SAML Authentication

On this page:

Related pages:

- Configure SAML for Microsoft Active Directory Federation Services
- Configure SAML for Microsoft Active Directory on Azure
- Configure SAML for Okta
- Configure SAML for OneLogin

The AppDynamics Controller can use an external SAML (Security Assertion Markup Language) identity provider to authenticate and authorize users. This topic describes how to set up and administer SAML authentication.

How SAML Authentication Works

With SAML authentication enabled, the Controller UI redirects credentials entered in the log in page to the external SAML identity provider. To be able to log in to the Controller UI, the user needs to be able to access both the Controller and the identity provider service by network from their computer.

User privileges in the Controller UI are governed by roles (see Roles and Permissions for more information on roles). You can configure the Controller to assign roles to authenticated users based on group attributes in their SAML responses. See About Roles and SAML Groups for more information on mapping SAML attribute to roles.

When SAML authentication is enabled, users can authenticate with local, AppDynamics credentials by clicking the **Use Local Login** link. The link appears at the bottom of the Login page. Local authentication is useful if you haven't mapped a particular role to SAML attributes, such as AppDynamics administrators, or if you need to disable SAML authentication.

Sample SAML Request

The SAML request that the external identity provider receives from the Controller looks something like the following:

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="_16df-3c17-4a60-9158-
b7b" Version="2.0" IssueInstant="2016-04-08T18:58:09.42Z" ProtocolBinding="urn:oasis:names:tc:
SAML:2.0:bindings:HTTP-POST" AssertionConsumerServiceURL="http://appd.example.com/controller/saml-
auth?accountName=customer2″>
   <saml:Issuer>http://appd.example.com/controller</saml:Issuer>
   <samlp:NameIDPolicy Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
AllowCreate="true" />
</samlp:AuthnRequest>
```

About Roles and SAML Groups

The Controller can assign roles to SAML-authenticated users using one of the following mechanisms:

- SAML group attributes: You can map SAML group membership attributes to roles in AppDynamics. Using this method, each time the user authenticates, the Controller checks the SAML assertion and updates the role assignment if needed.
- Internal AppDynamics account roles: If a SAML-authenticated user has the same username as an AppDynamics internal user account and the SAML assertion does not contain mapped SAML group attributes, the Controller gives the user the roles for the internal AppDynamics account.
- Default role: If there are no SAML group attributes in a user's identity assertion, the authenticated user is assigned the SAML default role upon first log in. An AppDynamics administrator can verify and adjust the roles for users manually in AppDynamics once the accounts are created for those users. Manually adjusted roles are preserved across subsequent log-ins.

To use SAML group attributes as the basis for AppDynamics role assignments, configure the SAML group attribute value mapping. If using internal account role associations, you can simply enable SAML authentication and configure basic SAML authentication settings.

The default role is not associated with any AppDynamics roles out-of-the-box, so you need to configure the default role to use it.

Enabling SAML Authentication

The following steps assume that you have an account with a supported identity provider. You need to know the SAML Login URL and have the x.509 certificate supplied by your identity provider. You should also be familiar with the format of the SAML identify response from your SAML provider.

Enable SAML authentication as follows:

1. As a user with AppDynamics account administrator privileges in the Controller UI, go to **Settings > Administration**.
2. Click on the **Authentication Provider** tab and select **SAML** as the authentication provider**.**
3. Enter the following SAML Configuration settings:
    - **Login URL**: The SAML Login URL where the Controller will route Service Provider (SP)-initiated login requests. This is required.
    - **Logout URL**: The URL where the Controller will redirect users after they log out. If you do not specify a logout URL, users will get the AppDynamics login screen when they log out.
    - **Certificate**: The x.509 certificate from your identity provider configuration. Paste the certificate between the BEGIN CERTIFICATE and END CERTIFICATE delimiters. Avoid duplicating "BEGIN CERTIFICATE" and "END CERTIFICATE" delimiters from the source certificate itself.
4. In the SAML Attribute Mappings settings, specify how SAML-authenticated users are identified in the AppDynamics Controller:
    - **Username Attribute**: Unique identifier for the user in the SAML response. This value corresponds to the AppDynamics username field, so the value must be unique among all SAML users in the Controller account. Given the sample response below, the value for this setting would be `User.OpenIDName`.
    - **Display Name Attribute**: The informal name for the user corresponding to the AppDynamics Name field. Given the sample response, this value would be `User.fullName`.
    - **Email Attribute**: The user's email address, corresponding to AppDynamics email field. Given the sample response, this value would be `User.email`.
    - **Account Name**: If the Controller is in multi-tenant mode, the SAML response must contain a custom SAML attribute accountName that indicates the user's AppDynamics account name. You cannot change this field mapping in the Controller.

