Set up SSL/TLS/HTTPS

1. SSH into the new server
2. Ensure that the server has been fully installed, and an instance has been installed as well
    1. To install run: ts install
3. Solution 1, SSL offload using nginx
    1. Install nginx, run: ts install-proxy
    2. Setup a proxy, run: ts setup-proxy
4. Solution 2, SSL connector in Tomcat
    1. Run: ts install-routing
5. Install certbot, run: ts install-ssl
6. When the install finishes, select Y, or run: ts setup-ssl
7. Follow the prompts

- Old implementation
- Automated renewals
- External Certificate
- Problems with wrappers

Old implementation

The following is the old, manual, way of installing SSL certs.

Tomcat 7 automatic installation

Using the TS commandline tools, you specify the domain and your email

```
tsinstallssl.sh server.acme.com sslresponsible@acme.com
```

After a couple of minutes you will be rquired to enter the domain an email again, and accept the terms of service

Tomcat 7 manual installation

Install and configure letsencrypt

Download an build certbot (letsencrypt client)

```
sudo yum -y install python27-devel git (deprecated)
```

```
sudo yum -y install python36 python36-pip
sudo yum -y install git-all
sudo git clone https://github.com/letsencrypt/letsencrypt /opt/letsencrypt
/opt/letsencrypt/letsencrypt-auto --debug --agree-tos
```

Create a config file

```
sudo touch /etc/letsencrypt/config.ini
sudo chmod 777 /etc/letsencrypt/config.ini
sudo echo "rsa-key-size = 4096" >> /etc/letsencrypt/config.ini
sudo echo "email = kpe@tempusserva.dk" >> /etc/letsencrypt/config.ini
```

Generate PKCS12 certificate

Generate a certificate

```
sudo mkdir /usr/share/tomcat7/webapps/ROOT
/opt/letsencrypt/letsencrypt-auto certonly --debug --webroot -w /usr/share/tomcat7/webapps/ROOT -d
letsencrypt.tempusserva.dk --config /etc/letsencrypt/config.ini --agree-tos
```

Convert to pkcs12 format

```
 sudo -s
```

```
cd /etc/letsencrypt/live/letsencrypt.tempusserva.dk
openssl pkcs12 -export -out bundle.pfx -inkey privkey.pem -in cert.pem -certfile chain.pem -password
pass:TempusServaSecret
chmod 755 bundle.pfx
chmod 755 /etc/letsencrypt/live
```

```
 Press: ctrl + d
```

Install certificate in Tomcat

Edit Tomcat configuration

```
sudo nano  /usr/share/tomcat7/conf/server.xml
```

```
        <Connector
        protocol="org.apache.coyote.http11.Http11NioProtocol"
        port="8443" maxThreads="200"
        scheme="https" secure="true" SSLEnabled="true"
        keystoreFile="/etc/letsencrypt/live/letsencrypt.tempusserva.dk/bundle.pfx"
keystorePass="TempusServaSecret"

ciphers="TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WI
TH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA"
        clientAuth="false" sslProtocol="TLS" keystoreType="PKCS12"/>
```

Reboot the server

```
service tomcat7 restart
```

Automated renewals

Before starting test that the renewal process works

```
/opt/letsencrypt/letsencrypt-auto renew --dry-run
```

Make sure the path is accessible from cron

```
sudo chmod go+x /etc/letsencrypt/archive
sudo chmod go+x /etc/letsencrypt/live
```

Make a script file

```
sudo nano /usr/bin/tsrefreshcerts.sh
```

.... containing the following commands

```
/opt/letsencrypt/letsencrypt-auto renew
cd /etc/letsencrypt/live/letsencrypt.tempusserva.dk
openssl pkcs12 -export -out bundle.pfx -inkey privkey.pem -in cert.pem -certfile chain.pem -password
pass:TempusServaSecret
/usr/bin/tstomcatrestart.sh
```

Now add a job to the crontab

```
sudo crontab -l > tempcron
echo "0 0 1 * * /usr/bin/tsrefreshcerts.sh" >> tempcron
sudo crontab tempcron
rm tempcron
```

Problems with Amazon Linux?

In case the autorenewal process fails try updating the dependencies and pip

```
sudo /opt/eff.org/certbot/venv/bin/pip2 install cryptography zope interface
sudo /opt/eff.org/certbot/venv/bin/pip2 install --upgrade pip
sudo rsync -avz /opt/eff.org/certbot/venv/lib64/python2.7/dist-packages/
```

```
/opt/eff.org/certbot/venv/lib/python2.7/dist-packages/
```

Still got problems with Amazon Linux?

In case certbot cant find the root folder try and run it manually

```
sudo /opt/letsencrypt/letsencrypt-auto certonly
```

Choose the following values when prompted

```
2: Place files in webroot directory (webroot)
<domain>
2: Renew & replace the cert (may be subject to CA rate limits)
/usr/share/tomcat7/webapps/ROOT/
```

Need manual crontab install?

Steps

- sudo crontab -e
- press INSERT
- move to bottom of file
- paste this

```
0 0 1 * * /usr/bin/tsrefreshcerts.sh
```

- press ESC
- press :wq

External Certificate

Acquire certificate

Buy a certificate from a provider. Note that the max lifetime is currently 1 year, så buying a 5 year certificate only help on pricing.

```
https://www.ssls.com/
```

After issuing the files you will have

- A private key - ex: movia.tempusserva.dk.pfx
- A certificate - ex: movia_tempusserva_dk_key.txt

Install certificate

1. Upload the files

2. Convert to a pfx file format

```
openssl pkcs12 -export -out movia.tempusserva.dk.pfx -inkey movia_tempusserva_dk_key.txt -in movia.tempusserva.dk.crt
```

Write the password down

Check alias if needed

```
openssl pkcs12 -nokeys -info -in movia.tempusserva.dk.pfx -passin pass:TempusServaFTW!
```

4. Install in tomcat Add the following code to <tomcat>\conf\server.xml

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true" URIEncoding="UTF-8"
  scheme="https" secure="true" maxHttpHeaderSize="8192"
  maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
  enableLookups="false" acceptCount="100" disableUploadTimeout="true"
  keystoreFile="/mnt/sda/certs/movia.tempusserva.dk.pfx" keystorePass="TempusServaFTW!"
```

```
keystoreType="PKCS12"
  clientAuth="false" sslProtocol="TLS" sslEnabledProtocols="TLSv1.2"
  ciphers="TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,
          TLS_RSA_WITH_AES_128_CBC_SHA,
          TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
          TLS_RSA_WITH_AES_128_CBC_SHA256,
          TLS_RSA_WITH_AES_128_GCM_SHA256,
          TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
          TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
          TLS_RSA_WITH_AES_256_CBC_SHA,
          TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
          TLS_RSA_WITH_AES_256_CBC_SHA256,
          TLS_RSA_WITH_AES_256_GCM_SHA384,
          TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
          TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384"
  compression="on" compressionMinSize="2048" nocompressionUserAgents="gozilla, traviata"
     compressableMimeType="text/html,text/xml,text/plain,application/xml"
/>
```

5. Restart the server

Problems with wrappers

The usage of wrappers can result in SSL warnings.

If your solution is depending on the use of Wrappers, please tjeck the following

- All style, script and image references are made with HTTPS
- No referenced stylesheets depends on images using HTTP

If the wrapper cannot be transformed from HTTP to HTTPS, referenced ressources should be copied to the server

- Stylesheets copied to TS stylesheet
- Images downloaded and copied to the media library

After changes are made remmeber to flush caches: Both Chrome and IE sometimes caches longer than expected.