

CSE 3035

PRINCIPLES OF CLOUD COMPUTING



Lab Assessment – 1

L15+L16 | SJT501
Dr. Sivaprakash S

FALL SEMESTER 2022-23

by

SHARADINDU ADHIKARI
19BCE2105

Assessment 1 .

Experiment 1

Title: VM config (Local & Cloud).

Aim: To Create, Configure, and access a Virtual machine (VM) on my local machine and Cloud with the basic resources. Allocate CPU, memory, and storage space as per a specified requirement. Install the Guest OS image in that instance, launch the same and confirm the successful installation of the OS by performing a few OS commands in the Terminal.

Background Theory:

Virtualization:

- Uses software to create an abstraction layer over computer hardware that allows the hardware elements of a single computer—processors, memory, storage and more—to be divided into multiple virtual computers, commonly called virtual machines (VMs).
- Each VM runs its own operating system (OS) and behaves like an independent computer, even though it is running on just a portion of the actual underlying computer hardware.

Virtual Machine:

- Virtual machines (VMs) are virtual environments that simulate physical computing in software form.
- They normally comprise several files containing the VM's configuration, the storage for the virtual hard drive, and some snapshots of the VM that preserve its state at a particular point in time.

Hypervisors:

- A hypervisor is the software layer that coordinates VMs.
- It serves as an interface between the VM and the underlying physical hardware, ensuring that each has access to the physical resources it needs to execute.
- It also ensures that the VMs don't interfere with each other by impinging on each other's memory space or compute cycles.

Hypervisor Types:

- Type 1 or “bare-metal” hypervisors interact with the underlying physical resources, replacing the traditional operating system altogether. They most commonly appear in virtual server scenarios.
- Type 2 hypervisors run as an application on an existing OS. Most commonly used on endpoint devices to run alternative operating systems, they carry a performance overhead because they must use the host OS to access and coordinate the underlying hardware resources.

Requirements for conducting the experiment:

Local Instance:

- Oracle VM VirtualBox (or)
- VMware Workstation Player (or)
- VMware Workstation Pro (or)
- Any other Virtual Machine Setup
- Any OS ISO Image (preferably Ubuntu)

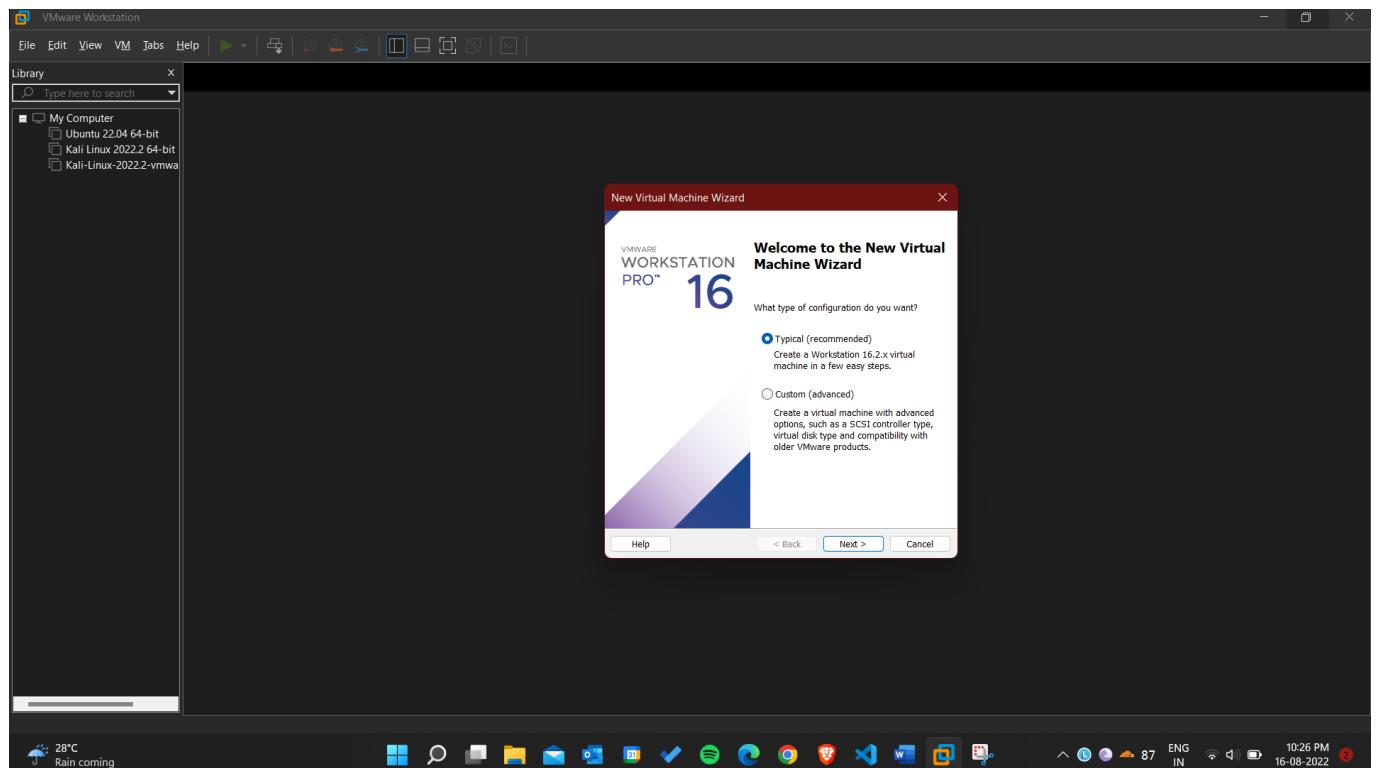
Cloud:

- AWS Licence

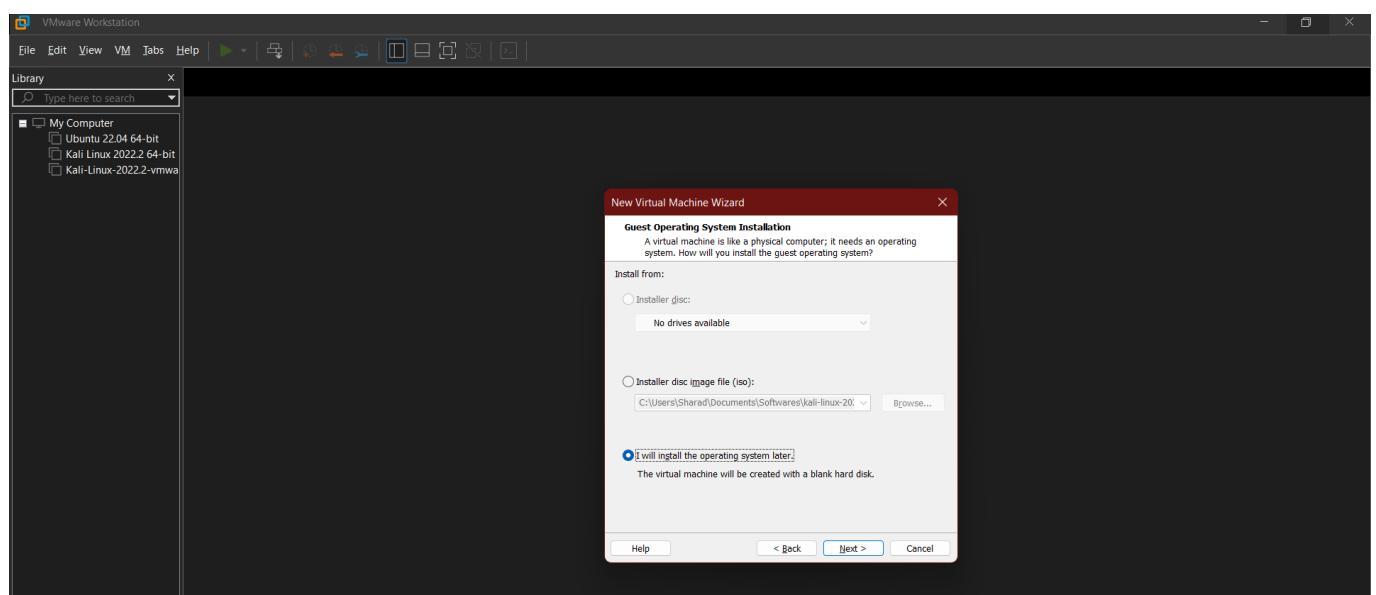
Procedure & Screenshots:

(1) Local Instance:

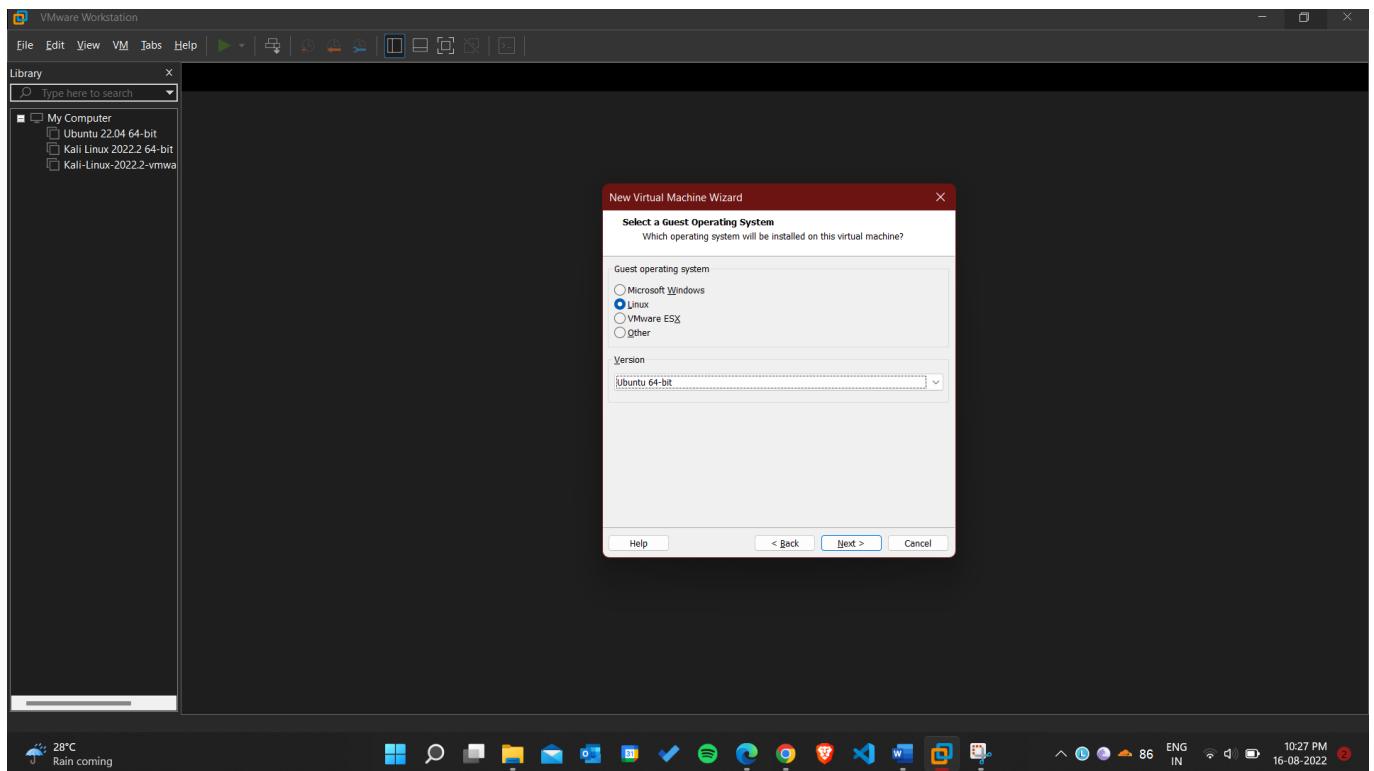
Step 1: Type of Installation → Typical, where we do not go into the SCSI Controller type, Virtual Disk Type etc.



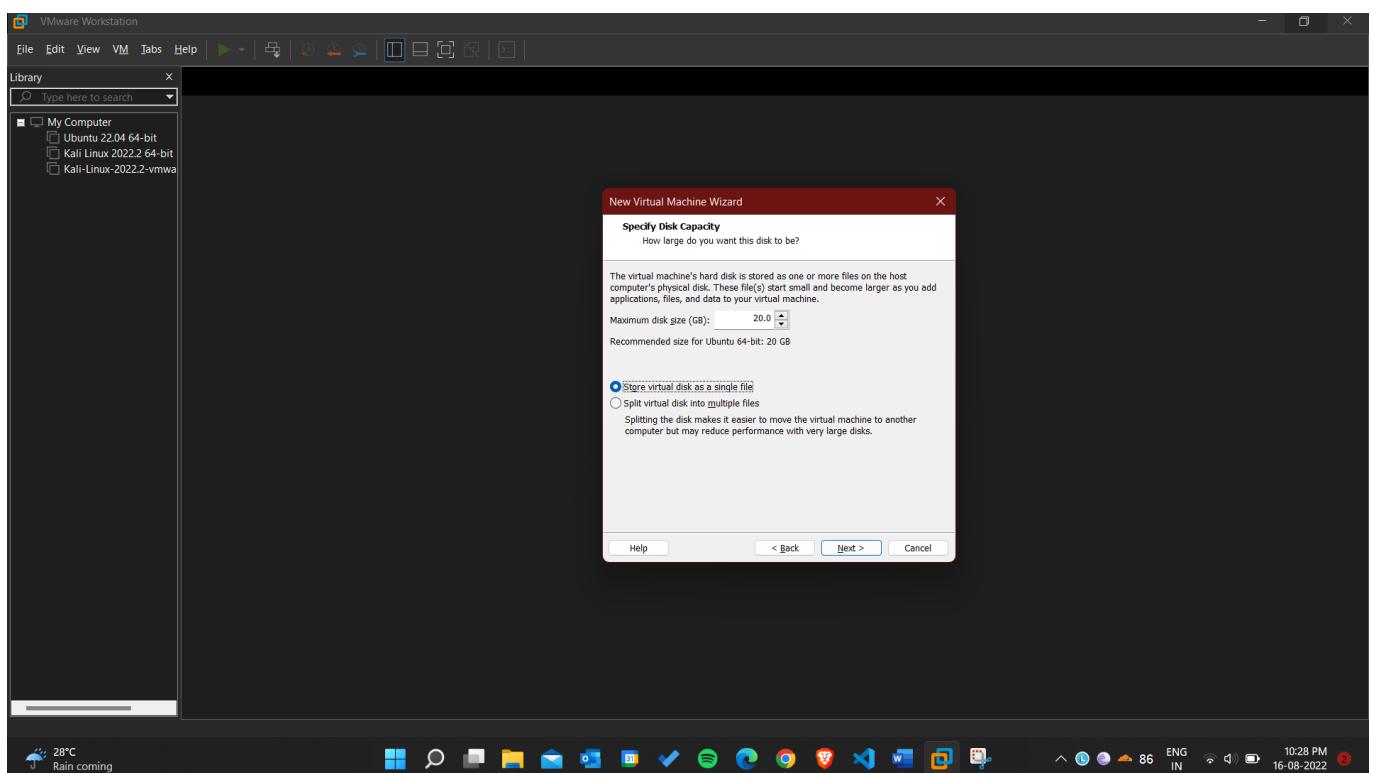
Step 2: We skip adding the iso file, for now, just creating the virtual machine setup.



