Acme Madrugá – Https Configuration

# Configuring our project

To configure our project go to the webapp folder and locate the web.xml file.

We are going to create a new security tag, in this tag we will indicate the name of our project (Acme-Madruga), the url pattern (put “/\*” to use https in all our website, not just in the login process), and the transport guarantee constraint “Confidential” to tell the server that the use of SSL is required.

<security-constraint>

<web-resource-collection>

<web-resource-name>Acme-Madruga</web-resource-name>

<url-pattern>/\*</url-pattern>

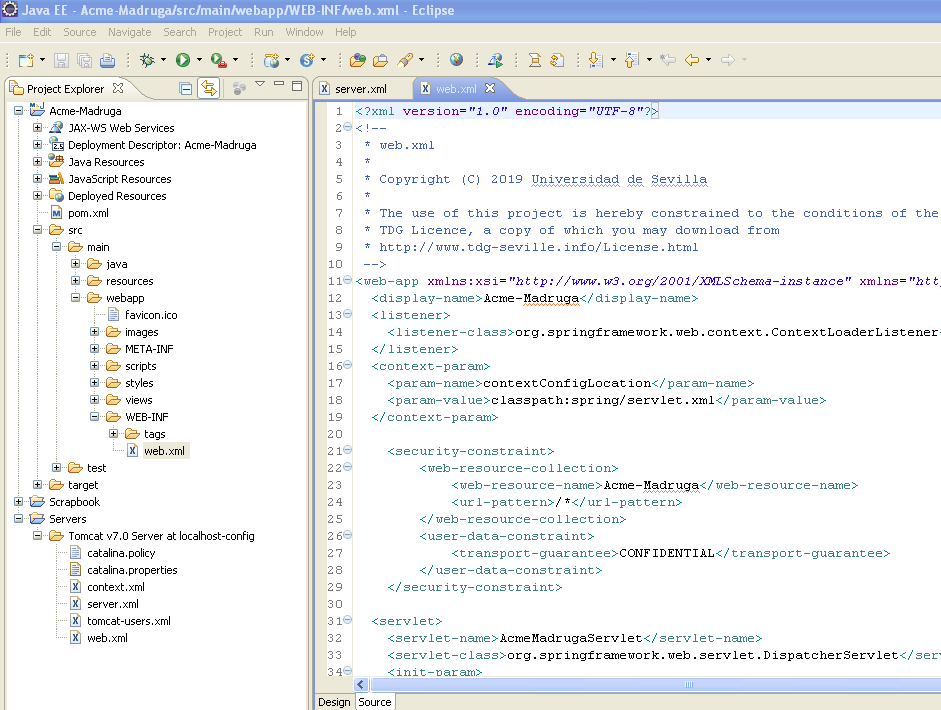
</web-resource-collection>

<user-data-constraint>

<transport-guarantee>CONFIDENTIAL</transport-guarantee>

</user-data-constraint>

</security-constraint>



# Creating the SSL certificate

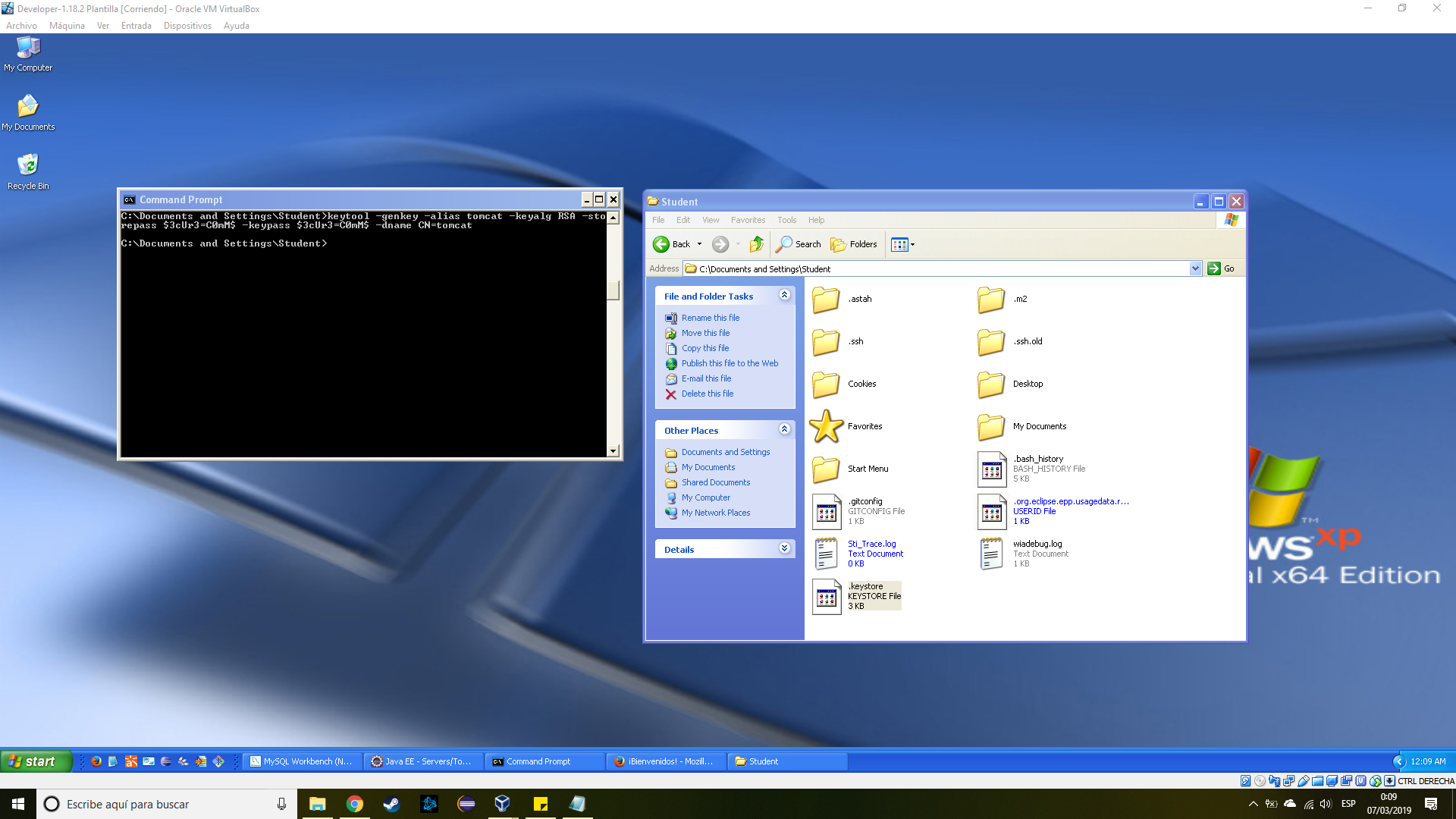
To create our certificate we are going to open a command prompt in the location where we are going to stablish the certificate and execute the next command:

keytool -genkey -alias tomcat -keyalg RSA -storepass $3cUr3=C0mM$ -keypass $3cUr3=C0mM$ -dname

CN=tomcat

Where keypass is the password of our certificate, in our case we decided to use “$3cUr3=C0mM$”.

It will generate the certificate in the user folder.



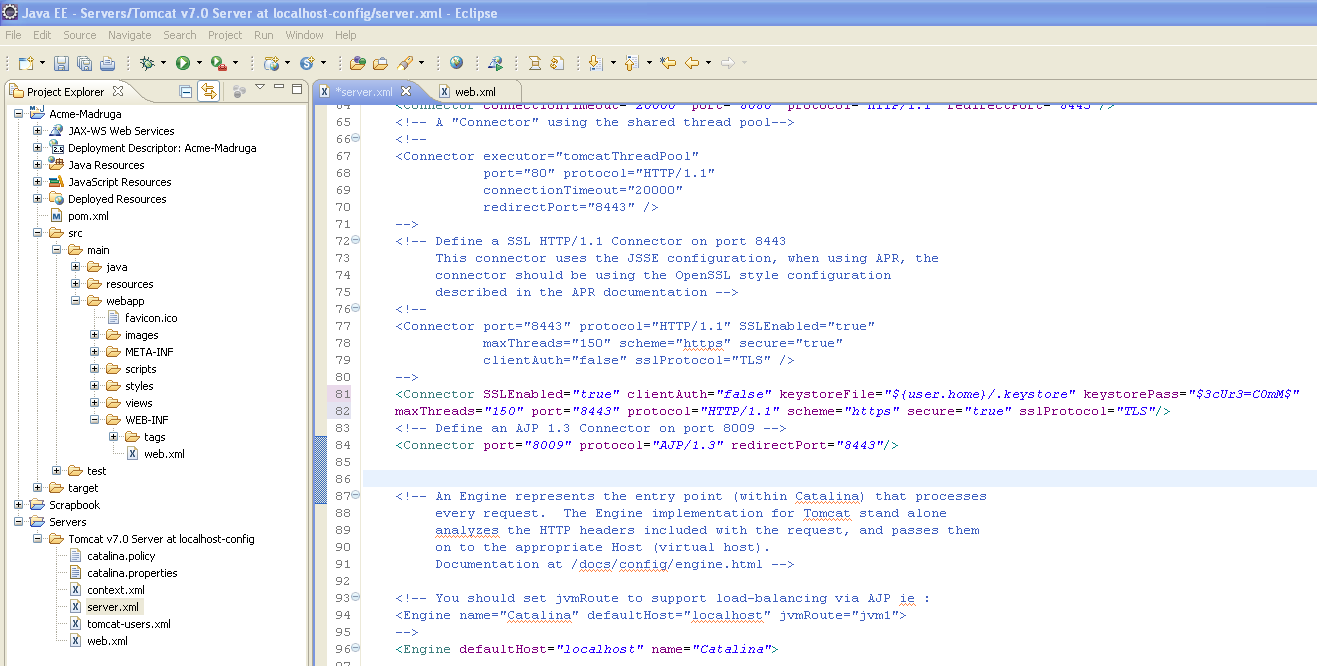
## Configuring Tomcat

Configuring Tomcat is really easy, go to your server folder an open server.xml, we are going to define a new Connector tag.

This tag already exist in the file but is commented, you just need to add two new params to indicate the location and password of the certificate:

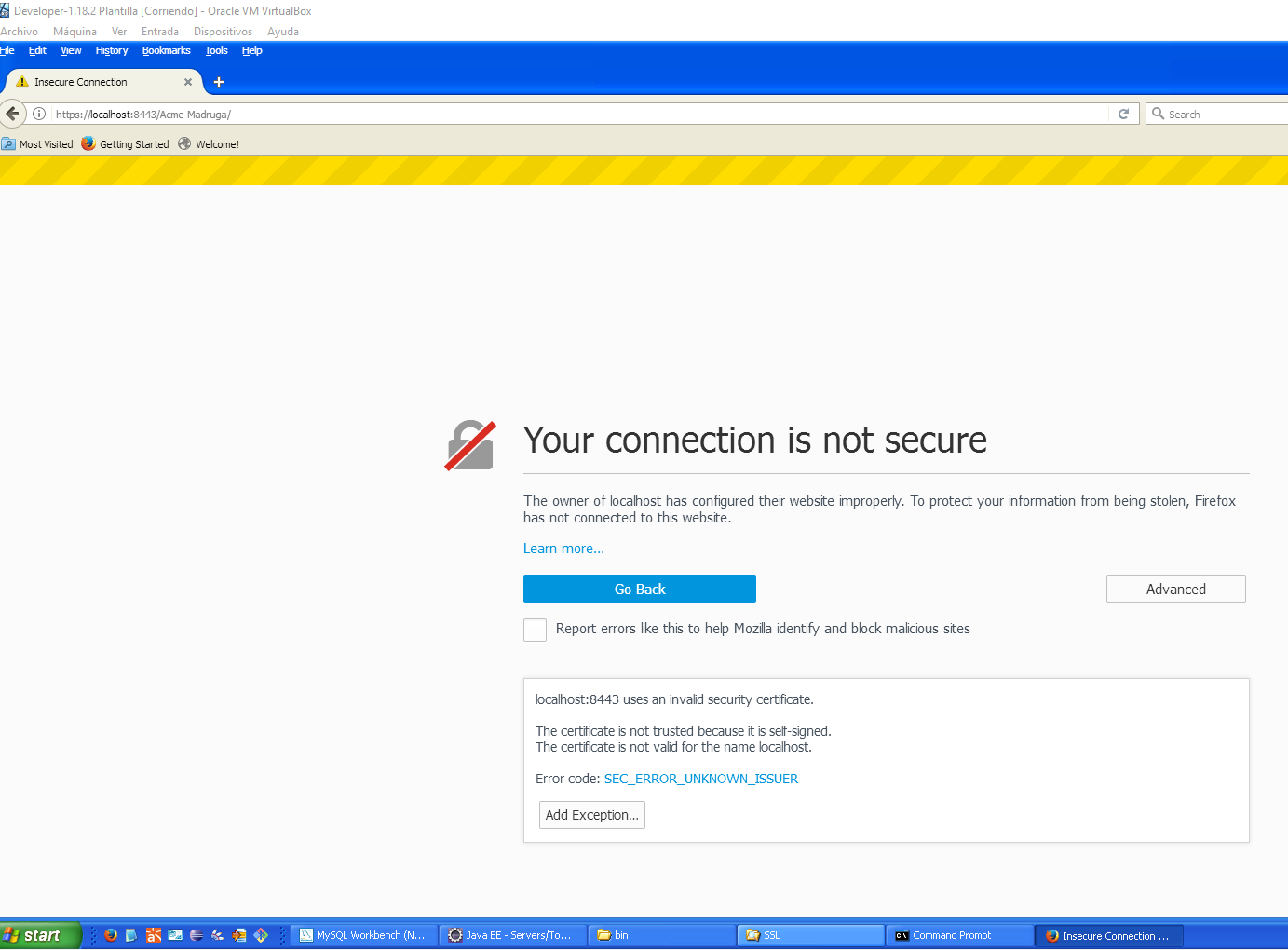
<Connector SSLEnabled="true" clientAuth="false" keystoreFile="${user.home}/.keystore" keystorePass="$3cUr3=C0mM$"

maxThreads="150" port="8443" protocol="HTTP/1.1" scheme="https" secure="true" sslProtocol="TLS"/>



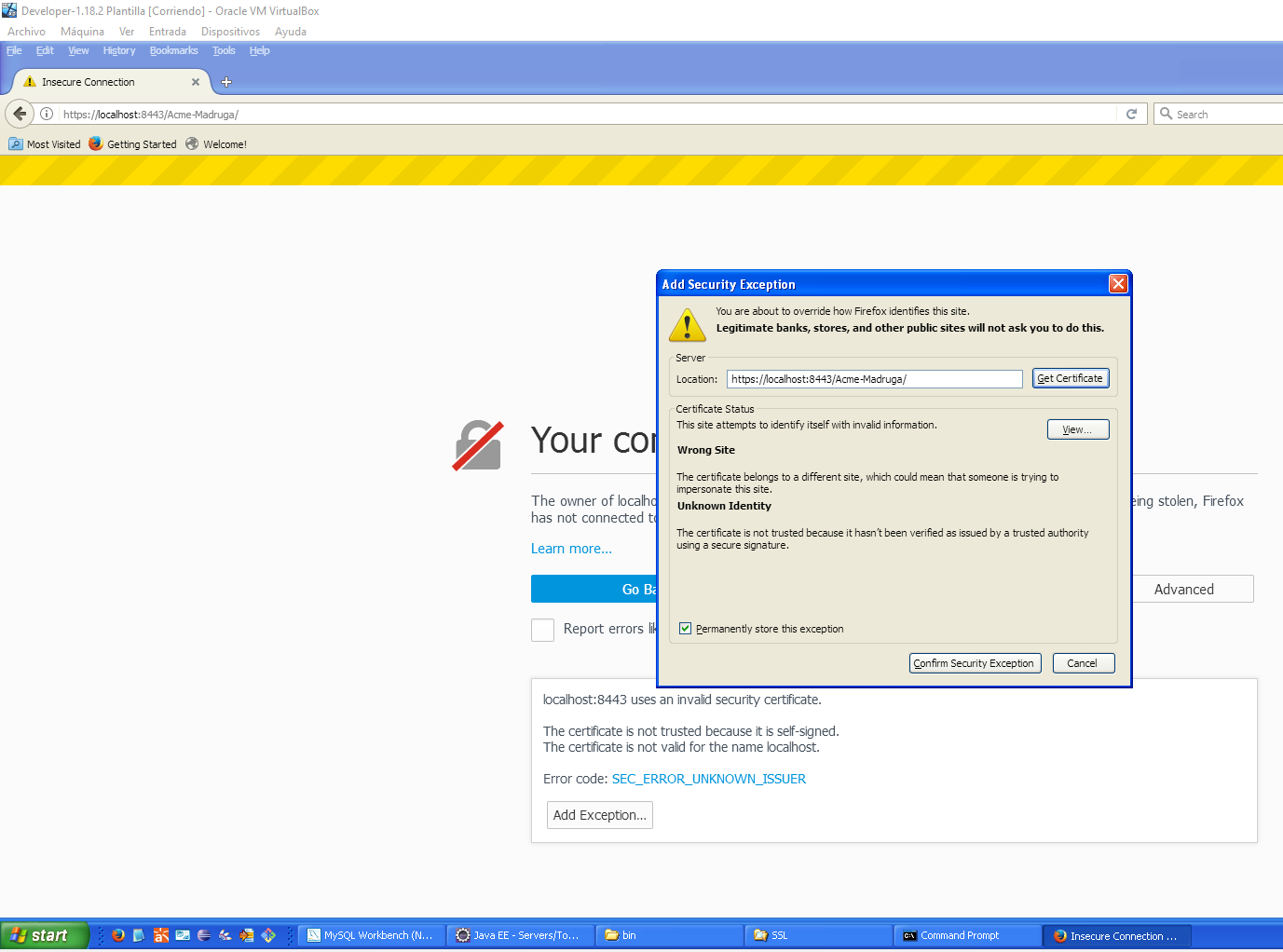
# Checking the results

Now that our project is ready, go to <http://localhost:8080/Acme-Madruga/>



If we see this page, we succeeded configuring https for our application, but since we created our own certificate, the browser will warn us that is probably an insecure page.

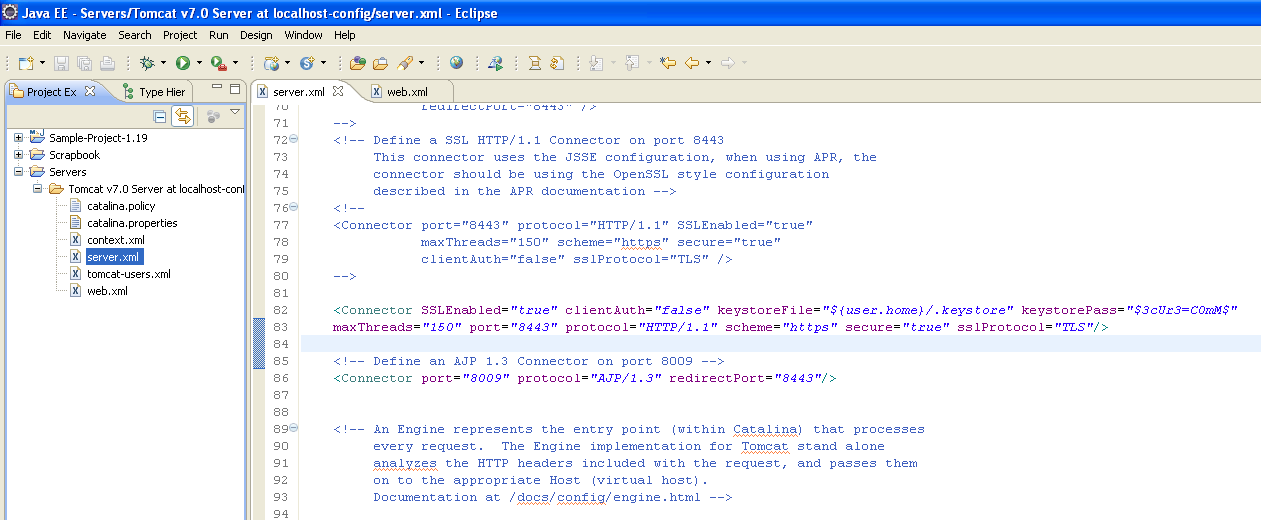
Add the exception to the browser and confirm the security exception.



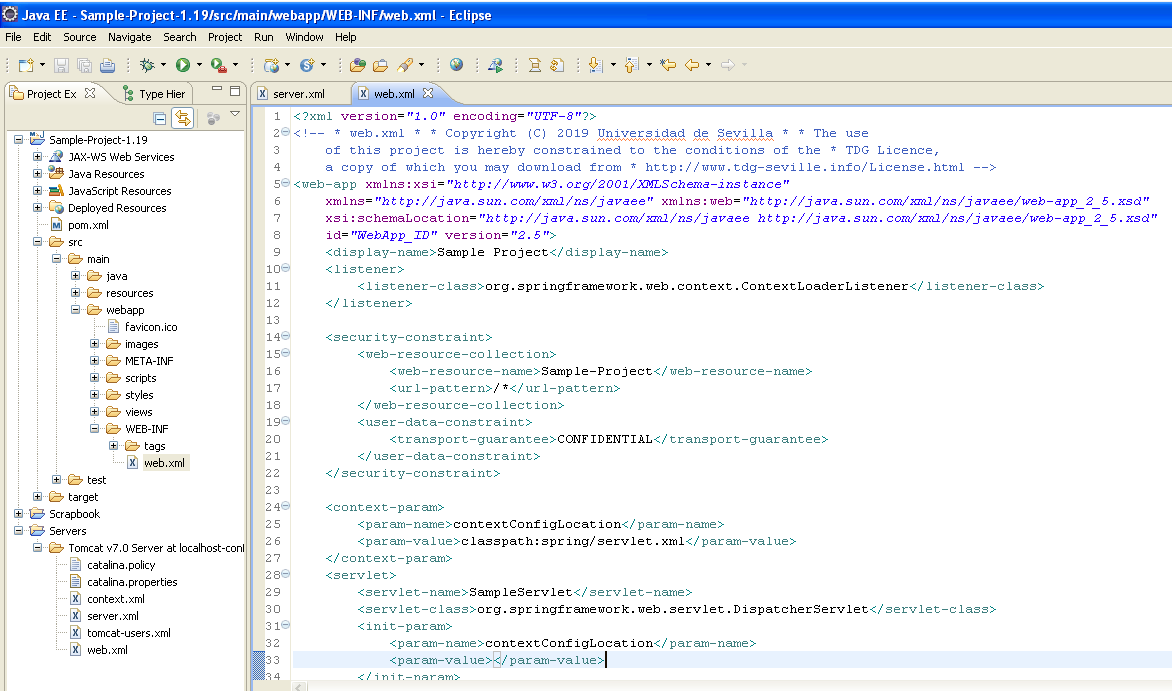
# Sample-project

Following this same steps will configure https to the sample-project:

Tomcat server.xml:



Sample project web.xml:



The certificate is the same that we used before in Acme-Madruga.

Results:

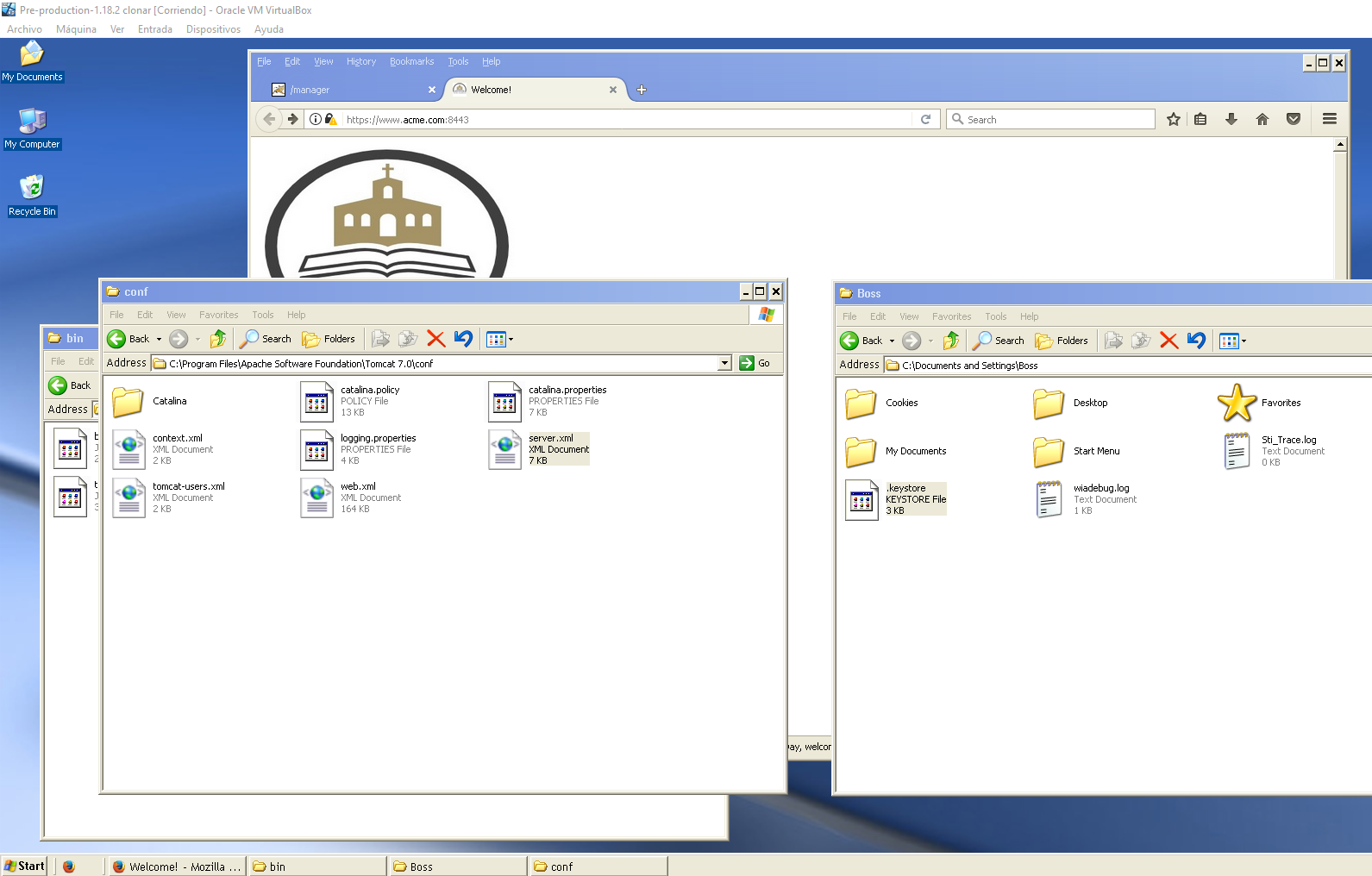


If you want to check this results please use the sample project and sample workspace that we attached to item 2 and move the certificate (inside sample workspace folder) to the users folder, in our case is C:\Documents and Settings\Student.

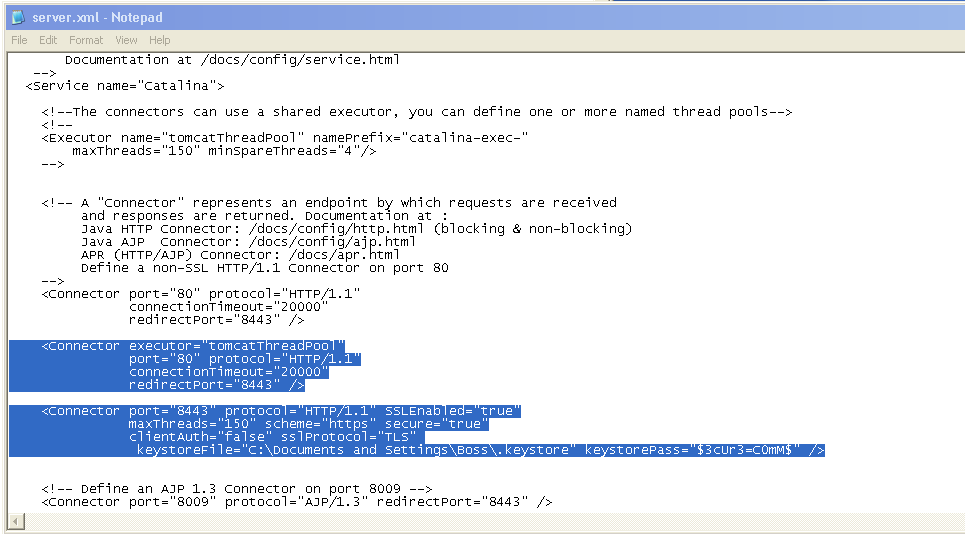
# Pre-Production

To deploy with https configuration in the pre-production machine we need to do some additional steps.

First, copy the key generated previously (.keystore) and put it in C:\Documents and Settings\Boss. Once you do this, go to C:\Program Files\Apache Software Foundation\Tomcat 7.0\conf\server.xml and open it with notepad.



This file is practically the same file we modified in the developer machine but now we need to uncomment and additional line. Make sure you add the parameters keystore and keypass to the connector port like this:



<Connector executor="tomcatThreadPool"

port="80" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443" />

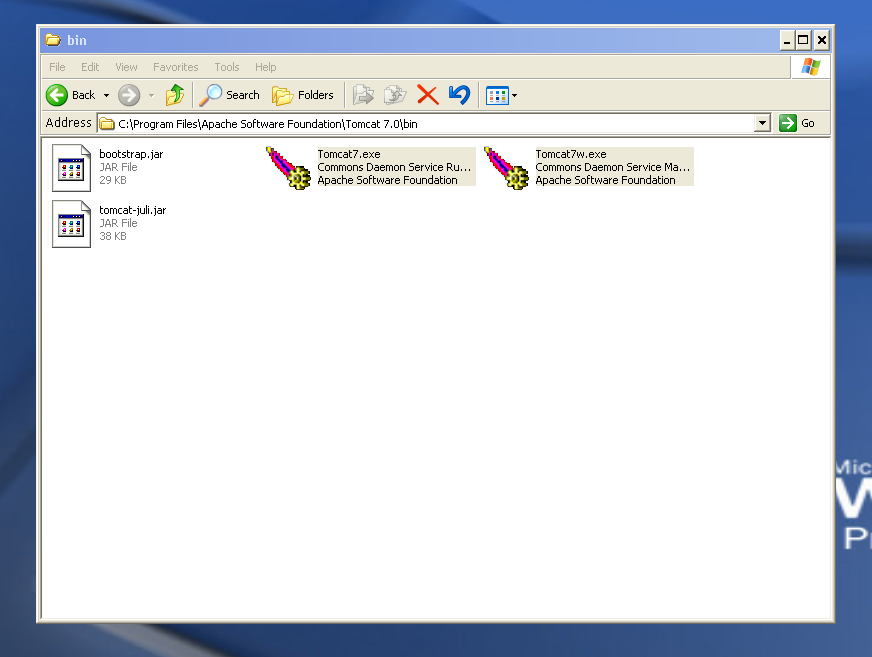
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS"

keystoreFile="C:\Documents and Settings\Boss\.keystore" keystorePass="$3cUr3=C0mM$" />

Now we need to restart Tomcat. Go to C:\Program Files\Apache Software Foundation\Tomcat 7.0\bin and execute TomCat7w.exe and Tomcat7.exe , in this order, to restart the server.



Create the database and deploy the war like always and open [www.acme.com](http://www.acme.com) in the browser, it will change automatically to <https://www.acme.com:8443/> allowing us to use https.

