

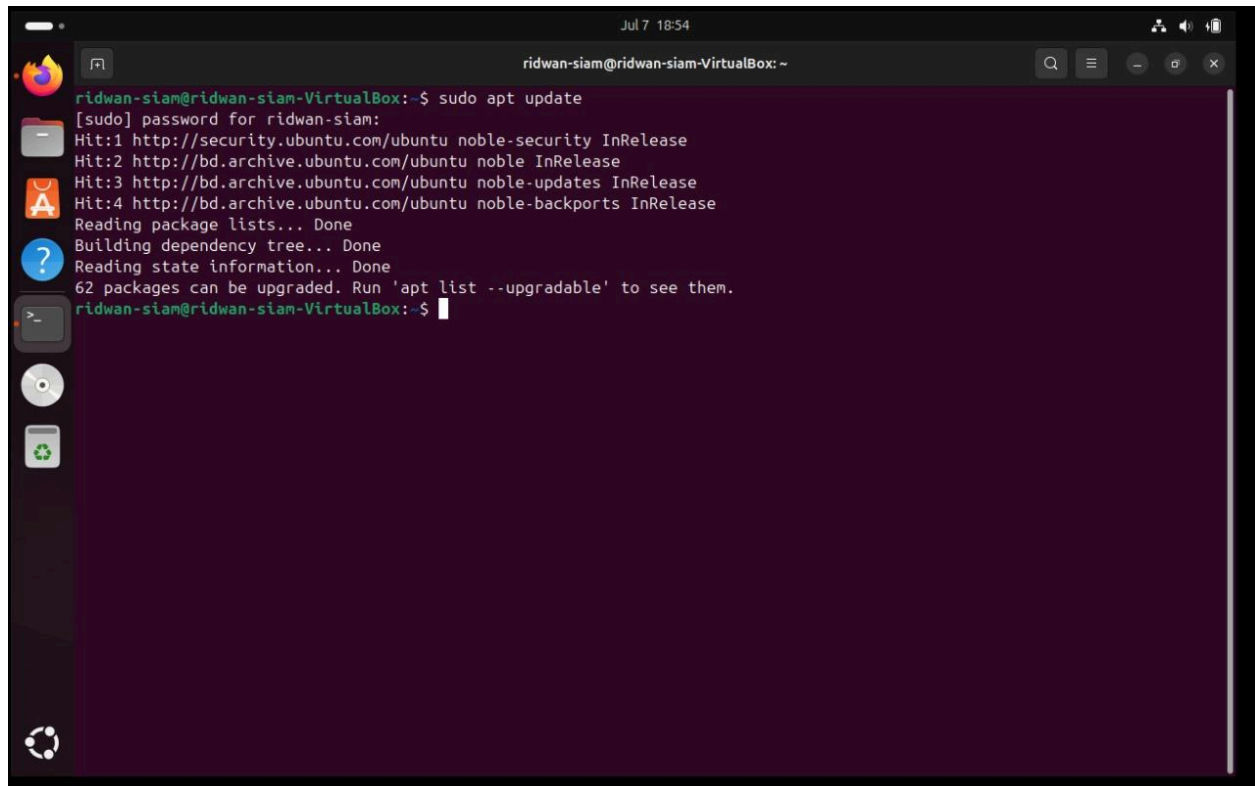
# Task-1: Setting up an Apache web server

## Step 1 — Installing Apache

Apache is available within Ubuntu's default software repositories, making it possible to install it using conventional package management tools.

Let's begin by updating the local package index to reflect the latest upstream changes. If apt is not recognised as a command, try apt-get instead of apt.

**sudo apt update.**

A screenshot of a terminal window titled 'ridwan-siam@ridwan-siam-VirtualBox: ~'. The terminal shows the command 'sudo apt update' being executed. The output includes the password prompt, four 'Hit' messages for security and archive repositories, and status messages for reading package lists, building the dependency tree, and reading state information. It concludes by stating that 62 packages can be upgraded and suggests running 'apt list --upgradable' to see them. The prompt returns to the user.

```
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo apt update
[sudo] password for ridwan-siam:
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://bd.archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://bd.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://bd.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
62 packages can be upgraded. Run 'apt list --upgradable' to see them.
ridwan-siam@ridwan-siam-VirtualBox:~$
```

Then, install the apache2 package:

**sudo apt install apache2**

After confirming the installation, apt will install Apache and all required dependencies

```
Jul 7 18:59
ridwan-siam@ridwan-siam-VirtualBox: ~
Hit:3 http://bd.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://bd.archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
62 packages can be upgraded. Run 'apt list --upgradable' to see them.
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo apt install apache2
[sudo] password for ridwan-siam:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
0 upgraded, 8 newly installed, 0 to remove and 62 not upgraded.
Need to get 1,896 kB of archives.
After this operation, 7,452 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://bd.archive.ubuntu.com/ubuntu noble/main amd64 libapr1t64 amd64 1.7.2-3.1build2 [107 kB]
Get:2 http://bd.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1t64 amd64 1.6.3-1.1ubuntu7 [91.9 kB]
Get:3 http://bd.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.3-1.1ubuntu7 [11.2 kB]
Get:4 http://bd.archive.ubuntu.com/ubuntu noble/main amd64 libaprutil1-ldap amd64 1.6.3-1.1ubuntu7 [9,116 B]
Get:5 http://bd.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-bin amd64 2.4.58-1ubuntu8.1 [1,327 kB]
Get:6 http://bd.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-data all 2.4.58-1ubuntu8.1 [163 kB]
Get:7 http://bd.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2-utils amd64 2.4.58-1ubuntu8.1 [96.2 kB]
Get:8 http://bd.archive.ubuntu.com/ubuntu noble-updates/main amd64 apache2 amd64 2.4.58-1ubuntu8.1 [90.2 kB]
Fetched 1,896 kB in 3s (616 kB/s)
Selecting previously unselected package libapr1t64:amd64.
(Reading database ... 147852 files and directories currently installed.)
Preparing to unpack .../0-libapr1t64_1.7.2-3.1build2_amd64.deb ...
```

## Step 2 — Adjusting the Firewall

Before testing Apache, it's necessary to modify the firewall settings to allow outside access to the default web ports. This is necessary if you try to access your web site from a separate machine. Assuming that you followed the instructions in the prerequisites, you should have a UFW firewall configured to restrict access to your server.

List the ufw application profiles by typing:

```
sudo ufw app list
```

It is recommended that you enable the most restrictive profile that will still allow the traffic you've configured. Since we haven't configured SSL for our server yet in this guide, we will only need to allow traffic on port 80:

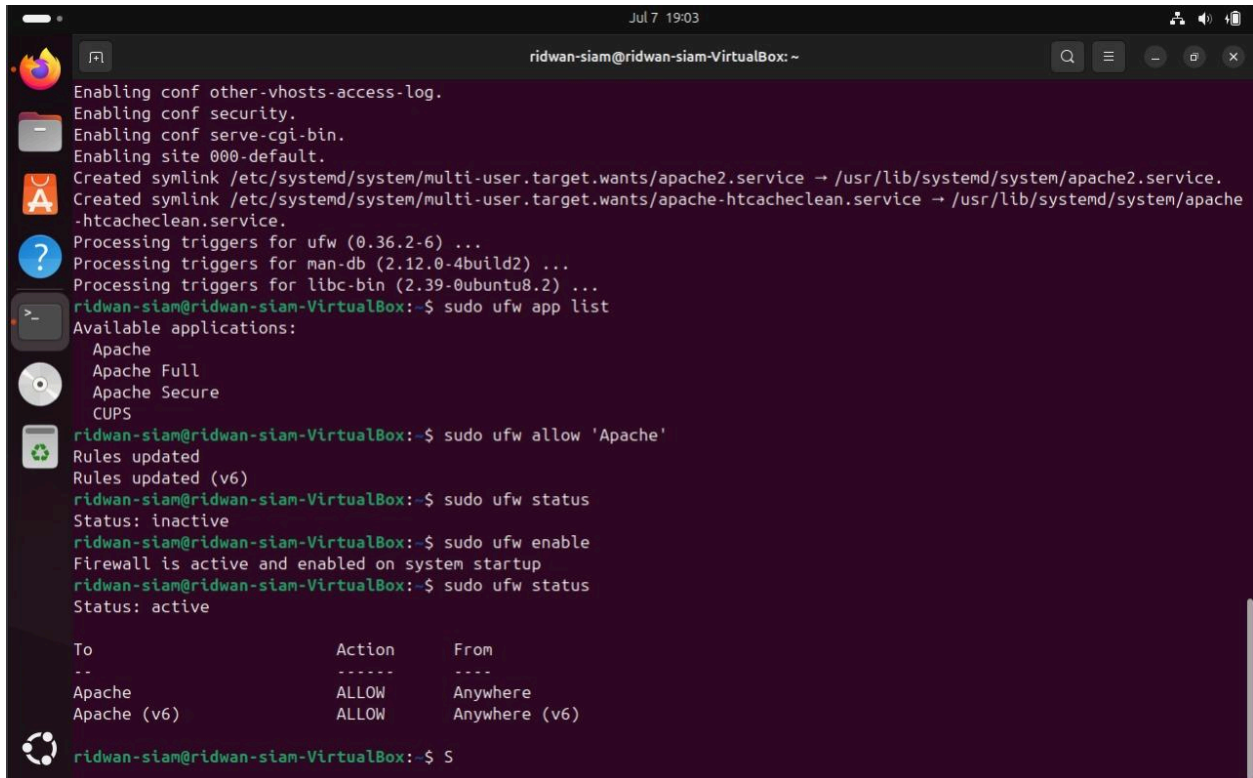
```
sudo ufw allow 'Apache'
```

You can verify the change by typing:

```
sudo ufw status
```

```
sudo ufw enable
```

