Aim -
Creating and running virtual machines on Hosted Hypervisor like Virtual Box.

Theory -
Kernel-based Virtual Machine (KVM) is an open source virtualization technology built into Linux®. 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).
KVM is part of Linux. If you've got Linux 2.6.20 or newer, you've got KVM. KVM was first announced in 2006 and merged into the mainline Linux kernel version a year later. Because KVM is part of existing Linux code, it immediately benefits from every new Linux feature, fix, and advancement without additional engineering.

How does KVM work?
KVM converts Linux into a type-1 (bare-metal) hypervisor. All hypervisors need some operating system-level components—such as a memory manager, process scheduler, input/output (I/O) stack, device drivers, security manager, a network stack, and more—to run VMs. KVM has all these components because it's part of the Linux kernel. Every VM is implemented as a regular Linux process, scheduled by the standard Linux scheduler, with dedicated virtual hardware like a network card, graphics adapter, CPU(s), memory, and disks.

KVM features
KVM is part of Linux. Linux is part of KVM. Everything Linux has, KVM has too. But there are specific features that make KVM an enterprise's preferred hypervisor.

- Security
  KVM uses a combination of security-enhanced Linux (SELinux) and secure virtualization (sVirt) for enhanced VM security and isolation. SELinux establishes security boundaries around VMs. sVirt extends SELinux's capabilities, allowing Mandatory Access Control (MAC) security to be applied to guest VMs and preventing manual labeling errors.

- Storage
  KVM is able to use any storage supported by Linux, including some local disks and network-attached storage (NAS). Multipath I/O may be used to improve storage and provide redundancy. KVM also supports shared file systems so VM images may be shared by multiple hosts. Disk images support thin provisioning, allocating storage on demand rather than all up front.

- Hardware support
  KVM can use a wide variety of certified Linux-supported hardware platforms. Because hardware vendors regularly contribute to kernel development, the latest hardware features are often rapidly adopted in the Linux kernel.

- Memory management
  KVM inherits the memory management features of Linux, including non-uniform memory access and kernel same-page merging. The memory of a VM can be swapped, backed by large volumes for better performance, and shared or backed by a disk file.

- Live migration
  KVM supports live migration, which is the ability to move a running VM between physical hosts with no service interruption. The VM remains powered on, network connections remain active, and applications continue to run while the VM is relocated. KVM also saves a VM's current state so it can be stored and resumed later.

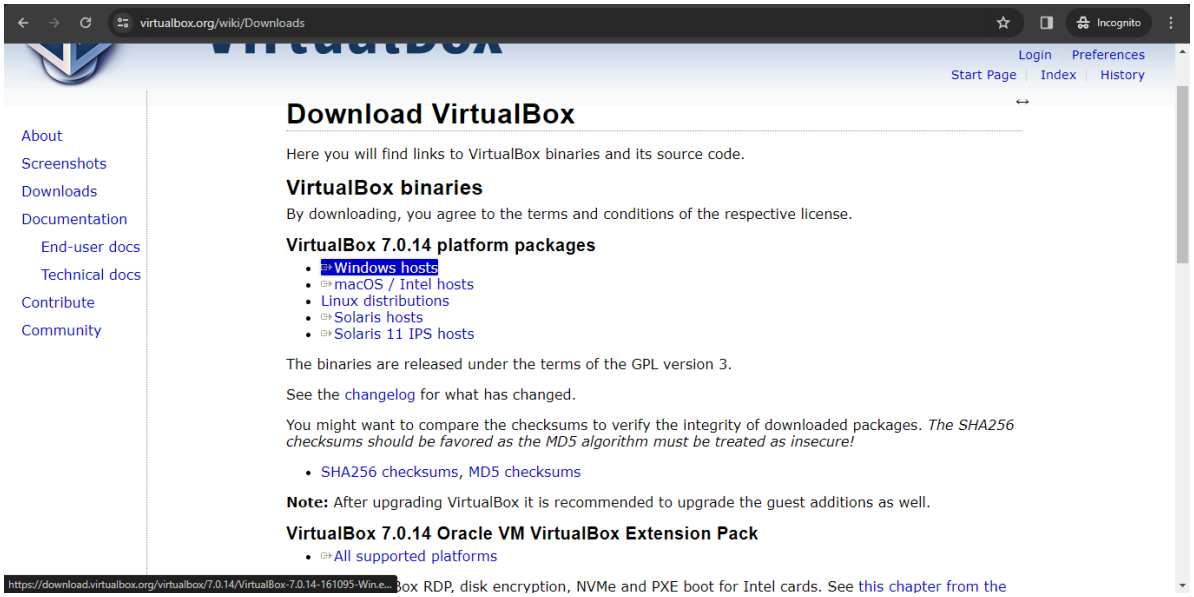
- Performance and scalability
  KVM inherits the performance of Linux, scaling to match demand load if the number of guest machines and requests increases. KVM allows the most demanding application workloads to be virtualized and is the basis for many enterprise virtualization setups, such as data centers and private clouds (via OpenStack®).