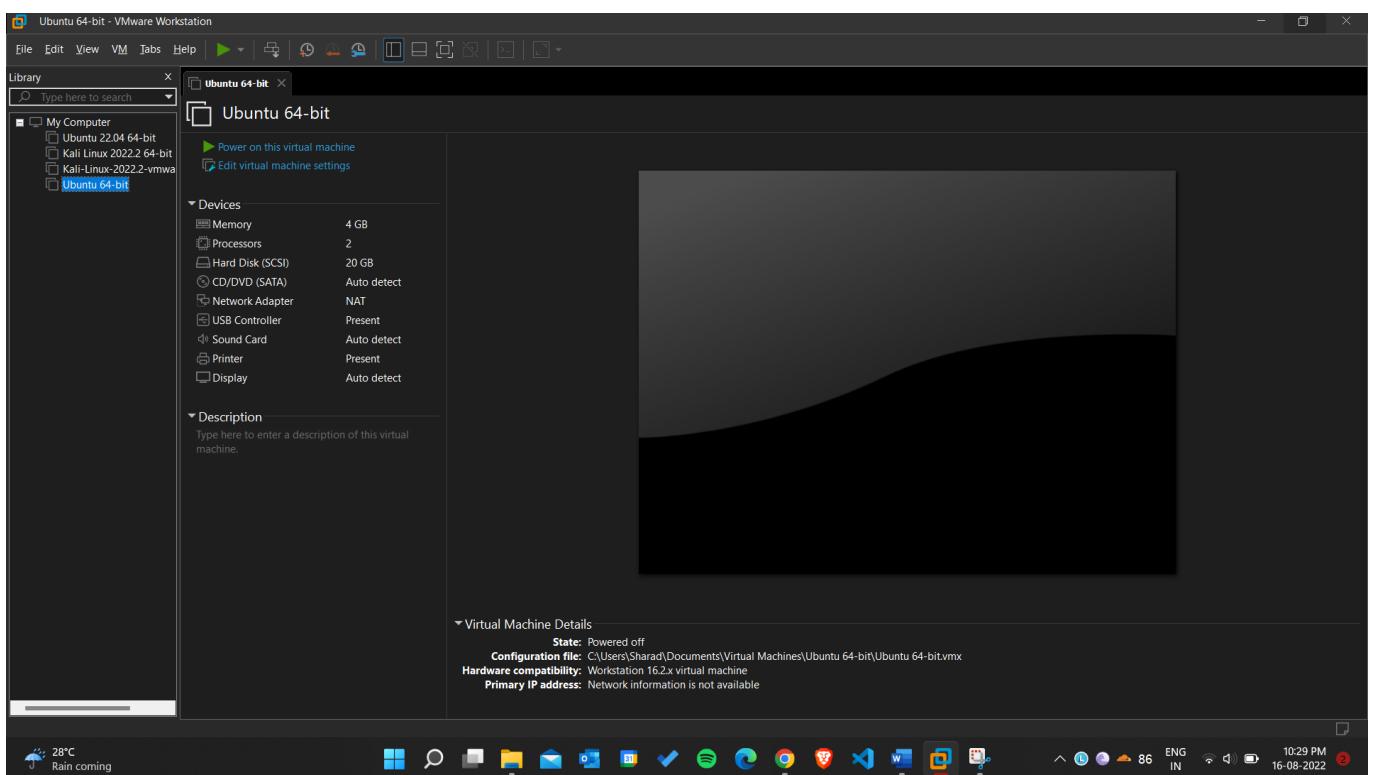
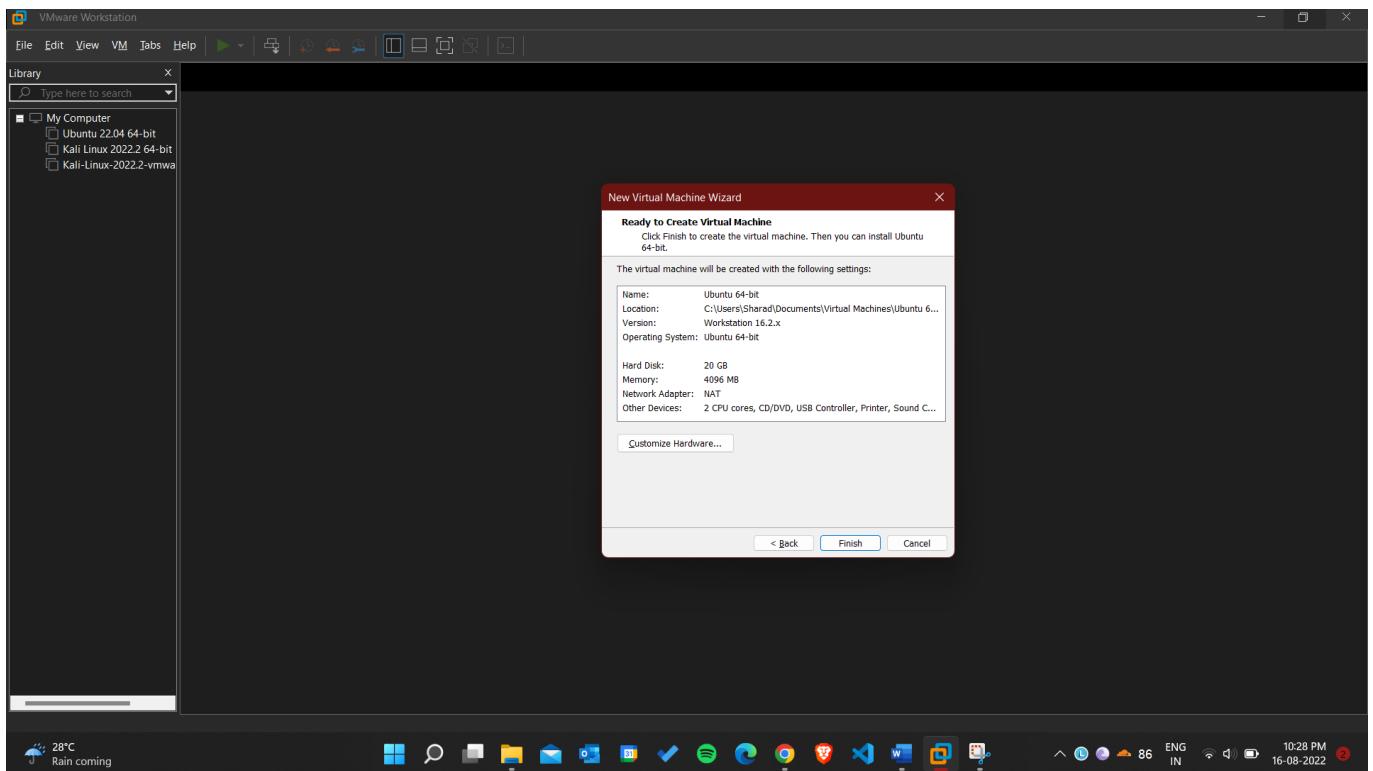
Step 3: Select the type of OS we're planning on installing.



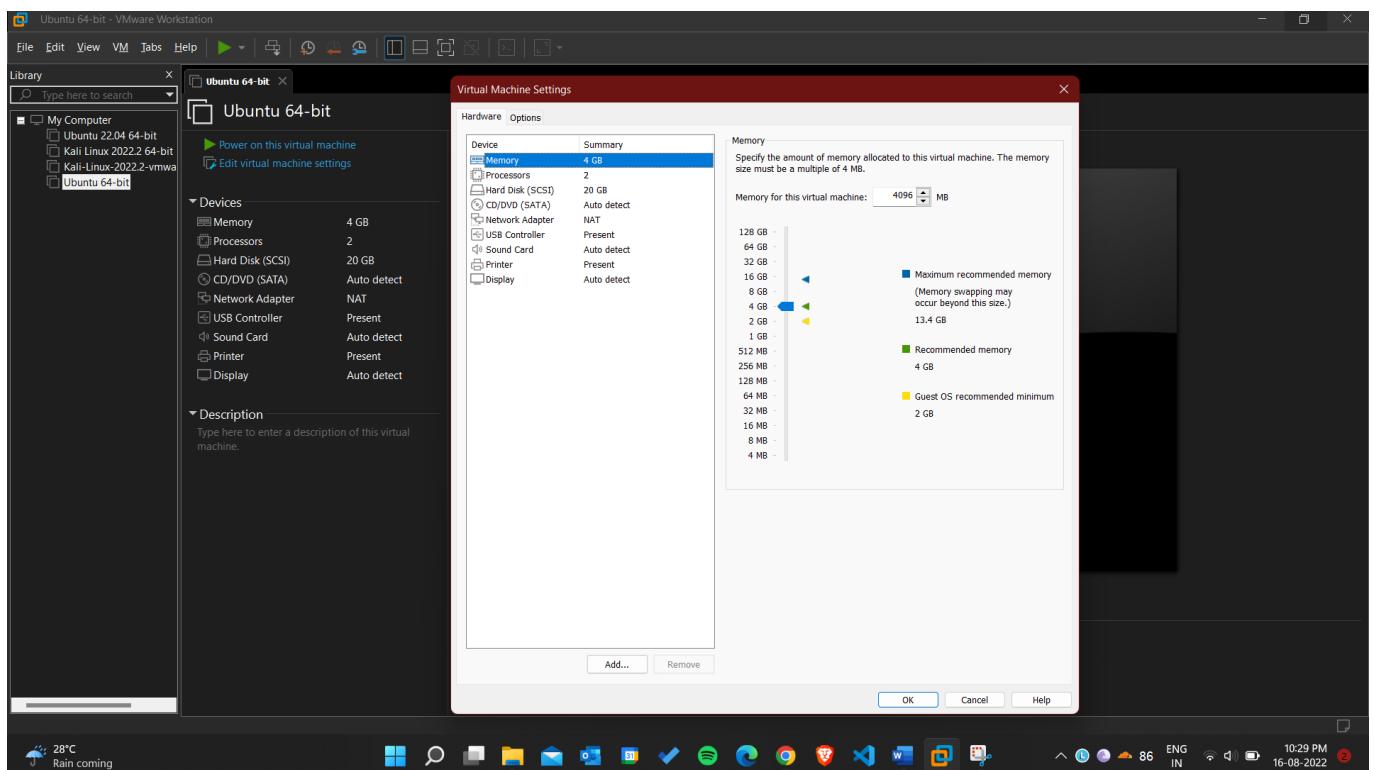
Step 4: Select the VM disk storage type.



