

# Enforcing repository management policies in your enterprise

## In this article

- About policies for repository management in your enterprise
- Configuring the default visibility of new repositories
- Enforcing a policy for base repository permissions
- Enforcing a policy for repository creation
- Enforcing a policy for forking private or internal repositories
- Enforcing a policy for inviting collaborators to repositories
- Enforcing a policy for the default branch name
- Enforcing a policy for changes to repository visibility
- Enforcing a policy for repository deletion and transfer
- Enforcing a policy for deleting issues
- Enforcing a policy for Git push limits
- Enforcing a policy for the display of member names in your repositories
- Configuring the merge conflict editor for pull requests between repositories
- Configuring force pushes
- Configuring anonymous Git read access

You can enforce policies for repository management 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 repository management in an enterprise.

## About policies for repository management in your enterprise

You can enforce policies to control how members of your enterprise on GitHub Enterprise Server manage repositories. You can also allow organization owners to manage policies for repository management. For more information, see "[Creating and managing repositories](#)" and "[Organizations and teams documentation](#)."

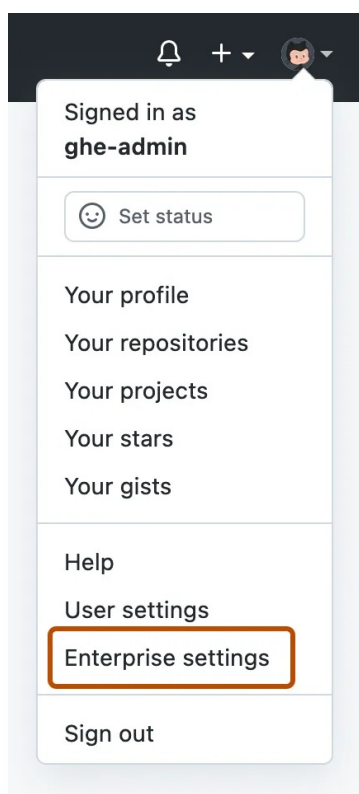
## Configuring the default visibility of new repositories

Each time someone creates a new repository within your enterprise, that person must choose a visibility for the repository. When you configure a default visibility setting for the enterprise, you choose which visibility is selected by default. For more information on repository visibility, see "[About repositories](#)."

If an enterprise owner disallows members from creating certain types of repositories, members will not be able to create that type of repository even if the visibility setting

defaults to that type. For more information, see "[Enforcing a policy for repository creation](#)."

- 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 **Options**.
- 4 Under "Default repository visibility", use the drop-down menu and select a default visibility.

**Warning:** If you add an image attachment to a pull request or issue comment, anyone can view the anonymized image URL without authentication, even if the pull request is in a private repository, or if private mode is enabled. To prevent unauthorized access to the images, ensure that you restrict network access to the systems that serve the images, including your GitHub Enterprise Server instance.

## Enforcing a policy for base repository permissions [↗](#)

Across all organizations owned by your enterprise, you can set a base repository permission level (none, read, write, or admin) for organization members, or allow owners to administer the setting on the 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 **Repositories**.
- 4 Under "Base permissions", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click **View your organizations' current configurations**.

All organizations: Enabled ▾

[View your organizations' current configurations](#) without the enterprise's policy.

- 5 Under "Base permissions", select the dropdown menu and click a policy.

## Enforcing a policy for repository creation [↗](#)

Across all organizations owned by your enterprise, you can allow members to create repositories, restrict repository creation to organization owners, or allow owners to administer the setting on the organization level.

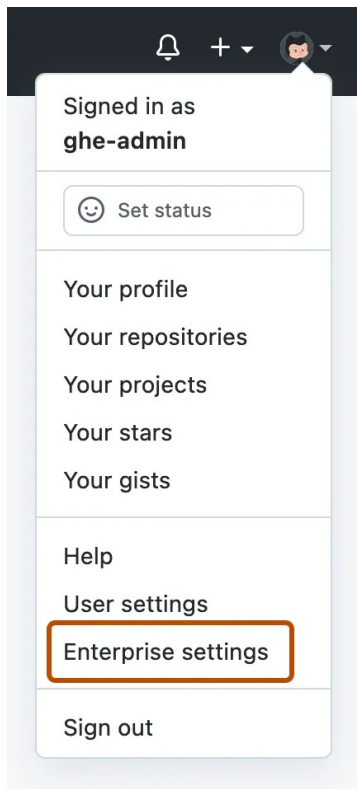
If you allow members to create repositories in your organizations, you can choose which types of repositories (public, private, and internal) that members can create.

