Home Server Project

Full Documentation | 9-25-2023

Purpose

I have a good fundamental understanding of networking but not much hands-on experience, however. The goal of this project is to increase my understanding of networking and specifically how servers' function, as well as common use-cases for them. This project will also provide the foundation for many more projects to come.

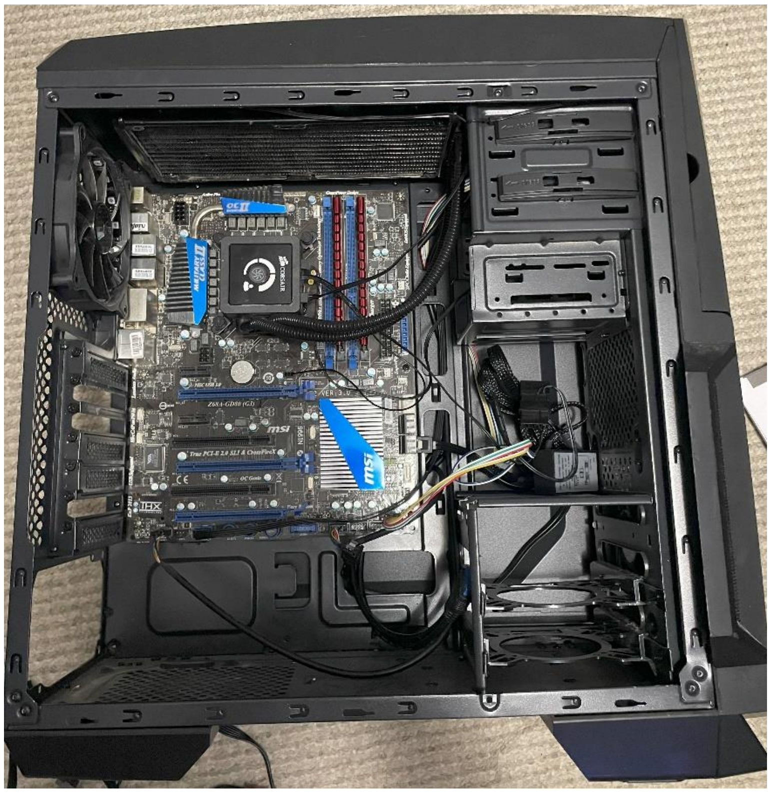
Scope

There are key objectives that will be accomplished in this project. These objectives aim to simulate servers seen in real-world environments as well as the usefulness of having one in a private environment. The key objectives are as follows:

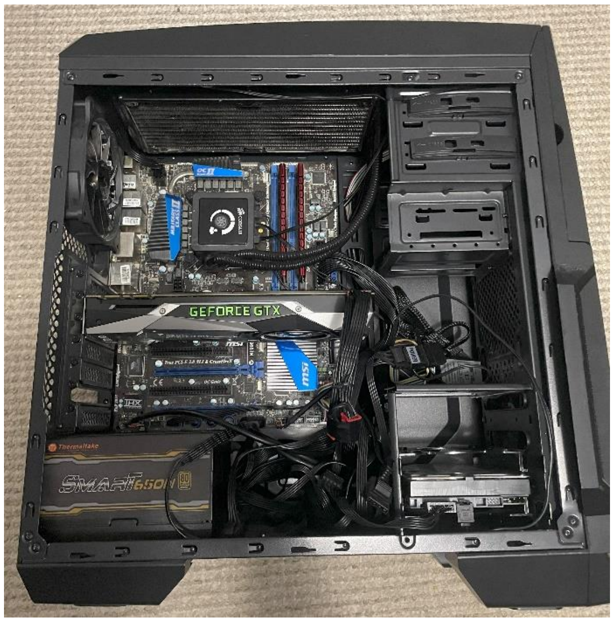
- Build the physical server.
- Flash the operating system ISO and install it on the server.
- Successfully get the server up and running.
- Perform updates or patches on the server.
- Harden the system.
- Enable remote access for maintenance and control.
- Configure a Network-Attached Storage (NAS).
- Download a hypervisor and an operating system.
- Install Apache and host a webpage.

Project

The first step in this project is to build the physical server. Luckily, I already had all of the components needed, most of which are from my first gaming pc. There were only a few components that I needed to configure as can be seen in the picture below.

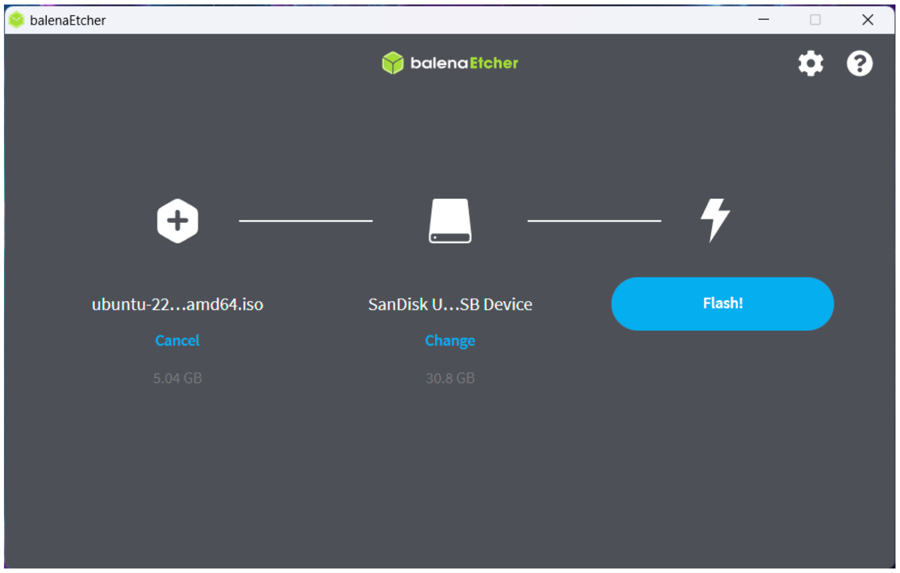


From here, all I need to install is the GPU and power supply. Luckily, this saved a lot of time and money due to the fact that I did not need to build a system from scratch. The finished build can be seen in the pictures below.

