University of Texas at San Antonio

Unlocking Security: Building a PKI with Ubuntu and OpenSSL

Daniella Carbuccia

IS 3513: Information Assurance and Security

04 Oct. 2024

In this lab, we set up a virtual machine using Ubuntu Server and explored key concepts in managing server environments. We focused on setting up a Certificate Authority, generating self-signed certificates, and configuring SSL for secure communication. By working through tasks like cloning virtual machines and managing certificates, we gained hands-on experience with the tools and techniques crucial for cybersecurity professionals. Following the lab instructions, we installed and configured software, troubleshooted errors, and learned how these processes enhance security and system administration.

To start off the lab we had to setup our new virtual machine using Ubuntu Server. Ubuntu server was first created in 2004 alongside the desktop version by Canonical. It is a specialized version of the Ubuntu operating system that uses a command-line interface and is designed to be run on servers instead of personal computers. According to an article from Server Academy, "Ubuntu Server is optimized for running server applications and services. If you need an operating system for web hosting, file servers, database management, or enterprise-level applications, Ubuntu Server is the right choice." (Hill). Ubuntu server is ultimately a good choice if you need efficiency when running server applications. However, if you would like to use something that is more user-friendly, you would use Ubuntu Desktop. Ubuntu desktop uses a graphical user interface and acts more like a personal computer which allows web browsing, office work, etc. I have used the Ubuntu desktop in a previous class before so in my opinion it was very easy to navigate. When we were instructed to install Ubuntu server, the instructions were easy to follow and was overall a smooth experience. However, I did have to watch some Youtube videos on installation because I wasn't sure what options to pick for the settings and at one point once the installation was complete and it started booting up, it asked me to remove the CD ROM. After watching a few videos, I was able to figure it out by shutting down the machine

and going into the settings and removing the disk. From there the server booted up and I was able to update it using sudo apt update and I installed the webserver Apache onto it. According to an article from DigitalOcean, “The Apache HTTP server is the most widely used web server in the world. It provides many powerful features, including dynamically loadable modules, robust media support, and extensive integration with other popular software.” (Glass; Horcasitas). This will be helpful later on in the certificates portion of the lab.

We were then instructed to make a clone of our new virtual machine. Making a clone of your virtual machine is typically for testing changes without disrupting the original machine, and serves as a backup incase of something like a system failure happens. There are a lot of advantages to making a clone as opposed to making one from scratch such as, reducing setup times, risk-free testing, all software versions and settings are the same, and as I said before it acts as a backup in case of any failures. Now, there are some disadvantages to cloning a machine. Full cloning takes up a good amount of space, whereas if you build it from scratch, you can install only necessary components. Another disadvantage is when you want different setups on your virtual machines, because when you clone your machine, it is the same as the original so it will take more time to customize it to your liking. If I had to choose between making a clone or a whole new machine, I would pick the clone because I have lost progress many times in other classes due to something crashing and it helps knowing there is a backup of my original system. I also like that it is faster to setup than making a whole new machine. To make the clone I right clicked on the original machine and chose clone. From there I set it up and chose “The current state in the virtual machine” to get a full replica and followed the rest of the steps to clone it. For the most part it cloned successfully but I had a few hiccups. I tried to start up the machine a few times and it would completely freeze up to where I had to close out of VMware completely.

What I ended up doing to resolve this was delete both machines and double check the setup was done correctly on both, but I noticed it was still happening. So, to resolve this issue, I started up my original machine and then once I was signed in, I powered on the clone, and it worked. My overall takeaway from this part of the lab was how different Ubuntu server looked from Ubuntu desktop. When I opened up the server, I thought I downloaded the wrong program because there was no GUI like I expected. Now I understand what functions the server has compared to the desktop, and it has a lot of cool features that the desktop version doesn't have.

After the setup was complete, I began following the lab instructions on my new virtual machine and clone. The first part we worked on was configuring the certificate authority. According to an article from TechTarget, "A certificate authority (CA) is a trusted entity that issues Secure Sockets Layer (SSL) certificates. These digital certificates are data files used to cryptographically link an entity with a public key. Web browsers use them to authenticate content sent from web servers, ensuring trust in content delivered online." (Awati). So to configure it, I updated the packages on my CA virtual machine and installed OpenSSL using sudo apt install openssl. Once that was completed, I created a directory for my CA using the mkdir command and from there created two more for private keys and certificates. I then created an empty index file using the touch command, and then a serial file which uses the echo command. If the echo command is done correctly, anytime a certificate is created, it will save into that file incrementing up by 1 on the file name. The next step in the lab is to generate a private key using the command: openssl genrsa -out private/ca.key 4096. This private key is essential in proving authenticity as it is used to sign certificates. I then created the self-signed certificate with that private key and entered the information in that was requested. I ran into a road bump on this step that I wasn't aware of until a few steps later in the process, but I will explain after a few steps. I

then made a backup of my openssl.cnf file using the cp command and then used sudo nano to edit the file. We were instructed to change the base directory to direct to our CA directory and update the certificate to read ca.crt which is where to find the self-signed certificate. We were also instructed to edit the path of the private key to $dir/private/ca.key. Lastly, we were told to add one last line and that is the crlDistributionPoints line. This line ensures that clients accessing the CA's I have issued can also check for their revocation status. I wasn't sure at first what a lot of these commands were doing while I was doing the actual steps on my machine, but when I was confused, I would look up what certain commands did or what the file paths were for and once we used those files, I understood what they did.

I moved onto the issuing certificates portion of the lab. This part of the lab starts on the clone WS machine we made earlier, and I generated a private key with: openssl genrsa -out server.key 2048 and created a certificate signing request (CSR) for a server. For this part I used openssl req -new -key server.key -out server.csr and filled in the details. After doing that part I transferred the server.csr to my CA VM using secure copy: scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2. On the [OPTION] portion I entered my file: server.csr, in [user@SRC_HOST:]file1 I put my username and IP address of my CA VM, and on the [user@]DEST_HOST:]file2 I put the path I want the file to go in. The last step of this part of the lab is once the transfer was complete, I went back to my CA VM and signed the CSR to issue the certificate. Remember earlier how I said I had issues with the information I inputted when I created the self-signed certificate? Well, when I tried to sign the CSR, it gave me an error regarding the organization name, something along the lines of it not matching. I had trouble finding resolutions, but what I did to fix it was recreate both certificates on both machines and made sure to leave the default options for the organization name. Once I tried to sign the CSR

again, it finally worked. This portion of the lab was pretty confusing, and I noticed a few other students were stuck as well but I learned that sometimes it just means I have to restart some steps from the beginning and usually it fixes the problem.

The last step we did before testing the certificates was setting up the webserver with HTTPS. To do this, I went back to my WS VM and enable the SSL module using sudo a2enmod ssl. I then restarted the webserver to see the changes. We then needed to prepare the SSL certificate and key, and on this portion, I had a lot of trouble. We were first instructed to copy the server.crt to the WS machine, when I attempted to secure copy it there, I kept getting error messages saying I don't have permission. I made several attempts, including using the sudo command and still nothing. After thinking about it I figured out that I could transfer it to the home directory and from there use the cp command to the right file and it worked. Instead of having the same issues like with the server.crt, when we were told to transfer the server.key file to the WS machine, I used the same method of transferring to the home directory first. We then were instructed to edit the SSL config. File using sudo nano to direct to our server.crt and server.key files. After that, I enabled the SSL site and restarted Apache using similar steps to what we used at the beginning of the lab. From here, we can now test the certificates.

