

Quickstart for Actions Runner Controller

In this article

Introduction

Prerequisites

Installing Actions Runner Controller

Configuring a runner scale set

Using runner scale sets

Next steps

Legal notice

Try out Actions Runner Controller in 5 minutes.

[Legal notice](#)

Introduction

Actions Runner Controller (ARC) is a Kubernetes operator that orchestrates and scales self-hosted runners for GitHub Actions. For more information, see [Operator pattern](#) in the Kubernetes documentation.

With ARC, you can create runner scale sets that automatically scale based on the number of workflows running in your repository, organization, or enterprise. Because controlled runners can be ephemeral and based on containers, new runner instances can scale up or down rapidly and cleanly. For more information about autoscaling, see "[Autoscaling with self-hosted runners](#)."

You can set up ARC on Kubernetes using Helm, then create and run a workflow that uses runner scale sets. For more information about runner scale sets, see "[Deploying runner scale sets with Actions Runner Controller](#)."

Prerequisites

In order to use ARC, ensure you have the following.

- A Kubernetes cluster
 - For a managed cloud environment, you can use AKS. For more information, see [Azure Kubernetes Service](#) in the Azure documentation.
 - For a local setup, you can use minikube or kind. For more information, see [minikube start](#) in the minikube documentation and [kind](#) in the kind documentation.

Note: OpenShift clusters are currently unsupported.

- Helm 3

- For more information, see [Installing Helm](#) in the Helm documentation.

Installing Actions Runner Controller

- 1 To install the operator and the custom resource definitions (CRDs) in your cluster, do the following.
 - a. In your Helm chart, update the `NAMESPACE` value to the location you want your operator pods to be created. This namespace must allow access to the Kubernetes API server.
 - b. Install the Helm chart.

The following example installs the latest version of the chart. To install a specific version, you can pass the `--version` argument along with the version of the chart you wish to install. You can find the list of releases in the [GitHub Container Registry](#).

Bash

```
NAMESPACE="arc-systems"
helm install arc \
  --namespace "${NAMESPACE}" \
  --create-namespace \
  oci://ghcr.io/actions/actions-runner-controller-charts/gha-runner-scale-
set-controller
```

For additional Helm configuration options, see [values.yaml](#) in the ARC documentation.

- 2 To enable ARC to authenticate to GitHub, generate a personal access token (classic). For more information, see [Authenticating to the GitHub API](#).

Configuring a runner scale set

- 1 To configure your runner scale set, run the following command in your terminal, using values from your ARC configuration.

When you run the command, keep the following in mind.

- Update the `INSTALLATION_NAME` value carefully. You will use the installation name as the value of `runs-on` in your workflows. For more information, see "[Workflow syntax for GitHub Actions](#)."
- Update the `NAMESPACE` value to the location you want the runner pods to be created.
- Set `GITHUB_CONFIG_URL` to the URL of your repository, organization, or enterprise. This is the entity that the runners will belong to.
- This example command installs the latest version of the Helm chart. To install a specific version, you can pass the `--version` argument with the version of the chart you wish to install. You can find the list of releases in the [GitHub Container Registry](#).

Note:

- As a security best practice, create your runner pods in a different namespace than the namespace containing your operator pods.
- As a security best practice, create Kubernetes secrets and pass the secret

references. Passing your secrets in plain text via the CLI can pose a security risk. For more information, see "[Deploying runner scale sets with Actions Runner Controller](#)."

Bash

```
INSTALLATION_NAME="arc-runner-set"
NAMESPACE="arc-runners"
GITHUB_CONFIG_URL="https://github.com/<your_enterprise/org/repo>"
GITHUB_PAT="<PAT>"
helm install "${INSTALLATION_NAME}" \
  --namespace "${NAMESPACE}" \
  --create-namespace \
  --set githubConfigUrl="${GITHUB_CONFIG_URL}" \
  --set githubConfigSecret.github_token="${GITHUB_PAT}" \
  oci://ghcr.io/actions/actions-runner-controller-charts/gha-runner-
scale-set
```

For additional Helm configuration options, see [values.yaml](#) in the ARC documentation.

- 2 From your terminal, run the following command to check your installation.

Bash

```
helm list -A
```

You should see an output similar to the following.

NAME	NAMESPACE	REVISION	UPDATED	APP VERSION
arc	arc-systems	1	2023-04-12	
11:45:59.152090536 +0000 UTC	deployed	gha-runner-scale-set-		
controller-0.4.0	0.4.0			
arc-runner-set	arc-runners	1	2023-04-12	
11:46:13.451041354 +0000 UTC	deployed	gha-runner-scale-set-0.4.0		
0.4.0				

- 3 To check the manager pod, run the following command in your terminal.

Bash

```
kubectl get pods -n arc-systems
```

If everything was installed successfully, the status of the pods shows as **Running**.

NAME	READY	STATUS
arc-gha-runner-scale-set-controller-594cdc976f-m7cjs	1/1	Running
64s		
arc-runner-set-754b578d-listener	1/1	Running
12s		

If your installation was not successful, see "[Troubleshooting Actions Runner Controller errors](#)" for troubleshooting information.

Using runner scale sets [↗](#)

Now you will create and run a simple test workflow that uses the runner scale set runners.

- 1 In a repository, create a workflow similar to the following example. The `runs-on` value should match the Helm installation name you used when you installed the autoscaling runner set.


For more information on adding workflows to a repository, see "[Quickstart for GitHub Actions](#)."

YAML 

```
name: Actions Runner Controller Demo
on:
  workflow_dispatch:

jobs:
  Explore-GitHub-Actions:
    # You need to use the INSTALLATION_NAME from the previous step
    runs-on: arc-runner-set
    steps:
      - run: echo " This job uses runner scale set runners!"
```

- 2 Once you've added the workflow to your repository, manually trigger the workflow. For more information, see "[Manually running a workflow](#)."
- 3 To view the runner pods being created while the workflow is running, run the following command from your terminal.

Bash 

```
kubectl get pods -n arc-runners
```

A successful output will look similar to the following.

NAMESPACE	NAME	READY
STATUS	RESTARTS	AGE
arc-runners	arc-runner-set-rmrgw-runner-p9p5n	1/1
Running	0	21s

Next steps [↗](#)

Actions Runner Controller can help you efficiently manage your GitHub Actions runners. Ready to get started? Here are some helpful resources for taking your next steps with ARC:

- For detailed authentication information, see "[Authenticating to the GitHub API](#)."
- For help using ARC runners in your workflows, see "[Using Actions Runner Controller runners in a workflow](#)."
- For deployment information, see "[Deploying runner scale sets with Actions Runner Controller](#)."

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.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)