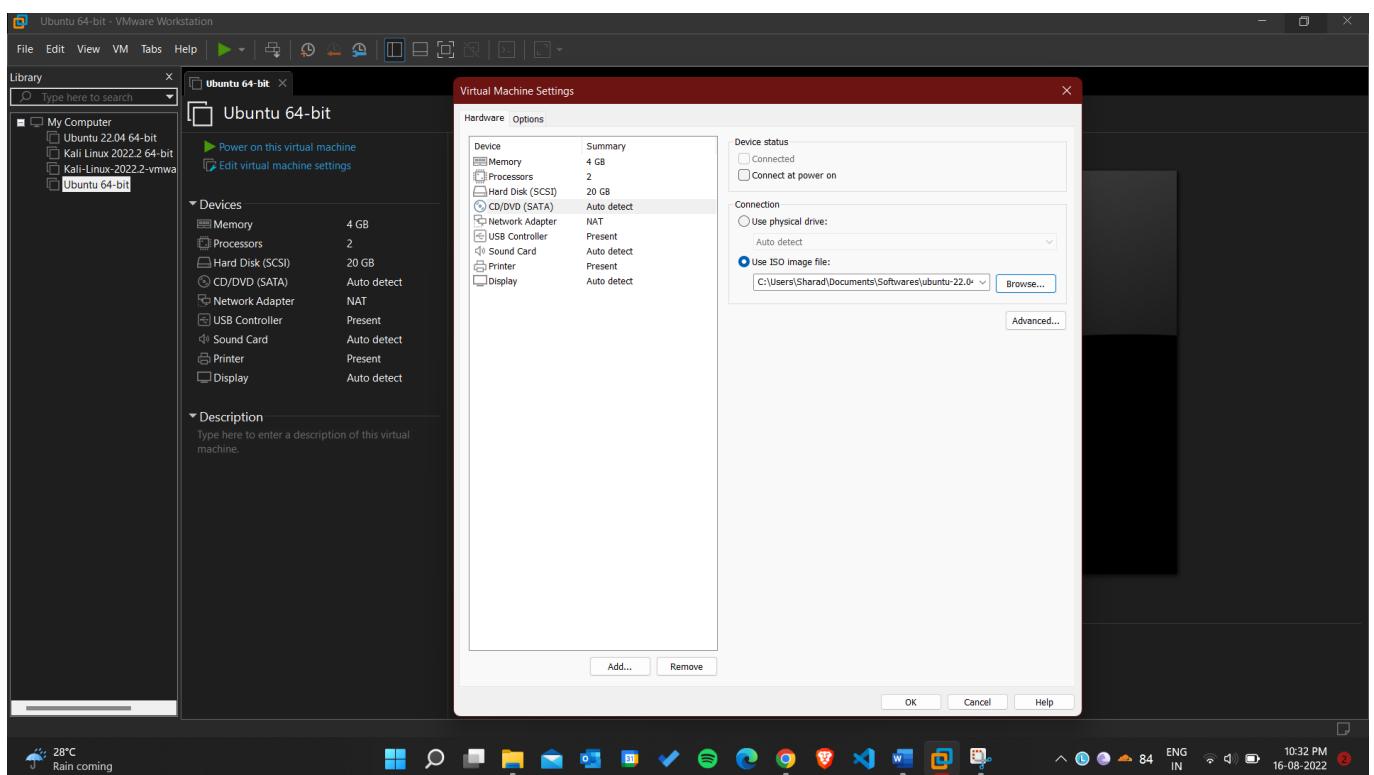
Step 5: Finish setting up the VM (for now).



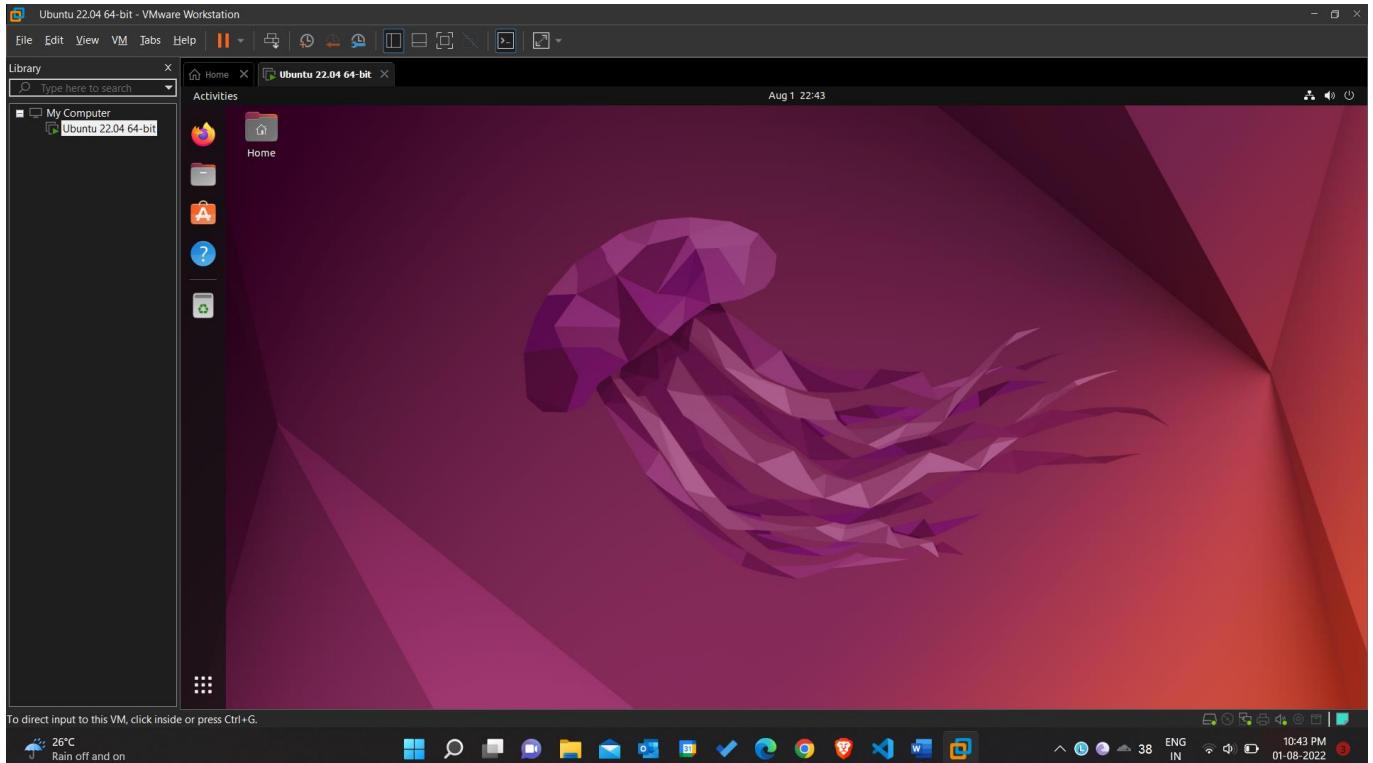
Step 6: Editing the VM configurations



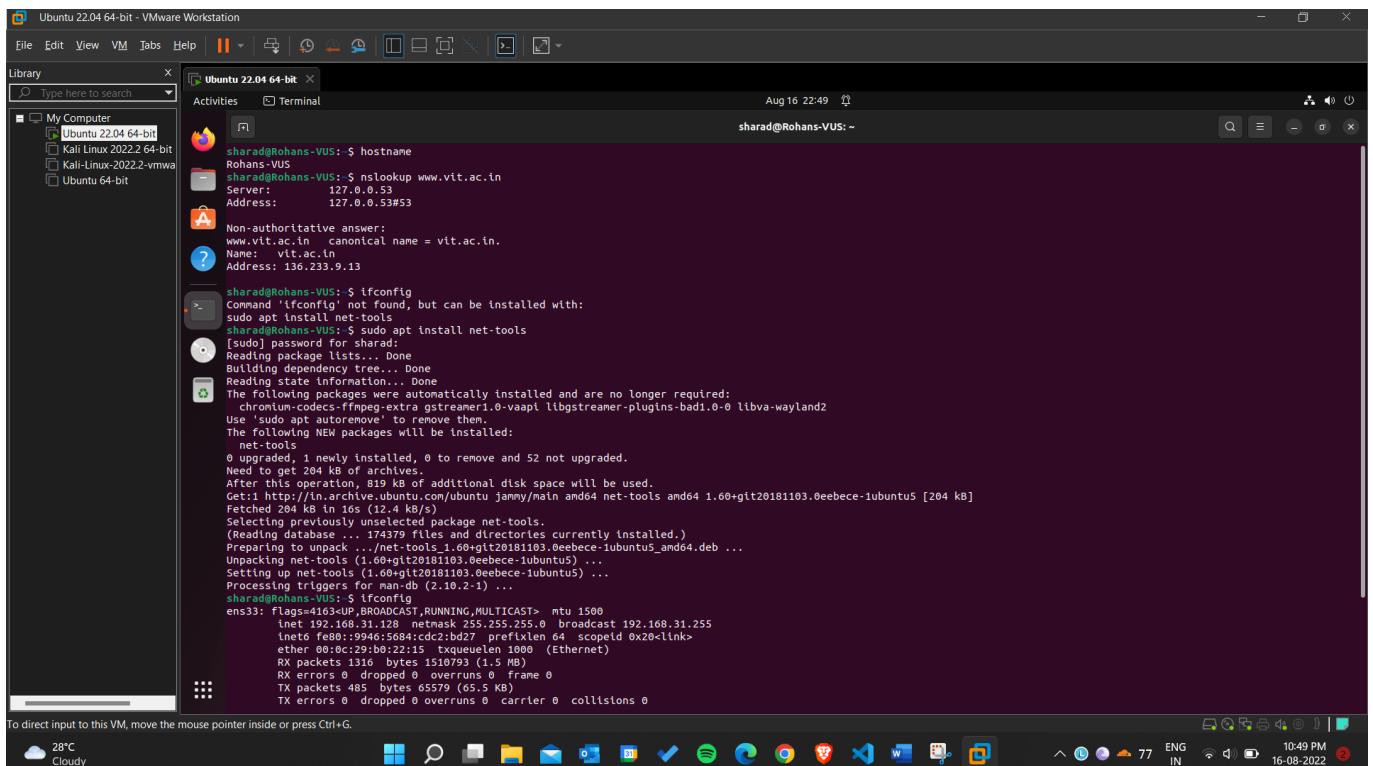
Step 7: Adding the iso file to install the OS.



Step 8: Install the OS (Ubuntu here). Then boot it up.



Step 9: Running basic Linux commands in the Terminal.



(2) Cloud:

Step 1: Create an Instance with the following configuration: (a) ubuntu 64-bit architecture, (b) type: t2.micro, (c) 8 gib, gpu2 root volume

The screenshot shows the 'Launch an instance' wizard on the AWS EC2 console. The configuration is as follows:

- Name and tags:** Name is set to 'POCC Assessment'.
- Software Image (AMI):** Canonical, Ubuntu, 22.04 LTS, ami-068257025f72f470d
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB
- Free tier information:** A tooltip indicates that the first year includes 750 hours of t2.micro or t3.micro usage in specific regions.
- Buttons:** 'Cancel' and 'Launch instance'.

The screenshot shows the 'Launch an instance' wizard with the following configuration:

- Application and OS Images (Amazon Machine Image):** Ubuntu Server 22.04 LTS (HVM), SSD Volume Type, ami-068257025f72f470d (64-bit (x86)) / ami-02b401fea53d5bc05 (64-bit (Arm))
- Description:** Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2022-06-09
- Architecture:** 64-bit (x86)
- AMI ID:** ami-068257025f72f470d
- Free tier eligible:** A dropdown menu shows 'Free tier eligible'.
- Summary:** Shows the same configuration as the previous screenshot, including the free tier information.
- Buttons:** 'Cancel' and 'Launch instance'.

The screenshot shows the AWS Lambda console interface. A modal window titled 'Create Lambda Function' is open, prompting for the function name 'HelloWorld'. The 'Code' tab is selected, showing the Lambda@Edge code for the function. The code is as follows:

```

function handler(event, context) {
    const response = {
        statusCode: 200,
        headers: {
            'Content-Type': 'text/plain'
        },
        body: 'Hello, world from Lambda!' + JSON.stringify(event)
    };
    context.succeed(response);
}

```

The 'Function code' section contains the same code. Below the code, there are sections for 'Environment variables', 'Triggers', and 'Logs'. The 'Logs' section shows a log entry with the timestamp '2022-08-01T11:44:59+00:00' and the message 'Hello, world from Lambda! {>}'.

The screenshot shows the AWS Lambda console interface. A modal window titled 'Create Lambda Function' is open, prompting for the function name 'HelloWorld'. The 'Code' tab is selected, showing the Lambda@Edge code for the function. The code is as follows:

```

function handler(event, context) {
    const response = {
        statusCode: 200,
        headers: {
            'Content-Type': 'text/plain'
        },
        body: 'Hello, world from Lambda!' + JSON.stringify(event)
    };
    context.succeed(response);
}

```

The 'Function code' section contains the same code. Below the code, there are sections for 'Environment variables', 'Triggers', and 'Logs'. The 'Logs' section shows a log entry with the timestamp '2022-08-01T11:44:59+00:00' and the message 'Hello, world from Lambda! {>}.'

To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info

Advanced

1x 8 GiB gp2 Root volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

The selected AMI contains more instance store volumes than the instance allows. Only the first 0 instance store volumes from the AMI will be accessible from the instance

0 x File systems Edit

Advanced details Info

Summary

Number of instances Info

1

Virtual server type Instance type

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel **Launch instance**

Step 2: Launch cloud instance

Instances (2) Info

Search

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
#pocc #exp1	i-057613d3ff744a20e	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-2-129-
-	i-03db1d6f7d2229a83	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-43-205-11

Select an instance

Launch instances

New EC2 Experience Tell us what you think

- EC2 Dashboard
- EC2 Global View
- Events
- Tags
- Limits
- Instances
 - Instances** New
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances New
 - Dedicated Hosts
 - Capacity Reservations
- Images
 - AMIs New
 - AMI Catalog
- Elastic Block Store

Feedback Looking for language selection? Find it in the new Unified Settings Info

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

Cloudy 28°C 10:56 PM 75 ENG IN 16-08-2022

Instance summary for i-057613d3ff744a20e (#pocc #exp1)

Instance ID	Public IPv4 address	Private IPv4 addresses
i-057613d3ff744a20e (#pocc #exp1)	65.2.129.169 open address	172.31.41.184
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-65-2-129-169.ap-south-1.compute.amazonaws.com open address
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-41-184.ap-south-1.compute.internal	ip-172-31-41-184.ap-south-1.compute.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPv4 (A)	t2.micro	Opt-in to AWS Compute Optimizer for recommendation s.
Auto-assigned IP address	VPC ID	Learn more
65.2.129.169 [Public IP]	vpc-0c06adf23f1fe1f67	Auto Scaling Group name
IAM Role	Subnet ID	-
-	subnet-0c53008b735ea3d86	

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance details | Info

Conclusion:

I've created a virtual machine setup on my local host computer, and another on the cloud (in AWS). All of the steps are chronologically depicted, and screenshots at each milestone are added.

