CL-2006
Operating System

LAB - 03

Network Services in Linux, Network Configuration of Static IP and Dynamic IP using DHCP in Ubuntu, Restart Network Manager, Installation and Configuration of SAMBA for File sharing

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

Spring 2023

Contents

NETWORK SERVICES IN LINUX

Services in Linux and other operating systems are applications or a set of applications that run in the background, enabling certain capabilities as and when they become necessary. This lesson takes a look at Linux services that deal with network capabilities like connecting to a network and file sharing.

The following three services are required for network activities on a Linux system:

1. network - This service turns on the network card or powers the modem.
2. iptables - This is the kernel based Packet Filtering firewall service. Various Linux firewalls are based on this service.
3. xinetd - This is the server that monitors and controls other servers. This service frees up CPU load by allowing other services like FTP (file transfer) to be available, but only run when needed, as opposed to running continuously and consuming unnecessary processing power.

Let's look at some other network services that can be used:

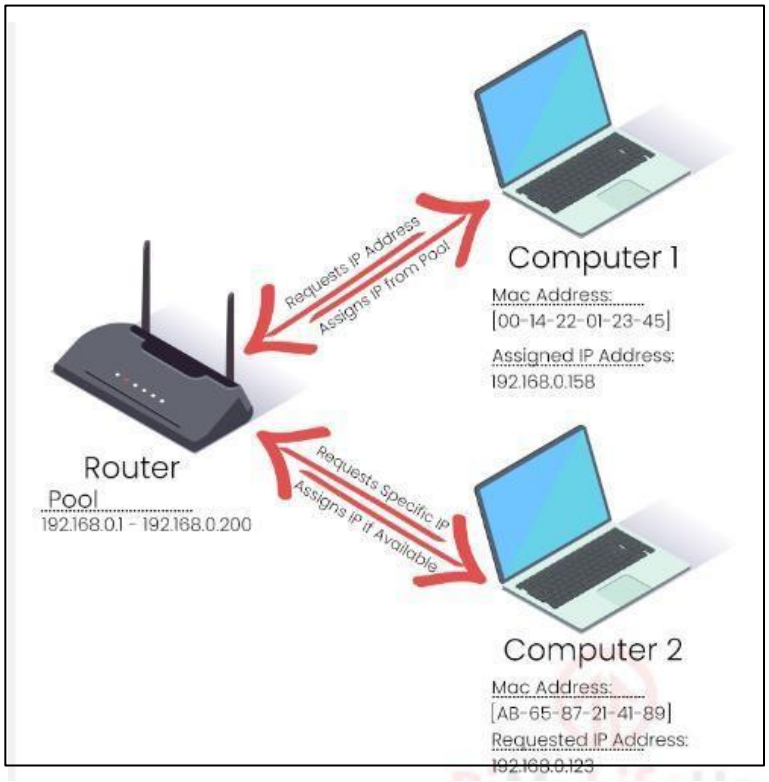
- arpwatch - This monitors and pairs remote IP addresses with hostnames.
- dhcpd - This enables the DHCP server to dynamically provide IP addresses to the local network. This is used in situations where the local network is large enough such that static IP addresses are no longer convenient.
- iplog - This is a network monitoring tool that logs TCP, UDP, and ICMP connections with hostnames of a remote host.
- netplugd - This is the Network Card Daemon, which monitors the network interface and enables it or disables it depending on whether it detects a signal. It's used mainly for laptops, as they may not always be connected to a network.
- nfs - This is the Network File Share service that's required for computers that are part of a network file sharing system.
- nfsfs - This is the Network File Share Server that powers the computer, which acts as the server in a network file sharing system.
- nfslock - This is a service that allows for NFS file locking in a network file sharing system.
- saned - This is the Network Scanner Service, which allows any machine on the network to use the scanner.
- snmpd - This is the Simple Network Management Protocol, which is used in a small network environment, like a home office network.

NETWORK CONFIGURATION

1. HOW TO SET STATIC IP IN UBUNTU SERVER

Ubuntu Server 22.04 uses netplan for default network configurations. You can adjust your Ubuntu 22.04 device's network configuration so that it will request a static IP address. By changing the interfaces configuration, the device will request a specific IP address when it connects to the router. If the router has the IP address available in its pool, it will be assigned to your 22.04 Ubuntu device.

s



a) Identifying the Network Interface Name

We can tell that any interface that starts with the text "en" is an Ethernet network interface. If you are dealing with a wireless network interface, the name would be prefixed with the text "wl". We can utilize the ip command to list the available network interfaces on your Ubuntu Server 22.04 machine.

Use the command below to list out the currently active network interfaces on your device.

ip link

By using the command, you should end up with a result as we have below. This result shows you your network interface's details, including their name.

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:0c:29:15:24:5e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
```

You can see that our Ubuntu device currently has two active network interfaces.

The first network interface that is listed is the loopback device. This interface has the name "lo" and is used for your computer to communicate with itself. The second network interface utilizes the name "enp0s3". As this interface starts with "en", we can tell that it is our Ethernet network interface. This interface is the one that we want to set the static IP address for.

Note: Before proceeding, make sure that you make a note of your network interface's name. From this example, the network name we want to set a static IP address for will be called "ens33".

Using Netplan to Define a Static IP Address

We can use this tool to make the operating system request a specific IP address from the router when it connects.

1. As your Ubuntu Server installation might come with different configuration files, we need to list the files out of the config directory. We can use the ls command to list the files in the "/etc/netplan/" directory.

ls /etc/netplan

In our clean installation of Ubuntu Server 22.04, this file was called "01-network-manager-all.yaml".

```
ayesha@ayesha-virtual-machine:~$ ls /etc/netplan
01-network-manager-all.yaml
```

2. We can begin editing the Netplan configuration file by utilizing the nano text editor. Make sure you change the filename in the following command

sudo nano /etc/netplan/ 01-network-manager-all.yaml

3. In this file, you should have text similar to what we have shown below. Using this file, we can control the behavior of our network interfaces.

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
    dhcp4: yes
```

- **network**: – When modifying a network interface's behaviour, the file must always start with "network:".
- All settings that modify the network will be stored underneath this block.
- **version**: – Netplan requires you to specify what version of its configuration markup that you are using.
- **ethernets:** – This block is where we define settings that will modify our ethernet interfaces' behavior.
- If you are trying to modify a wireless connection, you should see the text "wifi" instead.
- **ens33**: – Finally, you should see the name of your network interface. Settings defined under this block only affects that device.
- **dhcp4**: – This option allows us to control whether Netplan will automatically fetch an IP address from the router using the DHCP client.

By default, the DHCP client will be enabled with the setting set to "yes".

4. Now that you understand the default configuration, we can now modify it to suit our needs. Please note that indentation is crucial to the Netplan configuration file. Anything inside a block needs to have an extra two spaces added before it.
    a) Our first task will be to disable the network interface from retrieving an IP address automatically from the router's DHCP server. Disabling the DHCP client is as simple as changing the value of this setting from "yes" to "no".

```
dhcp4: no
```

    b) With the DHCP client disabled, we can now define a static IP address for our Ubuntu 22.04 device. We need to add a new option, that being "addresses:". Using this option, we can specify the IP address you want to retrieve. The IP addresses you specify here must be surrounded by square brackets ([ ]. If you add multiple IPs, you must separate them using a comma (,). With our example below, we will be telling our device to try and use the IP address 192.168.0.123. Remember you can get your current ip address by using the ip a command. Now set your static ip address

```
addresses: [192.168.0.123/24]
```

c) Since we have disabled the DHCP client, we need to add an option so that we can specify the gateway address. This option is “gateway4:“. Using this, we can specify the IP we want to route through. Typically, this IP address will be the internal IP of a device such as your network router. For our example, we will be using the IP address for our network router, “192.168.1.1“.

gateway4: 192.168.1.1

d) Additionally, we need to define the IP address for our nameservers. Since the “nameservers” block has several options, so everything after this block will be on a new line.

```
nameservers:
```

e) Within this block, we need to define the nameservers that we want to connect to by using the “addresses:” option. Inside the nameservers block, we need to define the DNS servers’ IP addresses that you want to connect to. This option will be started with “addresses:“. To showcase how this works, we will be using Google DNS servers. In this case, the two IP addresses would be “8.8.4.4” and “8.8.8.8“.

```
nameservers:
  addresses: [8.8.8.8,8.8.4.4]
```

5. At the end of these steps, you should have a configuration file that looks like we have below. The IP addresses and network interface names will likely differ from what we have here.

