

# Disabling or limiting GitHub Actions for your organization

## In this article

- About GitHub Actions permissions for your organization
- Managing GitHub Actions permissions for your organization
- Limiting the use of self-hosted runners
- Adding a required workflow to an organization
- Enabling workflows for private repository forks
- Setting the permissions of the GITHUB\_TOKEN for your organization
- Managing GitHub Actions cache storage for your organization

Organization owners can disable, enable, and limit GitHub Actions for an organization.

**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 GitHub Actions permissions for your organization

By default, after GitHub Actions is enabled on your GitHub Enterprise Server instance, it is enabled on all repositories and organizations. You can choose to disable GitHub Actions or limit it to actions in your enterprise. For more information about GitHub Actions, see "[Learn GitHub Actions](#)."

You can enable GitHub Actions for all repositories in your organization. When you enable GitHub Actions, workflows are able to run actions located within your repository and any other public or internal repository. You can disable GitHub Actions for all repositories in your organization. When you disable GitHub Actions, no workflows run in your repository.

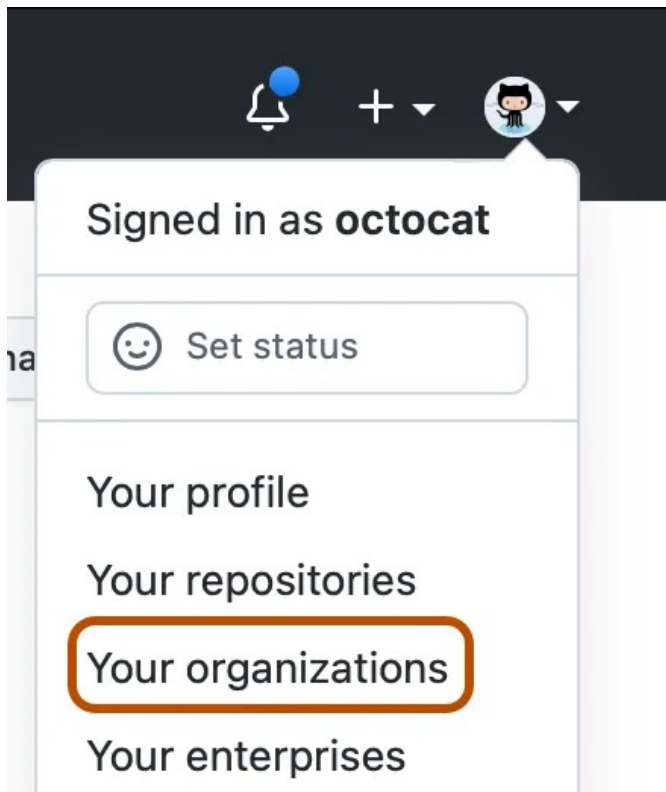
Alternatively, you can enable GitHub Actions for all repositories in your organization but limit the actions a workflow can run.

## Managing GitHub Actions permissions for your organization

You can choose to disable GitHub Actions for all repositories in your organization, or only allow specific repositories. You can also limit the use of public actions, so that people can only use local actions that exist in your enterprise.

**Note:** You might not be able to manage these settings if your organization is managed by an enterprise that has overriding policy. For more information, see "[Enforcing policies for GitHub Actions in your enterprise](#)."

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




- 2 Next to the organization, click **Settings**.
- 3 In the left sidebar, click **Actions**, then click **General**.
- 4 Under "Policies", select an option.

If you choose **Allow select actions**, actions within your enterprise are allowed, and there are additional options for allowing other specific actions. For more information, see "[Allowing select actions to run](#)."

- 5 Click **Save**.

