This document will provide step by steps instructions on how to install and configure Ubuntu Desktop 20.04, Microsoft Windows 10, Samba and also show you the steps on how to set up static IP addresses, creating a virtual network, adding a second Network adaptor, creating new users on Ubuntu, creating folder and text files and showing how to access and share/store files over the network. If you want to enable SSH on Ubuntu please refer to Milestone 1 documentation.

Host Computer: MAC OS

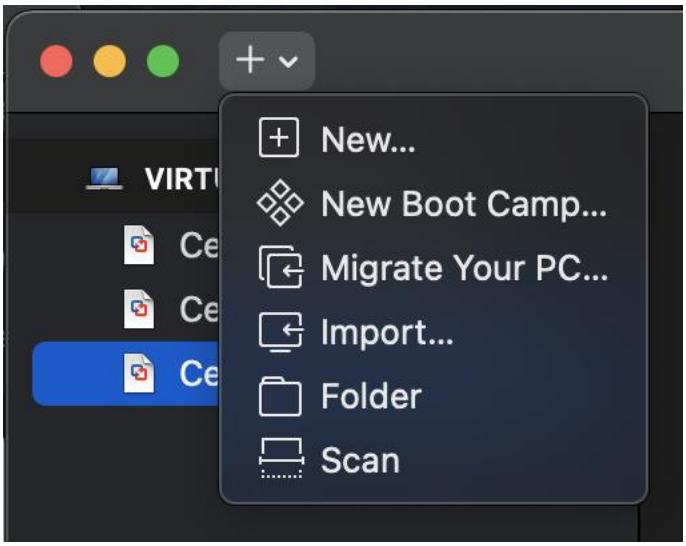
Prerequisites before starting this process:

1. Must have a hypervisor installed in order to install a Virtual Machine. (Ex. VMware Fusion)
2. Ubuntu Desktop ISO file which can be found at this website.
3. A Microsoft Windows ISO file which can be found on their website as well or you can use Microsoft Azure.

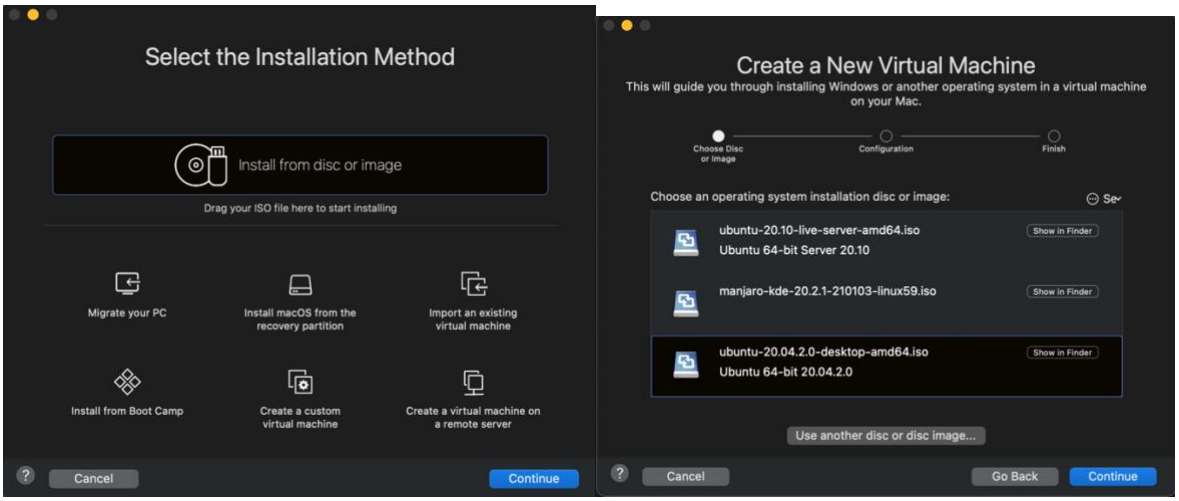
*An ISO file is a file that contains an image of data found on an optical disc, like a CD or DVD.

Creating an Ubuntu Desktop 20.04 VM (VMware Fusion v12 is being used).

