

Setting up a virtual home lab

This repository documents the design and deployment of a virtualized IT and cybersecurity lab environment, built entirely with free and open-source tools.

The objective of this project is to simulate enterprise-level infrastructure for hands-on testing, analysis, and skill development in system administration, networking, and cybersecurity operations.

This approach is fast, flexible, and cost-effective, providing an excellent platform to build practical IT and cybersecurity skills, all without the need for physical hardware.

Follow along as I walk you through how I built it and how you can set up your own virtual lab environment, too.

Index

- **Virtualization Overview**
- **Free Virtualization Software**
- **Downloading and Installing VirtualBox**
- **Creating a Virtual Machine**
- **Creating a Virtual Network (Organization Network and connecting VMs to the same NatNetwork)**
- **Configuring Base Memory (RAM Allocation)**
- **Installing an Operating System on Your Lab VMs**
- **Conclusion**

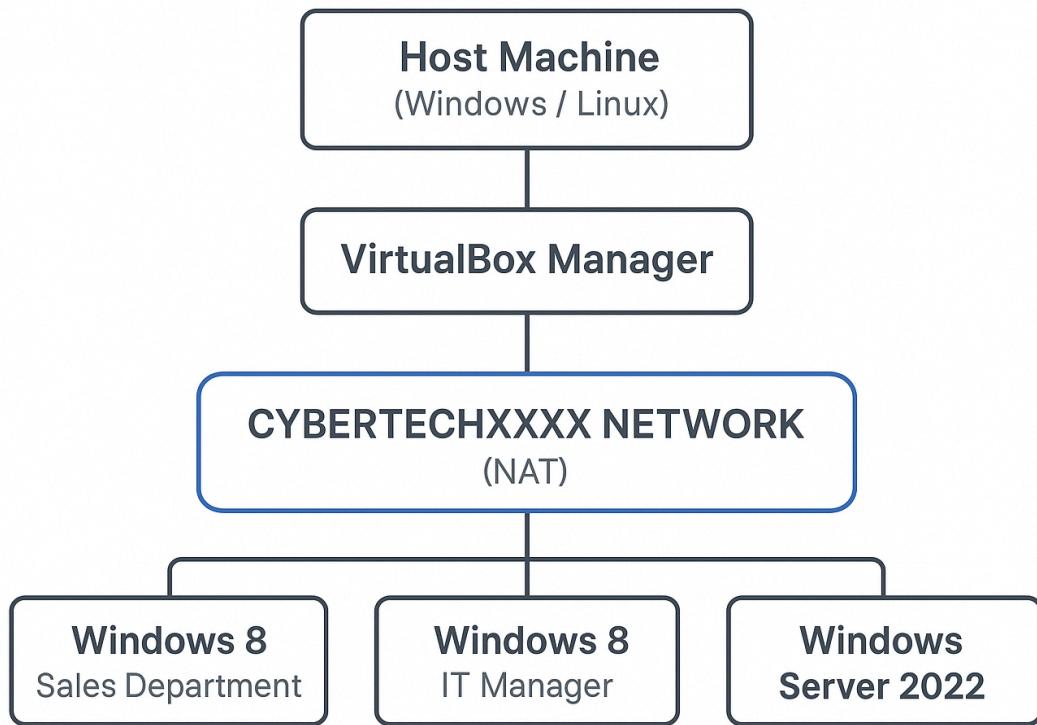
Virtualization Overview

So, what exactly is virtualization?

In simple terms, virtualization is the process of **emulating a computer system**. Instead of needing multiple physical machines, you can create several **virtual machines (VMs)** that run as software on your main computer.

Imagine your main computer is the **host**, the one doing all the work. On this host, you can run one or more **guest virtual machines**, each acting like its own separate computer with its own operating system and settings.

Think of it as running a full computer *inside* your existing one, like opening up a new world on your desktop!



Virtual Machines —> Computers Within a Computer

Virtual Machines (VMs) operate as isolated computing environments that run as software within a host system. Each VM is equipped with its own virtual hardware, operating system, and applications, allowing it to function just like a standalone physical computer.

In cybersecurity, this isolation is essential. Virtual machines provide a controlled and secure sandbox for:

- Penetration testing and ethical hacking without impacting the host system
- Malware analysis in a quarantined, disposable environment

- Network segmentation and traffic inspection
- Configuration testing and system hardening before production deployment

By leveraging virtualization, you can safely build, break, and rebuild complex infrastructures, gaining hands-on experience and technical proficiency without the risk or expense of additional hardware.

Free Virtualization Software

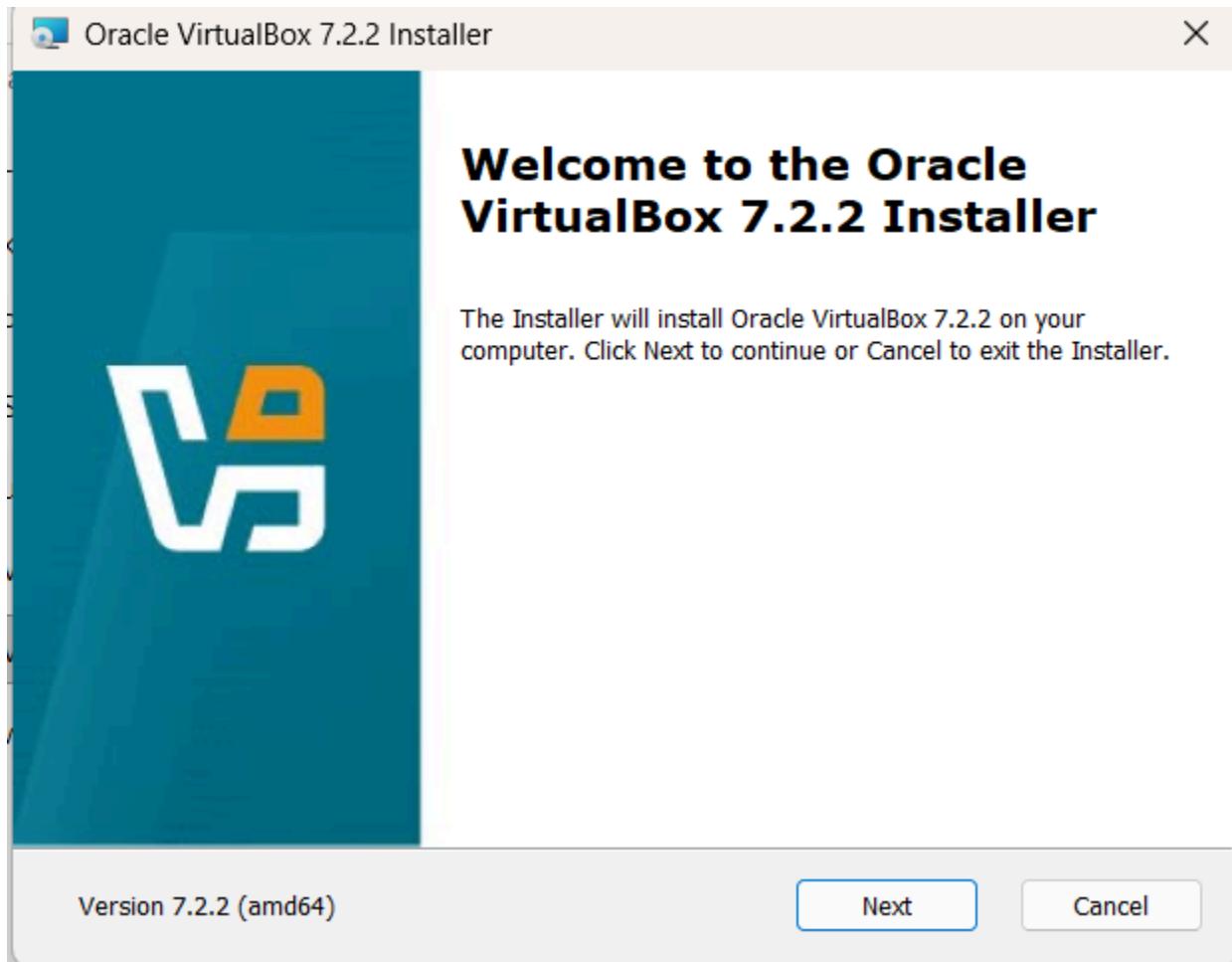
There are several virtualization platforms available, and selecting the “best” one depends on your operating system, hardware, and specific requirements. In practice, there’s no single universal solution — each platform has its own strengths and limitations.