## Allowing select actions to run [↗](#)

When you choose **Allow select actions**, local actions are allowed, and there are additional options for allowing other specific actions:

- **Allow actions created by GitHub:** You can allow all actions created by GitHub to be used by workflows. Actions created by GitHub are located in the `actions` and `github` organizations. For more information, see the [actions](#) and [github](#) organizations.
- **Allow Marketplace actions by verified creators:** This option is available if you have GitHub Connect enabled and configured with GitHub Actions. For more information, see "[Enabling automatic access to GitHub.com actions using GitHub Connect](#)." You can allow all GitHub Marketplace actions created by verified creators to be used by workflows. When GitHub has verified the creator of the action as a partner organization, the  badge is displayed next to the action in GitHub Marketplace.
- **Allow specified actions:** You can restrict workflows to use actions in specific organizations and repositories. Specified actions cannot be set to more than 1000.

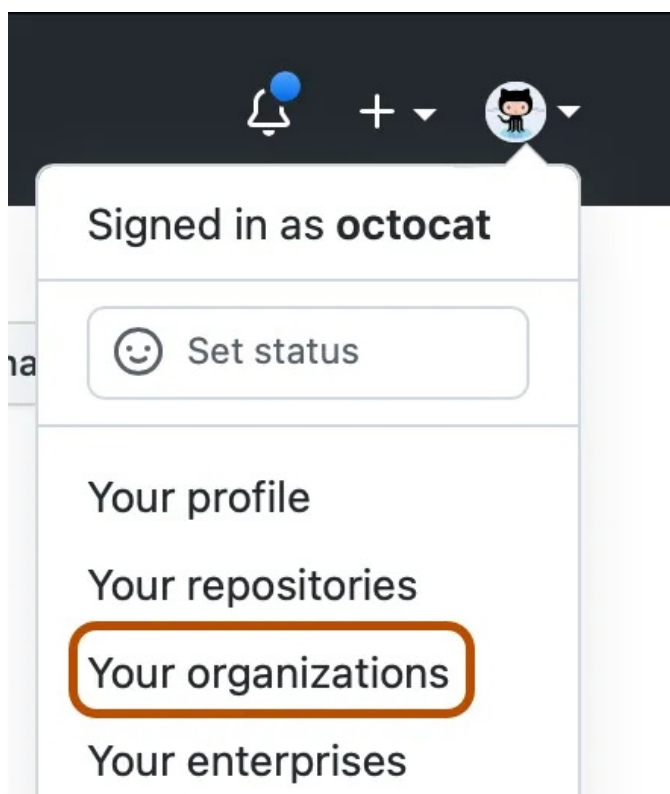
To restrict access to specific tags or commit SHAs of an action, use the same syntax used in the workflow to select the action.

- For an action, the syntax is `OWNER/REPOSITORY@TAG-OR-SHA`. For example, use `actions/javascript-action@v1.0.1` to select a tag or `actions/javascript-action@a824008085750b8e136effc585c3cd6082bd575f` to select a SHA. For more information, see "[Finding and customizing actions](#)."

You can use the `*` wildcard character to match patterns. For example, to allow all actions in organizations that start with `space-org`, you can specify `space-org/*/*`. To allow all actions in repositories that start with `octocat`, you can use `*/octocat**@*`. For more information about using the `*` wildcard, see "[Workflow syntax for GitHub Actions](#)."

This procedure demonstrates how to add specific actions to the allow list.

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



- 2 Next to the organization, click **Settings**.
- 3 In the left sidebar, click **Actions**, then click **General**.
- 4 Under "Policies", select **Allow select actions** and add your required actions to the list.
- 5 Click **Save**.

## Limiting the use of self-hosted runners [↗](#)

There is no guarantee that self-hosted runners for GitHub Enterprise Server will be hosted on ephemeral, clean virtual machines. As a result, they may be compromised by untrusted code in a workflow.

Similarly, anyone who can fork the repository and open a pull request (generally those with read access to the repository) can compromise the self-hosted runner environment,

including gaining access to secrets and the `GITHUB_TOKEN` which, depending on its settings, can grant write access to the repository. Although workflows can control access to environment secrets by using environments and required reviews, these workflows are not run in an isolated environment and are still susceptible to the same risks when run on a self-hosted runner.

