

## Introduction: Setting up a Virtual Home Lab

I am creating a virtual lab project with detailed, step-by-step documentation to practice networking, server management, and cybersecurity. The lab is divided into two subnets: **Operations** and **IT**.

In the first phase, I am setting up the **Operations subnet**, which includes a **Windows Server** and two Windows 8 PCs, labeled **Sales Manager PC** and **IT Manager PC**. The **IT subnet** will be implemented in the next phase and will feature systems such as **Ubuntu**, **Parrot OS**, **Kali Linux**, and **PfSense**, providing a flexible environment for experimenting with different operating systems and security tools.

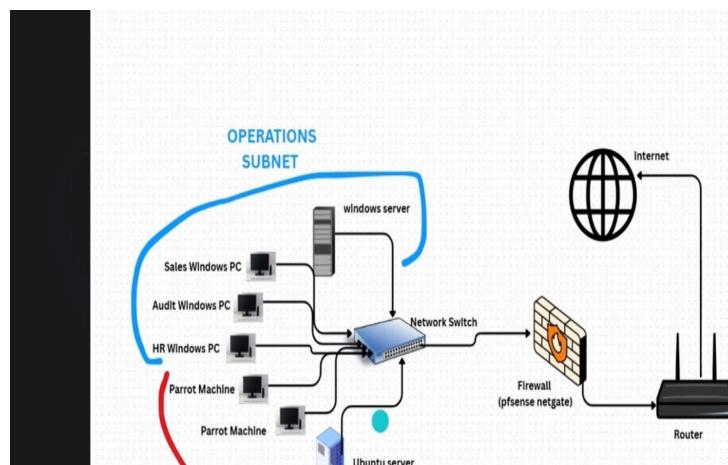
This setup allows me to explore **network communication, device management, and cybersecurity practices** in a controlled and practical environment.

### Lab Components:

- Virtualization Manager: Virtual box
- Windows Server ( Domain Controller): Handles Active Directory, DNS, DHCP
- Ubuntu Server: Hosts web apps, databases, or containers (e.g Docker).
- FireWall( Pfsense Netgate): Stimulates internal traffic and provides NAT.
- Windows 8 for: Sales manager and IT Manager PCs
- Kali Linux
- Router: Connect LAN to the internet.

This virtual lab simulates a small organizational network with two subnets: **Operations** and **IT**.

- **Operations Subnet:** Includes a **Windows Server** (domain controller) and two Windows 8 PCs (**Sales Manager PC** and **IT Manager PC**) for daily operations and IT management tasks.
- **IT Subnet:** These include **Ubuntu**, **Parrot OS**, **Kali Linux**, and **PfSense** for cybersecurity testing and experimentation.



## What is Virtualization ?

**Virtualization** is a technology that allows a single physical computer to operate multiple independent environments or resource instances simultaneously. It enhances system efficiency, scalability, and flexibility by enabling several operating systems and applications to run separately on the same hardware platform.

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!



## Core Concepts of Virtualization

### ➤ Virtual Machines [VMs]

A virtual machine is a software-created computer that mimics the functions of a physical one. It runs its own operating system and applications independently, just like a real physical system.

### ➤ Hypervisors

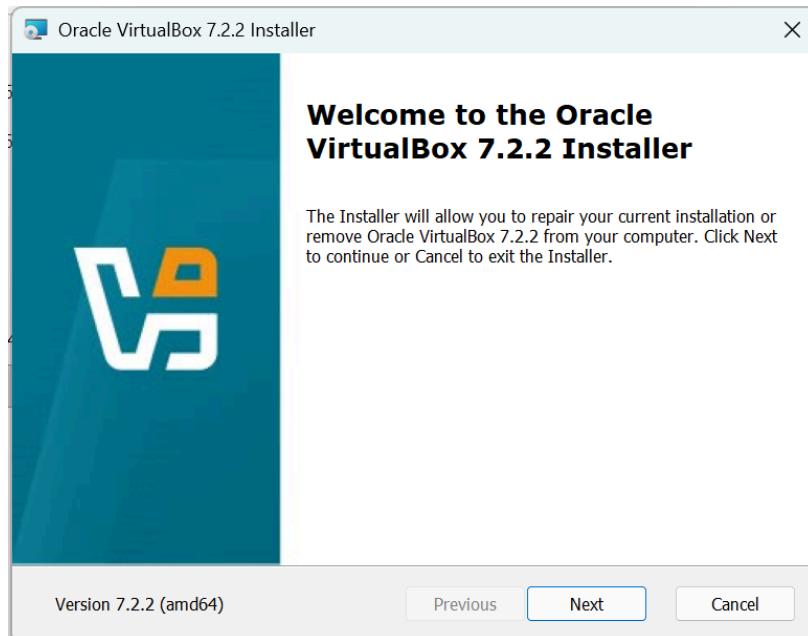
A hypervisor is the software that makes virtualization work. It helps share the system's hardware between different virtual machines. There are two main kinds of hypervisors:

- **Type 1 (Bare-metal Hypervisors):** Installed directly on the physical hardware, offering high performance and efficiency. Examples: VMware ESXi, Microsoft Hyper-V.
- **Type 2 (Hosted Hypervisors):** Runs on top of a host operating system, suitable for desktop or development environments. Examples: Oracle VirtualBox, VMware Workstation.

For this lab, I will be using Oracle VM VirtualBox since it runs well on both Windows and Linux. You can use whatever virtualization software you like; most of the steps will be very similar.

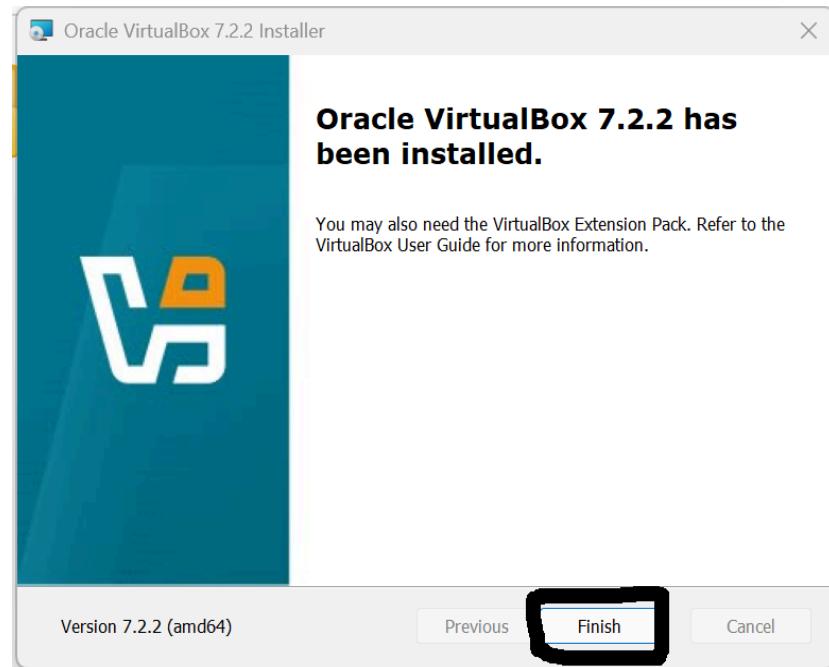
## Downloading and Installing VirtualBox

**Step 1:** I visited the official virtualbox download page and click the link [Oracle VirtualBox](#) then I downloaded Oracle virtual box and preceded with the installation



**Step 2:** I went through the installation using **all the default options**.

Whenever you're prompted with a **yes/no question**, simply choose "**Yes**" to continue



Then Click **Finish** and launch Virtual Box.

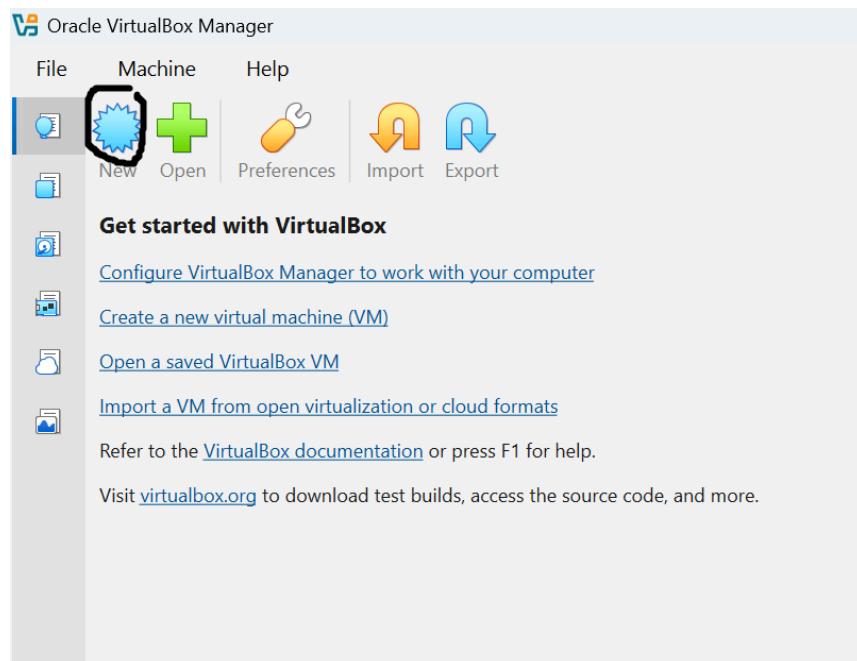
### **Creating a Virtual Machine**

When I launch VirtualBox, the first thing I see is this main window, which shows all my virtual machines and options.

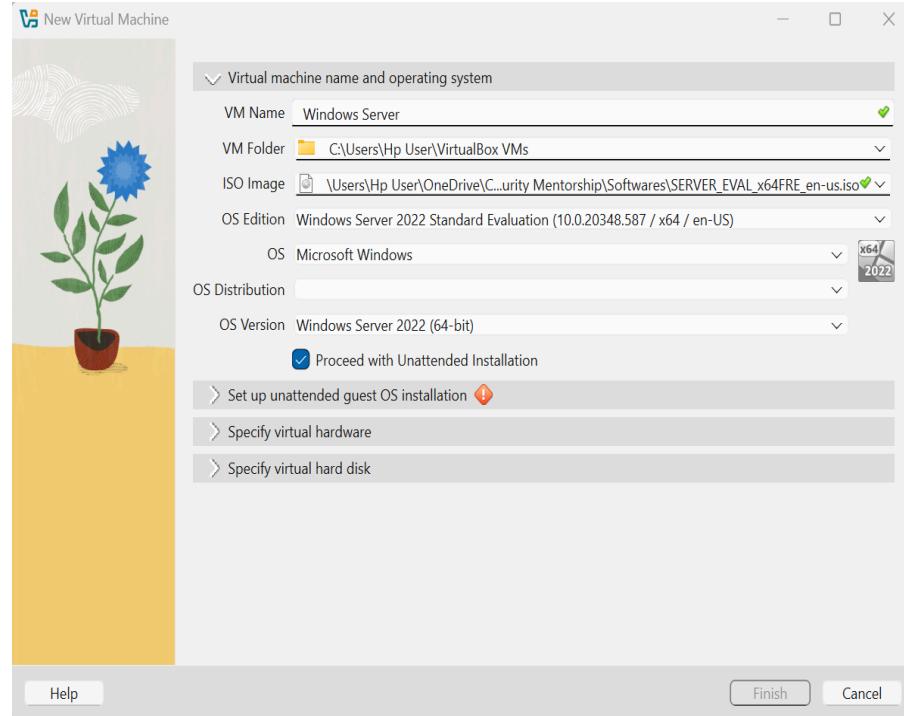


## Creating “Window Server” Virtual Machine

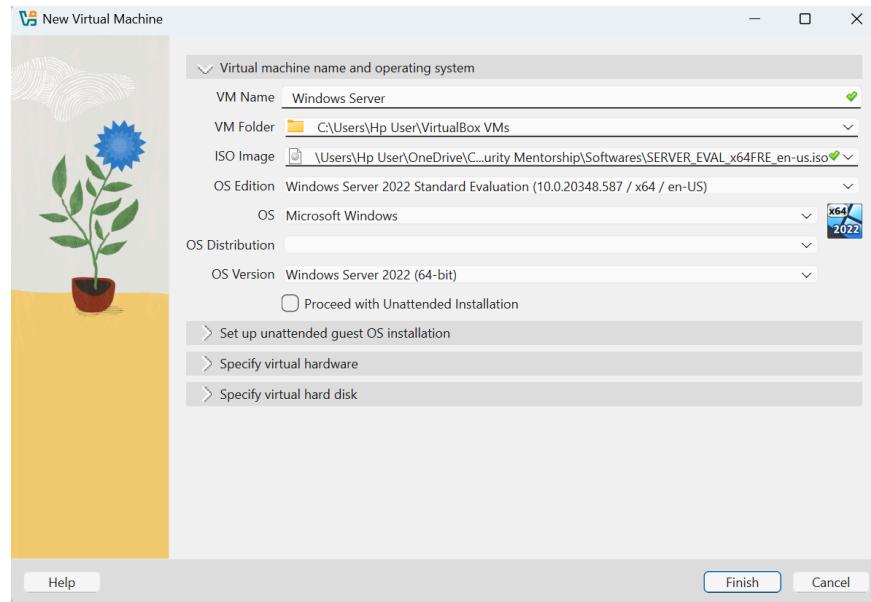
**Step 1:** To create a virtual machine for Windows Server. I clicked on “New” option