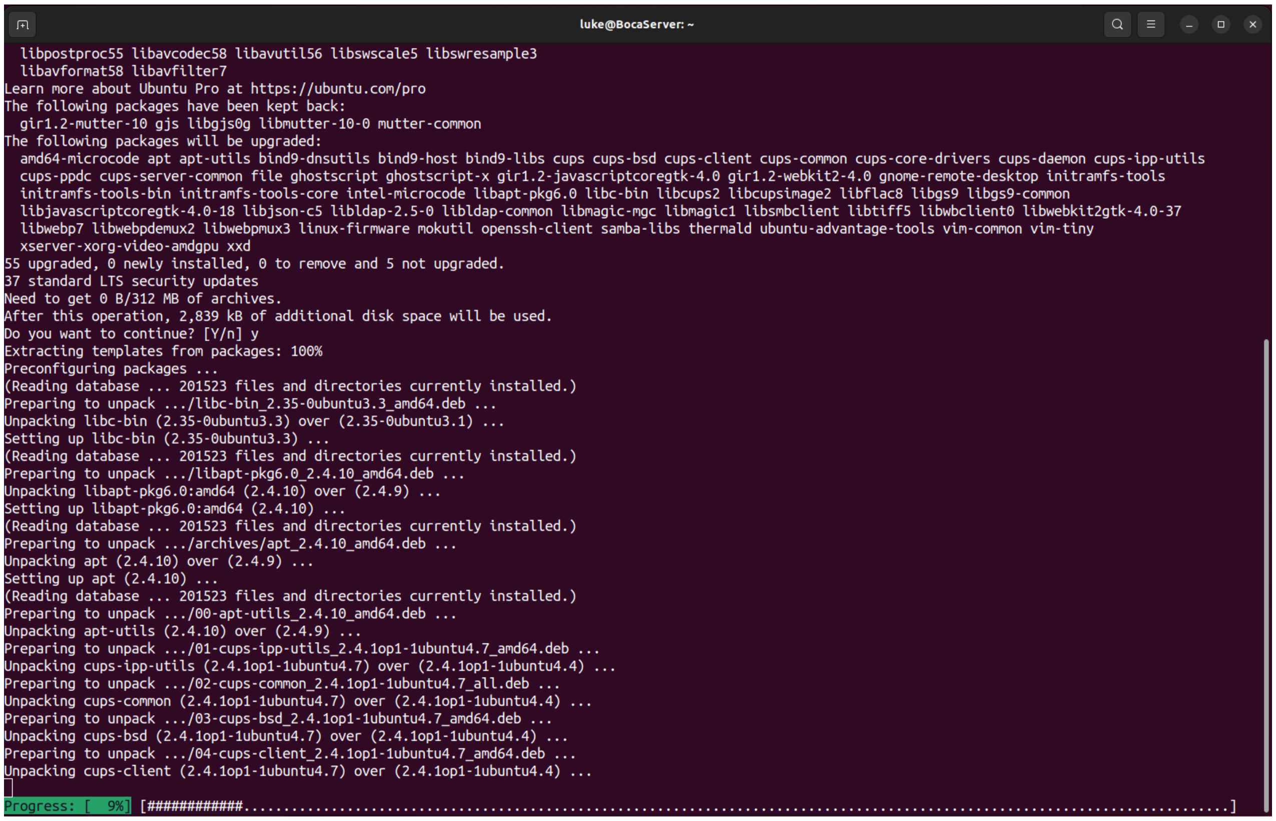


Now that the physical server has been built and 'POSTed' successfully, it is time to set the server aside and get the operating system ready for installation.

I decided to use Ubuntu for a few reasons. The main reason is that it is a well-respected Linux distribution and one that I am unfamiliar with. After deciding what OS to use, I navigated to ubuntu.com and installed the ISO image. From here, it was time to flash this ISO image on a thumb drive in order to install Ubuntu on the server. I used BalenaEtcher to achieve this and can be seen in the screenshot below. I then booted from the flash drive and installed Ubuntu on the server.

