How to enable HTTPS protocol with Apache 2 on Ubuntu 20.04

Published on: 18 June 2021 Apache Security SSL Ubuntu

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details >

Accept all

Customise >

Reject all

Powered by **Cookiebot by Usercentrics**

Contents

Configuring an SSL (Secure Sockets Layer) connection, allows you to add an additional asymmetric encryption protocol to the common HTTP. The SSL protocol can be used to strengthen either the authentication system of a website or the data exchange between an app and the server. In this guide you will see how to configure an SSL connection and enable HTTPS on Apache with Ubuntu 20.04.

First, connect to your server via an SSH connection. If you haven’t done so yet, following our guide is recommended to securely connect with the SSH protocol. In case of a local server, go to the next step and open the terminal of your server.

Getting an SSL Certificate

To establish a secure connection, Apache will need an SSL certificate that can be obtained from a Certification Authority (CA). For convenience, in this example we will use a self-signed or self-signed certificate, used only in test and development environments. To obtain a self-signed certificate, refer to our guide to Create a Self-Signed SSL Certificate.

If you are interested in obtaining a free SSL certificate issued by a Certification Authority, follow our guide on How to secure Apache with Let's Encrypt and Ubuntu 18.04 .

Important note:

During the creation of the certificate, enter your server’s IP address and or domain name when asked for the Common Name:

```
Common Name (e.g. server FQDN or YOUR name) []: domain.c
om
```

After obtaining the certificate, create the /etc/certificate folder:

```
$ sudo mkdir /etc/certificate
```

Then save both the certificate and the private key in it.

Configuring the Apache SSL parameters

Proceed by setting the directives for the secure connection that Apache will create. To do so, create the ssl-params.conf file in the Apache conf-available directory:

```
$ sudo nano /etc/apache2/conf-available/ssl-params.conf
```

Paste the following basic configuration into the newly created file:

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES2
56+EDH

    SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

    SSLHonorCipherOrder On


    Header always set X-Frame-Options DENY

    Header always set X-Content-Type-Options nosniff

    # Requires Apache >= 2.4

    SSLCompression off

    SSLUseStapling on

    SSLStaplingCache "shmcb:logs/stapling-cache(150000)"


    # Requires Apache >= 2.4.11

    SSLSessionTickets Off
```

Then save and close the file.

How to change the Virtual Host

Then, modify the SSL configuration of the Virtual Host of the domain you want to protect with SSL connection. In this tutorial the SSL configuration of the default Apache Virtual Host will be used, as an example.

Open the Virtual Host SSL configuration:

```
$ sudo nano /etc/apache2/sites-available/default-ssl.con
f
```

You'll find a file structured as follows :

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details ›

Accept all

Customise ›

Reject all

Powered by **Cookiebot by Usercentrics**

```
<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>
```

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details ›

Accept all

Customise ›

Reject all

Powered by **Cookiebot by Usercentrics**

Set up the ServerAdmin directive correctly by entering your email and add the ServerName directive followed by your domain or your server's IP address.

Finally, change the path indicated by the SSLCertificateFile and SSLCertificateKeyFile directives, entering respectively the path of your certificate and private key .

You will get a result similar to the following :

```
<IfModule mod_ssl.c>

            <VirtualHost _default_:443>

                    ServerAdmin john@mydomain.com

                    ServerName mydomain.com


                    DocumentRoot /var/www/html


                    ErrorLog ${APACHE_LOG_DIR}/error.log

                    CustomLog ${APACHE_LOG_DIR}/access.log combined


                    SSLEngine on


                    SSLCertificateFile      /etc/certificate/certificate.crt

                    SSLCertificateKeyFile /etc/certificate/private.key


                    <FilesMatch "\.(cgi|shtml|phtml|php)$">

                                    SSLOptions +StdEnvVars

                    </FilesMatch>

                    <Directory /usr/lib/cgi-bin>

                                    SSLOptions +StdEnvVars

                    </Directory>


            </VirtualHost>

    </IfModule>
```

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details ›

Accept all

Customise ›

Reject all

Powered by **Cookiebot by Usercentrics**

Then save and close the file.

How to configure the Firewall

In case of a firewall on your system, set it up to enable HTTP traffic and HTTPS traffic to your machine.

When using the UFW firewall, some pre-installed profiles for Apache are available. So let's see how to enable them.

To check the available profiles installed in the UFW firewall, run this command:

```
$ sudo ufw app list
```

A list similar to the following will be displayed on the screen:

```
Available applications:

      Apache

      Apache Full

      Apache Secure

      OpenSSH
```

To allow HTTP (Port 80) and HTTPS (Port 443) traffic, use the "Apache Full" profile.

Check the profile information as follows:

```
$ sudo ufw app info "Apache Full"
```

The screen profile description will be displayed :

```
Profile: Apache Full

    Title: Web Server (HTTP,HTTPS)
    Description: Apache v2 is the next generation of the
omnipresent Apache web
    server.

    Ports:
    80,443/tcp
```

After verifying the profile, enable it:

```
$ sudo ufw allow in "Apache Full"
```

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details ›

Accept all

Customise ›

Reject all

Powered by **Cookiebot by Usercentrics**

How to configure Apache

At this point changes to the Apache configuration can be made.

Enable the mod_ssl and mod_headers modules:

```
$ sudo a2enmod ssl

    $ sudo a2enmod headers
```

Enable reading of the SSL configuration created earlier:

```
$ sudo a2enconf ssl-params
```

Enable the default SSL Virtual Host:

```
$ sudo a2ensite default-ssl
```

Check that you have not made syntax errors in the Apache configuration files:

```
$ sudo apache2ctl configtest
```

If the message "Syntax OK" appears on the screen, proceed by restarting Apache:

```
$ sudo systemctl restart apache2
```

How to check the secure connection

Open your browser by connecting to the domain or IP address of the Virtual Host you configured, making sure to use the https protocol

```
https://mydomain.com
```

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details >

Accept all

Customise >

Reject all

Powered by **Cookiebot by Usercentrics**

A complete Cloud environment for developing your projects

Cloud PRO
Learn more >

Cloud VPS
Learn more >

Virtual Private Cloud
Learn more >

Cloud Backup
Learn more >

Database as a Service
Learn more >

Cloud Object Storage
Learn more >

Domain Center

Learn more ›

Jelastic Cloud

Learn more ›

This website uses cookies

To offer you an ever better browsing experience, this website uses its own cookies and those of selected third-party partners. Third-party cookies may also be profiling cookies. Please read our **information on the use of cookies** to find out more or go to "Customise" to manage your settings. By clicking "Accept" you consent to the storage of cookies on your device. By clicking "Reject", you accept the storage of only necessary cookies.

Show details ›

Accept all

Customise ›

Reject all

Powered by **Cookiebot by Usercentrics**