For this project, I’m using Oracle VM VirtualBox because it’s free, open-source, and cross-platform, supporting both Windows and Linux hosts. It offers stable performance and includes essential features such as network bridging, snapshots, and multi-VM management, making it ideal for building virtual lab environments.

However, you’re not limited to VirtualBox. You can use alternative platforms like VMware Workstation Player, Microsoft Hyper-V, or Proxmox VE. The setup and configuration steps are generally similar across these tools, so choose the one that best aligns with your workflow and system capabilities.

Downloading and Installing VirtualBox

Download Oracle VM VirtualBox by [clicking here](#). Once the download is completed, I launched the VirtualBox 7.2.2



Installation

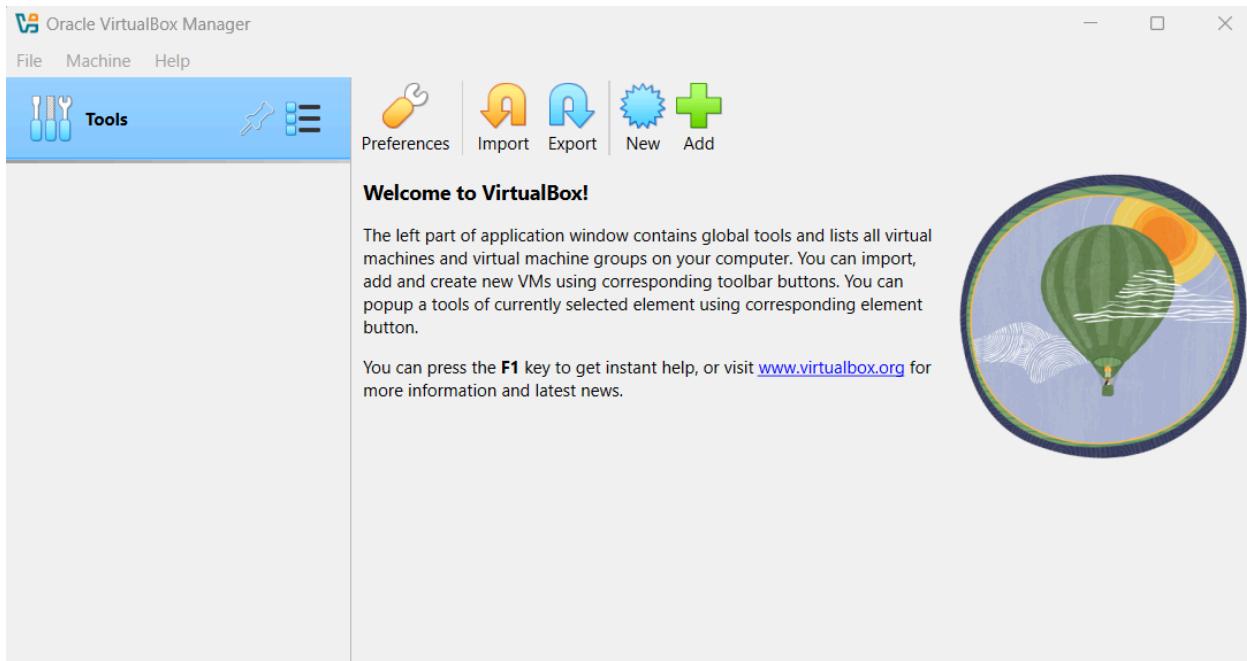
Proceed with the installation using the **default settings**.

When prompted with any confirmation dialogs (e.g., *Yes/No*), simply select "**Yes**" to continue.



Check “Start Oracle VirtualBox 7.2.2” for the VirtualBox to launch and Click “Finish”.

Upon launching we will have the VirtualBox Main interface/VirtualBox Manager.