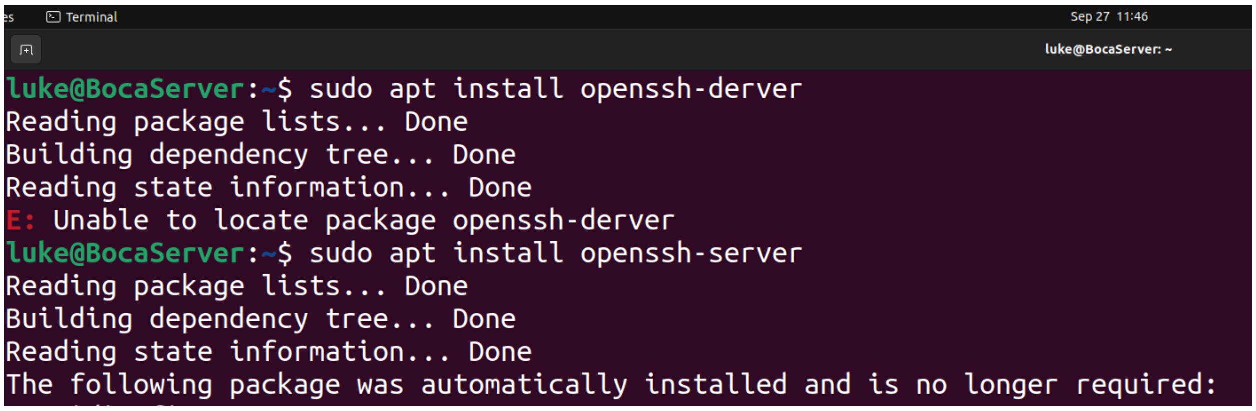


Ubuntu is now officially installed on the server ready to go. The first step now is to update or patch the system because there is no telling how old the ISO image is. To accomplish this, I navigate to the terminal and issue the commands "`sudo apt update && sudo apt upgrade`." This is a combined command to tell the system to both update and upgrade its software and packages. This is an important step for many reasons, and in this case, I am practicing hardening the server by patching it.



```
  libpostproc55 libavcodec58 libavutil56 libswscale5 libswresample3
  libavformat58 libavfilter7
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following packages have been kept back:
  gir1.2-mutter-10 gjs libgjs0g libmutter-10-0 mutter-common
The following packages will be upgraded:
  amd64-microcode apt apt-utils bind9-dnsutils bind9-host bind9-libs cups cups-bsd cups-client cups-common cups-core-drivers cups-daemon cups-ipp-utils
  cups-ppdc cups-server-common file ghostscript ghostscript-x gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0 gnome-remote-desktop initramfs-tools
  initramfs-tools-bin initramfs-tools-core intel-microcode libapt-pkg6.0 libc-bin libcups2 libcupsimage2 libflac8 libgs9 libgs9-common
  libjavascriptcoregtk-4.0-18 libjson-c5 libldap-2.5-0 libldap-common libmagic-mgc libmagic1 libsmbclient libtiff5 libwbclient0 libwebkit2gtk-4.0-37
  libwebp7 libwebpdemux2 libwebpmux3 linux-firmware mokutil openssh-client samba-libs thermald ubuntu-advantage-tools vim-common vim-tiny
  xserver-xorg-video-amdgpu xxd
55 upgraded, 0 newly installed, 0 to remove and 5 not upgraded.
37 standard LTS security updates
Need to get 0 B/312 MB of archives.
After this operation, 2,839 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Extracting templates from packages: 100%
Preconfiguring packages ...
(Reading database ... 201523 files and directories currently installed.)
Preparing to unpack .../libc-bin_2.35-0ubuntu3.3_amd64.deb ...
Unpacking libc-bin (2.35-0ubuntu3.3) over (2.35-0ubuntu3.1) ...
Setting up libc-bin (2.35-0ubuntu3.3) ...
(Reading database ... 201523 files and directories currently installed.)
Preparing to unpack .../libapt-pkg6.0_2.4.10_amd64.deb ...
Unpacking libapt-pkg6.0:amd64 (2.4.10) over (2.4.9) ...
Setting up libapt-pkg6.0:amd64 (2.4.10) ...
(Reading database ... 201523 files and directories currently installed.)
Preparing to unpack .../archives/apt_2.4.10_amd64.deb ...
Unpacking apt (2.4.10) over (2.4.9) ...
Setting up apt (2.4.10) ...
(Reading database ... 201523 files and directories currently installed.)
Preparing to unpack .../00-apt-utils_2.4.10_amd64.deb ...
Unpacking apt-utils (2.4.10) over (2.4.9) ...
Preparing to unpack .../01-cups-ipp-utils_2.4.1op1-1ubuntu4.7_amd64.deb ...
Unpacking cups-ipp-utils (2.4.1op1-1ubuntu4.7) over (2.4.1op1-1ubuntu4.4) ...
Preparing to unpack .../02-cups-common_2.4.1op1-1ubuntu4.7_all.deb ...
Unpacking cups-common (2.4.1op1-1ubuntu4.7) over (2.4.1op1-1ubuntu4.4) ...
Preparing to unpack .../03-cups-bsd_2.4.1op1-1ubuntu4.7_amd64.deb ...
Unpacking cups-bsd (2.4.1op1-1ubuntu4.7) over (2.4.1op1-1ubuntu4.4) ...
Preparing to unpack .../04-cups-client_2.4.1op1-1ubuntu4.7_amd64.deb ...
Unpacking cups-client (2.4.1op1-1ubuntu4.7) over (2.4.1op1-1ubuntu4.4) ...

Progress: [  9%] [#############..........................................................................................................................]
```

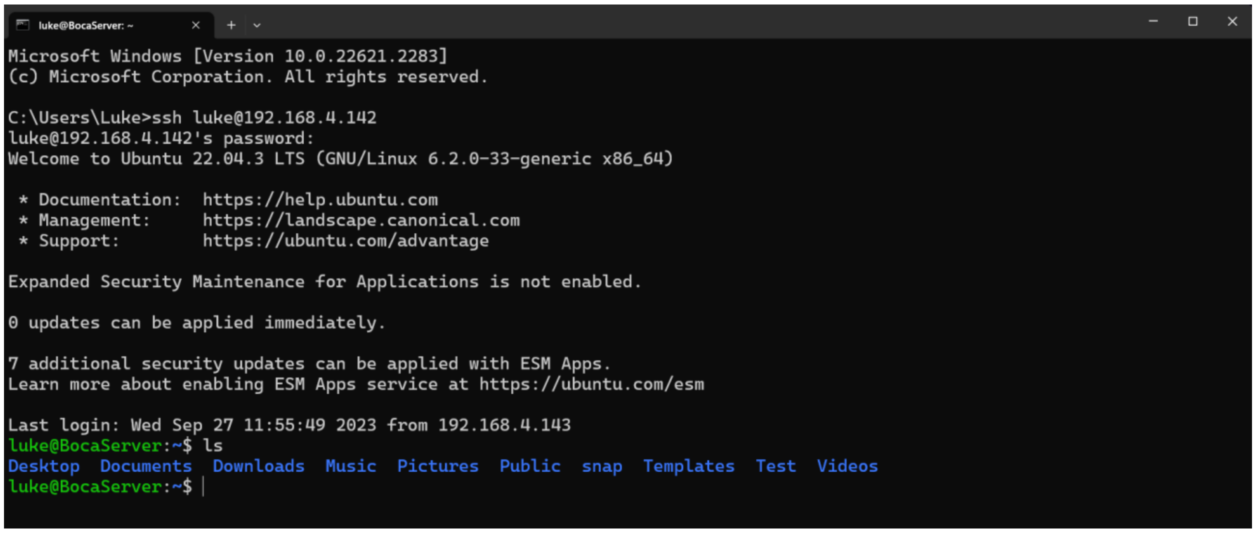
The next objective is to enable remote access to the machine, and this will be achieved through the SSH protocol. First, I install Openssh Server with the command "`sudo apt install openssh-server`." The next step is to make sure SSH is allowed through the firewall, so I issue the command "`sudo ufw allow openssh`." This can be seen in the screenshot below.



```
luke@BocaServer:~$ sudo apt install openssh-derver
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package openssh-derver
luke@BocaServer:~$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
```

