

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.

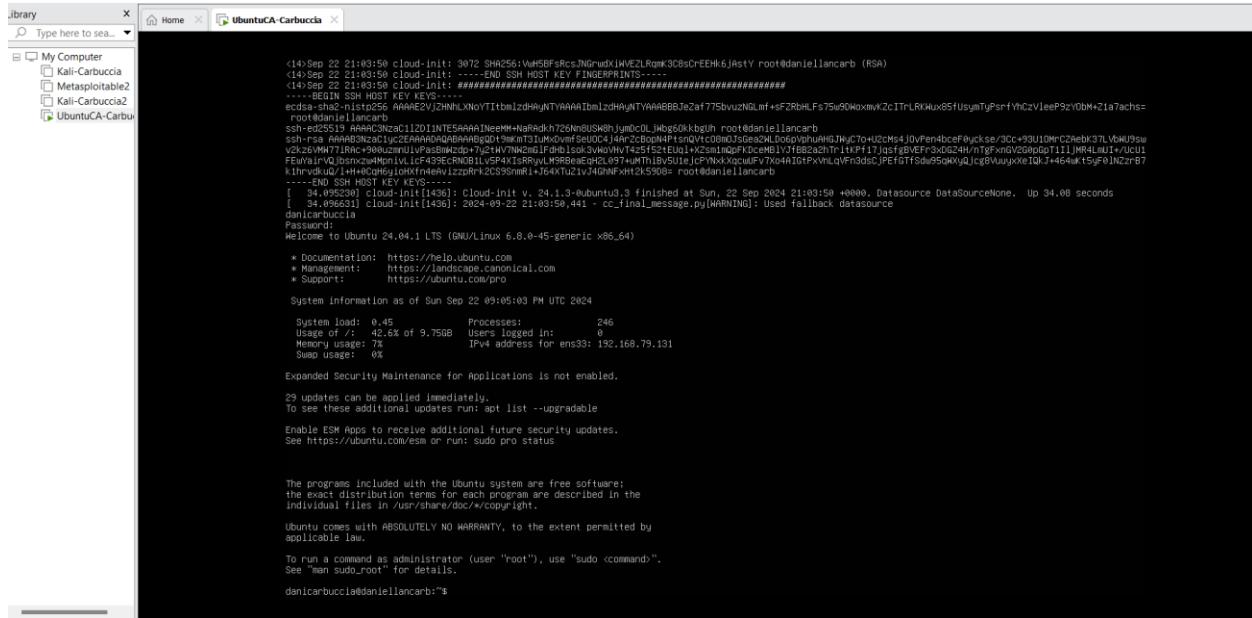
Gitlan, Dionisie. “What Is Openssl and How Does It Work?” *SSL Dragon*, 5 Apr. 2024, www.ssldragon.com/blog/what-is-openssl/.

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.



```

Library Type here to search... Home UbuntuCA-Carbuccia
My Computer
  Kali-Carbuccia
  Metasploitable2
  Kali-Carbuccia2
  UbuntuCA-Carbu
  ed5519 AAAHACnzcaci120iINTE5AAAIneem+NaEdkhtZ6Km8u8hJymc0LJhg50kkgjh root@danielancarb
  ssh-rsa AAAHABJ4c2EAAQABRgq19mkT3Iu0vfmfSeUDCJAAnZcbpH4tsnQVic0d00JseacWLDo5pghuHhByC7o46CMs4 [REDACTED] 03c+93U10rCfekcGTLV0hPsw
  FEAvwirQJbcnzwMpv1vJcf439EcRNB0LLyG4Xt5RrvuM938ewaCek0L974uMThBvSutieJcPvnxxXorufV7x0d16PxvnqlqFn9dc5FET671S9e959k0qjlg89vuyxxtQk3464ukt9jP9jNz2r87
  k1hrvdkuU1h+Hc6g61onXfn4neAvizzPrn2CS5Smrl+Jc4XTu2iV4GnfXh2k590s+ root@danielancarb
  -----END SSH HOST KEY KEYS-----
  [ 34.096339] cloud-init[1468]: Cloud-init v. 24.1.3-ubuntu0.3 finished at Sun, 22 Sep 2024 21:03:50 +0000. Datasource DataSourceNone. Up 34.00 seconds
  [ 34.096339] cloud-init[1468]: 2024-09-22 21:03:56,441 + cc_final_message.py[KRNINQ]: Used fallback datasource
  danielancarb
  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 09:05:03 PM UTC 2024

  System load: 0.45 Processes: 246
  Usage of /: 22.6% of 9.75GB Users logged in: 0
  Memory usage: 76% IPv4 address for ens3: 192.168.79.131
  Swap usage: 0%

  Expanded Security Maintenance for Applications is not enabled.

  29 updates can be applied immediately.
  To see these additional updates run: apt list --upgradable

  Enable ESM Apps to receive additional future security updates.
  See https://ubuntu.com/esm or run: sudo pro status

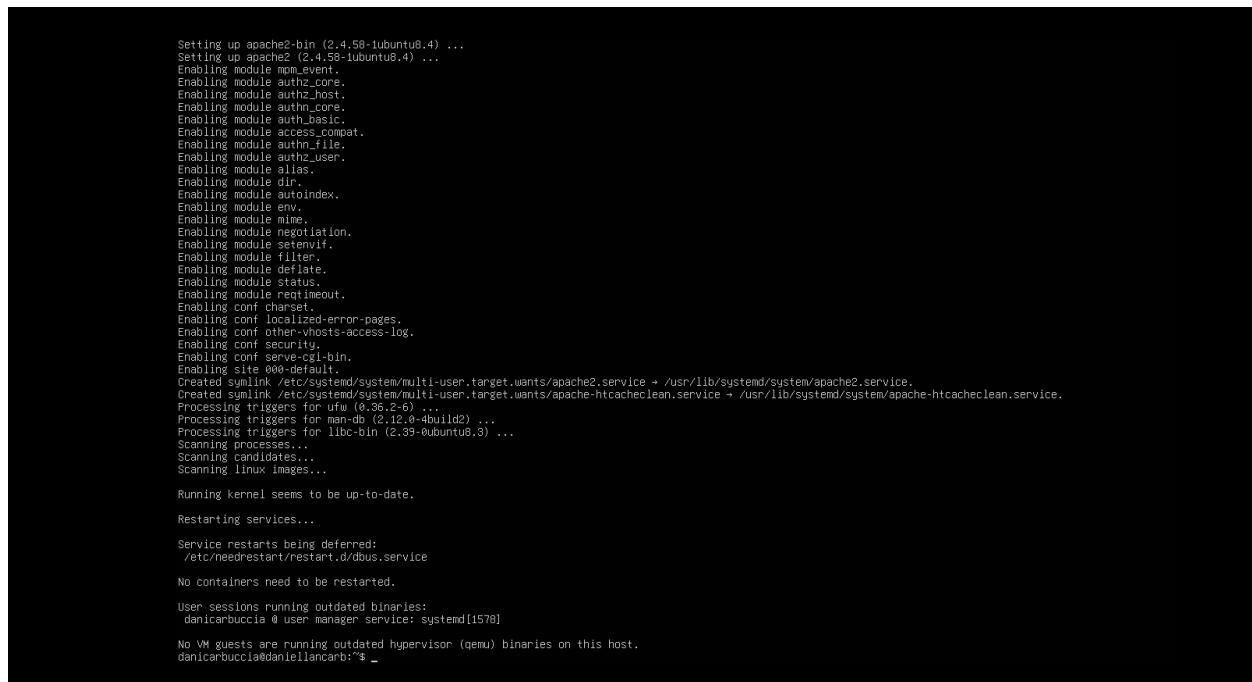
  The programs included with the Ubuntu system are free software;
  the exact distribution terms for each program are described in the
  individual files in /usr/share/doc/<copyright>.

  Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
  applicable law.

  To run a command as administrator (user "root"), use "sudo <command>".
  See "man sudo_root" for details.

  danielancarb@danielancarb:~$
```

