Configuring TWCloud Admin

On this page

Teamwork Cloud features the new Webapp Platform-based TWCloud Admin Console. As such, it is a standalone application that communicates with Teamwork Cloud using the REST API.

Configuration of its communication with Teamwork Cloud is located in *<installation_directory>/WebAppPlatform/shared/conf/webappplatform.properties*.

In this section, we will review the various settings which you may have to adjust in order to establish communications between the admin console and Teamwork Cloud. Changes to these settings are only necessary if one is not using a default installation.

```
#
# Authentication server properties
#
# Authentication server address
# http/https depending on setup of Authentication server.
authentication.server.uri=https://IP_ADDRESS:8555/authentication
```

Authserver access

If you are accessing via a hostname or FQDN, especially if you are using a signed certificate, use the applicable FQDN or hostname instead of the IP address.

If you have configured authserver to use HTTP or to run on a different port, make sure that the URI reflects the correct values.

```
#
# Teamwork Cloud server properties
#
twc.admin.username=Administrator
twc.admin.password=Administrator
# Teamwork Cloud server address
# http/https depending on setup of Authentication server.
twc.url=https://IP_ADDRESS:8111
```

TWCloud access

Please make sure these credentials for **twc.admin.username** and **twc.admin.password** match those of a user with administrative privileges.

If you are accessing via a hostname or FQDN, especially if you are using a signed certificate, use the applicable FQDN or hostname instead of the IP

If you have configured TWCloud to use HTTP or to run on a different port, make sure that the URI reflects the correct values.

If you change any of the configuration parameters, you will need to restart the WebApp service.

Setting server protocol

By default, and in order to enforce a higher level of security, the admin console is accessed via HTTPS. In order to change the mode of operation to HTTP (not recommended), various configuration changes must be made.

The default port for the admin console is *8443*. In this example, we will make the changes necessary to run over HTTP on the default port of *8443*.

The WebApp server configuration is located in *<installation_directory>/WebAppPlatform/conf/server.xml.*

The following section:

```
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
```

needs to be edited to:

```
    <Connector executor="tomcatThreadPool"
               port="8443" protocol="HTTP/1.1"
               connectionTimeout="20000" />
```

The changes which we implemented consist of changing the port from *8080* to *8443*, and removing a redirect that would route to the handler on port *8443*.

Since we have configured this connector to listen on port *8443*, we now need to remove the existing connector handler on port *8443*.

The following section:

```
    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               sslImplementationName="org.apache.tomcat.util.net.jsse.JSSEImplementation"
               maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
        <Certificate certificateKeystoreFile="../configuration/keystore.p12"
                     certificateKeystorePassword="nomagic"
                     type="RSA" />
      </SSLHostConfig>
    </Connector>
```

needs to be commented out as follows:

```
<!--
         <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               sslImplementationName="org.apache.tomcat.util.net.jsse.JSSEImplementation"
               maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
        <Certificate certificateKeystoreFile="../configuration/keystore.p12"
                     certificateKeystorePassword="nomagic"
                     type="RSA" />
      </SSLHostConfig>
    </Connector>
-->
```

By default, for security reasons, we have established a security policy requiring access to be encrypted. To disable this, we need to edit *<installation_directory>/WebAppPlatform/conf/web.xml*. This section is located at the very bottom of the file.

The following section:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>webapp</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

needs to be edited as follows:

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>webapp</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <user-data-constraint>
        <transport-guarantee>NONE</transport-guarantee>
    </user-data-constraint>
</security-constraint>
```

In the next example, we will configure the Admin Console to run HTTPS on a different port (*8444*).

The following code section:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               sslImplementationName="org.apache.tomcat.util.net.jsse.JSSEImplementation"
               maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
        <Certificate certificateKeystoreFile="../configuration/keystore.p12"
                     certificateKeystorePassword="nomagic"
                     type="RSA" />
      </SSLHostConfig>
    </Connector>
```

needs to be edited as follows:

```
<Connector port="8444" protocol="org.apache.coyote.http11.Http11NioProtocol"
               sslImplementationName="org.apache.tomcat.util.net.jsse.JSSEImplementation"
               maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
        <Certificate certificateKeystoreFile="../configuration/keystore.p12"
                     certificateKeystorePassword="nomagic"
                     type="RSA" />
      </SSLHostConfig>
    </Connector>
```

As can be seen, the only change is the definition of the port number, which changed from 8443 to 8444.

If you change either the protocol or the port from the default, you need to edit **authentication.redirect.uri.whitelist,** located in *<installation_directory>/AuthServer/config/authserver.properties* accordingly.

Changing the SSL certificate

By default, the Admin console uses a self-signed certificate that is provided with the build. This is the same keystore used by TWCloud and Authserver, and is located in *<install_directory>/configuration/keystore.p12*.

If a signed certificate is being used to replace the self-signed certificate, we need to update configurations in three files: *<installation_directory>/configuration/application.conf***,** *<installation_directory>/AuthServer/config/authserver.properties* and *<installation_directory>/WebAppPlatform/conf/server.xml.*

To list the aliases of the using the command:

```
 <path_to_java_bin_directory>/keytool -v -list -keystore <keystorefile>
```

For this example, the keystore file is the default **keystore.p12**. The command is being executed from the same directory where **keystore.p12** is located. When the command is executed, you will be prompted for the keystore password. For our self-signed certificate (**keystore.p12**), it is *nomagic*.

```
# <path_to_keytool>/keytool -v -list -keystore keystore.p12
Enter keystore password:
Keystore type: PKCS12
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: teamworkcloud
Creation date: Oct 30, 2018
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=10.254.254.56
```

You will execute this command on whichever certificate you will be using. In this case, the alias is **teamworkcloud** and the certificate was generated for an **Owner** with a common name (CN) of 10.254.254.56, which happens to be a self-signed certificate for a machine with IP 10.254.254.56. Your keystore may contain multiple certificates with different aliases. You will identify the relevant one based on the Owner information. Once we have this information, we can proceed with the configuration.

For this example, we will assume that our new certificate is named **server.p12**, the keystore password is "*mypassword*" and the alias is "*myserver*", and that we will export the certificate into a file named **myserver.crt**.

First, copy it to the *<install_directory>/configuration/* directory.

Next, we need to export the certificate so that we can import it into the truststore (*<teamwork_cloud_install_directory>/AuthServer/config/truststore.jks*):

```
<path_to_keytool>/keytool -export -keystore <teamwork_cloud_install_directory>/configuration/server.p12 -
storepass mypassword -alias myserver -file <teamwork_cloud_install_directory>/AuthServer/config/truststore
/myserver.crt
```

Now we will proceed to edit **application.conf**.

```
                ssl {
                        keystorePath = "configuration/server.p12"
                        keystoreType = "pkcs12"
                        keystorePassword = "mypassword"
                        keyPassword = "mypassword"
                }
```

```
 https {

                        # the file name of the certificate or the key store (should be a full path)
                        file = "AuthServer/config/truststore/myserver.crt"

                        # certificate_mode: "true" if the file is a certificate; "false" if the file is a key
store.
                        is_certificate_file = true

                        # key store password
                        password = ""
                }
```

Next, we proceed to edit **authserver.properties**.

```
server.ssl.key-store=../configuration/server.p12
server.ssl.key-store-type=PKCS12
server.ssl.key-store-password=mypassword
server.ssl.key-password=mypassword
server.ssl.key-alias=myserver
```

Next, we need to delete the truststore (*<teamwork_cloud_install_directory>/AuthServer/config/truststore.jks*), so that it will be recreated upon restarting authserver.

Finally, we will edit **server.xml**.

```
    <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
               sslImplementationName="org.apache.tomcat.util.net.jsse.JSSEImplementation"
               maxThreads="150" SSLEnabled="true">
      <SSLHostConfig>
        <Certificate certificateKeystoreFile="../configuration/server.p12"
                     certificateKeystorePassword="mypassword"
                     certificateKeyAlias="myserver"
                     type="RSA" />
      </SSLHostConfig>
    </Connector>
```

Please note the addition of "*certificateKeyAlias*". This is not always necessary, but we do it for good measure. Tomcat will load the first certificate in the keystore. If there are multiple certificates, the alias is necessary in order to load the correct certificate.

After completing the configuration changes, all 3 services (Teamwork Cloud, Authserver, and Webapp) must be restarted.