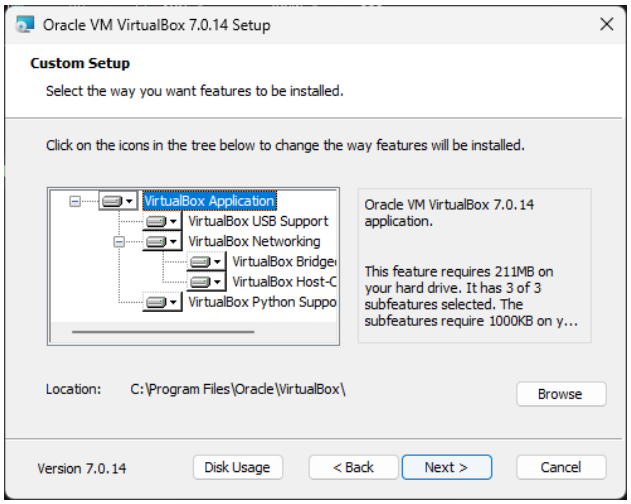
- Scheduling and resource control
  In the KVM model, a VM is a Linux process, scheduled and managed by the kernel. The Linux scheduler allows fine-grained control of the resources allocated to a Linux process and guarantees a quality of service for a particular process. In KVM, this includes the completely fair scheduler, control groups, network name spaces, and real-time extensions.

- Lower latency and higher prioritization
  The Linux kernel features real-time extensions that allow VM-based apps to run at lower latency with better prioritization (compared to bare metal). The kernel also divides processes that require long computing times into smaller components, which are then scheduled and processed accordingly.

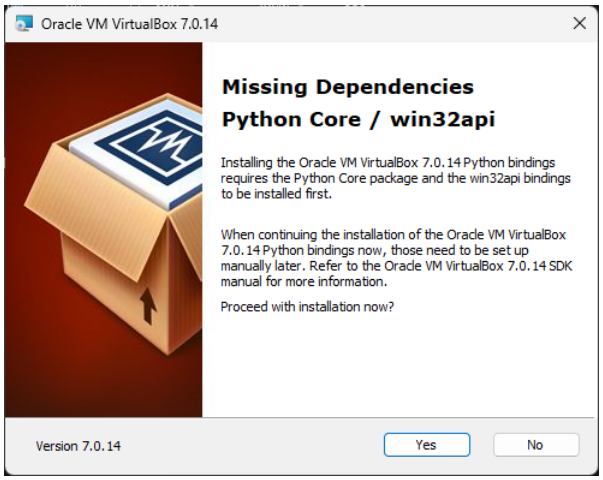
Procedure -

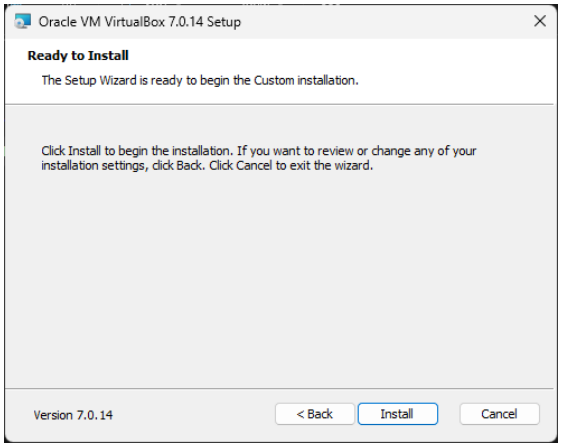
1. Download and install VirtualBox.



