Red Hat Ansible Automation Platform 2.5

Access management and authentication

Configure role based access control, authenticators and authenticator maps in Ansible Automation Platform

Last Updated: 2024-10-03

Red Hat Ansible Automation Platform 2.5 Access management and authentication

Configure role based access control, authenticators and authenticator maps in Ansible Automation Platform

Legal Notice

Abstract

This guide provides requirements, options, and recommendations for controlling access to Red Hat Ansible Automation Platform resources.

Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

If you have a suggestion to improve this documentation, or find an error, you can contact technical support at https://access.redhat.com to open a request.

CHAPTER 1. OVERVIEW OF ACCESS MANAGEMENT AND AUTHENTICATION

Ansible Automation Platform features a platform interface where you can set up centralized authentication, configure access management, and configure global and system level settings from a single location.

The first time you log in to the Ansible Automation Platform, you must enter your subscription information to activate the platform. For more information about licensing and subscriptions, refer to Managing Ansible Automation Platform licensing, updates and support .

A system administrator can configure access, permissions and system settings through the following tasks:

- Configuring authentication in the Ansible Automation Platform , where you set up simplified login for users by selecting from several authentication methods available and define permissions and assign them to users with authenticator maps.
- Configuring access to external applications with token-based authentication , where you can configure authentication of third-party tools and services with the platform through integrated OAuth 2 token support.
- Managing access with role based access control , where you configure user access based on their role within a platform organization.
- Configuring Ansible Automation Platform, where you can configure global and system level settings for the platform and services.

CHAPTER 2. MANAGING ANSIBLE AUTOMATION PLATFORM LICENSING, UPDATES AND SUPPORT

Ansible is an open source software project and is licensed under the GNU General Public License version 3, as described in the Ansible Source Code .

You must have valid subscriptions attached before installing Ansible Automation Platform.

For more information, see Attaching Subscriptions.

2.1. TRIAL AND EVALUATION

A license is required to run Ansible Automation Platform. You can start by using a free trial license.

- Trial licenses for Ansible Automation Platform are available at the Red Hat product trial center .
- Support is not included in a trial license or during an evaluation of the Ansible Automation Platform.

2.2. COMPONENT LICENSES

To view the license information for the components included in Ansible Automation Platform, see **/usr/share/doc/automation-controller-<version>/README**.

where **<version>** refers to the version of automation controller you have installed.

To view a specific license, see **/usr/share/doc/automation-controller-<version>/*.txt**.

where ***** is the license file name to which you are referring.

2.3. NODE COUNTING IN LICENSES

The Ansible Automation Platform license defines the number of Managed Nodes that can be managed as part of your subscription.

A typical license says "License Count: 500", which sets the maximum number of Managed Nodes at 500.

For more information about managed node requirements for licensing, see How are "managed nodes" defined as part of the Red Hat Ansible Automation Platform offering.

NOTE

Ansible does not recycle node counts or reset automated hosts.

2.4. SUBSCRIPTION TYPES

Red Hat Ansible Automation Platform is provided at various levels of support and number of machines as an annual subscription.

- **Standard:**
  - Manage any size environment

  - Enterprise 8x5 support and SLA
  - Maintenance and upgrades included
  - Review the SLA at Product Support Terms of Service
  - Review the Red Hat Support Severity Level Definitions

- **Premium:**
  - Manage any size environment, including mission-critical environments
  - Premium 24x7 support and SLA
  - Maintenance and upgrades included
  - Review the SLA at Product Support Terms of Service
  - Review the Red Hat Support Severity Level Definitions

All subscription levels include regular updates and releases of automation controller, Ansible, and any other components of the Ansible Automation Platform.

For more information, contact Ansible through the Red Hat Customer Portal or at the Ansible site.

2.5. ATTACHING YOUR RED HAT ANSIBLE AUTOMATION PLATFORM SUBSCRIPTION

You **must** have valid subscriptions attached on all nodes before installing Red Hat Ansible Automation Platform. Attaching your Ansible Automation Platform subscription provides access to subscription-only resources necessary to proceed with the installation.

Procedure

1. Make sure your system is registered:

   ```
   $ sudo subscription-manager register --username <$INSERT_USERNAME_HERE> --password <$INSERT_PASSWORD_HERE>
   ```

2. Obtain the **pool_id** for your Red Hat Ansible Automation Platform subscription:

   ```
   $ sudo subscription-manager list --available --all | grep "Ansible Automation Platform" -B 3 -A 6
   ```

   > **NOTE**
   >
   > Do not use MCT4022 as a **pool_id** for your subscription because it can cause Ansible Automation Platform subscription attachment to fail.

   Example

   An example output of the **subsciption-manager list** command. Obtain the **pool_id** as seen in the **Pool ID:** section:

```
Subscription Name: Red Hat Ansible Automation, Premium (5000 Managed Nodes)
  Provides: Red Hat Ansible Engine
  Red Hat Ansible Automation Platform
  SKU: MCT3695
  Contract: ````
  Pool ID: <pool_id>
  Provides Management: No
  Available: 4999
  Suggested: 1
```

3. Attach the subscription:

```
$ sudo subscription-manager attach --pool=<pool_id>
```

   You have now attached your Red Hat Ansible Automation Platform subscriptions to all nodes.

4. To remove this subscription, enter the following command:

```
$ sudo subscription-manager remove --pool=<pool_id>
```

Verification

- Verify the subscription was successfully attached:

```
$ sudo subscription-manager list --consumed
```

Troubleshooting

- If you are unable to locate certain packages that came bundled with the Ansible Automation Platform installer, or if you are seeing a ***Repositories disabled by configuration*** message, try enabling the repository by using the command:
  Red Hat Ansible Automation Platform 2.5 for RHEL 8

```
$ sudo subscription-manager repos --enable ansible-automation-platform-2.5-for-rhel-8-
x86_64-rpms
```

  Red Hat Ansible Automation Platform 2.5 for RHEL 9

```
$ sudo subscription-manager repos --enable ansible-automation-platform-2.5-for-rhel-9-
x86_64-rpms
```

2.6. OBTAINING A MANIFEST FILE

You can obtain a subscription manifest in the Subscription Allocations section of Red Hat Subscription Management. After you obtain a subscription allocation, you can download its manifest file and upload it to activate Ansible Automation Platform.

To begin, login to the Red Hat Customer Portal using your administrator user account and follow the procedures in this section.

2.6.1. Create a subscription allocation

Creating a new subscription allocation allows you to set aside subscriptions and entitlements for a system that is currently offline or air-gapped. This is necessary before you can download its manifest and upload it to Ansible Automation Platform.

Procedure

1. From the Subscription Allocations page, click **New Subscription Allocation**.
2. Enter a name for the allocation so that you can find it later.
3. Select **Type: Satellite 6.8** as the management application.
4. Click **Create**.

Next steps

- Add the subscriptions needed for Ansible Automation Platform to run properly.

2.6.2. Adding subscriptions to a subscription allocation

Once an allocation is created, you can add the subscriptions you need for Ansible Automation Platform to run properly. This step is necessary before you can download the manifest and add it to Ansible Automation Platform.

Procedure

1. From the Subscription Allocations page, click on the name of the **Subscription Allocation** to which you would like to add a subscription.
2. Click the **Subscriptions** tab.
3. Click **Add Subscriptions**.
4. Enter the number of Ansible Automation Platform Entitlement(s) you plan to add.
5. Click **Submit**.

Next steps

- Download the manifest file from Red Hat Subscription Management.

2.6.3. Downloading a manifest file

After an allocation is created and has the appropriate subscriptions on it, you can download the manifest from Red Hat Subscription Management.

Procedure

1. From the Subscription Allocations page, click on the name of the **Subscription Allocation** to which you would like to generate a manifest.
2. Click the **Subscriptions** tab.
3. Click **Export Manifest** to download the manifest file.
   This downloads a file *manifest*<allocation name>_<date>.zip_ to your default downloads folder.

Next steps

- Upload the manifest file to activate Red Hat Ansible Automation Platform.

2.7. ACTIVATING RED HAT ANSIBLE AUTOMATION PLATFORM

Red Hat Ansible Automation Platform uses available subscriptions or a subscription manifest to authorize the use of Ansible Automation Platform. To obtain a subscription, you can do either of the following:

1. Use your Red Hat customer or Satellite credentials when you launch Ansible Automation Platform.
2. Upload a subscriptions manifest file either using the Red Hat Ansible Automation Platform interface or manually in an Ansible playbook.

2.7.1. Activate with credentials

When Ansible Automation Platform launches for the first time, the Ansible Automation Platform Subscription screen automatically displays. You can use your Red Hat credentials to retrieve and import your subscription directly into Ansible Automation Platform.

> **NOTE**
>
> You are opted in for Automation Analytics by default when you activate the platform on first time log in. This helps Red Hat improve the product by delivering you a much better user experience. You can opt out, after activating Ansible Automation Platform, by doing the following:
>
> 1. From the navigation panel, select **Settings** → **System**.
> 2. Clear the **Gather data for Automation Analytics**option.
> 3. Click **Save**.

Procedure

1. Log in to Red Hat Ansible Automation Platform.
2. Select **Username / password**
3. Enter your Red Hat username and password.
4. Select your subscription from the **Subscription** list.

   > **NOTE**
   >
   > You can also use your Satellite username and password if your cluster nodes are registered to Satellite through Subscription Manager.

5. Review the End User License Agreement and select **I agree to the End User License Agreement**.
6. Click **Finish**.

Verification

After your subscription has been accepted, subscription details are displayed. A status of *Compliant* indicates your subscription is in compliance with the number of hosts you have automated within your subscription count. Otherwise, your status will show as *Out of Compliance*, indicating you have exceeded the number of hosts in your subscription. Other important information displayed include the following: Hosts automated:: Host count automated by the job, which consumes the license count Hosts imported:: Host count considering all inventory sources (does not impact hosts remaining) Hosts remaining:: Total host count minus hosts automated

2.7.2. Activate with a manifest file

If you have a subscriptions manifest, you can upload the manifest file either by using the Red Hat Ansible Automation Platform interface.

NOTE

You are opted in for Automation Analytics by default when you activate the platform on first time log in. This helps Red Hat improve the product by delivering you a much better user experience. You can opt out, after activating Ansible Automation Platform, by doing the following:

1. From the navigation panel, select **Settings** → **System**.
2. Uncheck the **Gather data for Automation Analytics**option.
3. Click **Save**.

Prerequisites

You must have a Red Hat Subscription Manifest file exported from the Red Hat Customer Portal. For more information, see Obtaining a manifest file .

Procedure

1. Log in to Red Hat Ansible Automation Platform.
2. If you are not immediately prompted for a manifest file, go to **Settings** → **Subscription**.
3. Select **Subscription manifest**
4. Click **Browse** and select the manifest file.
5. Review the End User License Agreement and select **I agree to the End User License Agreement**.
6. Click **Finish**.

NOTE

If the **BROWSE** button is disabled on the License page, clear the **USERNAME** and **PASSWORD** fields.

Verification

After your subscription has been accepted, subscription details are displayed. A status of *Compliant* indicates your subscription is in compliance with the number of hosts you have automated within your subscription count. Otherwise, your status will show as *Out of Compliance*, indicating you have exceeded the number of hosts in your subscription. Other important information displayed include the following: Hosts automated:: Host count automated by the job, which consumes the license count Hosts imported:: Host count considering all inventory sources (does not impact hosts remaining) Hosts remaining:: Total host count minus hosts automated

Next steps

- You can return to the license screen by selecting **Settings → Subscription** from the navigation panel and clicking **Edit subscription**.

CHAPTER 3. CONFIGURING AUTHENTICATION IN THE ANSIBLE AUTOMATION PLATFORM

Using the authentication settings in Ansible Automation Platform, you can set up a simplified login through several authentication methods, such as LDAP and SAML. Depending on the authentication method you select, you will be required to enter different information to complete the configuration. Be sure to include all the information required for your configuration needs.

3.1. PREREQUISITES

- A running installation of Ansible Automation Platform 2.5
- A running instance of your authentication source
- Administrator rights to the Ansible Automation Platform
- Any connection information needed to connect Ansible Automation Platform 2.5 to your source (see individual authentication types for details).

3.2. PLUGGABLE AUTHENTICATION

Authentication is the process of verifying a user’s identity to the Ansible Automation Platform (that is, to establish that a user is who they say they are). This can be done in a number of ways but would traditionally be associated with a **username** and **password**.

Ansible Automation Platform 2.5 uses a pluggable authentication system with a configuration wizard that provides a common, simplified method of configuring different types of authenticators such as LDAP and SAML. The pluggable system also allows you to configure multiple authenticators of the same type.

In the pluggable system we have a couple of concepts:

Authenticator Plugin

A plugin allows Ansible Automation Platform to connect to a source system, such as, LDAP or SAML. Ansible Automation Platform includes a variety of authenticator plugins. Authenticator plugins are similar to Ansible collections, in that all of the required code is in a package and can be versioned independently if needed.

Authenticator

An authenticator is an instantiation of an authenticator plugin and allows users from the specified source to log in. For example, the LDAP authenticator plugin defines a required LDAP server setting. When you instantiate an authenticator from the LDAP authentication plugin, you must provide the authenticator the LDAP server URL it needs to connect to.

Authenticator Map

Authenticator maps are applied to authenticators and tell Ansible Automation Platform what permissions to give a user logging into the system.

3.3. CREATING AN AUTHENTICATION METHOD

The **Create Authentication** wizard guides you through the steps to create a new authentication method for your organization. The wizard is launched during the create authentication process.

Creating an authenticator involves the following procedures:

1. Authentication type, where you select the type of authenticator plugin you want to configure.
2. Authentication details, where you configure the authentication details for the plugin you selected.
3. Mapping, where you define mapping rule types and triggers to control access to the system.
4. Mapping order, where you can define the mapping precedence.

> **NOTE**
>
> Mapping order is only available if you have defined one or more authenticator maps.

5. Review, where you can review and confirm the authentication settings before creating the authentication method.

3.3.1. Selecting an authentication type

On the first screen of the wizard you can select the type of authenticator plugin you want to configure.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
   The **Create Authentication** wizard is displayed, where you can follow the prompts to configure your preferred authentication method.
3. Select the authenticator type from the **Authentication type** list. See Configuring an authentication type for the complete list of authentication plugins available.
4. Click **Next** to Configure authentication details.

3.3.2. Configuring authentication details

Different authenticator plugins require different types of information. See the respective sections in Configuring an authentication type for the required details.

For all authentication types you can enter a **Name**, **Additional Authenticator Fields** and **Create Objects**.

Procedure

1. Enter a unique **Name** for the authenticator. The name is required, must be unique across all authenticators, and must not be longer than 512 characters. This becomes the unique identifier generated for the authenticator.

> NOTE
>
> Changing the name does not update the unique identifier of the authenticator. For example, if you create an authenticator with the name “My Authenticator” and later change it to “My LDAP Authenticator” you will not be able to create another authenticator with the name “My Authenticator” because the unique identifier is still in use.

2. Use the **Additional Authenticator Fields** to send arbitrary data back to the libraries behind the authenticators. This is an advanced feature and any values provided in this field are not validated.

   > NOTE
   >
   > Values defined in this field override the dedicated fields provided in the UI. For example, if you enter a URL in a dedicated field on this page and then add a URL entry into the Additional Authentication Fields, the URL defined in Additional Authentication Fields overrides the definition in the dedicated field.

3. Enable or disable **Enabled** to specify if the authenticator should be enabled or disabled. If enabled, users are able to login from the authenticator. If disabled, users will not be allowed to login from the authenticator.

4. Enable or disable **Create Object** to specify whether the authenticator should create teams and organizations in the system when a user logs in.

   Enabled

   Teams and organizations defined in the authenticator maps are created and the users added to them.

   Disabled

   Organizations and teams defined in the authenticator maps will not be created automatically in the system. However, if they already exist (i.e. created by a superuser), users who trigger the maps are granted access to them.

5. Enable or disable **Remove Users**. If enabled, any access previously granted to a user is removed when they authenticate from this source. If disabled, permissions are only added or removed from the user based on the results of this authenticator’s authenticator mappings.
   For example, assume a user has been granted the **is_superuser** permission in the system. And that user will log into an authenticator whose maps will not formulate an opinion as to whether or not the user should be a superuser. If **Remove Users** is enabled, the **is_superuser** permission will be removed from the user, the authenticator maps will not have an opinion as to whether it should be there or not so, after login the user will not have the **is_superuser** permission.

   If **Remove Users** is disabled, the **is_superuser** permission *will not* be removed from the user. The authenticator maps will not have an opinion as to whether it should be there or not so after login the user *will* have the **is_superuser** permission.

6. Click **Next** to Define authentication mapping rules and triggers.

3.3.3. Defining authentication mapping rules and triggers

Authentication map types can be used with any type of authenticator. Each map has a trigger that defines when the map should be evaluated as true.

Procedure

1. Click **Add authentication mapping** to see the list of available map types and select the map type you want to create. See Authenticator map types for detailed descriptions of the different map types. Choices include:
   - Allow
   - Organization
   - Team
   - Role
   - Is Superuser
2. Enter a unique rule **Name** to identify the rule.
3. Select a **Trigger** from the list. See Authenticator map triggers for more details. Choices include:
   - **Always**
   - **Never**
   - **Group**
   - **Attribute**
4. Repeat steps 1-3 to add additional triggers to the authenticator.
5. Click **Next** to optionally Adjust the Mapping order.

   NOTE

   The mapping order setting is only available if there is more than one authenticator map defined.

3.3.4. Adjusting the Mapping order

If you have one or more authenticator maps defined, you can manage the order of the maps. Authenticator maps are run in order when logging in lowest order to highest. If one authenticator map determines a user should be a member of a team but a subsequent map determines the user should not be a member of the same team the ruling form the second map will take precedence over the result of the first map. Authenticator maps with the same order are executed in an undefined order.

For example, if the first authenticator map is of type **is_superuser** and the trigger is set to **never**, any user logging into the system would never be granted the **is_superuser** flag.

And, if the second map is of type **is_superuser** and the trigger is based on the user having a specific group, any user logging in would initially be denied the **is_superuser** permission. However, any user with the specified group would subsequently be granted the **is_superuser** permission by the second rule.

Procedure

1. Adjust the mapping order by dragging and dropping the mappings up or down in the list using the draggable icon.

NOTE

The mapping precedence is determined by the order in which the mappings are listed.

2. After your authenticator maps are in the correct order, click **Next** to Review the authentication settings.

3.3.5. Reviewing the authentication settings

After you have defined the authentication details, configured the authentication maps, and specified the mapping order precedence, you can review and verify, or modify the settings before creating the authenticator.

Procedure

1. Review and verify the authentication settings.
2. Click **Finish** to create the authenticator.
   A notification is displayed if there are any issues with the authenticator or the map. If you encounter issues, click **Back** or select a wizard section from the wizard menu to go back and add missing data or correct inaccurate data.

3.4. CONFIGURING AN AUTHENTICATION TYPE

Ansible Automation Platform provides multiple authenticator plugins that you can configure to simplify the login experience for your organization. These are the authenticator plugins that are provided:

- Local
- LDAP
- SAML
- TACACS+
- Radius
- Azure
- Google OAuth
- Generic OIDC
- Keycloak
- GitHub
- GitHub organization
- GitHub team
- GitHub enterprise
- GitHub enterprise organization

- GitHub enterprise team

3.4.1. Configuring local authentication

As a platform administrator, you can configure local system authentication. With local authentication, users and their passwords are checked against local system accounts.

> **NOTE**
>
> A local authenticator is automatically created by the Ansible Automation Platform installation process, and is configured with the specified admin credentials in the inventory file before installation. After successful installation, you can log in to the Ansible Automation Platform using those credentials.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
3. Select **Local** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this Local configuration. The configuration name is required, must be unique across all authenticators, and must not be longer than 512 characters.
5. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   > **NOTE**
   >
   > Values defined in this field override the dedicated fields provided in the UI.

6. To automatically create organizations, users, and teams upon successful login, select **Create objects**.
7. To enable this authentication method upon creation, select **Enabled**.
8. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.
9. Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.2. Configuring LDAP authentication

As a platform administrator, you can configure LDAP as the source for account authentication information for Ansible Automation Platform users.

When LDAP is configured, an account is created for any user who logs in with an LDAP username and password and they can be automatically placed into organizations as either regular users or organization administrators.

Users created through an LDAP login should not change their username, first name, last name, or set a local password for themselves. Any changes made to this information is overwritten the next time the user logs in to the platform.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
3. Select **LDAP** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this LDAP configuration. The configuration name is required, must be unique across all authenticators, and must not be longer than 512 characters.
5. In the **LDAP Server URI** field, enter or modify the list of LDAP servers to which you want to connect. This field supports multiple addresses.
6. In the **LDAP Bind DN text** field, enter the Distinguished Name (DN) to specify the user that the Ansible Automation Platform uses to connect to the LDAP server. For example:

   ```
   CN=josie,CN=users,DC=website,DC=com
   ```

7. In the **LDAP Bind Password** text field, enter the password to use for the binding user.
8. Select a group type from the **LDAP Group Type** list.

   NOTE

   The LDAP group types that are supported by the Ansible Automation Platform use the underlying django-auth-ldap library.

9. To use **LDAP User DN Template** as an alternative to user search, enter the name of the template. This approach is more efficient for user lookups than searching if it is usable in your organizational environment. If this setting has a value it will be used instead of the **LDAP User Search** setting.
10. **LDAP Start TLS** is disabled by default. To enable TLS when the LDAP connection is not using SSL, set the switch to **On**.
11. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

    NOTE

    Values defined in this field override the dedicated fields provided in the UI.

12. Enter any **LDAP Connection Options** to set for the LDAP connection.
13. In the **LDAP Group Type Parameters** field, set the parameters required for the LDAP G

13. In the **LDAP Group Type Parameters** field, enter the parameters required for the LDAP Group Type you previously selected to identify LDAP groups and the members that belong to those groups.
    To determine the parameters that a specific **LDAP Group Type** requires, refer to the django_auth_ldap documentation on the classes **init** parameters.

14. In the **LDAP Group Search** field, specify which groups should be searched and how to search them.

15. In the **LDAP User Attribute Map** field, enter user attributes to map LDAP fields to your Ansible Automation Platform users for example, email or first_name.

16. In the **LDAP User Search** field, enter where to search for users during authentication.

17. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

18. To enable this authentication method upon creation, select **Enabled**.

19. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

20. Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.3. Configuring SAML authentication

SAML allows the exchange of authentication and authorization data between an Identity Provider (IdP) and a Service Provider (SP). Ansible Automation Platform is a SAML SP that can be configured to talk with one or more SAML IdPs in order to authenticate users. Based on groups and attributes optionally provided by the IdP users can be placed into teams and organizations in Ansible Automation Platform based on authenticator maps tied to this authenticator.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.

2. Click **Create authentication**.

3. Select **SAML** from the **Authentication type** list and click **Next**.

4. Enter a **Name** for this SAML configuration.

5. Enter the application-defined unique identifier used as the audience of the SAML service provider configuration in the **SAML Service Provider Entity ID** field. This is usually the URL for the service.

6. Include the certificate content in the **SAML Service Provider Public Certificate** field.

7. Include the private key content in the **SAML Service Provider Private Key**

8. Enter the URL to redirect the user to for login initiation in the **IdP Login URL** field.

9. Enter the public cert used for secrets coming from the **IdP in the IdP Public Cert**field.

10. Enter the entity ID returned in the assertion in the **Entity ID**. .Enter user details in the **Groups**, **User Email**, **Username**, **User Last Name**, **User First Name** and **User Permanent ID** fields.

11. The **SAML Assertion Consumer Service (ACS) URL**field registers the service as a service provider (SP) with each identity provider (IdP) you have configured.

12. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

    NOTE

    Values defined in this field override the dedicated fields provided in the UI.

13. In the **SAML Service Provider Organization Info**field, provide the URL, display name, and the name of your app.

```
{
  "en-US": {
    "url": "http://www.example.com",
    "displayname": "Example",
    "name": "example"
  }
}
```

14. In the **SAML Service Provider Technical Contact**field, provide the name and email address of the technical contact for your service provider.

```
{
"givenName": "Some User",
"emailAddress": "suser@example.com"
}
```

15. In the **SAML Service Provider Support Contact**field, provide the name and email address of the support contact for your service provider.

```
{
"givenName": "Some User",
"emailAddress": "suser@example.com"
}
```

16. Optional: Provide extra configuration data in the **SAML Service Provider extra configuration data** field. This field is the equivalent to the **SOCIAL_AUTH_SAML_SP_EXTRA** in the API. For more information, see OneLogin's SAML Python Toolkit to learn about the valid service provider extra (SP_EXTRA) parameters.

17. Optional: Provide security settings in the **SAML Security Config** field. This field is the equivalent to the **SOCIAL_AUTH_SAML_SECURITY_CONFIG** field in the API.

```
// Indicates whether the <samlp:AuthnRequest> messages sent by this SP // will be signed.
[Metadata of the SP will offer this info]
"authnRequestsSigned": false,
// Indicates a requirement for the <samlp:Response>, <samlp:LogoutRequest> // and
<samlp:LogoutResponse> elements received by this SP to be signed.
"wantMessagesSigned": false,

// Indicates a requirement for the <saml:Assertion> elements received by // this SP to be
signed. [Metadata of the SP will offer this info]
"wantAssertionsSigned": false,
```

For more information and additional options, see OneLogin’s SAML Python Toolkit.

18. Optional: In the **SAML IDP to extra_data attribute mapping** field, enter values to map IDP attributes to extra_data attributes. For more information, see advanced SAML settings.

19. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

20. To enable this authentication method upon creation, select **Enabled**.

21. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

22. Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.4. Configuring TACACS+ authentication

Terminal Access Controller Access-Control System Plus (TACACS+) is a protocol that handles remote authentication and related services for networked access control through a centralized server. TACACS+ provides authentication, authorization and accounting (AAA) services, in which you can configure Ansible Automation Platform to use as a source for authentication.

> **NOTE**
>
> This feature is deprecated and will be removed in a future release.

Procedure

1. From the navigation panel, select **Access Management → Authentication Methods**.

2. Click **Create authentication**.

3. Select **TACACS+** from the **Authentication type** list and click **Next**.

4. Enter a **Name** for this TACACS+ configuration.

5. Enter the following information:

    - Hostname of TACACS+ Server: Provide the hostname or IP address of the TACACS+ server with which to authenticate. If you leave this field blank, TACACS+ authentication is disabled.

- TACACS+ Authentication Protocol: The protocol used by the TACACS+ client. The options are ascii or pap.
- Shared secret for authenticating to TACACS+ server: The secret key for TACACS+ authentication server.

6. The **TACACS+ client address sending enabled** is disabled by default. To enable client address sending, select the checkbox.
7. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   NOTE

   Values defined in this field override the dedicated fields provided in the UI.

8. To automatically create organizations, users, and teams upon successful login, select **Create objects**.
9. To enable this authentication method upon creation, select **Enabled**.
10. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.
    Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.5. Configuring Microsoft Azure active directory authentication

To set up enterprise authentication for Microsoft Azure Active Directory (AD), you need to obtain an OAuth2 key and secret by registering your organization-owned application from Azure using the Quickstart: Register an application with the Microsoft identity platform .

Each key and secret must belong to a unique application and cannot be shared or reused between different authentication backends. To register the application, you must supply it with your webpage URL, which is the Callback URL shown in the Authenticator details for your authenticator configuration. See dDisplaying authenticator details for instructions on accessing this information.

Procedure

1. From the navigation panel, select **Access Management → Authentication Methods**.
2. Click **Create authentication**.
3. Select **Azuread** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this authentication configuration.
5. Click **Edit**, copy and paste Microsoft’s **Application (Client) ID** to the **OIDC Key** field.
   Following instructions for registering your application with the Microsoft identity platform , supply the key (shown at one time only) to the client for authentication.

6. Copy and paste the secret key created for your Microsoft Azure AD application to the OIDC Secret field.

7. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   NOTE

   Values defined in this field override the dedicated fields provided in the UI.

8. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

9. To enable this authentication method upon creation, select **Enabled**.

10. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.
    Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

Additional resources

For application registering basics in Microsoft Azure AD, see the What is the Microsoft identity platform? overview.

3.4.6. Configuring Google OAuth2 authentication

To set up social authentication for Google, you must obtain an OAuth2 key and secret for a web application. To do this, you must first create a project and set it up with Google.

For instructions, see Setting up OAuth 2.0 in the Google API Console Help documentation.

If you have already completed the setup process, you can access those credentials by going to the Credentials section of the Google API Manager Console. The OAuth2 key (Client ID) and secret (Client secret) are used to supply the required fields in the UI.

Procedure

1. From the navigation panel, select **Access Management → Authentication Methods**.

2. Click **Create authentication**.

3. Select **Google OAuth** from the **Authentication type** list and click **Next**.

4. Enter a **Name** for this authentication setting.

5. The **Google OAuth2 Key** and **Google OAuth2 Secret** fields are pre-populated.
   If not, use the credentials Google supplied during the web application setup process. Save these settings for use in the following steps.

6. Copy and paste Google's Client ID into the **Google OAuth2 Key** field.

7. Copy and paste Google's Client secret into the **Google OAuth2 Secret** field.

8. Optional: Enter information for the following fields using the tooltips provided for instructions and required format:

   - **Access Token URL**
   - **Access Token Method**
   - **Authorization URL**
   - **Revoke Token Method**
   - **Revoke Token URL**
   - **OIDC JWT Algorithm(s)**
   - **OIDC JWT**

9. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   > NOTE
   >
   > Values defined in this field override the dedicated fields provided in the UI.

10. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

11. To enable this authentication method upon creation, select **Enabled**.

12. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.
    Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.7. Configuring generic OIDC authentication

OpenID Connect (OIDC) uses the OAuth 2.0 framework. It enables third-party applications to verify the identity and obtain basic end-user information. The main difference between OIDC and SAML is that SAML has a service provider (SP)-to-IdP trust relationship, whereas OIDC establishes the trust with the channel (HTTPS) that is used to obtain the security token. To obtain the credentials needed to set up OIDC with Ansible Automation Platform, see the documentation from the IdP of your choice that has OIDC support.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
3. Select **Generic OIDC** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this authentication configuration.
5. Enter the following information:
    - **OIDC Provider URL**: The URL for your OIDC provider.
    - **OIDC Key**: The client ID from your third-party IdP.
    - **OIDC Secret**: The client secret from your IdP.
6. Optional: Select the HTTP method to be used when requesting an access token from the **Access Token Method** list. The default method is **POST**.
7. Optionally enter information for the following fields using the tooltips provided for instructions and required format:
    - **Access Token Method** - The default method is **POST**.
    - **Access Token URL**
    - **Access Token Method**
    - **Authorization URL**
    - **ID Key**
    - **ID Token Issuer**
    - **JWKS URI**
    - **OIDC Public Key**
    - **Revoke Token Method** - The default method is **GET**.
    - **Revoke Token URL**
    - **Response Type**
    - **Token Endpoint Auth Method**
    - **Userinfo URL**

- Username Key

8. Use the **Verify OIDC Provider Certificate** to enable or disable the OIDC provider SSL certificate verification.

9. Use the **Redirect** State to enable or disable the state parameter in the redirect URI. It is recommended that this is enabled to prevent Cross Site Request Forgery (CSRF) attacks.

10. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

> NOTE
>
> Values defined in this field override the dedicated fields provided in the UI.

11. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

12. To enable this authentication method upon creation, select **Enabled**.

13. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

14. Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.8. Configuring keycloak authentication

You can configure Ansible Automation Platform to integrate Keycloak to manage user authentication.

Procedure

1. From the navigation panel, select **Access Management → Authentication Methods**.

2. Click **Create authentication**.

3. Select **Keycloak** from the **Authentication type** list and click **Next**.

4. Enter a **Name** for this keycloak configuration. The configuration name is required, must be unique across all authenticators, and must not be longer than 512 characters.

5. Enter the location where the user’s token can be retrieved in the **Keycloak Access Token URL** field.

6. Optional: Enter the redirect location the user is taken to during the login flow in the **Keycloak Provider URL** field.

7. Enter the Client ID from your Keycloak installation in the **Keycloak OIDC Key** field.

8. Enter the RS256 public key provided by your Keycloak realm in the **Keycloak Public Key** field.

9. Enter the OIDC secret (Client Secret) from your Keycloak installation in the **Keycloak OIDC Secret** field.

10. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

    NOTE

    Values defined in this field override the dedicated fields provided in the UI.

11. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

12. To enable this authentication method upon creation, select **Enabled**.

13. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

14. Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.9. Configuring GitHub authentication

You can connect GitHub identities to Ansible Automation Platform using OAuth. To set up GitHub authentication, you need to obtain an OAuth2 key and secret by registering your organization-owned application from GitHub using the registering the new application with GitHub .

The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI. To register the application, you must supply it with your webpage URL, which is the Callback URL shown in the Authenticator details for your authenticator configuration. See Displaying authenticator details for instructions on accessing this information.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.

2. Click **Create authentication**.

3. Select **GitHub** from the **Authentication type** list and click **Next**.

4. Enter a **Name** for this authentication configuration.

5. When the application is registered, GitHub displays the **Client ID** and **Client Secret**:

   a. Copy and paste the GitHub Client ID into the GitHub OAuth2 Key field.

   b. Copy and paste the GitHub Client Secret into the GitHub OAuth2 Secret field.

6. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

NOTE

Values defined in this field override the dedicated fields provided in the UI.

7. To automatically create organizations, users, and teams upon successful login, select **Create objects**.
8. To enable this authentication method upon creation, select **Enabled**.
9. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.
10. Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.10. Configuring GitHub organization authentication

When defining account authentication with either an organization or a team within an organization, you should use the specific organization and team settings. Account authentication can be limited by an organization and by a team within an organization. You can also choose to permit all by specifying non-organization or non-team based settings. You can limit users who can log in to the platform by limiting only those in an organization or on a team within an organization.

To set up social authentication for a GitHub organization, you must obtain an OAuth2 key and secret for a web application using the registering the new application with GitHub .

The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI. To register the application, you must supply it with your webpage URL, which is the Callback URL shown in the Authenticator details for your authenticator configuration. See Displaying authenticator details for instructions on accessing this information.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
3. Select **GitHub organization** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this authentication configuration.
5. When the application is registered, GitHub displays the **Client ID** and **Client Secret**:
    a. Copy and paste the GitHub Client ID into the GitHub OAuth2 Key field.

   b. Copy and paste the GitHub Client Secret into the GitHub OAuth2 Secret field.

6. Enter the name of your GitHub organization, as used in your organization’s URL, for example, **https://github.com/<yourorg>/** in the **GitHub OAuth Organization Name** field.

7. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   NOTE

   Values defined in this field override the dedicated fields provided in the UI.

8. Enter the authorization scope for users in the **GitHub OAuth2 Scope** field. The default is **read:org**.

9. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

10. To enable this authentication method upon creation, select **Enabled**.

11. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

12. Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.11. Configuring GitHub team authentication

To set up social authentication for a GitHub team, you must obtain an OAuth2 key and secret for a web application using the instructions provided in registering the new application with GitHub .

The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI. To register the application, you must supply it with your webpage URL, which is the **Callback URL** shown in the Authenticator details for your authenticator configuration. See Displaying authenticator details for instructions on accessing this information.

Each key and secret must belong to a unique application and cannot be shared or reused between different authentication backends. The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI.

Procedure

1. From the navigation panel, select **Access Management → Authentication Methods**.

2. Click **Create authentication**.

3. Select **GitHub team** from the **Authentication type** list and click **Next**.

4. Enter a **Name** for this authentication configuration.

5. When the application is registered, GitHub displays the **Client ID** and **Client Secret**:

    a. Copy and paste the GitHub Client ID into the GitHub OAuth2 Key field.

    b. Copy and paste the GitHub Client Secret into the GitHub OAuth2 Secret field.

6. Copy and paste GitHub's team ID in the **GitHub OAuth2 Team ID** field.

7. Enter the authorization scope for users in the GitHub OAuth2 Scope field. The default is **read:org**.

8. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

    NOTE

    Values defined in this field override the dedicated fields provided in the UI.

9. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

10. To enable this authentication method upon creation, select **Enabled**.

11. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

12. Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.12. Configuring GitHub enterprise authentication

To set up social authentication for a GitHub enterprise, you must obtain a GitHub Enterprise URL, an API URL, OAuth2 key and secret for a web application.

To obtain the URLs, refer to the GitHub Enterprise administration documentation .

The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI. To register the application, you must supply it with your webpage URL, which is the **Callback URL** shown in the Authenticator details for your authenticator configuration. See Displaying authenticator details for instructions on accessing this information.

Each key and secret must belong to a unique application and cannot be shared or reused between different authentication backends. The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
3. Select **GitHub enterprise** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this authentication configuration.
5. When the application is registered, GitHub displays the **Client ID** and **Client Secret**:
   a. Copy and paste the GitHub Client ID into the GitHub OAuth2 Key field.
   b. Copy and paste the GitHub Client Secret into the GitHub OAuth2 Secret field.
6. In the **Base URL** field, enter the hostname of the GitHub Enterprise instance, for example, **https://github.example.com**.
7. In the **Github OAuth2 Enterprise API URL** field, enter the API URL of the GitHub Enterprise instance, for example, **https://github.example.com/api/v3**.
8. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   NOTE

   Values defined in this field override the dedicated fields provided in the UI.

9. To automatically create organizations, users, and teams upon successful login, select **Create objects**.
10. To enable this authentication method upon creation, select **Enabled**.
11. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.
12. Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.13. GitHub Enterprise Organization settings

To set up social authentication for a GitHub Enterprise Organization, you must obtain a GitHub Enterprise Organization URL, an Organization API URL, an Organization OAuth2 key and secret for a web application.

To obtain the URLs, refer to the GitHub Enterprise administration documentation .

The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI. To register the application, you must supply it with your webpage URL, which is the **Callback URL** shown in the Authenticator details for your authenticator configuration. See Displaying authenticator details for instructions on accessing this information.

Each key and secret must belong to a unique application and cannot be shared or reused between different authentication backends. The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. Click **Create authentication**.
3. Select **GitHub enterprise organization** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this authentication configuration.
5. When the application is registered, GitHub displays the **Client ID** and **Client Secret**:
   a. Copy and paste the GitHub Client ID into the GitHub OAuth2 Key field.
   b. Copy and paste the GitHub Client Secret into the GitHub OAuth2 Secret field.
6. In the **Base URL** field, enter the hostname of the GitHub Enterprise instance, for example, **https://github.example.com**.
7. In the **Github OAuth2 Enterprise API URL** field, enter the API URL of the GitHub Enterprise instance, for example, **https://github.example.com/api/v3**.
8. Enter the name of your GitHub Enterprise organization, as used in your organization’s URL, for example, **https://github.com/<yourorg>/** in the **GitHub OAuth2 Enterprise Org Name** field.
9. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   NOTE

   Values defined in this field override the dedicated fields provided in the UI.

10. To automatically create organizations, users, and teams upon successful login, select **Create objects**.
11. To enable this authentication method upon creation, select **Enabled**.
12. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

13. Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.14. Configuring GitHub enterprise team authentication

To set up social authentication for a GitHub enterprise team, you must obtain a GitHub Enterprise Organization URL, an Organization API URL, an Organization OAuth2 key and secret for a web application.

To obtain the URLs, refer to the GitHub Enterprise administration documentation .

To obtain the key and secret, you must first register your enterprise organization-owned application at **https://github.com/organizations/<yourorg>/settings/applications**.

The OAuth2 key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI. To register the application, you must supply it with your webpage URL, which is the **Callback URL** shown in the Authenticator details for your authenticator configuration. See Displaying authenticator details for instructions on accessing this information.

Each key and secret must belong to a unique application and cannot be shared or reused between different authentication backends. The OAuth2key (Client ID) and secret (Client Secret) are used to supply the required fields in the UI.

Procedure

1. From the navigation panel, select **Access Management → Authentication Methods**.
2. Click **Create authentication**.
3. Select **GitHub enterprise team** from the **Authentication type** list and click **Next**.
4. Enter a **Name** for this authentication configuration.
5. When the application is registered, GitHub displays the **Client ID** and **Client Secret**:
    a. Copy and paste the GitHub Client ID into the GitHub OAuth2 Key field.
    b. Copy and paste the GitHub Client Secret into the GitHub OAuth2 Secret field.
6. In the **Base URL** field, enter the hostname of the GitHub Enterprise instance, for example, **https://github.orgexample.com**.
7. In the **Github OAuth2 Enterprise API URL** field, enter the API URL of the GitHub Enterprise instance, for example, **https://github.example.com/api/v3**.

8. Enter the OAuth2 key (Client ID) from your GitHub developer application in the **GitHub OAuth2** team ID field.

9. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

   NOTE

   Values defined in this field override the dedicated fields provided in the UI.

10. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

11. To enable this authentication method upon creation, select **Enabled**.

12. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

13. Click **Next**.

Verification

To verify that the authentication is configured correctly, log out of Ansible Automation Platform and check that the login screen displays the logo of your authentication chosen method to enable logging in with those credentials.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.4.15. Configuring RADIUS authentication

You can configure Ansible Automation Platform to centrally use RADIUS as a source for authentication information.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.

2. Click **Create authentication**.

3. Select **Radius** from the **Authentication type** list and click **Next**.

4. Click **Create authentication**.

5. Enter the host or IP of the RADIUS server in the **RADIUS Server** field. If you leave this field blank, RADIUS authentication is disabled.

6. Enter the **Shared secret for authenticating to RADIUS server**

7. Optional: Enter any **Additional Authenticator Fields** that this authenticator can take. These fields are not validated and are passed directly back to the authenticator.

NOTE

Values defined in this field override the dedicated fields provided in the UI.

8. To automatically create organizations, users, and teams upon successful login, select **Create objects**.

9. To enable this authentication method upon creation, select **Enabled**.

10. To remove a user for any groups they were previously added to when they authenticate from this source, select **Remove users**.

11. Click **Next**.

Next steps

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or to what groups they belong, continue to Mapping.

3.5. MAPPING

To control which users are allowed into the Ansible Automation Platform server, and placed into Ansible Automation Platform organizations or teams based on their attributes (like username and email address) or what groups they belong to, you can configure authenticator maps.

Authenticator maps allow you to add conditions that must be met before a user is given or denied access to a resource type. Authenticator maps are associated with an authenticator and are given an order. The maps are processed in order when the user logs in. These can be thought of like firewall rules or mail filters.

3.5.1. Authenticator map types

Ansible Automation Platform supports the following rule types:

Allow

Determine if the user is allowed to log into the system.

Organization

Determine if a user should be put into an organization.

Team

Determine if the user should be a member of a team.

Role

Determine if the user is a member of a role (for example, *System Auditor*).

Is Superuser

Determine if the user is a superuser in the system.

These authentication map types can be used with any type of authenticator.

3.5.2. Authenticator map triggers

Each map has a trigger that defines when the map should be evaluated as true. Trigger types include the following:

Always

The trigger should always be fired.

Never

The trigger should never be fired.

Group

The map is true or false based on a user having, not having or having multiple groups in the source system. When defining a group trigger, the authentication mapping expands to include the following selections:

- **Operation:** This field includes conditional settings that trigger the handling of the rule based on the specified **Groups** criteria. The choices include **and** and **or**. For example, if you select **and** the user logging in must be a member of all of the groups specified in the **Groups** field for this trigger to be true. Alternatively, if you select **or** the user logging in must be a member of any of the specified **Groups** in order for the trigger to fire.

  NOTE

  If you are only keying off one group it doesn't matter if you select **"and"** or **"or"**.

- **Groups:** This is a list of one or more groups coming from the authentication system that the user must be a member of. See the **Operation** field to determine the behavior of the trigger if more than one group is specified in the trigger.

Attribute

The map is true or false based on a users attributes coming from the source system. When defining an attribute trigger, the authentication mapping expands to include the following selections:

- **Operation:** This field includes conditional settings that trigger the handling of the rule based on the specified **Attribute** criteria. In version 2.5 this field indicates what will happen if the source system returns a list of attributes instead of a single value. For example, if the source system returns multiple emails for a user and **Operation** was set to **and**, all of the given emails must match the **Comparison** for the trigger to be *True*. If **Operation** was set to **or**, any of the returned emails will set the trigger to *True* if they match the **Comparison** in the trigger.

  NOTE

  If you would like to experiment with multiple attribute maps you can do that through the API but the UI form will remove multi-attribute maps if the authenticator is saved through the UI. When adding multiple attributes to a map, the **Operation** will also apply to the attributes.

- **Attribute:** The name of the attribute coming from the source system this trigger will be evaluated against. For example, if you wanted the trigger to fire based on the user's last name and the last name field in the source system was called **users_last_name** you would enter the value 'users_last_name' in this field.

- **Comparison:** Tells the trigger how to evaluate the value of the users. **Attribute** in the source system compared to the **Value** specified on the trigger. Available options are: **contains**, **matches**, **ends with**, **in**, or **equals**. Below is a breakdown of each **Comparison** type:

  - **contains**: The specified character sequence in **Value** is contained within the attributes value returned from the source. For example, given an attribute value of ‘John’ from the source the contains **Comparison** would set the trigger to *True* if the trigger **Value** was set to ‘Jo’ and *False* if the trigger **Value** was ‘Joy’.

- **matches**: The **Value** on the trigger is treated as a python regular expression and does an Regular expression match (re.match) (with case ignore on) between the specified **Value** and the value returned from the source system. For example, if the trigger’s **Value** was ‘Jo’ the trigger would return *True* if the value from the source was ‘John’ or ‘Joanne’ or any other value which matched the regular expression ‘Jo’. The trigger would return *False* if the sources value for the attribute was ‘Dan’ because ‘Dan’ does not match the regular expression ‘Jo’.

- **ends with**: The trigger will see if the value provided by the source ends with the specified **Value** of the trigger. For example, if the source provided a value of ‘John’ the trigger would be *True* if its **Value** was set to ‘n’ or ‘on’. The trigger would be *False* if its **Value** was set to ‘z’ because the value ‘John’ coming from the source does not end with the value ‘z’ specified by the trigger.

- **equal**: The trigger will see if the value provided by the source is equal to (in its entirety) the specified **Value** of the trigger. For example, if the source returned the value ‘John’, the trigger would be *True* if its **Value** was set to ‘John’. Any value other than ‘John’ returned from the source would set this trigger to *False*.

- **in**: The **in** condition will see if the value matches one of several values. When **in** is specified as the **Comparison**, the **Value** field can be a comma separated list. For example, if a trigger had a **Value** of ‘John,Donna’ the trigger would be *True* if the attribute coming from the source had either the value ‘John’ or ‘Donna’. Otherwise, the trigger would be *False*.

- **Value**: The value that a users attribute will be matched against based on the **Comparison** field. See examples in the **Comparison** definition in this section.

NOTE

If the **Comparison** type is **in**, this field can be a comma separated list (without spaces).

3.5.3. Authenticator map examples

- Make this user a superuser if they have an attribute called **aap_superuser** with a value of *True*.

- Add this user to a team if they have the group **cn=Administrators,ou=AAP,ou=example,o=com** or **cn=Operators,ou=AAP,ou=example,o=com**.

- Never allow access to the system if the user has an attribute called **disabled** with a value of *True*, *Yes* or *Until Further Notice*.

Since maps are executed in order, it is possible to create exceptions. Expanding on the previous example for “Never allow access to the system if the user has an attribute called disabled with a value of *True*, *Yes* or *Until Further Notice*.

You can add another rule with a higher order, such as, “Allow access to the system for a **disabled** user if they are in the group **Emergency Contacts**.”

The first rule prevents the disabled user from accessing the system, but the second rule alters that decision to grant access to the system for the disabled user if they are in the **Emergency Contacts** group.

3.5.4. Allow mapping

With allow mapping, you can control which users have access to the system by defining the conditions that must be met.

Procedure

1. After configuring the authentication details for your authentication method, select **Allow** from the **Add authentication mapping** list.
2. Enter a unique rule **Name** to identify the rule.
3. Select a **Trigger** from the list. See Authenticator map triggers for more information about map triggers.
4. Select **Revoke** to deny user access to the system when none of the trigger conditions are matched.
5. Click **Next**.

Next steps

1. You can manage the authentication mappings order by dragging and dropping the mapping up or down in the list.

   NOTE

   The mapping precedence is determined by the order in which the mappings are listed.

2. Click **Next** to review and verify the mapping configurations.
3. Click **Finish**.

3.5.5. Organization mapping

You can control which users are placed into which Ansible Automation Platform organizations based on attributes like their username and email address or based on groups provided from an authenticator.

When organization mapping is positively evaluated, a specified organization is created, if it does not exist if the authenticator tied to the map is allowed to create objects.

Procedure

1. After configuring the authentication details for your authentication type, select **Organization** from the **Add authentication mapping** list.
2. Enter a unique rule **Name** to identify the rule.
3. Select a **Trigger** from the list. See Authenticator map triggers for more information about map triggers.

4. Select **Revoke** to deny user access to the system when none of the trigger conditions are matched.
5. Select the **Organization** to which matching users are added or blocked.
6. Select a **Role** to be applied or removed for matching users (for example, **Organization Admin** or **Organization Member**).
7. Click **Next**.

Next steps

1. You can manage the authentication mappings order by dragging and dropping the mapping up or down in the list.

   NOTE

   The mapping precedence is determined by the order in which the mappings are listed.

2. Click **Next** to review and verify the mapping configurations.
3. Click **Finish**.

3.5.6. Team mapping

Team mapping is the mapping of team members (users) from authenticators.

You can define the options for each team's membership. For each team, you can specify which users are automatically added as members of the team and also which users can administer the team.

Team mappings can be specified separately for each account authentication.

When Team mapping is positively evaluated, a specified team and its organization are created, if they don't exist if the related authenticator is allowed to create objects.

Procedure

1. After configuring the authentication details for your authentication type, select **Team** from the **Add authentication mapping** list.
2. Enter a unique rule **Name** to identify the rule.
3. Select a **Trigger** from the list. See Authenticator map triggers for more information about map triggers.
4. Select **Revoke** to deny user access to the system when none of the trigger conditions are matched.
5. Select the **Team** and **Organization** to which matching users are added or blocked.
6. Select a **Role** to be applied or removed for matching users (for example, **Team Admin** or **Team Member**).
7. Click **Next**.

Next steps

1. You can manage the authentication mappings order by dragging and dropping the mapping up or down in the list.

   NOTE

   The mapping precedence is determined by the order in which the mappings are listed.

2. Click **Next** to review and verify the mapping configurations.
3. Click **Finish**.

3.5.7. Role mapping

Role mapping is the mapping of a user either to a global role, such as Platform Auditor, or team or organization role.

When a Team and/or Organization is specified together with the appropriate Role, the behavior is identical with Organization mapping or Team mapping.

Role mapping can be specified separately for each account authentication.

Procedure

1. After configuring the authentication details for your authentication type, select **Team** from the **Add authentication mapping** list.
2. Enter a unique rule **Name** to identify the rule.
3. Select a **Trigger** from the list. See Authenticator map triggers for more information about map triggers.
4. Select **Revoke** to remove the role for the user when none of the trigger conditions are matched.
5. Select a **Role** to be applied or removed for matching users.
6. Click **Next**.

Next steps

1. You can manage the authentication mappings order by dragging and dropping the mapping up or down in the list.

   NOTE

   The mapping precedence is determined by the order in which the mappings are listed.

2. Click **Next** to review and verify the mapping configurations.
3. Click **Finish**.

3.5.8. Superuser mapping

Role mapping is the mapping of a user either to a global role, such as Platform Auditor, or team or organization role.

When a Team and/or Organization is specified together with the appropriate Role, the behavior is identical with Organization mapping or Team mapping.

Role mapping can be specified separately for each account authentication.

Procedure

1. After configuring the authentication details for your authentication type, select **Team** from the **Add authentication mapping** list.
2. Enter a unique rule **Name** to identify the rule.
3. Select a **Trigger** from the list. See Authenticator map triggers for more information about map triggers.
4. Select **Revoke** to remove the superuser role from the user when none of the trigger conditions are matched.
5. Click **Next**.

Next steps

1. You can manage the authentication mappings order by dragging and dropping the mapping up or down in the list.

   NOTE

   The mapping precedence is determined by the order in which the mappings are listed.

2. Click **Next** to review and verify the mapping configurations.
3. Click **Finish**.

3.6. MANAGING AUTHENTICATION IN ANSIBLE AUTOMATION PLATFORM

After you have configured your authentication settings, you can view a list of authenticators, search, sort and view the details for each authenticator configured on the system.

3.6.1. Authentication list view

On the **Authentication Methods** page, you can view and manage the configured authentication methods for your organization.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**. The **Authentication Methods** page is displayed.

2. Click **Create authentication** and follow the steps for creating an authentication method in Configuring an authentication type. Otherwise, proceed to step 3.

3. From the menu bar, you can sort the list of authentication methods by using the arrows in the menu bar for **Order**, **Name** and **Authentication type**.

4. Click the toggles to **Enable** or **Disable** authenticators.

3.6.2. Searching for an authenticator

You can search for a previously configured authenticator from the Authentication list view.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.

2. In the search bar, enter an appropriate keyword for the authentication method you want to search for and click the arrow icon.

3. If you don’t find what you’re looking for, you can narrow your search. From the filter list, select **Name** or **Authentication type** depending on the search term you want to use.

4. Scroll through the list of search results and select the authenticator you want to review.

3.6.3. Displaying authenticator details

After you locate the authenticator you want to review, you can display the configuration details:

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.

2. In the list view, select the authenticator name displayed in the **Name** column.
   The authenticator **Details** page is displayed.

3. From the **Details** page, you can review the configuration settings applied to the authenticator.

3.6.4. Editing an authenticator

You can modify the settings of previously configured authenticators from the **Authentication** list view.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.

2. In the list view, you can either:

   a. Select the **Edit** ✏ icon next to authenticator you want to modify, or

   b. Select the authenticator name displayed in the **Name** column and click **Edit authenticator** from the **Details** page.

3. Modify the authentication details or mapping configurations as required.

4. Click **Save**.

3.6.5. Deleting an authenticator

You can modify the settings of previously configured authenticators from the **Authentication** list view.

Procedure

1. From the navigation panel, select **Access Management** → **Authentication Methods**.
2. In the list view, select the checkbox next to the authenticator you want to delete.
3. Select **Delete authentication** from the ⋮ list.

> NOTE
>
> You can delete multiple authenticators by selecting the checkbox next to each authenticator you want to remove, and clicking **Delete selected authentication** from the ⋮ list on the menu bar.

CHAPTER 4. CONFIGURING ACCESS TO EXTERNAL APPLICATIONS WITH TOKEN-BASED AUTHENTICATION

Token-based authentication permits authentication of third-party tools and services with the platform through integrated OAuth 2 token support, and allows you to access external applications without having to store your password on disk.

For more information on the OAuth2 specification, see The OAuth 2.0 Authorization Framework.

For more information on using the **manage** utility to create tokens, see Token and session management.

4.1. APPLICATIONS

Create and configure token-based authentication for external applications such as ServiceNow and Jenkins. With token-based authentication, external applications can easily integrate with Ansible Automation Platform.

With OAuth 2 you can use tokens to share data with an application without disclosing login information. You can configure these tokens as read-only.

You can create an application that is representative of the external application you are integrating with, then use it to create tokens for the application to use on behalf of its users.

Associating these tokens with an application resource enables you to manage all tokens issued for a particular application. By separating the issue of tokens under **OAuth Applications**, you can revoke all tokens based on the application without having to revoke all tokens in the system.

4.1.1. Getting started with OAuth Applications

You can access the **OAuth Applications** page from the navigation panel by selecting **Access Management → OAuth Applications**. From there you can view, create, sort and search for applications currently managed by Ansible Automation Platform and automation controller.

If no applications exist, you can create one by clicking **Create OAuth application**.

4.1.1.1. Access Rules for applications and tokens

Access rules for applications are as follows:

- System administrators can view and manipulate all applications in the system.
- Organization administrators can view and manipulate all applications belonging to organization members.
- Other users can only view, update, and delete their own applications, but cannot create any new applications.
- Tokens, on the other hand, are resources used to authenticate incoming requests and mask the permissions of the underlying user.

Access rules for tokens are as follows:

- Users can create a token if they are able to view the related application and can also create a personal token for themselves.

- System administrators are able to view and manipulate every token in the system.
- Organization administrators are able to view and manipulate all tokens belonging to organization members.
- System Auditors can view all tokens and applications.
- Other normal users are only able to view and manipulate their own tokens.

NOTE

Users can only view the token or refresh the token value at the time of creation.

4.1.1.2. Application functions

Several OAuth 2 utilities are available for authorization, token refresh, and revoke. You can specify the following grant types when creating an application:

Password

This grant type is ideal for users who have native access to the web application and must be used when the client is the resource owner.

Authorization code

This grant type should be used when access tokens must be issued directly to an external application or service.

NOTE

You can only use the authorization code type to acquire an access token when using an application. When integrating an external web application with Ansible Automation Platform, that web application might need to create OAuth2 tokens on behalf of users in that other web application. Creating an application in the platform with the authorization code grant type is the preferred way to do this because:

- This allows an external application to obtain a token from Ansible Automation Platform for a user, using their credentials.
- Compartmentalized tokens issued for a particular application enables those tokens to be easily managed. For example, revoking *all* tokens associated with that application without having to revoke all tokens in the system.

4.1.1.2.1. Requesting an access token after expiration

The **Gateway access token expiration** defaults to 600 seconds (10 minutes).

The best way to set up application integrations using the **Authorization code** grant type is to allowlist the origins for those cross-site requests. More generally, you must allowlist the service or application you are integrating with the platform, for which you want to provide access tokens.

To do this, have your administrator add this allowlist to their local Ansible Automation Platform settings file:

```
CORS_ORIGIN_ALLOW_ALL = True
CORS_ALLOWED_ORIGIN_REGEXES = [
    r"http://django-oauth-toolkit.herokuapp.com*",
    r"http://www.example.com*"
]
```

Where **http://django-oauth-toolkit.herokuapp.com** and **http://www.example.com** are applications requiring tokens with which to access the platform.

4.1.2. Creating a new application

When integrating an external web application with automation controller the web application might need to create OAuth2 tokens on behalf of users of the web application.

Creating an application with the Authorization Code grant type is the preferred way to do this for the following reasons:

- External applications can obtain a token for users, using their credentials.
- Compartmentalized tokens issued for a particular application, enables those tokens to be easily managed. For example, revoking all tokens associated with that application.

Procedure

1. From the navigation panel, select **Access Management** → **OAuth Applications**.
2. Click **Create OAuth application**. The **Create Application** page opens.
3. Enter the following details:

   Name

   (required) Enter a name for the application you want to create.

   Description

   (optional) Include a short description for your application.

   Organization

   (required) Select an organization with which this application is associated.

   Authorization grant type

   (required) Select one of the grant types to use for the user to get tokens for this application. For more information, see Application functions for more information about grant types.

   Client Type

   (required) Select the level of security of the client device.

   Redirect URIS

   Provide a list of allowed URIs, separated by spaces. You need this if you specified the grant type to be **Authorization code**.

4. Click **Create OAuth application**, or click **Cancel** to abandon your changes.
   The **Client ID** and **Client Secret** display in a window. This will be the only time the client secret will be shown.

   NOTE

   The **Client Secret** is only created when the **Client type** is set to **Confidential**.

5. Click the copy icon and save the client ID and client secret to integrate an external application with Ansible Automation Platform.

4.2. ADDING TOKENS

You can view a list of users that have tokens to access an application by selecting the **Tokens** tab in the **OAuth Applications** details page.

> **NOTE**
>
> You can only create OAuth 2 Tokens for your own user, which means you can only configure or view tokens from your own user profile.

When authentication tokens have been configured, you can select the application to which the token is associated and the level of access that the token has.

Procedure

1. From the navigation panel, select **Access Management** → **Users**.
2. Select the username for your user profile to configure OAuth 2 tokens.
3. Select the **Tokens** tab.
   When no tokens are present, the **Tokens** screen prompts you to add them.
4. Click **Create token** to open the **Create Token** window.
5. Enter the following details:

   Application

   Enter the name of the application with which you want to associate your token. Alternatively, you can search for it by clicking **Browse**. This opens a separate window that enables you to choose from the available options. Select **Name** from the filter list to filter by name if the list is extensive.

   > **NOTE**
   >
   > To create a Personal Access Token (PAT) that is not linked to any application, leave the Application field blank.

   Description

   (optional) Provide a short description for your token.

   Scope

   (required) Specify the level of access you want this token to have. The scope of an OAuth 2 token can be set as one of the following:

   - **Write**: Allows requests sent with this token to add, edit and delete resources in the system.
   - **Read**: Limits actions to read only. Note that the write scope includes read scope.

6. Click **Create token**, or click **Cancel** to abandon your changes.

The Token information is displayed with **Token** and **Refresh Token** information, and the expiration date of the token. This will be the only time the token and refresh token will be shown. You can view the token association and token information from the list view.

7. Click the copy icon and save the token and refresh token for future use.

Verification

You can verify that the application now shows the user with the appropriate token using the Tokens tab on the Applications details page.

1. From the navigation panel, select **Access Management → OAuth Applications**.
2. Select the application you want to verify from the **Applications** list view.
3. Select the **Tokens** tab.
   Your token should be displayed in the list of tokens associated with the application you chose.

Additional resources

If you are a system administrator and have to create or remove tokens for other users, see the revoke and create commands in Token and session management.

4.2.1. Application token functions

The **refresh** and **revoke** functions associated with tokens, for tokens at the **/api/o/** endpoints can currently only be carried out with application tokens.

4.2.1.1. Refresh an existing access token

The following example shows an existing access token with a refresh token provided:

```
{
    "id": 35,
    "type": "access_token",
    ...
    "user": 1,
    "token": "omMFLk7UKpB36WN2Qma9H3gbwEBSOc",
    "refresh_token": "AL0NK9TTpv0qp54dGbC4VUZtsZ9r8z",
    "application": 6,
    "expires": "2017-12-06T03:46:17.087022Z",
    "scope": "read write"
}
```

The **/api/o/token/** endpoint is used for refreshing the access token:

```
curl -X POST \
    -d "grant_type=refresh_token&refresh_token=AL0NK9TTpv0qp54dGbC4VUZtsZ9r8z" \
    -u
"gwSPoasWSdNkMDtBN3Hu2WYQpPWCO9SwUEsKK22l:fI6ZpfocHYBGfm1tP92r0yIgCyfRdDQt0Tos
9L8a4fNsJjQQMwp9569eIaUBsaVDgt2eiwOGe0bg5m5vCSstClZmtdy359RVx2rQK5YlIWyPlrolpt2LEp
VeKXWaiybo" \
    http://<controller>/api/o/token/ -i
```

Where **refresh_token** is provided by **refresh_token** field of the preceding access token.

The authentication information is of format **<client_id>:<client_secret>**, where **client_id** and **client_secret** are the corresponding fields of the underlying related application of the access token.

NOTE

The special OAuth 2 endpoints only support using the **x-www-form-urlencoded Content-type**, so as a result, none of the **api/o/*** endpoints accept **application/json**.

On success, a response displays in JSON format containing the new (refreshed) access token with the same scope information as the previous one:

```
HTTP/1.1 200 OK
Server: nginx/1.12.2
Date: Tue, 05 Dec 2017 17:54:06 GMT
Content-Type: application/json
Content-Length: 169
Connection: keep-alive
Content-Language: en
Vary: Accept-Language, Cookie
Pragma: no-cache
Cache-Control: no-store
Strict-Transport-Security: max-age=15768000

{"access_token": "NDInWxGJI4iZgqpsreujjbvzCfJqgR", "token_type": "Bearer", "expires_in":
315360000000, "refresh_token": "DqOrmz8bx3srlHkZNKmDpqA86bnQkT", "scope": "read write"}
```

The refresh operation replaces the existing token by deleting the original and then immediately creating a new token with the same scope and related application as the original one.

Verify that the new token is present and the old one is deleted in the **/api/v2/tokens/** endpoint.

4.2.1.2. Revoke an access token

You can revoke an access token by deleting the token in the platform UI, or by using the **/api/o/revoke-token/** endpoint.

Revoking an access token by this method is the same as deleting the token resource object, but it enables you to delete a token by providing its token value, and the associated **client_id** (and **client_secret** if the application is **confidential**). For example:

```
curl -X POST -d "token=rQONsve372fQwuc2pn76k3IHDCYpi7" \
-u
"gwSPoasWSdNkMDtBN3Hu2WYQpPWCO9SwUEsKK22l:fI6ZpfocHYBGfm1tP92r0yIgCyfRdDQt0Tos
9L8a4fNsJjQQMwp9569eIaUBsaVDgt2eiwOGe0bg5m5vCSstClZmtdy359RVx2rQK5YlIWyPlrolpt2LEp
VeKXWaiybo" \
http://<controller>/api/o/revoke_token/ -i
```

NOTE

- The special OAuth 2 endpoints only support using the **x-www-form-urlencoded Content-type**, so as a result, none of the **api/o/*** endpoints accept **application/json**.

- The **Allow External Users to Create Oauth2 Tokens** (**ALLOW_OAUTH2_FOR_EXTERNAL_USERS** in the API) setting is disabled by default. External users refer to users authenticated externally with a service such as LDAP, or any of the other SSO services. This setting ensures external users cannot create their own tokens. If you enable then disable it, any tokens created by external users in the meantime will still exist, and are not automatically revoked. This setting can be configured from the **Settings → Platform gateway** menu.

Alternatively, to revoke OAuth2 tokens, you can use the **manage** utility, see Revoke oauth2 tokens.

On success, a response of **200 OK** is displayed. Verify the deletion by checking whether the token is present in the /**api**/**v2**/**tokens**/ endpoint.

4.2.2. Token and session management

Automation controller supports the following commands for OAuth2 token management:

- **create_oauth2_token**

- **revoke_oauth2_tokens**

- **cleartokens**

- **expire_sessions**

- **clearsessions**

4.2.2.1. create_oauth2_token

Use the following command to create OAuth2 tokens (specify the username for **example_user**):

```
$ awx-manage create_oauth2_token --user example_user

New OAuth2 token for example_user: j89ia8OO79te6IAZ97L7E8bMgXCON2
```

Ensure that you provide a valid user when creating tokens. Otherwise, an error message that you attempted to issue the command without specifying a user, or supplied a username that does not exist, is displayed.

4.2.2.2. revoke_oauth2_tokens

Use this command to revoke OAuth2 tokens, both application tokens and personal access tokens (PAT). It revokes all application tokens (but not their associated refresh tokens), and revokes all personal access tokens. However, you can also specify a user for whom to revoke all tokens.

To revoke all existing OAuth2 tokens use the following command:

```
$ awx-manage revoke_oauth2_tokens
```

- To revoke all OAuth2 tokens and their refresh tokens use the following command:

```
$ awx-manage revoke_oauth2_tokens --revoke_refresh
```

To revoke all OAuth2 tokens for the user with **id=example_user** (specify the username for **example_user**):

```
$ awx-manage revoke_oauth2_tokens --user example_user
```

To revoke all OAuth2 tokens and refresh token for the user with **id=example_user**:

```
$ awx-manage revoke_oauth2_tokens --user example_user --revoke_refresh
```

4.2.2.3. cleartokens

Use this command to clear tokens which have already been revoked.

For more information, see cleartokens in Django’s Oauth Toolkit documentation.

4.2.2.4. expire_sessions

Use this command to terminate all sessions or all sessions for a specific user.

Consider using this command when a user changes role in an organization, is removed from assorted groups in LDAP/AD, or the administrator wants to ensure the user can no longer execute jobs due to membership in these groups.

```
$ awx-manage expire_sessions
```

This command terminates all sessions by default. The users associated with those sessions are logged out. To only expire the sessions of a specific user, you can pass their username using the **--user** flag (replace **example_user** with the username in the following example):

```
$ awx-manage expire_sessions --user example_user
```

4.2.2.5. clearsessions

Use this command to delete all sessions that have expired.

For more information, see Clearing the session store in Django’s Oauth Toolkit documentation.

For more information on OAuth2 token management in the UI, see the Applications.

CHAPTER 5. MANAGING ACCESS WITH ROLE BASED ACCESS CONTROL

Role-based access control (RBAC) restricts user access based on their role within an organization to which they are assigned in Ansible Automation Platform. The roles in RBAC refer to the levels of access that users have to the Ansible Automation Platform components and resources.

You can control what users can do with the components of Ansible Automation Platform at a broad or granular level depending on your RBAC policy. You can designate whether the user is a system administrator or normal user and align roles and access permissions with their positions within the organization.

Roles can be defined with multiple permissions that can then be assigned to resources, teams and users. The permissions that make up a role dictate what the assigned role allows. Permissions are allocated with only the access needed for a user to perform the tasks appropriate for their role.

5.1. ORGANIZATIONS

An organization is a logical collection of users, teams, and resources. It is the highest level object in the Ansible Automation Platform object hierarchy. After you have created an organization, Ansible Automation Platform displays the organization details. You can then manage access and execution environments for the organization. Ansible Automation Platform automatically creates a default organization and the system administrator is automatically assigned to this organization. If you have a Self-support level license, you have only the default organization available and must not delete it.

5.1.1. Organizations list view

The **Organizations** page displays the existing organizations for your installation. From here, you can search for a specific organization, filter the list of organizations, or change the sort order for the list.

Procedure

1. From the navigation panel, select **menu:Access Management → Organizations**.
2. In the Search bar, enter an appropriate keyword for the organization you want to search for and click the arrow icon.
3. From the menu bar, you can sort the list of organizations by using the arrows for **Name** to toggle your sorting preference.
4. You can also sort the list by selecting **Name**, **Created** or **Last modified** from the **Sort** list.
5. You can view organization details by clicking an organization **Name** on the **Organizations** page.

5.1.2. Creating an organization

Ansible Automation Platform automatically creates a default organization. If you have a self-support level license, you have only the default organization available and cannot delete it.

Procedure

1. From the navigation panel, select **Access Management → Organizations**.
2. Click **Create organization**.

3. Enter the **Name** and optionally provide a **Description** for your organization.

   NOTE

   If automation controller is enabled on the platform, continue with Step 4. Otherwise, proceed to Step 6.

4. Select the name of the **Execution environment** or search for one that exists that members of this team can run automation.

5. Enter the name of the **Instance Groups** on which to run this organization.

6. Optional: Enter the **Galaxy credentials** or search from a list of existing ones.

7. Select the **Max hosts** for this organization. The default is 0. When this value is 0, it signifies no limit. If you try to add a host to an organization that has reached or exceeded its cap on hosts, an error message displays:

   > You have already reached the maximum number of 1 hosts allowed for your organization. Contact your System Administrator for assistance.

8. Click **Next**.

9. If you selected more than 1 instance group, you can manage the order by dragging and dropping the instance group up or down in the list and clicking **Confirm**.

   NOTE

   The execution precedence is determined by the order in which the instance groups are listed.

10. Click **Next** and verify the organization settings.

11. Click **Finish**.

5.1.3. Access to organizations

You can manage access to an organization by selecting an organization from the **Organizations** list view and selecting the associated tabs for providing access to Users, Administrators or Teams.

5.1.3.1. Adding a user to an organization

You can provide a user with access to an organization by adding them to the organization and managing the roles associated with the user. To add a user to an organization, the user must already exist. For more information, see Creating a user. To add roles for a user, the role must already exist. See Creating a role for more information.

Procedure

1. From the navigation panel, select **Access Management** → **Organizations**.

2. From the **Organizations** list view, select the organization to which you want to add a user.

3. Click the **Users** tab to add users.

4. Click **Add users** and select one or more users from the list by clicking the checkbox next to the name to add them as members.
5. Click **Next**.
6. Select the roles you want the selected user to have. Scroll down for a complete list of roles.

    NOTE

    If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

7. Click **Next** to review the roles settings.
8. Click **Finish** to apply the roles to the selected users, and to add them as members. The **Add roles** dialog displays the updated roles assigned for each user.

    NOTE

    A user with associated roles retains them if they are reassigned to another organization.

9. To remove a particular user from the organization, select **Remove user** from the **More actions** ⋮ list next to the user. This launches a confirmation dialog, asking you to confirm the removal.
10. To manage roles for users in an organization, click the ⚙ icon next to the user and select **Manage roles**.

5.1.3.2. Adding an administrator to an organization

You can add administrators to an organization which allows them to manage the membership and settings of the organization. For example, they can create new users and teams within the organization, and grant permission to users within the organization. To add an administrator to an organization, the user must already exist.

Procedure

1. From the navigation panel, select **Access Management** → **Organizations**.
2. From the Organizations list view, select the organization to which you want to add a user, administrator, or team.
3. Click the **Administrators** tab.
4. Click **Add administrators**.
5. Select the users from the list by clicking the checkbox next to the name to assign the administrator role to them for this organization.
6. Click **Add administrators**.
7. To remove a particular administrator from the organization, select **Remove administrator** f

7. To remove a particular administrator from the organization, select **Remove administrator** from the **More actions** ⋮ list next to the administrator name. This launches a confirmation dialog, asking you to confirm the removal.

   NOTE

   If the user had previously been added as a member to this organization, they will continue to be a member of this organization. However, if they were added to the organization when the administrator assignment was made, they will be removed from the organization.

5.1.3.3. Adding a team to an organization

You can provide team access to an organization by adding roles to the team. To add roles to a team, the team must already exist in the organization. For more information, see Creating a team. To add roles for a team, the role must already exist. See Creating a role for more information.

Procedure

1. From the navigation panel, select **Access Management** → **Organizations**.

2. From the Organizations list view, select the organization to which you want to add team access.

3. Click the **Teams** tab. If no teams exist, click **Create team** to create a team and add it to this organization.

4. Click **Add roles**.

5. Select the roles you want the selected team to have. Scroll down for a complete list of roles.

   NOTE

   If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

6. Click **Next** to review the roles settings.

7. Click **Finish** to apply the roles to the selected teams. The Add roles dialog displays the updated roles assigned for each team.

8. Click **Close**.

   NOTE

   A team with associated roles retains them if they are reassigned to another organization.

9. To manage roles for teams in an organization, click the ⚙ icon next to the user and select **Manage roles**.

5.1.3.4. Deleting an organization

Before you can delete an organization, you must be an Organization administrator or System administrator. When you delete an organization, the organization, team, users and resources are permanently removed from Ansible Automation Platform.

NOTE

When you attempt to delete items that are used by other resources, a message is displayed warning you that the deletion might impact other resources and prompts you to confirm the deletion. Some screens contain items that are invalid or have been deleted previously, and will fail to run.

Procedure

1. From the navigation panel, select **Access Management** → **Organizations**.
2. Click the ⋮ icon next to the organization you want removed and select **Delete organization**.
3. Select the confirmation checkbox and click **Delete organizations** to proceed with the deletion. Otherwise, click **Cancel**.

   NOTE

   You can delete multiple organizations by selecting the checkbox next to each organization you want to remove, and selecting **Delete selected organizations** from the **More actions** ⋮ list on the menu bar.

5.1.4. Working with notifiers

When automation controller is enabled on the platform, you can review any notifier integrations you have set up and manage their settings within the organization resource.

Procedure

1. From the navigation panel, select **Access Management** → **Organizations**.
2. From the **Organizations** list view, select the organization to which you want to manage notifications.
3. Select the **Notification** tab.
4. Use the toggles to enable or disable the notifications to use with your particular organization. For more information, see Enable and disable notifications .
5. If no notifiers have been set up, select **Automation Execution** → **Administration** → **Notifiers** from the navigation panel.

For information on configuring notification types, see Notification types.

5.1.5. Working with execution environments

When automation controller is enabled on the platform, you can review any execution environments you have set up and manage their settings within the organization resource.

For more information about execution environments, see Execution environments in *Using automation execution* guide.

Procedure

1. From the navigation panel, select **Access Management** → **Organizations**.
2. From the Organizations list view, select the organization whose execution environments you want to manage.
3. Select the **Execution Environments** tab.
4. If no execution environments are available, click **Create execution environment** to create one. Alternatively, you can create an execution environment from the navigation panel by selecting **Automation Execution** → **Infrastructure** → **Execution Environments**.
5. Click **Create execution environment**.

> NOTE
>
> After creating a new execution environments, return to **Access Management** → **Organizations** and select the organization in which you created the execution environment to update the list on that tab.

6. Select the execution environments to use with your particular organization.

5.2. TEAMS

A team is a subdivision of an organization with associated users, and resources. Teams provide a means to implement role-based access control schemes and delegate responsibilities across organizations. For instance, you can grant permissions to a Team rather than each user on the team.

You can create as many teams as needed for your organization. Teams can only be assigned to one organization while an organization can be made up of multiple teams. Each team can be assigned roles, the same way roles are assigned for users. Teams can also scalably assign ownership for credentials, preventing multiple interface click-throughs to assign the same credentials to the same user.

5.2.1. Teams list view

The Teams page displays the existing teams for your installation. From here, you can search for a specific team, filter the list of teams by team name or organization, or change the sort order for the list.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. In the **Search** bar, enter an appropriate keyword for the team you want to search for and click the arrow icon.
3. From the menu bar, you can sort the list of teams by using the arrows for **Name** and **Organization** to toggle your sorting preference.
4. You can view team details by clicking a team **Name** on the **Teams** page.
5. You can view organization details by clicking the link in the **Organization** column.

5.2.2. Creating a team

You can create new teams, assign an organization to the team, and manage the users and administrators associated with each team. Users associated with a team inherit the permissions associated with the team and any organization permissions to which the team has membership.

To add a user or administrator to a team, the user must have already been created.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Click **Create team**.
3. Enter a **Name** and optionally give a **Description** for the team.
4. Select an **Organization** to be associated with this team.

   NOTE

   Each team can only be assigned to one organization.

5. Click **Create team**.
   The **Details** page opens, where you can review and edit your team information.

5.2.3. Adding users to a team

To add a user to a team, the user must already have been created. For more information, see Creating a user. Adding a user to a team adds them as a member only. Use the **Roles** tab to assign a role for different resources to the selected team.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Select the team to which you want to add users.
3. Select the **Users** tab and click **Add users**.
4. Select one or more users from the list by clicking the checkbox next to the name to add them as members of this team.
5. Click **Add users**.

5.2.4. Removing users from a team

You can remove a user from a team from the Team list view.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Select the team from which you want to remove users.
3. Select the **Users** tab.
4. Click the **Remove user** icon next to the user you want to remove as a member of the team.

5. You can delete multiple users by selecting the checkbox next to each user you want to remove, and selecting **Remove selected users** from the **More actions** ⋮ list.

   NOTE

   If the user is a Team administrator, you can remove their membership to the team from the **Administrators** tab.

   This launches a confirmation dialog, asking you to confirm the removal.

5.2.5. Adding administrators to a team

You can add administrators to a team which allows them to manage the membership and settings of that team. For example, they can create new users and grant permission to users within the team. To add an administrator to a team, the administrator must already have been created. For more information, see Creating a user.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Select the team to which you want to add an administrator.
3. Select the **Administrators** tab and click **Add administrator(s)**.
4. Select one or more users from the list by clicking the checkbox next to the name to add them as administrators of this team.
5. Click **Add administrators**.

5.2.6. Adding roles to a team

You can assign permissions to teams, such as edit and administer resources and other elements. You can set permissions through an inventory, project, job template and other resources, or within the Organizations view.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Select the team **Name** to which you want to add roles.
3. Select the **Roles** tab and click **Add roles**.

   NOTE

   If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

4. Select a **Resource type** and click **Next**.
5. Select the resources to receive the new roles and click **Next**.

6. Select the roles to apply to the resources and click **Next**.
7. Review the settings and click **Finish**.
   The Add roles dialog displays indicating whether the role assignments were successfully applied, click **Close** to close the dialog.

5.2.7. Removing roles from a team

You can remove roles from a team by selecting the - icon next to the resource. This launches a confirmation dialog, asking you to confirm the removal.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Select the team **Name** from which you want to remove roles.
3. Select the **Roles** tab.

   > NOTE
   >
   > If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

4. Select the checkbox next to each resource you want to remove and click **Remove selected roles** from the ⋮ list on the menu bar.
5. Select the checkbox to confirm removal of the selected roles and click **Remove role**.

5.2.8. Deleting a team

Before you can delete a team, you must have team permissions. When you delete a team, the inherited permissions members got from that team are revoked.

Procedure

1. From the navigation panel, select **Access Management** → **Teams**.
2. Select the check box for the team that you want to remove.
3. Select the ⋮ icon and select **Delete team**.

   > NOTE
   >
   > You can delete multiple teams by selecting the checkbox next to each team you want to remove, and selecting **Delete teams** from the **More actions** ⋮ list.

5.3. USERS

Users associated with an organization are shown in the **Users** tab of the organization.

You can add other users to an organization, including a normal user or system administrator, but first, you must create them.

NOTE

Ansible Automation Platform automatically creates a default admin user so they can log in and set up Ansible Automation Platform for their organization. This user can not be deleted or modified.

You can sort or search the User list by **Username**, **First name**, **Last name**, or **Email**. Click the arrows in the header to toggle your sorting preference. You can view **User type** and **Email** beside the user name on the Users page.

5.3.1. Users list view

The **Users** page displays the existing users for your installation. From here, you can search for a specific user, filter the list of users, or change the sort order for the list.

Procedure

1. From the navigation panel, select **Access Management → Users**.
2. In the **Search** bar, enter an appropriate keyword for the user you want to search for and click the arrow icon.
3. From the menu bar, you can sort the list of users by using the arrows for **Username**, **Email**, **First name**, **Last name** or **Last login** to toggle your sorting preference.
4. You can view user details by selecting a **Username** from the **Users** list view.

5.3.2. Creating a user

There are three types of users in Ansible Automation Platform:

Normal user

Normal users have read and write access limited to the resources (such as inventory, projects, and job templates) for which that user has been granted the appropriate roles and privileges. Normal users are the default type of user when no other **User type** is specified.

Ansible Automation Platform Administrator

An administrator (also known as a Superuser) has full system administration privileges – with full read and write privileges over the entire installation. An administrator is typically responsible for managing all aspects of and delegating responsibilities for day-to-day work to various users.

Ansible Automation Platform Auditor

Auditors have read-only capability for all objects within the environment.

Procedure

1. From the navigation panel, select **Access Management → Users**.
2. Click **Create user**.
3. Enter the details about your new user in the fields on the **Create user** page. Fields marked with an asterisk (*) are required.

4. Normal users are the default when no **User type** is specified. To define a user as an administrator or auditor, select a **User type** checkbox.

   NOTE

   If you are modifying your own password, log out and log back in again for it to take effect.

5. Select the **Organization** to be assigned for this user. For information about creating a new organization, refer to Creating an organization.

6. Click **Create user**.

When the user is successfully created, the **User** dialog opens. From here, you can review and modify the user’s Teams, Roles, Tokens and other membership details.

NOTE

If the user is not newly-created, the details screen displays the last login activity of that user.

If you log in as yourself, and view the details of your user profile, you can manage tokens from your user profile by selecting the **Tokens** tab For more information, see Adding a token.

5.3.3. Editing a user

You can modify the properties of a user account after it is created.

Procedure

1. From the navigation panel, select **Access Management** → **Users**.

2. Select the check box for the user that you want to modify.

3. Click the **Pencil** icon and select **Edit user**.

4. The **Edit** user page is displayed where you can modify user details such as, **Password**, **Email**, **User type**, and **Organization**.

5. After your changes are complete, click **Save user**.

5.3.4. Deleting a user

Before you can delete a user, you must have normal user or system administrator permissions. When you delete a user account, the name and email of the user are permanently removed from Ansible Automation Platform.

Procedure

1. From the navigation panel, select **Access Management** → **Users**.

2. Select the checkbox for the user that you want to remove.

3. Click the ⋮ icon next to the user you want removed and select **Delete user**.

> NOTE
>
> You can delete multiple users by selecting the checkbox next to each user you want to remove, and clicking **Delete users** from the **More actions** ⋮ list.

5.3.5. Adding roles for a user

You can grant access for users to use, read, or write credentials by assigning roles to them.

Procedure

1. From the navigation panel, select **Access Management** → **Users**.
2. From the **Users** list view, click on the user to which you want to add roles.
3. Select the **Roles** tab to display the set of roles assigned to this user. These provide the ability to read, modify, and administer resources.
4. To add new roles, click **Add roles**.

   > NOTE
   >
   > If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

5. Select a Resource type and click **Next**.
6. Select the resources that will receive new roles and click **Next**.
7. Select the roles that will be applied to the resources and click **Next**.
8. Review the settings and click **Finish**.
   The Add roles dialog displays indicating whether the role assignments were successfully applied. Click **Close** to close the dialog.

5.3.6. Removing roles from a user

You can remove roles from a user by selecting the **-** icon next to the resource. This launches a confirmation dialog, asking you to confirm the removal.

Procedure

1. From the navigation panel, select **Access Management** → **Users**.
2. Select the user Name from which you want to remove roles.
3. Select the **Roles** tab.

NOTE

If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

4. Select the checkbox next to each resource you want to remove and click **Remove selected roles** from the **More actions** ⋮ list on the menu bar.

5. Select the checkbox to confirm removal of the selected roles and click **Remove role**.

5.4. RESOURCES

You can manage user access to Ansible Automation Platform resources and what users can do with those resources. Users are granted access through the roles to which they are assigned or through roles inherited through the role hierarchy, for example, through the roles they inherit through team membership. Ansible Automation Platform resources differ depending on the functionality you are configuring. For example, resources can be job templates and projects for automation execution or decision environments and rulebook activations for automation decisions.

5.4.1. Providing team access to a resource

You can grant users access based on their team membership. When you add a user as a member of a team, they inherit access to the roles and resources defined for that team.

Procedure

1. From the navigation panel, select a resource to which you want to provide team access. For example, **Automation Execution → Templates**.

2. Select the **Team Access** tab.

3. Click **Add roles**.

4. Click the checkbox beside the team to assign that team to your chosen type of resource and click **Next**.

5. Select the roles you want applied to the team for the chosen resource and click **Next**.

6. Review the settings and click **Finish**. The Add roles dialog displays indicating whether the role assignments were successfully applied.

7. You can remove resource access for a team by selecting the **Remove role** icon next to the team. This launches a confirmation dialog, asking you to confirm the removal.

5.4.2. Providing user access to a resource

You can grant users access to resources through the roles to which they are assigned.

Procedure

1. From the navigation panel, select a resource to which you want to provide team access. For example, **Automation Execution → Templates**.

2. Select the **User access** tab.

3. Click **Add roles**.

4. Click the checkbox beside the user to assign that user to your chosen type of resource and click **Next**.

5. Select the roles you want applied to the user for the chosen resource and click **Next**.

6. Review the settings and click **Finish**. The Add roles dialog displays indicating whether the role assignments were successfully applied.

7. You can remove resource access for a user by selecting the **Remove role** icon next to the user. This launches a confirmation dialog, asking you to confirm the removal.

CHAPTER 6. ROLES

Roles are units of organization in the Red Hat Ansible Automation Platform. When you assign a role to a team or user, you are granting access to use, read, or write credentials. Because of the file structure associated with a role, roles become redistributable units that enable you to share behavior among resources, or with other users. All access that is granted to use, read, or write credentials is handled through roles, and roles are defined for a resource.

6.1. DISPLAYING ROLES

You can display the roles assigned for component resources from the **Access Management** menu.

Procedure

1. From the navigation panel, select **Access Management** → **Roles**.

2. Select a tab for the component resource for which you want to create custom roles.

   NOTE

   If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

3. From the table header, you can sort the list of roles by using the arrows for **Name**, **Description**, **Created** and **Editable** or by making sort selections in the **Sort** list.

4. You can filter the list of roles by selecting **Name** or **Editable** from the filter list and clicking the arrow.

6.2. CREATING A ROLE

Ansible Automation Platform services provide a set of predefined roles with permissions sufficient for standard automation tasks. It is also possible to configure custom roles, and assign one or more permission filters to them. Permission filters define the actions allowed for a specific resource type.

Procedure

1. From the navigation panel, select **Access Management** → **Roles**.

2. Select a tab for the component resource for which you want to create custom roles.

   NOTE

   If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

3. Click **Create role**.

4. Provide a **Name** and optionally include a **Description** for the role.

5. Select a **Content Type**.
6. Select the **Permissions** you want assigned to this role.
7. Click **Create role** to create your new role.

6.3. EDITING A ROLE

Built in roles can not be changed, however, you can modify custom roles from the **Roles** list view. The **Editable** column in the **Roles** list view indicates whether a role is *Built-in* or *Editable*.

Procedure

1. From the navigation panel, select **Access Management** → **Roles**.
2. Select a tab for the component resource for which you want to modify a custom role.

   NOTE

   If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

3. Click the **Edit role** icon ✎ next to the role you want and modify the role settings as needed.
4. Click **Save role** to save your changes.

6.4. DELETING A ROLE

Built in roles can not be deleted, however, you can delete custom roles from the **Roles** list view.

Procedure

1. From the navigation panel, select **Access Management** → **Roles**.
2. Select a tab for the component resource for which you want to create custom roles.

   NOTE

   If you have multiple Ansible Automation Platform components installed, you will see selections for the roles associated with each component in the **Roles** menu bar. For example, Automation Execution for automation controller roles, Automation Decisions for Event-Driven Ansible roles.

3. Click the **More Actions** icon ⋮ next to the role you want and select **Delete role**.
4. To delete roles in bulk, select the roles you want to delete from the **Roles** list view, click the **More Actions** icon ⋮, and select **Delete roles**.

CHAPTER 7. CONFIGURING ANSIBLE AUTOMATION PLATFORM

You can configure Ansible Automation platform from the **Settings** menu using the following selections:

- **Subscriptions**
- **Platform gateway**
- **User Preferences**
- **Troubleshooting**

NOTE

The other selections available from the **Settings** menu are specific to automation execution. For more information, refer to the Configuring automation execution guide.

7.1. CONFIGURING SUBSCRIPTIONS

You can use the **Subscription** menu to view the details of your subscription, such as compliance, host-related statistics, or expiry, or you can apply a new subscription.

Procedure

1. From the navigation panel, select **Settings → Subscription**. The **Subscription** page is displayed.
2. Click **Edit subscription**.
3. You can either enter your Red Hat Username and Password, or attach a current Subscription Manifest in the **Welcome** page.
4. Click **Next** and agree to the terms of the license agreement.
5. Click **Next** to review the subscription settings.
6. Click **Finish** to complete the configuration.

7.2. PLATFORM GATEWAY

The Platform gateway is the service that handles authentication and authorization for Ansible Automation Platform. It provides a single ingress into the Platform and serves the Platform's user interface.

From the **Settings → Platform gateway** menu, you can configure **Platform gateway**, **Security**, **Session**, **Platform Security**, **Custom Login**, and **Other** settings.

Procedure

1. From the navigation panel, select **Settings → Platform gateway**.
2. The **Platform gateway settings** page is displayed.

3. To configure the options, click **Edit platform gateway settings**.

4. You can configure the following platform gateway options:

   - **Platform gateway proxy url:** URL to the platform gateway proxy layer.

   - **Platform gateway proxy url ignore cert** Ignore the certificate to the platform gateway proxy layer.

5. Click **Save platform gateway settings** to save the changes or proceed to configure the other platform options available.

7.2.1. Configuring platform security

From the **Platform gateway settings** page, you can configure platform security settings.

Procedure

1. From the navigation panel, select **Settings → Platform gateway**.

2. The **Platform gateway settings** page is displayed.

3. To configure the options, click **Edit**.

4. You can configure the following **Security** settings:

   - **Allow admin to set insecure:** Whether a superuser account can save an insecure password when editing any local user account.

   - **Gateway basic auth enabled:** Enable basic authentication to the platform gateway API. Turning this off prevents all basic authentication (local users), so customers need to make sure they have their alternative authentication mechanisms correctly configured before doing so.

     Turning it off with only local authentication configured also prevents all access to the UI.

     Social auth username in full email Enabling this setting alerts social authentication to use the full email as username instead of the full name.

     Gateway token name: The header name to push from the proxy to the backend service.

     > **WARNING**
     >
     > If this name is changed, backends must be updated to compensate.

   - **Gateway access token expiration:** How long the access tokens are valid for.

   - **Jwt private key**: The private key used to encrypt the JWT tokens sent to backend services. This should be a private RSA key and one should be generated automatically on installation.

> NOTE
>
> Use caution when rotating the key as it will cause current sessions to fail until their JWT keys are reset.

- (Read only) **Jwt public key**: The private key used to encrypt the JWT tokens sent to backend services.
  This should be a private RSA key and one should be generated automatically on installation.

> NOTE
>
> See other services' documentation on how they consume this key.

5. Click **Save changes** to save the changes or proceed to configure the other platform options available.

7.2.2. Configuring platform sessions

From the **Platform gateway settings** page, you can configure platform session settings.

Procedure

1. From the navigation panel, select **Settings → Platform gateway**.
2. The **Platform gateway settings** page is displayed.
3. To configure the options, click **Edit platform gateway settings**.
4. Enter the time in seconds before a session expires in the **Session cookie age** field.
5. Click **Save platform gateway settings** to save the changes or proceed to configure the other platform options available.

7.2.3. Configuring a platform password security policy

From the **Platform gateway settings** page, you can configure a password security policy.

Procedure

1. From the navigation panel, select **Settings → Platform gateway**.
2. The **Platform gateway settings** page is displayed.
3. To configure the options, click **Edit platform gateway settings**.
4. You can configure the following **Password Security** options:
   - **Password minimum uppercase letters**: How many uppercase characters need to be in a local password.
   - **Password minimum length**: The minimum length of a local password.
   - **Password minimum numerical digits**: How many numerical characters need to be in a local password.

- **Password minimum special characters**: How many special characters need to be in a local password.

5. Click **Save platform gateway settings** to save the changes or proceed to configure the other platform options available.

7.2.4. Configuring a custom platform log in

From the **Platform gateway settings** page, you can configure the custom log in options.

Procedure

1. From the navigation panel, select **Settings → Platform gateway**.
2. The **Platform gateway settings** page is displayed.
3. To configure the options, click **Edit platform gateway settings**.
4. You can configure the following **Custom Login** options:
   - **Custom login info**: Provide specific information (such as a legal notice or a disclaimer) to a text box in the login modal. For example, you can include a company banner with a statement such as, "This is only to be used for **<COMPANY_NAME>**, etc."
   - **Custom logo** : Provide an image file for setting up a custom logo (must be a data URL with a base64-encoded GIF, PNG, or JPEG image).
5. Click **Save platform gateway settings** to save the changes or proceed to configure the other platform options available.

7.2.5. Configuring additional platform options

From the **Platform gateway settings** page, you can configure additional platform options.

Procedure

1. From the navigation panel, select **Settings → Platform gateway**.
2. The **Platform gateway settings** page is displayed.
3. Click **Edit platform gateway settings**.
4. You can configure the following **Other settings**:
   - **Jwt expiration buffer in seconds**: The number of seconds before a JWT token's expiration to revoke from the cache.
     When authentication happens a JWT token is created for the user and that token is cached. When subsequent calls happen to services such as automation controller or Event-Driven Ansible, the token is taken from the cache and sent to the service. Both the token and the cache of the token have an expiration time. If the token expires while in the cache the authentication process attempts results in a 401 error (unauthorized). This setting gives Red Hat Ansible Automation Platform a buffer by removing the JWT token from the cache before the token expires. When a token is revoked from cache a new token with a new expiration is generated and cached for the user. As a result, expired tokens from the cache

are never used. This setting defaults to 2 seconds. If you have a large latency between platform gateway and your services and observe 401 responses you must increase this setting to lower the number of 401 responses.

- **Status endpoint backend timeout seconds**: Timeout (in seconds) for the status endpoint to wait when trying to connect to a backend.
- **Status endpoint backend verify**: Specifies whether SSL certificates of the services are verified when calling individual nodes for statuses.
- **Request timeout**: Specifies, in seconds, the length of time before the proxy will report a timeout and generate a 504.
- *Allow external users to create OAuth2 tokens *: For security reasons, users from external authentication providers, such as LDAP, SAML, SSO, Radius, and others, are not allowed to create OAuth2 tokens. To change this behavior, enable this setting. Existing tokens are not deleted when this setting is turned off.

5. Click **Save platform gateway settings** to save the changes or proceed to configure the other platform options available.

7.3. USER PREFERENCES

You can use the **User preferences** page to customize your platform experience. Use this menu to control theming, layout options and formatting.

NOTE

User preferences are stored locally in your browser. This means that they are unique to you and your machine.

Procedure

1. From the navigation panel, select **Settings → User Preferences**.
2. The **User Preferences page** is displayed.
3. Click **Edit**.
4. You can configure the following options:
   - **Refresh interval**: Select the refresh interval for the page.
     This refreshes the data on the page at the selected interval.

     The refresh happens in the background and does not reload the page.
   - **Color theme**: Select from:
     - Dark theme
     - Light theme
     - System default
   - **Table layout**: Select from:

    - Comfortable
    - Compact
  - **Form columns**: Select from:
    - Multiple columns of inputs
    - Single column of inputs
  - **Date format** Select from:
    - Shows dates **Relative** to the current time
    - Shows dates as **Date and time**
  - **Preferred data format**: Sets the default format for when editing and displaying data.

5. Click **Save user preferences**.

7.4. TROUBLESHOOTING OPTIONS

You can use the **Troubleshooting** page to enable or disable certain flags that aid in debugging issues within Ansible Automation Platform.

Procedure

1. From the navigation panel, select **Settings → Troubleshooting**.
2. The **Troubleshooting** page is displayed.
3. Click **Edit**.
4. You can select the following options:
   - **Enable or Disable tmp dir cleanup** Select this to enable or disable the cleanup of tmp directories generated during execution of a job after job execution completes.
   - **Debug Web Requests** Select this to enable or disable web request profiling for debugging slow web requests.
   - **Release Receptor Work** Select this to turn on or off the deletion of job pods after they complete or fail. This can be helpful in debugging why a job failed.
   - **Keep receptor work on error**: Select this to prevent receptor work from being released when an error is detected.
5. Click **Save** to save your changes.