As you can see, the profile has been activated to allow access to the web server.



```
Jul 7 19:03
ridwan-siam@ridwan-siam-VirtualBox: ~

Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created 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 libc-bin (2.39-0ubuntu8.2) ...
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw allow 'Apache'
Rules updated
Rules updated (v6)
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw status
Status: inactive
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw status
Status: active

To           Action      From
--           -
Apache       ALLOW       Anywhere
Apache (v6)  ALLOW      Anywhere (v6)
```

### Step 3 —Checking your Web Server

At the end of the installation process, Ubuntu 18.04 starts Apache. The web server should already be up and running.

Check with the systemd init system to make sure the service is running by typing:

**sudo systemctl status apache2**

```
Jul 7 19:06
ridwan-siam@ridwan-siam-VirtualBox: ~
Rules updated (v6)
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw status
Status: inactive
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo ufw status
Status: active

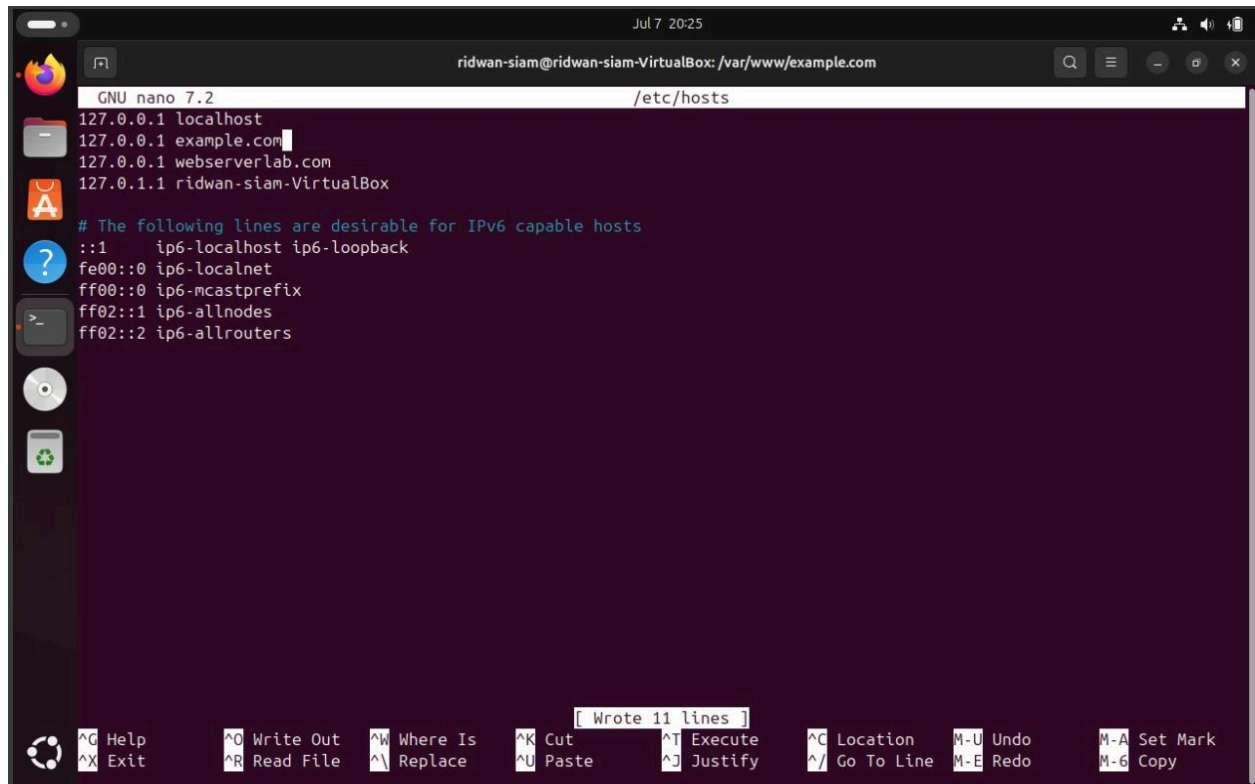
To Action From
--
Apache ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)

ridwan-siam@ridwan-siam-VirtualBox:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-07-07 18:58:00 +06; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 6171 (apache2)
      Tasks: 55 (limit: 4616)
    Memory: 5.4M (peak: 5.5M)
       CPU: 112ms
    CGroup: /system.slice/apache2.service
            └─6171 /usr/sbin/apache2 -k start
              6173 /usr/sbin/apache2 -k start
              6174 /usr/sbin/apache2 -k start

Jul 07 18:58:00 ridwan-siam-VirtualBox systemd[1]: Starting apache2.service - The Apache HTTP Server...
Jul 07 18:58:00 ridwan-siam-VirtualBox apache2[6170]: AH00558: apache2: Could not reliably determine the server's full
Jul 07 18:58:00 ridwan-siam-VirtualBox systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
ridwan-siam@ridwan-siam-VirtualBox:~$
```

You can access the default Apache landing page to confirm that the software is running properly through your IP address or by just typing localhost (127.0.0.1) in the browser. Let us use webserverlab.com as our domain name. To get our computers recognise this domain name, let us add the following entry to /etc/hosts; this entry basically maps the domain name webserverlab.com to our localhost (i.e., 127.0.0.1):

- 127.0.0.1 webserverlab.com



The screenshot shows a terminal window titled 'ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com'. Inside, the GNU nano 7.2 editor is open, displaying the /etc/hosts file. The file contains the following entries:

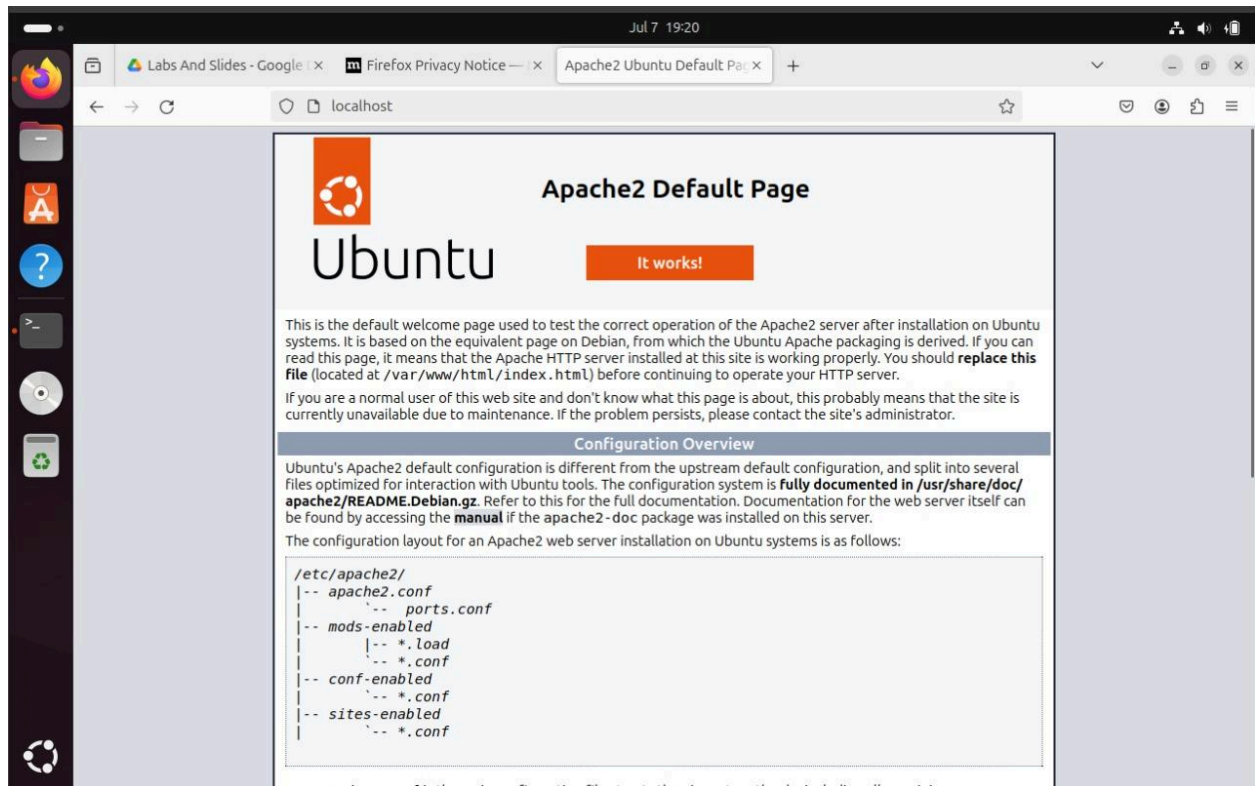
```
127.0.0.1 localhost
127.0.0.1 example.com
127.0.0.1 webserverlab.com
127.0.1.1 ridwan-siam-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

The bottom of the terminal shows a status bar with various keyboard shortcuts: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\_ Replace, ^K Cut, ^U Paste, ^T Execute, ^J Justify, ^C Location, ^\_ Go To Line, M-U Undo, M-E Redo, M-A Set Mark, M-C Copy. A message '[ Wrote 11 lines ]' is also visible.

Now, to check the installation of Apache, enter this domain or its IP address into your browser's address bar:

<http://webserverlab.com> or <http://localhost> or <http://127.0.0.1> or [http://ip\\_address](http://ip_address)



## Step 2 — Setting up a single virtual host

Create the directory for example.com as follows, using the `-p` flag to create any necessary parent directories:

```
sudo mkdir -p /var/www/example.com/html
```

Next, assign ownership of the directory with the `$USER` environment variable:

```
sudo chown -R $USER:$USER /var/www/example.com/html
```

The permissions of your web roots should be correct if you haven't modified your unmask value, but you can make sure by typing:

```
sudo chmod -R 755 /var/www/example.com
```



```
Jul 8 03:16
ridwan-siam@ridwan-siam-VirtualBox: /etc/apache2/sites-enabled

File Edit View Search Terminal Tabs Help

ridwan-siam@ridwan-siam-VirtualBox: /etc/ap... x ridwan-siam@ridwan-siam-VirtualBox: /etc/ap... x ridwan-siam@ridwan-siam-VirtualBox: /etc/ssl/... x

command 'sumo' from deb sumo (1.18.0+dfsg-3build2)
command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
Try: sudo apt install <deb name>
ridwan-siam@ridwan-siam-VirtualBox: /etc$ sudo nano hosts
ridwan-siam@ridwan-siam-VirtualBox: /etc$ cd /var/www/html
ridwan-siam@ridwan-siam-VirtualBox: /var/www/html$ cd
ridwan-siam@ridwan-siam-VirtualBox: ~$ cd /etc
ridwan-siam@ridwan-siam-VirtualBox: /etc$ sudo nano hosts
ridwan-siam@ridwan-siam-VirtualBox: /etc$ cd /var/www/html
ridwan-siam@ridwan-siam-VirtualBox: /var/www/html$ ls
index.html
ridwan-siam@ridwan-siam-VirtualBox: /var/www/html$ cd
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo mkdir -p /var/www/example.com/html
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo chown -R $USER:$USER /var/www/example.com/html
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo chmod -R 755 /var/www/example.com
ridwan-siam@ridwan-siam-VirtualBox: ~$ nano /var/www/example.com/html/index.html
ridwan-siam@ridwan-siam-VirtualBox: ~$ nano /var/www/example.com/html/index.html
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo nano /etc/apache2/sites-available/example.com.conf
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo a2ensite example.com.conf
Enabling site example.com.
To activate the new configuration, you need to run:
systemctl reload apache2
ridwan-siam@ridwan-siam-VirtualBox: ~$ ^C
ridwan-siam@ridwan-siam-VirtualBox: ~$ ^C
ridwan-siam@ridwan-siam-VirtualBox: ~$ systemctl reload apache2
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo a2ensite example.com.conf
Site example.com already enabled
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo a2dissite 000-default.conf
Site 000-default disabled
```

Next, create a sample index.html page using nano or your favorite editor:

**nano /var/www/example.com/html/index.html**

```
Jul 7 19:33
ridwan-siam@ridwan-siam-VirtualBox: ~

GNU nano 7.2 /var/www/example.com/html/index.html *

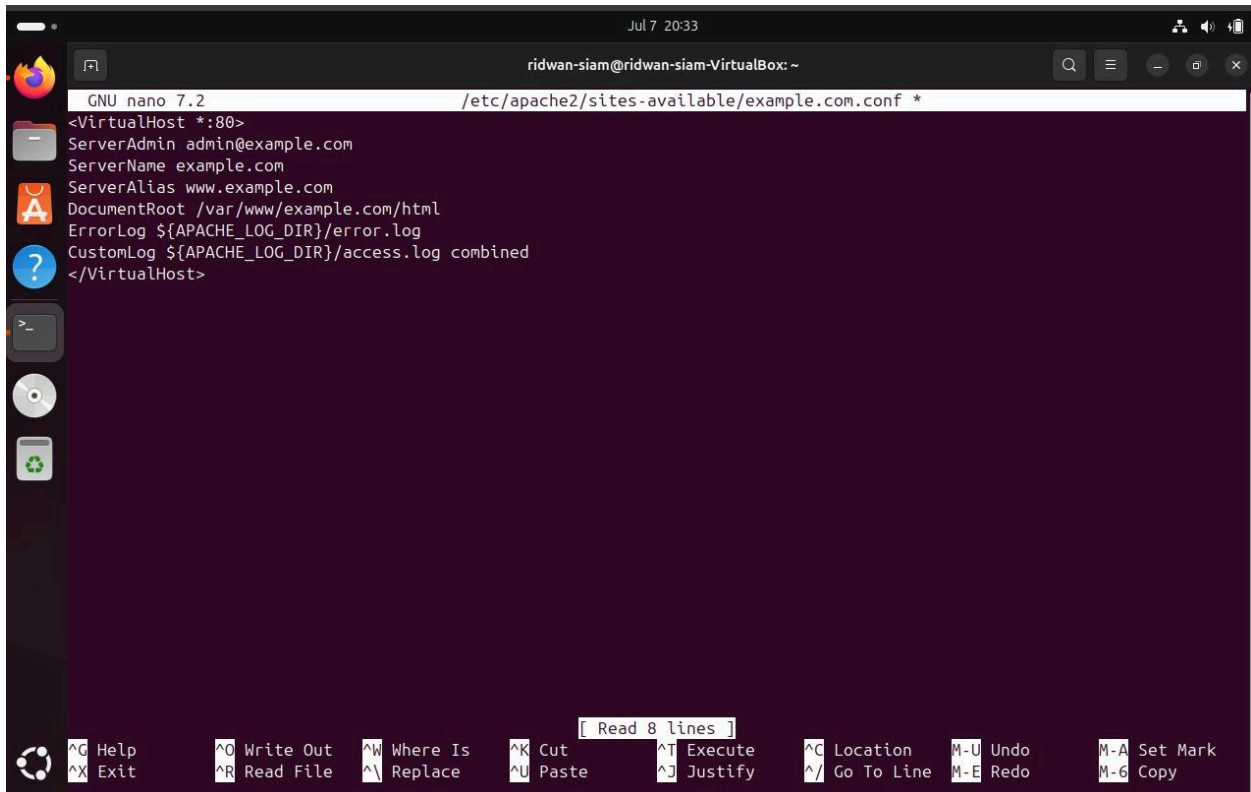
<html>
<head>
<title>Welcome to Example.com!</title>
</head>
<body>
<h1>Congratulations Siamboss! The example.com server block is working!</h1>
</body>
</html>

[ Read 8 lines ]
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^_ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo      M-C Copy
```

Save and close the file when you are finished.

In order for Apache to serve this content, it's necessary to create a virtual host file with the correct directives. Instead of modifying the default configuration file located at `/etc/apache2/sites-available/000-default.conf` directly, let's make a new one at `/etc/apache2/sites-available/example.com.conf`:

**`sudo nano /etc/apache2/sites-available/example.com.conf`**



```
GNU nano 7.2 /etc/apache2/sites-available/example.com.conf *
<VirtualHost *:80>
ServerAdmin admin@example.com
ServerName example.com
ServerAlias www.example.com
DocumentRoot /var/www/example.com/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Paste in the following configuration block, which is similar to the default, but updated for our new directory and domain name:

Notice that we've updated the `DocumentRoot` to our new directory and `ServerAdmin` to an email that the `example.com` site administrator can access. We've also added two directives: `ServerName`, which establishes the base domain that should match for this virtual host definition, and `ServerAlias`, which defines further names that should match as if they were the base name.

Save and close the file when you are finished.