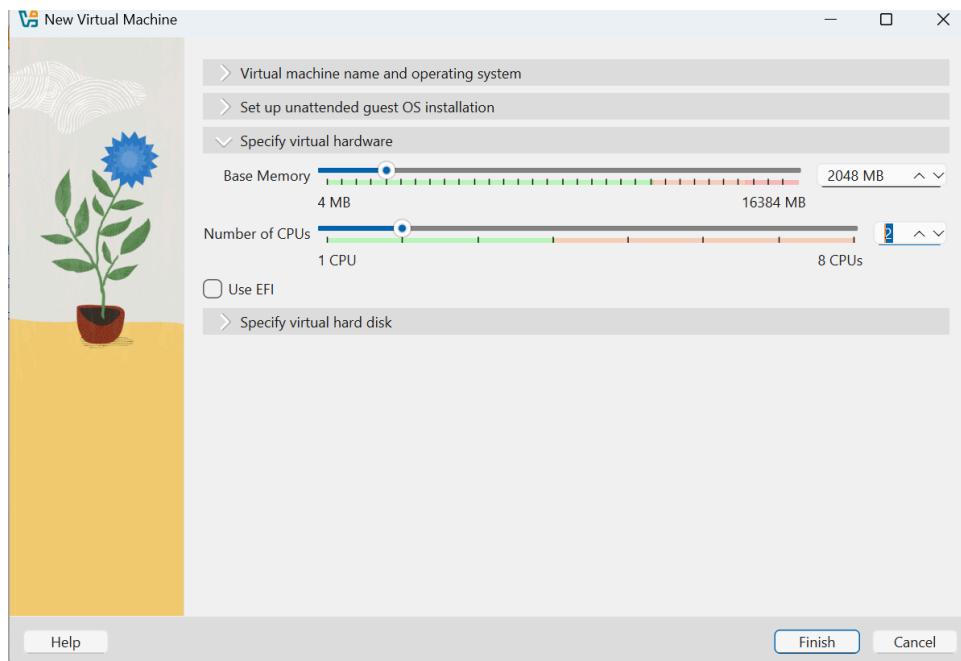
**Step 2:** I named it “**Windows Server**”, and selected the Windows Server ISO image then inserted the Window server ISO file.



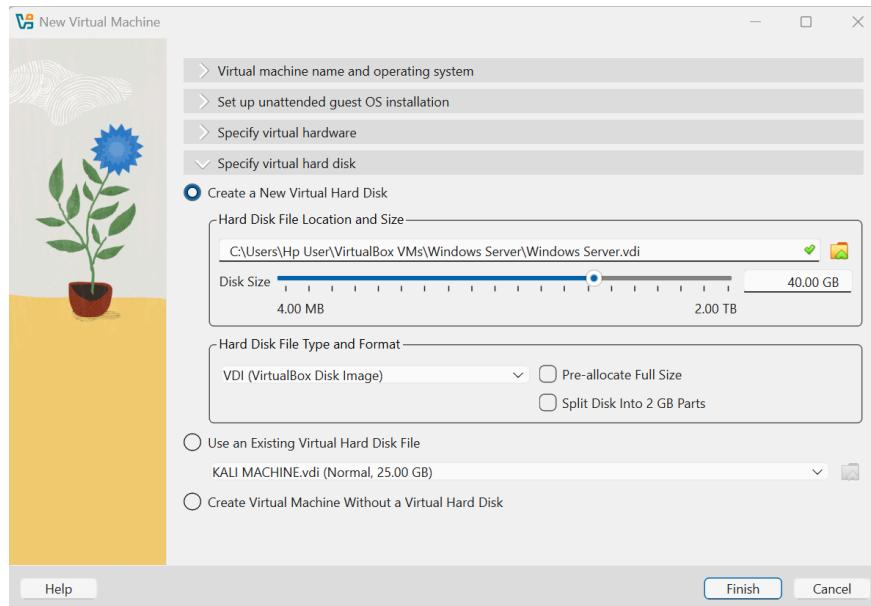
**Step 3:** I disabled the option to “proceed with **unattended installation**” before proceeding.



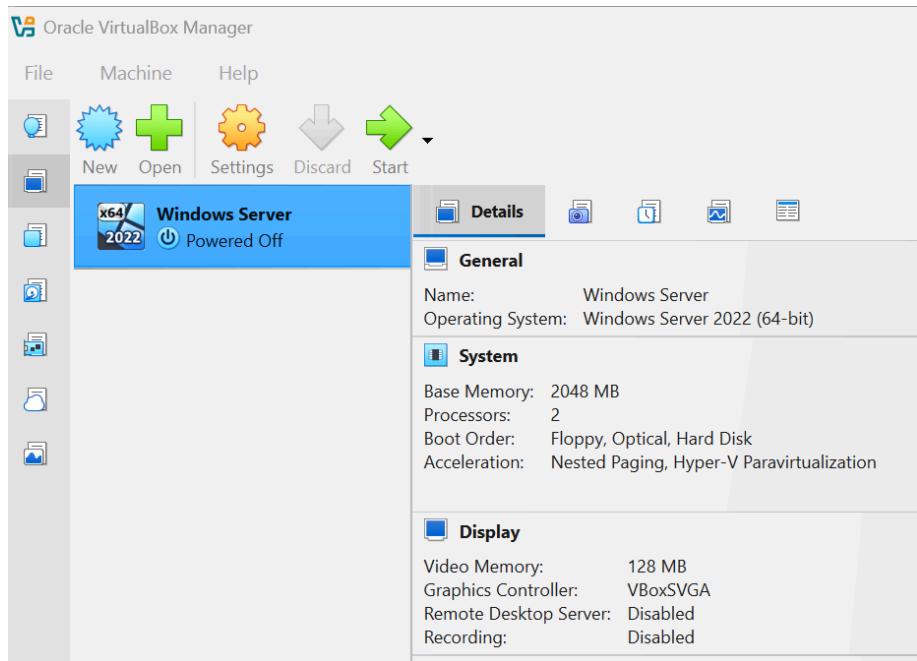
**Step 4:** I then configured the virtual hardware by default allocating **2048 MB of RAM** and setting the **CPU to 2 cores** to ensure better performance.



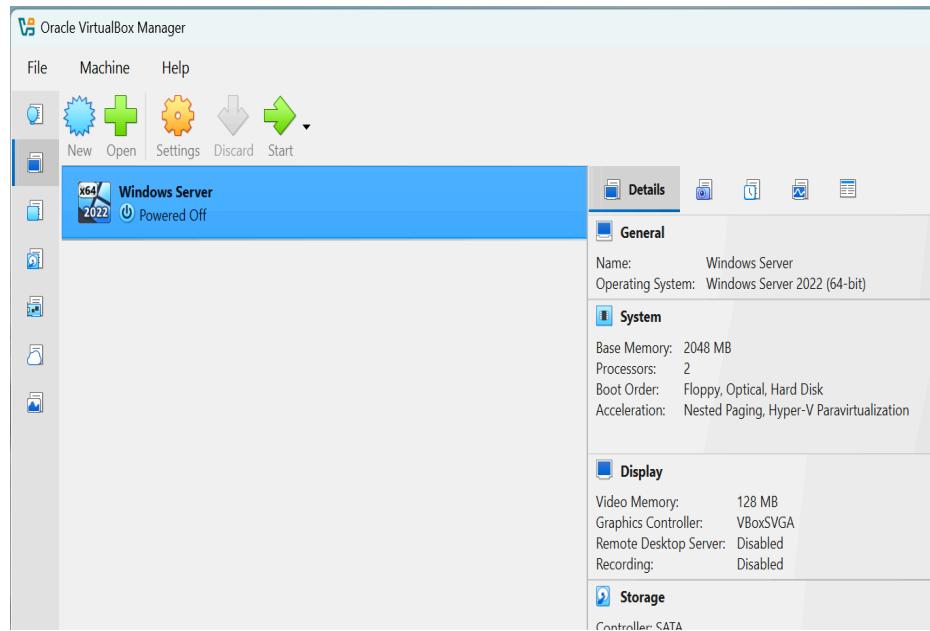
**Step 5:** Lastly, I specified the **virtual hard disk size as 40 GB** to provide sufficient storage for the operating system then select **finish**.



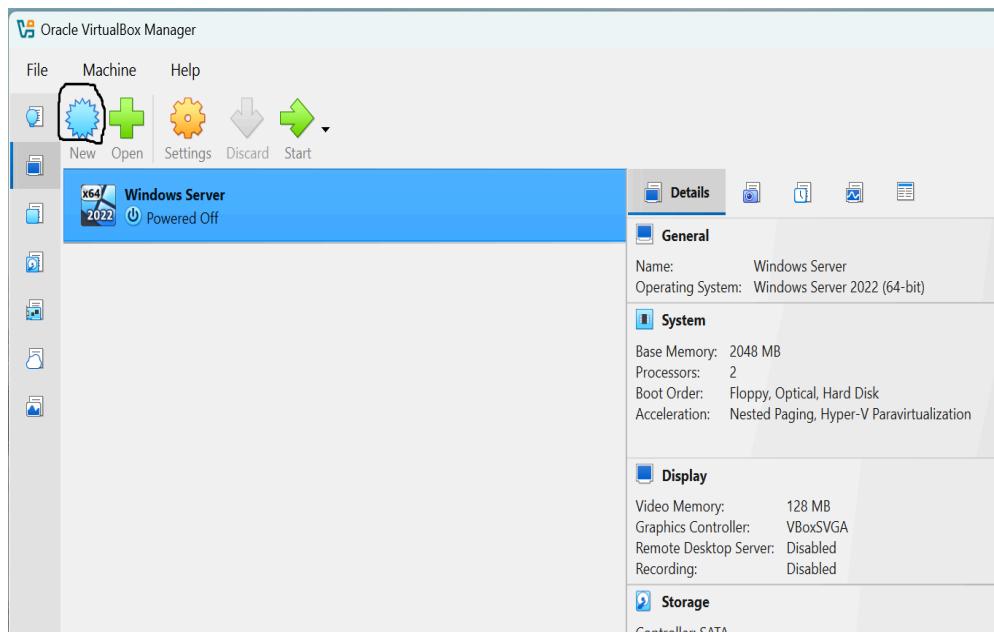
Finally, my **Windows Server VM** has been created successfully and is ready to boot up for installation.