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: no
      addresses: [192.168.0.123/24]
      gateway4: 192.168.1.1
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]
```

You can save your changes to the configuration file by pressing CTRL + X, followed by Y, then the ENTER key.

6. Now to check whether your configuration has been accepted or not, type the below commands

sudo netplan try

which will show you the below output

```
Changes will revert in 111 seconds
Configuration accepted.
```

7. With the configuration file modified, we need to get the Netplan tool to apply the changes. Applying the changes is as simple as using the following command on your Ubuntu device.

sudo netplan apply

8. Verifying that the Ubuntu 22.04 operating system is now utilizing a static IP address, we can use the ip command again.

ip a

```
ayesha@ayesha-virtual-machine:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:15:24:5e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.123/24 brd 192.168.0.255 scope global noprefixroute ens33
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe15:245e/64 scope link
       valid_lft forever preferred_lft forever
```

From this command, you should end up finding a result for your Ethernet device. As you can see, our ip address changed to 196.162.0.128 as we added in the configuration file above.

DISADVANTAGES OF STATIC IP

Static IP address has a few drawbacks:

- The router will never guarantee that the IP is actually available. As there is no reservation in place, the IP can be assigned to another device.
- Additionally, suppose the static IP address is not available when your Ubuntu 22.04 device connects. In that case, it will fail to connect to the router.
- As your device asks for a specific IP, the router will not fall back to assigning an IP from the router's pool.

b) HOW TO SET DYNAMIC IP IN UBUNTU SERVER

In the following procedure, we will see how to configure the interface to receive a dynamic IP address from DHCP.

Edit the netplan configuration file using any text editor. Here we are using the Nano text editor for this purpose.

sudo nano /etc/netplan/01-network-manager-all.yaml

Then add the following lines by replacing the interface name with your system's network interface.

```
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    ens33:
      dhcp4: yes
      addresses: []
```

Once done, save and close the file.

Now test the new configuration using the following command:

sudo netplan try

If it validates the configuration, you will receive the configuration accepted message, otherwise, it rolls back to the previous configuration.

```
Do you want to keep these settings?


Press ENTER before the timeout to accept the new configuration


Changes will revert in 118 seconds
Configuration accepted.
```

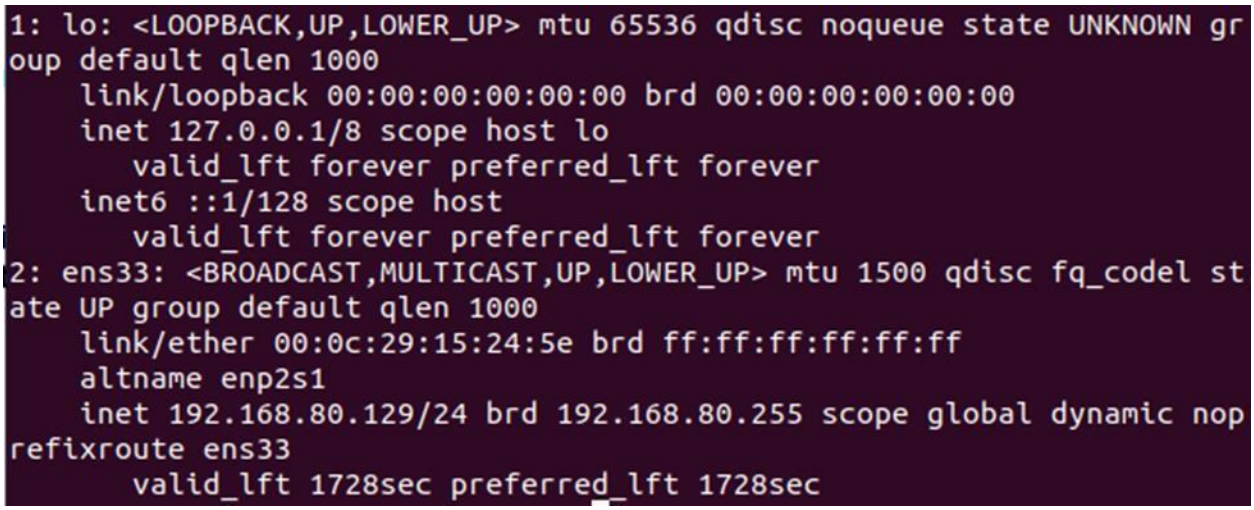
Next, run the following command to apply the new configurations.

sudo netplan apply

After this, check the IP address of your machine using the following command:

ip a

As you can see, your ip, that you previously set using static method to 192.168.0.123 as changed to 192.168.80.129



RESTART NETWORK MANAGER SERVICE

There are various situations where you may have to restart the network on Ubuntu. It may be because the network settings were changed. It may be because the network connection is acting weird. Generally, whenever there's a problem with the system, a common treatment is performing a reboot. However, if it's a problem related to the network, then it's possible to just restart the network.

This is one of the easiest ways of restarting the network service. In this example we have used systemd to restart the network. You can use other tools like nmcli, ifup, nmtui, etc.

Systemd offers an array of system components to the system. Part of it is handling the services. Use the below command to restart your network manager

sudo systemctl restart NetworkManager.service

INSTALLATION AND CONFIGURATION OF SAMBA SERVER ON UBUNTU 22.04 LTS

INTRODUCTION TO SAMBA

Samba is an open source implementation of the Server Message Block (SMB) protocol. It allows the networking of Microsoft Windows®, Linux, UNIX, and other operating systems together, enabling access to Windows-based file and printer shares. Samba's use of SMB allows it to appear as a Windows server to Windows clients.

Some examples of common services offered by Samba are:

- Share one or more directory trees
- Provide a Distributed Filesystem (MS-DFS) namespace

- Centrally manage printers, print settings, and their associated drivers for access from Windows clients
- Assist clients with network browsing
- Authenticate clients logging onto a Windows domain
- Provide or assist with Windows Internet Name Service (WINS) name-server resolution

SAMBA FEATURES

Samba is a powerful and versatile server application. Even seasoned system administrators must know its abilities and limitations before attempting installation and configuration.

What Samba can do:

- Serve directory trees and printers to Linux, UNIX, and Windows clients
- Assist in network browsing (with or without NetBIOS)
- Authenticate Windows domain logins
- Provide Windows Internet Name Service (WINS) name server resolution
- Act as a Windows NT®-style Primary Domain Controller (PDC)
- Act as a Backup Domain Controller (BDC) for a Samba-based PDC
- Act as an Active Directory domain member server
- Join a Windows NT/2000/2003 PDC

INTSALLATION & CONFIGURATION OF SAMBA

1. To install SAMAB on your Ubuntu use the following command

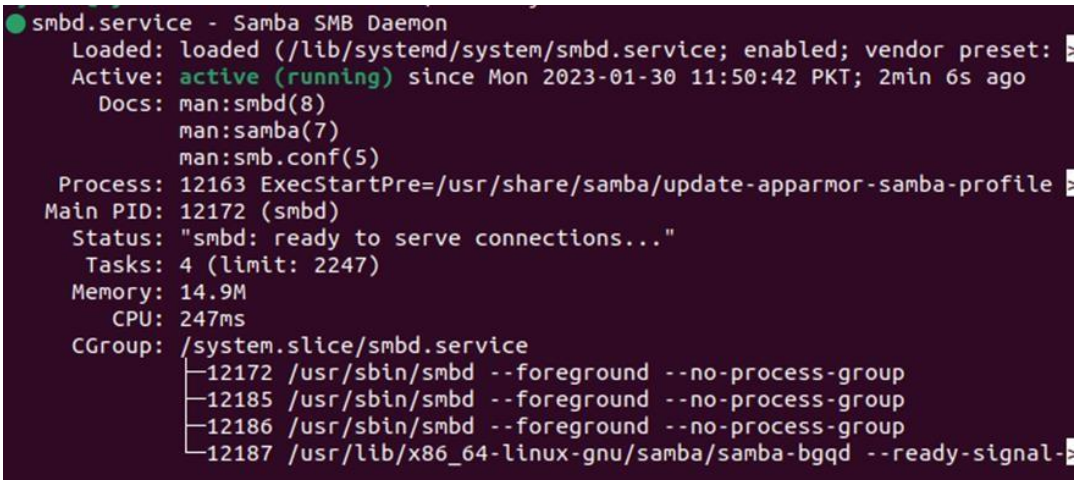
sudo apt-get install samba

2. After installing check SAMBA stsus using the following command

sudo systemctl enable –now smbd

sudo systemctl status smbd

This will show that your SAMBA server is active/running.



3. Now create a directory that will be shared across platforms

mkdir /home/ayesha/share

(Note: Please use your username file creating a directory. My username was Ayesha)

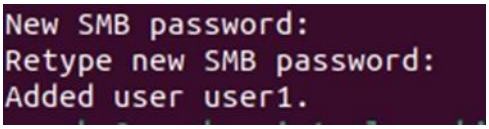
4. Now give permissions to the above directory named sharingFolder

chmod 777 /home/ayesha/share

5. Now set SAMBA user password

sudo smbpasswd –a ayesha

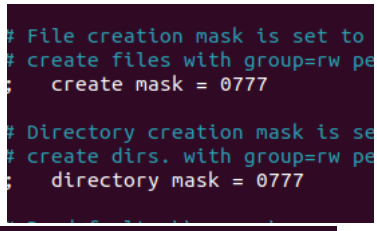
After typing the above command, you will be shown the below image.

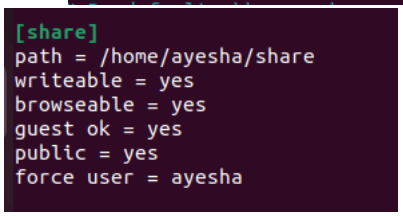


6. Now go to SAMBA configuration file to enable sharing of the directory sharingFolder

sudo nano /etc/samba/smb.conf

After opening the file. Go to the bottom of the file using ctrl+V and type the below commands, make directory and create masks as: **0777**





- Path – set the path where your directory is located
- Valid users – Which users can access this file
- Read list – Who can access the directory with only read access
- Write list – Who can access the directory with write only access
- Browseable – Is it browseable or not

Now Press CTRL+C, then press y and then press enter to save, Now run the following commands.

