Bahir Dar University
Institute of Technology
Faculty of Computing

Department of Software Engineering
Operating System and System Programming
Individual Assignment

Name:
Tnsaie Hunegnaw merid
ID: BDU 1203170
Section: A

Submitted to :
Mr. Wendmu Baye
Submission
Date :16/08/2017
E.C

BACKGROUND

Debian is a widely respected, stable, and versatile open-source operating system based on the Linux kernel. It has been at the heart of many other distributions (like Ubuntu), and is known for its robustness, package management (APT), and strong community support. Due to its stability, it's frequently used in both desktop and server environments.

On the other hand, VMware Workstation is a powerful virtualization tool that allows users to run multiple operating systems simultaneously on a single physical machine. It is especially popular among developers, testers, and students who need isolated environments for different tasks without needing separate hardware.

MOTIVATION

So, **why run Debian OS on VMware Workstation?** Great question—and here's where the motivation really shines:

✅ Learning and Experimentation

Running Debian on VMware is a *safe and flexible playground*. You can test commands, install packages, break things, fix them again—all without affecting your host OS. This is especially useful for students, developers, or anyone learning Linux system administration, programming, or networking.

✅ Software Development and Testing

Developers often need to test applications in different environments. Using a Debian VM allows you to ensure your software runs well on Debian-based systems. You can snapshot, revert, or clone VMs easily for different stages of testing.

✅ Stability for Long-Term Projects

Debian is known for its long-term support and rock-solid reliability. When combined with VMware's flexibility, it becomes ideal for setting up reliable, persistent development environments or simulated server environments.

✅ Cross-Platform Advantage

VMware Workstation supports Windows and Linux hosts, so running Debian as a VM gives you the best of both worlds. You can work in a Windows environment while still developing or experimenting in Linux.

✅ Networking and Server Practice

Want to practice setting up a server, DNS, DHCP, or firewall on Linux? With Debian on VMware, you can simulate an entire network setup on your laptop or PC without touching your real network.

The primary objective of deploying **Debian OS on VMware Workstation** is to create a flexible, isolated, and controlled environment for **learning, testing, development, and system administration** purposes. This setup enables users to:

Gain Practical Experience with Linux (Debian)
Develop hands-on skills in using the Debian operating system, including command-line navigation, package management, and system configuration.

Explore Virtualization Concepts
Understand and apply virtualization techniques using VMware Workstation, including VM creation, snapshotting, network configuration, and resource allocation.

Practice System and Network Administration
Simulate real-world server setups such as web servers, DNS, DHCP, file sharing, or firewall configuration within the Debian environment.

Support Software Development and Testing
Create a stable platform for compiling, debugging, and testing software in a Linux environment without affecting the host machine.

Enhance Learning without Risk
Provide a risk-free platform for experimentation where users can explore advanced Linux functionalities, troubleshoot errors, and recover from mistakes easily using VMware snapshots and backups.

Improve Academic and Career Readiness
Build a strong foundation in Linux system management and virtualization—valuable skills for IT professionals, engineers, and students preparing for roles in system administration, DevOps, or cybersecurity.

This objective makes your work sound serious, purposeful, and clearly aligned with practical skills and academic goals. Want me to help you write it in a more formal tone or adapt it for a specific report or assignment? Just say the word!

HARDWARE AND SOFTWARE REQUIREMENTS

✅ Hardware Requirements (Host Machine)

To run Debian smoothly inside VM ware Workstation, your physical computer (the *host*) should meet or exceed the following minimum specifications:

Component	Minimum Requirement	Recommended for Better Performance
Processor (CPU)	Dual-core 64-bit processor (Intel/AMD)	Quad-core or higher with virtualizatio support(Intel VT-x / AMD-V)
RAM (Memory)	4 GB	8 GB or more (especially if running multiple VMs)
Storage	At least 20 GB free disk space	SSD with 40 GB+ free space for better performance
Display	1024×768 resolution	1920×1080 or higher
Other	Virtualization support enabled in BIOS/UEFI	—

✅ Software Requirements

You’ll need the following software components

1.VMware Workstation / VMware Workstation Player

VMware Workstation Pro (paid) or VMware Workstation Player (free for personal use) Compatible with your operating system (Windows/Linux)

Debian ISO File

Download the latest Debian stable release from the official Debian website

Choose the appropriate architecture (usually *amd64* for modern systems)

Operating System (Host)

Windows 10/11 or any modern Linux distressdistro

VM ware Tools (Optional but Recommended)

Enhances VM performance (graphics, mouse, file sharing, etc.)

