

Managing access to self-hosted runners using groups

Host your own runners

7 of 8 in learning path

Next: [Monitoring and troubleshooting self-hosted runners](#)

In this article

- About runner groups
- Creating a self-hosted runner group for an organization
- Creating a self-hosted runner group for an enterprise
- Changing which organizations can access a runner group
- Changing which repositories can access a runner group
- Changing which workflows can access a runner group
- Changing the name of a runner group
- Automatically adding a self-hosted runner to a group
- Moving a self-hosted runner to a group
- Removing a self-hosted runner group
- Using unique names for runner groups

You can use policies to limit access to self-hosted runners that have been added to an organization or enterprise.

Who can use this feature

Enterprise accounts, organizations owned by enterprise accounts, and organizations using GitHub Team can create and manage additional runner groups.

Note: GitHub-hosted runners are not currently supported on GitHub Enterprise Server. You can see more information about planned future support on the [GitHub public roadmap](#).

About runner groups

To control access to runners at the organization and/or enterprise levels, enterprise and organization owners can use runner groups.

Runner groups are used to collect sets of runners and create a security boundary around them. You can then decide which organizations or repositories are permitted to run jobs on those sets of machines. Enterprise administrators can configure access policies that control which organizations and workflows in an enterprise have access to the runner group.

When you grant access to a runner group, you can see the runner group listed in the organization's runner settings. Optionally, you can assign additional granular repository and workflow access policies to the runner group.

When new runners are created, they are automatically assigned to the default group unless otherwise specified. Runners can only be in one group at a time. You can move runners from one runner group to another. For more information, see "[Moving a runner to a group](#)."

For information on how to route jobs to runners in a specific group, see "[Choosing the runner for a job](#)."

Creating a self-hosted runner group for an organization


Warning: We recommend that you only use self-hosted runners with private repositories. This is because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.

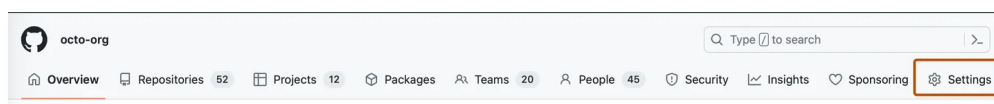
For more information, see "[About self-hosted runners](#)."


All organizations have a single default runner group. Organizations within an enterprise account can create additional groups. Organization admins can allow individual repositories access to a runner group. For information about how to create a runner group with the REST API, see "[Actions](#)."

If no group is specified during the registration process, runners are automatically added to a default group. You can later move the runner from the default group to a custom group. For more information, see "[Moving a runner to a group](#)."

When creating a group, you must choose a policy that defines which repositories and workflows have access to the runner group.

- 1 On your GitHub Enterprise Server instance, navigate to the main page of the organization.
- 2 Under your organization name, click  **Settings**. If you cannot see the "Settings" tab, select the ... dropdown menu, then click **Settings**.



- 3 In the left sidebar, click  **Actions**, then click **Runner groups**.
- 4 In the "Runner groups" section, click **New runner group**.
- 5 Enter a name for your runner group.
- 6 Assign a policy for repository access.

You can configure a runner group to be accessible to a specific list of repositories, or to all repositories in the organization. By default, only private repositories can access runners in a runner group, but you can override this. This setting can't be overridden if configuring an organization's runner group that was shared by an enterprise.

- 7 Assign a policy for workflow access.

You can configure a runner group to be accessible to a specific list of workflows, or to all workflows. This setting can't be overridden if you are configuring an organization's runner group that was shared by an enterprise. If you specify what workflow can access the runner group, you must use the full path to the workflow,

including the repository name and owner, and you must pin the workflow to a branch, tag, or full SHA. For example: `octo-org/octo-repo/.github/workflows/build.yml@v2`, `octo-org/octo-repo/.github/workflows/deploy.yml@d6dc6c96df4f32fa27b039f2084f576ed2c5c2a5`, `monalisa/octo-test/.github/workflows/test.yml@main`.

Only jobs directly defined within the selected workflows will have access to the runner group. Organization-owned runner groups cannot access workflows from a different organization in the enterprise; instead, you must create an enterprise-owned runner group.

- 8 Click **Create group** to create the group and apply the policy.

