Enforcing policies for GitHub Actions in your enterprise

Configure GitHub Actions
2 of 6 in learning path

Next: **Enabling automatic access to GitHub.com actions using GitHub Connect**

In this article

You can enforce policies for GitHub Actions within your enterprise's organizations, or allow policies to be set in each organization.

> **Who can use this feature**
> Enterprise owners can enforce policies for GitHub Actions in an enterprise.

About policies for GitHub Actions in your enterprise

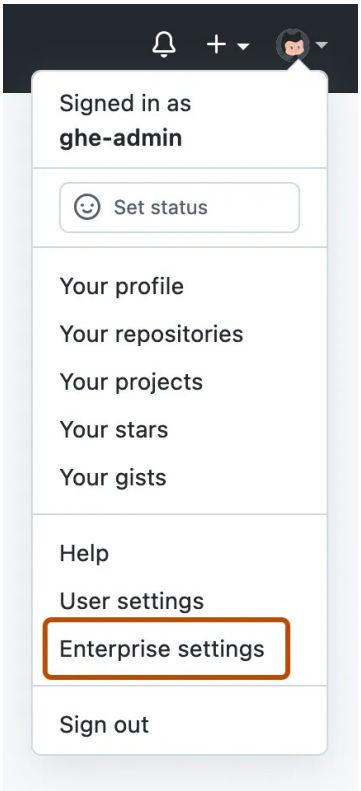
GitHub Actions helps members of your enterprise automate software development workflows on GitHub Enterprise Server. For more information, see "Understanding GitHub Actions."

If you enable GitHub Actions, any organization on your GitHub Enterprise Server instance can use GitHub Actions. You can enforce policies to control how members of your enterprise on GitHub Enterprise Server use GitHub Actions. By default, organization owners can manage how members use GitHub Actions. For more information, see "Disabling or limiting GitHub Actions for your organization."

Enforcing a policy to restrict the use of GitHub Actions in your enterprise

You can choose to disable GitHub Actions for all organizations in your enterprise, or only allow specific organizations. You can also limit the use of public actions , so that people can only use local actions that exist in your 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. Under "Policies", select your options.

   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."

   > **Note:** To enable access to public actions, you must first configure your GitHub Enterprise Server instance to connect to GitHub.com. For more information, see "Enabling automatic access to GitHub.com actions using GitHub Connect."

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.

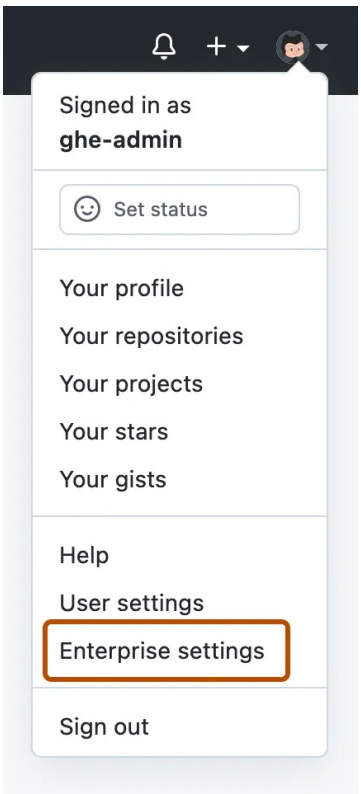
  To restrict access to specific tags or commit SHAs of an action, use the same syntax used in the workflow to select the action.

  - 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 **Enterprise settings**.



2. In the enterprise account sidebar, click ⚖ **Policies**.
3. Under "⚖ Policies", click **Actions**.
4. Under "Policies", select **Allow select actions** and add your required actions to the list.

Disabling repository-level 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.

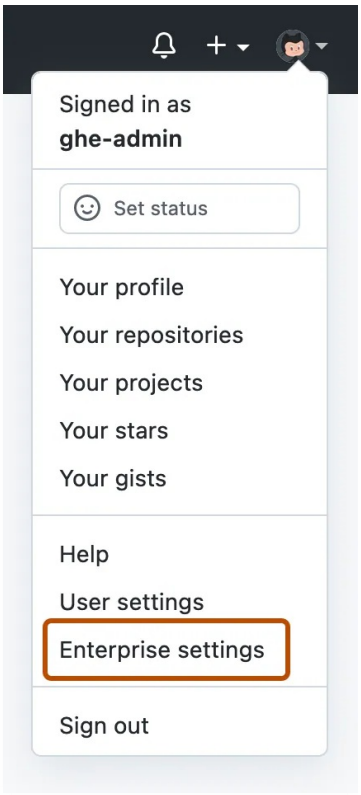
Similarly, anyone who can fork the repository and open a pull request (generally those with read access to the repository) can compromise the self-hosted runner environment, including gaining access to secrets and the `GITHUB_TOKEN` which, depending on its settings, can grant write access to the repository. 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. For more information on creating self-hosted runners at the repository level, see "Adding self-hosted runners."