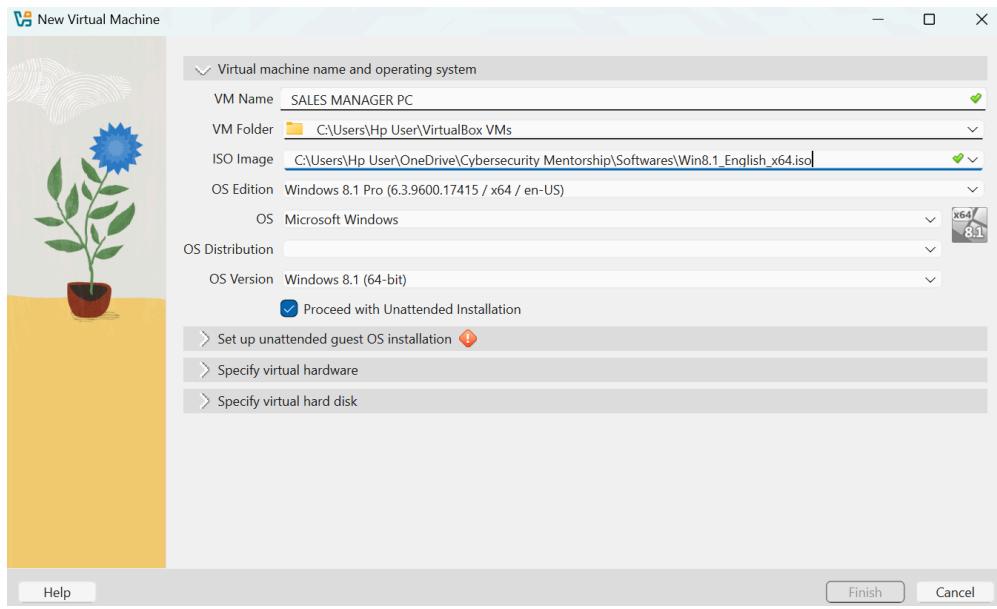
## Setting Up the First Windows 8 PC – Sales Manager PC



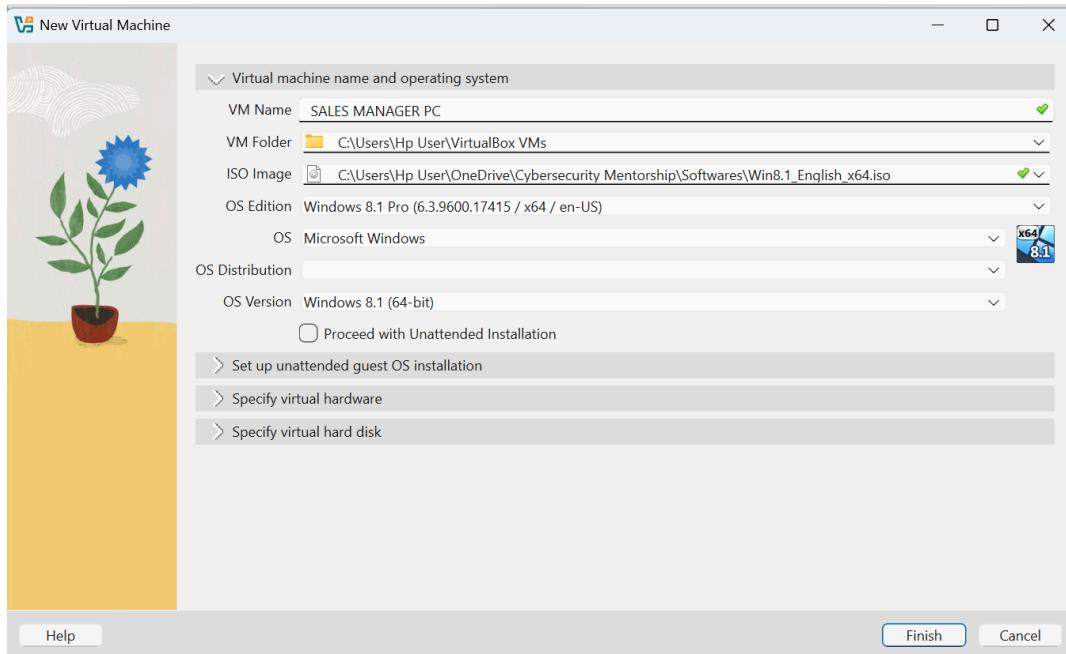
**Step 1:** I click on the “New” button at the top to start creating a new virtual machine.



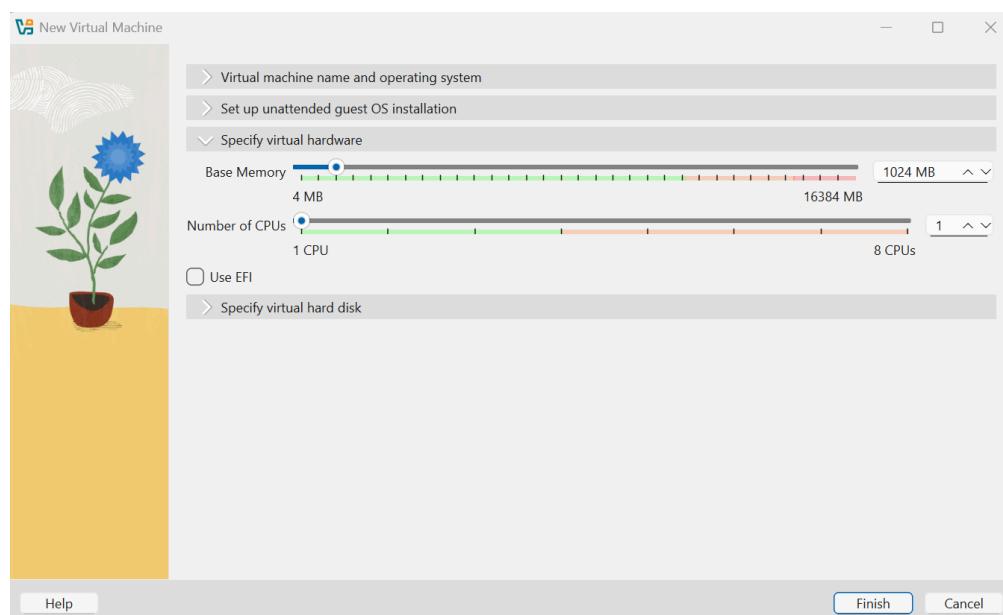
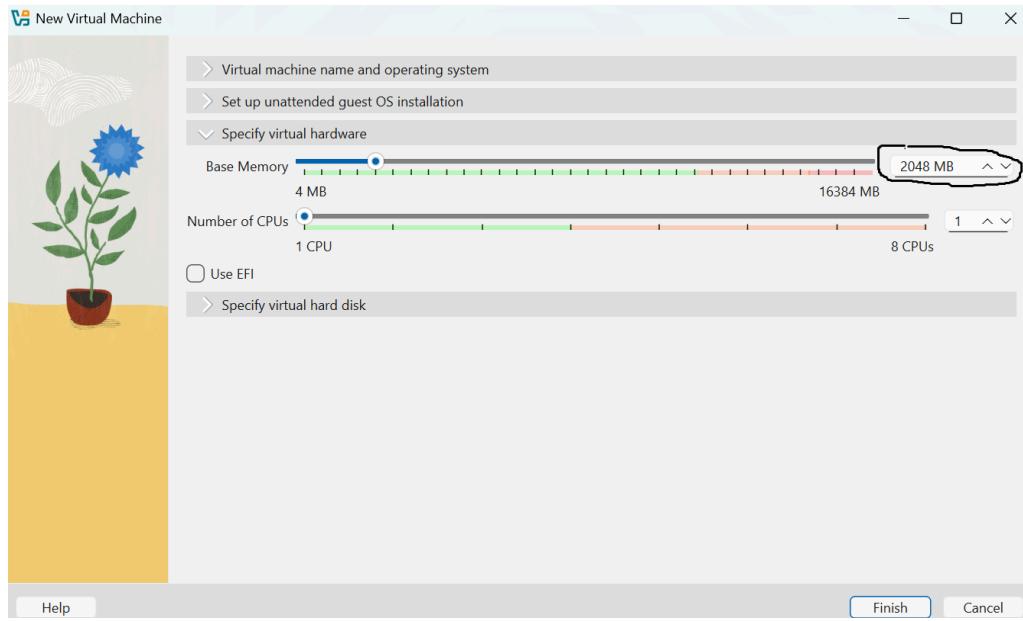
**Step 2:** In the new window, I named the virtual machine “SALES MANAGER PC” and clicked on the ISO image and selected the Windows 8 ISO file to use for the installation.



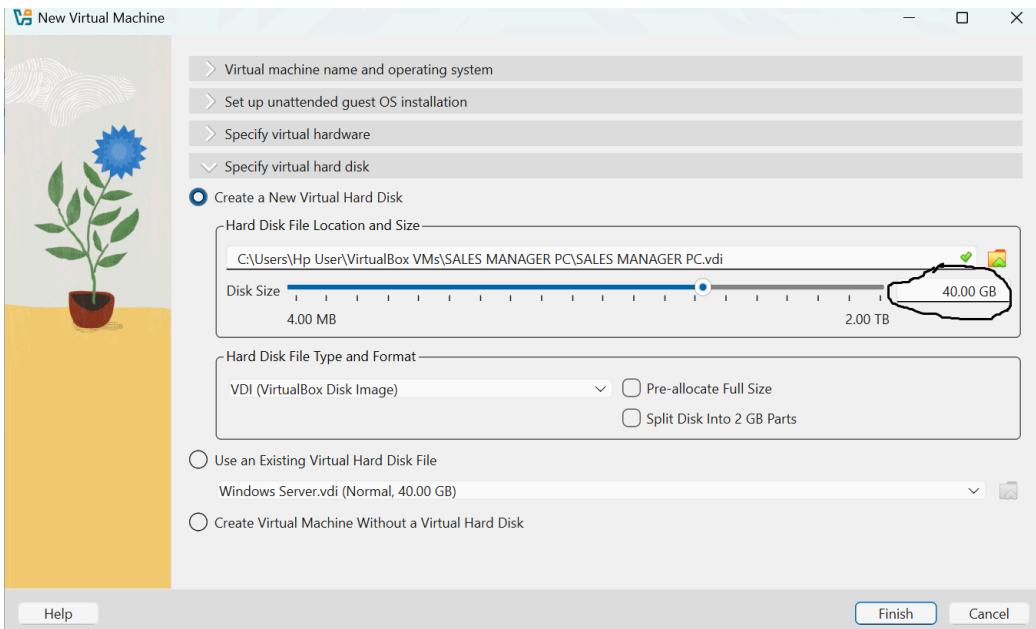
**Step 3:** I deselected the button “Proceed with Unattended Installation”



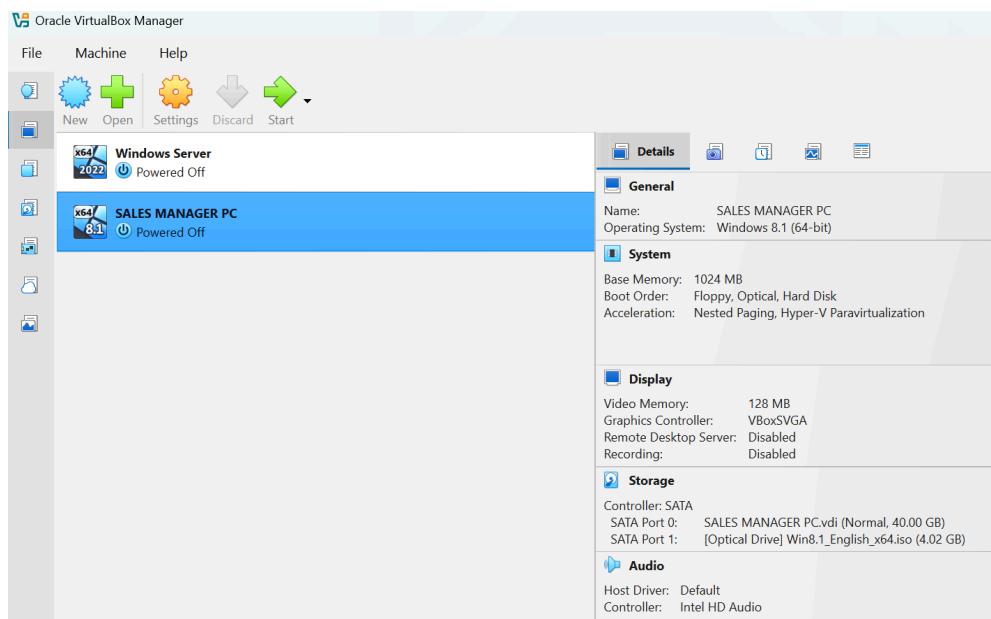
**Step 4:** I chose to lower the allocated virtual hardware memory from the default memory of 2048 to **1024 MB** to optimize resource usage and leave the number of CPU untouched.