Creating a self-hosted runner group for an enterprise

Warning: We recommend that you only use self-hosted runners with private repositories. This is because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.

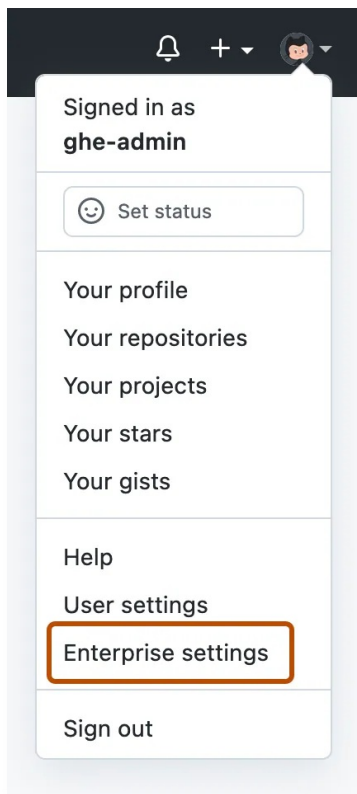
For more information, see "[About self-hosted runners](#)."

Enterprises can add their runners to groups for access management. Enterprises can create groups of runners that are accessible to specific organizations in the enterprise account or to specific workflows. Organization owners can then assign additional granular repository or workflow access policies to the enterprise runner groups. For information about how to create a runner group with the REST API, see the enterprise endpoints in the [GitHub Actions REST API](#).

If no group is specified during the registration process, runners are automatically added to a default group. You can later move the runner from the default group to a custom group. For more information, see "Moving a runner to a group."

When creating a group, you must choose a policy that defines which organizations have access to the runner group.

- 1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click **Policies**.
- 3 Under "Policies", click **Actions**.
- 4 Click the **Runner groups** tab.
- 5 Click **New runner group**.
- 6 Under "Group name", type a name for your runner group.
- 7 To choose a policy for organization access, select the **Organization access** dropdown menu and click a policy. You can configure a runner group to be accessible to a specific list of organizations, or all organizations in the enterprise. By default, only private repositories can access runners in a runner group, but you can override this.
- 8 Assign a policy for workflow access.

You can configure a runner group to be accessible to a specific list of workflows, or to all workflows. This setting can't be overridden if you are configuring an organization's runner group that was shared by an enterprise. If you specify what workflow can access the runner group, you must use the full path to the workflow, including the repository name and owner, and you must pin the workflow to a branch, tag, or full SHA. For example: `octo-org/octo-repo/.github/workflows/build.yml@v2`, `octo-org/octo-repo/.github/workflows/deploy.yml@d6dc6c96df4f32fa27b039f2084f576ed2c5c2a5`, `monalisa/octo-test/.github/workflows/test.yml@main`.

Only jobs directly defined within the selected workflows will have access to the runner group.

- 9 Click **Save group** to create the group and apply the policy.

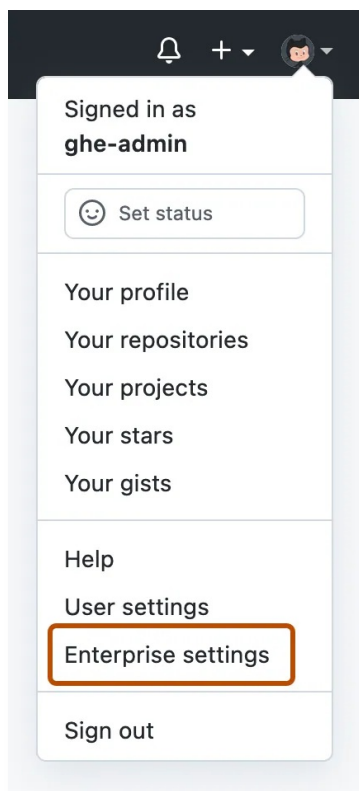
Changing which organizations can access a runner group


Warning: We recommend that you only use self-hosted runners with private repositories. This is because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.

For more information, see "[About self-hosted runners](#)."

For runner groups in an enterprise, you can change what organizations in the enterprise can access a runner group.

