



Creación de certificados SSL Let's Encrypt e instalación de Coturn en CentOS 8 para OpenMeetings 6.1.0

El presente tutorial está hecho básicamente para aquellos que tienen instalado OpenMeetings 6.1.0 en su computadora tras un router NAT. Está testeado y funciona correctamente.

Si no lo tuviera instalado puede descargar directamente la guía de instalación desde aquí:

[Descargar Instalación OpenMeetings 6.1.0 en CentOS 8](#)

Doy las gracias a Maxim Solodovnik y a Carlos Heras, sin cuya colaboración en las pruebas prácticas no habría podido confirmar el correcto funcionamiento y así poder publicar el presente tutorial.

Igualmente doy las gracias a todos aquellos que han contribuido, tales como Marcus Schulz y Daniel Baker. Gracias a todos ellos.

Comenzamos...

1)

----- Creación del certificads Let's Encrypt SSL -----

Instalamos certbot, necesario para hacer los certificados:

```
sudo yum install certbot
```

Es importante que su servidor-pc no tenga en uso el puerto 80 con algún servidor web o algún otro. Si fuera así deténgalo y continúe con este paso. Cuando concluya los certificados podrá lanzarlo nuevamente. Este puerto ha de estar abierto en el router y en el firewall.

Let's Encrypt valida "SSL Certificate Authority (CA)" el o los dominios de tu servidor.

Lo ejecutaremos con el parámetro `--standalone`, para que usted pueda añadir al final, cada dominio que requiera un certificado, por ejemplo: `-d nuevoejemplo.com`
Cambie "`ejemplo.com`" por el verdadero dominio de su servidor:

```
sudo certbot certonly --standalone -d ejemplo.com -d www.ejemplo.com
```

Preguntará por una dirección de correo de administración. Ponga uno verdadero para que le mantenga informado acerca de los certificados:

Installation succeeded.

Saving debug log to /var/log/letsencrypt/letsencrypt.log

Plugins selected: Authenticator standalone, Installer None

Enter email address (used for urgent renewal and security notices) (Enter 'c' to cancel): **...aquí ponga su dirección de correo y pulse Enter**

Preguntará si está de acuerdo;

Please read the Terms of Service at

<https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf>. You must agree in order to register with the ACME server at <https://acme-v02.api.letsencrypt.org/directory>

(A)gree/(C)ancel: **...escriba... a ...y pulse Enter**

Preguntará si quiere compartir su dirección de correo:

Would you be willing to share your email address with the Electronic Frontier Foundation, a founding partner of the Let's Encrypt project and the non-profit organization that develops Certbot? We'd like to send you email about our work encrypting the web, EFF news, campaigns, and ways to support digital freedom.

(Y)es/(N)o: **...escriba... n ...y pulse Enter**

...cuando finalice de hacer los certificados con éxito, mostrará lo siguiente:

IMPORTANT NOTES:

- **Congratulations!** Your certificate and chain have been saved at:
/etc/letsencrypt/live/**tu_dominio**/fullchain.pem
Your key file has been saved at:
/etc/letsencrypt/live/**tu_dominio**/privkey.pem
Your cert will expire on 2020-06-24. To obtain a new or tweaked version of this certificate in the future, simply run letsencrypt-auto again. To non-interactively renew **all** of your certificates, run "letsencrypt-auto renew"
- If you like Certbot, please consider supporting our work by:

Donating to ISRG / Let's Encrypt: <https://letsencrypt.org/donate>
Donating to EFF: <https://eff.org/donate-le>

2)

----- **Chequear certificado de dominio** -----

Vamos a ver donde están almacenados los certificados que acabamos de crear, que en nuestro caso es /etc/letsencrypt/live:

```
sudo ls /etc/letsencrypt/live
```

...mostrará el nombre de su dominio: **tu_dominio**

Todos los dominios que usted haya especificado en el paso anterior, se hallarán en el mismo certificado.

3)

----- **Renovación del certificado SSL** -----

El certificado Let's Encrypt tiene un inconveniente, y es que su validez es solo de 90 días, por lo que habremos de renovarlo.