**Step 5:** I decided to keep the virtual hard disk to 40GB as it is, but you can adjust them based on your preference

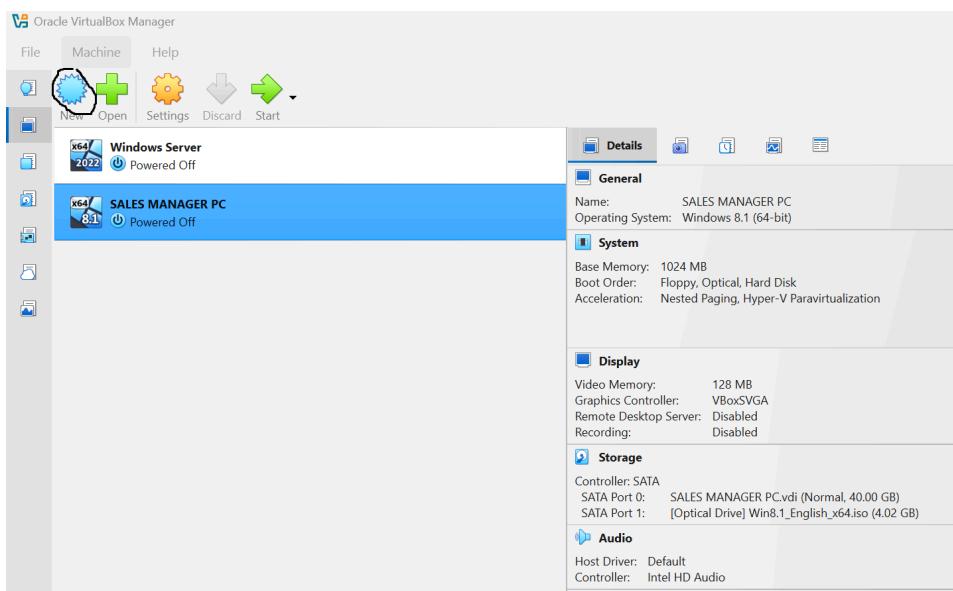


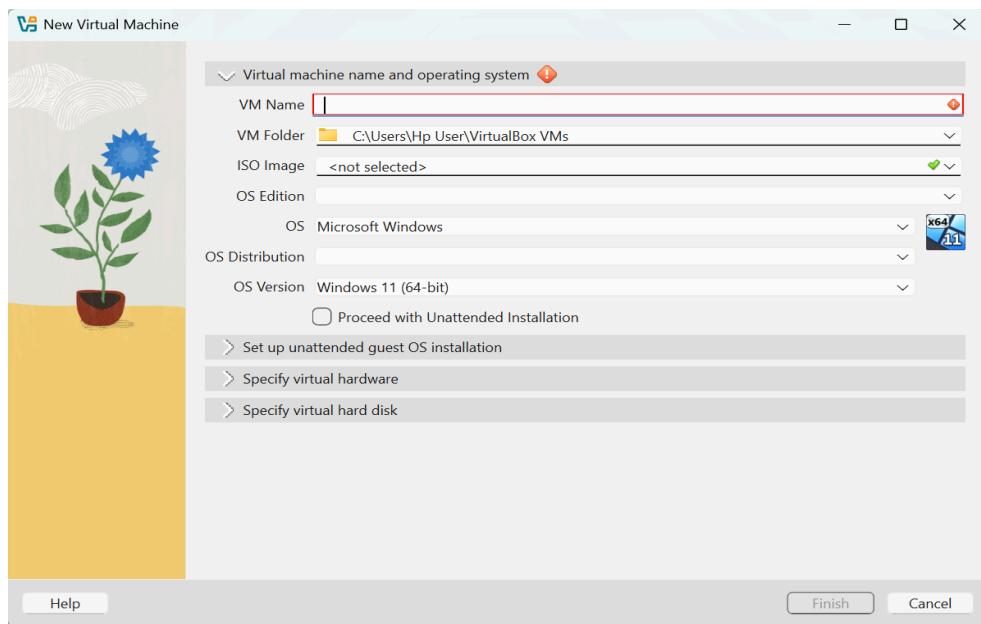
**Step 6:** Then I clicked “Finish”, and the Sales Manager PC was successfully launched.



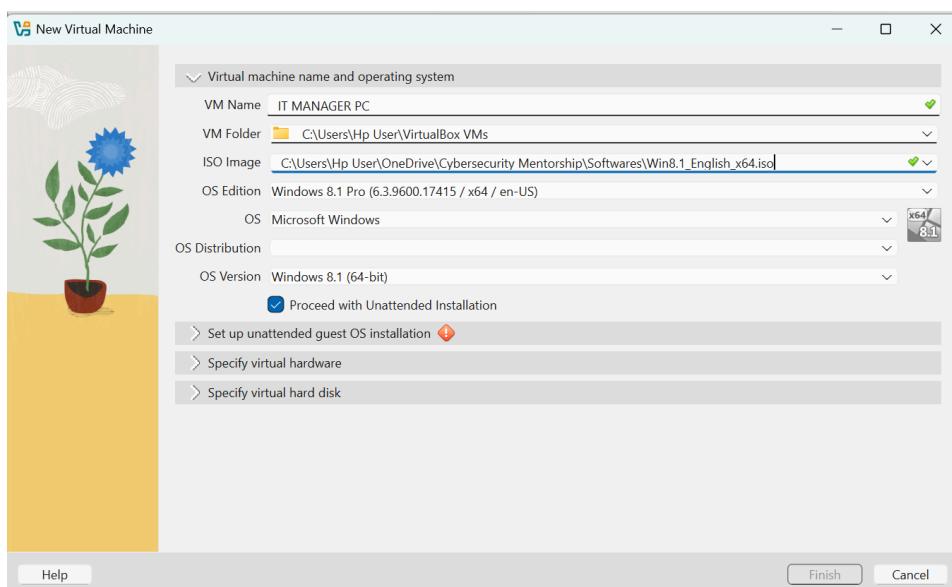
## Setting Up the Second Windows 8 PC – IT Manager PC

**Step 1:** I clicked the “New” button at the top to begin creating another virtual machine.

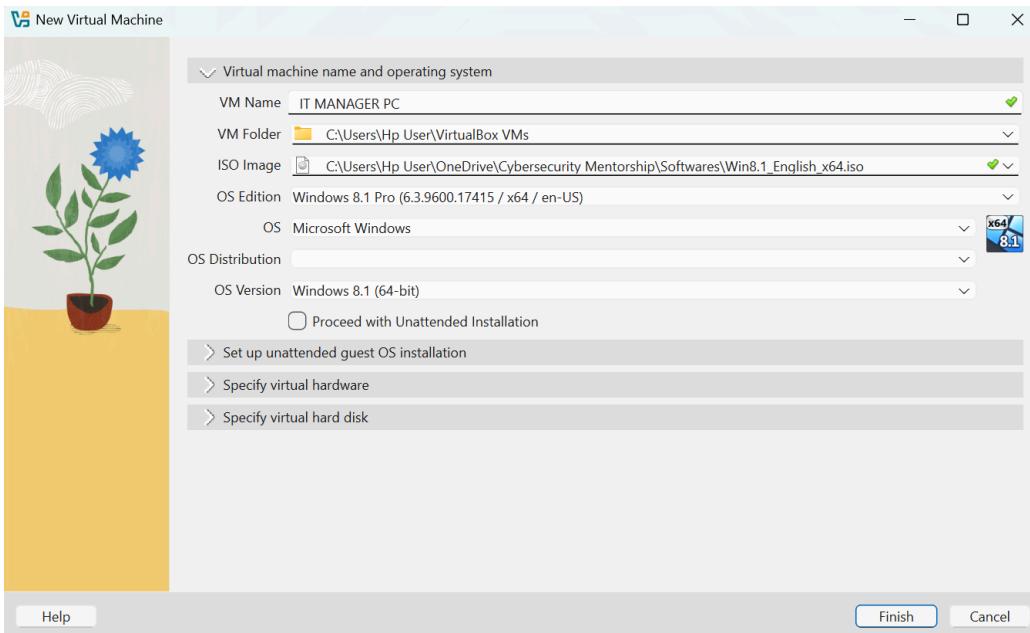




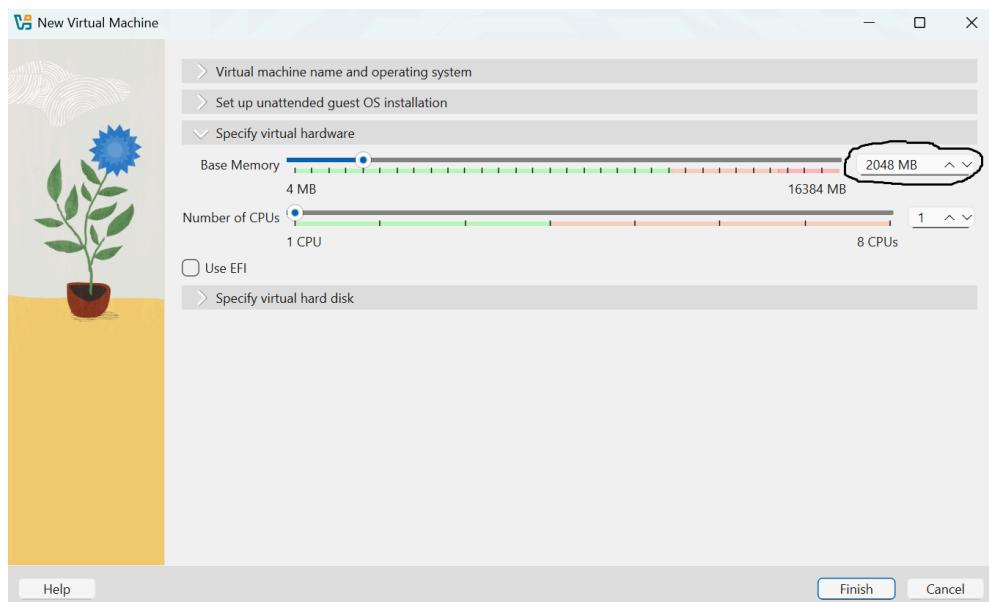
**Step 2:** I entered IT Manager PC as the virtual machine name and selected the Windows 8 ISO file that will be used for the installation process.



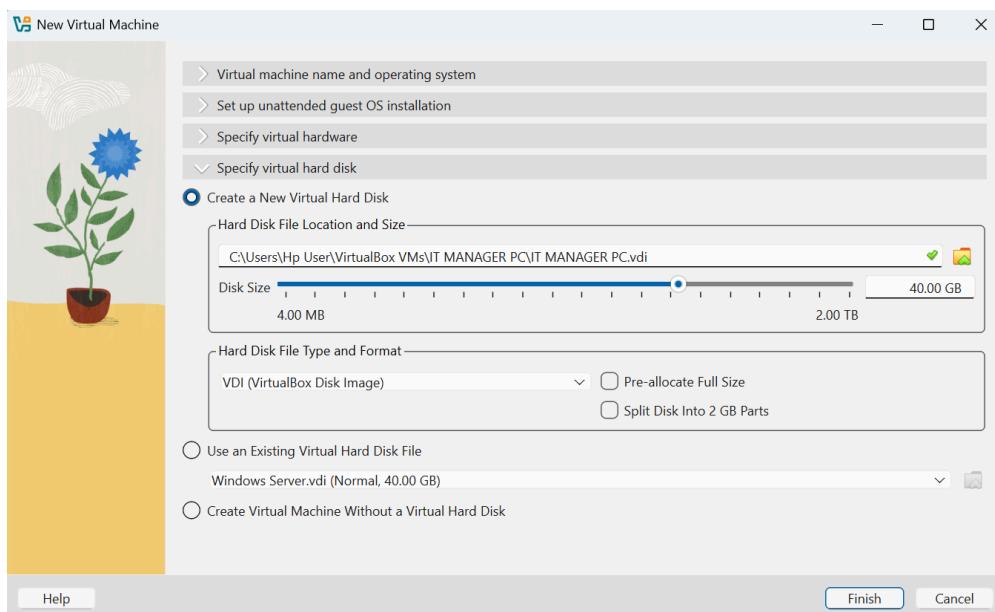
**Step 3:** I deselected the button to “ Proceed with Unattended Installation”



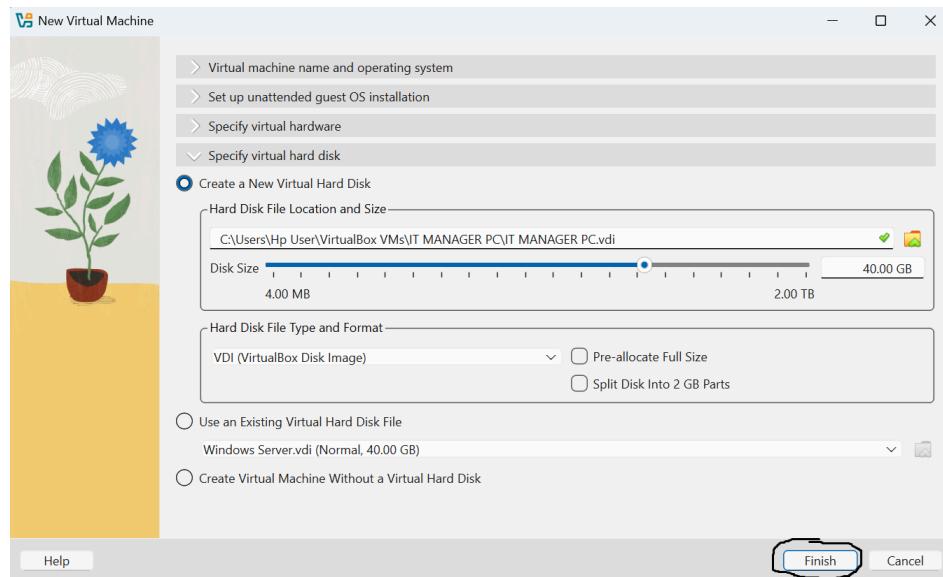
**Step 4:** I then opened the Virtual Hardware settings, reduced the default memory from 2048 MB to 1024 MB, and kept the number of CPUs at 1.