1. Create a new Virtual Machine.



2. Once your Ubuntu Desktop ISO file is downloaded you can drag and drop the file into the installation box or click continue to search for that downloaded file.



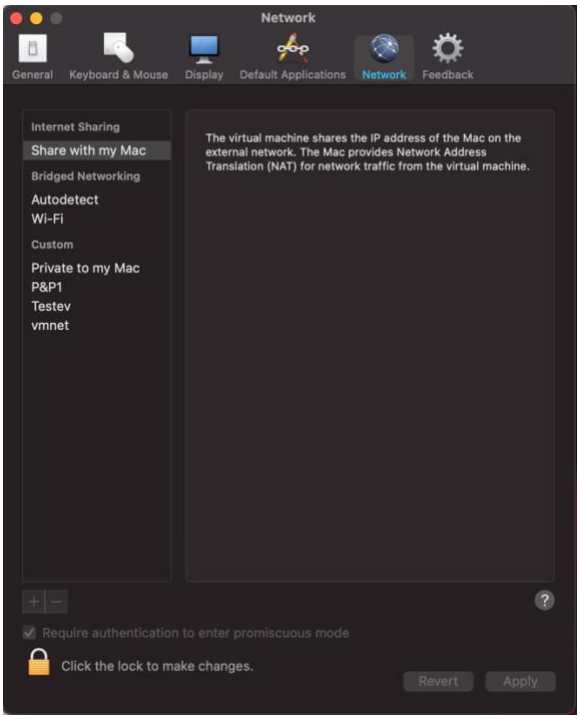
3. Once that file is selected click on continue and now you should be at the Linux Easy Install screen. You will want to set up a password for the admin. Click Continue.
4. Next, should be the finish screen and to rename your Virtual Machine click on customized settings and name your VM UbuntuFileServer. Click Finish. All DONE!

Creating a Virtual Network

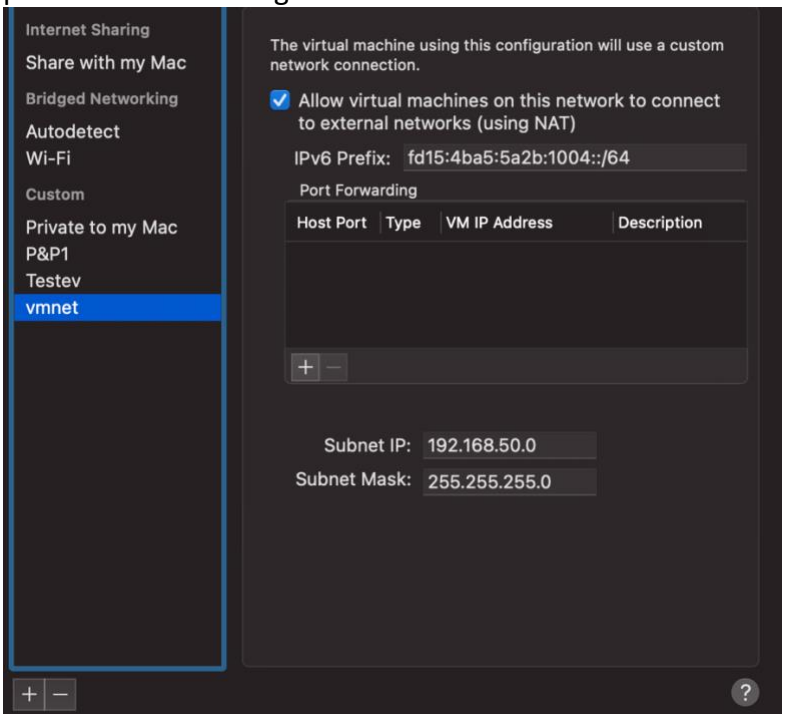
Before we start installing Ubuntu Desktop, we will create a Virtual Network and a second Network Adapter.

1. While you are on the setting of your new VM at the top left in the menu bar click on VMware Fusion -> Preferences.

2. Click on Network at the top of the General dialogue box. Click on the lock button and enter your password or use Touch-ID. *The password is the one for your host machine not the Ubuntu password you created earlier.