Experiment 2

Title: Nested Virtual Machines

Aim: To install the KVM emulator (VM Manager) and Configure a Nested Virtual Machine (VM under another VM) in the cloud and local machine. To Install OS images and work with a few OS commands.

Background Theory:

Virtualization Technology:

- Virtualization relies on software to simulate hardware functionality and create a virtual computer system. This enables IT organizations to run more than one virtual system – and multiple operating systems and applications – on a single server. The resulting benefits include economies of scale and greater efficiency.

Intel® VT:

- Intel® Virtualization Technology (Intel® VT) represents a growing portfolio of technologies and features that make virtualization practical by eliminating performance overheads and improving security. Intel® Virtualization Technology (Intel® VT) provides hardware assist to the virtualization software, reducing its size, cost, and complexity. Special attention is also given to reduce the virtualization overheads occurring in cache, I/O, and memory.
- Over the last decade or so, a significant number of hypervisor vendors, solution developers, and users have been enabled with Intel® Virtualization Technology (Intel® VT), which is now serving a broad range of customers in the consumer, enterprise, cloud, communication, technical computing, and many more sectors.

AMD-V®:

- AMD virtualization (AMD-V®) is a virtualization technology developed by Advanced Micro Devices. AMD-V technology takes some of the tasks that virtual machine managers perform through software emulation and simplifies those tasks through enhancements in the processor's instruction set.
- AMD-V® was first announced in 2004 and added to AMD's Pacifica 64-bit x86 processor designs. By 2006, AMD's Athlon 64 X2 and Athlon 64 FX processors appeared with AMD-V technology, and today, the technology is available on Turion 64 X2, second- and third-generation Opteron, Phenom, and Phenom II processors.

KVM Tool:

- KVM (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on hardware containing virtualization extensions. It consists of a loadable kernel module that exposes virtualization APIs to userspace for use by applications such as QEMU.
- Specifically, KVM lets you turn Linux into a hypervisor that allows a host machine to run multiple, isolated virtual environments called guests or virtual machines (VMs).

Requirements for conducting the experiment:

Local Instance:

- Oracle VM VirtualBox (or)
- VMware Workstation Player (or)
- VMware Workstation Pro (or)
- Any other Virtual Machine Setup
- Any OS ISO Image (preferably Ubuntu)

- Virtual Machine with Guest Operating System

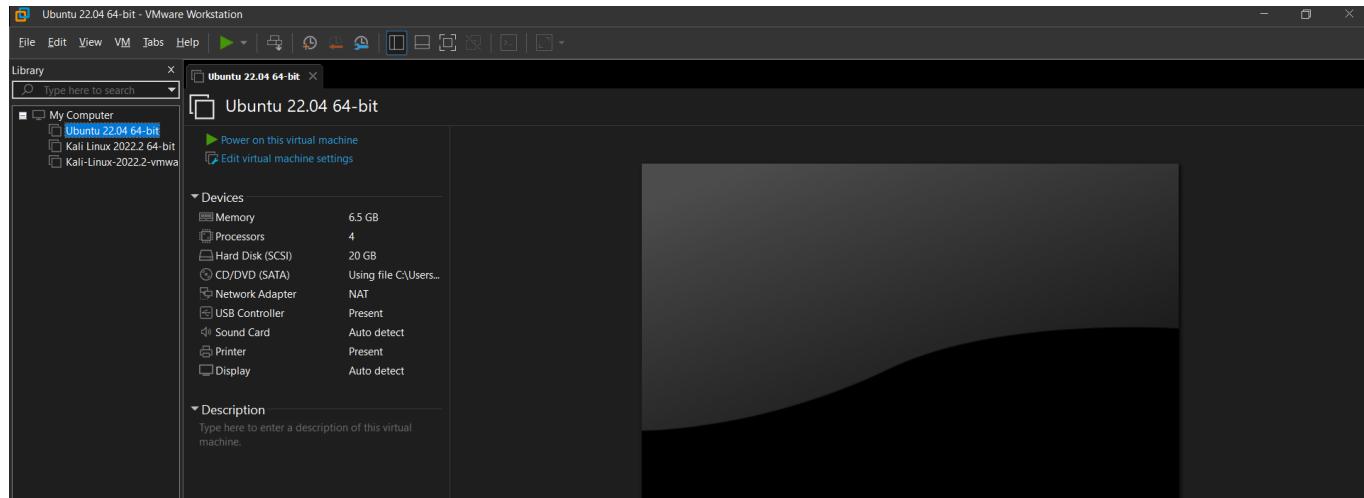
Cloud:

- AWS Licence

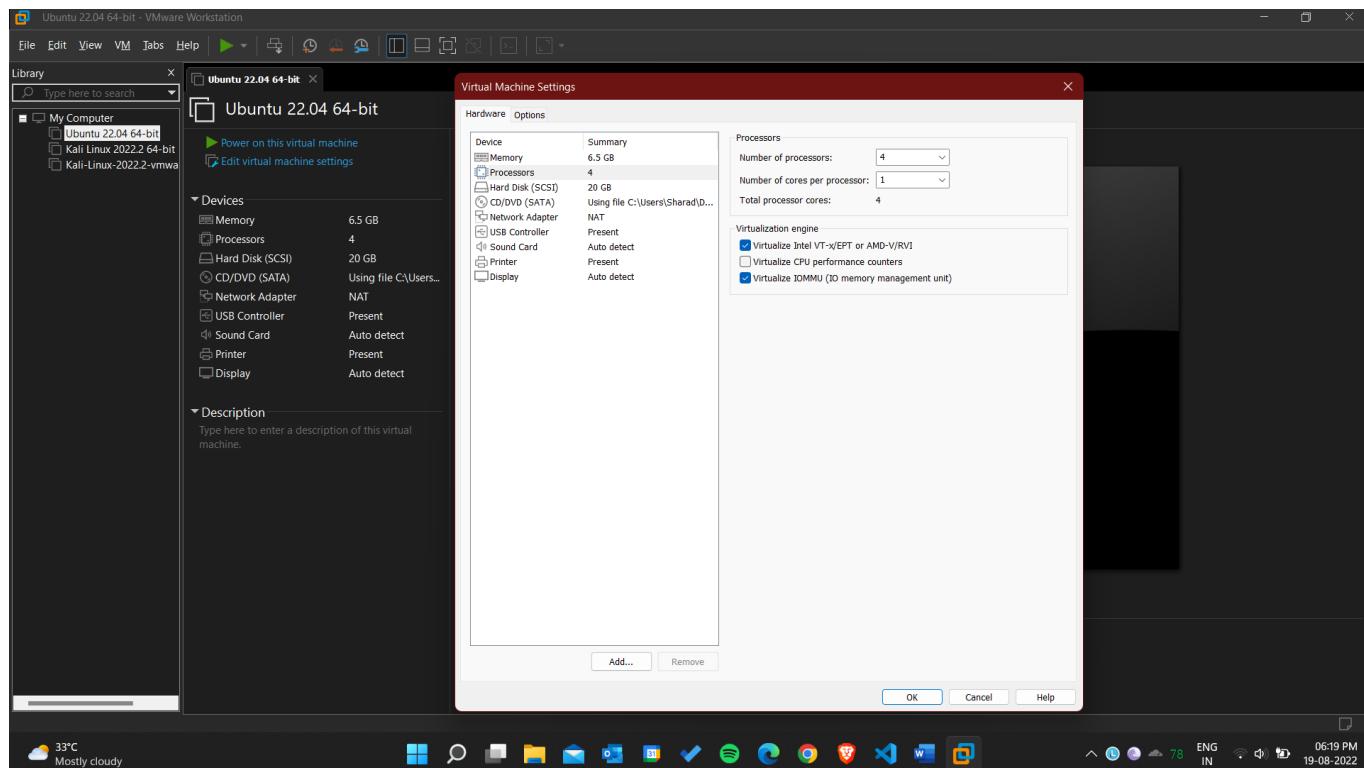
Procedure & Screenshots:

(1) Local Instance:

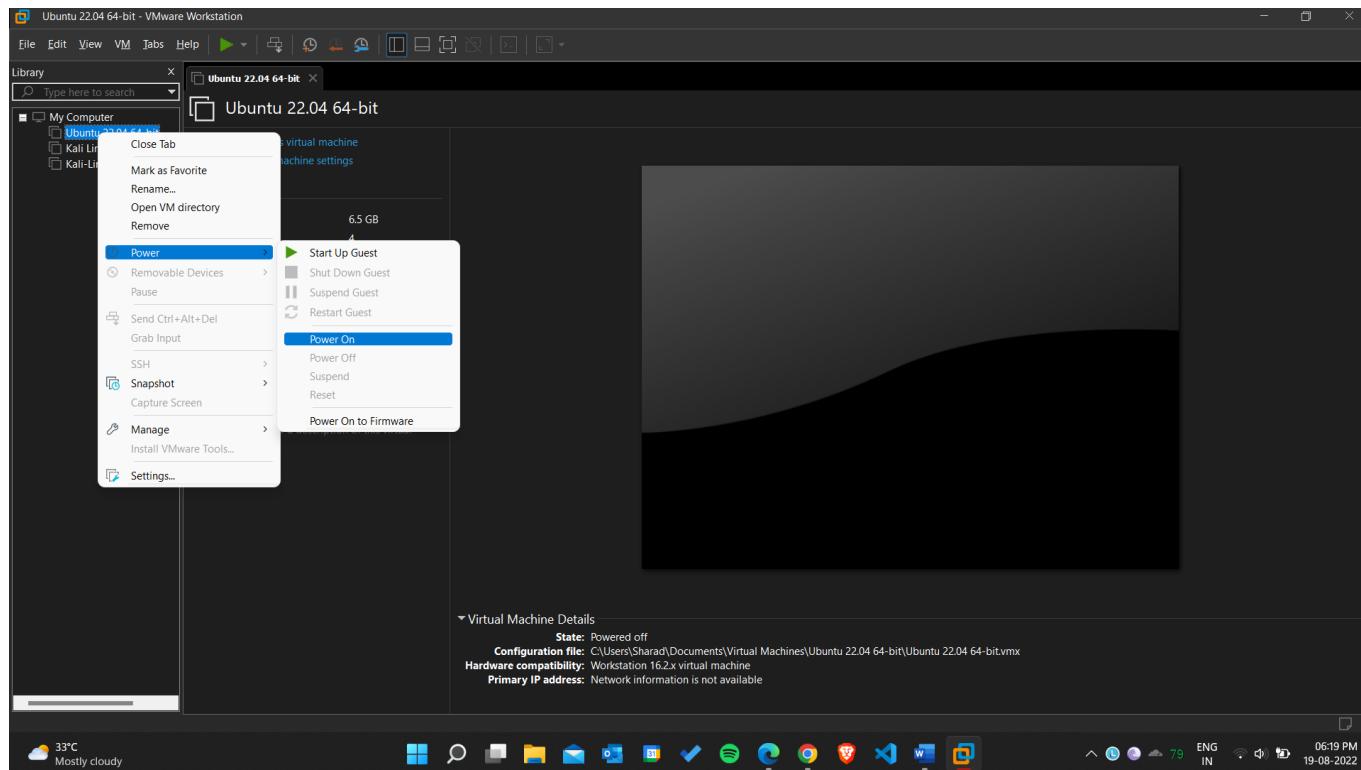
Step 1: Open VMWare Workstation Pro and select the Virtual Machine (here, Ubuntu 22.04 64-bit which I'd installed in Experiment 1)



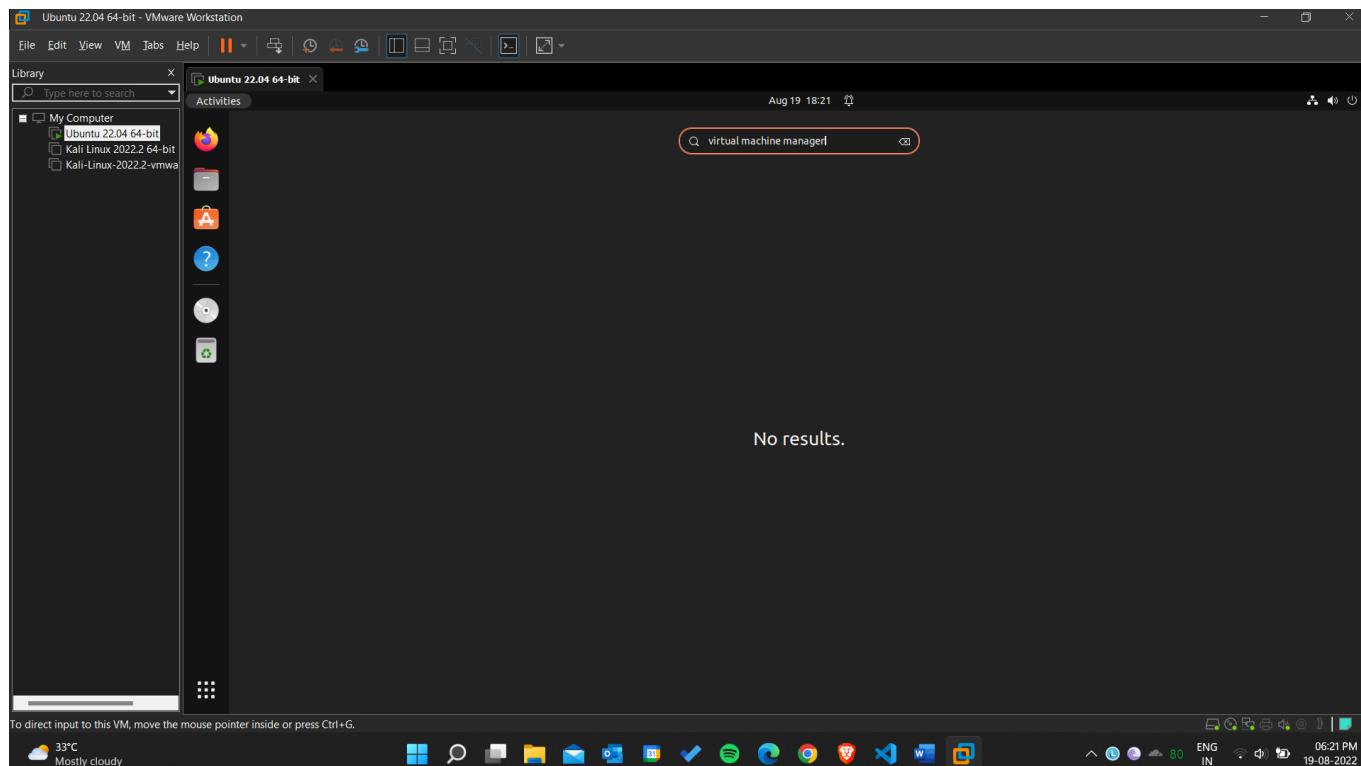
Step 2: Then go to VM (at the top bar) → Settings → Processors, and enable the options that are under Virtualization engine ‘Virtualize Intel VT-x/EPT or AMD-V/RVI’ and ‘Virtualize IOMMU’ and then click on ‘OK’.