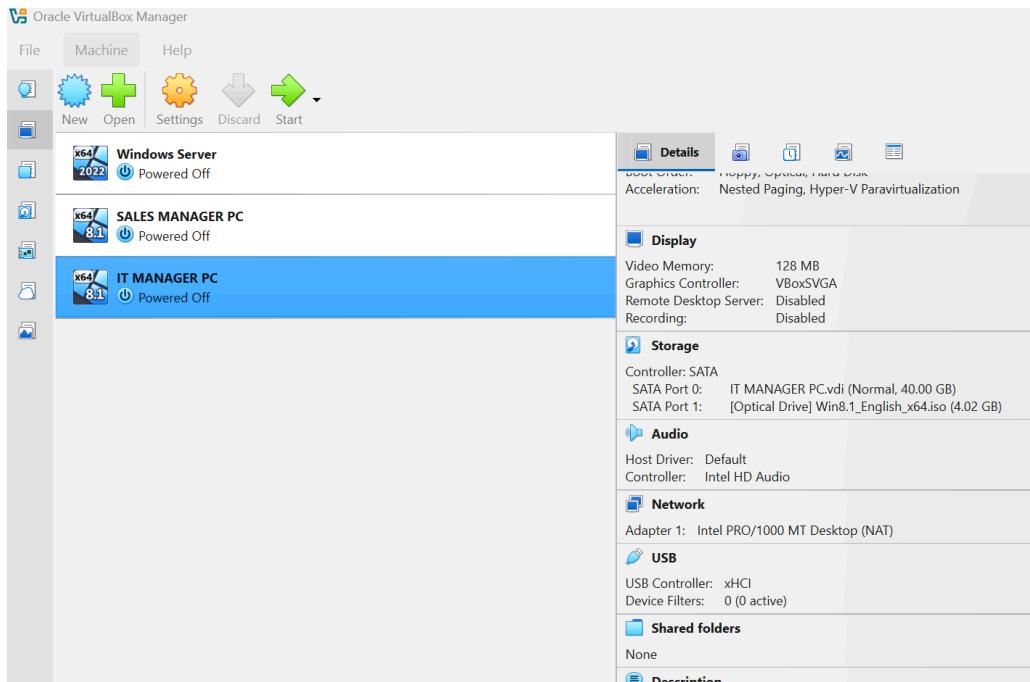
**Step 5:** I went to ‘Specify Virtual Hard Disk’, where the default storage size was 40 GB. I left this setting as it was since it’s sufficient for this setup, but you can change it depending on your storage needs.



**Step 6:** I then clicked “*Finish*” to complete the setup of the IT Manager PC.

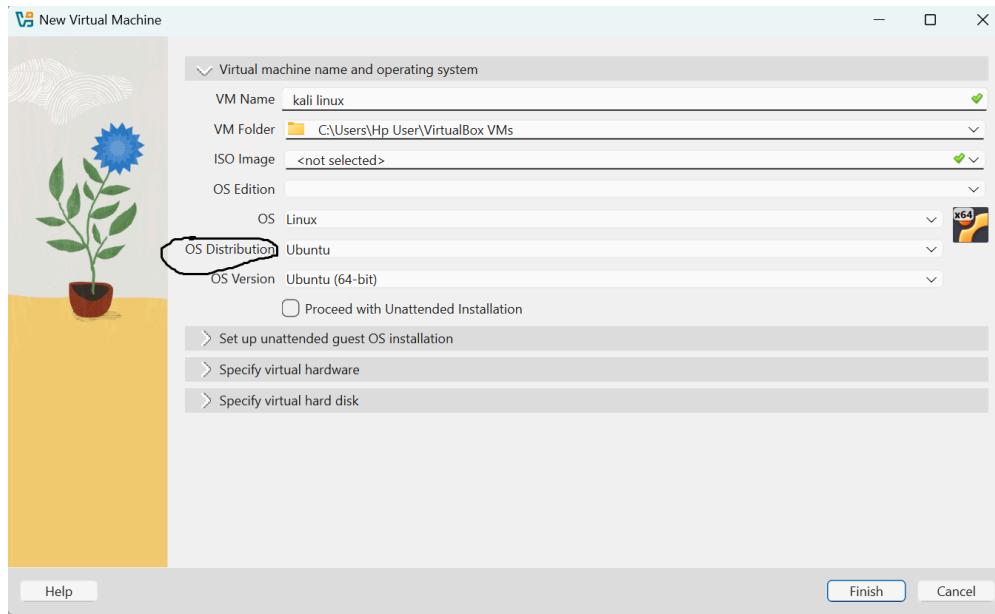


**Finally, our IT Manager PC is ready.**

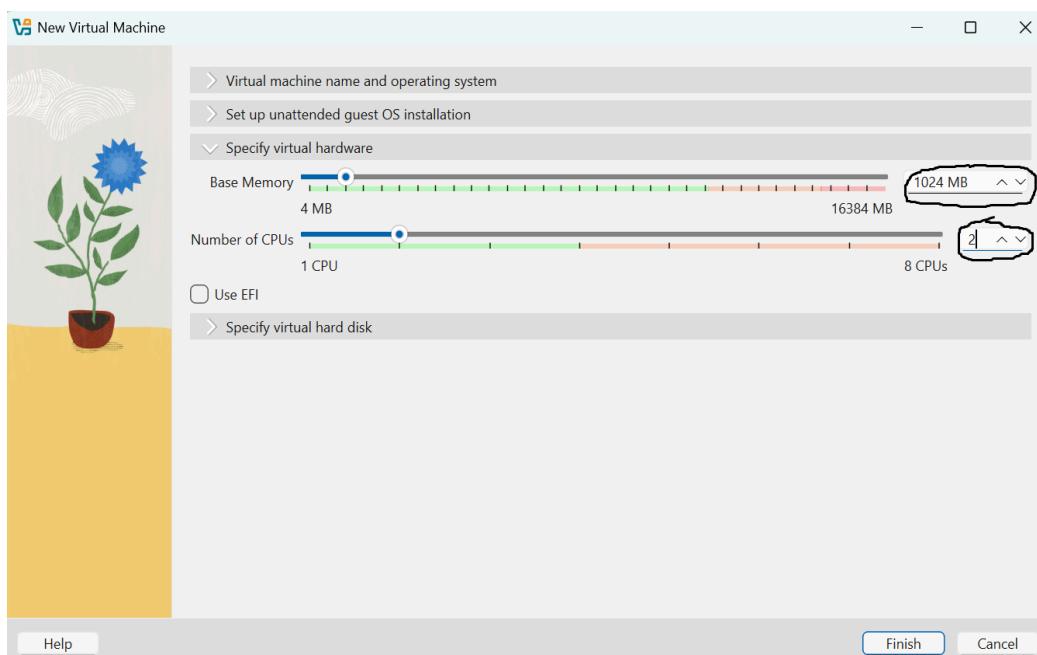


## CREATING KALI LINUX

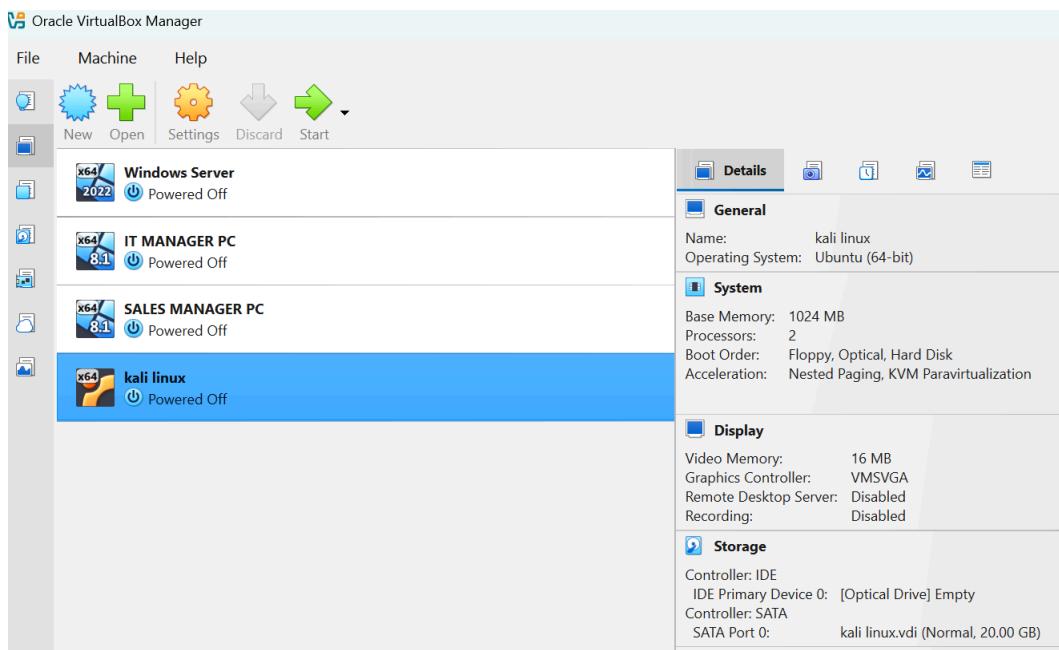
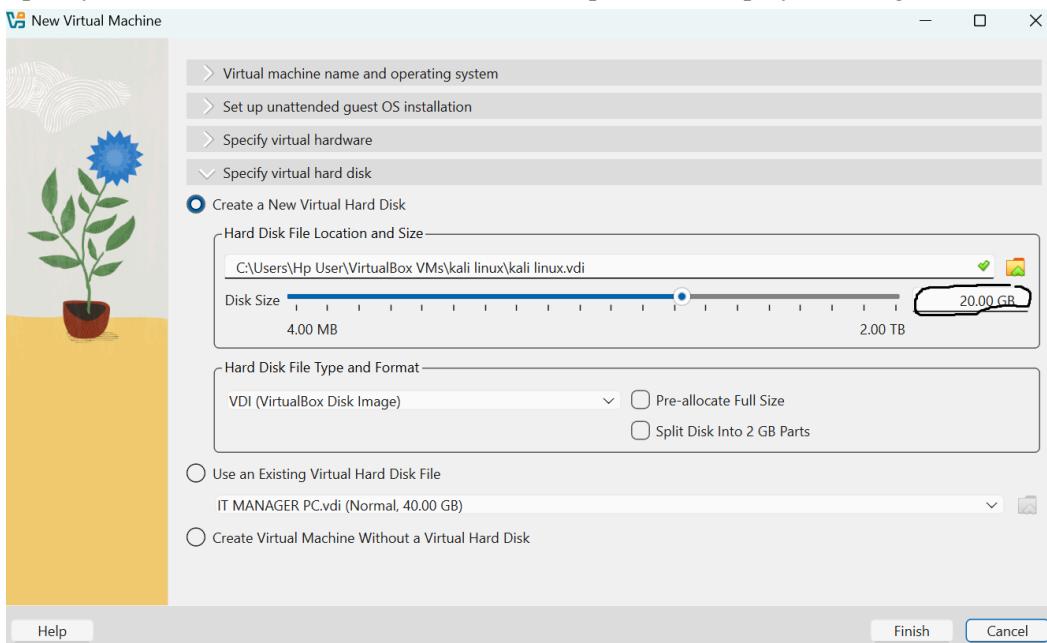
**Step 1:** Let's mount the Kali Linux ISO onto the virtual machine and name it "**Kali Machine**." The process follows the same steps used when mounting the **Windows Server** and **Windows 8** ISOs but select ubuntu for the OS Distribution . Here's the link:[Kali linux download](#)



