

# Controlling access to larger runners

## In this article

- About runner groups
  - Creating a runner group for an organization
  - Changing which repositories can access a runner group
  - Changing the name of a runner group
  - Moving a runner to a group
  - Removing a runner group
- 

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

### Who can use this feature

Larger runners are only available for organizations and enterprises using the GitHub Team or GitHub Enterprise Cloud plans.

**Note:** The information and instructions in this article only apply to larger runners with Linux and Windows operating systems.

## About runner groups

---

To control access to runners at the organization level, organizations using the GitHub Team plan 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. Organization owners can configure access policies that control which repositories in an organization 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 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](#)."

## Managing access to your runners

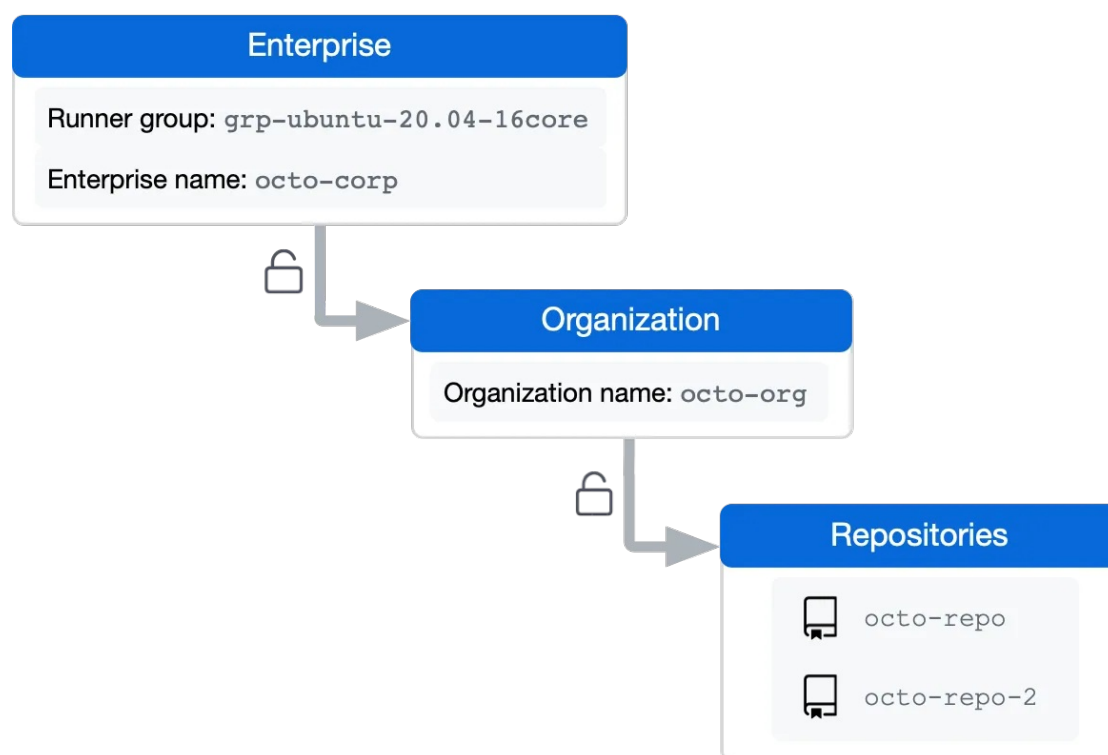
**Note:** Before your workflows can send jobs to larger runners, you must first configure permissions for the runner group. See the following sections for more information.

Runner groups are used to control which repositories can run jobs on your larger

runners. You must manage access to the group from each level of the management hierarchy, depending on where you've defined the larger runner:

- **Runners at the enterprise level:** By default, repositories in an organization do not have access to enterprise-level runner groups. To give repositories access to enterprise runner groups, organization owners must configure each enterprise runner group and choose which repositories have access.
- **Runners at the organization level:** By default, all repositories in an organization are granted access to organization-level runner groups. To restrict which repositories have access, organization owners must configure organization runner groups and choose which repositories have access.

For example, the following diagram has a runner group named `grp-ubuntu-20.04-16core` at the enterprise level. Before the repository named `octo-repo` can use the runners in the group, you must first configure the group at the enterprise level to allow access to the `octo-org` organization. You must then configure the group at the organization level to allow access to `octo-repo`.



## Creating a runner group for an organization [↗](#)

**Warning:** If you are using a Fixed IP range, we recommend that you only use larger runners with private repositories. Forks of your repository can potentially run dangerous code on your larger runner by creating a pull request that executes the code in a workflow.

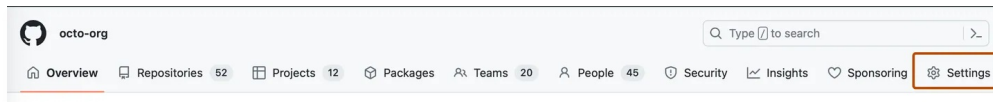
All organizations have a single default runner group. Organizations using the GitHub Team plan can create additional groups. Organization admins can allow individual repositories access to a runner group. For information about how to create a runner group with the REST API, see "[Actions](#)."

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 have access to the runner group.

- 1 On GitHub.com, 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 Click **Create group** to create the group and apply the policy.

## Changing which repositories can access a runner group [↗](#)

**Warning:** If you are using a Fixed IP range, we recommend that you only use larger runners with private repositories. Forks of your repository can potentially run dangerous code on your larger runner by creating a pull request that executes the code in a workflow.

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**.

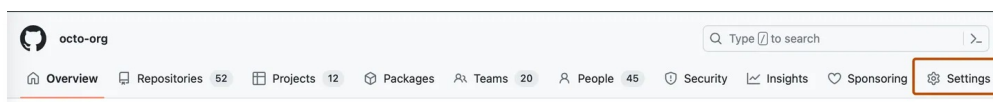
## Changing the name of a runner group [↗](#)

- 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 Enter the new runner group name in the text field under "Group name."
- 6 Click **Save**.

## Moving a 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.

- 1 On GitHub.com, 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.

## Removing a runner group [🔗](#)

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

### Legal