For these and other reasons, you may decide to prevent people creating self-hosted runners at the repository level.




**Note:** If your organization belongs to an enterprise, creation of self-hosted runners at the repository level may have been disabled as an enterprise-wide setting. If this has been done, you cannot enable repository-level self-hosted runners in your organization settings. For more information, see "[Enforcing policies for GitHub Actions in your enterprise](#)."

If a repository already has self-hosted runners when you disable their use, these will be listed with the status "Disabled" and they will not be assigned any new workflow jobs.

## Runners

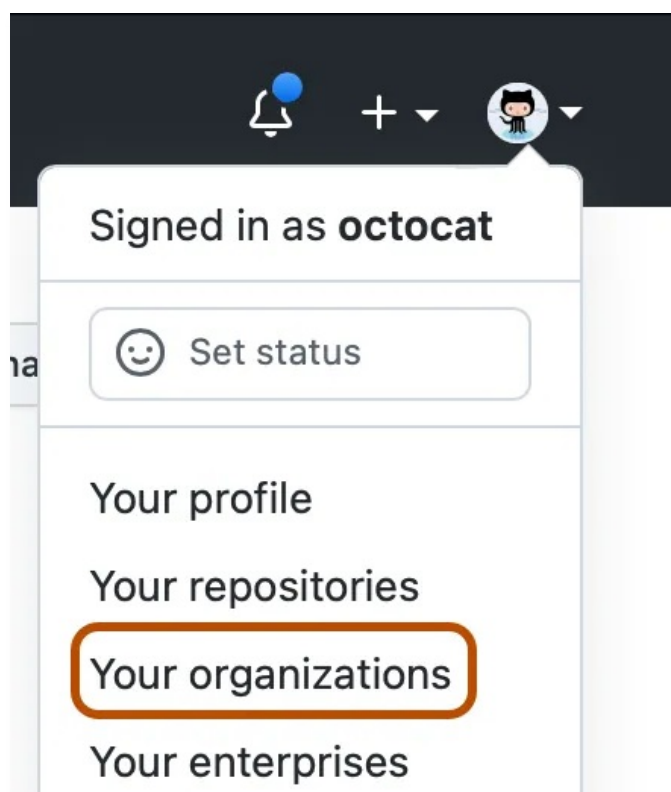
Host your own runners and customize the environment used to run jobs in your GitHub Actions workflows. [Learn more about self-hosted runners](#).

Self-hosted runners were disabled by your organization admin.



1 runner	Status
 Self-hosted runner self-hosted-runner	 Disabled 

**Note:** When creation of repository-level self-hosted runners is disabled, workflows can still access self-hosted runners that have been set up at the enterprise or organization level.

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



- 2 Next to the organization, click **Settings**.

- 3 In the left sidebar, click  **Actions**, then click **General**.
- 4 Under "Runners," use the dropdown menu to choose your preferred setting:
  - **All repositories** - self-hosted runners can be used for any repository in your organization.
  - **Selected repositories** - self-hosted runners can only be used for the repositories you select.
  - **Disabled** - self-hosted runners cannot be created at the repository level.
- 5 If you choose **Selected repositories**:
  - a. Click .
  - b. Select the check boxes for the repositories for which you want to allow self-hosted runners.
  - c. Click **Select repositories**.

## Adding a required workflow to an organization

**Note:** GitHub is deprecating support for required workflows for GitHub Actions.

You can configure required workflows to run in all or selected repositories in an organization where you are an owner. Required workflows are triggered by `pull_request` and `pull_request_target` default events and must pass before a pull request can be merged. For more information, see "[Required workflows](#)."