Can be installed inside the Debian Vm

✅ Optional (But Useful) Tools

Terminal Emulator or SSH Client (like PuTTY or Terminator) for remote access and practice

Text Editor (like VS Code, Sublime Text, or Nano inside Debian)

Internet Connection for package updates and software installation

Installation Steps

🔧 Pre-requisites

VMware Workstation installed on your system.

Debian 12 ISO image (Download from debian.org)

Installation Steps

🔧 Pre-requisites

VMware Workstation installed on your system.

Debian 12 ISO image (Download from debian.org).

Installation Steps

1. Create a New Virtual Machine

Open **VMware Workstation**.

Click **"Create a New Virtual Machine"** (or `File > New Virtual Machine`).

Choose **"Typical (recommended)"**, click **Next**.

Select **"Installer disc image file (iso)"**, then **browse to your Debian 12 ISO**, click **Next**.

2. Guest Operating System

Choose **Linux** as the Guest OS.

Select **Debian 11 or later versions** (if Debian 12 isn't listed), then click **Next**.

3. Name the VM & Location

Give your VM a name, e.g., **Debian12-VM**.

Choose a location for VM files, click **Next**.

4. Specify Disk Capacity

Set disk size (e.g., **20 GB or more**, depending on use case).

Choose **"Store virtual disk as a single file"**.

Click **Next**.

5. Customize Hardware (Optional)

-

You can click **"Customize Hardware"** to:

Allocate more **RAM** (e.g., 2048 MB or more).

Increase **CPU cores**.

Adjust **network settings**.

Click **Close**, then **Finish**.

Debian Installation Process

Once the VM boots into the ISO:

6. Choose Installation Mode

Select **Graphical Install** (recommended).

7. Language, Location, Keyboard

Select your **language**, **location**, and **keyboard layout**.

8. Configure Network

The installer will attempt to configure your network via DHCP.

You can set hostname (e.g., `debian12`) and domain name (optional).

9. Set Up Users

Set **root password** (if prompted), or just create a **regular user** with sudo privileges.

Create your username and password.

10. Partition Disks

Choose **Guided - use entire disk** (recommended for VMs).

Select the virtual disk.

Choose a partition scheme (e.g., **All files in one partition**).

Confirm to **write changes to disk**

11. Install Base System

Let the installer copy and install base system files.

12. Configure Package Manager

Choose a **mirror location** (e.g., `deb.debian.org`).

You may be asked about **proxy settings** – leave blank unless needed.

13. Software Selection

Select desired environments:

Debian desktop environment (default GNOME, or choose KDE, XFCE, etc.)

Standard system utilities

SSH server (optional)