Podemos hacer esto manualmente (siempre conectados a Internet):

```
sudo certbot renew
```

4)

----- **Configuración de Tomcat-OpenMeetings con los certificados SSL** -----

Esta configuración que haremos ahora es solo para la serie **5 ó 6.xx** (no 4) de OpenMeetings. **Este paso número 4 hay que repetirlo cada 80 días, tras actualizar los certificados, pues son 90 los días válidos de Letsencrypt.**

He seguido la ruta de instalación de OM que muestran los tutoriales de OpenMeetings que se encuentran en su sitio wiki oficial. Es decir `/opt/open610`.

Si usted hubiera hecho la instalación en una ruta distinta, modifique lo que indico a continuación. Ya hicimos los certificados letsencrypt de nuestro dominio en el paso 1.

Ahora vamos a crear un PKCS12 que contenga la full chain y la privada. Es necesario tener instalado openssl. Lo instalamos si no:

```
sudo yum install openssl
```

Ahora lanzamos el siguiente comando:

(En una sola línea con espacio entre cada una de ellas)

```
sudo openssl pkcs12 -export -out /tmp/example.com_fullchain_and_key.p12  
-in /etc/letsencrypt/live/example.com/fullchain.pem  
-inkey /etc/letsencrypt/live/example.com/privkey.pem -name tomcat
```

...sustituir `example.com` por tu verdadero dominio (el mismo de cuando hemos hecho los certificados letsencrypt) Pedirá una contraseña, elija a su gusto y guardela en un archivo de texto,

...y ahora convertimos esa PKCS12 en un JKS empleando java keytool:

(En una sola línea con espacio entre cada una de ellas)

```
sudo keytool -importkeystore -deststorepass samplePassword -destkeypass samplePassword  
-destkeystore /tmp/example.com.jks -srckeystore /tmp/example.com_fullchain_and_key.p12  
-srestoretype PKCS12 -srcstorepass samplePassword -alias tomcat
```

...sustituya `example.com` por tu verdadero dominio (dos veces), y `samplePassword` (tres veces) por la contraseña que haya elegido anteriormente y que había guardado en un archivo de texto.

Copiamos el archivo generado `example.com.jks` al directorio de instalación de Tomcat-OpenMeetings:

```
sudo cp /tmp/example.com.jks /opt/open610/conf
```

...sustituya `example.com` por tu verdadero dominio.

Pasamos a configurar Tomcat con la Java Keystore que hemos generado.

Para ello editamos el archivo `server.xml`:

```
sudo nano /opt/open610/conf/server.xml
```

...vamos a la sección:

```
<Connector port="5443" protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="150" SSLEnabled="true">
  <SSLHostConfig>
    <Certificate certificateKeystoreFile="conf/localhost.jks"
      certificateKeystorePassword="openmeetings"
      certificateKeystoreType="JKS"
      certificateVerification="false"
      sslProtocol="TLS"
      type="RSA" />
  </SSLHostConfig>
```

...y lo modificamos dejándolo así:

```
<Connector port="5443" protocol="org.apache.coyote.http11.Http11NioProtocol"
  maxThreads="150" SSLEnabled="true">
  <SSLHostConfig>
    <Certificate certificateKeystoreFile="conf/example.com.jks"
      certificateKeystorePassword="samplePassword"
      certificateKeystoreType="JKS"
      certificateVerification="false"
      sslProtocol="TLS"
      type="RSA" />
  </SSLHostConfig>
```

...sustituya **example.com** por su verdadero dominio, y **samplePassword** por la contraseña que recién haya escogido (la que acaba de guardar en un archivo de texto)

...salimos del editor nano pulsando las teclas **Ctrl+x**, preguntará si guarda y pulsamos **S** y después **Enter** para salir.

5)

----- Instalación de Coturn y configuración del servidor Turn -----

Instalaremos Coturn.

```
sudo yum install coturn
```

```
# Configuramos turn server.
```

Comenzamos creando una contraseña que necesitaremos para ponerla en el archivo de configuración y más tarde en un archivo de OpnMeetings. La creamos:

```
sudo openssl rand -hex 32
```

...generará algo similar a esto:

```
751c45cae60a2839711a94c8d6bf0089e78b2149ca602fdXXXXXXXXXXXXXXXXXX
```

...copie la larga contraseña y péguela en un archivo de texto guardándolo.