By default anyone with admin access to a repository can add a self-hosted runner for the repository. The enterprise settings allow you to disable the use of repository-level self-hosted runners across all repositories in your enterprise. If you allow repository-level self-hosted runners for your enterprise, organization owners can choose to allow or prevent creation of repository-level self-hosted runners for some or all repositories in their organization. For more information see, "Disabling or limiting GitHub Actions for your organization."

> **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 **Enterprise settings**.



2. In the enterprise account sidebar, click ⚖ **Policies**.
3. Under "⚖ Policies", click **Actions**.
4. In the "Runners" section, select **Disable for all organizations**.
5. Click **Save** to apply the change.

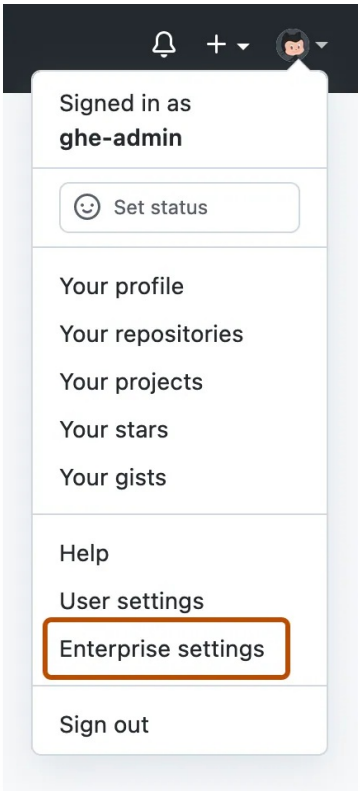
Enforcing a policy for artifact and log retention in your enterprise

GitHub Actions can store artifact and log files. For more information, see "Downloading workflow artifacts."

By default, the artifacts and log files generated by workflows are retained for 90 days before they are automatically deleted. You can change this retention period to anywhere between 1 day or 400 days.

When you customize the retention period, it only applies to new artifacts and log files, and does not retroactively apply to existing objects. For managed repositories and organizations, the maximum retention period cannot exceed the limit set by the managing organization or 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. In the "Artifact, log, and cache settings" section, under **Artifact and log retention**, enter a new value.
5. Click **Save** to apply the change.

Enforcing a policy for fork pull requests in your enterprise

You can enforce policies to control how GitHub Actions behaves for your GitHub Enterprise Server instance when members of your enterprise run workflows from forks.

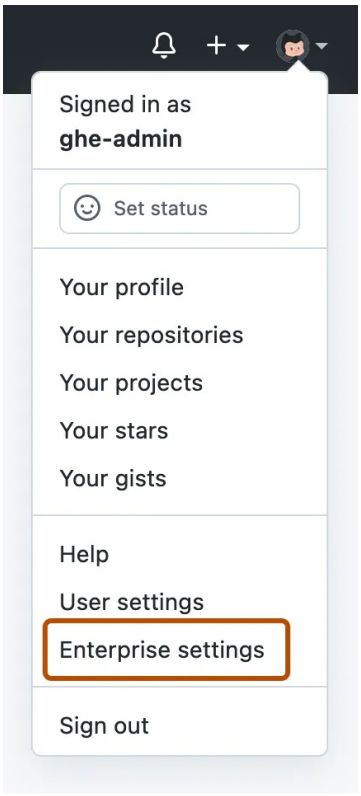
Enforcing a policy for fork pull requests in private repositories

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 enabled for an enterprise, the policy can be selectively disabled in individual organizations or repositories. If a policy is disabled for an enterprise, individual organizations or repositories cannot enable it.

- **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.

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. Under **Fork pull request workflows**, select your options.
5. Click **Save** to apply the settings.

Enforcing a policy for workflow permissions in your enterprise

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.

