Question 3:

Virtual Machine (VM) - Overview

A virtual machine (VM) is a software emulation of a physical computer that runs an operating system and applications just like a physical machine. VMs allow you to create multiple isolated environments on a single physical host, each with its own virtual hardware configuration. This virtualization technology provides several advantages, such as the ability to run multiple operating systems on a single physical machine, easy migration of VMs between different hosts, and improved resource utilization.

The virtualization layer called the hypervisor, is responsible for managing the VMs and enabling communication between them and the underlying physical hardware. There are two types of hypervisors: Type 1 hypervisor (bare-metal) runs directly on the physical hardware, while Type 2 hypervisor runs on top of an existing operating system.

Oracle VirtualBox - Overview

Oracle VirtualBox is a popular Type 2 hypervisor that allows you to create and manage virtual machines on your desktop or laptop. It supports a wide range of guest operating systems, including Windows, Linux, macOS, and more. VirtualBox is free and open-source, making it an excellent choice for developers, testers, and anyone interested in exploring virtualization.

How to Install VirtualBox

Here's a step-by-step guide to installing Oracle VirtualBox on your Windows, macOS, or Linux computer:

Step 1: Download VirtualBox

1. Go to the official VirtualBox website: https://www.virtualbox.org/

2. Click on the "Downloads" link in the top navigation menu.

Step 2: Choose the Correct Package

1. On the Downloads page, you'll see various packages for different host operating systems. Select the appropriate package for your OS (e.g., Windows, macOS, or Linux).

Step 3: Install VirtualBox

1. For Windows:

- Download the installer for Windows and double-click on the downloaded file to start the installation.

- Follow the on-screen instructions and accept the license agreement. - Choose the components you want to install and the installation path. - Complete the installation process.

2. For macOS:

- Download the macOS version of VirtualBox.

- Double-click on the downloaded DMG file to open it.

- Double-click on the VirtualBox package icon to start the installation. - Follow the on-screen instructions to complete the installation.

3. For Linux:

- Download the appropriate package for your Linux distribution (e.g., .deb for Debian/Ubuntu-based systems, .rpm for Red Hat/Fedora-based systems). - Install VirtualBox using the package manager of your Linux distribution. For example, for Ubuntu, use the following command in the terminal:

sudo dpkg -i <VirtualBox\_package\_name>.deb

- You may need to install additional dependencies if prompted by the package manager.

Step 4: Post-installation Configuration (All Operating Systems)

1. After installation, you might need to add your user account to the "vboxusers" group (Linux) or "VirtualBox Users" group (Windows) to grant permissions to manage VMs.

Step 5: Launch VirtualBox

1. Once the installation is complete, you can launch VirtualBox from your application menu (Windows and Linux) or from the Applications folder (macOS).

Congratulations! You now have Oracle VirtualBox installed on your computer and can start creating and managing virtual machines for various purposes, including development, testing, and exploration of different operating systems.

Once the VM has been installed, visit https://www.osboxes.org/ download a Ubuntu 22.04 image and start it through your VirtualBox.

Install Nginx inside the Ubuntu machine and host a website.

Come back to your host machine (windows/Linux/mac) and scan the virtual machine using Nmap. Create the documentation of the process and the output of the scan. Observe the ports which are open.

1. **Download VirtualBox** from <https://www.virtualbox.org/wiki/Downloads> , downloaded for windows with right version.
   * **install VirtualBox** in **Windows**: Run the .exe installer and follow the prompts.
   * **linux (Ubuntu)**:
     + Download the .deb package from the VirtualBox website.
     + Use the terminal to install the package. First, navigate to the directory where you downloaded the .deb file and run:- sudo dpkg -i <VirtualBox\_package\_name>.deb

If there are any missing dependencies, run:

sudo apt-get install -f

1. **Post-Installation Setup (Linux Only)**:
   * Add your user to the vboxusers group to avoid permission issues:

sudo usermod -aG vboxusers $USER

1. **Launch VirtualBox**:
   * Once installation is complete, open VirtualBox from your system's applications menu or folder.

**Part 2: Download and Setup Ubuntu VM**

**Step 1: Download Ubuntu 22.04 VM Image**

1. **Visit the Osboxes website**: <https://www.osboxes.org/ubuntu/>
2. **Scroll down** to find **Ubuntu 22.04 LTS (Jammy Jellyfish)**.
3. **Download the VirtualBox Image (VDI)**:
   * You will see an option to download the **Ubuntu 22.04 LTS** VirtualBox image (VDI format). It should be a file around **2.59GB** in size with a SHA256 checksum of 7c7641bdb3f3dfa1beaa76b4edd7d3a2a95a4aa27c88b21cb569e8b40cd8e925.

**Step 2: Import the VM Image into VirtualBox**

1. **Open VirtualBox**.
2. **Create a new Virtual Machine**:
   * Click New, and name your VM (e.g., "Ubuntu 22.04").
   * Choose Linux as the type and Ubuntu (64-bit) as the version.
   * Allocate memory (RAM) for the VM (at least 2GB recommended).
3. **Use the downloaded VDI**:
   * Instead of creating a virtual hard disk, select **"Use an existing virtual hard disk file"** and browse to the .vdi file you downloaded from Osboxes.
4. **Finish creating the VM**.

**Step 3: Start the Ubuntu VM**

1. Once the VM is created, select it from the VirtualBox interface and click **Start**.
2. The VM will boot, and you should be taken to the Ubuntu login screen. Use the default credentials provided by Osboxes, typically:
   * **Username**: osboxes
   * **Password**: osboxes.org

**Part 3: Install Nginx Inside the Ubuntu VM**

Now that your Ubuntu 22.04 VM is up and running, the next step is to install **Nginx** and host a simple website.

**Step 1: Update the Package List**

Once you're logged into the VM, open a terminal and run:

sudo apt update

**Step 2: Install Nginx**

Install the Nginx web server by running:

sudo apt install nginx -y

**Step 3: Start Nginx**

After installation, start the Nginx service:

sudo systemctl start nginx

To ensure that Nginx starts automatically on boot:

sudo systemctl enable nginx

**Step 4: Verify Nginx is Running**

You can verify that Nginx is running by checking its status:

sudo systemctl status nginx

Nginx runs on port 80 by default. If everything is working, open a web browser on your host machine and type the IP address of your virtual machine (you can get it by running ip a on the VM).

You should see the default Nginx welcome page.

**Part 4: Scan the Ubuntu VM Using Nmap**

Once Nginx is running, you will scan the VM from your **host machine** using **Nmap**.

**Step 1: Install Nmap**

If you don’t have Nmap installed on your host machine, install it:

* **Windows**: Download and install Nmap from https://nmap.org/download.html.
* **macOS**: Install via Homebrew:

brew install nmap

* **Linux** (Ubuntu):

sudo apt install nmap

**Step 2: Find the IP Address of Your VM**

In your Ubuntu VM, run:

ip a

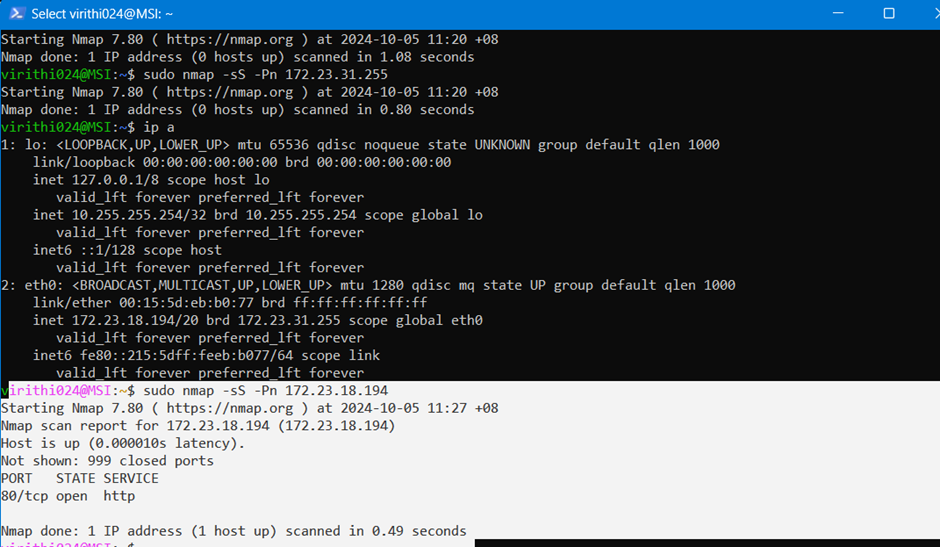
This will show you the IP address of the VM. It will usually be under an interface like eth0 or enp0s3.

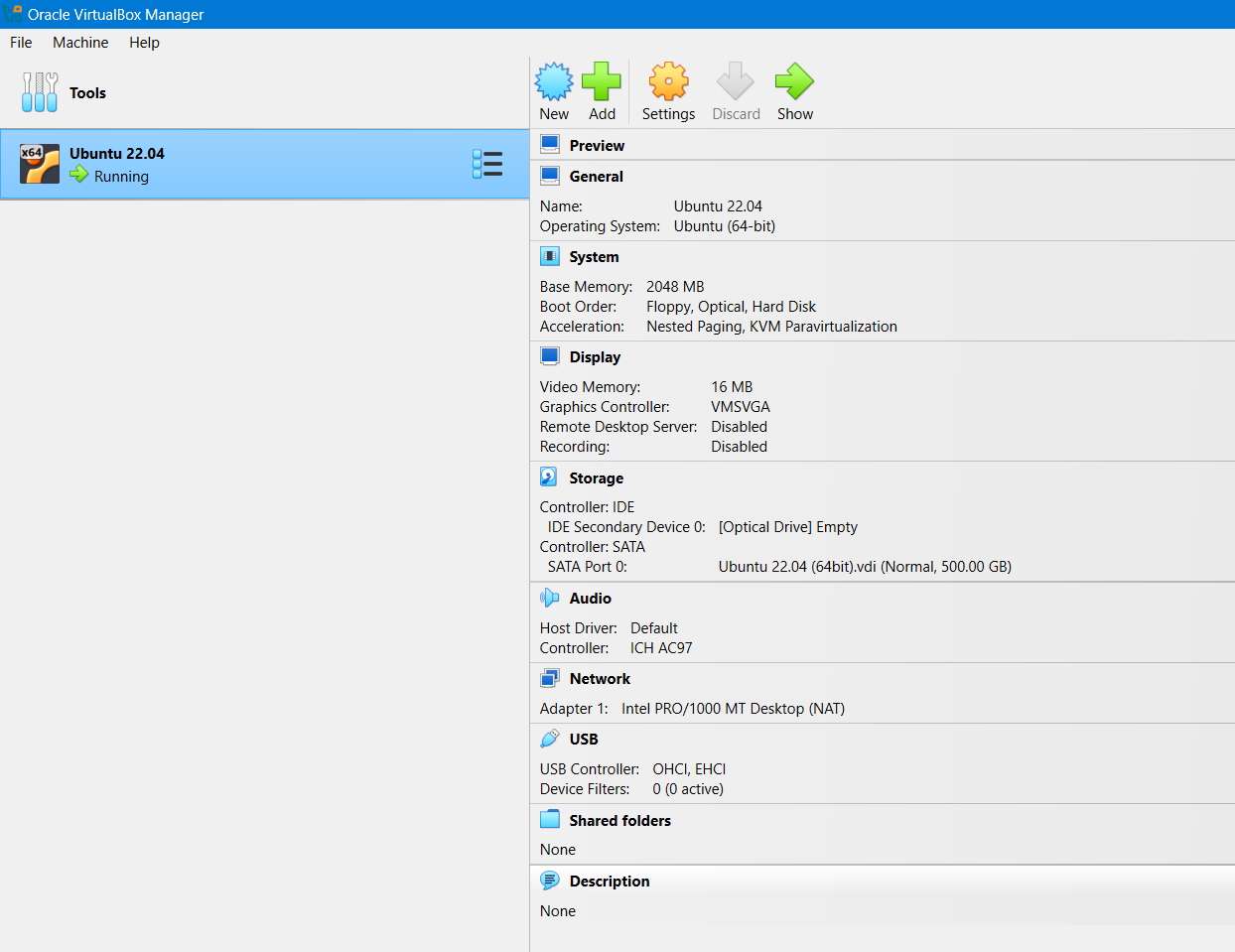
**Step 3: Run the Nmap Scan**

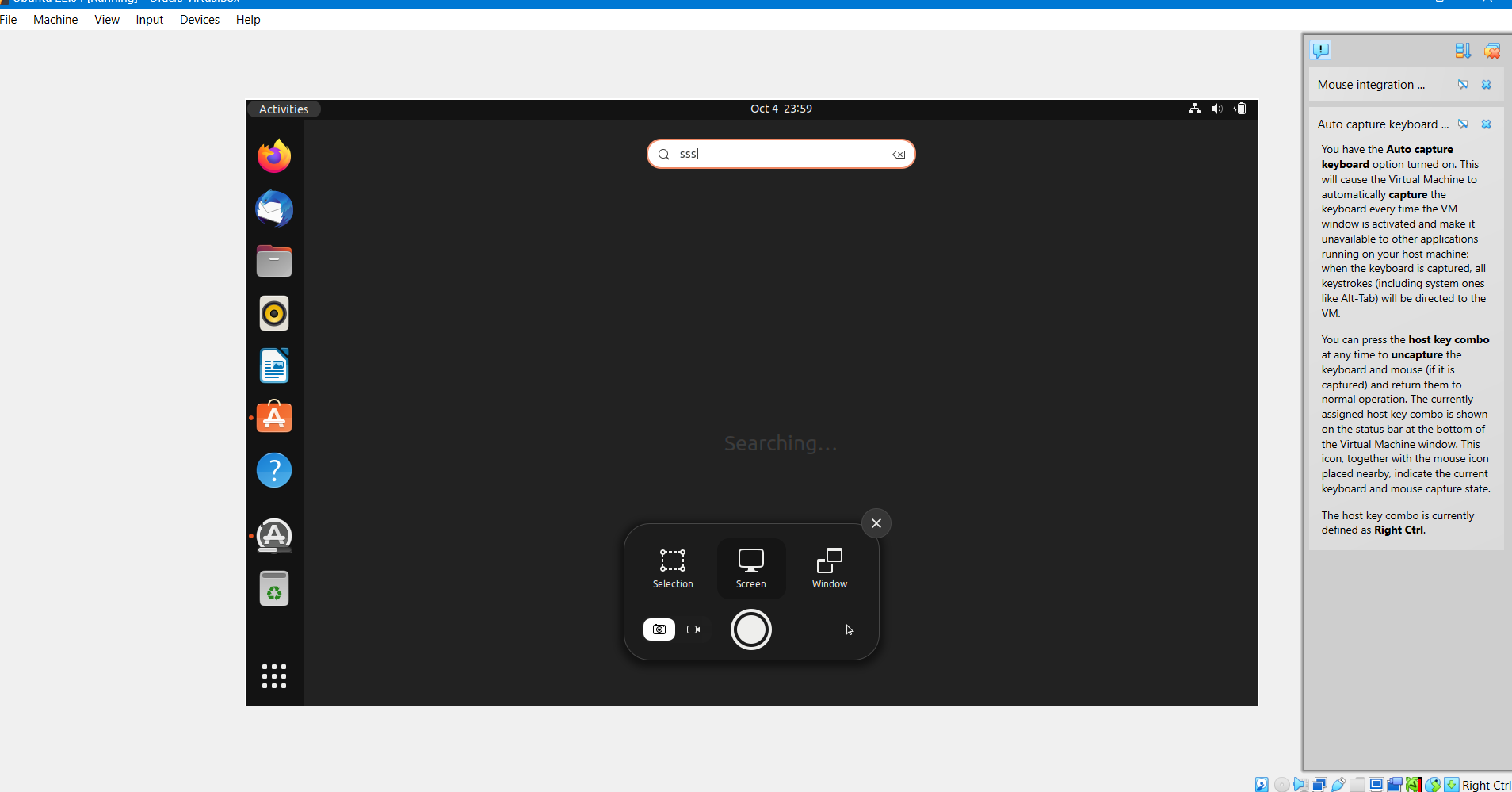
Now, from your **host machine**, run the following command, replacing <VM\_IP\_Address> with the actual IP address of your VM: sudo nmap -sS 172.23.31.255

This will perform a **TCP SYN scan** to identify open ports on the VM. You should see that port 80 (used by Nginx) is open

**Output:**







x