How to install ssl certs in jump server [secure connection] ?

207 Manu Chacko June 23, 2026 Tweaks & Configuration 20020



How to Create a SSL Certificate on jump server Apache for Ubuntu 20/ Ubuntu 22/Ubuntu 24/ Debian12/ Debian11 ?

Step 1. Install Apache2

```
root@gateway:~# apt-get install apache2
```

Step 2. Create a New Directory

We need to create a new directory where we will store the server key and certificate

```
root@gateway:~# mkdir /etc/certs/ssl
```

Step 3. Create a Self Signed Certificate

When we request a new certificate, we can specify how long the certificate should remain valid by changing the 365 to the number of days we prefer. As it stands this certificate will expire after one year.

```
root@gateway:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

With this command, we will be both creating the self-signed SSL certificate and the server key that protects it, and placing both of them into the new directory.
This command will prompt terminal to display a lists of fields that need to be filled in.

```
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]:US

State or Province Name (full name) [Some-State]  :New  York

Locality Name (eg, city) []  :NYC

Organization Name (eg, company) [Internet Widgits Pty Ltd]  :Awesome  Inc

Organizational Unit Name (eg, section) []  :Dept  of Merriment

Common Name (e.g. server FQDN or YOUR name) []:example.com

Email Address []:webmaster@awesomeinc.com

-----
```

Step 4. Modify the Default Apache SSL Virtual Host File.
Let's modify **/etc/apache2/sites-available/default-ssl.conf** default Apache SSL Virtual Host file.
Before we go any further, let's back up the original SSL Virtual Host file:

```
root@gateway:~# cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/default-ssl.conf.bak
```

Now, open the SSL Virtual Host file to make adjustments:

```
root@gateway:~# nano /etc/apache2/sites-available/default-ssl.conf
```

Find the section that begins with <VirtualHost _default_:443> and **Uncomment the DocumentRoot and ServerName line and replace example.com with your domain name or server IP address.** Also **uncomment SSLCertificateFile, SSLCertificateKeyFile, SSLEngine on & add the correct path of cert file & key file.**

```
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin your_email@example.com
                ServerName server_domain_or_IP
                DocumentRoot /var/www/html


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


                SSLEngine on
                SSLCertificateFile      /etc/ssl/certs/apache-selfsigned.crt
                SSLCertificateKeyFile   /etc/ssl/private/apache-selfsigned.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

    </VirtualHost>
</IfModule>
```

Save & Exit the file.

Step 5. Enable the Changes in Apache

```
root@gateway:~# a2enmod ssl
root@gateway:~# a2ensite default-ssl
```

Also **enforce ssl** in ezeelogin gui.

Restart Apache

```
root@gateway:~# systemctl restart apache2
```

How to Create a SSL Certificate on ezeelogin jump server Apache for Almalinux 9/ Almalinux 8/ Rocky Linux 9/ Rocky Linux 8?

Overview: This article covers creating SSL certificates on Ezeelogin jump servers by installing necessary packages, creating directories for storing keys and certificates, generating self-signed certificates, and configuring Apache virtual hosts. This ensures secure HTTPS connections on CentOS and Ubuntu/Debian systems.

Step 1. Install Mod SSL

```
root@gateway:~# yum install mod_ssl openssl
```

Step 2. Create a New Directory

We need to create a new directory where we will store the server key and certificate

```
root@gateway:~# mkdir /etc/httpd/ssl
```

Step 3. Create a Self Signed Certificate

When we request a new certificate, we can specify how long the certificate should remain valid by changing the 365 to the number of days we prefer. As it stands this certificate will expire after one year.

```
root@gateway:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/apache.key -out /etc/httpd/ssl/apache.crt
```

With this command, we will be both creating the self-signed SSL certificate and the server key that protects it, and placing both of them into the new directory.
This command will prompt terminal to display a lists of fields that need to be filled in.

```
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]:US

State or Province Name (full name) [Some-State] :New York

Locality Name (eg, city) [] :NYC

Organization Name (eg, company) [Internet Widgits Pty Ltd] :Awesome Inc

Organizational Unit Name (eg, section) [] :Dept of Merriment

Common Name (e.g. server FQDN or YOUR name) []:example.com

Email Address []:webmaster@awesomeinc.com

-----
```

Step 4. Set up the virtual hosts to display the new certificate.

Open up the SSL config file:

```
root@gateway:~# vi /etc/httpd/conf.d/ssl.conf
```

Find the **section that begins with <VirtualHost _default_:443>** and **Uncomment the DocumentRoot and ServerName line and replace example.com with your domain name or server IP address**

```
#uncomment & Replace the example.com with your domain
  ServerName example.com:443

  DocumentRoot "/var/www/html"

  ServerName www.example.com:443
```

Find the following three lines, and make sure that they match the extensions below:

```
SSLEngine on

SSLCertificateFile /etc/httpd/ssl/apache.crt

SSLCertificateKeyFile /etc/httpd/ssl/apache.key
```

Your virtual host is now all set up! Save and Exit.

Restart Apache

```
root@gateway:~# systemctl restart httpd
```

Note:

Ezeelogin uses two distinct sets of SSL certificate and key files — one for the web interface and another for its backend services. Correctly identifying these file paths before making any changes is essential to avoid service disruptions.

This applies whenever you need to:
Renew an expiring SSL certificate
Replace an existing SSL certificate with a new one
Troubleshoot SSL-related errors across Ezeelogin components

SSL Files for the Web Interface
The Ezeelogin web interface uses the following files to serve secure HTTPS access:
/etc/ssl/certs/localhost.crt
/etc/pki/tls/private/localhost.key

SSL Files for Backend Services
Ezeelogin's internal services — including the Web Portal, WebSSH, and RDP — rely on a separate set of certificate and key files:
/usr/local/etc/ezlogin/tls_cert.pem
/usr/local/etc/ezlogin/tls_key.pem

Always back up all existing certificate and key files before applying updates. This ensures you can quickly roll back to the previous configuration if anything goes wrong. Once the backup is in place, replace the files with the newly generated certificate and key to restore secure communication across all Ezeelogin components.

Related Articles:

Check the versions of SSL/TLS , HTTPS and SSH used in Ezeelogin Server

How To Create a Self-Signed SSL Certificate for Nginx on debian

SSL Certificate failed with MySQL SSL

Install Master/Slave Ezeelogin with MySQL SSL

Online URL: https://www.ezeelogin.com/kb/article/how-to-install-ssl-certs-in-jump-server-secure-connection-207.html