**Step 2:** Adjust the base memory as you wish and the CPU.

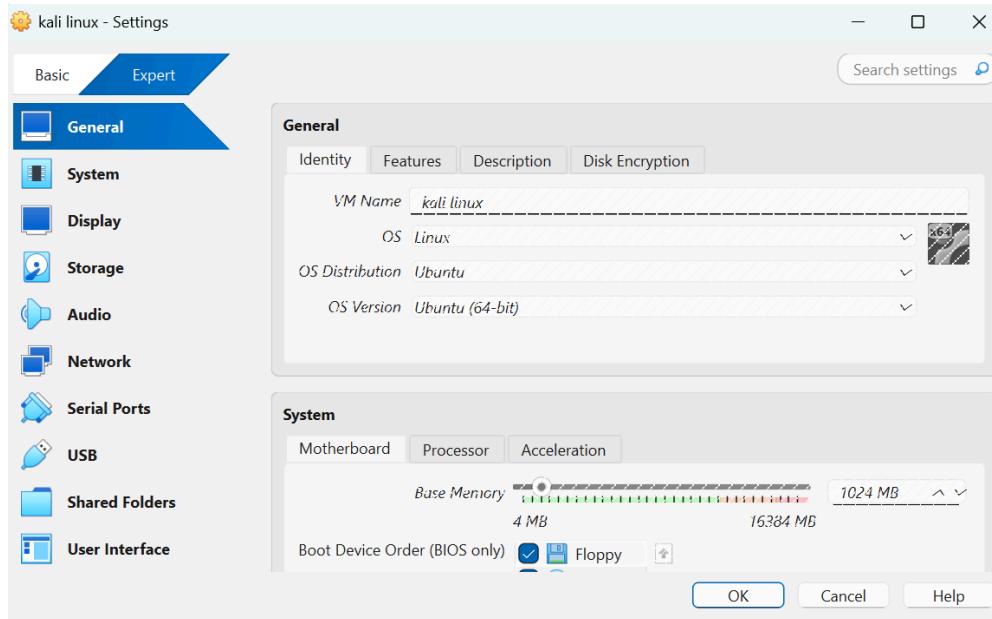


**Step 3 :** Specify the size of the virtual hard disk, then complete the setup by selecting **Finish**.

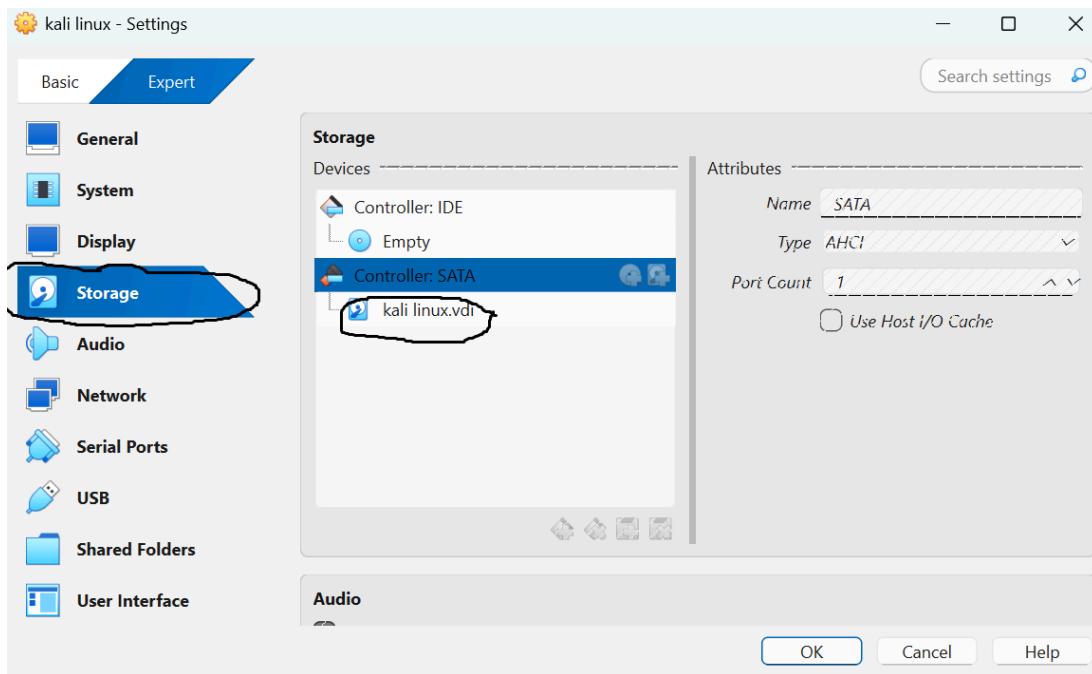


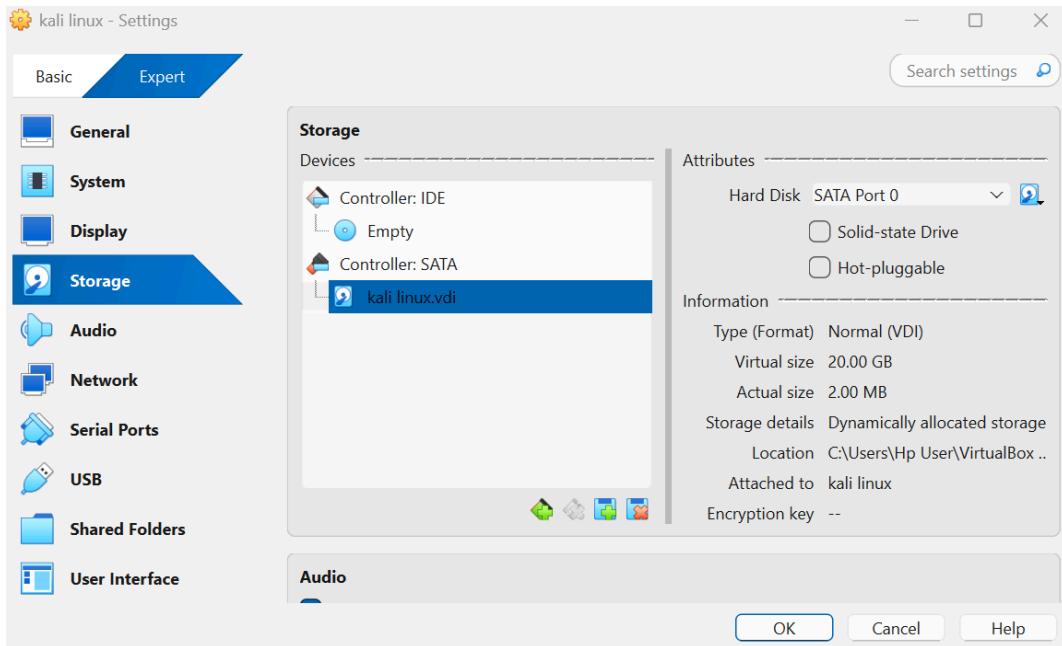
Note: To boot up Kali Linux, you need to add the operating system ISO file. Follow these diagram steps to add it correctly.

### Step 1: click on kali linux setting

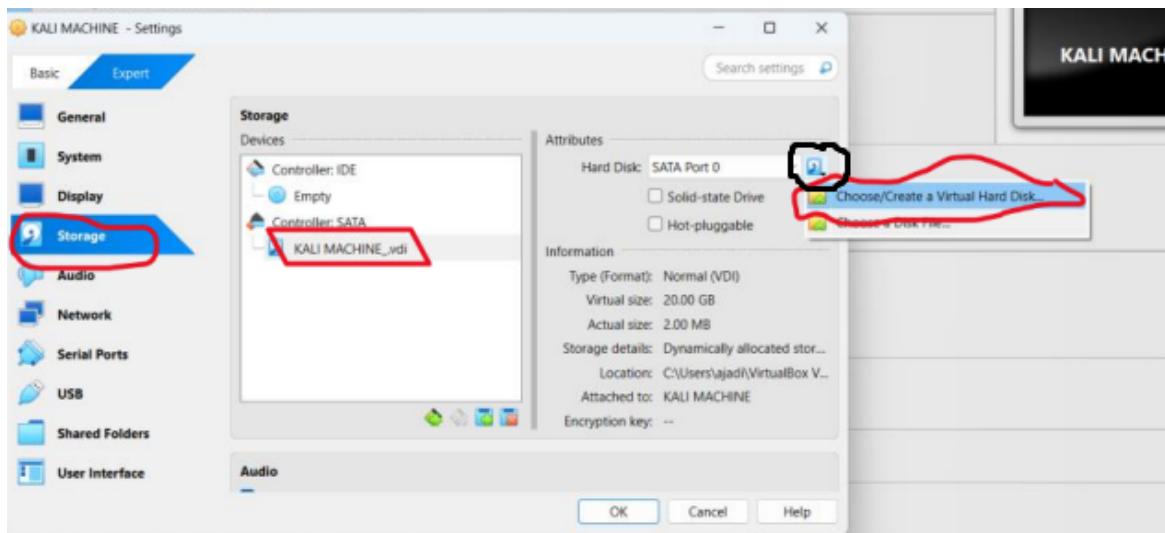


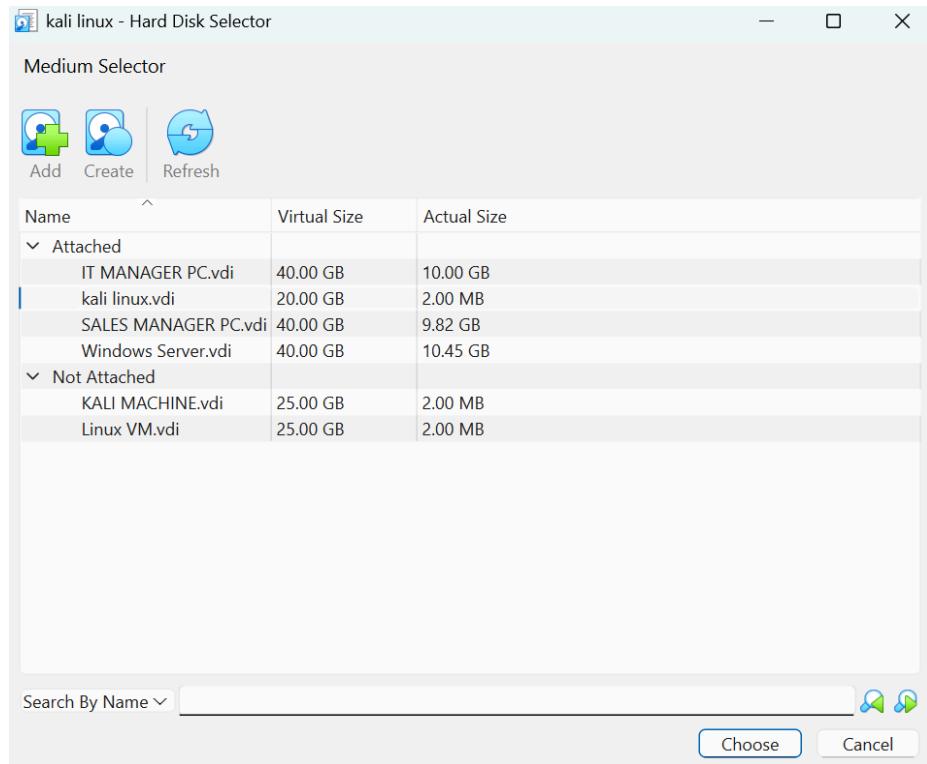
### Step 2: Select storage and click on the “kali linux vdi” file.



