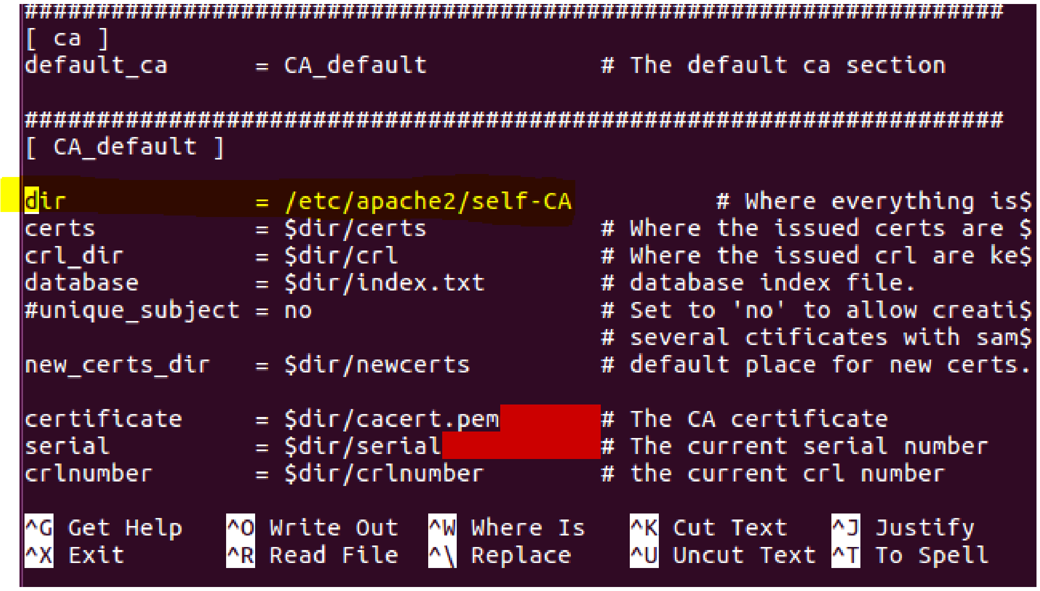
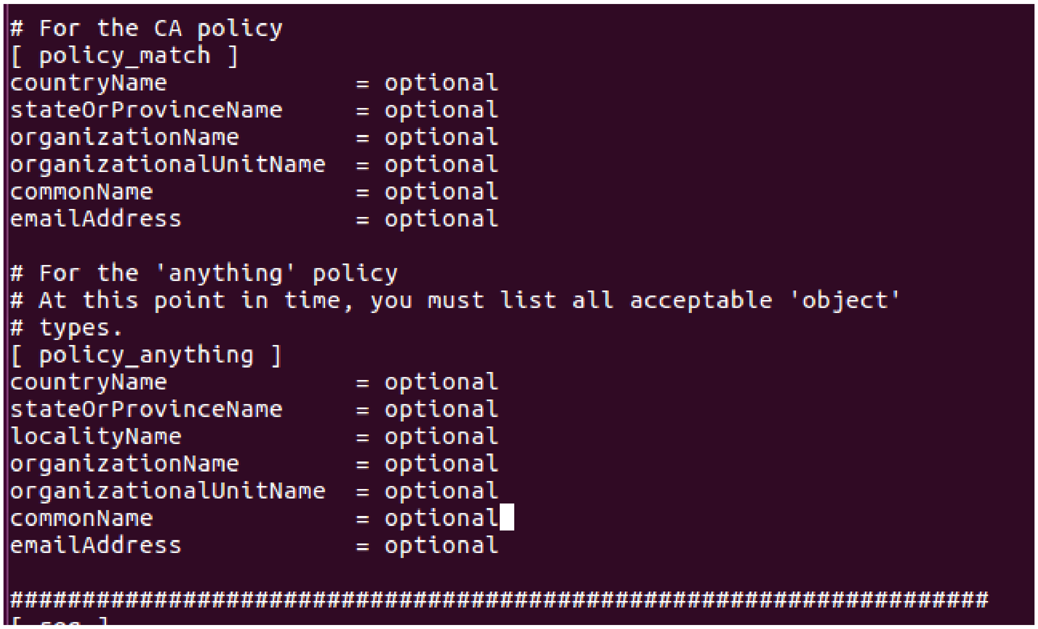
Create a certificate authority using OpenSSL on Linux (Ubuntu)

1. On the terminal verify Openssl is installed. (on Ubuntu it comes by default)
2. Use command “which openssl”
3. Then locate the openssl.cnf file using the command “locate openssl.cnf”.
4. Go to the location /etc/ssl (openssl.cnf is present at this location.)
5. Sudo -s to change to the root mode.
6. Create a directory named self-CA inside /etc/apache2 to store our certificates. Open the openssl.cnf file and change the directory to /etc/apache2/self-CA under the CA_Default section. Openssl will look into this directory for certificates.



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

####################################################################
[ CA_default ]

dir             = /etc/apache2/self-CA          # Where everything is$
certs           = $dir/certs            # Where the issued certs are $
crl_dir         = $dir/crl              # Where the issued crl are ke$
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creati$
                                        # several ctificates with sam$
new_certs_dir   = $dir/newcerts         # default place for new certs.

certificate     = $dir/cacert.pem[redacted]# The CA certificate
serial          = $dir/serial[redacted]# The current serial number
crlnumber       = $dir/crlnumber        # the current crl number

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Uncut Text  ^T To Spell
```

7. Also I have changed the CA policy match to optional (Since this is a lab assignment). But in a production network they have to match.



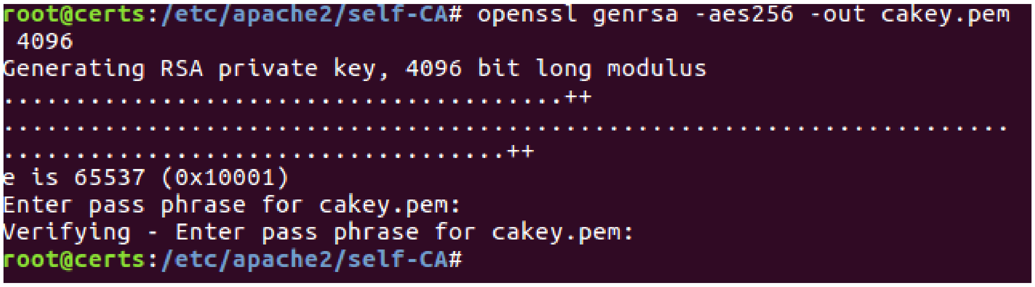
```
# For the CA policy
[ policy_match ]
countryName             = optional
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = optional
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = optional
emailAddress            = optional