You can also prevent users from creating repositories owned by their user accounts.

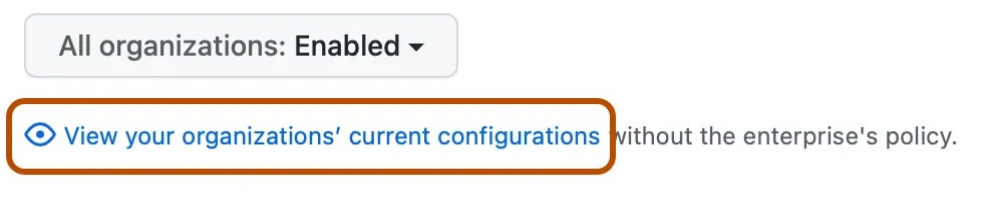
For more information about internal repositories, see "[Creating a new repository](#)."

Organization owners can always create any type of repository, and outside collaborators can never create any type of repository. For more information, see "[About repositories](#)."

- 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 **Repositories**.
- 4 Under "Repository creation", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click **View your organizations' current configurations**.



- 5 Under "Repository creation", select a policy.
- 6 If you selected **Members can create repositories**, select one or more repository types.
- 7 Optionally, to prevent enterprise members from creating repositories owned by their user accounts, select **Block the creation of user namespace repositories**.

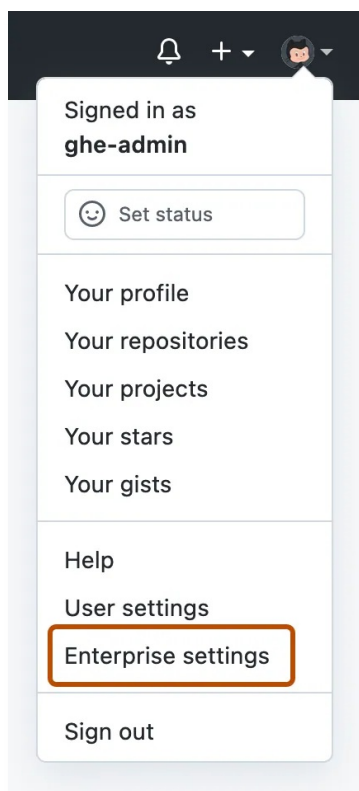
## Enforcing a policy for forking private or internal repositories

Across all organizations owned by your enterprise, you can allow people with access to a private or internal repository to fork the repository, never allow forking of private or internal repositories, or allow owners to administer the setting on the organization level.

People with admin permissions can set a more granular forking policy. For more information, see "[Managing the forking policy for your organization](#)."

**Note:** If your "Repository creation" policy prevents enterprise members from creating repositories owned by their user accounts, members will not be allowed to fork a repository in their user accounts, regardless of your "Repository forking" policy.

- 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 **Repositories**.
- 4 Under "Repository forking", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click **View your organizations' current configurations**.

All organizations: Enabled ▾

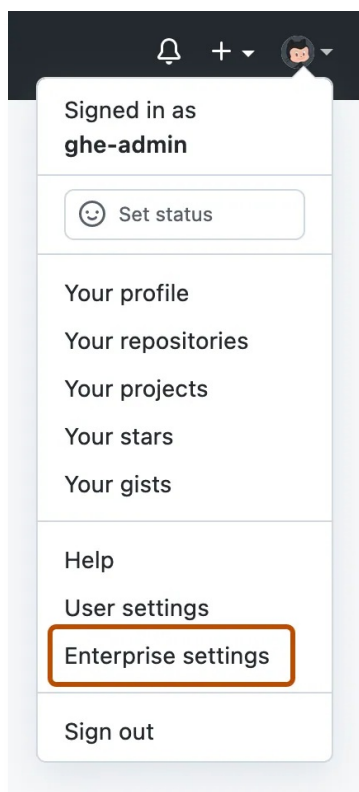
[View your organizations' current configurations](#) without the enterprise's policy.