Step 3: Now after enabling the options ‘Power On’ the OS.

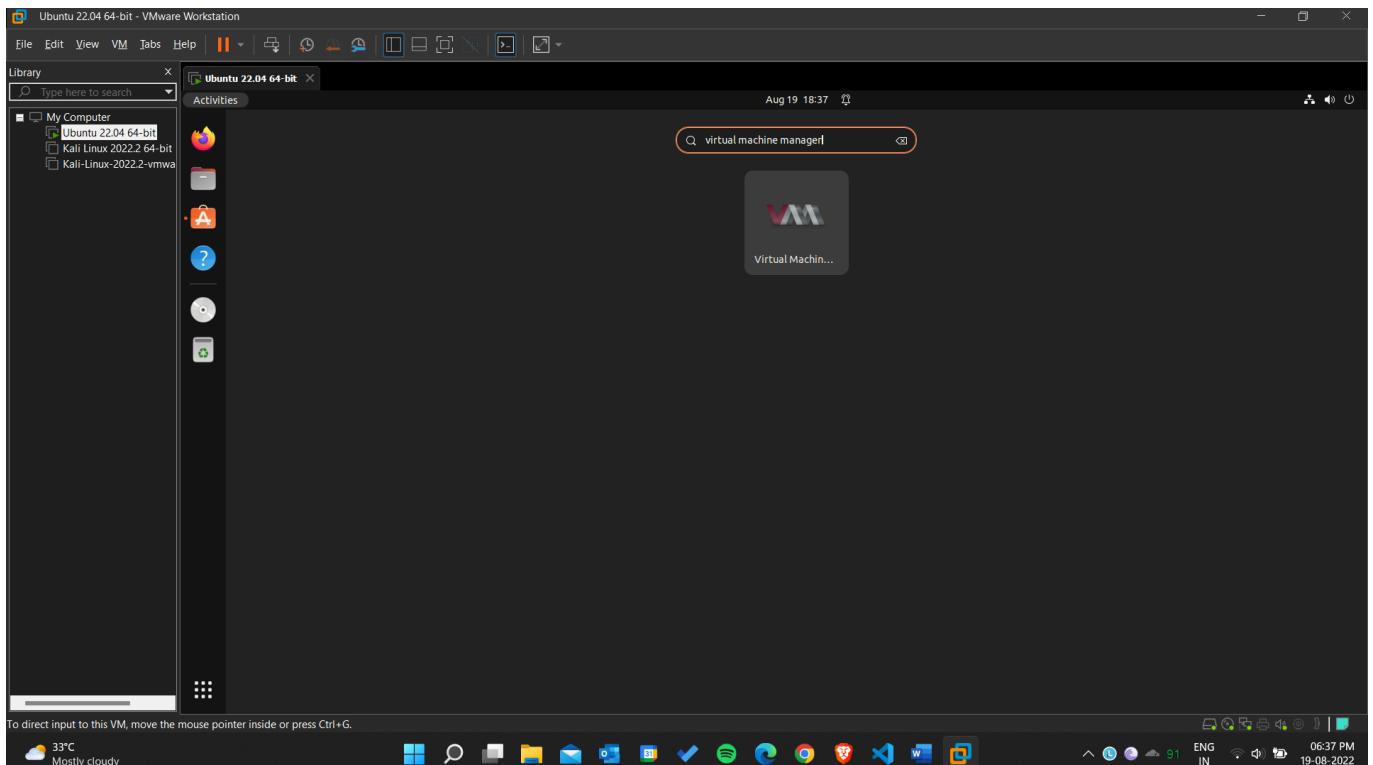


Step 4: Search for ‘virtual machine manager’. If it is not installed go to the terminal.

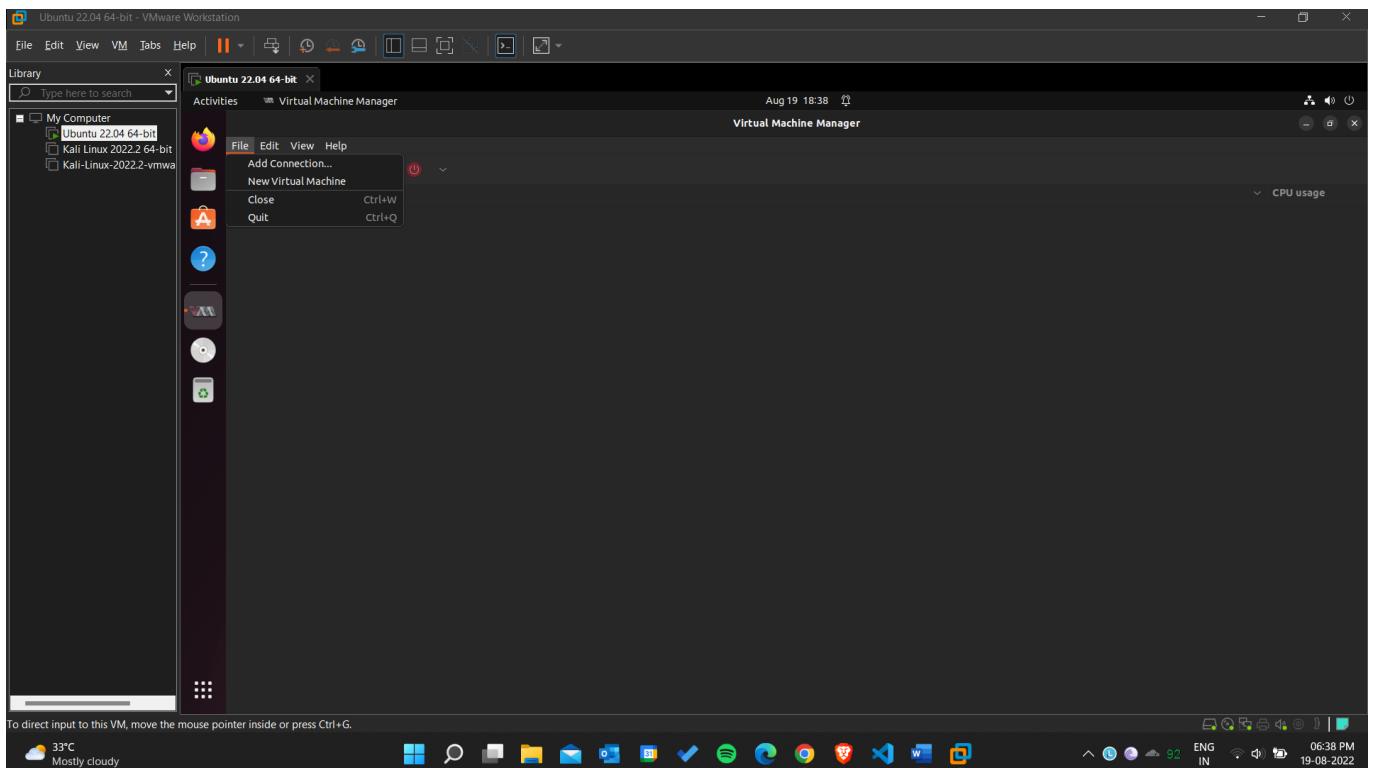


Step 5: Type in ‘sudo apt-get install virt-manager’ to install the virtual machine manager. If this way, it shows any error, go to Applications → Ubuntu Software Center → Search for "virt-manager" & install.

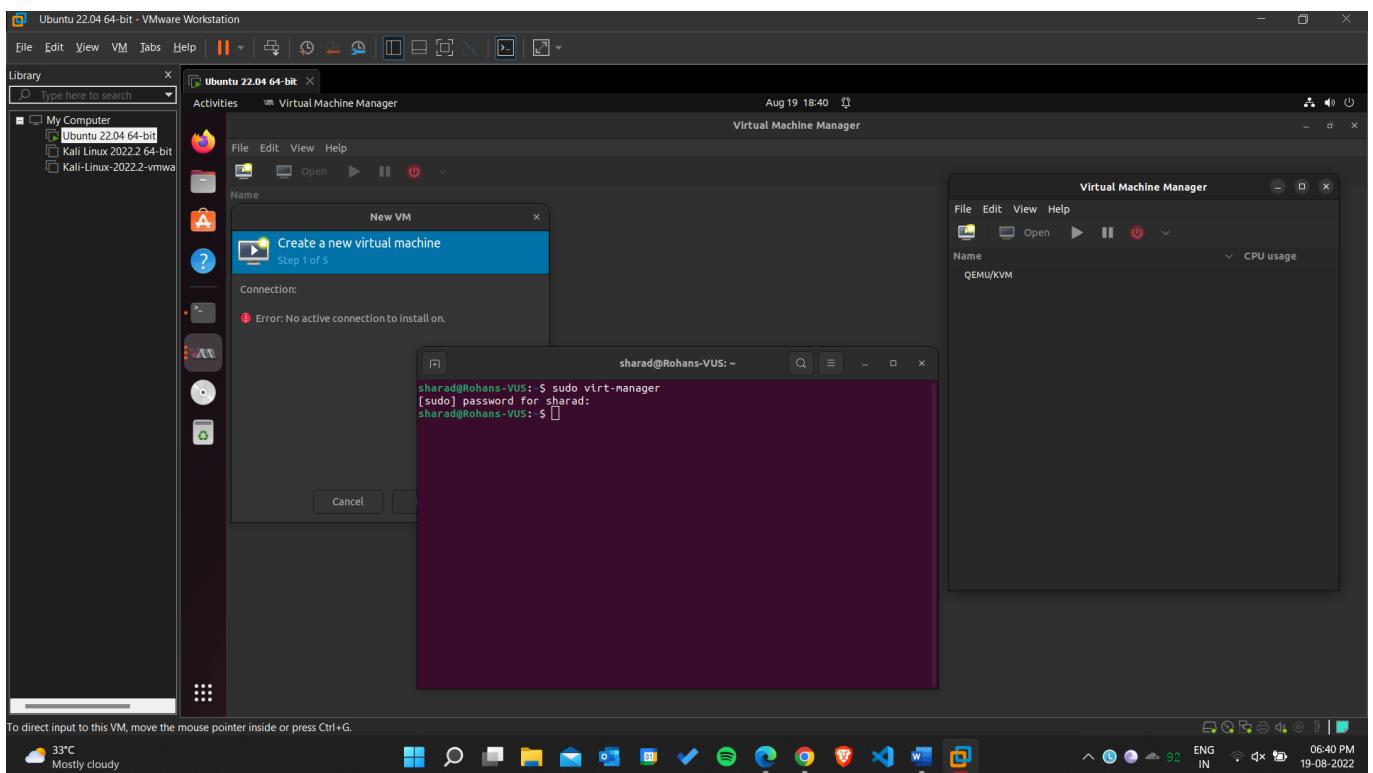
Step 6: Looks like our manager is here now.