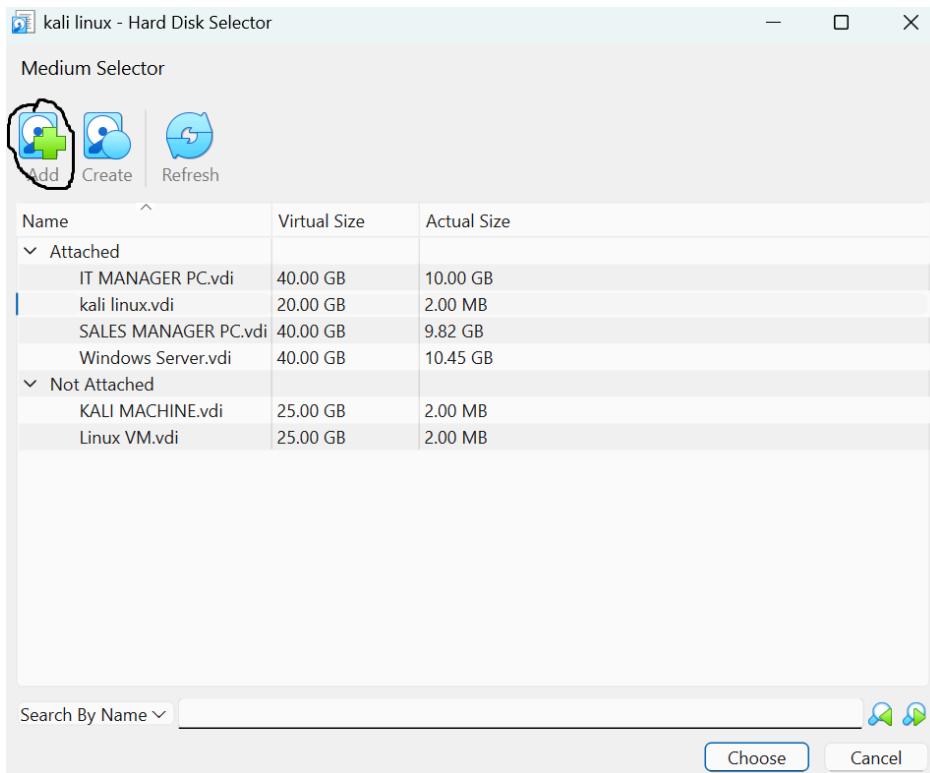


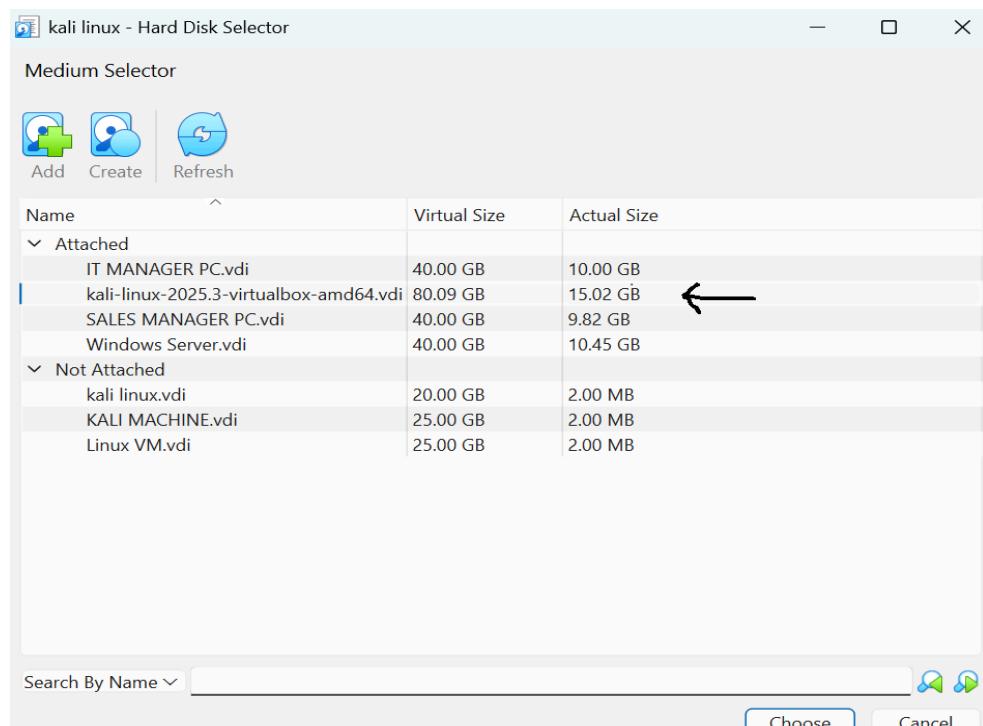
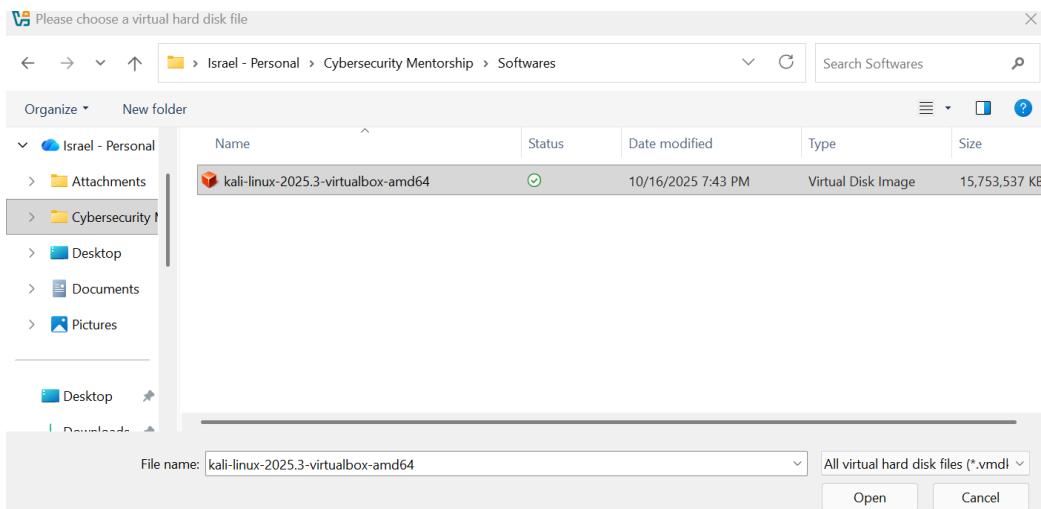
**Step 3:** Select the blue icon and click on “choose/create a virtual hard disk”



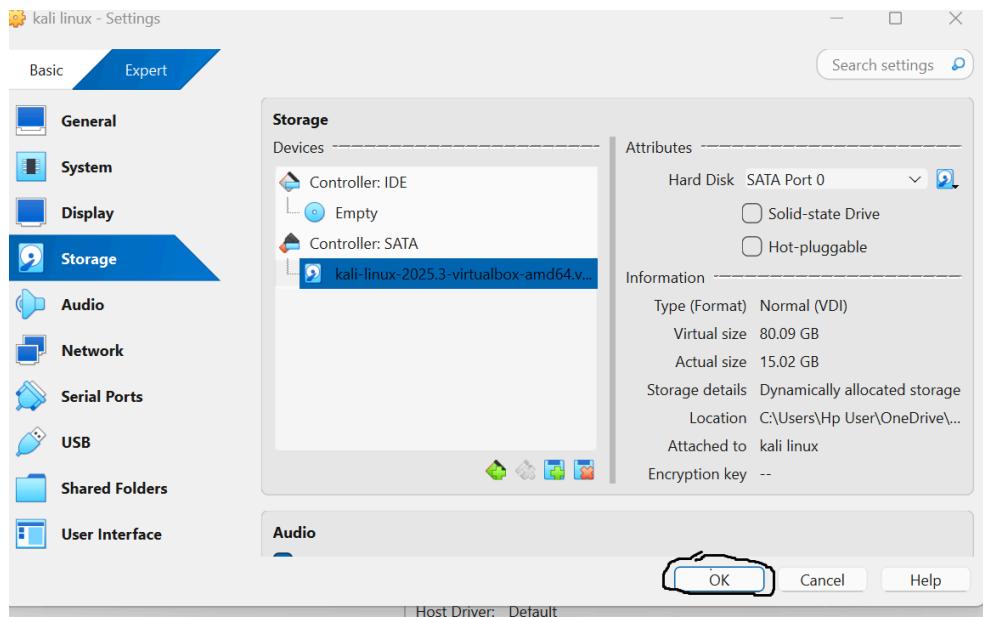


#### Step 4: Select add to add the file





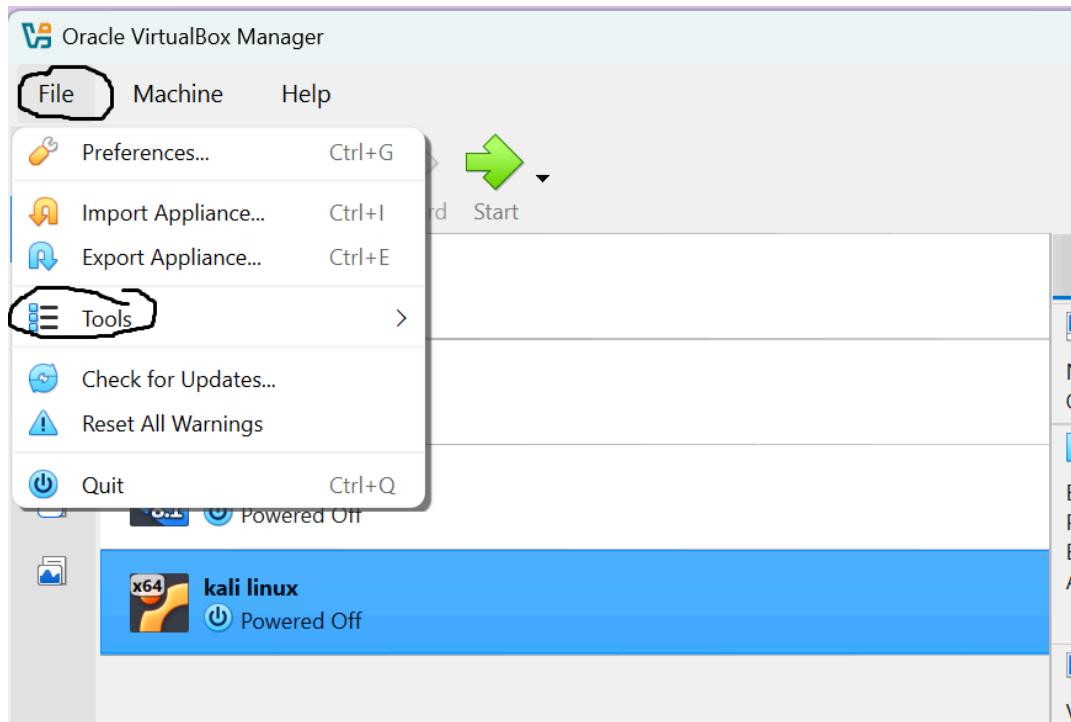
Then click the “OK” button to successfully launch.



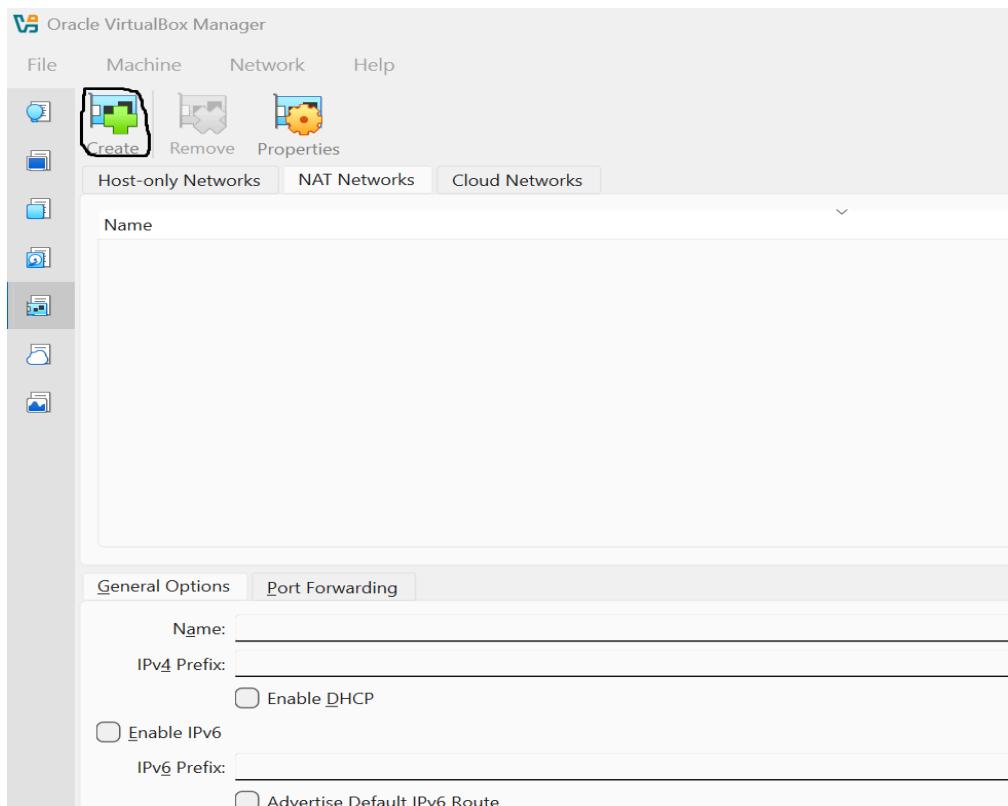
## Setting Up Operations and IT Subnets Using NAT Networks in VirtualBox

My reason for doing this is to simulate a real-world network environment. In most organizations, different departments or systems are separated to manage traffic better, improve security, and make the network easier to control.

