Parul® University
Vadodara, Gujarat

NAAC GRADE A++

FACULTY OF ENGINEERING AND TECHNOLOGY

BACHELOR OF TECHNOLOGY

Cloud Computing (303105364)

SEMESTER 6th (3rd YEAR)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Laboratory Manual

Parul® University
Vadodara, Gujarat

NAAC GRADE A++

PARUL UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

PARUL INSTITUTE OF ENGINEERING & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

STUDENT DETAILS

NAME	:	
ENROLLMENT NO	:	
SUBJECT	:	Cloud Computing
SUBJECT CODE	:	303105364
SEMESTER	:	6TH
FACULTY NAME	:	Pathan Bilalkhan R.

Parul® University
Vadodara, Gujarat

NAAC GRADE A++

FACULTY OF ENGINEERING AND TECHNOLOGY

BACHELOR OF TECHNOLOGY

CERTIFICATE

This is to certify that Mr./Ms _____ with Enrollment no: _____has Successfully completed his/her Laboratory Experiments in the Cloud Computing (303105364) from the Department of Computer Engineering during the academic year 2025-26

Date of Submission: .....

Staff In charge: ..... **Head of Department: .....**

INDEX

Sr. No	Experiment Title	Number of Hours	Page No.	Date of Performance	Date of Assessment	Marks (out of 10)	Sign
1	Understanding single core and multi core Architecture.	2					
2	Understanding Computer Network fundamentals and Designing LANs.	2					
3	Implementation of Infrastructure as a service (IaaS) using Hypervisors.	2					
4	Implementation of private cloud platform using open stack cloud.	2					
5	Working with IaaS of Public cloud platforms.	2					
6	Implementation of Platform as a service (PaaS) in private cloud environment.	2					
7	Implementation Platform as a service (PaaS) in public cloud environment.	2					
8	Implementation Software as a service (SaaS) in private cloud environment.	2					
9	Implementing Software as a service (SaaS) in public cloud environment.	2					
10	Implementation of Storage as a service (SaaS).	2					

Practical-1

Aim: Understanding single core and multi core Architecture.

Core:

In computing, a core refers to the processing unit within a CPU that executes instructions from programs. A single core can handle one task (or thread) at a time, while modern CPUs typically have multiple cores to enable parallel processing, improving performance. Each core has its own ALU (Arithmetic Logic Unit) and control unit, allowing independent operations. Multi-core processors are common in devices, supporting multitasking and efficient execution of complex applications. The number of cores directly impacts the ability to run simultaneous processes smoothly.

Types of Cores:

Single-Core Processor: Contains one core to execute tasks sequentially. It performs one operation at a time and is suitable for simple tasks but struggles with multitasking and modern applications.

Multi-Core Processor: Includes two or more cores in a single CPU, enabling parallel processing for improved performance and efficiency in multitasking and demanding applications.

Need of Multi-Core Processor:

Single-core processors are adequate for basic operations like simple browsing or office tasks. Multi-core processors are essential for modern systems that run complex applications, multitasking, gaming, or heavy computations, as they divide workloads across multiple cores for better speed and responsiveness.

Architecture of a Single-Core Processor

Design

1. **Simplicity in Design:** Compared to multi-core processors, designing a single-core processor is simpler since only one core is present. There's no need to implement complex communication mechanisms among multiple cores.
2. **Ease of Software Development:** Developing software for a single-core processing system is easier compared to multi-core systems, as there are no concerns about thread synchronization or resource contention among cores.

Throughput

1. **Sequential Task Execution**: Throughput is lower because tasks are executed sequentially, one after the other.
2. **Limited Parallelism:** While multitasking can reduce CPU idle time, true parallelism cannot be achieved, limiting performance for computationally intensive workloads.

Cost and Power Consumption

1. **Cost Efficiency:** The cost of a single-core system is significantly lower since only one core is used, reducing manufacturing complexity and silicon usage.
2. **Power Efficiency:** Single-core processors consume less power, making them suitable for devices with strict energy constraints.

Applications

1. **Low-Power, Simple Systems**: Despite the dominance of multi-core processors, single-core systems are still used in applications where low power consumption and simplicity are priorities.
2. **Embedded Systems:** Devices such as microwaves, washing machines, and IoT devices, which do not require high computational power, continue to rely on single-core processors.

Architecture of a Multi-Core Processor

Design

1. **Complex Design:** Multi-core processors are more complex than single-core processors due to the need for effective communication mechanisms, such as interconnects, shared buses, or ring architectures, to allow cores to collaborate efficiently.
2. **Software Development Challenges**: Developing software for multi-core processors is challenging as it requires parallel programming techniques, thread management, and synchronization mechanisms to ensure optimal performance and avoid issues like race conditions or deadlocks.

Throughput

1. **High Throughput:** By allowing multiple cores to execute tasks simultaneously, multi-core processors achieve significantly higher throughput compared to single-core processors.
2. **Parallel Task Execution:** Supports true parallelism, enabling better performance for multithreaded applications such as video rendering, gaming, and scientific simulations.

Cost and Power Consumption

1. **Higher Cost:** Multi-core processors are more expensive to manufacture due to the complexity of integrating multiple cores on a single chip and ensuring efficient communication among them.
2. **Increased Power Consumption:** Although more power-efficient per task compared to increasing clock speed, the overall power consumption is higher than single-core processors, especially in high-performance configurations.

Applications

1. **High-Performance Systems:** Multi-core processors are used in systems requiring significant computational power, such as servers, gaming consoles, and workstations.
2. **Parallel Computing:** Applications like machine learning, big data processing, and scientific simulations benefit from the parallel processing capabilities of multi-core processors.
3. **Consumer Electronics:** Modern smartphones, tablets, and laptops use multi-core processors for efficient multitasking and responsiveness.

Physical and Logical Cores:

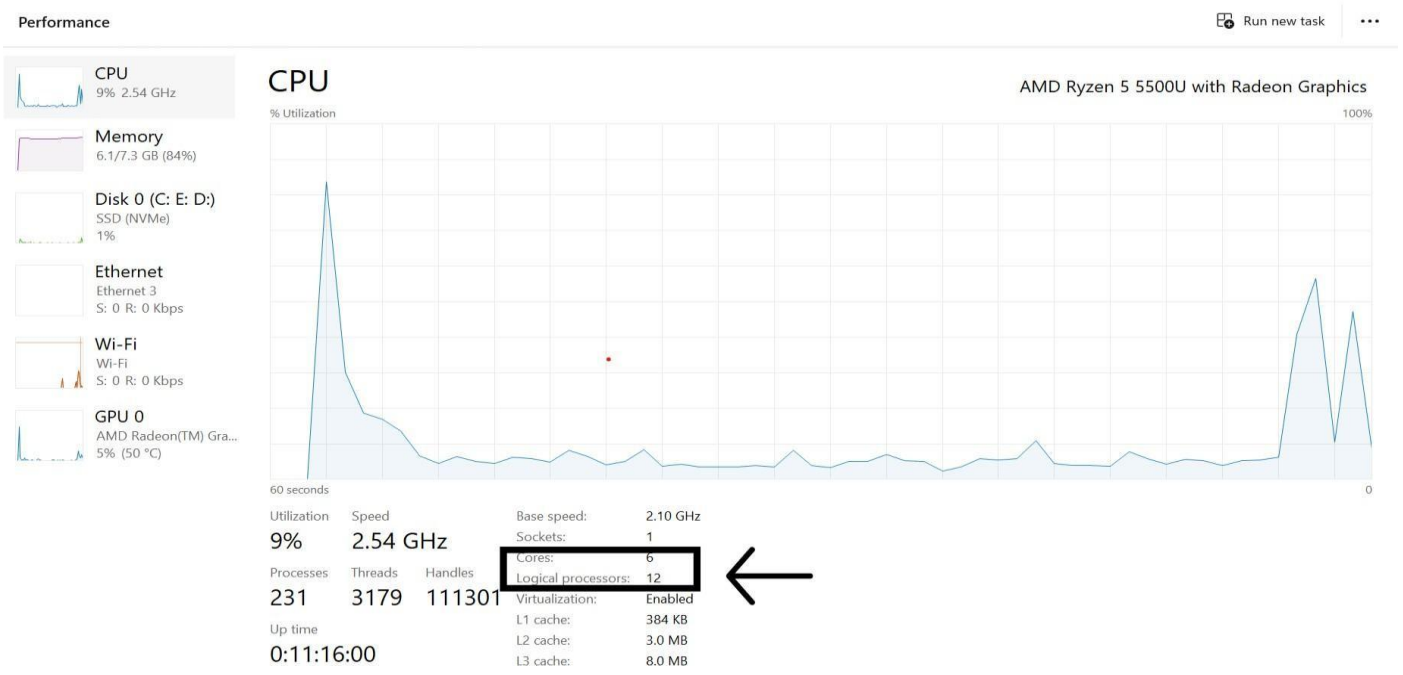
A multi-core processor contains multiple physical cores that allow parallel processing to improve performance. To further enhance core utilization, Hyper-Threading Technology (HTT) is employed. This technology creates logical cores by enabling a single physical core to handle multiple threads simultaneously. Logical cores help reduce the idle time of the CPU by optimizing how tasks are executed, allowing it to switch between threads efficiently.

