Basic Unix and Web Server setup

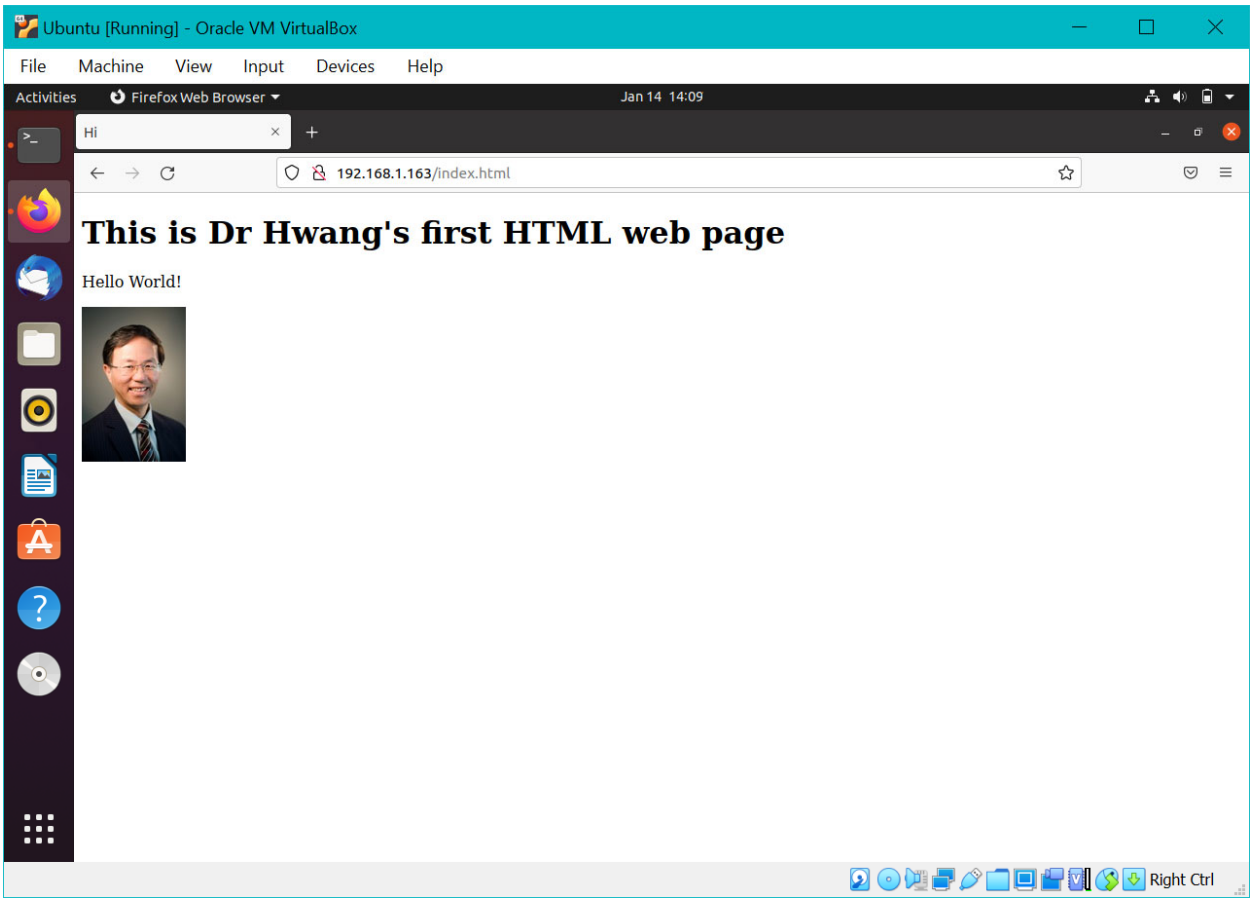
We will be using Apache for our web server. The Apache web server is among the most popular web servers in the world. It is well documented, and has been in wide use for much of the history of the web, which makes it a great default choice for hosting a website.

Follow the LAMP document instructions to install Apache if you have not already done so.

Global web pages

Global web pages (i.e. web pages not associated with a user account) are stored in the directory **/var/www/html**.

To browse to these web pages, type in a browser's address bar the ip address of the web server followed by the name of the webpage you want to access separated by slashes /. For example the URL address **192.168.1.163/index.html** accesses the **index.html** webpage on the server whose ip address is **192.168.1.163**. If you do not specify a webpage name and just enter the ip address, the default webpage to load is index.html.



If you put web pages in subdirectories under /var/www/html then you need to specify the path to the web page in the address, for example, **192.168.1.163/products/orange.html**, will retrieve the webpage **orange.html** under the **products** directory.