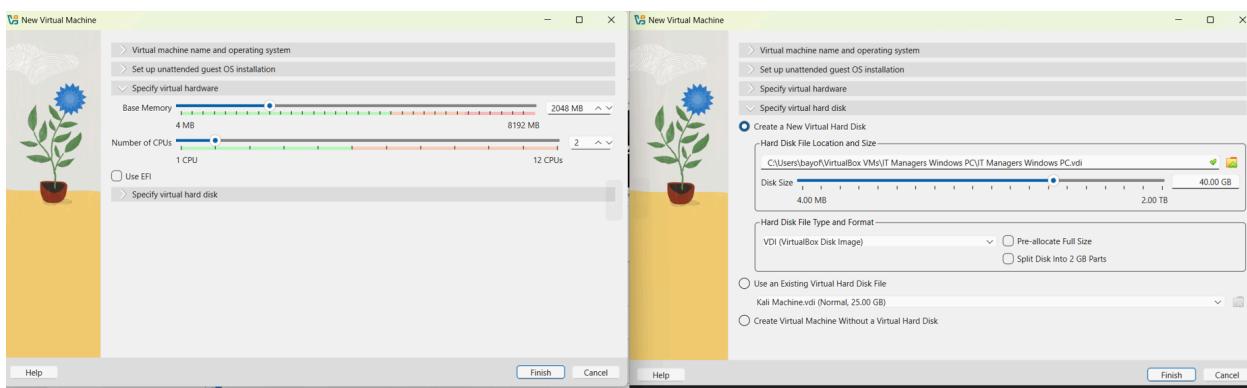
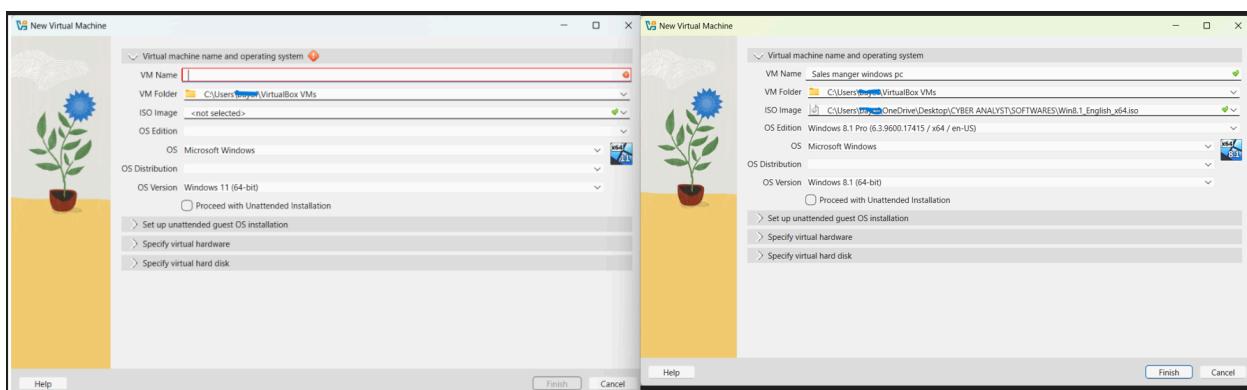
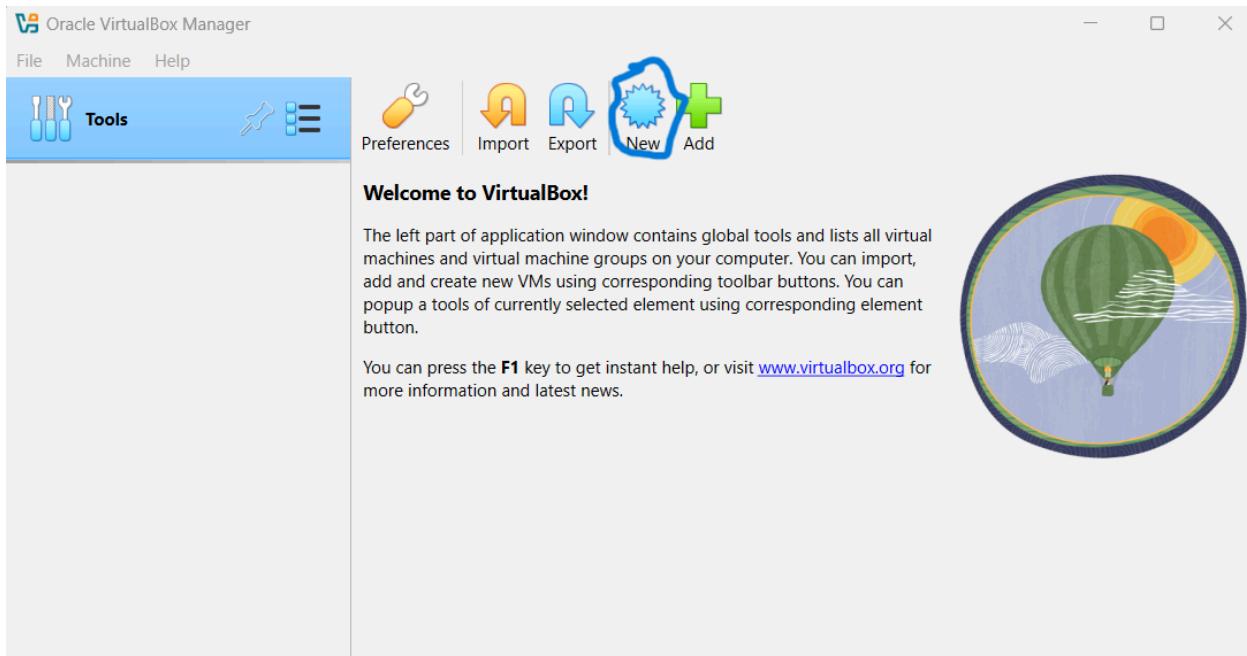


Creating a Virtual Machine

Now that we've decided on the type of virtual network to use, it's time to **create a Virtual Machine (VM)**.

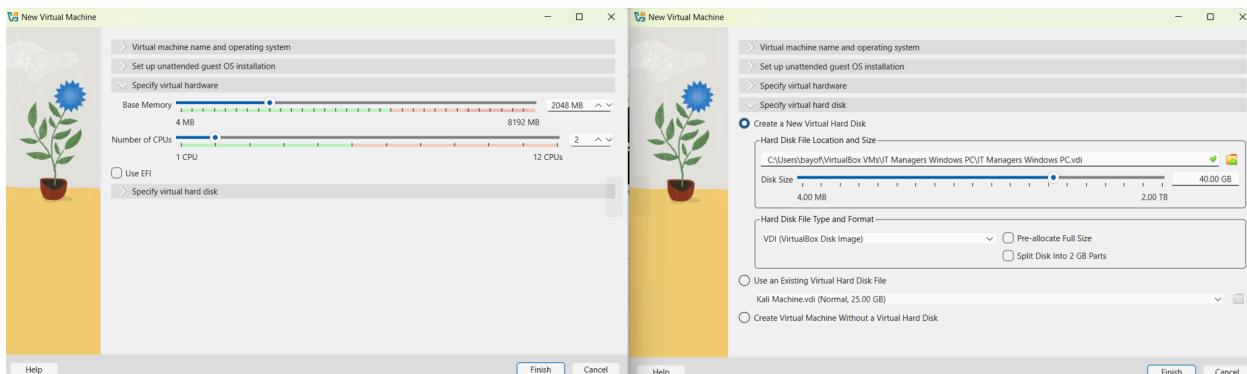
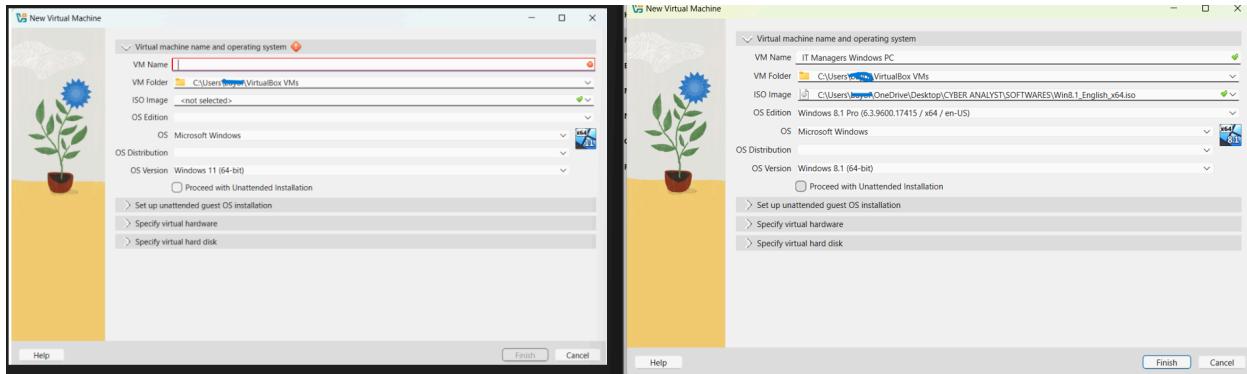
Step 1→Create a New Virtual Machine

