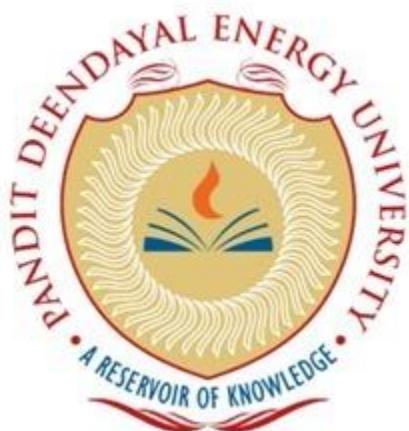


**Pandit Deendayal Energy University**  
**School of Technology**



**Cloud Computing Lab (20CP322P)**  
**B.Tech-Computer Science & Engineering (Sem-VI)**

**PATEL VEDANT H.**

**19BCP138**

**DIVISION – 2 (G4)**

## **Table of Contents:**

<b>List Of Experiments</b>	<b>Page No.</b>
1. Study of Various Virtualization Software	3 - 11
2. XenServer & XenCenter	12 - 20
3. To Install and Configure VMware Workstation Pro	21 - 33
4. To Install and Configure VMware Workstation Pro for creating Virtual Machines, Cloning and Deleting VMs.	34 - 44
5. Docker	45 - 50
6. Cloud Analyst	51 - 64
7. Microsoft Azure	65 - 73
8. CloudSim	74 - 80

# **Experiment - 1**

## **Study of Various Virtualization Software**

❖ **Aim:** Study of Various Virtualization Software such as VMware, QEMU, KVM, XEN Hypervisor and Oracle Virtual Box.

- Perform the following:
1. Compare their Virtualization type and methodologies.
  2. Identify architectural difference.
  3. List out various advantages and disadvantages.
  4. Applications of virtualization software.

### **1. Compare their Virtualization type and methodologies:**

#### **➤ VMware:**

VMware is the leading virtualization software in the web hosting field. VM virtual machine makes it possible for more than one instance of the operating systems to run on the same server. VMware also allows you to allocate hard disk space from the available hard disk storage pool. The drives you set up will appear as if it is a physical hard disk that is installed on the server. VMware supports all kinds of OS including Windows, Linux, and Unix.

The VMware design methodology is broken down into three or sometimes four phases:

- The first phase is the architectural vision, during which the scope will be defined, requirements listed and goals assigned. This will serve as a guide for the rest of the project.
- The second is the architectural analysis. This is a much more in-depth look at the existing infrastructure. Information is gathered and analyzed on existing servers, storage, and networks. This information will be needed later on, so compiling detailed information is vital.
- The third phase is the technology architecture, where all the information recorded earlier or is used to create conceptual, logical, and physical designs.
- The fourth stage, if used, is a migration planning phase, where the information is valuable for their migration from physical to virtual.

## ➤ QEMU:

QEMU is virtualization software that emulates processors and supports a variety of virtual devices (such as HDD, RAM, sound, Ethernet, USB, VGA, etc.). QEMU is a type 1 hypervisor that is free and open-source. It uses dynamic binary translation to imitate the machine's processor and provides a choice of hardware and device models for the machine, allowing it to run a range of guest operating systems.

QEMU has various operating modes such as:

- User-mode emulation: QEMU runs single Linux or Darwin/macOS programs that were compiled for a different instruction set.
- System emulation: QEMU emulates a full computer system, including peripherals. It can be used to provide virtual hosting of several virtual computers on a single computer.
- KVM hosting: QEMU deals with the setting up and migration of KVM images.
- Xen hosting: QEMU is involved only in the emulation of hardware; the execution of the guest is done within Xen and is hidden from QEMU.

## ➤ KVM:

KVM is a type of virtualization. KVM (Kernel-based Virtual Machine) is an open-source virtualization technology that comes pre-installed with Linux®. KVM, in particular, allows you to turn Linux into a hypervisor, allowing you to operate several, and segregated virtual environments called guests or virtual machines on a single host machine (VMs). It consists of a loadable kernel module, kvm.ko that provides the core virtualization infrastructure and a processor-specific module, kvm-intel.ko or kvm-amd.ko. Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

## ➤ XEN Hypervisor:

Xen is an open-source type-1 or BareMetal hypervisor, which makes it possible to run many instances of an operating system or indeed different operating systems in parallel on a single machine (or host). Xen is the only type-1 hypervisor that is available as open-source. Xen is used as the basis for several different commercial and open-source applications, such as server virtualization, Infrastructure as a Service (IaaS), desktop virtualization, security applications, embedded and hardware appliances. Xen enables users to increase server utilization, consolidate server farms, reduce complexity, and decrease the total cost of ownership.

Xen offers five approaches to running the guest operating system:

- HVM (hardware virtual machine)
- HVM with PV drivers
- PVHVM (paravirtualization with full hardware virtualization, i.e., HVM with PVHVM drivers)
- PVH (PV in an HVM container)
- PV (paravirtualization).

## ➤ Oracle Virtual Box:

Virtual Box is a Type 2 hypervisor. That is to say that it is virtualization host software that runs as an application on an established operating system. Hardware-assisted software virtualization is available with CPU chips with built-in virtualization support. Recently, with the introduction of the Intel VT and AMD-V technology, this virtualization type has become commoditized. This technology was first introduced on the IBM System/370 computer. It is similar to software virtualization, with the exception that some hardware functions are accelerated and assisted by hardware technology. The type 2 hypervisor works with both full software virtualization and hardware-assisted software virtualization. Oracle VM Virtual Box is a type 2 hypervisor-based product.

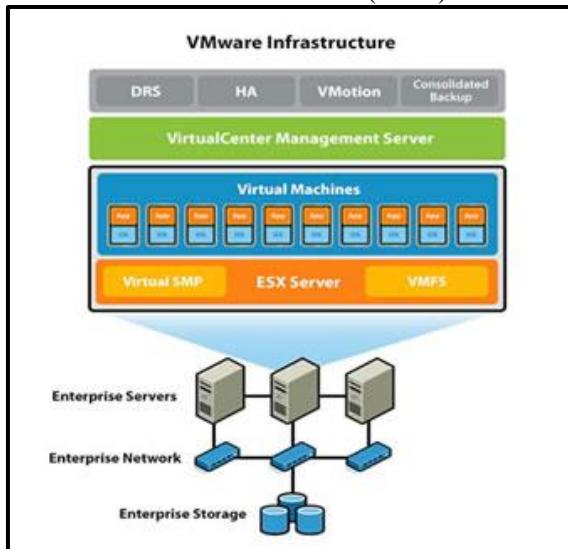
## 2. Identify Architectural Differences:

### ➤ VMware:

VMware Infrastructure is a suite of infrastructure virtualization that provides a complete picture of application availability, virtualization, resource optimization, management, and operational automation. VMware Infrastructure consolidates and virtualizes hardware resources between two or more systems.

In a virtual environment, you provide a virtual resource pool for your data center. High availability, consolidated backup, fine-grain, policy-driven, and resource allocation of the whole virtual data center are all elements of VMware Infrastructure's distributed architecture. These distributed architecture services play a critical part in establishing an IT organization by ensuring that service level agreements and production are met cost-effectively.

The architecture includes VMware ESX Server, Virtual Center Management Server (Virtual Center Server), VMware High Availability, Virtual Infrastructure Client (VI Client), Virtual Infrastructure Web Access, VMware Virtual Machine File System, VMware Virtual Symmetric Multi-Processing (SMP), VMware Virtual Motion, VMware Backup, VMware Infrastructure SDK, Distributed Resource Scheduler (DRS).



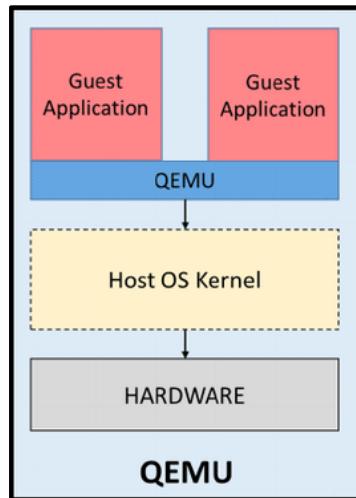
## ➤ QEMU:

QEMU is a fast, cross-platform Open-Source machine emulator which can emulate a huge number of hardware architectures for you. QEMU lets you run a completely unmodified operating system (VM Guest) on top of your existing system (VM Host Server).

You can also use QEMU for debugging purposes - you can easily stop your running virtual machine, inspect its state and save and restore it later.

QEMU consists of the following parts:

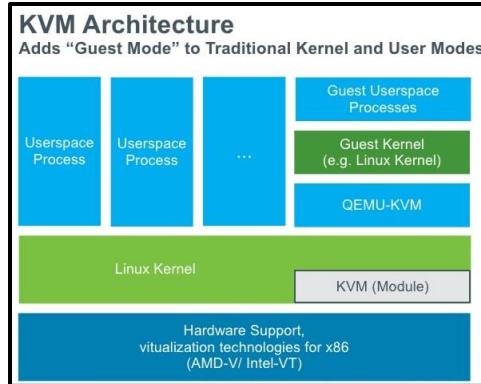
- Processor Emulator (x86, s390x, PowerPC, Sparc ...)
- Emulated Devices (graphic card, network card, hard drives, mice ...)
- Generic devices used to connect the emulated devices to the related host devices
- Descriptions of the emulated machines (PC, Power Mac ...)
- Debugger
- The user interface used to interact with the emulator



## ➤ KVM:

QEMU is a user-space process that runs on top of the Linux kernel using the KVM module, and it runs a guest kernel on the emulated hardware.

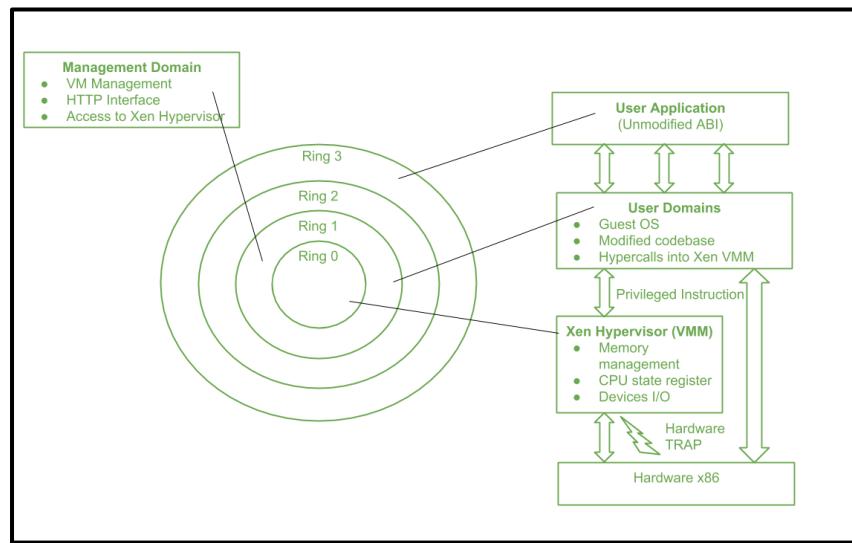
For hardware-based virtualization, QEMU and KVM can function together (Intel VT-x or AMD AMD-V). QEMU is extremely fast since it does not have to replicate all CPU instructions thanks to hardware-based virtualization. we run this on x86-64 based architecture CPU, with the help of KVM support (hardware-based virtualization), the emulated chipset should be q35, and the size of memory should be 8GB, and so on.



## ➤ XEN Hypervisor:

The below figure describes the Xen Architecture and its mapping onto a classic x86 privilege model. A Xen-based system is handled by Xen hypervisor, which is executed in the most privileged mode and maintains the access of the guest operating system to the basic hardware.

Guest operating systems are run between domains, which represents virtual machine instances. In addition, particular control software, which has privileged access to the host and handles all other guest OS, runs in a special domain called Domain 0. This is the only one loaded once the virtual machine manager has fully booted, and hosts an HTTP server that delivers requests for virtual machine creation, configuration, and termination. This component establishes the primary version of a shared virtual machine manager (VMM), which is a necessary part of a Cloud computing system delivering an Infrastructure-as-a-Service (IaaS) solution. Various x86 implementations support four distinct security levels, termed rings.

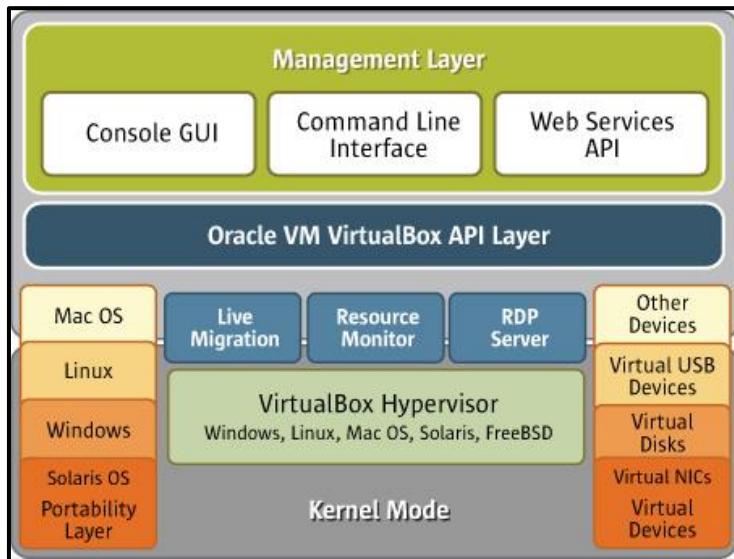


## ➤ Oracle Virtual Box:

Virtual Box uses a layered architecture consisting of a set of kernel modules for running virtual machines, an API for managing the guests, and a set of user programs and services. At

the core is the hypervisor, implemented as a ring 0 (privileged) kernel service. Figure shows the relationships between all of these components. The kernel service consists of a device driver named vboxsrv, which is responsible for tasks such as allocating physical memory for the guest virtual machine, and several loadable hypervisor modules for things like saving and restoring the guest process context when a host interrupt occurs, turning control over to the guest OS to begin execution, and deciding when VT-x or AMD-V events need to be handled.

The hypervisor does not get involved with the details of the guest operating system scheduling. Instead, those tasks are handled completely by the guest during its execution. The entire guest is run as a single process on the host system and will run only when scheduled by the host. If they are present, an administrator can use host resource controls such as scheduling classes and CPU caps or reservations to give a very predictable execution of the guest machine.



## ➤ Comparison:

	Xen	KVM	VirtualBox	VMWare
<b>Para-virtualization</b>	Yes	No	No	No
<b>Full virtualization</b>	Yes	Yes	Yes	Yes
<b>Host CPU</b>	x86, x86-64, IA-64	x86, x86-64, IA64, PPC	x86, x86-64	x86, x86-64
<b>Guest CPU</b>	x86, x86-64, IA-64	x86, x86-64, IA64, PPC	x86, x86-64	x86, x86-64
<b>Host OS</b>	Linux, UNIX	Linux	Windows, Linux, UNIX	Proprietary UNIX
<b>Guest OS</b>	Linux, Windows, UNIX	Linux, Windows, UNIX	Linux, Windows, UNIX	Linux, Windows, UNIX
<b>VT-x / AMD-v</b>	Opt	Req	Opt	Opt
<b>Cores supported</b>	128	16	32	8
<b>Memory supported</b>	4TB	4TB	16GB	64GB
<b>3D Acceleration</b>	Xen-GL	VMGL	Open-GL	Open-GL, DirectX
<b>Live Migration</b>	Yes	Yes	Yes	Yes
<b>License</b>	GPL	GPL	GPL/proprietary	Proprietary

### **3. List out various Advantages and Dis-Advantages:**

#### **➤ VMware:**

##### **❖ Advantages:**

- Cost-Effective Software
- Fast Rollback Feature
- Adding New Virtual Machine is Easy
- Easy to Delete Virtual Machine

##### **❖ Dis-Advantages:**

- Need Handling Skills
- Not Best for Complete Physical Product Testing
- Lack of Reliability
- Low Performance

#### **➤ QEMU:**

##### **❖ Advantages:**

- Supports emulating IA-32 (x86) PCs, x86-64|AMD64 PCs, MIPS R4000, Suns SPARC sun3 and PowerPC (PReP and Power Macintosh) architectures.
- Support for other architectures in both host and emulated systems.
- Increased speed—some applications can run in close to real time.
- Can save and restore the state of the machine. (Programs running, etc.)

##### **❖ Dis-Advantages:**

- Incomplete support for Microsoft Windows and other host operating systems. (emulation of these systems is just fine)
- Incomplete support for less frequently-used architectures.
- Used on x86, it still isn't as fast as a virtual machine such as VMware, unless using thekqemu accelerator.
- More difficult to install and use than comparable emulators.

#### **➤ KVM:**

##### **❖ Advantages:**

- KVM comes in-built in most of Linux distributions.
- KVM is open source and hence free to use.
- KVM ensures performance and stability.
- It has been developed by reputable organizations hence making it trustworthy to be used.

##### **❖ Dis-Advantages:**

- Available only for Linux distributions.
- The host computer needs to be powerful enough to take the load of multiple VMs.
- Using KVM has a learning curve attached to it.
- Since the hardware is centralized, hence the risk of losing data in case of system failure increases exponentially.

## ➤ Oracle Virtual Box:

### ❖ Advantages:

- Multiple OS in same computer
- No effect to host machine
- Easy to use and install
- Latest hardware support
- Trying out new OS is easy

### ❖ Dis-Advantages:

- Less efficient
- Dependent on host machine
- Effected by the host machine weakness

## 4. Applications of Virtualization Software:

Virtualization software allows multiple operating systems and applications to run on the same server at the same time, and, as a result, lowers costs and increases the efficiency of a company's existing hardware. It's a fundamental technology that powers cloud computing. Virtualization thus emulates hardware. Cloud computing is a service that results from that manipulation and is an external service. Cloud computing almost always assumes virtualization of certain resources (storage or data) that will be then delivered to the customer on demand.

## ➤ VMware:

- Browser Testing
- Software Development
- Product Demos
- Testing Multiple Operating Systems and Applications

## ➤ QEMU:

- Using QEMU allows developers to develop without the need for physical hardware, while still being able to use real-world data for testing.
- To offer a virtual machine environment, QEMU is used to imitate devices. On x86-based Linux machines, it is often used to execute Windows and DOS software.

## ➤ KVM:

- KVM are used in IT environments across a wide range of industries for benefits ranging from reducing desktop clutter to enabling remote access of network devices.

## ➤ XEN Hypervisor:

- Consolidation leading to increased utilization
- Rapid provisioning

- Dynamic fault tolerance against software failures (through rapid bootstrapping or rebooting)
- Hardware fault tolerance (through migration of a virtual machine to different hardware)
- Secure separations of virtual operating systems
- Support for legacy software as well as new OS instances on the same computer

➤ **Oracle Virtual Box:**

- Provides the solution to run different operating systems on a single machine.
- Provides community support to the user.
- Supports all types of operating systems so there are such kinds of boundaries.
- Used for the cross platform to the users.
- It is used to faster server provisioning.
- It is used for virtual labs

# **Experiment – 2**

## **XenServer & XenCenter**

❖ **Aim:** Hands on virtualization using **XenServer**.

❖ **Introduction:**

Citrix Hypervisor is an industry-leading, open-source platform for cost-effective application, desktop, cloud, and server virtual infrastructures. Citrix Hypervisor enables organizations of any size or type to consolidate and transform compute resources into virtual workloads for today's data center requirements while ensuring a seamless pathway for moving workloads to the cloud. This platform is used by virtualization administrators to deploy, host and manage VMs. It's also used to distribute hardware resources – CPU, memory, networking, storage to VMs.

❖ **Theory:**

➤ **XenServer:**

The main architectural components of Citrix XenServer are the 64-bit hypervisor platform, XenCenter integrated management, and XenMotion live migration. Key features of Citrix XenServer aim to ease virtualization infrastructure management. VM templates are a significant aspect of this. For example, you can create VM templates from snapshots. Another key feature of this platform is XenMotion, which allows you to live to migrate VMs between hosts. With the Enterprise version of XenServer 7.1, you can also live patch hosts with no downtime. Centralized support for Open vSwitch is another important feature of the XenServer platform.

The XenServer management console, which is a Windows-based client, sets this platform apart from other Xen stacks because it gives you the ability to manage VMs running on multiple hosts in a resource pool. Critics of XenServer argue that its ecosystem isn't as established as those of its major competitors, but the

fact that there is a 3 3 Cloud Computing Lab open-source version of this hypervisor makes it a good choice for smaller IT shops. Another drawback of Citrix XenServer is that it lacks built-in automation capabilities. To automate tasks, you have to use Power Shell scripts. In addition, compared to its competitors, XenServer doesn't have the same level of integration or available certifications.

#### ➤ **XenCenter:**

XenCenter is the management GUI for XenServer that runs on Windows. This is the front end that lets users configure new virtual machines and enables full virtual machine installation, configuration, administration, and lifecycle management. Using this tool, we configure the remote storage and manage networks, including VLANs and internal networks, and bonded and dedicated NICs. XenCenter displays the VMs' performance stats and lets us take snapshots of guest machines and provide access to the virtual machines' consoles. These are just some basic functions of XenCenter.

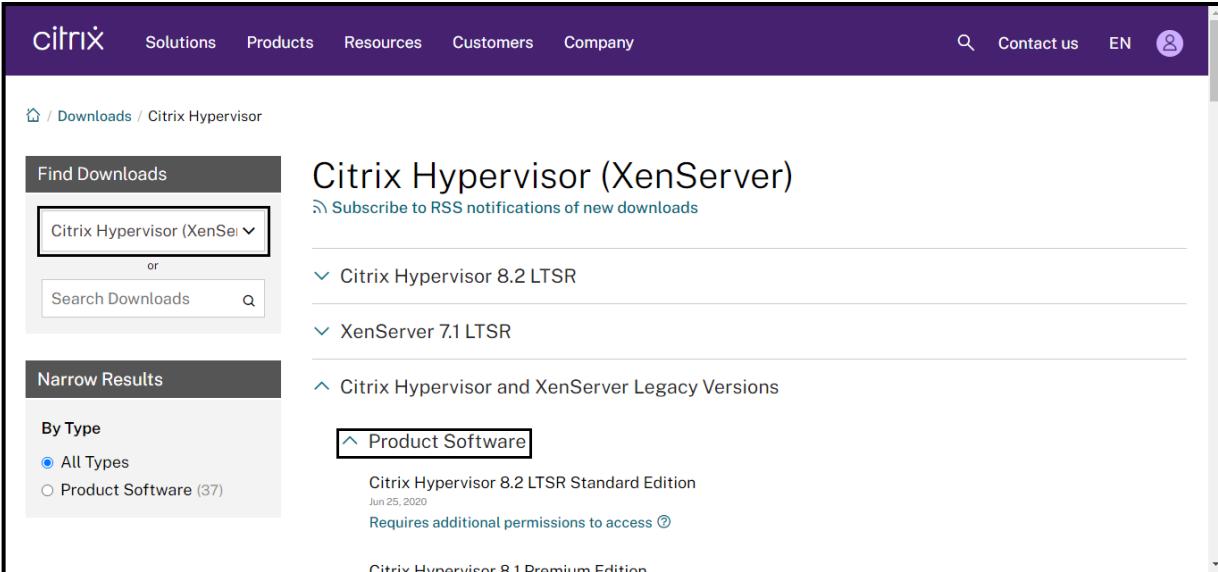
#### ❖ **Procedure:**

Install XenCenter & XenServer on the Virtual Machine and operate it from XenCenter.

#### ❖ **Instructions:**

##### ➤ **Installing XenServer:**

**Step-1:** Go to the official website of Citrix (<https://www.citrix.com/en-in/downloads/citrixhypervisor/product-software/hypervisor-80-express-edition.html>) for downloading the XenServer and Sign In with your credentials. Click on the Dialog box and select Citrix Hypervisor 8.0.0 Base Installation ISO – Express Edition. It will lead to the file we want to download. Agree to the terms and condition and download the file.

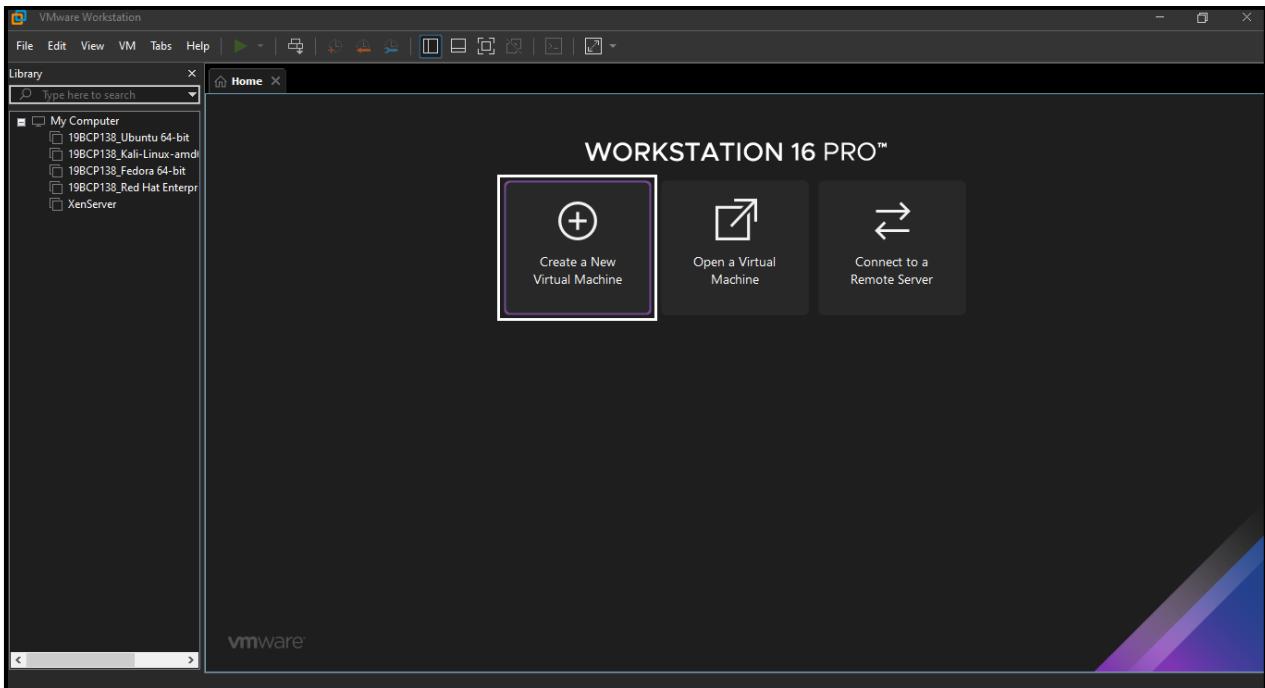


The screenshot shows the Citrix website's download section for Citrix Hypervisor (XenServer). The top navigation bar includes links for Solutions, Products, Resources, Customers, Company, a search icon, Contact us, EN, and a user profile icon. The main content area shows a "Find Downloads" sidebar with a dropdown menu set to "Citrix Hypervisor (XenSe" and a search bar. Below this is a "Narrow Results" sidebar with a "By Type" section where "All Types" is selected. The main content area displays a list of download options under "Citrix Hypervisor (XenServer)". The first item is "Citrix Hypervisor 8.2 LTSR", followed by "XenServer 7.1 LTSR", and then "Citrix Hypervisor and XenServer Legacy Versions". Under "Citrix Hypervisor and XenServer Legacy Versions", there is a "Product Software" section with a link to "Citrix Hypervisor 8.2 LTSR Standard Edition" (Jun 25, 2020), which requires additional permissions to access.

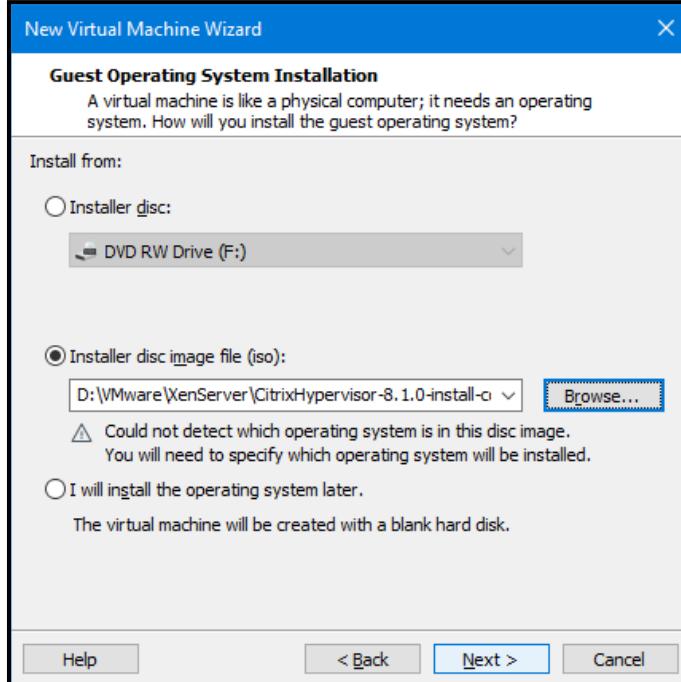


The screenshot shows the specific download page for the "Citrix Hypervisor 8.0.0 Base Installation ISO-Express Edition". The page header is "Citrix Hypervisor 8.0.0 Base Installation ISO-Express Edition". It shows the download date as "Apr 25, 2019" and the file size as "581MB (.iso)". A prominent blue "Download File" button is centered. Below the button, the "Checksums" section lists the SHA-256 hash: "SHA-256-63349fae8111eddfa586b706183257ba7d64f5ad326dfd75a47073b2d6d50bc5".

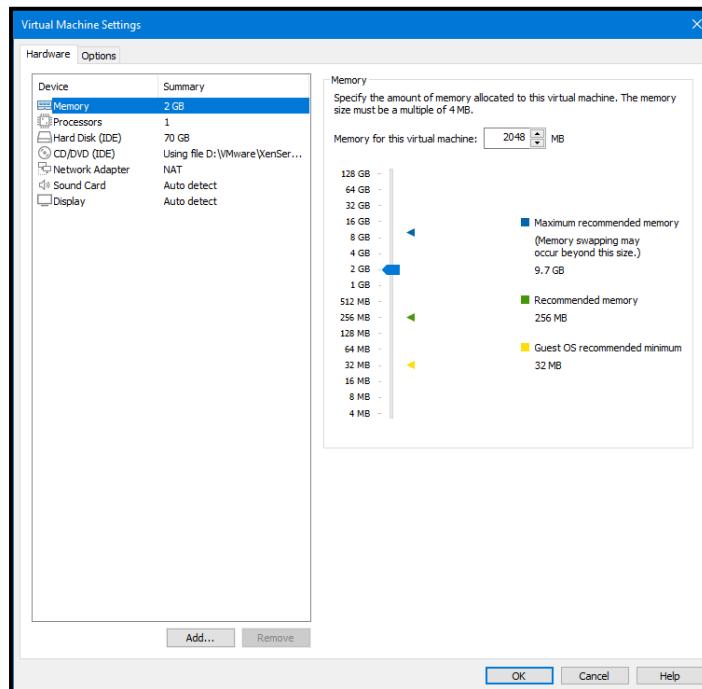
**Step-2:** Open VMware and Create New Virtual Machine using the ISO File you downloaded in Step 1.