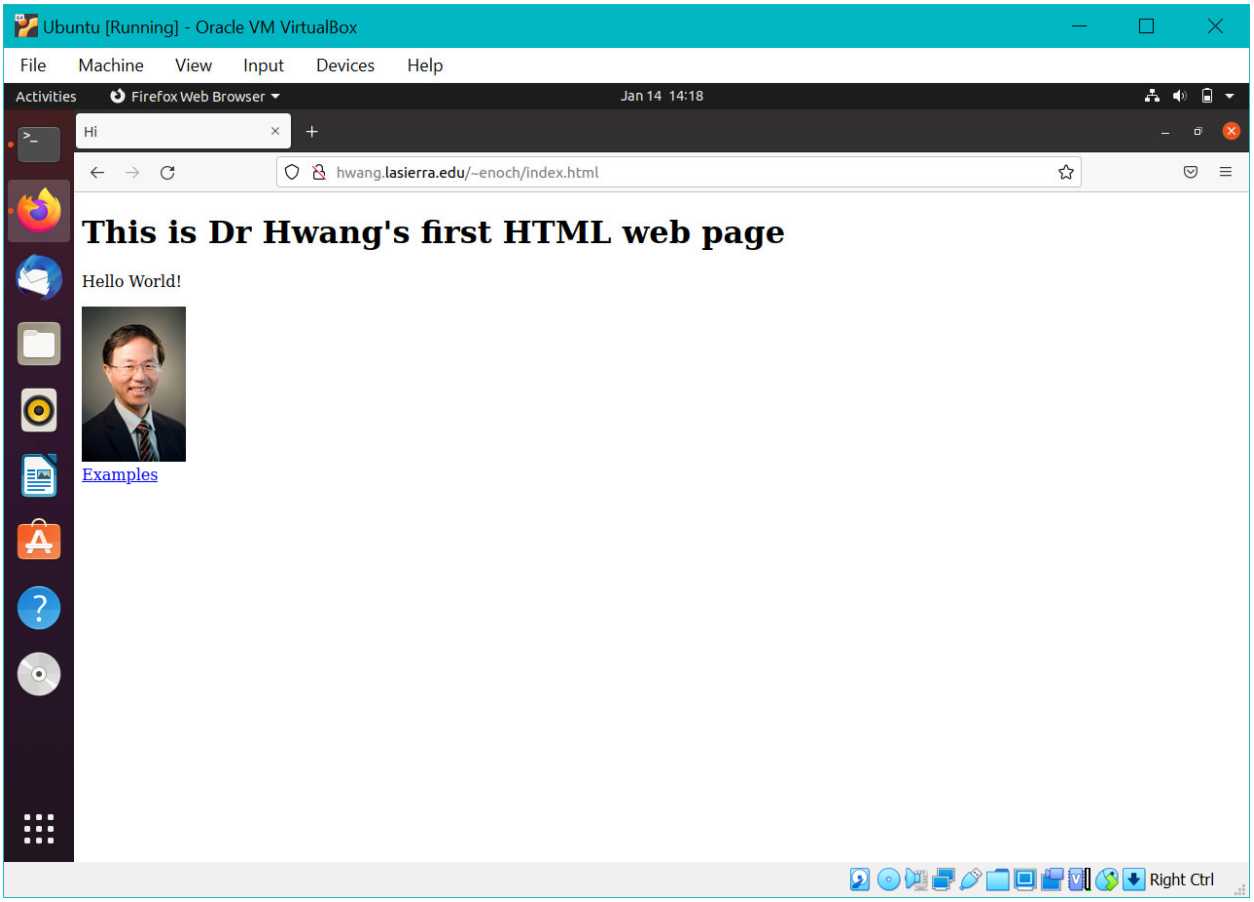
User web pages

User web pages are stored in the user's home directory under **public_html**.
To enable this, execute the **a2enmod** command and then restart apache

```
$ sudo a2enmod userdir
$ sudo systemctl restart apache2
```

To enable PHP files stored in public_html, edit the file **/etc/apache2/mods-enabled/php7.4.conf** (replace the version number with the installed version number). At the end of the file, comment out the lines from <IfModule …> to </IfModule>. Restart apache with the systemctl command shown above.

Browsing to the address **hwang.lasierra.edu/~enoch/index.html** will bring up the webpage **index.html** in the directory **public_html** in user **enoch** on server **hwang.lasierra.edu**. Again the file name index.html is optional. The full path to this file is **/home/enoch/pubic_html/index.html**.



Refer to the table at the end of this document for a list of the commonly used Unix commands. In your Ubuntu system open up a terminal window.

Creating your first web page

Change directory to /var/www/html.
Create a html file named MyFirstHTML.html

```
$ cd /var/www/html
$ sudo nano MyFirstHTML.html
```

Type in the following html code in the file and save it.

```
<!DOCTYPE html>
<html>
<head>
        <title>Hi</title>
</head>

<body>
        <h1>This is Dr Hwang's first HTML web page</h1>
        <p>Hello World!</p>
        <img src="Enoch.jpg" alt="Dr Hwang" width="107" height="160">
        <img src="http://hwang.lasierra.edu/Enoch.jpg" alt="Dr Hwang"
width="107" height="160">
</body>
</html>
```

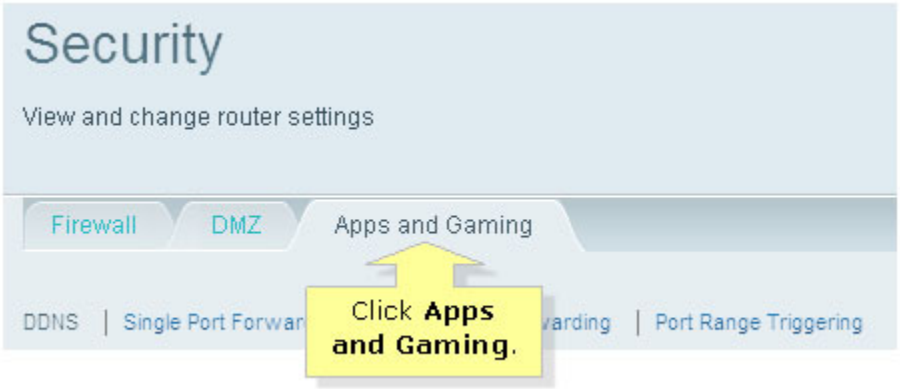
Browse to this webpage with the address 192.168.1.163/MyFirstHTML.html. Replace the ip address with your ip address. You will see that the image has a broken link because the file Enoch.jpg does not exist in your directory.