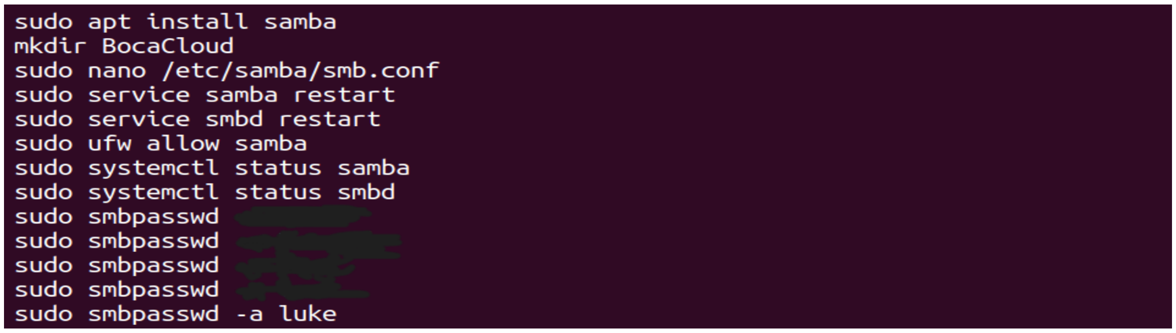
An extra security measure to further harden the system is to disable root login over SSH. The reason being that it is very risky to leave this option on as an attacker just has to brute force a password in order to gain root access.

To verify that SSH is working properly, I switched to my Windows Desktop and navigated to the command prompt. I then successfully connected to the server from my Windows machine.



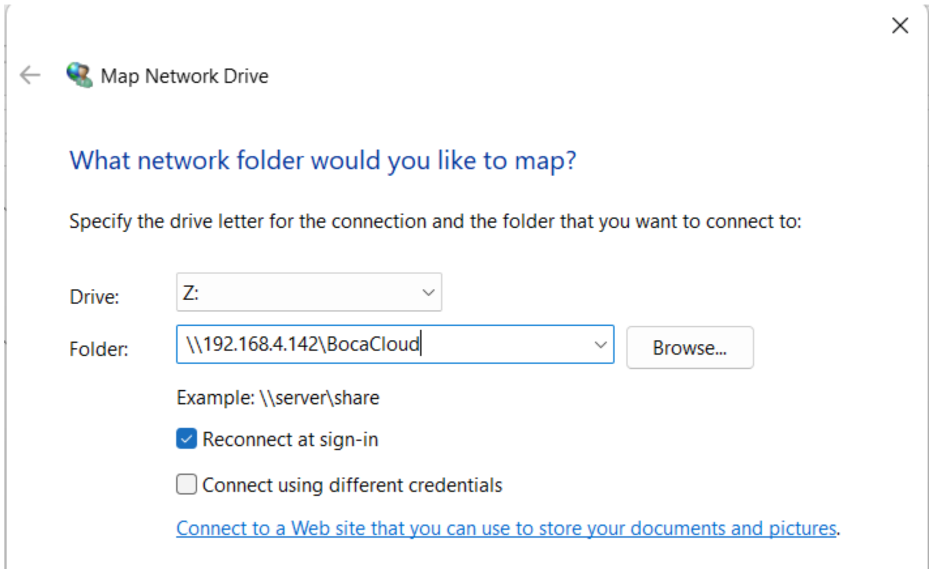
```
luke@BocaServer: ~

Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Luke>ssh luke@192.168.4.142
luke@192.168.4.142's password:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Wed Sep 27 11:55:49 2023 from 192.168.4.143
luke@BocaServer:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  snap  Templates  Test  Videos
luke@BocaServer:~$ 
```

The next objective is to create a Network-Attached Storage which will be referred to as a NAS. This function will serve as additional storage for anyone on the network to access with credentials. This is just awesome to have and will feel like private cloud storage. The first step is to install Samba on the server which will utilize the Server Message Block (SMB) protocol. This can be achieved by issuing the command "`sudo apt install samba.`" From here, I went ahead and created the directory where the NAS will live. For this, I issued the command "`mkdir BocaCloud`." After that, I edited /etc/samba/smb.conf file to configure Samba. Next, I made sure that the firewall did not block SMB, so I issued the command "`sudo ufw allow samba`." Lastly, I configured a password in order to map the network drive and access the NAS with the command "`sudo smbpasswd -a luke`." These commands can be seen in the screenshot below.



```
sudo apt install samba
mkdir BocaCloud
sudo nano /etc/samba/smb.conf
sudo service samba restart
sudo service smbd restart
sudo ufw allow samba
sudo systemctl status samba
sudo systemctl status smbd
sudo smbpasswd [redacted]
sudo smbpasswd
sudo smbpasswd
sudo smbpasswd
sudo smbpasswd -a luke
```

The NAS is now configured on the server-side. Now it is time to map the NAS on my windows machine. From here, I navigate to the file explorer and right click on “My PC” and select “Map Network Drive.” I then enter IP address and path of the directory which can be seen in the screenshot below.



Map Network Drive

What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: \\192.168.4.142\BocaCloud