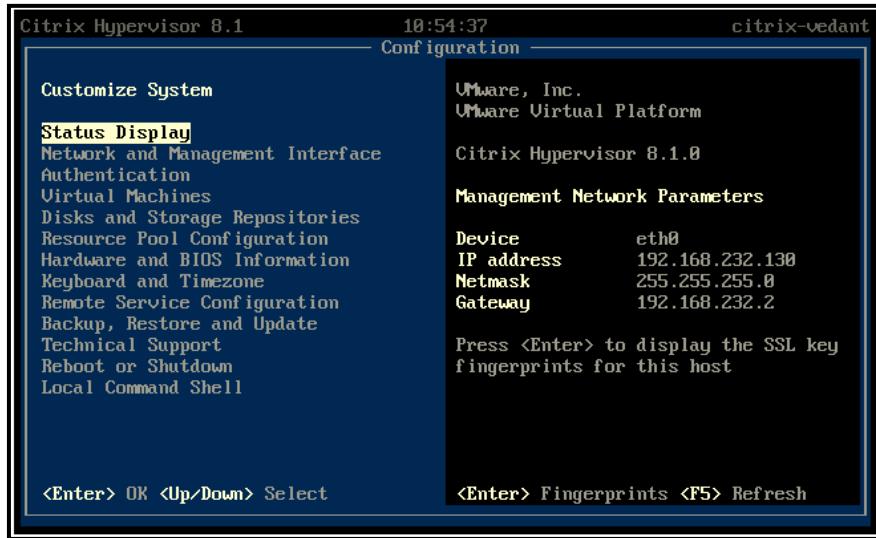
Guest Operating System should be kept other and Version as other too. You can choose any location to install it and also specify the disk size (Preferred is 4GB) and disk split type.



In the last step click on customize hardware and specify the main memory to be minimum 2048 MB. Then Click on finish and Run the VM.



**Step-3:** You will be prompted for creation of Login Credentials. Do it. When installation part is completed, you will see following success prompt. Click OK and restart the machine. After Restarting machine, it will install some more files and can take time to finish the process. After the completion, you will see below screen as a success prompt.



### ➤ Installing XenCenter:

**Step-1:** Go to the official website of Citrix (<https://www.citrix.com/en-in/downloads/citrixhypervisor/product-software/hypervisor-80-express-edition.html>) for downloading the XenCenter and Sign In with your credentials. Click on the Dialog box and select Hypervisor 8.0.0 Windows Management Console. It will lead to the file we want to download. Agree to the terms and condition and download the file.

The screenshot shows the Citrix Downloads page for Citrix Hypervisor (XenServer). The top navigation bar includes links for Solutions, Products, Resources, Customers, Company, and a search bar. The main content area is titled "Citrix Hypervisor (XenServer)" and features a "Find Downloads" section with a dropdown menu set to "Citrix Hypervisor (XenServer)". Below this is a "Narrow Results" section with a "By Type" filter where "All Types" is selected. The main list shows three categories: "Citrix Hypervisor 8.2 LTSR", "XenServer 7.1 LTSR", and "Citrix Hypervisor and XenServer Legacy Versions". Under "Citrix Hypervisor 8.2 LTSR", there is a link to "Citrix Hypervisor 8.2 LTSR Standard Edition" from "Jun 25, 2020". At the bottom of the page, there is a note about "Requires additional permissions to access".

## XenCenter 8.0.0 Windows Management Console

Apr 25, 2019

41MB -windows-(.msi)

[Download File](#)

XenCenter allows you to manage your Citrix Hypervisor environment and deploy, manage, and monitor Virtual Machines from a Windows desktop.

### Checksums

SHA-256 -8379dc49b7f772380a76e253a2ef06e718b3c2ed6a708ea26edca31afacb9e41

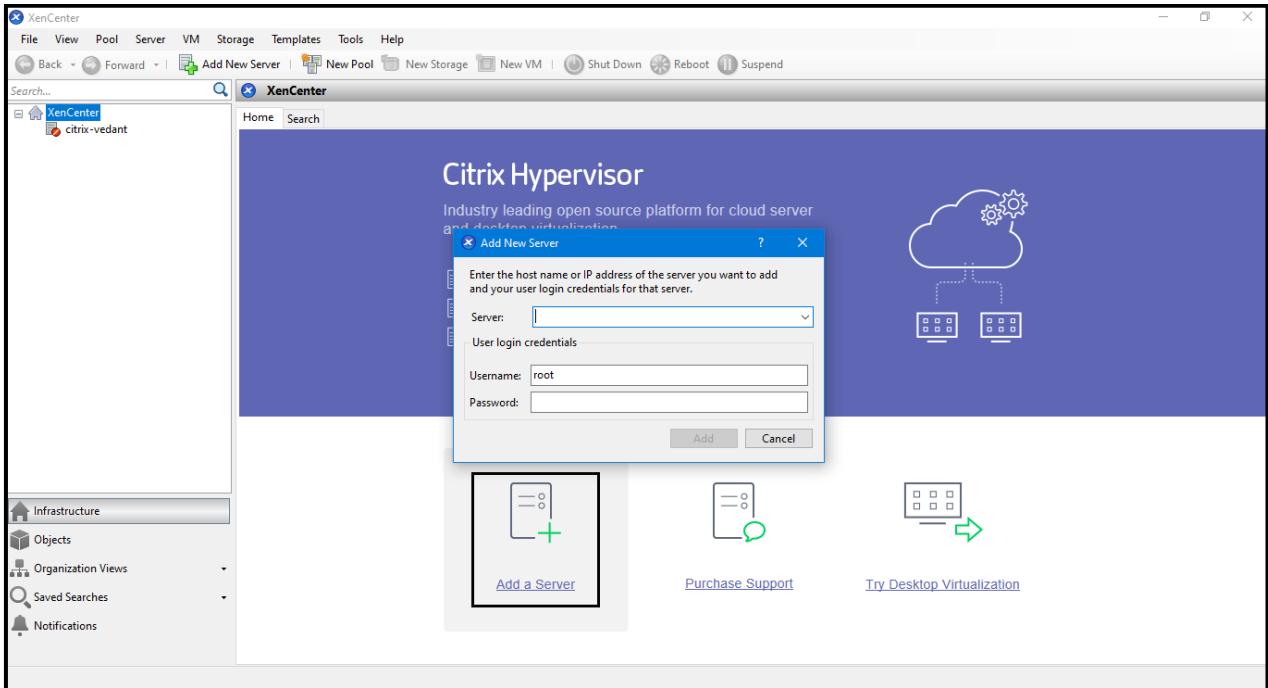
**Step-2:** Run the MSI file you just downloaded. (You need to follow normal Windows Installation Procedure).



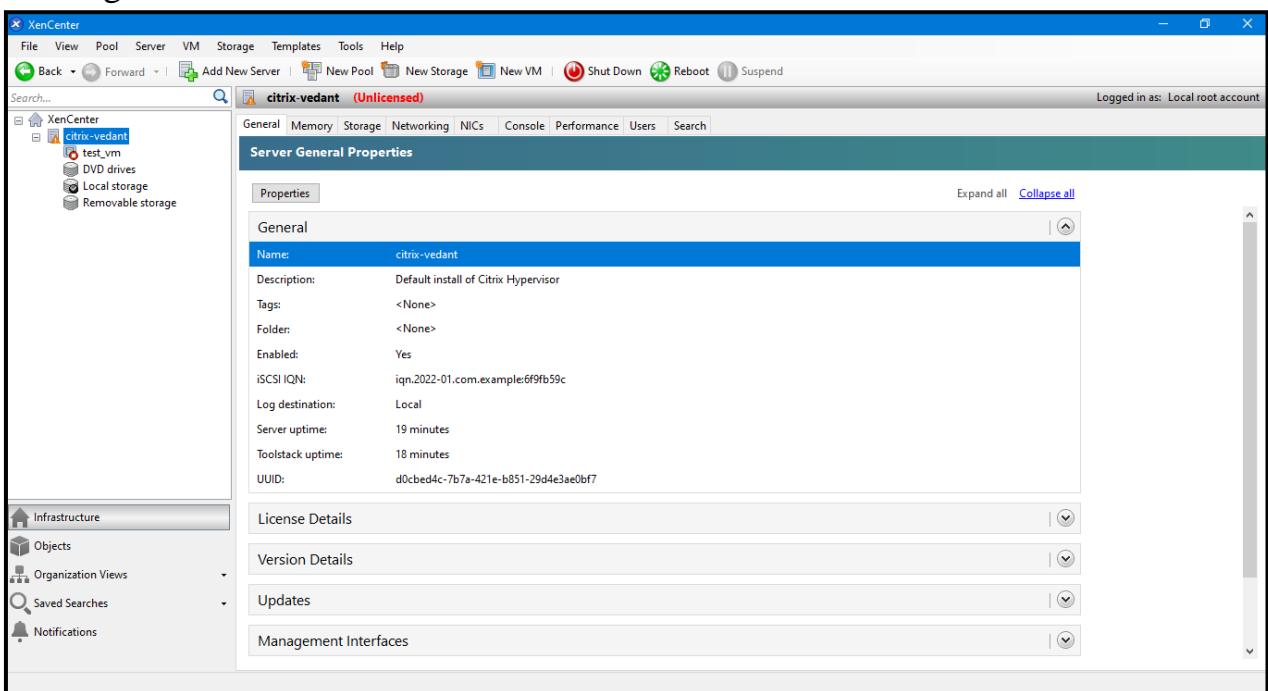
**Step-3:** Run the installed XenCenter. Click on Add Server and give the credential you have used while installing XenServer and you will get server after running XenServer. First Start the XenServer from VMware.



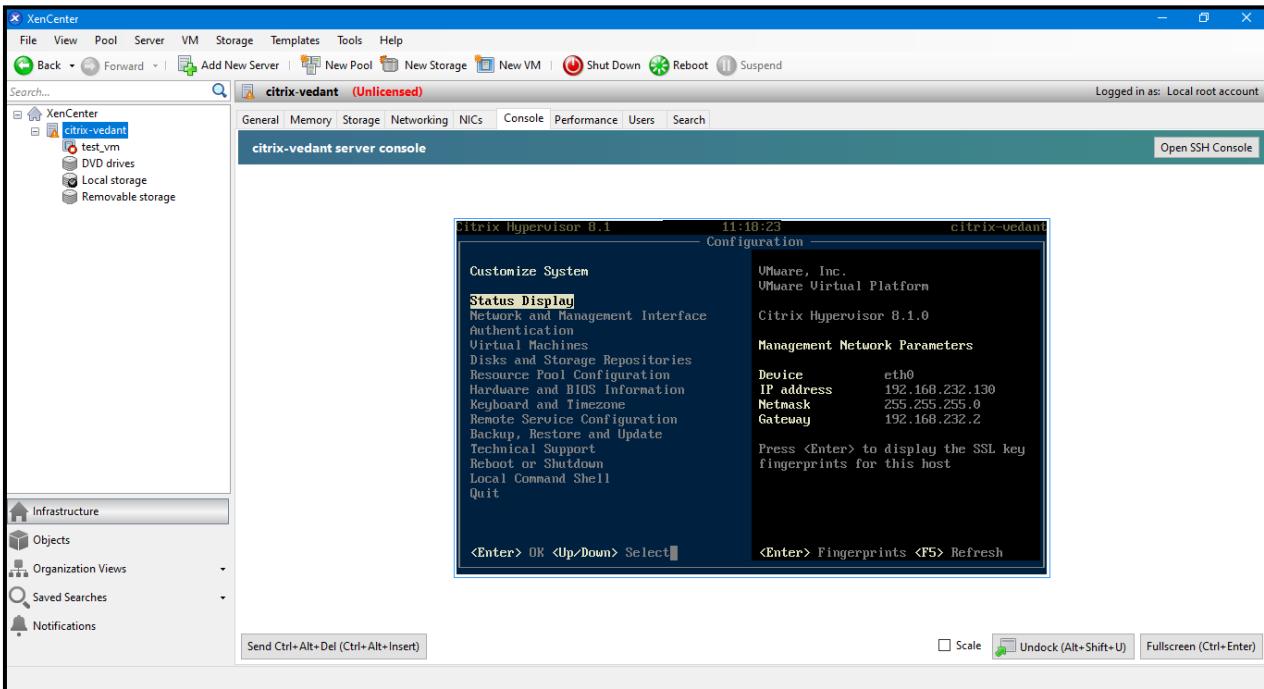
Then Add Server in XenCenter by writing the IP address, Username and Password.



**Step-4:** After adding the server you can see the details of the server and can manage the XenServer from XenCenter and can take a look over to the server.



Can manage the XenServer running in VMware from XenCenter.



## ❖ Observation and Learning:

XenServer is pretty sleek and detail-oriented software and is also used in the world's largest cloud services and enterprises. It was easy to install in a few basic steps. It's found that this platform (XenServer) is used by virtualization administrators to deploy, host and manage VMs. It's also used to distribute hardware resources – CPU, memory, networking, storage to VMs. Likewise; XenCenter is the management GUI for XenServers that run on Windows. This is the front end that lets users configure new virtual machines and enables full virtual machine installation, configuration, administration, and lifecycle management. It displays VMware's performance stats, lets us take snapshots of guest machines, and provides access to the virtual machines' consoles.

## ❖ Conclusion:

We learned to download, install and operate XenServer in the VMware. We also learned to download, install and operate XenCenter software and to manage the XenServers from the XenCenter.

## **❖ Reference:**

Website to Download the XenServer .iso file and XenCenter .msi file:

- [Citrix Hypervisor 8.0 Express Edition - Citrix India](#)

Theory on XenServer and XenCenter:

- [Install XenServer and use XenCenter for configuration \(techtarget.com\)](#)

Installation Process and Theory:

PDEU Cloud Computing Lab Lectures

# Experiment - 3

## To Install and Configure VMWare Workstation Pro

❖ **Aim:** Exploring VMWare Workstation Pro to create the Virtual Machines.

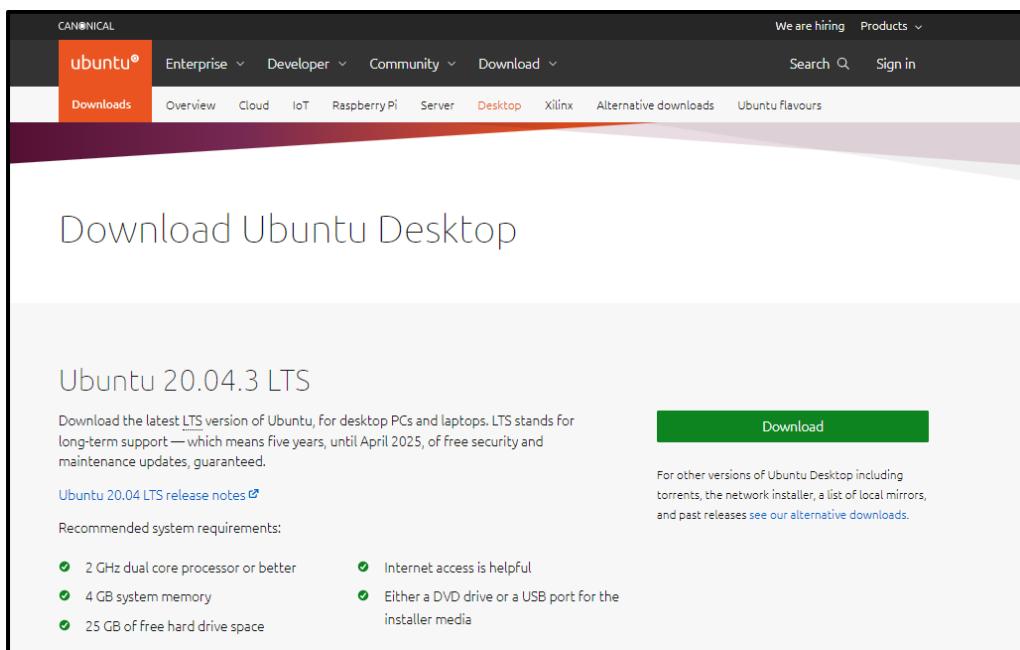
➤ Perform the following:

5. Create three or more Virtual Machine and Assign resources.
6. Install two or more Guest Operating System on all the VMs.
7. Run simple applications or programs on all the VMs.

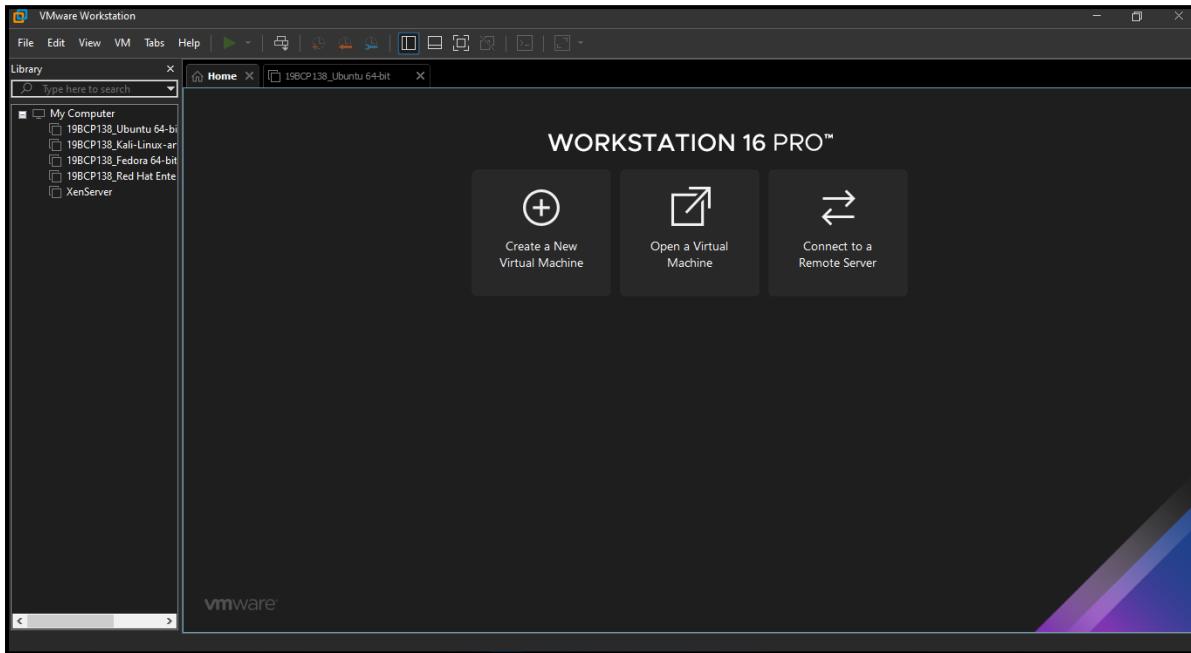
### 1. VM1: Ubuntu Virtual Machine

Steps to create Ubuntu Virtual Machine:

1. Download .iso file Ubuntu from the website.



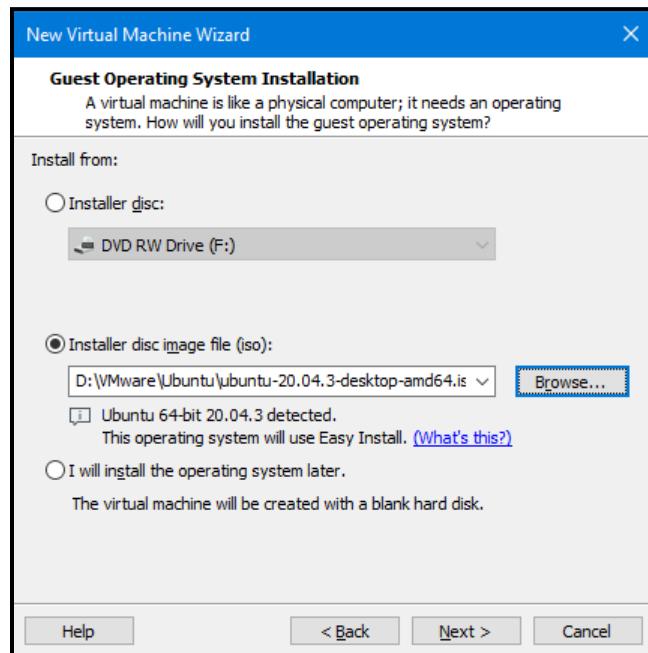
2. Open VMWare Workstation and Click on Create a New Virtual Machine.



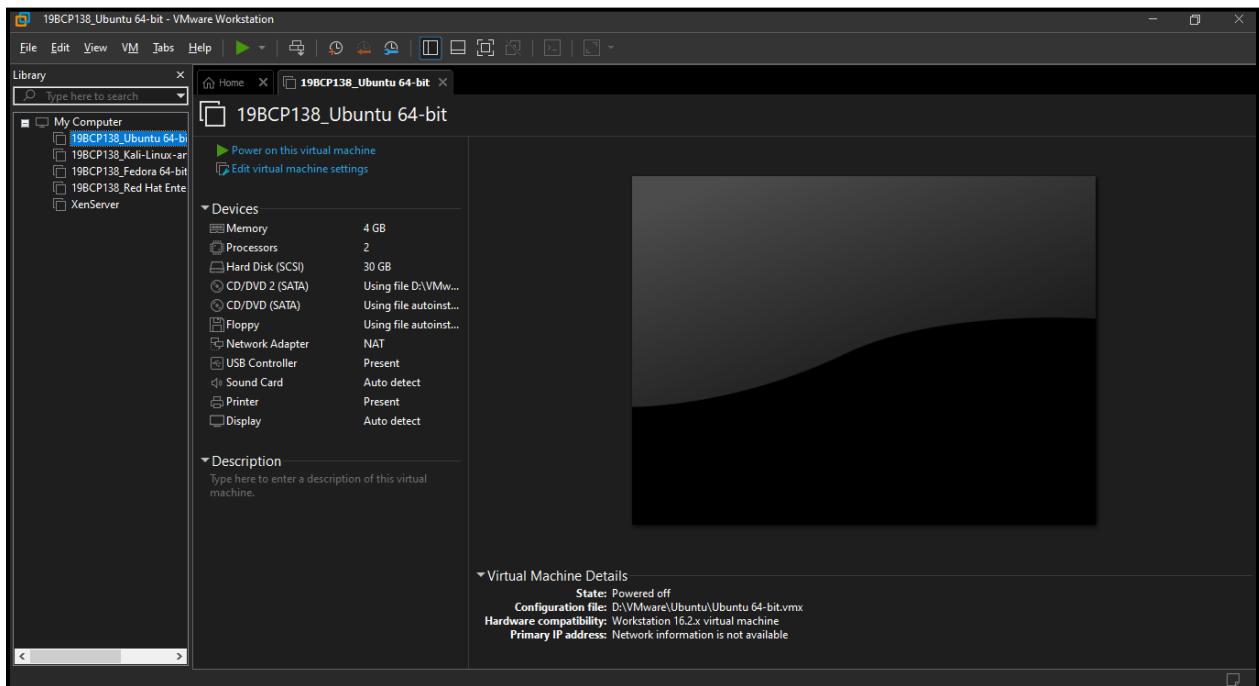
3. Select Typical and Click Next.



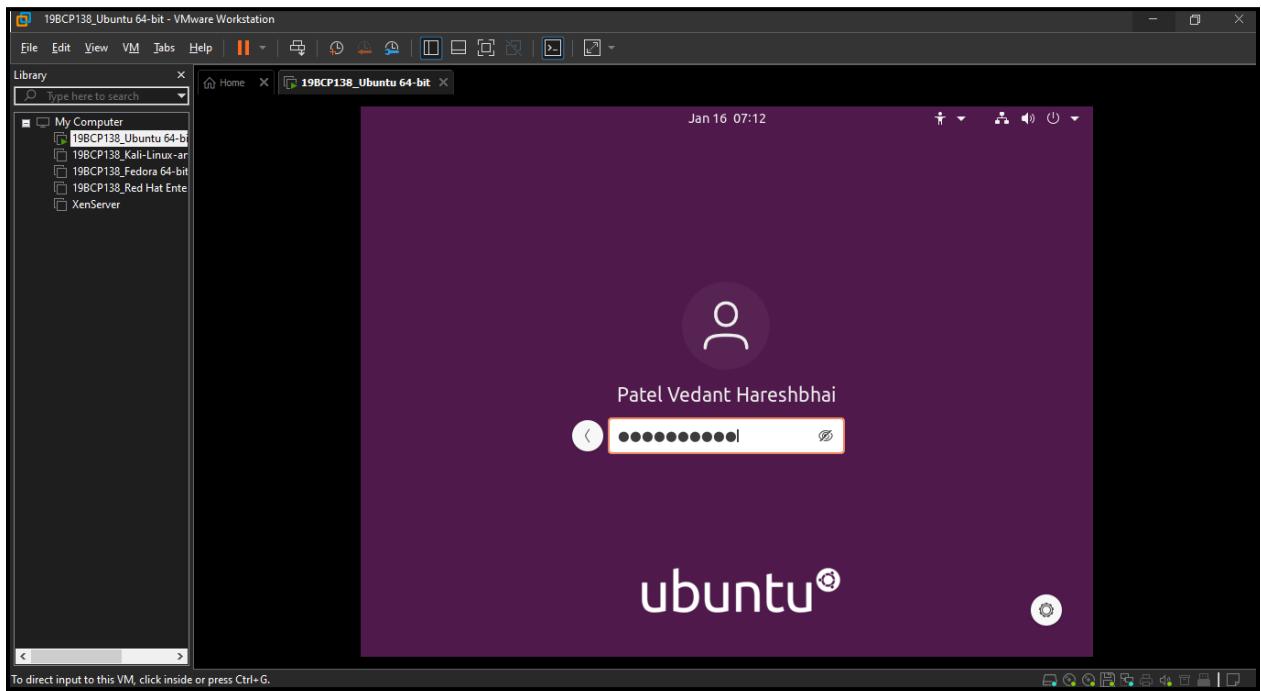
4. Select the .iso file of Ubuntu Downloaded and Click Next.



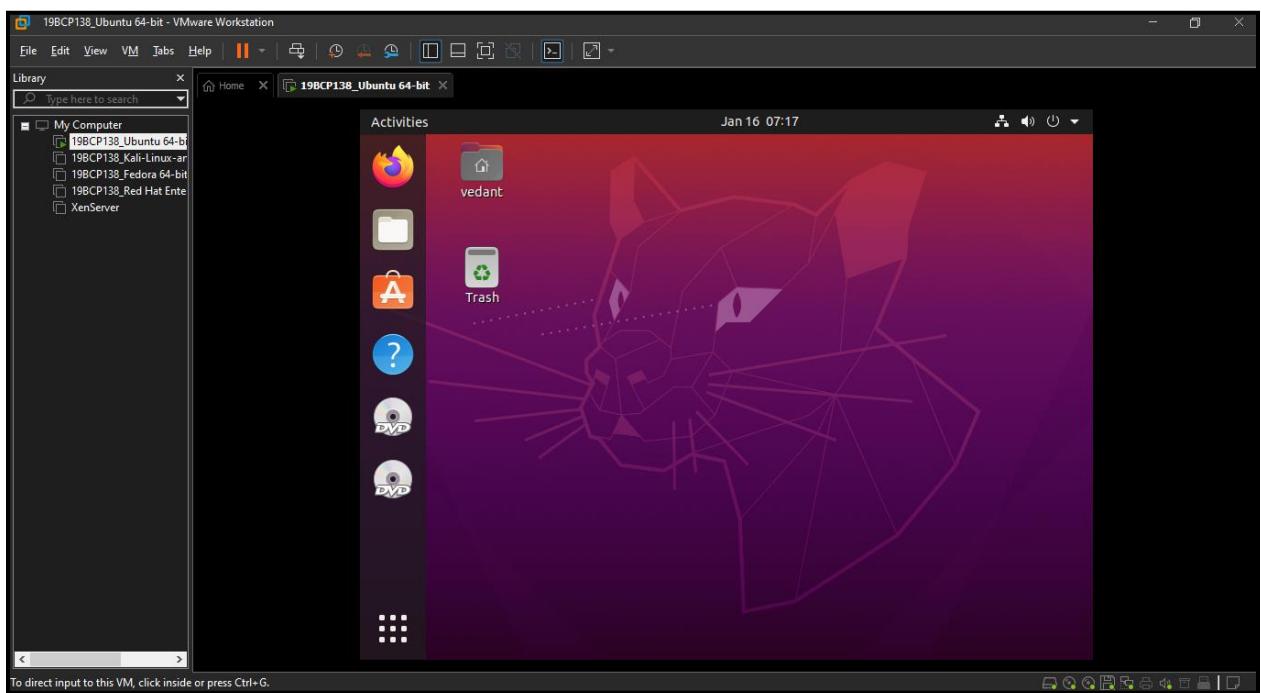
5. Assign the Recommended Memory and Requirement. Then Power On the Virtual Machine.



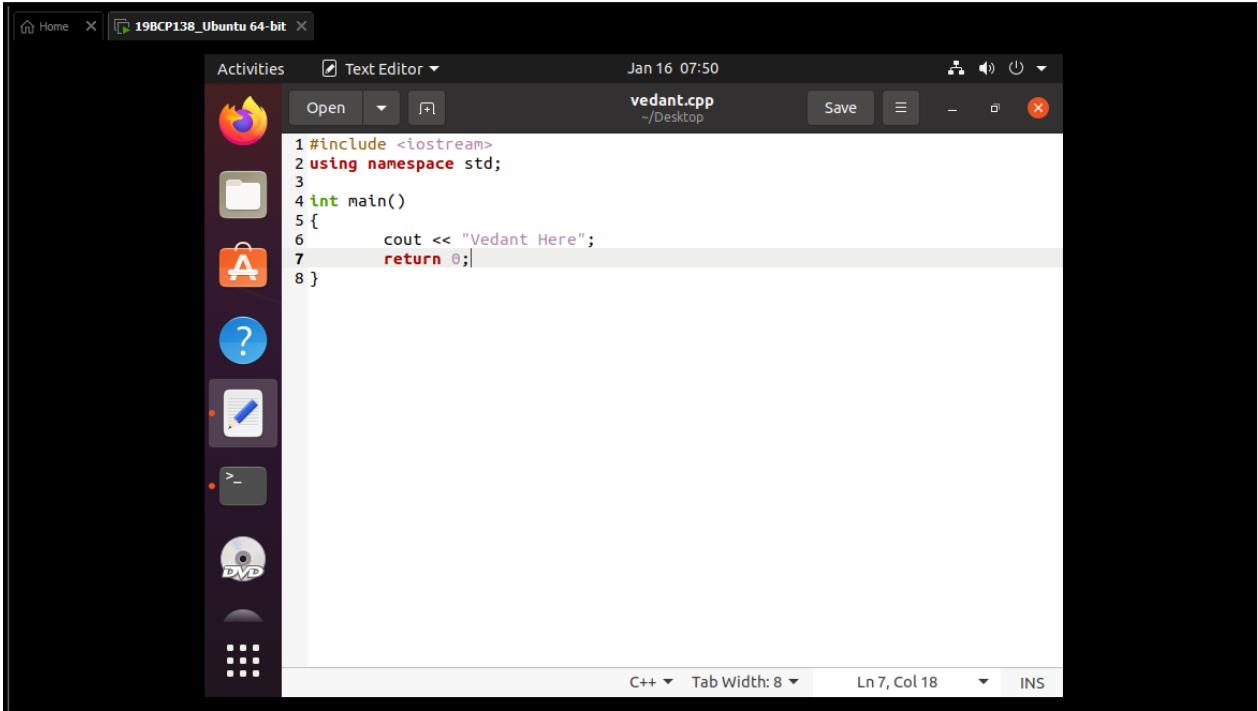
6. Enter the Password and then Ubuntu will start in VMware.



