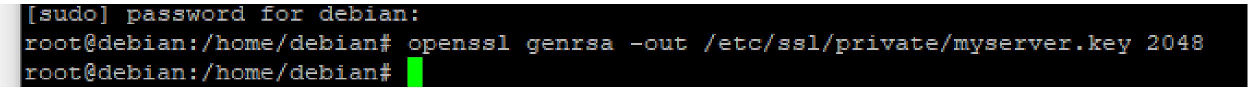
2026-02-02-set-up-an-SSL-in-openSSL-with-a-secure-server

Open a terminal and run the following command to generate a 2048-bit RSA private key:

```
openssl genrsa -out /etc/ssl/private/myserver.key 2048
```



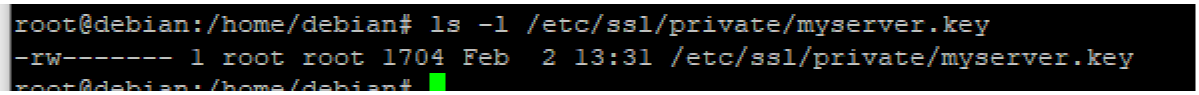
```
[sudo] password for debian:
root@debian:/home/debian# openssl genrsa -out /etc/ssl/private/myserver.key 2048
root@debian:/home/debian#
```

Verify that the file was created:

```
ls -l /etc/ssl/private/myserver.key
```

Expected result:

```
-rw------- 1 root root 1675 Jun  4 18:30 /etc/ssl/private/myserver.key
```



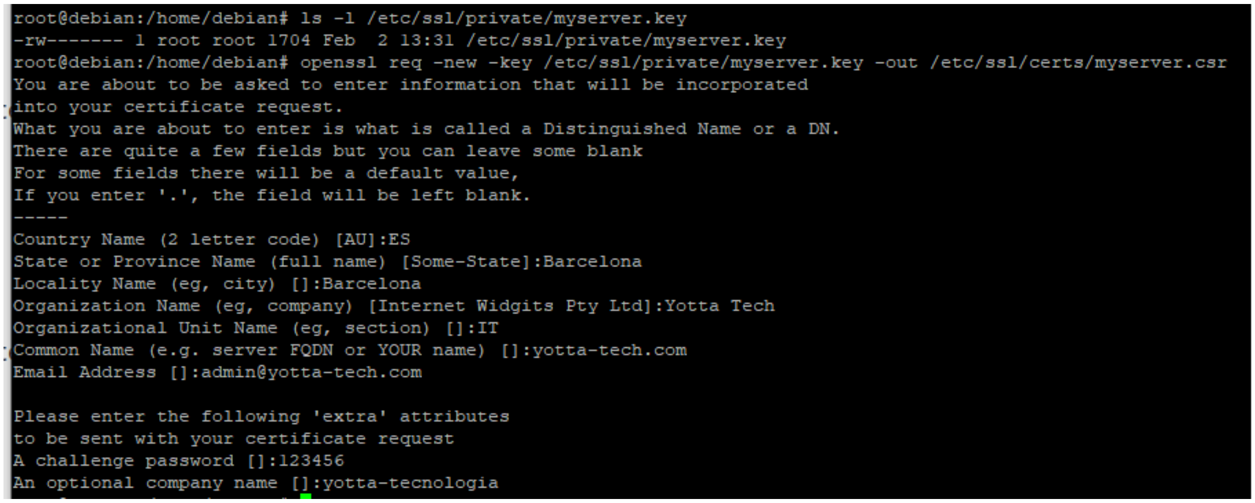
```
root@debian:/home/debian# ls -l /etc/ssl/private/myserver.key
-rw------- 1 root root 1704 Feb  2 13:31 /etc/ssl/private/myserver.key
root@debian:/home/debian#
```

Use the following command to generate a CSR that will contain the public information to be included in the certificate:

```
openssl req -new -key /etc/ssl/private/myserver.key -out /etc/ssl/certs/myserver.csr
```

During the process, you will be prompted to enter information about your organization. (Here is an example of how you can complete it):

- Country Name (2 letter code): ES
- State or Province Name (full name): Barcelona
- Locality Name (eg, city): Barcelona
- Organization Name (eg, company): Yotta-tech
- Organizational Unit Name (eg, section): IT
- Common Name (eg, fully qualified host name): yotta-tech.com
- Email Address: admin@yotta-tech.com