- 1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click **Policies**.
- 3 Under "**Policies**", click **Actions**.
- 4 Click the **Runner groups** tab.
- 5 Under "Organization access," use the dropdown menu to click **Selected organizations**.
 - a. To the right of the dropdown menu, click .
 - b. In the popup, use the checkboxes to select organizations that can use this runner group.
- 6 Click **Save group**.

Changing which repositories can access a runner group

Warning: We recommend that you only use self-hosted runners with private repositories. This is

because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.

For more information, see "[About self-hosted runners](#)."

For runner groups in an organization, you can change what repositories in the organization can access a runner group.

- 1 Navigate to the main page of the organization where your runner groups are located.
- 2 Click ⚙️ **Settings**.
- 3 In the left sidebar, click ▶️ **Actions**, then click **Runner groups**.
- 4 In the list of groups, click the runner group you'd like to configure.
- 5 Under "Repository access," use the dropdown menu to click **Selected organizations**.
 - a. To the right of the dropdown menu, click ⚙️.
 - b. In the popup, use the checkboxes to select repositories that can access this runner group.
- 6 Click **Save group**.

Changing which workflows can access a runner group

Warning: We recommend that you only use self-hosted runners with private repositories. This is because forks of your public repository can potentially run dangerous code on your self-hosted runner machine by creating a pull request that executes the code in a workflow.

For more information, see "[About self-hosted runners](#)."

You can configure a runner group to run either selected workflows or all workflows. For example, you might use this setting to protect secrets that are stored on runners or to standardize deployment workflows by restricting a runner group to run only a specific reusable workflow. This setting cannot be overridden if you are configuring an organization's runner group that was shared by an enterprise.

- [Changing which workflows can access an organization runner group](#)
- [Changing which workflows can access an enterprise runner group](#)

Changing which workflows can access an organization runner group

- 1 Navigate to the main page of the organization where your runner groups are located.
- 2 Click ⚙️ **Settings**.
- 3 In the left sidebar, click ▶️ **Actions**, then click **Runner groups**.
- 4 In the list of groups, click the runner group you'd like to configure.

- 5 Under **Workflow access**, select the dropdown menu and click **Selected workflows**.

- 6 Click .

- 7 Enter a comma separated list of the workflows that can access the runner group. Use the full path, including the repository name and owner. Pin the workflow to a branch, tag, or full SHA. For example: `octo-org/octo-repo/.github/workflows/build.yml@v2, octo-org/octo-repo/.github/workflows/deploy.yml@d6dc6c96df4f32fa27b039f2084f576ed2c5c2a5, monalisa/octo-test/.github/workflows/test.yml@main`.

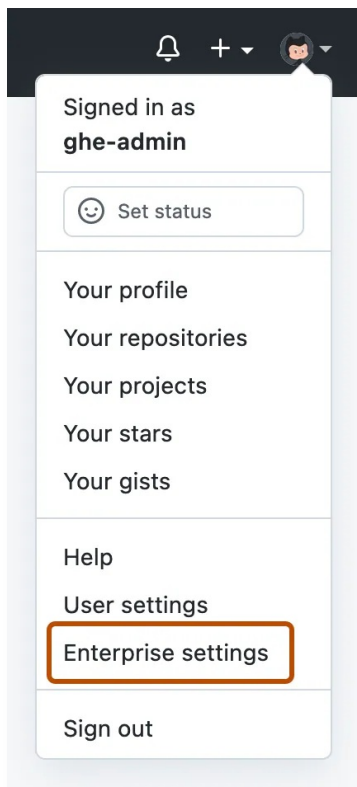
Only jobs directly defined within the selected workflows will have access to the runner group.



Organization-owned runner groups cannot access workflows from a different organization in the enterprise; instead, you must create an enterprise-owned runner group.

- 8 Click **Save**.

Changing which workflows can access an enterprise runner group

- 1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click  **Policies**.
- 3 Under " Policies", click **Actions**.
- 4 Click the **Runner groups** tab.
- 5 In the list of groups, click the runner group you'd like to configure.

- 6 Under **Workflow access**, select the dropdown menu and click **Selected workflows**.

- 7 Click .

- 8 Enter a comma separated list of the workflows that can access the runner group. Use the full path, including the repository name and owner. Pin the workflow to a branch, tag, or full SHA. For example: `octo-org/octo-repo/.github/workflows/build.yml@v2`, `octo-org/octo-repo/.github/workflows/deploy.yml@d6dc6c96df4f32fa27b039f2084f576ed2c5c2a5`, `monalisa/octo-test/.github/workflows/test.yml@main`.