7. Ubuntu desktop will open as seen.

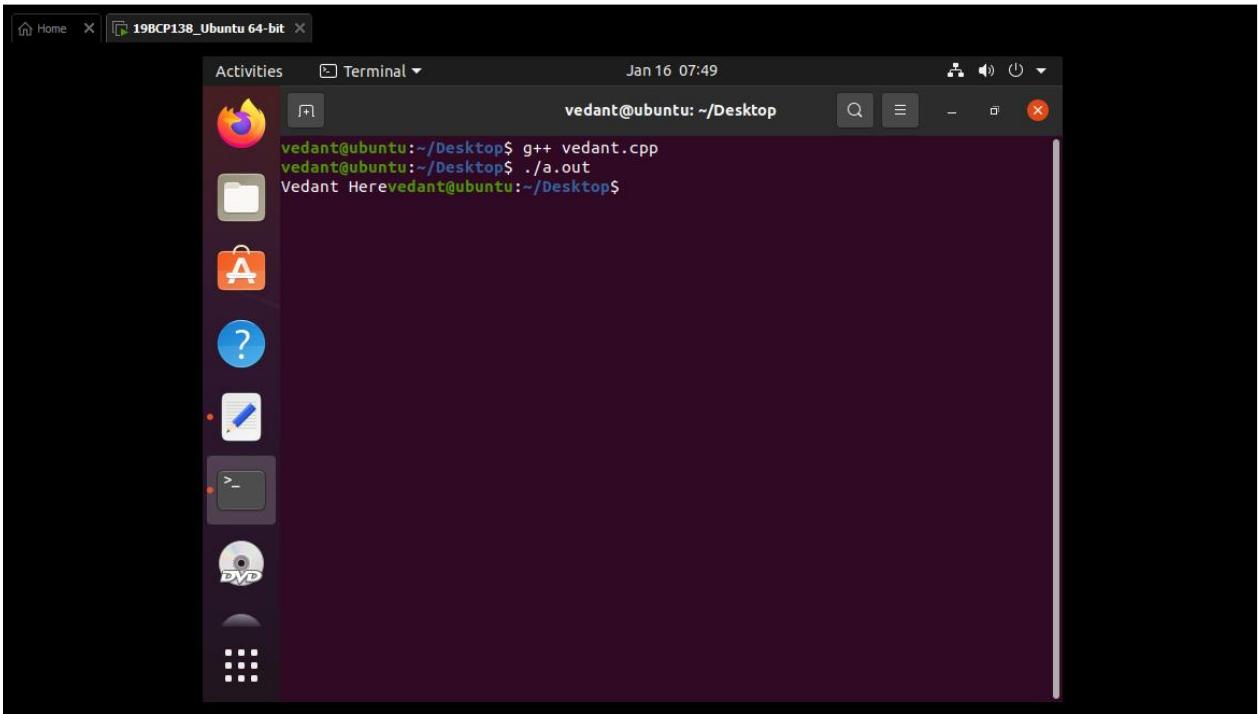


8. Opening Text Editor and write a code.



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Vedant Here";
7     return 0;
8 }
```

9. Executing the code.

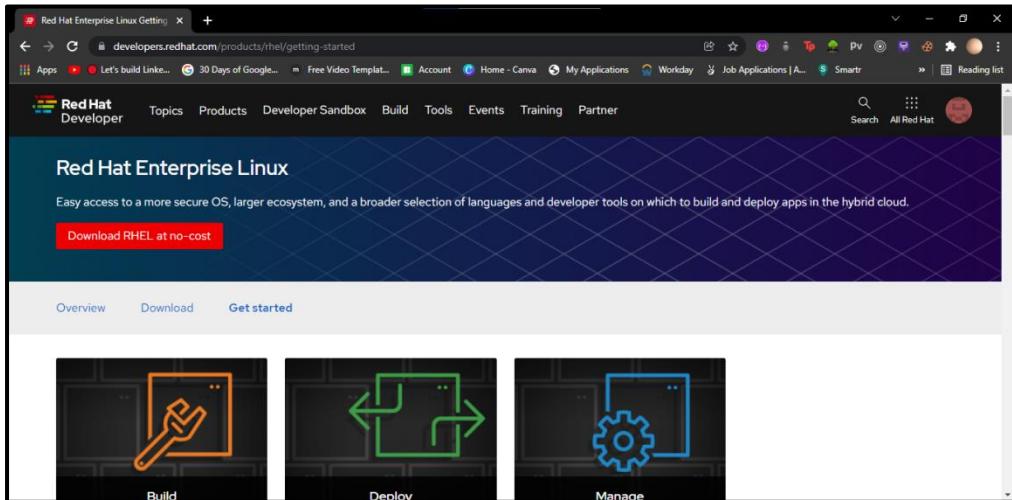