This means that each physical core is virtualized as multiple logical cores (commonly two per physical core), enabling better multitasking and improving performance in workloads that benefit from parallel execution, such as video rendering or running virtual machines. However, it is important to note that logical cores share the same physical core's resources (e.g., cache and execution units), so their performance boost depends on the workload.

Exploring cores in personal computer:

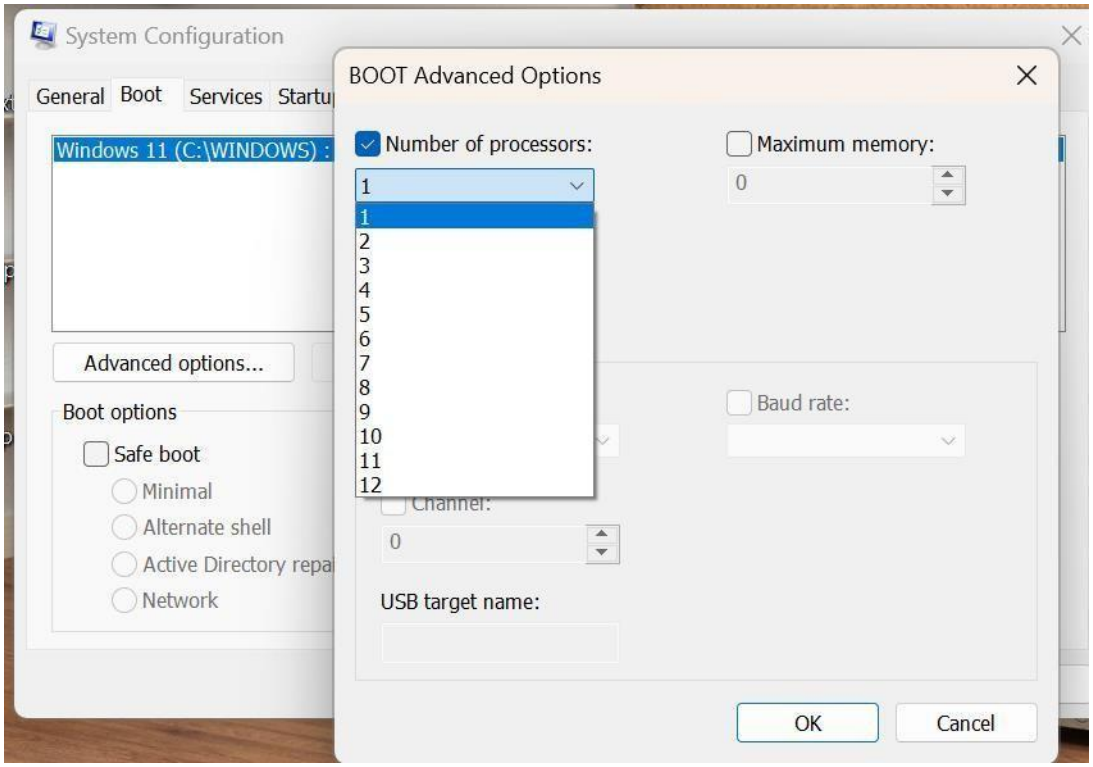
First way:

1. Go to Task manager
2. Click on the CPU
3. Each core can work with 2 threads simultaneously using Hyper-Threading Technology. Hence Logical cores will be double of the Physical cores



Second Way:

1. Open System Configuration:
2. Press Win + R to open the Run dialog.
3. Type msconfig and press Enter.
4. Navigate to the Boot Tab:
5. In the System Configuration window, go to the Boot tab.
6. Open Advanced Options:
7. Under the Boot tab, click on Advanced Options.
8. Check Processor Count:
9. In the Boot Advanced Options window, you'll see a checkbox labelled Number of processors.
10. Check the box, and the drop-down menu below will display the number of cores available in your CPU.
11. Exit Without Changing:
12. If you are just checking the core count, click Cancel to avoid making unintended changes, and then close the System Configuration window.



Comparing Single and Multi-Core Processor:

Feature	Single-Core Processor	Multi-Core Processor
Definition	A processor with one core or processing unit	A processor with multiple cores or processing units
Parallel Processing	Limited to executing one instruction at a time	Can execute multiple instructions simultaneously
Performance	Lower performance for multitasking and complex tasks	Higher performance for multitasking and complex tasks
Power Consumption	Generally lower power consumption	Higher power consumption but more efficient per task
Heat Generation	Generates less heat	Generates more heat due to multiple cores
Efficiency	Less efficient for modern applications	More efficient for modern applications that require parallel processing
Cost	Typically cheaper	Generally more expensive
Suitable For	Basic tasks like web browsing, word processing	Intensive tasks like gaming, video editing, and scientific computations
Example	Early desktop computers	Modern desktops, laptops, and servers

Practical-2

Aim: Understanding Computer Network fundamentals and Designing LANs.

Computer Network:

A computer network is a group of devices that are connected and can share resources and data with each other. Computer networks are made up of nodes and links, which are the basic building blocks of the network. Nodes are devices that send and receive data, such as modems, hubs, and switches. Links are the connections between nodes, such as cable wires or optical fibres.

Network Model:

Sharing data on the network involves many challenges and Network Model is a layered structure which solves every challenge in different layers enabling smooth functioning in the network.

Types:

1. TCP/IP Model
2. OSI Model

TCP/IP Model:

Stands for Transmission Control Protocol/Internet Protocol. This model consists of 4 layers namely,

1. Application Layer
2. Transport Layer
3. Network Layer
4. Network Access Layer

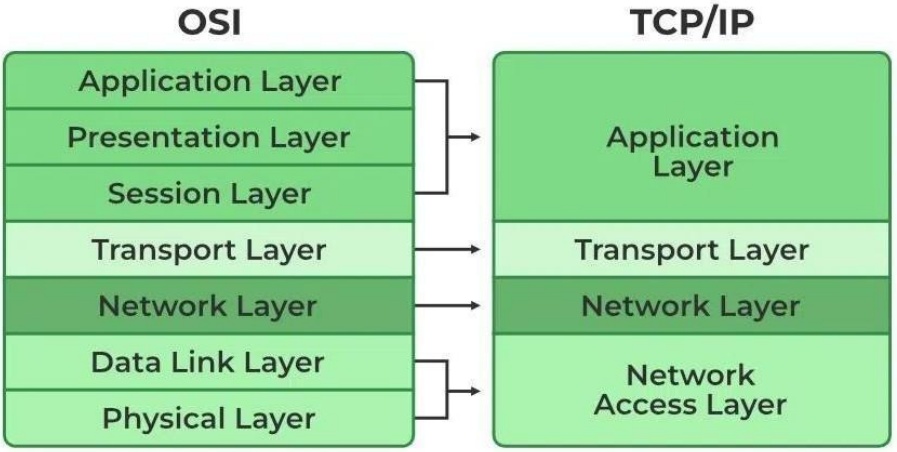
TCP/IP model is one the oldest and most widely used model till date.

OSI Model:

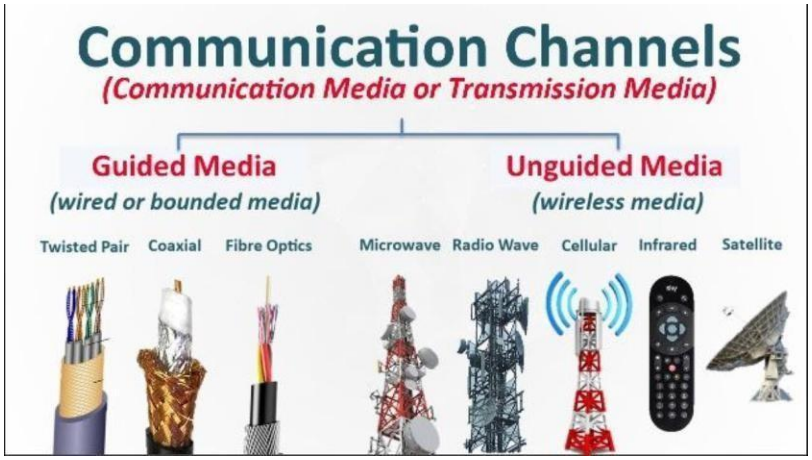
Stands for Open System Interconnection. This model consists of 7 layers namely,

1. Application layer
2. Presentation layer
3. Session layer
4. Transport layer
5. Network layer
6. Data link layer
7. Physical layer.