Port forwarding through router

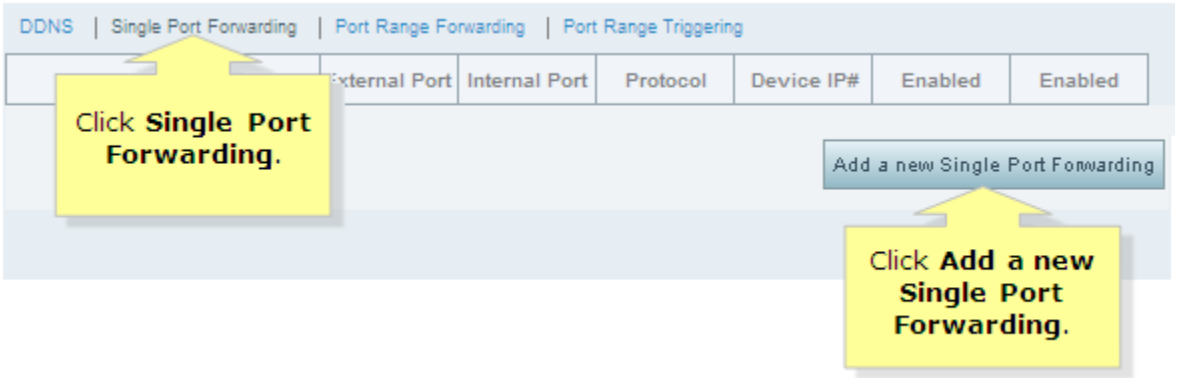
With your current setup, you can only browse to your webpages from inside your router/home. If you connect to the internet from outside your router you will not be able to get through your router's firewall and to your server. For example, from where I am, I can't browse to your webpages. A quick way to see that is to turn off the wifi on your phone and just use your cellular data, and try to browse to your webpage from your phone. You wouldn't be able to get to the webpage. But if you turn your wifi back on then you should be able to browse to it. (Do you know why that is so?)

What you need to do is to do a port forwarding on your router to open a port through the firewall. Every router's setup user interface is different and I can't cover all cases, but the idea is the same. The following instruction is for a Linksys router.

1. Browse to your router's webpage. The ip address is usually 192.168.1.1.
2. Enter the router's admin username and password. The default username is blank and the default password is **admin**. For other routers, just google for the default username and password for your brand.
3. Go to the Settings page.
   For other routers just find where Port Forwarding is, or google for Port Forwarding instructions for your router and continue with step 8 below.
4. Click on the Security tab.
5. Click on the Apps and Gaming tab.



6. Click Single Port Forwarding.



7. Click Add a new single port forwarding
8. Enter in the correct port you want to forward. The information that you need to enter in is the same for all router brands.
    - Application name: type in any name you like, for example, **web**
    - External port: **80**.[1]
    - Internal port: **80**.
    - Protocol: **Both** UDP and TCP. If it doesn't give you the **Both** option, then you need to add two lines, one for UDP and one for TCP.
    - Device IP: the ip address of your server. In my case it'll be 192.168.1.163

    Make sure you save the changes.

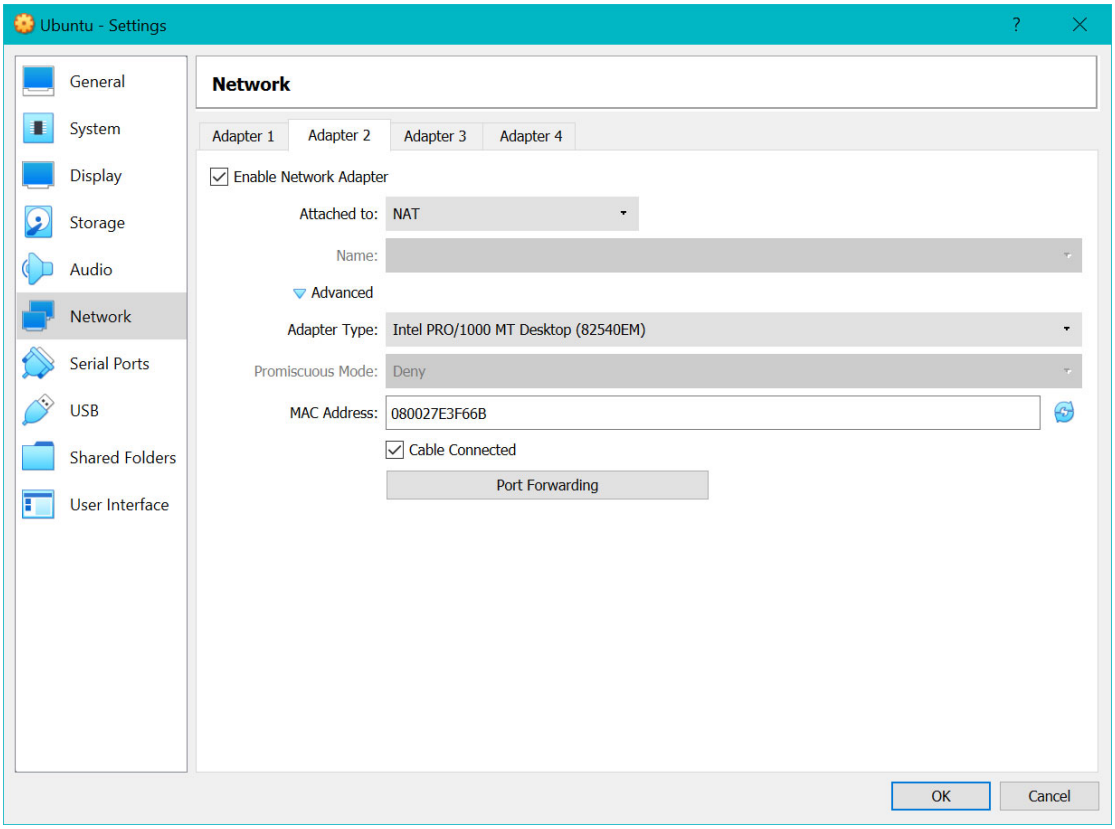
[1] Port 80 is the default port for web browsing. If you use another number then you need to configure Apache to listen to this other number.