```
vedant@ubuntu:~/Desktop$ g++ vedant.cpp
vedant@ubuntu:~/Desktop$ ./a.out
Vedant Herevedant@ubuntu:~/Desktop$
```

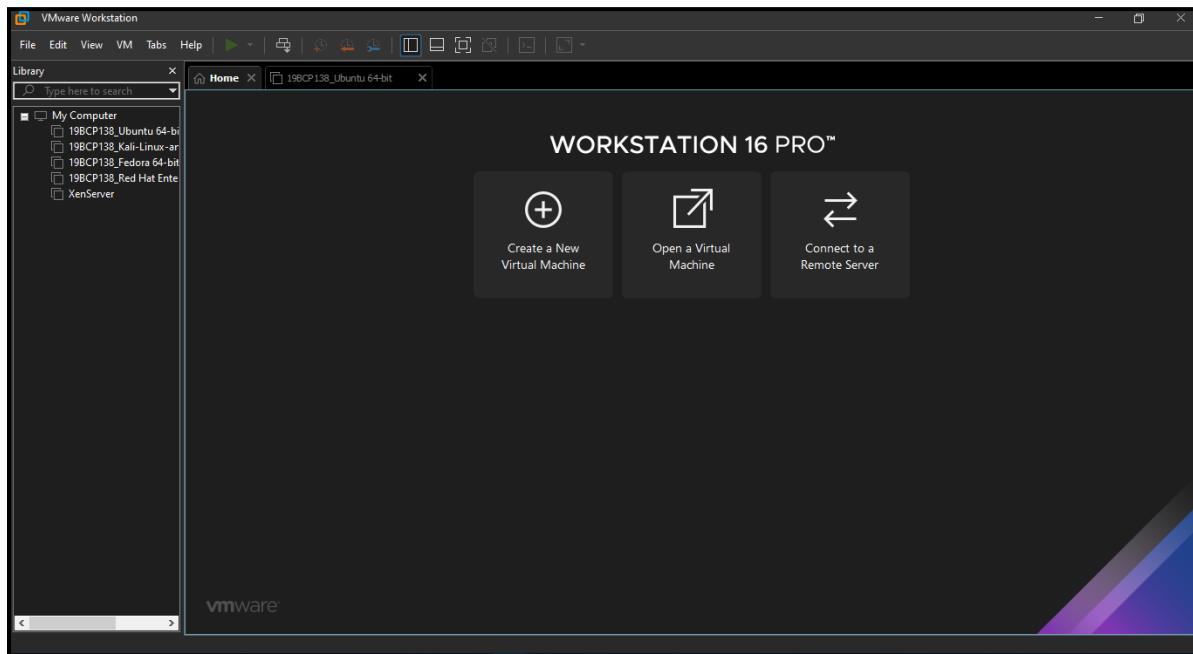
## 2. VM2: Red Hat Linux 8 Virtual Machine

Steps to create Red Hat Linux 8 Virtual Machine:

1. Download the .iso file of Red Hat but first need to do Login.



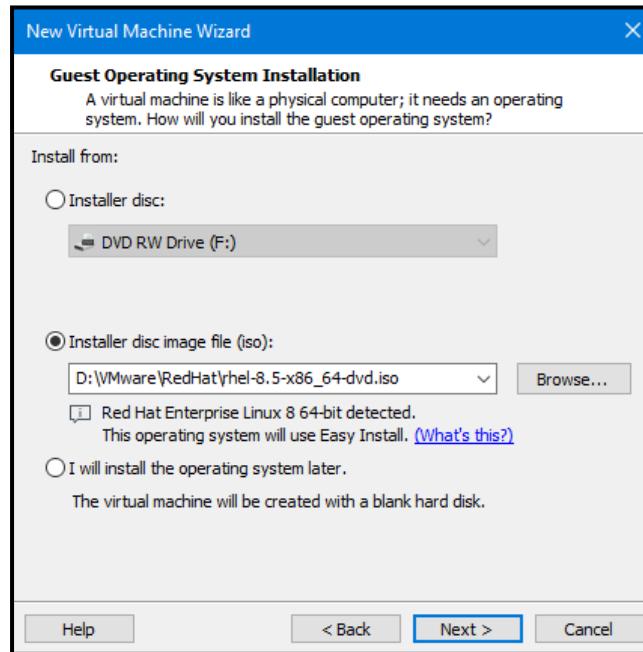
2. Open VMware Workstation and Click on Create a New Virtual Machine.



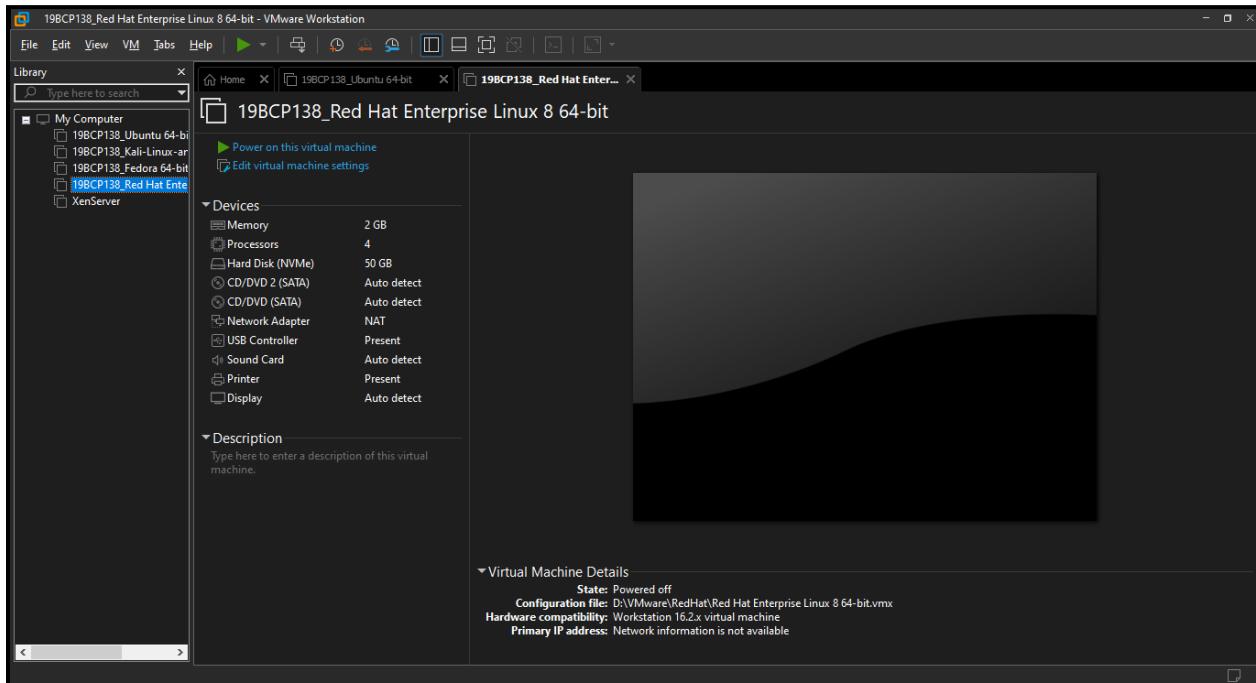
3. Select Typical and Click Next.



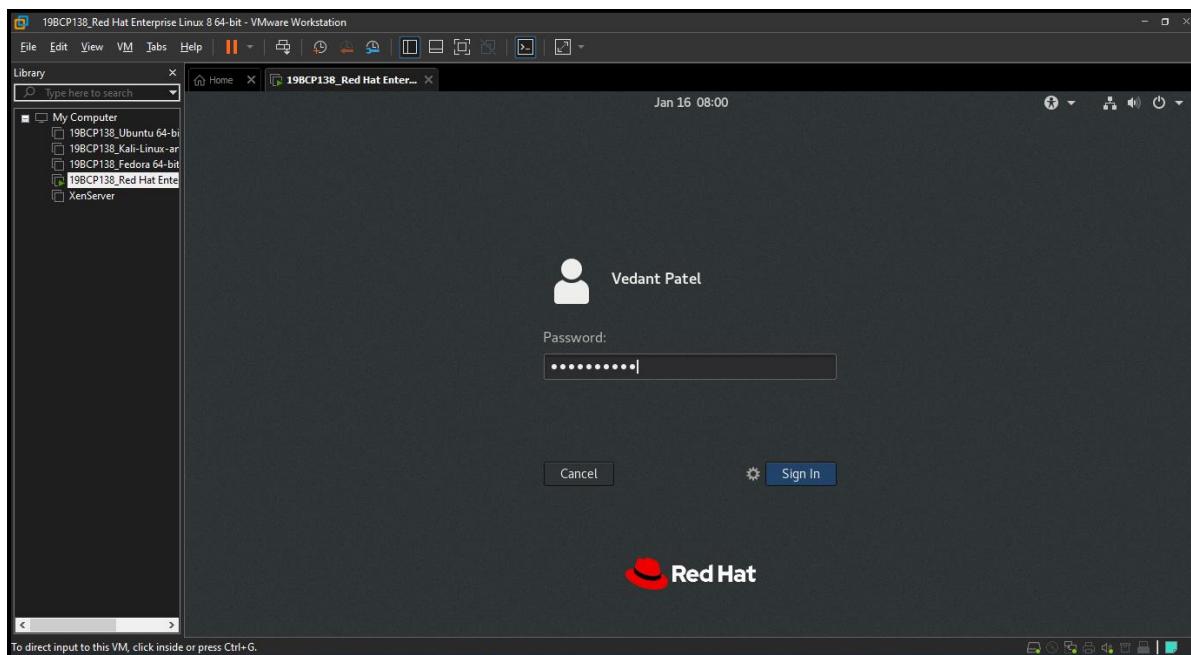
4. Select the .iso file of Red Hat Downloaded and Click Next.



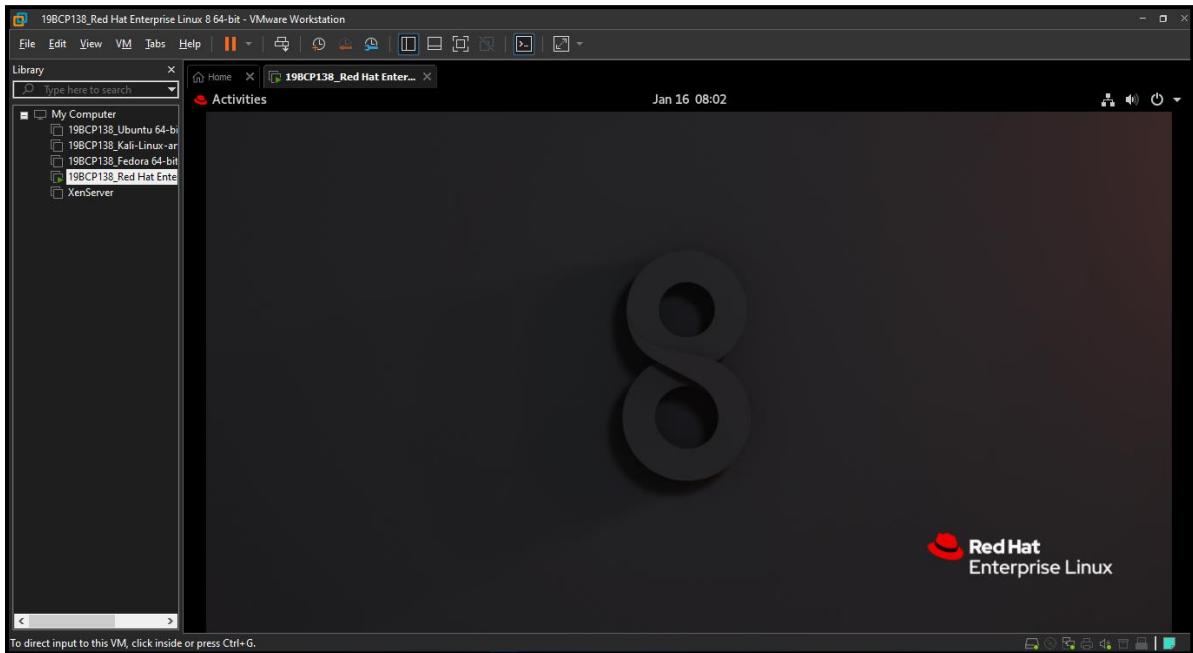
## 5. Assign the Recommended Memory and Requirement. Then Power On the Virtual Machine.



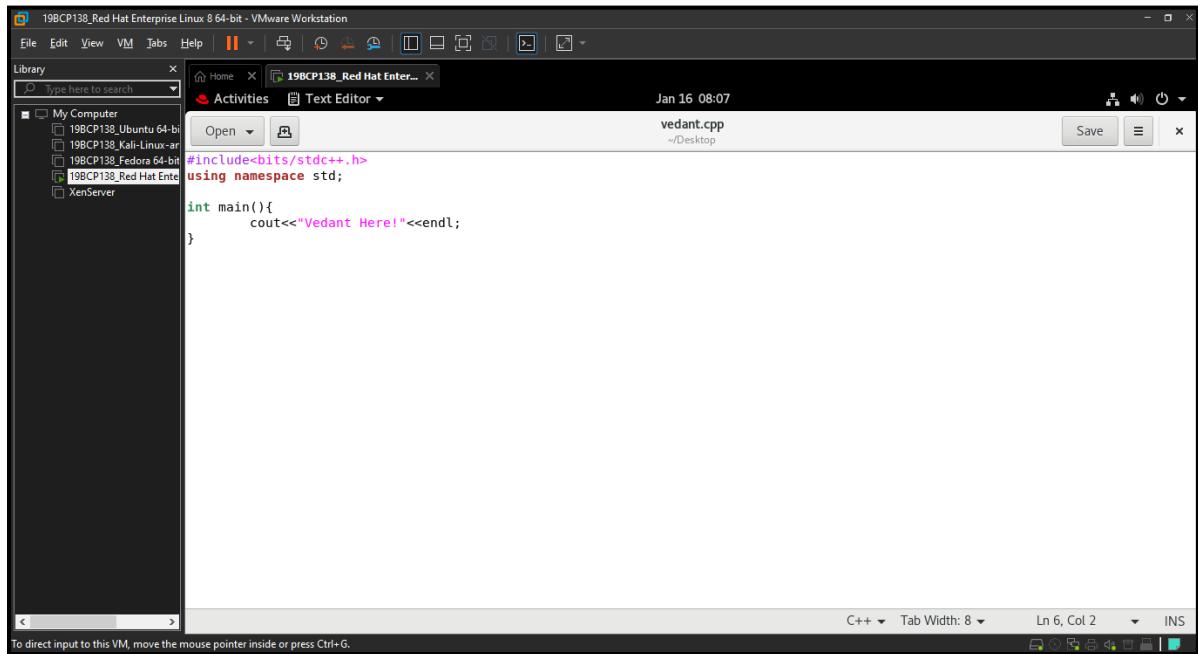
## 6. Enter the Password and then Red Hat will start in VMware.



7. Red Hat desktop will open as seen.



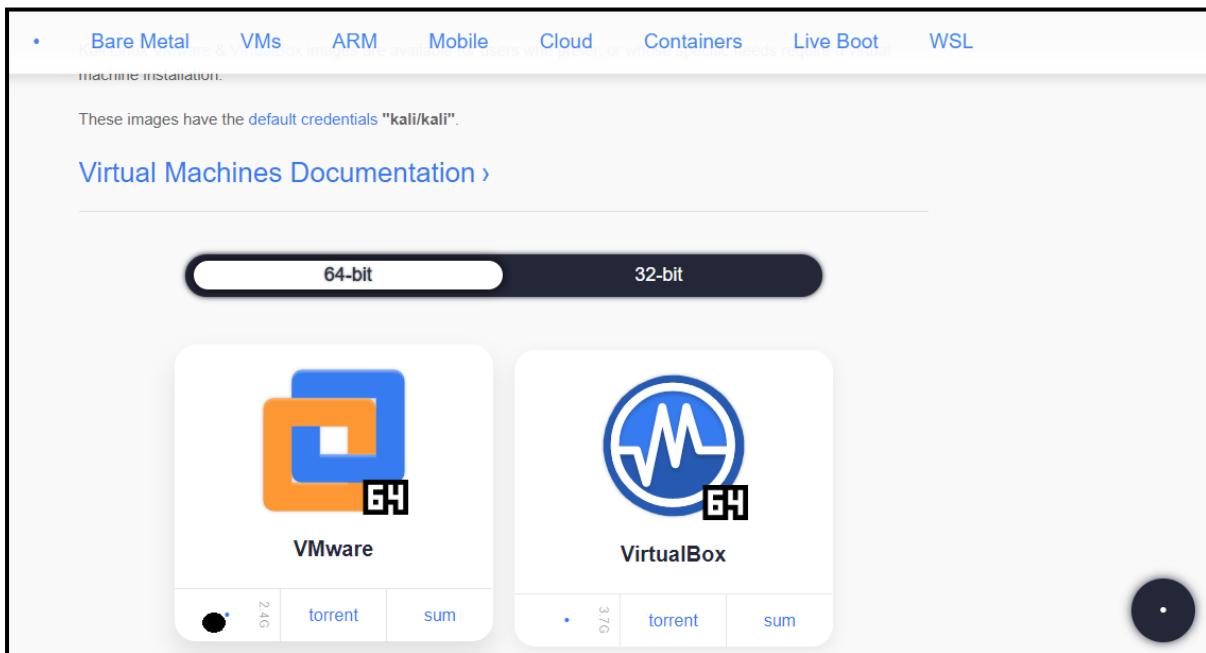
8. Opening Text Editor and write a code.



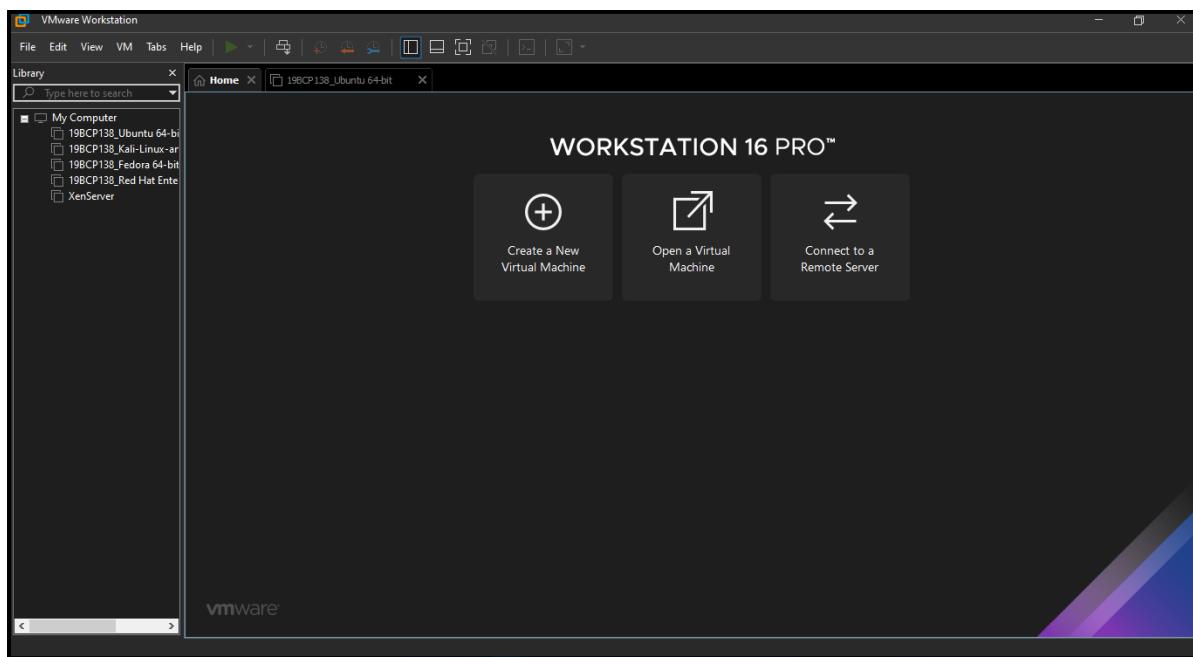
### 3. VM3: Kali Linux Virtual Machine

Steps to create Kali Linux Virtual Machine:

1. Download .iso file of Kali Linux from the website (Click where the black spot is drawn.)



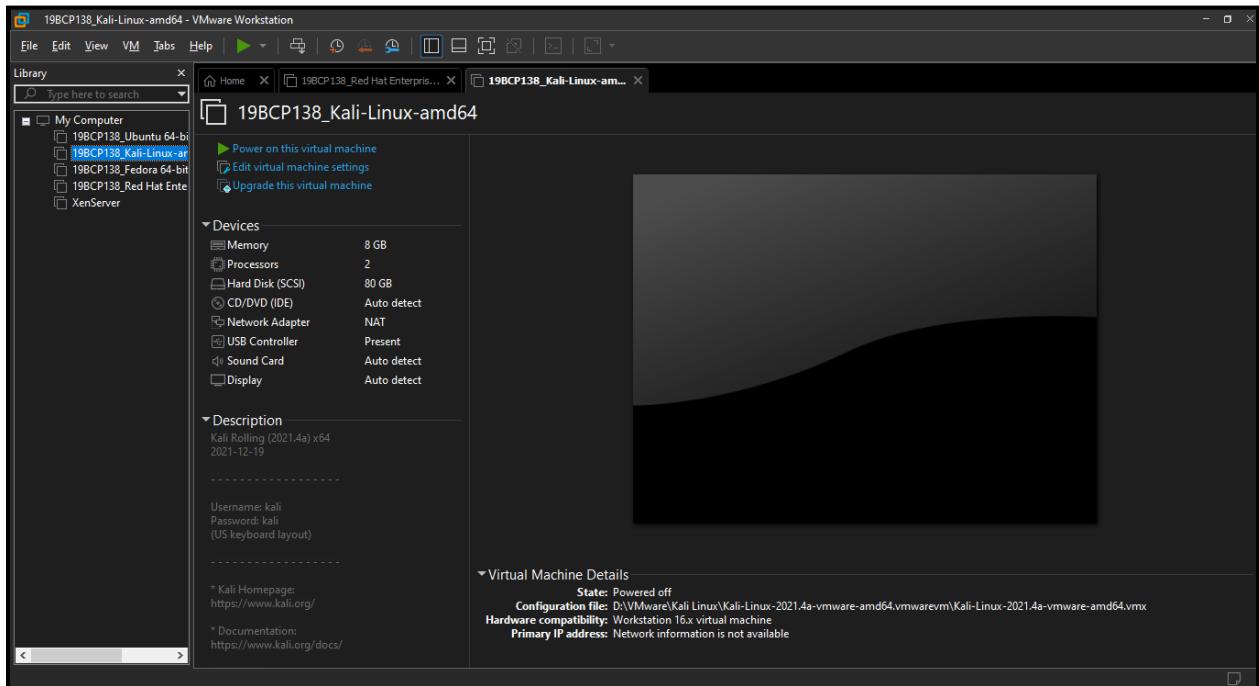
2. Open VMware Workstation and Click on Create a New Virtual Machine.



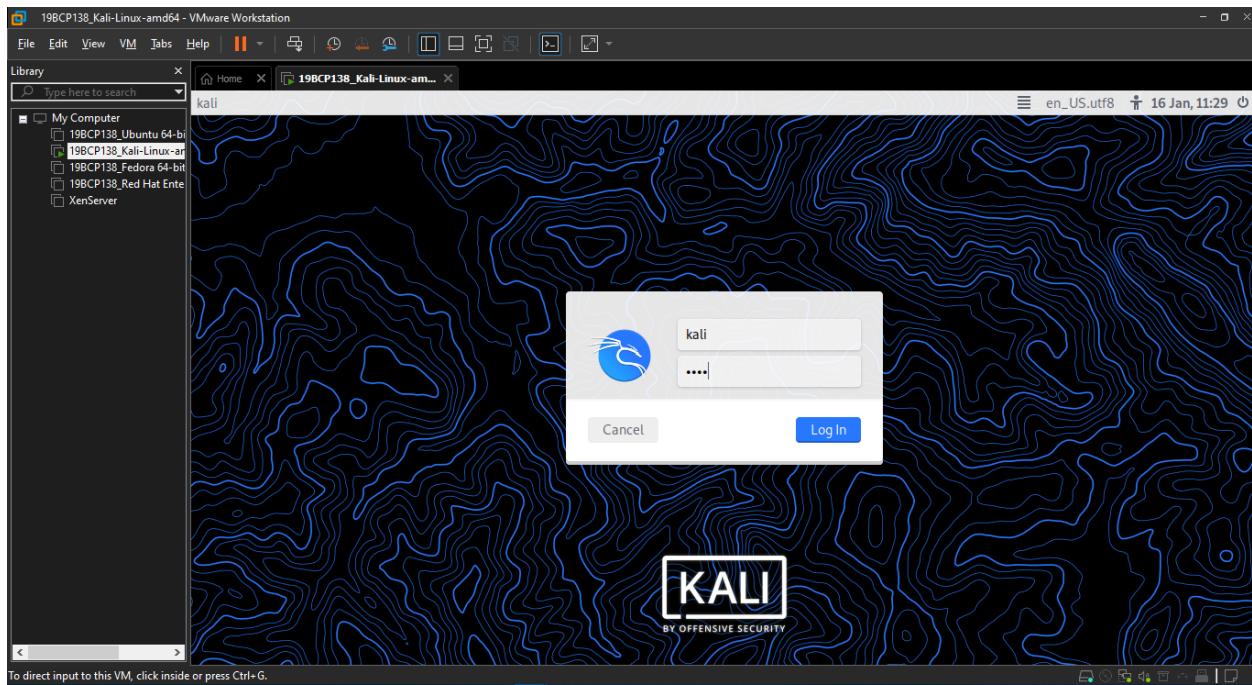
3. Select Typical and Click Next.



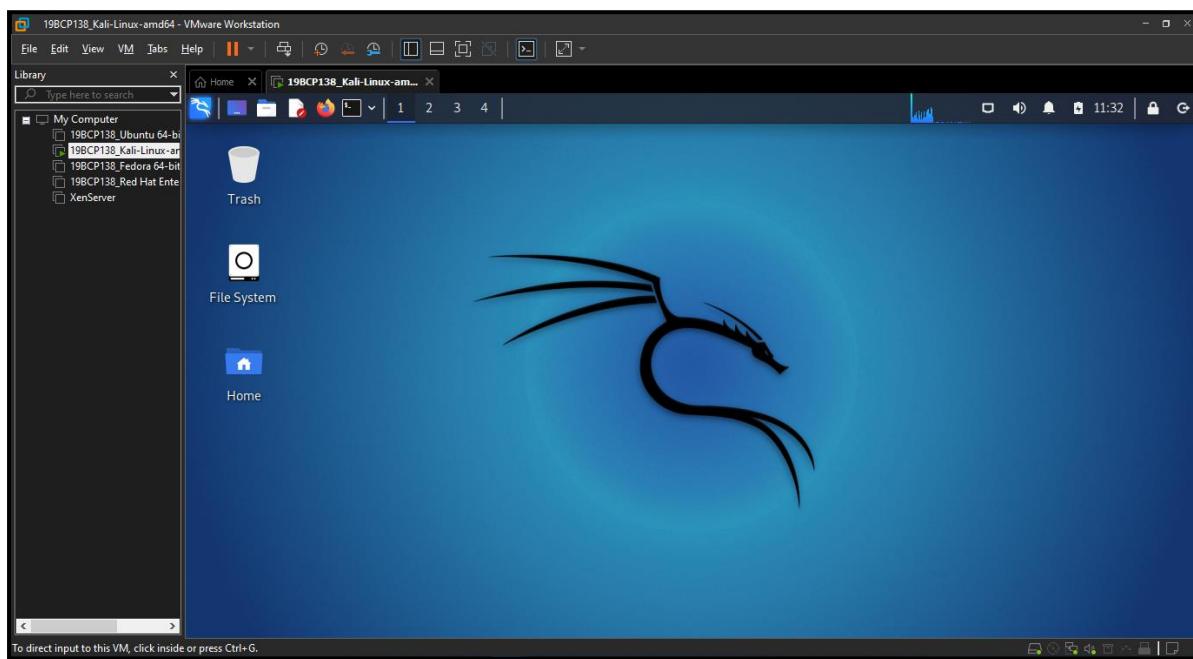
4. Assign the Recommended Memory and Requirement. Then Power On the Virtual Machine.



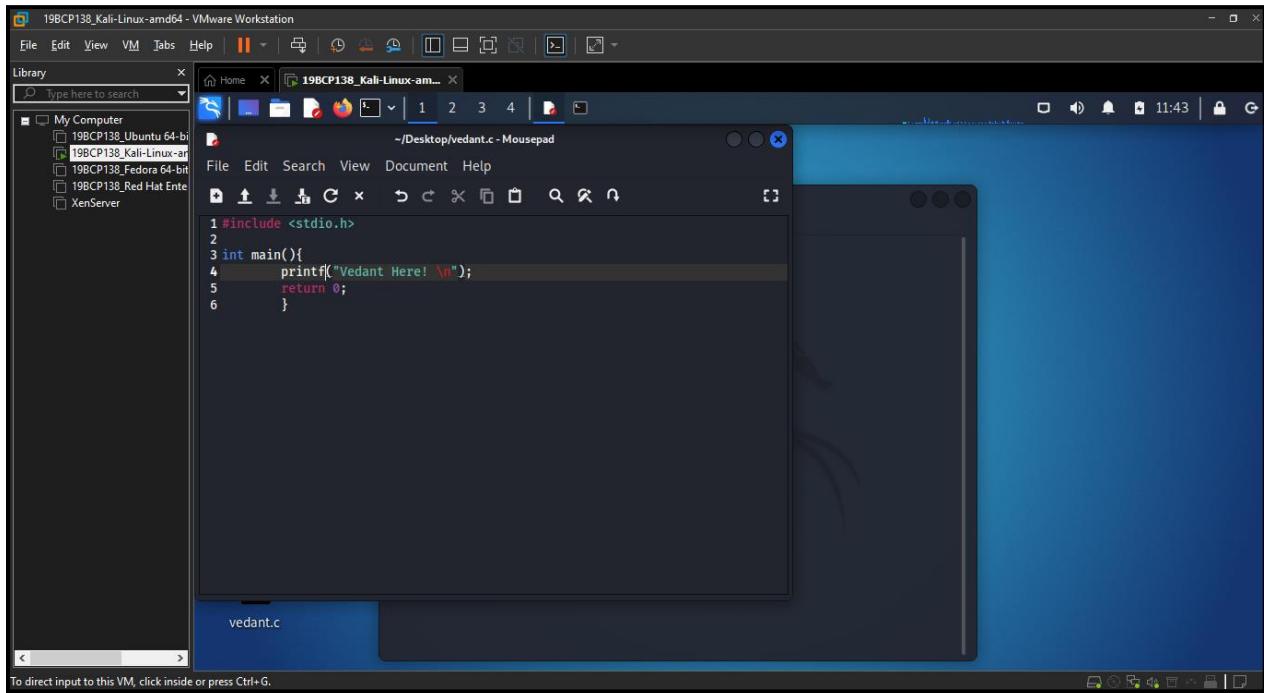
5. Enter the Username & Password and then Kali Linux will start in VMware.



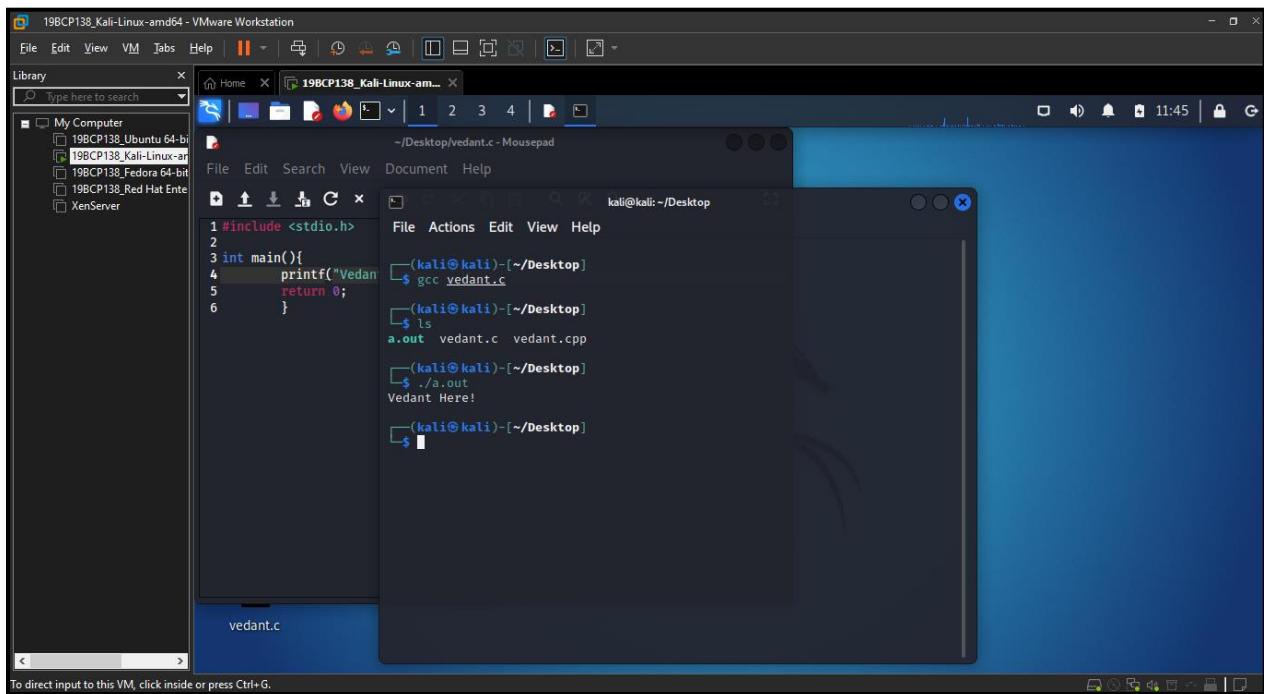
6. Kali Linux desktop will open as seen.



## 7. Opening Text Editor and write a code.



## 8. Executing the code.



## Experiment - 4

### To Install and Configure VMware Workstation Pro for creating the Virtual Machines, Cloning and Deleting VMs.

❖ **Aim:** Exploring VMware Workstation Pro to create the Virtual Machines.

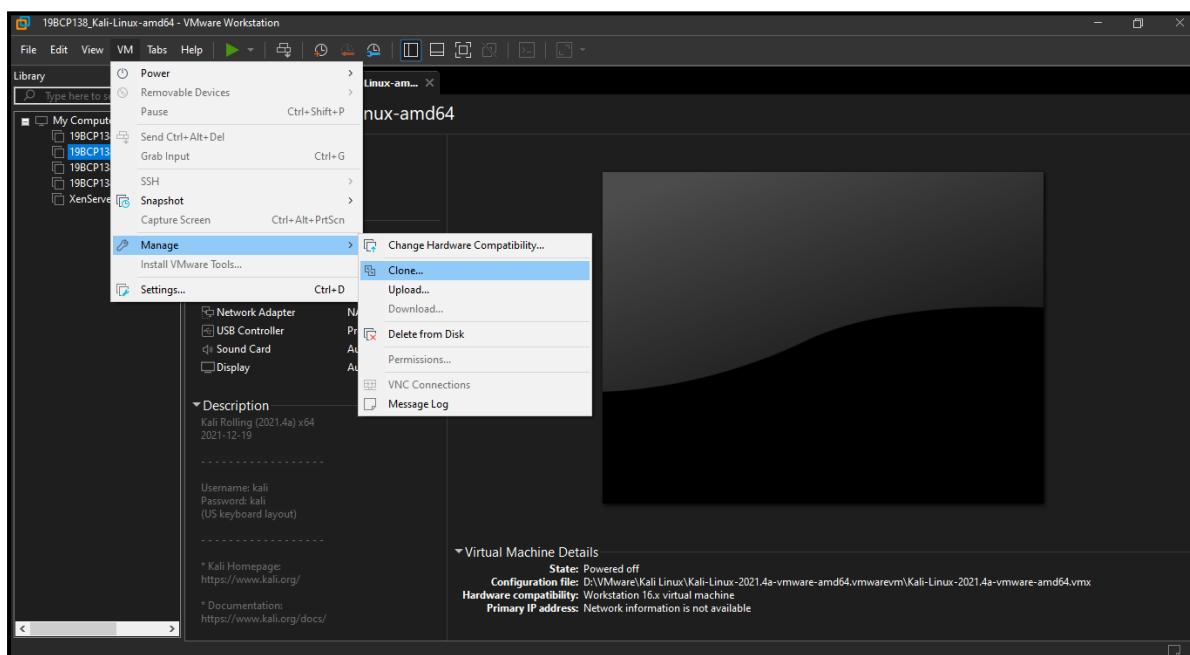
➤ Perform the following:

1. Cloning of Virtual Machines: Full Cloning, Linked Cloning
2. Add a USB Controller to Virtual Machine
3. Closing Virtual Machines & Exiting Workstation Pro
  - a. Hard and Soft Power Off
  - b. Configure Virtual Machine to Run in the Background
  - c. Pause and Unpause a Virtual Machine
  - d. Pause all Virtual Machine without interacting Workstation Pro
  - e. Suspend and Resume Virtual Machines
4. Delete a Virtual Machine from the disk

#### 1. Cloning of Virtual Machines: Full Cloning, Linked Cloning

➤ Steps to do Cloning of Virtual Machine: **Linked Cloning**

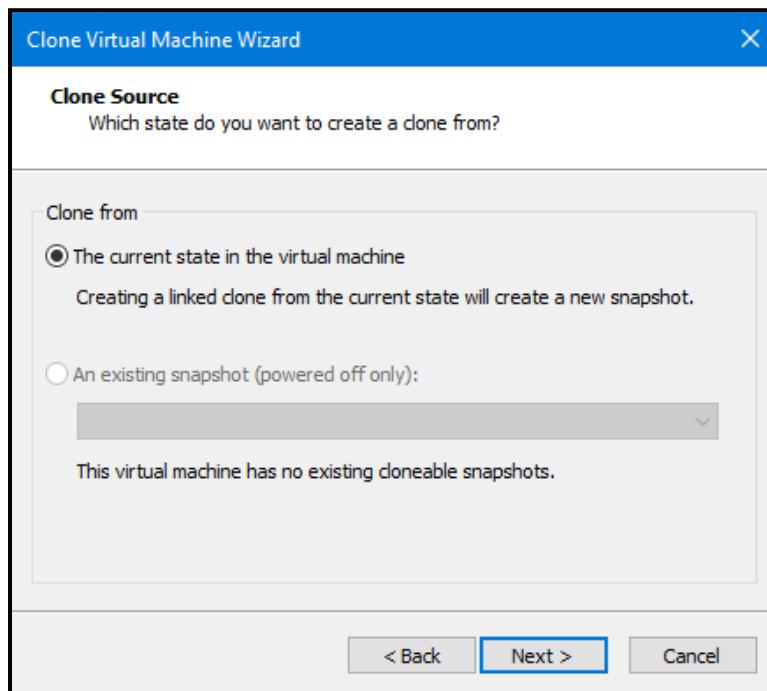
1. Select VM → Manage → Clone.



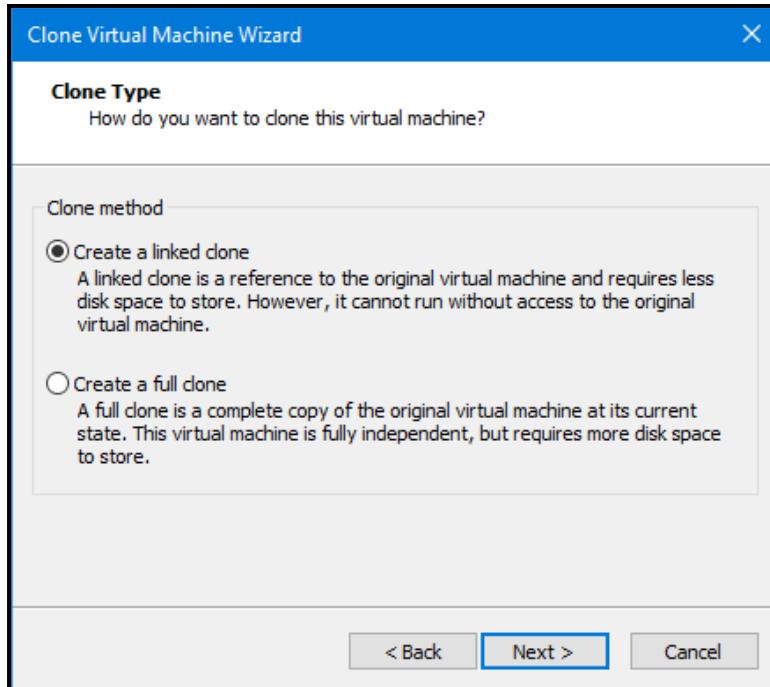
2. Select Next.



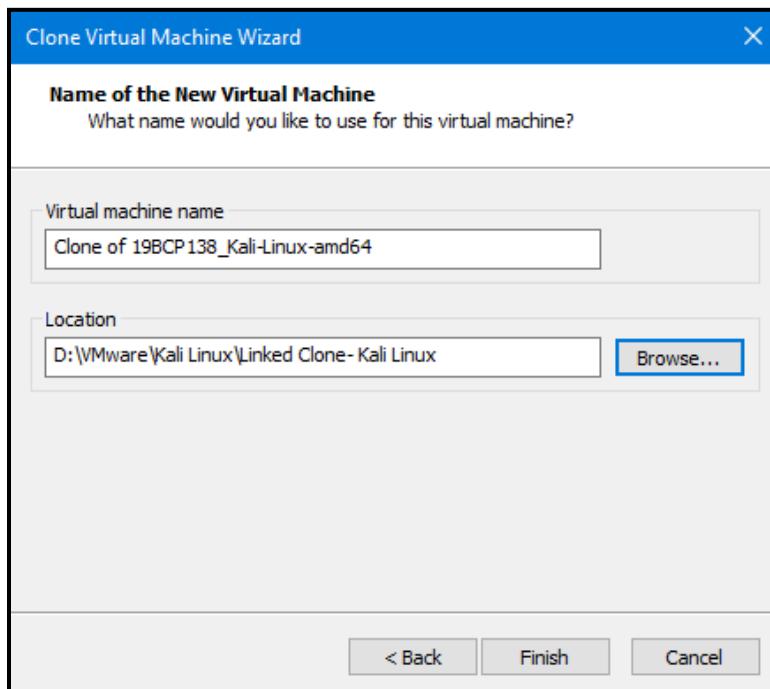
3. Select (The current state in the virtual machine) and Select Next.



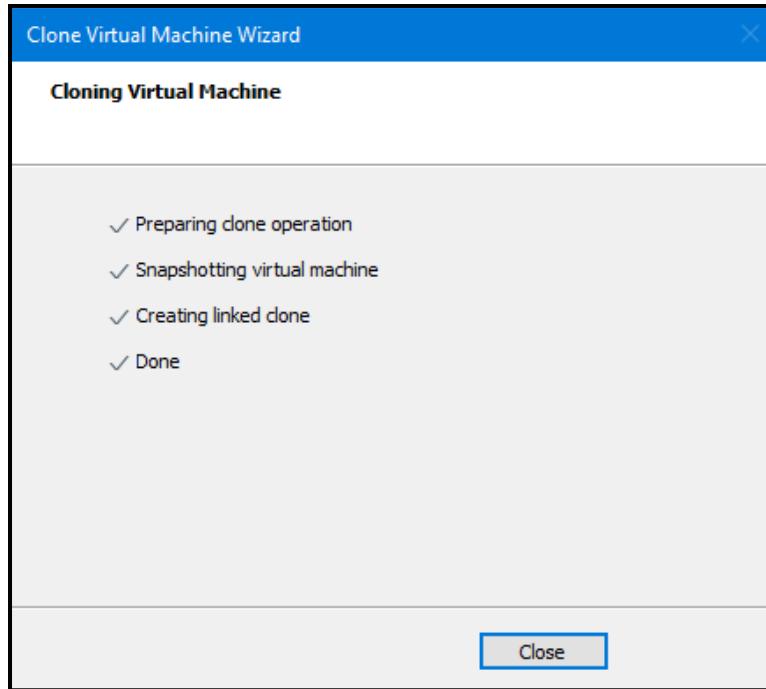
4. Select Linked Clone and go Next.



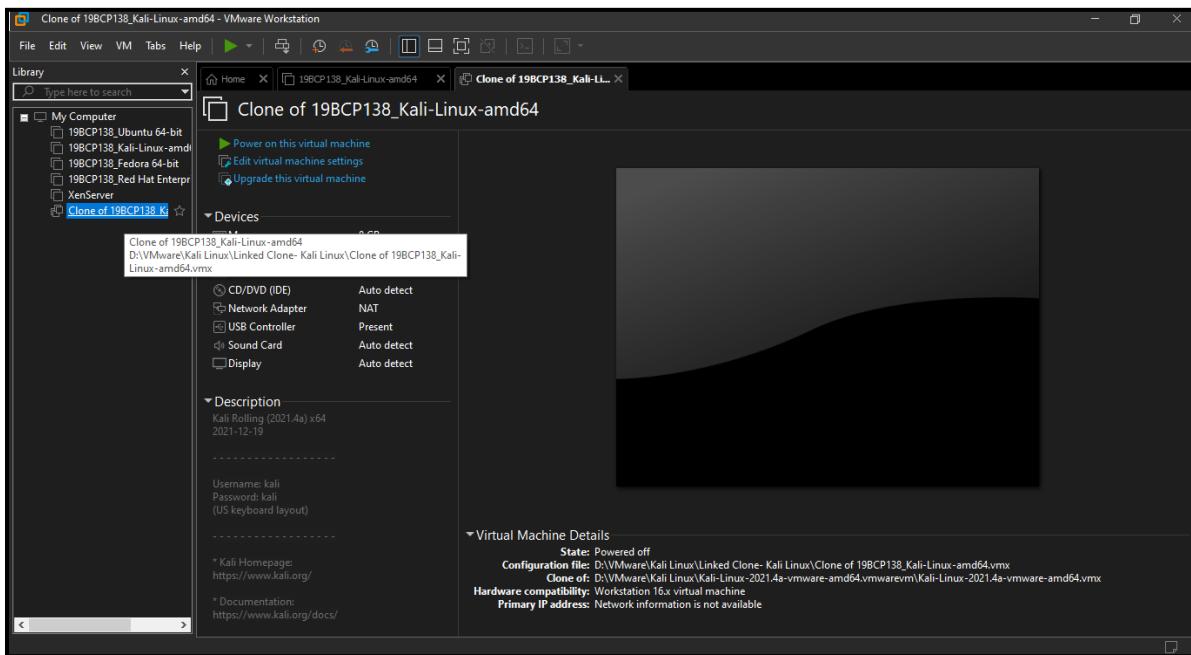
5. Name Virtual Machine and Select the Location where to Save and Select Finish.



## 6. Now Linked Clone - Virtual Machine will be Created.

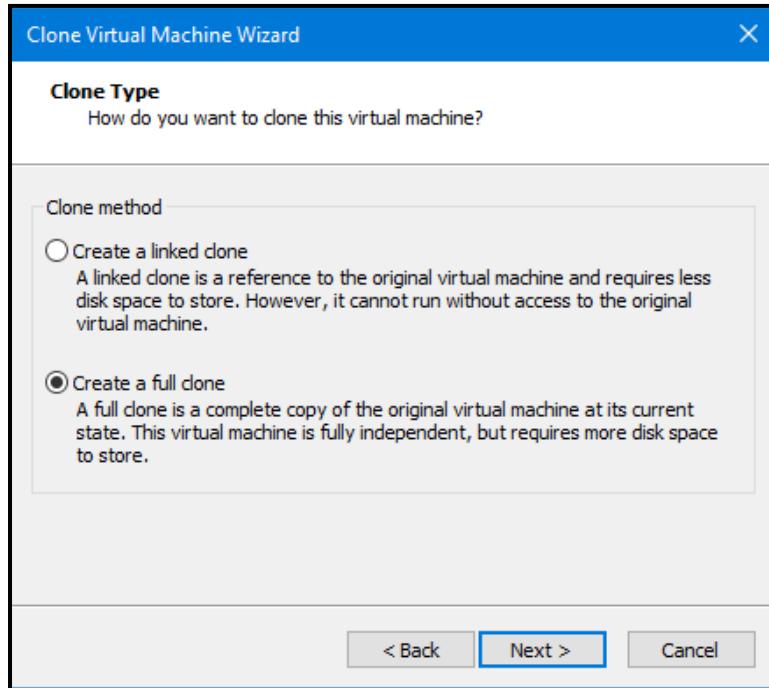


## 7. Now As we can see Linked Clone of Kali Linux is Created.

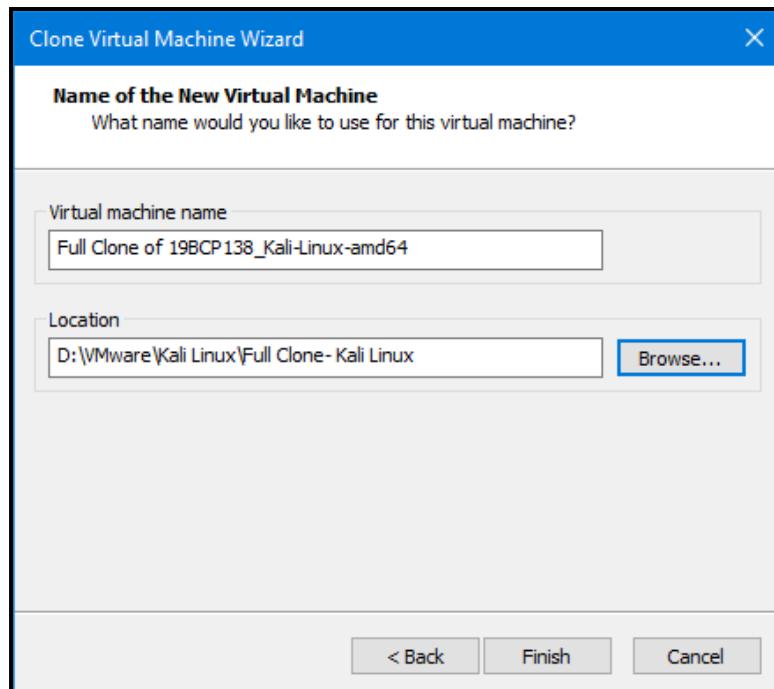


➤ Steps to do Cloning of Virtual Machine: **Full Cloning**

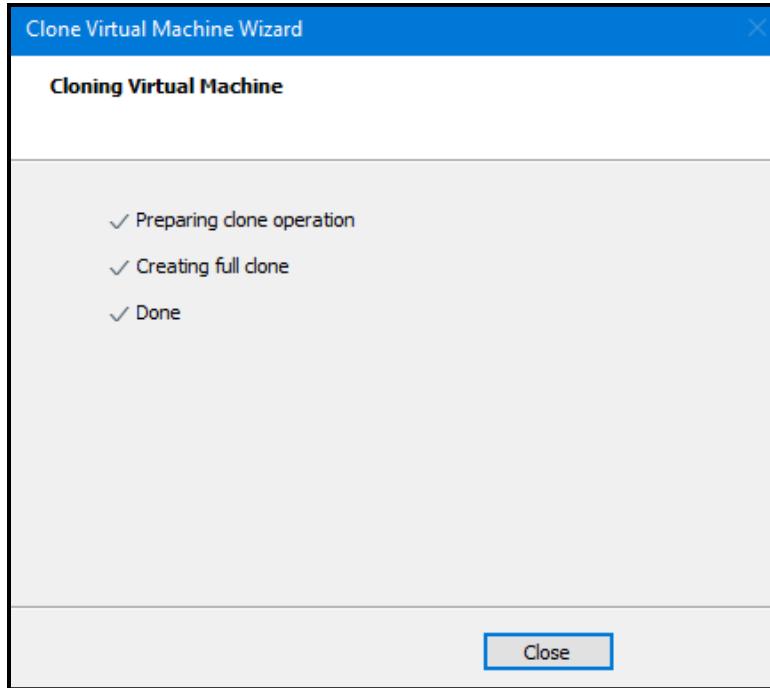
1. Follow the Steps-1,2&3 same as Linked Cloning then Select Full Clone and go Next.



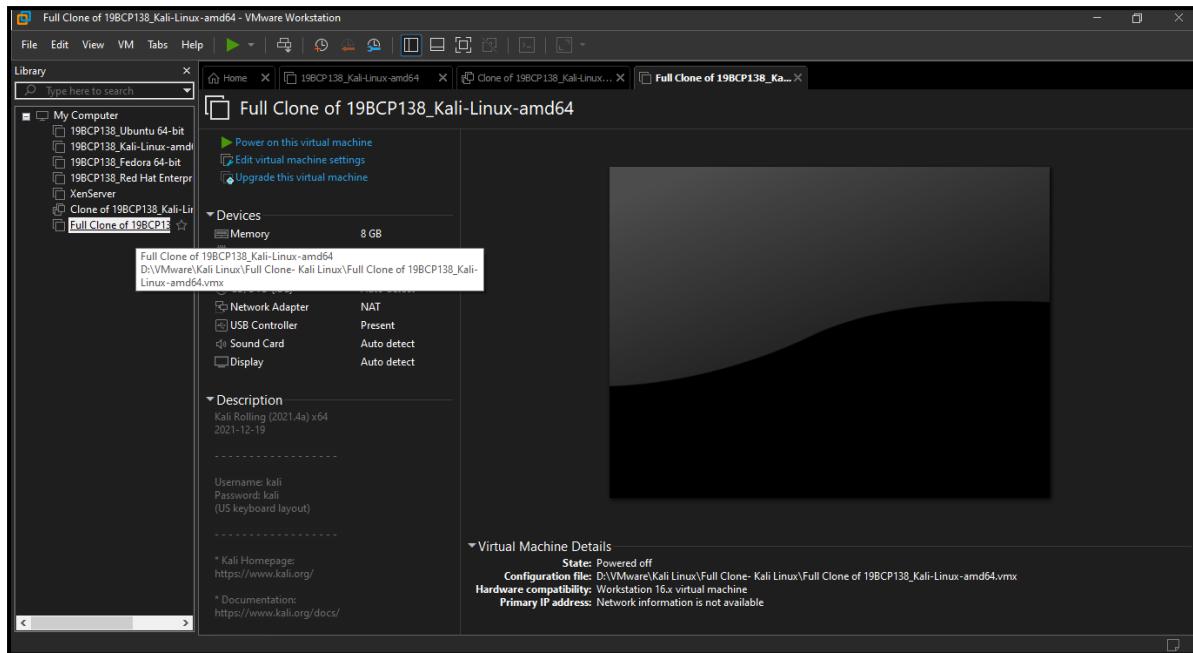
2. Name Virtual Machine and Select the Location where to Save and Select Finish.



3. Now Full Clone - Virtual Machine will be Created.



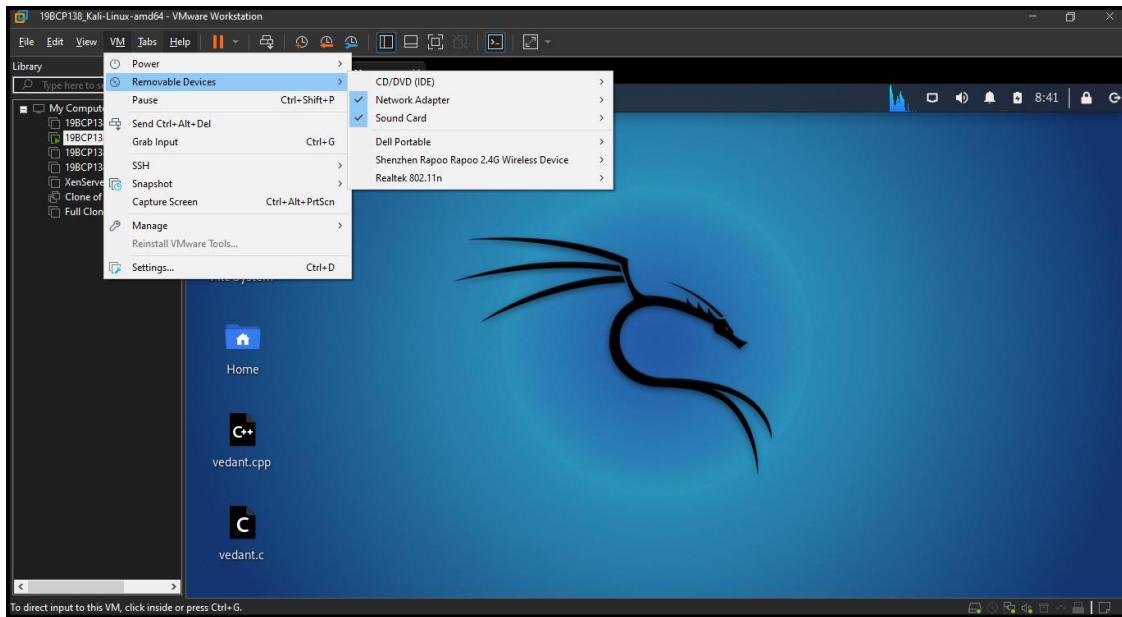
4. Now As we can see Full Clone of Kali Linux is Created.



## 2. Add a USB Controller to Virtual Machine

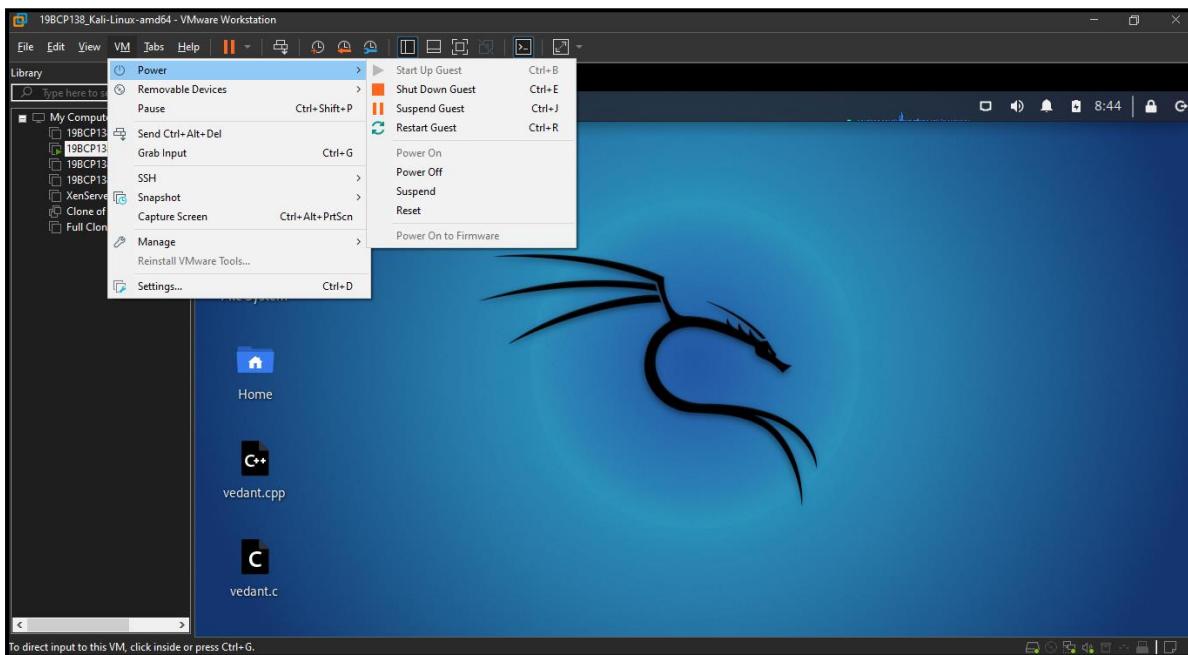
➤ Steps to add USB Controller:

1. Select VM → Removable device → *external\_hard\_drive\_name*, and then click Connect. Here USB Controller is already Present.



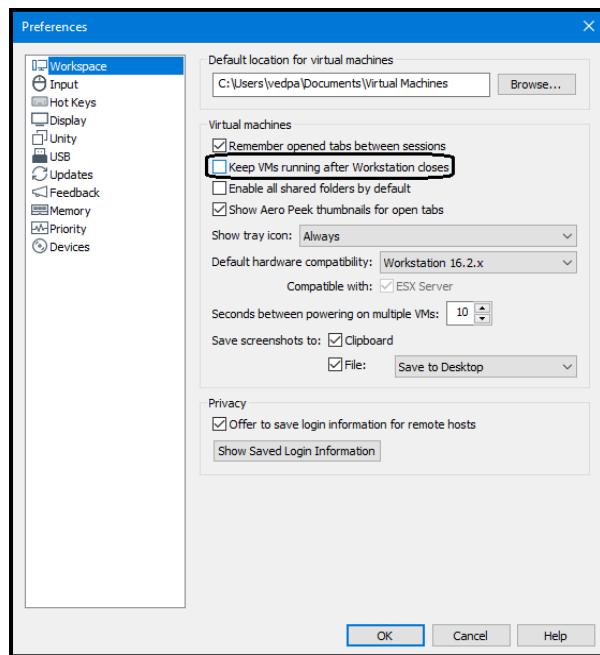
## 3. Closing Virtual Machine & Exiting Workstation Pro

a. Hard and Soft Power Off.

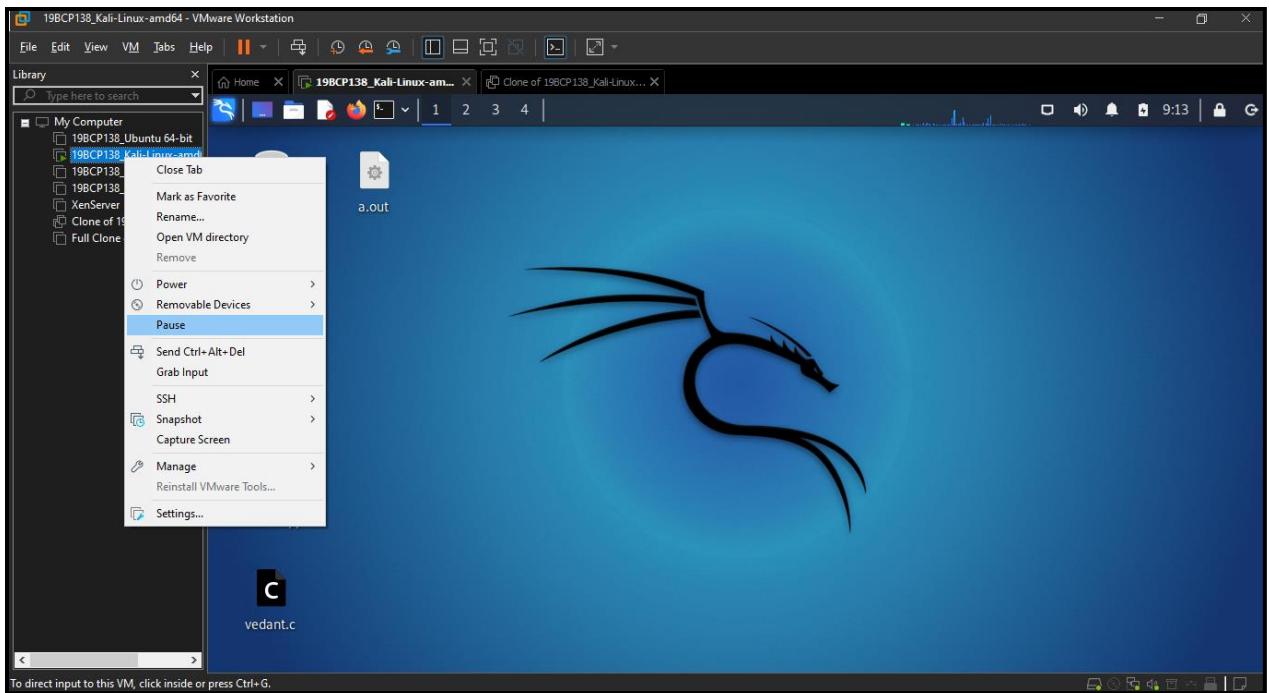


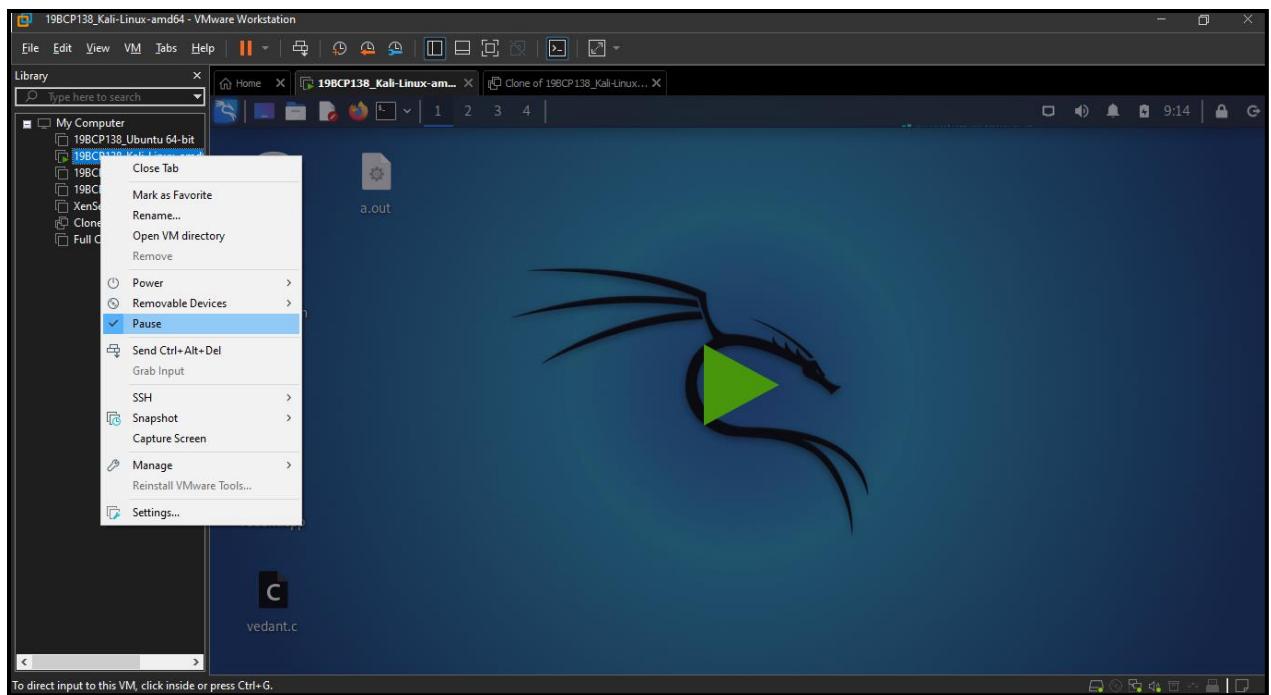
b. Configure Virtual Machine to Run in the Background.

1. Select Edit → Preferences... Then a Dialogue Box will pop-up and check-in a box.



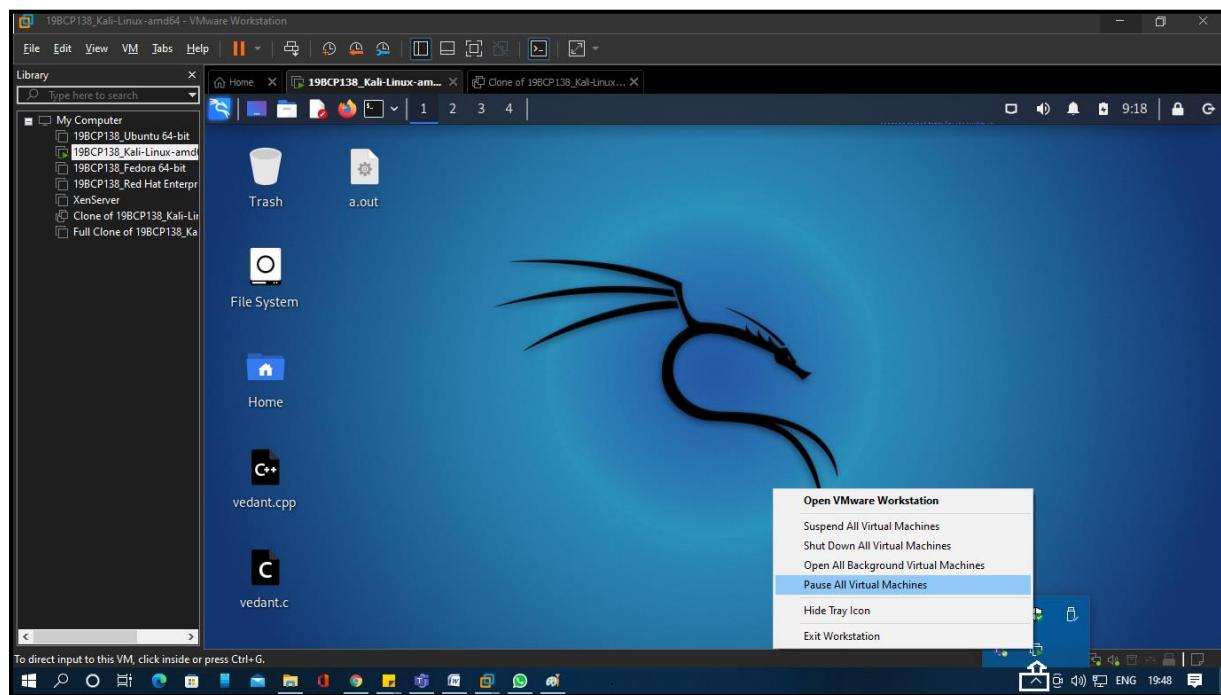
c. Pause and Unpause a Virtual Machine.



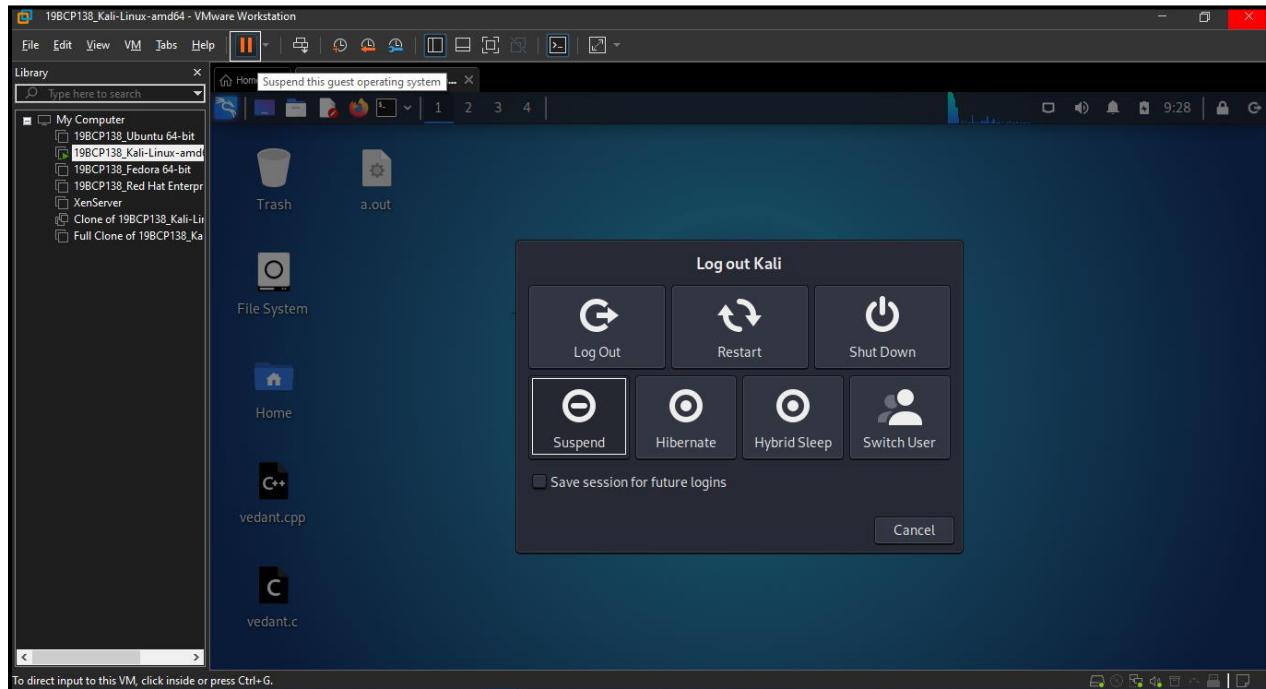


UNPAUSE

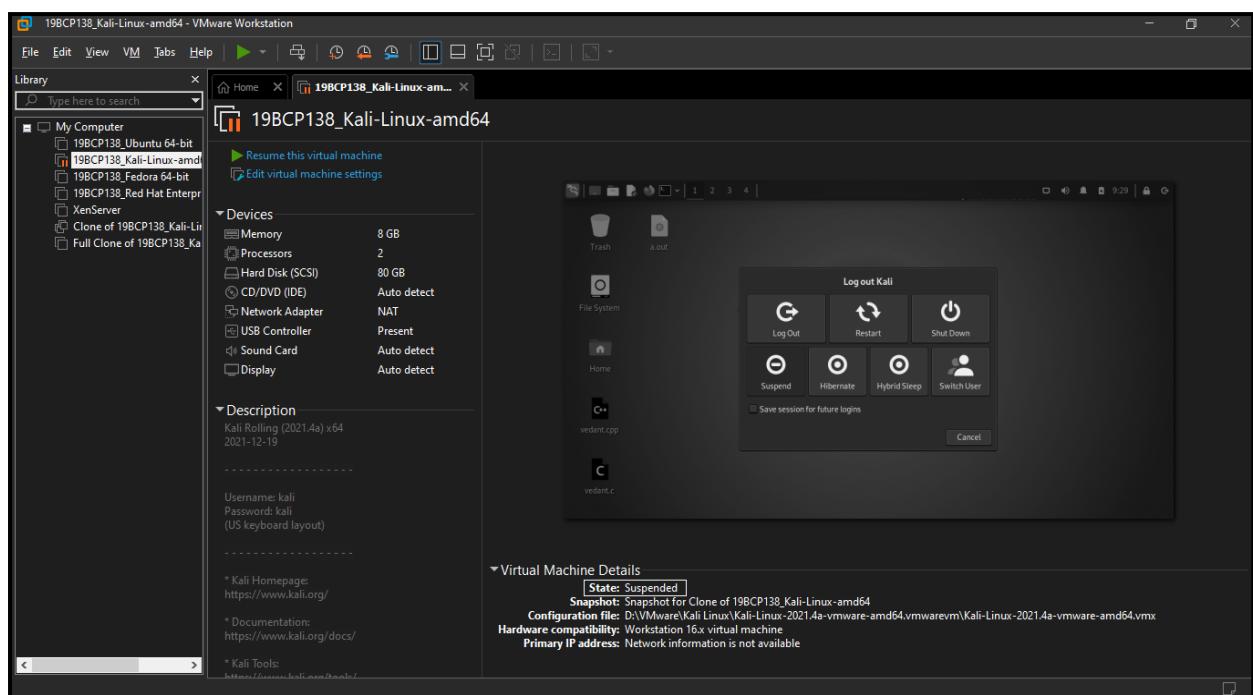
d. Pause all Virtual Machine without interacting Workstation Pro.



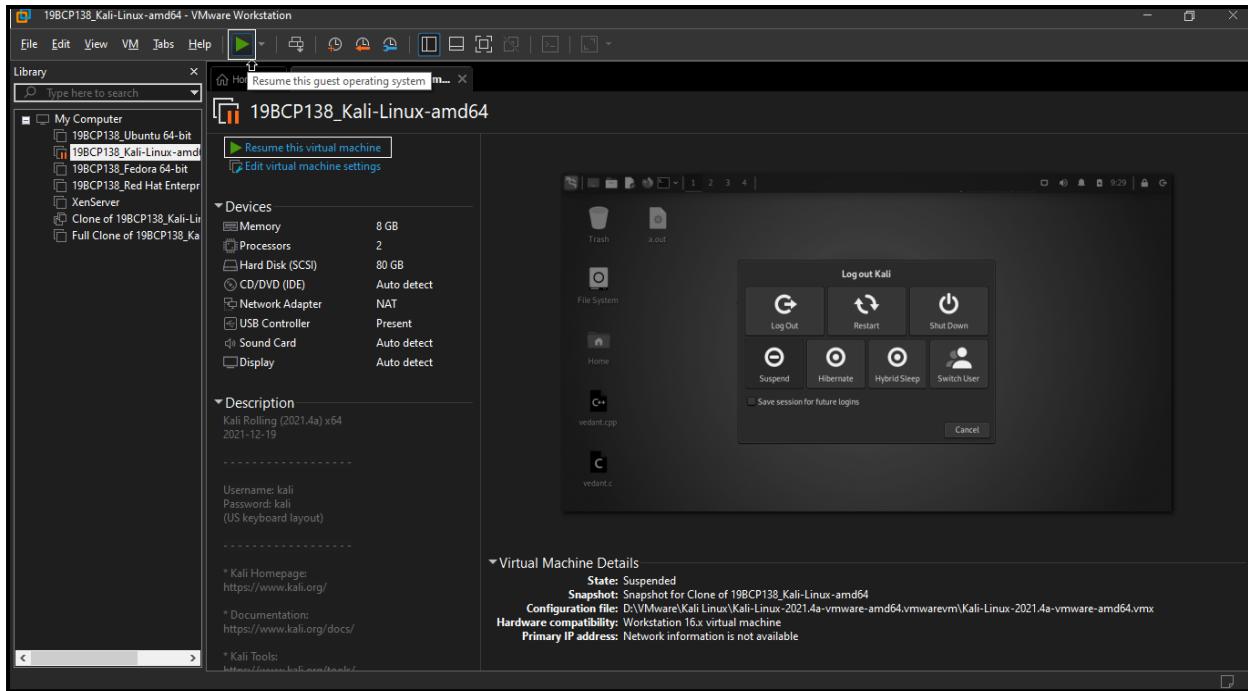
## e. Suspend and Resume Virtual Machines.



SUSPEND



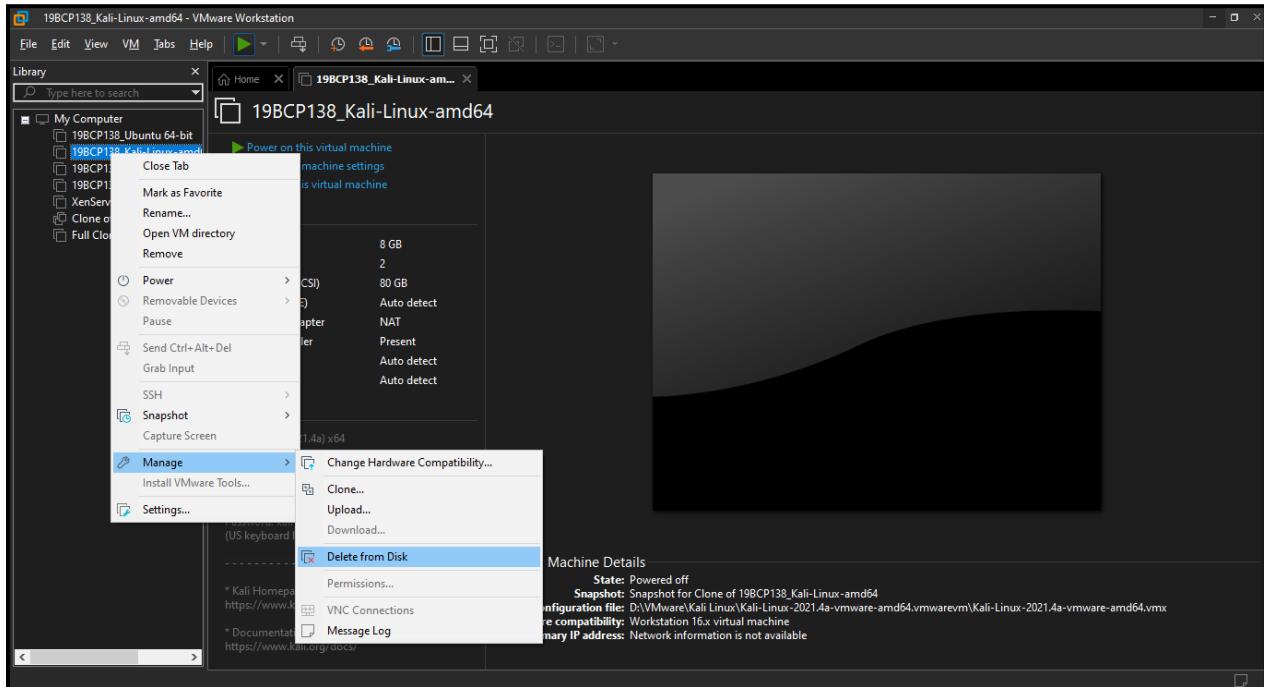
SUSPENDED



## RESUME

## 4. Delete a Virtual Machine from Disk

➤ Steps to Delete a Virtual Machine:



# **Experiment – 5**

## **Docker**

❖ **Aim:** To Explore Docker Container.

❖ **Introduction:**

Docker is a set of platforms as a service product that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their software, libraries, and configuration files; they can communicate with each other through well-defined channels.

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

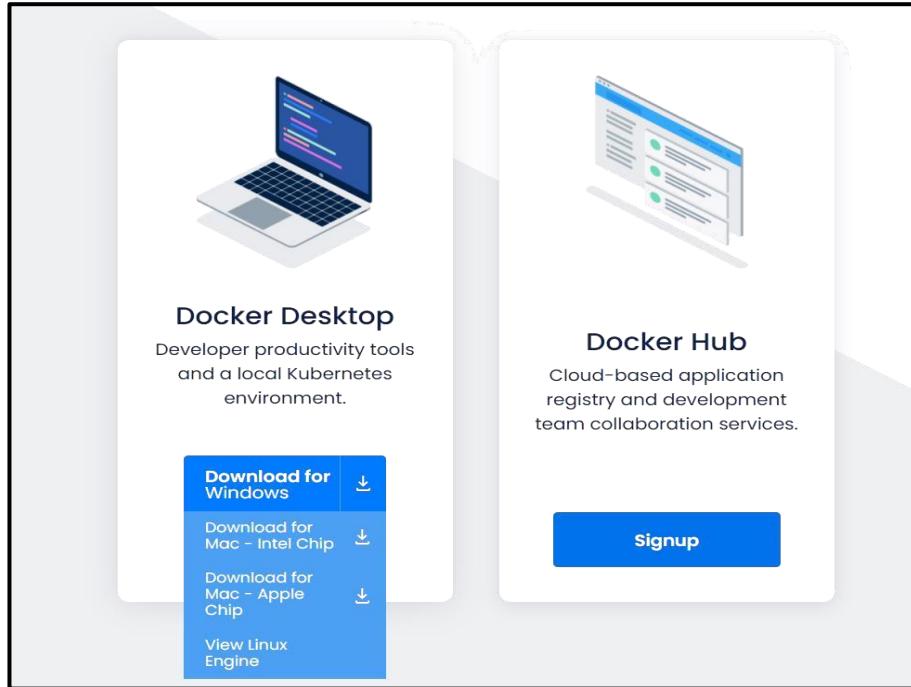
Docker works by providing a standard way to run your code. Docker is an operating system for containers. Similar to how a virtual machine virtualizes (removes the need to directly manage) server hardware, containers virtualize the operating system of a server. Docker is installed on each server and provides simple commands you can use to build, start, or stop containers.

Using Docker lets you ship code faster, standardize application operations, seamlessly move code, and save money by improving resource utilization. With Docker, you get a single object that can reliably run anywhere. Docker's straightforward syntax gives you full control. Wide adoption means there's a robust ecosystem of tools and off-the-shelf applications that are ready to use with Docker.

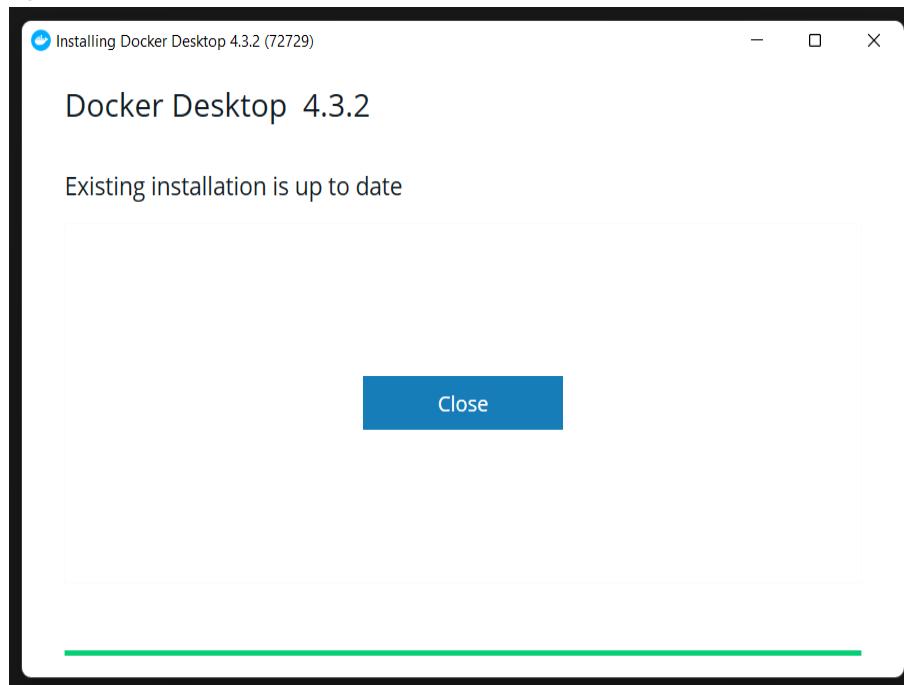
You can use Docker containers as a core building block creating modern applications and platforms. Docker makes it easy to build and run distributed micro services architectures, deploy your code with standardized continuous integration and delivery pipelines, build highly-scalable data processing systems, and create fully-managed platforms for your developers.

## ❖ Procedure:

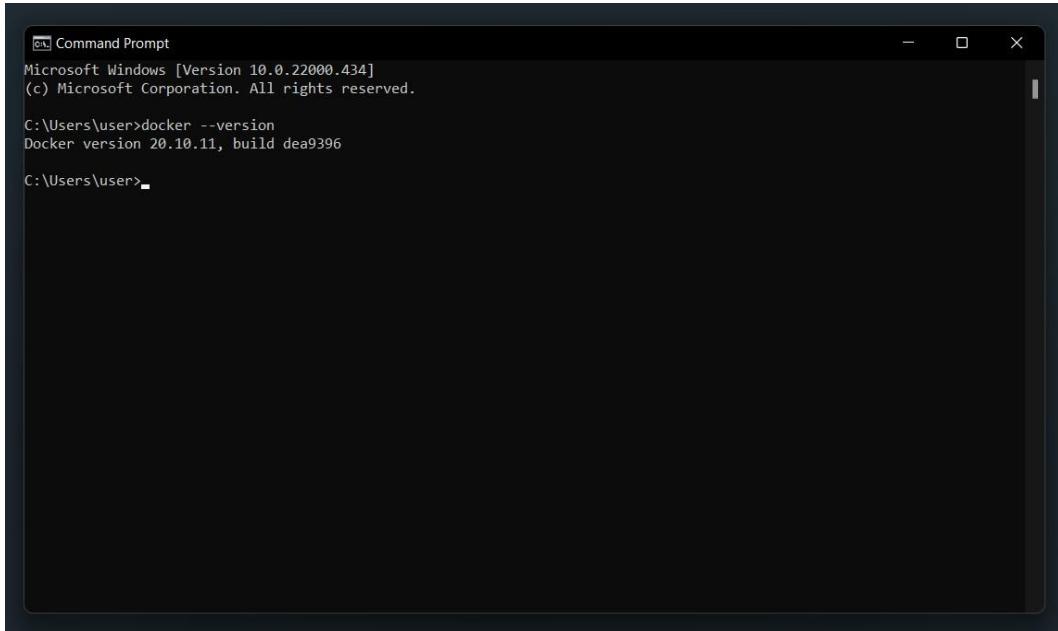
**Step 1:** Go to (<https://www.docker.com/get-started>) and download Docker for your Operating System.



**Step 2:** Click on the downloaded .exe file to install Docker in your system. Follow the normal procedure to install app. As it's already installed in my system, it is showing this.



**Step 3:** Go to Terminal/Command and run `docker --version` to verify whether the Docker is installed properly or not.



