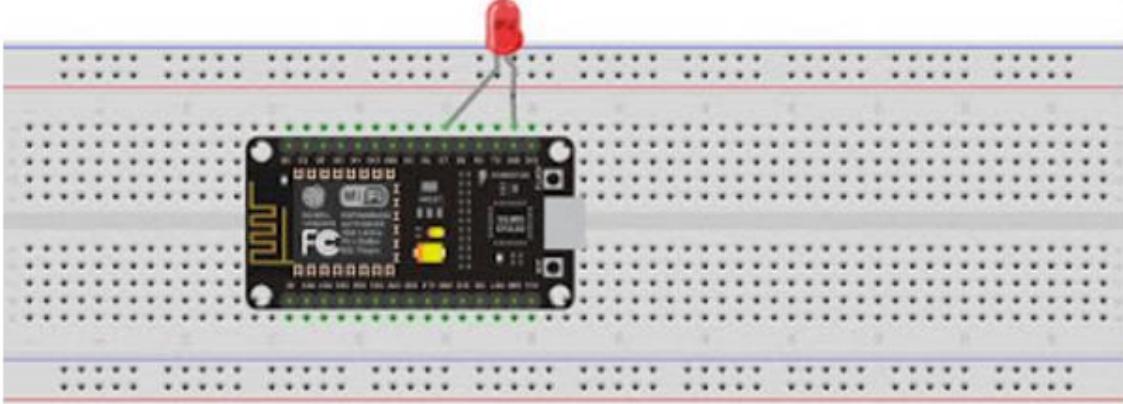
Procedure:

1. Make the connection as shown in the below table.

Note: Long leg of the LED is positive. Short leg is negative.

LED	NodeMCU
LED +ve	GPIO13

LED -ve	GND	

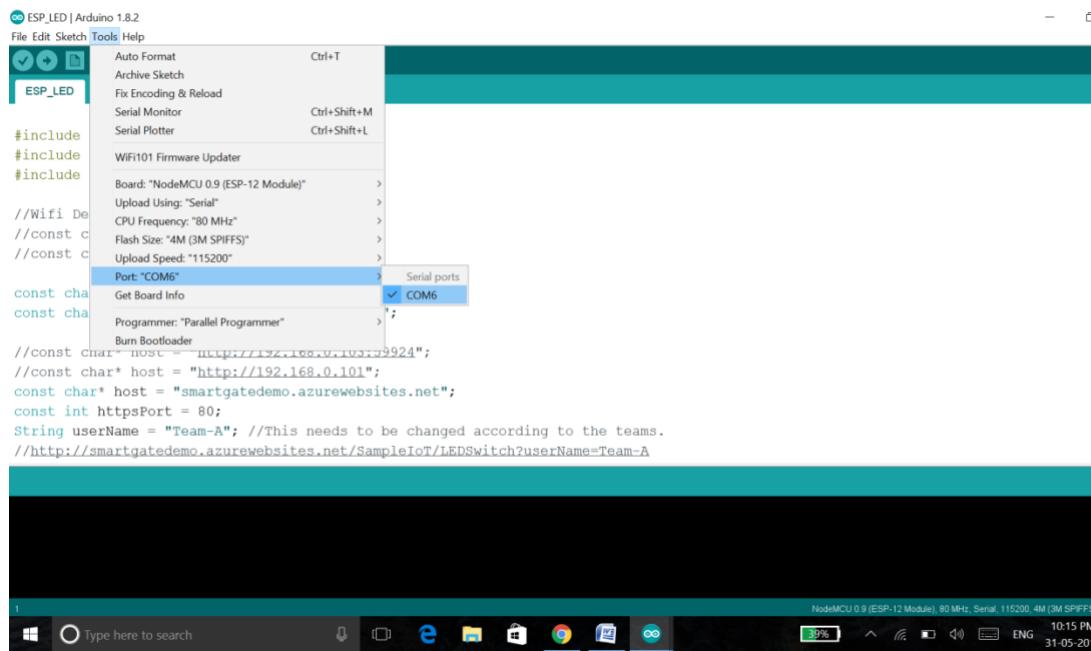


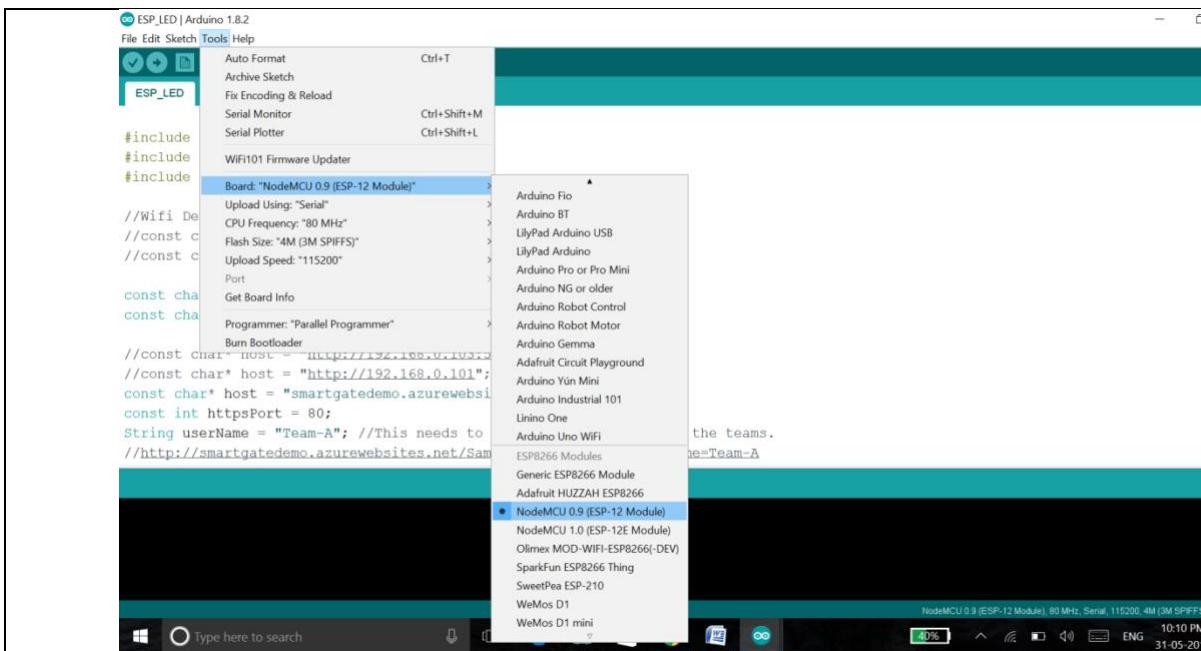
2. Connect one end of the USB cable to the NodeMCU and the other end to the computer.
3. Open the Arduino IDE → File → New. Copy the below Arduino program, change ssid & password of WiFi accordingly in the program and save the file.

```

const char* ssid = "<Wifi_Username>";
const char* password = "<wifi_password>";
```

4. Select the correct COM Port and board type NodeMCU 0.9 (ESP-12 Module) as shown below in the screen shot.





5. Compile and upload the arduino program.

Input/Output

- Open the application hosted in cloud by typing the url mentioned below in the browser. <http://revalabexercises.azurewebsites.net/IoT/LEDSwitch>
- Select the group (Team A or Team B) as specified in the program.
- Change the switch status and observe the reflection on the led.

Arduino IDE Program

```
#include <SPI.h>

#include <ESP8266WiFi.h>

//Wifi Details Starts from Here

const char* ssid = "<Wifi_Username>";

const char* password = "<wifi_password>";

const char* host = "revalabexercises.azurewebsites.net"; //Cloud server

const int httpsPort = 80;

String groupName = "Group-A"; //This needs to be changed according to the teams.

WiFiClient client;

//Wifi Details Ends Here
```

```
int reading;      // the current reading from the input pin

int previous = LOW; // the previous reading from the input pin

/* the follow variables are long's because the time, measured in miliseconds, will quickly become
a bigger number than can be stored in an int. */

longcustomtime = 0; // the last time the output pin was toggled

longdebounce = 200;

#define led 13 //GPIO13 is connected to LED (long leg), GND of ESP8266 is connected to LED
(short leg)

void setup() {

Serial.begin(115200);

pinMode(led, OUTPUT);

// Connect to wifi Starts From Here

Serial.print("connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

// Connect to wifi Ends Here

}

void loop()

{

String url = "/api/IoTServices/GetSwitchStatus?groupName=" + groupName;
```

```
String line = GetLEDStatus(url);
}

//Call API to get LED status

String GetLEDStatus(String Url)
{
    // Use WiFiClientSecure class to create TLS connection

    WiFiClient client;

    Serial.print("connecting to ");

    Serial.println(host);

    Serial.println("Port is :");

    Serial.println(httpsPort);

    if (!client.connect(host, httpsPort)) {

        Serial.println("connection failed");

    }

    Serial.println("requesting URL: ");

    Serial.println(Url);

    client.print(String("GET ") + Url + " HTTP/1.1\r\n" +

        "Host: " + host + "\r\n" +

        "User-Agent: BuildFailureDetectorESP8266\r\n" +

        "Connection: close\r\n\r\n");

    Serial.println("request sent");

    while (client.connected()) {

        String line = client.readStringUntil('\n');

        if (line == "\r") {

            Serial.println("headers received");

            break;

        }

    }

}
```

```
String line = client.readStringUntil('\n');

if (line.startsWith("{\"state\":\"success\"})) {

Serial.println("esp8266/Arduino CI successfull!");

} else {

Serial.println("esp8266/Arduino CI has failed");

}

Serial.println("Response is:");

Serial.println("----***----");

Serial.println(line);

Serial.println("----***----");

Serial.println("closing connection");

if(line == "1")

{

Serial.println("Glow Led");

digitalWrite(led,HIGH);

}

else

{

Serial.println("Switch Off");

digitalWrite(led,LOW);

}

delay(2000);

return "\r";

}
```

Cloud API

1. Use below URL to access cloud API in Arduino program to get LED status.

<http://revalabexercises.azurewebsites.net/api/IoTServices/GetSwitchStatus?groupName=<groupName>>

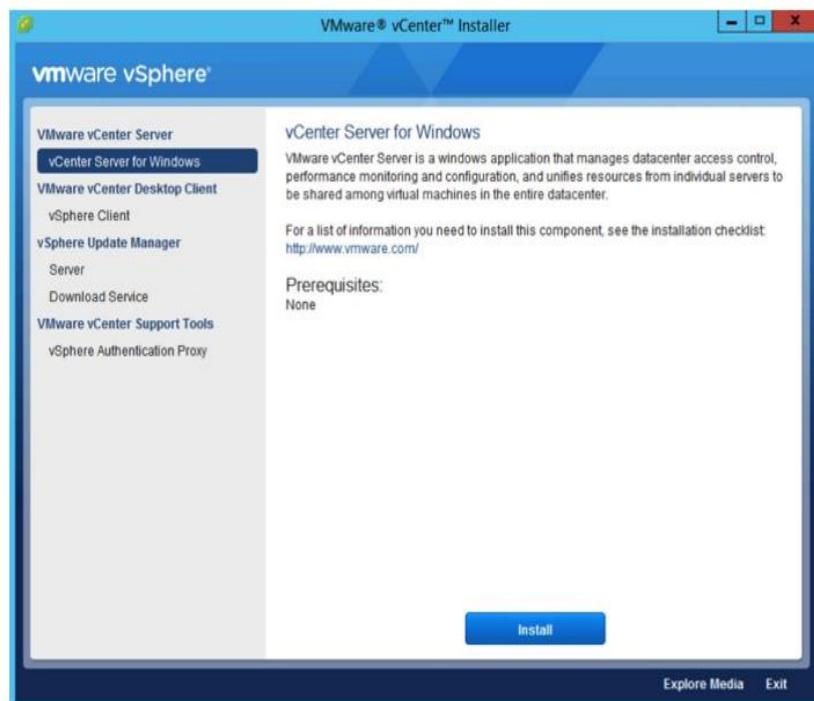
Cloud Application

1. Type the below URL to access cloud application to control the LED.
<http://revalabexercises.azurewebsites.net/IoT/LEDswitch>
2. Select the group you have mentioned in the arduino program and switch on and off the button in the display page and observe the changes.

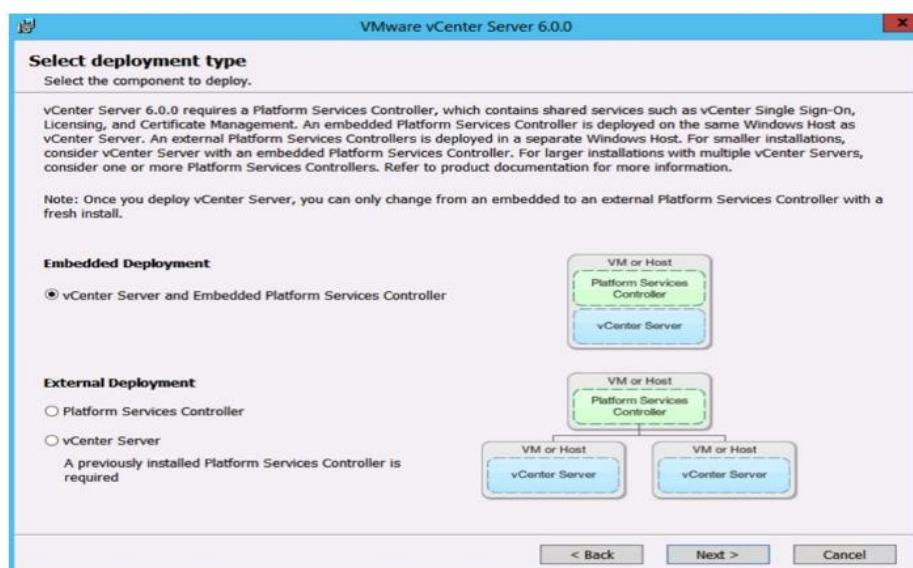
Additional Session 1	
1	Problem Statement:
Create a data center using vCenter for multiple VM's.	
2	Student Learning Outcomes:
To install and configure the vCenter server Create a final data center for different VM's	
3	Theoretical Description:
<p>VMware VCenter Server is a data center management server application developed by VMware Inc. to monitor virtualized environments.</p> <p>VCenter Server provides centralized management and operation, resource provisioning and performance evaluation of virtual machines residing on a distributed virtual data center. VMware VCentre Server is designed primarily for VSphere, VMware's platform for building virtualized cloud infrastructures.</p>	
WMware VCenter Server was previously known as VMware VirtualCenter	
4	Requirements
To install vCenter Server on a Windows virtual machine or physical server, your system must meet specific hardware and software requirements. <ul style="list-style-type: none"> ▪ Synchronize the clocks of the virtual machines on which you plan to install vCenter Server and the Platform Services Controller. ▪ Verify that the DNS name of the virtual machine or physical server matches the actual full computer name. ▪ Verify that the host name of the virtual machine or physical server that you are installing or upgrading vCenter Server on complies with RFC 1123 guidelines. ▪ Verify that the system on which you are installing vCenter Server is not an Active Directory domain controller. ▪ If your vCenter Server service is running in a user account other than the Local System account, verify that the user account in which the vCenter Server service is running has the following permissions: <ul style="list-style-type: none"> ▪ Member of the Administrators group ▪ Log on as a service ▪ Act as part of the operating system (if the user is a domain user) ▪ If the system that you use for your vCenter Server installation belongs to a workgroup rather than a domain, not all functionality is available to vCenter Server. If assigned to a workgroup, the vCenter Server system is not able to discover all domains and systems available on the network when using some features. Your host machine must be connected to a domain if you want to add Active Directory identity sources after the installation. ▪ Verify that the LOCAL SERVICE account has read permission on the folder in which vCenter Server is installed and on the HKLM registry. 	
Verify that the connection between the virtual machine or physical server and the domain controller is working.	
5	Procedure

Windows Deployment

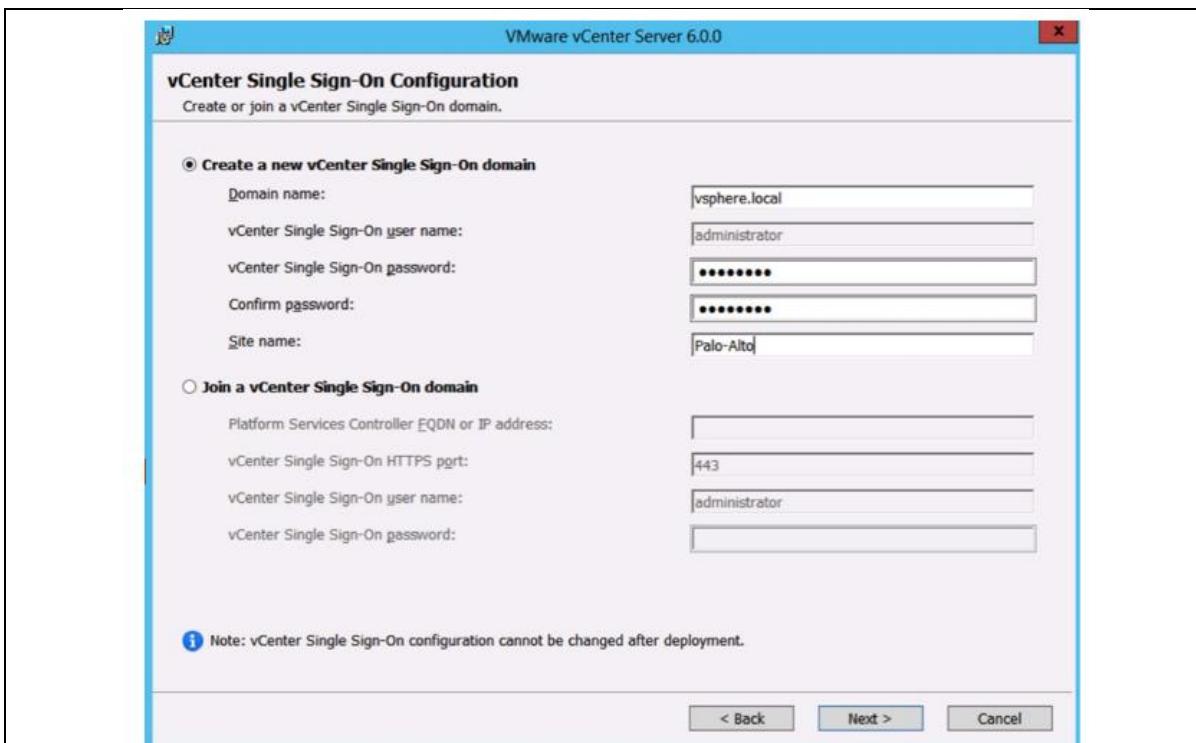
1. Verify all prerequisites.
2. If using a remote database, ensure that a 64-bit DSN has been created. DSN aliases are not supported. This step is not necessary if using the local PostgreSQL database.
3. Mount the vCenter Server 6.0 ISO image.
4. If autorun does not start, execute autorun.exe.
5. Select vCenter Server for Windows and click Install.



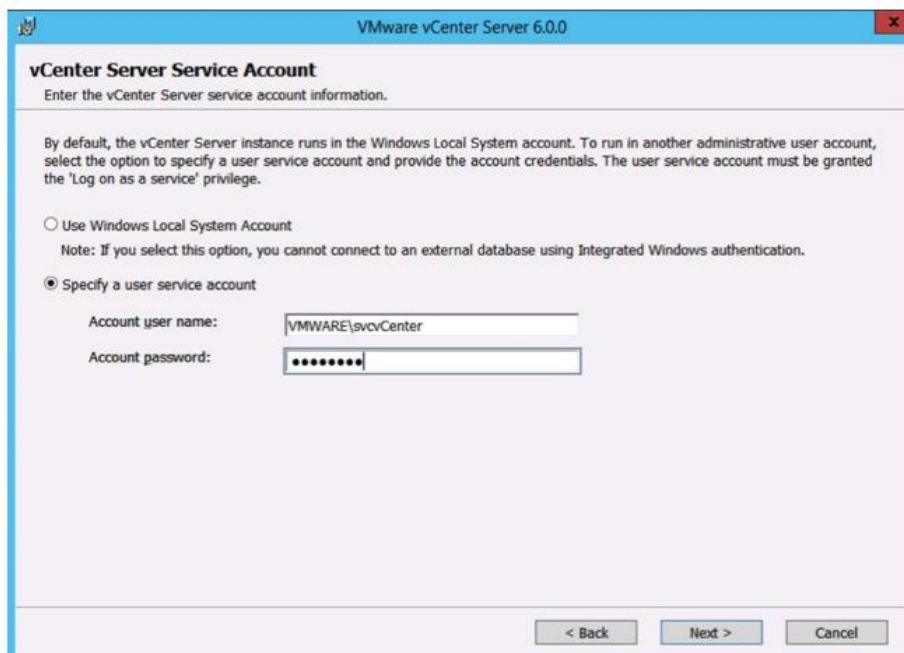
6. Click Next.
7. Accept the license agreements.
8. Select Embedded Deployment and click Next.