```
ayesha@ayesha-virtual-machine:~/share$ sudo smbcontrol smbd reload-config
ayesha@ayesha-virtual-machine:~/share$ sudo service smbd restart
ayesha@ayesha-virtual-machine:~/share$ sudo ufw allow samba
Skipping adding existing rule
Skipping adding existing rule (v6)
ayesha@ayesha-virtual-machine:~/share$ S
```

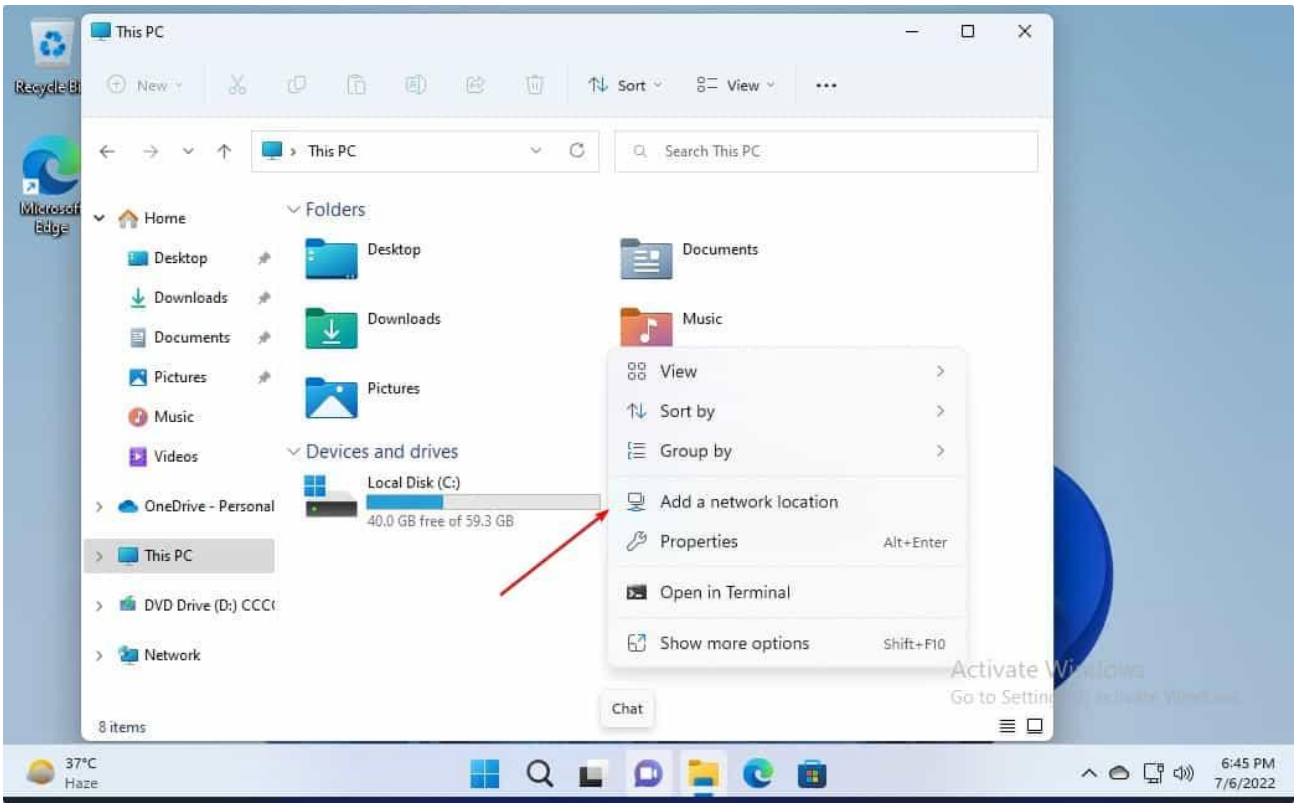
7. Now to check the sharing of your folder between Linux and Windows OS. Press Window + R to open run terminal and insert the ip address of your linux. To get the ip address use the following command

ip a

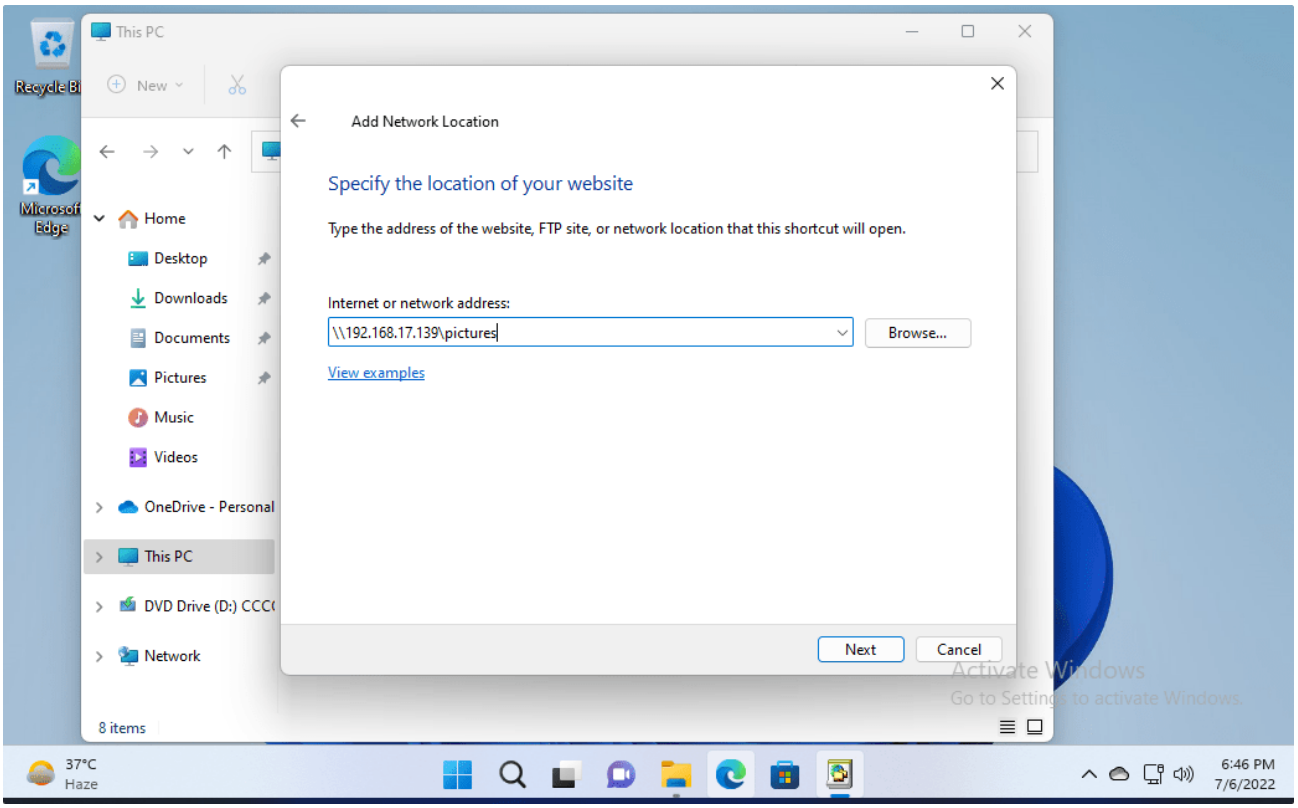
You will see the following lines in which your ip address will be written in the circled part. In my case it is 192.168.141.128 so I will be writing this ip.

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:1f:a2:71 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.141.128/24 brd 192.168.141.255 scope global dynamic noprefixrou
te ens33