```
Command Prompt
Microsoft Windows [Version 10.0.22000.434]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>docker --version
Docker version 20.10.11, build dea9396

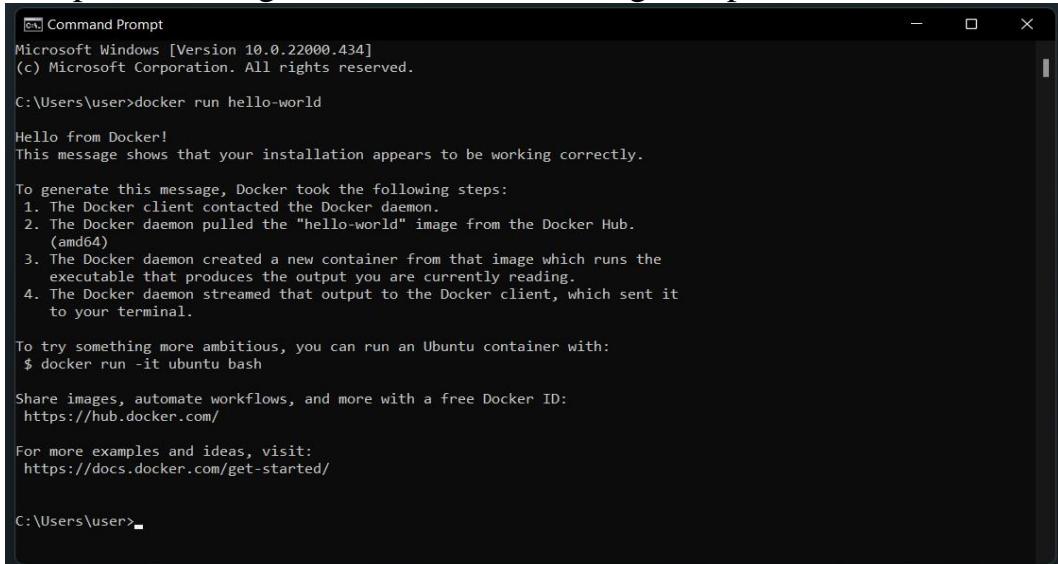
C:\Users\user>
```

**Step 4:** Open a command-line terminal like Power Shell, and try out some Docker commands!

Run **docker run hello-world** to verify that Docker can pull and run images. It didn't find `hello-world` in local so it downloaded from Docker Hub.

## ❖ Basic Commands:

**docker run:** The docker run command first creates a writeable container layer over the specified image, and then starts it using the specified command.