Only jobs directly defined within the selected workflows will have access to the runner group.

Organization-owned runner groups cannot access workflows from a different organization in the enterprise; instead, you must create an enterprise-owned runner group.



- 9 Click **Save**.

Changing the name of a runner group

You can edit the name of your runner groups at the enterprise and organization levels.

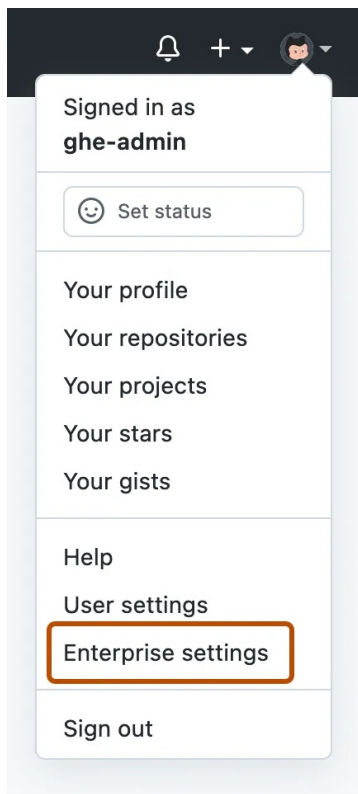
- [Changing the name of an organization runner group](#)
- [Changing the name of an enterprise runner group](#)

Changing the name of an organization runner group

- 1 Navigate to the main page of the organization where your runner groups are located.
- 2 Click  **Settings**.
- 3 In the left sidebar, click  **Actions**, then click **Runner groups**.
- 4 In the list of groups, click the runner group you'd like to configure.
- 5 Enter the new runner group name in the text field under "Group name."
- 6 Click **Save**.

Changing the name of an enterprise runner group

- 1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click **Policies**.
- 3 Under "Policies", click **Actions**.
- 4 Click the **Runner groups** tab.
- 5 In the list of groups, click the runner group you'd like to configure.
- 6 Enter the new runner group name in the text field under "Group name."
- 7 Click **Save**.

Automatically adding a self-hosted runner to a group [↗](#)

You can use the configuration script to automatically add a new runner to a group. For example, this command registers a new runner and uses the `--runnergroup` parameter to add it to a group named `rg-runnergroup`.

```
./config.sh --url $org_or_enterprise_url --token $token --runnergroup rg-runnergroup
```

The command will fail if the runner group doesn't exist:

```
Could not find any self-hosted runner group named "rg-runnergroup".
```

Moving a self-hosted runner to a group [↗](#)

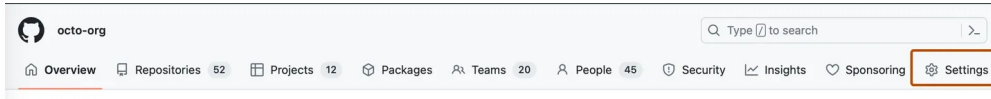
If you don't specify a runner group during the registration process, your new runners are automatically assigned to the default group, and can then be moved to another group.

- [Moving an organization runner to a group](#)

- [Moving an enterprise runner to a group](#)

Moving an organization runner to a group [↗](#)

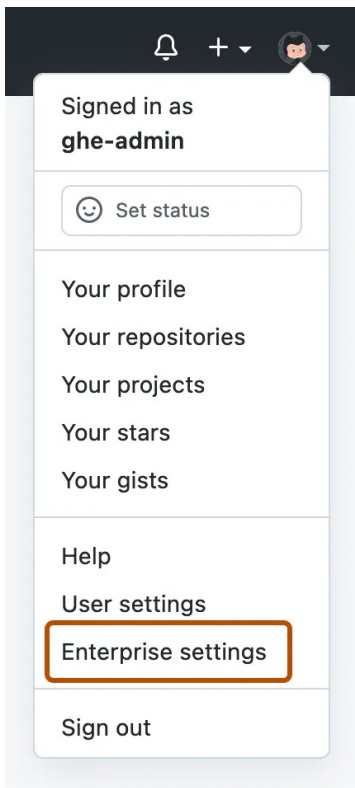
- 1 On your GitHub Enterprise Server instance, navigate to the main page of the organization.
- 2 Under your organization name, click ⚙️ **Settings**. If you cannot see the "Settings" tab, select the ... dropdown menu, then click **Settings**.



