Troubleshooting Actions Runner Controller errors

In this article

Learn how to troubleshoot Actions Runner Controller errors.

[Legal notice](#)

Logging 🔗

The Actions Runner Controller (ARC) resources, which include the controller, listener, and runners, write logs to standard output (`stdout`). We recommend you implement a logging solution to collect and store these logs. Having logs available can help you or GitHub support with troubleshooting and debugging. For more information, see [Logging Architecture](#) in the Kubernetes documentation.

Resources labels 🔗

Labels are added to the resources created by Actions Runner Controller, which include the controller, listener, and runner pods. You can use these labels to filter resources and to help with troubleshooting.

Controller pod 🔗

The following labels are applied to the controller pod.

```
app.kubernetes.io/component=controller-manager
app.kubernetes.io/instance=<controller installation name>
app.kubernetes.io/name=gha-runner-scale-set-controller
app.kubernetes.io/part-of=gha-runner-scale-set-controller
app.kubernetes.io/version=<chart version>
```

Listener pod 🔗

The following labels are applied to listener pods.

```
actions.github.com/enterprise= # Will be populated if githubConfigUrl is an
enterprise URL
actions.github.com/organization= # Will be populated if githubConfigUrl is an
organization URL
actions.github.com/repository= # Will be populated if githubConfigUrl is a
repository URL
actions.github.com/scale-set-name= # Runners scale set name
actions.github.com/scale-set-namespace= # Runners namespace
app.kubernetes.io/component=runner-scale-set-listener
app.kubernetes.io/part-of=gha-runner-scale-set
app.kubernetes.io/version= # Chart version
```

Runner pod 🔗

The following labels are applied to runner pods.

```
actions-ephemeral-runner= # True | False
actions.github.com/organization= # Will be populated if githubConfigUrl is an
organization URL
actions.github.com/scale-set-name= # Runners scale set name
actions.github.com/scale-set-namespace= # Runners namespace
app.kubernetes.io/component=runner
app.kubernetes.io/part-of=gha-runner-scale-set
app.kubernetes.io/version= # Chart version
```

Checking the logs of the controller and runner set listener 🔗

To check the logs of the controller pod, you can use the following command.

Bash

```bash
kubectl logs -n <CONTROLLER_NAMESPACE> -l app.kubernetes.io/name=gha-runner-
scale-set-controller
```

To check the logs of the runner set listener, you can use the following command.

Bash

```bash
kubectl logs -n <CONTROLLER_NAMESPACE> -l auto-scaling-runner-set-namespace=arc-
systems -l auto-scaling-runner-set-name=arc-runner-set
```

Using the charts from the `master` branch 🔗

We recommend you use the charts from the latest release instead of the `master` branch.
The `master` branch is highly unstable, and we cannot guarantee that the charts in the
`master` branch will work at any given time.

Troubleshooting the listener pod 🔗

If the controller pod is running, but the listener pod is not, inspect the logs of the
controller first and see if there are any errors. If there are no errors and the runner set
listener pod is still not running, ensure the controller pod has access to the Kubernetes
API server in your cluster.

If you have a proxy configured or you're using a sidecar proxy that's automatically injected, such as [Istio](), ensure it's configured to allow traffic from the controller container (manager) to the Kubernetes API server.

If you have installed the autoscaling runner set, but the listener pod is not created, verify that the `githubConfigSecret` you provided is correct and that the `githubConfigUrl` you provided is accurate. See "[Authenticating to the GitHub API]()" and "[Deploying runner scale sets with Actions Runner Controller]()" for more information.

Runner pods are recreated after a canceled workflow run 🔗

Once a workflow run is canceled, the following events happen.

- The cancellation signal is sent to the runners directly.
- The runner application terminates, which also terminates the runner pods.
- On the next poll, the cancellation signal is received by the listener.

There might be a slight delay between when the runners receive the signal and when the listener receives the signal. When runner pods start terminating, the listener tries to bring up new runners to match the desired number of runners according to the state it's in. However, when the listener receives the cancellation signal, it will act to reduce the number of runners. Eventually the listener will scale back down to the desired number of runners. In the meantime, you may see extra runners.

Error: `Name must have up to n characters` 🔗

ARC uses the generated names of certain resources as labels for other resources. Because of this requirement, ARC limits resource names to 63 characters.

Because part of the resource name is defined by you, ARC imposes a limit on the number of characters you can use for the installation name and namespace.

```
Error: INSTALLATION FAILED: execution error at (gha-runner-scale-
set/templates/autoscalingrunnerset.yaml:5:5): Name must have up to 45 characters

Error: INSTALLATION FAILED: execution error at (gha-runner-scale-
set/templates/autoscalingrunnerset.yaml:8:5): Namespace must have up to 63
characters
```

Error: `Access to the path /home/runner/_work/_tool is denied` 🔗

You may see this error if you're using Kubernetes mode with persistent volumes. This error occurs if the runner container is running with a non-root user and is causing a permissions mismatch with the mounted volume.

To fix this, you can do one of the following things.

- Use a volume type that supports `securityContext.fsGroup`. `hostPath` volumes do not support this property, whereas `local` volumes and other types of volumes do support it. Update the `fsGroup` of your runner pod to match the GID of the runner. You can do this by updating the `gha-runner-scale-set` helm chart values to include the following. Replace `VERSION` with the version of the `actions-runner` container image you want to use.

```
YAML                                                                    ⧉
```

```yaml
spec:
    securityContext:
        fsGroup: 123
    containers:
    - name: runner
    image: ghcr.io/actions/actions-runner:latest
    command: ["/home/runner/run.sh"]
```

- If updating the `securityContext` of your runner pod is not a viable solution, you can work around the issue by using `initContainers` to change the mounted volume's ownership, as follows.

YAML

```yaml
template:
spec:
    initContainers:
    - name: kube-init
    image: ghcr.io/actions/actions-runner:latest
    command: ["sudo", "chown", "-R", "1001:123", "/home/runner/_work"]
    volumeMounts:
        - name: work
        mountPath: /home/runner/_work
    containers:
    - name: runner
    image: ghcr.io/actions/actions-runner:latest
    command: ["/home/runner/run.sh"]
```

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.
```