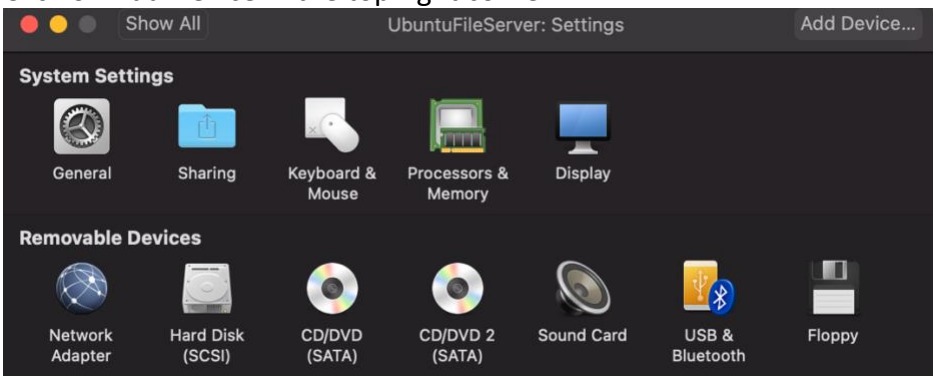
3. Click on the plus sign that is right about the lock symbol and a new virtual network will appear. You can rename by click on the name once. Check 'Allow virtual machines on this network to connect to external networks (using NAT) and in the Subnet IP box enter in 192.168.50.0 and your Subnet Mask should be 255.255.255.0. Click lock symbol to prevent further changes. **ALL DONE!**



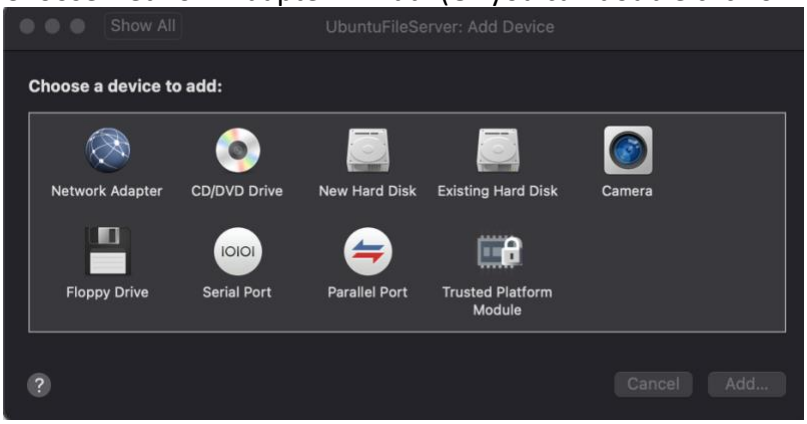
Creating a second Network Adapter

Back to the settings for your **UbuntuFileServer.**

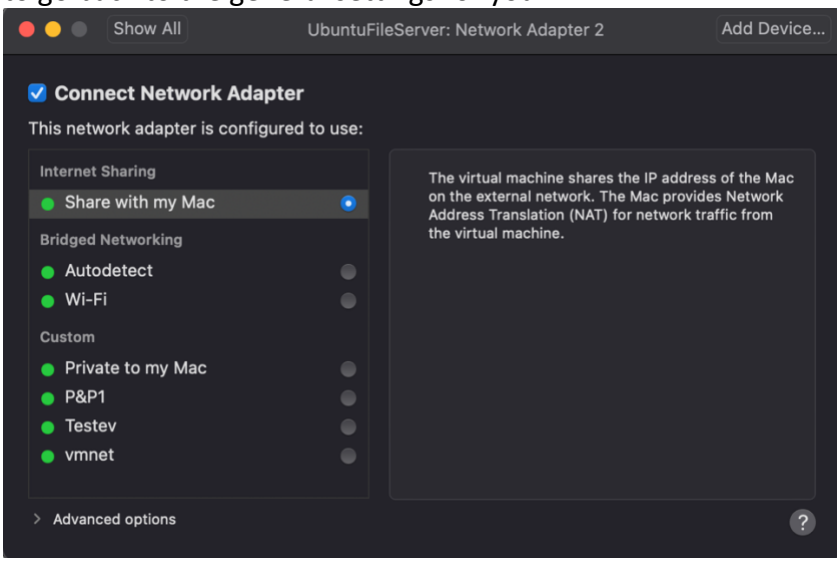
1. Click on Add Device in the top right corner.