Port forwarding through VirtualBox

Reference: https://nsrc.org/workshops/2014/btnog/raw-attachment/wiki/Track2Agenda/ex-virtualbox-portforward-ssh.htm

Port forwarding through your VirtualBox from your host machine (Windows) to your virtual Ubuntu machine is only needed if in your router's port forwarding setup page you cannot directly enter in the ip address of your Ubuntu virtual machine. Instead you are only given a list of devices that you can select from and the Ubuntu virtual machine doesn't show up in the list. Only your host Windows machine shows up in the list. By port forwarding from your host machine to your virtual machine, you can select your host Windows machine from the list, and the traffic will be sent to your virtual machine.

Turn off your virtual Ubuntu machine
Select your Ubuntu machine.
Select **Settings** | **Network**
Fill in the page as shown below
Select **Adapter 2**
Check **Enable Network Adapter**
Select **NAT** for Attached to:



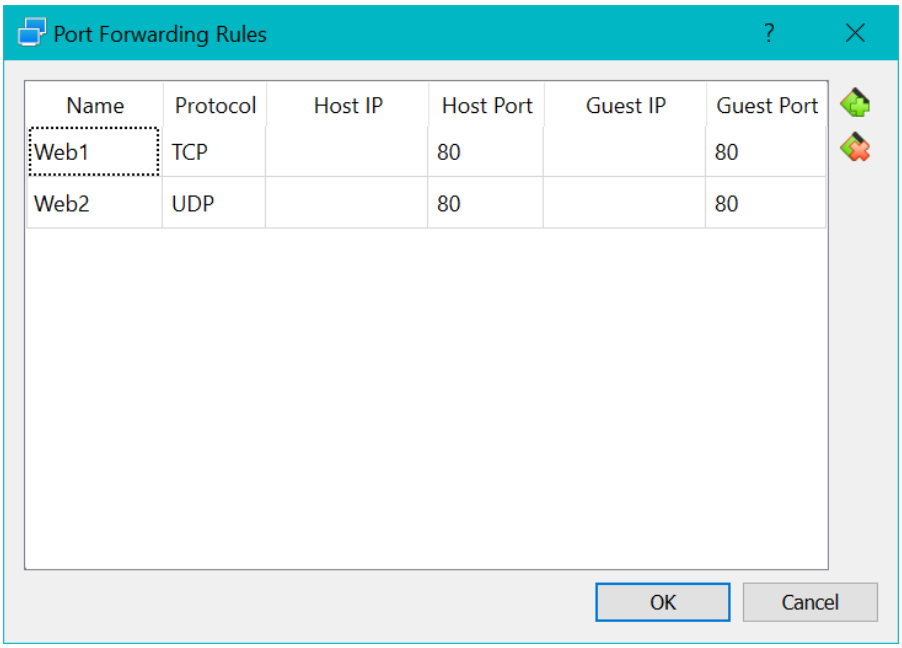
Click on **Advanced**
Click on **Port Forwarding**

Click on the green plus sign and add the first rule shown below

- Name: Web1
- Protocol: TCP
- Host Port: 80
- Guest Port: 80

Click on the green plus sign a second time and add the second rule shown below

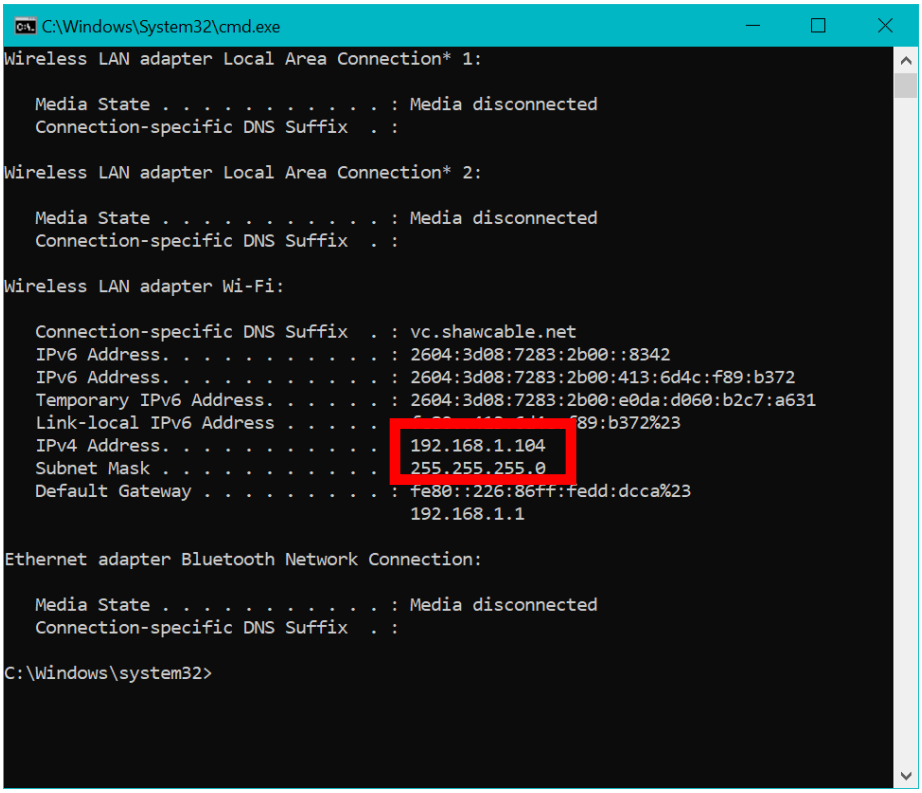
- Name: Web2
- Protocol: UDP
- Host Port: 80
- Guest Port: 80