####################################################################
[ req ]
```

8. Change the host to “certs.asu.cs.com” using the command - hostname certs.asu.cs.com. Also make the change inside the /etc/hosts file accordingly.

9. Now go to the /etc/apache2/self-CA location. First generate a RSA private key which will be used to sign the root certificates.

   Create root CA private key

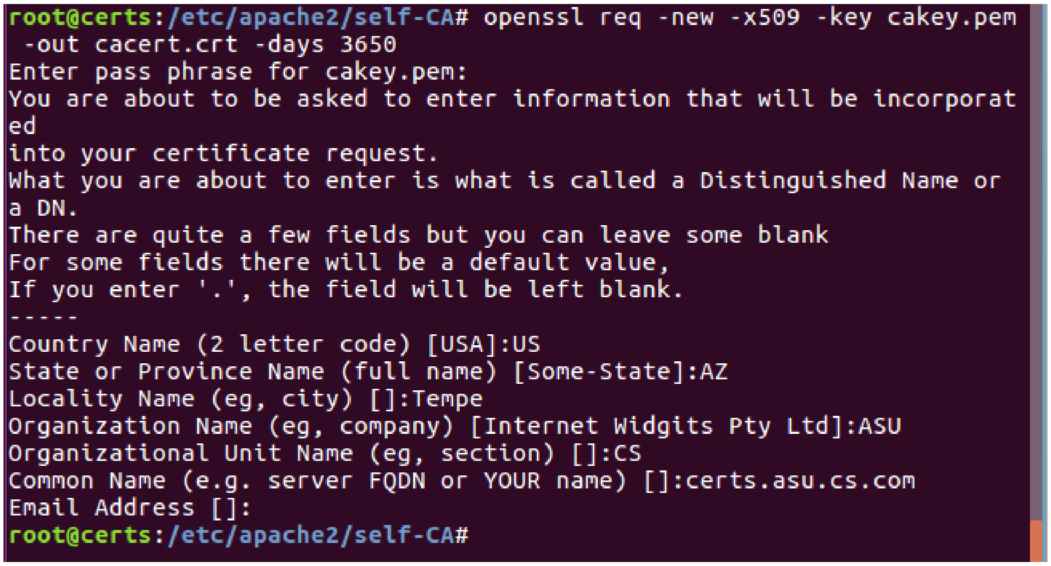
   Command – openssl genrsa –aes256 –out cakey.pem 4096



```
root@certs:/etc/apache2/self-CA# openssl genrsa -aes256 -out cakey.pem
 4096
Generating RSA private key, 4096 bit long modulus
.......................................++
......................................................................
.....................................++
e is 65537 (0x10001)
Enter pass phrase for cakey.pem:
Verifying - Enter pass phrase for cakey.pem:
root@certs:/etc/apache2/self-CA#
```

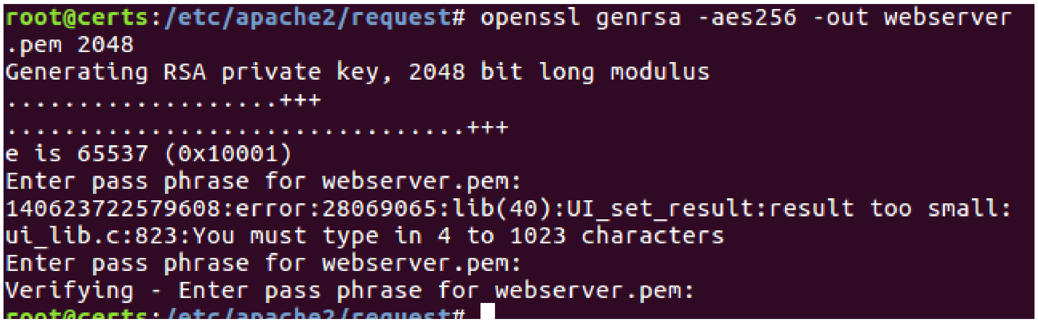
10. Now create a Root Certificate using the private key. While creating the root cert it will ask below details. Root certificate is valid for 10 years.

    command : openssl req –new –x509 –key cakey.pem –out cacert.crt –day 3650



```
root@certs:/etc/apache2/self-CA# openssl req -new -x509 -key cakey.pem
 -out cacert.crt -days 3650
Enter pass phrase for cakey.pem:
You are about to be asked to enter information that will be incorporat
ed
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) [USA]:US
State or Province Name (full name) [Some-State]:AZ
Locality Name (eg, city) []:Tempe
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ASU
Organizational Unit Name (eg, section) []:CS
Common Name (e.g. server FQDN or YOUR name) []:certs.asu.cs.com
Email Address []:
root@certs:/etc/apache2/self-CA#
```

11. **A webserver will request the root CA for a certificate where the webserver will have its public key. First, the webserver will create its private key.**

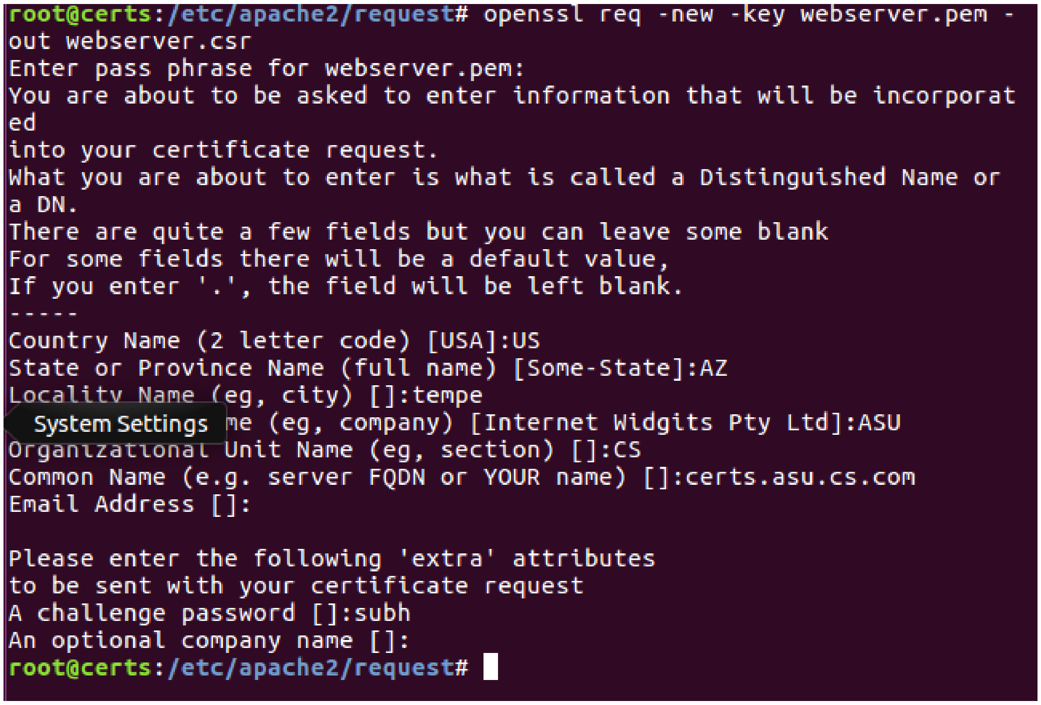


