Fediz Websphere

IBM Websphere Plugin

Apache CXF Fediz ships a plugin to secure an IBM Websphere 7/8 Application Server using WS-Federation.

This page describes how to enable Federation for a IBM Websphere Application Server (WAS) instance hosting Relying Party (RP) applications. This configuration is not for a Websphere instance hosting the Fediz IDP and IDP STS WARs but for applications that use SAML assertions for authentication. After this configuration is done, the Websphere-RP instance will validate the incoming SignInResponse created by the IDP server.

Prior to doing this configuration, make sure you've first deployed the Fediz IDP and STS on the separate Servlet Container instance as discussed here, and can view the STS WSDL at the URL given on that page. That page also provides some tips for running multiple Tomcat instances on your machine.

Websphere Security

A **Trust Authentication Interceptor (TAI)** is a pluggable security component that is installed and configured at the IBM WebSphere Application Cell level. As such, any managed server on the Cell will have this component installed in and activated once defined in the WAS Security configuration.
A TAI implements the WAS specific interface `com.ibm.wsspi.security.tai.TrustAssociationInterceptor`. The WAS specific API for security layer customization is explained in details at the following:

http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.base.doc%2Finfo%2Faes%2Fae%2Frsec_taisubcreate.html

The Fediz Plugin for Websphere provides a TAI implementation which leverages the **Fediz Core**.

WAS security runtime supports a notion of a security session using a specific security token called *LTPA Token* which is implemented as a HTTP cookie. The cookie lifetime is specified at the WAS administrative *Cell* level, which implies that it is not possible to configure this value per request based on the requirements for an application.
The TAI is no more involved after login once the LTPA Token is set which means a Web Application level component must intercept each request to check the security token (ex. SAML) lifetime and redirect the browser back to the IDP for re-authentication.
The Fediz Plugin Websphere ships a Java Servlet Filter which enforces the validity of the lifetime of the security token. This Servlet Filter must be configured in each Web Application module `web.xml` that is deployed on WAS.

Build Fediz Websphere Library

You have to build the Fediz plugin on your own as it depends on IBM Websphere libraries.

- Build the Websphere sources packaged within the downloadable distribution archive or checkout the Fediz sources here
- Add the library `runtime.jar` of IBM Rational Application Developer to your Maven repository
  ```
  mvn install:install-file -Dfile=<path-to-file> -DgroupId=com.ibm.ws -
  DartifactId=runtime -Dversion=7 -Dpackaging=jar
  ```
- run the maven command
  `mvn clean install -Pwebsphere`
  The Maven profile `websphere` enforces building the module `plugins/websphere`.
- You'll find the required libraries in `plugins/websphere/target/...zip-with-dependencies.zip`

Installation

Pre-Requisites

The Administrative and Application security must be activated for the WAS security layer to be able to intercept secured resources access requests. The local User Registry must be properly configured and at least one group of users must be declared in the registry prior any application installation.
At runtime, the WAS security layer will use the defined User/Group registry and the Fediz plugin maps the roles in the SAML token to WAS groups from this registry using the specified *Role to Group* mapper.
At deployment time, the declared J2EE security roles will need to be mapped to these groups, either using the Administrative Console or using the WAS binding files.

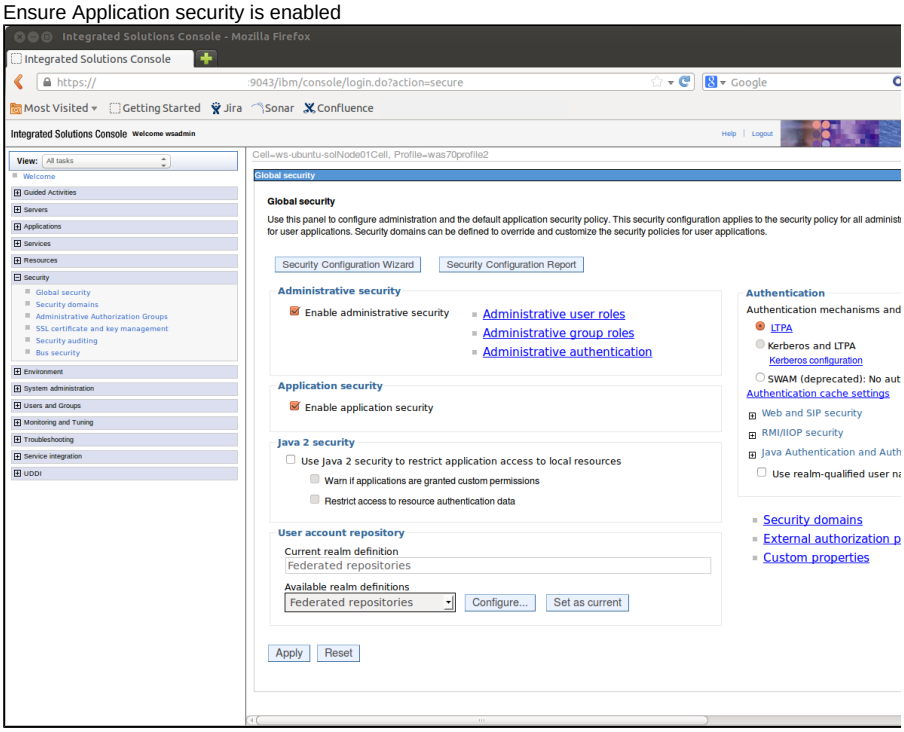
Plugin Installation

The Fediz Websphere plugin and its dependencies must be copied in the `WAS_INSTALL_ROOT>/lib/ext` directory of WebSphere Application Server, on each configured Node of the Cell (including the Deployment Manager)

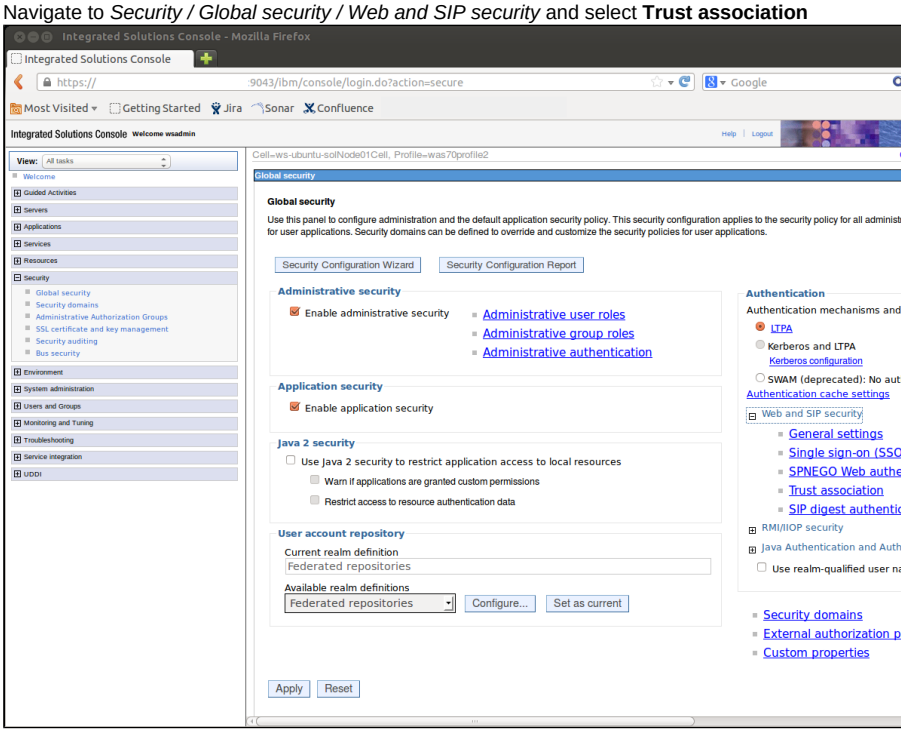
The Fediz configuration file (ex. `fediz-config.xml`) and the configured truststore should be copied in a directory with read permission for the WAS runtime user, on each configured Node of the Cell (including the Deployment Manager).
Note: Using a shared filesystem is recommended.

Web Application configuration

1. Open the Administative Console with Administrator privileges and navigate to Security / Global security
2. Ensure Application security is enabled

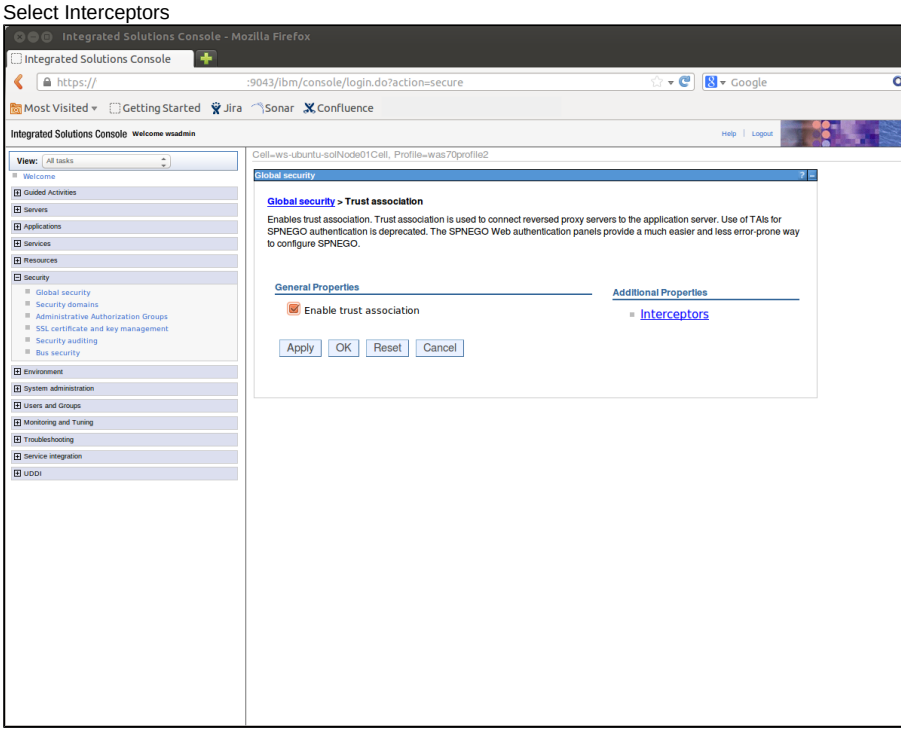


3. Navigate to *Security / Global security / Web and SIP security* and select **Trust association**

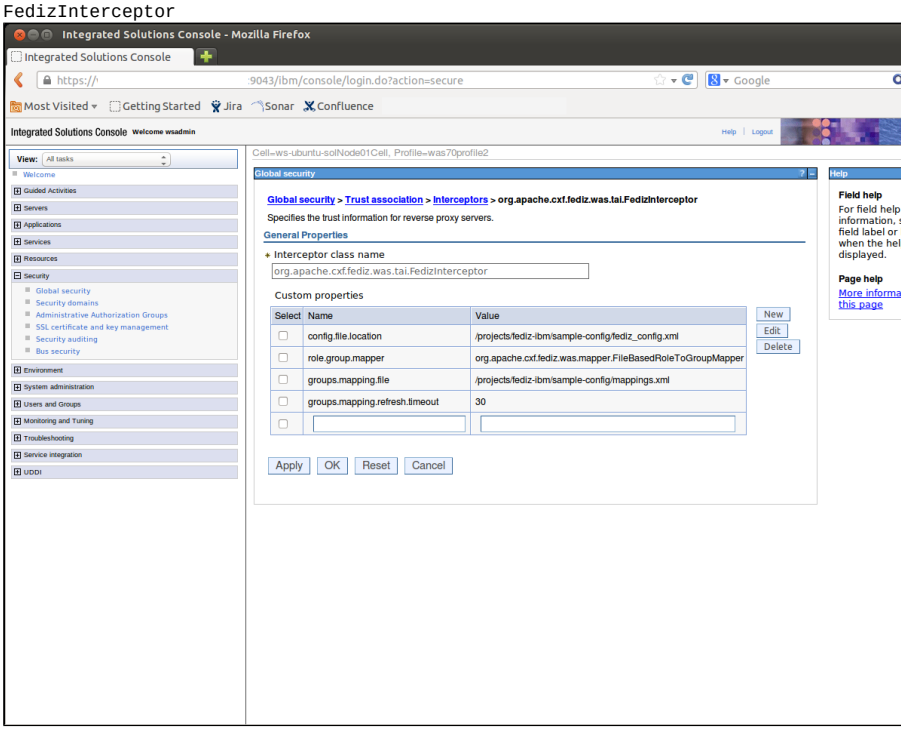


4. Check the **Enable trust association** check box

5. Select Interceptors



6. Click on New and specify the Interceptor class name as `org.apache.cxf.fediz.was.tai.FedizInterceptor`



Property	Value
config.file.location	Specify the path to the fediz-config.xml file
role.group.mapper	Specify the class of the Role to Group Mapper `org.apache.cxf.fediz.was.mapper.FileBasedRoleToGroupMapper`
groups.mapping.file	Specify the path to the Role - Group mapping file
groups.mapping.refresh.timeout	Specify the refresh time (in sec) to reload the Group mapping file

The file defined in `groups.mapping.file` must have the following structure:

Listing 1. roleGroupMapping.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<mapping>
```

```
  <samlToJ2EE>
    <claim>User</claim>
    <groups>
      <j2eeGroup>Authenticated</j2eeGroup>
      <j2eeGroup>Users</j2eeGroup>
    </groups>
  </samlToJ2EE>
  <samlToJ2EE>
    <claim>Manager</claim>
      <groups>
        <j2eeGroup>Manager</j2eeGroup>
        <j2eeGroup>Authenticated</j2eeGroup>
      </groups>
  </samlToJ2EE>
  <samlToJ2EE>
    <claim>Admin</claim>
      <groups>
        <j2eeGroup>Admin</j2eeGroup>
        <j2eeGroup>Authenticated</j2eeGroup>
      </groups>
  </samlToJ2EE>
</mapping>
```

A role value defined in element `claim` is mapped to a list of the Websphere JEE groups defined in `j2eeGroup`. Finally, these Websphere groups must be mapped to JEE roles. This indirection is required within Websphere.

Fediz configuration

The Fediz related configuration is done in a Servlet Container independent configuration file which is described here.

Federation Metadata document

The Webpshere Fediz plugin supports publishing the WS-Federation Metadata document which is described h ere.