```
<saml:AttributeStatement>
   <saml:Attribute Name="User.OpenIDName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
xs:string">adynamo</saml:AttributeValue>
   </saml:Attribute>
   ...
   <saml:Attribute Name="User.email" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
xs:string">Ajay.Dynamo@example.com</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="User.fullName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
      <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="
xs:string">Mr. Ajay Dynamo</saml:AttributeValue>
   </saml:Attribute>
   ...
</saml:AttributeStatement>
```

5. To map SAML group attributes to AppDynamics roles, configure the **SAML Group Mappings** settings. The settings you use depends on the structure of the SAML group attribute in the response, as described in Map SAML Groups to AppDynamics Roles. If you are using internal AppDynamics accounts to map user roles, you can skip this step.
6. Optionally specify a master SAML Access Attribute included in the SAML response from your provider. When enabled, the Controller only grants access to users when the SAML assertion contains a matching value for the attribute. In the sample response below, this attribute value is `AccessControl`.

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" Name="
AccessControl">
   <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:
string">{access}</saml:AttributeValue>
</saml:Attribute>
```

7. Click **Save** to apply your changes. The Controller immediately starts using the SAML identity provider you configured for user authentication.

Configuring Default Permissions

Instead of mapping SAML attributes to roles, you can allow all users to get a default role with the permissions you specify.

To use default permissions, edit the **Default Permissions** settings in the **SAML Group Mappings** list.

In the Default Group Mapping dialog, choose the AppDynamics roles that all authenticated users get.

Mapping SAML Group to Roles

If the identity assertion from the SAML provider includes group attributes that correspond to AppDynamics roles, you can configure mappings between those attributes and roles. The SAML Group Mappings settings in the SAML configuration page control the mappings, as described here.

To configure SAML attribute to role mapping:

1. In the **SAML Group Attribute Name** field, enter the `Name` attribute value that identifies the SAML Attribute element with group affiliations for the user. For example, given the following response snippet, use `SAML groups-Membership` in the **SAML Group Attribute Name** field.

```
<saml:Attribute NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic" Name="Groups-Membership">
   <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">
      {group1};{group2}
   </saml:AttributeValue>
</saml:Attribute>
```

2. Use the **Group Attribute Value** and **Mapping of Group to Roles** settings to describe the structure of the SAML group attribute from which AppDynamics needs to extract the group value, and the roles associated with those values. The Controller can extract Group Attribute values based on the following options
   - Singular Group Values: The response contains an `AttributeValue` element with a single group-mapping value.
   - Multiple Nested Group Values: The response contains more than one `AttributeValue` element, each with a single group-mapping value.
   - Singular Delimited Group Value: The response contains a single `AttributeValue` element with multiple, delimiter-separated group-mapping values.
   - Regex on Singular Group Value: The response contains a single `AttributeValue` element from which you want to extract the group-mapping value with a regular expression.

   The sections below provide more information on and examples for each option.
3. With any options selected, select the **Value is in LDAP Format** checkbox if the value or values returned by the group attribute value is in LDAP format. For example: "OU=AppDynamics-Users". With this option enabled, only "AppDynamics-Users" is used to map to the SAML Group name.

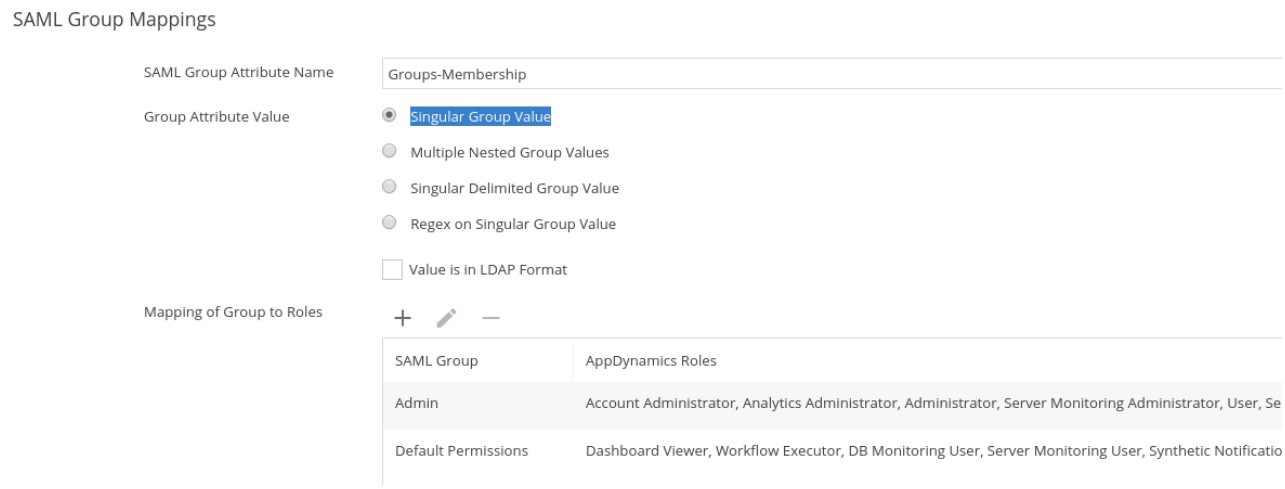
The following sections describe the SAML group attribute value mapping options.

Singular Group Values

Choose Singular Group Value if the SAML group attribute contains a single group, as in the following example.

```
<saml:AttributeStatement>
   <saml:Attribute Name="Groups-Membership" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
      <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string">Admin</saml:AttributeValue>
   </saml:Attribute>
</saml:AttributeStatement>
```

For this example, AppDynamics would extract the value Admin and associate the user with a SAML Group with the same name. In the following sample configuration, the user would get the roles configured assigned to the Admin SAML group in the example in the following figure—Account Administrator, Analytics Administrator, and so on:

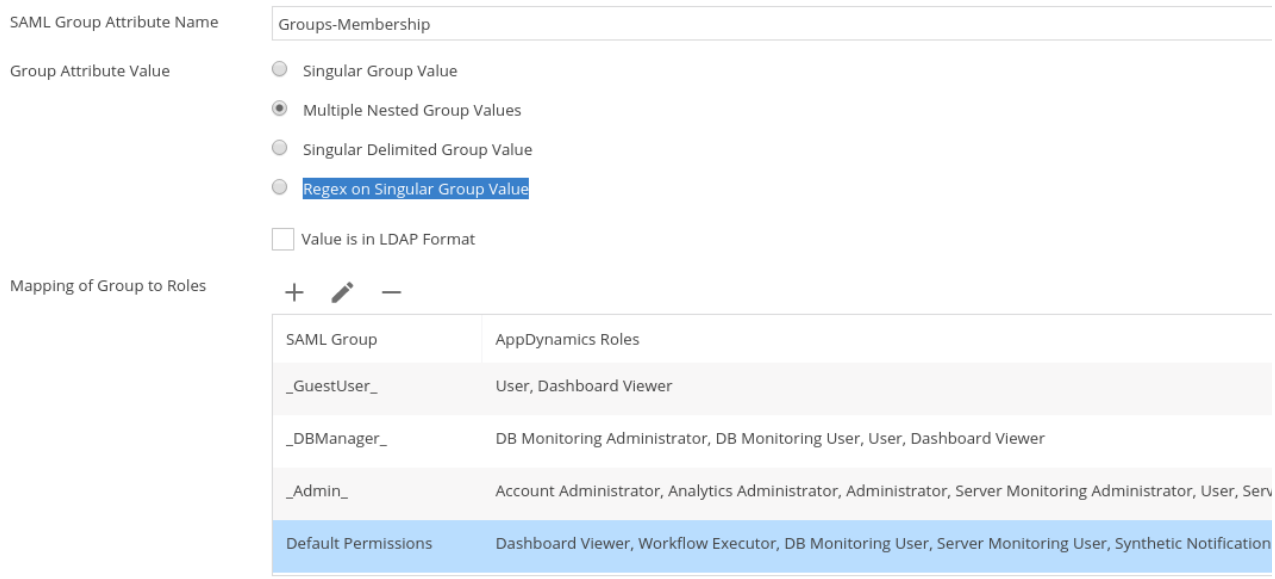


Multiple Nested Group Values

With this option selected, AppDynamics expects multiple `AttributeValue` child elements under the SAML Attribute with the group information, as in the following example:

```
<saml:Attribute Name="Groups-Membership" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:
basic">
   <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string"
>_Admin_</saml:AttributeValue>
   <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string"
>_DBManager_</saml:AttributeValue>
</saml:Attribute>
```

AppDynamics would extract _Admin_ and _DBManager_ from the example. Given the following sample configuration, the user with the previous response would receive the roles from the _Admin_ and _DBManager_ groups.



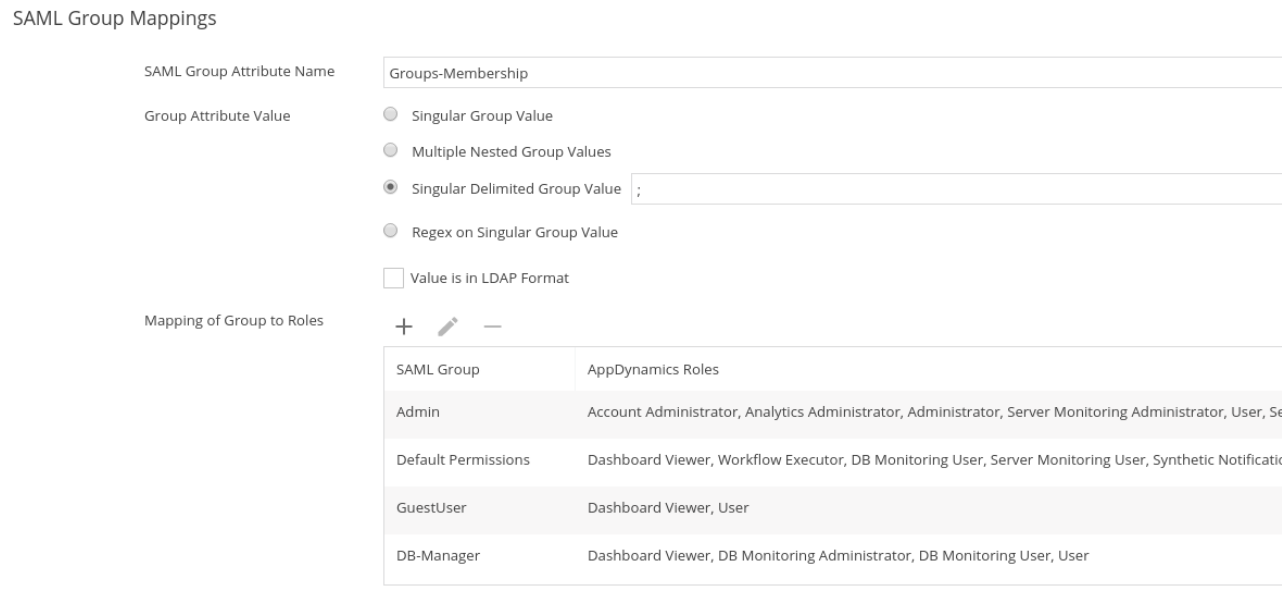
Singular Delimited Group Value

With this option selected, AppDynamics expects a single `AttributeValue` element with multiple, delimiter-separated values, as in the following example:

```
<saml:Attribute Name="Groups-Membership" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:
basic">
   <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string"
>Admin;DB-Manager</saml:AttributeValue>
</saml:Attribute>
```

Specify the delimiter that separate the values to extract, such as a semi-colon in the example.

Given the following sample configuration, the user would get the AppDynamics roles associated with both the Admin and DB-Manager groups—Dashboard Viewer, User, DB Monitoring Administrator, and so on.

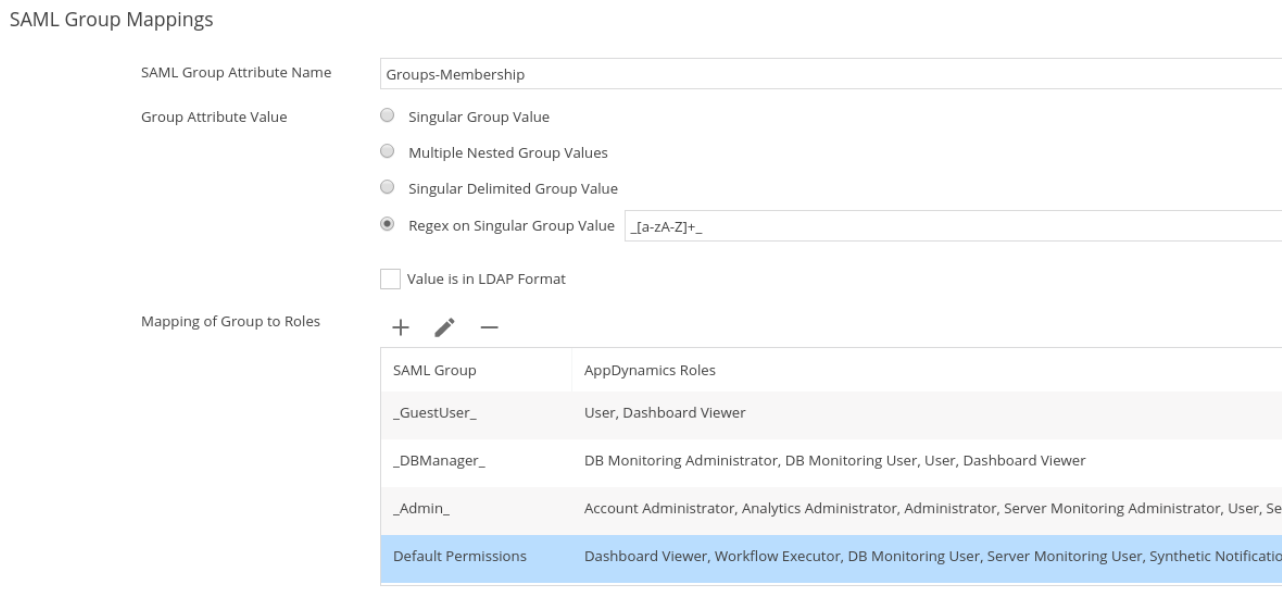


Regular Expression on Singular Group Value

Choose this option to have AppDynamics extract group mapping values using a regular expression. Regular expressions enable you to pull group values from unstructured contexts, such as from within a larger string, as in the following response example:

```
<saml:AttributeStatement>
   <saml:Attribute Name="Groups-Membership" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:basic">
   <saml:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="xs:string"
>User memberships in _Admin_ and _DBManager_ groups.</saml:AttributeValue>
   </saml:Attribute>
</saml:AttributeStatement>
```

In the example, the group names _Admin_ and _DBManager_ are embedded in the `AttributeValue` string. To extract those names, you can use a regular expression such as `_[a-zA-Z]_`. Like other types of group attribute sources, AppDynamics assigns all roles associated with both the _Admin_ and _DBManager_ SAML Groups, as follows:



Disabling SAML Authentication

To disable SAML authentication, log in as an administrator using the local login link on the log in page and restore the default authentication mode, AppDynamics authentication.