Let's enable the file with the `a2ensite` tool:

**`sudo a2ensite example.com.conf`**



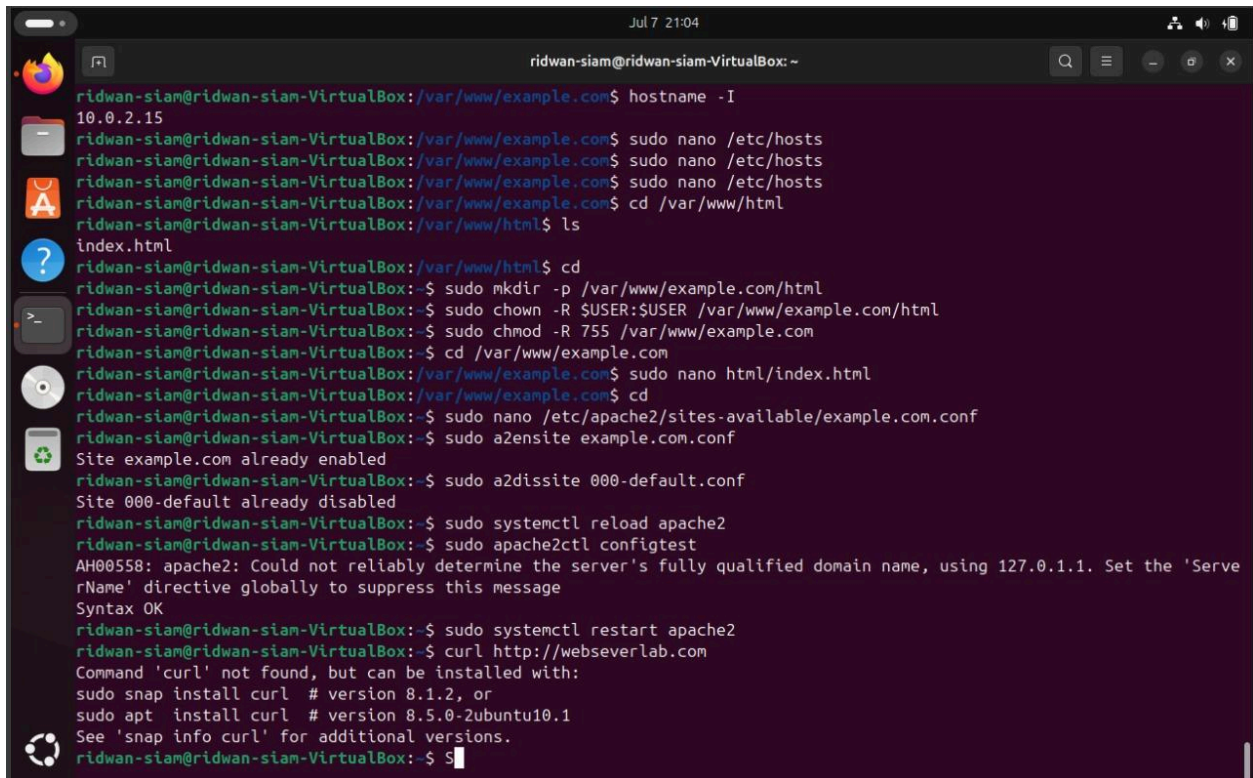
Disable the default site defined in 000-default.conf:

**sudo a2dissite 000-default.conf**

Next, let's test for configuration errors:

**sudo apache2ctl configtest**

If you see a "Syntax OK" output, then it's properly configured.

A terminal window titled 'ridwan-siam@ridwan-siam-VirtualBox: ~' with a timestamp of 'Jul 7 21:04'. The terminal shows a series of commands and their outputs for configuring Apache2. The user starts in the directory /var/www/example.com and runs 'hostname -I' to get the IP address 10.0.2.15. They then edit /etc/hosts with 'sudo nano /etc/hosts'. Next, they move to /var/www/html and list the files, seeing 'index.html'. They create the directory /var/www/example.com/html with 'sudo mkdir -p /var/www/example.com/html', set permissions with 'sudo chown -R \$USER:\$USER /var/www/example.com/html', and 'sudo chmod -R 755 /var/www/example.com'. They then edit the index.html file with 'sudo nano html/index.html'. After that, they edit the site configuration file /etc/apache2/sites-available/example.com.conf with 'sudo nano /etc/apache2/sites-available/example.com.conf'. They run 'sudo a2ensite example.com.conf' and see 'Site example.com already enabled'. Then they run 'sudo a2dissite 000-default.conf' and see 'Site 000-default already disabled'. They reload Apache with 'sudo systemctl reload apache2' and test the configuration with 'sudo apache2ctl configtest'. The output shows a warning from AH00558 about the server's fully qualified domain name but a 'Syntax OK' status. Finally, they restart Apache with 'sudo systemctl restart apache2'. The prompt returns to the user's shell.

```
ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com$ hostname -I
10.0.2.15
ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com$ sudo nano /etc/hosts
ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com$ sudo nano /etc/hosts
ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com$ sudo nano /etc/hosts
ridwan-siam@ridwan-siam-VirtualBox: /var/www/html$ cd /var/www/html
ridwan-siam@ridwan-siam-VirtualBox: /var/www/html$ ls
index.html
ridwan-siam@ridwan-siam-VirtualBox: /var/www/html$ cd
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo mkdir -p /var/www/example.com/html
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo chown -R $USER:$USER /var/www/example.com/html
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo chmod -R 755 /var/www/example.com
ridwan-siam@ridwan-siam-VirtualBox: ~$ cd /var/www/example.com
ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com$ sudo nano html/index.html
ridwan-siam@ridwan-siam-VirtualBox: /var/www/example.com$ cd
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo nano /etc/apache2/sites-available/example.com.conf
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo a2ensite example.com.conf
Site example.com already enabled
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo a2dissite 000-default.conf
Site 000-default already disabled
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo systemctl reload apache2
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
ridwan-siam@ridwan-siam-VirtualBox: ~$ sudo systemctl restart apache2
ridwan-siam@ridwan-siam-VirtualBox: ~$ curl http://webserverlab.com
Command 'curl' not found, but can be installed with:
sudo snap install curl # version 8.1.2, or
sudo apt install curl # version 8.5.0-2ubuntu10.1
See 'snap info curl' for additional versions.
ridwan-siam@ridwan-siam-VirtualBox: ~$
```

Restart Apache to implement your changes:

**sudo systemctl restart apache2**

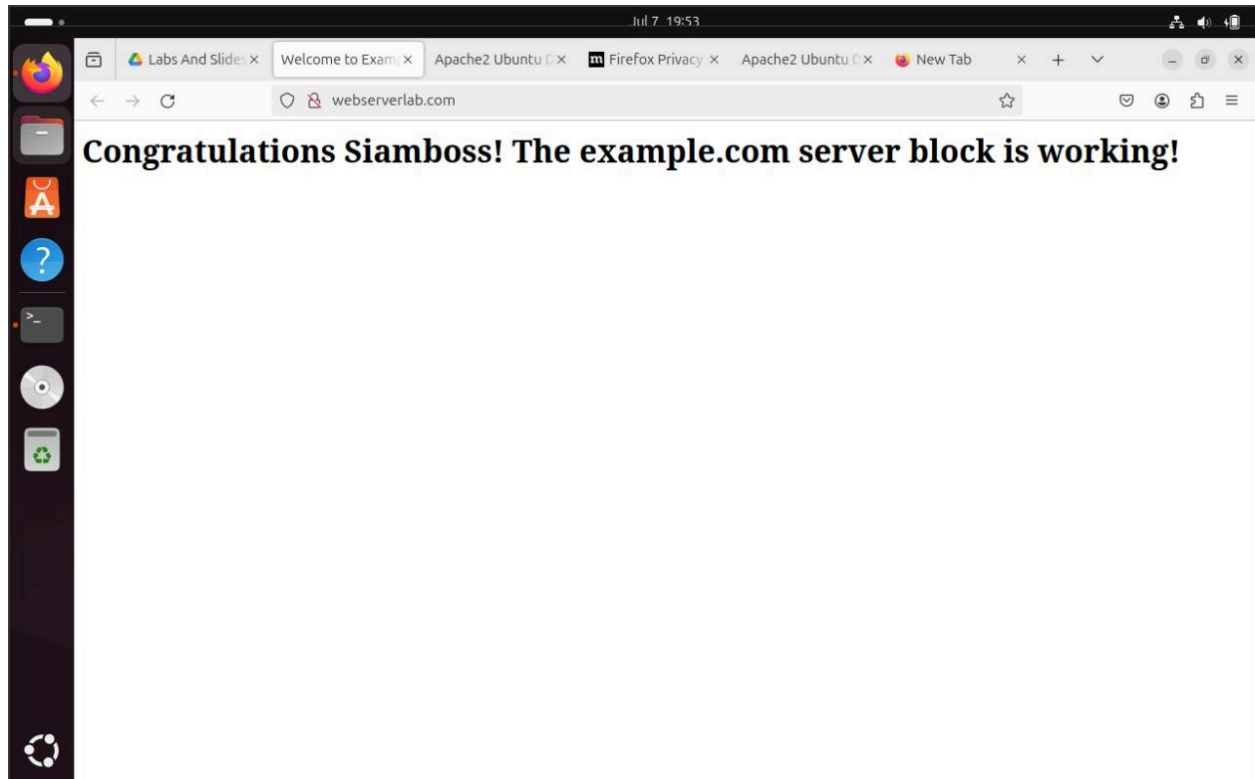
Apache should now be serving your domain name. Now issue the following command:

**sudo a2ensite example.com.conf**

Restart Apache to implement your changes:

**sudo systemctl restart apache2**

Try navigating to <http://webserverlab.com>, observe what happens. Think about what is happening. Try to navigate to <http://127.0.0.1>. What happened and why?