- 5 Under "Repository forking", select the dropdown menu and click a policy.
- 6 If forking is enabled, select a policy for where users are allowed to fork repositories.

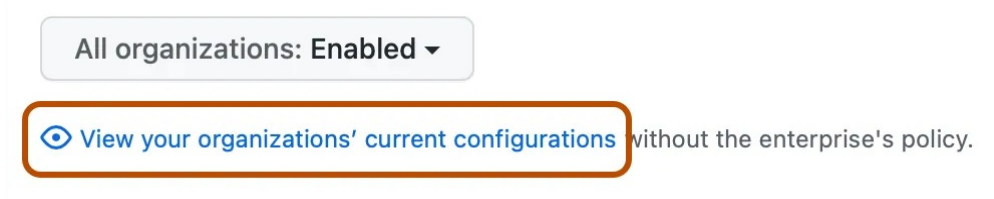
## Enforcing a policy for inviting collaborators to repositories [↗](#)

Across all organizations owned by your enterprise, you can allow members to invite collaborators to repositories, restrict invitations to organization owners, restrict invitations to enterprise owners, or allow organization owners to administer the setting on the 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 **Repositories**.
- 4 Under "Repository invitations", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click [View your organizations' current configurations](#).

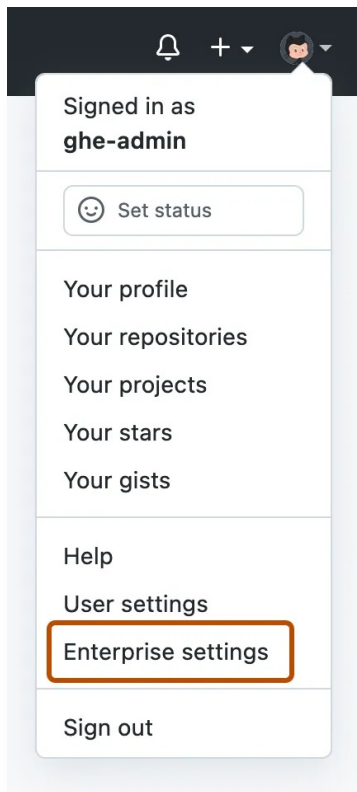




- 5 Under "Repository invitations", select the dropdown menu and click a policy.

## Enforcing a policy for the default branch name [🔗](#)

Across all organizations owned by your enterprise, you can set the default branch name for any new repositories that members create. You can choose to enforce that default branch name across all organizations or allow individual organizations to set a different one.

- 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 **Repositories**.
- 4 Under "Default branch name", enter the default branch name that new repositories should use.
- 5 Optionally, to enforce the default branch name for all organizations in the enterprise, select **Enforce across this enterprise**.
- 6 Click **Update**.

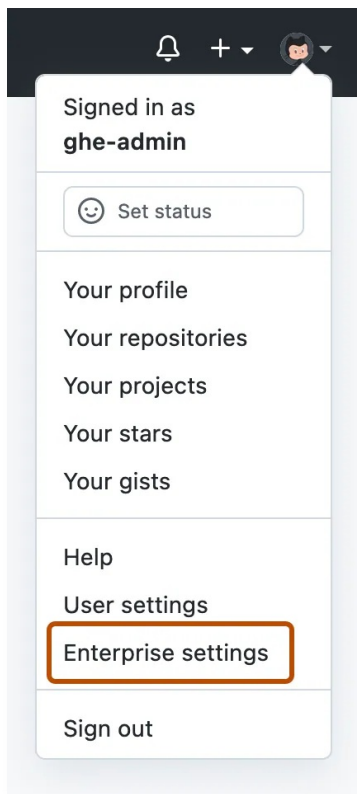
## Enforcing a policy for changes to repository visibility