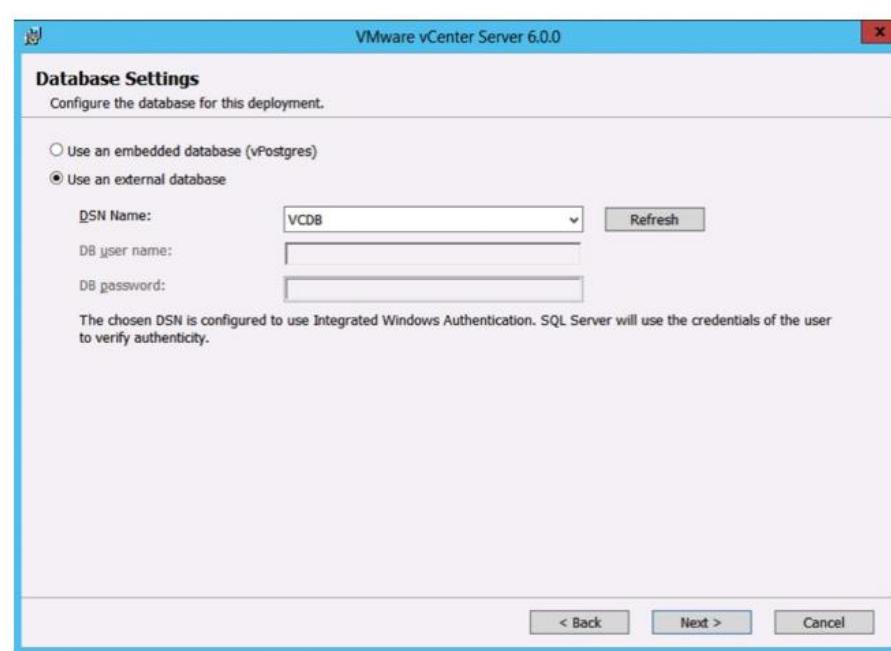
9. Verify that the FQDN is correct and click Next.
10. Enter a password and Site name for vCenter Single Sign-On and click Next.



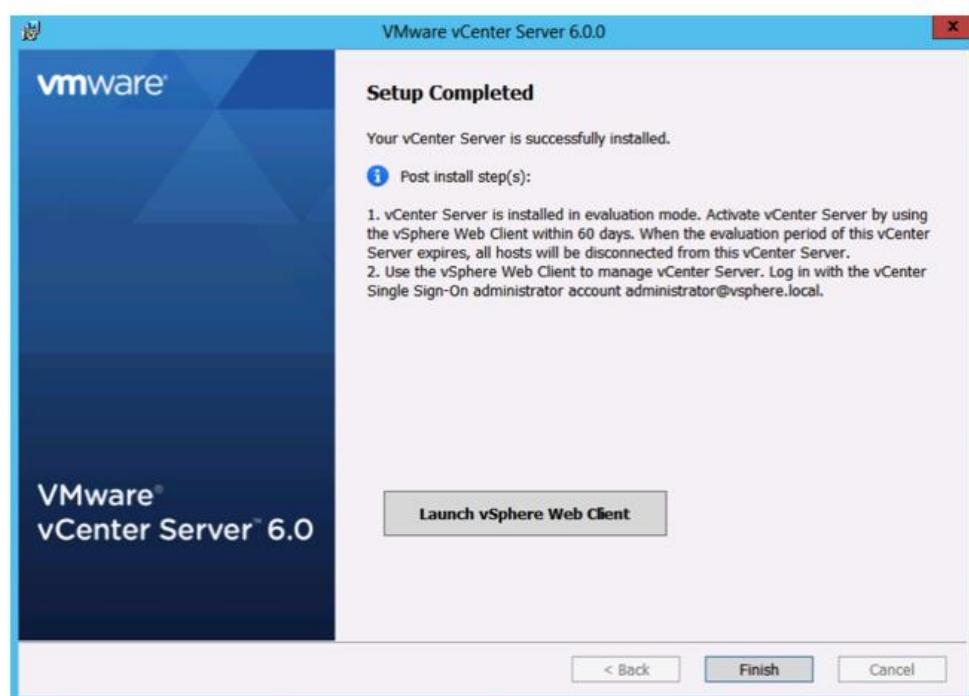
11. Select the local system account or enter the service account user name and password.



12. Select Use an embedded database (vPostgres) or Use an external database server's DSN Name and click Next.



13. Unless required, leave all ports at their defaults and click Next.
14. Unless required, leave the default paths for installation and click Next.
15. Review and then click Install.



Additional Session 2	
1	Problem Statement:
	Install and configure the vSphere Web client
2	Student Learning Outcomes:
	To install and configure the vSphere web Client to access the vCenter server Create a final data center for different VM's
3	Theoretical Description:
	vSphere Web client is a <ul style="list-style-type: none">• Web application.• Cross platform.• Can connect to only vCenter Server.• Subset of full functionality, focused on virtual machine deployment and basic monitoring functions. Cannot configure hosts, clusters, networks, datastores, or datastore clusters• Extensible plug-in-based architecture.
4	Requirements <ul style="list-style-type: none">• Download the vCenter Server installer.• Verify that you are a member of the Administrators group on the system.• Verify that the system has an Internet connection.• Verify that the system meets the software requirements for the vSphere Web Client. See vSphere Client and vSphere Web Client Software Requirements. The vSphere Web Client requires a 64-bit operating system for installation.
5	Procedure <ol style="list-style-type: none">1. In the software installer directory, double-click the autorun.exe file to start the installer.2. Select VMware vSphere® Web Client (Server) and click Install.3. Follow the wizard prompts to complete the installation When the vSphere Web Client installation is finished, a browser opens.4. Register one or more vCenter Servers on the vSphere Web Client Administration Application page in the browser.5. If the browser fails to open or to display the Administration Application page correctly, open the application from the shortcut From the Windows Start menu, select Programs > VMware > VMware vSphere Web Client > vSphere Administration Application.

Additional Session 3	
1	Problem Statement:
Illustration of vMotion to move the VM's from ESXi to ESXi	
2	Student Learning Outcomes:
<p>To install and configure the vSphere web Client to access the vCenter server</p> <p>Create a final data center for different VM's</p> <p>Identify the way of moving the VM's from one ESXi to another.</p> <p>Migrate the VM's from one to another ESXi.</p>	
3	Theoretical Description:
<p>VMware VMotion enables the live migration of running virtual machines from one physical server to another with zero downtime, continuous service availability, and complete transaction integrity. It is transparent to users.</p> <p>VMotion lets you:</p> <ul style="list-style-type: none"> ▪ Automatically optimize and allocate entire pools of resources for maximum hardware utilization and availability. ▪ Perform hardware maintenance without any scheduled downtime. ▪ Proactively migrate virtual machines away from failing or underperforming servers. 	
4	Requirements
<ul style="list-style-type: none"> ▪ Migration with vMotion requires a correctly configured vMotion network interface on both source and target hosts. ▪ Configure each host with at least one vMotion network interface. To ensure secure data transfer, the vMotion network must be a secure network, accessible only to trusted parties. Because vMotion performance improves significantly with additional bandwidth, dedicate at minimum a physical 1 Gigabit Ethernet (GigE) NIC to vMotion. As a best practice, provision at least one additional physical NIC as a failover NIC. ▪ Ensure that virtual machines have access to the same subnets on source and destination hosts. ▪ If you are using standard switches for networking, ensure that the network labels used for virtual machine port groups are consistent across hosts. ▪ During a migration with vMotion, vCenter Server assigns virtual machines to port groups. Configure hosts for vMotion with shared storage to ensure that virtual machines are accessible to both source and target hosts. ▪ If you use vMotion to migrate virtual machines with raw device mapping (RDM) files, ensure that the LUN IDs for RDMs are consistent across all participating hosts. 	
5	Procedure

Configure a vMotion interface using vSphere Client

To configure a vMotion Interface:

1. Log into the vCenter Server using vSphere Client.
2. Click to select the host.
3. Click the **Configuration** tab.
4. Click **Networking** under Hardware.
5. Click **Add Networking**.
6. Select **VMkernel** and click **Next**.
7. Select the existing vSwitch, or select **Create a vSphere standard switch** to create a new vSwitch and click **Next**.
8. Enter a name in the **Network Label** to identify the network that vMotion uses.
9. Select a VLAN ID from the **VLAN ID (Optional)** dropdown if applicable.
10. Select **Use this port group for vMotion** and click **Next**.
11. Enter the **IP address** and **Subnet Mask** of the host's vMotion Interface.
12. Click **Next**, then click **Finish**.

Note: For multiple hosts, repeat steps 2 to 12. Use a unique IP address for each host vMotion interface.

Configure a vMotion interface using vSphere Web Client:

1. In the vSphere Web Client, navigate to the Host.
2. Under **Manage**, select **Networking** and then select **VMkernel adapters**.
3. Click **Add host networking**.
4. On the Select connection type page, select **VMkernel Network Adapter** and click **Next**.
5. On the Select target device page, select either an **existing standard switch** or a **New vSphere standard switch**.
6. On the **Port properties**, enable vMotion Traffic and select **Next**.
7. Configure network for the vMotion VMkernel interface and click **Next**.
8. Review the settings and click **Finish**.

Additional Session 4

Problem Statement:

Design and Demonstrate a program using CloudSim, showing how to create a datacenter with one host and run one cloudlet on it.

Student Learning Outcomes:

Creating datacenter and a cloudlet using Cloud Sim.

Theoretical Description:

CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centres, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning.

Datacenter class is a CloudResource whose hostList are virtualized. It deals with processing of VM queries (i.e., handling of VMs) instead of processing Cloudlet-related queries.

A cloudlet is a mobility-enhanced small-scale cloud datacenter that is located at the edge of the Internet. The main purpose of the cloudlet is supporting resource-intensive and interactive mobile applications by providing powerful computing resources to mobile devices with lower latency.

Requirements:

1. Eclipse IDE

Procedure:

1. Initialize the CloudSim package and also initialize CloudSim library. (It should be called before creating any entities.)
2. Create Datacenters

(Datacenters are the resource providers in CloudSim. We need at least one of them to run a CloudSim simulation.)

3. Create Broker
4. Create one virtual machine with VM description.
5. Add the VM to the vmList.
6. Submit vmList to the broker.
7. Create one Cloudlet with cloudlet properties.
8. Add the cloudlet to the list.
9. Submit cloudlet list to the broker.
10. Start the simulation.

Steps needed to create a PowerDatacenter:

1. We need to create a list to store our machine

(A Machine contains one or more PEs or CPUs/Cores. In this example, it will have only one core.)

2. Create PEs and add these into a list
3. Create Host with its id and list of PEs and add them to the list of machines
4. Create a Datacenter Characteristics object that stores the properties of a data center: architecture, OS, list of Machines, allocation policy: time- or space-shared, time zone
and its price (G\$/Pe time unit).
5. Create a PowerDatacenter object.

(The users are encouraged to develop their own broker policies, to submit vms and cloudlets according to the specific rules of the simulated scenario.)

6. Prints the Cloudlet objects.

Source Code:

```
package org.cloudbus.cloudsim.examples;

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

import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
```

```
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerTimeShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

/**
 * Showing how to create a data center with one host and run one cloudlet on it.
 */

public class CloudSimExample1 {

    /** The cloudlet list. */
    private static List<Cloudlet> cloudletList;

    /** The vmlist. */
    private static List<Vm> vmlist;

    /**
     * Creates main() to run this example.
     *
     * @param args the args
     */
    @SuppressWarnings("unused")
    public static void main(String[] args) {
        Log.printLine("Starting CloudSimExample1...");

        try {
            // First step: Initialize the CloudSim package. It should be called before
            // creating any entities.

            int num_user = 1; // number of cloud users
        }
    }
}
```

```
Calendar calendar = Calendar.getInstance(); // Calendar whose fields have  
been initialized with the current date and time.  
  
boolean trace_flag = false; // trace events  
  
/* Comment Start - Dinesh Bhagwat  
 * Initialize the CloudSim library.  
 * init() invokes initCommonVariable() which in turn calls initialize() (all  
these 3 methods are defined in CloudSim.java).  
 * initialize() creates two collections - an ArrayList of SimEntity Objects  
(named entities which denote the simulation entities) and  
 * a LinkedHashMap (named entitiesByName which denote the  
LinkedHashMap of the same simulation entities), with name of every SimEntity as the key.  
 * initialize() creates two queues - a Queue of SimEvents (future) and  
another Queue of SimEvents (deferred).  
 * initialize() creates a HashMap of Predicates (with integers as keys) -  
these predicates are used to select a particular event from the deferred queue.  
 * initialize() sets the simulation clock to 0 and running (a boolean flag) to  
false.  
 * Once initialize() returns (note that we are in method  
initCommonVariable() now), a CloudSimShutDown (which is derived from SimEntity) instance is  
created  
 * (with numuser as 1, its name as CloudSimShutDown, id as -1, and state  
as RUNNABLE). Then this new entity is added to the simulation  
 * While being added to the simulation, its id changes to 0 (from the earlier  
-1). The two collections - entities and entitiesByName are updated with this SimEntity.  
 * the shutdownId (whose default value was -1) is 0  
 * Once initCommonVariable() returns (note that we are in method init()  
now), a CloudInformationService (which is also derived from SimEntity) instance is created  
 * (with its name as CloudInformationService, id as -1, and state as  
RUNNABLE). Then this new entity is also added to the simulation.  
 * While being added to the simulation, the id of the SimEntitiy is changed  
to 1 (which is the next id) from its earlier value of -1.  
 * The two collections - entities and entitiesByName are updated with this  
SimEntity.  
 * the cisId(whose default value is -1) is 1  
 * Comment End - Dinesh Bhagwat  
 */
```

```
CloudSim.init(num_user, calendar, trace_flag);

// Second step: Create Datacenters
// Datacenters are the resource providers in CloudSim. We need at
// list one of them to run a CloudSim simulation
Datacenter datacenter0 = createDatacenter("Datacenter_0");

// Third step: Create Broker
DatacenterBroker broker = createBroker();
int brokerId = broker.getId();

// Fourth step: Create one virtual machine
vmlist = new ArrayList<Vm>();

// VM description
int vmid = 0;
int mips = 1000;
long size = 10000; // image size (MB)
int ram = 512; // vm memory (MB)
long bw = 1000;
int pesNumber = 1; // number of cpus
String vmm = "Xen"; // VMM name

// create VM
Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm,
new CloudletSchedulerTimeShared());

// add the VM to the vmList
vmlist.add(vm);

// submit vm list to the broker
```

```
broker.submitVmList(vmlist);

// Fifth step: Create one Cloudlet
cloudletList = new ArrayList<Cloudlet>();

// Cloudlet properties
int id = 0;
long length = 400000;
long fileSize = 300;
long outputSize = 300;
UtilizationModel utilizationModel = new UtilizationModelFull();

Cloudlet cloudlet =
new Cloudlet(id, length, pesNumber, fileSize,
outputSize, utilizationModel, utilizationModel,
utilizationModel);
cloudlet.setUserId(brokerId);
cloudlet.setVmId(vmid);

// add the cloudlet to the list
cloudletList.add(cloudlet);

// submit cloudlet list to the broker
broker.submitCloudletList(cloudletList);

// Sixth step: Starts the simulation
CloudSim.startSimulation();

CloudSim.stopSimulation();
```

```
//Final step: Print results when simulation is over

List<Cloudlet> newList = broker.getCloudletReceivedList();

printCloudletList(newList);

Log.println("CloudSimExample1 finished!");

} catch (Exception e) {

    e.printStackTrace();

    Log.println("Unwanted errors happen");

}

}

/***
 * Creates the datacenter.
 *
 * @param name the name
 *
 * @return the datacenter
 */

private static Datacenter createDatacenter(String name) {

    // Here are the steps needed to create a PowerDatacenter:
    // 1. We need to create a list to store
    // our machine
    List<Host> hostList = new ArrayList<Host>();

    // 2. A Machine contains one or more PEs or CPUs/Cores.
    // In this example, it will have only one core.
    List<Pe> peList = new ArrayList<Pe>();
    int mips = 1000;
```

```
// 3. Create PEs and add these into a list.  
  
peList.add(new Pe(0, new PeProvisionerSimple(mips))); // need to store Pe id and  
MIPS Rating  
  
  
// 4. Create Host with its id and list of PEs and add them to the list  
// of machines  
  
int hostId = 0;  
  
int ram = 2048; // host memory (MB)  
  
long storage = 1000000; // host storage  
  
int bw = 10000;  
  
hostList.add(  
  
    new Host(  
  
        hostId,  
  
        new RamProvisionerSimple(ram),  
  
        new BwProvisionerSimple(bw),  
  
        storage,  
  
        peList,  
  
        new VmSchedulerTimeShared(peList)  
  
    )  
  
) // This is our machine  
  
  
// 5. Create a DatacenterCharacteristics object that stores the  
// properties of a data center: architecture, OS, list of  
// Machines, allocation policy: time- or space-shared, time zone  
// and its price (G$/Pe time unit).  
  
String arch = "x86"; // system architecture  
  
String os = "Linux"; // operating system  
  
String vmm = "Xen";  
  
double time_zone = 10.0; // time zone this resource located  
  
double cost = 3.0; // the cost of using processing in this resource  
  
double costPerMem = 0.05; // the cost of using memory in this resource
```

SAN

```
double costPerStorage = 0.001; // the cost of using storage in this
                                // resource

double costPerBw = 0.0; // the cost of using bw in this resource

LinkedList<Storage> storageList = new LinkedList<Storage>(); // we are not adding

// devices by now

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

// 6. Finally, we need to create a PowerDatacenter object.

Datacenter datacenter = null;

try {
    datacenter = new Datacenter(name, characteristics, new
VmAllocationPolicySimple(hostList), storageList, 0);

} catch (Exception e) {
    e.printStackTrace();
}

return datacenter;
}

// We strongly encourage users to develop their own broker policies, to
// submit vms and cloudlets according
// to the specific rules of the simulated scenario

/**
 * Creates the broker.
 *
 * @return the datacenter broker
 */
private static DatacenterBroker createBroker() {

    DatacenterBroker broker = null;

    try {
```

```
        broker = new DatacenterBroker("Broker");

    } catch (Exception e) {

        e.printStackTrace();

        return null;

    }

    return broker;

}

/**



 * Prints the Cloudlet objects.

 *

 *

 * @param list list of Cloudlets

*/



private static void printCloudletList(List<Cloudlet> list) {

    int size = list.size();

    Cloudlet cloudlet;





    String indent = "    ";

    Log.printLine();

    Log.printLine("===== OUTPUT =====");

    Log.printLine("Cloudlet ID" + indent + "STATUS" + indent

                + "Data center ID" + indent + "VM ID" + indent + "Time" + indent

                + "Start Time" + indent + "Finish Time");



    DecimalFormat dft = new DecimalFormat("##.##");



    for (int i = 0; i < size; i++) {

        cloudlet = list.get(i);

        Log.print(indent + cloudlet.getCloudletId() + indent + indent);





        if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS) {

            Log.print("SUCCESS");

        }

    }

}
```

```
        Log.println(indent + indent + cloudlet.getResourceId())
                + indent + indent + indent + cloudlet.getVmId()
                + indent + indent
                + dft.format(cloudlet.getActualCPUTime()) +
        indent
                + indent + dft.format(cloudlet.getExecStartTime())
                + indent + indent
                + dft.format(cloudlet.getFinishTime()));

    }
}

}
```

OUTPUT:

```
gk@gk:~$ java org.cloudbus.cloudsim.examples.CloudSimExample1
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
<400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID      STATUS      Data center ID      VM ID      Time      Start Time      Finish Time
          0      SUCCESS           2              0       400        0.1        400.1
CloudSimExample1 finished!
gk@gk:~$
```

Additional Session 5

Problem Statement:

Design and Demonstrate using CloudSim, showing how to create two datacenters with one host each and run cloudlets of two users on them.