Each layer gives services to the layer just below it and take services from layer above it.



Transmission Media:



Transmission media talks about how data is exactly transferred. It can be wired in other words known as **Guided Media** as well. **Unguided Media** involves data transfer based on electromagnetic waves.

Twisted Pair Cable: To reduce electromagnetic inference two insulated copper wires are twisted together. Generally used in telephone systems, Ethernet networks.

Coaxial Cable: Broadband Internet, TV cable are examples of this coaxial cable. This cable contains single copper wire with multiple layers of protection. This is expensive than Twisted Pair Cable.

Fibre Optics: Works on the principle of total internal reflection. This is made of sand which looks very similar to glass. Uses light signals to transmit data.

Unguided Media: This method involves data transfer without the need of cables. But this method involves various security issues. Radio, Cellular Network works on Unguided Media.

Networking Devices: These are hardware devices in the computer networks which ensures smooth communication over the internet. Various devices work at various levels of network model.

Active Devices: Routers, Switches

Passive Devices: Hubs, Repeaters

End Devices: Servers, Workstations

Routers:

This device works at network layer. Router connects LAN to WAN by providing access to the internet with the help of Ip address.

Switches:

Mainly used in Data Link Layer but sometimes used in network layer as well. Switch connect various devices within same network. Based on MAC address switch forwards data to target machine only.

Network Topologies:

- Star
- Ring
- Bus
- Mesh

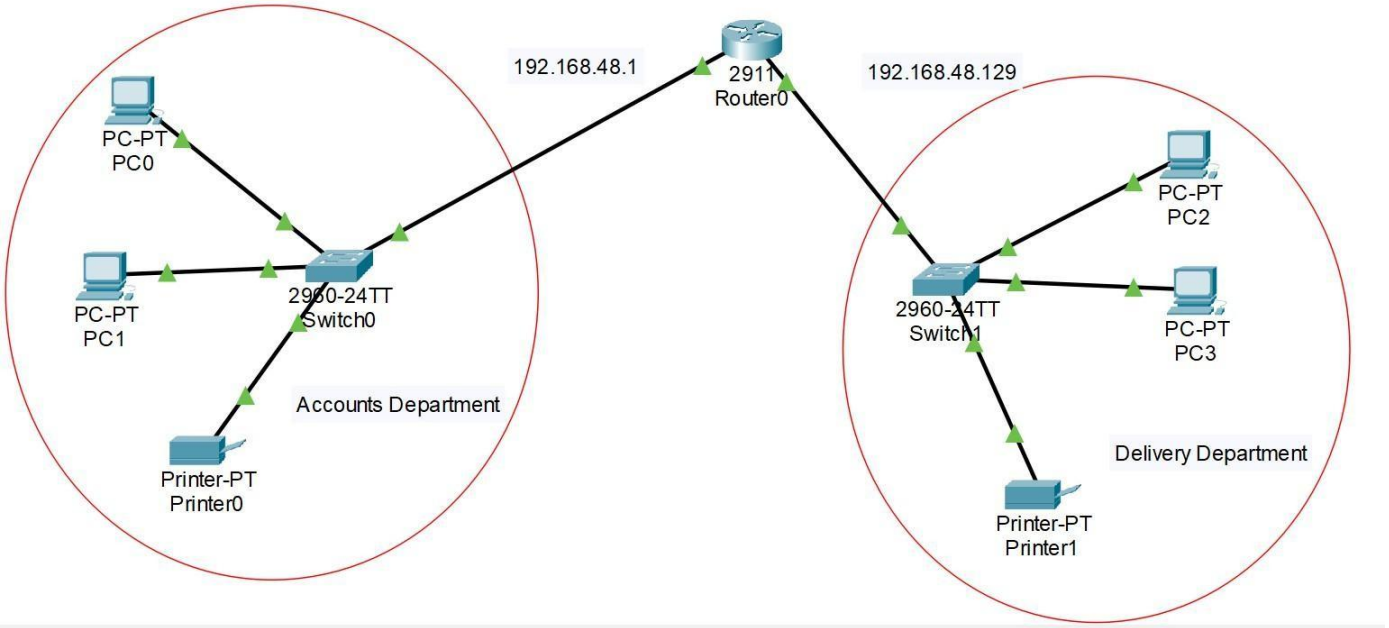
Network Protocols:

- TCP
- UDP

Designing a Network:

Create a network with 2 departments. Use appropriate number of switches and routers.

Use the network addresses 192.168.40.0. use appropriate cables to connect all the devices.



Output:

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Printer1	ICMP		0.000	N	0	(edit)	(delete)
	Successful	PC1	PC2	ICMP		0.000	N	1	(edit)	(delete)
	Successful	Printer0	Printer1	ICMP		0.000	N	2	(edit)	(delete)

Practical-3

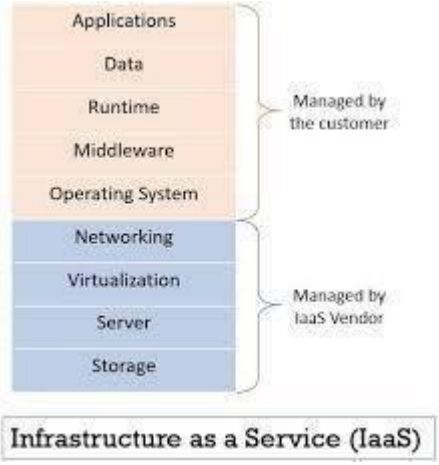
Aim: Implementation of Infrastructure as a service (IaaS) using Hypervisors.

Iaas: Before getting into the practical let us try to understand about infrastructure as a service.

For Hosting a software application, we need lot of hardware in the backend. Hardware includes server, storage, networking etc.

This becomes costly and headache for developers to always look after the working of this underlaying hardware in addition to maintaining their software application.

Iaas solves this problem by providing virtualized resources via internet. All we need to good internet connection. We can get server, storage, networking etc without setting up on our own. Iaas commits that cloud provider will be solely responsible for managing underlaying



Infrastructure as a Service (IaaS)

infrastructure. But operating system and runtime environments should be seen by user itself.

Hypervisor: Hypervisors are the implementation of vitalization concept. This is the backbone of IaaS, because only Hypervisors enable resource sharing and dynamic allocation of virtualized environments. Hypervisor allows multiple operating systems and virtual machines to run on the same physical hardware by abstracting and allocating resources such as CPU, Memory, Storage and Network.

Types of Hypervisors:

Type 1 Hypervisor (Bare-Metal Hypervisors):

These run directly on the physical hardware without requiring underlying OS.

Example: KVM (kernel virtual machine) which we are going to implement now and other hypervisors also exist like VMware ESXi, Microsoft Hyper-V etc.

Type-2 Hypervisors (Hosted Hypervisors):

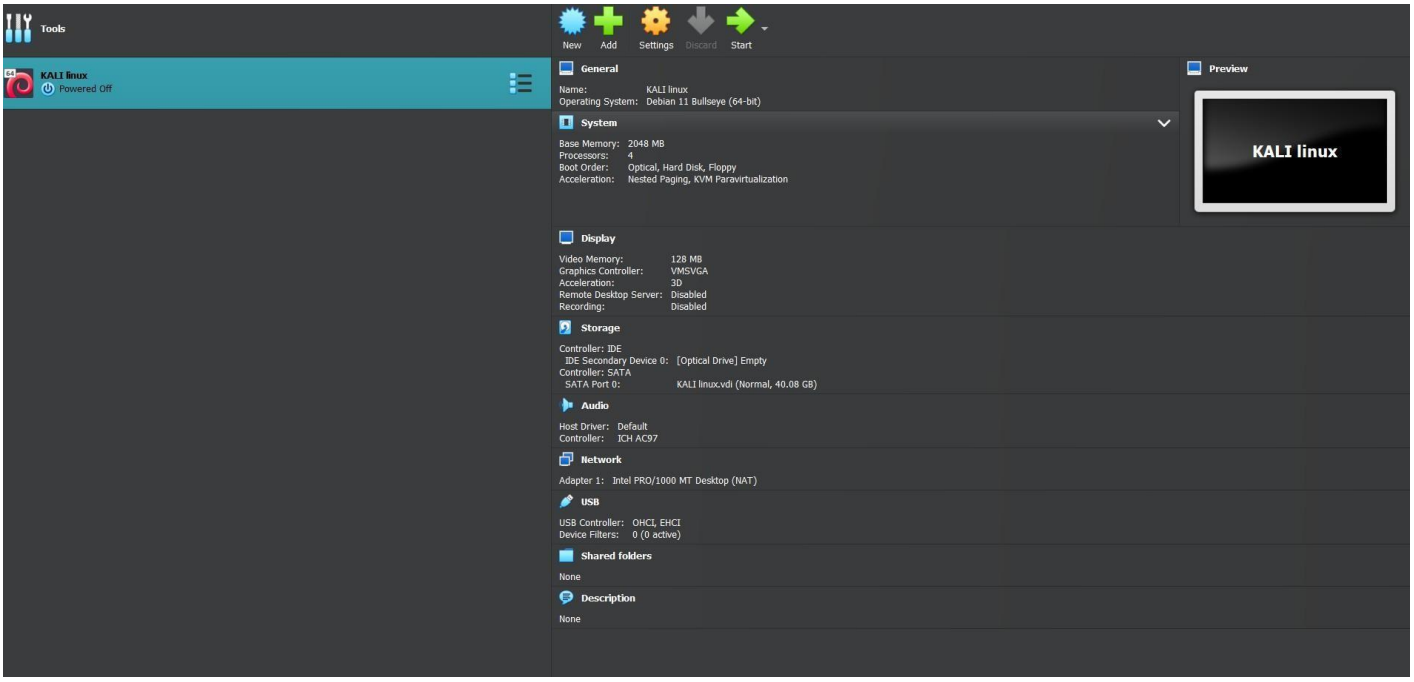
These run on top of an existing operating system and relay on the host to manage hardware resources. This will be useful for development, testing and for small scale virtualized environments.

Example: VMware workstation, VirtualBox.

Let us create a virtual machine using KVM:

Things to keep in mind:

1. We are on windows operating system, and we have installed virtual box, on top of virtual box we had Linux system.
2. Using Linux machine present in the virtual box we need to create virtual machine with the help of KVM and virtual manager.



Use lscpu command:

In this section, we have virtualization features. It should be showing full. And hypervisor type will be KVM (kernal virtual machine)

Now use the following command:

sudo apt update && sudo apt install -y qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils virt-manager

qemu-kvm

- QEMU is an open-source emulator that allows running virtual machines.
- KVM (Kernel-based Virtual Machine) is a Linux-based hypervisor for running VMs efficiently.

libvirt-daemon-system

- Libvirt is a toolkit for managing virtual machines.
- libvirt-daemon-system provides system-wide services for managing VMs.

libvirt-clients

- It contains command-line tools like virsh to interact with virtual machines.

bridge-utils

- It provides network bridging capabilities.
- Used for connecting virtual machines to the internet or the host network.

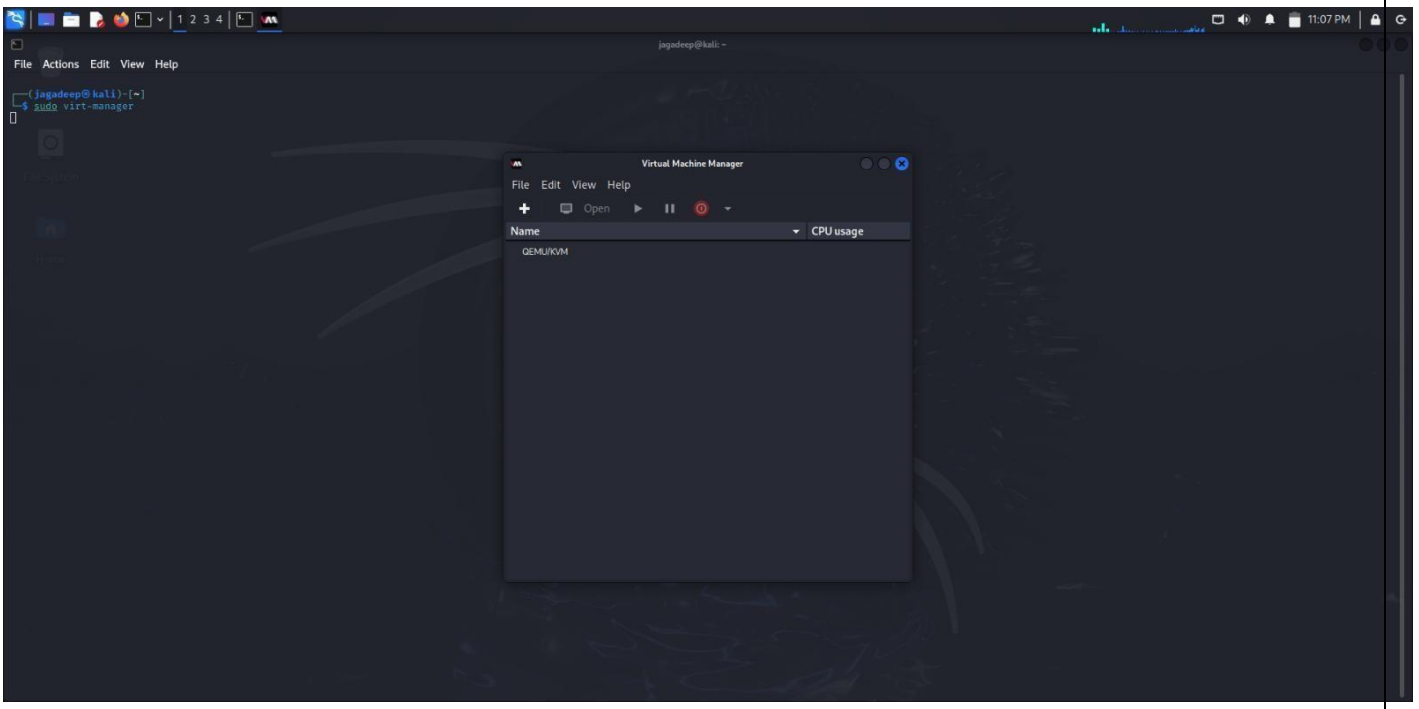
virt-manager

- Virt-Manager (Virtual Machine Manager) is a GUI tool for managing VMs.

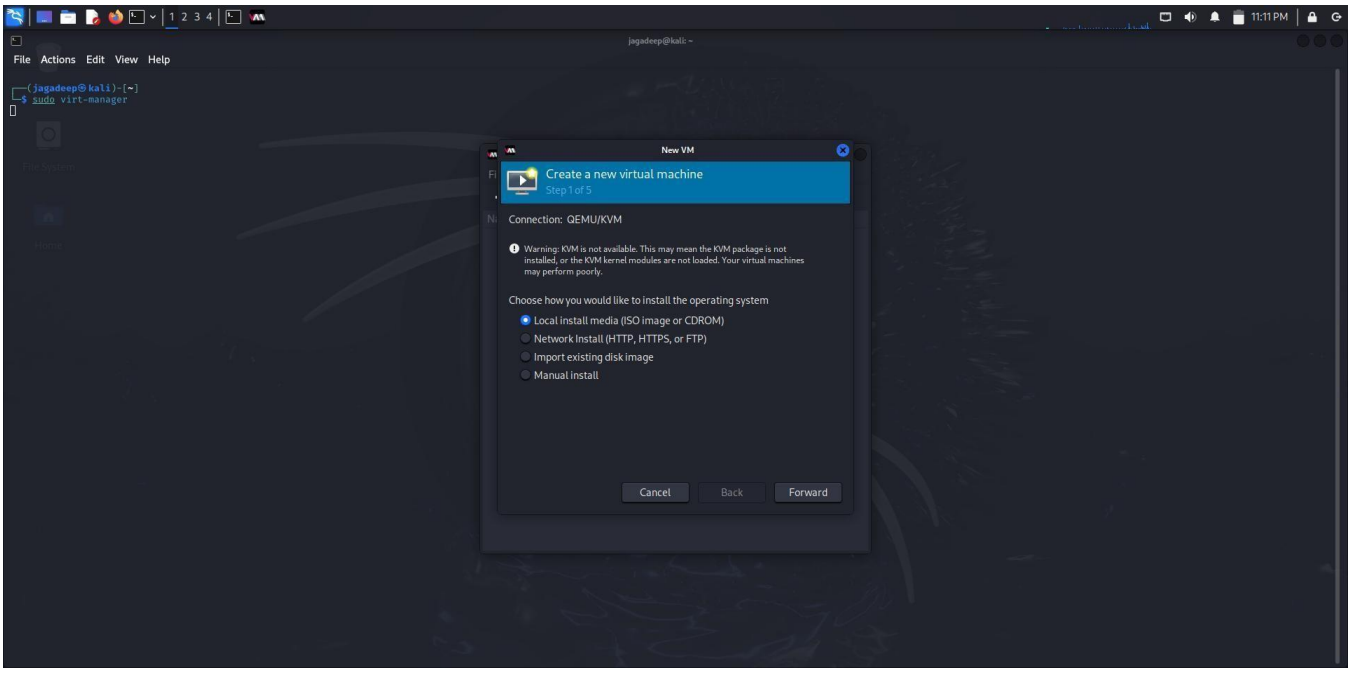
- It provides an easy-to-use interface

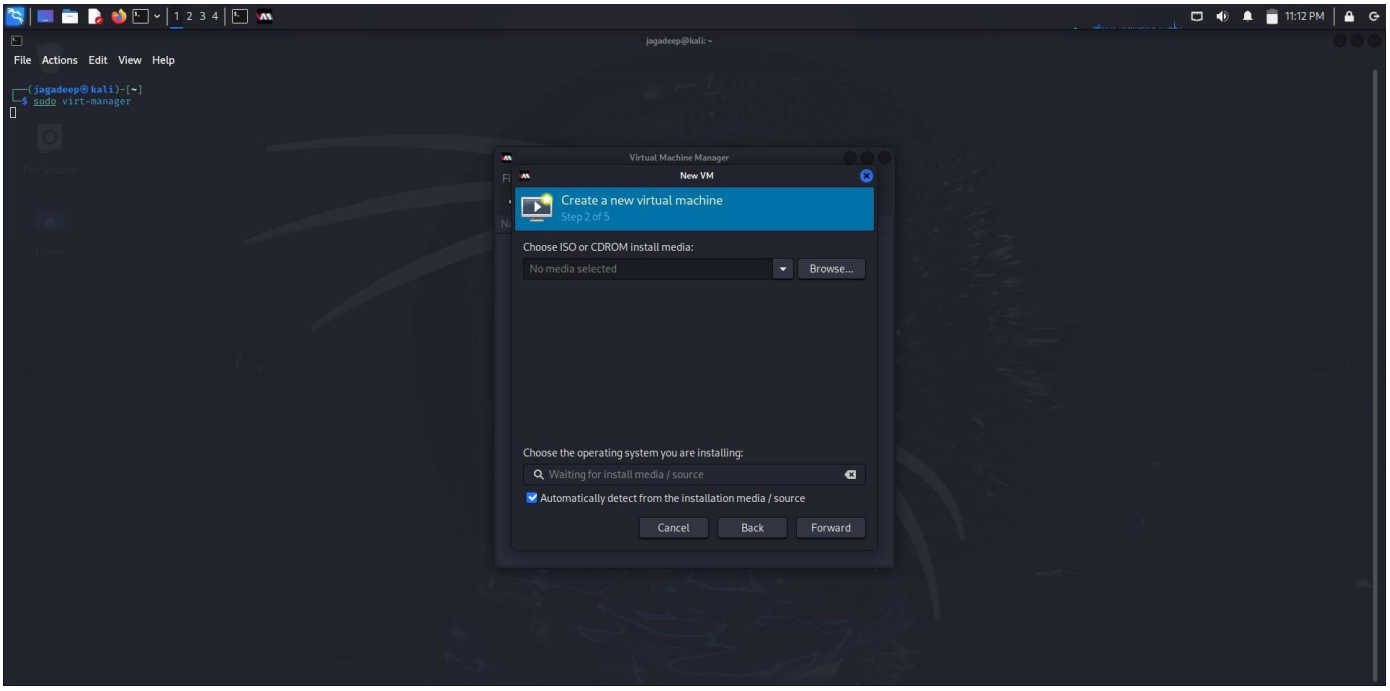
to create, delete, and manage virtual machines.

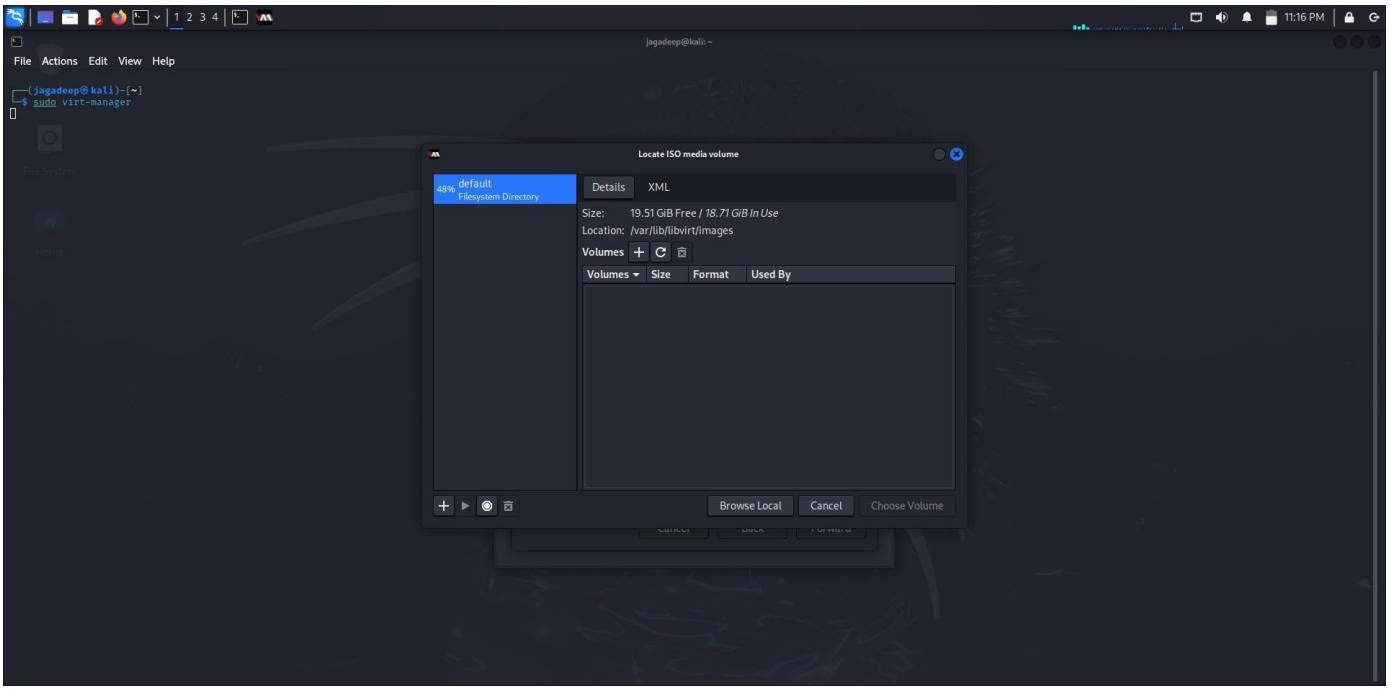
Now sudo virt-manager should open up an GUI for creating VM's



Now click on file, cerate new VM. And follow the further steps in setting up the vm. We will be asked to select iso file as well.



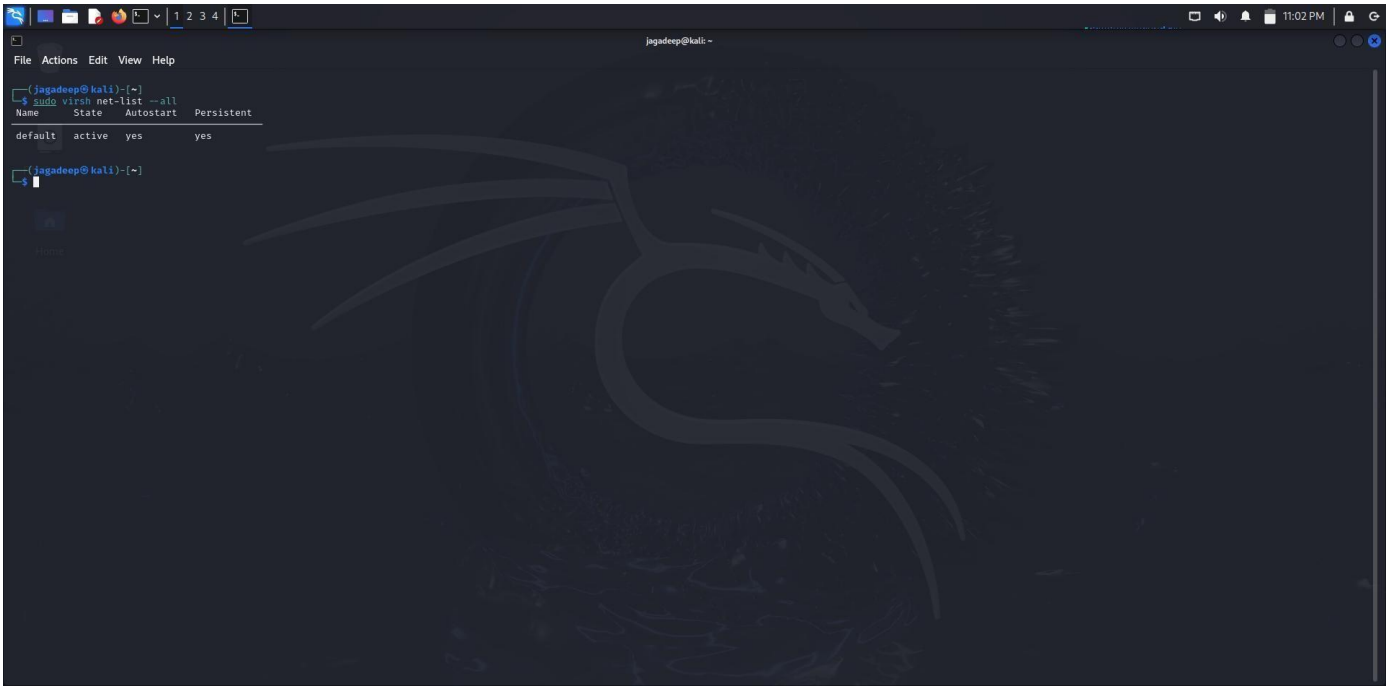




After setting up the virtual machine we can confirm it by using the following command

Sudo virsh net-list –all

This will list all the virtual machine running. We can confirm the status of the created virtual machine.



To enter the created virtual machine, there are various options available

Method	Command
GUI (virt-manager)	`virt-manager` → Click VM
Terminal Console	`virsh console ubuntu-vm1`
SSH (if configured)	`ssh user@vm-ip`
VNC (for remote GUI)	`virsh vncdisplay ubuntu-vm1`

Conclusion:

P

ractical-4 Aim: Working with IaaS of Public

cloud platforms **Google Cloud:**

In the following manual we are going to work Google Cloud Platform. GCP is now providing more than 200 services for its users. Let was take an overview before heading into practical.

Google Cloud IaaS (Infrastructure as a Service) provides scalable computing, storage, and networking resources on demand.

Compute Engine is its core IaaS service, offering customizable virtual machines (VMs) for various workloads.

Persistent Disk and Local SSDs provide high-performance storage options for VMs.

Cloud Load Balancing distributes traffic across multiple instances for reliability and performance.

VPC (Virtual Private Cloud) allows secure networking with fine-grained control over IP ranges and firewall rules.

Preemptible VMs and Spot VMs offer cost-effective options for non-critical workloads.

Autoscaling dynamically adjusts the number of VMs based on demand.

Identity and Access Management (IAM) ensures secure access control for cloud resources.

Cloud Storage integrates with IaaS for scalable object storage.

Compute Engine integrates with Kubernetes (GKE) and AI/ML services for advanced workloads.

IAAS Offering by GCP:

GCP offers several Infrastructure as a Service (IaaS) option, including Compute Engine, Cloud Storage, Cloud SQL, Cloud Spanner, and Google Kubernetes Engine (GKE).

Now Let us get familiar with Compute Engine Service in GCP.

Google Compute Engine:

Google Cloud **Compute Engine** is an **Infrastructure as a Service (IaaS)** offering that provides scalable, flexible, and customizable **virtual machines (VMs)** to run workloads in the cloud. Here are key aspects of Compute Engine:

1. Virtual Machines (VMs)

- Offers different machine types: Standard, High-Memory, High-CPU, and GPU-based VMs.
- Supports custom VM configurations with specific CPU and RAM requirements.

2. Persistent Storage Options
   - Persistent Disks (Standard & SSD): Durable and high-performance storage.
   - Local SSDs: Ultra-fast, temporary storage attached to VMs.
   - File store: Managed file storage for applications requiring shared file access.
3. Networking Features
   - Virtual Private Cloud (VPC): Custom networks, firewall rules, and subnets.
   - Load Balancing: Distributes traffic across multiple VM instances.
   - Cloud NAT: Provides outbound internet access without exposing internal VMs.
4. Preemptible and Spot VMs
   - Preemptible VMs: Short-lived, cost-effective instances (can be terminated anytime).
   - Spot VMs: Even cheaper and suitable for batch jobs or fault-tolerant applications.
5. Autoscaling & Instance Groups
   - Managed Instance Groups (MIGs): Automatically scales VM instances based on demand.
   - Unmanaged Instance Groups: Manually managed set of VM instances.
6. Security & Identity Management
   - IAM (Identity and Access Management) controls access to VMs.
   - Shielded VMs provide advanced protection against rootkits and malware.
7. Operating System Support
   - Supports Linux and Windows Server operating systems.
   - Custom OS images can be uploaded and used in VMs.
8. Integration with Other GCP Services
   - Works seamlessly with Cloud Storage, BigQuery, Kubernetes (GKE), and AI/ML services.
9. Cost Management

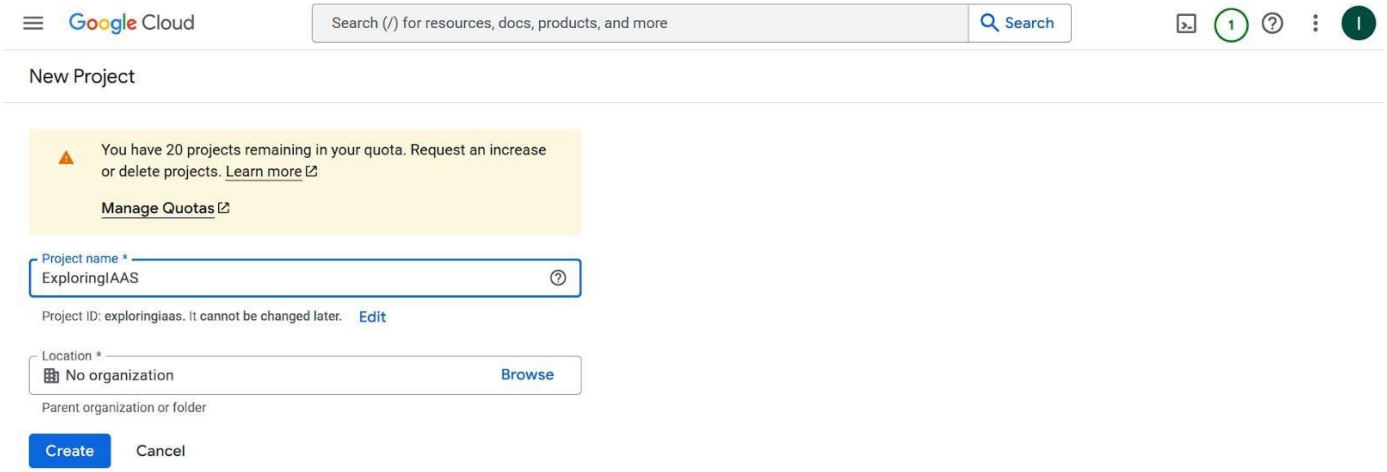
- Offers Sustained Use Discounts (automatic cost reduction for long-running VMs).
- Committed Use Contracts for predictable workloads with further cost savings.

10. Machine Learning & AI Acceleration

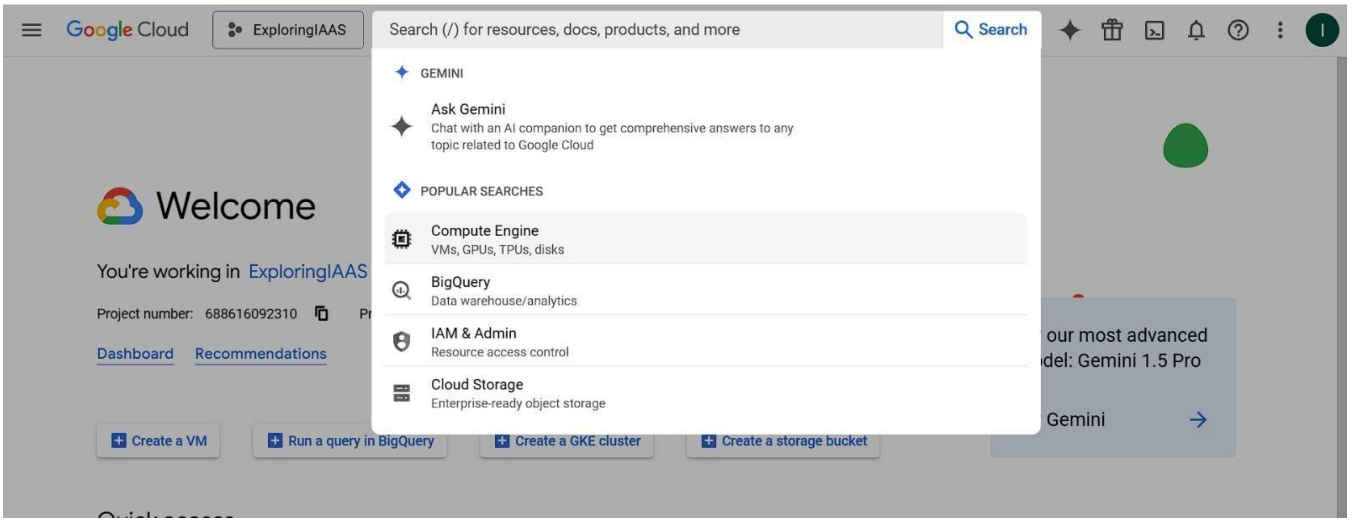
- Supports GPUs and TPUs for AI/ML workloads and high-performance computing (HPC)

Using Compute Engine Service:

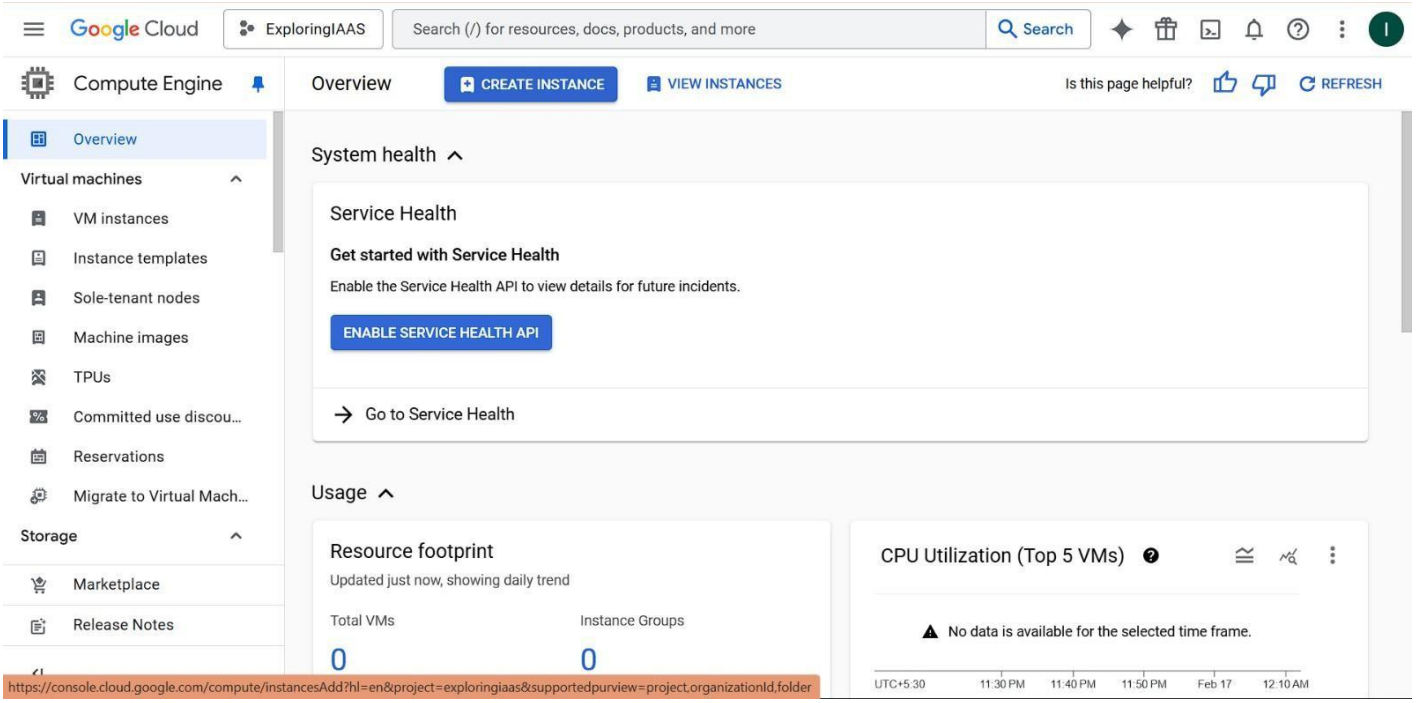
First step is to Create a project. I has Kept name as ExploringIAAS.



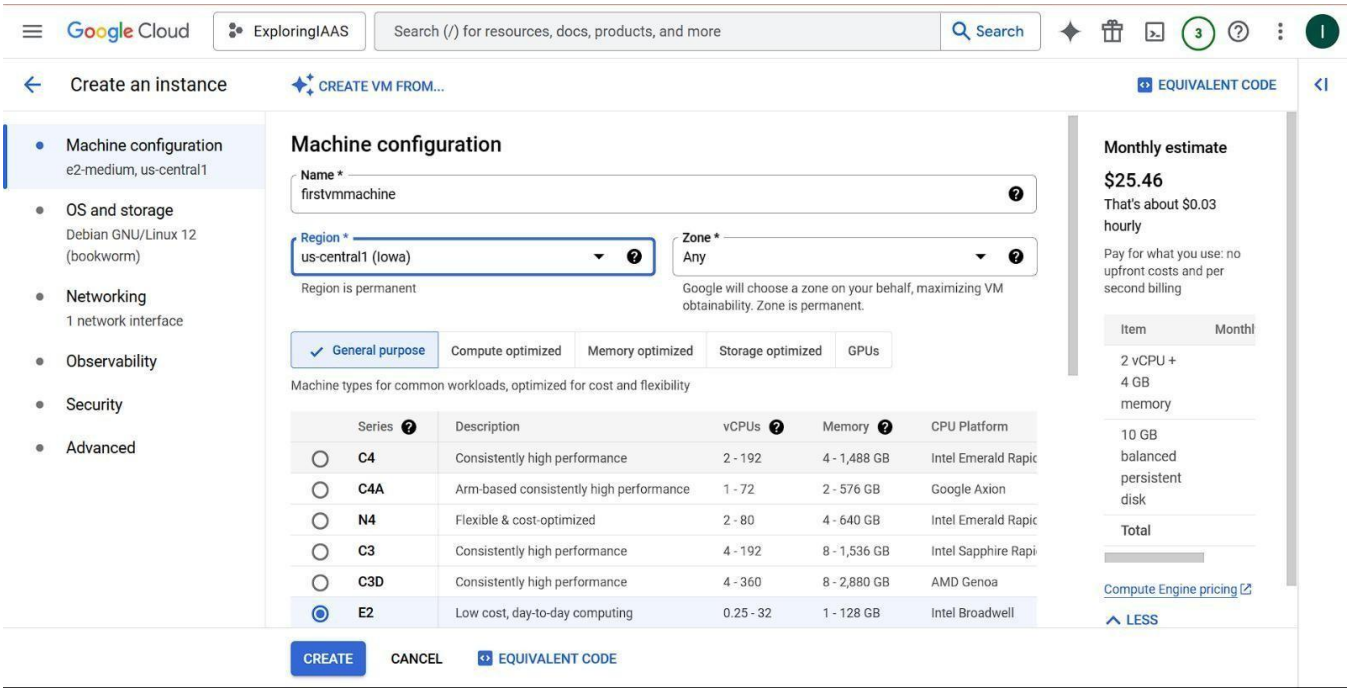
2nd step is to select compute engine service from the search box.



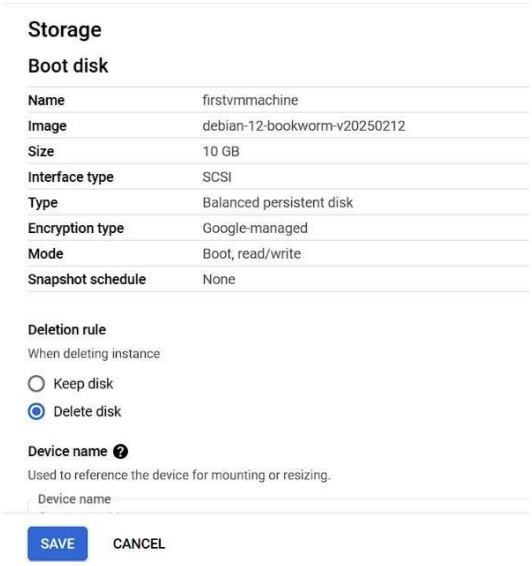
Now 3rd step is to click on create instance option in the compute engine page.