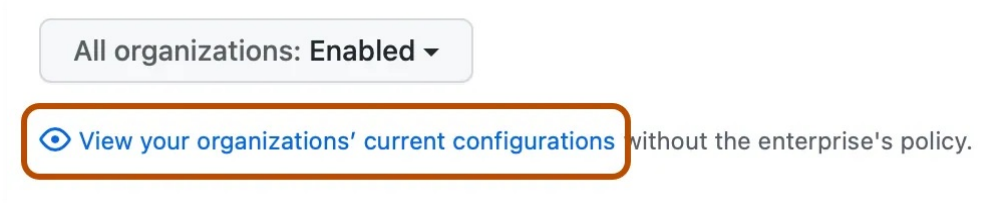
Across all organizations owned by your enterprise, you can allow members with admin access to change a repository's visibility, restrict repository visibility changes to organization owners, or allow owners to administer the setting on the organization level. When you prevent members from changing repository visibility, only enterprise owners can change the visibility of a repository.

If an enterprise owner has restricted repository creation to organization owners only, then members will not be able to change repository visibility. If an enterprise owner has restricted member repository creation to private repositories only, then members will only be able to change the visibility of a repository to private. For more information, see "[Enforcing a policy for repository creation](#)."

- 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 **Repositories**.
- 4 Under "Repository visibility change", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click **View your organizations' current configurations**.



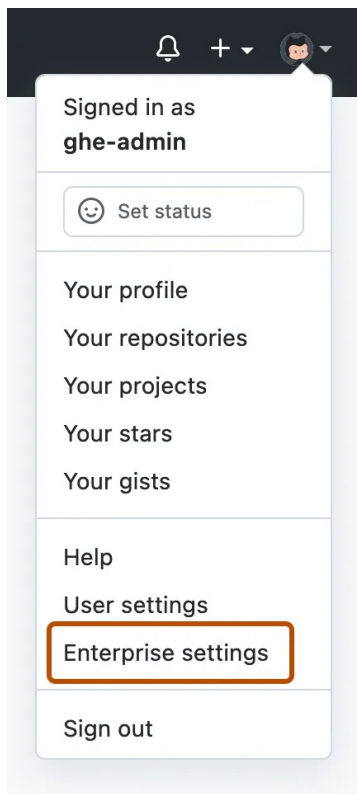
- 5 Under "Repository visibility change", select the dropdown menu and click a policy.

## Enforcing a policy for repository deletion and transfer

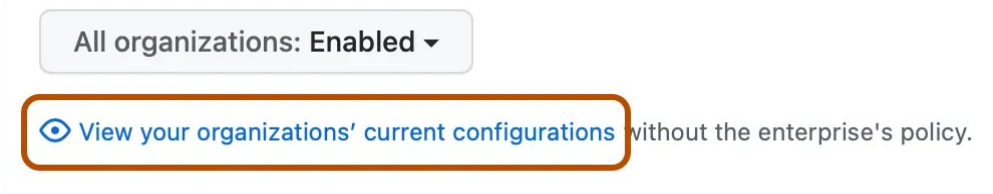
Across all organizations owned by your enterprise, you can allow members with admin permissions to delete or transfer a repository, restrict repository deletion and transfers to organization owners, or allow owners to administer the setting on the organization level.

- 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 **Repositories**.
- 4 Under "Repository deletion and transfer", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click **View your organizations' current configurations**.

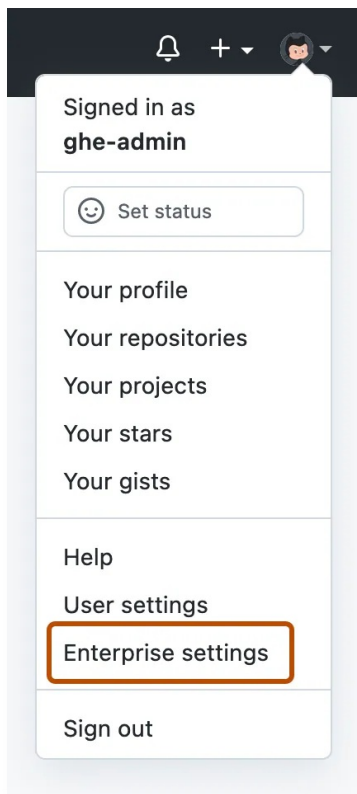


- 5 Under "Repository deletion and transfer", select the dropdown menu and click a policy.