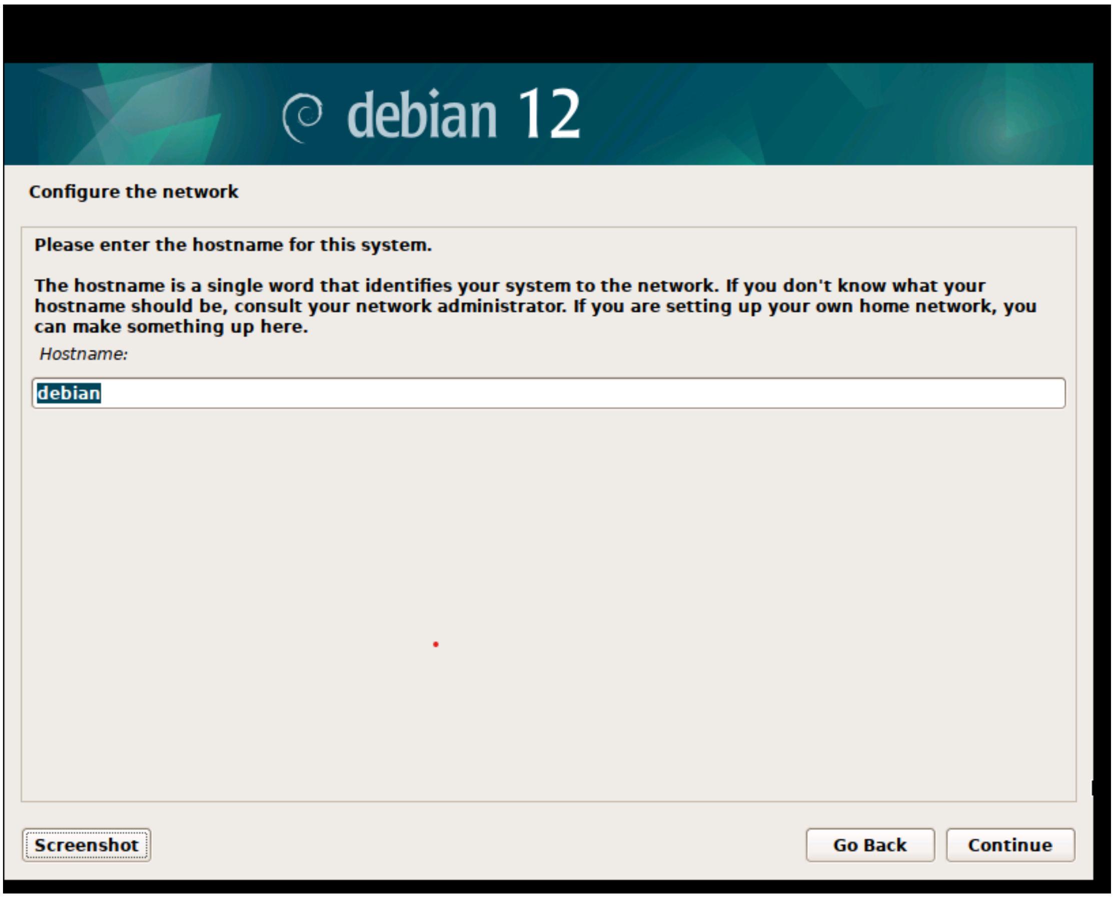
Continue to install selected packages.

14. Install GRUB Bootloade Choose to install GRUB to the master boot record Select the target disk (e.g., `/dev/sda`).

✅ 15. Finish Installatio

After installation, remove ISO when prompted

Click **Continue** to reboot into your new Debian 12 VM.



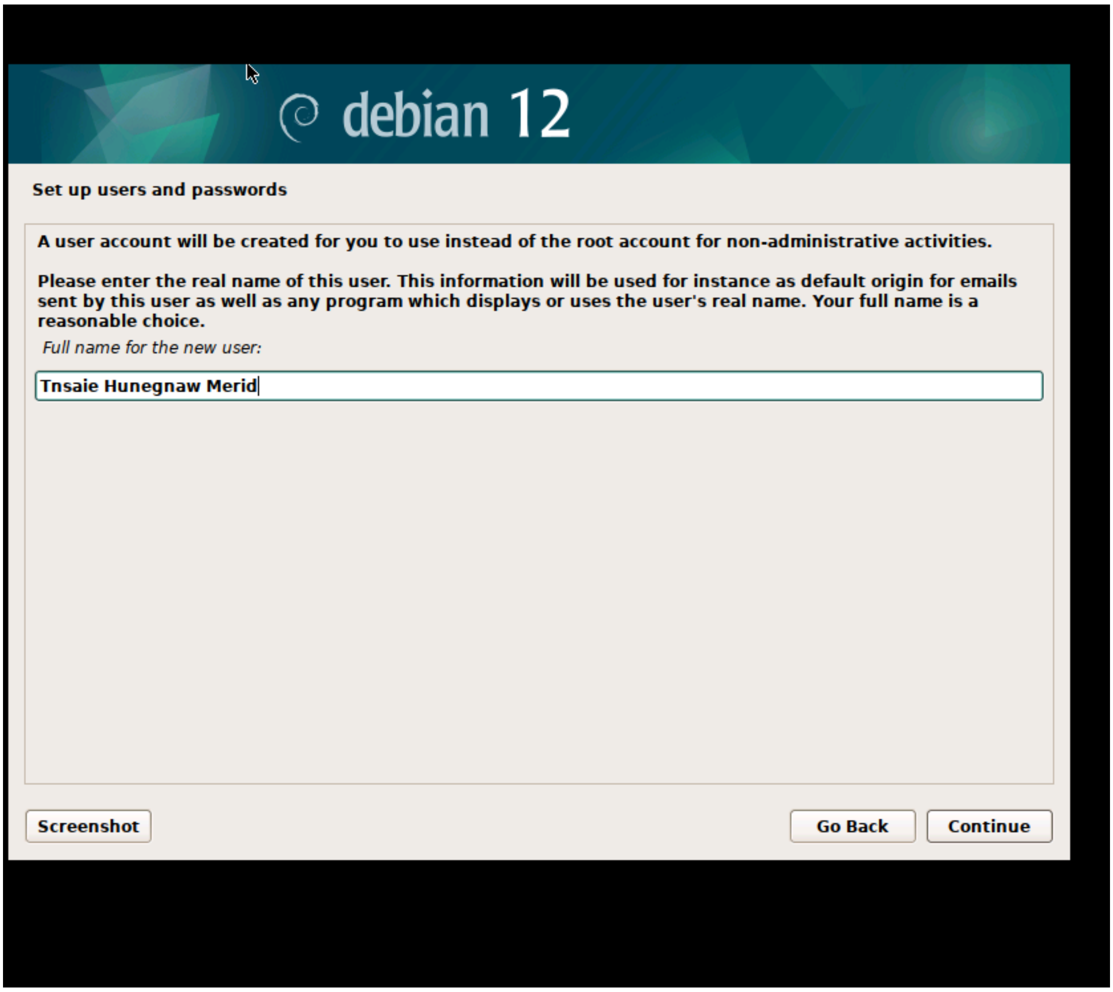
Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

