

Integrating authentication server with ForgeRock

On this page:

- [Integrating the authentication server with ForgeRock](#)
- [Deploying the authentication server](#)
- [Configuring SAMLv2 Remote Service Provider](#)

This page contains the instructions to integrate the authentication server with the ForgeRock application (ForgeRock is the company that develops open-source identity and access management products for cloud, social, mobile, and enterprise environments). This documentation is based on ForgeRock version 13.0.0.

Integrating the authentication server with ForgeRock

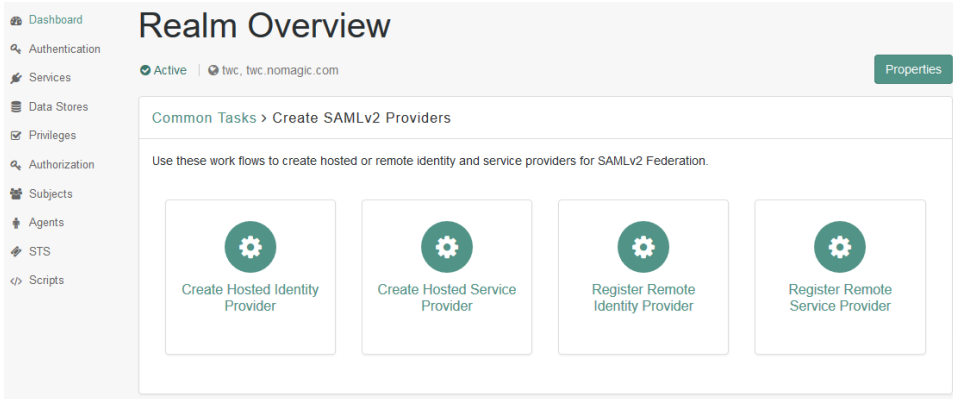
To integrate the authentication server with ForgeRock

1. Log in to ForgeRock as administrator.
2. Either select or create the realm, which will be used for integration. Make sure that the realm uses stateful sessions. The realm in this example is "twc".
3. In the realm overview select **Authentication > Settings**.
4. Click the **General** tab and clear the **Use Stateless Sessions** check box.
5. Click **Save Changes**.

The screenshot shows the 'Authentication Settings' page with the 'General' tab selected. The left sidebar contains a navigation menu with options like Dashboard, Authentication, Settings, Chains, Modules, Services, Data Stores, Privileges, Authorization, Subjects, Agents, STS, and Scripts. The main content area has tabs for Core, User Profile, Account Lockout, General (active), Security, and Post Authentication Processing. Under the General tab, there are several settings: 'Default Authentication Locale' set to 'en_US', 'Identity Types' with 'agent' and 'user' selected, 'Pluggable User Status Event Classes' (empty), 'Use Stateless Sessions' (unchecked), 'Two Factor Authentication Mandatory' (unchecked), and 'Default Authentication Level' set to '0'. At the bottom right are 'Revert' and 'Save Changes' buttons.

6. Go back to **Dashboard** and in the realm overview, select **Create SAMLv2 Providers** and then **Create Hosted Identity Provider**.

The screenshot shows the 'Realm Overview' page for the realm 'twc'. The left sidebar is the same as in the previous screenshot. The main content area shows the realm is 'Active' and provides a 'Properties' button. Below this is a 'Common Tasks' section with eight task cards, each with a gear icon: 'Create SAMLv2 Providers', 'Configure OAuth Provider', 'Create Fedlet', 'Configure Google Apps', 'Configure Salesforce CRM', 'Configure Social Authentication', 'Get Product Documentation', and 'Create a Soap STS Deployment'.



7. Fill in the Identity Provider data by selecting the realm, the signing key, and entering a new circle of trust's name.

Create a SAMLv2 Identity Provider on this Server

This page allows you to configure this instance of OpenAM server as an Identity Provider (IDP). You can provide a Name for the provider, Circle of Trust (COT), its metadata of the provider and optionally Signing Certificate. A COT is a group of IDPs and Service Providers (SPs) that trust each other and in effect represents the confines within which all federation communications are performed. Metadata represents the configuration necessary to execute federation protocols (eg SAMLv2) as well as the mechanism to communicate this configuration to other entities (eg SPs) in a COT. We shall generate the metadata if you do not have one. You are required to pick a realm for this provider if there are more than one realm in the system. Otherwise, this provider will be configured under the root realm.

Do you have metadata for this provider?: ☐ Yes ☒ No

metadata

* Realm: ?

* Name: ?

Signing Key: ?

Circle of Trust

Choose from existing circles of trust listed or provide one to be created in which to include this IDP. A COT is a group of IDPs and SPs that trust each other and provides the confines within which all SAMLv2 communications are performed.

* New Circle of Trust:

Attribute Mapping

Mapping attributes helps to ensure that both the Service Provider (SP) and the Identity Provider (IDP) can recognize the same attributes that may have unique names. For example, the SP may have an attribute called UserName but the IDP may call it UserID. Eliminating these inconsistencies by mapping the attributes will guarantee that the data will be passed correctly.

| Name in Assertion | Local Attribute Name |
|----------------------|----------------------|
| <input type="text"/> | <input type="text"/> |

Select an attribute:

8. Click **Configure** to save the Identity Provider.
9. Return to the main page.
10. Select **FEDERATION** menu item and select to modify the newly created Identity Provider in the **Entity Providers** table in the **Circle of Trust Configuration** section.
11. Remove the value "urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName=" in **NameID Value Map** in the **Assertion Content** tab, if any.
12. Add the value "urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName=uid" to the map. By default, this nameID is used in the Authentication Server for user identification (`authserver.properties` parameter `authentication.saml.name.id.format`).

NameID Value Map

Current Values

- urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress=mail
- urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName=
- urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName=
- urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos=
- urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified=
- urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName=uid

New Value

13. Save the configuration changes.

Deploying the authentication server

To deploy the authentication server

1. During the deployment process add the SAML integration configuration parameters to the file `authserver.properties`. The full list of SAML integration parameters is listed on the page [Configuration Parameters, SAML Integration](#) section.
 - Set `authentication.saml.enabled` to `true`.
 - If the authentication server is deployed on a cluster, set value for the parameter `authentication.saml.entity.id` (default is `com.nomagic.authentication.server`). This value should be unique for each authentication server instance.

- Set **authentication.saml.idp.metadata.url** to the appropriate ForgeRock server address (should be <http://<server>:<port>/openam/saml2/jsp/exportmetadata.jsp?entityid=http://<server>:<port>/openam&realm=/<realm name>>).
- Set the name of the button to the parameter **authentication.saml.link**. Users will click this button to log in using ForgeRock as the Identity Provider.

2. Save the **authserver.properties** file and register/start the Authentication Server.

Configuring SAMLv2 Remote Service Provider

To configure SAMLv2 Remote Service Provider

1. Go back to ForgeRock and, in the realm overview, select **Create SAMLv2 Providers** and then **Register Remote Service Provider**.
2. Select the realm.
3. Specify the URL of Authentication Server metadata: [http\[s\]://<auth-server-host>:<auth-server-port>/authentication/saml/metadata](http[s]://<auth-server-host>:<auth-server-port>/authentication/saml/metadata) (if such URL is accessible from ForgeRock) or select a file of stored Authentication Server metadata (usually used in **https** case).
4. Enter or select the same circle of trust as that of the Identity Provider and click **Configure**.

Create a SAMLv2 Remote Service Provider

This page allows you to register a remote Service Provider (SP). You need two things: Circle of Trust (COT) and metadata of the provider. A COT is a group of Identity Providers (IDPs) and SPs that trust each other and in effect represents the confines within which all federation communications are performed. Metadata represents the configuration necessary to execute federation protocols (eg SAMLv2) as well as the mechanism to communicate this configuration to other entities (eg IDPs) in a COT.

* Realms: ⓘ

Where does the metadata file reside?: ☒ URL ☐ File ⓘ

* URL where metadata is located: ⓘ

Circle of Trust

Choose from existing circles of trust listed or provide one to be created in which to include this SP. A COT is a group of IDPs and SPs that trust each other and provides the confines within which all SAMLv2 communications are performed.

* New Circle of Trust:

Attribute Mapping

| Attributes Mapping | |
|--|---|
| <input type="button" value="Delete"/> | |
| <input type="text" value="Name in Assertion"/> | <input type="text" value="Local Attribute Name"/> |
| <input type="text"/> | <input type="text"/> |
| <input type="button" value="Add"/> | |

ⓘ

5. A Service Provider with the default name **com.nomagic.authentication.server** will be created. The name of the service provider is configured in the Authentication Server's **authserver.properties** file (parameter **authentication.saml.entity.id**).

Once you have completed the steps on this page, you should be able to log in through ForgeRock by clicking the SAML integration button on the **Authentication Server** login page. If later the Authentication Server configuration, related to SAML or server keystore file is changed, delete the remote service provider and add a new one (see [Create SAMLv2 Providers > Register Remote Service Provider](#)).