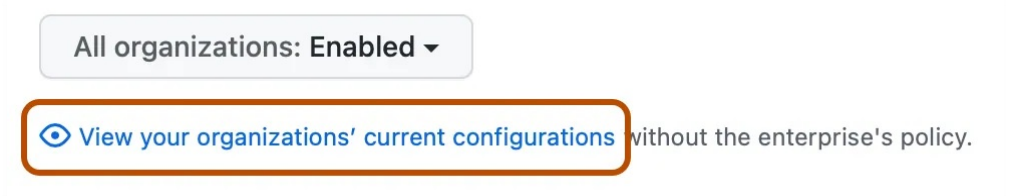
## Enforcing a policy for deleting issues [🔗](#)

Across all organizations owned by your enterprise, you can allow members with admin access to delete issues in a repository, restrict issue deletion to organization owners, or allow owners to administer the setting on the 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 On the **Repository policies** tab, under "Repository issue deletion", review the information about changing the setting. Optionally, to view the current configuration for all organizations in the enterprise account before you change the setting, click **View your organizations' current configurations**.



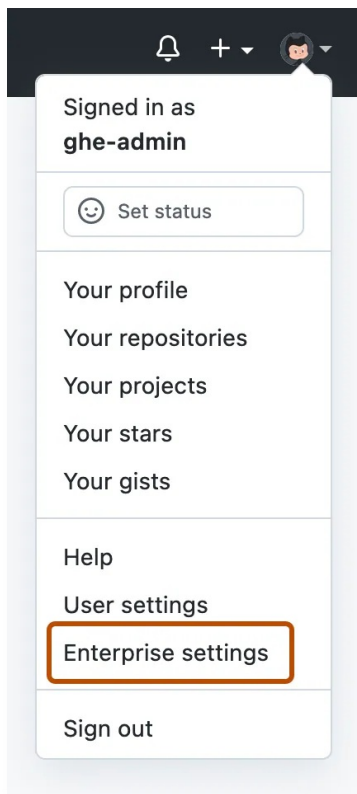
- 4 Under "Repository issue deletion", select the dropdown menu and click a policy.

## Enforcing a policy for Git push limits [🔗](#)

To keep your repository size manageable and prevent performance issues, you can configure a file size limit for repositories in your enterprise.

By default, when you enforce repository upload limits, people cannot add or update files larger than 100 MB.

- 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 **Options**.
- 4 Under "Repository upload limit", use the drop-down menu and click a maximum object size.
- 5 Optionally, to enforce a maximum upload limit for all repositories in your enterprise, select **Enforce on all repositories**

#### Repository upload limit

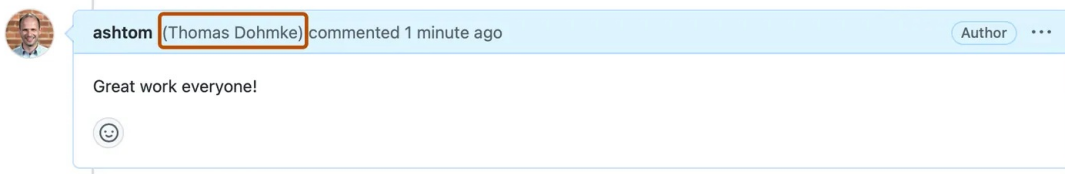
The maximum size of Git objects that can be pushed to repositories. Allowing large objects to be pushed into Git can degrade performance, consider other options (e.g. git-lfs) before increasing this value. If enforced, the setting cannot be changed for individual repositories.