Install of apache2:



```

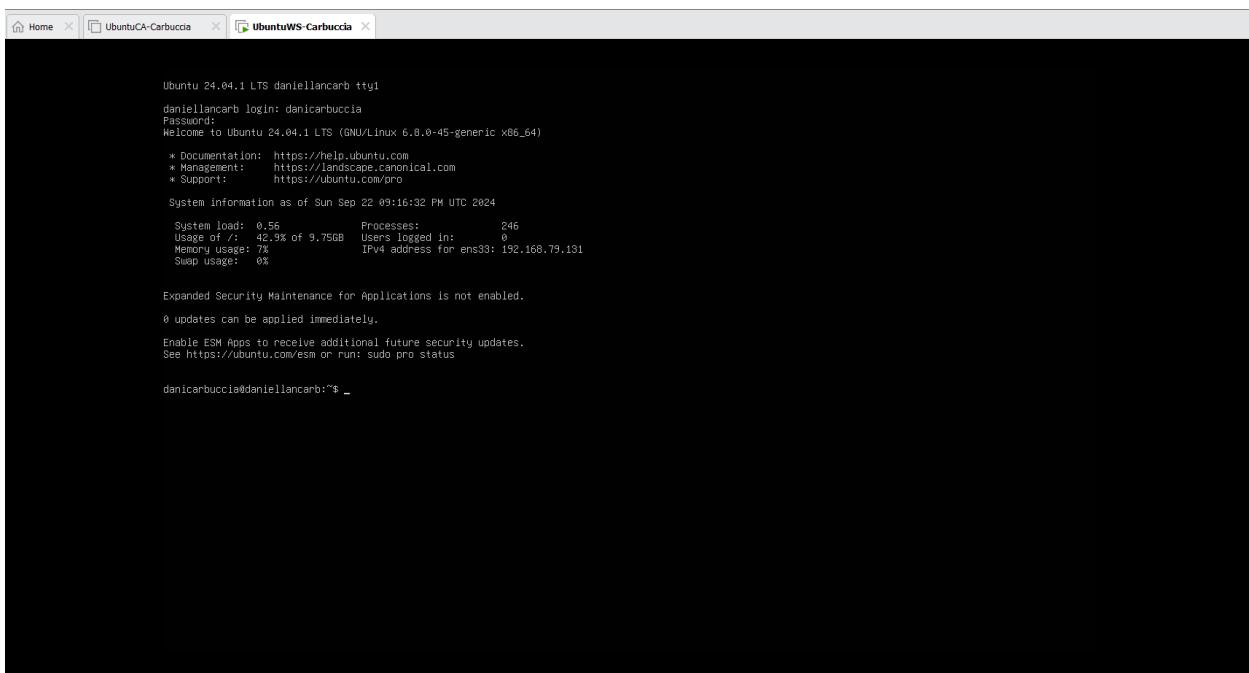
Setting up apache2-bin (2.4.58-1ubuntu0.4) ...
Setting up apache2 (2.4.58-1ubuntu0.4) ...
  Enabling module mpm_event.
  Enabling module authz_core.
  Enabling module authz_host.
  Enabling module authn_core.
  Enabling module authn_basic.
  Enabling module access_compat.
  Enabling module authn_file.
  Enabling module authz_user.
  Enabling module alias.
  Enabling module dir.
  Enabling module autoindex.
  Enabling module env.
  Enabling module mime.
  Enabling module negotiation.
  Enabling module setenvif.
  Enabling module deflate.
  Enabling module deflate.
  Enabling module status.
  Enabling module rewrite.
  Enabling conf charset.
  Enabling conf localized-error-pages.
  Enabling conf other-vhosts-access-log.
  Enabling conf security.
  Enabling conf serve-cgi-bin.
  Enabling sites-available.
  Creating symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.
  Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /usr/lib/systemd/system/apache-htcacheclean.service.
  Processing triggers for ufw (0.36.2-6) ...
  Processing triggers for man-db (2.12.0-4build2) ...
  Processing triggers for liblc-bin (2.39-0ubuntu0.3) ...
  Scanning processes...
  Scanning candidates...
  Scanning linux images...

  Running kernel seems to be up-to-date.

  Restarting services...
  Service restarts being deferred:
  /etc/needrestart/restart.d/dbus.service
  N containers need to be restarted.

  User sessions running outdated binaries:
  danielancarb@danielancarb:~$_

  No VM guests are running outdated hypervisor (qemu) binaries on this host.
  danielancarb@danielancarb:~$ _
```

