1. Goal

Learn how to generate a self-signed certificate and use it to allow Tomcat to work over HTTPS.

2. Lesson Notes

First, note that the entire process of downloading and setting up a Tomcat server on an EC2 instance is covered in Module 9 - Lesson 1 (Setting Up Jenkins and The First Job).

2.1. The Self-Signed Certificate

Let's get started here with the commands to generate the self-signed certificate:

```
/usr/lib/jvm/java-8-oracle/bin/keytool -genkey -alias tomcat -keyalg RSA
```

The result is the keystore file, under the home directory of the current user:

```
~/.keystore
```

We can also use *keytool* to verify the keystore that we just generated:

```
/usr/lib/jvm/java-8-oracle/bin/keytool  -list -keystore ~/.keystore
```

Finally, let's configure Tomcat with a new connector for HTTPS:

```
vim $TOMCAT_INSTALL_DIR/tomcat/conf/server.xml
```

Unable the HTTP connector and add these two lines to the HTTPS connector:

```
keystoreFile="${user.home}/.keystore"
keystorePass="changeit"
```

2.2. A Real Certificate

A self-signed certificate is perfectly fine during development.

But of course, as soon as we start going towards production, we need to obtain a real certificate signed by an actual Certificate Authority.

Now, that is beyond the score of this lesson, so we won't cover that here, but if you're at that point - this is a good place to start.

3. Resources

- SSL/TLS Configuration HOW-TOSSL/TLS Configuration HOW-TO