1. Open **VirtualBox Manager** and click **New**.
2. Enter a **name** for the virtual machine, for example, "**Sales Manager Windows PC**" to make it easy to identify later.
3. Under **ISO Image**, select the **downloaded Windows ISO file** to mount it.
4. Click **Specify Virtual Hardware**, then allocate **2 GB of base memory (RAM)** and **2 CPUs** for improved performance.
5. Click **Specify Virtual Hard Disk**, leaving the default size at **40 GB**.
6. Click **Finish** to complete the VM creation.
7. The new **Sales Manager Windows PC** VM will now appear in the **VirtualBox Manager**. Start it and follow the **Windows installation process** as prompted.



Step 2→Create a New Virtual Machine

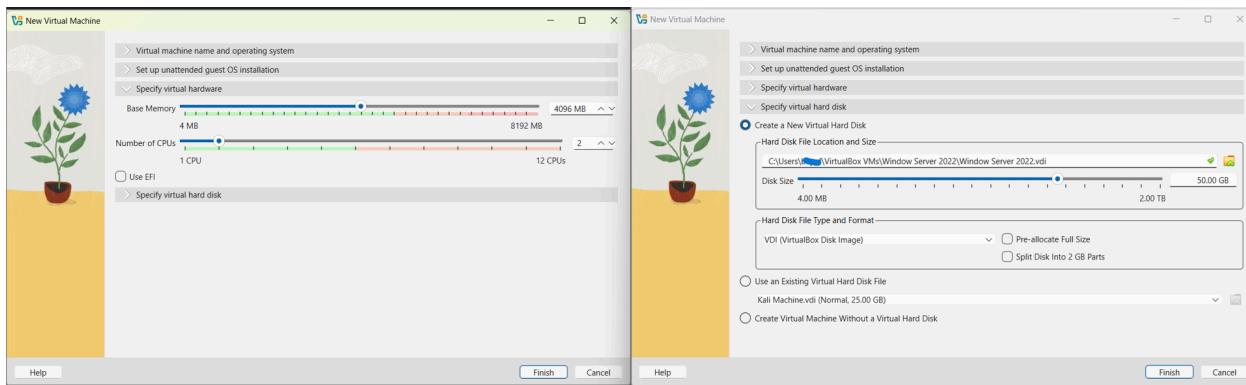
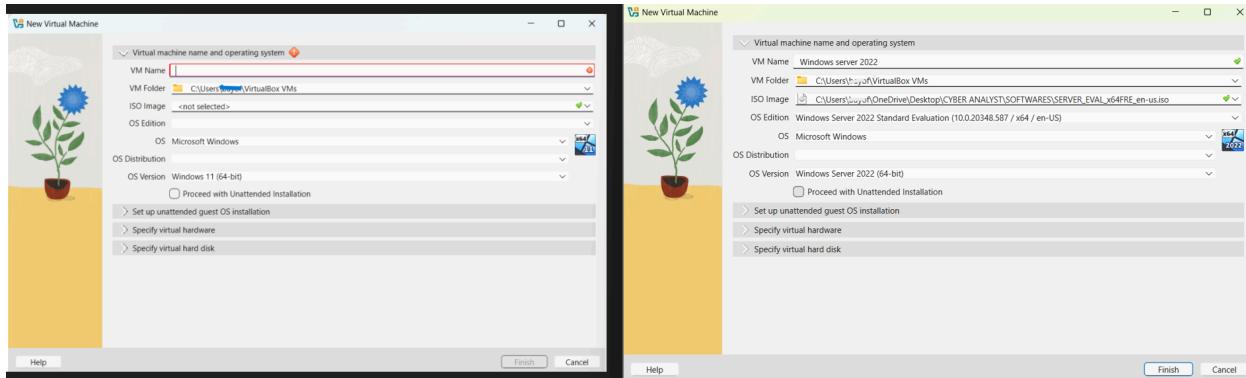
1. Open **VirtualBox Manager** and click **New**.
2. Enter a **name** for the virtual machine, for example, “**IT Manager window pc**” to make it easy to identify later.
3. Follow Step 1 Numbers 3-7



Step 3→Create Window server 2022 Virtual Machine

1. Open **VirtualBox Manager** and click **New**.
2. Enter a **name** for the virtual machine, for example, “**Window server 2022**” to make it easy to identify later.
3. Under **ISO Image**, select the **downloaded Windows ISO file** to mount it.

4. Click **Specify Virtual Hardware**, then allocate **4 GB of base memory (RAM)** and **2 CPUs** for improved performance.
5. Click **Specify Virtual Hard Disk**, leaving the default size at **50 GB**.
6. Click **Finish** to complete the VM creation.
7. The new **Sales Manager Windows PC** VM will now appear in the **VirtualBox Manager**. Start it and follow the **Windows installation process** as prompted



All three virtual Machines are showing on the VirtualBox.

The screenshot shows the Oracle VirtualBox Manager interface. On the left is a sidebar with icons for Home, Machines, Extensions, Media, Network, Cloud, and Resources. The 'Machines' icon is selected. The main area displays three virtual machines:

- Sales manager win...**: x64, Version 8.1, Powered Off
- IT Managers Win...**: x64, Version 8.1, Powered Off
- Window Server 2022**: x64, Version 2022, Running

Details for the selected 'Window Server 2022' machine are shown in the right panel:

- General**
 - Name: Window Server 2022
 - Operating System: Windows Server 2022 (64-bit)
- System**
 - Base Memory: 2048 MB
 - Processors: 2
 - Boot Order: Floppy, Optical, Hard Disk
 - Acceleration: Nested Paging, Hyper-V Paravirtualization
- Display**
 - Video Memory: 128 MB
 - Graphics Controller: VBoxSVGA
 - Remote Desktop Server: Disabled
 - Recording: Disabled
- Storage**
 - Controller: SATA
 - SATA Port 0: Window Server 2022.vdi (Normal, 50.00 GB)
 - SATA Port 1: [Optical Drive] SERVER_EVAL_x64FRE_en-us.iso (4.00 GB)
- Audio**
 - Host Driver: Default
 - Controller: Intel HD Audio
- Network**
 - Adapter 1: Intel PRO/1000 MT Desktop (NAT Network, 'CYBERTEC')
- USB**
 - USB Controller: xHCI
 - Device Filters: 0 (0 active)
- Shared folders**
 - None
- Description**
 - None