The second to last step of the lab is testing the secure connection. On my own computer outside of the virtual machine, I searched up my WS machine using: https://[VM ip address] on Google Chrome. At first, it showed me an error page saying the site wasn't secure, so I was a little confused. However, when I allowed the webpage to let me through, it took me to the Apache2 default page like in the example in the lab. I also checked the SSL certificate, and I verified the information such as the common name, validity period and certificate chain and everything matched up to the results it should show. I thought this portion of the lab was really

cool because I didn't know that you can search up anything about your virtual machine outside of it but I liked that I could see the work I did show up on a webpage. I believe learning this skill is essential in Cyber Security because it is good to understand how encryption ensures confidentiality.

The last step in the lab is revoking certificates. On my CA machine, I used the command: openssl ca -revoke newcerts/1000.pem -config along with the file path to revoke the certificate in the openssl.cnf file. I first had to go to the file by using the cd command to see what certificate I had, and I only saw the 1000.pem file so I chose to revoke it. I then followed the instructions to create a Certificate Revocation List. From there I created a directory for the CRL and copied it to the new directory. Finally, we were told to test the revoked certificate by going back to the website to see what happens. This part did confuse me because when I went back to the website and refreshed it still showed the Apache2 default page. I couldn't get it to show me the results, but I did get confirmation that the certificate I had was revoked. I would say this part was easy to get going but towards the end I had that roadblock, and I couldn't figure out where the issue was. I retraced my steps, but I have no idea where it happened. Overall, I did like learning how I would revoke a certificate if I needed to, but I will just have to be more careful in the steps along the way. I plan to redo this lab on my own time to see if I can start over from scratch.

Completing this lab helped me understand how to set up and manage a secure server environment using self-signed certificates and SSL. It emphasized the importance of certificates for secure communication and gave insight into managing virtual machines and troubleshooting common issues. By following the steps closely, I improved my practical knowledge of how certificates are issued and revoked, which is vital for maintaining network security.

Works Cited

"About the Ubuntu Project." *Ubuntu*, ubuntu.com/about. Accessed 4 Oct. 2024.

Awati, Rahul, and Peter Loshin. "What Is a Certificate Authority (CA)?" *Security*, TechTarget, 1 Sept. 2021, www.techtarget.com/searchsecurity/definition/certificate-authority.

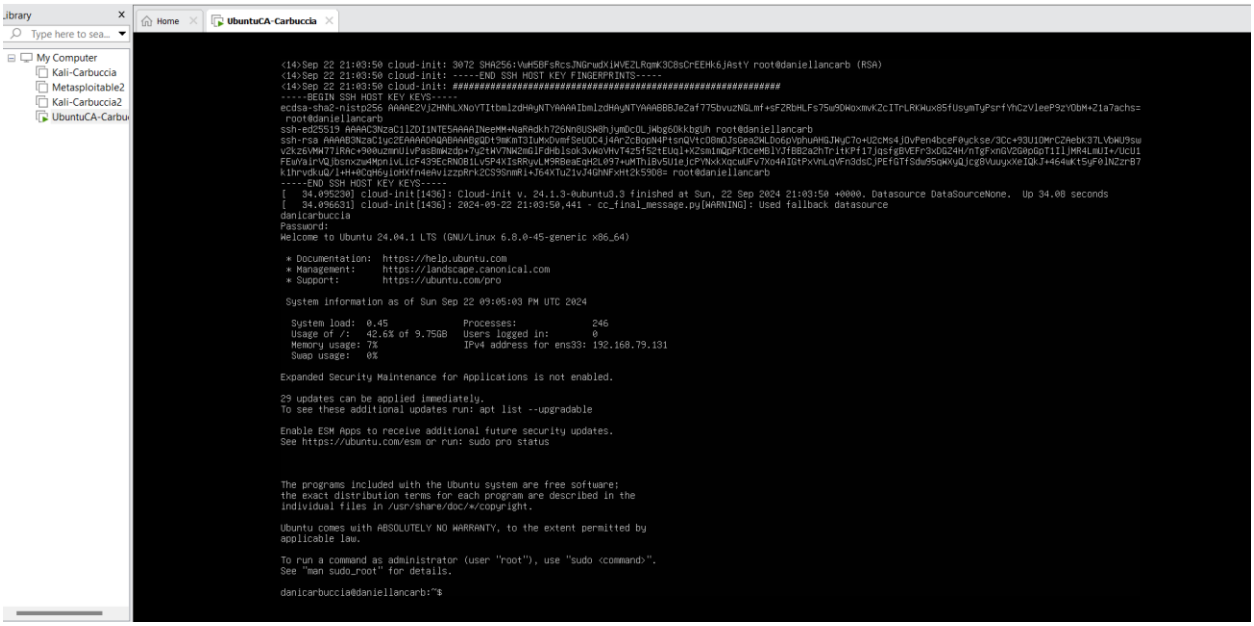
Gitlan, Dionisie. "What Is Openssl and How Does It Work?" *SSL Dragon*, 5 Apr. 2024, www.ssldragon.com/blog/what-is-openssl/.

Glass, Erin, and Jeanelle Horcasitas. "How to Install the Apache Web Server on Ubuntu." *DigitalOcean*, 26 Apr. 2022, www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-22-04.

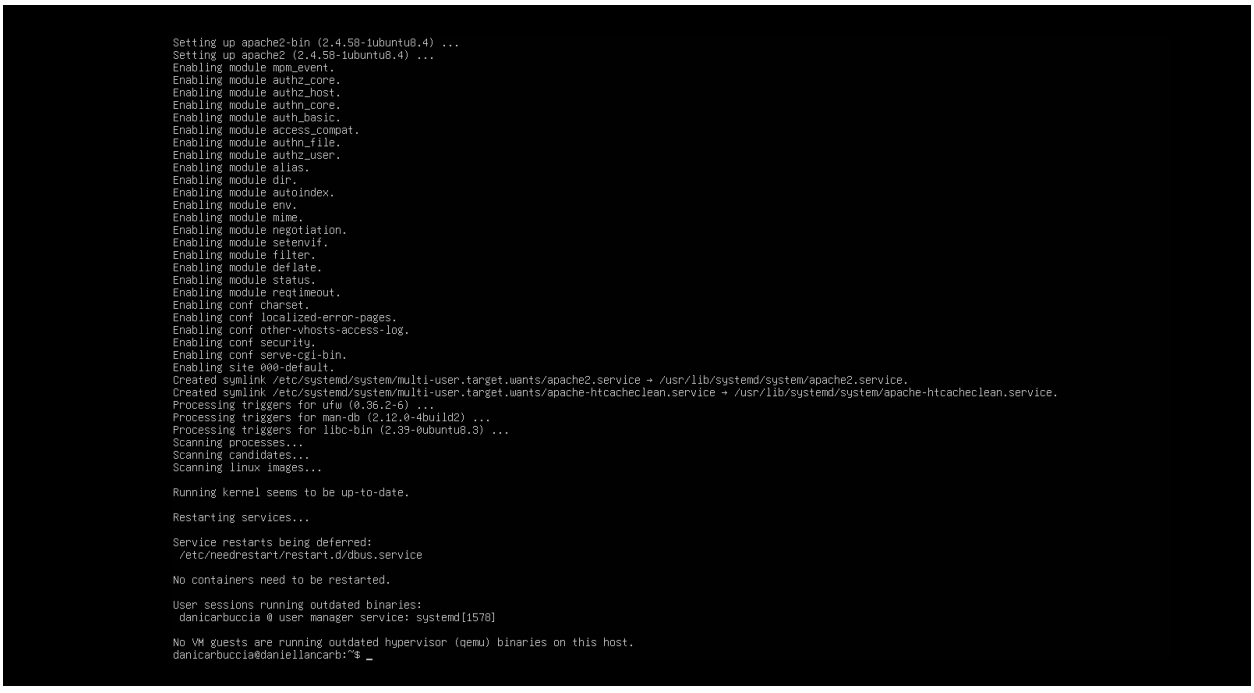
Hill, Paul. "Ubuntu Server VS Desktop." *Server Academy*, 9 June 2024, www.serveracademy.com/blog/ubuntu-server-vs-desktop/.

r2schools. "How to Download and Install Ubuntu 24.04 LTS on VMWare | Install Ubuntu Server 24.04 on VMWare 2024." *YouTube*, YouTube, 26 Apr. 2024, www.youtube.com/watch?v=WYd3S9Ozajs.