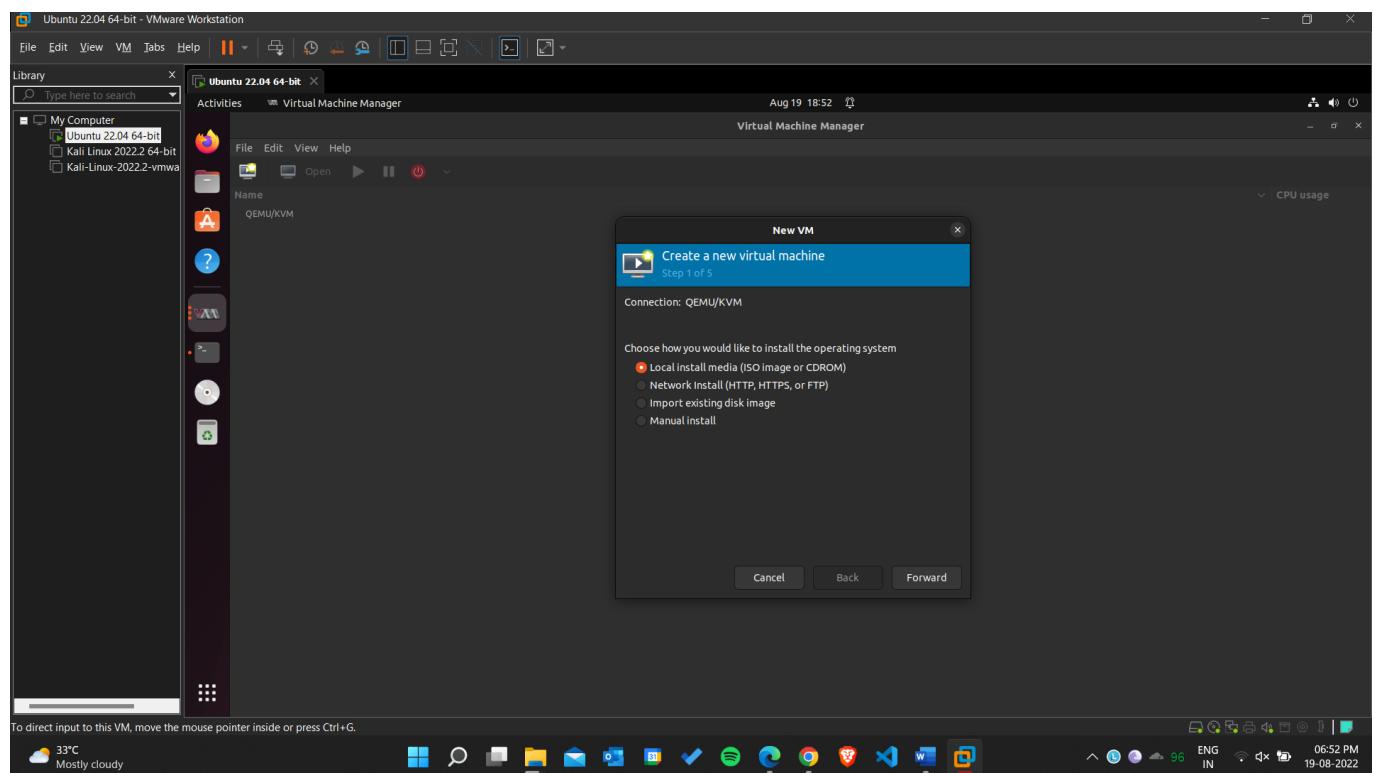
Step 7: Open the virtual machine manager, Click on ‘File’ and then click on ‘New Virtual Machine’.



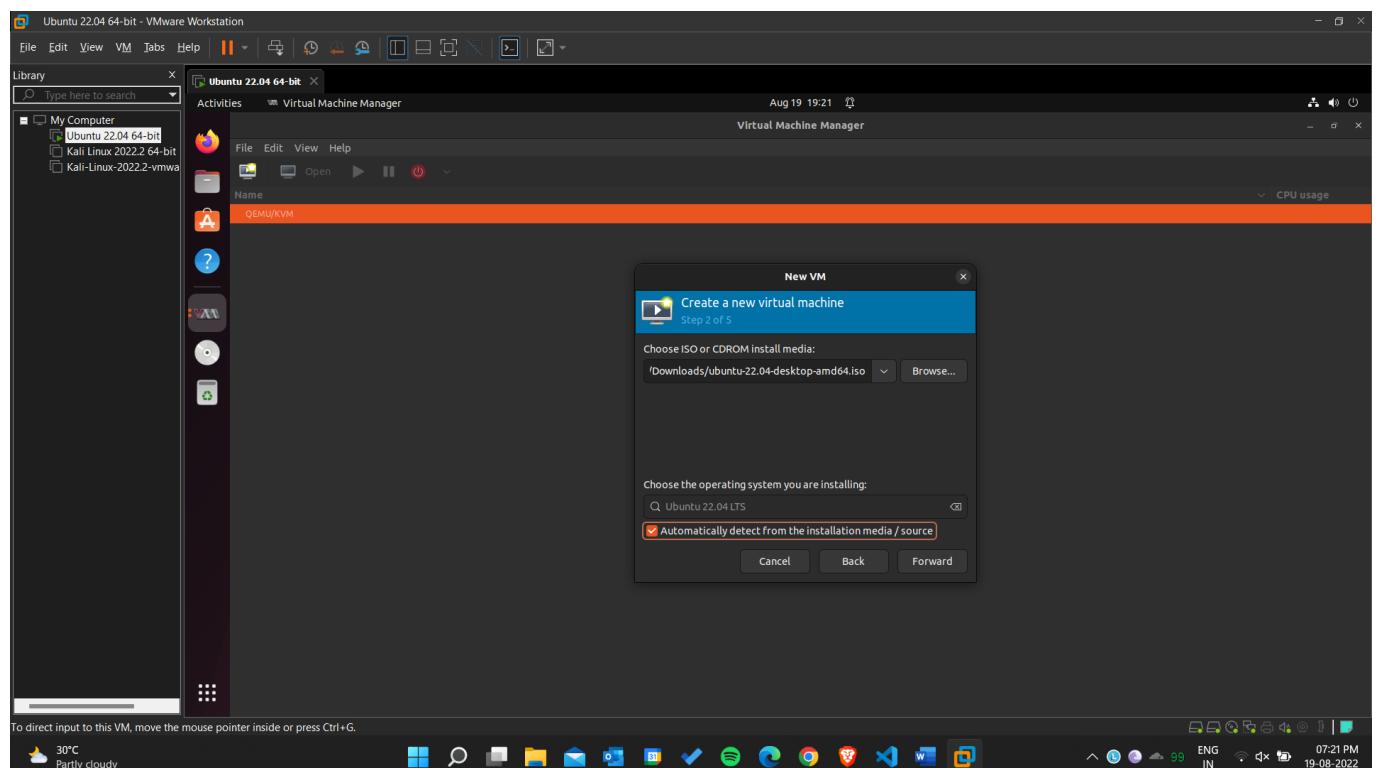
Step 8: Sometimes after doing so, it might show that there isn't any active connection. To mitigate this, just open the Terminal, and type in 'sudo virt-manager'. It'll open the manager again, in another window, which has the connection established.



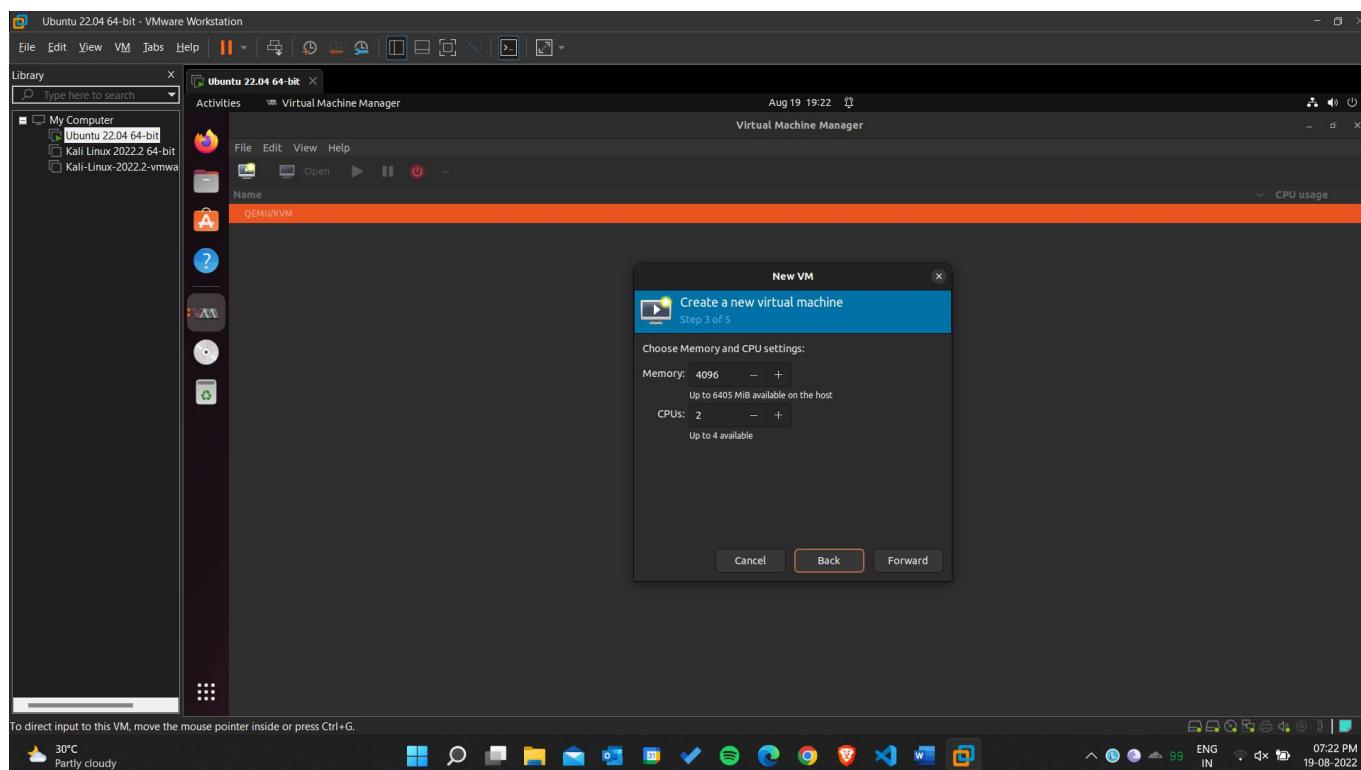
Step 9: In this new window, click on File → New Virtual Machine. Then Select ‘Local install media’ and click on ‘Forward’.



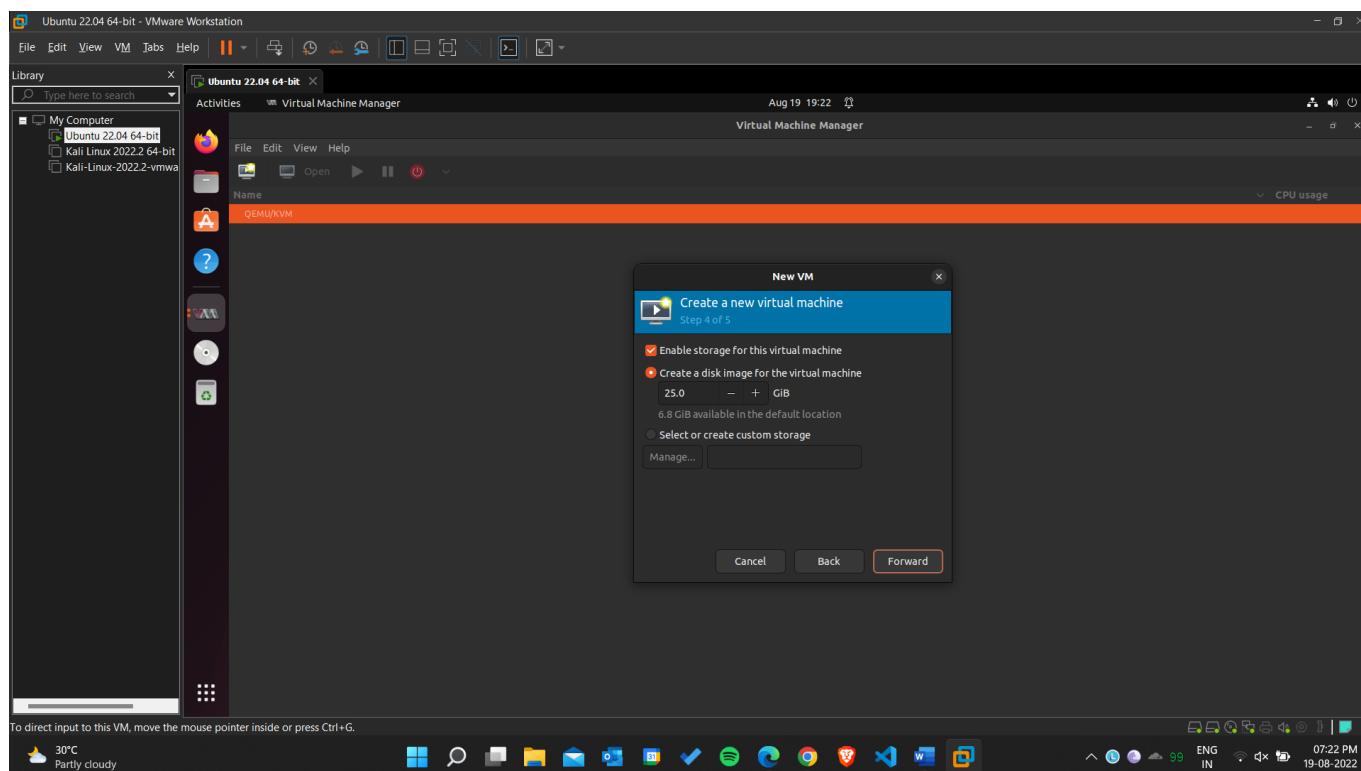
Step 10: Select the iso file.