Clone of VM:

The screenshot shows a terminal window titled "UbuntuWS-Carbuccia" with the following content:

```
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: 0.58      Processes: 246
Usage of /: 22.3% of 9.75GB  Users logged in:
Memory usage: 7%
Swap usage: 0%

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:

The screenshot shows a terminal window titled "UbuntuCA-Carbuccia" with two tabs: "UbuntuCA-Carbuccia" and "UbuntuWS-Carbuccia". The terminal displays the following text:

```
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 09:21:34 PM UTC 2024
System load: 1.9 Processes: 250
Usage of /: 42.9% of 9.75GB Users logged in: 0
Memory usage: 8% IPv4 address for ens33: 192.168.79.131
Swap usage: 0%

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:~$ sudo apt update
[sudo] password for danicarbuccia:
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
danicarbuccia@daniellancarb:~$ sudo apt install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.13-0ubuntu3.4).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
danicarbuccia@daniellancarb:~$
```

Prepare CA directory structure:

```

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 09:21:34 PM UTC 2024

System load: 1.9          Processes:      250
Usage of /: 42.9% of 9.75GB  Users logged in:      0
Memory usage: 8%           IPv4 address for ens3: 192.168.79.131
Swap usage:  0%

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

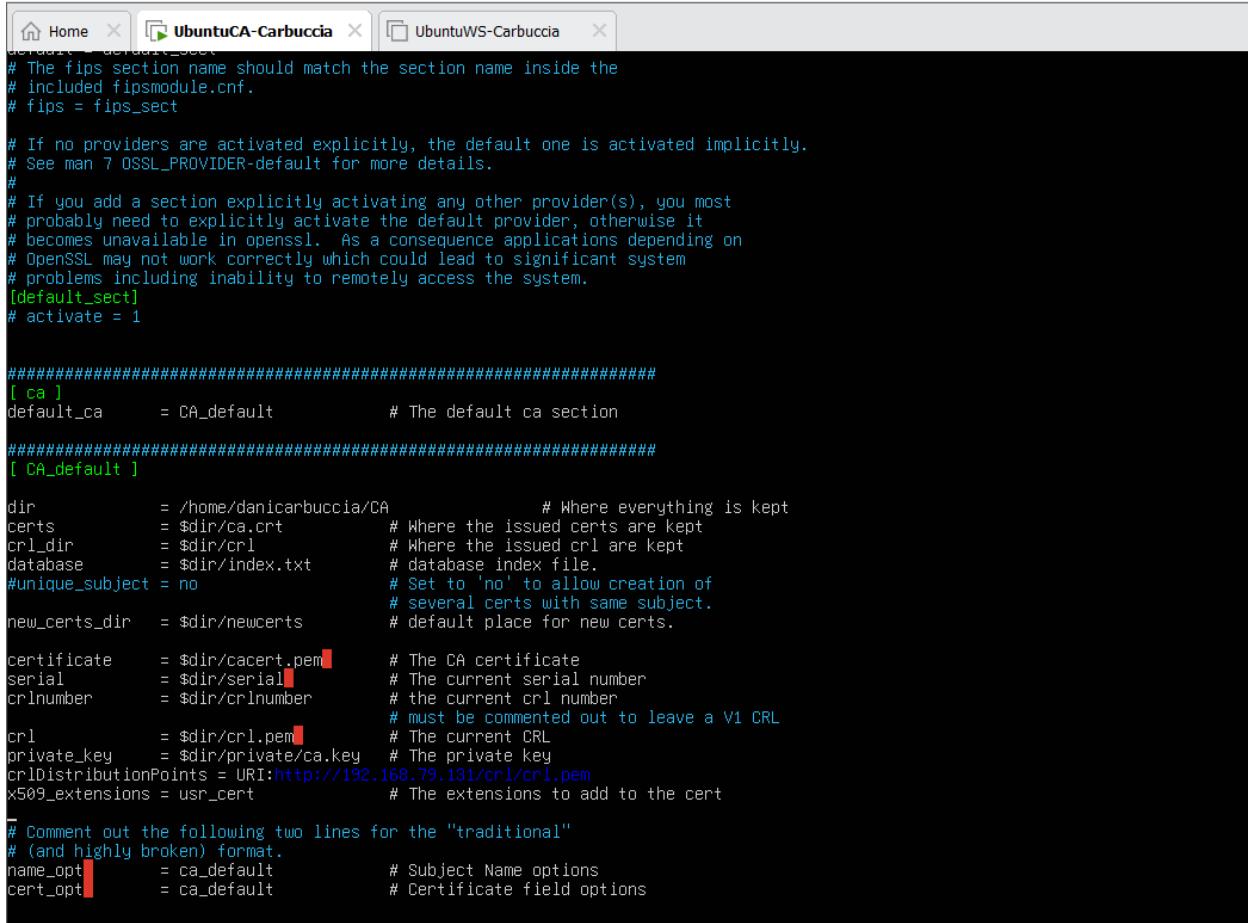
danicarbuccia@daniellancarb:~$ sudo apt update
[sudo] password for danicarbuccia:
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
danicarbuccia@daniellancarb:~$ sudo apt install openssl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssl is already the newest version (3.0.13-0ubuntu3.4).
openssl set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
danicarbuccia@daniellancarb:~$ mkdir ~/CA
danicarbuccia@daniellancarb:~$ mkdir ~/CA/certs ~/CA/newcerts
danicarbuccia@daniellancarb:~$ touch ~/CA/index.txt
danicarbuccia@daniellancarb:~$ echo '1000' > ~/CA/serial
danicarbuccia@daniellancarb:~$ echo '1000' > ~/CA/crlnumber
danicarbuccia@daniellancarb:~$
```

Create CA's self-signed certificate:

```

danicarbuccia@daniellancarb:~/CA$ openssl req -x509 -new -nodes -key private/ca.key -sha256 -days 1024 -out ca.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Texas
Locality Name (eg, city) []:San Antonio
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Daniella Carbuccia
Email Address []:carbucciadaniella@gmail.com
danicarbuccia@daniellancarb:~/CA$
```

Modify /etc/ssl/openssl.cnf:



```

[default_sect]
# The fips section name should match the section name inside the
# included fipsmodule.cnf.
# fips = fips_sect

# If no providers are activated explicitly, the default one is activated implicitly.
# See man 7 OSSL_PROVIDER-default for more details.
#
# If you add a section explicitly activating any other provider(s), you most
# probably need to explicitly activate the default provider, otherwise it
# becomes unavailable in openssl. As a consequence applications depending on
# OpenSSL may not work correctly which could lead to significant system
# problems including inability to remotely access the system.
[default_sect]
# activate = 1

#####
[ ca ]
default_ca      = CA_default          # The default ca section

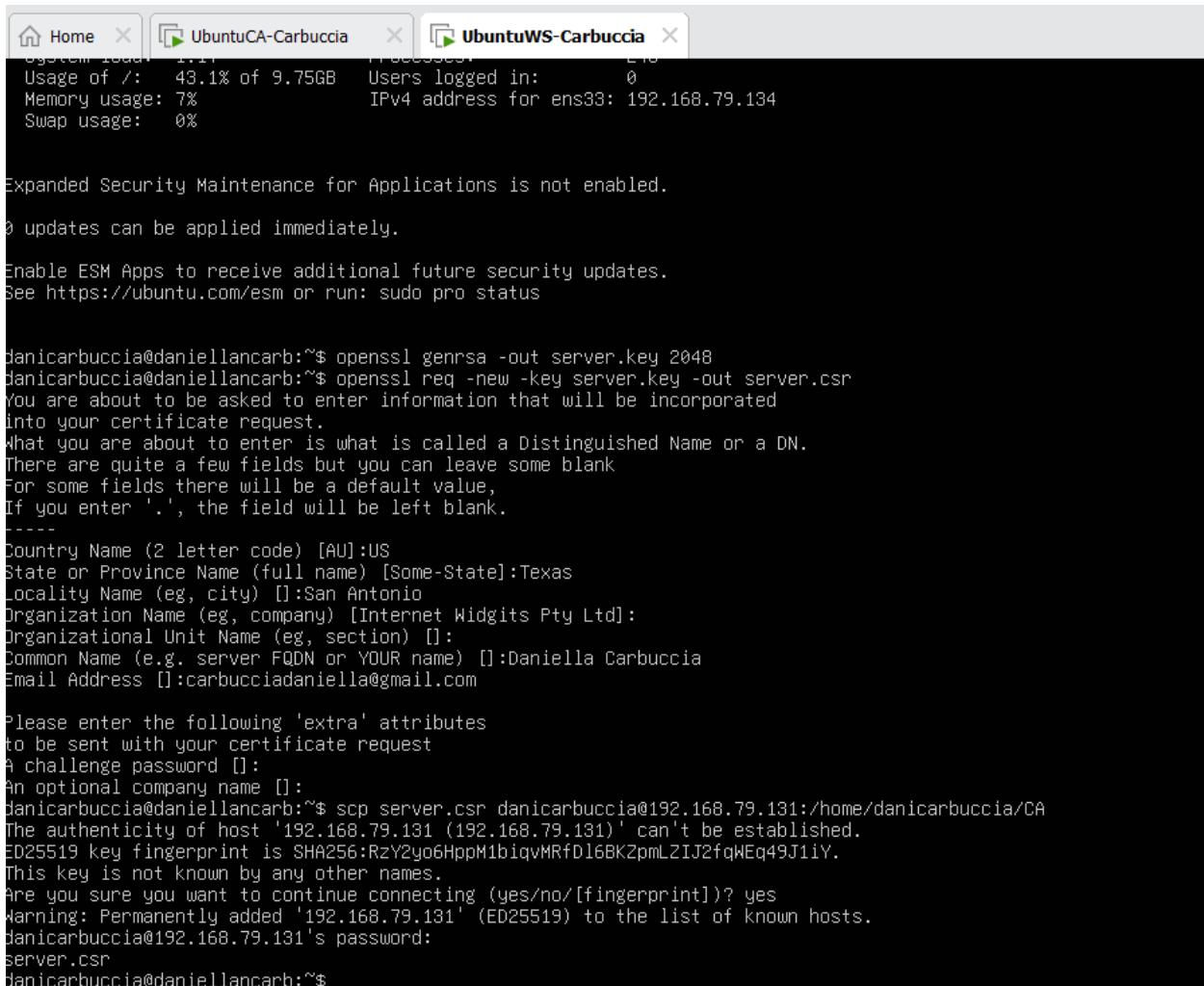
#####
[ CA_default ]
dir              = /home/danicarbuccia/CA          # Where everything is kept
certs            = $dir/ca.crt           # Where the issued certs are kept
crl_dir          = $dir/crl             # Where the issued crl are kept
database         = $dir/index.txt        # database index file.
unique_subject   = no                  # Set to 'no' to allow creation of
# several certs with same subject.
new_certs_dir    = $dir/newcerts        # default place for new certs.