- 3 In the left sidebar, click ▶ **Actions**, then click **Runners**.
- 4 In the "Runners" list, click the runner that you want to configure.
- 5 Select the **Runner group** drop-down.
- 6 In "Move runner to group", choose a destination group for the runner.

Moving an enterprise runner to a group [↗](#)

- 1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click ⚙️ **Policies**.
- 3 Under "⚙️ Policies", click **Actions**.
- 4 Click the **Runners** tab.
- 5 In the "Runners" list, click the runner that you want to configure.

- 6 Select the **Runner group** drop-down.
- 7 In "Move runner to group", choose a destination group for the runner.

Removing a self-hosted runner group [🔗](#)

In order to remove a runner group, you must first move or remove all of the runners from the group.

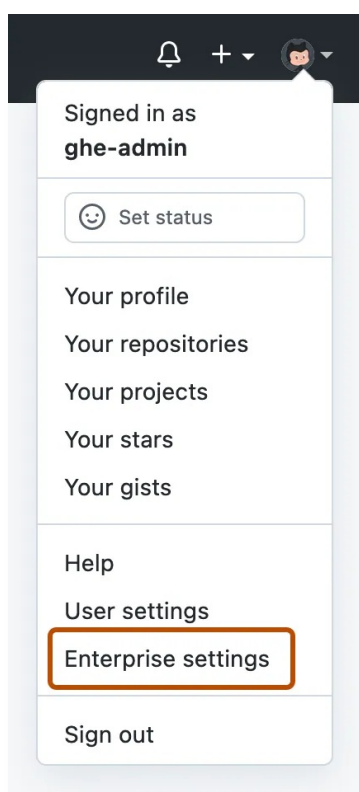
- [Removing a runner group from an organization](#)
- [Removing a runner group from an enterprise](#)



Removing a runner group from an organization [🔗](#)

- 1 Navigate to the main page of the organization where your runner groups are located.
- 2 Click ⚙️ **Settings**.
- 3 In the left sidebar, click ▶️ **Actions**, then click **Runner groups**.
- 4 In the list of groups, to the right of the group you want to delete, click ⋮.
- 5 To remove the group, click **Remove group**.
- 6 Review the confirmation prompts, and click **Remove this runner group**.

Removing a runner group from an enterprise [🔗](#)

- 1 In the top-right corner of GitHub Enterprise Server, click your profile photo, then click **Enterprise settings**.



- 2 In the enterprise account sidebar, click  **Policies**.
- 3 Under " Policies", click **Actions**.
- 4 Click the **Runner groups** tab.
- 5 In the list of groups, to the right of the group you want to delete, click **⋮**.
- 6 To remove the group, click **Remove group**.
- 7 Review the confirmation prompts, and click **Remove this runner group**.

Using unique names for runner groups

GitHub Actions requires that runner group names must be unique at the organization level. This means that an organization will no longer be able to create a runner group with the same name as one in the enterprise. In addition, users will see a warning banner on any runner groups that share the same name as a group in the enterprise, suggesting that the organization group should be renamed.

To avoid ambiguity, a workflow will fail if there are duplicate runner groups in the organization and enterprise. To address this, you can either rename one of your runner groups in the organization or enterprise, or you can update your workflow file to add a prefix to the runner group name:

- `org/` or `organization/`
- `ent/` or `enterprise/`

Example: Using prefixes to differentiate runner groups

For example, if you have a runner group named `my-group` in the organization and another named `my-group` in the enterprise, you can update your workflow file to use `org/my-group` or `ent/my-group` to differentiate between the two.

Using `org/` :

```
runs-on:
  group: org/my-group
  labels: [ self-hosted, label-1 ]
```

Using `ent/` :

```
runs-on:
  group: ent/my-group
  labels: [ self-hosted, label-1 ]
```

Previous
[Using self-hosted runners in a workflow](#)

Next
[Monitoring and troubleshooting self-hosted runners](#)

Legal