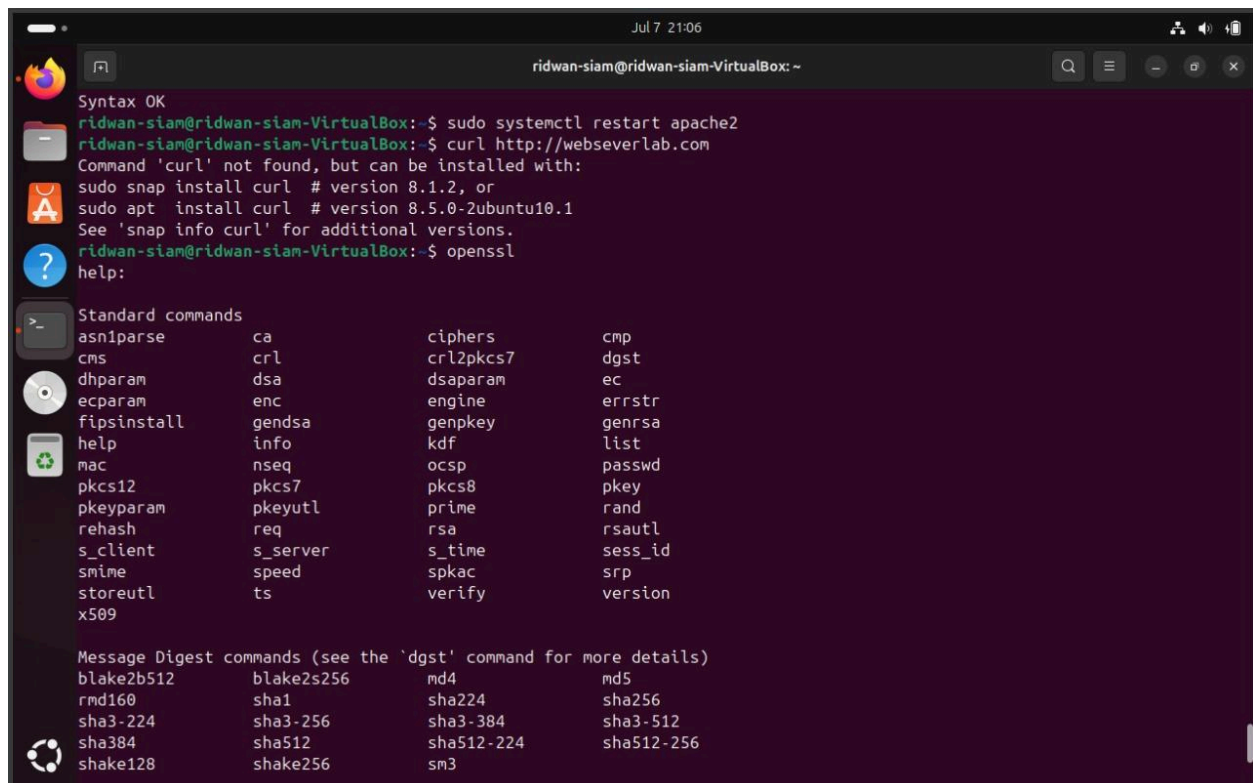


# Lab 5: Securing Apache Web Server

## Task-1: Becoming a certificate authority

In this lab, you will need to create digital certificates, but you will not be going to pay to any commercial CA. You will become a root CA, and then use this CA to issue certificate for others (e.g. servers). In this task, you will make yourself a root CA, and generate a certificate for this CA. Unlike other certificates, which are usually signed by another CA, the root CA's certificates are self-signed. Root CA's certificates are usually pre-loaded into most operating systems, web browsers, and other software that rely on certificate-based security. Root CA's certificates are unconditionally trusted.

For this, you will use OpenSSL which are already familiar with from Lab-3.



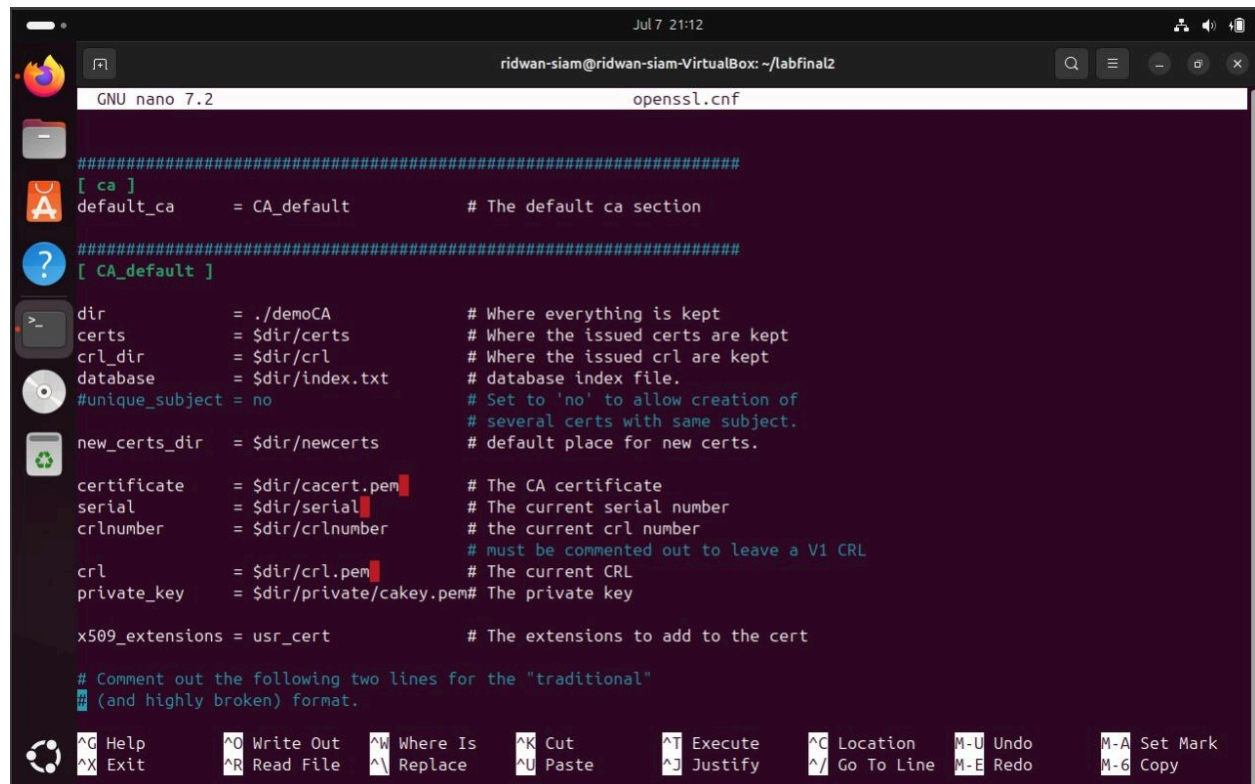
```
Jul 7 21:06
ridwan-siam@ridwan-siam-VirtualBox: ~
Syntax OK
ridwan-siam@ridwan-siam-VirtualBox:~$ sudo systemctl restart apache2
ridwan-siam@ridwan-siam-VirtualBox:~$ curl http://webseverlab.com
Command 'curl' not found, but can be installed with:
sudo snap install curl # version 8.1.2, or
sudo apt install curl # version 8.5.0-2ubuntu10.1
See 'snap info curl' for additional versions.
ridwan-siam@ridwan-siam-VirtualBox:~$ openssl
help:

Standard commands
asn1parse      ca              ciphers         cmp
cms            crl             crl2pkcs7       dgst
dhparam        dsa            dsaparam        ec
ecparam        enc            engine          errstr
fipsinstall    gendsa         genpkey         genrsa
help           info           kdf             list
mac            nseq           ocsf            passwd
pkcs12         pkcs7          pkcs8           pkey
pkeyparam      pkeyutl        prime           rand
rehash         req            rsa             rsautl
s_client       s_server       s_time          sess_id
smime          speed          spkac           srp
storeutl       ts             verify          version
x509

Message Digest commands (see the 'dgst' command for more details)
blake2b512     blake2s256     md4             md5
rmd160         sha1            sha224          sha256
sha3-224       sha3-256       sha3-384        sha3-512
sha384         sha512         sha512-224     sha512-256
shake128       shake256        sm3
```

To start this task, create a folder for this task and cd into it. In this folder, you will need to create a particular configuration file as discussed below.