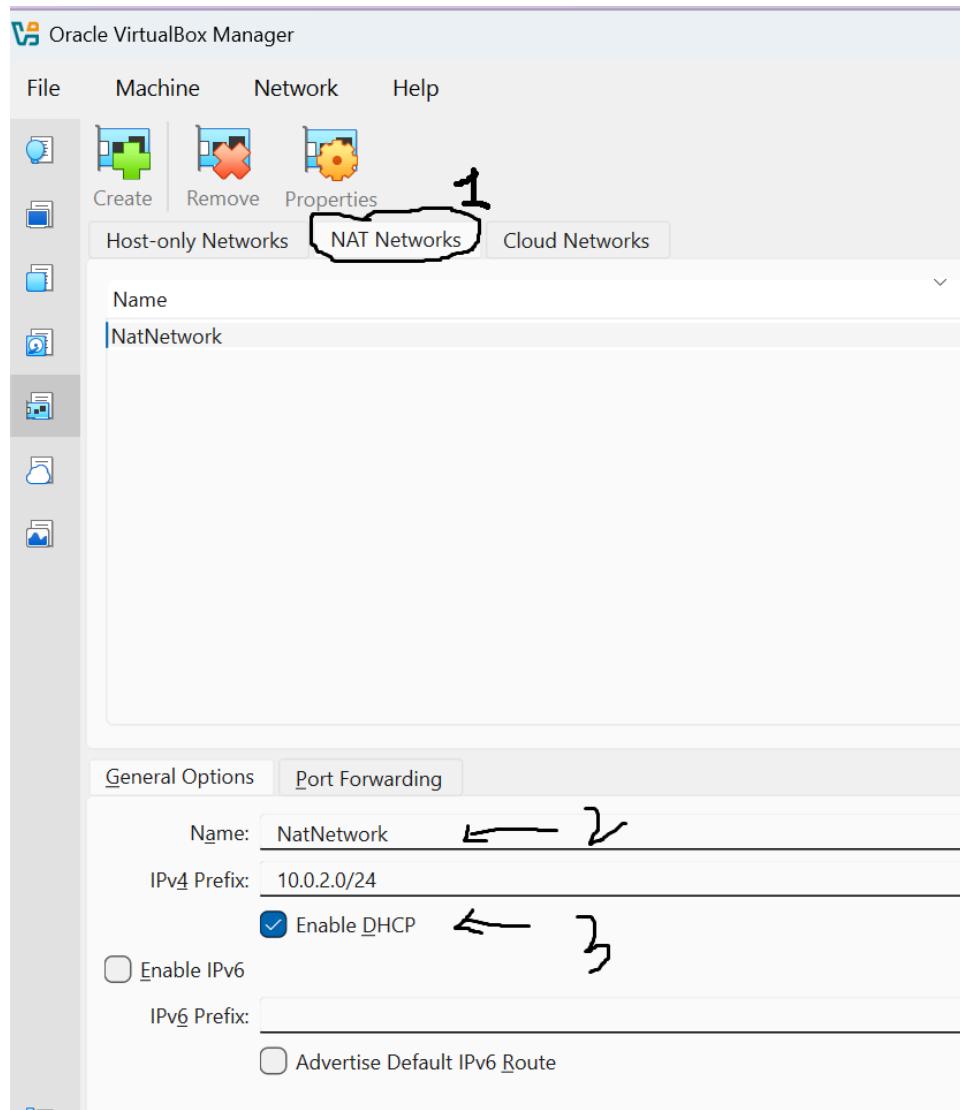
**Step 1:** Click on the “File” menu at the top-left corner of the VirtualBox window and select “ Tools”



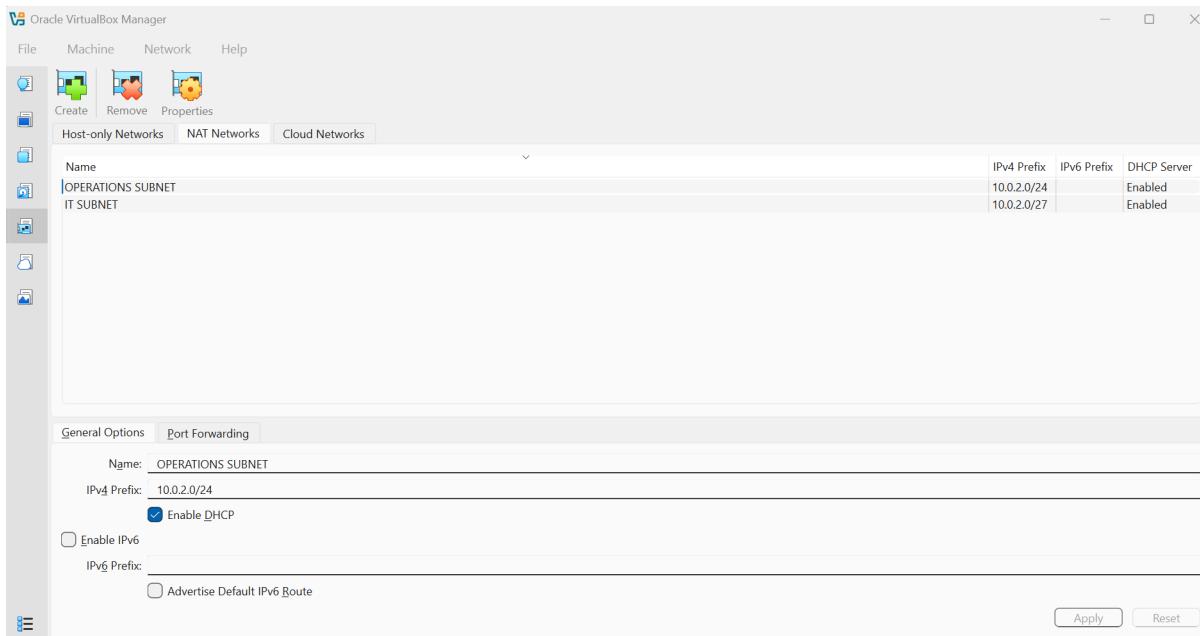
**Step 2:** In the window that opens, go to the Network tab, you should see this then click on the “Create” menu



**Step 3:** After clicking the **Create** menu, go to **NAT Networks** to confirm you are on the correct network. Then, edit the network name and change it to **Operations Subnet**. Make sure to enable **DHCP** so that IP addresses are assigned automatically on “**Apply**”



**Step 4:** Do the same for the **IT subnet**, then the page should look like this. That means the both Subnets that have been successfully configured under the NAT Networks.

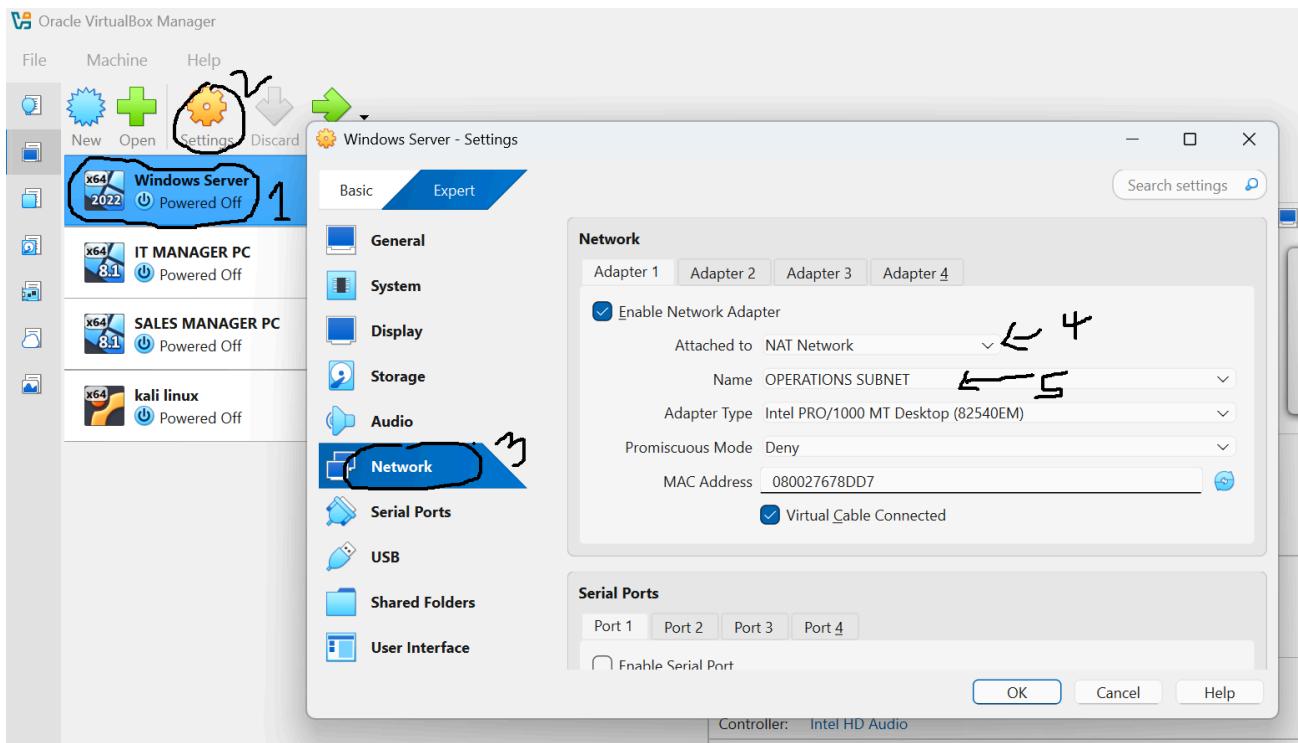


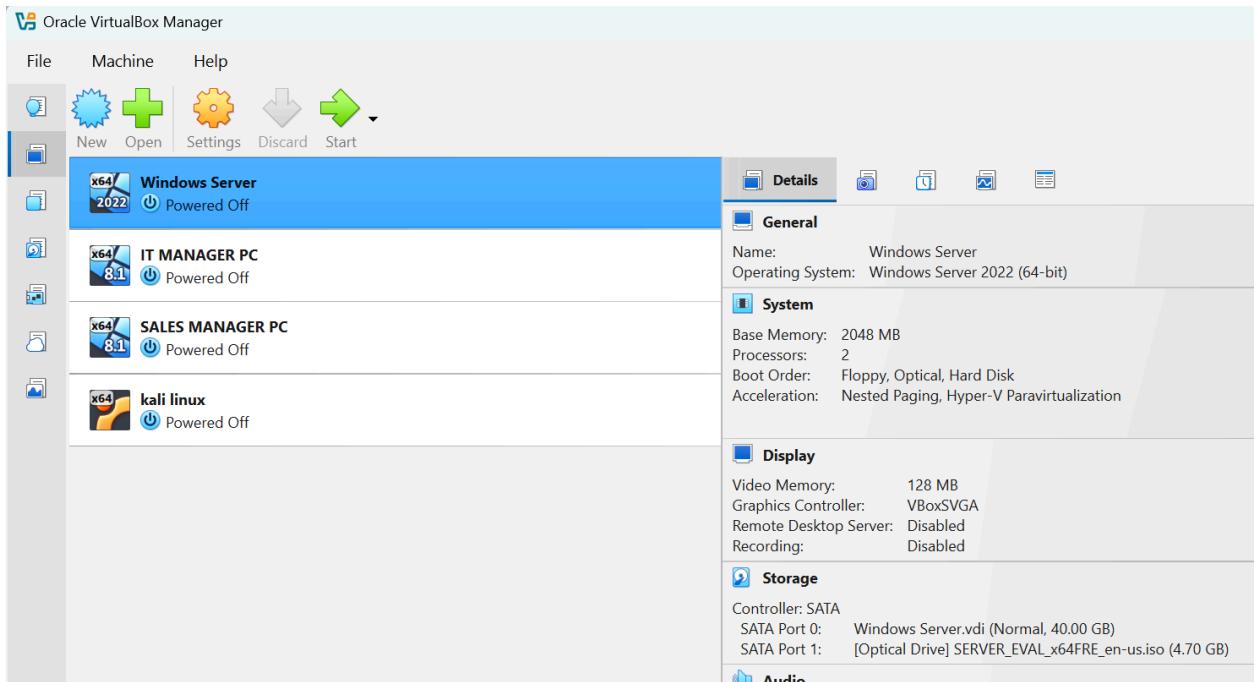
## Configuring Virtual Machines to Join a NAT Network

1. select the virtual machine you want to configure. For example, choose Windows Server
2. Click **Settings**, then go to the **Network** section.
3. Under **Adapter 1**, click **Attach to** and choose **NAT Network**.
4. From the dropdown menu, select the network name **Operations Subnet**.
5. Click “OK”

**Note:** For other VMs, like Kali Linux, repeat the same steps but select their corresponding NAT network, in this case “IT Subnet”.

This ensures that each VM is assigned to the correct subnet and can communicate with other machines within the same network.





## Conclusion

I have successfully installed the operating systems and completed my virtual home lab setup. Thank you for following along, and I hope this guide helps anyone looking to build and learn from their own virtual lab.