### Prerequisites

Before configuring a required workflow, note the following prerequisites:

- GitHub Actions must be enabled for a repository in the organization's settings in order for required workflows to run. Once enabled at an organization-level, required workflows will run even when GitHub Actions is disabled in the repository's settings. For more information on managing GitHub Actions in your organization's repositories, see "[Disabling or limiting GitHub Actions for your organization](#)."
- Required workflows are available for organizations and only in repositories where the organization's plan supports required status checks. If required status checks are not supported, the workflow will still run, but it will not be a required check and will not block merging. For more information about support for required status checks, see "[About protected branches](#)."
- The repository's default branch must match the organization's default branch setting in order for required workflows to run as required status checks. If the default branch names do not match, the workflow will still run, but it will not be a required check. For more information about managing default branch names, see "[Managing the default branch name for repositories in your organization](#)" and "[Changing the default branch](#)."
- For required workflows to run, the pull request's source repository must be in the same organization as the target repository. GitHub Enterprise Server will source the required workflow from a specified branch, tag, or commit SHA from the repository containing the workflow.
- Secrets used in a required workflow should be created at either the organization level or in the target repositories.

- Secrets in the source repository will not be fetched when a workflow runs in the target repository.
- When a workflow is run as a required workflow it will ignore all the filters in the `on:` section, for example: `branches`, `branches-ignore`, `paths`, `types` etc. The required workflow will run only for the `pull_request` and `pull_request_target` default events. For more information on default activity types, see "[Events that trigger workflows](#)."
- Required workflows are not automatically triggered on already existing pull requests even though they automatically appear as expected checks. To trigger required workflows for an already existing pull request, push a new change to that pull request.

## Restrictions and behaviors for the source repository

Note the following restrictions and behaviors for the source repository and workflow:

- Required workflows can be stored in any repository folder and are not restricted to the `.github/workflows` folder like normal workflows. If a required workflow calls a reusable workflow, the reusable workflow must be stored in the `.github/workflows` folder. When calling a reusable workflow, a required workflow must use the full path and ref to the reusable workflow. For example,  
`{owner}/{repo}/.github/workflows/{filename}@{ref}`.
- If the required workflow is contained in a private or internal repository, you must ensure that workflows within the repository are accessible by other repositories in your organization. For more information, see "[Managing GitHub Actions settings for a repository](#)" and "[Managing GitHub Actions settings for a repository](#)."
- Workflows stored in a public repository can be configured as required workflows for any repository in your organization. Workflows stored in a private repository can only be configured as required workflows for other private repositories in your organization. Workflows stored in internal repositories can be configured as required workflows for internal and private repositories in your organization.
- CodeQL is not supported in required workflows because CodeQL requires configuration at the repository level. For information on configuring code scanning, see "[Configuring advanced setup for code scanning](#)."
- Required workflows can be referenced using any branch, tag, or commit SHA from the repository containing the workflow file.

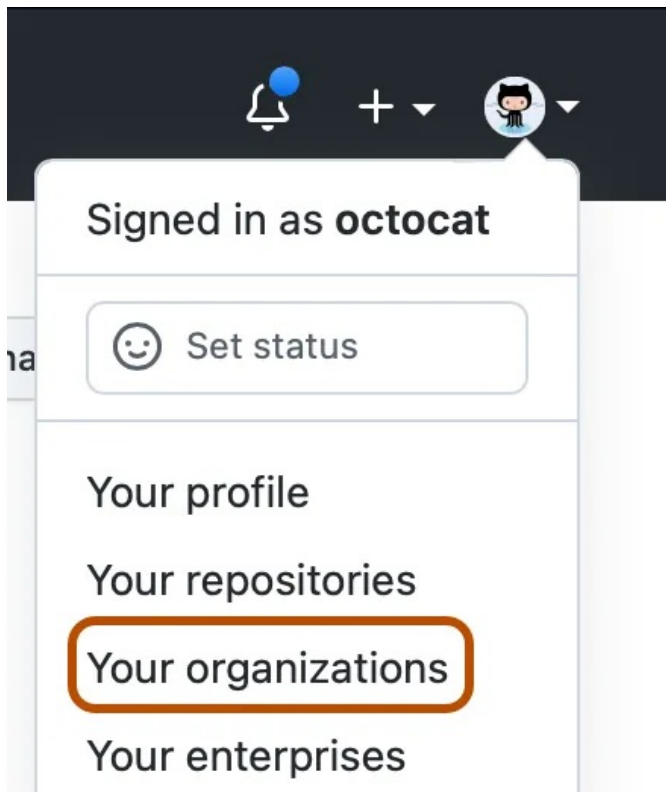
## Restrictions and behaviors for the target repository