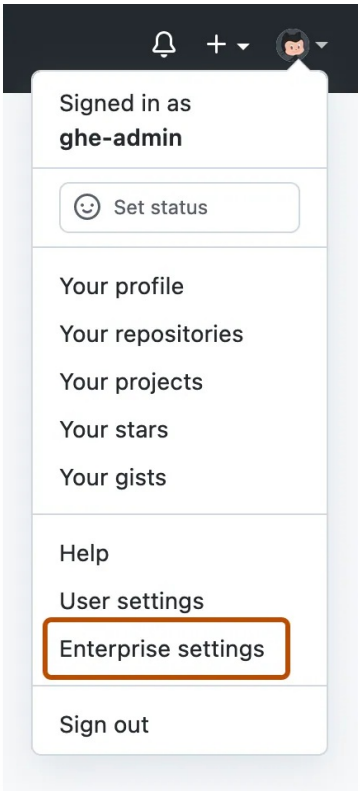
You can set the default permissions for the `GITHUB_TOKEN` in the settings for your enterprise, organizations, or repositories. If you choose a restricted option as the default in your enterprise settings, this prevents the more permissive setting being chosen in the organization or repository settings.

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 enterprise, `GITHUB_TOKEN` only has read access for the `contents` and `packages` scopes.

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



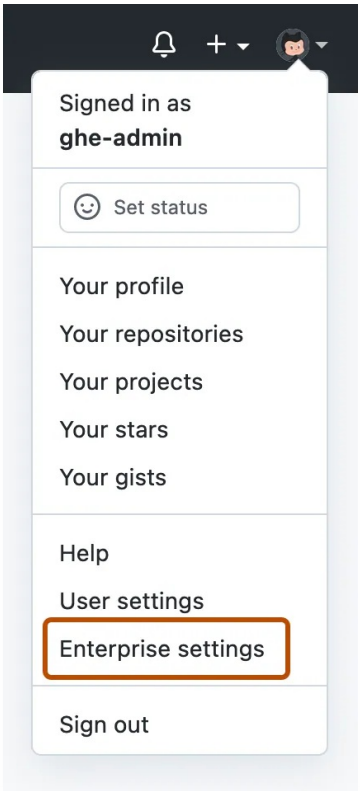
2. In the enterprise account sidebar, click ⚖ **Policies**.
3. Under "⚖ Policies", click **Actions**.
4. Under "Workflow permissions", choose whether you want the `GITHUB_TOKEN` to have read and write access for all scopes (the permissive setting), or just read access for the `contents` and `packages` scopes (the restricted setting).
5. 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 enterprise, 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 **Enterprise settings**.



2. In the enterprise account sidebar, click ⚖ **Policies**.
3. Under "⚖ Policies", click **Actions**.
4. 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.
5. Click **Save** to apply the settings.

Enforcing a policy for cache storage in your enterprise

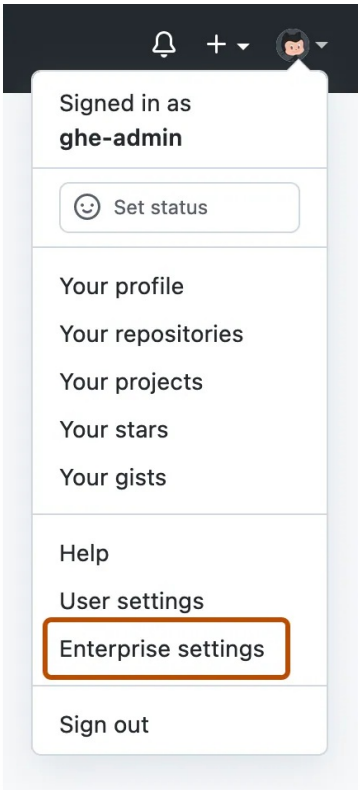
By default, the total cache storage that GitHub Actions uses on the external storage for your GitHub Enterprise Server instance is limited to a maximum of 10 GB per repository, and the maximum allowed size that can be set for a repository is 25 GB. If you exceed the limit, GitHub will save the new cache but will begin evicting caches until the total size is less than the repository limit.

However, you can set an enterprise policy to customize both the default total cache size for each repository, as well as the maximum total cache size allowed for a repository. For example, you might want the default total cache size for each repository to be 5 GB, but

also allow organization owners and repository administrators to configure a total cache size up to 15 GB if necessary.

Organization owners can set a lower total cache size that applies to each repository in their organization. People with admin access to a repository can set a total cache size for their repository up to the maximum cache size allowed by the enterprise or organization policy setting.

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. In the "Artifact, log, and cache settings" section, under **Maximum cache size limit**, enter a value, then click **Save** to apply the setting.
5. In the "Artifact, log, and cache settings" section, under **Default cache size limit**, enter a value, then click **Save** to apply the setting.

Previous
Getting started with GitHub Actions for GitHub Enterprise Server

Next
Enabling automatic access to GitHub.com actions using GitHub Connect