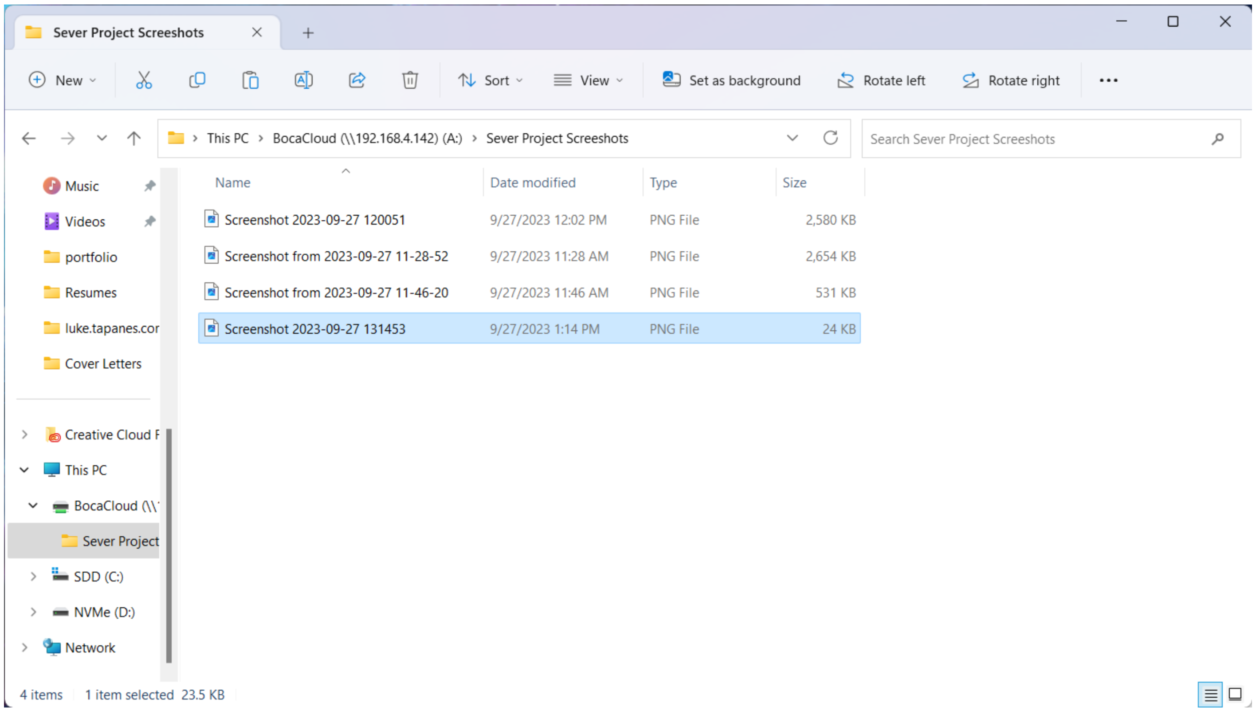
Example: \\server\share

Reconnect at sign-in [✓]

Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures.

After successfully mapping the network drive, I tested to see if it worked by uploading files from my server to my desktop and sure enough, it worked!



Sever Project Screeshots

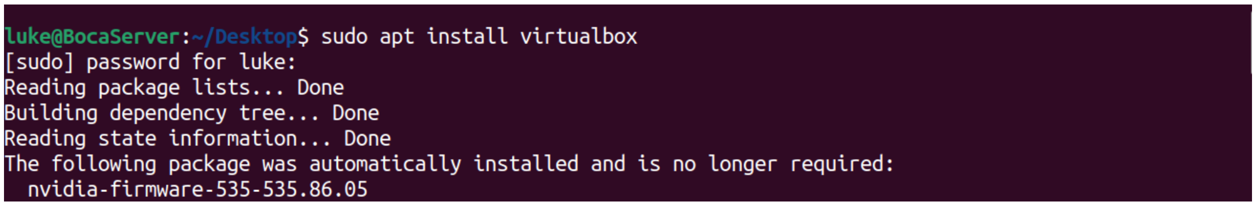
This PC > BocaCloud (\\192.168.4.142) (A:) > Sever Project Screeshots

Name	Date modified	Type	Size
Screenshot 2023-09-27 120051	9/27/2023 12:02 PM	PNG File	2,580 KB
Screenshot from 2023-09-27 11-28-52	9/27/2023 11:28 AM	PNG File	2,654 KB
Screenshot from 2023-09-27 11-46-20	9/27/2023 11:46 AM	PNG File	531 KB
Screenshot 2023-09-27 131453	9/27/2023 1:14 PM	PNG File	24 KB

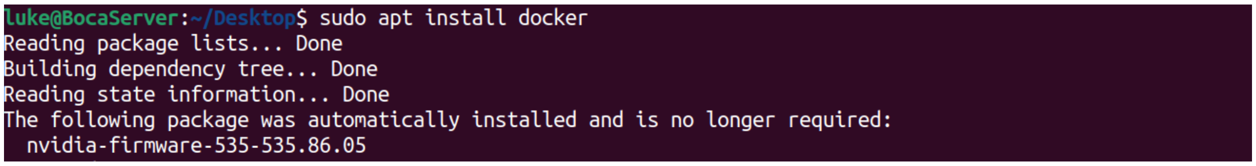
4 items 1 item selected 23.5 KB

The next objective is to install a hypervisor along with an OS to run on the hypervisor. The idea here is to simulate a real server environment as many servers operate multiple virtual machines and containers to host multiple services at the same time. The reason for this is to maximize the server's hardware utilization rather than waste an entire machine on one service. The hypervisor I chose to use for this objective is VirtualBox. I have used VirtualBox a ton in the past and know it will certainly get the job done. The operating system that I chose for this objective is Windows Server 2019. The reason being that I plan to do a future project involving Windows Server, so stay tuned for that.

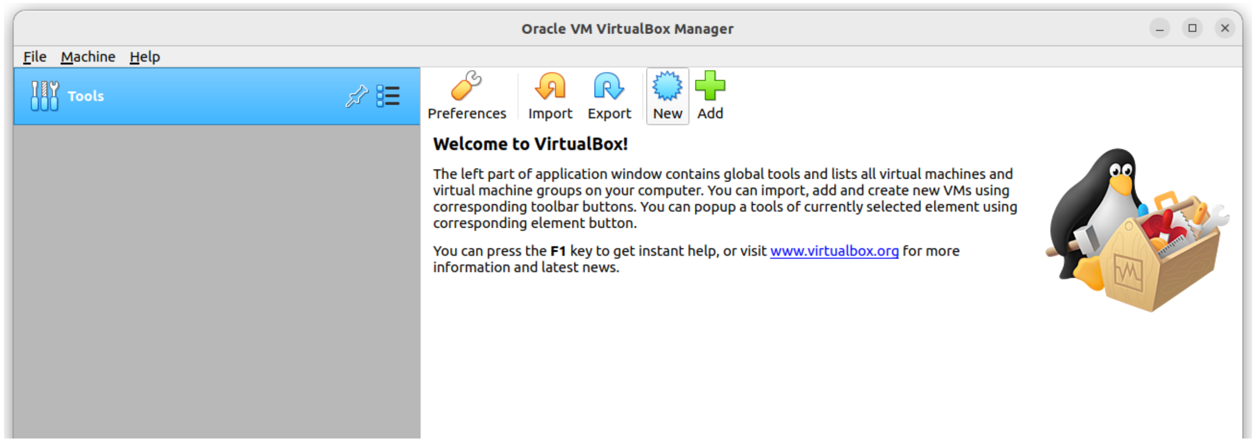
The first step is to install VirtualBox, and it is pretty straightforward as I issued the command "`sudo apt install virtualbox`." I also installed Docker while I was at it; however, I will not be utilizing docker in this project, but I went ahead and installed it for good measure. I then opened the application to verify that it was installed correctly, which can be seen in the screenshots below.



```
luke@BocaServer:~/Desktop$ sudo apt install virtualbox
[sudo] password for luke:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  nvidia-firmware-535-535.86.05
```



```
luke@BocaServer:~/Desktop$ sudo apt install docker
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  nvidia-firmware-535-535.86.05
```