Note the following restrictions and behaviors for the target repositories:

- When configuring a required workflow to run on all or selected repositories, the visibility of the repository containing the required workflow will affect which repositories in your organization the workflow runs on. Required workflows stored in public repositories will run on all repositories. Required workflows stored in private repositories will only run on other private repositories. Required workflows stored in internal repositories will run on internal and private repositories.
- Required workflows cannot be configured to run in the repository the workflow is created in. You should consider creating a separate repository to store your required workflows.
- When configuring a required workflow to run on all or selected repositories, required workflows will not run in repositories where actions is disabled in the organization settings.

## Configuring a required workflow for your organization

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



- 2 Next to the organization, click **Settings**.
- 3 In the left sidebar, click **Actions**, then click **General**.
- 4 To the right of "Required Workflows", click **Add workflow**.
- 5 Under "Required workflow", use the drop-down menu to select the repository that contains the workflow. Then, enter the path to the workflow in the text field. You can reference any branch, tag, or commit SHA from the repository containing the workflow file using the `{path}@{ref}` syntax.
- 6 Optionally, to specify target branches on which to enforce the required workflow, enter the branch or multiple branches in the text field under "Target branches". If you do not enter a target branch, the required workflow will be enforced on the default branch for the repository.
- 7 Under "Apply to repositories...", use the drop-down menu to select which repositories the required workflow applies to. Select **All repositories** to apply the required workflow to all repositories in your organization, or **Selected repositories** to choose which repositories it will apply to.
- 8 Optionally, if you chose "Selected repositories", click  to open the repository selection modal, then use the checkboxes to select the repositories, and click **Apply selection**. You can use filters to narrow down your search.
- 9 To add the required workflow, click **Add workflow**.

## Enabling workflows for private repository forks

If you rely on using forks of your private repositories, you can configure policies that control how users can run workflows on `pull_request` events. Available to private and internal repositories only, you can configure these policy settings for your enterprise,

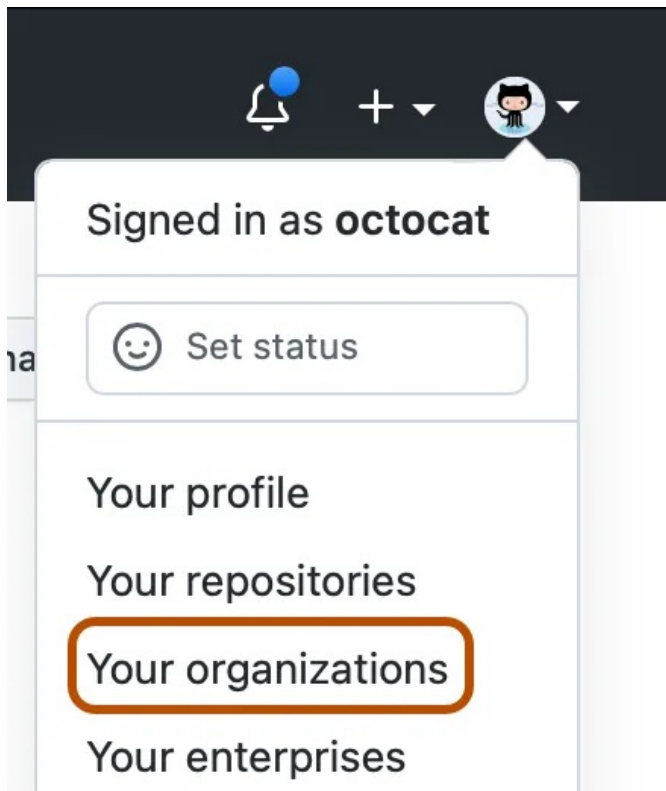
organizations, or repositories.