debian



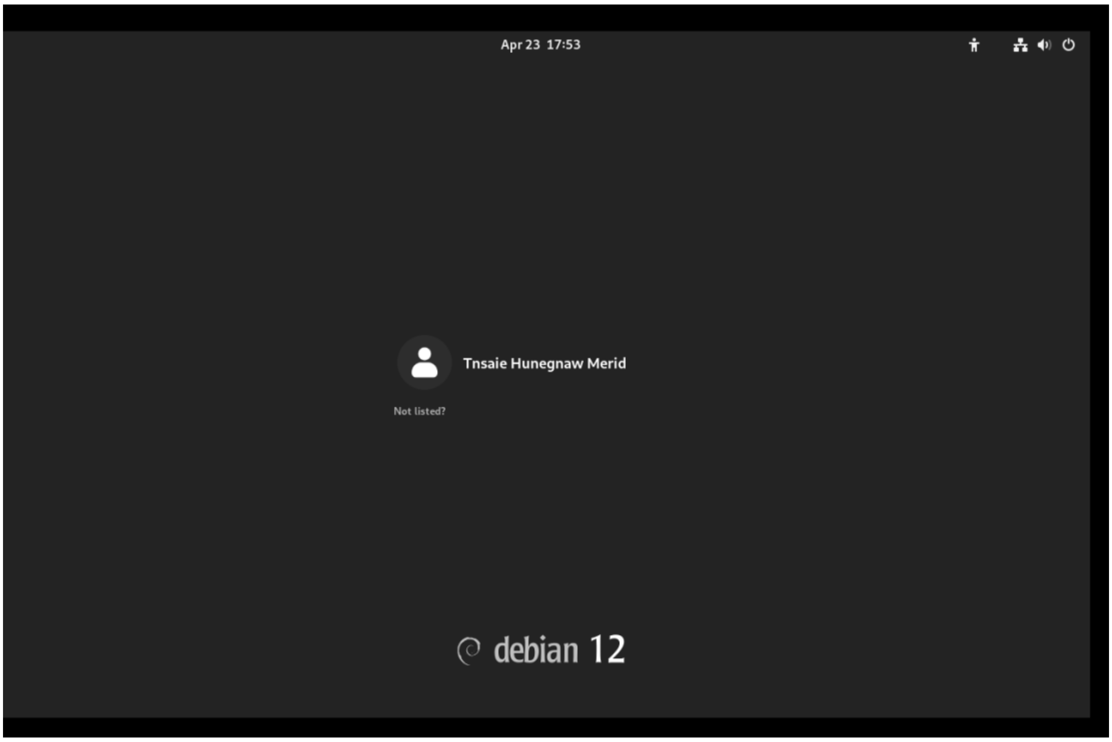
Set up users and passwords

A user account will be created for you to use instead of the root account for non-administrative activities.

Please enter the real name of this user. This information will be used for instance as default origin for emails sent by this user as well as any program which displays or uses the user's real name. Your full name is a reasonable choice.