Default (100MB) ▾

☐ Enforce for all repositories on the instance

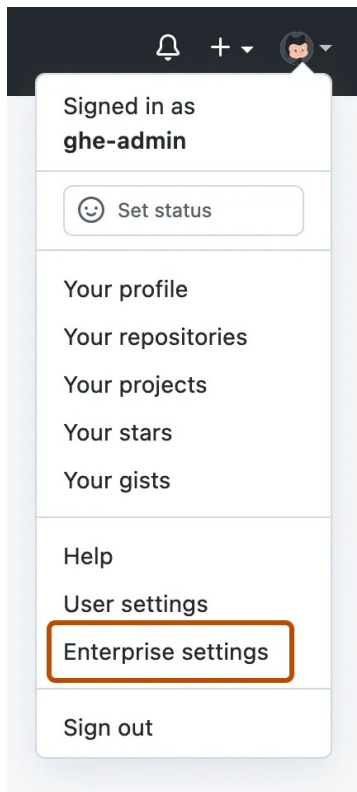
## Enforcing a policy for the display of member names in your repositories [🔗](#)

Across all organizations owned by your enterprise, you can allow members to see a comment author's profile name, in addition to their username, in issues and pull requests for public and internal repositories.



**Note:** When this policy is enforced for all repositories in the enterprise, it overrides the organization setting for private repositories. For more information, see "[Managing the display of member names in your organization](#)".

- 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 **Options**.
- 4 Under "Allow members to see the comment author's profile name in public and internal repositories", select the dropdown menu and click a policy.
- 5 Optionally, to enforce the display of profile names for all repositories in your enterprise, select **Enforce for all repositories on the instance**.

#### Allow members to see comment author's profile name in public and internal repositories

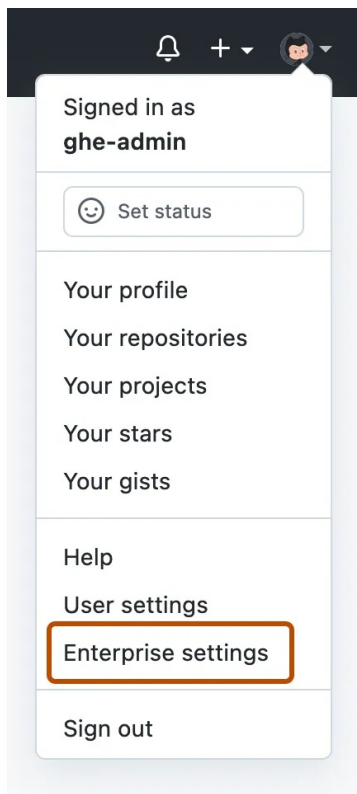
If enabled, members will be able to see comment author's profile name in issues and pull requests for public and internal repositories.

Disabled ▾ ☐ Enforce for all repositories on the instance

## Configuring the merge conflict editor for pull requests between repositories [🔗](#)

Requiring users to resolve merge conflicts locally on their computer can prevent people from inadvertently writing to an upstream repository from a fork.

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



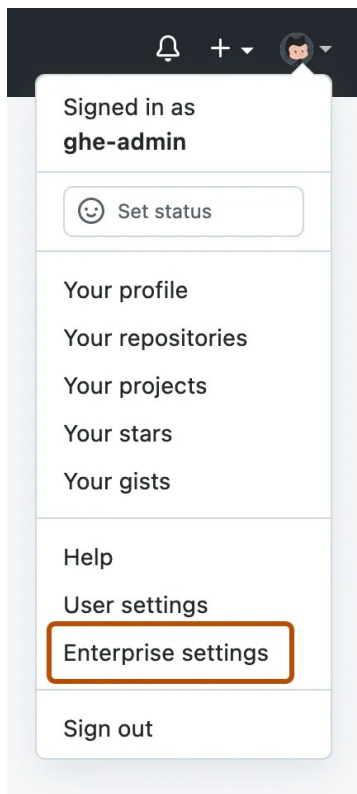
- 2 In the enterprise account sidebar, click **Policies**.
- 3 Under **Policies**, click **Options**.
- 4 Under "Conflict editor for pull requests between repositories", use the drop-down menu, and click **Disabled**.