       valid_lft 1157sec preferred_lft 1157sec
    inet6 fe80::73b2:bd6b:1672:e50f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Remember your ip:

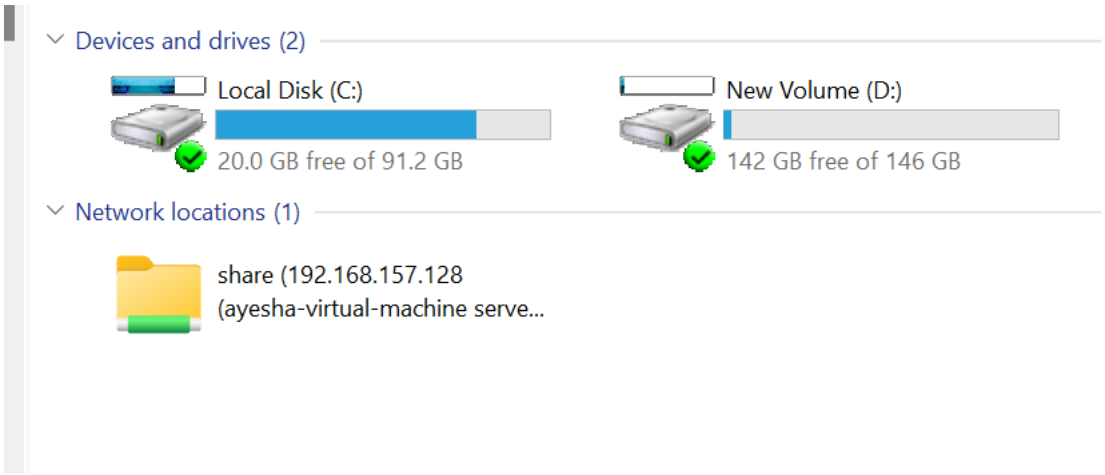


Add your ubuntu ip address and your shared folder in this format:



As you will press ok you will see that your directory **share** which you created in Linux OS will be accessible in Windows as well.

When you will access the folder. You will be asked to add credentials. Now remember, add the credentials of the Samba user and Samba password you to open your folder.



How to install virtual box 7.0 in Ubuntu 22.04

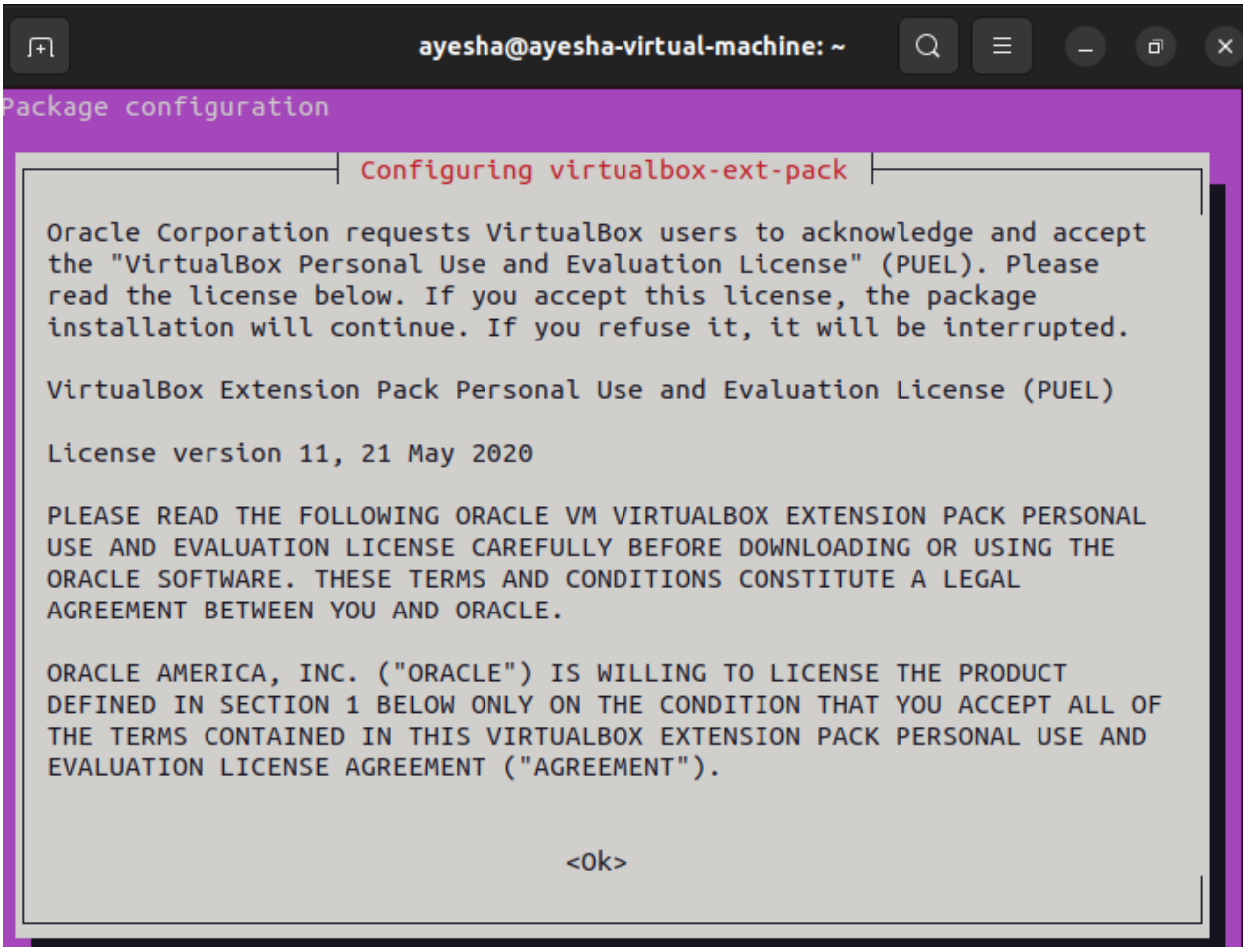
1. Update Ubuntu packages

   sudo apt update –y

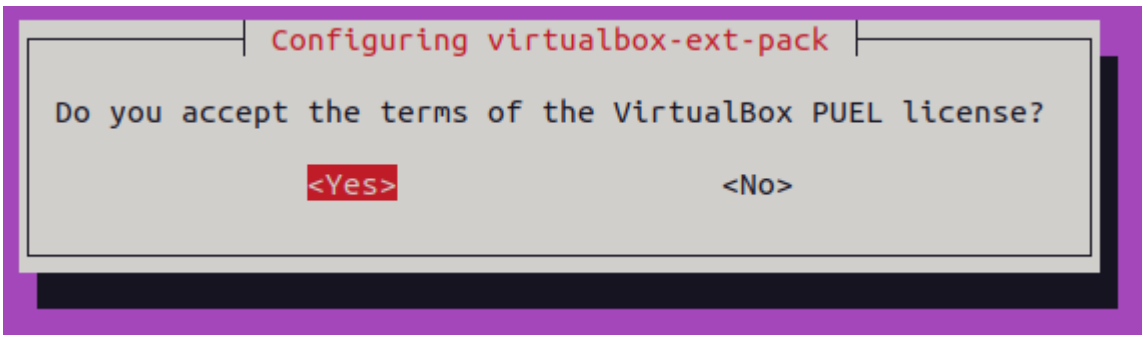
2. Install VMA

   sudo apt install virtualbox virtualbox-ext-pack -y

3. Select OK, and press Enter after reading the VirtualBox Extension Pack Personal Use and Evaluation License (PUEL) terms.



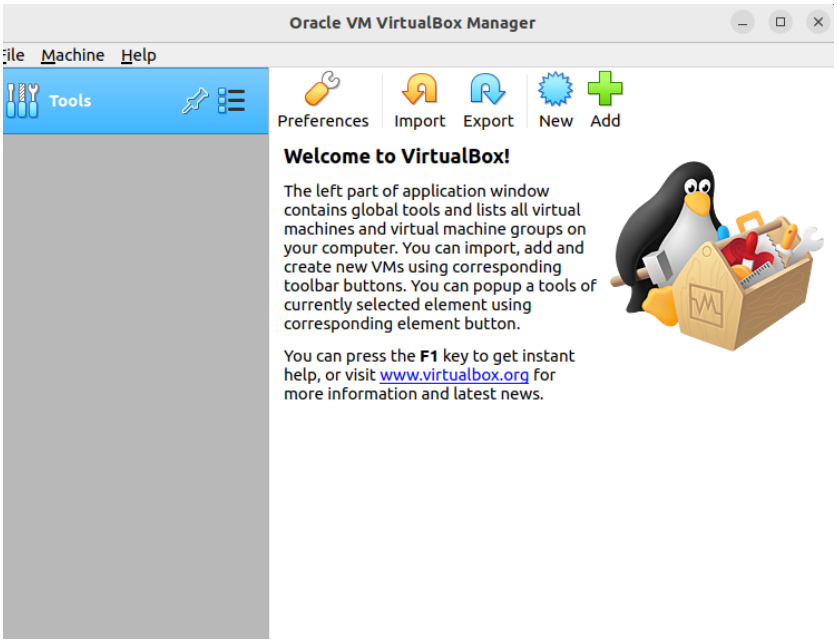