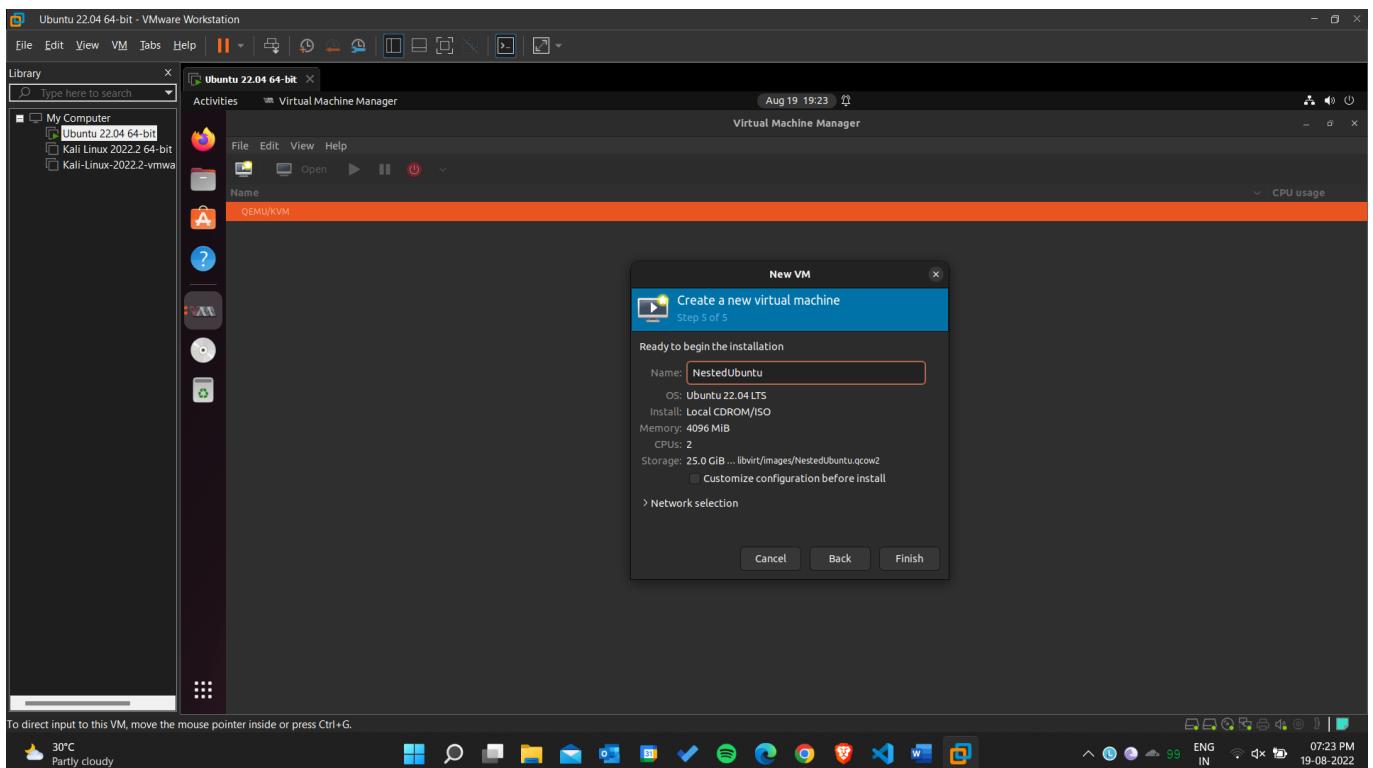
Step 11: Allocate the memory and CPU to the Virtual Machine Manager



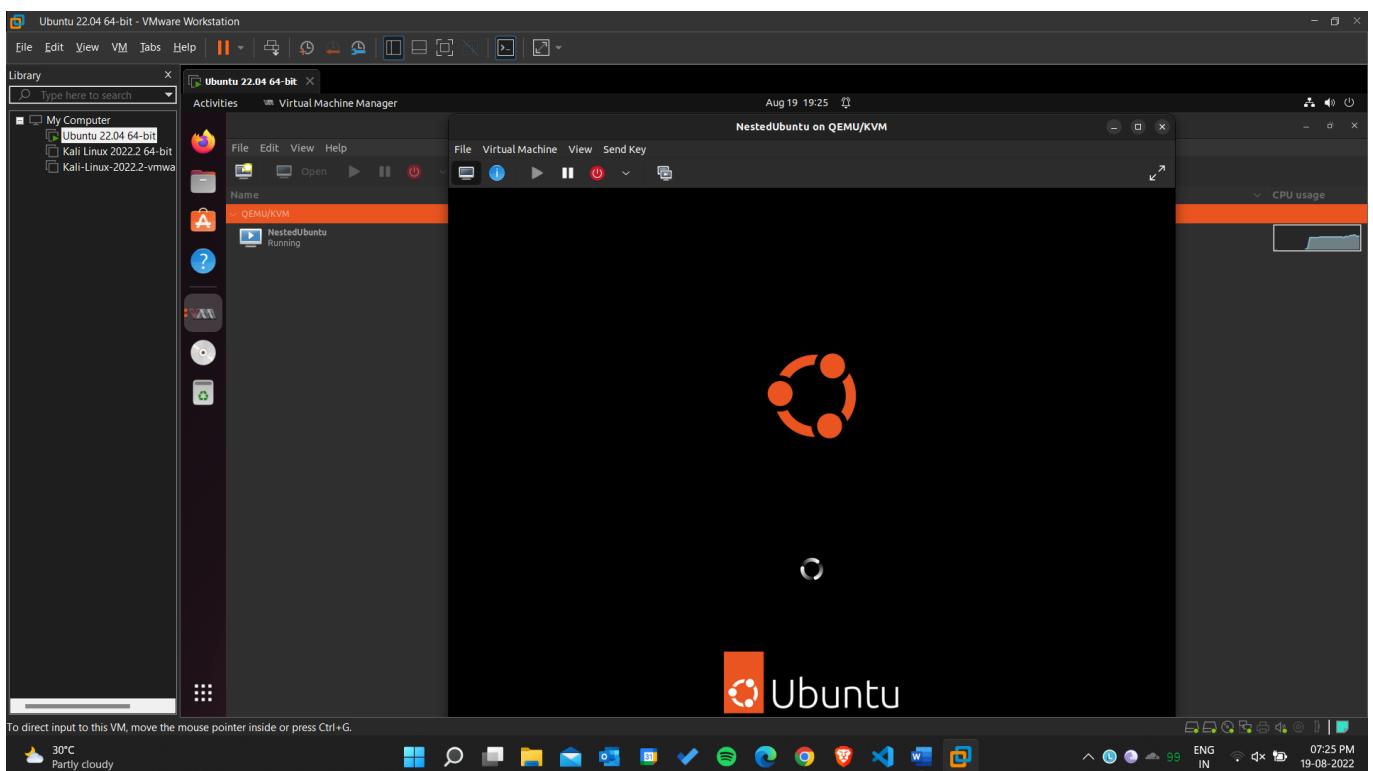
Step 12: Allocate disk size to the new VM



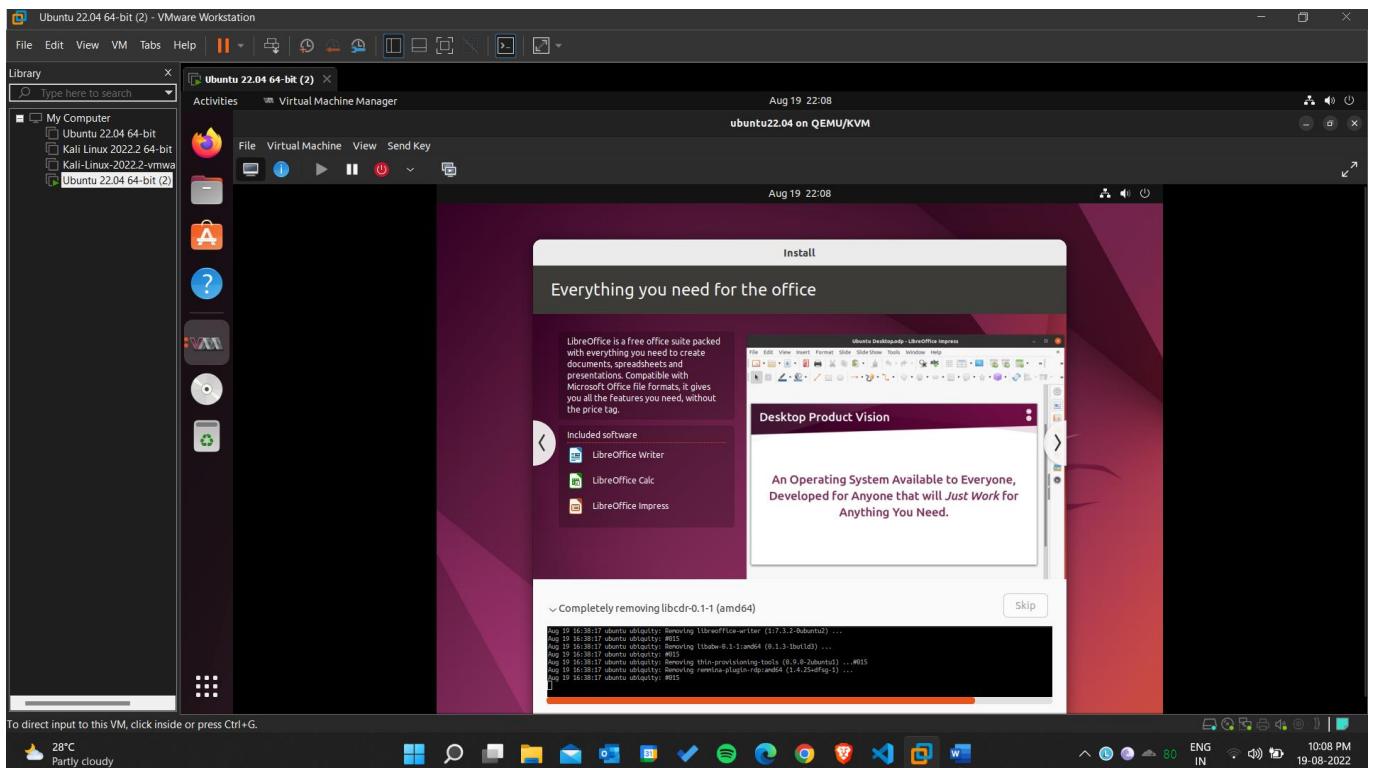
Step 13: Give a name to the nested VM.



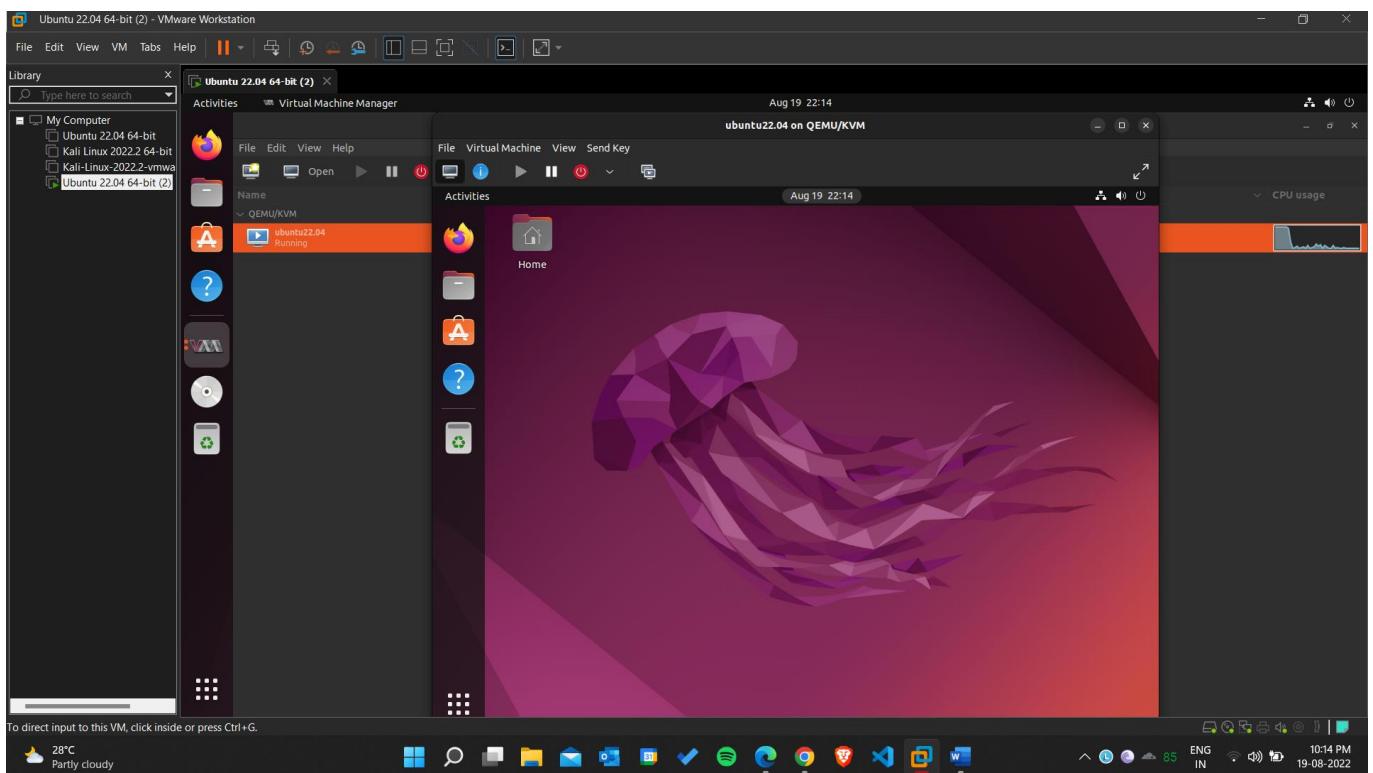
Step 14: After clicking Finish, OS in Nested VM starts installing



Step 15: Getting through the installation.



Step 16: OS in Nested VM has been successfully installed.



(2) Cloud:

Step 1: In the dashboard, go to the Images, and click on AMIs. Click on ‘Public images’.

The screenshot shows the AWS Management Console with the EC2 Management Console - Images page selected. On the left sidebar, under the 'Images' section, 'AMIs' is highlighted. In the main content area, there is a table with columns: 'AMI ID', 'AMI name', 'Source', and 'Owner'. A dropdown menu above the table is set to 'Owned by me', with 'Public images' selected. A message at the top of the table area says 'You do not have any images in this Region.' Below the table, a modal window titled 'Select an AMI' is displayed.

Step 2: Then, in the search bar, add ‘Owner alias = amazon’, and ‘hyper-v’. Select the most recent AMI from there.