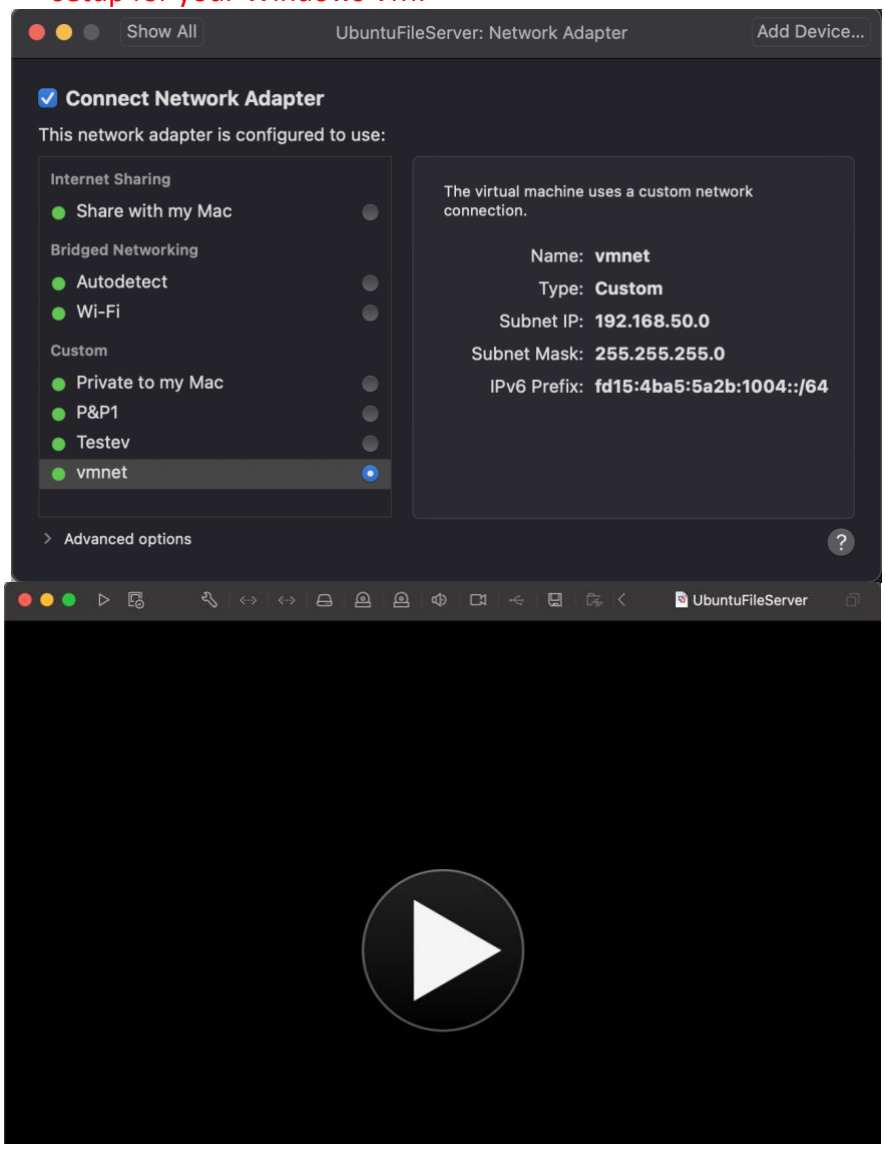
2. Choose Network Adapter -> Add. (Or you can double click on Network Adapter)



3. It should automatically go into your second network adapter settings and should be on **Share with my Mac** (No changes needed for this adapter). click **Show All** at the top left to go back to the general settings for your VM.