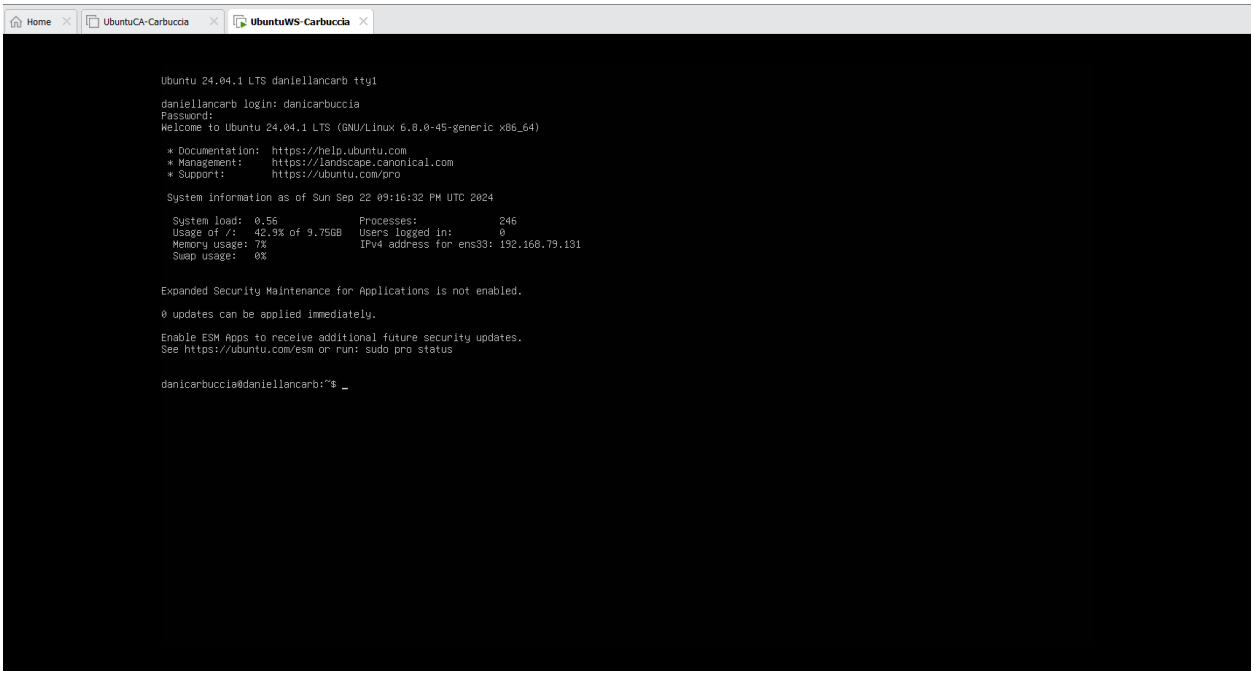
Ubuntu installed, however I had some confusion installing because it failed and told me to disconnect CD ROM, but I removed it through the settings and it worked.



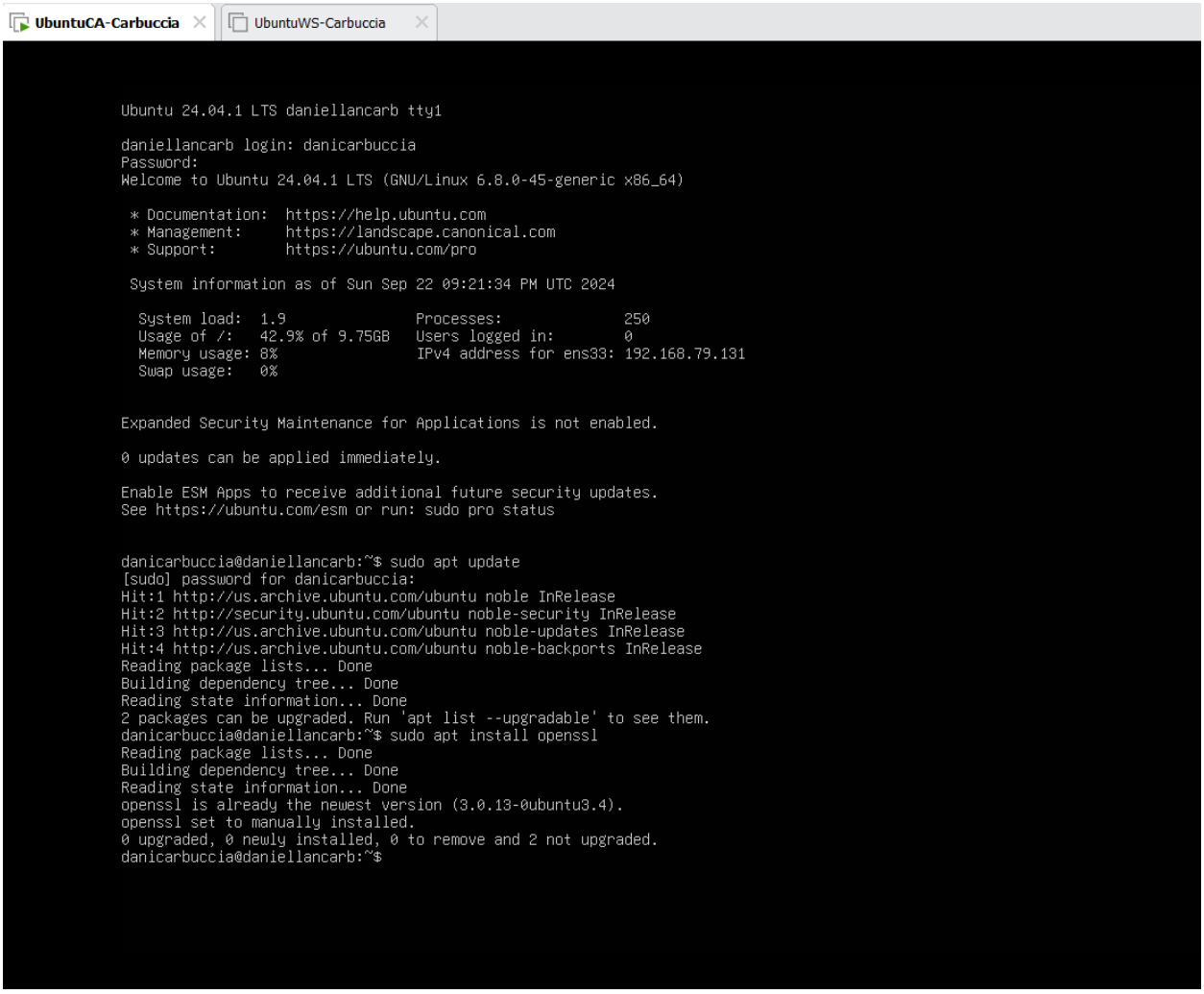
Install of apache2:



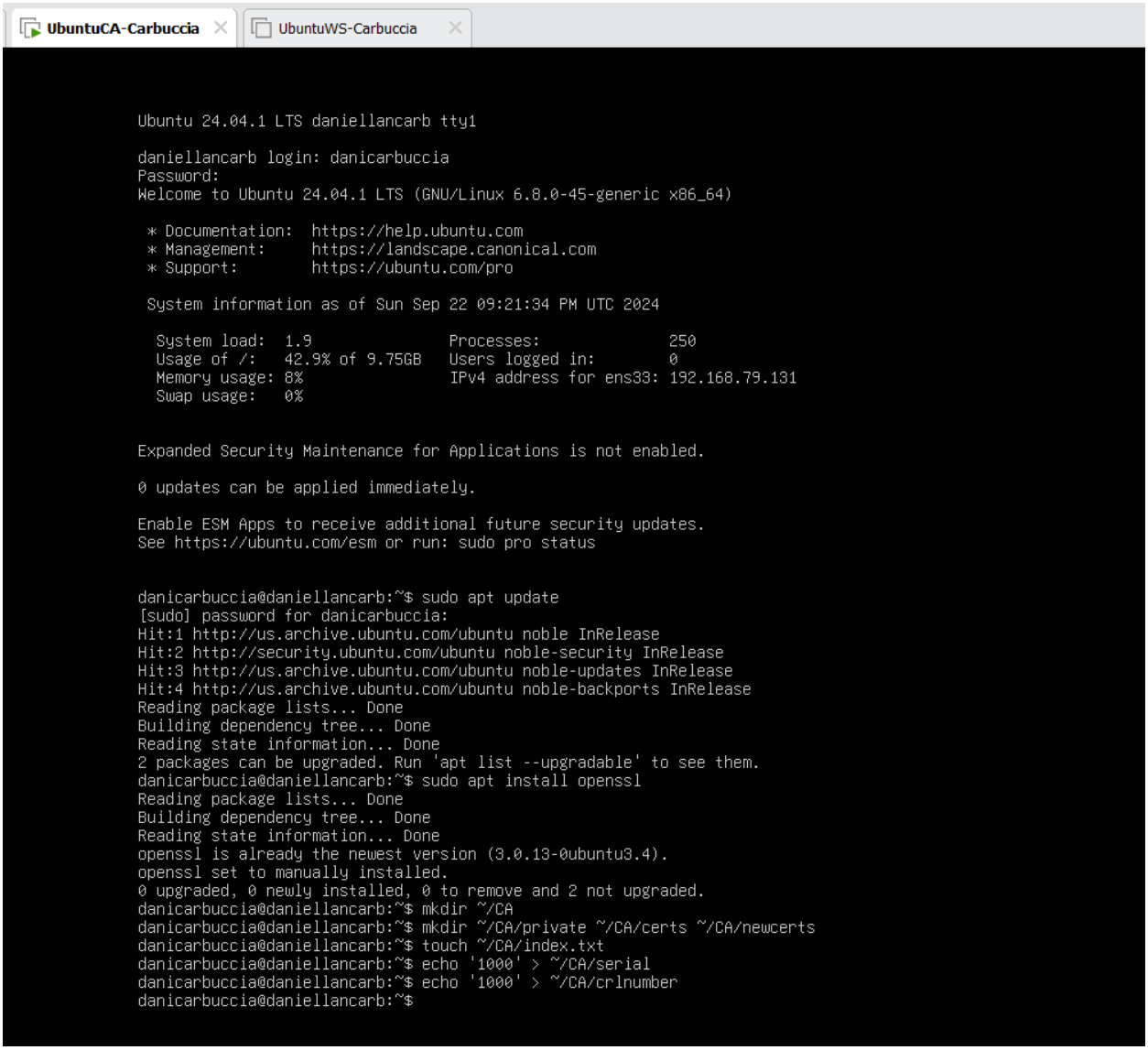
Clone of VM:


Home
UbuntuCA-Carbuccia
UbuntuWS-Carbuccia
Ubuntu 24.04.1 LTS daniellancarb tty1
daniellancarb login: danicarbuccia
Password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-45-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Sun Sep 22 09:16:32 PM UTC 2024
System load: 0.56
Usage of /: 42.9% of 9.75GB
Memory usage: 7%
Swap usage: 0%
Processes: 246
Users logged in: 0
IPv4 address for ens33: 192.168.79.131
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
danicarbuccia@daniellancarb:~$

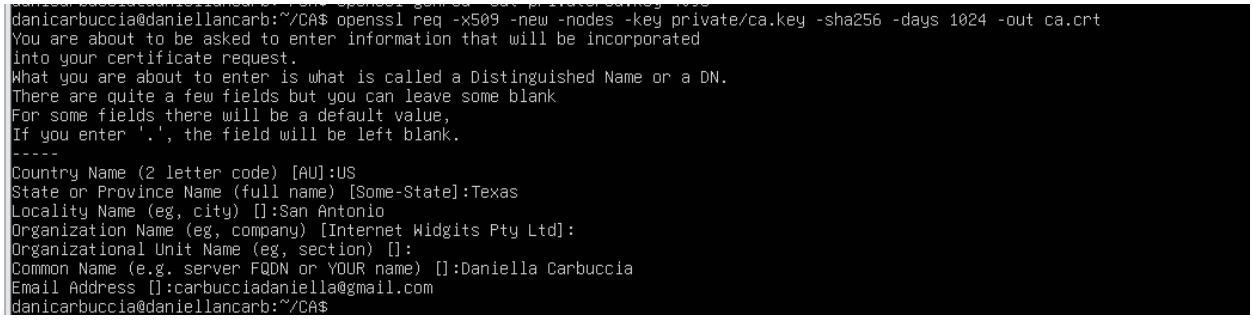
Install of openssl:



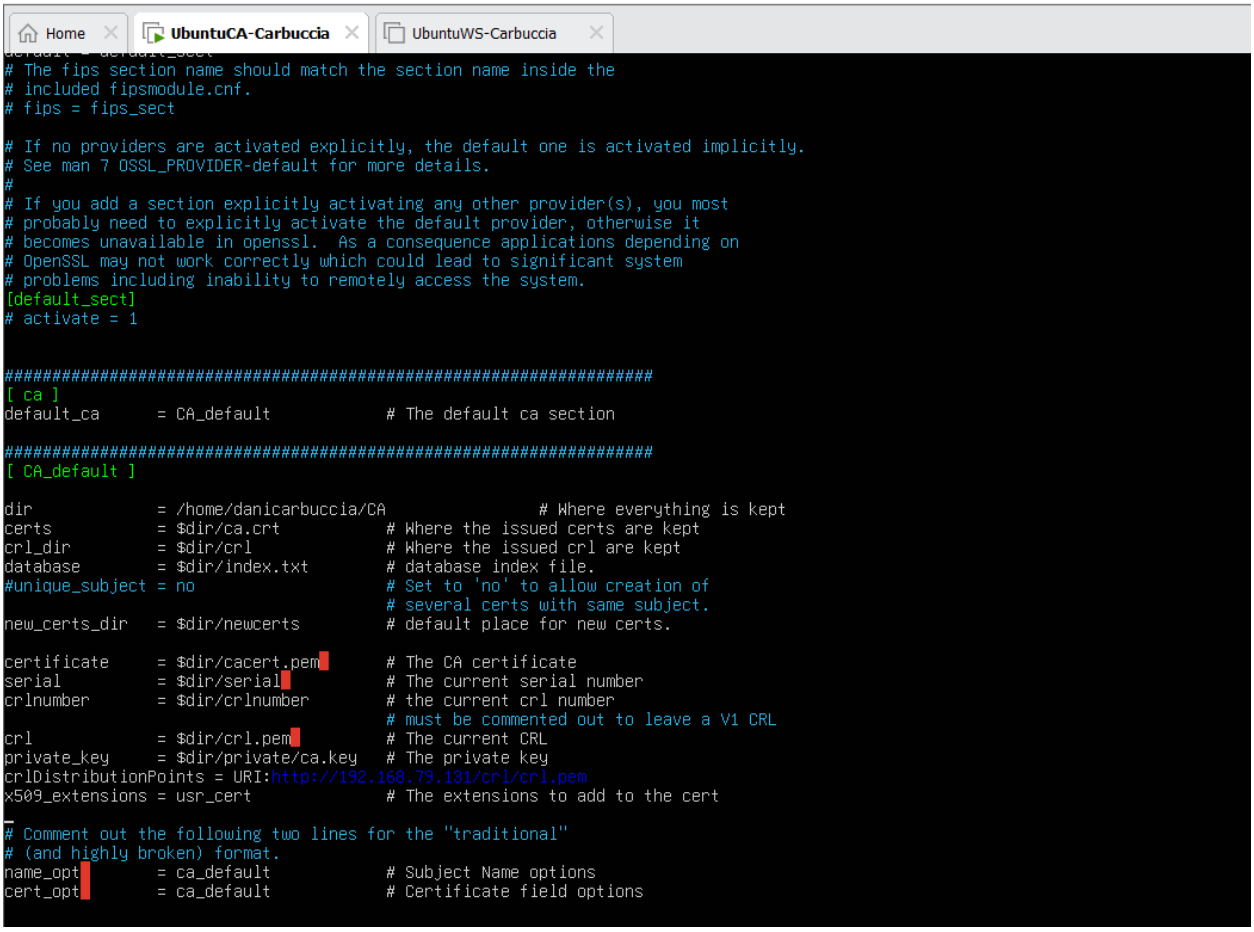
Prepare CA directory structure:



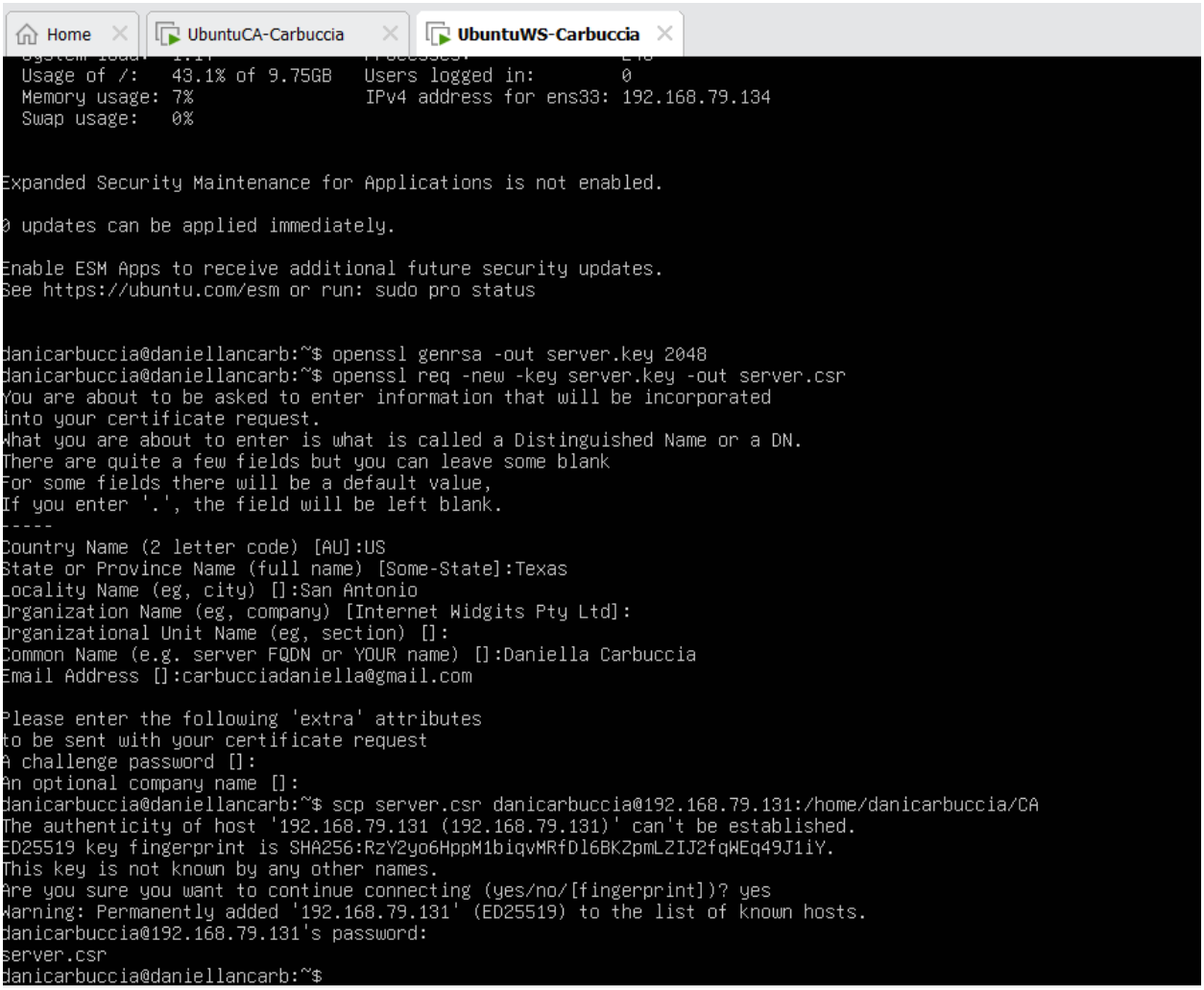
Create CA's self-signed certificate:



Modify /etc/ssl/openssl.cnf:



Created server.csr in ubuntuWS-Carbuccia:



Signed certificate:

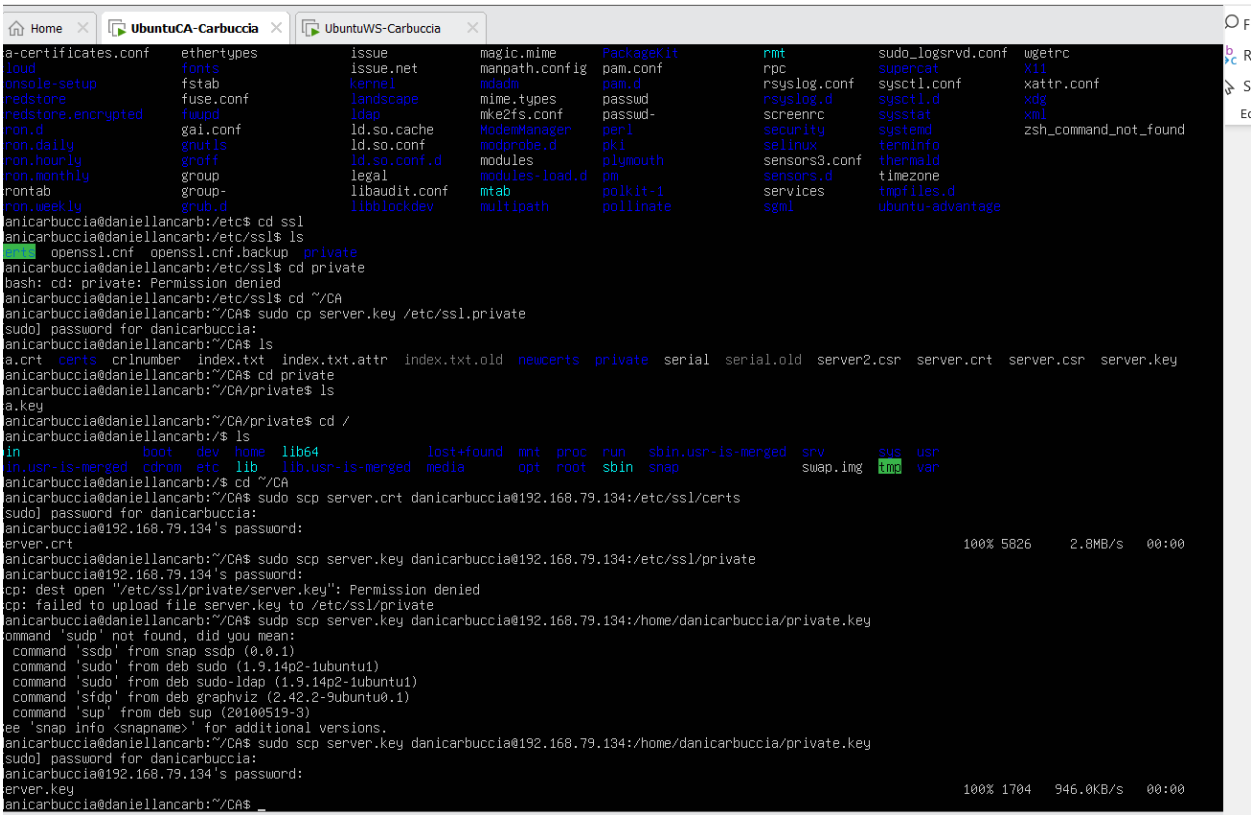
```
danicarbuccia@daniellancarb:~/CA$ openssl ca -in server2.csr -out server.crt -cert ca.crt -keyfile private/ca.key -config /et
Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
The organizationName field is different between
CA certificate (Internet Widgits Pty Ltd) and the request (Daniella Carbuccia)
danicarbuccia@daniellancarb:~/CA$ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile private/ca.key -config /etc
Using configuration from /etc/ssl/openssl.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4096 (0x1000)
        Validity
            Not Before: Sep 22 22:58:14 2024 GMT
            Not After : Sep 22 22:58:14 2025 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = Texas
            organizationName          = Internet Widgits Pty Ltd
            commonName                = Daniella Carbuccia
            emailAddress              = carbucciadaniella@gmail.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Subject Key Identifier:
                C3:73:B8:71:5A:5F:D1:1F:85:E6:88:CF:4B:A7:51:0E:65:09:88:30
            X509v3 Authority Key Identifier:
                8D:EE:DE:C4:9A:99:FF:5D:65:33:B1:21:AA:94:C1:93:6F:5C:3C:FB
Certificate is to be certified until Sep 22 22:58:14 2025 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
danicarbuccia@daniellancarb:~/CA$ _
```

