Installing local SSL certificate

https://hostadvice.com/how-to/configure-apache-with-tls-ssl-certificate-on-ubuntu-18/

https://reactpaths.com/how-to-get-https-working-in-localhost-development-environment-f17de34af046

https://www.freecodecamp.org/news/how-to-get-https-working-on-your-local-development-environment-in-5-minutes-7af615770eec/

https://www.digicert.com/kb/csr-ssl-installation/ubuntu-server-with-apache2-openssl.htm

## Step 1:  Generate Certificate

1. Create a directory place to store the file

        $ mkdir ~/certificates

        $ cd ~/certificates

2. Generate a CSR and private key using following command

openssl genrsa -des3 -out rootCA.key 2048

openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem

After successfully running the command it will ask for the information of certificate request. Complete it using the appropriate information.

Country Name (2 letter code) [AU]: US  
State or Province Name (full name) [Some-State]: FL  
Locality Name (eg, city) []: Miami  
Organization Name (eg, company) [My Company]: My Company  
Organizational Unit Name (eg, section) []:

The common name is your domain name or the server IP address.

Common Name (e.g. server FQDN or YOUR name) []: 192.168.2.3  
Email Address []:dummy@example.com

3. Now move the certificate into the same folder you created using the following commands

$ mkdir /etc/apache2/ssl

$ mv ~/certificates/\* /etc/apache2/ssl/.

4. We are done creating the certificate now we will make the certificate work with Apache.

# Step 2: Install SSL certificate in Ubuntu

sudo mkdir /usr/local/share/ca-certificates/extra

sudo cp rootCA.pem \

/usr/local/share/ca-certificates/extra/rootCA.crt

sudo update-ca-certificates

## Install certificate file CentOS:

Cp rootCA.pem *etc*/pki/ca-trust/source/anchors

updae-ca-trust extract

## Step 3: Domain SSL certificate

The root SSL certificate can now be used to issue a certificate specifically for your local development environment located at localhost.

Create a new OpenSSL configuration file server.csr.cnf so you can import these settings when creating a certificate instead of entering them on the command line.

[req]

default\_bits = 2048

prompt = no

default\_md = sha256

distinguished\_name = dn

[dn]

C=US

ST=RandomState

L=RandomCity

O=RandomOrganization

OU=RandomOrganizationUnit

emailAddress=hello@example.com

CN = localhost

Create a v3.ext file in order to create a [X509 v3 certificate](https://en.wikipedia.org/wiki/X.509). Notice how we’re specifying subjectAltName here.

authorityKeyIdentifier=keyid,issuer

basicConstraints=CA:FALSE

keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment

subjectAltName = @alt\_names

[alt\_names]

DNS.1 = localhost

Create a certificate key for localhost using the configuration settings stored in server.csr.cnf. This key is stored in server.key.

openssl req -new -sha256 -nodes -out server.csr -newkey rsa:2048 -keyout server.key -config <( cat server.csr.cnf )

A certificate signing request is issued via the root SSL certificate we created earlier to create a domain certificate for localhost. The output is a certificate file called server.crt.

openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.csr -days 500 -sha256 -extfile v3.ext

Create CRT:

openssl x509 -req -days 365 -in ca.csr -signkey ca.key -out ca.crt



## Step 4: Configuring the Firewall

1. Make sure that the TCP port 443 is open. Instead of port 80 SSL uses port 443. We will be using Uncomplicated Firewall (UFW)
2. To enable UFW use the following command

$ sudo ufw enable

1. Allow the predefined apache settings for the firewall using the following command

$ sudo ufw allow 'Apache Full'

$ sudo ufw allo “OpenSSH’

1. You  can check the list of current rules using the following command.

$ sudo ufw status

The configuration should be similar to this:

**To                         Action      From  
--                         ------      ----  
Apache Full                ALLOW       Anywhere  
OpenSSH                    ALLOW       Anywhere  
Apache Full (v6)           ALLOW       Anywhere (v6)  
OpenSSH (v6)               ALLOW       Anywhere (v6)**

1. To allow future connections allow OpenSSH

$ sudo ufw allow 'OpenSSH'

## Step 5: Apache virtual host configuration

1. Navigate to the default Apache site config directory using the following command

$ sudo nano /etc/apache2/sites-available/default-ssl.conf

This config file tells the server where to find SSL certificate. It should look like this:

**<IfModule mod\_ssl.c>  
<VirtualHost \_default\_:443>  
ServerAdmin webmaster@localhost**

DocumentRoot /var/www/html

ErrorLog ${APACHE\_LOG\_DIR}/error.log  
CustomLog ${APACHE\_LOG\_DIR}/access.log combined

SSLEngine on

SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem  
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

<FilesMatch ".(cgi|shtml|phtml|php)$">  
SSLOptions +StdEnvVars  
</FilesMatch>  
<Directory /usr/lib/cgi-bin>  
SSLOptions +StdEnvVars  
</Directory>

**</VirtualHost>  
</IfModule>** 1. Edit this: ServerAdmin webmaster@localhost to this :

ServerAdmin email@example.net

1. Add this right below the ServerAdmin line:

ServerName ADD\_YOUR\_IP\_OR\_DOMAIN\_NAME\_HERE

1. Now, edit these lines with our certificate location:

SSLCertificateFile    /etc/apache2/ssl/server.crt  
SSLCertificateKeyFile /etc/apache2/ssl/server.key

Our file should look like this:

**<IfModule mod\_ssl.c>  
<VirtualHost \_default\_:443>  
ServerAdmin email@example.net  
ServerName 203.0.113.122**

DocumentRoot /var/www/html

ErrorLog ${APACHE\_LOG\_DIR}/error.log  
CustomLog ${APACHE\_LOG\_DIR}/access.log combined

SSLEngine on

SSLCertificateFile    /etc/apache2/ssl/apache.crt  
SSLCertificateKeyFile /etc/apache2/ssl/apache.key

<FilesMatch ".(cgi|shtml|phtml|php)$">  
SSLOptions +StdEnvVars  
</FilesMatch>  
<Directory /usr/lib/cgi-bin>  
SSLOptions +StdEnvVars  
</Directory>

**</VirtualHost>  
</IfModule>**

1. Save the file, and close it.

## Step 6: Enable the Apache SSL module

1. Enable the SSL module using following command

$ sudo a2enmod ssl

1. Now enable the site we have just edited:

$ sudo a2ensite default-ssl.conf

1. Restart Apache:

$ sudo service apache2 restart