certificate     = $dir/cacert.pem       # The CA certificate
serial          = $dir/serial           # The current serial number
crlnumber        = $dir/crlnumber        # the current crl number
# must be commented out to leave a V1 CRL
crl              = $dir/crl.pem         # The current CRL
private_key      = $dir/private/ca.key    # The private key
crlDistributionPoints = URI:http://192.168.79.131/crl/crl.pem
x509_extensions = usr_cert           # The extensions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
name_opt         = ca_default          # Subject Name options
cert_opt          = ca_default          # Certificate field options

```

Created server.csr in ubuntuWS-Carbuccia:



The screenshot shows a terminal window with three tabs: 'Home', 'UbuntuCA-Carbuccia', and 'UbuntuWS-Carbuccia'. The 'UbuntuWS-Carbuccia' tab is active, displaying the following command-line session:

```

System load: 1.11   17:00:53+01:00
Usage of /: 43.1% of 9.75GB  Users logged in: 0
Memory usage: 7%          IPv4 address for ens33: 192.168.79.134
Swap usage: 0%          

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:~$ openssl genrsa -out server.key 2048
danicarbuccia@daniellancarb:~$ openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Texas
Locality Name (eg, city) []:San Antonio
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Daniella Carbuccia
Email Address []:carbucciadaniella@gmail.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
danicarbuccia@daniellancarb:~$ scp server.csr danicarbuccia@192.168.79.131:/home/danicarbuccia/CA
The authenticity of host '192.168.79.131 (192.168.79.131)' can't be established.
ED25519 key fingerprint is SHA256:RzY2yo6HppM1biqvMRfDl6BK2pmLZIJ2fqWEq49J1iY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.79.131' (ED25519) to the list of known hosts.
danicarbuccia@192.168.79.131's password:
server.csr
danicarbuccia@daniellancarb:~$
```

Signed certificate:

```
danicarbuccia@daniellancarb:~/CA$ openssl ca -in server2.csr -out server.crt -cert ca.crt -keyfile private/ca.key -config /etc/ssl/openssl.cnf
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/ssl/openssl.cnf
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_G0_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_OIF_A62634068.pem	Security_Communication_RootCA2.pem
b0e59380_0	Security_Communication_RootCA3.pem
b1159c4c_0	Security_Communication_Root_CA.pem
b433981b_0	server.crt
b65398e9_0	ssl-cert-snakeoil.pem
b727005e_0	SSL.com_EV_Root_Certification_Authority_ECC.pem
b7a5b043_0	SSL.com_EV_Root_Certification_Authority_RSA_R2.pem
b81093f0_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_Certificate_Authority_-_G2.pem
c01eb047_0	SwissSign_Gold_CA_-_G2.pem
c28a8a30_0	SwissSign_Silver_CA_-_G2.pem
ca664ad9_0	S2AFIR_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_B3.pem
cc450945_0	TrustAsia_Global_Root_CA_B4.pem
cd58d51e_0	Trustwave_Global_Certification_Authority.pem
cd800d63_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.pem	TWCAC_Global_Root_CA.pem
Certum_EC-384_CA.pem	TWCAC_Root_Certification_Authority.pem
Certum_Trusted_Network_CA_2.pem	UCA_Extended_Validation_Root.pem
Certum_Trusted_Network_CA.pem	UCA_Global_G2_Root.pem
Certum_Trusted_Root_CA.pem	USERTrust_ECC_Certification_Authority.pem
CFCA_EV_ROOT.pem	USERTrust_RSA_Certification_Authority.pem
CommScope_Public_Trust_ECC_Root-01.pem	vTrus_ECC_Root_CA.pem
CommScope_Public_Trust_ECC_Root-02.pem	vTrus_Root_CA.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:

```

Home × UbuntuCA-Carbuccia × UbuntuWS-Carbuccia ×
a-certificates.conf ethertypes issue magic.mime PackageKit rmt sudo_logsrvd.conf wgetrc
cloud fonts issue.net manpath.config pam.conf rpc supercat X11
console-setup fstab kernel mime_types pam.d rsyslog.conf sysctl.conf xattr.conf
restore fuse.conf landscape mime_types passwd rsuslog.d sysctl.d xdg
restore.encrypted fixup ldap mke2fs.conf passwd screenrc sysstat xml
ron.d gai.conf ld.so.cache ModemManager peri security systemd zsh_command_not_found
ron.daily gnutls ld.so.conf modprobe.d pk1 sensorsd terminfo
ron.hourly groff ld.so.conf.d modules plymouth sensorsd timezone
ron.monthly group legal modules-load.d pm services tmpfiles.d
rontab group libaudit.conf mtab polkit-1 sgml ubuntu-advantage
ron.weekly grub.d libblockdev multipath pollinate

danicarbuccia@danieldlancarb:~$ cd ss
danicarbuccia@danieldlancarb:~/ss$ ls
[redacted] openssl.cnf openssl.cnf.backup private
danicarbuccia@danieldlancarb:~/ss$ cd private
bash: cd: private: Permission denied
danicarbuccia@danieldlancarb:~/ss$ cd ~/CA
danicarbuccia@danieldlancarb:~/CA$ sudo cp server.key /etc/ssl.private
[sudo] password for danicarbuccia:
danicarbuccia@danieldlancarb:~/CA$ ls
a.crt certs crlnumber index.txt index.txt.attr index.txt.old newcerts private serial serial.old server2.csr server.crt server.csr server.key
danicarbuccia@danieldlancarb:~/CA$ cd private
danicarbuccia@danieldlancarb:~/CA/private$ ls
a.key
danicarbuccia@danieldlancarb:~/CA/private$ cd /
danicarbuccia@danieldlancarb:$ ls
[redacted] boot dev home lib64 lost+found mnt proc run sbin usr-is-merged srv sus usr
[redacted] usr-is-merged cdrom etc lib lib usr-is-merged media opt root sbin snap swap.img tmp var
danicarbuccia@danieldlancarb:$ cd ~/CA
danicarbuccia@danieldlancarb:~/CA$ sudo scp server.crt danicarbuccia@192.168.79.134:/etc/ssl/certs
[sudo] password for danicarbuccia:
danicarbuccia@192.168.79.134's password:
server.crt                                         100% 5826    2.8MB/s   00:00
danicarbuccia@danieldlancarb:~/CA$ sudo scp server.key danicarbuccia@192.168.79.134:/etc/ssl/private
danicarbuccia@192.168.79.134's password:
cp: dest open "/etc/ssl/private/server.key": Permission denied
cp: failed to upload file server.key to /etc/ssl/private
danicarbuccia@danieldlancarb:~/CA$ sudo scp server.key danicarbuccia@192.168.79.134:/home/danicarbuccia/private.key
command 'sudo' not found, did you mean:
  command 'ssdp' from snap ssdp (0.0.1)
  command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
  command 'sfdo' from deb graphviz (2.42.2-9ubuntu0.1)
  command 'sup' from deb sup (20100519-3)
see 'snap info <snapname>' for additional versions.
danicarbuccia@danieldlancarb:~/CA$ sudo scp server.key danicarbuccia@192.168.79.134:/home/danicarbuccia/private.key
[sudo] password for danicarbuccia:
danicarbuccia@192.168.79.134's password:
server.key                                         100% 1704    946.0KB/s   00:00
danicarbuccia@danieldlancarb:~/CA$ _