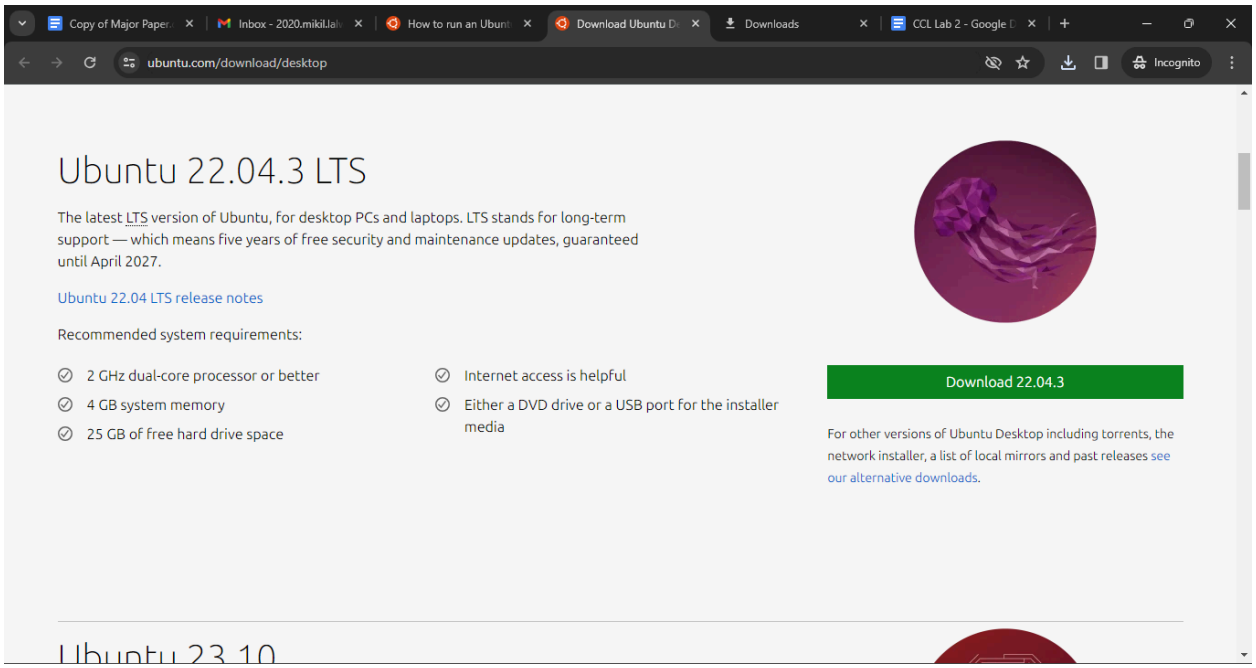



Oracle VM VirtualBox 7.0.14
Warning:
Network Interfaces
Installing the Oracle VM VirtualBox 7.0.14 Networking feature will reset your network connection and temporarily disconnect you from the network.
Proceed with installation now?
Version 7.0.14
Yes
No