Full name for the new user:

Tnsaie Hunegnaw Merid



File systems support

File Systems Supported by Debian 12

File System	Type	Common Use Case	Why Use It
ext4	Native	Default file system	Reliable, stable, mature, good performance
ext3	Native	Older systems	Journaling support, backward compatible
ext2	Native	USB drives, low-resource systems	No journaling = less write overhead
XFS	High-performance	Servers, large files	High-performance, scales well
Btrfs	Advanced	Snapshots, RAID, self-healing	Modern, supports subvolumes, checksums
F2FS	Flash storage	SSDs, embedded devices	Optimized for flash memory

File System	Type	Common Use Case	Why Use It
NTFS	Windows	Interoperability	Read/write support via ntfs-3g
exFAT	External drives	USBs shared with Windows/macOS	No 4GB limit, cross-platform
VFAT/FAT32	Legacy	Older USB drives, boot partitions	Very compatible, but limited features
ZFS	Advanced	Data integrity, enterprise	Snapshot, compression, self-healing (needs external packages like zfs-dkms)
ISO 9660 / UDF	Optical media	CD/DVD support	Read-only or universal disk format

💡 Why Use These File Systems?

ext4: Default for a reason — it’s robust, tested, supports journaling, and performs well in most general use cases.

XFS: Great for systems that manage large files (e.g., video editing, servers).

Btrfs: Ideal if you want features like snapshots, built-in RAID, or a modern copy-on-write FS (think of it as Linux’s version of APFS or ZFS).

F2FS: Specifically designed for flash memory, improving performance and longevity of SSDs.

NTFS / exFAT / FAT32: Mostly used for compatibility with other operating systems, especially when sharing drives with Windows.

ZFS: Known for its data integrity features and ability to scale, although it’s not included by default due to licensing issues (needs to be installed separately).

Advantage and Disadvantage of Installing Debian 12 server on Vmware workstation

✅ Advantages

1. 💻 Safe Testing Environment

You can try updates, scripts, or configurations without risking your main system.

Perfect for **learning Linux server administration**.

2. Snapshots & Rollbacks

VMware allows **snapshots**—easily roll back if something breaks.

Great for testing server configs or updates

3. Multi-VM Setup

Create a **lab environment** with multiple VMs (e.g., one as a web server, another as a DB).

Simulate networked services or clustering.

4. Convenience

Run Debian server alongside your host OS (e.g., Windows).

No need for dual-boot or a dedicated physical server.

5. Portability

VMs are **just files** — easy to back up, move, or clone

6 Isolation

Any crashes or security issues in Debian won’t affect your host machine.

Disadvantages

1. Performance Overhead

VM is **slower than bare metal** — you share CPU, RAM, and disk I/O with the host OS.

Not ideal for heavy workloads like databases or large-scale web apps.

2. Hardware Access is Limited

No direct access to GPU, advanced network adapters, or specific hardware peripherals.

3. VMware Dependency

Requires **VMware Workstation license** (though there's a free Player version).

Not a "true" server deployment; can behave differently than real hardware.

4. Networking Complexity

Bridged vs NAT vs Host-only networking can be confusing to set up, especially for services that need public access.

5. Not Production-Grade

VMware Workstation is a **desktop virtualization tool**, not intended for production server deployments (unlike VMware ESXi or Proxmox).

⚖ Conclusion: When It Makes Sense

Use **Debian 12 Server on VMware Workstation** when you need:

A sandbox to learn, test, or prototype.

A flexible, isolated, and easily resettable environment.

To avoid dedicating real hardware to your project

Avoid it if you're building a **high-performance**, **production**, or **hardware-specific** deployment

Conclusion about Installing Debian 12 server on Vmware workstation

✅ Conclusion

Installing **Debian 12 Server** on **VMware Workstation** provides a reliable and flexible virtual environment ideal for development, testing, or production use. This process involves configuring the virtual machine, selecting the appropriate ISO image, and following Debian’s guided installation steps. Once completed, users benefit from a stable, secure, and lightweight server environment with full virtualization capabilities. With proper setup and updates, your Debian 12 server can serve as a powerful foundation for a wide range of applications and services.

Future out look about Installing Debian 12 server on Vmware workstation

🔮 Future Outlook

As virtualization continues to play a central role in IT infrastructure and development environments, the combination of **Debian 12 Server** and **VMware Workstation** remains highly relevant. Debian’s long-standing reputation for stability and security makes it a solid choice for running virtualized servers, especially in development, lab, and even some production environments.

