Configuring a Shibboleth Identity Provider to join the Tuakiri Federation

For a Shibboleth Identity Provider to join one of the Tuakiri Federations (Test/Dev or Production), the following steps have to be done:

- Registering the IdP in the Federation Registry
- Configuring the IdP to load the federation metadata
- Configuring the IdP to release the attributes required by the federation.

There are two federations available, both fully operational:

- **Tuakiri Test/Development Environment** (**Tuakiri-TEST**) for testing deployments and developing new features
- **Tuakiri Production Federation Service** (**Tuakiri**) for ready-for-production Identity Providers and services.

We recommend first registering a Test system into Tuakiri-TEST and after successful testing, register a production-ready system into Tuakiri Production.

Federation Details

Federation name	Tuakiri Production	Tuakiri TEST
Metadata name	`tuakiri.ac.nz`	`test.tuakiri.ac.nz`
Metadata distribution point	https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-signed.xml	https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-signed.xml
Metadata signing certificate	https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem	https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-cert.pem
Federation Registry URL	https://registry.tuakiri.ac.nz/federationregistry/	https://registry.test.tuakiri.ac.nz/federationregistry/
Discovery Service / WAYF URL	https://directory.tuakiri.ac.nz/ds/DS	https://directory.test.tuakiri.ac.nz/ds/DS

Registering an IdP into the Federation Registry

Go to the respecting Federation Registry URL and:

- Register an *Organisation* for your institution (if not already registered)
  - For Contact Details, do not use a shared mailbox, alias or mailing list when entering an email address because the confirmation email contains a single-use link and may cause some confusion should more than one person attempt to use it.
  - For *Organization Name*, enter your **DNS domain name**.
  - For *Organization Display Name*, enter your actual organization name.
- Wait for the Organisation to be approved
- Register your IdP under that Organisation
  - Provide the Contact Details for the IdP admin (again, do not use a shared mailbox).
  - Select the organisation and provide a name and description for your IdP.
  - Enter the base URL for your IdP (`https://idp.example.org`).
  - Enter the PEM encoded certificate used by your IdP for signing Shibboleth assertions (the default is `$IDP_HOME/credentials/idp.pem`).
  - Select the attributes the IdP will be able to release to the federation.
  - Select supported NameID formats. By default, `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` is already selected.
  - Submit the details and wait for your IdP to be approved.
  - After having your IdP registration approved, click on the link sent to you to become an Administrator of the IdP's registration.

> ⚠ **Confirmation email**
>
> - It is important to click on the link in the confirmation email, as this makes the recipient of the email an administrator of the Identity Provider being registered in the Tuakiri Federation Registry.
>   - The link in the confirmation email can only be used once.
>   - Same applies for the link sent for the Organization registration.

ECP support

If supporting ECP, advertise also your ECP SSO EndPoint:

As of version 2.6.0, the Federation Registry automatically registers the ECP endpoint on new registrations, so no explicit action should be required. To add an ECP endpoint to an existing IdP registration, perform the following:

In the Federation Registry registration for your IdP:

- Add a new "Single Sin On Service" Endpoint
- Select Binding: `urn:oasis:names:tc:SAML:2.0:bindings:SOAP`
- Enter Location: `https://idp.example.org/idp/profile/SAML2/SOAP/ECP` (substituting your IdP hostname)

The IdP also needs to be configured to support ECP.

Configuring your IdP to load the federation metadata

The code snippets in this section have values for Tuakiri Production federation. Please update them accordingly as per the table above if configuring your IdP to join the Tuakiri TEST/DEV federation. (The key code snippets are for convenience given in the "Tuakiri-TEST specific" box below).

NOTE: Check what your *IdP home directory* is: the directory is typically called `shibboleth-idp` - and on Debian and Ubuntu systems, it's commonly `/usr/local/shibboleth-idp`, while on RedHat and CentOS it's `/opt/shibboleth-idp`. The snippets below are referring to the IdP home directory as `$IDP_HOME`

To configure a 3.x IdP to Load the Tuakiri metadata:

- Configure the IdP to load the Federation Metadata in `/opt/shibboleth-idp/conf/metadata-providers.xml` by adding the following snippet into the `Chaining MetadataProvider`.

```
    <MetadataProvider id="TuakiriMetadata"
                      xsi:type="FileBackedHTTPMetadataProvider"
                refreshDelayFactor="0.125"
                maxRefreshDelay="PT2H"
                backingFile="%{idp.home}/metadata/tuakiri-metadata.xml"
                metadataURL="https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-signed.xml">

            <MetadataFilter xsi:type="SignatureValidation"
                    certificateFile="${idp.home}/credentials/tuakiri-metadata-cert.pem"
                    requireSignedRoot="true">
            </MetadataFilter>
            <MetadataFilter xsi:type="EntityRoleWhiteList">
                    <RetainedRole>md:SPSSODescriptor</RetainedRole>
            </MetadataFilter>

    </MetadataProvider>
```

  - Note: validity checking is implicitly turned on, so it is not needed to explicitly add the `RequiredValidUntil` metadata filter, which would only be useful to reject metadata published with a validity longer then maxValidityInterval milliseconds. We recommend to rely on signature validation. The Tuakiri metadata are being generated with a validity of one week.
  - Note: by default, metadata get refreshed only every 3 hours (0.75 factor out of 4 hours maximum refresh interval).
    - To make metadata changes propagate faster (reload every 15 minutes), set the maximum refresh interval to 2 hours and the factor to 0.125 as above.
    - To avoid re-fetching the file even when not changed, turn on caching (memory caching is enough as we already do have a backing file)
  - See the IDP30 https://wiki.shibboleth.net/confluence/display/IDP30/MetadataConfiguration and https://wiki.shibboleth.net/confluence/display/IDP30/FileBackedHTTPMetadataProvider documentation for more information.
  - Note: in IdP 3.0.0, the `RetainedRole` element was incorrectly using the namespace `samlmd` - as of 3.1.1, the namespace declared in metadata-providers.xml and used in the examples is `md`, consistent with other use.

- This definition is referring to a certificate used to verify the signature - store the certificate in `/opt/shibboleth-idp/credentials`

```
wget https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem -O $IDP_HOME/credentials/tuakiri-metadata-cert.pem
```

Tuakiri-TEST specific

When building a TEST IdP and registering into Tuakiri-TEST instead, please load instead the Tuakiri-TEST metadata with:

```
    <MetadataProvider id="TuakiriTESTMetadata"
                      xsi:type="FileBackedHTTPMetadataProvider"
                  refreshDelayFactor="0.125"
                  maxRefreshDelay="PT2H"
                  backingFile="%{idp.home}/metadata/tuakiri-test-metadata.xml"
                  metadataURL="https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-
metadata-signed.xml">

            <MetadataFilter xsi:type="SignatureValidation"
                   certificateFile="${idp.home}/credentials/tuakiri-test-metadata-cert.pem"
                   requireSignedRoot="true">
            </MetadataFilter>
            <MetadataFilter xsi:type="EntityRoleWhiteList">
                   <RetainedRole>md:SPSSODescriptor</RetainedRole>
            </MetadataFilter>

    </MetadataProvider>
```

and fetch the Tuakiri-TEST metadata signing certificate instead:

```
wget https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-cert.pem -O $IDP_HOME
/credentials/tuakiri-test-metadata-cert.pem
```

For archival purposes, we also keep the original instructions for configuring the Tuakiri metadata into a 2.x IdP - unfold the box below to see the IdP 2.x compatible syntax:

- Download the metadata signing certificate into `$IDP_HOME/credentials`:

```
wget https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem -O $IDP_HOME/credentials
/tuakiri-metadata-cert.pem
```

- In `$IDP_HOME/conf/relying-party.xml`
  - Add the following snippet into the `ChainingMetadataProvider`:

```
        <!-- Tuakiri -->
        <metadata:MetadataProvider id="Tuakiri" xsi:type="metadata:
ResourceBackedMetadataProvider">
          <metadata:MetadataFilter xsi:type="metadata:ChainingFilter" xmlns="urn:mace:
shibboleth:2.0:metadata">
            <metadata:MetadataFilter xsi:type="metadata:SignatureValidation" xmlns="urn:mace:
shibboleth:2.0:metadata"
                                trustEngineRef="shibboleth.MetadataTrustEngine"
                                requireSignedMetadata="true" />
          </metadata:MetadataFilter>
          <metadata:MetadataResource xsi:type="resource:FileBackedHttpResource"
                              url="https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-
signed.xml"
                              file="/opt/shibboleth-idp/metadata/tuakiri-metadata.xml" />
        </metadata:MetadataProvider>
```

  - And add the following snippet into the `<security:TrustEngine id="shibboleth.MetadataTrustEngine" xsi:type="security:StaticExplicitKeySignature">` element:

```
       <security:Credential id="Tuakiri-FederationCredentials" xsi:type="security:
X509Filesystem">
            <security:Certificate>/opt/shibboleth-idp/credentials/tuakiri-metadata-cert.pem<
/security:Certificate>
       </security:Credential>
```

⚠ Remember to uncomment the `<security:TrustEngine id="shibboleth.MetadataTrustEngine" xsi:type="security:StaticExplicitKeySignature">` element if it is still commented out (it is commented out in the default configuration).

Configure attribute release/filtering through the federation

To configure a 3.x IdP to Load the Tuakiri-managed attribute filter:

- Contact the federation administrators (by emailing tuakiri@reannz.co.nz) and request a URL for the Attribute Filter for your IdP.
  - In the request, please include:
    - The name (hostname or entityID) of your IdP
    - An email address that should receive notifications whenever the attribute filter changes (these are notifications only, no action will be required).
  - The attribute filter may have to be manually added to the list of attribute filters published. Once created, the URL will have the form of: **`https://directory.tuakiri.ac.nz/attribute-filter/<institution-domain>.xml`**
- Edit `$IDP_HOME/conf/services.xml` and add the additional attribute filter as an additional resource in the `shibboleth.AttributeFilterResources util:list` bean, using the built-in FileBackedHTTPResource:

```
       <bean id="TuakiriAttributeFilterResource" class="net.shibboleth.ext.spring.resource.
FileBackedHTTPResource"
             c:client-ref="shibboleth.MemoryCachingHttpClient"
             c:url="https://directory.tuakiri.ac.nz/attribute-filter/institution.domain.ac.nz.xml"
             c:backingFile="%{idp.home}/conf/tuakiri-attribute-filter.xml"/>
```

- For Tuakiri-TEST, the configuration would be the same, just the URL would be different - please use the URL provided by the federation administrators.

For archival purposes, we also keep the original instructions for configuring the Tuakiri-managed attribute filter into a 2.x IdP - unfold the box below to see the IdP 2.x compatible syntax:

After requesting the attribute filter:

- Add the following entry into `<srv:Service id="shibboleth.AttributeFilterEngine"` in `$IDP_HOME/conf/service.xml`(note that the URL varies for each IdP and has to be obtained from the federation administrators):

```
       <srv:ConfigurationResource xsi:type="resource:FileBackedHttpResource"
                            url="https://directory.tuakiri.ac.nz/attribute-filter/<institution-
domain>.xml"
                            file="/opt/shibboleth-idp/conf/tuakiri-attribute-filter.xml" />
```

⚠ Note: if your `$IDP_HOME` is different from `/opt/shibboleth-idp`, change the file path in the above snippet accordingly.

If configuring this in Shibboleth IdP 2.1.x, do not use the srv: namespace prefix - i.e., use just:

```
        <ConfigurationResource xsi:type="resource:FileBackedHttpResource"
                        url="https://directory.tuakiri.ac.nz/attribute-filter/<institution-
domain>.xml"
                        file="/opt/shibboleth-idp/conf/tuakiri-attribute-filter.xml" />
```

- We also strongly recommend you configure your IdP to periodically reload this file - we recommend at 2 hour intervals. This is documented in detail in the IdP Install Manual: Reloading configuration section and Load AAF Atribute Filter sections. The simple step is to add the `configurationResourcePollingFrequency="PT2H0M0.000S"` and `configurationResourcePollingRetryAttempts="10"` attributes to the `<srv:Service id="shibboleth.AttributeFilterEngine"`element. If you already have these attributes set for reloading the local configuration file - with a shorter interval, please adjust them accordingly to 2 hours for the remotely loaded attribute filter:

```
    <srv:Service id="shibboleth.AttributeFilterEngine"
+             configurationResourcePollingFrequency="PT2H0M0.000S"
configurationResourcePollingRetryAttempts="10"
             xsi:type="attribute-afp:ShibbolethAttributeFilteringEngine">
```

Now your IdP should be able to access service providers within the Tuakiri federation.

Appendix A - Alternative implementation

Loading the metadata and the attribute filter files from a remote URL makes the IdP depend on the accessibility of the remote URL. While for metadata itself, the IdP software should be sufficiently resilient, for attribute filter configuration, this is not the case. Tuakiri will be running their servers serving these XML files according to best practices. However, some sites may prefer not to take on the risk and put the XML file loading outside of the IdP, into a separate process. This section describes this alternative implementation. This implementation first downloads the XML file into a temporary file on the local machine. Once this is completed it then replaces the original configuration file with the new one, and this will be detected by the IdP and will cause a reload of this file.

This implementation is based on using an external script (`fetch-xml.sh`). This script loads the XML file (over an HTTPS connection), checks the XML document for well-formedness, optionally verifies the signature on the downloaded XML document - and if all tests are passed, replaces the original file with a single "mv".

The script takes three arguments: the remote URL, the local file name, and an email address to send any errors to (no email sent if everything goes well).

An extra optional step (documented below) is to install XmlSecTool for verifying the signature. Otherwise, downloading the file over HTTPS and checking the XML structure provides also reasonable guarantees. If using XmlSecTool, the script takes a fourth argument, the certificate to check the signature with. And in this case, XmlSecTool must be found either in the `PATH` or in the `XMLSECTOOL` environment variable.

To deploy this solution without XmlSecTool:

- Download the fetch-xml.sh script into `/opt/shibboleth-idp/bin`

```
wget -O /opt/shibboleth-idp/bin/fetch-xml.sh https://github.com/REANNZ/Tuakiri-public/raw/master/scripts
/fetch-xml.sh
chmod +x /opt/shibboleth-idp/bin/fetch-xml.sh
```

- Determine the URLs you will be loading the files (metadata and attribute filter) from and locations you will be putting them into - same as in the standard implementation above.
- Download the metadata signing certificate into `$IDP_HOME/credentials`:

```
wget http://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem -O $IDP_HOME/credentials/tuakiri-
metadata-cert.pem
```

- Invoke `fetch-xml.sh` once to download the metadata:

```
/opt/shibboleth-idp/bin/fetch-xml.sh https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-signed.
xml /opt/shibboleth-idp/metadata/tuakiri-metadata-signed.xml errors@institution.domain.ac.nz
```

- Invoke `fetch-xml.sh` once to download the attribute filter for your IdP (note that you have to request one to be published, same as in the standard implementation above):

```
/opt/shibboleth-idp/bin/fetch-xml.sh http://directory.tuakiri.ac.nz/attribute-filter/institution.domain.
ac.nz.xml /opt/shibboleth-idp/conf/tuakiri-attribute-filter.xml errors@institution.domain.ac.nz
```

- Configure a 3.x IdP to load the Tuakiri metadata and atribute filter files:
  - Configure the IdP to load the Federation Metadata in `/opt/shibboleth-idp/conf/metadata-providers.xml` by adding the following snippet into the `ChainingMetadataProvider`.

```
    <MetadataProvider id="TuakiriMetadata"
                      xsi:type="FilesystemMetadataProvider"
                  refreshDelayFactor="0.125"
                  maxRefreshDelay="PT2H"
                  metadataFile="%{idp.home}/metadata/tuakiri-metadata.xml">

            <MetadataFilter xsi:type="SignatureValidation"
                    certificateFile="${idp.home}/credentials/tuakiri-metadata-cert.pem"
                    requireSignedRoot="false">
            </MetadataFilter>
            <MetadataFilter xsi:type="EntityRoleWhiteList">
                    <RetainedRole>md:SPSSODescriptor</RetainedRole>
            </MetadataFilter>

    </MetadataProvider>
```

    - Please see the notes in the main instructions for Configuring an IdP to load the Tuakiri metadata for additional information about the parameters in this snippet.

  - Edit `$IDP_HOME/conf/services.xml` and add the additional attribute filter as an additional value in the `shibboleth.AttributeFilterResources` `util:list` bean:

```
<value>%{idp.home}/conf/tuakiri-attribute-filter.xml</value>
```

- For archival purposes, we also keep the original instructions for configuring the Tuakiri metadata and attribute filter on a 2.x IdP - unfold the box below to see the IdP 2.x compatible syntax:

    - Load the metadata from the local file: add the following into `$IDP_HOME/conf/relying-party.xml` (the variation from the standard implementation above is using a `FilesystemResource` instead of a `FileBackedHttpResource`)
      - Add the following snippet into the `ChainingMetadataProvider`:

```
        <!-- Tuakiri -->
        <metadata:MetadataProvider id="Tuakiri" xsi:type="metadata:
ResourceBackedMetadataProvider">
          <metadata:MetadataFilter xsi:type="metadata:ChainingFilter" xmlns="urn:mace:
shibboleth:2.0:metadata">
            <metadata:MetadataFilter xsi:type="metadata:SignatureValidation" xmlns="urn:
mace:shibboleth:2.0:metadata"
                              trustEngineRef="shibboleth.MetadataTrustEngine"
                              requireSignedMetadata="true" />
          </metadata:MetadataFilter>
          <metadata:MetadataResource xsi:type="resource:FilesystemResource" file="/opt
/shibboleth-idp/metadata/tuakiri-metadata.xml" />
        </metadata:MetadataProvider>
```

- Same as in the standard implementation, uncomment the `<security:TrustEngine id="shibboleth.MetadataTrustEngine" xsi:type="security:StaticExplicitKeySignature">` element if it is still commented out and add in this snippet to load the metadata signing certificate

```
        <security:Credential id="Tuakiri-FederationCredentials" xsi:type="security:
X509Filesystem">
            <security:Certificate>/opt/shibboleth-idp/credentials/tuakiri-metadata-cert.
pem</security:Certificate>
        </security:Credential>
```

- Load the attribute filter from a local file: Add the following entry into `<srv:Service id="shibboleth.AttributeFilterEngine"` in `$IDP_HOME/conf/service.xml`:

```
        <srv:ConfigurationResource file="/opt/shibboleth-idp/conf/tuakiri-attribute-filter.
xml" xsi:type="resource:FilesystemResource" />
```

- Create a cron job to periodically (every 2 hours) download the metadata and the attribute filter: run `crontab -e` and add the following entry (matching the command you had run on the command line earlier):

```
02 */2 * * * /opt/shibboleth-idp/bin/fetch-xml.sh https://directory.tuakiri.ac.nz/metadata/tuakiri-
metadata-signed.xml /opt/shibboleth-idp/metadata/tuakiri-metadata.xml errors@institution.domain.ac.nz
02 */2 * * * /opt/shibboleth-idp/bin/fetch-xml.sh https://directory.tuakiri.ac.nz/attribute-filter
/institution.domain.ac.nz.xml /opt/shibboleth-idp/conf/tuakiri-attribute-filter.xml errors@institution.
domain.ac.nz
```

Optional: Installing XmlSecTool

- Download latest version (2.0.0 as of July 2016) from http://www.shibboleth.net/downloads/tools/xmlsectool/ into `~/inst`
  - Unzip into `/opt/xmlsectool-$XMLSECTOOL_VERSION`
  - Symlink as `/opt/xmlsectool`

```
export XMLSECTOOL_VERSION="2.0.0"
wget -P ~/inst/ http://www.shibboleth.net/downloads/tools/xmlsectool/$XMLSECTOOL_VERSION
/xmlsectool-$XMLSECTOOL_VERSION-bin.zip
cd /opt
unzip ~/inst/xmlsectool-$XMLSECTOOL_VERSION-bin.zip
ln -s xmlsectool-$XMLSECTOOL_VERSION xmlsectool
```

- Set JAVA_HOME to your Java installation:

```
export JAVA_HOME=/usr/lib/jvm/java
```

- Invoke as **`/opt/xmlsectool/xmlsectool.sh`**

- Modify `fetch-xml.sh`cron jobs to use XmlSecTool to verify signature:
  - Add `/opt/shibboleth-idp/credentials/tuakiri-metadata-cert.pem` as an additional argument (the certificate to verify signatures with)
  - Prefix the commands with environment variable settings to tell the script where to find XmlSecTool and tell XmlSecTool where to find Java: `JAVA_HOME=/usr/lib/jvm/java XMLSECTOOL=/opt/xmlsectool/xmlsectool.sh`
  - The final form of the cron jobs is:

```
02 */2 * * * JAVA_HOME=/usr/lib/jvm/java XMLSECTOOL=/opt/xmlsectool/xmlsectool.sh /opt/shibboleth-
idp/bin/fetch-xml.sh https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-signed.xml /opt
/shibboleth-idp/metadata/tuakiri-metadata.xml errors@institution.domain.ac.nz /opt/shibboleth-idp
/credentials/tuakiri-metadata-cert.pem
02 */2 * * * JAVA_HOME=/usr/lib/jvm/java XMLSECTOOL=/opt/xmlsectool/xmlsectool.sh /opt/shibboleth-
idp/bin/fetch-xml.sh http://directory.tuakiri.ac.nz/attribute-filter/institution.domain.ac.nz.xml
/opt/shibboleth-idp/conf/tuakiri-attribute-filter.xml errors@institution.domain.ac.nz /opt
/shibboleth-idp/credentials/tuakiri-metadata-cert.pem
```