Student Learning Outcomes:

Create two datacenters with one host each and run cloudlets of two users on them using CloudSim.

Theoretical Description:

CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centres, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning.

Datacenter class is a CloudResource whose hostList are virtualized. It deals with processing of VM queries (i.e., handling of VMs) instead of processing Cloudlet-related queries.

A cloudlet is a mobility-enhanced small-scale cloud datacenter that is located at the edge of the Internet. The main purpose of the cloudlet is supporting resource-intensive and interactive mobile applications by providing powerful computing resources to mobile devices with lower latency.

Requirements:

1. Eclipse IDE

Procedure:

1. Initialize the CloudSim package and also initialize CloudSim library. (It should be called before creating any entities.)
2. Create Datacenters

(Datacenters are the resource providers in CloudSim. We need at least one of them to run a CloudSim simulation.)

3. Create Broker
4. Create one virtual machine for each broker/user
(the first VM belongs to user1 and the second VM belongs to user 2)
5. Add the VMs to the vmLists
6. Submit vmList to the broker
7. Create two Cloudlets
8. Add the cloudlets to the lists: each cloudlet belongs to one user
9. Submit cloudlet list to the brokers
10. Starts the simulation

Steps needed to create a PowerDatacenter:

1. We need to create a list to store our machine

(A Machine contains one or more PEs or CPUs/Cores. In this example, it will have only one core.)

2. Create PEs and add these into a list
3. Create Host with its id and list of PEs and add them to the list of machines
4. Create a Datacenter Characteristics object that stores the properties of a data center: architecture, OS, list of Machines, allocation policy: time- or space-shared, time zone
and its price (G\$/Pe time unit).
5. Create a PowerDatacenter object.

(The users are encouraged to develop their own broker policies, to submit vms and cloudlets according to the specific rules of the simulated scenario.)

Prints the Cloudlet objects.

Source Code:

```
package org.cloudbus.cloudsim.examples;

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

import org.cloudbus.cloudsim.Cloudlet;
import org.cloudbus.cloudsim.CloudletSchedulerTimeShared;
import org.cloudbus.cloudsim.Datacenter;
import org.cloudbus.cloudsim.DatacenterBroker;
import org.cloudbus.cloudsim.DatacenterCharacteristics;
import org.cloudbus.cloudsim.Host;
import org.cloudbus.cloudsim.Log;
import org.cloudbus.cloudsim.Pe;
import org.cloudbus.cloudsim.Storage;
import org.cloudbus.cloudsim.UtilizationModel;
import org.cloudbus.cloudsim.UtilizationModelFull;
```

```
import org.cloudbus.cloudsim.Vm;
import org.cloudbus.cloudsim.VmAllocationPolicySimple;
import org.cloudbus.cloudsim.VmSchedulerSpaceShared;
import org.cloudbus.cloudsim.core.CloudSim;
import org.cloudbus.cloudsim.provisioners.BwProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.PeProvisionerSimple;
import org.cloudbus.cloudsim.provisioners.RamProvisionerSimple;

/**
 * A simple example showing how to create
 * two datacenters with one host each and
 * run cloudlets of two users on them.
 */
public class CloudSimExample5 {

    /** The cloudlet lists. */
    private static List<Cloudlet> cloudletList1;
    private static List<Cloudlet> cloudletList2;

    /** The vmlists. */
    private static List<Vm> vmlist1;
    private static List<Vm> vmlist2;

    /**
     * Creates main() to run this example
     */
    public static void main(String[] args) {
        Log.printLine("Starting CloudSimExample5...");
```

```
try {  
    // First step: Initialize the CloudSim package. It should be called  
    // before creating any entities.  
    int num_user = 2; // number of cloud users  
    Calendar calendar = Calendar.getInstance();  
    boolean trace_flag = false; // mean trace events  
  
    // Initialize the CloudSim library  
    CloudSim.init(num_user, calendar, trace_flag);  
  
    // Second step: Create Datacenters  
    //Datacenters are the resource providers in CloudSim. We need at least one  
    // of them to run a CloudSim simulation  
    @SuppressWarnings("unused")  
    Datacenter datacenter0 = createDatacenter("Datacenter_0");  
    @SuppressWarnings("unused")  
    Datacenter datacenter1 = createDatacenter("Datacenter_1");  
  
    //Third step: Create Brokers  
    DatacenterBroker broker1 = createBroker(1);  
    int brokerId1 = broker1.getId();  
  
    DatacenterBroker broker2 = createBroker(2);  
    int brokerId2 = broker2.getId();  
  
    //Fourth step: Create one virtual machine for each broker/user  
    vmlist1 = new ArrayList<Vm>();  
    vmlist2 = new ArrayList<Vm>();  
  
    //VM description  
    int vmid = 0;
```

```
int mips = 250;  
  
long size = 10000; //image size (MB)  
  
int ram = 512; //vm memory (MB)  
  
long bw = 1000;  
  
int pesNumber = 1; //number of cpus  
  
String vmm = "Xen"; //VMM name  
  
//create two VMs: the first one belongs to user1  
  
Vm vm1 = new Vm(vmid, brokerId1, mips, pesNumber, ram, bw, size,  
vmm, new CloudletSchedulerTimeShared());  
  
//the second VM: this one belongs to user2  
  
Vm vm2 = new Vm(vmid, brokerId2, mips, pesNumber, ram, bw, size,  
vmm, new CloudletSchedulerTimeShared());  
  
//add the VMs to the vmlists  
  
vmlist1.add(vm1);  
  
vmlist2.add(vm2);  
  
//submit vm list to the broker  
  
broker1.submitVmList(vmlist1);  
  
broker2.submitVmList(vmlist2);  
  
//Fifth step: Create two Cloudlets  
  
cloudletList1 = new ArrayList<Cloudlet>();  
  
cloudletList2 = new ArrayList<Cloudlet>();  
  
//Cloudlet properties  
  
int id = 0;  
  
long length = 40000;  
  
long fileSize = 300;
```

```
long outputSize = 300;

UtilizationModel utilizationModel = new UtilizationModelFull();

Cloudlet cloudlet1 = new Cloudlet(id, length, pesNumber, fileSize,
outputSize, utilizationModel, utilizationModel, utilizationModel);

cloudlet1.setUserId(brokerId1);

Cloudlet cloudlet2 = new Cloudlet(id, length, pesNumber, fileSize,
outputSize, utilizationModel, utilizationModel, utilizationModel);

cloudlet2.setUserId(brokerId2);

//add the cloudlets to the lists: each cloudlet belongs to one user

cloudletList1.add(cloudlet1);

cloudletList2.add(cloudlet2);

//submit cloudlet list to the brokers

broker1.submitCloudletList(cloudletList1);

broker2.submitCloudletList(cloudletList2);

// Sixth step: Starts the simulation

CloudSim.startSimulation();

// Final step: Print results when simulation is over

List<Cloudlet> newList1 = broker1.getCloudletReceivedList();

List<Cloudlet> newList2 = broker2.getCloudletReceivedList();

CloudSim.stopSimulation();

Log.print("===== User "+brokerId1+" =====");

printCloudletList(newList1);
```

```
        Log.print("===== User "+brokerId2+" ");
        printCloudletList(newList2);
        Log.printLine("CloudSimExample5 finished!");
    }

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

private static Datacenter createDatacenter(String name){

    // Here are the steps needed to create a PowerDatacenter:
    // 1. We need to create a list to store
    // our machine
    List<Host> hostList = new ArrayList<Host>();

    // 2. A Machine contains one or more PEs or CPUs/Cores.
    // In this example, it will have only one core.
    List<Pe> peList = new ArrayList<Pe>();
    int mips=1000;
    // 3. Create PEs and add these into a list.
    peList.add(new Pe(0, new PeProvisionerSimple(mips))); // need to store Pe id and
MIPS Rating

    //4. Create Host with its id and list of PEs and add them to the list of machines
    int hostId=0;
    int ram = 2048; //host memory (MB)
    long storage = 1000000; //host storage
    int bw = 10000;
    //in this example, the VMAllocatonPolicy in use is SpaceShared. It means that only
one VM
```

```
//is allowed to run on each Pe. As each Host has only one Pe, only one VM can run  
on each Host.
```

```
hostList.add(  
    new Host(  
        hostId,  
        new RamProvisionerSimple(ram),  
        new BwProvisionerSimple(bw),  
        storage,  
        peList,  
        new VmSchedulerSpaceShared(peList)  
    )  
) // This is our first machine
```

```
// 5. Create a DatacenterCharacteristics object that stores the  
// properties of a data center: architecture, OS, list of  
// Machines, allocation policy: time- or space-shared, time zone  
// and its price (G$/Pe time unit).
```

```
String arch = "x86"; // system architecture  
String os = "Linux"; // operating system  
String vmm = "Xen";  
double time_zone = 10.0; // time zone this resource located  
double cost = 3.0; // the cost of using processing in this resource  
double costPerMem = 0.05; // the cost of using memory in this  
resource  
double costPerStorage = 0.001; // the cost of using storage in this resource  
double costPerBw = 0.0; // the cost of using bw in this  
resource  
LinkedList<Storage> storageList = new LinkedList<Storage>(); //we are not  
adding SAN devices by now  
DatacenterCharacteristics characteristics = new DatacenterCharacteristics(  
    arch, os, vmm, hostList, time_zone, cost, costPerMem, costPerStorage, costPerBw);
```

```
// 6. Finally, we need to create a PowerDatacenter object.

Datacenter datacenter = null;

try {

    datacenter = new Datacenter(name, characteristics, new
VmAllocationPolicySimple(hostList), storageList, 0);

} catch (Exception e) {

    e.printStackTrace();

}

return datacenter;

}

//We strongly encourage users to develop their own broker policies, to submit vms and
cloudlets according

//to the specific rules of the simulated scenario

private static DatacenterBroker createBroker(int id){

    DatacenterBroker broker = null;

    try {

        broker = new DatacenterBroker("Broker"+id);

    } catch (Exception e) {

        e.printStackTrace();

        return null;

    }

    return broker;

}

/***
 * Prints the Cloudlet objects
 *
 * @param list list of Cloudlets
 */

private static void printCloudletList(List<Cloudlet> list) {

    int size = list.size();
```

```
Cloudlet cloudlet;

String indent = "    ";
Log.printLine();
Log.printLine("===== OUTPUT =====");
Log.printLine("Cloudlet ID" + indent + "STATUS" + indent +
            "Data center ID" + indent + "VM ID" + indent + "Time" + indent +
            "Start Time" + indent + "Finish Time");

DecimalFormat dft = new DecimalFormat("###.##");

for (int i = 0; i < size; i++) {
    cloudlet = list.get(i);
    Log.print(indent + cloudlet.getCloudletId() + indent + indent);

    if (cloudlet.getCloudletStatus() == Cloudlet.SUCCESS){
        Log.print("SUCCESS");

        Log.print( indent + indent + cloudlet.getResourceId() + indent +
                  indent + indent + cloudlet.getVmId() +
                  indent + indent +
                  dft.format(cloudlet.getActualCPUTime()) + indent + indent +
                  dft.format(cloudlet.getExecStartTime()) +
                  indent + indent +
                  dft.format(cloudlet.getFinishTime()));

    }
}

}
```

```
gk@gk:~$ java org.cloudbus.cloudsim.examples.CloudSimExample1
Starting CloudSimExample1...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 1 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
400.1: Broker: All Cloudlets executed. Finishing...
400.1: Broker: Destroying VM #0
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

===== OUTPUT =====
Cloudlet ID      STATUS      Data center ID    VM ID      Time      Start Time      Finish Time
          0      SUCCESS           2            0       400        0.1        400.1
CloudSimExample1 finished!
gk@gk:~$
```

Appendix A

Problem Statement:

Install and Configure hadoop.

Student Learning Outcomes:

To identify the different ways to install and setup the hadoop.

Theoretical Description:

Hadoop: Hadoop is a framework written in Java for running applications on large clusters of commodity hardware.

Requirements

- Ubuntu Linux 10.04 LTS (deprecated: 8.10 LTS, 8.04, 7.10, 7.04)
- Hadoop 1.0.3, released May 2012
- Hadoop requires a working Java 1.5+ (aka Java 5) installation. However, using Java 1.6 (aka Java 6) is recommended for running Hadoop.

Procedure

1. Install Java as follows:

```
# Add the Ferramosca Roberto's repository to your apt
repositories
# See https://launchpad.net/~ferramroberto/
#
$ sudo apt-get install python-software-properties
$ sudo add-apt-repository ppa:ferramroberto/java

# Update the source list
$ sudo apt-get update

# Install Sun Java 6 JDK
$ sudo apt-get install sun-java6-jdk

# Select Sun's Java as the default on your machine.
# See 'sudo update-alternatives --config java' for more
information.
#
$ sudo update-java-alternatives -s java-6-sun
```

The full JDK which will be placed in /usr/lib/jvm/java-6-sun (well, this directory is actually a symlink on Ubuntu).

After installation, make a quick check whether Sun's JDK is correctly set up:

```
user@ubuntu:~# java -version
java version "1.6.0_20"
Java(TM) SE Runtime Environment (build 1.6.0_20-b02)
Java HotSpot(TM) Client VM (build 16.3-b01, mixed mode, sharing)
```

2. Adding a dedicated Hadoop system user.

```
$ sudo addgroup hadoop  
$ sudo adduser --ingroup hadoop hduser
```

3. Configuring SSH.

First, we have to generate an SSH key for the hduser user.

```
1 user@ubuntu:~$ su - hduser  
2 hduser@ubuntu:~$ ssh-keygen -t rsa -P ""  
3 Generating public/private rsa key pair.  
4 Enter file in which to save the key  
(/home/hduser/.ssh/id_rsa):  
5 Created directory '/home/hduser/.ssh'.  
6 Your identification has been saved in  
/home/hduser/.ssh/id_rsa.  
7 Your public key has been saved in  
/home/hduser/.ssh/id_rsa.pub.  
8 The key fingerprint is:  
9 9b:82:ea:58:b4:e0:35:d7:ff:19:66:a6:ef:ae:0e:d2  
hduser@ubuntu  
10 The key's randomart image is:  
11 [...snipp...]  
12 hduser@ubuntu:~$
```

The second line will create an RSA key pair with an empty password.

Second, you have to enable SSH access to your local machine with this newly created key.

```
hduser@ubuntu:~$ cat $HOME/.ssh/id_rsa.pub >>  
$HOME/.ssh/authorized_keys
```

The final step is to test the SSH setup by connecting to your local machine with the hduser user.

```
1 hduser@ubuntu:~$ ssh localhost  
2 The authenticity of host 'localhost (::1)' can't be  
established.
```

```
3 RSA key fingerprint is  
d7:87:25:47:ae:02:00:eb:1d:75:4f:bb:44:f9:36:26.  
  
4 Are you sure you want to continue connecting (yes/no)? yes  
  
5 Warning: Permanently added 'localhost' (RSA) to the list of  
known hosts.  
  
6 Linux ubuntu 2.6.32-22-generic #33-Ubuntu SMP Wed Apr 28  
13:27:30 UTC 2010 i686 GNU/Linux  
  
7 Ubuntu 10.04 LTS.  
  
8 [...snipp...]  
  
9 hduser@ubuntu:~$
```

If the SSH connect should fail, these general tips might help:

- Enable debugging with `ssh -vvv localhost` and investigate the error in detail.
- Check the SSH server configuration in `/etc/ssh/sshd_config`, in particular the options `PubkeyAuthentication` (which should be set to `yes`) and `AllowUsers` (if this option is active, add the `hduser` user to it). If you made any changes to the SSH server configuration file, you can force a configuration reload with `sudo /etc/init.d/ssh reload`.

Disabling IPv6:

One problem with IPv6 on Ubuntu is that using 0.0.0.0 for the various networking-related Hadoop configuration options will result in Hadoop binding to the IPv6 addresses of my Ubuntu box.

To disable IPv6 on Ubuntu 10.04 LTS, open `/etc/sysctl.conf` in the editor of your choice and add the following lines to the end of the file:

```
/etc/sysctl.conf  
1 # disable ipv6  
2 net.ipv6.conf.all.disable_ipv6 = 1  
3 net.ipv6.conf.default.disable_ipv6 = 1  
4 net.ipv6.conf.lo.disable_ipv6 = 1
```

You have to reboot your machine in order to make the changes take effect.

You can check whether IPv6 is enabled on your machine with the following command:

```
1 $ cat /proc/sys/net/ipv6/conf/all/disable_ipv6
```

A return value of 0 means IPv6 is enabled, a value of 1 means disabled (that's what we want).

4. Install Hadoop.

Download Hadoop from the Apache Download Mirrors and extract the contents of the Hadoop package to a location of your choice. I picked `/usr/local/hadoop`. Make sure to change the owner of all the files to the `hduser` user and `hadoop` group, for example:

```
1 $ cd /usr/local
2 $ sudo tar xzf hadoop-1.0.3.tar.gz
3 $ sudo mv hadoop-1.0.3 hadoop
4 $ sudo chown -R hduser:hadoop hadoop
```

Update `$HOME/.bashrc`

Add the following lines to the end of the `$HOME/.bashrc` file of user `hduser`. If you use a shell other than bash, you should of course update its appropriate configuration files instead of `.bashrc`.

`$HOME/.bashrc`

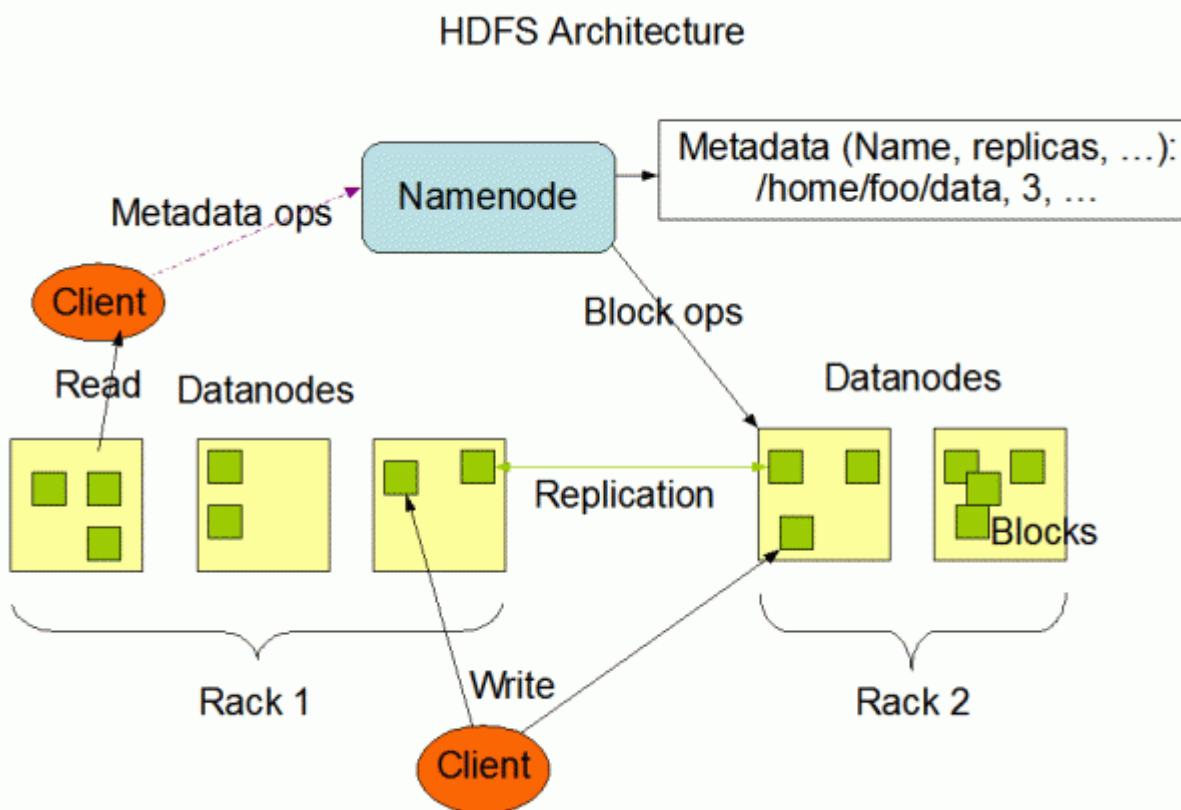
```
1 # Set Hadoop-related environment variables
2 export HADOOP_HOME=/usr/local/hadoop
3
4 # Set JAVA_HOME (we will also configure JAVA_HOME directly
# for Hadoop later on)
5 export JAVA_HOME=/usr/lib/jvm/java-6-sun
6
7 # Some convenient aliases and functions for running
# Hadoop-related commands
8 unalias fs &> /dev/null
9 alias fs="hadoop fs"
10 unalias hls &> /dev/null
11 alias hls="fs -ls"
12
13 # If you have LZO compression enabled in your Hadoop
# cluster and
14 # compress job outputs with LZOP (not covered in this
# tutorial):
15 # Conveniently inspect an LZOP compressed file from the
# command
16 # line; run via:
17 #
18 # $ lzohead /hdfs/path/to/lzop/compressed/file.lzo
19 #
20 # Requires installed 'lzop' command.
21 #
22 lzohead () {
23     hadoop fs -cat $1 | lzop -dc | head -1000 | less
24 }
```

```

25
26 # Add Hadoop bin/ directory to PATH
27 export PATH=$PATH:$HADOOP_HOME/bin

```

Hadoop Distributed File System (HDFC):



Configuration:

`hadoop-env.sh`

The only required environment variable we have to configure for Hadoop in this tutorial is `JAVA_HOME`.