With ongoing support from both the Debian community and VMware, users can expect continued compatibility, improved hardware integration, and security patches. Additionally, as Debian evolves and VMware introduces enhancements (like better resource handling, virtual hardware upgrades, and more seamless host-guest integration), the experience of running Debian in a virtualized setting will only become more efficient and user-friendly.

Looking ahead, this setup is likely to remain a go-to solution for developers, system admins, and learners seeking a versatile and dependable Linux server environment without the need for dedicated hardware.

Virtualization in Modern Operating Systems

What is Virtualization?

Virtualization refers to the creation of virtual versions of physical resources, such as servers, storage devices, or operating systems, on a host machine. In modern operating systems, virtualization enables the simulation of hardware and the running of multiple virtual instances (virtual machines or VMs) on a single physical machine.

Types of Virtualization:

Hardware Virtualization: Simulates physical hardware to run multiple OS instances on a host.

OS-level Virtualization: Runs multiple isolated instances of the same OS (e.g., containers).

Application Virtualization: Separates the application from the underlying OS environment, allowing apps to run in different environments without modification

Why is Virtualization Important?

Virtualization offers a wide range of benefits in modern operating systems and computing environments:

Resource Efficiency:

Virtualization enables better resource utilization by allowing multiple VMs to share the same physical resources (CPU, RAM, storage).

Instead of having underutilized physical servers, organizations can run several virtual machines on one physical server, optimizing both hardware and energy consumption

Isolation and Security:

Virtual machines are isolated from one another, meaning that issues or crashes in one VM do not affect others. This isolation enhances security and stability, especially in multi-tenant environments.

Virtualization enables easy segmentation of workloads and the ability to run different operating systems on the same hardware, improving security.

Scalability and Flexibility:

Virtualization allows easy scaling of resources. New VMs can be created quickly, and existing VMs can be adjusted as needed. This dynamic environment is crucial for businesses that need to respond rapidly to changing demands.

Cost Savings:

Virtualization reduces the need for physical hardware, leading to cost savings in terms of both hardware and maintenance

It also enables organizations to consolidate resources, reducing physical space requirements and energy consumption.

Simplified Management:

Virtual environments offer centralized management tools for monitoring and administering multiple VMs, simplifying operations and reducing the complexity of managing physical servers.

Disaster Recovery and High Availability

Virtual machines can be backed up, cloned, and migrated easily between physical servers, which supports robust disaster recovery strategies.

Features like VM snapshots and live migration provide high availability and allow organizations to minimize downtime.

○

How Does Virtualization Work in Modern Operating Systems?

Virtualization in modern OSes works by leveraging a **hypervisor**, which is a layer that sits between the hardware and the operating systems (or VMs). There are two main types of hypervisors:**Type 1 Hypervisor (Bare-Metal** This type runs directly on the physical hardware and does not require a host OS. Examples include **VMware ESXi**, **Microsoft Hyper-V**, and **Xen**.

Type 1 hypervisors provide better performance and are often used in data centers and enterprise environments.

Type 2 Hypervisor (Hosted):

This type runs on top of an existing operating system. Examples include **VMware Workstation**, **Oracle VirtualBox**, and **Parallels**.

Type 2 hypervisors are typically used in desktop environments, making them ideal for developers and individuals who want to run multiple OS instances on their personal computers.

Here’s how the process typically works:

Step 1: Host Machine – The physical machine that provides resources (CPU, memory, storage).

Step 2: Hypervisor Layer – The hypervisor is installed on the host machine and manages multiple virtual machines by allocating resources to them.

Step 3: Virtual Machines – Each VM runs its own operating system, which may be different from the host OS. The hypervisor emulates the underlying hardware for the VMs, creating an isolated environment for each one.

VMs can run different operating systems (e.g., Linux, Windows) and have their own applications, configurations, and settings. The hypervisor ensures that the virtual machines function independently of one another and are properly allocated the resources they need.

Virtualization in Modern Operating Systems:

Linux: Linux supports various virtualization technologies, including KVM (Kernel-based Virtual Machine), Xen, and LXC (Linux Containers), allowing users to run multiple virtual environments and containers efficiently.

Windows: Windows supports Hyper-V for virtualization and integrates well with Microsoft's cloud offerings. It also supports Docker containers, enabling the virtualization of applications.

macOS: macOS users can use virtualization tools like Parallels or VMware Fusion to run other OSes on their systems.