**Q-1.** Deploy a website on localhost using either apache2 or Nginx. Create a DNS name for this website as ‘awesomeweb’. You can use any web template you want or can write your own simple HTML code.

**Write the detailed documentation with the steps involved.**

Steps:

* Open powershell in administrator mode
* type wsl and press enter
* To install Nginx, type the below commands

sudo apt update

sudo apt install nginx

To verify the installation, type the below command

sudo systemctl status nginx

* Create a directory for the website and add an html file

sudo mkdir /var/www/awesomeweb

sudo touch /var/www/awesomeweb/index.html

sudo vi /var/www/awesomeweb/index.html

* Create and update configuration file to setup dns name for the website

sudo vi /etc/nginx/sites-available/awesomeweb

* Enable the site and restart nginx instance to view the changes

sudo ln -s /etc/nginx/sites-available/awesomeweb /etc/nginx/sites-enabled/

sudo systemctl restart nginx

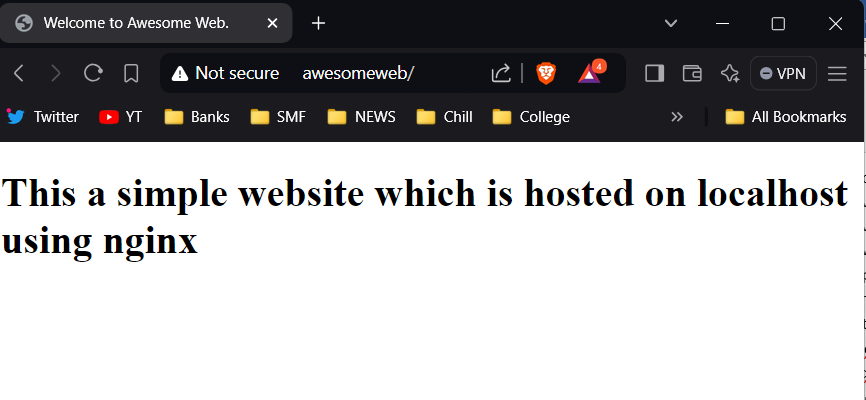
* Update the hosts file

sudo vi /etc/hosts

127.0.0.1 awesomeweb

and save the file

* Open the browser and type "http://awesomeweb/" to verify the website setup



**Q-2.** A website can have many subdomains and different services are running on them. Write a Python script to check the status of the subdomains which are up or down. The script should automatically check the status every minute and should update it in tabular format on the screen.

Steps:

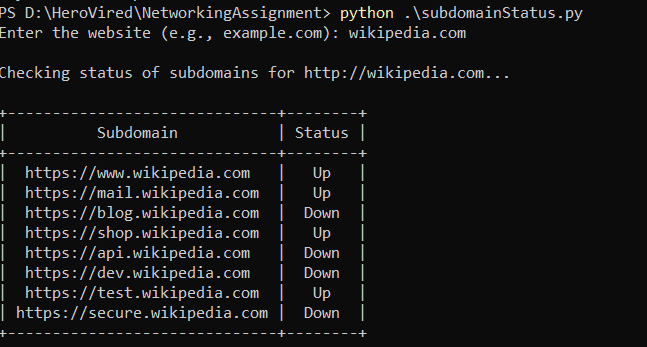
* To view the results in tabular format, "prettytable" library needs to be installed

pip install requests prettytable

* Create a python file named "subdomainStatus.py" and update the code as per the requirements
* Run the script by typing the following command in powershell or any ide

python /path\_to\_the\_file/subdomainStatus.py

* Enter the website name you want to check the subdomains status when prompted
* This script checks the status of the specified subdomains every minute
* Tested the script against "youtube.com" and "wikipedia.com" for validation



**Q3.** Install Oracle VirtualBox, download a Ubuntu 22.04 image and start it through virtual box. Install Nginx inside Ubuntu 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

Steps:

1. Installing VirtualBox – Navigated to <https://www.virtualbox.org/wiki/Downloads> and downloaded Windows hosts package.



1. For installation, ran the downloaded file. Installation failed as there was no “Microsoft Visual C++ 2019 Redistributable Package” installed on the machine. Installed the required package from <https://learn.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-170#visual-studio-2015-2017-2019-and-2022> , restarted the machine and proceeded with virtualbox installation on the windows.



1. Add your user account to the “VirtualBox Users” to grant permissions to manage VMs

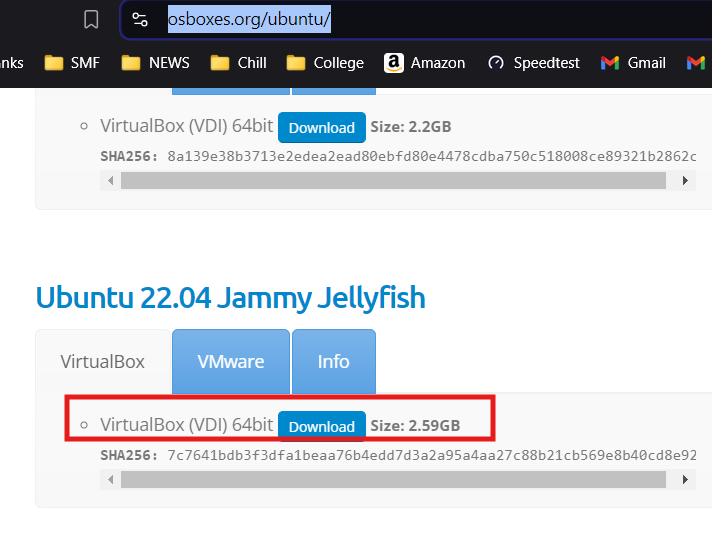
Steps:

* Right-click on the Start button and select "Computer Management."
* In the left pane, expand "Local Users and Groups."
* Click on "Groups."
* Find and double-click on "VirtualBox Users."
* In the "VirtualBox Users Properties" window, click "Add."
* Type your username in the "Enter the object names to select" field and click "Check Names" to ensure it's correct.
* Click "OK" to add your account.
* Click "OK" in the VirtualBox Users Properties window to save your changes.
* Close the Computer Management window.
* Restart Your Computer (if necessary)

1. Download and install ubuntu image in virtualbox

Steps:

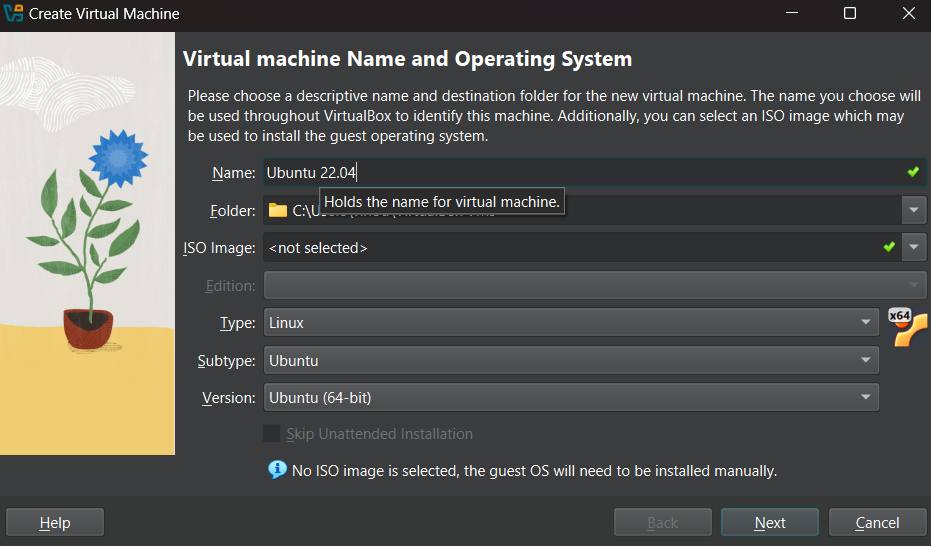
* Download ubuntu 22.04 image from <https://www.osboxes.org/ubuntu/>



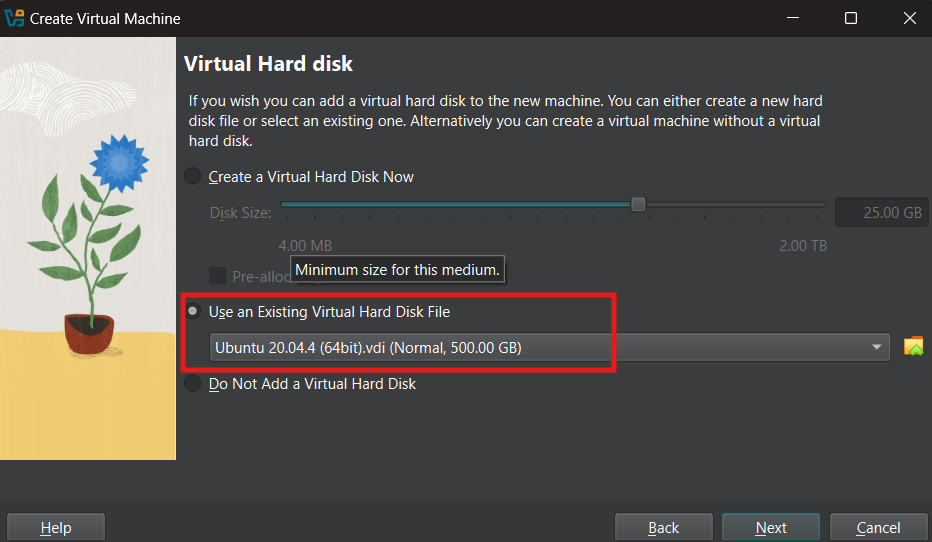
* Once downloaded, extract the files
* Go to the virtualbox, click on new



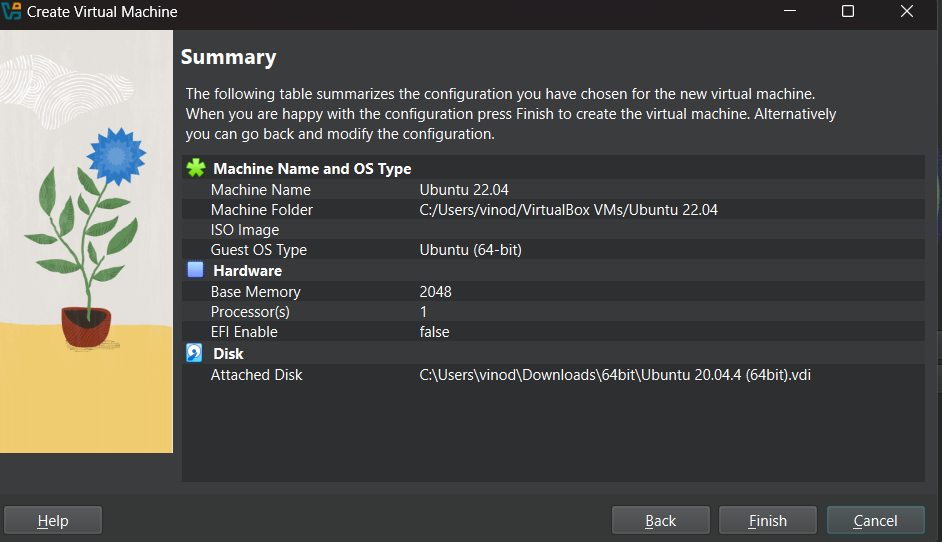
* Enter the name and type of VM



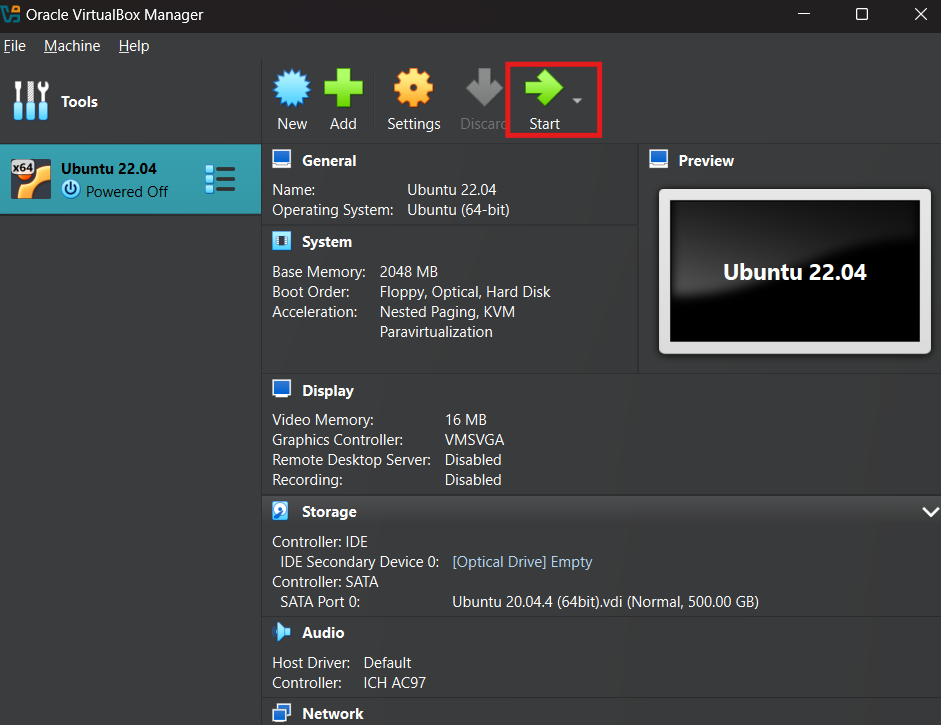
* Select the downloaded and extracted VDI file



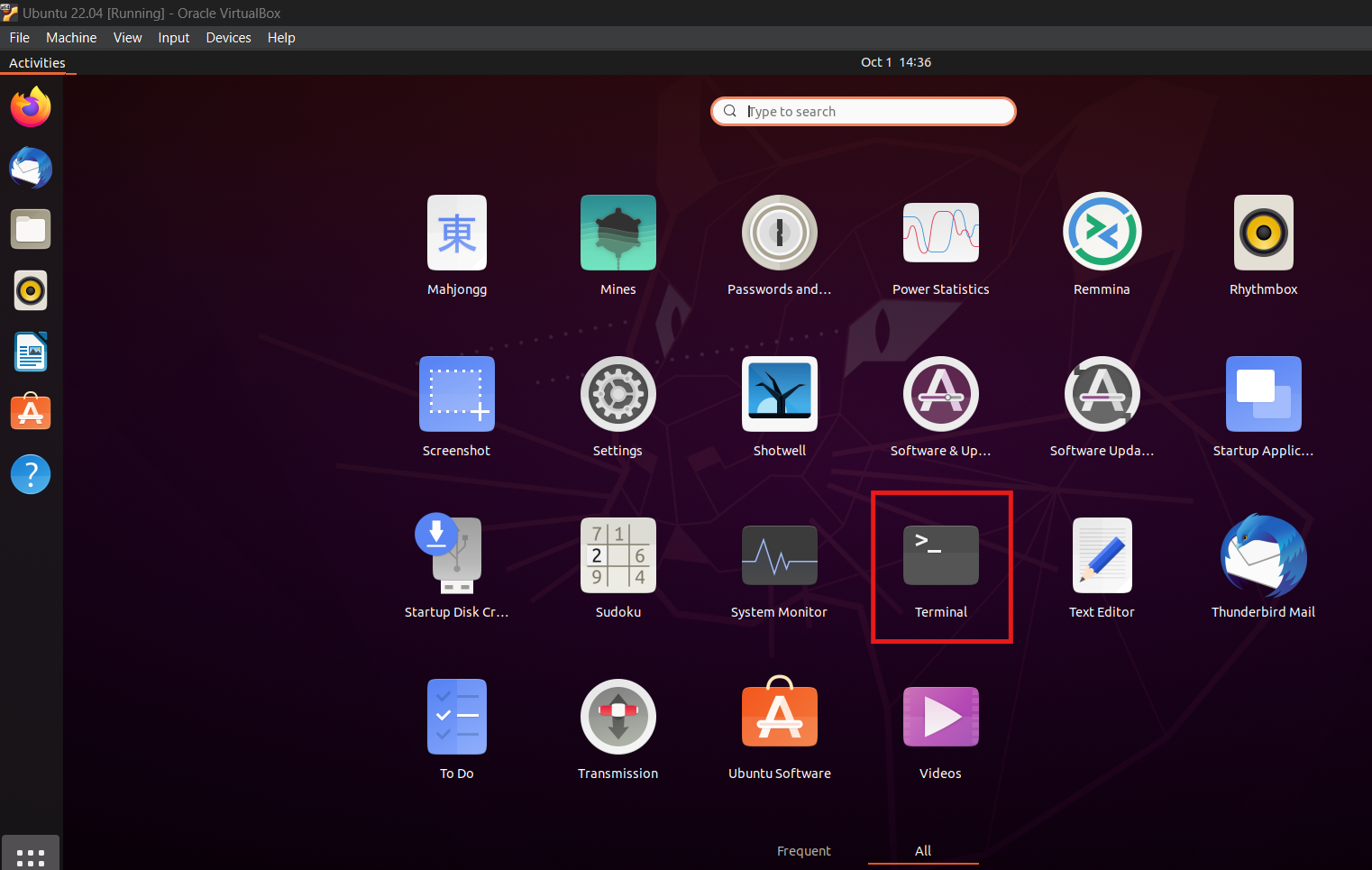
* Verify the summary and click on “Finish”



* Once created, click on “Start” to start the vm



* Open the terminal to install nginx and host a website



Sudo apt update

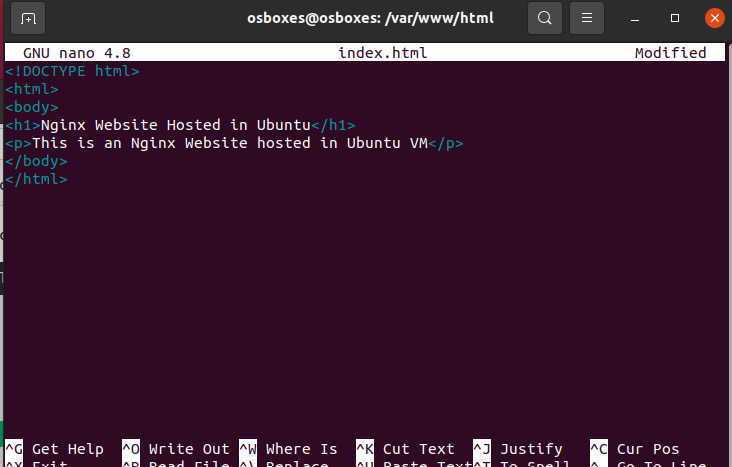
Sudo apt install nginx

Sudo systemctl start nginx

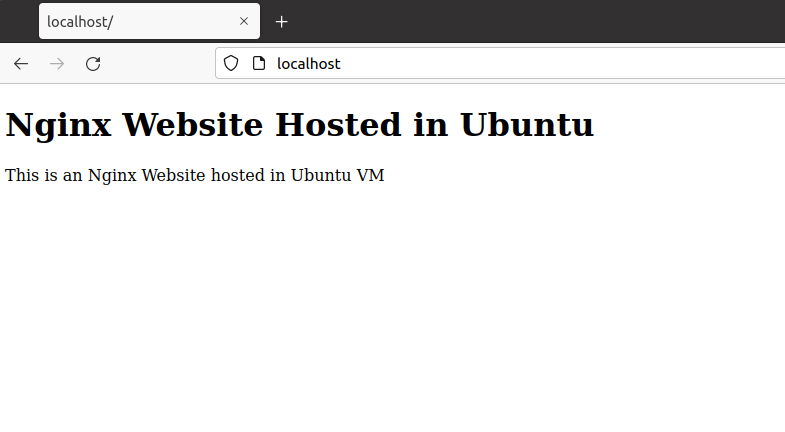
Cd /var/www/html

Sudo touch index.html

Sudo nano index.html



* Navigate to <http://localhost> to validate the hosted website



1. Scan the vm from host machine using Nmap and observe the ports that are open

Steps:

* Download and install nmap from nmap.org
* Press Windows+R (for Windows OS system), type cmd and press enter
* Type nmap -Pn <ipaddressOfVm> to view the results