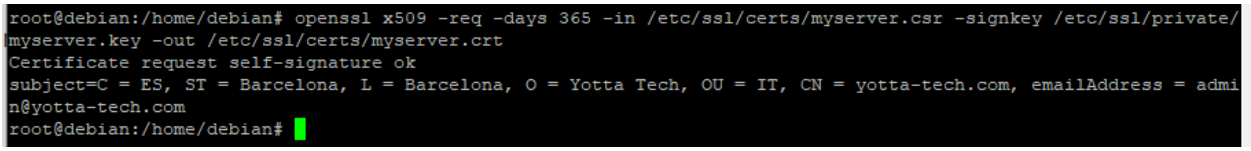


```
root@debian:/home/debian# ls -l /etc/ssl/private/myserver.key
-rw------- 1 root root 1704 Feb  2 13:31 /etc/ssl/private/myserver.key
root@debian:/home/debian# openssl req -new -key /etc/ssl/private/myserver.key -out /etc/ssl/certs/myserver.csr
You are about to be asked to enter information that will 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]:ES
State or Province Name (full name) [Some-State]:Barcelona
Locality Name (eg, city) []:Barcelona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Yotta Tech
Organizational Unit Name (eg, section) []:IT
Common Name (e.g. server FQDN or YOUR name) []:yotta-tech.com
Email Address []:admin@yotta-tech.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:123456
An optional company name []:yotta-tecnologia
```

For the purposes of this practice, we will sign the CSR with our own private key to obtain a self-signed certificate, use the following command (This will generate a self-signed certificate valid for 365 days):

```
openssl x509 -req -days 365 -in /etc/ssl/certs/myserver.csr -signkey /etc/ssl/private/myserver.key -out /etc/ssl/certs/myserver.crt
```

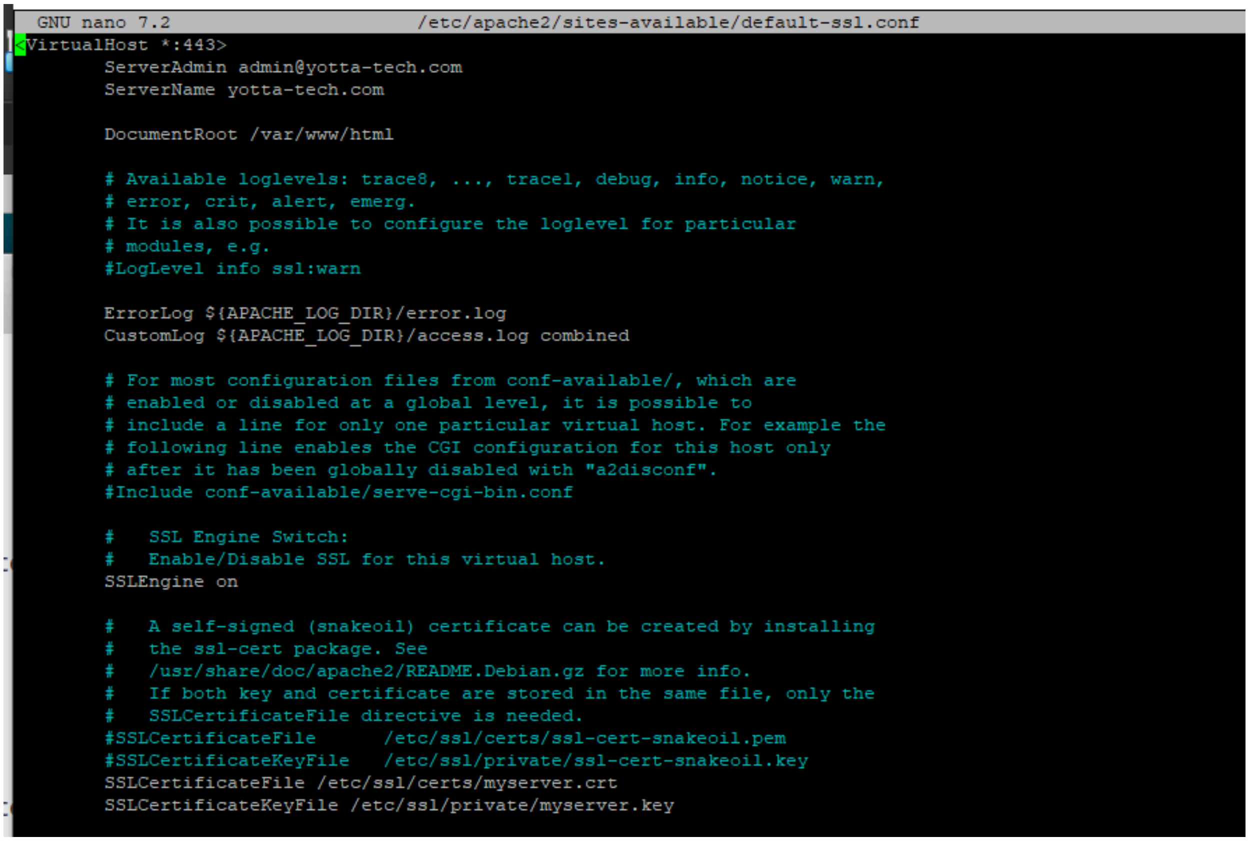
- `x509`: Generates a certificate in standard X.509 format.
- `-req`: Indicates that the input file is a CSR request.
- `-days 365`: The certificate will be valid for 365 days.
- `-in /etc/ssl/certs/myserver.csr`: CSR file that we will sign.
- `-signkey /etc/ssl/private/myserver.key`: Private key used to sign the CSR.
- `-out /etc/ssl/certs/myserver.crt`: Name of the resulting file (the final certificate).



```
root@debian:/home/debian# openssl x509 -req -days 365 -in /etc/ssl/certs/myserver.csr -signkey /etc/ssl/private/
myserver.key -out /etc/ssl/certs/myserver.crt
Certificate request self-signature ok
subject=C = ES, ST = Barcelona, L = Barcelona, O = Yotta Tech, OU = IT, CN = yotta-tech.com, emailAddress = admi
n@yotta-tech.com
root@debian:/home/debian#
```