```

Screenshot of private.key it correct directory:

```

bin usr-is-merged cdrom etc lib lib usr-is-merged media opt root sbin snap swap.img tmp var
danicarbuccia@daniellancarb:~$ cd etc
danicarbuccia@daniellancarb:~/etc$ cd ssl
danicarbuccia@daniellancarb:~/etc/ssl$ cd private
danicarbuccia@daniellancarb:~/etc/ssl/private$ ls
server.key ssl-cert-snakeoil.key
danicarbuccia@daniellancarb:~/etc/ssl/private$ cd ~
danicarbuccia@daniellancarb:~$ ls
private.key server.crt server.csr server.key WS
danicarbuccia@daniellancarb:~$ cp private.key /etc/ssl/private
danicarbuccia@daniellancarb:~$ cd /
-bash: cd: No such file or directory
danicarbuccia@daniellancarb:~$ cd /
danicarbuccia@daniellancarb:~/etc$ cd ssl
danicarbuccia@daniellancarb:~/etc/ssl$ cd private
danicarbuccia@daniellancarb:~/etc/ssl/private$ ls
private.key server.key ssl-cert-snakeoil.key
danicarbuccia@daniellancarb:~/etc/ssl/private$ cat private.key
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKUwgSJgEaOIBAQx8HzxXuVqbX+9
2rK1amao7Tl3n2DRl25uSj4Lagx0TXmcvko9Jx02tiQuem+I1yhn5Wj0h19C7B5
tYIebNjVtprAMRdfM18AYf/pljHmN0DozVzehsIsJ5J1ITyaizbz18/TduNvQDN2
j+ZSvbod4NAx0iEnk2mLrtxJ0fsID070hJr9XZRYNBLEgg4dkh17rS8ekuH0FQ3N
5hxgSmf4gm/tvTJ0e8/Zvrh4ugYlhJ0np3qHtzkFRNkg1AgbxTbIEET4luE26Inq
80Fnz7s0J3xkjg1W6faVp97xyfUa6BDIH143ddPF/Sn,jq17eb0lb/FhdRaWB8w0H
1/6nUToXhgMBhAEcggeAJWskJVfeh15n1dc1TncfBpHTKJ13NSvmcHHicav00fmr
Zx4xnLuHza1ltsu225junc52eUPjYYAe0n9v3cGHZcgHx7KKRwVcWw1nk
v1JuVSe+6a5aI3QNhjt/78JQkbm+ubog46AiYf662nfThk/1Qy2ibqm30UbuaIOA
nRmlt52Gr4L1T7pxNggSaduIJXxuM1iy//7XHmUud2KF+4+eckr/4vJLKr8H0oI
qcVSjI0q4KD+zb8f4EMo61YhLFLr76p9BT7+TCmcOpS0huiaus4KscZMc12fL+
bkHm4Khzkn1bfHDQ084J3N+JuflNjyhbzmcT2s03xtfSv6gtcfJ7r3mUXmqvIBM
Nb+zBI/uKtJaug/Y0YBSLx9eXgvGrndoeff44JHfHWIr8pcuapHirXY3AS3g61oGU
c9MR7rgsSHNPgkXlof/kptFgQK8gQC8B/2rVa2hcdfr4e8l6L2Iugehs/xelPH
1o39HkhBseevuHBfkuerUS19/Vc2uPKf831Stmu1ch/tan1rY9KxfZSMcZb+ufy
GVXlNUUQkrXUBM1HyKfuc5n4cIXmr1J3w06fr3r5nU8mxYSZE23XkcfXv0o
MV1a8XYNrwkBgQDhLHF03n7qq1PGoCs4VUlpf5e5J8CbDE441HRcXAugBD/cs
T2H4JrJc4r/PszwP51oxMoRJnhCuheRoT+qHAN1zYkJ6bz8Qn1lf9u+IgkH2H
Fk+J2xGLof02+oXTqx+5UytQW01Ssgx19Xra/mzObkuAcElV/7fm2QKbgAvd
1TBk78wubq96151JP3vEoJhPKc@0@R0e9RY075An6zpx2322o38YIES91n2vCeDe
VE8f//CcCx10PMxy52QcunXULvnNT6U3Bv7CatcfyYapHxa6nru0mr5c2uDENkvX
tKkmYtigcjpctWkU/dxdSB46/TETBDBbLTEYHAMxAGG@DTLF3hNBuoSXPK+1dtv
14pLYrol4tGroK0Lw+2eMtF104cd1lxgiplozjw+4+IvhjY7AlCryqxDXcmcf
5EDUWkanFyzqobha0EwzhWhkIo92tP6AV6Pr32D87pLpnqHebzclVGF0Tn0@X+Cd
fkphluN5PC06kE7iuKuXlrA=Y
-----END PRIVATE KEY-----
danicarbuccia@daniellancarb:~/etc/ssl/private$ cd /
danicarbuccia@daniellancarb:~$ .