```
Command Prompt
Microsoft Windows [Version 10.0.22000.434]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

C:\Users\user>
```

**docker pull:** Can be used to pull images from docker hub.

The screenshot shows the Docker Desktop interface. At the top, there are tabs for 'LOCAL' and 'REMOTE REPOSITORIES'. Below the tabs is a search bar and a checkbox for 'In Use only'. A table lists two images: 'hello-world' and 'ubuntu'. The 'hello-world' row has an 'IN USE' button next to it. The 'ubuntu' row also has an 'IN USE' button. The table columns are 'NAME', 'TAG', 'IMAGE ID', 'CREATED', and 'SIZE'. The 'hello-world' image was created 4 months ago and is 13.26 KB. The 'ubuntu' image was created 1 day ago and is 72.78 MB. Below the table is a command prompt window titled 'Command Prompt'. It shows the command 'docker pull ubuntu' being run, followed by the output of the pull process, which includes the digest and status message. The command prompt ends with 'C:\Users\user>'.

NAME	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	feb5d9fea6a5	4 months ago	13.26 KB
ubuntu	latest	54c9d81ccb44	1 day ago	72.78 MB

```
C:\Users\user>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
08c01a0ec47e: Pull complete
Digest: sha256:669e01b58ba5b5beb2836b253c1fd5768333f0d1dbcb834f7c07a4dc93f474be
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\user>
```

**docker images:** The default docker images will show all top-level images, their repository and tags, and their size.

The screenshot shows the Docker Desktop interface. At the top, there are tabs for 'LOCAL' and 'REMOTE REPOSITORIES'. Below the tabs is a search bar and a checkbox for 'In Use only'. A table lists the same two images as before. Both rows have an 'IN USE' button next to them. The table columns are 'NAME', 'TAG', 'IMAGE ID', 'CREATED', and 'SIZE'. The 'hello-world' image was created 4 months ago and is 13.26 KB. The 'ubuntu' image was created 1 day ago and is 72.78 MB. Below the table is a command prompt window titled 'Command Prompt'. It shows the command 'docker images' being run twice, once for 'ubuntu' and once for 'hello-world'. The output shows the repository name, tag, image ID, creation time, and size for each. The command prompt ends with 'C:\Users\user>'.

NAME	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	feb5d9fea6a5	4 months ago	13.26 KB
ubuntu	latest	54c9d81ccb44	1 day ago	72.78 MB

```
C:\Users\user>docker images ubuntu
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu latest 54c9d81ccb44 28 hours ago 72.8MB

C:\Users\user>docker images hello-world
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest feb5d9fea6a5 4 months ago 13.3kB

C:\Users\user>
```

**docker start:** A stopped container can be restarted with all its previous changes intact using docker start.

**docker save:** Produces a tarred repository to the standard output stream.

**docker rename:** The docker rename command renames a container.