Change

`conf/hadoop-env.sh`

```

1 # The java implementation to use. Required.
2 # export JAVA_HOME=/usr/lib/j2sdk1.5-sun

```

to

`conf/hadoop-env.sh`

```

1 # The java implementation to use. Required.
2 export JAVA_HOME=/usr/lib/jvm/java-6-sun

```

conf/*-site.xml

In this section, we will configure the directory where Hadoop will store its data files, the network ports it listens to, etc.

Now we create the directory and set the required ownerships and permissions:

```
1 $ sudo mkdir -p /app/hadoop/tmp  
2 $ sudo chown hduser:hadoop /app/hadoop/tmp  
3 # ...and if you want to tighten up security, chmod from  
4 # 755 to 750...  
4 $ sudo chmod 750 /app/hadoop/tmp
```

Add the following snippets between the `<configuration> ... </configuration>` tags in the respective configuration XML file.

In file `conf/core-site.xml`:

conf/core-site.xml

```
1 <property>  
2   <name>hadoop.tmp.dir</name>  
3   <value>/app/hadoop/tmp</value>  
4   <description>A base for other temporary directories.</description>  
5 </property>  
6  
7 <property>  
8   <name>fs.default.name</name>  
9   <value>hdfs://localhost:54310</value>  
10  <description>The name of the default file system. A URI whose  
11    scheme and authority determine the FileSystem implementation. The  
12    uri's scheme determines the config property (fs.SCHEME.impl) naming  
13    the FileSystem implementation class. The uri's authority is used to  
14    determine the host, port, etc. for a filesystem.</description>  
15 </property>
```

In file `conf/mapred-site.xml`:

conf/mapred-site.xml

```
1 <property>  
2   <name>mapred.job.tracker</name>  
3   <value>localhost:54311</value>  
4   <description>The host and port that the MapReduce job tracker runs  
5    at. If "local", then jobs are run in-process as a single map  
6    and reduce task.  
7   </description>  
8 </property>
```

In file `conf/hdfs-site.xml`:

`conf/hdfs-site.xml`

```
1 <property>
2   <name>dfs.replication</name>
3   <value>1</value>
4   <description>Default block replication.
5   The actual number of replications can be specified when the file is created.
6   The default is used if replication is not specified in create time.
7   </description>
8 </property>
```

Formatting the HDFS filesystem via the NameNode:

The first step to starting up your Hadoop installation is formatting the Hadoop filesystem which is implemented on top of the local filesystem of your “cluster”. You need to do this the first time you set up a Hadoop cluster.

To format the filesystem (which simply initializes the directory specified by the `dfs.name.dir` variable), run the command:

```
hduser@ubuntu:~$ /usr/local/hadoop/bin/hadoop namenode -format
```

The output will look like this:

```
hduser@ubuntu:/usr/local/hadoop$ bin/hadoop namenode -format
1 10/05/08 16:59:56 INFO namenode.NameNode: STARTUP_MSG:
2 ****
3 STARTUP_MSG: Starting NameNode
4 STARTUP_MSG:   host = ubuntu/127.0.1.1
5 STARTUP_MSG:   args = [-format]
6 STARTUP_MSG:   version = 0.20.2
7 STARTUP_MSG:   build =
8 https://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20 -r
9 911707; compiled by 'chrisdo' on Fri Feb 19 08:07:34 UTC 2010
10 ****
11 10/05/08 16:59:56 INFO namenode.FSNamesystem: fsOwner=hduser,hadoop
12 10/05/08 16:59:56 INFO namenode.FSNamesystem: supergroup=supergroup
13 10/05/08 16:59:56 INFO namenode.FSNamesystem: isPermissionEnabled=true
14 10/05/08 16:59:56 INFO common.Storage: Image file of size 96 saved in 0
seconds.
15 10/05/08 16:59:57 INFO common.Storage: Storage directory .../hadoop-
hduser/dfs/name has been successfully formatted.
16 10/05/08 16:59:57 INFO namenode.NameNode: SHUTDOWN_MSG:
17 ****
18 SHUTDOWN_MSG: Shutting down NameNode at ubuntu/127.0.1.1
19 ****
hduser@ubuntu:/usr/local/hadoop$
```

Starting your single-node cluster

Run the command:

```
1 hduser@ubuntu:~$ /usr/local/hadoop/bin/start-all.sh
```

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on your machine.

The output will look like this:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/start-all.sh
2 starting namenode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-namenode-
ubuntu.out
3 localhost: starting datanode, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-
datanode-ubuntu.out
4 localhost: starting secondarynamenode, logging to /usr/local/hadoop/bin/../logs/hadoop-
hduser-secondarynamenode-ubuntu.out
5 starting jobtracker, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-jobtracker-
ubuntu.out
6 localhost: starting tasktracker, logging to /usr/local/hadoop/bin/../logs/hadoop-hduser-
tasktracker-ubuntu.out
7 hduser@ubuntu:/usr/local/hadoop$
```

Use the tool [jps](#) for checking whether the expected Hadoop processes are running.

```
1 hduser@ubuntu:/usr/local/hadoop$ jps
2 2287 TaskTracker
3 2149 JobTracker
4 1938 DataNode
5 2085 SecondaryNameNode
6 2349 Jps
7 1788 NameNode
```

Stopping your single-node cluster

Run the command

```
hduser@ubuntu:~$ /usr/local/hadoop/bin/stop-all.sh
```

to stop all the daemons running on your machine.

Example output:

```
1 hduser@ubuntu:/usr/local/hadoop$ bin/stop-all.sh
2 stopping jobtracker
3 localhost: stopping tasktracker
4 stopping namenode
5 localhost: stopping datanode
6 localhost: stopping secondarynamenode
7 hduser@ubuntu:/usr/local/hadoop$
```

Appendix B	
1	Problem Statement:
Install and Configure VMware vSphere ESXi.	
2	Student Learning Outcomes:
To identify the different ways to install and setup the VMware ESXi, which can act as server for creating multiple VM's.	
3	Theoretical Description:
<p>vSphere: formerly VMware Infrastructure, is VMware's cloud computing virtualization platform.</p> <p>ESXi: formerly ESX, is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers. As a type-1 hypervisor, ESXi is not a software application that one installs in an operating system (OS); instead, it includes and integrates vital OS components, such as a kernel.</p>	
4	Requirements
To install or upgrade ESXi 6.0, your hardware and system resources must meet the following requirements:	
<ul style="list-style-type: none"> • Supported server platform. For a list of supported platforms, see the <i>VMware Compatibility Guide</i> at http://www.vmware.com/resources/compatibility. • ESXi 6.0 requires a host machine with at least two CPU cores. • ESXi 6.0 supports 64-bit x86 processors released after September 2006. This includes a broad range of multi-core processors. • ESXi 6.0 requires the NX/XD bit to be enabled for the CPU in the BIOS. • ESXi requires a minimum of 4GB of physical RAM. It is recommended to provide at least 8 GB of RAM to run virtual machines in typical production environments. • To support 64-bit virtual machines, support for hardware virtualization (Intel VT-x or AMD RVI) must be enabled on x64 CPUs. • One or more Gigabit or faster Ethernet controllers. For a list of supported network adapter models. • SCSI disk or a local, non-network, RAID LUN with unpartitioned space for the virtual machines. • For Serial ATA (SATA), a disk connected through supported SAS controllers or supported on-board SATA controllers. SATA disks will be considered remote, not local. These disks will not be used as a scratch partition by default because they are seen as remote. 	
5	Procedure

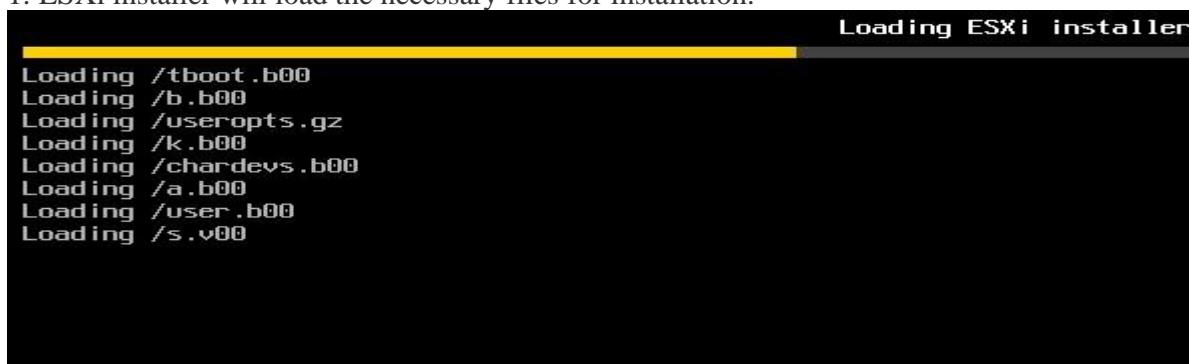
I. Installation of ESXi 6.0.0 on Bare metal

Insert the VMWARE ESXi6.0.0DVD and Power on the Computer, the computer will BOOT from the CDROM, and the following BOOT screen will be displayed

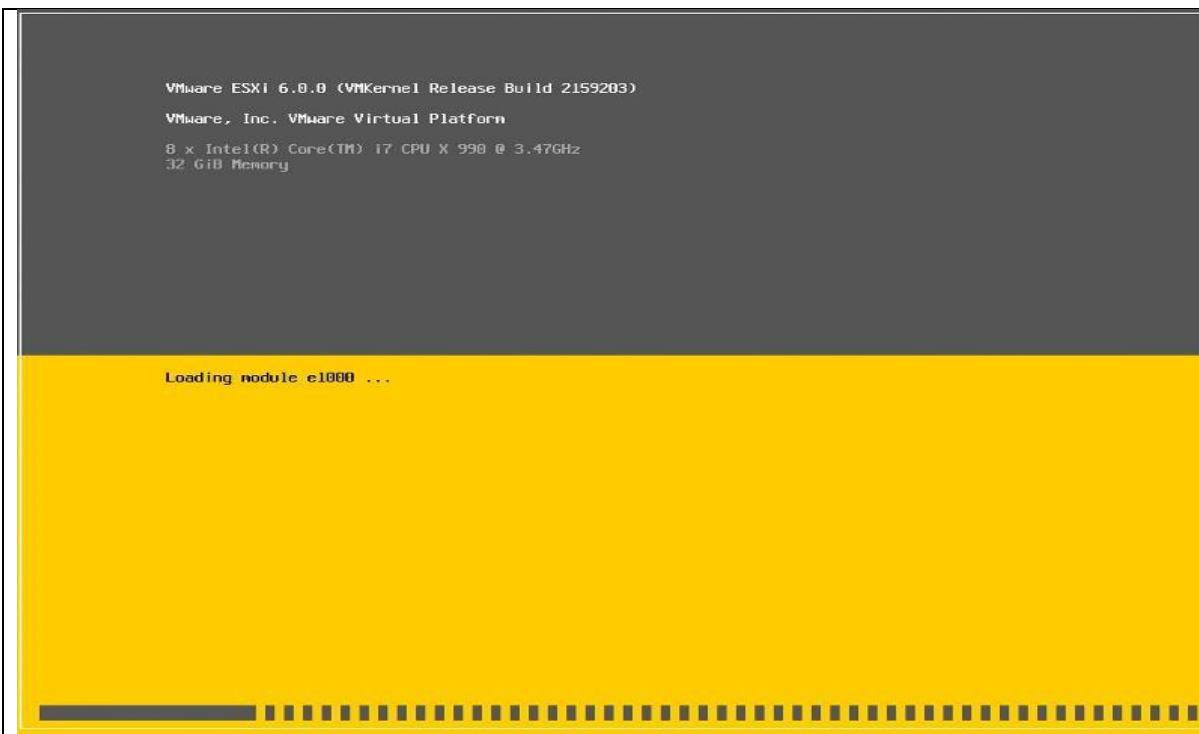


The default option is to BOOT ESXi-6.0.0-2159203-standard Installer after 10 seconds, hit any key will pause the BOOT process.

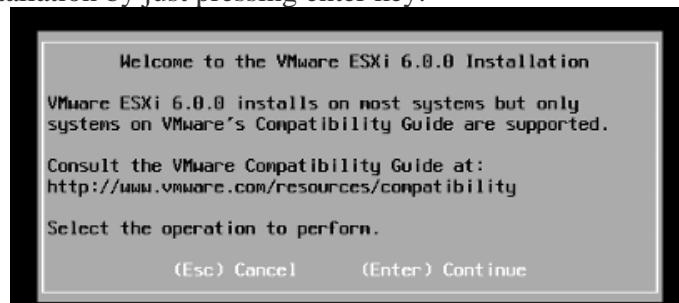
1. ESXi installer will load the necessary files for installation.



