IDP 3 - Step 2 - Shibboleth installation

Previous step: Install and configure Java and Tomcat

Table of contents for step 2

Install the IDP software

Before running the installer safe off the current umask or your `root` user (so that we can later restore it) and change it to the value given below:

```
oldumask=$(umask)
umask 0022
```

The IDP installer will ask for two passwords: One to protect a newly generated PKCS#12 keystore (for a SOAP/backchannel connector, configuration of which which we've dropped from this documentation), the other as Cookie/localStorage encryption key (for client-side session storage). So generate *two* random strings to be used as passwords and note them down somewhere temporarily but securely, indicating their purpose (backchannel, cookie encryption). The cookie encryption password will be written to `/opt/shibboleth-idp/conf/idp.properties` by the installer, though.

```
openssl rand -hex 16  # run twice to generate two random strings
```

Download and unpack the latest Shibboleth IDP software, adjusting the value of $VER to the latest/current version. Optional (but recommended, if you understand how PGP and the Web of Trust work) commands for verification of the software using cryptographic signatures from the Shibboleth devlopers are also included below.

```
export VER=3.4.8
cd /usr/local/src
curl -s https://shibboleth.net/downloads/PGP_KEYS | gpg --import -
curl -O "https://shibboleth.net/downloads/identity-provider/latest3/shibboleth-identity-provider-$VER.tar.gz{,.
asc}"
gpg --verify shibboleth-identity-provider-$VER.tar.gz.asc
tar xzf shibboleth-identity-provider-$VER.tar.gz
cd shibboleth-identity-provider-$VER
./bin/install.sh < /dev/null
```

When prompted:

⚠ If the installer appears to be "hung" it's probably just sitting there *waiting for you* to enter something or to hit `<Return>` to continue!

1. `Source (Distribution) Directory`": Accept the current directory by hitting `<Return>`
2. `Installation Directory`: Accept the default (`/opt/shibboleth-idp`)
3. `Hostname`: Enter the publicly visible FQDN of your IDP's webserver as hostname – the one you generated a TLS server certificate for previously
4. `SAML EntityID`: Accept the suggested default (unless you already have an IDP this install should replace, then enter *your current IDP's entityID*)
5. `Attribute Scope`: Enter the canonical DNS domain for your institution, e.g. "univie.ac.at", to be used for scoped attributes (or your currently used scope)
6. `Backchannel PKCS12 Password`: Enter the previously generated password for the (to be generated, but ignored by this documentation) backchannel keystore
7. `Cookie Encryption Key Password`: Enter the previously generated password to protect the (to be generated) Cookie encryption key

This should result in a `BUILD SUCCESSFUL` message and a Web Archive file in `/opt/shibboleth-idp/war/idp.war`

Adjust Tomcat configuration

Since we want the IDP (and hence Apache Tomcat and the JVM) to be run as a non-priviledged user we'll need to adjust a couple of file system permissions:

```
chown tomcat /opt/shibboleth-idp/{logs,metadata}
chgrp tomcat -R /opt/shibboleth-idp/{credentials,conf}
chmod g+r -R /opt/shibboleth-idp/conf
chmod 640 /opt/shibboleth-idp/credentials/*
chmod 750 /opt/shibboleth-idp/credentials
chmod g+w /opt/shibboleth-idp/credentials/sealer.*
```

As per the Shibboleth IDP documentation for Tomcat we'll need to make a few more adjustments:

Add a Context Deployment Fragment to Tomcat so it knows where to find the IDP's war file:

```
echo '<Context docBase="/opt/shibboleth-idp/war/idp.war"
    privileged="true"
    antiResourceLocking="false"
    swallowOutput="true" />' > /etc/tomcat9/Catalina/localhost/idp.xml
```

Following the recommendations from the Shibboleth wiki we also uncomment (i.e., make active) the line `<Manager pathname="" />` in Tomcat's `context.xml`. And since we have to change that file anyway let's replace it with a minimalist version that also avoids scanning (most) of the IDP's JAR files during startup, see section Slow Startup towards the end of that Shibboleth wiki page.

```
cp -a /etc/tomcat9/context.xml /etc/tomcat9/context.xml.`date -u +%Y%m%dT%H%M%S`

JARS=$(unzip -l /opt/shibboleth-idp/war/idp.war | grep WEB-INF/lib/. | sed -r 's/^.*WEB-INF\/lib\/(.+-)[0-9\.]+-?(RELEASE|GA|Final|[Bb]eta.?|.*avoid-conflict.*)?(-jre)?.jar$/\1*.jar,/' | tr '\n' ' ' | sed 's/, $//')

echo "<Context>
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <WatchedResource>\${catalina.base}/conf/web.xml</WatchedResource>
  <Manager pathname=\"\" />
  <JarScanner>
    <JarScanFilter
        pluggabilitySkip=\"\${tomcat.util.scan.StandardJarScanFilter.jarsToSkip}, $JARS\" />
  </JarScanner>
</Context>" > /etc/tomcat9/context.xml
```

Finally, to make the `status.sh` script work we'll need to add the Java Server Tag Library to the IDP that Tomcat is not re-distributing:

```
cd /opt/shibboleth-idp/edit-webapp/WEB-INF/lib/
curl -sSLO "https://build.shibboleth.net/nexus/service/local/repositories/thirdparty/content/javax/servlet/jstl/1.2/jstl-1.2.jar{,.asc}"
gpg --verify jstl-1.2.jar.asc && rm jstl-1.2.jar.asc
/opt/shibboleth-idp/bin/build.sh < /dev/null
```

Restart Tomcat, which may take a bit, and check the logs for `WARN` and `ERROR` messages: By default the IDP logs to `/opt/shibboleth-idp/logs/idp-process.log` but if something is seriously wrong and the IDP isn't even able to start up you'll have to look at Tomcat's journal entries:

```
systemctl restart tomcat9
multitail /opt/shibboleth-idp/logs/idp-process.log -l 'journalctl -u tomcat9.service -f'  # exit with 'q'
```

You can test whether the IdP is properly installed with the status command line utility:

```
/opt/shibboleth-idp/bin/status.sh
```

With these steps the installation – and therefore most of the OS-specific and GNU/Linux distribution-specific details – is done!
You can now restore a potentially deviating umask value to its previous value to close this chapter.

```
umask $oldumask
```

Now on to the configuration!

Next step: Configure Metadata