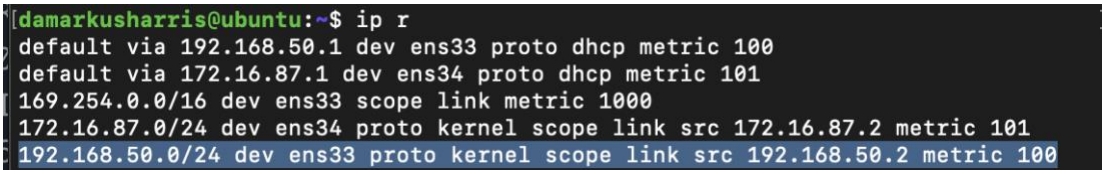
4. Click on your first Network Adapter and configure it to use the new Virtual Network you created earlier. In the screenshot below mines is labeled **vmnet.** Just click on the circle next to yours and it will highlight blue like in the screenshot. You can now press the Play button on your Ubuntu VM to start the installation. **ALL DONE!** *This will be the same setup for your Windows VM.



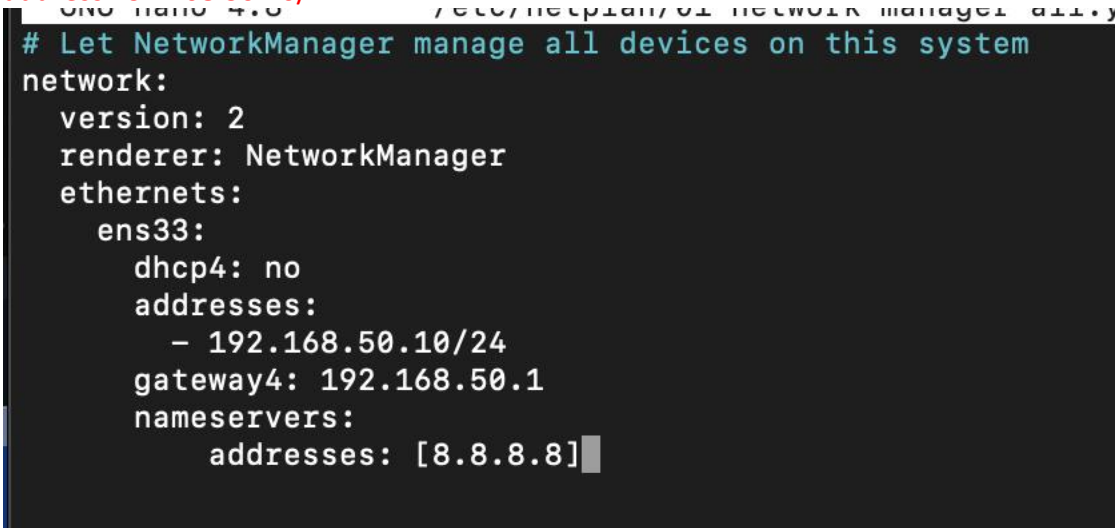
Setting up Static IP address

We will be using Netplan since it is the default network management tool on Ubuntu. Netplan uses configuration files in YAML syntax. Netplan currently supports two renderers NetworkManager and Systemd-networkd. NetworkManager is mostly used on Desktop machines while Systemd-networkd is used on servers without a GUI. I will show how to use the Command line.

1. Enter command: **ip r** *Take note of your network interface name mines in **ens33** and take note of your **dhcp address** as shown in the screenshot below. Notice there are a few that show. Knowing which one to use is by looking at the network id that shows first.



2. Let's navigate to our Netplan directory by using command: **cd /etc/netplan**
3. Open the .yaml file by using command: **sudo nano /etc/netplan/01-netcfg.yaml**
4. We will edit the .yaml file to look similar to the screenshot below. *Make sure the appropriate changes are made to reflect the name of your interface and use the static ip address 192.168.50.10/24.