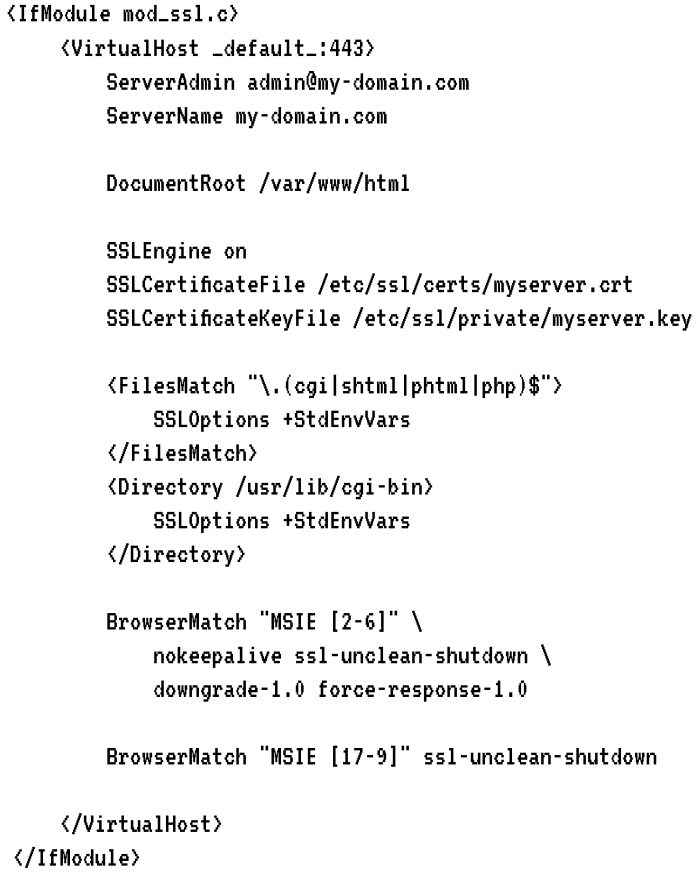
Check the SSL configuration file for Apache:

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



```
  GNU nano 7.2                      /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:443>
        ServerAdmin admin@yotta-tech.com
        ServerName yotta-tech.com

        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

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

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf

        #   SSL Engine Switch:
        #   Enable/Disable SSL for this virtual host.
        SSLEngine on

        #   A self-signed (snakeoil) certificate can be created by installing
        #   the ssl-cert package. See
        #   /usr/share/doc/apache2/README.Debian.gz for more info.
        #   If both key and certificate are stored in the same file, only the
        #   SSLCertificateFile directive is needed.
        #SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
        #SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key
        SSLCertificateFile /etc/ssl/certs/myserver.crt
        SSLCertificateKeyFile /etc/ssl/private/myserver.key
```

Make sure the file contains the following:



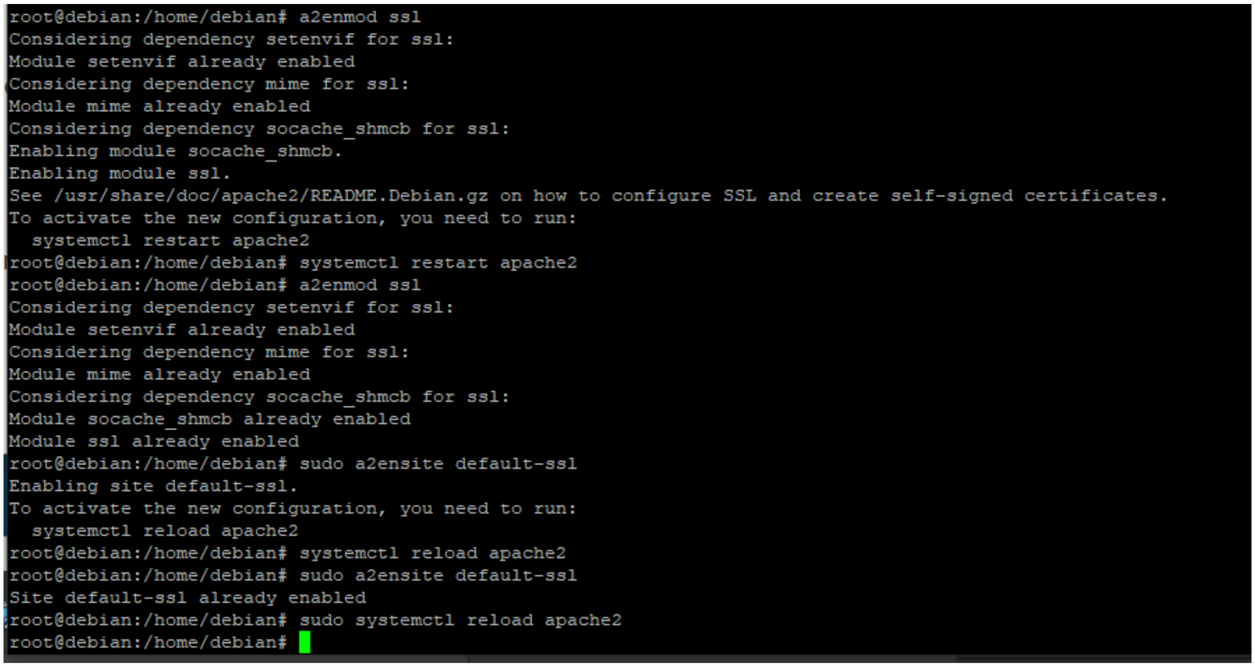
```
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin admin@my-domain.com
        ServerName my-domain.com

        DocumentRoot /var/www/html

        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/myserver.crt
        SSLCertificateKeyFile /etc/ssl/private/myserver.key

        <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>
```

Ensure that:

- The ServerName matches the Common Name (CN) of the self-signed certificate.
- SSLEngine on: The SSL engine for this site is enabled to on.
- SSLCertificateFile: Path to the .crt file (the public certificate)
- SSLCertificateKeyFile: Path to the .key file (the private key associated with the certificate).

Use the following commands to enable the SSL module and load the HTTPS site configuration:

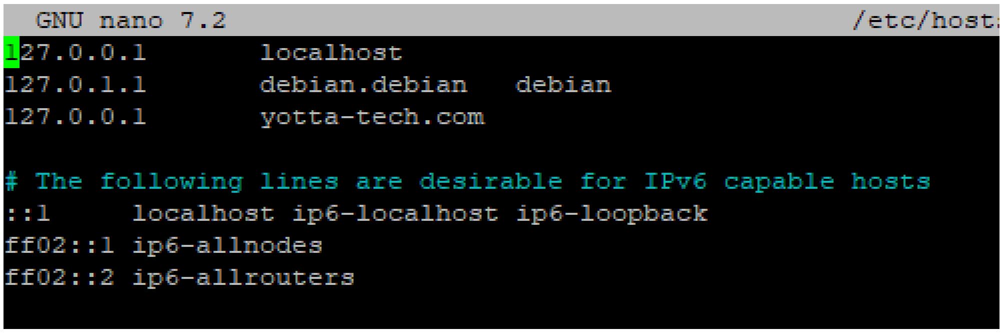
```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo systemctl reload apache2
```



```
root@debian:/home/debian# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@debian:/home/debian# systemctl restart apache2
root@debian:/home/debian# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
root@debian:/home/debian# sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@debian:/home/debian# systemctl reload apache2
root@debian:/home/debian# sudo a2ensite default-ssl
Site default-ssl already enabled
root@debian:/home/debian# sudo systemctl reload apache2
root@debian:/home/debian# 
```

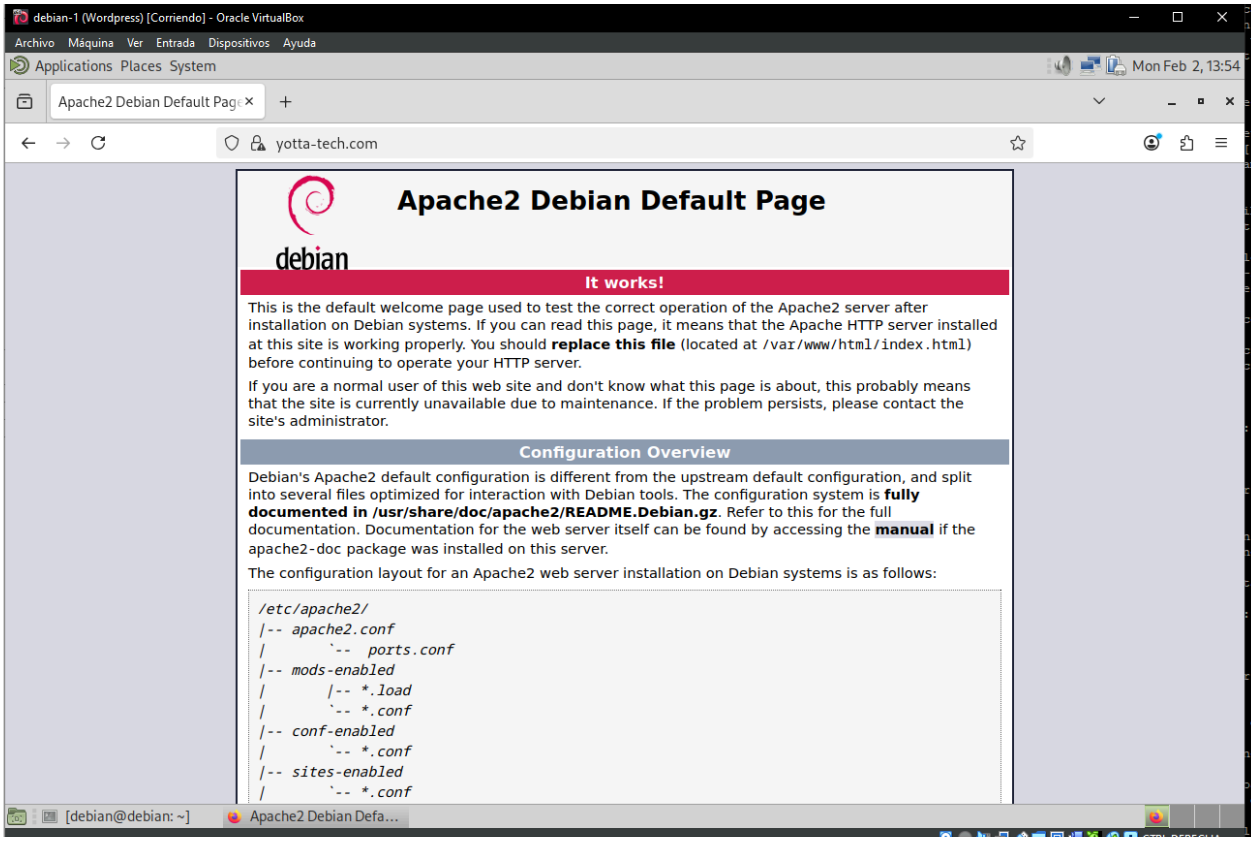
Check the /etc/hosts file on your local machine (from where you access the virtual machine) to ensure that my-domain.com resolves to 127.0.0.1

```
sudo nano /etc/hosts
```



```
  GNU nano 7.2                                                  /etc/host
127.0.0.1       localhost
127.0.1.1       debian.debian   debian
127.0.0.1       yotta-tech.com

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

Open a web browser and enter the URL https://my-domain.com. You should see a security warning due to the self-signed certificate. Accept the risk and continue to see the default Apache page served over HTTPS.


debian-1 (Wordpress) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Applications Places System
Mon Feb 2, 13:54
Apache2 Debian Default Pag
yotta-tech.com
debian
[debian@debian: ~]
Apache2 Debian Defa...

Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

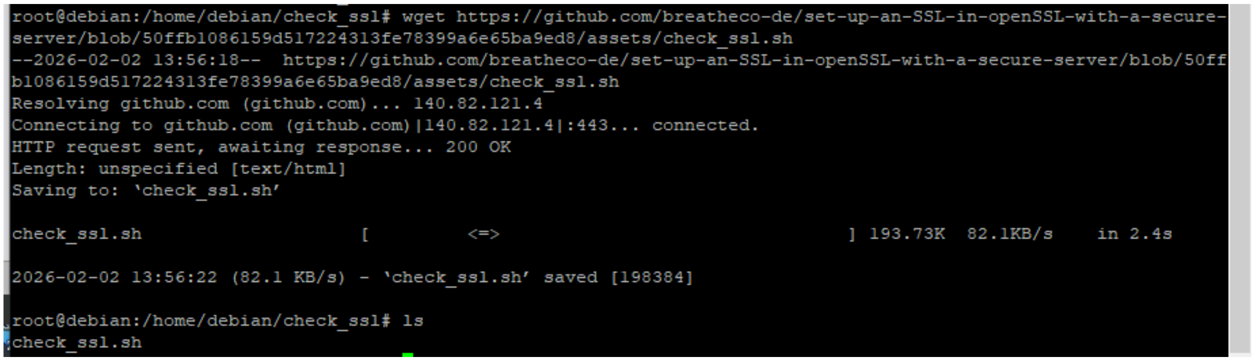
Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

In the ./assets folder, you will find the script check_ssl.sh that you should copy and paste onto the desktop of your Debian virtual machine.



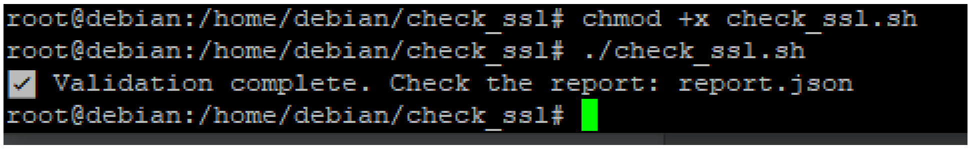
```
root@debian:/home/debian/check_ssl# wget https://github.com/breatheco-de/set-up-an-SSL-in-openSSL-with-a-secure-
server/blob/50ffb1086159d517224313fe78399a6e65ba9ed8/assets/check_ssl.sh
--2026-02-02 13:56:18--  https://github.com/breatheco-de/set-up-an-SSL-in-openSSL-with-a-secure-server/blob/50ff
b1086159d517224313fe78399a6e65ba9ed8/assets/check_ssl.sh
Resolving github.com (github.com)... 140.82.121.4
Connecting to github.com (github.com)|140.82.121.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: ‘check_ssl.sh’