If a policy is disabled for an enterprise, it cannot be enabled for organizations. If a policy is disabled for an organization, it cannot be enabled for repositories. If an organization enables a policy, the policy can be disabled for individual repositories.

- **Run workflows from fork pull requests** - Allows users to run workflows from fork pull requests, using a `GITHUB_TOKEN` with read-only permission, and with no access to secrets.
- **Send write tokens to workflows from pull requests** - Allows pull requests from forks to use a `GITHUB_TOKEN` with write permission.
- **Send secrets to workflows from pull requests** - Makes all secrets available to the pull request.
- **Require approval for fork pull request workflows** - Workflow runs on pull requests from collaborators without write permission will require approval from someone with write permission before they will run.

## Configuring the private fork policy for an organization [🔗](#)

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



- 2 Next to the organization, click **Settings**.
- 3 In the left sidebar, click **Actions**, then click **General**.
- 4 Under **Fork pull request workflows**, select your options.
- 5 Click **Save** to apply the settings.

## Setting the permissions of the `GITHUB_TOKEN` for your organization [🔗](#)



You can set the default permissions granted to the `GITHUB_TOKEN` . For more information about the `GITHUB_TOKEN` , see "[Automatic token authentication](#)." You can choose a restricted set of permissions as the default, or apply permissive settings.

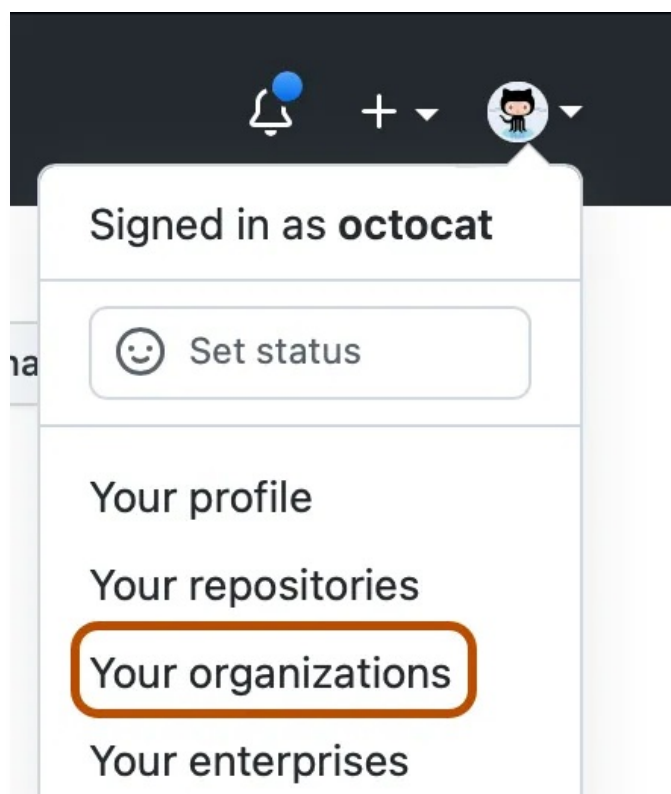
You can set the default permissions for the `GITHUB_TOKEN` in the settings for your organization or your repositories. If you select a restrictive option as the default in your organization settings, the same option is selected in the settings for repositories within your organization, and the permissive option is disabled. If your organization belongs to a GitHub Enterprise account and a more restrictive default has been selected in the enterprise settings, you won't be able to select the more permissive default in your organization settings.

Anyone with write access to a repository can modify the permissions granted to the `GITHUB_TOKEN` , adding or removing access as required, by editing the `permissions` key in the workflow file. For more information, see [permissions](#) .