```

Configure SSL on webserver:



The screenshot shows a terminal window with three tabs: "Home", "UbuntuCA-Carbuccia", and "UbuntuWS-Carbuccia". The "UbuntuWS-Carbuccia" tab is active and displays the Apache configuration file (httpd.conf) with SSL settings. The configuration includes directives for LogLevel, ErrorLog, CustomLog, SSLEngine, SSLCertificateFile, SSLCertificateKeyFile, and SSLCertificateChainFile. Below the configuration, a series of terminal commands are shown for enabling the site, restarting the service, and checking for systemctl command availability.

```
# It is also possible to configure the loglevel for particular
# modules, e.g.
LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/certs/server.crt
SSLCertificateKeyFile   /etc/ssl/private/server.key

# Server Certificate Chain:
# Point SSLCertificateChainFile at a file containing the
# concatenation of PEM encoded CA certificates which form the
# certificate chain for the server certificate. Alternatively
# the referenced file can be the same as SSLCertificateFile
# when the CA certificates are directly appended to the server
# certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

# Certificate Authority (CA):
# Set the CA certificate verification path where to find CA
# certificates for client authentication or alternatively one
# huge file containing all of them (file must be PEM encoded)

danicarbuccia@daniellancarb:/$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
danicarbuccia@daniellancarb:/$ sudo systemctl restart apache2
sudo: systemctl: command not found
danicarbuccia@daniellancarb:/$ sudo systemctl restart apache2
danicarbuccia@daniellancarb:/$
```

Apache2 searched with ip address (3 images):

The screenshot shows a web browser window with four tabs open. The active tab is titled 'Apache2 U' and displays the Apache2 Default Page. The page content includes the Ubuntu logo, the text 'Apache2 Default Page', 'Ubuntu', and a red button with the text 'It works!'. Below this, there is a note: 'This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.' A dark blue bar at the bottom contains the text 'Configuration Overview'.

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at /var/www/html/index.html) before continuing to operate your HTTP server.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and splits into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   '-- ports.conf
|-- mods-enabled
|   |-- *.Load
|   '-- *.conf
|-- conf-enabled
```

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

Issuer

▼ Validity

Not Before

Field Value

emailAddress = carbucciadaniella@gmail.com
CN = Daniella Carbuccia
O = Internet Widgits Pty Ltd
L = San Antonio
ST = Texas

Revoking certificate (2 images):

```
danicarbuccia@daniellancarb:~/CA$ openssl ca -revoke newcerts/1000.pem -config /etc/ssl/openssl.cnf
Using configuration from /etc/ssl/openssl.cnf
Revoking Certificate 1000.
Database updated
danicarbuccia@daniellancarb:~/CA$ cd
danicarbuccia@daniellancarb:~$ ls
CA
danicarbuccia@daniellancarb:~$ cd ~/CA
danicarbuccia@daniellancarb:~/CA$ ls
ca.crt  crlnumber  index.txt.attr      index.txt.old  private  serial.old  server.crt  serve
certs   index.txt  index.txt.attr.old  newcerts     serial    server2.csr  server.csr
danicarbuccia@daniellancarb:~/CA$ cd newcerts
danicarbuccia@daniellancarb:~/CA/newcerts$ ls
1000.pem
danicarbuccia@daniellancarb:~/CA/newcerts$
```