The Configuration File openssl.cnf: In order to use OpenSSL to create certificates, you have to have a configuration file. The configuration file usually has an extension .cnf. It is used by three OpenSSL commands: ca, req and x509. The manual page of openssl.cnf can be found using Google search. You can also get a copy of the configuration file from /usr/lib/ssl/openssl.cnf. After copying this file into your current directory, you need to create several sub-directories as specified in the configuration file (look at the [CA default] section):



The screenshot shows a terminal window with the nano 7.2 editor open to the file openssl.cnf. The file contains configuration for the OpenSSL CA. The [ca] section sets default\_ca to CA\_default. The [CA\_default] section defines several paths: dir (./demoCA), certs (\$dir/certs), crl\_dir (\$dir/crl), database (\$dir/index.txt), new\_certs\_dir (\$dir/newcerts), certificate (\$dir/cacert.pem), serial (\$dir/serial), crlnumber (\$dir/crlnumber), crl (\$dir/crl.pem), and private\_key (\$dir/private/cakey.pem). It also sets unique\_subject to no and x509\_extensions to usr\_cert. Comments explain the purpose of each path and the unique\_subject setting. At the bottom, there are instructions to comment out lines for the 'traditional' format. The terminal window title is 'ridwan-siam@ridwan-siam-VirtualBox: ~/labfinal2' and the date/time is 'Jul 7 21:12'.

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

#####
[ CA_default ]

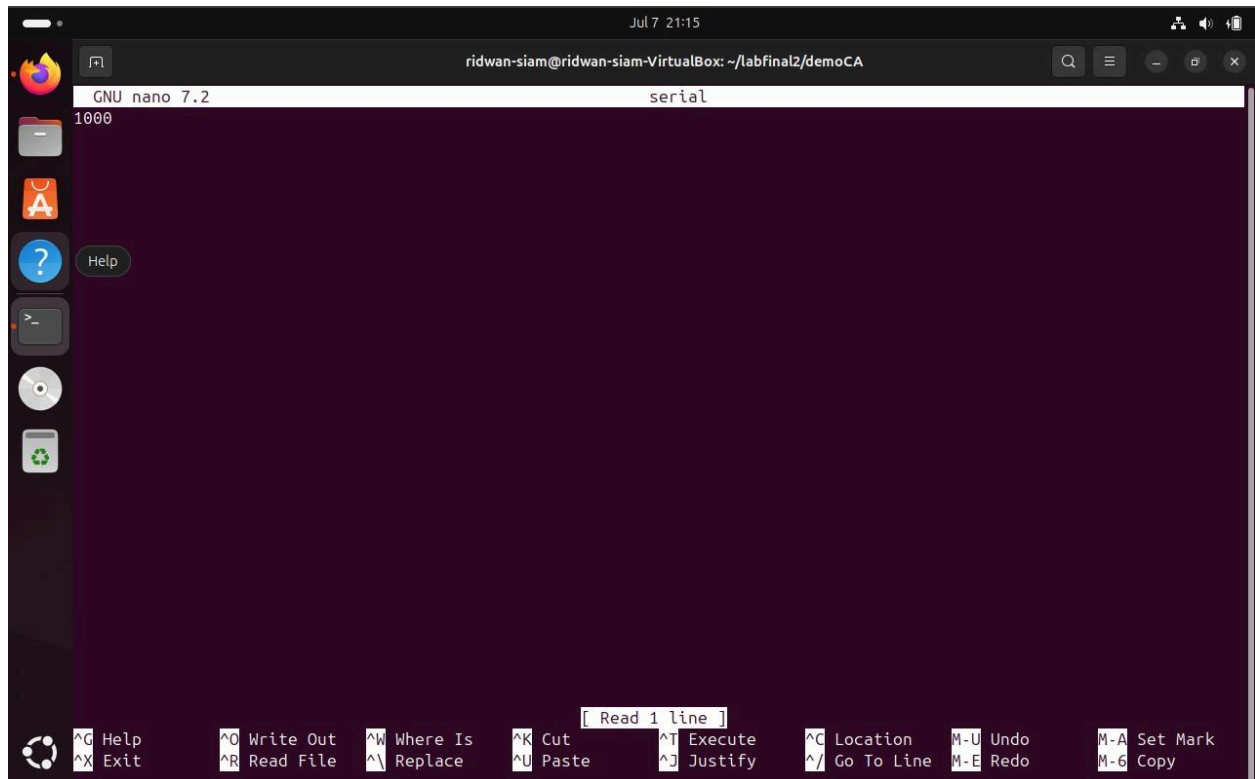
dir                = ./demoCA      # Where everything is kept
certs              = $dir/certs    # Where the issued certs are kept
crl_dir            = $dir/crl      # Where the issued crl are kept
database           = $dir/index.txt # database index file.
#unique_subject    = no           # Set to 'no' 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/cakey.pem # The private key

x509_extensions    = usr_cert       # The extensions to add to the cert

# Comment out the following two lines for the "traditional"
# (and highly broken) format.
```

For the index.txt file, simply create an empty file. For the serial file, put a single number in string format (e.g. 1000) in the file. Once you have set up the configuration file openssl.cnf, you can create and issue certificates.

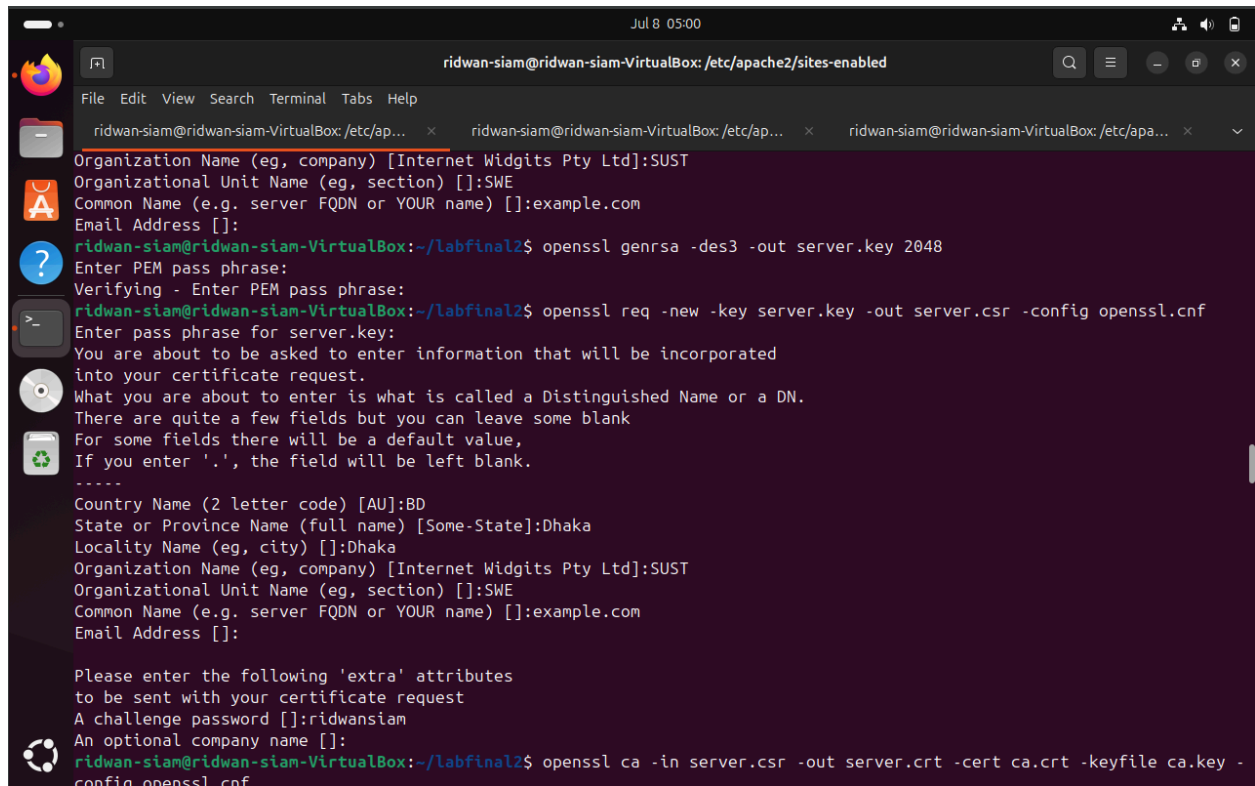


Certificate Authority (CA): As described before, you need to generate a self-signed certificate for our CA. This means that this CA is totally trusted, and its certificate will serve as the root certificate. You can run the following command to generate the self-signed certificate for the CA:

**openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf**