4. Now, select Yes, and press Enter to accept the Oracle Binary Code License Agreement for the VirtualBox Extension Pack.



5. Finally, run the following command to start using VirtualBox or open it from the Applications menu.

virtualbox

The main VirtualBox Manager window appears, as shown below, where you can manage your VMs.

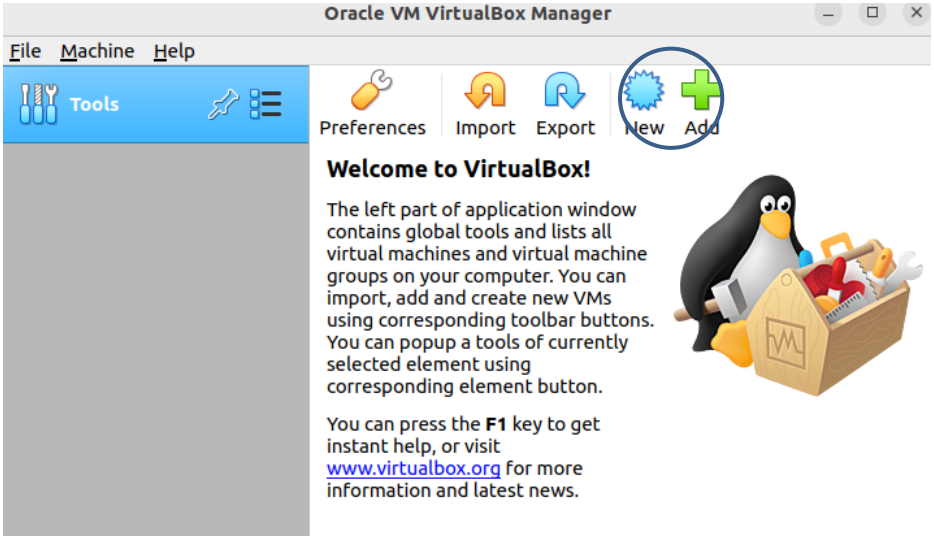


Creating Your First VM

You've successfully installed VirtualBox on your machine and are ready to create your first VM. This tutorial uses a Windows 10 image to create a VM, but you can choose any you prefer.

To create your first VM:

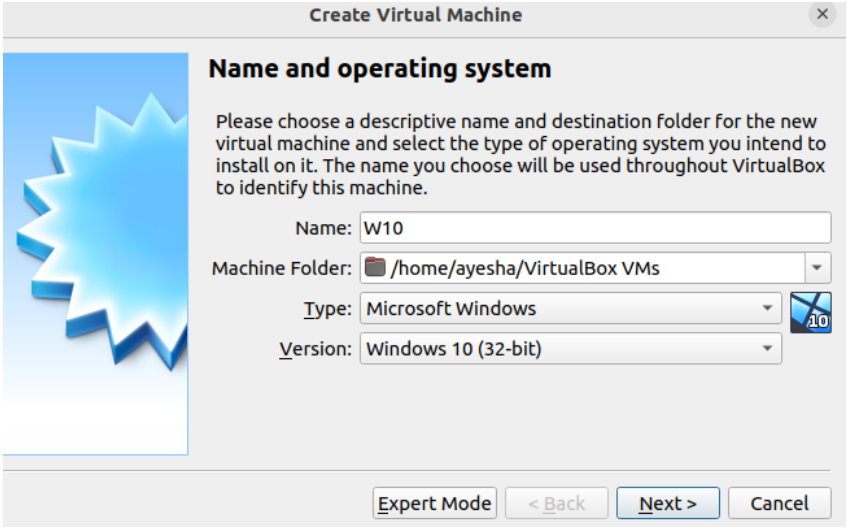
1. Click the New button on the toolbar to create a new VM.



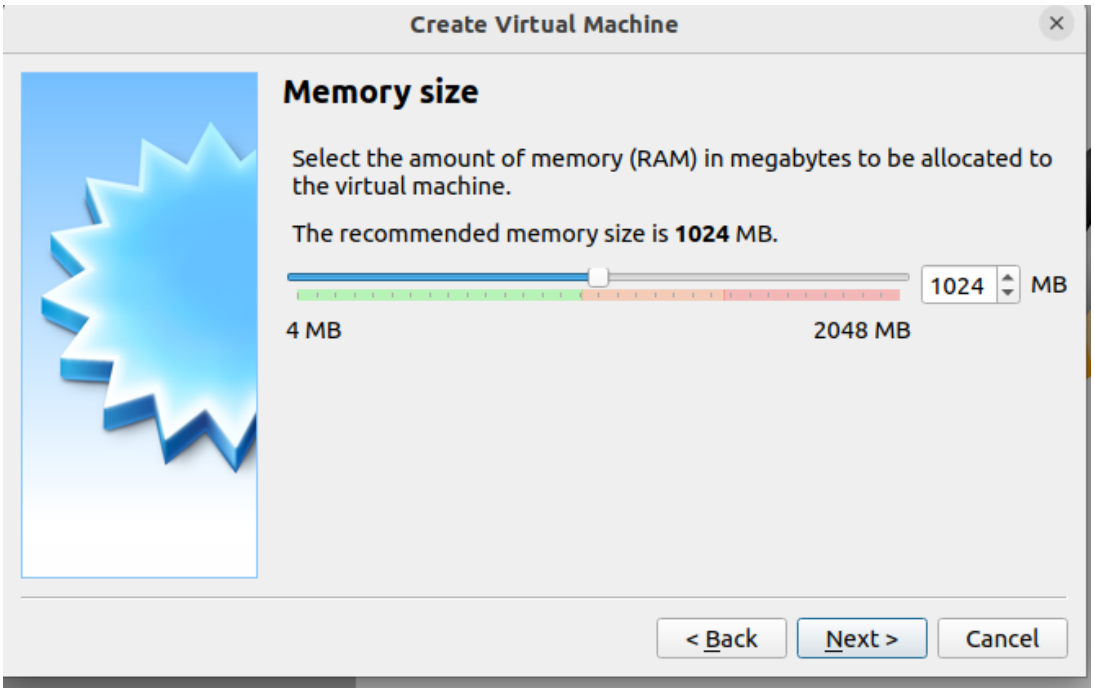
2. Configure the name and operating system (OS) for your VM with the following:

    - Name – Provide a name for your VM. This tutorial's choice is W10.
    - Machine Folder – Choose the folder where you want your VM to reside.
    - Type – Select the type of OS for your VM. This example will use Windows 10 (32-bit).

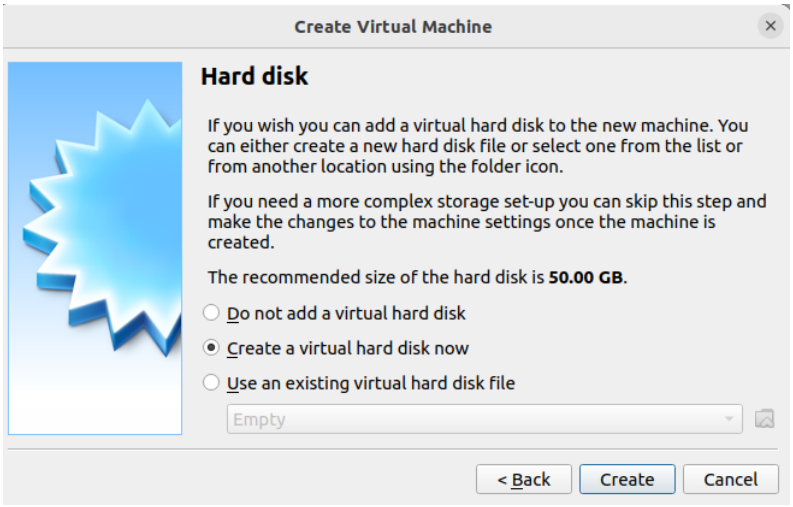
Click Next to continue.



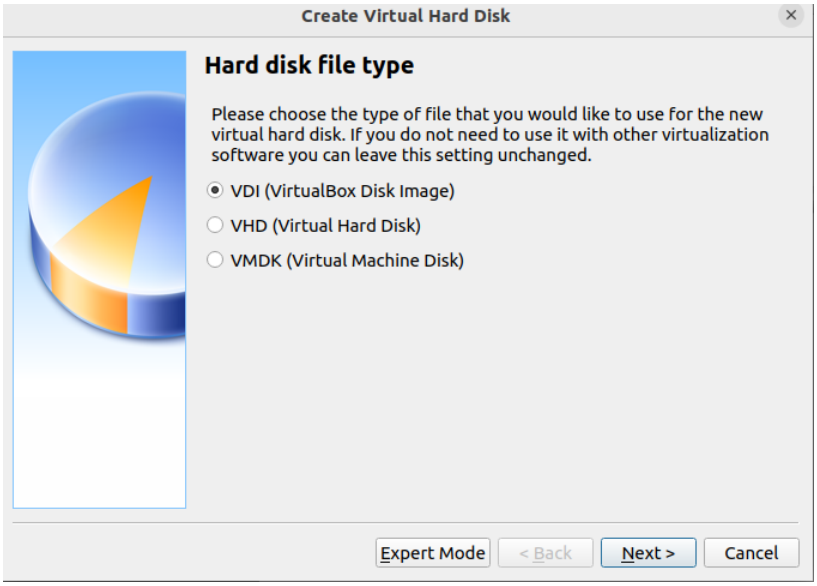
3. Now, specify how much memory (RAM in MB) you want to allocate for your VM, and click Next.



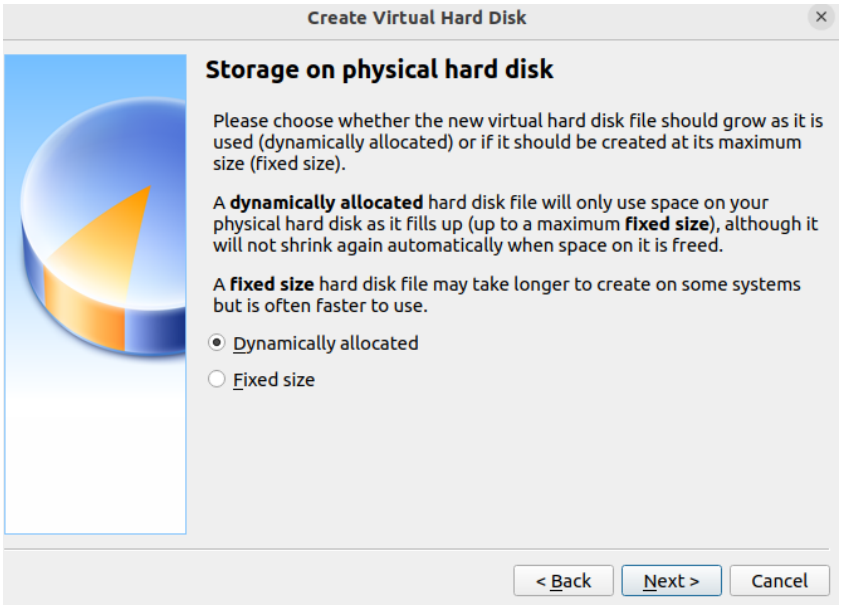
4. Select the Create a virtual hard disk now option on the next screen, and click Create. This option lets you create a virtual hard disk (a file) that stores all the data for your VM, including the operating system, applications, and files.



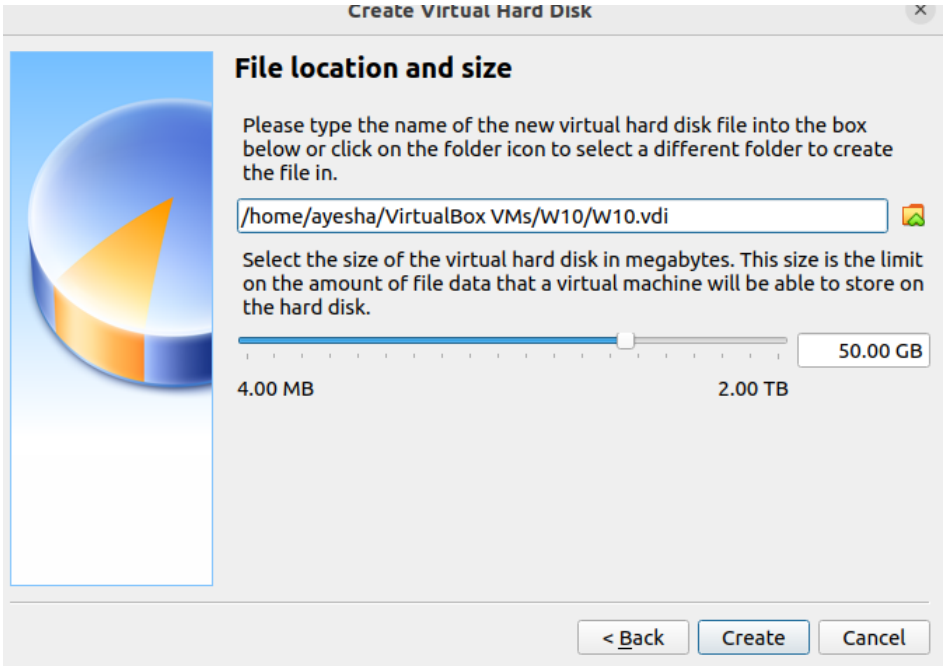
5. Next, choose the VDI (VirtualBox Disk Image) option, which is a good choice for most users, and click Next.



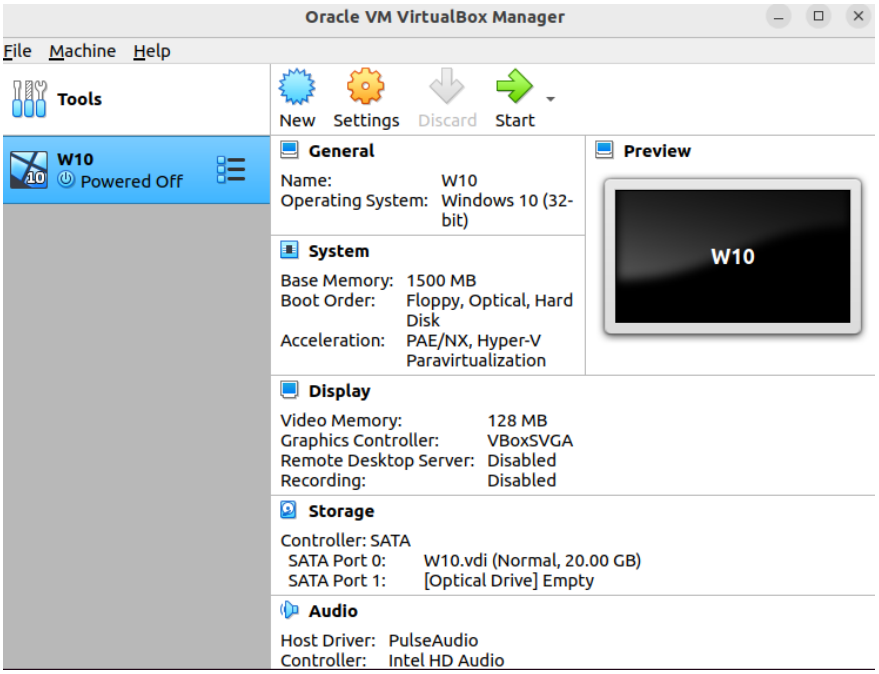
6. On the next screen, choose the storage type for your virtual hard disk. But for this tutorial, select the default option (Dynamically allocated) and click Next. Why use dynamic allocation? This option is more efficient with storage space because it only allocates the amount of disk space the VM uses.



7. Choose a name for the virtual hard disk, allocate the storage space for your VM, and click Create. You can choose the storage size you prefer, but this example's choice is 50 GB.



8. Once the VM is created, you'll see the VM listed in the left pane of the VirtualBox window with all details.

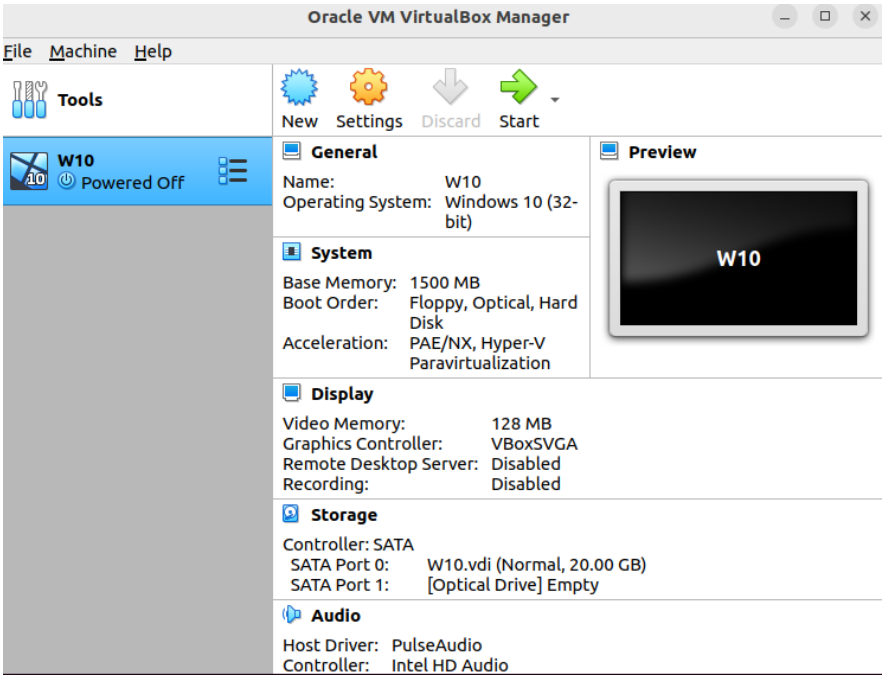


Attaching a Bootable Media

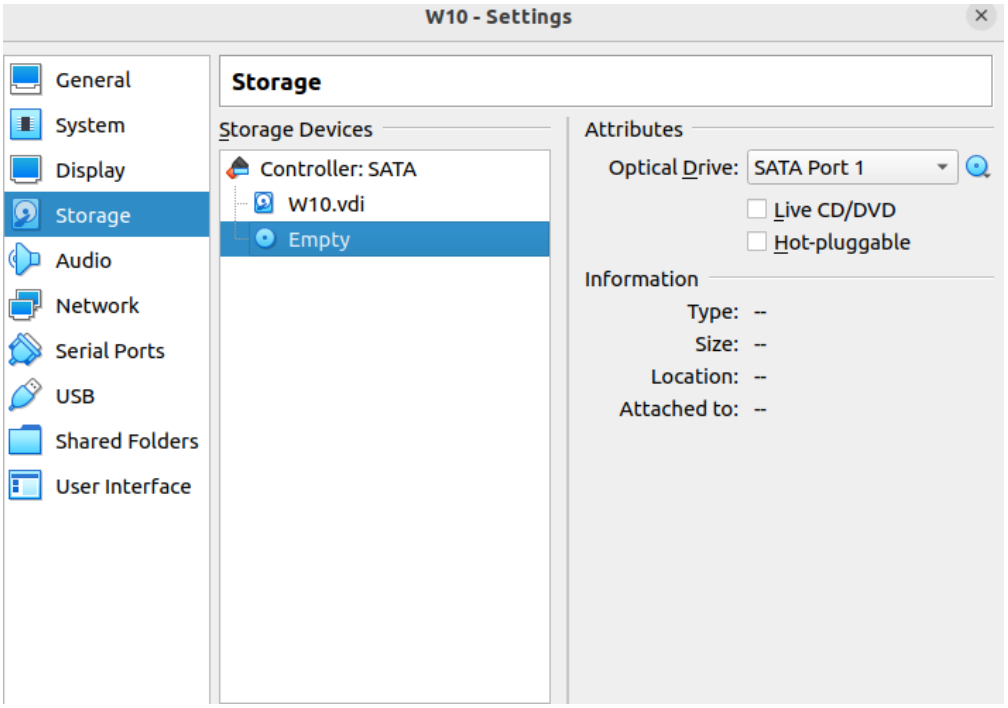
You've just created your first VM on VirtualBox. But right now, even if you start the VM, it won't do anything since you haven't attached any bootable media to the VM.

To attach a bootable media to your VM:

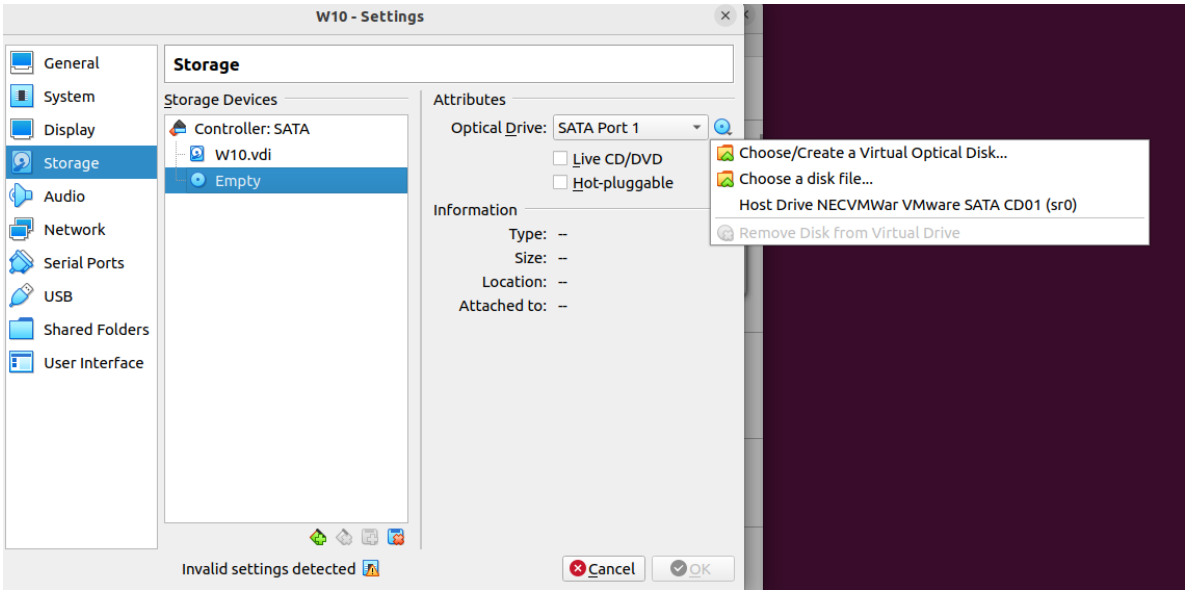
1. Click on your VM in the left pane, and click Settings from the toolbar to access your VM's settings.