## Configuring force pushes [↗](#)

Each repository inherits a default force push setting from the settings of the user account or organization that owns the repository. Each organization and user account inherits a default force push setting from the force push setting for the enterprise. If you change the force push setting for the enterprise, the policy applies to all repositories owned by any user or organization.

## Blocking force pushes to all repositories [↗](#)


- 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 **Options**.
- 4 Under "Force pushes", select the dropdown menu, and click **Allow**, **Block**, or **Block to the default branch**.
- 5 Optionally, to override organization and repository level settings for force pushes, select **Enforce on all repositories**.

## Blocking force pushes to a specific repository

**Note:** Each repository automatically inherits default settings from the organization or user that owns it. You cannot override the default setting if the repository's owner has enforced the setting on all of their repositories.

- 1 Sign in to your GitHub Enterprise Server instance at `http(s)://HOSTNAME/login`.
- 2 From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click .
- 3 If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
- 4 Under "Search users, organizations, teams, repositories, gists, and applications", type the name of the repository in the text field. Then to the right of the field, click **Search**.

Search users, organizations, teams, repositories, gists, and applications

Users are found by login, email, SSH key SHA256 fingerprint, GPG key, or database ID.

Organizations are found by login, email, or database ID.

Teams are found by organization/team, GraphQL object ID, or database ID.

Repositories are found by name, "username/repository", deploy key SHA256 fingerprint, or database ID.


Gists are found by name or "username/repository".

OAuth applications are found by name, client ID or application ID.

GitHub Apps are found by name or integration ID.

GitHub App installation are found by installation ID.

Webhooks are found by hook ID.

- 5 Under "Search results – Repositories", click the name of the repository.
- 6 In the upper-right corner of the page, click  **Admin**.



- 7 Under "Push and Pull", to the right of "Force pushes", select the dropdown menu, and click **Block** or **Block to the default branch**.

## Blocking force pushes to repositories owned by a user account or organization

Repositories inherit force push settings from the user account or organization to which they belong. User accounts and organizations in turn inherit their force push settings from the force push settings for the enterprise.

You can override the default inherited settings by configuring the settings for a user account or organization.

- 1 Sign in to your GitHub Enterprise Server instance at `http(s)://HOSTNAME/login`.
- 2 From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click .
- 3 If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
- 4 Under "Search users, organizations, teams, repositories, gists, and applications", type the name of the user or organization in the text field. Then to the right of the field, click **Search**.

Search users, organizations, teams, repositories, gists, and applications

Users are found by login, email, SSH key SHA256 fingerprint, GPG key, or database ID.

Organizations are found by login, email, or database ID.

Teams are found by organization/team, GraphQL object ID, or database ID.

Repositories are found by name, "username/repository", deploy key SHA256 fingerprint, or database ID.

Gists are found by name or "username/repository".

OAuth applications are found by name, client ID or application ID.

GitHub Apps are found by name or integration ID.

GitHub App installation are found by installation ID.

Webhooks are found by hook ID.

- 5 In the search results, click the name of the user or organization.


user

Search results – Accounts

Fuzzy matches

 user2

 user1

- 6 In the upper-right corner of the page, click  **Admin**.


 **Admin**  Security  Content  Collaboration

- 7 Under "Repository default settings" in the "Force pushes" section, select a policy.
- To block force pushes to all branches, select **Block**.
  - To only block force pushes to the default branch, select **Block to the default branch**.
- 8 Optionally, to override repository-specific settings, select **Enforce on all repositories**. Note that this will **not** override an enterprise-wide policy.

Repository default settings

Force pushes

Allow or block force pushes made to repositories within this organization.

 Allow ☒ Enforce on all repositories

## Configuring anonymous Git read access [↗](#)

### Warnings:



- The Git protocol is unauthenticated and unencrypted. An attacker could intercept repository data transferred over connections using this protocol.
- If you enable anonymous Git read access, you're responsible for all access and use of the feature. GitHub is not responsible for any unintended access, security risks, or misuse of the feature.
- You may not use this feature to violate your license from GitHub, including the limit on the number of user licenses for your GitHub Enterprise Server instance.