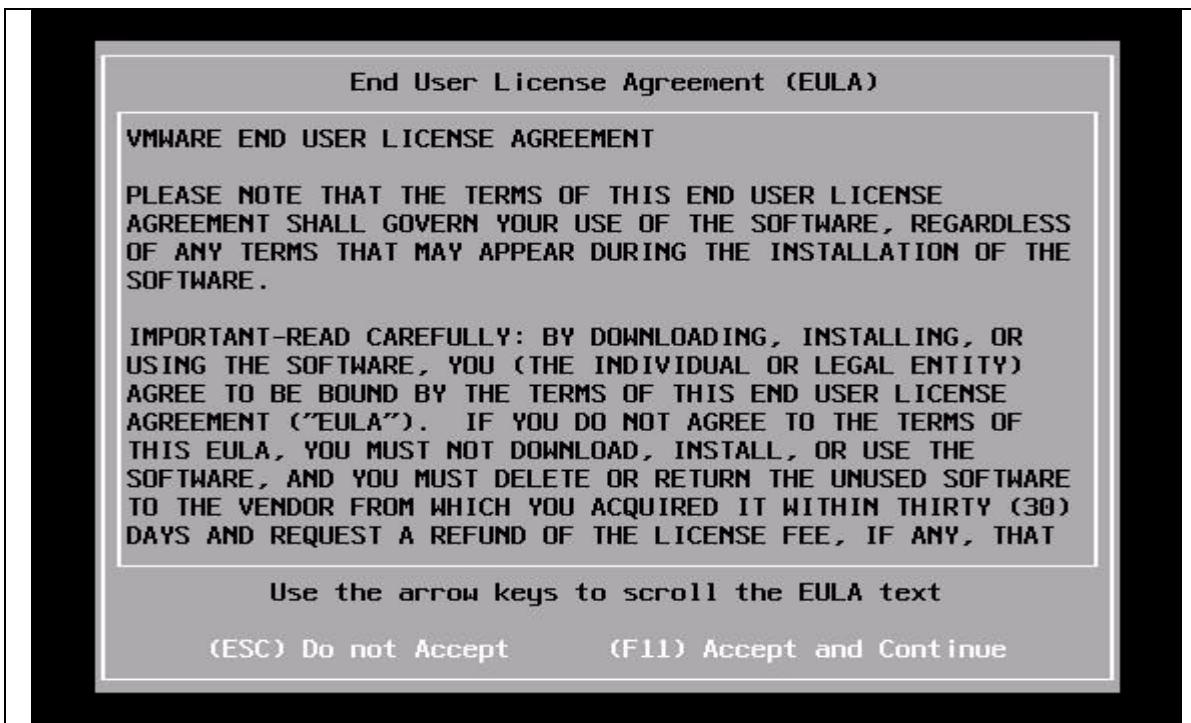
2. Once all the modules are loaded successfully, you will get the below screen with server configuration.



3. Continue the installation by just pressing enter key.



4. By default, you have to accept the license to continue the installation by pressing F11.



5. Select the disk in which you would like to install VMWARE ESXi 5.1. It requires minimum 4~ to 5~GB disk space. Enter to continue.



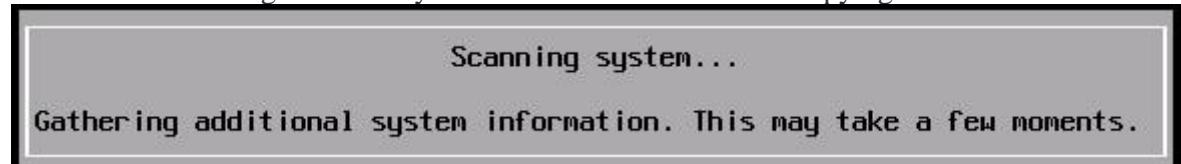
6. Select the keyboard layout. Here i am selecting "US Default".



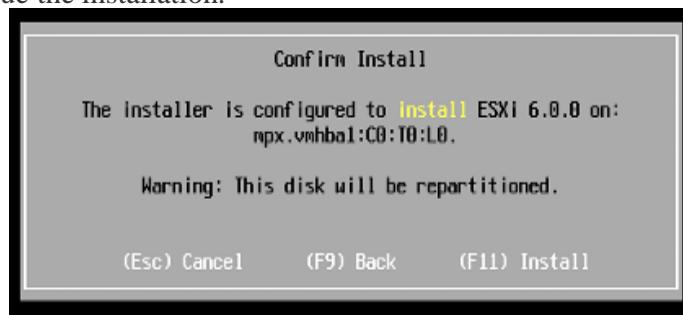
7. Its time to secure ESXi by providing the complex root password. Enter to continue.



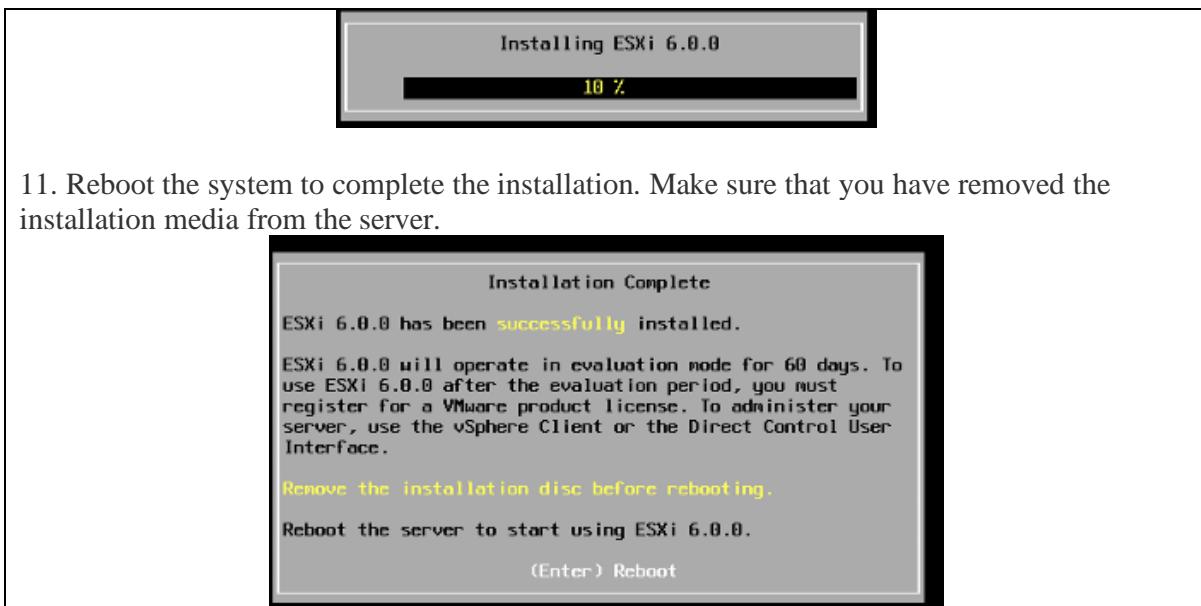
8. Here the Installer gathers the system information before start copying the files to the disk.



9. Here will be the final confirmation you need to give before destroying any data on that disk. Press F11 to continue the installation.



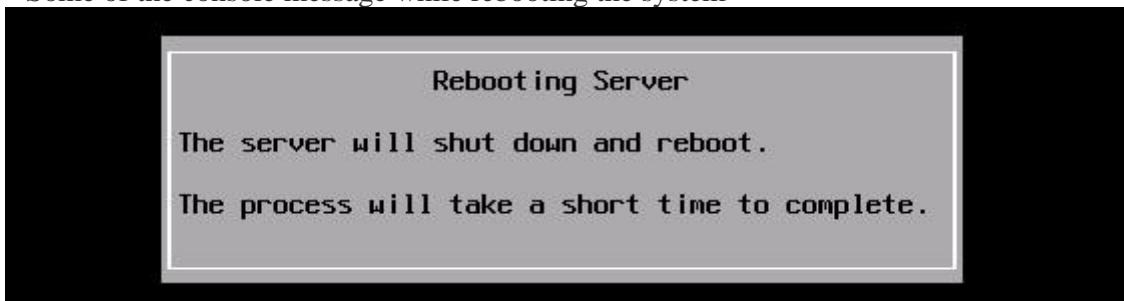
10. Installation begins.



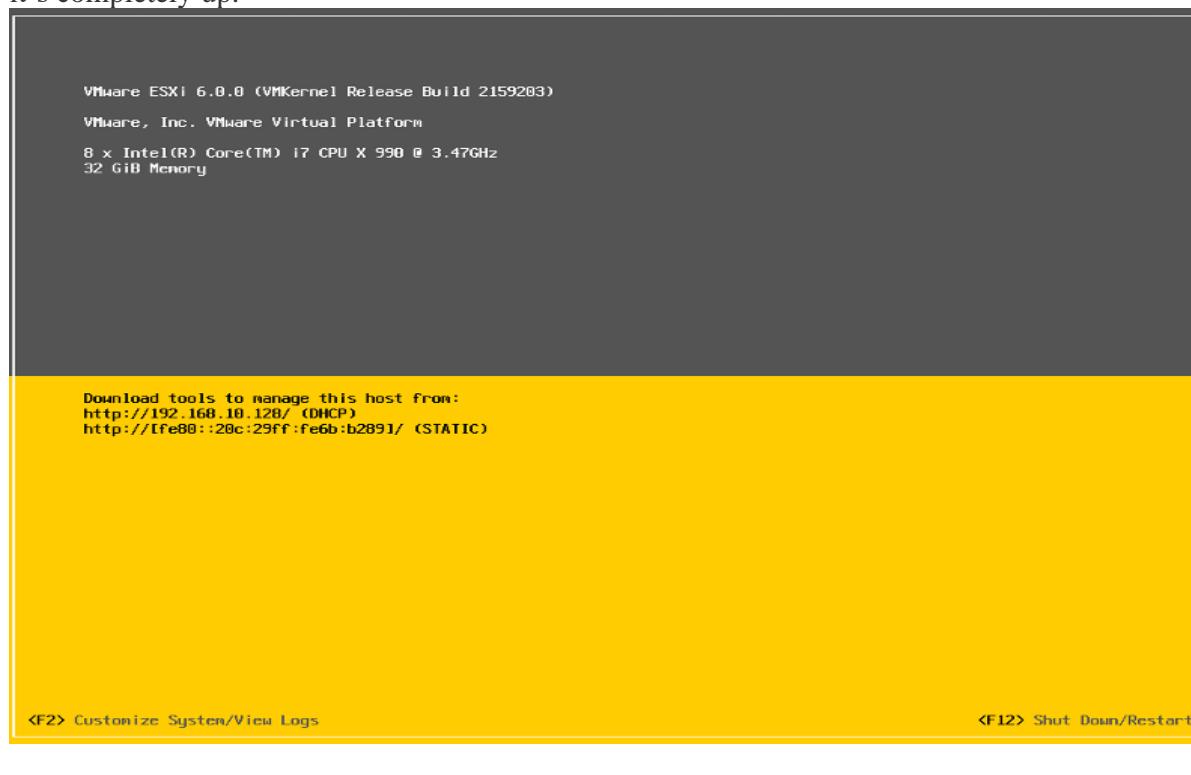
11. Reboot the system to complete the installation. Make sure that you have removed the installation media from the server.



Some of the console message while rebooting the system



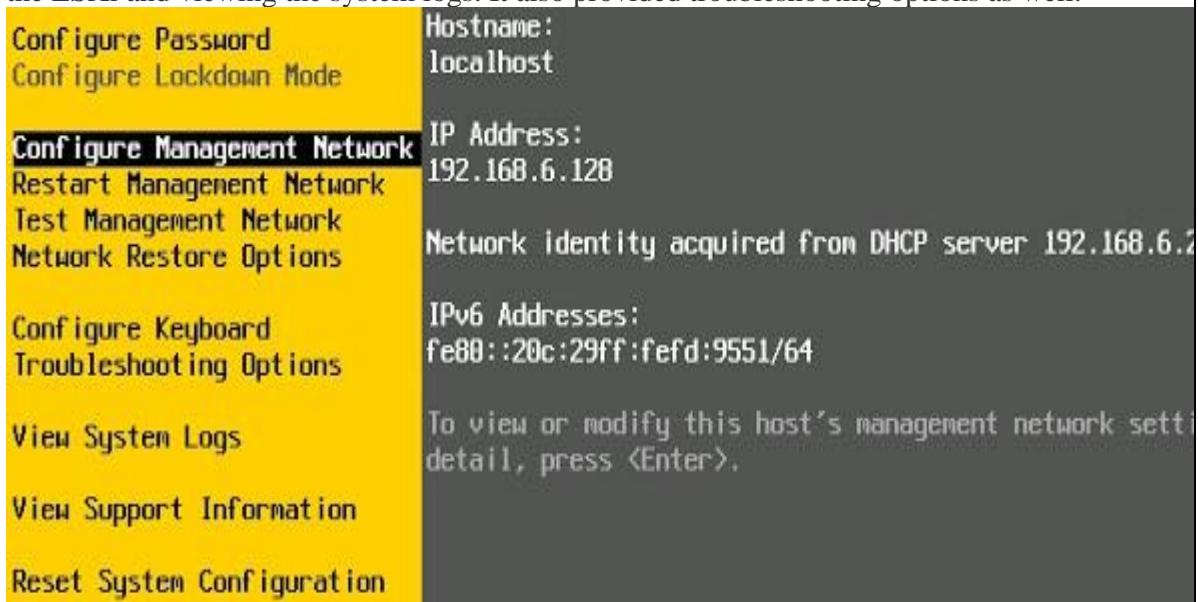
12. Now vmware EXSi 5.1 will boot from hard disk and you will get the below screen, once it's completely up.



13. Many of them wonder that, how to login to the system. You can get login screen by just pressing F2.



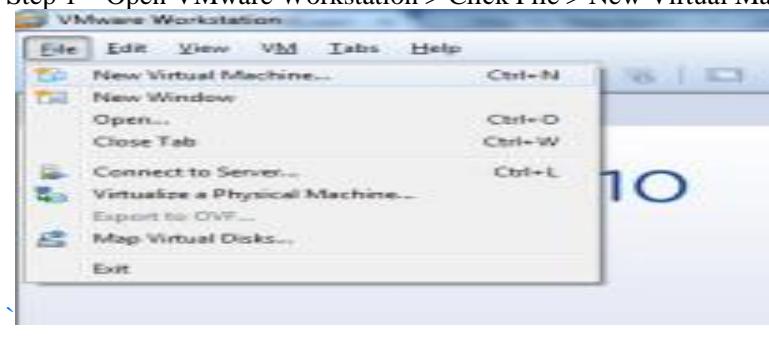
14. After logging to the system, you can see there are many options provided for configuring the ESXi and viewing the system logs. It also provided troubleshooting options as well.



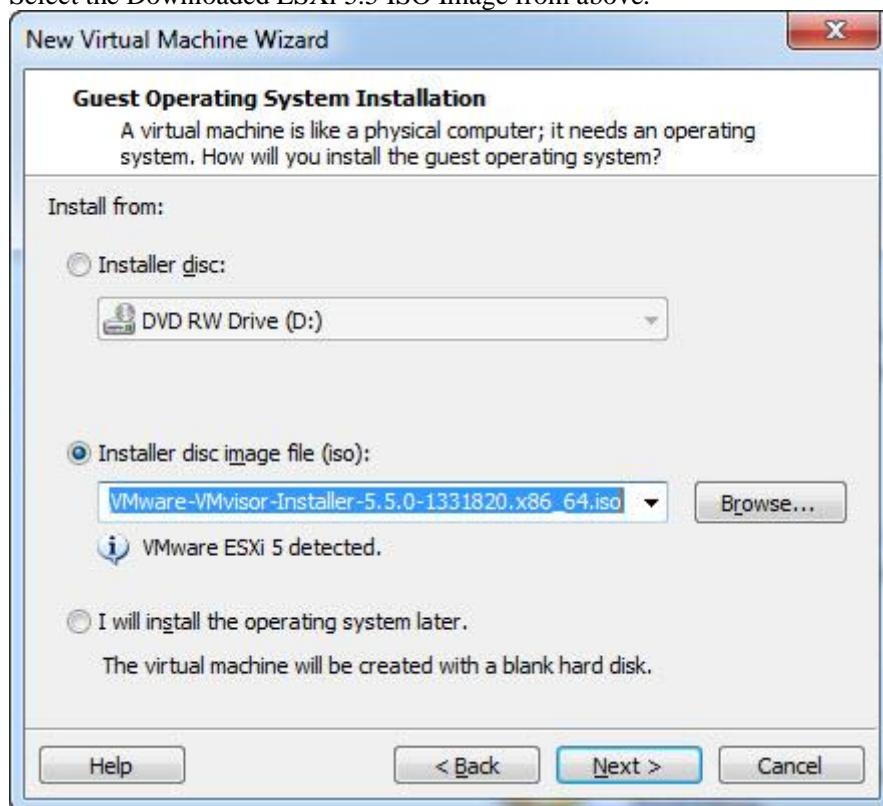
II. Set-Up Virtual Machine in VMware Workstation

The ESXi can also be installed and configured on VMware Workstation, not directly on hardware of the physical machine.

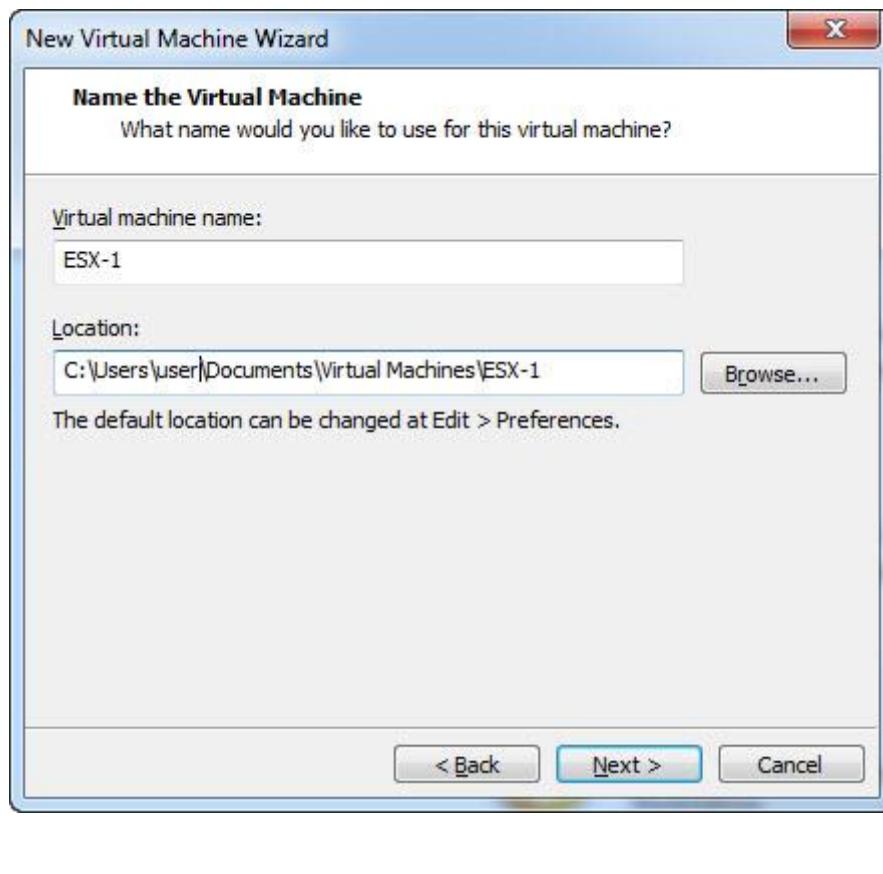
Step 1 – Open VMware Workstation > Click File > New Virtual Machine (Wizard)



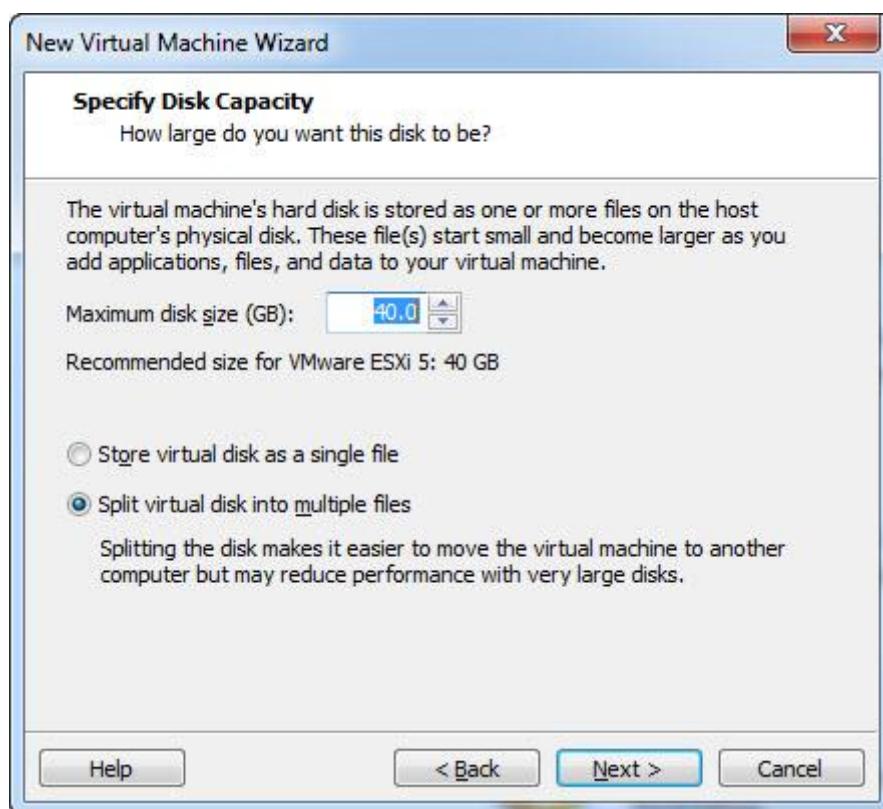
Step 2 – The New Virtual Machine will appear > Select Installer Disk Image File (ISO) > Browse & Select the Downloaded ESXi 5.5 ISO Image from above.



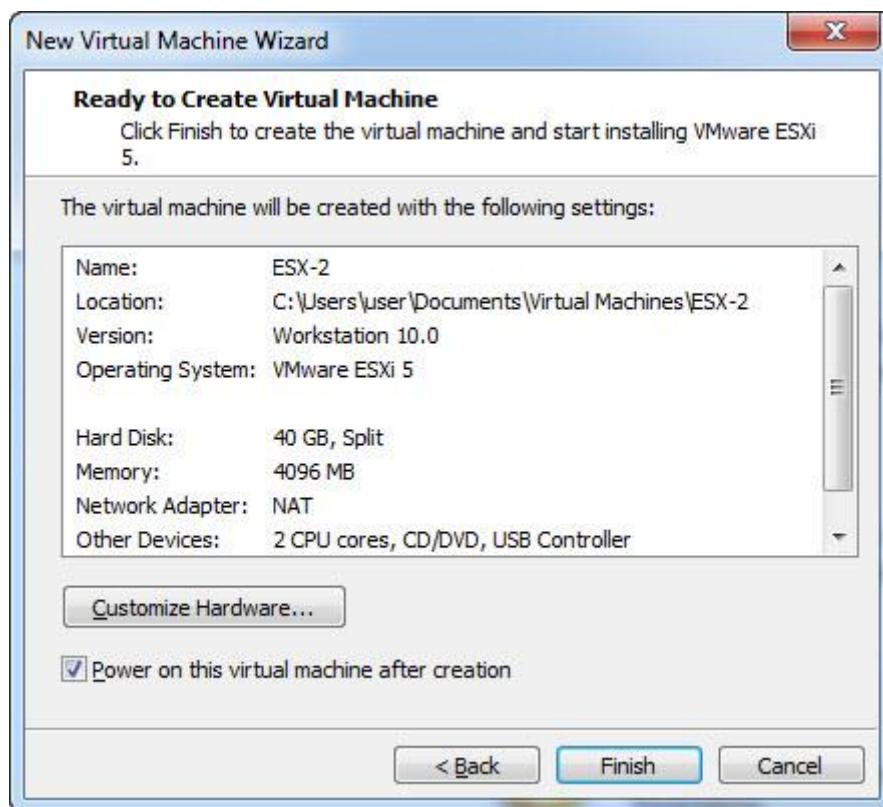
Step 3 – Give your virtual machine a name. For the purposes of this LAB it will be ESX-1.



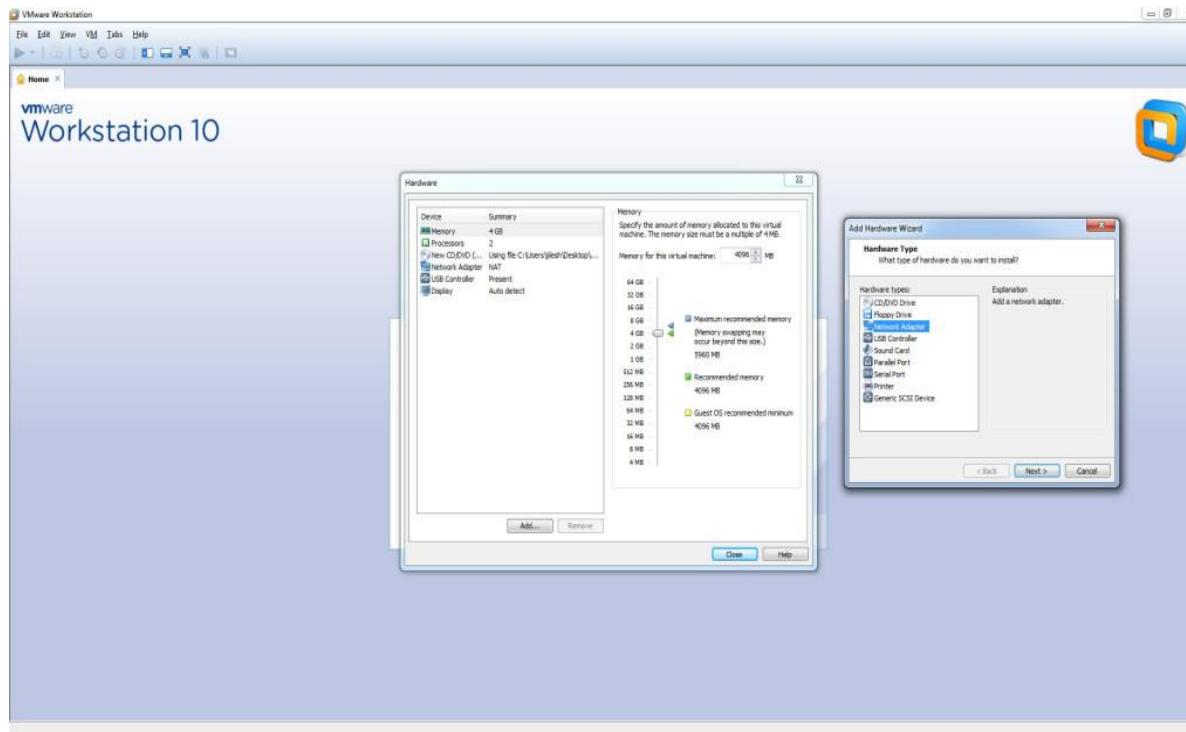
Step 4 – Specify the disk size. 40GB is the default and is more than enough.



Step 5 – Before you click Finish, select “Customise Hardware”.



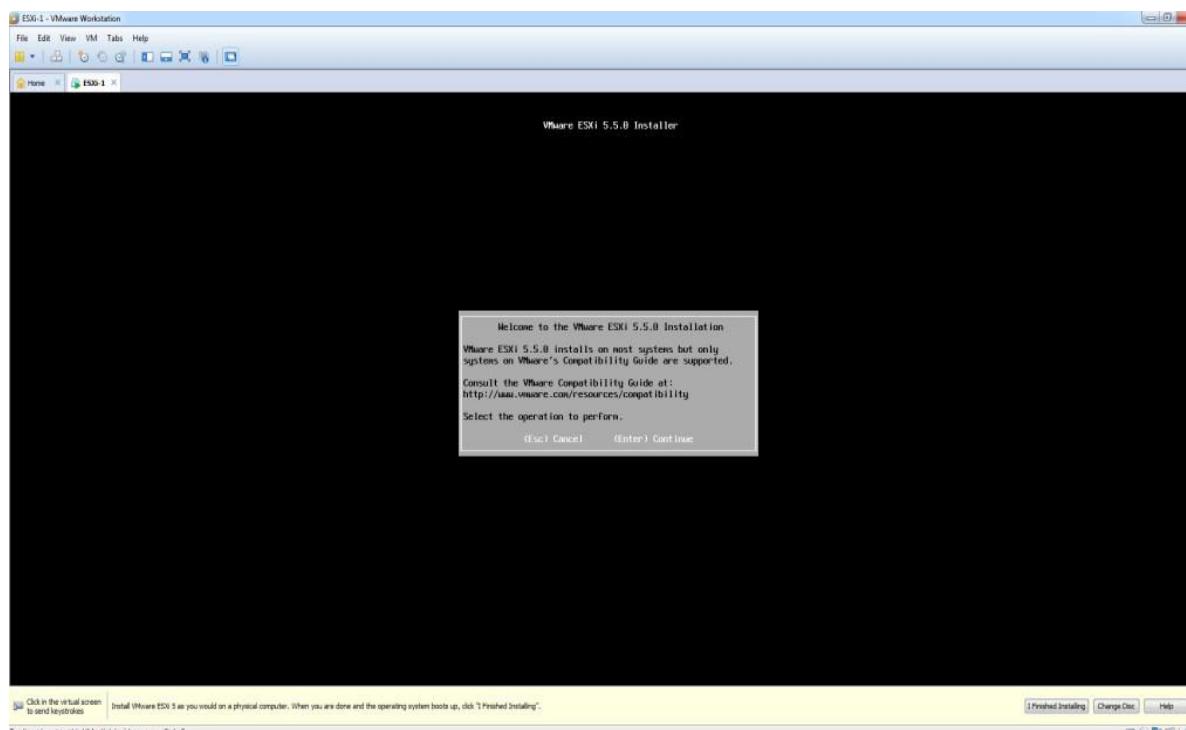
Step 6 – 3 Virtual Network Cards are required for your lab per ESXi Host. Select “Add” to open the Add Hardware Wizard and follow the steps to add 2 more network cards to your virtual machine.