## Configuring the default `GITHUB_TOKEN` permissions [↗](#)

By default, when you create a new organization, the setting is inherited from what is configured in the enterprise settings.

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



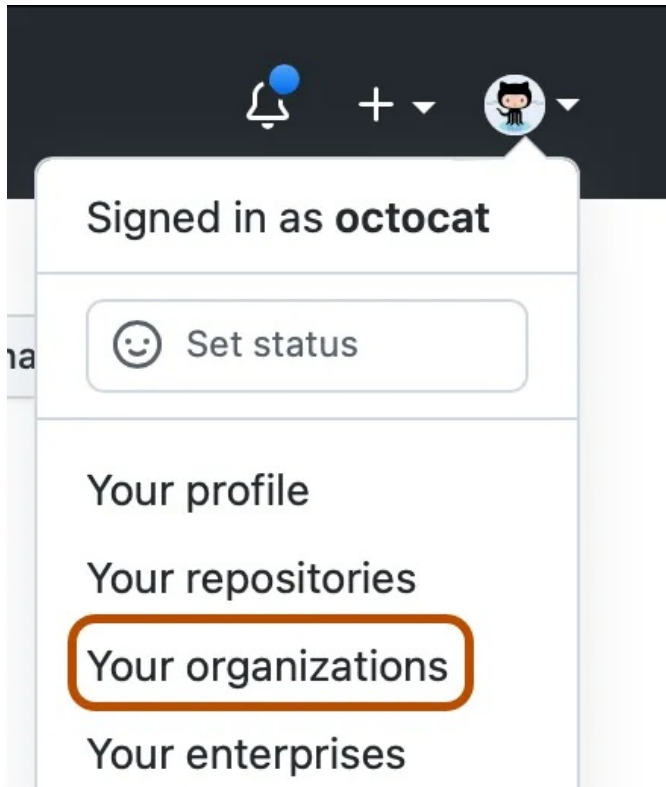
- 3 Next to the organization, click **Settings**.
- 4 In the left sidebar, click **Actions**, then click **General**.
- 5 Under "Workflow permissions", choose whether you want the `GITHUB_TOKEN` to have read and write access for all scopes (the permissive setting), or just read access for the `contents` and `packages` scopes (the restricted setting).
- 6 Click **Save** to apply the settings.

## Preventing GitHub Actions from creating or approving pull requests [🔗](#)

You can choose to allow or prevent GitHub Actions workflows from creating or approving pull requests.

By default, when you create a new organization, workflows are not allowed to create or approve pull requests.

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



- 3 Next to the organization, click **Settings**.
- 4 In the left sidebar, click **Actions**, then click **General**.
- 5 Under "Workflow permissions", use the **Allow GitHub Actions to create and approve pull requests** setting to configure whether `GITHUB_TOKEN` can create and approve pull requests.
- 6 Click **Save** to apply the settings.

## Managing GitHub Actions cache storage for your organization [🔗](#)

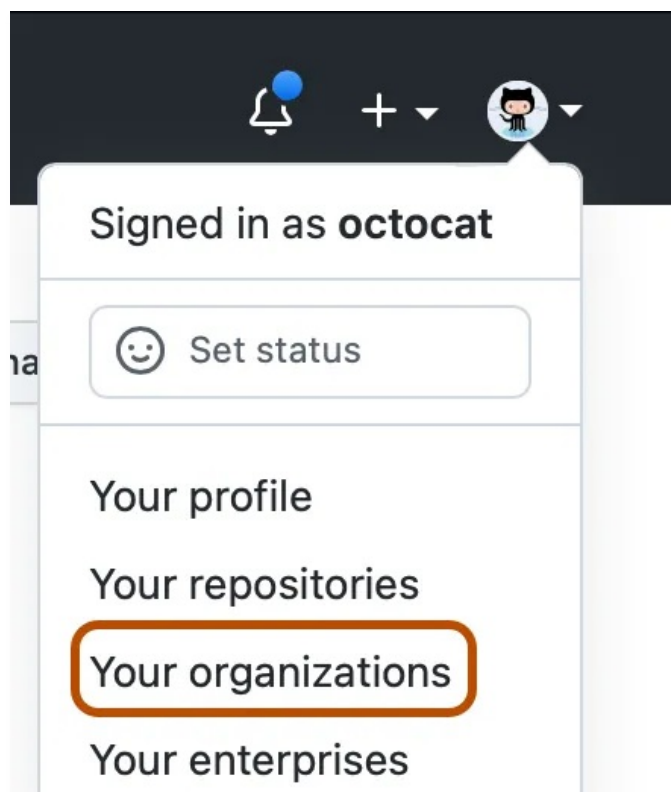
Organization administrators can view and manage GitHub Actions cache storage for all repositories in the organization.

