About permissions and visibility of forks

In this article

The permissions and visibility of forks depend on whether the upstream repository is public or private, whether it is owned by an organization, and the policies of your enterprise.

About permissions for creating forks

You can fork a private or internal repository to your personal account or to an organization on your GitHub Enterprise Server instance where you have permission to create repositories, provided that the settings for the repository and your enterprise policies allow forking. Generally, you can fork any public repository to your personal account or to an organization where you have permission to create repositories.

If you fork a private repository that belongs to a personal account, external collaborators also get access to the fork. If you fork a private or internal repository that belongs to an organization, teams within the organization get access to the fork, but external collaborators do not. You can add an external collaborator to the fork, but only if the external collaborator also has access to the upstream repository.

Organizations can allow or prevent the forking of any private repositories owned by the organization, and enterprises can enforce policies to specify where members can create forks of private or internal repositories. Policies control the options available to the enterprise's organizations.. For more information, see "Managing the forking policy for your organization" and "Enforcing repository management policies in your enterprise."

About visibility of forks

A fork is a new repository that shares code and visibility settings with the upstream repository. All forks of public repositories are public. You cannot change the visibility of a fork.

All repositories belong to a repository network. A repository network contains the upstream repository, the upstream repository's direct forks, and all forks of those forks. All forks in the repository network have the same visibility setting. For more information, see “Understanding connections between repositories.”

If you delete a repository or change the repository's visibility settings, you will affect the repository's forks. For more information, see "What happens to forks when a repository is deleted or changes visibility?"

About permissions of forks

Private forks inherit the permissions structure of the upstream repository. This helps owners of private repositories maintain control over their code. For example, if the upstream repository is private and gives read/write access to a team, then the same team will have read/write access to any forks of the private upstream repository. Only team permissions (not individual permissions) are inherited by private forks.

Public forks do not inherit the permissions structure of the upstream repository. If you fork a public repository to your personal account, make changes, then open a pull request to propose your changes to the upstream repository, you can give anyone with push access to the upstream repository permission to push changes to your pull request branch (including deleting the branch). This speeds up collaboration by allowing repository maintainers to make commits or run tests locally to your pull request branch from a user-owned fork before merging. You cannot give push permissions to a fork owned by an organization. For more information, see "Allowing changes to a pull request branch created from a fork."

Important security considerations

If you work with forks, or if you're the owner of a repository or organization that allows forking, it's important to be aware of the following security considerations.

- Forks have their own permissions separate from the upstream repository.
- The owners of a repository that has been forked have read permission to all forks in the repository's fork network.
- Organization owners of a repository that has been forked have admin permission to forks created in personal user namespaces, including the ability to delete the fork and its branches.
- Organization owners of a repository that has been forked have read permission to forks created in organizations, but do not have the ability to delete the fork or its branches.
- Forks created in another organization will not be deleted when individual access is removed from the upstream repository.
- Commits to any repository in a fork network can be accessed from any repository in the same fork network, including the upstream repository.

About forks within an organization

Forks within the same organization copy the collaborators and team settings of their upstream repositories. If a repository is owned by an organization:

- That organization controls the permissions of its forks.
- Any teams from the upstream permission structure that exist and are visible in the target organization or user namespace will have their permissions copied.
- Admin permissions remain with the upstream owner, except when a user forks into a different organization.
- If that repository is forked to a user namespace, the organization maintains admin permissions and any teams with access maintain access.