Q1. 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 detailed documentation with the steps involved

Steps 1: First install the Nginx using the below command

```
sudo apt update
sudo apt install nginx
```

Step 2: Check the status of nginx

```
systemctl status nginx
```

Step 3: If Nginx shows not running then we can start the Nginx

```
sudo systemctl start nginx
```

Step 4: We can check the status again whether it has started or not

```
systemctl status nginx
```

If it shows the output active (running) that means the service is started.

Step 5: If you need to host a static website and would like to write your simple HTML Code.

get into to directory using cd /var/www/html
delete the index.html file using sudo rm -rf index.html
create your HTML page using vi index.html write the below code and save it

```
<html>
        <head><title>Nginx</title></head>
        <body>
                <h1>Hello Visitors</h1>
                <p>Welcome to our websites</p>
        </body>
</html>
```

Step 6: We must restart the nginx as we have done the configuration changes

```
sudo systemctl restart nginx
```

Step 7: to create the DNS name for this, we need to get into to below path and change hostname to awesomeweb and save and exit

su vi /etc/hosts

```
172.0.0.1       awesomeweb
```

Step 8: Restart the Nginx service because the configuration changes

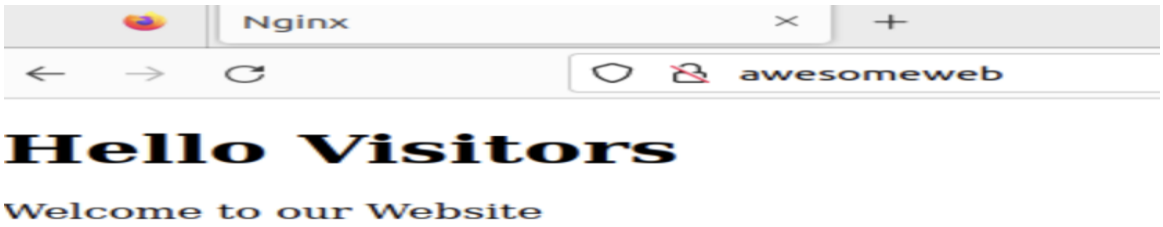
```
sudo systemctl restart nginx
```

Step 9: If the host restarts or shuts down and would like the nginx web service to Start automatically. to start up at boot.

```
sudo systemctl enable nginx
```

Step 10: Open a web browser and try to access http://awesomeweb

We can see below the output



Q2. 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. Write a detailed documentation of it.

Step 1: Write the below program subdomain.py

```
import requests
import time
from prettytable import PrettyTable

def check_subdomain_status(subdomains, base_domain):
    table = PrettyTable(['Subdomain', 'Status Code'])

    for subdomain in subdomains:
        url = f"http://{subdomain}.{base_domain}"
        try:
            response = requests.get(url)
            status_code = response.status_code
            table.add_row([subdomain, status_code])
        except requests.ConnectionError:
            table.add_row([subdomain, "Connection Error"])

    print(table)

def main():
    base_domain = "example.com"  # Change this to the base domain of your website
    subdomains = ["sub1", "sub2", "sub3"]  # Add the list of subdomains you want to
check

    while True:
        check_subdomain_status(subdomains, base_domain)
        time.sleep(60)  # Check status every minute
```

```
if __name__== "_main_":
   main()
```

Step 2: when we run the program it will show an error request and prettytable. The module is not installed, we can install both modules using the below command.

python -m pip install requests

python -m pip install -U prettytable

Step 3: run the program.

python3 subdomain.py

it will show the below output, it will show the status of the subdomain every minute.

```
+-----------+-----------------+
| Subdomain |   Status Code   |
+-----------+-----------------+
|   sub1    | Connection Error |
|   sub2    | Connection Error |
|   sub3    | Connection Error |
+-----------+-----------------+
```

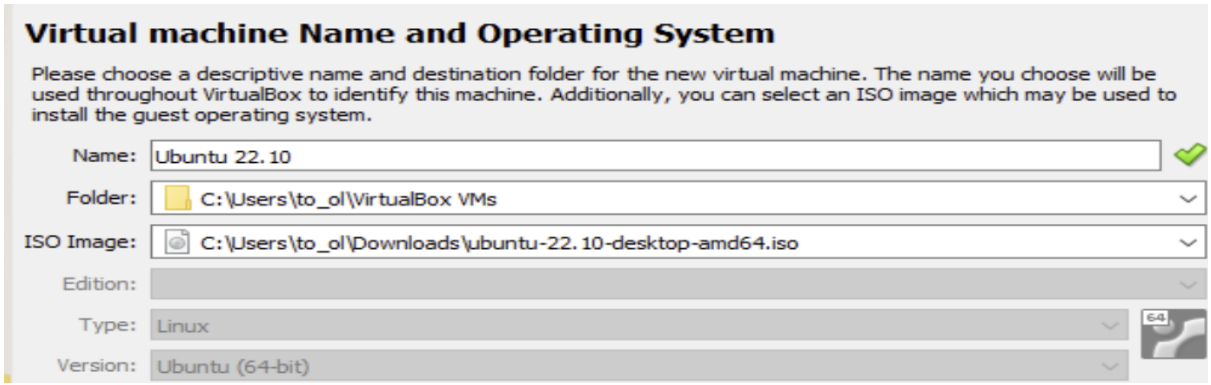
Q3. Hosting and scanning a website on a virtual box.

Step 1: Download the virtual box on Windows/Mac/Linux, below link to download. The virtual box. https://www.virtualbox.org/wiki/Download

Step 2: run the virtual box setup file from the download folders

Step 3: Once the virtual box is installed, Download the Ubuntu server using the below link https://ubuntu.com/download/server

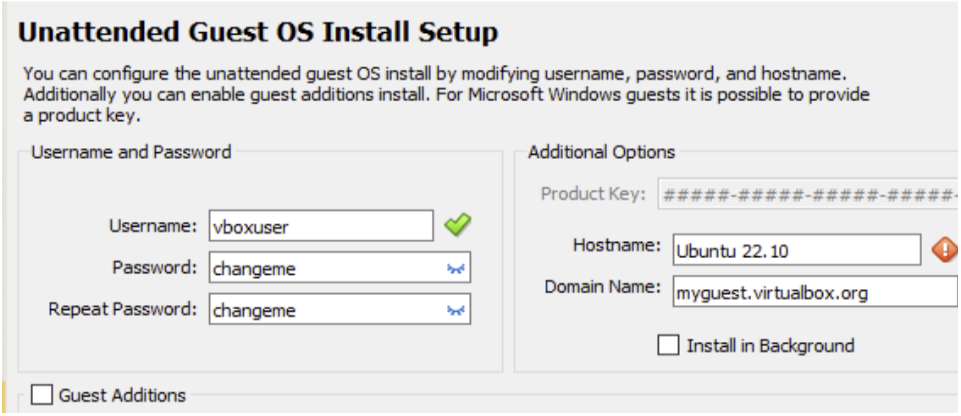
Step 4: Launch the virtual box from the start menu, and click new to create a virtual machine, create a name and select the Folder where you want to install and the location where ubuntu.iso If there



Step 5: The default user will be created.

Username-vboxuser

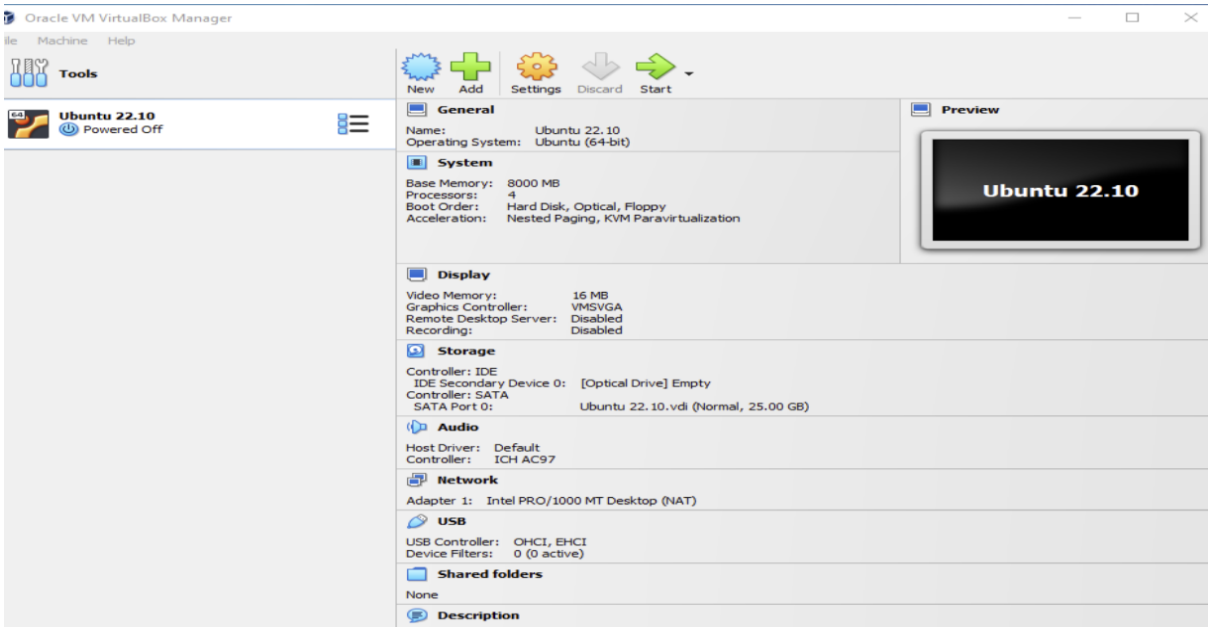
Password-changeme

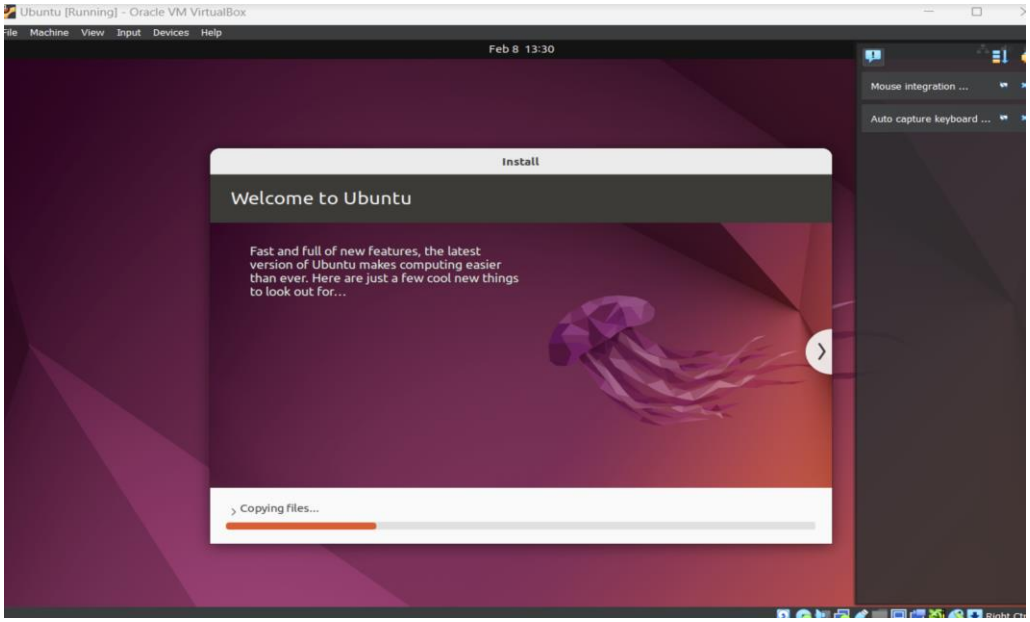


Step 6: We need to define the virtual machine resource.

RAM, Hard Disk Size and finish initialising machine

Step 7: Now we can install the image and click start to launch the virtual machine.





Once installation is completed it will show, and it will ask us to log in.

Step 8: update and upgrade the apt

sudo apt update
sudo apt upgrade -y

Step 9: Add the user account to the "vboxusers" group (Linux) or "VirtualBox Users" group (Windows) to grant permissions to manage.

Open terminal

The command will create the vboxusers group if it does not already exist

```
sudo groupadd vboxusers
```

we are going to add a user to the vboxusers group with this command

```
sudo usermod -aG vboxusers vboxusers
```

We can verify user has successfully been added to a group or not.

```
groups username
```

Step 9: Install the nginx to the virtual machine.

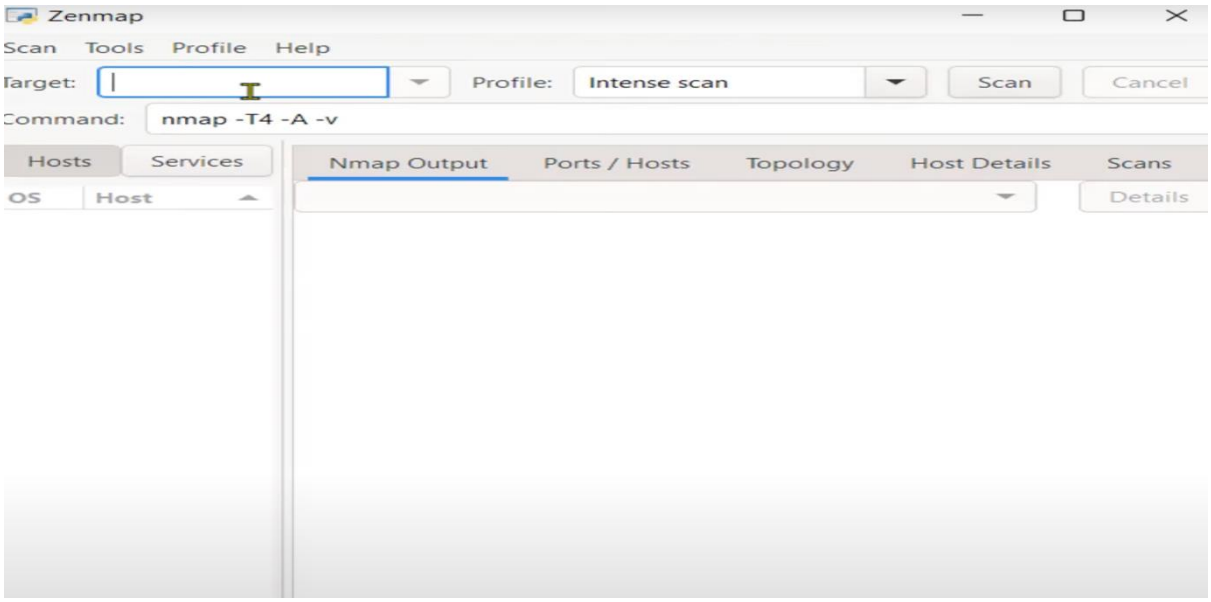
sudo apt update
sudo apt install nginx
Check the status of the Nginx
systemctl status nginx
sudo systemctl start nginx
Start automatically. to start up at boot.
sudo systemctl enable nginx

Step 10: install Nmap from https://nmap.org/ on your Windows host machine

Step 11: Check the IP of the virtual machine where nginx website is hosted

Step 12: Launch the Zenmap

enter IP: 127.0.0.1 under Target




```
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-08 15:24 India Standard Time
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 15:25
Completed NSE at 15:25, 0.00s elapsed
Initiating NSE at 15:25
Completed NSE at 15:25, 0.00s elapsed
Initiating NSE at 15:25
Completed NSE at 15:25, 0.00s elapsed
Initiating Parallel DNS resolution of 1 host. at 15:25
Completed Parallel DNS resolution of 1 host. at 15:25, 0.02s elapsed
Initiating SYN Stealth Scan at 15:25
Scanning 127.0.0.1 [1000 ports]
Discovered open port 445/tcp on 127.0.0.1
Discovered open port 135/tcp on 127.0.0.1
Discovered open port 902/tcp on 127.0.0.1
Discovered open port 2869/tcp on 127.0.0.1
Discovered open port 912/tcp on 127.0.0.1
Completed SYN Stealth Scan at 15:25, 0.08s elapsed (1000 total ports)
Initiating Service scan at 15:25
Scanning 5 services on 127.0.0.1
Completed Service scan at 15:25, 21.02s elapsed (5 services on 1 host)
Initiating OS detection (try #1) against 127.0.0.1
NSE: Script scanning 127.0.0.1.
Initiating NSE at 15:25
Completed NSE at 15:26, 64.22s elapsed
Initiating NSE at 15:26
Completed NSE at 15:26, 7.07s elapsed
Initiating NSE at 15:26
Completed NSE at 15:26, 0.00s elapsed
Nmap scan report for 127.0.0.1
Host is up (0.0039s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE            VERSION
135/tcp  open  msrpc              Microsoft Windows RPC
445/tcp  open  microsoft-ds?
902/tcp  open  ssl/vmware-auth    VMware Authentication Daemon 1.10 (Uses VNC, SOAP)
912/tcp  open  vmware-auth        VMware Authentication Daemon 1.0 (Uses VNC, SOAP)
2869/tcp open  http               Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Device type: general purpose
Running: Microsoft Windows 10
OS CPE: cpe:/o:microsoft:windows_10:1607
OS details: Microsoft Windows 10 1607
Uptime guess: 5.713 days (since Fri Feb  2 22:19:29 2024)
Network Distance: 0 hops
TCP Sequence Prediction: Difficulty=259 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2024-02-08T09:55:25
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
```