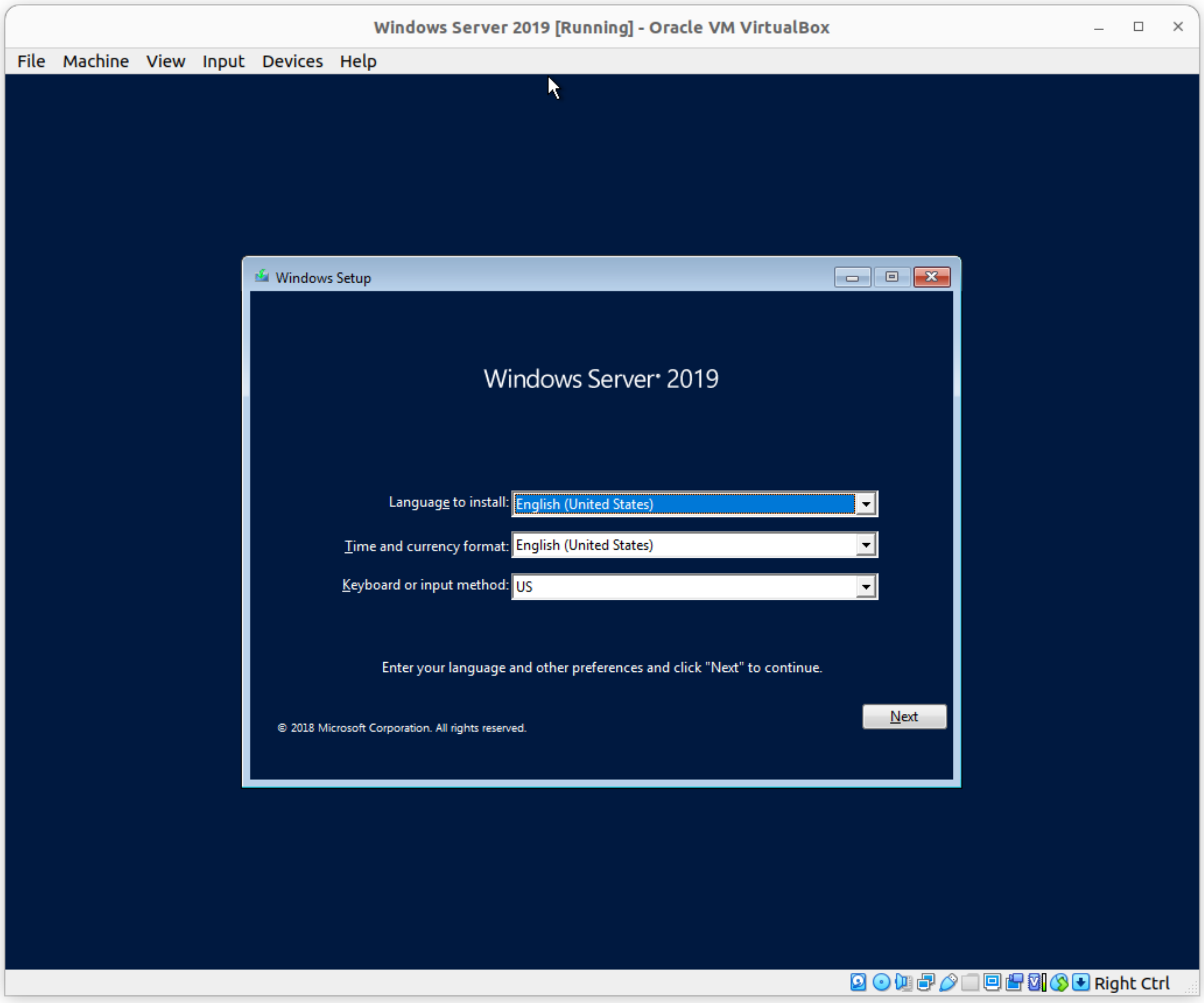

Oracle VM VirtualBox Manager
File Machine Help
Tools
Preferences Import Export New Add

Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

Now that the hypervisor is ready, it is time to download the operating system to go with it. I navigated to Microsoft.com and installed Windows Server 2019. The installation took about 10 minutes and then the ISO image was ready. From here, I added the ISO image in virtual box and allocated system resources to the VM. After that, I booted up the VM to verify if it was configured currently and it was. As soon as I was greeted with the Windows setup screen, I powered off the VM and the rest of the configuration will be included in the next project.



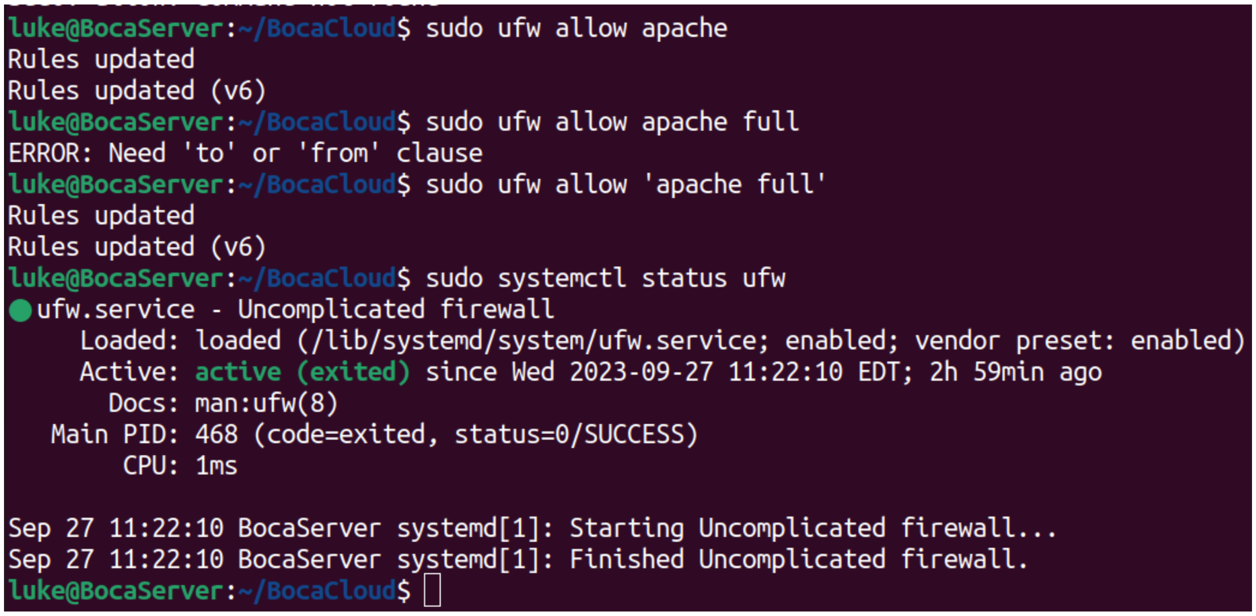
Language to install: English (United States)
Time and currency format: English (United States)
Keyboard or input method: US