If you have [enabled private mode](#) for your GitHub Enterprise Server instance, you can allow repository administrators to enable anonymous Git read access to public repositories.

Enabling anonymous Git read access allows users to bypass authentication for custom tools on your enterprise. When you or a repository administrator enable this access setting for a repository, unauthenticated Git operations (and anyone with network access to GitHub Enterprise Server) will have read access to the repository without authentication.

Anonymous Git read access is disabled by default.

If necessary, you can prevent repository administrators from changing anonymous Git access settings for repositories on your enterprise by locking the repository's access settings. After you lock a repository's Git read access setting, only a site administrator can change the setting.

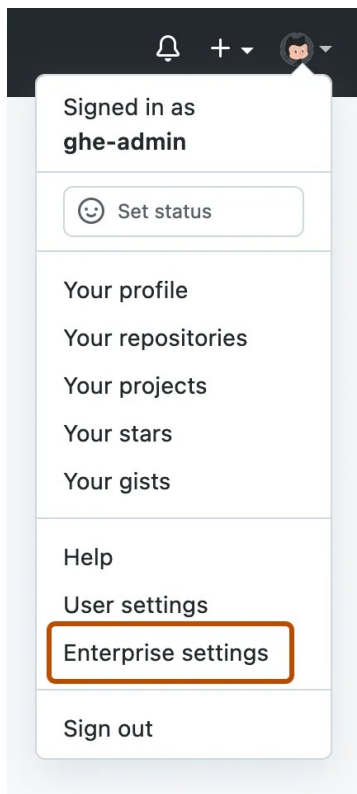
To see the repositories with anonymous Git read access enabled, filter the repositories list in the site admin dashboard.

#### Notes:

- You cannot change the Git read access settings for forked repositories since they inherit their access settings from the root repository by default.
- If a public repository becomes private, then anonymous Git read access will automatically be disabled for that repository and its forks.
- If a repository with anonymous authentication contains Git LFS assets, it will fail to download the Git LFS assets since they still require authentication. We strongly recommend not enabling anonymous Git read access for a repository with Git LFS assets.

## Setting anonymous Git read access for all repositories

- 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 **Options**.
- 4 Under "Anonymous Git read access", use the drop-down menu, and click **Enabled**.
- 5 Optionally, to prevent repository admins from changing anonymous Git read access settings in all repositories on your enterprise, select **Prevent repository admins from changing anonymous Git read access**.

## Setting anonymous Git read access for a specific repository [🔗](#)

- 1 From an administrative account on GitHub Enterprise Server, in the upper-right corner of any page, click **⚙**.
- 2 If you're not already on the "Site admin" page, in the upper-left corner, click **Site admin**.
- 3 Under "Search users, organizations, teams, repositories, gists, and applications", type the name of the repository in the text field. Then to the right of the field, click **Search**.

Search users, organizations, teams, repositories, gists, and applications

Users are found by login, email, SSH key SHA256 fingerprint, GPG key, or database ID.

Organizations are found by login, email, or database ID.

Teams are found by organization/team, GraphQL object ID, or database ID.

Repositories are found by name, "username/repository", deploy key SHA256 fingerprint, or database ID.

Gists are found by name or "username/repository".

OAuth applications are found by name, client ID or application ID.


GitHub Apps are found by name or integration ID.

GitHub App installation are found by installation ID.

Webhooks are found by hook ID.

Search

4 Under "Search results – Repositories", click the name of the repository.

5 In the upper-right corner of the page, click  **Admin**.



6 Under "Danger Zone", next to "Enable Anonymous Git read access", click **Enable**.

Danger Zone

Toggle visibility

Make this repository private.

Make private

Archive repository

Mark this repository as archived and read-only.

Archive

Disable repository

Disable access to the repository and all of its forks.

Disable access

Enable anonymous Git read access

Enable read access to this repository for Git operations without authentication.

Enable

☐ Prevent repository admins from enabling anonymous Git read access.

7 Review the changes. To confirm, click **Yes, enable anonymous Git read access**.

8 Optionally, to prevent repository admins from changing this setting for this repository, select **Prevent repository admins from disabling anonymous Git read access**.

## Legal