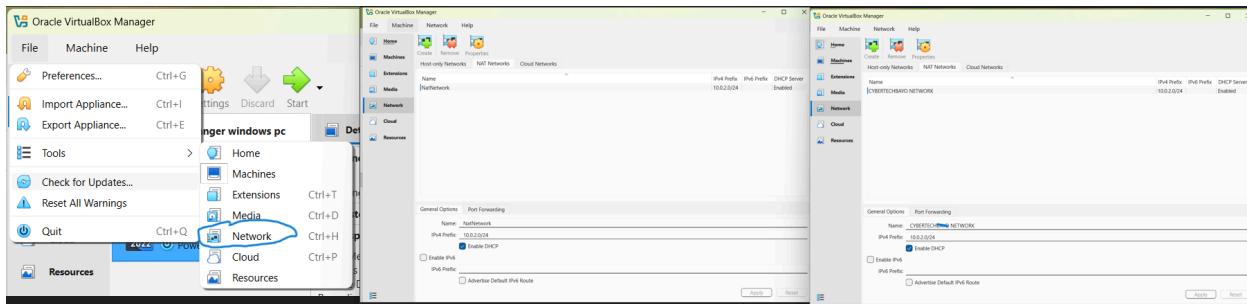
Creating a Virtual Network with VirtualBox

After creation, you'll see “**Windows Server 2022**,” “**IT Manager Windows 8**,” and “**Sales Manager Windows PC**” listed in the **VirtualBox Manager**, as shown above.

Once all three VMs are set up, navigate to the **VirtualBox Network Settings** to create a shared network for them:

1. Click **File** → **Tools** → **Network**.
2. Under **NAT Networks**, click **Create**.
3. A new network (named **NatNetwork** by default) will appear. Click on it to rename it to an organizational name of your choice in this example, “**CYBERTECHxxx NETWORK**.”
4. Check the box for **Enable DHCP**, then click **Apply** to save your configuration.

This creates a dedicated **NAT Network** that all three VMs can use to communicate internally while maintaining secure internet access through the host machine.



After completing the setup, you'll return to the **VirtualBox Manager**, where all **three virtual machines (VMs)** will be listed.

Next, we need to connect all three VMs to the same virtual network “**CYBERTECHXXXX NETWORK**” which we created earlier.

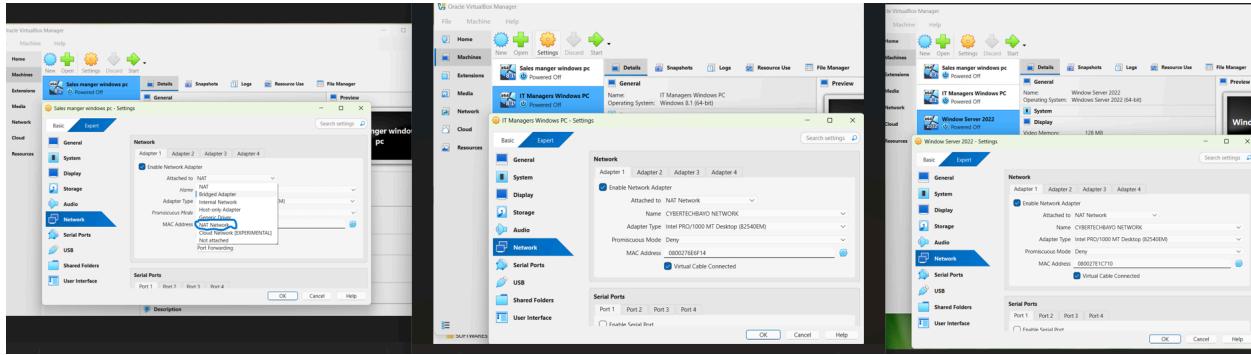
To configure each VM:

1. Select the VM (starting with “**Sales Manager Windows PC**”) and click **Settings**.
2. Navigate to the **Network** tab.
3. Under **Attached to**, select **NAT Network** from the dropdown menu.

4. Choose “**CYBERTECHXXXX NETWORK**” as the **Network Name**.

5. Click **OK** to save the changes.

Repeat these steps for all three VMs to ensure they are connected to the same organizational network.



Configuring Base Memory (RAM Allocation)

Base memory, commonly referred to as **RAM**, is one of the most important performance factors when running multiple virtual machines (VMs).