Enter your language and other preferences and click "Next" to continue.

© 2018 Microsoft Corporation. All rights reserved.

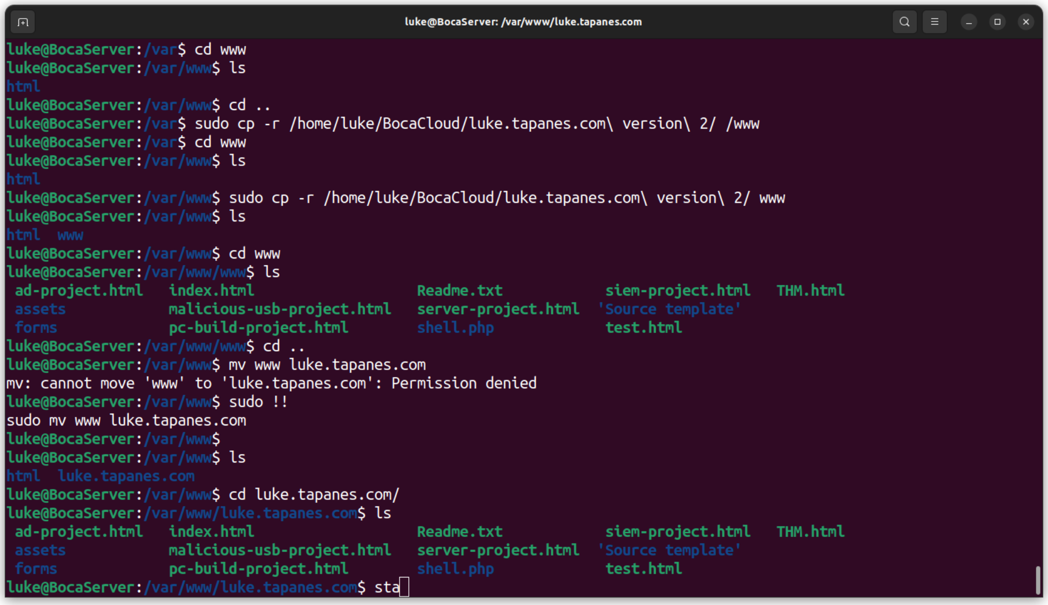
The final objective for this project is to install Apache and host a web page. For this project, the webpage will only be hosted on the network and is meant to simulate an intranet environment. A future project will involve hosting a webpage on the internet.

I first installed Apache with the simple command “`sudo apt install apache2`.” I then made sure to allow Apache through the firewall with the command “`sudo ufw allow ‘Apache’`” and “`sudo ufw allow ‘Apache Full’`.” To verify that these rules are active, I issued the command “`sudo systemctl status ufw`.” This can be seen in the screenshot below.



```
luke@BocaServer:~/BocaCloud$ sudo ufw allow apache
Rules updated
Rules updated (v6)
luke@BocaServer:~/BocaCloud$ sudo ufw allow apache full
ERROR: Need 'to' or 'from' clause
luke@BocaServer:~/BocaCloud$ sudo ufw allow 'apache full'
Rules updated
Rules updated (v6)
luke@BocaServer:~/BocaCloud$ sudo systemctl status ufw
● ufw.service - Uncomplicated firewall
     Loaded: loaded (/lib/systemd/system/ufw.service; enabled; vendor preset: enabled)
     Active: active (exited) since Wed 2023-09-27 11:22:10 EDT; 2h 59min ago
       Docs: man:ufw(8)
   Main PID: 468 (code=exited, status=0/SUCCESS)
        CPU: 1ms

Sep 27 11:22:10 BocaServer systemd[1]: Starting Uncomplicated firewall...
Sep 27 11:22:10 BocaServer systemd[1]: Finished Uncomplicated firewall.
luke@BocaServer:~/BocaCloud$ 
```

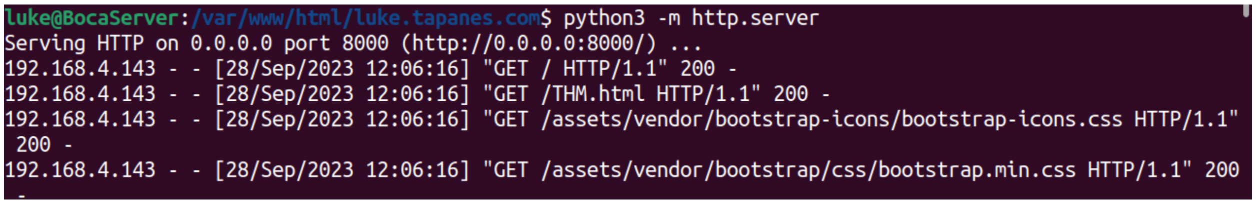
Now that Apache is configured, it is time to copy the website files over to the correct directory. I copied all the website files and directories that I already had from building a website and pasted them in the /var/www/html directory. This can be seen in the screenshot below.



