# Setting up Apache HTTP Server with SSL support, self-signed certificate and virtual hosts on Ubuntu

I needed to set up a server to host a secure helpdesk application. So, first i installed httpd (the Apache HTTP Server) and configured it to allow SSL connections.

Installing Apache wasn't hard with Ubuntus software center. I have chosen to install the whole LAMP stack

```
$ sudo apt-get update
$ sudo apt-get install lamp-server
```

I tested the installation was OK, I opened http://apacheserver in the browser. The browser displayed that Apache 'It works!' page!

The module mod_ssl (http://httpd.apache.org/docs/2.0/mod/mod_ssl.html) provides SSL/TLS support to httpd. It is available in the httpd installation as a part of the apache2-common package. On Ubuntu, use the following command to enable SSL

```
$ sudo a2ensite default-ssl
$ sudo service apache2 restart
```

I tested the installation was OK, I opened https://apacheserver in the browser. The browser, again, displayed that Apache 'It works!' page!

To use a self-signed certificate, the package ssl-cert must be installed, which it was on my installation. I wanted to configure my own self-signed certificate for the server and to store it in /etc/apache2/ssl. To do so, run the following command from the terminal:

```
$ sudo mkdir /etc/apache2/ssl
$ sudo /usr/sbin/make-ssl-cert /usr/share/ssl-cert/ssleay.cnf
/etc/apache2/ssl/apache.crt
```

The command prompts you to enter the hostname to use in the certificate. Once done, you can now see that there is a new file in the /etc/apache2/ssl directory:

```
drwxr-xr-x 2 root root 4096 2011-12-16 14:40 ./
drwxr-xr-x 8 root root 4096 2011-12-16 14:12 ../
lrwxrwxrwx 1 root root 10 2011-12-16 14:40 a9630d61 -> apache.crt
-rw——- 1 root root 2685 2011-12-16 14:40 apache.crt
```

That last command will have generated an apache.crt file that contains both the certificate and the key. Let’s now separate that file into two files:

1. apache.pem to store the certificate
2. apache.key to store the key

I will simply copy the original apache.crt file twice, one with each name and edit each file.

```
$ cd /etc/apache2/ssl
$ sudo cp apache.crt apache.pem
$ sudo cp apache.crt apache.key
```

The apache.pem file must contain everything from the beginning line to the ending line of the certificate

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

The apache.key file must contain everything from the beginning line to the ending line of the key

```
-----BEGIN PRIVATE KEY-----
...
-----END PRIVATE KEY-----
```

Now, I have to configure httpd to use my new certificate. To do so, I edit the configuration with text editor of your choice

```
$ sudo vim /etc/apache2/sites-enabled/default-ssl
```

We have to update the following two lines

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

With the following two lines

```
SSLCertificateFile /etc/apache2/ssl/apache.pem
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

The private key shall only be readable by root:

```
$ sudo chmod 600 /etc/apache2/ssl/apache.key
```

Let’s now restart Apache2 again

```
$ sudo /etc/init.d/apache2 restart
```

If you need a virtual host, create a virtual host in /etc/apache2/sites-avaliable, and edit it:

```
$ cd /etc/apache2/sites-avaliable
$ touch example.com.ssl.conf
$ vim example.com.ssl.conf
```

Paste the following:

```
<IfModule mod_ssl.c>
        <VirtualHost *:443>
                ServerAdmin root@example.com
                ServerName example.com
                ServerAlias www.example.com
                DocumentRoot /var/www/example.com

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

                SSLEngine on
                SSLCertificateFile      /etc/ssl/examplecom.crt
                SSLCertificateKeyFile   /etc/ssl/examplecom.key
                SSLCertificateChainFile /etc/ssl/sub.class1.server.ca.pem
                <FilesMatch "\.(cgi|shtml|phtml|php)$">
                                SSLOptions +StdEnvVars
                </FilesMatch>
                <Directory /usr/lib/cgi-bin>
                                SSLOptions +StdEnvVars
                </Directory>
                BrowserMatch "MSIE [2-6]" \
                                nokeepalive ssl-unclean-shutdown \
                                downgrade-1.0 force-response-1.0
                BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

        </VirtualHost>
</IfModule>
```

Enable it, and you're done

```
$ sudo a2ensite example.com.ssl.conf
```

If you don't want a self-signed certificate, use StartSSL which gives free certificates for non commercial use: https://www.startssl.com Installation instructions here