```
root@certs:/etc/apache2/request# openssl genrsa -aes256 -out webserver
.pem 2048
Generating RSA private key, 2048 bit long modulus
..................+++
...............................+++
e is 65537 (0x10001)
Enter pass phrase for webserver.pem:
140623722579608:error:28069065:lib(40):UI_set_result:result too small:
ui_lib.c:823:You must type in 4 to 1023 characters
Enter pass phrase for webserver.pem:
Verifying - Enter pass phrase for webserver.pem:
root@certs:/etc/apache2/request#
```

(I have created a request folder where the webserver will generate its private key)

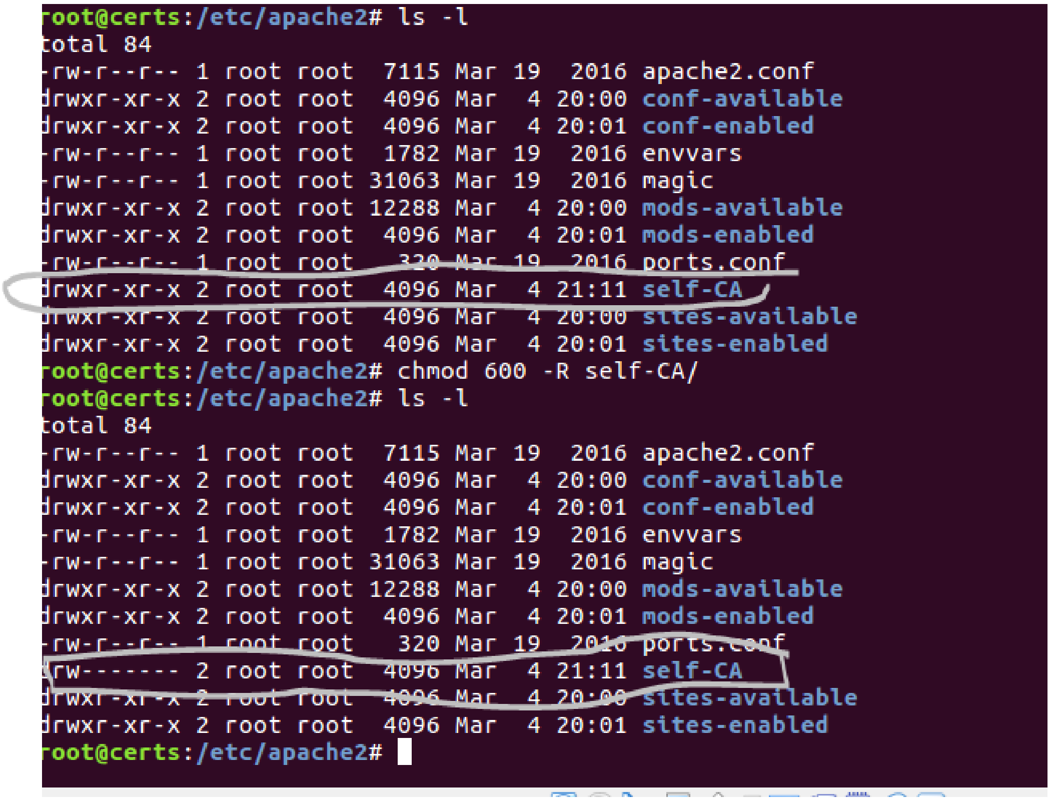
12. **Generate CSR: The webserver generates a certificate signing request using its private key.**

    command: openssl req –new –key webserver.pem –out webserver.csr