### Viewing GitHub Actions cache storage by repository [🔗](#)

For each repository in your organization, you can see how much cache storage a

repository is using, the number of active caches, and if a repository is near the total cache size limit. For more information about the cache usage and eviction process, see "[Caching dependencies to speed up workflows](#)."

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



- 3 Next to the organization, click **Settings**.
- 4 In the left sidebar, click **Actions**, then click **Caches**.
- 5 Review the list of repositories for information about their GitHub Actions caches. You can click on a repository name to see more detail about the repository's caches.

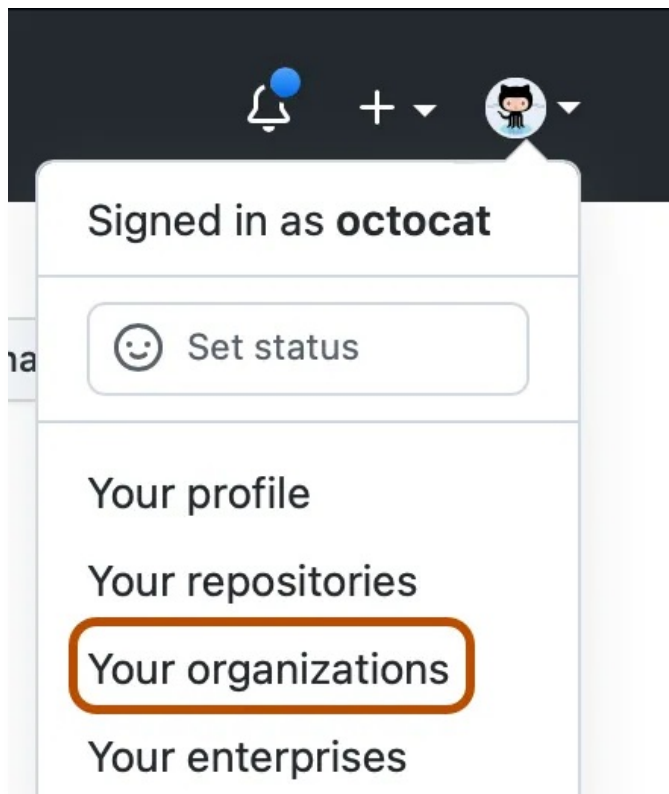
## Configuring GitHub Actions cache storage for your organization



By default, the total cache storage that GitHub Actions uses on the external storage for your GitHub Enterprise Server instance is limited to a maximum of 10 GB per repository, and the maximum allowed size that can be set for a repository is 25 GB.

You can configure the size limit for GitHub Actions caches that will apply to each repository in your organization. The cache size limit for an organization cannot exceed the cache size limit set in the enterprise policy. Repository admins will be able to set a smaller limit in their repositories.

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



- 3 Next to the organization, click **Settings**.
- 4 In the left sidebar, click ▶ **Actions**, then click **General**.
- 5 Under **Cache size limit**, enter a new value.
- 6 Click **Save** to apply the change.

## Legal

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