```
Jul 8 05:00
ridwan-siam@ridwan-siam-VirtualBox: /etc/apache2/sites-enabled

Organization Name (eg, company) [Internet Widgits Pty Ltd]:SUST
Organizational Unit Name (eg, section) []:SWE
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ openssl genrsa -des3 -out server.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ openssl req -new -key server.key -out server.csr -config openssl.cnf
Enter pass phrase for server.key:
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]:BD
State or Province Name (full name) [Some-State]:Dhaka
Locality Name (eg, city) []:Dhaka
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SUST
Organizational Unit Name (eg, section) []:SWE
Common Name (e.g. server FQDN or YOUR name) []:example.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:ridwansiam
An optional company name []:
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -
config openssl.cnf
```

Step 3: Generating Certificates. The CSR file needs to have the CA's signature to form a certificate. In the real world, the CSR files are usually sent to a trusted CA for their signature. In this lab, you will use our own trusted CA to generate certificates:

**openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -config openssl.cnf**

```
Jul 7 21:27
ridwan-siam@ridwan-siam-VirtualBox: ~/labfinal2
An optional company name []:
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ openssl ca -in server.csr -out server.crt -cert ca.crt -keyfile ca.key -
config openssl.cnf
Using configuration from openssl.cnf
Enter pass phrase for ca.key:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 4096 (0x1000)
  Validity
    Not Before: Jul  7 15:26:26 2024 GMT
    Not After : Jul  7 15:26:26 2025 GMT
  Subject:
    countryName           = BD
    stateOrProvinceName   = Dhaka
    organizationName      = SUST
    organizationalUnitName = SWE
    commonName            = example.com
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    X509v3 Subject Key Identifier:
      A5:3F:3B:49:51:0C:87:C2:79:05:CC:A4:AB:2A:B4:B3:8A:79:2C:3C
    X509v3 Authority Key Identifier:
      AD:FA:DB:91:5E:5A:DF:BA:1A:04:67:62:33:D2:D8:FA:26:73:F0:98
  Certificate is to be certified until Jul  7 15:26:26 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
```

If OpenSSL refuses to generate certificates, it is very likely that the names in your requests do not match with those of CA. Fix this and re-issue the above command.

Next, let us launch a simple web server with the certificate generated in the previous task. OpenSSL allows us to start a simple web server using the `s_server` command. Use the following steps:

Step 1: Combine the secret key and certificate into one file

**cp server.key server.pem**

**cat server.crt >> server.pem**

Step 2: Launch the web server using server.pem

**openssl s\_server -cert server.pem -www**

By default, the server will listen on port 4433. You can alter that using the `-accept` option.

Now, you can access the server using the following URL: `https://example.com:4433/`. Most likely, you will get an error message from the browser. In Firefox, you will see a message like the following: “example.com:4433 uses an invalid security certificate. The certificate is not trusted because the issuer certificate is unknown”.

```
Jul 8 04:58
ridwan-siam@ridwan-siam-VirtualBox: /etc/apache2/sites-enabled

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Database updated
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ cp server.key server.pem
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ cat server.crt >> server.pem
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ openssl s_server -cert server.pem -www
Enter pass phrase for server.pem:
Enter pass phrase for server.pem:
Using default temp DH parameters
ACCEPT
40B72E905C7E0000:error:0A000418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca:../ssl/record/rec_layer_s3.c:1590:SSL alert number 48
40B72E905C7E0000:error:0A000418:SSL routines:ssl3_read_bytes:tlsv1 alert unknown ca:../ssl/record/rec_layer_s3.c:1590:SSL alert number 48
^C
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ cd
ridwan-siam@ridwan-siam-VirtualBox:~$ cd labfinal2
ridwan-siam@ridwan-siam-VirtualBox:~/labfinal2$ mkdir ~/certs
cd ~/certs
ridwan-siam@ridwan-siam-VirtualBox:~/certs$ pwd
/home/ridwan-siam/certs
ridwan-siam@ridwan-siam-VirtualBox:~/certs$ openssl genrsa -des3 -out myCA.key 2048
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
ridwan-siam@ridwan-siam-VirtualBox:~/certs$ openssl req -x509 -new -nodes -key myCA.key -sha256 -days 1825 -out myCA.pem
Enter pass phrase for myCA.key:
You are about to be asked to enter information that will be incorporated
```

Had this certificate been assigned by VeriSign, you will not have such an error message, because VeriSign's certificate is very likely preloaded into Firefox's certificate repository already. Unfortunately, the certificate of example.com is signed by our own CA (i.e., using ca.crt), and this CA is not recognized by Firefox. There are two ways to get Firefox to accept our CA's self-signed certificate.

You can request Mozilla to include our CA's certificate in its Firefox software, so everybody using Firefox can recognize our CA. This is how the real CAs, such as VeriSign, get their certificates into Firefox. Unfortunately, our own CA does not have a large enough market for Mozilla to include our certificate, so you will not pursue this direction.

Load ca.crt into Firefox: You can manually add our CA's certificate to the Firefox browser by clicking the following menu sequence:

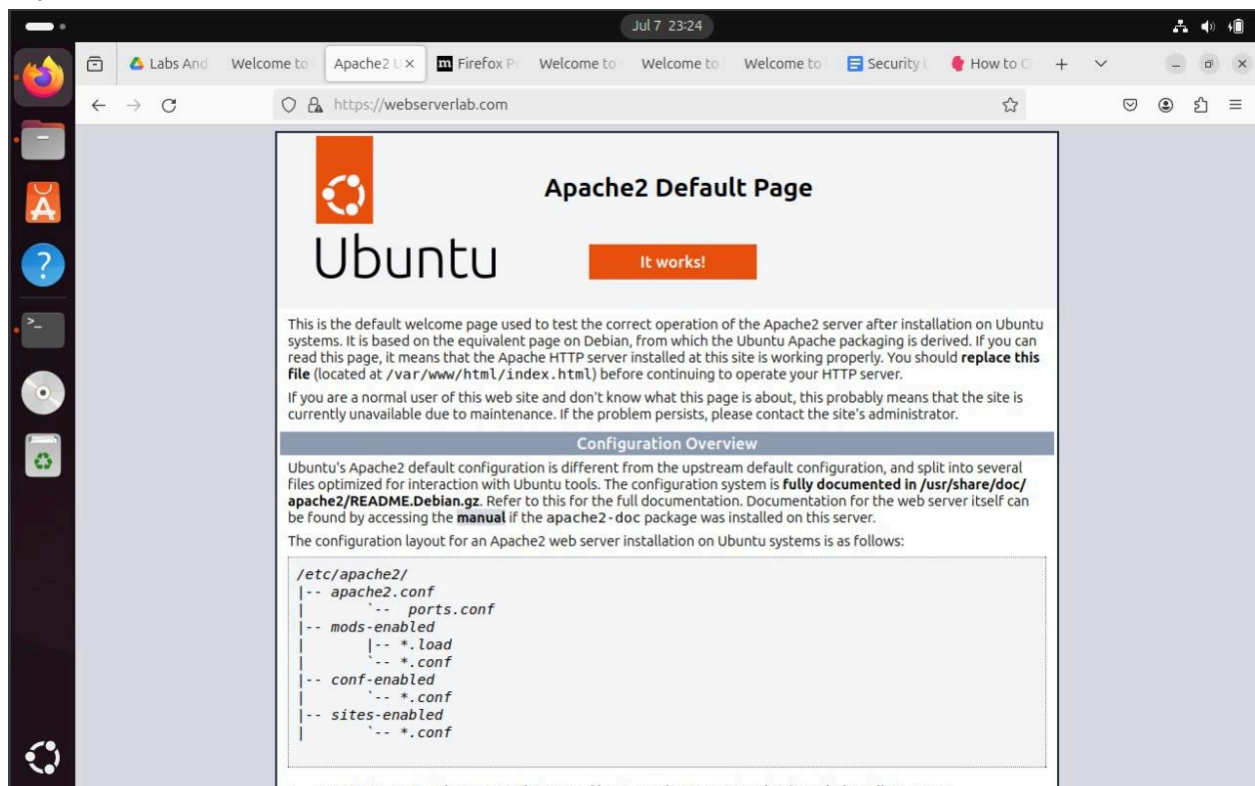
- Preference -> Advanced -> View Certificates

You will see a list of certificates that are already accepted by Firefox. From here, you can "import" our own certificate. Please import ca.crt, and select the following option: "Trust this CA to identify web sites". You will see that our CA's certificate is now in Firefox's list of the accepted certificates. Now, point the browser to <https://example.com:4433>.

Checkpoint – 1 (5 marks): Show this to your course teacher and explain what is happening. Since example.com points to 127.0.0.1, you can also use <https://localhost:4433> to load a web page shown by the OpenSSL server. Please do so, describe and explain your observations.