```
root@certs:/etc/apache2/request# openssl req -new -key webserver.pem -
out webserver.csr
Enter pass phrase for webserver.pem:
You are about to be asked to enter information that will be incorporat
ed
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) [USA]:US
State or Province Name (full name) [Some-State]:AZ
Locality Name (eg, city) []:tempe
               me (eg, company) [Internet Widgits Pty Ltd]:ASU
Organizational Unit Name (eg, section) []:CS
Common Name (e.g. server FQDN or YOUR name) []:certs.asu.cs.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:subh
An optional company name []:
root@certs:/etc/apache2/request#
```

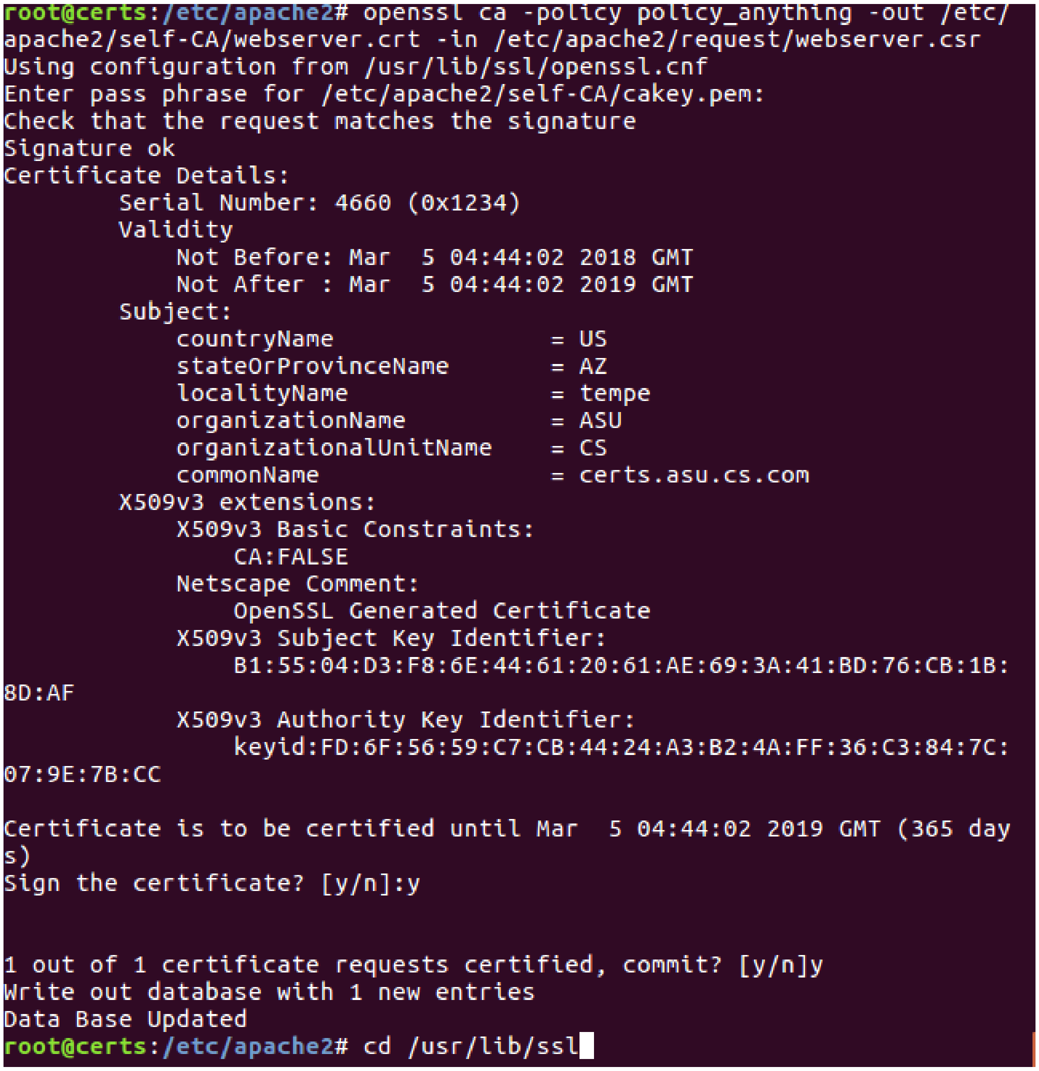
13. Change the permission of the self-CA folder so that no one else can make any changes.



```
root@certs:/etc/apache2# ls -l
total 84
-rw-r--r-- 1 root root  7115 Mar 19  2016 apache2.conf
drwxr-xr-x 2 root root  4096 Mar  4 20:00 conf-available
drwxr-xr-x 2 root root  4096 Mar  4 20:01 conf-enabled
-rw-r--r-- 1 root root  1782 Mar 19  2016 envvars
-rw-r--r-- 1 root root 31063 Mar 19  2016 magic
drwxr-xr-x 2 root root 12288 Mar  4 20:00 mods-available
drwxr-xr-x 2 root root  4096 Mar  4 20:01 mods-enabled
-rw-r--r-- 1 root root   320 Mar 19  2016 ports.conf
drwxr-xr-x 2 root root  4096 Mar  4 21:11 self-CA
drwxr-xr-x 2 root root  4096 Mar  4 20:00 sites-available
drwxr-xr-x 2 root root  4096 Mar  4 20:01 sites-enabled
root@certs:/etc/apache2# chmod 600 -R self-CA/
root@certs:/etc/apache2# ls -l
total 84
-rw-r--r-- 1 root root  7115 Mar 19  2016 apache2.conf
drwxr-xr-x 2 root root  4096 Mar  4 20:00 conf-available
drwxr-xr-x 2 root root  4096 Mar  4 20:01 conf-enabled
-rw-r--r-- 1 root root  1782 Mar 19  2016 envvars
-rw-r--r-- 1 root root 31063 Mar 19  2016 magic
drwxr-xr-x 2 root root 12288 Mar  4 20:00 mods-available
drwxr-xr-x 2 root root  4096 Mar  4 20:01 mods-enabled
-rw-r--r-- 1 root root   320 Mar 19  2016 ports.conf
drw------- 2 root root  4096 Mar  4 21:11 self-CA
drwxr-xr-x 2 root root  4096 Mar  4 20:00 sites-available
drwxr-xr-x 2 root root  4096 Mar  4 20:01 sites-enabled
root@certs:/etc/apache2#
```

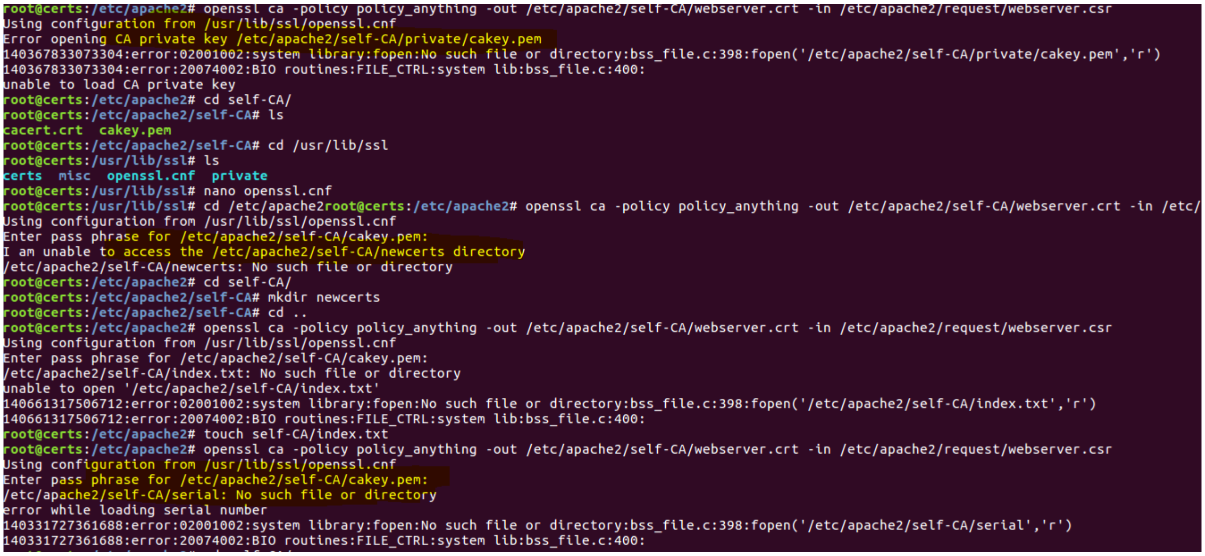
14. Now the Root-CA has to sign the Certificate Signing Request from the webserver.
command: openssl ca –policy policy_anything –out /etc/apache2/self-CA/webserver.crt –in /etc/apache2/request/webserver.csr

Since it is a test CA, we are putting policy_anything.