2. On the Settings window, click Storage in the left pane —> Empty drive under Storage Devices —> the disc icon under Attributes. A context menu opens where you can choose how to attach a bootable media for the VM.

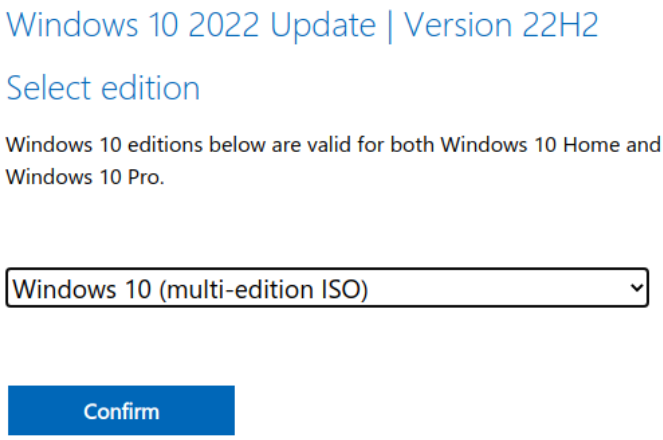


3. Select Choose disk file from the dropdown menu to look up your bootable media (ISO).

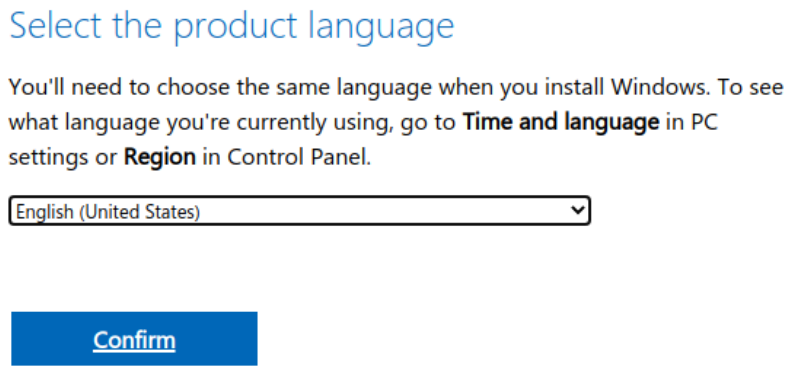


4. We will use Win10 32bit disk file. Download the Win10 disk file using the following steps.

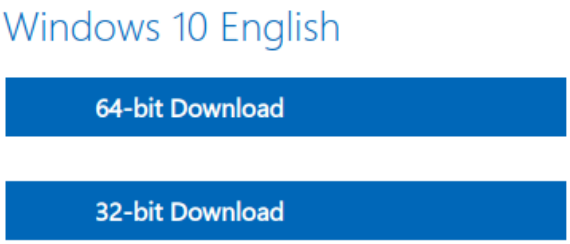
   a) Visit the following site to download W10 ISO file.
   https://www.microsoft.com/en-us/software-download/windows10ISO
   b) Select the Windows10 option and click confirm.



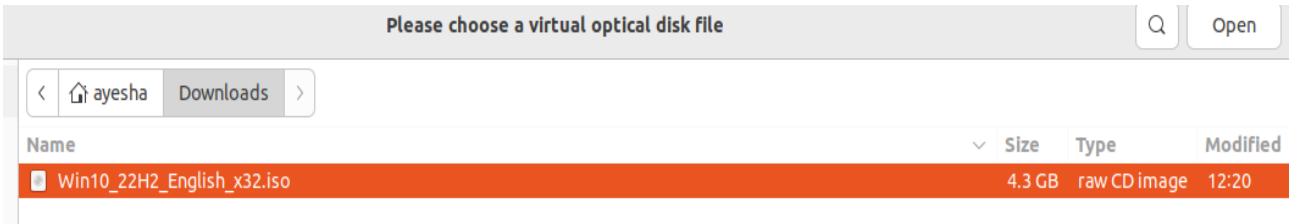
c) Select the language for your windows and click confirm.



d) Select the 32bit as when creating our Virtual Machine, we selected Windows 32 bit.



Now, locate and select your ISO image file.



5. Click OK to close the Settings window.
6. Now install the Windows 10 like you normally do.

sssssssss