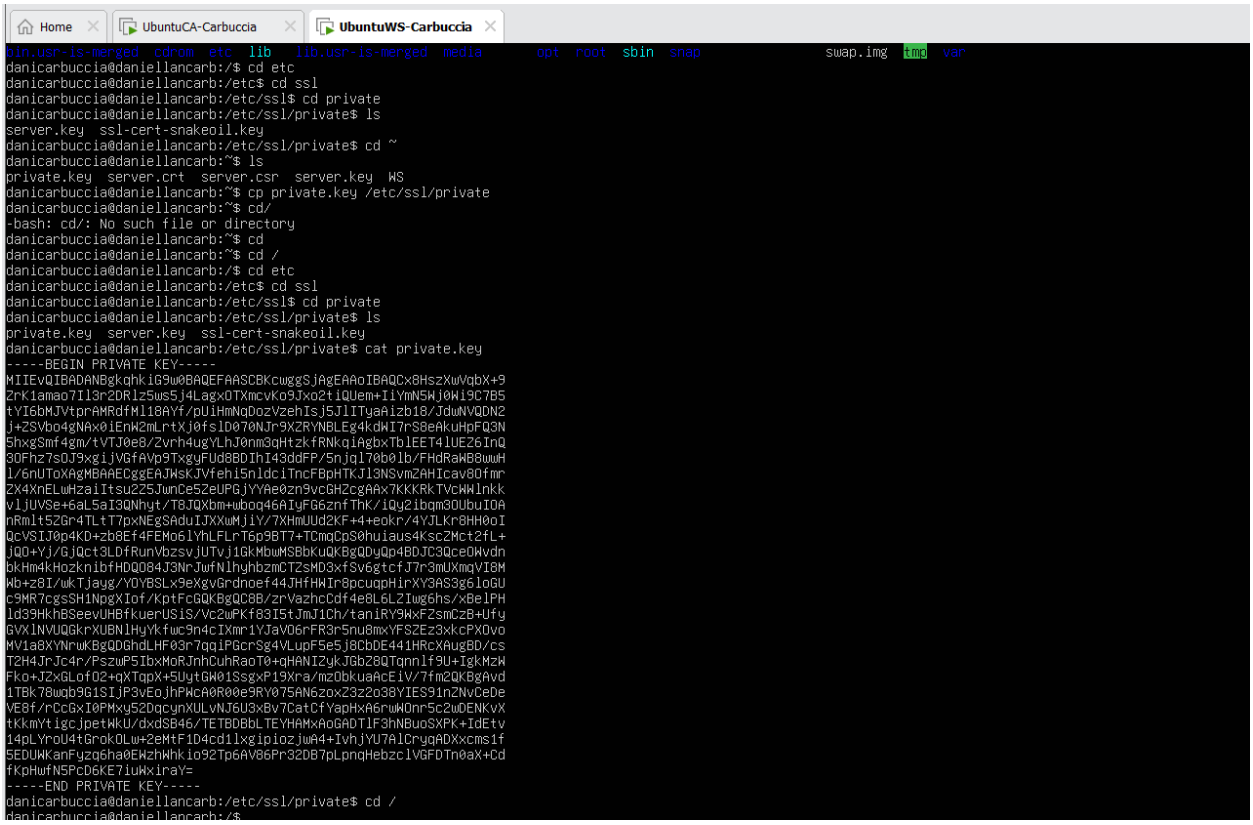
Server.crt copied to /etc/ssl/certs:

```
AffirmTrust_Commercial.pem                                  OISTE_WISeKey_Global_Root_GC_CA.pem
AffirmTrust_Networking.pem                                  QuoVadis_Root_CA_1_G3.pem
AffirmTrust_Premium_ECC.pem                                 QuoVadis_Root_CA_2_G3.pem
AffirmTrust_Premium.pem                                     QuoVadis_Root_CA_2.pem
Amazon_Root_CA_1.pem                                        QuoVadis_Root_CA_3_G3.pem
Amazon_Root_CA_2.pem                                        QuoVadis_Root_CA_3.pem
Amazon_Root_CA_3.pem                                        Sectigo_Public_Server_Authentication_Root_E46.pem
Amazon_Root_CA_4.pem                                        Sectigo_Public_Server_Authentication_Root_R46.pem
ANF_Secure_Server_Root_CA.pem                               Secure_Global_CA.pem
Atos_TrustedRoot_2011.pem                                   SecureSign_RootCA11.pem
Atos_TrustedRoot_Root_CA_ECC_TLS_2021.pem                   SecureTrust_CA.pem
Atos_TrustedRoot_Root_CA_RSA_TLS_2021.pem                   Security_Communication_ECC_RootCA1.pem
Autoridad_de_Certificacion_Firmaprofesional_CIF_A62634068.pem  Security_Communication_RootCA2.pem
b0e59380.0                                                  Security_Communication_RootCA3.pem
b1159c4c.0                                                  Security_Communication_Root_CA.pem
b433981b.0                                                  server.crt
b66938e9.0                                                  ssl-cert-snakeoil.pem
b727005e.0                                                  SSL.com_EV_Root_Certification_Authority_ECC.pem
b7a5b843.0                                                  SSL.com_EV_Root_Certification_Authority_RSA_R2.pem
b81b93f0.0                                                  SSL.com_Root_Certification_Authority_ECC.pem
Baltimore_CyberTrust_Root.pem                               SSL.com_Root_Certification_Authority_RSA.pem
bf53fb88.0                                                  SSL.com_TLS_ECC_Root_CA_2022.pem
BJCA_Global_Root_CA1.pem                                    SSL.com_TLS_RSA_Root_CA_2022.pem
BJCA_Global_Root_CA2.pem                                    Starfield_Class_2_CA.pem
Buypass_Class_2_Root_CA.pem                                 Starfield_Root_Certificate_Authority_-_G2.pem
Buypass_Class_3_Root_CA.pem                                 Starfield_Services_Root_Certificate_Authority_-_G2.pem
c01eb047.0                                                  SwissSign_Gold_CA_-_G2.pem
c28a8a30.0                                                  SwissSign_Silver_CA_-_G2.pem
ca6e4ad9.0                                                  SZAFIR_ROOT_CA2.pem
ca-certificates.crt                                         Telia_Root_CA_v2.pem
CA_Disig_Root_R2.pem                                        TeliaSonera_Root_CA_v1.pem
cbf06781.0                                                  TrustAsia_Global_Root_CA_G3.pem
cc450945.0                                                  TrustAsia_Global_Root_CA_G4.pem
cd58d51e.0                                                  Trustwave_Global_Certification_Authority.pem
cd8c0d63.0                                                  Trustwave_Global_ECC_P256_Certification_Authority.pem
ce5e74ef.0                                                  Trustwave_Global_ECC_P384_Certification_Authority.pem
Certainly_Root_E1.pem                                       T-TeleSec_GlobalRoot_Class_2.pem
Certainly_Root_R1.pem                                       T-TeleSec_GlobalRoot_Class_3.pem
Certigna.pem                                                TUBITAK_Kamu_SM_SSL_Kok_Sertifikasi_-_Surum_1.pem
Certigna_Root_CA.pem                                        TunTrust_Root_CA.pem
certSIGN_Root_CA_G2.pem                                     TWCA_Global_Root_CA.pem
certSIGN_ROOT_CA.pem                                        TWCA_Root_Certification_Authority.pem
Certum_EC-384_CA.pem                                        UCA_Extended_Validation_Root.pem
Certum_Trusted_Network_CA_2.pem                             UCA_Global_G2_Root.pem
Certum_Trusted_Network_CA.pem                               USERTrust_ECC_Certification_Authority.pem
Certum_Trusted_Root_CA.pem                                  USERTrust_RSA_Certification_Authority.pem
CFCA_EV_ROOT.pem                                            vTrus_ECC_Root_CA.pem
CommScope_Public_Trust_ECC_Root-01.pem                      vTrus_Root_CA.pem
CommScope_Public_Trust_ECC_Root-02.pem                      XRamp_Global_CA_Root.pem
danicarbuccia@daniellancarb:/etc/ssl/certs$ _
```

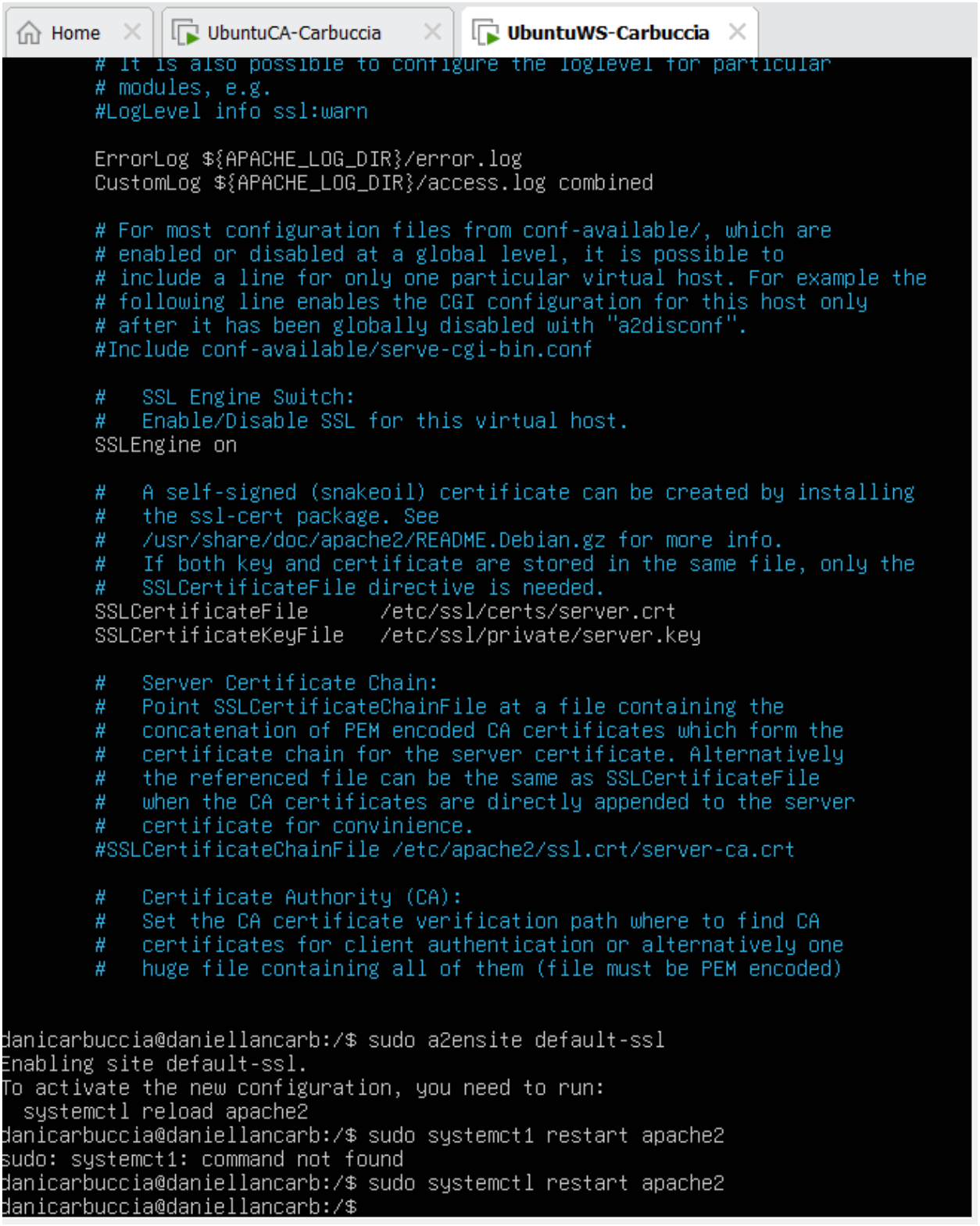
Had trouble sending a copy to the direct path in my WS server due to permission error but was able to transfer it to the home path and from there copy it to the correct directory:



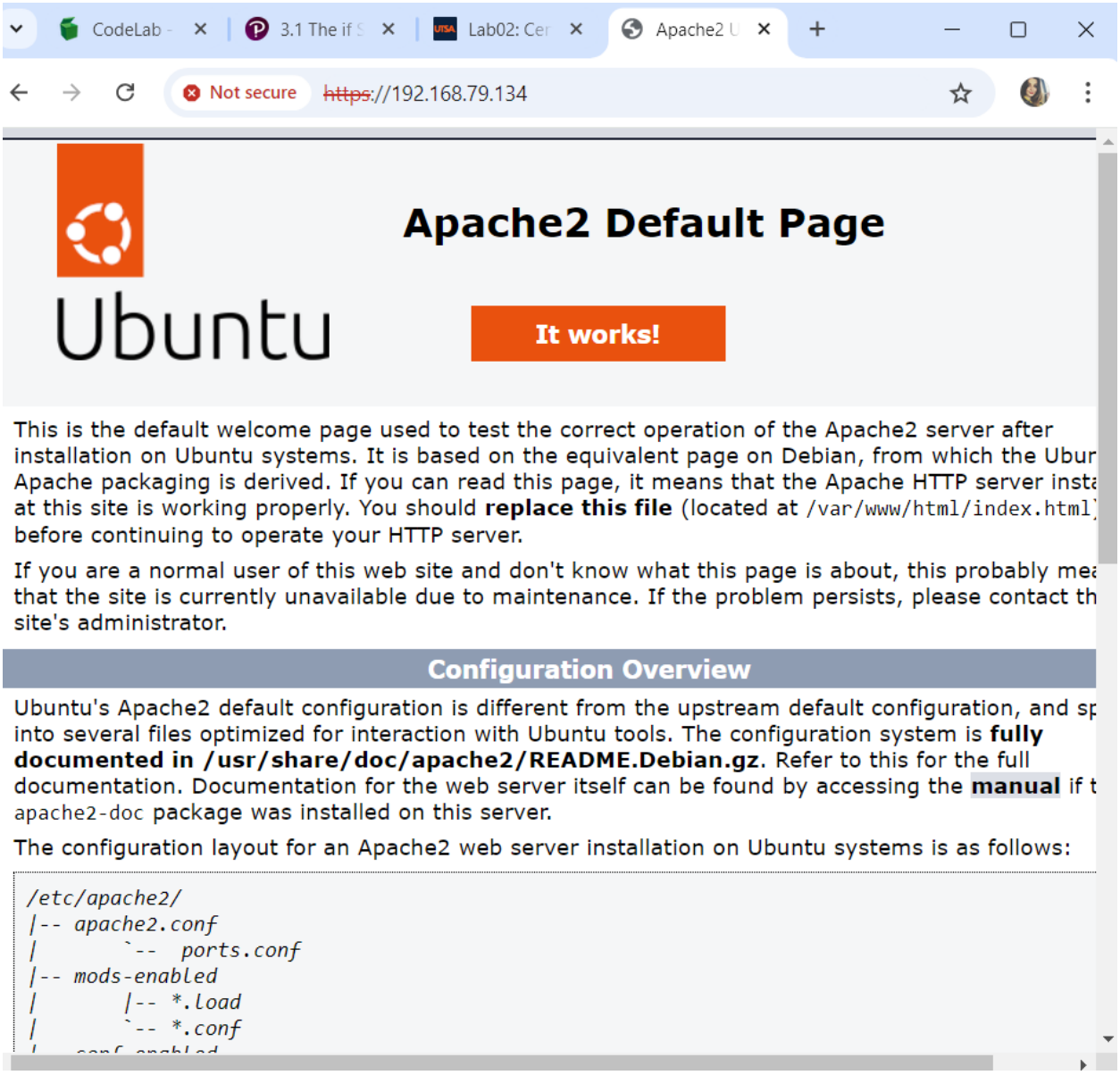
Screenshot of private.key it correct directory:



Configure SSL on webserver:



Apache2 searched with ip address (3 images):



Certificate Viewer: Daniella Carbuccia

General | Details

Issued To

Common Name (CN)	Daniella Carbuccia
Organization (O)	Internet Widgits Pty Ltd
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	Daniella Carbuccia
Organization (O)	Internet Widgits Pty Ltd
Organizational Unit (OU)	<Not Part Of Certificate>

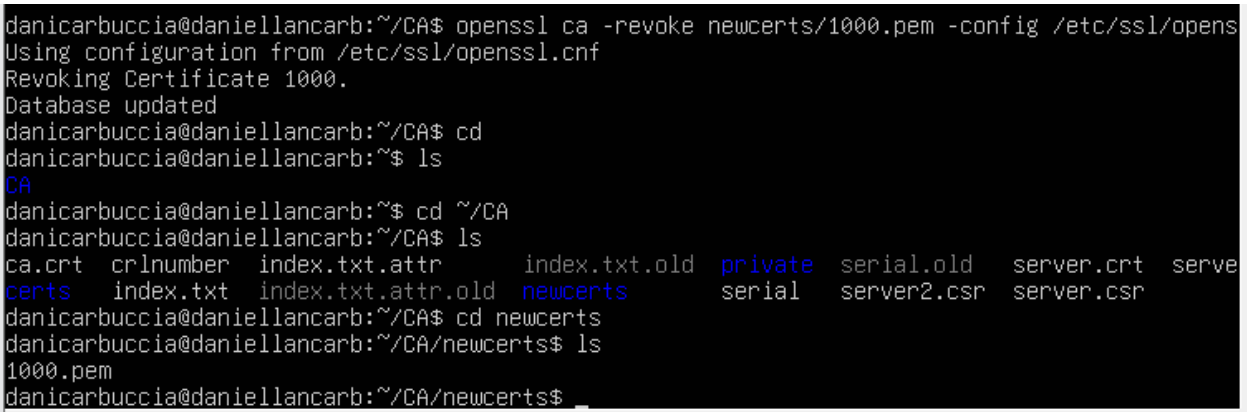
Validity Period

Issued On	Sunday, September 22, 2024 at 5:58:14 PM
Expires On	Monday, September 22, 2025 at 5:58:14 PM

SHA-256 Fingerprints

Certificate	a9174d8a4990055b6a77458fe9d182be8c3889e3fbe8e119fc0777aa801ad309
Public Key	8d70111d204c8eb55c4f24a5fd16a90c9442128aa98f6f9280f6b099ac4f6993

Revoking certificate (2 images):



SSL certificate: