ApacheTomcat8

These pages are examples and do not reflect any normative requirements or assumptions on the part of the IdP software and may be a mix of suggestions from both the project team and deployers. You should take any of this advice with a grain of local salt and consider general security /deployment considerations appropriate to the use of web software in your local environment.

The official information about containers and versions we support is solely maintained on the SystemRequirements page. If you wish to operate without complete responsibility for your Java servlet container, you may consider the Windows package we provide that includes an embedded container.

Monitoring Tomcat 8.0 with JMX prior to the full initialization of the IdP web application appears to destablize the Spring environment or trigger some kind of bug, as noted here.

Using Apache Tomcat 8.0

Within this documentation, `idp.home` will be used to refer to IdP installation directory (as specified during the installation process; default is `/opt/shibboleth-idp`). `CATALINA_BASE` will be used to refer to the location of the Tomcat installation.

Version Requirements/Recommendations

There are no known issues with any specific Tomcat 8.0 release. The latest stable version should be used.

Tomcat 8+ (including 8.5 and 9.0) are also supported but we don't have a specific page at present for Tomcat 8.5 and 9.0. See also SystemRequirement for more info. Earlier versions e.g. 8.5.4 had session bugs that render the IdP unstable.

Required Configuration Changes

In order to run the IdP, Tomcat must be informed of the location of the IdP warfile. This should be done with a context descriptor by creating the file `CATALINA_BASE/conf/Catalina/localhost/idp.xml` and placing the following content in it (replacing `idp.home` with your IdP's home directory):

```
<Context docBase="idp.home/war/idp.war"
         privileged="true"
         antiResourceLocking="false"
         swallowOutput="true">

        <!-- Work around lack of Max-Age support in IE/Edge for Tomcat 8.0.x -->
        <CookieProcessor alwaysAddExpires="true" />

</Context>
```

The above `<CookieProcessor>` line is only for Tomcat 8.0.x. It is safe to remove that line on other versions.

- Tomcat listens on ports 8080 and 8443 for user-facing web traffic by default. You will most likely need to modify these ports to 80 and 443 in `CATALINA_BASE/conf/server.xml`, and make arrangements for Tomcat to run as root, use a port forwarding approach, or rely on some other solution, cf. IdPLinuxNonRoot and IdPLinuxNonRootDebianUbuntu.
- Tomcat does not provide the Java Server Tag Library, which is required to use JSP pages as Spring views. The IdP status page at `/idp/status` is built with JSP and will not work without this library. You can download it from our Maven repository here (asc), place it into `idp.home/edit-webapp/WEB-INF/lib/`, then change to `idp.home` and run `./bin/build.sh` (or build.bat, depending on your platform).
  More details can be found at http://stackoverflow.com/tags/jstl/info
- Add the following parameters to the CATALINA_OPTS environment variable (on Windows, the CATALINA_OPTS variable can be set via the "Manage Tomcat" application in the "Java" Tab; on other systems, the file `bin/setenv.sh` can be created to set variables during startup):
  - If you chose to install to a location other than the default (/opt/shibboleth-idp):
    - **-Didp.home=<location>** (replacing <location> with your install location)
      Note: On windows in versions prior to 3.2.0 if the install location contains a space then you have to provide the shortname (**-Didp.home=c:\progra~1\Path\To\Install**). This is best achieved by using the `tomcatw.exe` program.
    - In V3.1.2 or later, idp.home can be set as a context-parameter in web.xml (copied to edit-webapp and then the war rebuilt using the build command)

```
<context-param>
    <param-name>idp.home</param-name>
    <param-value>J:/Downloads/Shibboleth/IdP</param-value>
</context-param>
```

- **-XX:+UseG1GC** - enables alternate garbage collector that reduces memory usage on larger metadata files
- **-Xmx1500m** - this is the maximum amount of memory that Tomcat may use, at least 1.5G is recommended for handling larger (> 25M) metadata files but you will need to test on your particular metadata configuration
- **-XX:MaxPermSize=128m** - the maximum amount of memory allowed for the permanent generation object space (this setting applies only to Java 7)

Recommended Configuration Changes

- Limit the allowed size of POST submissions to any HTTP or AJP connectors (including the SOAP connector below) by adding the `maxPostSize` attribute. A size of about 100K (100000) is a reasonable choice.
- Disable session persistence by uncommenting the `<Manager pathname="" />` line in *CATALINA_BASE/conf/context.xml* (as noted in the file). This prevents errors from being logged regarding the lack of persistence of the session objects created by the IdP when you stop the container. It is not possible to cluster the IdP using the Tomcat session manager.

Slow Startup

To minimize startup time, do NOT set `unpackWAR="false"` in *CATALINA_BASE/conf/Catalina/localhost/idp.xml*. See Tomcat 8 Deployment and this bug for details.

Tomcat is extremely inefficient in newer versions because of poor implementations of jarfile scanning required by the newest Servlet standard (see https://wiki.apache.org/tomcat/HowTo/FasterStartUp) This manifests in extremely slow startup time once the IdP warfile is deployed if the warfile is not unpacked (on the order of 4 minutes).

Adding the IdP library list to the list of jars to skip, as advised by that page, reduces this by 1-2 minutes, but creates a fairly large maintenance issue to keep up with.

To generate a list of jars to skip for a given install, a command such as this will generate output you can paste:

```
unzip -l /opt/shibboleth-idp/war/idp.war | grep WEB-INF/lib/. | sed 's/^.*WEB-INF\/lib\///' | awk '{print $1",
\\"}'
```

The list of comma-separated jarfiles is added into the file *CATALINA_BASE/conf/catalina.properties* to the property value named `tomcat.util.scan.StandardJarScanFilter.jarsToSkip`

While avoiding the `unpackWAR` option does fix this, it is almost a certainty that you will encounter problems with out of sync warfile content if you frequently make changes that required repacking and redeploying the warfile. Be aware of this and exercise caution. If you encounter unusual problems, manually purge unpacked warfile content in between restarts.

Logging

Any deployers that have advice on configuring logging in particular ways are welcome to edit this section.

Supporting SOAP Endpoints

The use of the back-channel is discussed in the SecurityAndNetworking topic, and you should review that to understand whether or not you need to support this feature.

If you do need this support, these connections generally require special security properties that are not appropriate for user-facing/browser use. Therefore an additional endpoint must be configured.

The plugin component that supports the requirements on the back-channel is available from here (asc) and needs to be copied to `CATALINA_BASE/lib`.

1. Add the following connector definition to `CATALINA_BASE/conf/server.xml`:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
        maxThreads="150"
        SSLEnabled="true"
        scheme="https"
        secure="true"
        clientAuth="want"
        keystoreFile="idp.home/credentials/idp-backchannel.p12"
        keystorePass="PASSWORD"
        keystoreType="PKCS12"
        trustManagerClassName="net.shibboleth.utilities.ssl.TrustAnyCertificate" />
```

2. Replace `idp.home` with the IdP home directory entered during installation.
3. Replace `PASSWORD` with the keystore password entered during installation.