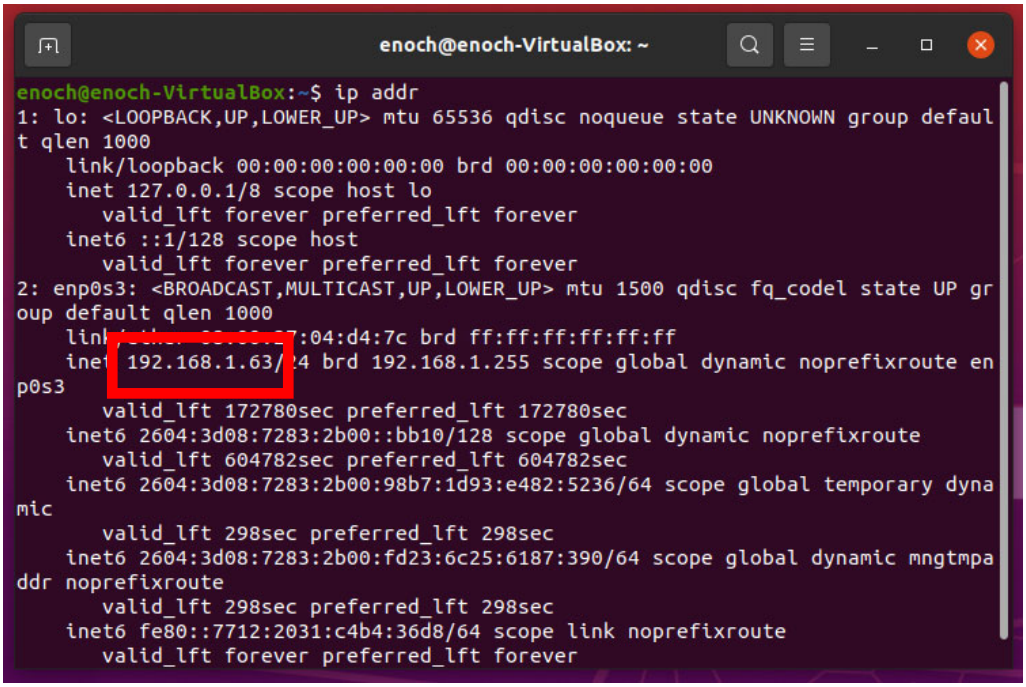
Click OK to save and exit

Start the virtual machine.
Now you should be able to browse to your Ubuntu web server using either your Windows host ip address or your Ubuntu guest web server ip address.

To find your Windows' ip address, open a **cmd** window. Type in the command **ipconfig**. As shown below my Windows' IPv4 ip address is 192.168.1.104

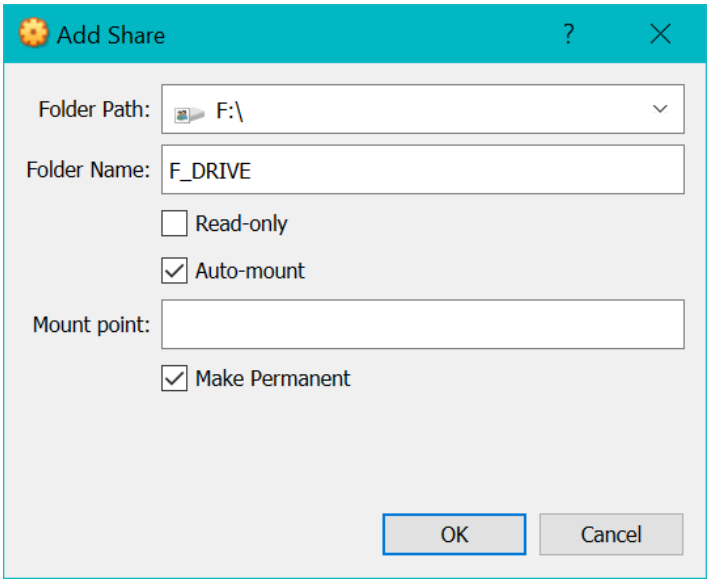


To find your Ubuntu's ip address, open a **terminal** window. Type in the command **ip addr**. As shown below my Ubuntu's IPv4 ip address is 192.168.1.63



VirtualBox Shared Folders

From the VirtualBox menu, select **Devices** | **Shared Folders** | **Shared Folders Settings**
Click on the **Add new shared folder** icon. Select the Folder Path you want to share. Check **Auto-mount** and **Make Permanent**.