```
Jul 7 21:36
https://example.com:4433
s_server -cert server.pem -www
Secure Renegotiation IS NOT supported
Ciphers supported in s_server binary
TLSv1.3 : TLS_AES_256_GCM_SHA384 TLSv1.3 : TLS_CHACHA20_POLY1305_SHA256
TLSv1.3 : TLS_AES_128_GCM_SHA256 TLSv1.2 : ECDHE-ECDSA-AES256-GCM-SHA384
TLSv1.2 : ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 : DHE-RSA-AES256-GCM-SHA384
TLSv1.2 : ECDHE-ECDSA-CHACHA20-POLY1305 TLSv1.2 : ECDHE-RSA-CHACHA20-POLY1305
TLSv1.2 : DHE-RSA-CHACHA20-POLY1305 TLSv1.2 : ECDHE-ECDSA-AES128-GCM-SHA256
TLSv1.2 : ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 : DHE-RSA-AES128-GCM-SHA256
TLSv1.2 : ECDHE-ECDSA-AES256-SHA384 TLSv1.2 : ECDHE-RSA-AES256-SHA384
TLSv1.2 : DHE-RSA-AES256-SHA256 TLSv1.2 : ECDHE-ECDSA-AES128-SHA256
TLSv1.2 : ECDHE-RSA-AES128-SHA256 TLSv1.2 : DHE-RSA-AES128-SHA256
TLSv1.0 : ECDHE-ECDSA-AES256-SHA TLSv1.0 : ECDHE-RSA-AES256-SHA
SSLv3 : DHE-RSA-AES256-SHA TLSv1.0 : ECDHE-ECDSA-AES128-SHA
TLSv1.0 : ECDHE-RSA-AES128-SHA SSLv3 : DHE-RSA-AES128-SHA
TLSv1.2 : RSA-PSK-AES256-GCM-SHA384 TLSv1.2 : DHE-PSK-AES256-GCM-SHA384
TLSv1.2 : RSA-PSK-CHACHA20-POLY1305 TLSv1.2 : DHE-PSK-CHACHA20-POLY1305
TLSv1.2 : ECDHE-PSK-CHACHA20-POLY1305 TLSv1.2 : AES256-GCM-SHA384
TLSv1.2 : PSK-AES256-GCM-SHA384 TLSv1.2 : PSK-CHACHA20-POLY1305
TLSv1.2 : RSA-PSK-AES128-GCM-SHA256 TLSv1.2 : DHE-PSK-AES128-GCM-SHA256
TLSv1.2 : AES128-GCM-SHA256 TLSv1.2 : PSK-AES128-GCM-SHA256
TLSv1.2 : AES256-SHA256 TLSv1.2 : AES128-SHA256
TLSv1.0 : ECDHE-PSK-AES256-CBC-SHA384 TLSv1.0 : ECDHE-PSK-AES256-CBC-SHA
SSLv3 : SRP-RSA-AES-256-CBC-SHA SSLv3 : SRP-AES-256-CBC-SHA
TLSv1.0 : RSA-PSK-AES256-CBC-SHA384 TLSv1.0 : DHE-PSK-AES256-CBC-SHA384
SSLv3 : RSA-PSK-AES256-CBC-SHA SSLv3 : DHE-PSK-AES256-CBC-SHA
SSLv3 : AES256-SHA TLSv1.0 : PSK-AES256-CBC-SHA384
SSLv3 : PSK-AES256-CBC-SHA TLSv1.0 : ECDHE-PSK-AES128-CBC-SHA256
TLSv1.0 : ECDHE-PSK-AES128-CBC-SHA SSLv3 : SRP-RSA-AES-128-CBC-SHA
SSLv3 : SRP-AES-128-CBC-SHA TLSv1.0 : RSA-PSK-AES128-CBC-SHA256
TLSv1.0 : DHE-PSK-AES128-CBC-SHA256 SSLv3 : RSA-PSK-AES128-CBC-SHA
SSLv3 : DHE-PSK-AES128-CBC-SHA SSLv3 : AES128-SHA
TLSv1.0 : PSK-AES128-CBC-SHA256 SSLv3 : PSK-AES128-CBC-SHA
---
Ciphers common between both SSL end points:
TLS_AES_128_GCM_SHA256 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_256_GCM_SHA384
ECDHE-ECDSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-GCM-SHA256 ECDHE-ECDSA-CHACHA20-POLY1305
ECDHE-RSA-CHACHA20-POLY1305 ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA ECDHE-ECDSA-AES128-SHA ECDHE-RSA-AES128-SHA
ECDHE-RSA-AES256-SHA AES128-GCM-SHA256 AES256-GCM-SHA384
AES128-SHA AES256-SHA
Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PS5+SHA256:RSA-PS5+SHA384:RSA-PS5+SHA512:RSA+SHA256:RSA+SHA384:RSA+SHA512:ECDSA+SHA1:RSA+SHA1
Shared Signature Algorithms: ECDSA+SHA256:ECDSA+SHA384:ECDSA+SHA512:RSA-PS5+SHA512:RSA-PS5+SHA384:RSA-PS5+SHA256:RSA+SHA256:RSA+SHA384:RSA+SHA512
Supported groups: x25519:secp256r1:secp384r1:secp521r1:ffdhe2048:ffdhe3072
```

Checkpoint – 2 (5 marks): Follow the same instructions for webserverlab.com and show this to your course teacher.



## Task-2: Deploy HTTPS into Apache

Now, you will deploy the HTTPS capability into Apache web server. At first, stop the Openssl webserver launched in the previous task. Now add the following lines into the example configuration file:

Apache is quite modular in the sense it supports the development of additional module which can add extended functionalities. For this lab, you will need to enable the ssl module in Apache which might not be enabled by default. Use the following command to enable the ssl module.

**sudo a2enmod ssl**

Next, use the following command to test the configuration.

**sudo apache2ctl configtest**

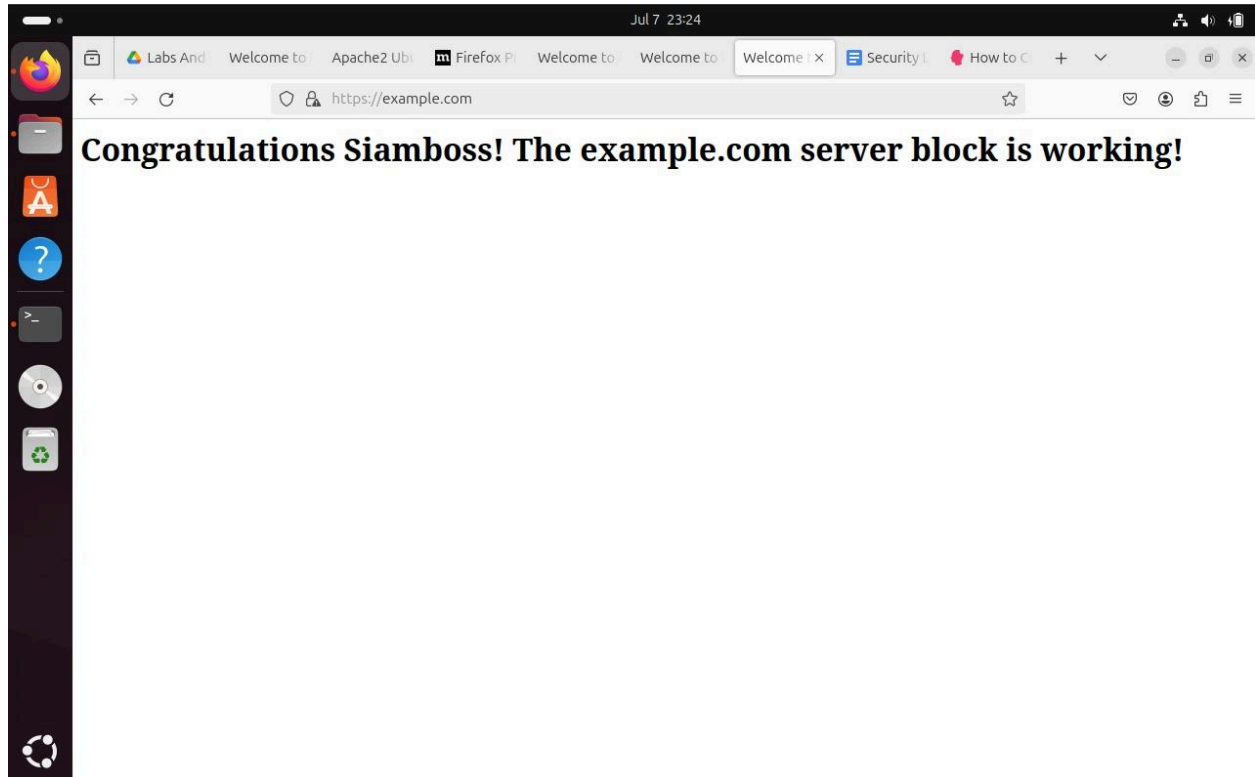
If a syntax is displayed onto the terminal, it indicates everything is okay.

Next restart the apache server using the restart command shown above.

Now, try to access the <https://example.com>. If everything is properly configured, you should be able to view the webpage in HTTPS.

If your browser is Firefox and it shows a warning, you can fix it by importing the CA certificate as described previously. If you use Chrome and it shows a similar warning, you can also import the CA certificate from the Manage certificate option under the Advanced setting in Chrome.

Checkpoint – 3 (5 marks): Access the <https://example.com> in your browser and show it to your teacher.



Checkpoint – 4 (5 marks): Set it up for webserverlab.com. Access it via HTTPS and show it to your teacher.