```
$ docker rename CONTAINER NEW_NAME
```

**docker rm:** Remove one or more containers. docker rmi: Remove one or more images.

## ❖ Observation and Learning:

It was found that using Docker lets you ship code faster, standardize application operations, seamlessly move code, and save money by improving resource utilization. With Docker, you get a single object that can reliably run anywhere. Docker's simple and straightforward syntax gives you full control. Wide adoption means there's a robust ecosystem of tools and off-the-shelf applications that are ready to use with Docker.

With this experiment, I got to know about the Docker, how it works, why to use Docker and when to use Docker.

## ❖ Conclusion:

Docker is a set of platforms as a service product that use OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries, and configuration files; they can communicate with each other through well-defined channels.

Docker is a software platform that allows you to build, test, and deploy applications quickly.

## ❖ Questions:

**Question: 1 →What is the difference between Docker and VMware?**

- VMware emulates machine hardware whereas Docker emulates the operating system in which your application runs. Docker is a much more lightweight virtualization technology since it does not have to emulate server hardware resources. The focus is on abstracting the environment required by the app, rather than the physical server.
- VMware, just like actual machine hardware, lets you install operating systems and other tasks that require a full server.

**Question: 2 →Where is docker images located?**

- The heaviest contents are usually images. If you use the default storage driver overlay2, then your Docker images are stored in /var/lib/docker/overlay2. There, you can find different files that represent read-only layers of a Docker image and a layer on top of it that contains your changes.

**Question: 3 →What do you mean by the term container?**

- A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. ... Secure: Applications are safer in containers and Docker provides the strongest default isolation capabilities in the industry.

## **Experiment – 6**

### **Cloud Analyst**

❖ **Aim:** Study of Load Balancing using **Cloud Analyst**.

#### **❖ Introduction:**

In cloud computing, load balancing is considered one of the most important topics. Load Balancing means, distributing the workload of multiple network links to achieve maximum throughput and minimum response time. Using various load balancing techniques, we can avoid the overloading that may occur in the cloud due to heavy workloads. There are mainly three aspects (performance, time, and cost) considering which, three different load balancing algorithms can be used.

Cloud Analyst is a tool developed at the University of Melbourne whose goal is to support the evaluation of social networks according to the geographical distribution of users and data centers. In this tool, communities of users (User Bases) and data centers supporting the social networks are characterized and based on their location; parameters such as user experience while using the social network application and load on the data center is obtained/logged. Cloud Analyst is used to model and analyze real-world problems through case studies of social networking applications deployed on the cloud.

#### **❖ Instructions:**

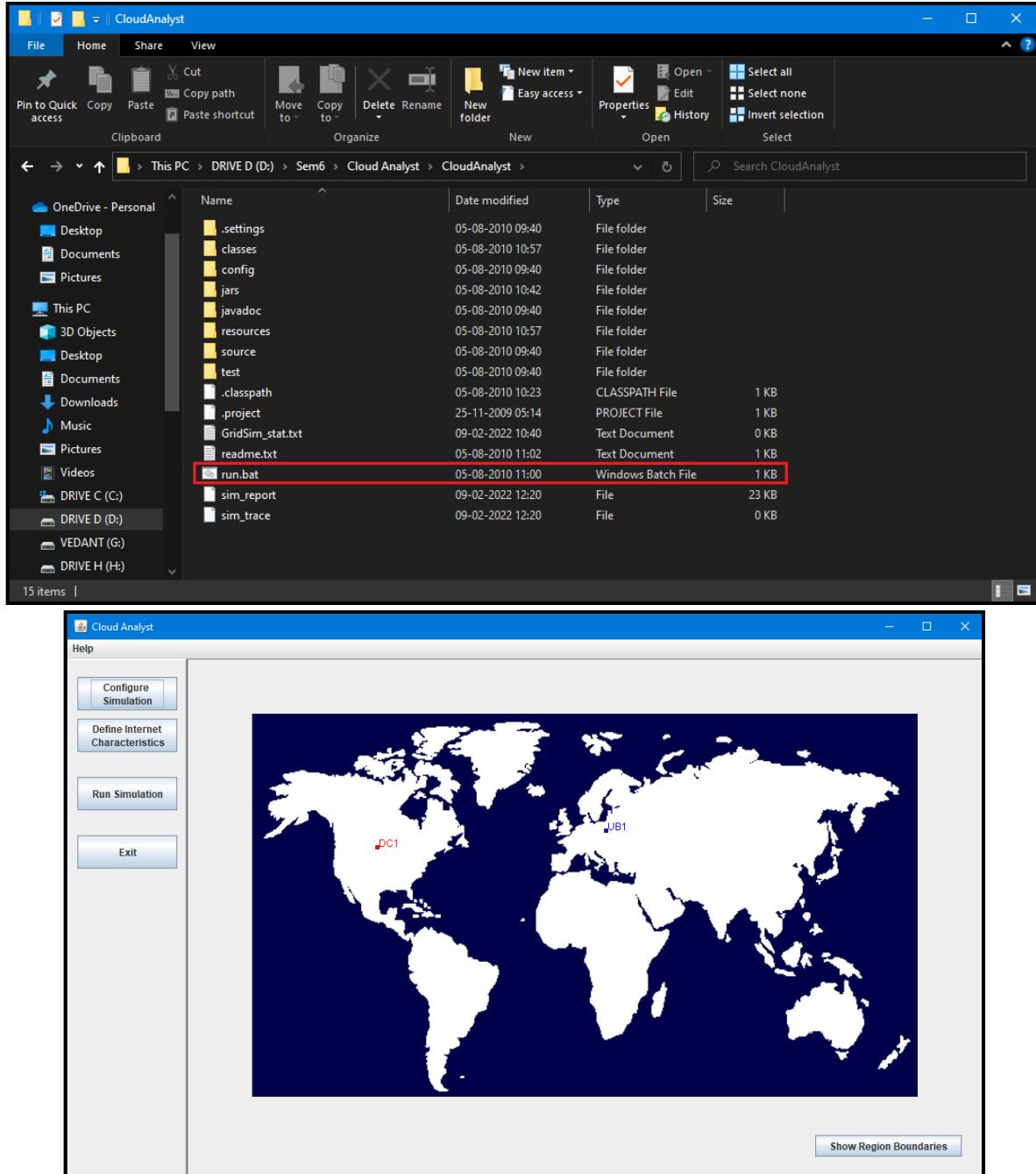
##### **➤ Installing of Cloud Analyst Simulator:**

**Step-1:** To install Cloud Analyst Simulator, first go to this site (<http://www.cloudbus.org/cloudsim/CloudAnalyst.zip>).

**Step-2:** Open the page, scroll down and you will find Cloud Analyst. Download the .zip file present there.

**Step-3:** Once the download is complete, extract this zip file.

**Step-4:** In the extracted files you will find, batch file named run.bat. Running this file will open the simulator.



Simulation Screen

Configure the DataCenter and Userbases according to the requirements-  
Scenarios to be completed: R0 – North America, R1 – South America, R2 – Europe, R3 – Asia, R4 – Africa, R5 – Australia.

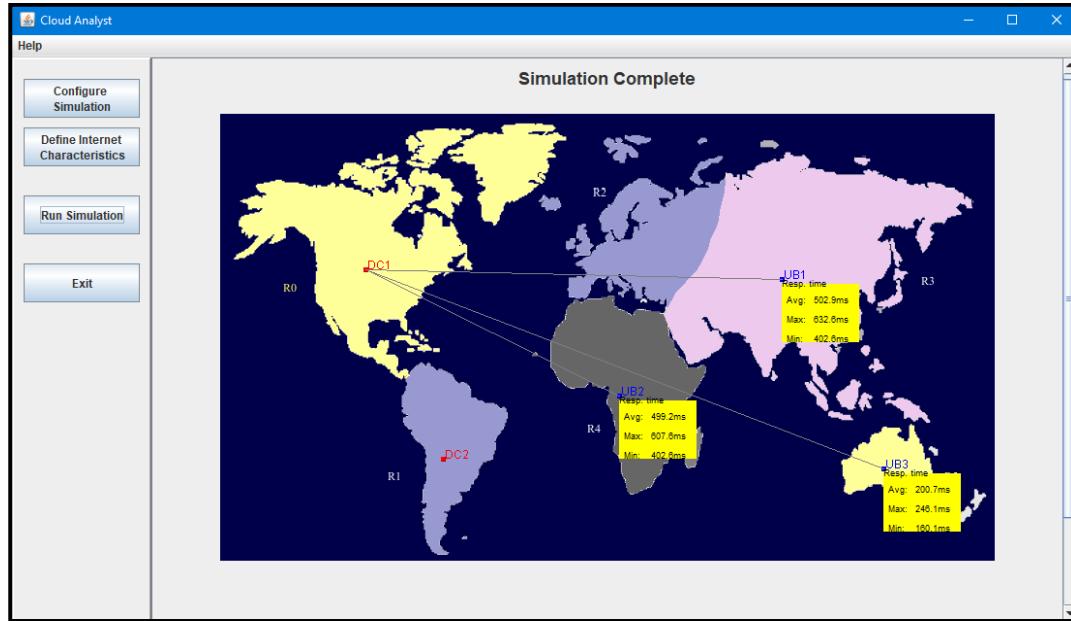
## ❖ Exercise:

➤ **Exercise - 1:** 02 Datacenters in R0 and R1 and Userbases in R3, R4 and R5.

Plot the graphs as shown above:

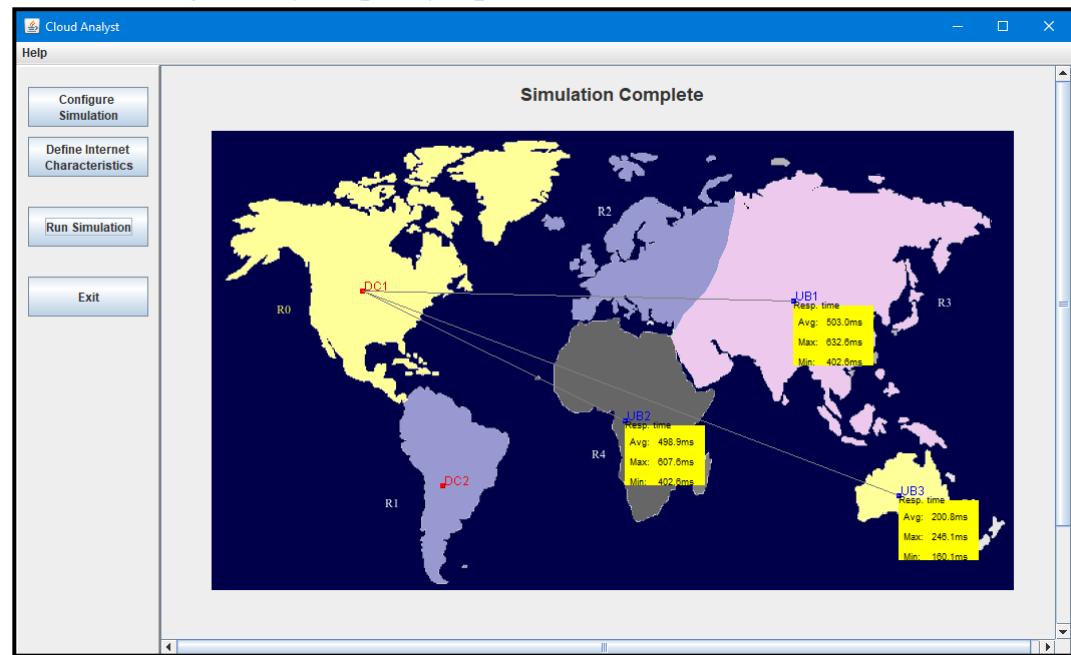
1. Server Broker Policy: Closest Data Center

Load Balancing Policy: Round Robin

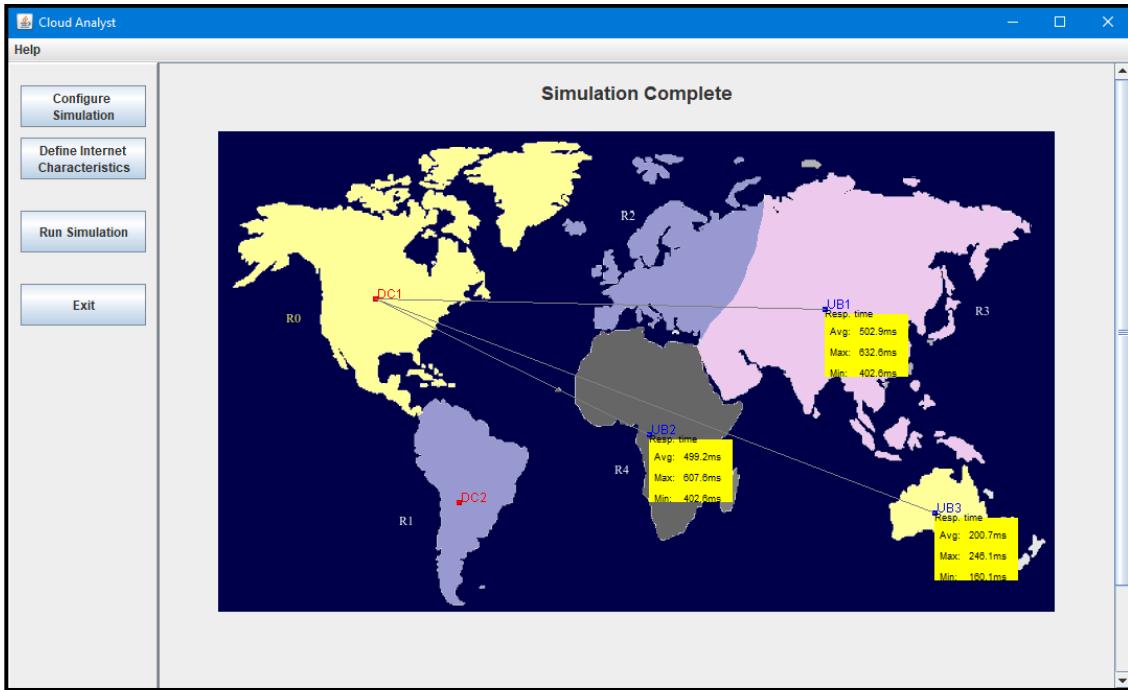


2. Server Broker Policy: Closest Data Center

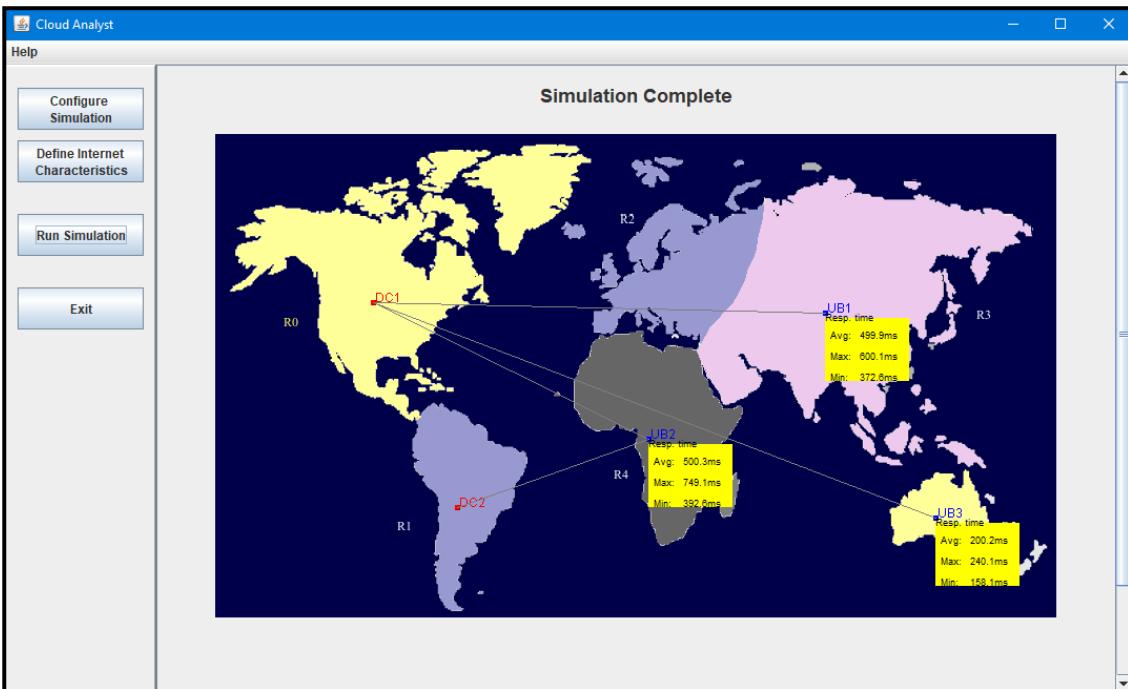
Load Balancing Policy: Equally Spread Current Execution Load



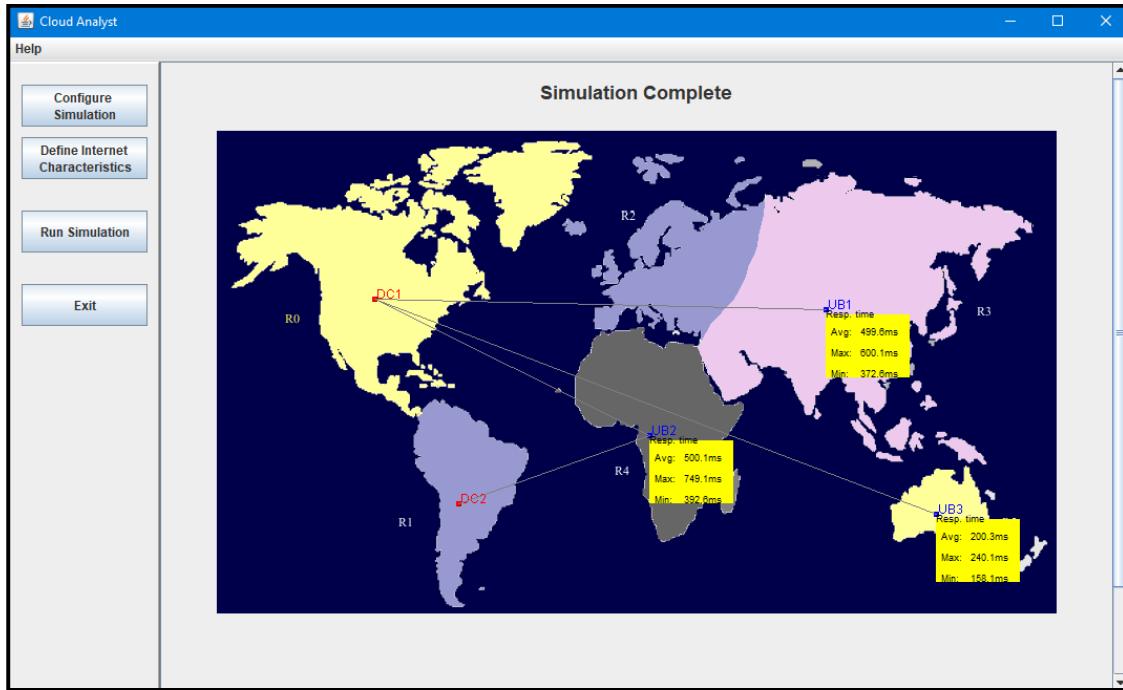
### 3. Server Broker Policy: Closest Data Center Load Balancing Policy: Throttled



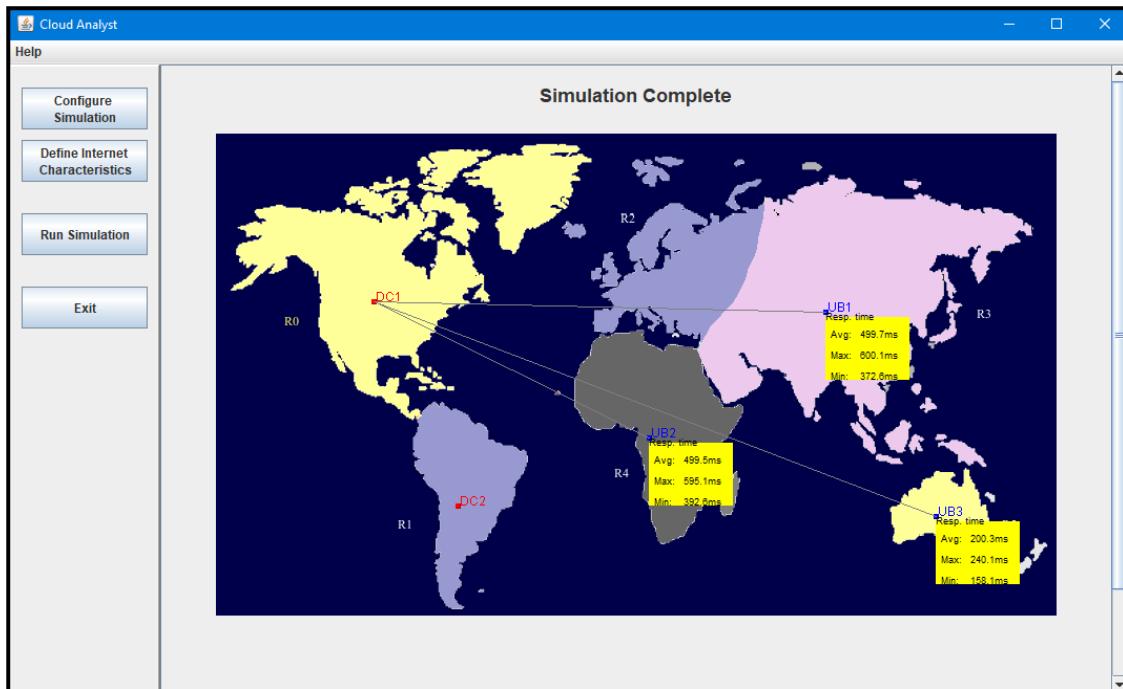
### 4. Server Broker Policy: Optimise Response Time Load Balancing Policy: Round Robin



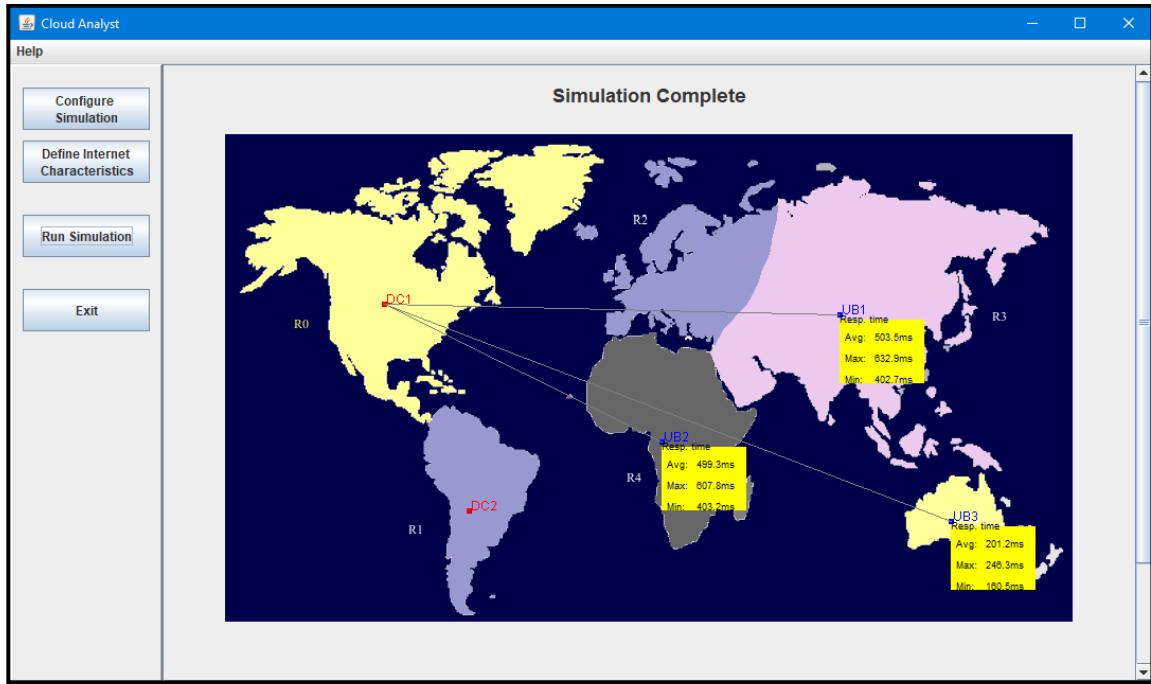
## 5. Server Broker Policy: Optimise Response Time Load Balancing Policy: Equally Spread Current Execution Load



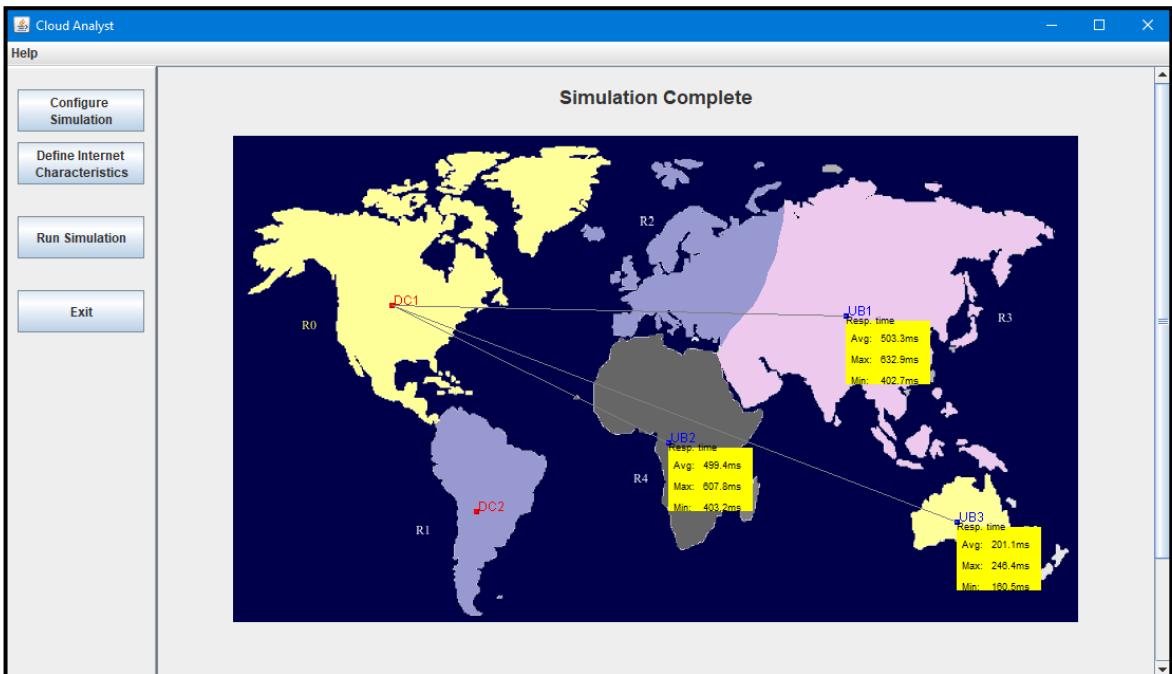
## 6. Server Broker Policy: Optimise Response Time Load Balancing Policy: Throttled



7. Server Broker Policy: Reconfigure dynamically...  
Load Balancing Policy: Round Robin

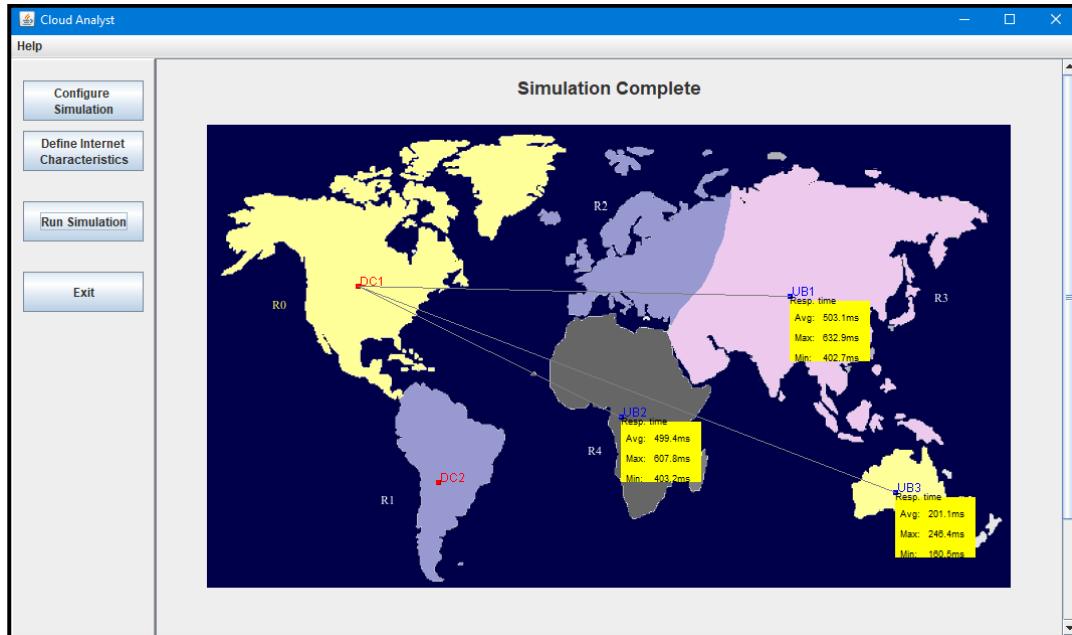


8. Server Broker Policy: Reconfigure dynamically...  
Load Balancing Policy: Equally Spread Current Execution Load



## 9. Server Broker Policy: Reconfigure dynamically...

Load Balancing Policy: Throttled

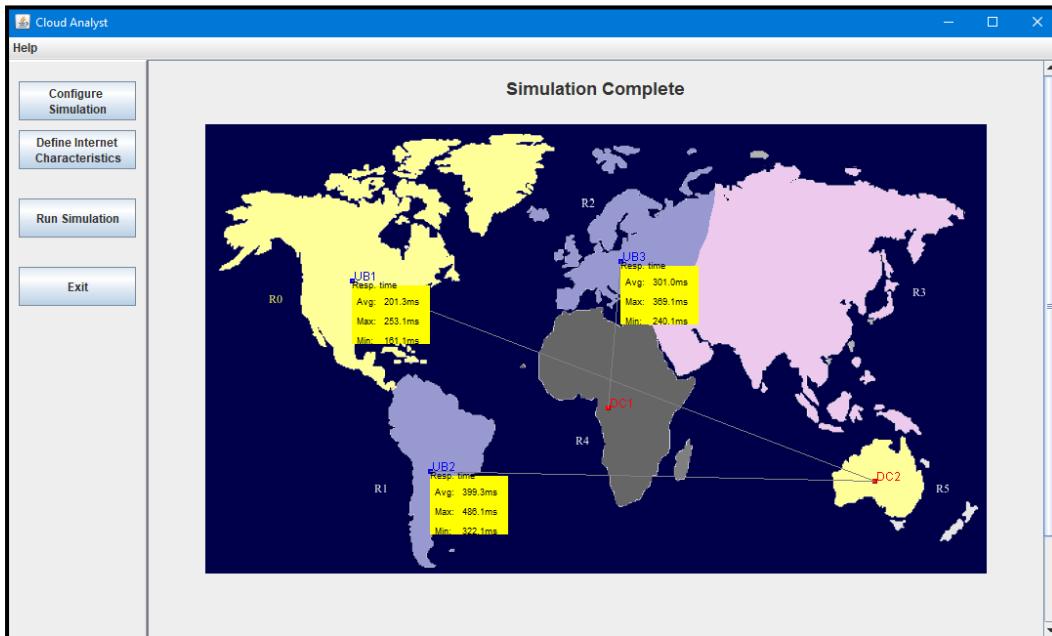


➤ **Exercise - 2:** 02 Datacenters in R4 and R5 and Userbases in R0, R1 and R2.

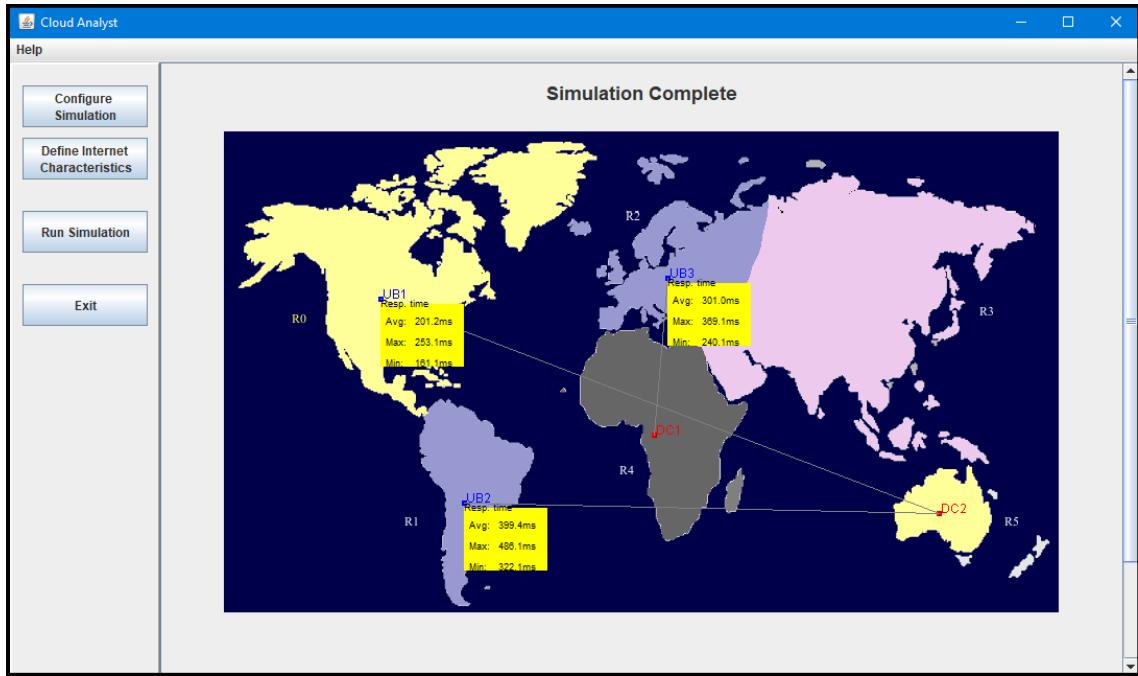
Plot the graphs as shown above:

1. Server Broker Policy: Closest Data Center

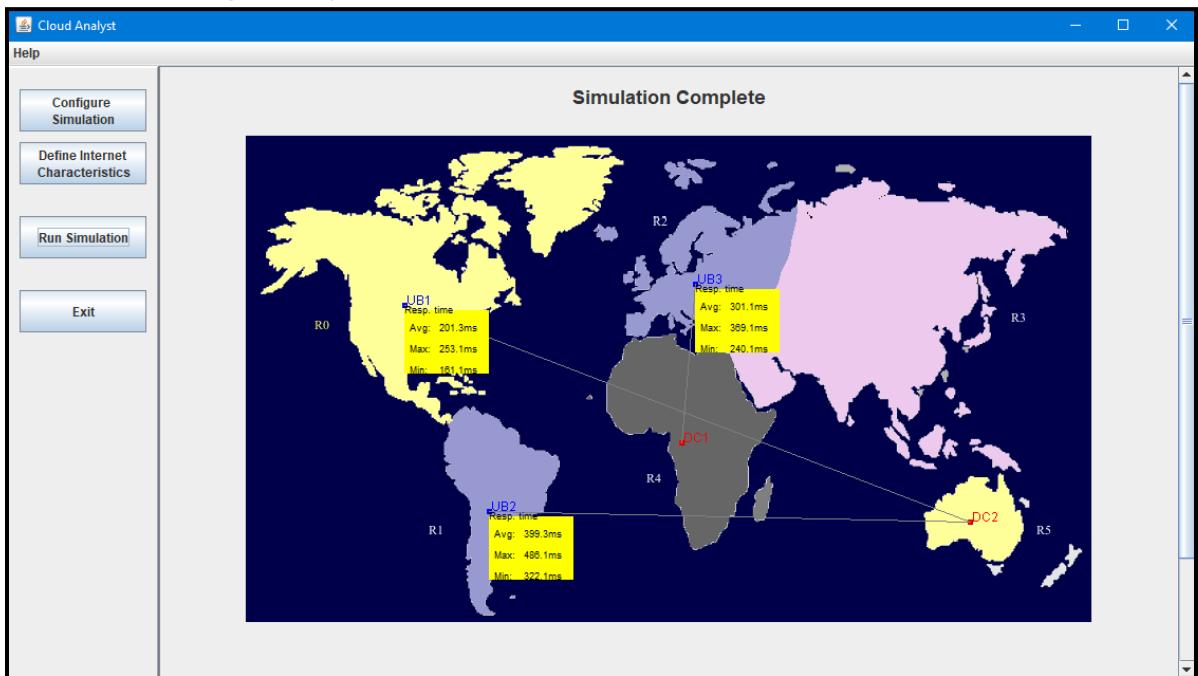
Load Balancing Policy: Round Robin



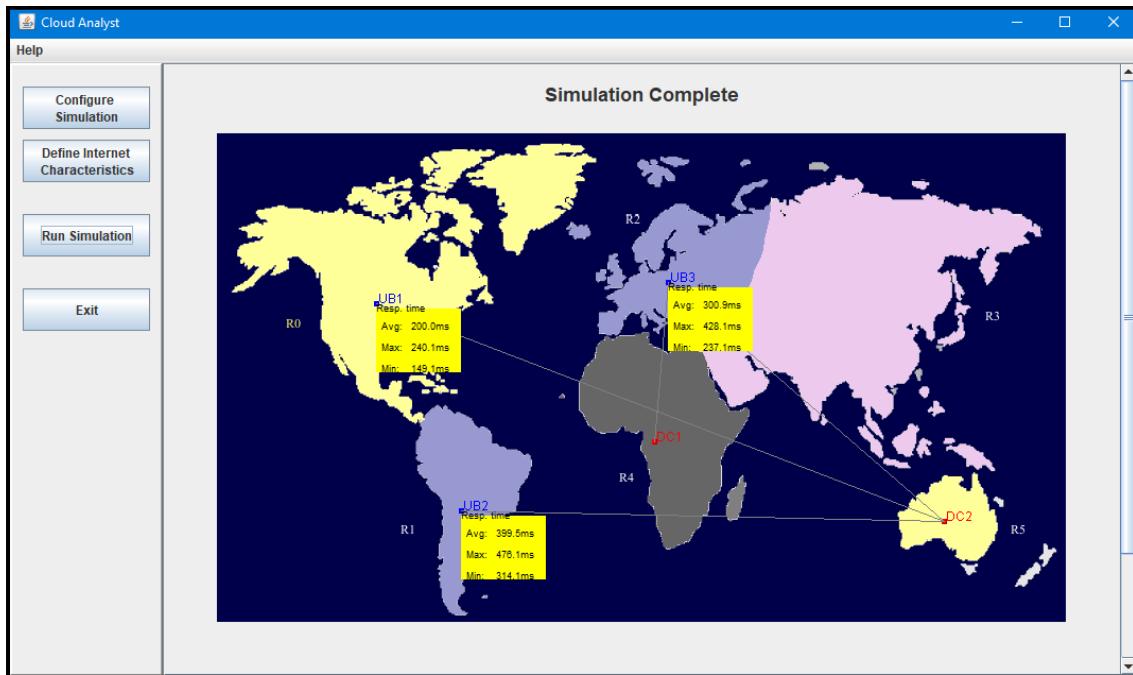
## 2. Server Broker Policy: Closest Data Center Load Balancing Policy: Equally Spread Current Execution Load



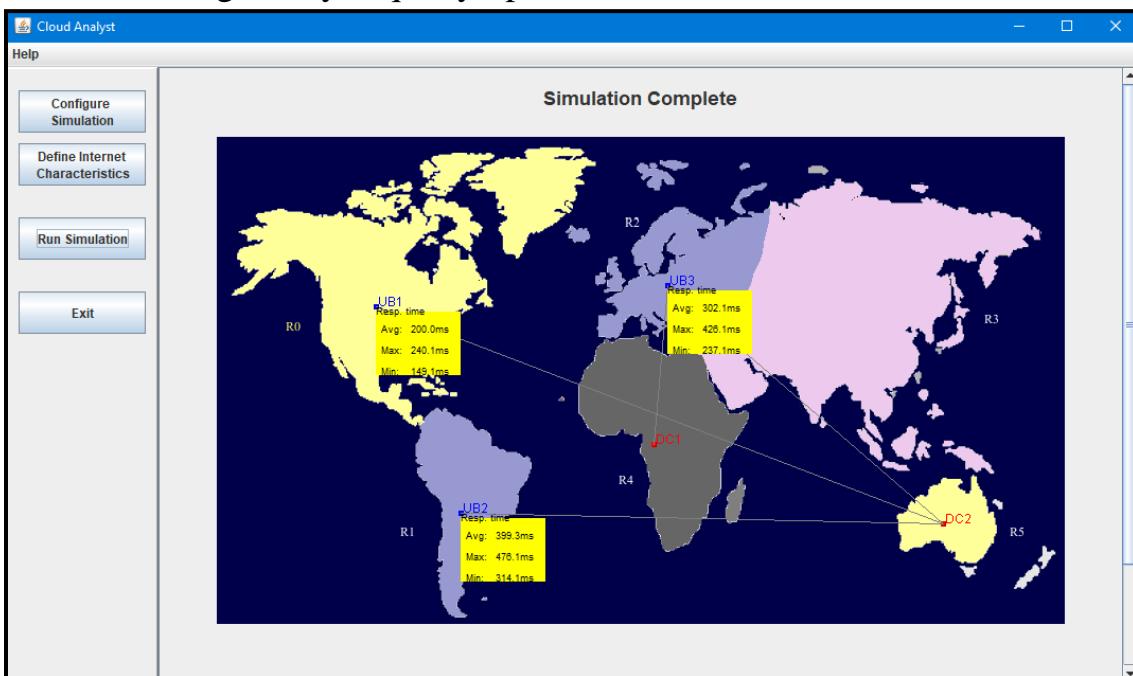
## 3. Server Broker Policy: Closest Data Center Load Balancing Policy: Throttled



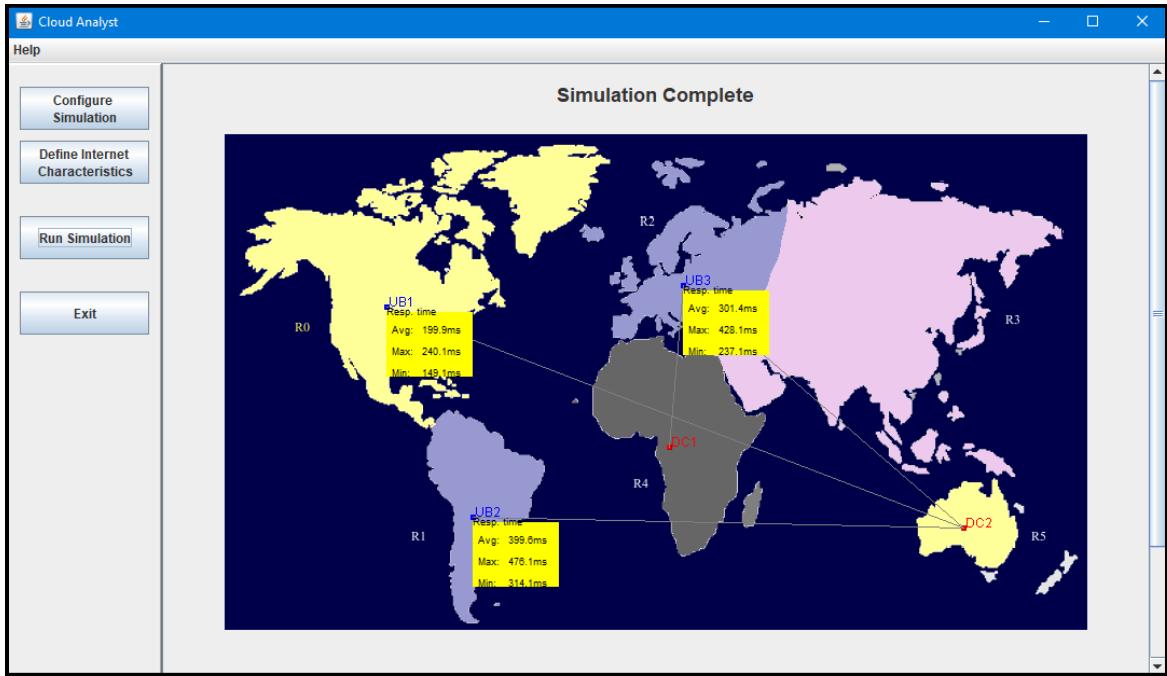
#### 4. Server Broker Policy: Optimise Response Time Load Balancing Policy: Round Robin



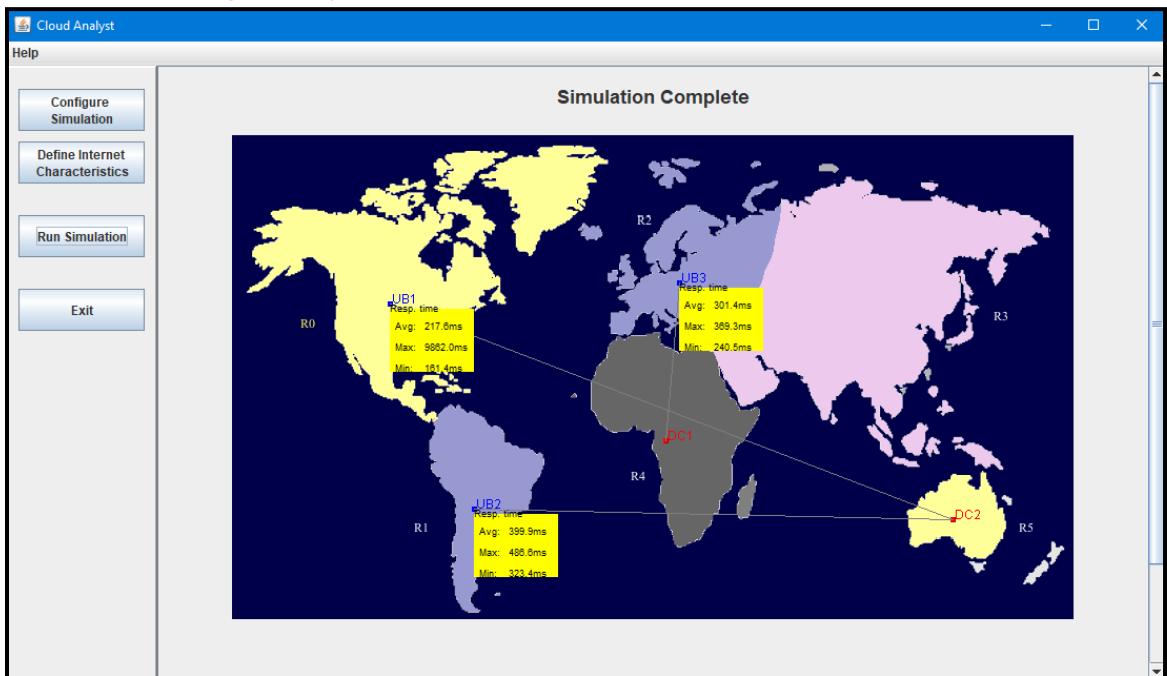
#### 5. Server Broker Policy: Optimise Response Time Load Balancing Policy: Equally Spread Current Execution Load



## 6. Server Broker Policy: Optimise Response Time Load Balancing Policy: Throttled

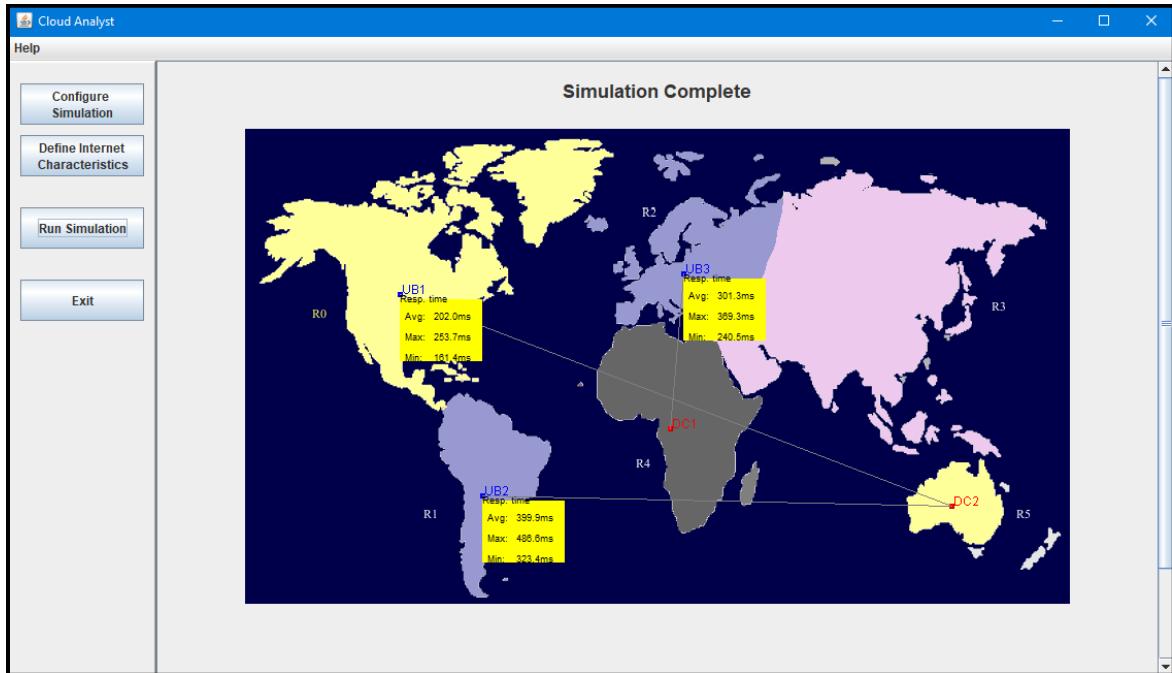


## 7. Server Broker Policy: Reconfigure dynamically... Load Balancing Policy: Round Robin



## 8. Server Broker Policy: Reconfigure dynamically...

Load Balancing Policy: Equally Spread Current Execution Load

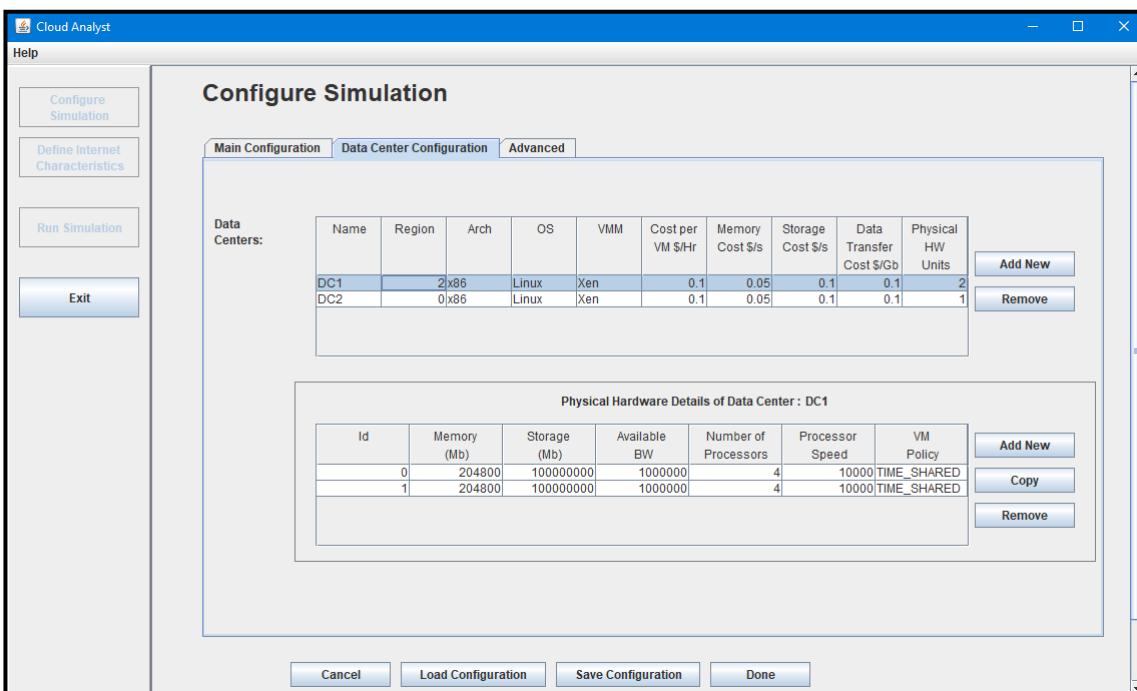
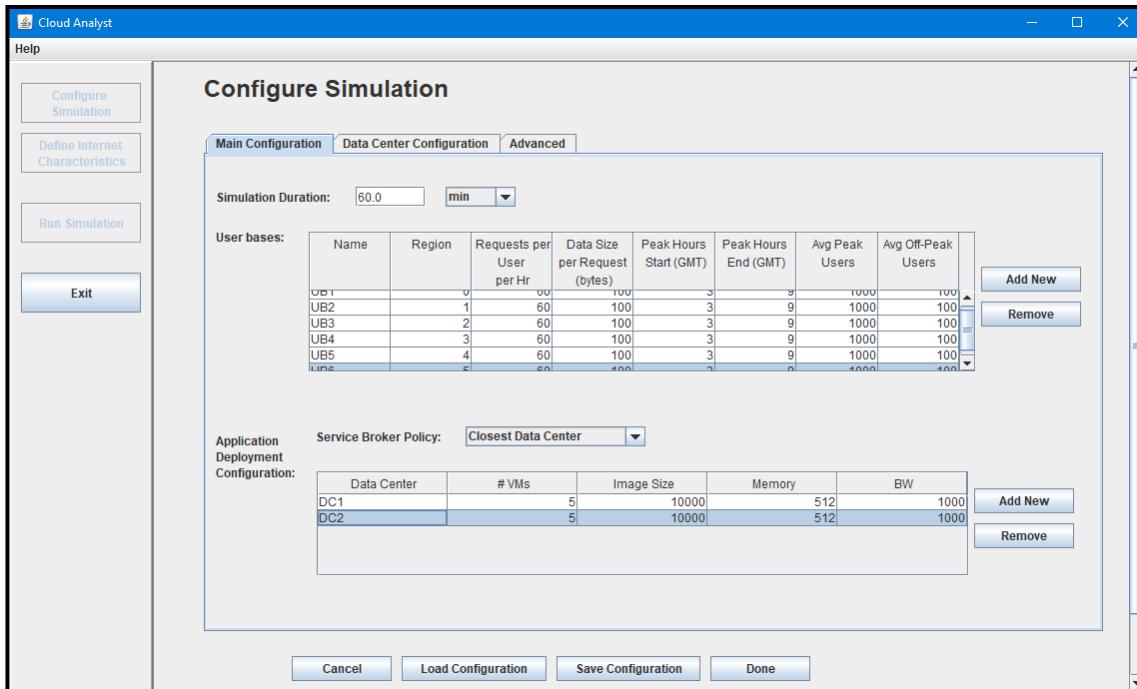


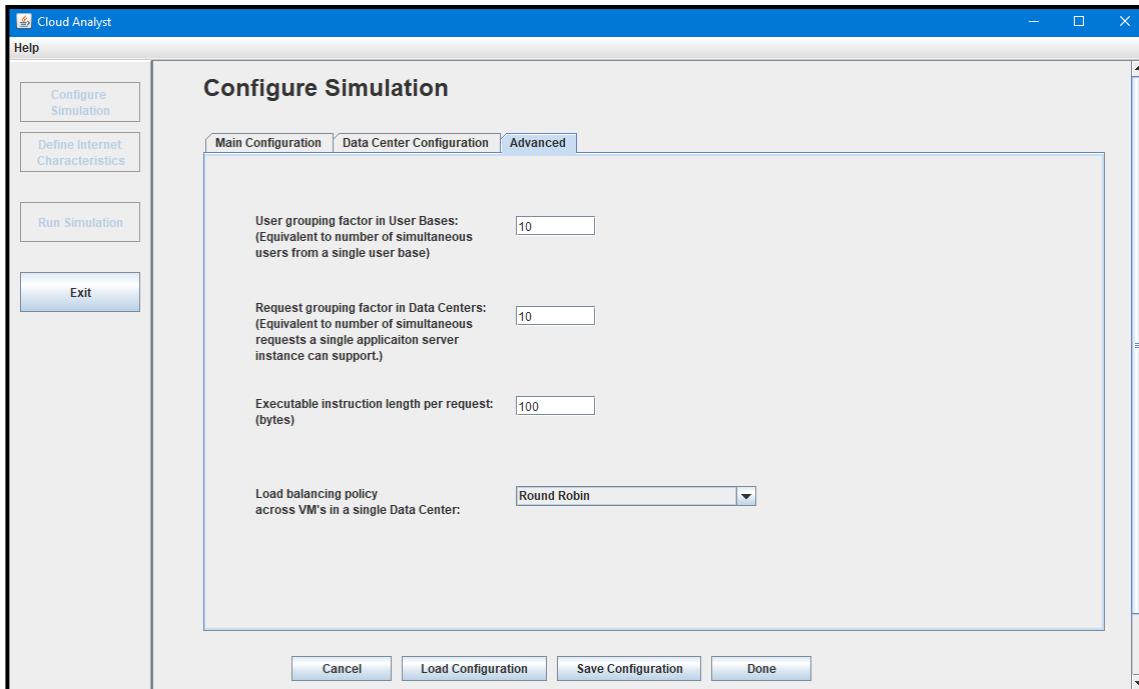
## 9. Server Broker Policy: Reconfigure dynamically...

Load Balancing Policy: Throttled



➤ **Exercise - 3:** Find out the best-case scenario for average time, minimum time and maximum time for given setup. User base in R0 to R5 (each User base having at least 10 applications), at least 2 Data Centres:





The least average response time, least minimum response time and least maximum response time, while using **Closest Data Center** as Service Broker Policy and **Round Robin** as Load Balancing technique. Here is the table of response time of the same setup:

### Overall Response Time Summary

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	184.34	37.63	358.61
Data Center Processing Time:	0.31	0.01	0.88

## ❖ Observation & Learning:

We learned the Load Balancing topic of Cloud Computing. Also, learned about the simulator known as Cloud Analyst, which is used to simulate various load balancing algorithms and service broker policy. Also, learned about three widely used load balancing strategies (Round Robin, Equally Spread Current Execution Load, and Throttled). Using Cloud Analyst, one can also estimate various things, for example, the cost of VM's that are running on the datacenters for a particular period.

## ❖ Conclusion:

There are different Load Balancing Algorithms in Cloud Computing, and any of these algorithms can be used based on the requirements.

## ❖ Questions:

### ➤ Question - 1: What are regions and zones?

Regions are geographical areas, which are independent of each other. On the other hand, zones are parts of the region. In the cloud, regions, and zones are logical abstractions of underlying physical resources that are provided in one or more physical data centers.

### ➤ Question - 2: What are availability zones?

Availability Zones (AZs) are isolated locations within data centre regions from which public cloud services originate and operate.

### ➤ Question - 3: What are collocation facilities in cloud?

A collocation facility, also known as colo, is a data center facility in which a business can rent space for servers and other computing hardware. Typically, a colo provides the building, cooling, power, bandwidth and physical security.

# **Experiment – 7**

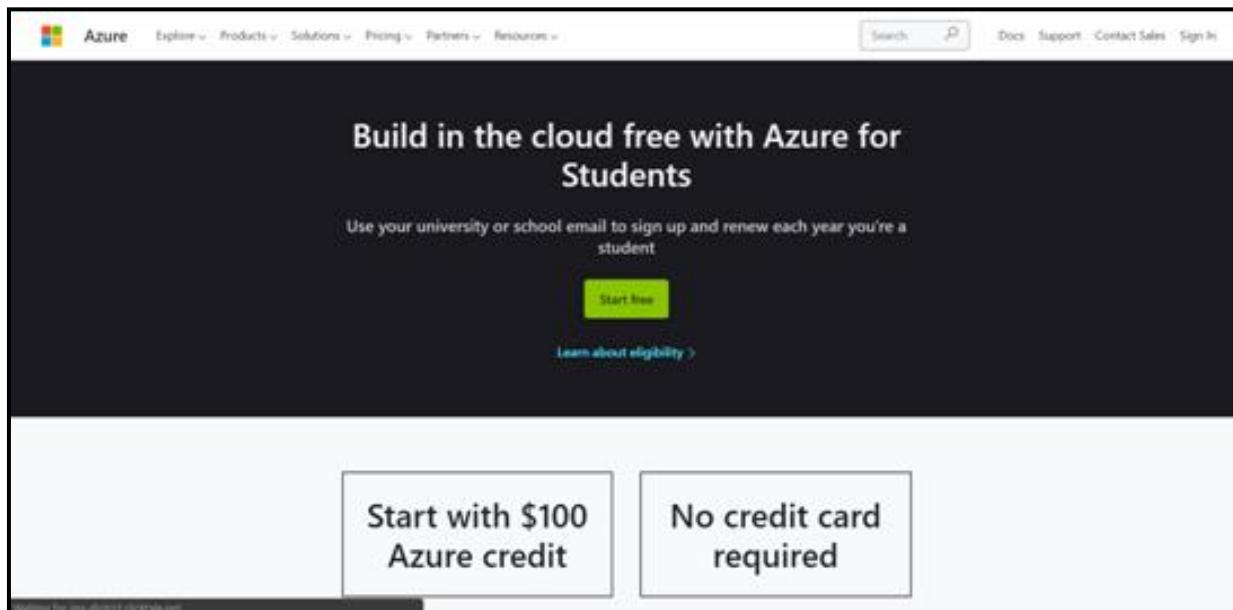
## **Microsoft Azure**

### **❖ Aim: To Explore Microsoft Azure.**

- Perform the following:
  8. Create a student account in Microsoft Azure
  9. Create a Virtual Machine with minimum configuration (Don't use the default suggested storage etc.)
    - a) Use Standard SSD option
    - b) Use password for authenticating the VM.
  10. Create a Resource Group and assign some resource to it
  11. Create a Virtual Network
  12. Create 3 subnets having 256 host in each subnet
  13. Create a virtual machine in each subnet and identify its public and private id address

### **5. Create a student account in Microsoft Azure:**

- Go to <https://azure.microsoft.com/en-in/free/students/> and create a free student account.



- Sign-in to create an account, after creating account you will be redirected to home page.

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a user profile. Below the header, there's a section for 'Azure services' with icons for Create a resource, Virtual machines, Quickstart Center, App Services, Storage accounts, SQL databases, Azure Cosmos DB, Kubernetes services, Function App, and All services. Under 'Navigate', there are links for Subscriptions, Resource groups, All resources, and Dashboard. In the 'Tools' section, there are links for Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. The 'Useful links' section includes Technical Documentation, Azure Services, and Recent Azure Updates. At the bottom right, there's an 'Azure mobile app' section with download links for the App Store and Google Play.

## 6. Create a Virtual Machine:

- Search for Virtual machine in search bar or select virtual machine from azure services.
- Select create from the top left navigation bar.

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The top navigation bar shows 'Home > Virtual machines'. The main title is 'Create a virtual machine'. Below it, there are tabs for Basics, Disks, Networking, Management, Advanced, Tags, and Review + create. The Basics tab is selected. A sub-section titled 'Project details' asks to select a subscription and resource group. The 'Subscription' dropdown is set to 'Azure for Students' and the 'Resource group' dropdown is set to '(New) Resource group' with a 'Create new' link. Below this, 'Instance details' include 'Virtual machine name' (empty), 'Region' (set to '(US) East US'), and 'Availability options' (set to 'No infrastructure redundancy required'). At the bottom, there are 'Review + create' and 'Next : Disks >' buttons.

- Enter VM name and select lowest size and create username and password for VM.

**Create a virtual machine**

Subscription: Azure for Students  
Resource group: (New) vedant138

Virtual machine name: vedantVM1

Region: (US) East US

Availability options: No infrastructure redundancy required

Security type: Standard

Image: Ubuntu Server 20.04 LTS - Gen2

Azure Spot instance:

Size: Standard\_D2s\_v3 - 2 vcpus, 8 GB memory (\$5,048.93/month)

Administrator account: [Review + create] < Previous Next : Disks >

- Select Standard SSD option in Disks.

**Create a virtual machine**

Basics Disks Networking Management Advanced Tags Review + create

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. [Learn more](#)

Disk options

OS disk type: Standard SSD (locally-redundant storage)  
If performance is critical for your workloads, choose Premium SSD disks for lower latency, higher IOPS and bandwidth, and bursting. [Learn more](#)

Delete with VM:

Encryption at host:

Encryption type: (Default) Encryption at-rest with a platform-managed key

Enable Ultra Disk compatibility:  Ultra disk is supported in Availability Zone(s) 1,2,3 for the selected VM size

Review + create < Previous Next : Networking >

- Go to review and create to create a VM.

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a user profile. Below it, the main title is "CreateVm-canonical.0001-com-ubuntu-server-focal-2-20220303135908 | Overview". On the left, a sidebar has tabs for "Overview", "Inputs", "Outputs", and "Template". The main content area says "Your deployment is complete" with deployment details like name, start time, and correlation ID. It also lists "Deployment details" (Download), "Next steps" (Setup auto-shutdown, Monitor VM health, Run a script), and two buttons: "Go to resource" and "Create another VM". To the right, there are promotional cards for "Cost Management", "Microsoft Defender for Cloud", and "Free Microsoft tutorials".

## 7. Create a Resource Group and assign some resources in it:

- Select Resource Group from navigate bar or search resource group in search bar and then click on create.
- Add name for resource group.

The screenshot shows the Microsoft Azure portal interface for creating a new resource group. The left sidebar shows "Resource groups" with a list of existing groups: "NetworkWatcherRG" and "vedant138". The main panel is titled "Create a resource group" and contains tabs for "Basics", "Tags", and "Review + create". Under "Project details", "Subscription" is set to "Azure for Students" and "Resource group" is a field where "vedant138" is typed. Under "Resource details", "Region" is set to "(US) East US". At the bottom, there are navigation buttons for "Page 1 of 1", "Review + create", "Previous", and "Next : Tags >".

- Create resource group.

Validation passed.

**Basics**

Subscription	Azure for Students
Resource group	RGPractice
Region	East US

**Tags**

None

**Create** | **< Previous** | **Next >** | Download a template for automation

- Assigned resources in resource group.

**vedant138** | Resource group

**Overview**

Subscription (move)  
Azure for Students

Subscription ID  
172d668d-61e1-42b0-b192-be424bf834dc

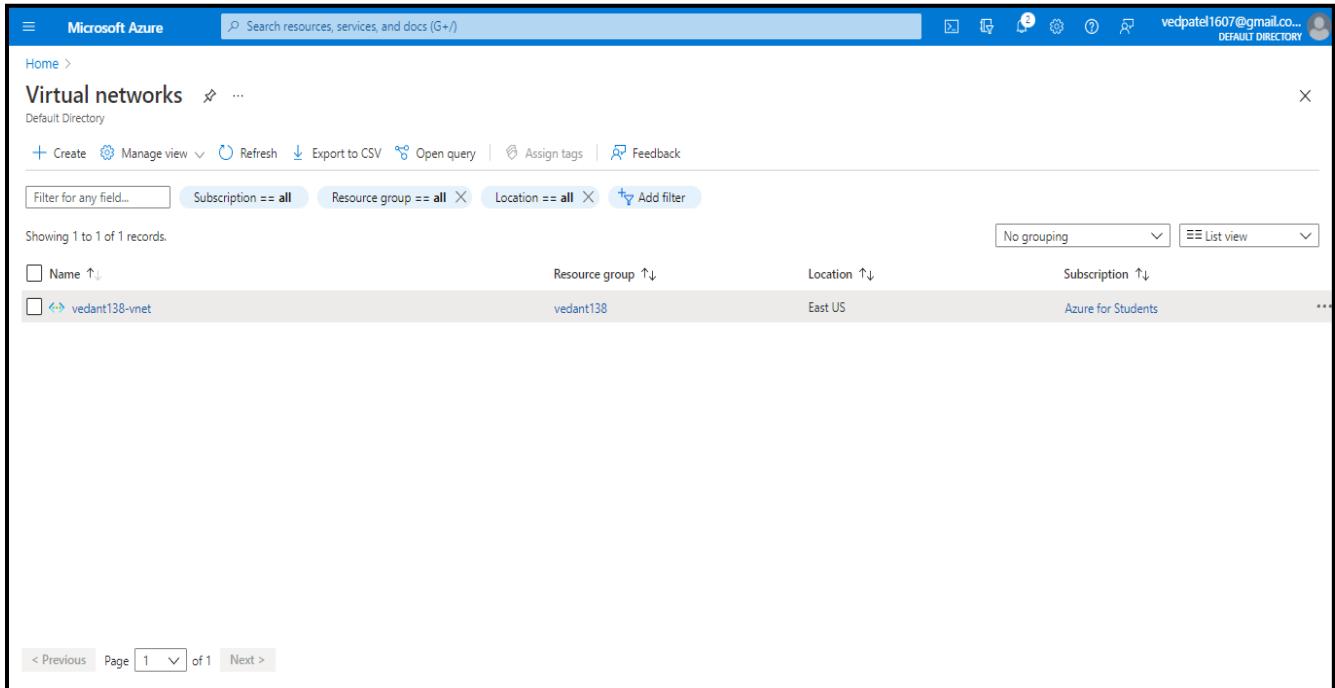
Tags (edit)  
Click here to add tags

**Resources**

Name	Type	Location
vedant138-vnet	Virtual network	East US
vedantVM1	Virtual machine	East US
vedantVM1-ip	Public IP address	East US
vedantVM1-nsg	Network security group	East US
vedantvm1598	Network interface	East US

## 8. Create a Virtual Network:

- Open Virtual Network and create a virtual network.

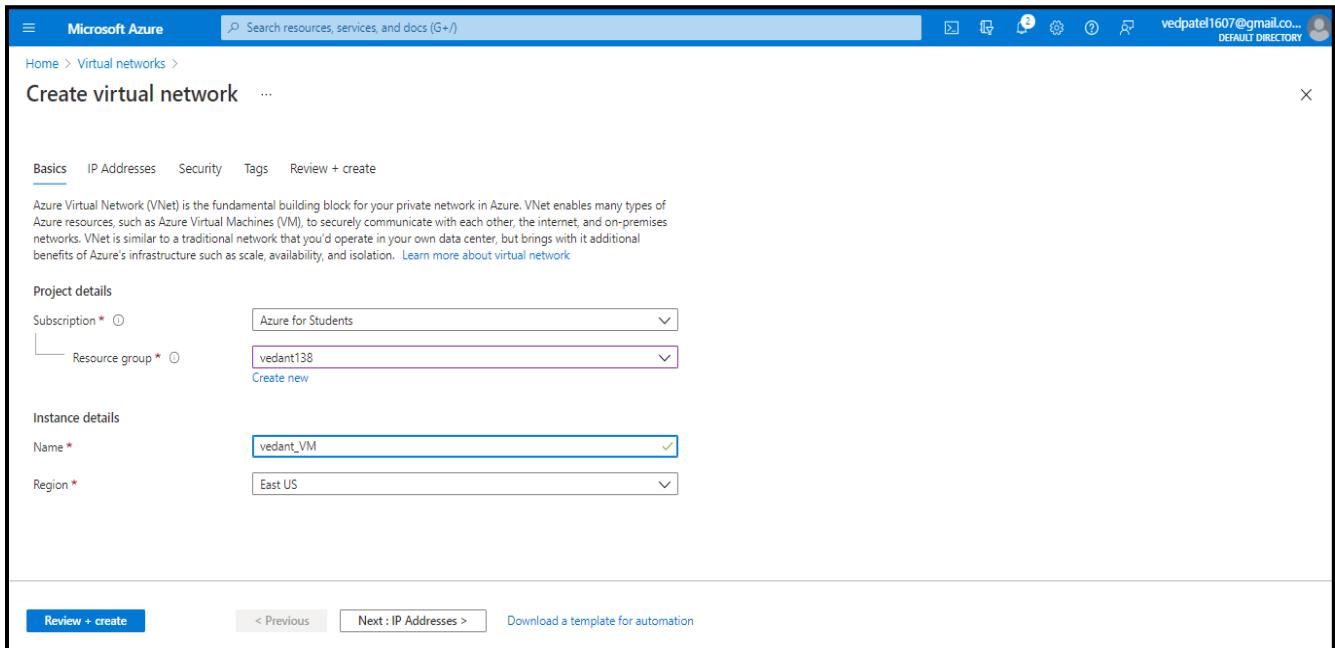


The screenshot shows the Microsoft Azure portal interface for managing Virtual networks. At the top, there's a search bar and a navigation bar with options like 'Create', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', 'Assign tags', and 'Feedback'. Below the header, the 'Virtual networks' section is displayed with a single item listed:

Name	Resource group	Location	Subscription
vedant138-vnet	vedant138	East US	Azure for Students

At the bottom of the page, there are navigation links for 'Page 1 of 1' and 'Next >'. The URL in the address bar is [https://portal.azure.com/#blade/HubsBlade/resourceType/virtualNetworks/resource/vedant138-vnet](#).

- Add name for virtual network.



The screenshot shows the 'Create virtual network' wizard in the Microsoft Azure portal, specifically on the 'Basics' step. The title bar says 'Create virtual network'.

The 'Basics' tab is selected, while other tabs like 'IP Addresses', 'Security', and 'Tags' are available. Below the tabs, a descriptive text explains what a Virtual Network (VNet) is and its benefits.

**Project details**

- Subscription: Azure for Students
- Resource group: vedant138 (selected)

**Instance details**

- Name: vedant\_VM
- Region: East US

At the bottom, there are buttons for 'Review + create' (highlighted in blue), '< Previous', 'Next : IP Addresses >', and 'Download a template for automation'.

## 9. Create 3 subnets having 256 hosts in each Subnet:

- Click on add subnet and add 3 subnets.

The screenshot shows the 'Create virtual network' wizard in the Microsoft Azure portal. The 'IPv4 address space' field is set to 10.1.0.0/16 (10.1.0.0 - 10.1.255.255 (65536 addresses)). Three subnets are listed: subnet1 (10.1.0.0/24), subnet2 (10.1.1.0/24), and subnet3 (10.1.2.0/24). A note indicates that a NAT gateway is recommended for outbound internet access. Navigation buttons at the bottom include 'Review + create', '< Previous', 'Next : Security >', and 'Download a template for automation'.

- Click on create to finally create virtual network.

The screenshot shows the 'Create virtual network' wizard in the Microsoft Azure portal. The validation has passed. The 'Review + create' tab is selected, showing the following details:

- Basics**: Subscription (Azure for Students), Resource group (vedant138), Name (vedant\_VM), Region (East US).
- IP addresses**: Address space (10.1.0.0/16), Subnet (subnet1 (10.1.0.0/24), subnet2 (10.1.1.0/24), subnet3 (10.1.2.0/24)).
- Tags**: None.

Navigation buttons at the bottom include 'Create', '< Previous', 'Next >', and 'Download a template for automation'.

## 10. VM for each subnet and its public and private id address:

- Click on add subnet and add 3 subnets.

The screenshot shows the 'Create a virtual machine' wizard in the Microsoft Azure portal. The 'Networking' section is being configured. A dropdown menu for 'Subnet' shows 'subnet1 (10.1.0.0/24)' selected. Other subnets listed are 'subnet1 (10.1.0.0/24)', 'subnet2 (10.1.1.0/24)', and 'subnet3 (10.1.2.0/24)'. The 'Public inbound ports' section has 'Allow selected ports' selected with 'SSH (22)' chosen. The 'Virtual network' dropdown shows 'vedant\_VM'.

- VM for Sub 1.

Virtual machine		Networking	
Computer name	vedantVMsub1	Public IP address	vedantVMsub1-ip
Health state	-	Public IP address (IPv6)	-
Operating system	Linux	Private IP address	10.1.0.4
Publisher	canonical	Private IP address (IPv6)	-
Offer	0001-com-ubuntu-server-focal	Virtual network/subnet	vedant_VM/subnet1
Plan	20_04-lts-gen2	DNS name	Configure
VM generation	V2		

- VM for Sub 2.

Virtual machine		Networking	
Computer name	vedantVMsub2	Public IP address	vedantVMsub2ip427
Health state	-	Public IP address (IPv6)	-
Operating system	Linux	Private IP address	10.1.1.5
Publisher	canonical	Private IP address (IPv6)	-
Offer	0001-com-ubuntu-server-focal	Virtual network/subnet	vedant_VM/subnet2
Plan	20_04-lts-gen2	DNS name	Configure
VM generation	V2		

- VM for Sub 3.

Virtual machine		Networking	
Computer name	vedantVMsub3	Public IP address	vedantVMsub2ip427
Health state	-	Public IP address (IPv6)	-
Operating system	Linux	Private IP address	10.1.1.5
Publisher	canonical	Private IP address (IPv6)	-
Offer	0001-com-ubuntu-server-focal	Virtual network/subnet	vedant_VM/subnet3
Plan	20_04-lts-gen2	DNS name	<a href="#">Configure</a>
VM generation	V2		

## **Experiment – 8**

Simulate a cloud scenario using CloudSim (<http://cloudbus.org/cloudsim/>.)

❖ **Aim:** To Simulate Cloud Environment by adding datacenters and cloudlets of different Topology.

➤ **Perform the following:**

14. Create a datacenter with one host and run one cloudlet on it.
15. Create two datacenters with one host and a network topology each and run two cloudlets on them.
16. Create two datacenters with one host each and run cloudlets of two users with network topology on them.
17. Create two datacenters with one host each and run two cloudlets on them.
18. Create two datacenters with one host each and run cloudlets of two users on them.
19. Pause and resume the simulation, and create simulation entities (a Datacenter Broker in this example) dynamically.
20. \*Perform VM migration between Datacenters considering them as different cloud providers.
  - a. Manually
  - b. Considering CPU Utilization threshold as criterion for VM migration.
  - c. When Simulation clock reaches at specific time.

\*Note: (You may use CloudSim Plus for VM Migration)

❖ **Introduction:**

**CloudSim** is a framework for modelling and simulation of cloud computing infrastructures and services. Originally built primarily at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, the University of Melbourne, Australia, CloudSim has become one of the most popular open-source cloud simulators in the research and academia. CloudSim is completely written in Java. CloudSim comprises huge volume of libraries for simulating the cloud-based computing systems. It offers necessary classes for creating users, applications, virtual machines, data centers, computational resources and policies for controlling the different parts of the system as provisioning and scheduling.

## ❖ Download CloudSim:

First of all go to the <http://cloudbus.org/cloudsim/> website and there you will see the screen as below:

The screenshot shows the homepage of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne. The header features the laboratory's name in blue and red text, and a logo for the CLOUDS LAB with the word 'MELBOURNE' above it. A vertical sidebar on the left contains a navigation menu with links such as Blog, Introduction, Research Probes, Team & Collaborators, Publications, Annual and Tech Reports, Dissertations, Presentations, Cloudbus Middleware, Gridbus Middleware, Applications, Grid Courseware, Master Degree in Distributed Computing, Melbourne-Chindia Cloud Network, Join us! Opportunities, In Press and News, Researcher Links, Sponsors, and Stay informed!. The main content area is titled 'CloudSim: A Framework For Modeling And Simulation Of Cloud Computing Infrastructures And Services'. It includes an 'Introduction' section with text about the evolution of cloud computing and its challenges, followed by a 'Main features' section.

Scroll down until the webpage as mentioned below and then click on GitHub link as mentioned below and it will open the new page of GitHub Page.

The screenshot shows the 'Documentation' section of the CloudSim website. It lists various resources including an online course, examples, release notes, installation instructions, a changelog, and container information. Below this is the 'Download' section, which provides a link to the CloudSim package on GitHub: <https://github.com/Cloudslab/cloudsim/releases>. The page also includes a note about the code being from a specific paper and links to a discussion group and related projects.

Click on the link named as “cloudsim-4.0.zip” as shown below in the Image.

May 24, 2016

cloudsim-4.0  
2d8f1c6

Compare ▾

## CloudSim 4.0 Latest

### Changes from CloudSim 3.0.3 to CloudSim 4.0

#### WHAT'S NEW

- added support for Container virtualization
- lots of bugfixes

#### Assets 4

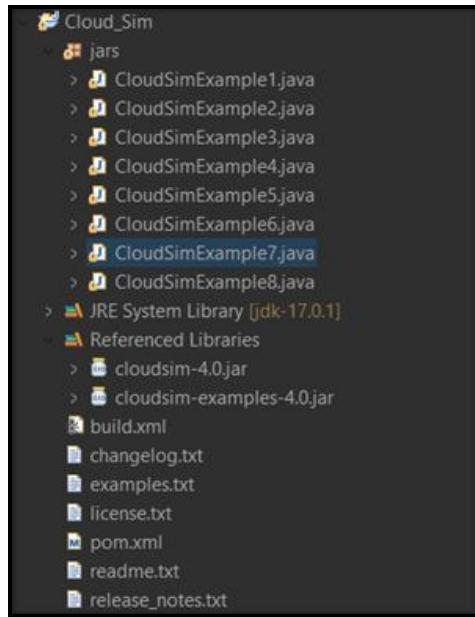
<a href="#">cloudsim-4.0.tar.gz</a>	3.47 MB
<a href="#">cloudsim-4.0.zip</a>	3.48 MB
<a href="#">Source code (zip)</a>	
<a href="#">Source code (tar.gz)</a>	

🕒 8 🌟 1 8 people reacted

The following files will be downloaded as shown in the Below Image.

Name	Date modified	Type	Size
jars	19-02-2022 08:51 PM	File folder	
build	24-05-2016 01:34 PM	XML Document	3 KB
changelog	24-05-2016 01:34 PM	Text Document	12 KB
examples	24-05-2016 01:34 PM	Text Document	5 KB
license	24-05-2016 01:34 PM	Text Document	9 KB
pom	24-05-2016 01:34 PM	XML Document	4 KB
readme	24-05-2016 01:34 PM	Text Document	3 KB
release_notes	24-05-2016 01:34 PM	Text Document	3 KB

After downloading these files, open the Eclipse Editor and then create a new Java Project and open the downloaded files and also add them to the Source Folder as shown in the below Image.



Once all of the above steps are completed, we just need to create different classes in the package jars and run those Java programs by taking the code from the GitHub Link mentioned below.

## ➤ Screenshots:

1. Create a datacenter with one host and run one cloudlet on it.

### Code 1 Link:

<https://github.com/Cloudslab/cloudsim/blob/master/modules/cloudsim-examples/src/main/java/org/cloudbus/cloudsim/examples/CloudSimExample1.java>

