3.2 – HTTPS and WSS (WebSocket secure connection)

Recommended to enable HTTPS and WSS. There is a self-signed certificate (keystore.jks) in the installation directory.

- Set `ssl=true` in gatway.conf file.
- Set your port to your desired SSL port, like 443: `port=443`
- Import your SSL certificate to a Java keystore, please check with your certificate issue and see how to buy and import certificate for Java application server.
- Set up keyStore and keyStorePassword in gateway.conf:

```
keyStore=D:\\test\\SV\\spark.p12
keyStorePassword=yourPassword
```

- Java 1.8 recommended which supports more and better cipher suites.
- Java 1.8 supports PKCS12 key store, it’s better to use PKCS12 format directly.
- Self-signed certificate may not work in some cases.
- You can have multiple certificates in the Java key store, but Java will always use the first one by default.
- Disable SSLV3, set sslProtocols = SSLv2Hello,TLSv1 in gateway.conf and restart. You can also add TLSv1.1, TLSv1.2 into it for Java 8.
- You can expand the DK key size to 2048 in Java 8 by adding this Java option: `-Djdk.tls.ephemeralDHKeySize=2048`
- You can choose the cipher suites you want to use by setting cipherSuites in gateway.conf. You'll need to install Java Cryptography Extension (JCE) to support all the cipher suites:

  http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html

  http://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html

  Recommended cipher suites for Java 11:
  cipherSuites = TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WIT

H_AES_128_CBC_SHA256,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384,TLS_DHE_RSA_WITH_AES_128_CBC_SHA,TLS_DHE_RSA_WITH_AES_256_CBC_SHA,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

Recommended cipher suites for Java 8:
cipherSuites = TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA,TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA

Set up Let's Encrypt (letsencrypt.org) certificate:

1. Apply for the certificate from letsencrypt.org and you'll get the certificate files: cert.perm, privkey.perm, chain.pem etc in /etc/letsencrypt/live/yourDomain/.
2. `openssl pkcs12 -export -in cert.pem -inkey privkey.pem -out cert_and_key.p12 -name spark -CAfile chain.pem -caname anyFriendlyName`
3. Add following entries in gateway.conf:

```
keyStore=/etc/letsencrypt/live/domain/cert_and_key.p12
keyStorePassword = yourExportPasswordInStep3
ssl = true
port = 443
```

4. Restart the gateway.

Renew and update the certificate automatically:

Create a cron job to update check the certificate every day at 2:30AM (crontab -e):

`30 2 * * * certbot renew --post-hook "sh /etc/letsencrypt/live/startme.biz/update.sh"`

update.sh:

```
#!/bin/bash
cd /etc/letsencrypt/live/domain/
openssl pkcs12 -export -in cert.pem -inkey privkey.pem -out cert_and_key.p12 -name spark -CAfile chain.pem -caname startme -passout pass:mypassword
systemctl stop SparkGateway
systemctl start SparkGateway
```

```
exit 0
```

Revision #3
Created 5 April 2022 08:14:30 by Julian
Updated 6 July 2023 13:50:44 by Julian