check_ssl.sh                [         <=>                                ] 193.73K  82.1KB/s    in 2.4s

2026-02-02 13:56:22 (82.1 KB/s) - ‘check_ssl.sh’ saved [198384]

root@debian:/home/debian/check_ssl# ls
check_ssl.sh
```

Once you have pasted the script check_ssl.sh on your Debian machine, open the terminal and navigate to the directory where the script is located, in our case ./Desktop, and make the script executable (if it is not already). This can be done using the chmod command:

```
chmod +x check_ssl.sh
```

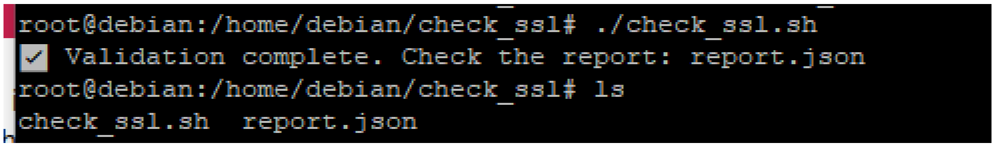
Run the script specifying its name. You may also provide any necessary arguments. Assuming no additional arguments are needed for this example, you should run:

```
./check_ssl.sh
```



```
root@debian:/home/debian/check_ssl# chmod +x check_ssl.sh
root@debian:/home/debian/check_ssl# ./check_ssl.sh
✅ Validation complete. Check the report: report.json
root@debian:/home/debian/check_ssl# 
```

Upload your results. Running the script will create a report.json file that you should copy and paste into the root of this project.



```
root@debian:/home/debian/check_ssl# ./check_ssl.sh
✅ Validation complete. Check the report: report.json
root@debian:/home/debian/check_ssl# ls
check_ssl.sh  report.json
```