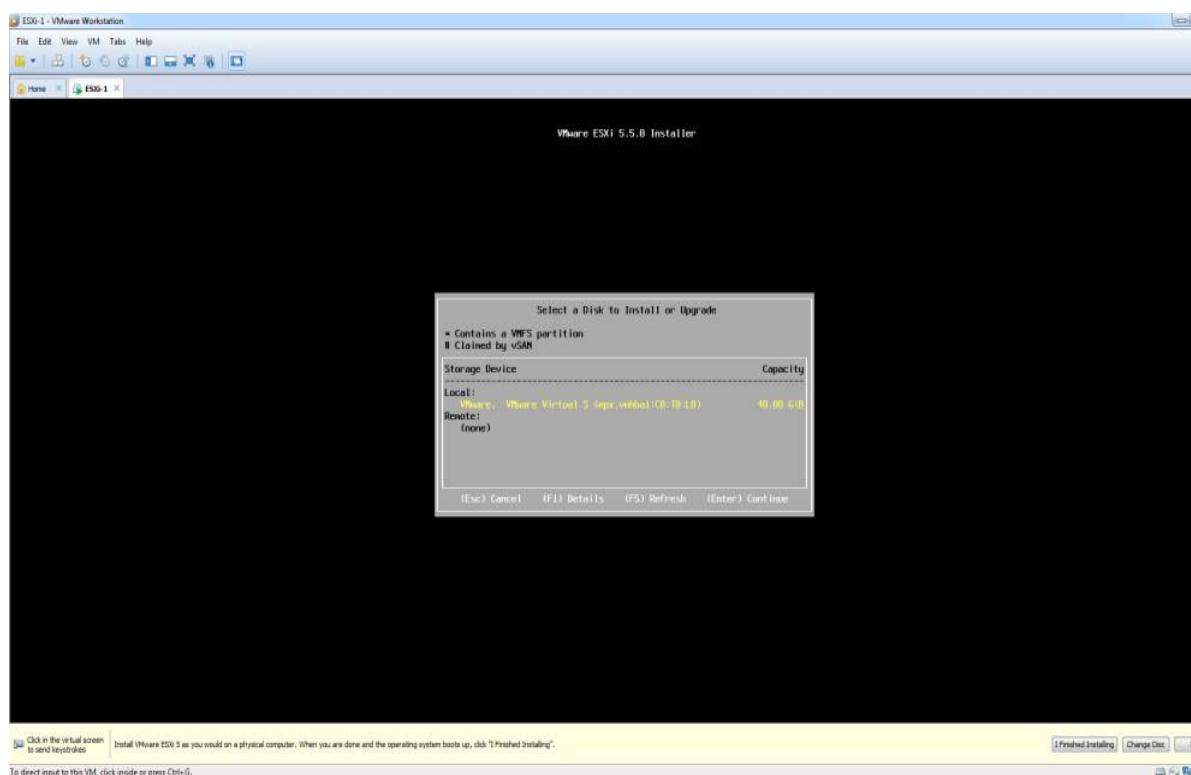


Step 7 – Click on Finish, and wait for your virtual machine to Power on Successfully.

INSTALL ESXI 5.5

Step 1 – Power on your Virtual / Physical Machine, and boot from the ESXi 5.5 ISO Image > Press Enter to Start the installation wizard.

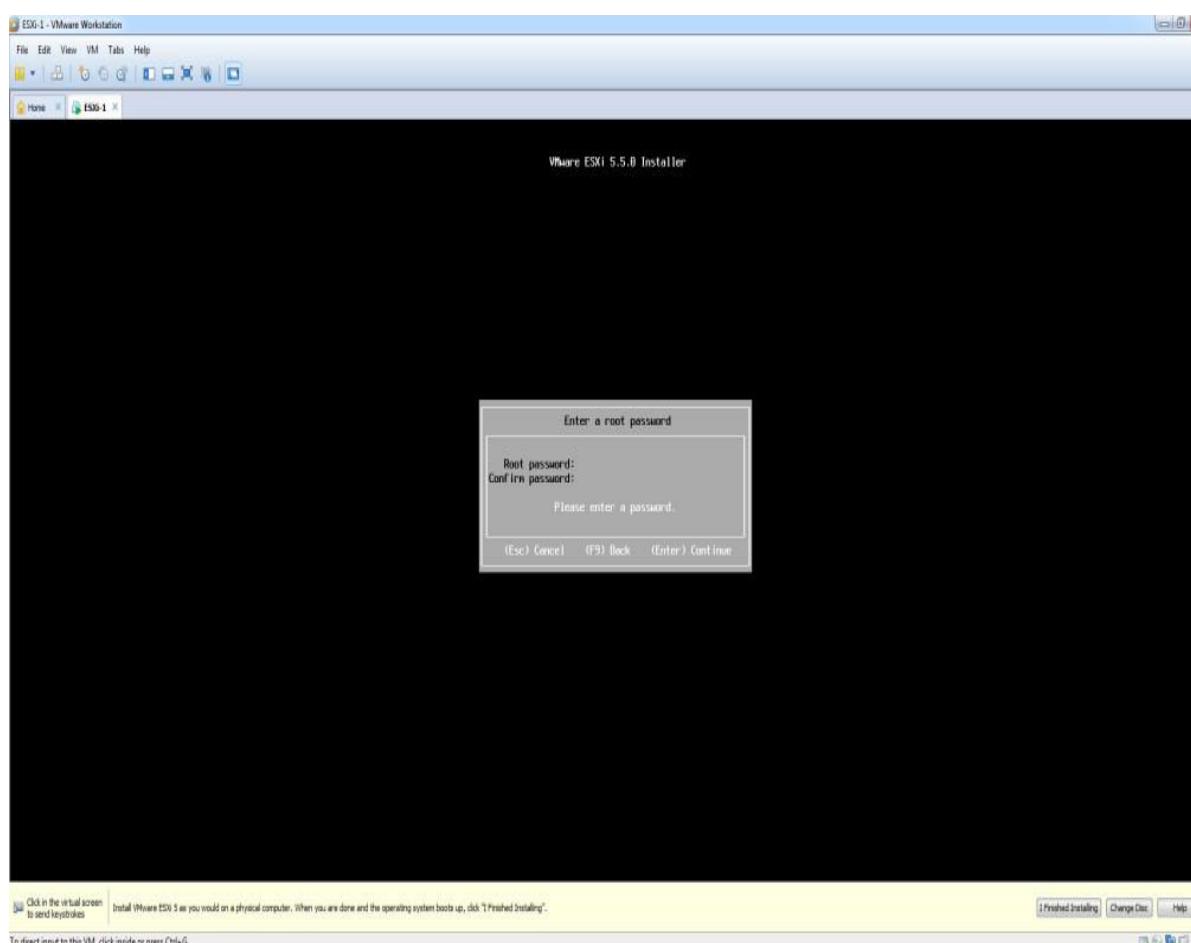


Step 2 – Accept EULA > Press F11**Step 3 – Select the Local Disk for the Installation > Press Enter**

Step 4 – Select your keyboard locale > Press Enter



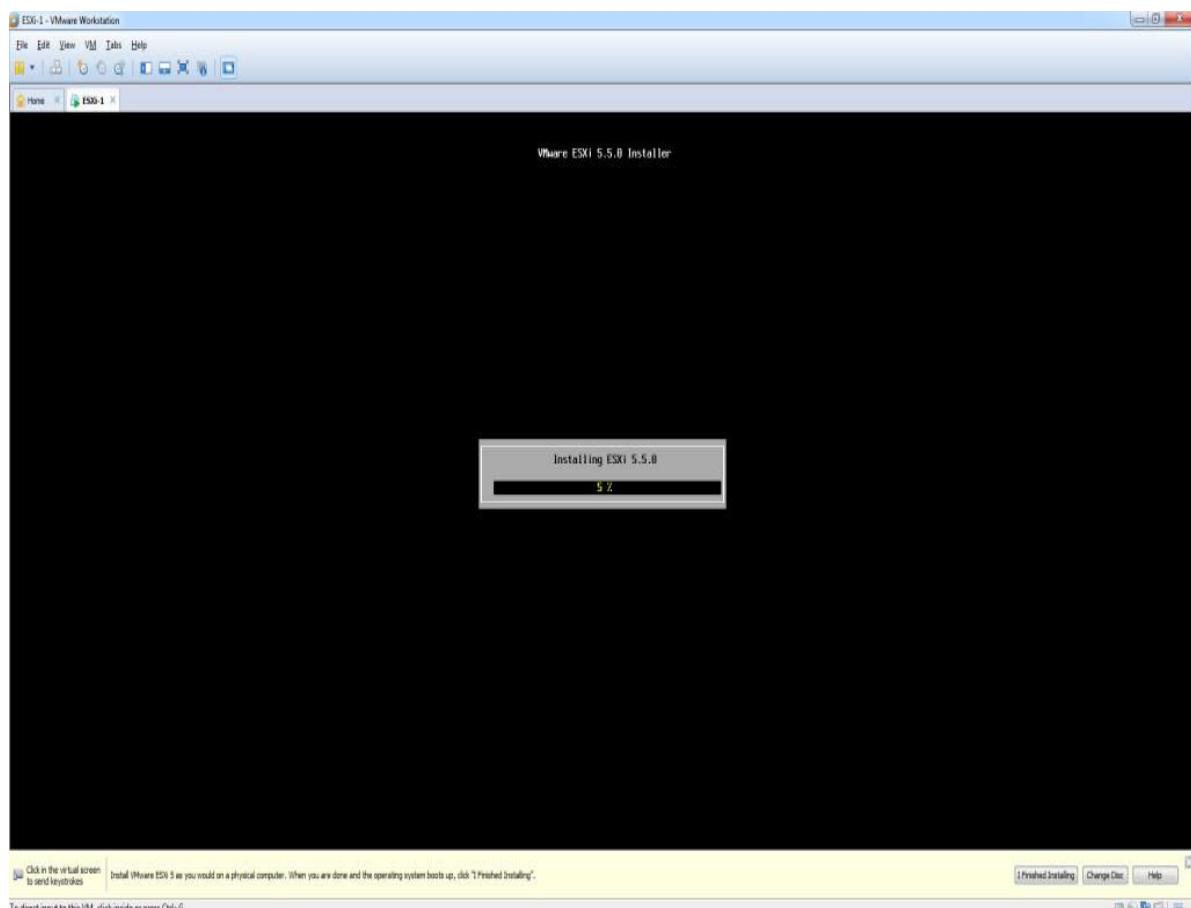
Step 5 – Type a complex password, with a minimum of 7 characters > Press Enter



Step 6 – Confirm > Press F11 to start the installation



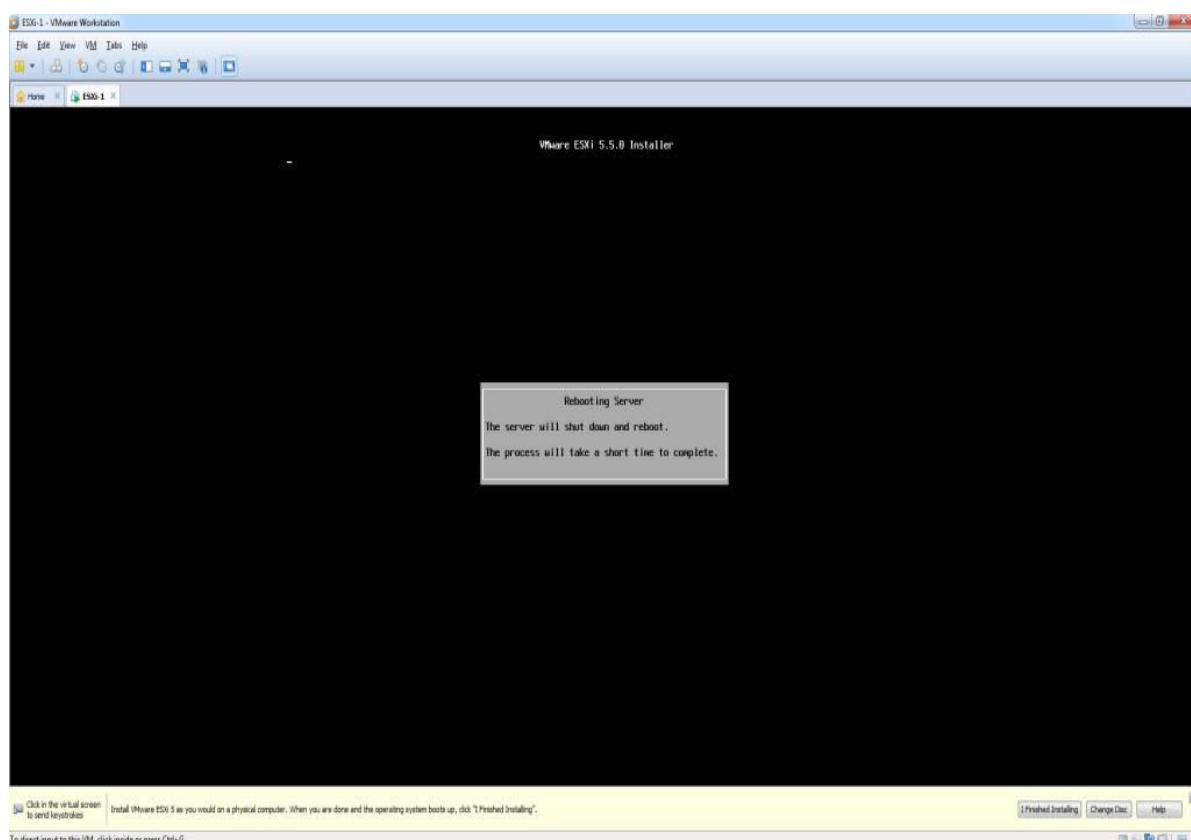
Step 7 – Installation will take about 2-5 minutes to complete.



Step 8 – Press Enter to Reboot once the installation has completed successfully.



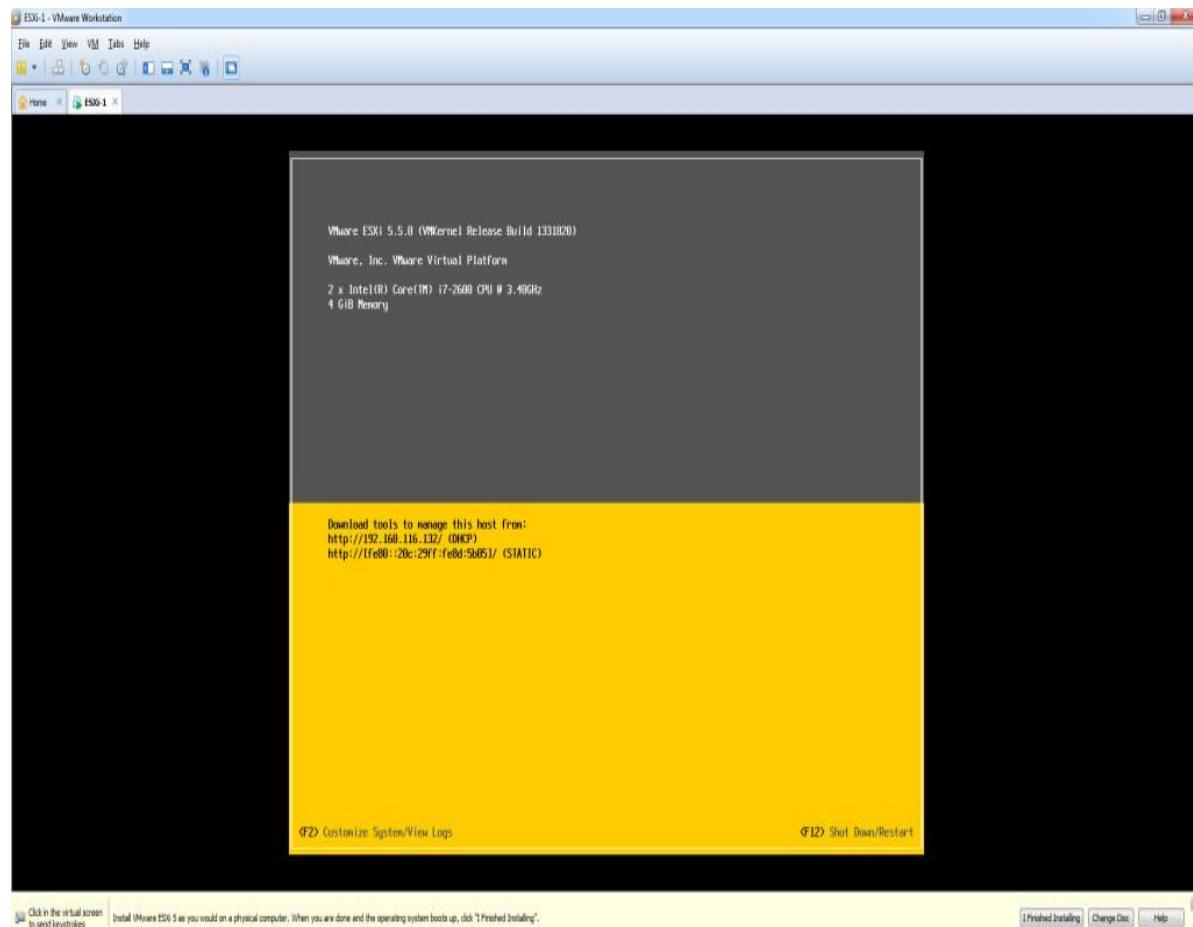
Step 9 – Wait for reboot to complete



III. CONFIGURE – HOSTNAME, IP ADDRESS, GATEWAY, DNS, PASSWORD ON ESXI 5.5

Once reboot has completed, your installation is now complete. The next step is to configure your ESXi 5.5 Host.

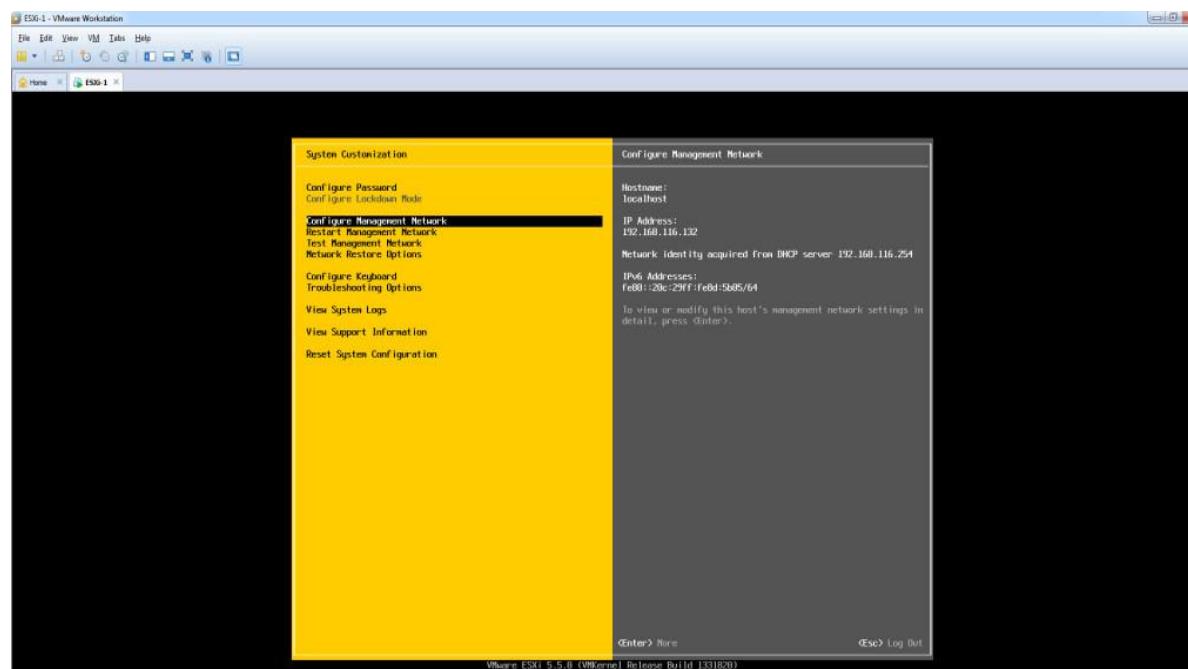
Step 1 – Press F2 to enter the configuration mode



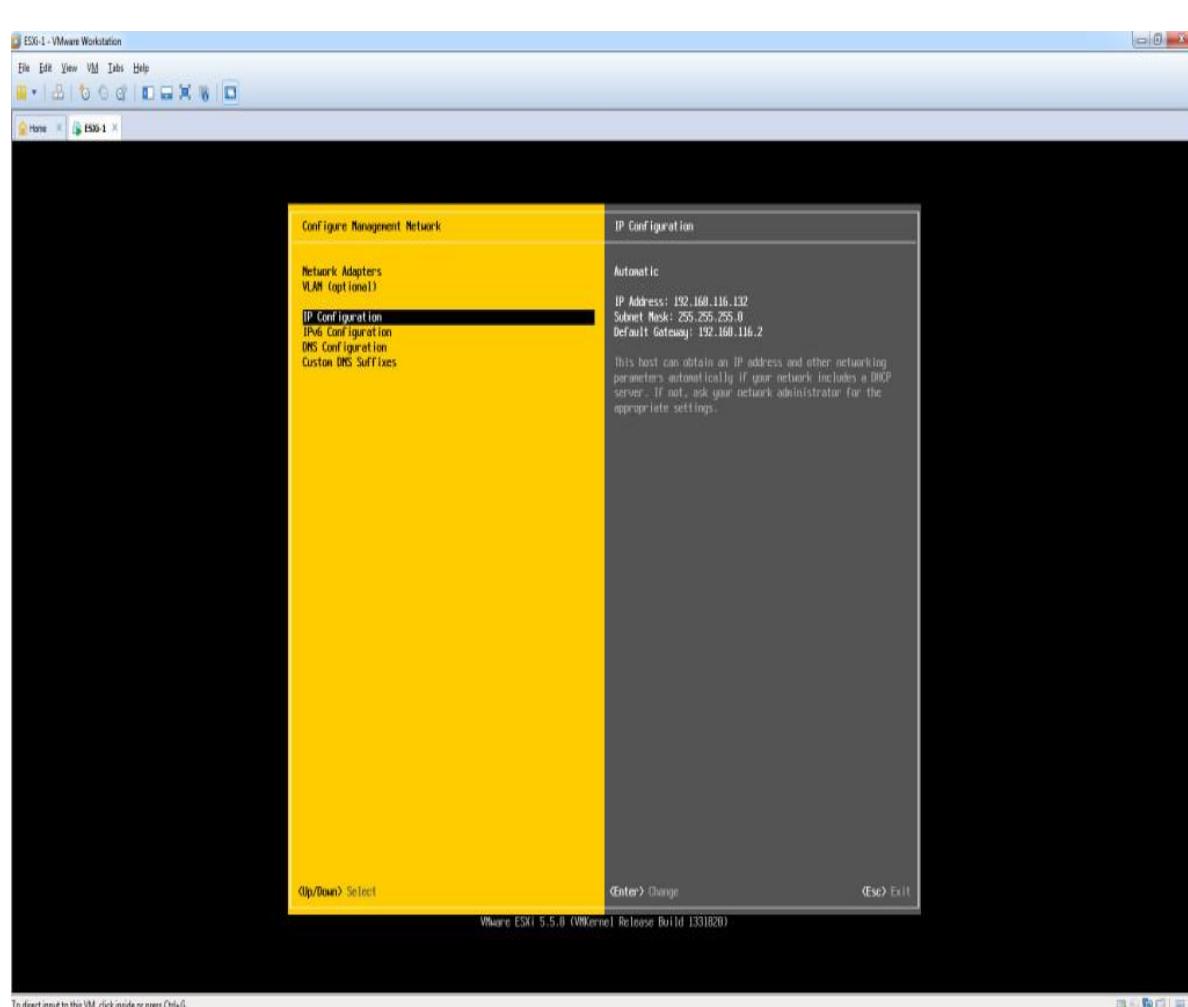
Step 2 – You will be prompted for the password which you created during the installation (note – the default user name is – root).



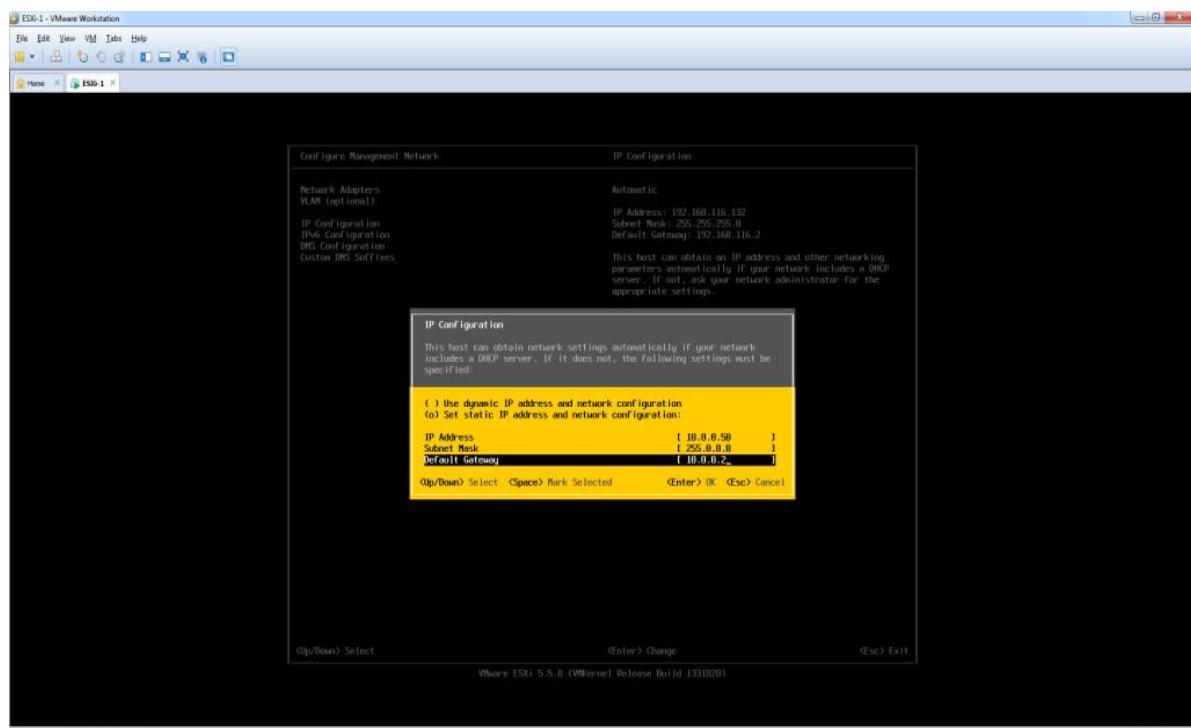
Step 3 – The next step is to configure the management network interface. Select “Configure Management Network” > Press Enter.



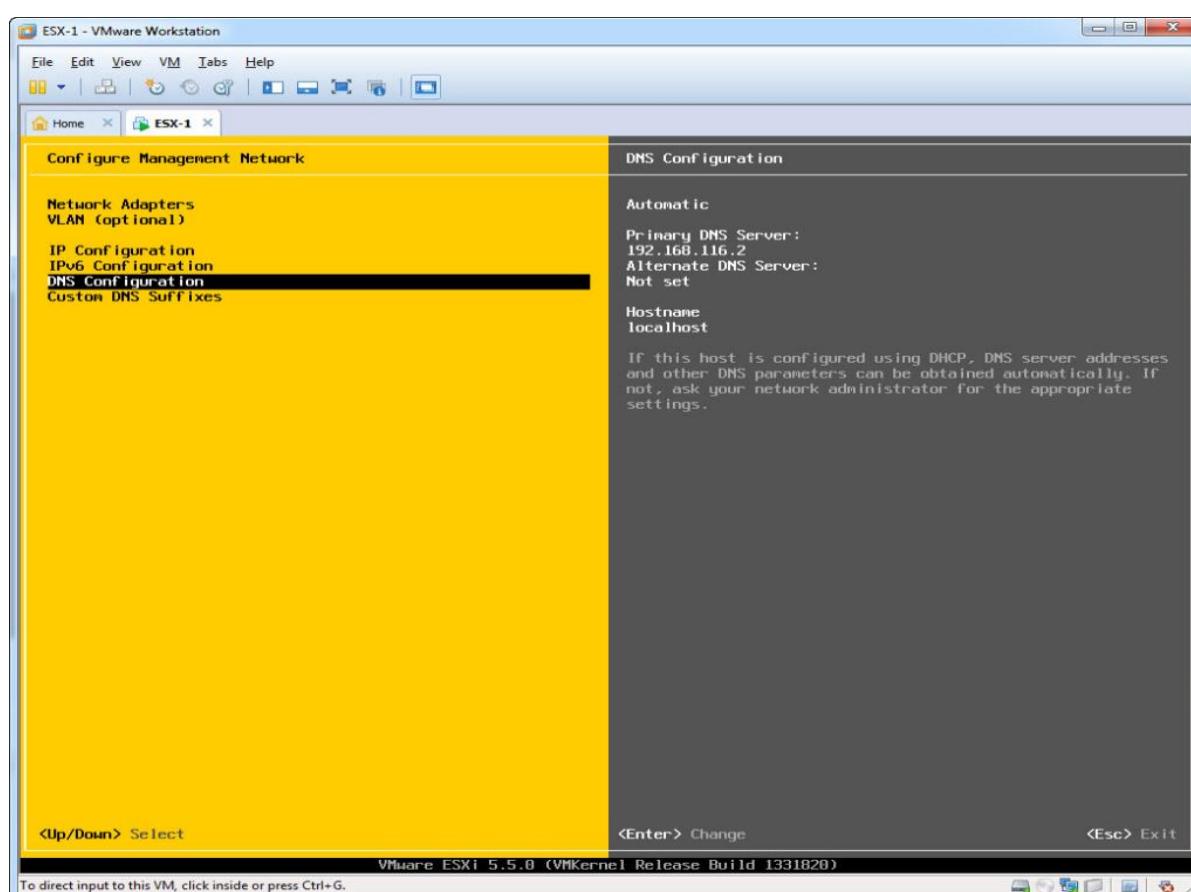
Step 4 – Select IP Configuration > Press Enter

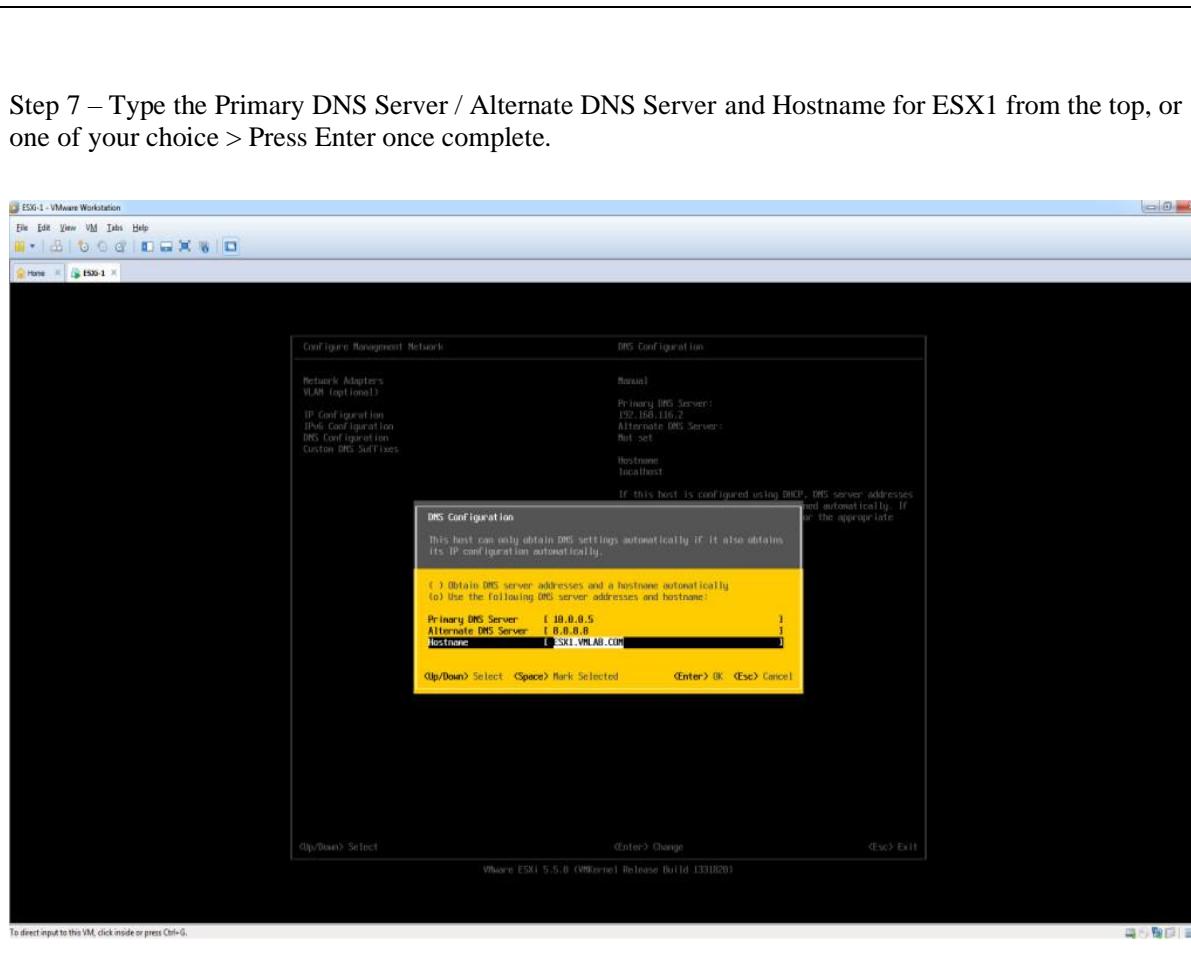


Step 5 – Select “Set Static IP Address and Network Configuration” > Type the static IP Address / Subnet mask and default gateway for ESX1 from the top, or one of your choice > Press Enter once complete.

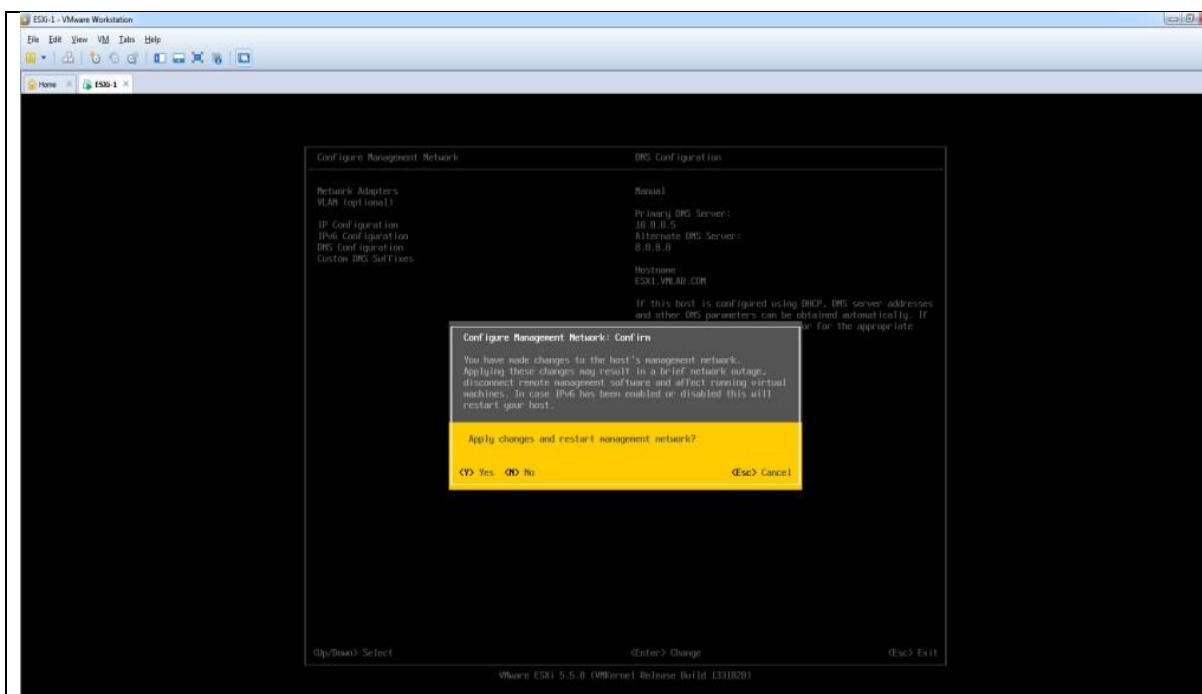


Step 6 – Select “DNS Configuration” > Press Enter

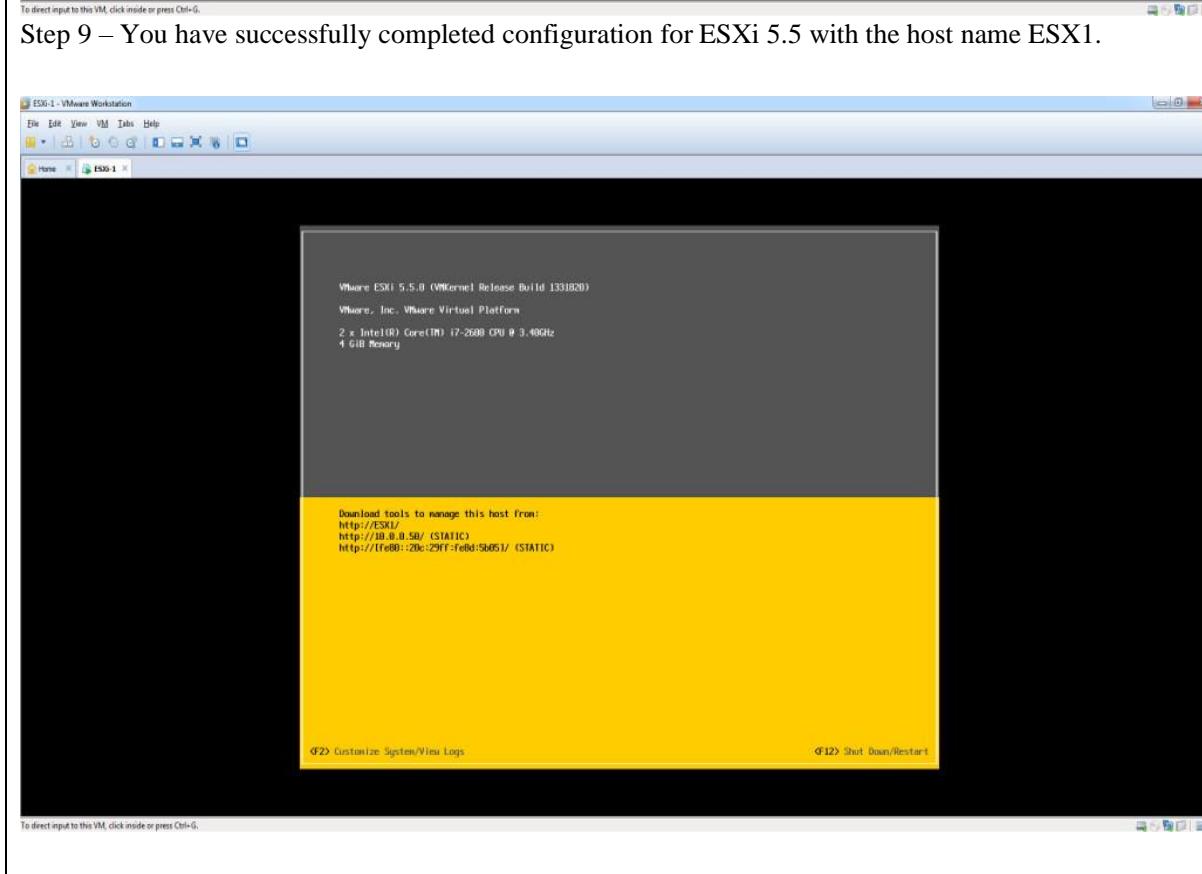




Step 8 – You will currently be in the “Configure Management Network” Menu. You have completed all configuration required for ESX 1. To Exit this menu and SAVE configuration > Press ESC > Type [Y] to “Apply changes and restart management network”. This will allow you to successfully exit the ESXi 5.5 configuration menu, back to the home screen.



Step 9 – You have successfully completed configuration for ESXi 5.5 with the host name ESX1.



Appendix C

1 Problem Statement:

Installation of VMware vSphere Client

2 Student Learning Outcomes:

To install and setup the VMware vSphere vClient.

3 Theoretical Description:

The vSphere Client is an interface for administering vCenter Server and ESXi.

The vSphere Client user interface is configured based on the server to which it is connected:

When the server is a vCenter Server system, the vSphere Client displays all the options available to the vSphere environment, according to the licensing configuration and the user permissions.

When the server is an ESXi host, the vSphere Client displays only the options appropriate to single host management

4 Requirements

Make sure that the vSphere Client hardware meets the minimum requirements.

- CPU – 1 CPU
- Processor – 500MHz or faster Intel or AMD processor (1GHz recommended)
- Memory – 1GB RAM
- Disk Storage – 1.5GB free disk space for a complete installation, which includes the following components:
 - Microsoft .NET 2.0
 - Microsoft .NET 3.0 SP1
 - Microsoft Visual J#
- Remove any previously installed versions of Microsoft Visual J# on the system where you are installing the vSphere Client.
- vSphere Client 6
- Networking – Gigabit connection recommended

5 Procedure

Login to any windows operating system where you want to configure vSphere Client to manage vCenter Server.

1. Copy the “VMware vSphere Client 6” directory to windows host and execute the exe file.

Name	Date modified	Type
checksums	9/26/2012 9:47 PM	Text Document
VMware-vclient-all-5.1.0-786111	9/26/2012 9:26 PM	Application



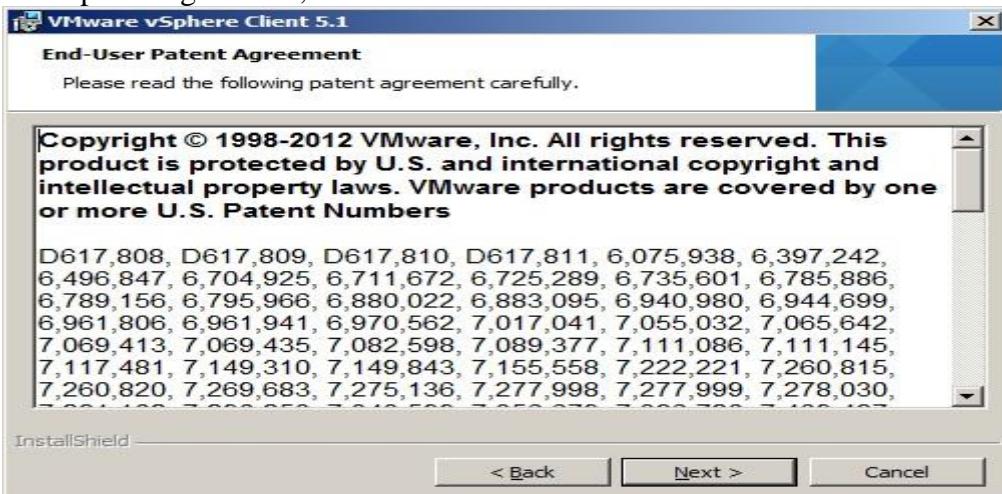
2. Press OK to continue the installation in English.



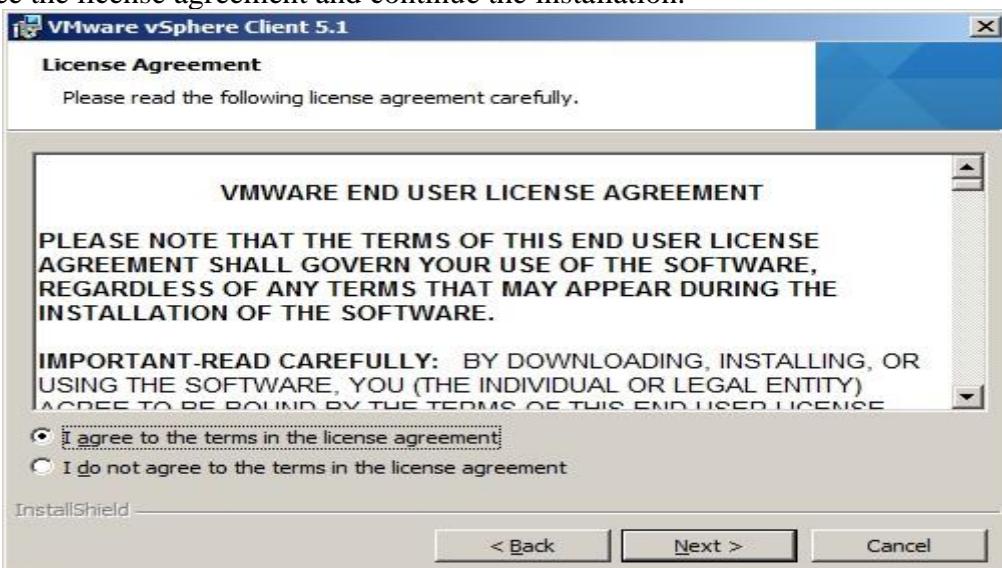
3. Click Next to continue the installation.



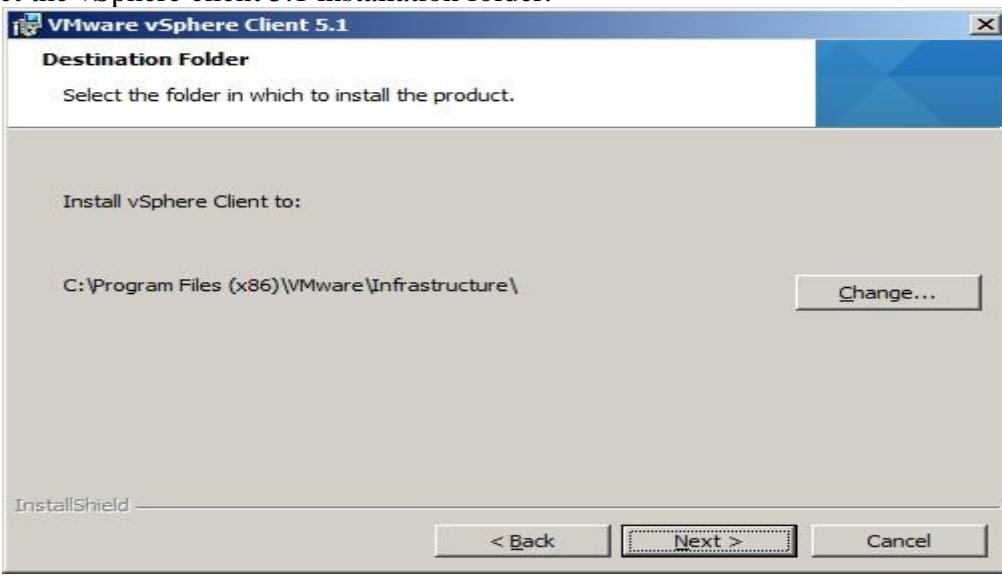
4.End User patent agreement, Click next to continue.



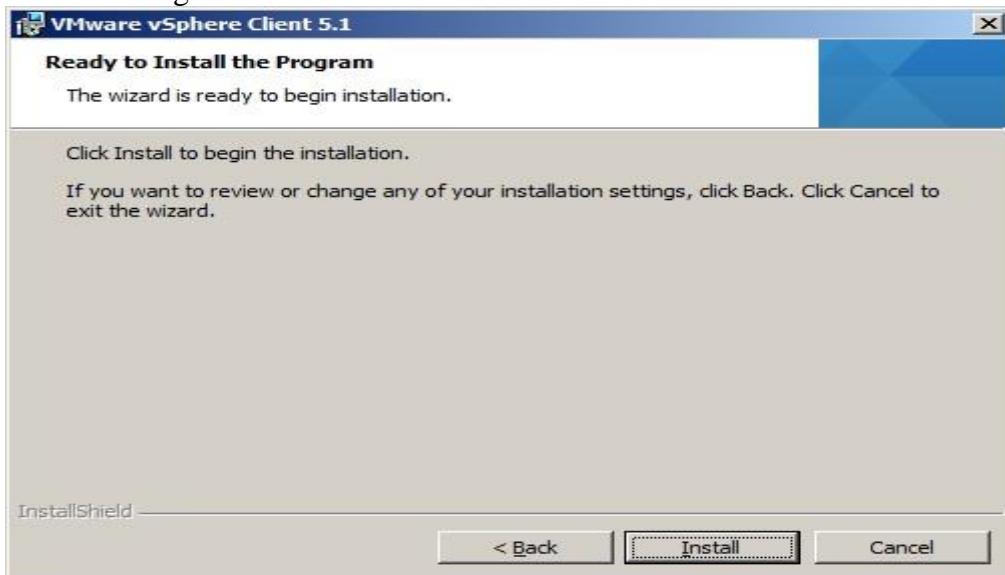
5.Agree the license agreement and continue the installation.



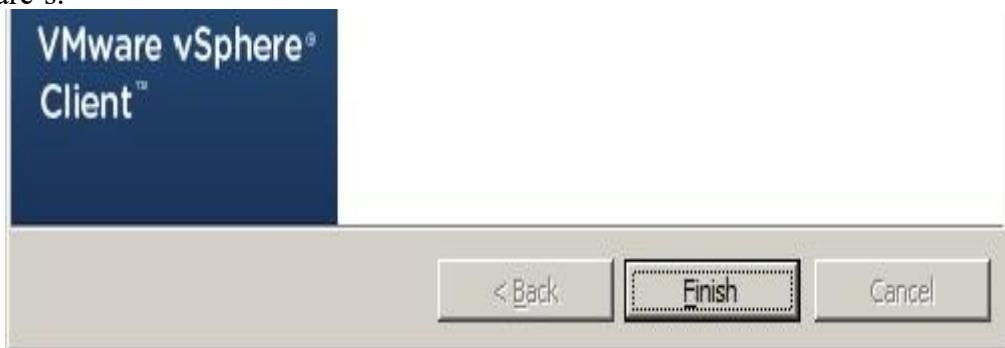
6.Select the vSphere client 5.1 installation folder.



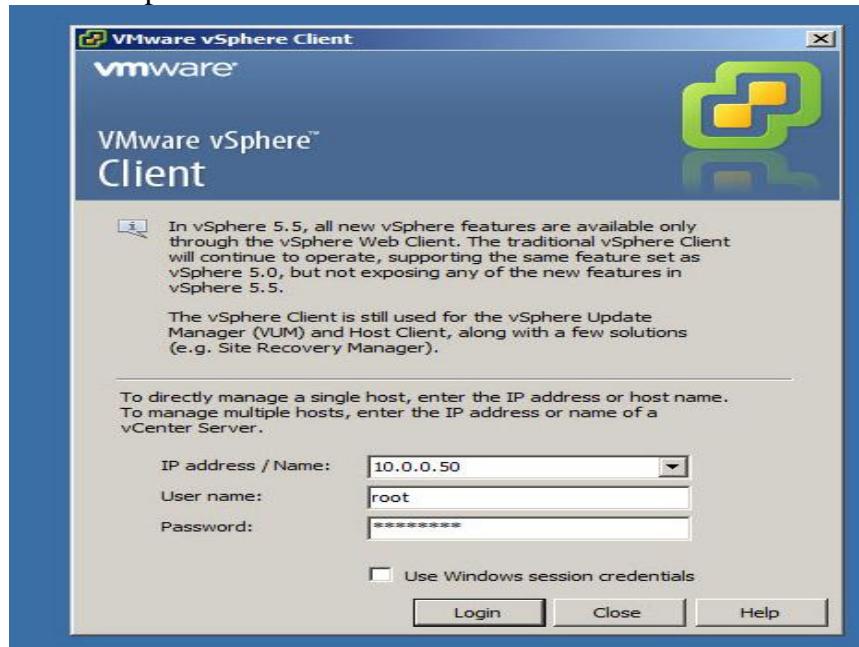
7.Click install to begin the installation.



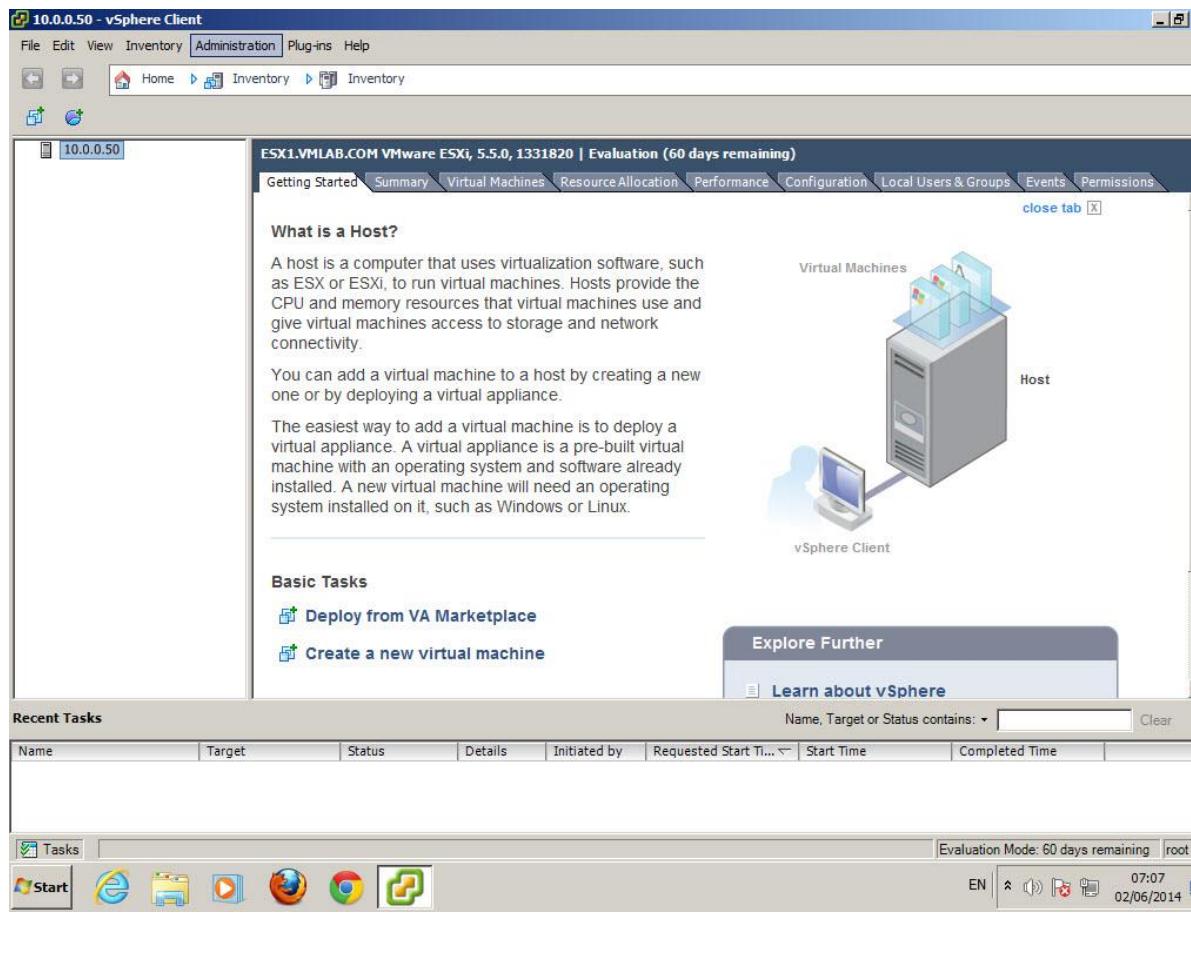
8. Click finish to complete the installation. The installation is so simple like other windows software's.



9.Start the Vmware vSphere Client 5.1 and connect to the VMware vCenter Server.



10. You are done with vSphere client installation and verified by logging in to vCenter server.



Recommended Course Resources:

1. <http://hadoop.apache.org/>
2. <https://www.macalester.edu/~shoop/sc13/hadoop/html/hadoop/wc-detail.html>
3. https://en.wikipedia.org/wiki/K-means_clustering
4. <https://github.com/himank/K-Means>
5. <https://www.arduino.cc/>
6. <https://www.arduino.cc/en/Main/arduinoBoardUno>
7. <https://en.wikipedia.org/wiki/Arduino>
8. www.arduino.org/
9. www.vmware.com
10. <http://www.vmware.com/products.html>