The screenshot shows the AWS Management Console with the EC2 Management Console - Images page selected. The search bar at the top contains the filters 'Owner alias = amazon' and 'hyper-v'. The main content area is a table listing AMIs. One specific row for the AMI ID 'ami-07625abd566238be0' is selected and highlighted with a blue border. The table has columns: 'Name', 'AMI ID', 'AMI name', 'Source', and 'Owner'. The 'AMI ID' column shows values like 'ami-011d65e005ba4a0', 'ami-0712870256c5472', etc. The 'AMI name' column shows values like 'Launch instance from AMI', 'Copy AMI', etc. The 'Source' and 'Owner' columns show 'amazon/Windows_Server-2016-English...' and '801119661308' respectively.

Step 3: Launch the instance from the AMI and Configure it.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: Request Spot instances

Network: vpc-0c06adf23f1fe1f67 (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: Use subnet setting (Enable)

Hostname type: Use subnet setting (IP name)

DNS Hostname: Enable IP name IPv4 (A record) DNS requests
 Enable resource-based IPv4 (A record) DNS requests
 Enable resource-based IPv6 (AAAA record) DNS requests

Placement group: Add instance to placement group

Capacity Reservation: Open

Domain join directory: No directory [Create new directory](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

Feedback Looking for language selection? Find it in the new Unified Settings [?](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences 28°C Partly cloudy ENG IN 10:29 PM 19-08-2022

Step 4: Add storage. Increase it from 30 to 50 GiB

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-082f7d876569376bf	50	General Purpose SSD (gp2)	150 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted
ephemeral0	/dev/nvme0n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral1	/dev/nvme1n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral2	/dev/nvme2n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral3	/dev/nvme3n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral4	/dev/nvme4n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral5	/dev/nvme5n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral6	/dev/nvme6n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted
ephemeral7	/dev/nvme7n1	N/A	1900	SSD (NVMe AMI required)	N/A	N/A	N/A	Hardware Encrypted

[Add New Volume](#) [Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Feedback Looking for language selection? Find it in the new Unified Settings [?](#) © 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences 28°C Partly cloudy ENG IN 10:30 PM 19-08-2022

Step 5: Add tag, 'hyper-v'.

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name	hyper-v			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Step 6: Configure the Security Group. Better to just go with the default option.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group Select an existing security group

Security Group ID	Name	Description	Actions
sg-0f9739c232826ac04	default	default VPC security group	Copy to new
sg-0540563309b5bb4cc	launch-wizard-1	launch-wizard-1 created 2022-08-01T18:12:26.926Z	Copy to new
sg-06ce479b43f45f031	launch-wizard-2	launch-wizard-2 created 2022-08-01T23:58:52.327+05:30	Copy to new
sg-097baceb4bd47482d	launch-wizard-3	launch-wizard-3 created 2022-08-10T04:52:35.630Z	Copy to new
sg-0fcfcbbfe60aecdd44	launch-wizard-4	launch-wizard-4 created 2022-08-16T18:10:53.098Z	Copy to new

Inbound rules for sg-0f9739c232826ac04 (Selected security groups: sg-0f9739c232826ac04)

Type	Protocol	Port Range	Source	Description
All traffic	All	All	sg-0f9739c232826ac04 (default)	

Cancel Previous Review and Launch

Step 7: Select the existing key pair (which was made in Experiment 1).

Step 7: Review Instance Launch

AMI Details
Windows_Server-2019-English-Full-Hyper
Microsoft Windows Server 2019 Locale English with .NET Framework 4.8
Root Device Type: ebs Virtualization type: hvm

Instance Type
I3.metal - 72 vCPUs

Security Groups
sg-0f9739c232826ac04

Select an existing key pair or create a new key pair
A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).
Choose an existing key pair
Select a key pair
pocc RSA
 I acknowledge that I have access to the corresponding private key file, and that without this file, I won't be able to log into my instance.

Launch Instances

Step 8: Launch the instance.

Launch Status

Your instances are now launching
The following instance launches have been initiated: [i-0484d12f923aad79c](#) [View launch log](#)

Get notified of estimated charges
Create [billing alerts](#) to get an email notification when estimated charges on your AWS bill exceed an amount you define (for example, if you exceed the free usage tier).

How to connect to your instances
Your instances are launching, and it may take a few minutes until they are in the **running** state, when they will be ready for you to use. Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.
Click [View Instances](#) to monitor your instances' status. Once your instances are in the **running** state, you can [connect](#) to them from the Instances screen. [Find out](#) how to connect to your instances.

Here are some helpful resources to get you started

- [How to connect to your Windows instance](#)
- [Learn about AWS Free Usage Tier](#)
- [Amazon EC2: User Guide](#)
- [Amazon EC2: Microsoft Windows Guide](#)
- [Amazon EC2: Discussion Forum](#)

While your instances are launching you can also

- [Create status check alarms](#) to be notified when these instances fail status checks. (Additional charges may apply)

Thereafter, we need Hyper-V (which is only available on Windows 10 Pro versions) and use that to create a Nested VM in AWS, which is out of the scope of my laptop. So I'm abandoning this part.

Conclusion: I've successfully created a nested VM in my local machine (inside VMWare), and partially done the process of the nest VM on the Cloud.

Experiment 3

Title: SSH Tunnelling

Aim: To Create an SSH tunnel between our server in the local machine and remote clients in EC2 instances and test the connections.

Background Theory:

SSH Tunnelling:

- SSH tunnelling, or SSH port forwarding, is a method of transporting arbitrary data over an encrypted SSH connection. SSH tunnels allow connections made to a local port (that is, to a port on your own desktop) to be forwarded to a remote machine via a secure channel.
- To protect our network services, not all of them are reachable directly from outside the ENCS network. If you are offsite and need to access a resource that is protected in this way, you can use ssh to tunnel through an accessible resource to reach the protected resource.

Usage of SSH:

- To connect to a remote host using terminal (On Linux): ssh [username]@[server_ip_or_hostname]
- For Windows-based systems use the tool called ‘PuTTY’.
- PuTTY is an SSH client with a GUI for SSH and Telnet.

Requirements for conducting the experiment:

- VM instances running on local machines and cloud
- PuTTY for Windows-based system
- Terminal for Linux
- SSH clients and server

Procedure & Screenshots:

Step 1: Create a new instance in AWS, after terminating the existing ones.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
-	i-0a542ae561e1069d1	Terminated	t2.micro	-	No alarms	ap-south-1a	-
-	i-0023247dbd8b11024	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-2-11-1
-	i-0484d12f923aad79c	Terminated	t2.micro	-	No alarms	ap-south-1a	-

The screenshot shows the AWS Management Console with the following tabs open: WhatsApp, Running Hyper-V on Amazon, AWS Management Console, Launch instance wizard | EC2, Instance details | EC2 Manager, 3.109.152.136, Apps, RR on Behance, Mitacs, Canada, MIT DL, MIT Challenge, Papers w/ code, Admitted 2ed, Unblockit, Carbon, ide.new, Acrobat, VIT Mail, MBA Deadlines, My Competitive Pro..., Mumbai, and Sharadindu.

The main view is the EC2 Instances page, showing the instance summary for i-0023247dbd8b11024. The instance is running and has a public IP of 65.2.11.194. It is an t2.micro instance type with an auto-assigned IP of 65.2.11.194. The subnet ID is subnet-0c53008b735ea3d86. The instance was updated less than a minute ago.

Instance summary for i-0023247dbd8b11024

Value	Description
Instance ID	i-0023247dbd8b11024
IPv6 address	-
Hostname type	IP name: ip-172-31-44-253.ap-south-1.compute.internal
Answer private resource DNS name	-
Auto-assigned IP address	65.2.11.194 [Public IP]
IAM Role	-

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

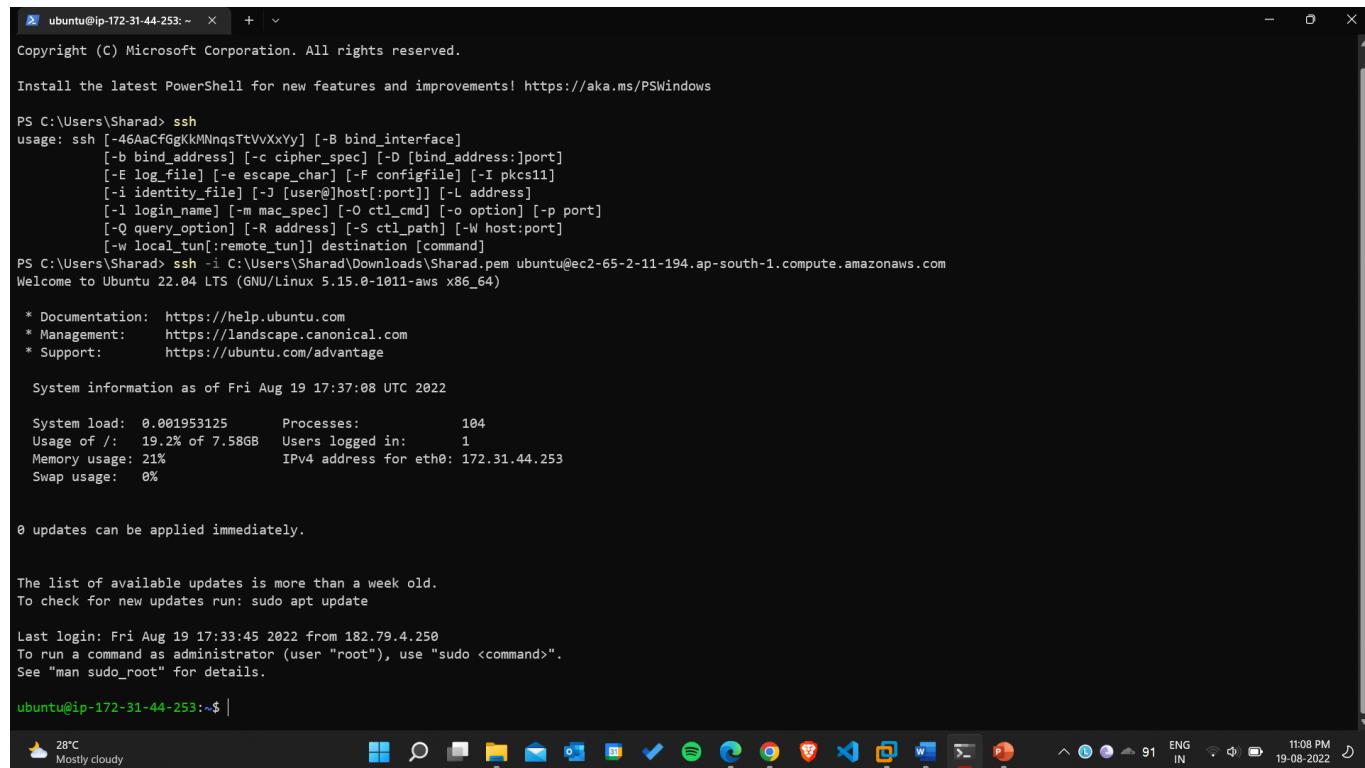
Instance details | Info

Step 2: Copy the Public DNS from the following page.

The screenshot shows the AWS Management Console with the URL ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#ConnectToInstance:instanceId=i-0023247dbd8b11024. The page title is "Connect to instance | EC2". The top navigation bar includes links for "AWS", "Services", and a search bar. The main content area is titled "Connect to instance" and provides instructions for connecting to the instance using "SSH client". It lists four steps: 1. Open an SSH client, 2. Locate your private key file (Sharad.pem), 3. Run chmod 400 Sharad.pem, and 4. Connect to your instance using its Public DNS (ec2-65-2-11-194.ap-south-1.compute.amazonaws.com). An example command is shown as "ssh -i "Sharad.pem" ubuntu@ec2-65-2-11-194.ap-south-1.compute.amazonaws.com". A note at the bottom states: "Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name."

Step 3: Open PowerShell. Type in 'SSH', to see if SSH is already configured on the laptop or not. In my system it is. So no additional steps are necessary at this point.

Step 4: Use the .pem file downloaded earlier (in experiment 1), and the public dns copied to type in the following command: "ssh -i C:\Users\Sharad\Downloads\Sharad.pem ubuntu@ec2-65-2-11-194.ap-south-1.compute.amazonaws.com"



```
ubuntu@ip-172-31-44-253: ~ + | ~
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Sharad> ssh [-i46AaCfGkKMNnqsTtVvXxYy] [-B bind_interface]
usage: ssh [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
PS C:\Users\Sharad> ssh -i C:\Users\Sharad\Downloads\Sharad.pem ubuntu@ec2-65-2-11-194.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Fri Aug 19 17:37:08 UTC 2022

System load: 0.001953125 Processes: 104
Usage of /: 19.2% of 7.58GB Users logged in: 1
Memory usage: 21% IPv4 address for eth0: 172.31.44.253
Swap usage: 0%

0 updates can be applied immediately.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Fri Aug 19 17:33:45 2022 from 182.79.4.250
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-44-253:~$ |
```

The screenshot shows a Windows PowerShell window with the title bar "ubuntu@ip-172-31-44-253: ~ + | ~". The command "ssh -i C:\Users\Sharad\Downloads\Sharad.pem ubuntu@ec2-65-2-11-194.ap-south-1.compute.amazonaws.com" is being typed. The output shows the connection to an Ubuntu 22.04 LTS instance with IP 172.31.44.253. The system information for the host shows a load average of 0.001953125, 104 processes, and 1 user logged in. The terminal also displays a message about available updates and the last login time.

Conclusion: I've successfully created an SSH tunnel between our server in the local machine and remote clients in an EC2 instance of AWS.