```

Home × UbuntuCA-Carbuccia × UbuntuWS-Carbuccia ×
1:09:be:6f:a1:7f:76:80:8c:2e:8a:6f:34:29:f5:b9:4f:13:
0d:43:3b:92:f1:24:b6:c0:27:9d:95:9d:2c:f6:e1:d8:d5:45:
64:02:39:a5:3e:1b:c9:90:6c:bd:f2:39:ab:95:29:c3:80:02:
da:de:18:06:39:99:1c:31:d2:f5:72:82:ec:0f:4c:a7:9d:c3:
9f:46:1f:82:3d:d5:31:f0:fc:71:85:29:a6:b6:53:1e:40:0d:
2b:87:dc:35:b1:ba:8a:cd:8f:94:06:9f:de:6d:00:60:03:a7:
2c:2d:60:23:a7:32:b7:df:f2:2c:a6:96:e0:4b:83:52:08:61:
18:f8:3f:a8:a4:94:2d:3c:3b:7b:d7:8b:b5:a5:04:cc:2a:55:
40:53:12:04:7b:24:56:29:aa:80:58:2c:5f:74:e0:7e:1f:00:
cd:68:9f:b3:33:a6:e3:b2:f6:be:38:2c:a5:40:f0:8c:30:12:
a9:55:5c:6b:60:fa:d6:1f:11:11:2b:b1:fd:1f:82:29:16:ef:
96:44:c5:e0:12:f4:5c:98:d6:01:50:04:3b:1f:d2:99:41:95:
0c:9a:21:56:4c:0a:cf:7e
-----BEGIN CERTIFICATE-----
MIIe8zCCAtugAwIBAgICEAAwDQYJKoZIhvcNAQELBQAwgZBxCzAJBgNVBAYTAlVT
MQ4wDAYDVQQIDAVU2XhhczEUMBIGA1UEBwwLU2FuIEFudG9uaW8xITAfBgNVBAoM
GEIudGVybmv0IFdpZGdpdHMgUHR5IEx0ZDEbMBkGA1UEAwuSRGFuaWVsbGEgQ2Fy
YnVjY2lhMSowKAYJKoZIhvcNAQkBFhtjYXJ1dWNjaWFkYW5pZWxsYUBnbWFpbC5j
b20whhcNMjQwOTIyMjI10DE0WhcNMjUwOTIyMjI10DE0WjCBiTELMAkGA1UEBhMC
vVMxDjAMBgNVBAgMBVRleGFzMSExHwYDVQQDBhJbnRlcmlCBXaWRnaXRzIFB0
esBMdGQxGzAZBgNvBMEkRhbm1lbGxhIENhcmJ1Y2NpYTEqMCgGCSqGSIB3DQEJ
ARYbY2FyYnVjY2lhZGFuaWVsbfGA221haWuuY29tMIIBiJANBgkqhkiG9w0BAQE
AAOCAQ8AMIIBCgKCAQEaujKhnNe/UM7lB1K9Ly1Fd6E+5DdL8sYBwAh6D2gV0Sj
r0JJCPsk6G+oMG4UVVGcQjsNb9Kvz+1J6JSnzbLzq7WHr0KgFbRVd0KcsY3p8W6
oDGEff/C/mLnebJF iAUWfVYmnDwnyoIdKx0/mDvi6Kexlw+wP5rIU+YvzC0ohp4W
kF40tL13007BIkWoBMKQkL50J0ccH1jmyyzEo9cVsrWHHHns02PkKotcraTcWMfs
an1Tvenl0AH7ALqSceGrf07YPUhLZyLzaW2K+ppdgEH65r4Nb/EIla2dDxxRRY0i
tPdR79ehx2uwFDhg0dVVFQwBWyugPJz9vaC+WhnwIDAQABo0wSzAJBgNVHRME
AjAAMB0GA1UdDgQWBBTDc7hxW1/RH4XmiM9Lp1E02QmIMDAfBgNVHSMEGDAwgbSN
7t7Empn/XWUzsSGq1MGTb1w8+zANBgkqhkiG9w0BAQsFAAOCAgEAIgcn4W/DQqnE
nWMtYn41YhNPQt8q+dkKtHispdpFlydEPAYGkcTq37Itwl9Er763c30uIsbnxh4I
2D2co/7LSn5sACKut9wD0j3Eo0jGjgh1YdGyaMlkzvBuv/Dv+q+GpaxVUwTzjNoh
7/0P0ngPP/x0Qdautr/iP11ttdrs73oy5BYMp5Jne1LJTC75iEmeomPYddKKYAMA
oqlufziPnm8v8LJaxYEEbiWMqUH7JsguJ0W/4bAnF7Q13Vc3h1UIs8c/an7pVMvq
ydsRx/d3yuWXxZzzAxzNxEKVinqchNxnoboCwqjE9Y7M0T4wyH12rdJ8DFxeErv
S/2phQN8zsWruCiiZtCOELBNikSrpZxxFzY6yH3wn5jmqJw6afhocQm+b6F/douM
LopvNCn1uU8TDUM7kvEktsAnn2WdlPbh09VF2AI5pT4by2BsvfI5q5Upw4AC2t4Y
Bjm2HDHS9XC7A9Mp53Dn0Yfgj3Vmfd8cyUppr2THkANK4fcNbG6is2PlAaf3m0A
YAOnLC1gI6cyt9/yLKa4EuDUghhGPg/qKSULTw7e9eLtaUEzCpVQFMSBHSkVimq
gFgsX3Tgfh8AzWifsz0m47L2vJgspUDwjDASqVVca2D61h8RESux/R+CKRbv1kTF
4BL0XjjWVAE0x/SmUGVDJohVkwKz34=
-----END CERTIFICATE-----
danicarbu...@daniellancarb:~/CA/newcerts$ cd ~/CA
danicarbu...@daniellancarb:~/CA$ openssl ca -gencrl -out crl.pem -config /etc/ssl/openssl.cnf
Using configuration from /etc/ssl/openssl.cnf
danicarbu...@daniellancarb:~/CA$ sudo mkdir /var/www/crl
[sudo] password for danicarbu...:
danicarbu...@daniellancarb:~/CA$ sudo cp crl.pem /var/www/crl/
danicarbu...@daniellancarb:~/CA$
```

SSL certificate:

```

certutil -H <command> : Print available options for the given command
certutil -H : Print complete help output of all commands and options
certutil --syntax : Print a short summary of all commands and options
danicarbu...@daniellancarb:~/CA$ openssl ca -revoke newcerts/1000.pem -config /etc/ssl/openssl.cnf
Using configuration from /etc/ssl/openssl.cnf
ERROR: Already revoked, serial number 1000
danicarbu...@daniellancarb:~/CA$
```