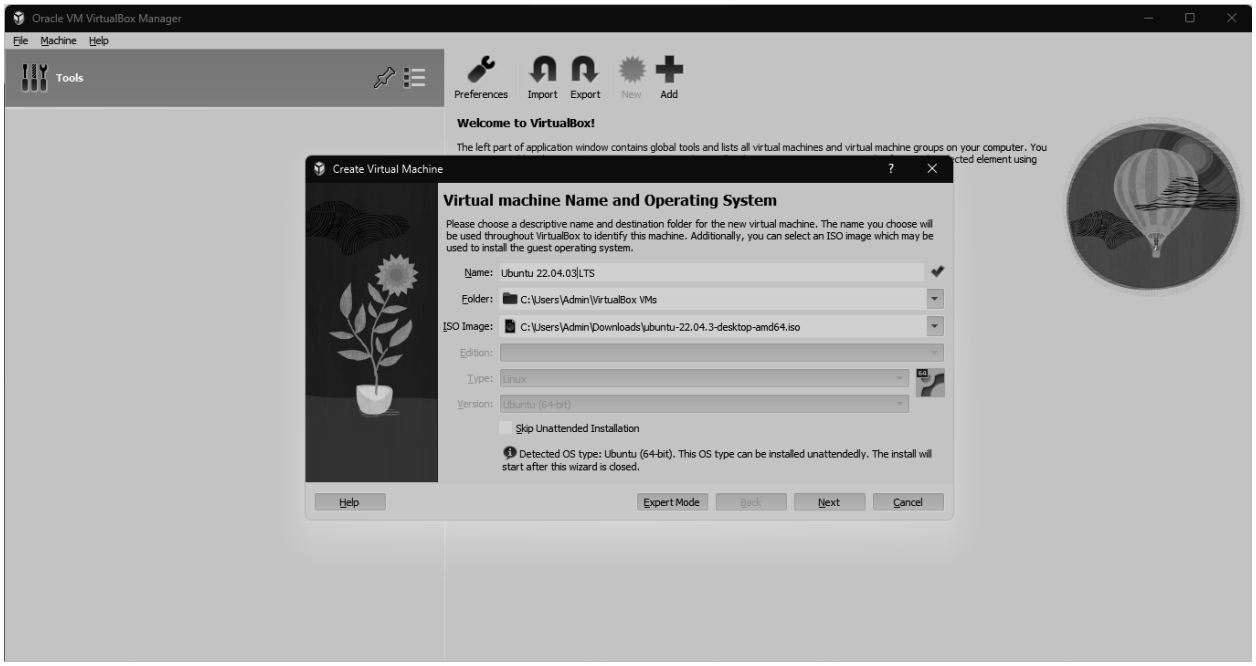

Oracle VM VirtualBox 7.0.14
Missing Dependencies
Python Core / win32api
Installing the Oracle VM VirtualBox 7.0.14 Python bindings requires the Python Core package and the win32api bindings to be installed first.
When continuing the installation of the Oracle VM VirtualBox 7.0.14 Python bindings now, those need to be set up manually later. Refer to the Oracle VM VirtualBox 7.0.14 SDK manual for more information.
Proceed with installation now?
Version 7.0.14
Yes
No


Oracle VM VirtualBox 7.0.14 Setup
Ready to Install
The Setup Wizard is ready to begin the Custom installation.
Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
Version 7.0.14
< Back
Install
Cancel

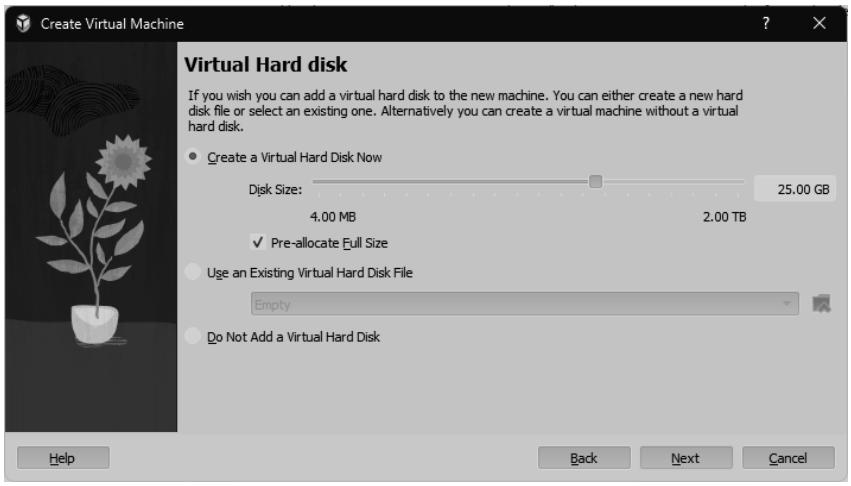
2. Download the latest Ubuntu ISO to create a new VM.

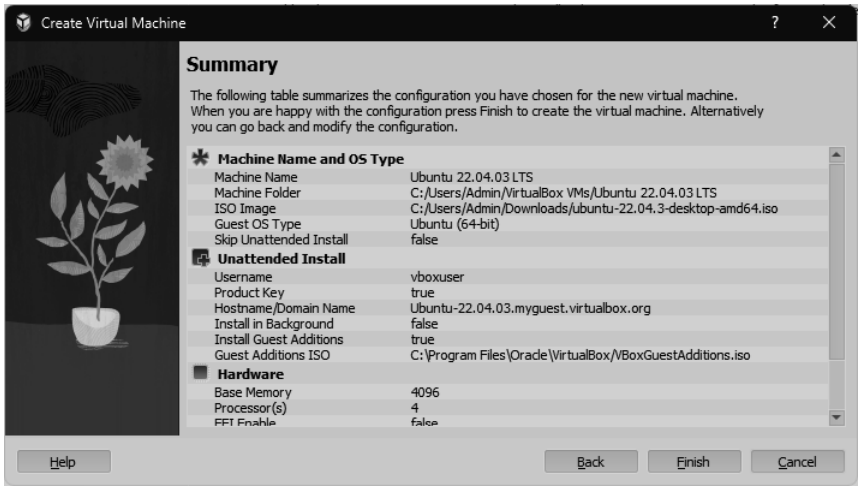


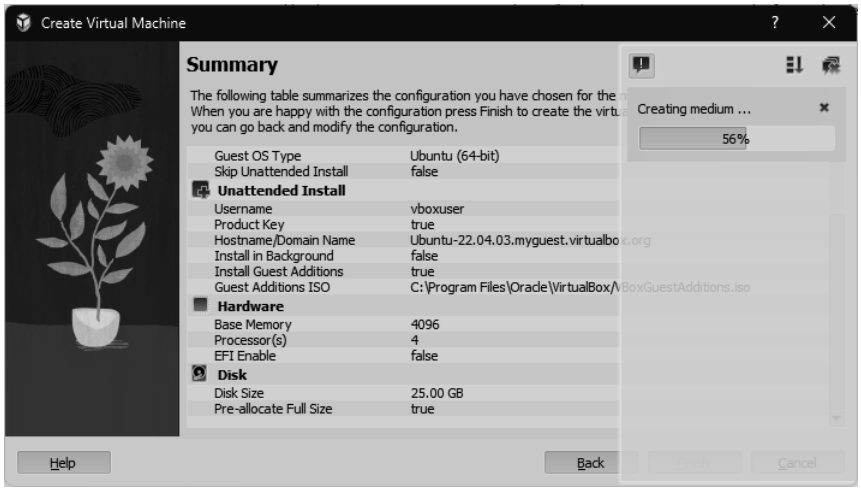
3. Create a new VM by clicking on “New” and follow the instructions below.


Oracle VM VirtualBox Manager
File Machine Help
Tools
Preferences Import Export New Add
Welcome to VirtualBox!
The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You
Create Virtual Machine
Virtual machine Name and Operating System
Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.
Name: Ubuntu 22.04.03|LTS
Folder: C:\Users\Admin\VirtualBox VMs
ISO Image: C:\Users\Admin\Downloads\ubuntu-22.04.3-desktop-amd64.iso
Edition:
Type: Linux
Version: Ubuntu (64-bit)
Skip Unattended Installation
Detected OS type: Ubuntu (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.
Help
Expert Mode
Back
Next
Cancel


Create Virtual Machine
Unattended Guest OS Install Setup
You can configure the unattended guest OS install by modifying username, password, and hostname. Additionally you can enable guest additions install. For Microsoft Windows guests it is possible to provide a product key.
Username and Password
Additional Options
Username: vboxuser
Password: changeme
Repeat Password: changeme
Product Key: #####-#####-#####-#####
Hostname: Ubuntu-22.04.03
Domain Name: myguest.virtualbox.org
Install in Background
Guest Additions
Guest Additions ISO: C:\Program Files\Oracle\VirtualBox\VBoxGuestAdditions.iso
Help
Back
Next
Cancel


Create Virtual Machine
Virtual Hard disk
If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.
Create a Virtual Hard Disk Now
Disk Size: 25.00 GB
4.00 MB
2.00 TB
Pre-allocate Full Size
Use an Existing Virtual Hard Disk File
Empty
Do Not Add a Virtual Hard Disk
Help
Back
Next
Cancel





Conclusion -

Thus we have successfully learned about creating and running virtual machines on Hosted Hypervisor like Virtual Box.