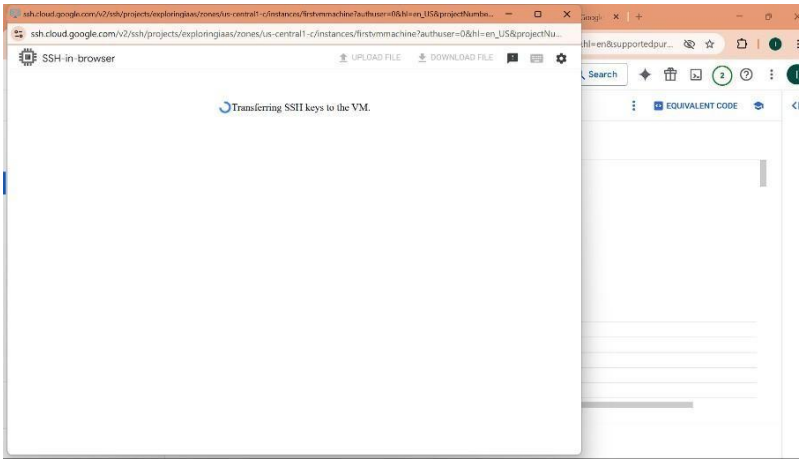
5th step is to add configurations to our virtual machine. This step includes selecting region, selecting virtual machine types, number of CPU cores, and selecting operating system.


Google Cloud
ExploringIAAS
Search (/) for resources, docs, products, and more
Search
Create an instance
CREATE VM FROM...
EQUIVALENT CODE
Machine configuration
e2-medium, us-central1
OS and storage
Debian GNU/Linux 12 (bookworm)
Networking
1 network interface
Observability
Security
Advanced
Machine configuration
Name *
firstvmmachine
Region *
us-central1 (Iowa)
Region is permanent
Zone *
Any
Google will choose a zone on your behalf, maximizing VM obtainability. Zone is permanent.
General purpose
Compute optimized
Memory optimized
Storage optimized
GPUs
Machine types for common workloads, optimized for cost and flexibility
Series
Description
vCPUs
Memory
CPU Platform
C4
Consistently high performance
2 - 192
4 - 1,488 GB
Intel Emerald Rapic
C4A
Arm-based consistently high performance
1 - 72
2 - 576 GB
Google Axion
N4
Flexible & cost-optimized
2 - 80
4 - 640 GB
Intel Emerald Rapic
C3
Consistently high performance
4 - 192
8 - 1,536 GB
Intel Sapphire Rapi
C3D
Consistently high performance
4 - 360
8 - 2,880 GB
AMD Genoa
E2
Low cost, day-to-day computing
0.25 - 32
1 - 128 GB
Intel Broadwell
Monthly estimate
$25.46
That's about $0.03 hourly
Pay for what you use: no upfront costs and per second billing
Item
Monthl
2 vCPU + 4 GB memory
10 GB balanced persistent disk
Total
Compute Engine pricing
LESS
CREATE
CANCEL
EQUIVALENT CODE

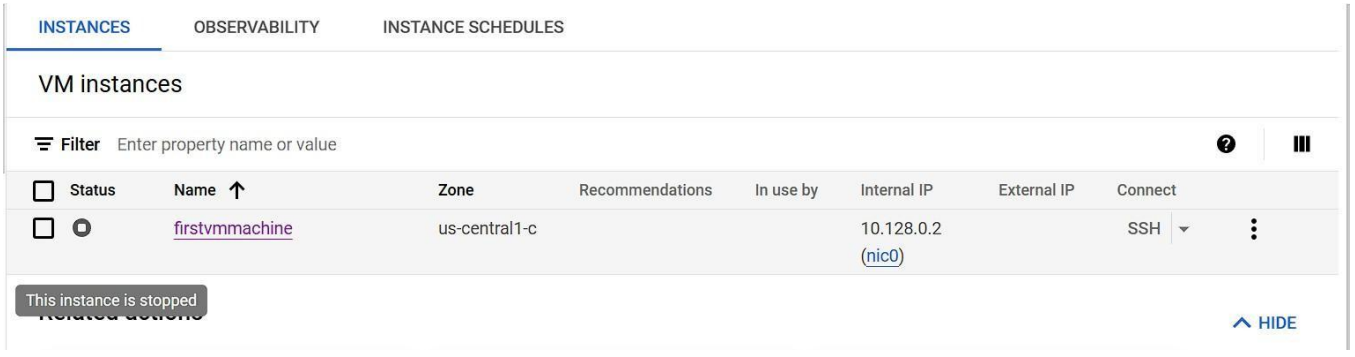
Next step is to select the boot disk configurations.

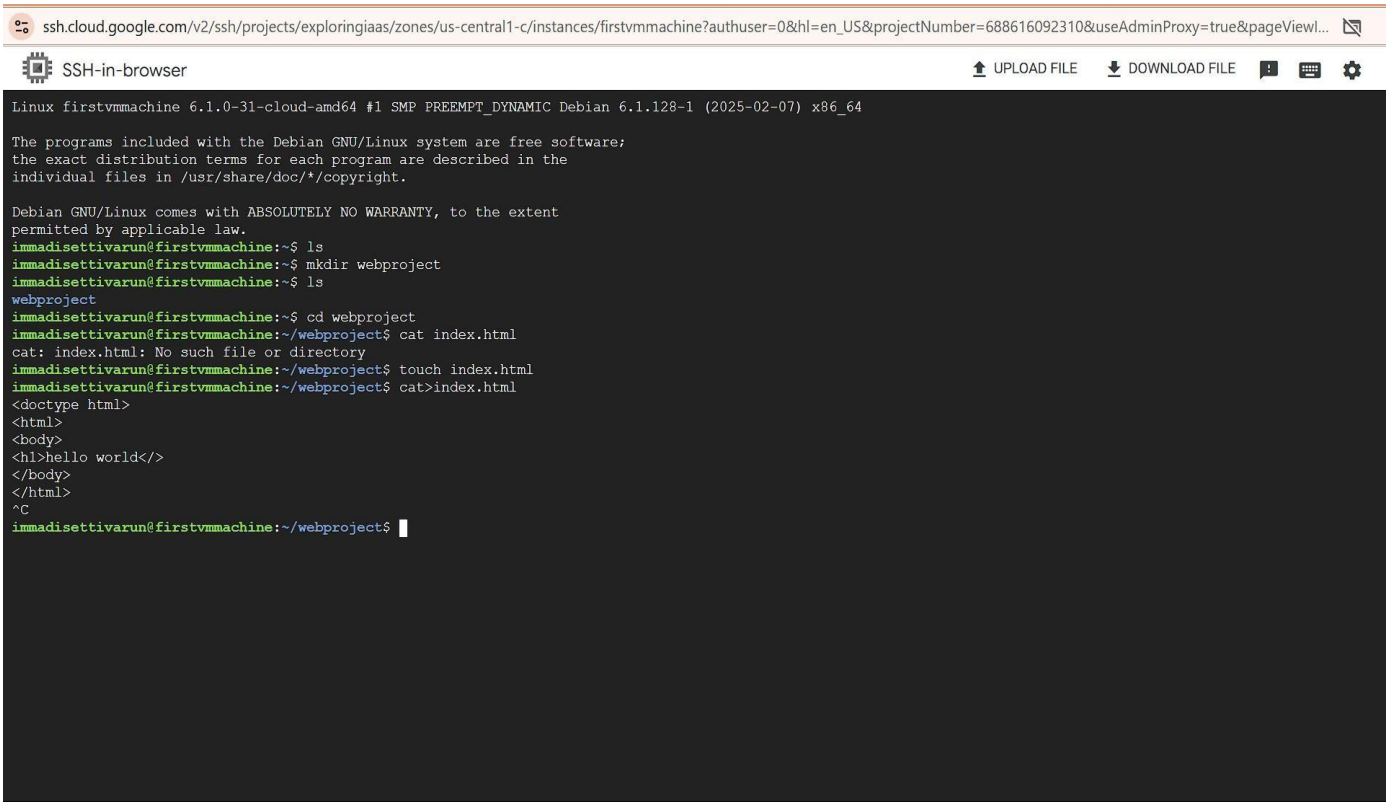


Now we can access the virtual machine using SSH options. After providing Authorization we can use the virtual machine. We can add workloads to the machine.



We can install Apache webservers in through the terminal and host applications on the virtual machine as well. It very important to turn off the virtual machine if it is no longer being user.




ssh.cloud.google.com/v2/ssh/projects/exploringiaas/zones/us-central1-c/instances/firstvmmachine?authuser=0&hl=en_US&projectNumber=688616092310&useAdminProxy=true&pageViewI...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Linux firstvmmachine 6.1.0-31-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.128-1 (2025-02-07) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
immadisettivarun@firstvmmachine:~$ ls
immadisettivarun@firstvmmachine:~$ mkdir webproject
immadisettivarun@firstvmmachine:~$ ls
webproject
immadisettivarun@firstvmmachine:~$ cd webproject
immadisettivarun@firstvmmachine:~/webproject$ cat index.html
cat: index.html: No such file or directory
immadisettivarun@firstvmmachine:~/webproject$ touch index.html
immadisettivarun@firstvmmachine:~/webproject$ cat>index.html
<doctype html>
<html>
<body>
<h1>hello world</>
</body>
</html>
^C
immadisettivarun@firstvmmachine:~/webproject$

Conclusion: