Authenticating to the GitHub API

In this article

Learn how to authenticate to the GitHub API to use Actions Runner Controller with your GitHub Enterprise Server instance.

[Legal notice](#)

Overview 🔗

You can authenticate Actions Runner Controller (ARC) to the GitHub API by using a GitHub App or by using a personal access token (classic).

> **Note:** You cannot authenticate using a GitHub App for runners at the enterprise level. For more information, see "[Managing access to self-hosted runners using groups](#)."

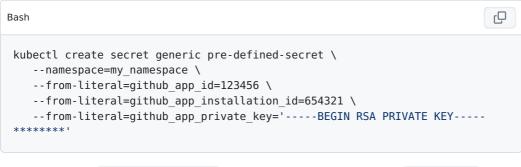## Authenticating ARC with a GitHub App 🔗

1. Create a GitHub App that is owned by an organization. For more information, see "[Registering a GitHub App](#)". Configure the GitHub App as follows.

   a. For "Homepage URL," enter `http://github.com/actions/actions-runner-controller`.

   b. Under "Permissions," click **Repository permissions**. Then use the dropdown menus to select the following access permissions.

      - **Administration**: Read and write
      - **Metadata**: Read-only

   c. Under "Permissions," click **Organization permissions**. Then use the dropdown menus to select the following access permissions.

      - **Self-hosted runners**: Read and write

2. After creating the GitHub App, on the GitHub App's page, note the value for "App ID". You will use this value later.

3. Under "Private keys", click **Generate a private key**, and save the `.pem` file. You will use this key later.

4. In the menu at the top-left corner of the page, click **Install app**, and next to your organization, click **Install** to install the app on your organization.

5. After confirming the installation permissions on your organization, note the app installation ID. You will use it later. You can find the app installation ID on the app installation page, which has the following URL format:

```
https://HOSTNAME/organizations/ORGANIZATION/settings/installations/INSTALLATION_
ID
```

6. Register the app ID, installation ID, and the downloaded `.pem` private key file from the previous steps to Kubernetes as a secret.

   To create a Kubernetes secret with the values of your GitHub App, run the following command.

   Bash

   ```bash
   kubectl create secret generic pre-defined-secret \
       --namespace=my_namespace \
       --from-literal=github_app_id=123456 \
       --from-literal=github_app_installation_id=654321 \
       --from-literal=github_app_private_key='-----BEGIN RSA PRIVATE KEY-----
   ********'
   ```

   Then using the `githubConfigSecret` property in your copy of the `values.yaml` file, pass the secret name as a reference.

   ```
   githubConfigSecret: pre-defined-secret
   ```

For additional Helm configuration options, see `values.yaml` in the ARC repository.

Authenticating ARC with a personal access token (classic) 🔗

ARC can use personal access tokens (classic) to register self-hosted runners.

> **Note:** Authenticating ARC with a personal access token (classic) is the only supported authentication method to register runners at the enterprise level.

1. Create a personal access token (classic) with the required scopes. The required scopes are different depending on whether you are registering runners at the repository, organization, or enterprise level. For more information on how to create a personal access token (classic), see "[Managing your personal access tokens](#)."

   The following is the list of required personal access token scopes for ARC runners.

   - Repository runners: `repo`
   - Organization runners: `admin:org`
   - Enterprise runners: `manage_runners:enterprise`

2. To create a Kubernetes secret with the value of your personal access token (classic), use the following command.

   Bash

```
kubectl create secret generic pre-defined-secret \
    --namespace=my_namespace \
    --from-literal=github_token='YOUR-PAT'
```

3   In your copy of the `values.yaml` file, pass the secret name as a reference.

```
githubConfigSecret: pre-defined-secret
```

For additional Helm configuration options, see `values.yaml` in the ARC repository.

Legal notice 🔗

Portions have been adapted from https://github.com/actions/actions-runner-controller/ under the Apache-2.0 license:

```
Copyright 2019 Moto Ishizawa

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```