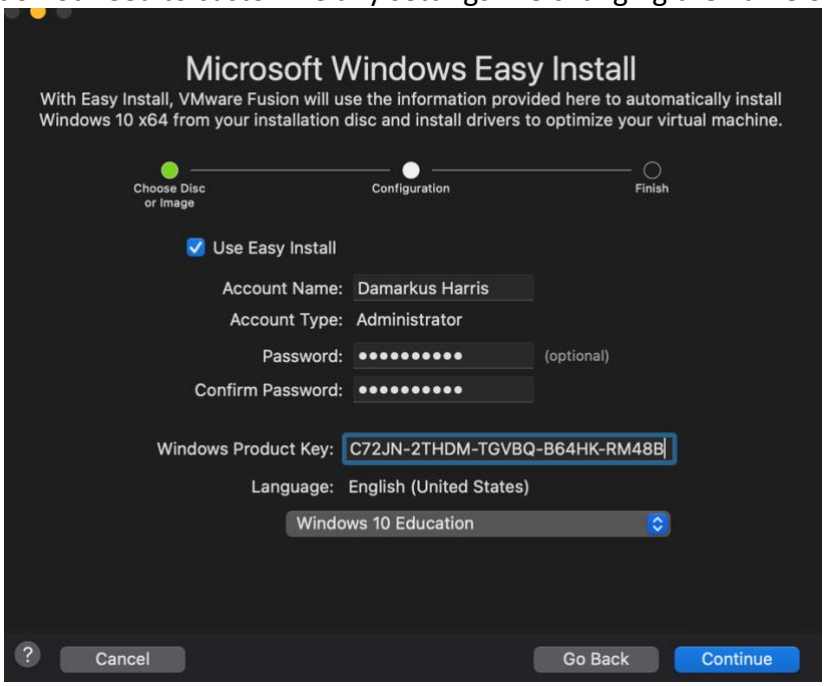
5. Once done use **^X** to exit, **Y** to save and click **Enter** to save (Make no changes to the file name)
6. Once done we need to apply the changes and verify your ip address by using the commands in the following screenshot. **ALL DONE!**

```
$ sudo netplan apply
```

```
$ ip addr show dev ens3
```

Creating a Windows VM

The same steps can be followed using the Creating an **Ubuntu Desktop 20.04 VM** section. The only difference is you will need to enter your Windows Product Key and Choose your Windows version. Click continue -> More Isolated on the Integration Screen. Click Finish if you do not need to customize any settings like changing the name of the VM.



Install and Configure Samba

Samba is a re-implementation of the SMB networking protocol. Samba provides file and print services for various Microsoft Windows clients. If you have not ran the command: **sudo apt-get update** then go ahead and execute that command to update the system. Here is the website for to install Samba. Let's start the process to install and configure Samba!

1. To install Samba use command: **sudo apt-get install samba**

2. Next, we will need to set up a password for the samba user. Use command: **sudo smbpasswd -a <user_name>** replace **<user_name>** with your actual username.

```
damarkusharris@ubuntu:~$ sudo smbpasswd -a damarkusharris
New SMB password:
Retype new SMB password:
Added user damarkusharris.
```

3. Here is a List of helpful tips from the help.ubuntu.com website.

```
Note: Samba uses a separate set of passwords than the standard Linux system accounts (stored in
/etc/samba/smbpasswd), so you'll need to create a Samba password for yourself. This tutorial
implies that you will use your own user and it does not cover situations involving other users
passwords, groups, etc...
```

```
Tip1: Use the password for your own user to facilitate.
```

```
Tip2: Remember that your user must have permission to write and edit the folder you want to
share.
Eg.:
sudo chown <user_name> /var/opt/blah/blahblah
sudo chown :<user_name> /var/opt/blah/blahblah
```

```
Tip3: If you're using another user than your own, it needs to exist in your system beforehand,
you can create it without a shell access using the following command :
sudo useradd USERNAME --shell /bin/false

You can also hide the user on the login screen by adjusting lightdm's configuration, in
/etc/lightdm/users.conf add the newly created user to the line :
hidden-users=
```

4. Create a directory to be shared using command in screenshot below.

```
mkdir /home/<user_name>/<folder_name>
```

5. Make a safe backup copy of the original smb.conf file to your home folder.

```
sudo cp /etc/samba/smb.conf ~
```

6. Edit the file “/etc/samba/smb.conf”

```
sudo nano /etc/samba/smb.conf
```

```
Once "smb.conf" has loaded, add this to the very end of the file:

[<folder_name>]
path = /home/<user_name>/<folder_name>
valid users = <user_name>
read only = no
```