luke@BocaServer: /var/www/luke.tapanes.com

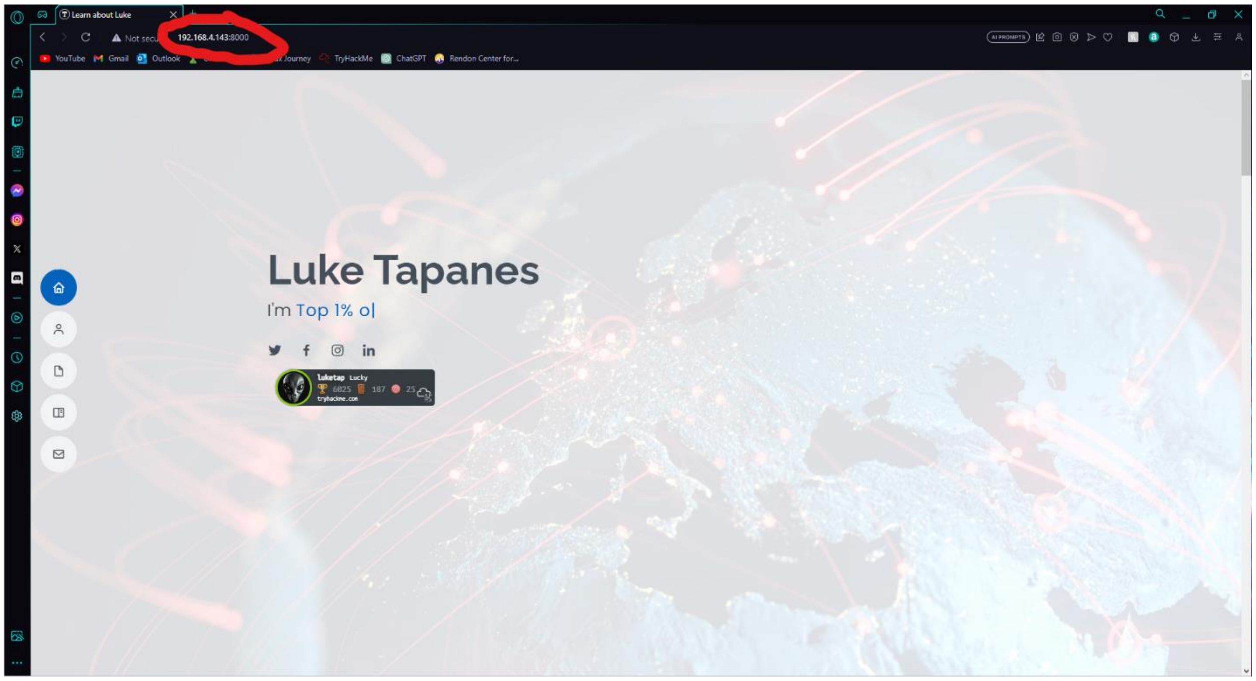
```
luke@BocaServer:/var$ cd www
luke@BocaServer:/var/www$ ls
html
luke@BocaServer:/var/www$ cd ..
luke@BocaServer:/var$ sudo cp -r /home/luke/BocaCloud/luke.tapanes.com\ version\ 2/ /www
luke@BocaServer:/var$ cd www
luke@BocaServer:/var/www$ ls
html
luke@BocaServer:/var/www$ sudo cp -r /home/luke/BocaCloud/luke.tapanes.com\ version\ 2/ www
luke@BocaServer:/var/www$ ls
html  www
luke@BocaServer:/var/www$ cd www
luke@BocaServer:/var/www/www$ ls
 ad-project.html   index.html                  Readme.txt            siem-project.html   THM.html
 assets            malicious-usb-project.html  server-project.html  'Source template'
 forms             pc-build-project.html       shell.php             test.html
luke@BocaServer:/var/www/www$ cd ..
luke@BocaServer:/var/www$ mv www luke.tapanes.com
mv: cannot move 'www' to 'luke.tapanes.com': Permission denied
luke@BocaServer:/var/www$ sudo !!
sudo mv www luke.tapanes.com
luke@BocaServer:/var/www$
luke@BocaServer:/var/www$ ls
html  luke.tapanes.com
luke@BocaServer:/var/www$ cd luke.tapanes.com/
luke@BocaServer:/var/www/luke.tapanes.com$ ls
 ad-project.html   index.html                  Readme.txt            siem-project.html   THM.html
 assets            malicious-usb-project.html  server-project.html  'Source template'
 forms             pc-build-project.html       shell.php             test.html
luke@BocaServer:/var/www/luke.tapanes.com$ sta
```

From here, I went ahead and started the Apache service with the command "`service apache2 start`." I then officially hosted the website with the command "`python3 -m http.server`." This can be seen in the screenshot below.



```
luke@BocaServer:/var/www/html/luke.tapanes.com$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.4.143 - - [28/Sep/2023 12:06:16] "GET / HTTP/1.1" 200 -
192.168.4.143 - - [28/Sep/2023 12:06:16] "GET /THM.html HTTP/1.1" 200 -
192.168.4.143 - - [28/Sep/2023 12:06:16] "GET /assets/vendor/bootstrap-icons/bootstrap-icons.css HTTP/1.1"
 200 -
192.168.4.143 - - [28/Sep/2023 12:06:16] "GET /assets/vendor/bootstrap/css/bootstrap.min.css HTTP/1.1" 200
```

The website is now live on the network! To test this, I switched to my desktop, pulled up the browser, and entered the IP address and port that the website was being hosted on. Sure enough, the website loaded flawlessly as seen in the screenshot below.



Lessons Learned

I had an absolute blast completing this project and learned a tremendous amount. I did run into some issues along the way but luckily, I troubleshooted and I successfully completed each objective. All-in-all, I achieved what I set out to do and learn and then some. This project has unlocked opportunities for many more projects to come and some were teased in the writeup. To recap this project, I successfully:

- Built the physical server.
- Flashed the operating system ISO and installed it on the server.
- Got the server up and running.
- Performed updates and patches on the server.
- Hardened the system.
- Enabled remote access for maintenance and control.
- Configured a Network-Attached Storage (NAS).
- Downloaded a hypervisor and an operating system.
- Installed Apache and hosted a webpage.

Below are a few YouTube videos that inspired and helped me with this project.

https://www.youtube.com/watch?v=e1lVV0nguZ8&t=6s

https://www.youtube.com/watch?v=wFjYzhkEBys&t=515s

https://www.youtube.com/watch?v=72D3MvPk3Xs&t=558s

https://www.youtube.com/watch?v=Wlmne44M6fQ