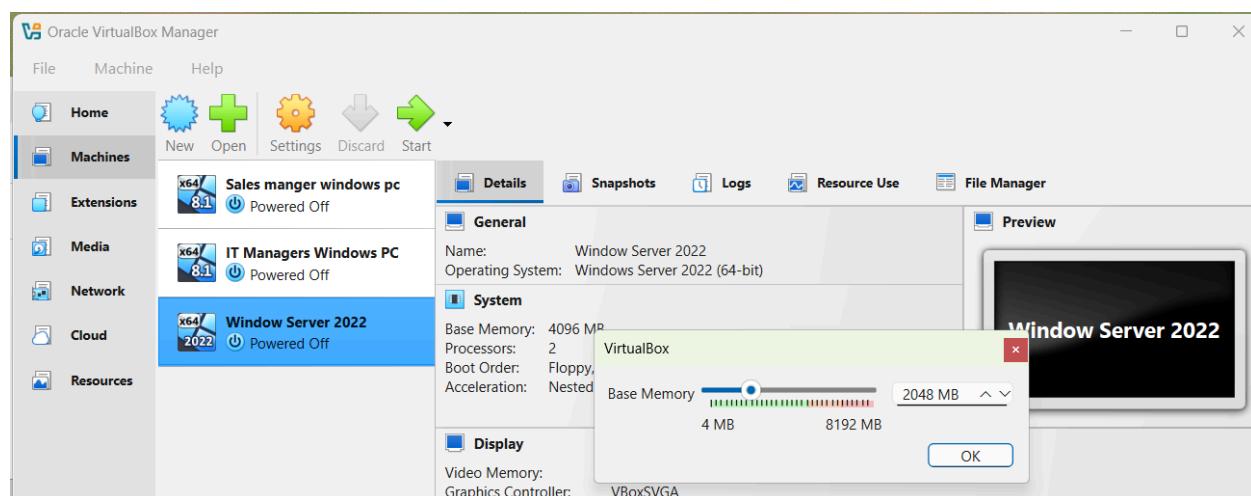
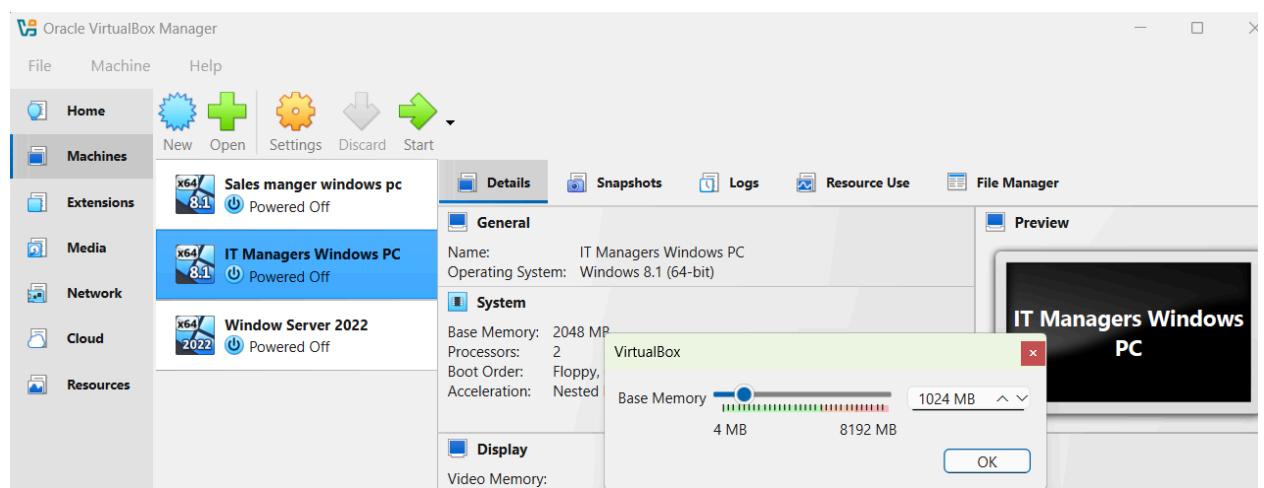
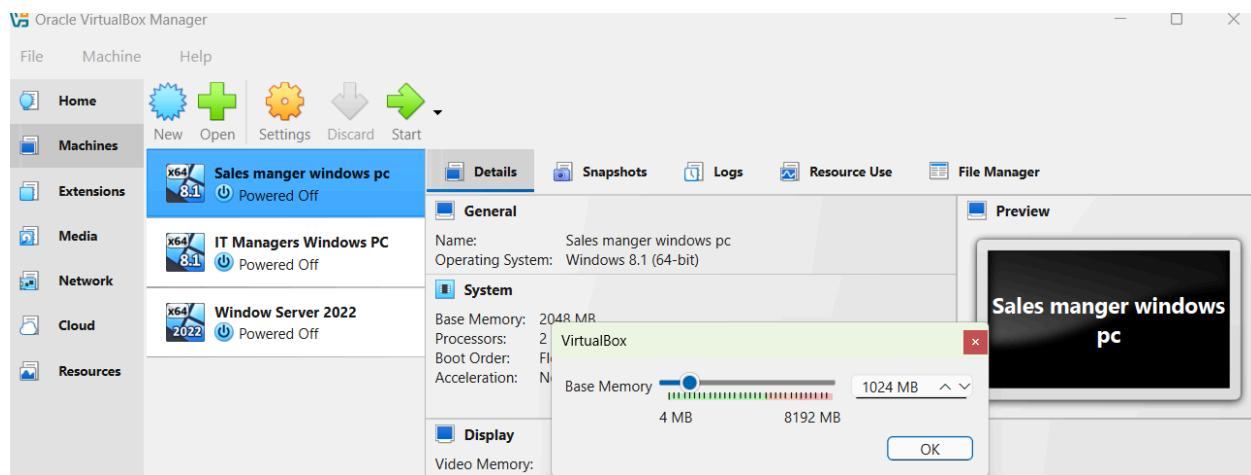
The host system in this setup has **8GB of RAM**. As a general best practice, it's recommended to allocate **no more than 50% of total system memory** across all active VMs to ensure both the host and guest systems run efficiently.

Initially, the three VMs were assigned **2GB**, **2GB**, and **4GB** of RAM respectively, which collectively consumed the entire 8GB leaving no memory available for the host system.

To optimize performance, I adjusted the memory allocation as follows:

- **Windows Server 2022:** 2GB
- **IT Manager Windows PC:** 1GB
- **Sales Manager Windows PC:** 1GB

This configuration balances memory usage, allowing all VMs to operate smoothly while maintaining sufficient resources for the host operating system with 4GB.



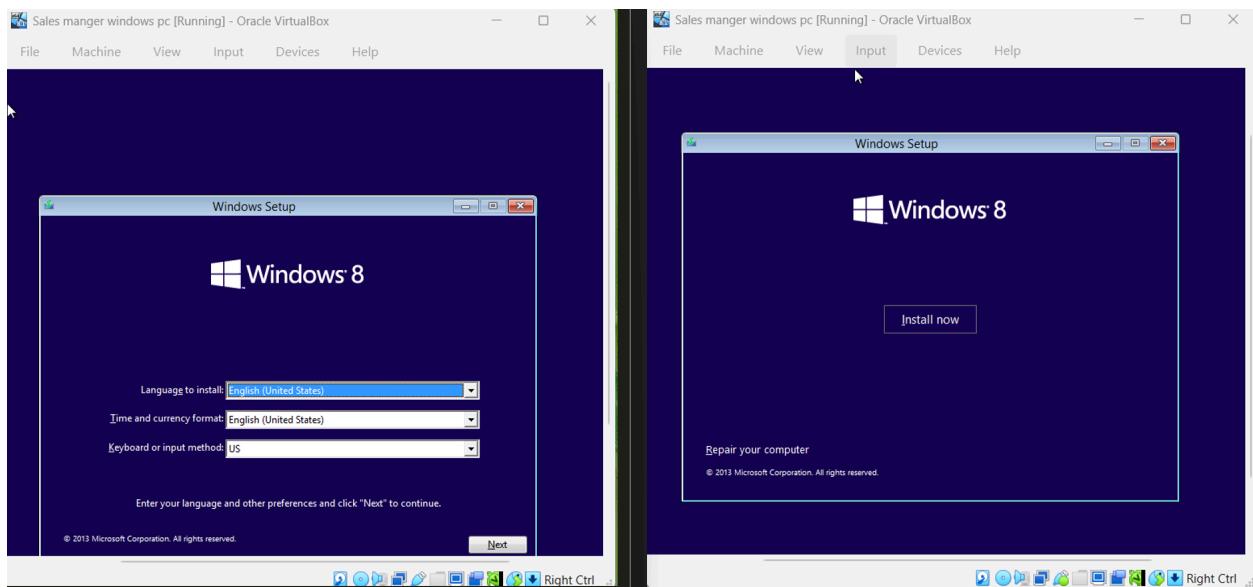
Installing an Operating System on Your Lab VMs

After creating your VM and assigning virtual hardware (RAM, CPU, and disk), Our next step is to install the operating system (OS) using an ISO image.