```

Problems Javadoc Declaration Console
terminated> CloudSimExample1 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (19-Feb-2022, 8:39:39 pm - 8:39:40 pm)
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!

```

- 2.** Create two datacenters with one host and a network topology each and run two cloudlets on them. **Code 2 Link:**

<https://github.com/Cloudslab/cloudsim/blob/master/modules/cloudsim-examples/src/main/java/org/cloudbus/cloudsim/examples/CloudSimExample2.java>

```

Problems Javadoc Declaration Console ×
<terminated> CloudSimExample2 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (19-Feb-2022, 8:40:11 pm – 8:40:12 pm)
Starting CloudSimExample2...
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
0:0: Broker: Trying to Create VM #1 in Datacenter_0
0:1: Broker: VM #0 has been created in Datacenter #2, Host #0
0:1: Broker: VM #1 has been created in Datacenter #2, Host #0
0:1: Broker: Sending cloudlet 0 to VM #0
0:1: Broker: Sending cloudlet 1 to VM #1
1000:1: Broker: Cloudlet 0 received
1000:1: Broker: Cloudlet 1 received
1000:1: Broker: All Cloudlets executed. Finishing...
1000:1: Broker: Destroying VM #0
1000:1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
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 1000 0.1 1000.1
1 SUCCESS 2 1 1000 0.1 1000.1
CloudSimExample2 finished!

```

- 3.** Create two datacenters with one host each and run cloudlets of two users with network topology on them. **Code 3 Link:**

<https://github.com/Cloudslab/cloudsim/blob/master/modules/cloudsim-examples/src/main/java/org/cloudbus/cloudsim/examples/CloudSimExample3.java>

```

Problems Javadoc Declaration Console ×
<terminated> CloudSimExample3 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (19-Feb-2022, 8:40:36 pm – 8:40:37 pm)
Starting CloudSimExample3...
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
0:0: Broker: Trying to Create VM #1 in Datacenter_0
0:1: Broker: VM #0 has been created in Datacenter #2, Host #0
0:1: Broker: VM #1 has been created in Datacenter #2, Host #1
0:1: Broker: Sending cloudlet 0 to VM #0
0:1: Broker: Sending cloudlet 1 to VM #1
0:1: Broker: Cloudlet 1 received
160:1: Broker: Cloudlet 0 received
160:1: Broker: All Cloudlets executed. Finishing...
160:1: Broker: Destroying VM #0
160:1: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.

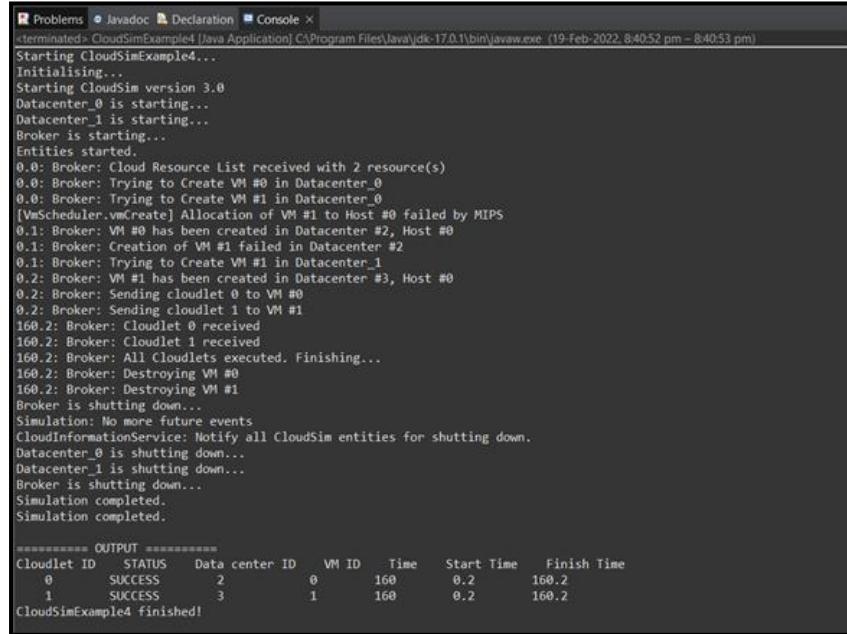
===== OUTPUT =====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
1 SUCCESS 2 1 80 0.1 80.1
0 SUCCESS 2 0 160 0.1 160.1
CloudSimExample3 finished!

```

#### 4. Create two datacenters with one host each and run two cloudlets on them.

##### Code 4 Link:

<https://github.com/Cloudslab/cloudsim/blob/master/modules/cloudsim-examples/src/main/java/org/cloudbus/cloudsim/examples/CloudSimExample4.java>

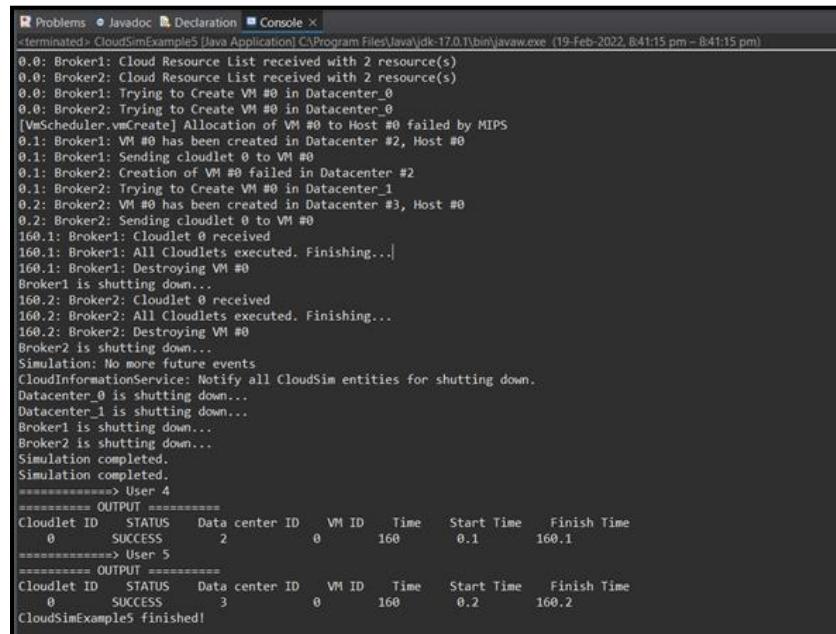


```
Starting CloudSimExample4...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #1 to Host #0 failed by MIPS
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: Creation of VM #1 failed in Datacenter #2
0.1: Broker: Trying to Create VM #1 in Datacenter_1
0.2: Broker: VM #1 has been created in Datacenter #3, Host #0
0.2: Broker: Sending cloudlet 0 to VM #0
0.2: Broker: Sending cloudlet 1 to VM #1
160.2: Broker: Cloudlet 0 received
160.2: Broker: Cloudlet 1 received
160.2: Broker: All Cloudlets executed. Finishing...
160.2: Broker: Destroying VM #0
160.2: Broker: Destroying VM #1
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.

=====
CloudSimExample4 finished!
```

#### 5. Create two datacenters with one host each and run cloudlets of two users on them. **Code 5 Link:**

<https://github.com/Cloudslab/cloudsim/blob/master/modules/cloudsim-examples/src/main/java/org/cloudbus/cloudsim/examples/CloudSimExample5.java>



```
Starting CloudSimExample5...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker1 is starting...
Broker2 is starting...
Entities started.
0.0: Broker1: Cloud Resource List received with 2 resource(s)
0.0: Broker2: Cloud Resource List received with 2 resource(s)
0.0: Broker1: Trying to Create VM #0 in Datacenter_0
0.0: Broker2: Trying to Create VM #0 in Datacenter_0
[VmScheduler.vmCreate] Allocation of VM #0 to Host #0 failed by MIPS
0.1: Broker1: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker1: Sending cloudlet 0 to VM #0
0.1: Broker2: Creation of VM #0 failed in Datacenter #2
0.1: Broker2: Trying to Create VM #0 in Datacenter_1
0.2: Broker2: VM #0 has been created in Datacenter #3, Host #0
0.2: Broker2: Sending cloudlet 0 to VM #0
160.1: Broker1: Cloudlet 0 received
160.1: Broker1: All Cloudlets executed. Finishing...
160.1: Broker1: Destroying VM #0
Broker1 is shutting down...
160.2: Broker2: Cloudlet 0 received
160.2: Broker2: All Cloudlets executed. Finishing...
160.2: Broker2: Destroying VM #0
Broker2 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker1 is shutting down...
Broker2 is shutting down...
Simulation completed.

=====> User 4
=====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 160 0.1 160.1
=====> User 5
=====
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 3 0 160 0.2 160.2
CloudSimExample5 finished!
```

**6.** Pause and resume the simulation, and create simulation entities (a Datacenter Broker in this example) dynamically. **Code 6 Link:**

<https://github.com/Cloudslab/cloudsim/blob/master/modules/cloudsim-examples/src/main/java/org/cloudbus/cloudsim/examples/CloudSimExample7.java>

```
Problems Javadoc Declaration Console 
terminated> CloudSimExample7 [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (19-Feb-2022, 8:42:10 pm – 8:42:17 pm)
519.996: Broker_1: Cloudlet 108 received
519.996: Broker_1: Cloudlet 100 received
519.996: Broker_1: Cloudlet 105 received
519.996: Broker_1: Cloudlet 102 received
519.996: Broker_1: Cloudlet 107 received
519.996: Broker_1: Cloudlet 104 received
519.996: Broker_1: Cloudlet 109 received
519.996: Broker_1: All Cloudlets executed. Finishing...
519.996: Broker_1: Destroying VM #100
519.996: Broker_1: Destroying VM #101
519.996: Broker_1: Destroying VM #102
519.996: Broker_1: Destroying VM #103
519.996: Broker_1: Destroying VM #104
Broker_1 is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker_0 is shutting down...
Broker_1 is shutting down...
Simulation completed.
Simulation completed.

***** OUTPUT *****
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time
0 SUCCESS 2 0 320 0.1 320.1
5 SUCCESS 2 0 320 0.1 320.1
1 SUCCESS 2 1 320 0.1 320.1
6 SUCCESS 2 1 320 0.1 320.1
2 SUCCESS 2 2 320 0.1 320.1
7 SUCCESS 2 2 320 0.1 320.1
4 SUCCESS 2 4 320 0.1 320.1
9 SUCCESS 2 4 320 0.1 320.1
3 SUCCESS 2 3 320 0.1 320.1
8 SUCCESS 2 3 320 0.1 320.1
CloudSimExample7 finished!
```