```
root@certs:/etc/apache2# openssl ca -policy policy_anything -out /etc/
apache2/self-CA/webserver.crt -in /etc/apache2/request/webserver.csr
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/apache2/self-CA/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 4660 (0x1234)
        Validity
            Not Before: Mar  5 04:44:02 2018 GMT
            Not After : Mar  5 04:44:02 2019 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = AZ
            localityName              = tempe
            organizationName          = ASU
            organizationalUnitName    = CS
            commonName                = certs.asu.cs.com
        X509v3 extensions:
            X509v3 Basic Constraints: 
                CA:FALSE
            Netscape Comment: 
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier: 
                B1:55:04:D3:F8:6E:44:61:20:61:AE:69:3A:41:BD:76:CB:1B:
8D:AF
            X509v3 Authority Key Identifier: 
                keyid:FD:6F:56:59:C7:CB:44:24:A3:B2:4A:FF:36:C3:84:7C:
07:9E:7B:CC

Certificate is to be certified until Mar  5 04:44:02 2019 GMT (365 day
s)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
root@certs:/etc/apache2# cd /usr/lib/ssl
```

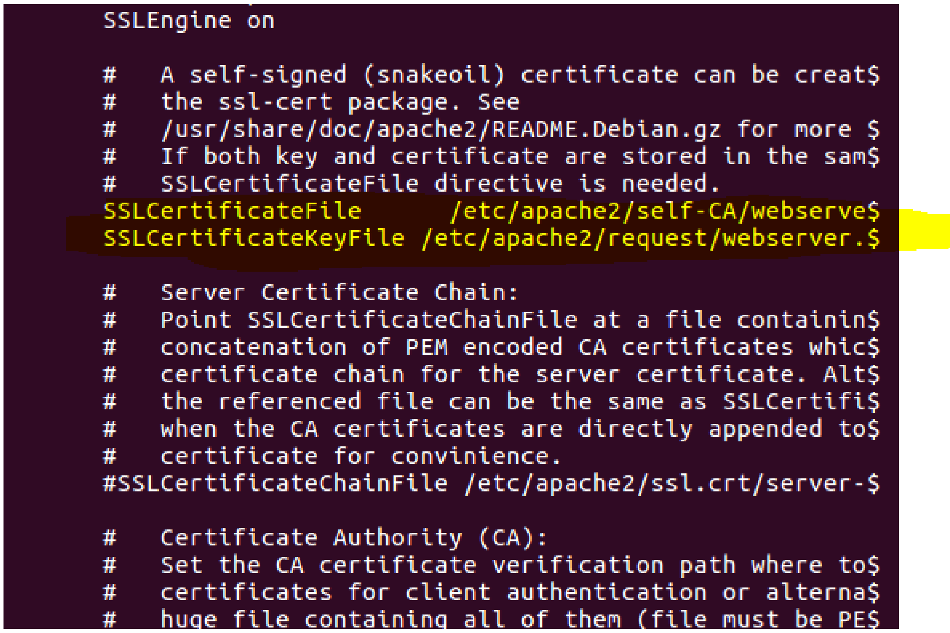
15. I realized few things while signing the csr as the root ca. First, create folders to hold certs, requests as per the structure of openssl.cnf file with in /usr/lib/ssl. I faced many errors since those folders were not present while signing the csr.



```
root@certs:/etc/apache2# openssl ca -policy policy_anything -out /etc/apache2/self-CA/webserver.crt -in /etc/apache2/request/webserver.csr
Using configuration from /usr/lib/ssl/openssl.cnf
Error opening CA private key /etc/apache2/self-CA/private/cakey.pem
140367833073304:error:02001002:system library:fopen:No such file or directory:bss_file.c:398:fopen('/etc/apache2/self-CA/private/cakey.pem','r')
140367833073304:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:400:
unable to load CA private key
root@certs:/etc/apache2# cd self-CA/
root@certs:/etc/apache2/self-CA# ls
cacert.crt  cakey.pem
root@certs:/etc/apache2/self-CA# cd /usr/lib/ssl
root@certs:/usr/lib/ssl# ls
certs  misc  openssl.cnf  private
root@certs:/usr/lib/ssl# nano openssl.cnf
root@certs:/usr/lib/ssl# cd /etc/apache2root@certs:/etc/apache2# openssl ca -policy policy_anything -out /etc/apache2/self-CA/webserver.crt -in /etc/
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/apache2/self-CA/cakey.pem:
I am unable to access the /etc/apache2/self-CA/newcerts directory
/etc/apache2/self-CA/newcerts: No such file or directory
root@certs:/etc/apache2# cd self-CA/
root@certs:/etc/apache2/self-CA# mkdir newcerts
root@certs:/etc/apache2/self-CA# cd ..
root@certs:/etc/apache2# openssl ca -policy policy_anything -out /etc/apache2/self-CA/webserver.crt -in /etc/apache2/request/webserver.csr
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/apache2/self-CA/cakey.pem:
/etc/apache2/self-CA/index.txt: No such file or directory
unable to open '/etc/apache2/self-CA/index.txt'
140661317506712:error:02001002:system library:fopen:No such file or directory:bss_file.c:398:fopen('/etc/apache2/self-CA/index.txt','r')
140661317506712:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:400:
root@certs:/etc/apache2# touch self-CA/index.txt
root@certs:/etc/apache2# openssl ca -policy policy_anything -out /etc/apache2/self-CA/webserver.crt -in /etc/apache2/request/webserver.csr
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/apache2/self-CA/cakey.pem:
/etc/apache2/self-CA/serial: No such file or directory
error while loading serial number
140331727361688:error:02001002:system library:fopen:No such file or directory:bss_file.c:398:fopen('/etc/apache2/self-CA/serial','r')
140331727361688:error:20074002:BIO routines:FILE_CTRL:system lib:bss_file.c:400:
```

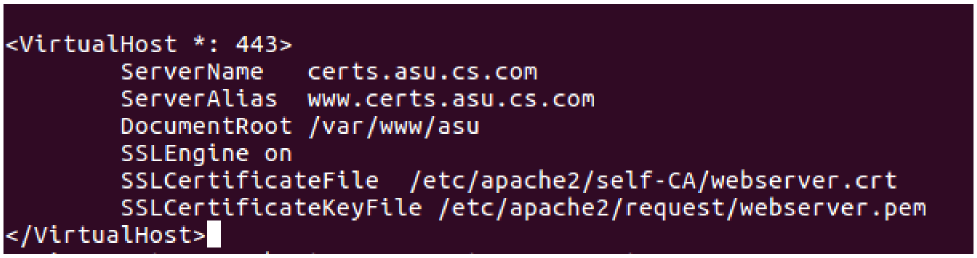
I had to create missing folders and I also had to make some adjustments in the openssl.cnf file within /usr/lib/ssl

16. Configure the web server to use the certificate.
    Go to the path /etc/apache2/sites-available. Open default-ssl.cnf file.
    (*apache2.conf need not to be modified.)
    Edit the path to the certificate and the private key of the webserver.



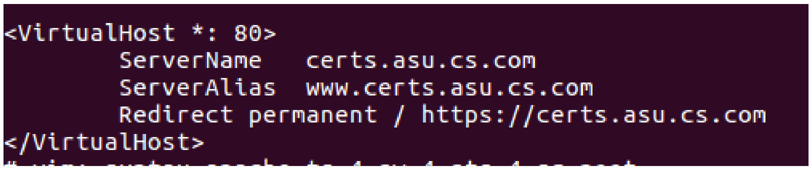
```
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be creat$
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more $
        #   If both key and certificate are stored in the sam$
        #   SSLCertificateFile directive is needed.
        SSLCertificateFile      /etc/apache2/self-CA/webserve$
        SSLCertificateKeyFile /etc/apache2/request/webserver.$

        #   Server Certificate Chain:
        #   Point SSLCertificateChainFile at a file containin$
        #   concatenation of PEM encoded CA certificates whic$
        #   certificate chain for the server certificate. Alt$
        #   the referenced file can be the same as SSLCertifi$
        #   when the CA certificates are directly appended to$
        #   certificate for convinience.
        #SSLCertificateChainFile /etc/apache2/ssl.crt/server-$

        #   Certificate Authority (CA):
        #   Set the CA certificate verification path where to$
        #   certificates for client authentication or alterna$
        #   huge file containing all of them (file must be PE$
```

17. To the 000-deafult.conf file inside /etc/apache2/sites-available, add a virtualhost as below.



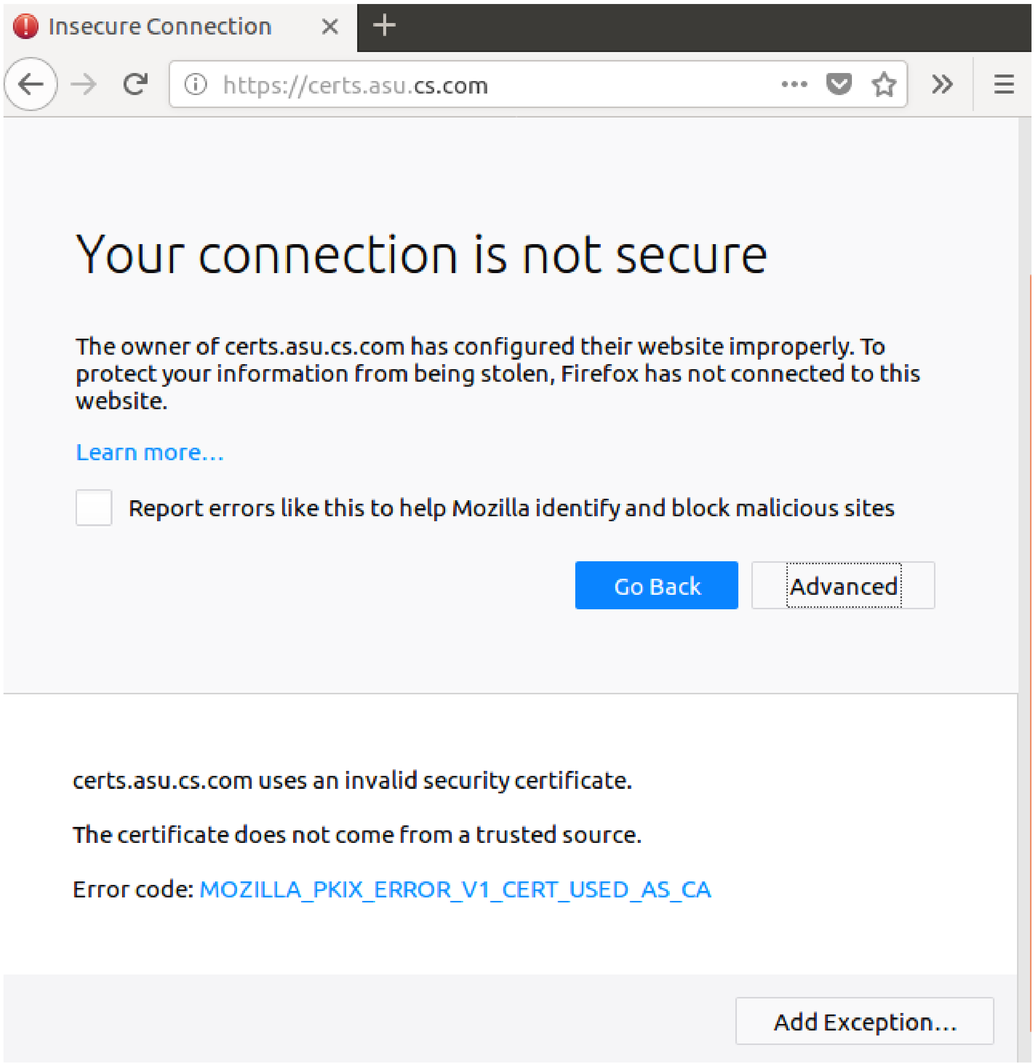
```
<VirtualHost *: 443>
        ServerName   certs.asu.cs.com
        ServerAlias  www.certs.asu.cs.com
        DocumentRoot /var/www/asu
        SSLEngine on
        SSLCertificateFile  /etc/apache2/self-CA/webserver.crt
        SSLCertificateKeyFile /etc/apache2/request/webserver.pem
</VirtualHost>
```

18. Add following for redirection of HTTP to HTTPS.



```
<VirtualHost *: 80>
        ServerName   certs.asu.cs.com
        ServerAlias  www.certs.asu.cs.com
        Redirect permanent / https://certs.asu.cs.com
</VirtualHost>
```

19. Since it is a self-signed certificate, the browser shows the warning. But go ahead and add to exception.



Your connection is not secure

The owner of certs.asu.cs.com has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

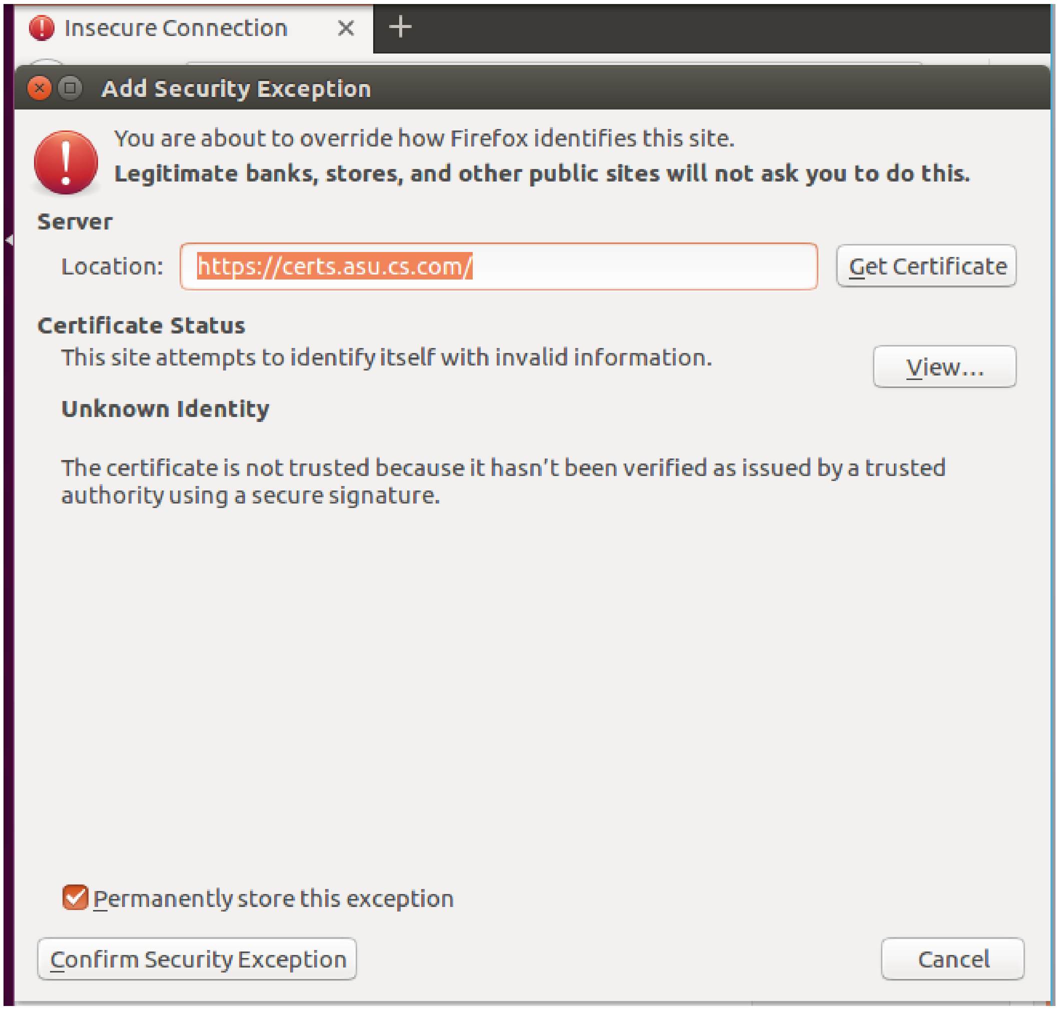
Learn more…

□ Report errors like this to help Mozilla identify and block malicious sites

certs.asu.cs.com uses an invalid security certificate.

The certificate does not come from a trusted source.

Error code: MOZILLA_PKIX_ERROR_V1_CERT_USED_AS_CA



You are about to override how Firefox identifies this site.

Legitimate banks, stores, and other public sites will not ask you to do this.

Server

Location: https://certs.asu.cs.com/

Certificate Status

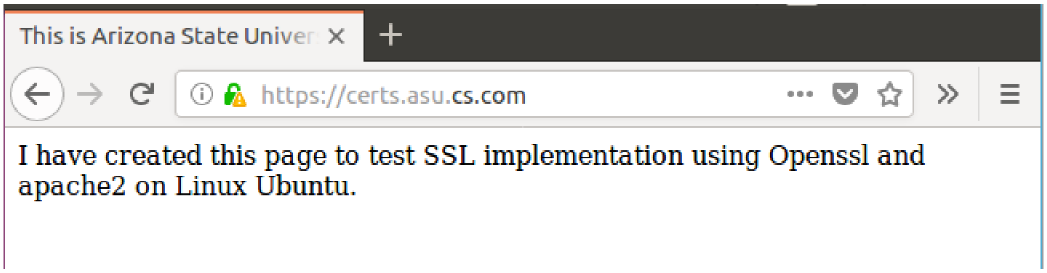
This site attempts to identify itself with invalid information.

Unknown Identity

The certificate is not trusted because it hasn't been verified as issued by a trusted authority using a secure signature.

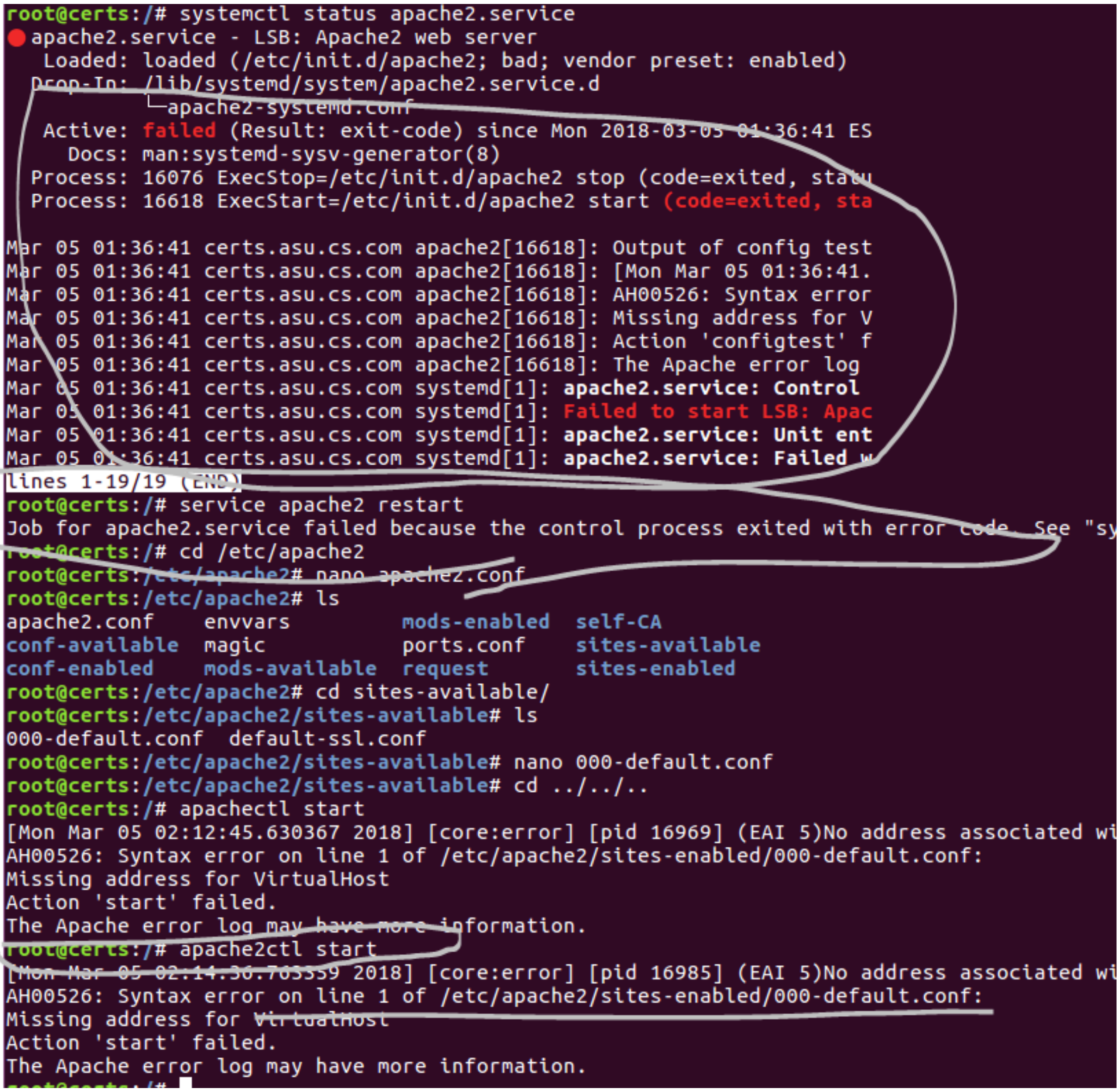
☑ Permanently store this exception

Now I am able to access my test website securely (https).



I have created this page to test SSL implementation using Openssl and apache2 on Linux Ubuntu.

Lesson Learned: many times I faced the error “Unable to start LSB” while restarting the apache2. It may happen due to wrong configuration in any of the apache conf file. Use apache2ctl start command every time you make any changes to conf file.



```
root@certs:/# systemctl status apache2.service
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: failed (Result: exit-code) since Mon 2018-03-05 01:36:41 ES
     Docs: man:systemd-sysv-generator(8)
  Process: 16076 ExecStop=/etc/init.d/apache2 stop (code=exited, statu
  Process: 16618 ExecStart=/etc/init.d/apache2 start (code=exited, sta

Mar 05 01:36:41 certs.asu.cs.com apache2[16618]: Output of config test
Mar 05 01:36:41 certs.asu.cs.com apache2[16618]: [Mon Mar 05 01:36:41.
Mar 05 01:36:41 certs.asu.cs.com apache2[16618]: AH00526: Syntax error
Mar 05 01:36:41 certs.asu.cs.com apache2[16618]: Missing address for V
Mar 05 01:36:41 certs.asu.cs.com apache2[16618]: Action 'configtest' f
Mar 05 01:36:41 certs.asu.cs.com apache2[16618]: The Apache error log
Mar 05 01:36:41 certs.asu.cs.com systemd[1]: apache2.service: Control
Mar 05 01:36:41 certs.asu.cs.com systemd[1]: Failed to start LSB: Apac
Mar 05 01:36:41 certs.asu.cs.com systemd[1]: apache2.service: Unit ent
Mar 05 01:36:41 certs.asu.cs.com systemd[1]: apache2.service: Failed w
lines 1-19/19 (END)
root@certs:/# service apache2 restart
Job for apache2.service failed because the control process exited with error code. See "sy
root@certs:/# cd /etc/apache2
root@certs:/etc/apache2# nano apache2.conf
root@certs:/etc/apache2# ls
apache2.conf    envvars          mods-enabled  self-CA
conf-available  magic            ports.conf    sites-available
conf-enabled    mods-available   request       sites-enabled
root@certs:/etc/apache2# cd sites-available/
root@certs:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf
root@certs:/etc/apache2/sites-available# nano 000-default.conf
root@certs:/etc/apache2/sites-available# cd ../../..
root@certs:/# apachectl start
[Mon Mar 05 02:12:45.630367 2018] [core:error] [pid 16969] (EAI 5)No address associated wi
AH00526: Syntax error on line 1 of /etc/apache2/sites-enabled/000-default.conf:
Missing address for VirtualHost
Action 'start' failed.
The Apache error log may have more information.
root@certs:/# apache2ctl start
[Mon Mar 05 02:14:36.703359 2018] [core:error] [pid 16985] (EAI 5)No address associated wi
AH00526: Syntax error on line 1 of /etc/apache2/sites-enabled/000-default.conf:
Missing address for VirtualHost
Action 'start' failed.
The Apache error log may have more information.
root@certs:/#
```

How to remove apache on Ubuntu:

Sudo -s

Service apache2 stop

`apt-get purge apache2 apache2-utils apache2.2-bin apache2-common`

`apt-get autoremove`

Finally, check if there is any configuration files or manual pages belonging to Apache2, which are still not removed.

Whereis apache2

Remove all of them using command : rm –rf /etc/apache2   (each path)

How to install apache2 on Ubuntu:

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

```
sudo a2dismod mpm_event
sudo a2enmod mpm_prefork
```

Openssl key generation https://jamielinux.com/docs/openssl-certificate-authority/online-certificate-status-protocol.html

https://rietta.com/blog/2012/01/27/openssl-generating-rsa-key-from-command/
Openssl CA https://www.youtube.com/watch?v=oCl0gzLPPMI
https://www.youtube.com/watch?v=Ei-ah2ruEkM

A very good tutorial on how to enable virtual host in apache - https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-16-04