Here how's mines look. Notice that we keep the brackets that samba is wrapped in and also one space before and after the = signs.

```
[samba]
path = /home/damarkusharris/samba
valid users = damarkusharris
read only = no
```

7. Restart the samba using command:

   ```
   sudo service smbd restart
   ```

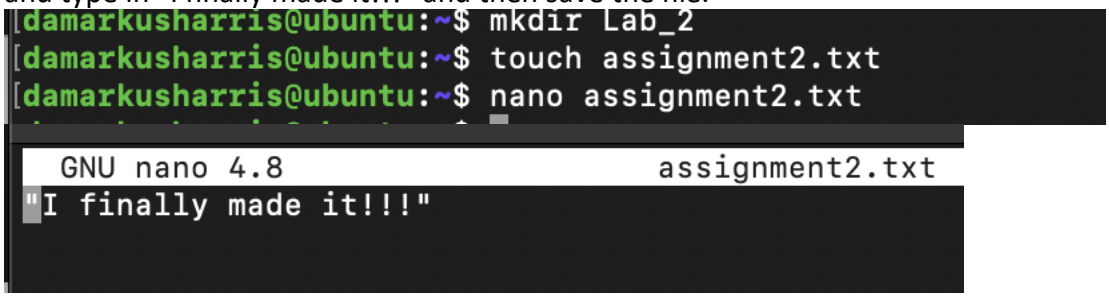
8. To check your smb.conf file for any syntax errors use this command:

   ```
   testparm
   ```

Creating a new user on Ubuntu

1. To add a new user use command: `sudo adduser userNameHere`
   *Replace userNameHere with the name of the user -> Follow the instructions that follow to you finally state the information is correct. The only thing that is required is the password everything else you can enter to continue to the end.

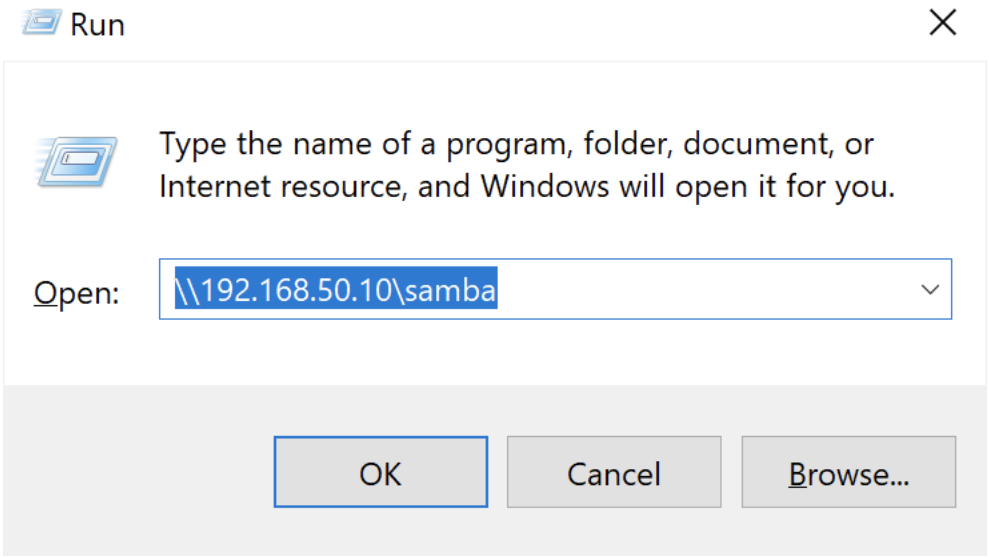
2. Create a folder **Lab_2** in your samba directory (**cd ~** to go back to your home directory) and add a text file **assignment2.txt** in your Lab_2 folder**.** Edit the text file and type in "I finally made it!!!" and then save the file.