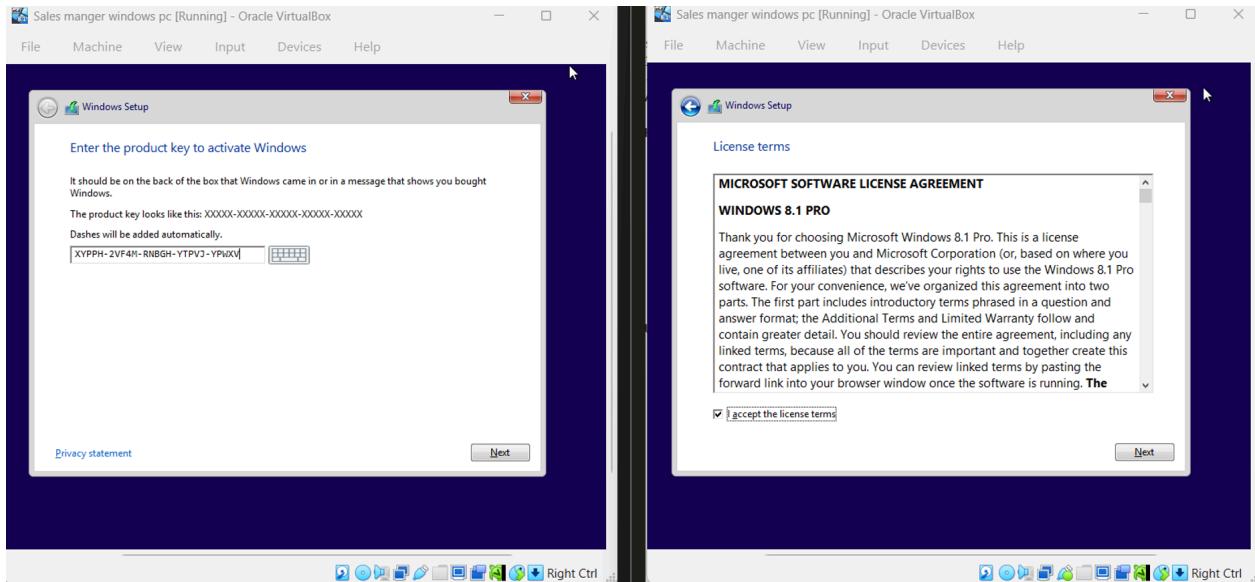
Step 1→Start the VM (I will be starting with ‘Sales Manager windows pc’)

- Open VirtualBox Manager.
- Select the VM you created (Sales Manager Windows PC).
- Click the green arrow Start (The system will boot and automatically detect the attached ISO image).

Note:- If it doesn't, you can manually mount the ISO by going to Devices → Optical Drives → Choose a disk file, and then select your OS ISO file.



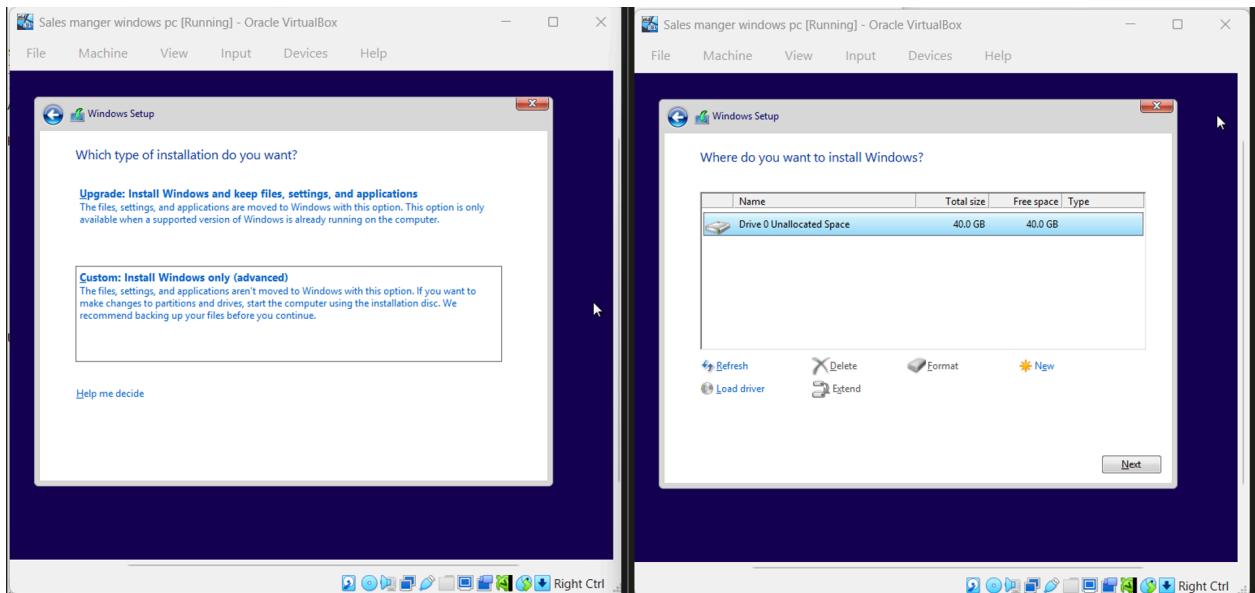
Step 2→Will be prompted to input the “Product Key to activate window” Insect and Click ‘Next’ and Accept the license term and click ‘Next’.



Step 3→ Select Installation Type

- Choose **Custom: Install Windows only (advanced)** to perform a clean installation.
- Select the virtual hard disk (the 40 GB disk you created earlier) → Click **Next**.

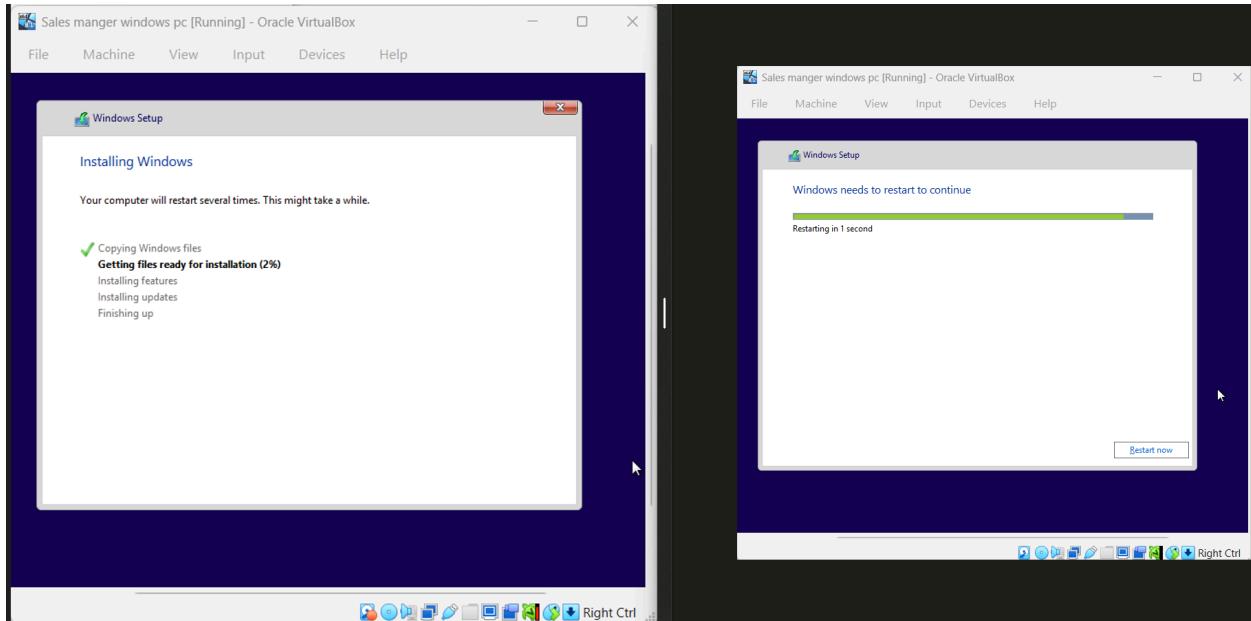
The installer will automatically partition and format the disk before continuing.



