A directory with tools is needed for this tutorial: *Tools_PicketLink_v1.0.*
All commands of this tutorial are done from the directory where the file above is located.

1/ How to install JBoss AS 7.2.0

PicketLink needs to get running in different servers :

- JBoss Application Server 5
- JBoss Application Server 7
- Apache Tomcat 6

In this tutorial, we use JBoss AS 7.2.0.
JBoss AS is an application server, written in Java, and implements the Java Platform, Enterprise Edition (Java EE) specifications.

A .zip file is available in the following directory :
Tools_PicketLink_v1.0/outils/zip/jboss-as-7.2.0.Final.zip

Unzip this file in directory /usr/local :

```
sudo unzip Tools_PicketLink_v1.0/outils/zip/jboss-as-7.2.0.Final.zip -d /usr/local/
```

A directory “jboss-as-7.2.0.Final” has been created in /usr/local but we cannot access it.
Change the access permissions :

```
sudo chmod -R 755 /usr/local/jboss-as-7.2.0.Final/
```

JBoss AS 7.2.0 is now available.

2/ How to install PicketLink 2.1.8

PicketLink is a project for security and identity management for Java Applications.
It provides an identity provider web application with the OASIS standard WS-Trust. It defines extensions that build on WS-Security to provide a framework for requesting and issuing security tokens.

The project is available in the following directory :
Tools_PicketLink_v1.0/outils/appli/picketlink-quickstarts/

The service that particularly interest us is the Security Token Service (STS) :
Tools_PicketLink_v1.0/outils/appli/picketlink-quickstarts/ws-trust/picketlink-sts/

Enter in the above directory and execute a maven build :

```
cd Tools_PicketLink_v1.0/outils/appli/picketlink-quickstarts/ws-trust/picketlink-sts/
```

```
mvn -Dbinding=jboss -Dbinding-version=as7 clean install
```

If the build was successful, a .war file is available in the directory target/ :
Tools_PicketLink_v1.0/outils/appli/picketlink-quickstarts/ws-trust/picketlink-sts/target/picketlink-sts-2.1.8.Final-jboss-as7.war

Then we have to copy the above .war file in the deployment directory of JBoss :

```
sudo cp Tools_PicketLink_v1.0/outils/appli/picketlink-quickstarts/ws-trust/picketlink-sts/target/picketlink-sts-2.1.8.Final-jboss-as7.war /usr/local/jboss-as-7.2.0.Final/standalone/deployments/
```

Now we have to add security-domain configurations to the standalone.xml.
Open the above file in a text editor (Vim for example) :

```
sudo vim /usr/local/jboss-as-7.2.0.Final/standalone/configuration/standalone.xml
```

Then add the following configurations between tags <security-domains> … </security-domains> :

```
<security-domain name="idp" cache-type="default">
   <authentication>
      <login-module code="UsersRoles" flag="required">
         <module-option name="usersProperties" value="users.properties" />
         <module-option name="rolesProperties" value="roles.properties" />
      </login-module>
   </authentication>
</security-domain>
<security-domain name="picketlink-sts" cache-type="default">
   <authentication>
      <login-module code="UsersRoles" flag="required">
         <module-option name="usersProperties" value="users.properties" />
         <module-option name="rolesProperties" value="roles.properties" />
      </login-module>
   </authentication>
</security-domain>
<security-domain name="sp" cache-type="default">
   <authentication>
      <login-module
code="org.picketlink.identity.federation.bindings.jboss.auth.SAML2LoginModule"
flag="required"/>
   </authentication>
</security-domain>
```

Finally, we have to execute a PicketLink installer in order to configure the JBoss Environment.
Enter in the directory and execute the command “ant”:

```
cd Tools_PicketLink_v1.0/outils/install/picketlink-installer-2.7.0.Final/
sudo ant
```

Some instructions are asked. Here is the answers we used in our JBoss configuration :
- Which JBoss Application Server are you using ? ([eap], wildfly)
eap
- Please enter the path to your JBoss Application Server installation:
/usr/local/jboss-as-7.2.0.Final

Now we can deploy the .war file :

```
sudo /usr/local/jboss-as-7.2.0.Final/bin/standalone.sh
```

If the connection to the JBoss Web Server is made for the first time, we have to create an user.
Execute the following command :

```
sudo /usr/local/jboss-as-7.2.0.Final/bin/add-user.sh
```

Then follow the steps :

```
- What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
a
- Enter the details of the new user to add.
*type Enter, do not write*
- Username
*choose an username*
- Password
*choose a password*
```

Answer yes to both final questions.

You can use your user information to access to to the Web Service available here :
http://127.0.0.1:9990/

The WSDL of PicketLink is available to the following web url (login : tomcat & password : tomcat) :
http://localhost:8080/picketlink-sts?wsdl

3/ How to check if the web service works

a) With a java class test

We can use an example of a client application that invokes PicketLink STS to get a custom token.
The project is available in the following directory :
Tools_PicketLink_v1.0/outils/appli/projectPL/

When importing this project in your Integrated Development Environment (IDE), some libraries are needed. They are available here :
Tools_PicketLink_v1.0/outils/lib/

Then run the following class :
projectPL/src/projectPL/WSTrustClientTest.java

If the web service works, the following message will appears :
Successfully issued a standard SAMLV2.0 Assertion!

Is assertion valid? true

b) With SoapUI

We can use the software SoapUI in order to generate a request.

Create a new SOAP Project.
File, New SOAP Project.

In the new window, enter the following adress in front of “Initial WSDL” :
http://localhost:8080/picketlink-sts?wsdl

The rest of information needed appears by themselves.
Then click on the OK button.

An authentication is asked :
Username : tomcat
Password : tomcast

In the tree at the left of the software, reach the “Request 1” file. A new window appears where we can write a request.
On the bottom, click on the menu tab “Auth” and add a new basic authorization with the same user information as above.

Now click on the menu tab “WS-A” and tick the four boxes available and select the Must understand option to FALSE.

Finally, use the following request in order to check the web service by clicking on the green arrow button :

```
<soap:Envelope xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/cd/ws-trust.xsd"
xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:picketlink:identity-
federation:sts">
   <soap:Header/>
   <soap:Body>
   <wst:RequestSecurityToken Context="context">
         <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</wst:TokenType>
         <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</wst:RequestType>
      </wst:RequestSecurityToken>
   </soap:Body>
</soap:Envelope>
```

Note : At the moment, the web service only works with the Must understand option to FALSE.