Ahora editamos el archivo de configuración de turn:

```
sudo nano /etc/coturn/turnserver.conf
```

...en este archivo habremos de descomentar (borrar #) solo las siguientes líneas:

```
verbose
```

```
use-auth-secret
```

```
static-auth-secret=751c45cae60a2839711a94c8d6bf0089e78b2149ca602fdXXXXXXXXXXXXXXXXXX
```

(en la línea de arriba pongan la larga contraseña que acabamos de guardar en un archivo de texto)

```
realm=su_verdadero_dominio    ...cambiar company.org por su verdadero dominio
```

```
log-file=/var/log/coturn/turnserver.log
```

```
simple-log
```

...salimos del editor nano pulsando las teclas **Ctrl+x**, preguntará si guarda y pulsamos **S** y después **Enter** para salir.

6)

----- Configuración de OpenMeetings 6.1.0 con Kurento media server -----

Editamos el archivo openmeetings.properties de OpenMeetings:

`sudo nano /opt/open610/webapps/openmeetings/WEB-INF/classes/openmeetings.properties`

...y en la sección `#### Kurento ####` modificamos solo las siguientes líneas::

```
#### Kurento ####
```

```
kurento.turn.url=  
kurento.turn.user=  
kurento.turn.secret=
```

...dejandolas así:

```
kurento.turn.url=IP publica de tu servidor:3478  
kurento.turn.user=  
kurento.turn.secret=751c45cae60a2839711a94c8d6bf0089e78b2149ca602fdXXXXXXXXXXXXXXXX
```

...arriba, en:

```
kurento.turn.secret=751c45cae60a2839711a94c8d6bf0089e78b2149ca602fdXXXXXXXXXXXXXXXX
```

...sustituya la línea:

```
751c45cae60a2839711a94c8d6bf0089e78b2149ca602fdXXXXXXXXXXXXXXXX
```

...por la larga contraseña que generamos en el paso 5 y que guardamos en un archivo de texto,

Salimos del editor nano pulsando las teclas **Ctrl+x**, preguntará si guarda y pulsamos **S** y después **Enter** para salir.

ES IMPORTANTE...reiniciamos la máquina y después vamos al siguiente paso 7.

Mas antes detendremos los servidores, si estuvieran corriendo:

```
sudo systemctl stop mariadb.service
```

```
sudo podman stop kms
```

```
sudo systemctl stop coturn.service
```

```
sudo /etc/init.d/tomcat34 stop
```

...y ahora reiniciamos:

```
sudo reboot
```

7)

----- Iniciar los servidores tras haber reiniciado la máquina -----

Lanzamos todos los servidores relacionados con OpenMeetings:

Iniciamos MariaDB: `sudo systemctl start mariadb.service`

Iniciamos Kurento: `sudo podman start kms`

Iniciamos Coturn: `sudo systemctl start coturn.service`

Tomcat-OpenMeetings: `sudo /etc/init.d/tomcat34 start`

8)

----- Abrir puertos necesarios para los servidores-----

Necesitamos abrir determinados puertos, tanto en el router como en el firewall, para que los servidores puedan ser accesibles.

Estos son:

3478 TCP UDP IN

5443 TCP IN

8888 TCP IN

49152:65535 UDP IN-OUT

Para abrirlos con IPTables estos son los comandos:

```
sudo iptables -A INPUT -p tcp -m tcp --dport 3478 -j ACCEPT
```

```
sudo iptables -A INPUT -p udp -m udp --dport 3478 -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp -m tcp --dport 5443 -j ACCEPT
```

```
sudo iptables -A INPUT -p tcp -m tcp --dport 8888 -j ACCEPT
```

```
sudo iptables -A INPUT -p udp --match multiport --dports 49152:65535 -j ACCEPT
```

```
sudo iptables -A OUT -p udp --match multiport --dports 49152:65535 -j ACCEPT
```


...tras haber lanzado los comandos guardamos los cambios:

```
sudo service iptables save
```

...y reiniciamos IPTables:

```
sudo service iptables restart
```

Y con esto concluimos.

Si tuviera alguna duda o pregunta, por favor planteala en los foros de Apache OpenMeetings:

<https://openmeetings.apache.org/mailling-lists.html>

OpenMeetings



Gracias.

Alvaro Bustos (PMC y Committer en Apache OpenMeetings)