SSH Server

The Secure SHell (SSH) allows you to use a terminal to remotely login to your server. After you login, you will be able to work on your server from a remote computer as if you are directly using the server. In order to use SSH, you first need to install the SSH server on your Ubuntu machine by executing the following commands:

```
$ sudo apt update
$ sudo apt install openssh-server
```

To verify that SSH is active and running

```
$ sudo systemctl status ssh
```

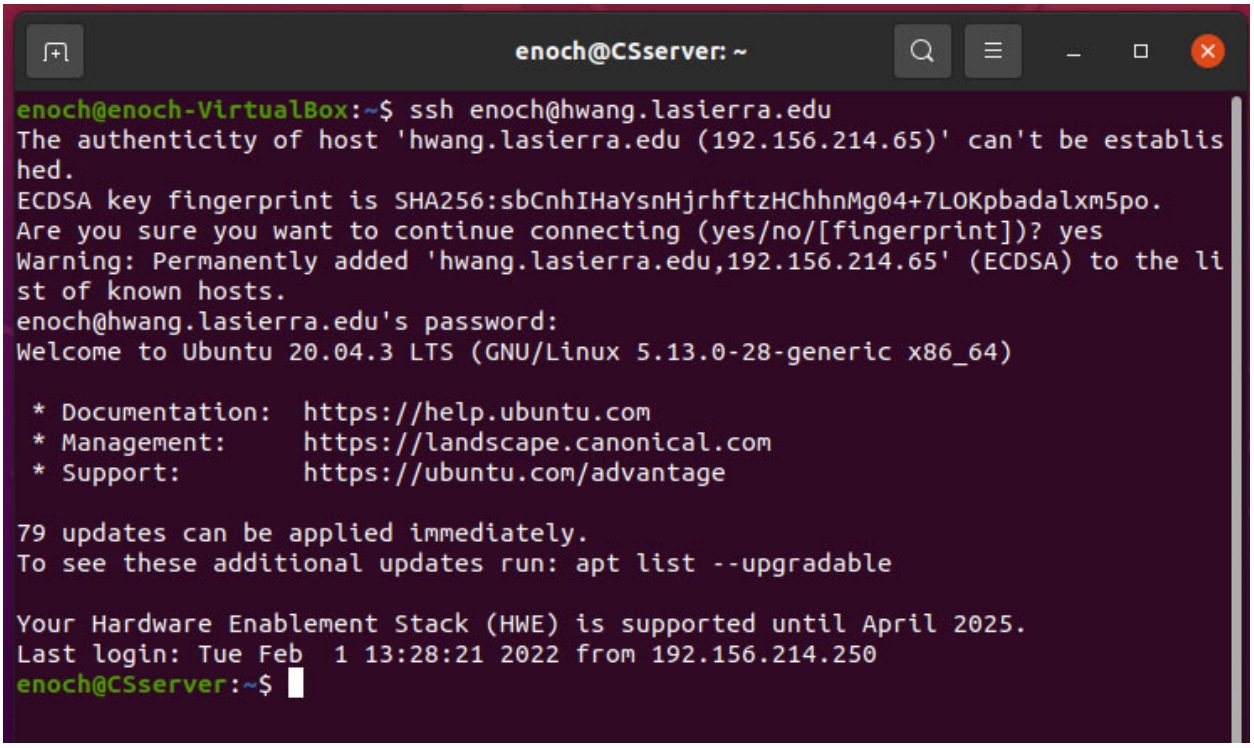
Open the SSH port through the firewall to allow traffic through

```
$ sudo ufw allow ssh
```

SSH Client

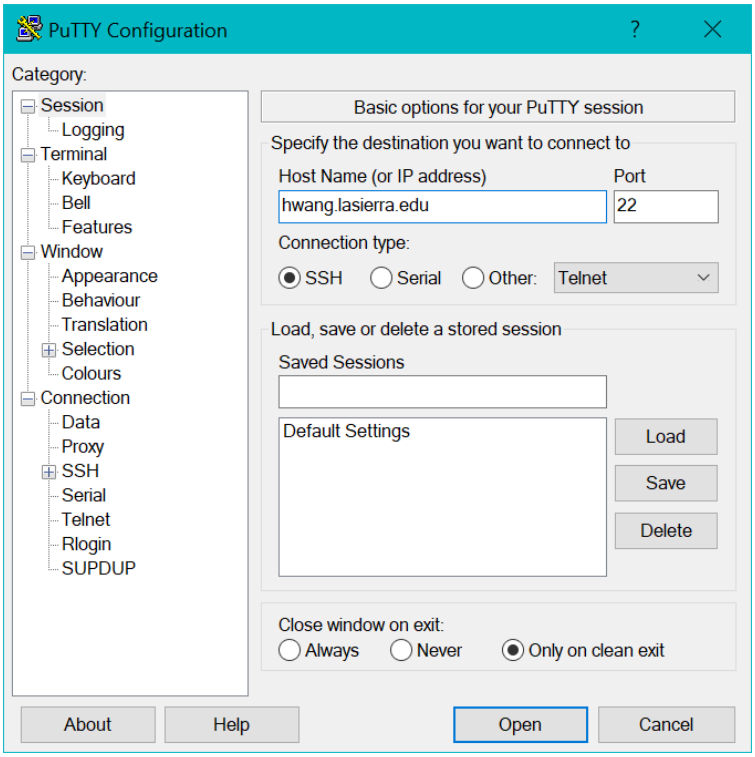
Remote Ubuntu or Mac machine

From a remote Ubuntu machine open a terminal window and type **ssh account@server_address** at the prompt to connect to the server.

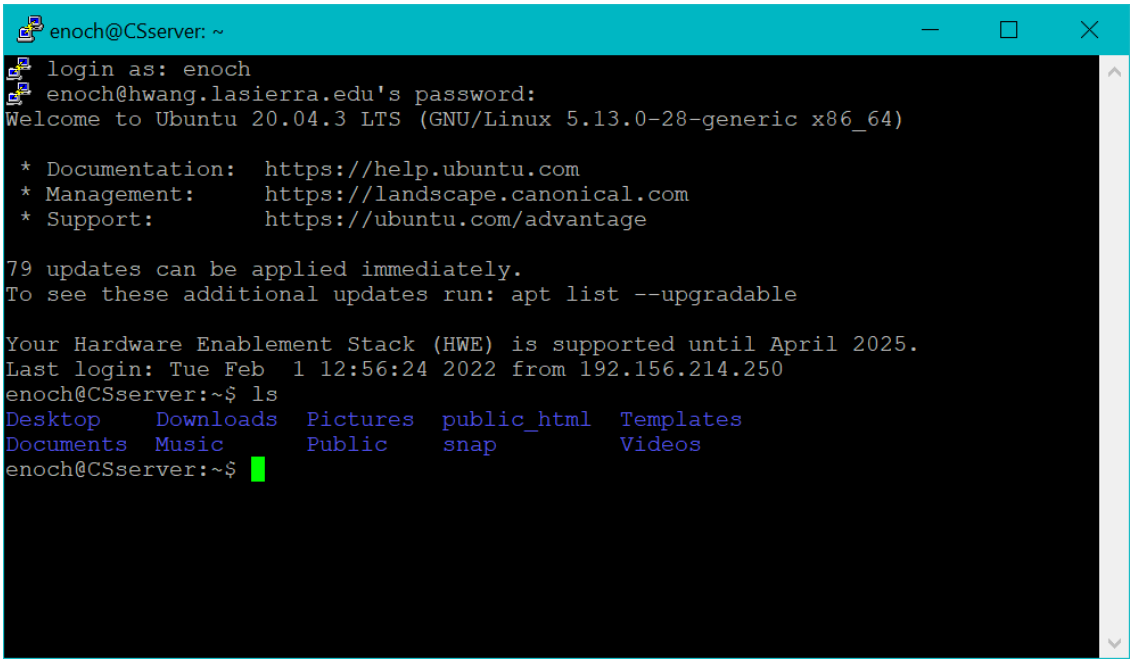


Remote Windows machine

From a remote Windows machine you need to first download **PuTTY** from **putty.org** and install it. Run the program. In the PuTTY Configuration window, type in the server address **hwang.lasierra.edu** and click Open.



After the connection is established with a terminal window, you will need to login with your user account and password.



SCP

Secure copy a file from Windows to Linux. To use scp, you need to have installed SSH. From a Windows' cmd window, type the **scp** command. The syntax of the command is **scp fromFile toFile**. The following are three examples of copying the file Enoch.jpg in the root directory of the C: drive to the directory public_html using the same name.

```
C:\scp c:/Enoch.jpg enoch@hwang.lasierra.edu:public_html/Enoch.jpg

C:\scp c:/Enoch.jpg enoch@hwang.lasierra.edu:public_html/.

C:\scp c:/Enoch.jpg
 enoch@hwang.lasierra.edu:/home/enoch/public_html/Enoch.jpg
```

FTP

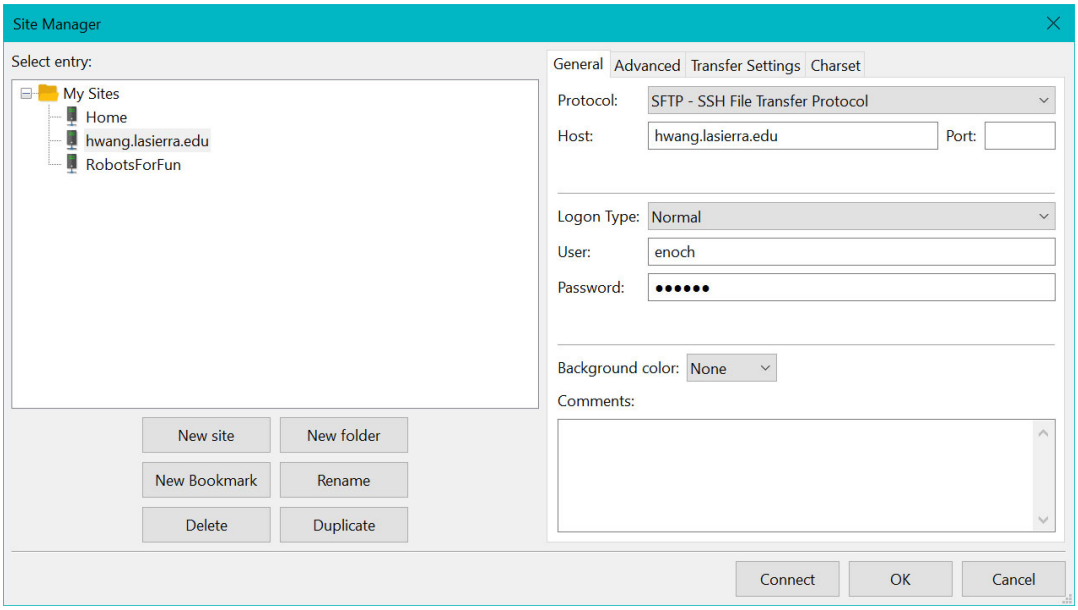
The File Transfer Protocol (FTP) is for copying files from one computer to another computer. In order to use FTP, you first need to install the FTP server **vsftpd** on your Ubuntu machine by executing the following commands:

```
$ sudo apt update
$ sudo apt install vsftpd
```

Reference: https://linuxize.com/post/how-to-setup-ftp-server-with-vsftpd-on-ubuntu-20-04/

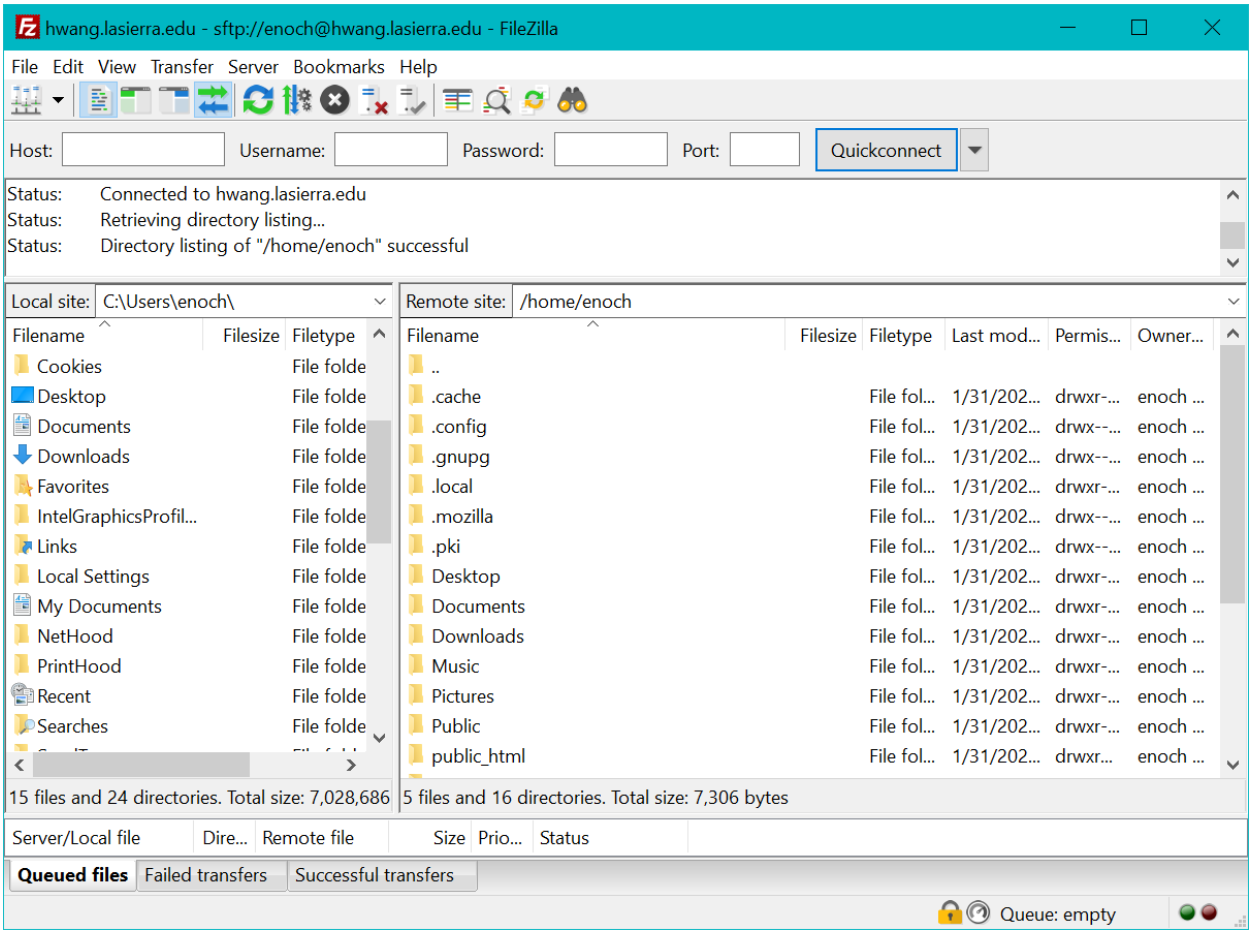
To use FTP to copy a file from a remote computer, such as your Windows computer, to your Ubuntu computer, you run a FTP client program on your Windows computer. A popular free FTP client program is FileZilla. Download the FileZilla client program from https://filezilla-project.org/.

In the FileZilla Site Manager window, create a new site and fill in the host address, user name and password. Select SFTP for the Protocol.



Click OK to save

Click Connect to connect to the server.



(Optional) Configure Apache to listen to another port

Port 80 is the default port for web browsing. If you want to use another port number you need to configure Apache to listen to this new number.

1. Edit /etc/apache2/ports.conf
2. Change the line **Listen 80** to **Listen 8081** where 8081 is the new port number that you want.
3. Edit /etc/apache2/sites-enabled/000-default.conf
4. Change the line **<VirtualHost *:80>** to **<VirtualHost *:8081>**
5. Restart Apache.

```
$ systemctl restart apache2
```

6. To check that Apache is using the new port number

```
$ sudo apt install net-tools
$ sudo netstat -tlpn | grep apache
```

It should print out a line that says 8081 apache2

To browse to your web server you now need to specify the port as part of the URL address in your browser. For example, **192.168.1.163:8081/index.html** will use port 8081. If you don't specify the port 8081, it will default to port 80 and the Apache web server will not respond.

Unix/Linux/Ubuntu Common Commands

Here are some Unix commands that you will commonly use. Unix commands **are** case sensitive. The first word is the command and the following words are parameters to the command.

Command	**Operation**	**Example**
cd	Go to your home directory	cd
cd /var/www/html	Change directory to /var/www/html	cd /var/www/html
cd ..	Go up one level	cd ..
pwd	Show current directory	pwd
cp fromfile tofile	Copy fromfile to tofile	cp in.txt index.html
mv fromfile tofile	Move/Rename fromfile to tofile	mv in.txt index.html
mkdir directory	Make (create) directory	mkdir images
rm filename	Remove (delete) file	rm in.txt
rm -r directory	Remove (delete) directory and everything inside it	rm -r images
chmod per file	Change read/write/execute permissions for a file or directory	chmod 777 index.html
nano file	Edit file	nano index.html
passwd	Change your account password	passwd
clear	Clear the terminal screen	clear
ping ip_address	Check connection to computer with given ip address	ping -4 google.com
ip addr	Get ip address of computer	ip addr
netstat	See what ports computer is listening on	netstat -a

Get more details on the commands here.