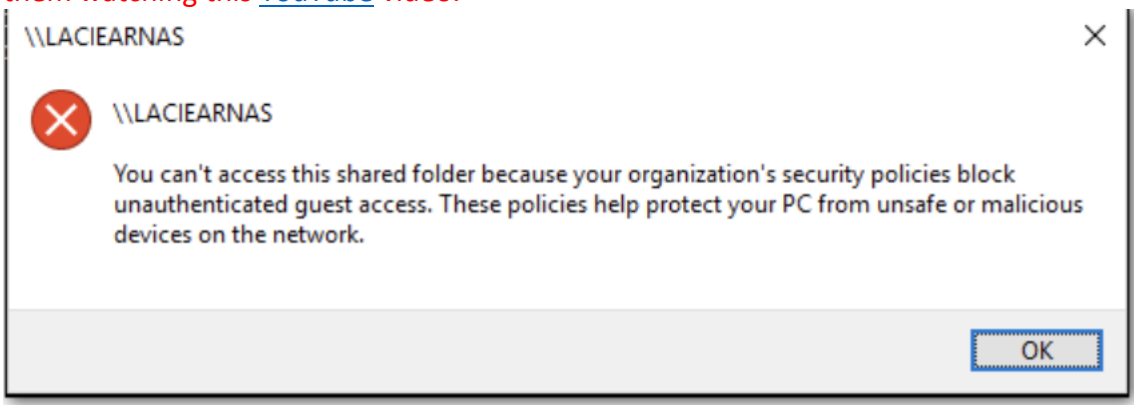
3. Access the Samba Ubuntu Server using Windows 10 VM.

   a. Disable the firewalls on Windows 10. In the search bar search for **Windows Defender Settings** -> **Click on Firewall & network protection** -> **Turn each one off by sliding the blue bar to off. Confirm after each one**.

b. On Windows 10, use **Run (Start menu > type Run) and enter \\<192.168.50.10>\samba**



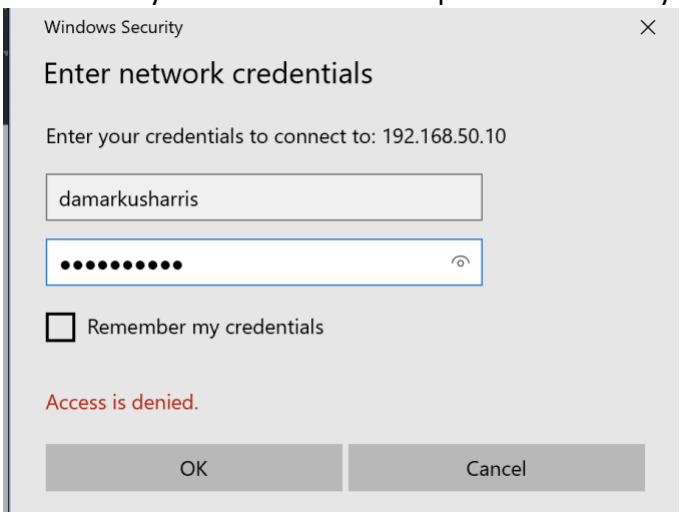
*If you receive an error code like the one below follow these steps to solve them watching this YouTube video.



*You may also need to allow samba through the Ubuntu firewall

```
damarkusharris@ubuntu:~$ sudo ufw allow samba
[sudo] password for damarkusharris:
Rules updated
Rules updated (v6)
```

5. Enter your credentials it should be what you created as your user on Ubuntu and Simba. Your ALL DONE! *Open up the assignment2.txt file and you should see what you wrote when on your Ubuntu OS. Edit the text and write “Yes, Samba works!”, and save the file. Go back to your Ubuntu OS and open that file and you should see the changes made.



Write-Up

Explain why having interoperability between the operating systems (OS) is critical in a corporate network. Relate these exercises to the Marconi Law project case.

With the Marconi Law project case we used network share to make a computer resource available from one host to other hosts on a computer network. It is important to have interoperability between OS because it makes it easier to share information/resources between disparate systems. Safety and protection of the data comes into play when sharing resources. This allows any type of user whether it is on a Windows or Linux system to share resources effectively. It overall makes communication quicker and efficient.