Step 3→ Installation Process

- The system will begin copying and installing files.
- This process may take **8 - 15 minutes**, depending on the host system's performance.

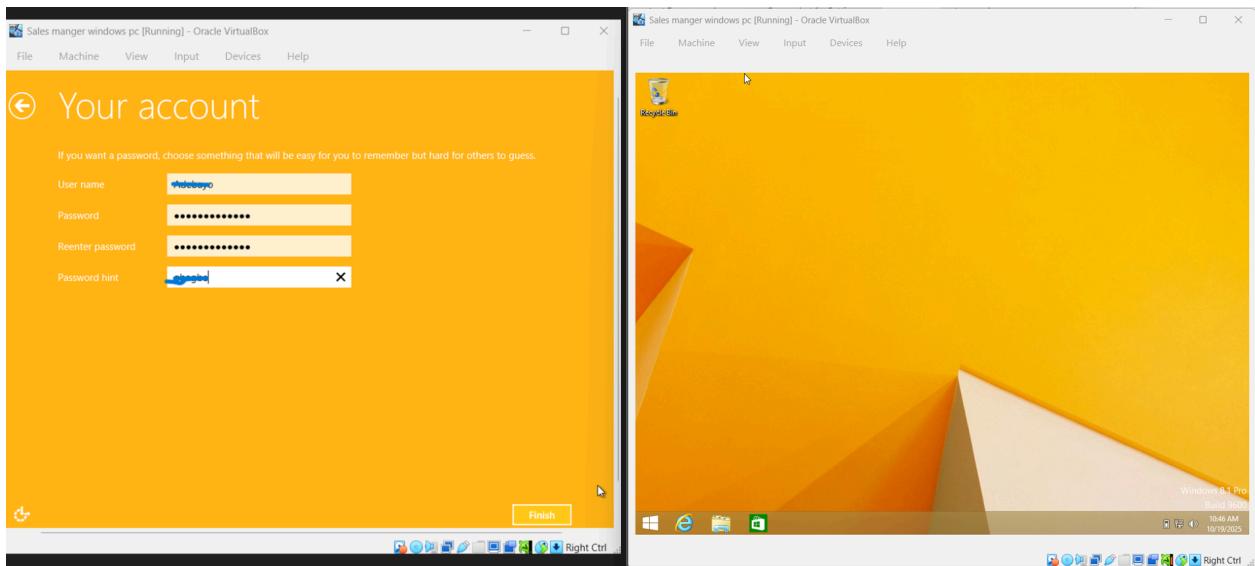
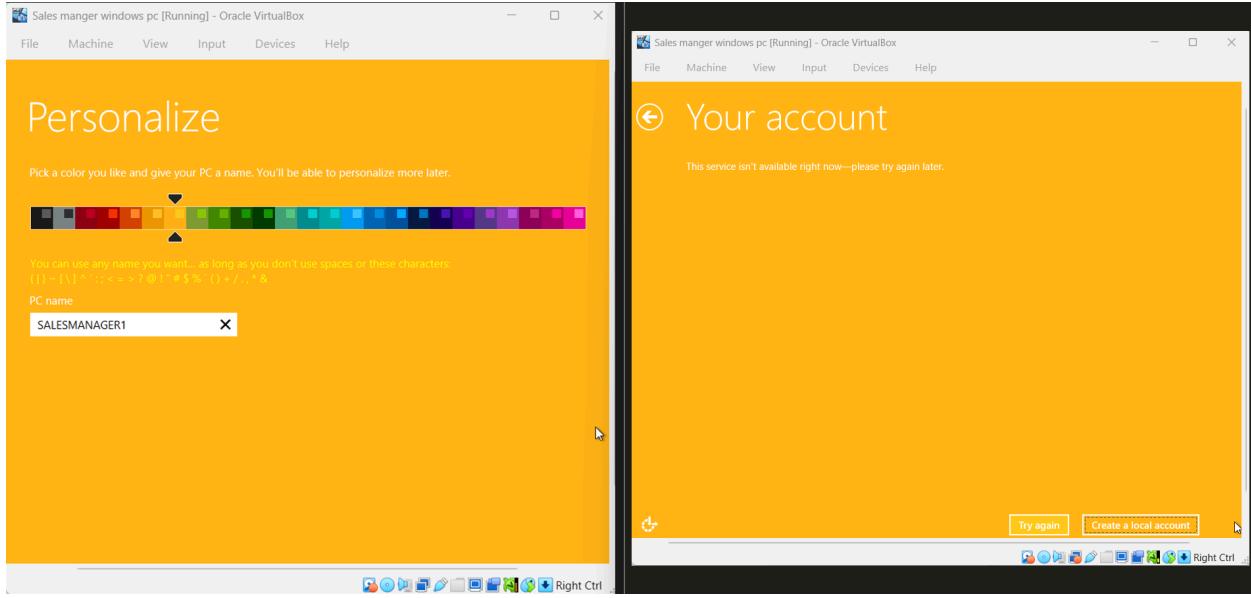
The VM will reboot automatically several times during the installation.



Step 4→ Initial Setup

After installation completes:

- Give your PC a name (SALESMANAGER1), Personalize the PC with your color choice for easy differentiation of vm (optional)
- Set your **administrator username** and **password**.
- Choose the **network location** (typically “Private Network” for a lab setup).
- Log in to the desktop for the first time.



Conclusion

Building a virtual IT and cybersecurity lab is one of the most effective ways to gain real-world, hands-on experience without the need for expensive hardware.

Through this project, I designed and deployed a fully virtualized enterprise environment using Oracle VM VirtualBox and Windows-based systems. The setup simulates a small to medium

business infrastructure, complete with a Windows Server, user workstations, and a shared virtual network all running securely within a single host system.

This lab environment allows for:

- Conducting **cybersecurity simulations, incident response, and ethical hacking exercises**
- Testing **Active Directory, DNS, and DHCP** deployments
- Exploring **endpoint management, patching, and policy enforcement**
- Practicing **system administration and network configuration**

By leveraging free and open-source virtualization tools, I was able to replicate enterprise-grade infrastructure in a home environment, enabling continuous learning, experimentation, and technical growth.