About rulesets

In this article

Rulesets help you to control how people can interact with branches and tags in a repository.

> **Who can use this feature**
> Anyone with read access to a repository can view the repository's rulesets. People with admin access to a repository, or a custom role with the "edit repository rules" permission, can create, edit, and delete rulesets for a repository and view ruleset insights. For more information, see "About custom repository roles."

Rulesets are available in public repositories with GitHub Free and GitHub Free for organizations, and in public and private repositories with GitHub Pro, GitHub Team, and GitHub Enterprise Cloud. For more information, see "GitHub's plans."

About rulesets

A ruleset is a named list of rules that applies to a repository, or to multiple repositories in an organization. You can create rulesets to control how people can interact with selected branches and tags in a repository. You can control things like who can push commits to a certain branch and how the commits must be formatted, or who can delete or rename a tag. For example, you could set up a ruleset for your repository's `feature` branch that requires signed commits and blocks force pushes for all users except repository administrators.

For each ruleset you create, you specify which branches or tags in your repository, or which repositories in your organization, the ruleset applies to. You can use `fnmatch` syntax to define a pattern to target specific branches, tags, and repositories. For example, you could use the pattern `releases/**/*` to target all branches in your repository whose name starts with the string `releases/`. For more information on `fnmatch` syntax, see "Creating rulesets for a repository."

When you create a ruleset, you can allow certain users to bypass the rules in the ruleset. This can be users with a certain role, such as repository administrator, or it can be specific teams or GitHub Apps.

There is a limit of 75 rulesets per repository, and 75 organization-wide rulesets.

About rulesets, protected branches, and protected tags

Rulesets work alongside any branch protection rules and tag protection rules in a repository. Many of the rules you can define in rulesets are similar to protection rules,

and you can start using rulesets without overriding any of your existing protection rules.

Additionally, you can import existing tag protection rules into repository rulesets. This will implement the same tag protections you currently have in place for your repository. For more information, see "Configuring tag protection rules."

Rulesets have the following advantages over branch and tag protection rules.

- Unlike protection rules, multiple rulesets can apply at the same time, so you can be confident that every rule targeting a branch or tag in your repository will be evaluated when someone interacts with that branch or tag. For more information, see "About rule layering."
- Rulesets have statuses, so you can easily manage which rulesets are active in a repository without needing to delete rulesets.
- Anyone with read access to a repository can view the active rulesets for the repository. This means a developer can understand why they have hit a rule, or an auditor can check the security constraints for the repository, without requiring admin access to the repository.

In addition, for organizations on a GitHub Enterprise plan, you can do the following things with rulesets.

- Quickly set up rulesets at the organization level to target multiple repositories in your organization. For more information, see "Managing rulesets for repositories in your organization."
- Create additional rules to control the metadata of commits entering a repository, such as the commit message and the author's email address. For more information, see "Available rules for rulesets."
- Use an "Evaluate" status to test a ruleset before making it active, and use an insights page to view which user actions are being affected by rules. For more information, see "Managing rulesets for a repository."

About rule layering

A ruleset does not have a priority. Instead, if multiple rulesets target the same branch or tag in a repository, the rules in each of these rulesets are aggregated. If the same rule is defined in different ways across the aggregated rulesets, the most restrictive version of the rule applies. As well as layering with each other, rulesets also layer with protection rules targeting the same branch or tag.

For example, consider the following situation for the `my-feature` branch of the `octo-org/octo-repo` repository.

- An administrator of the repository has set up a ruleset targeting the `my-feature` branch. This ruleset requires signed commits, and three reviews on pull requests before they can be merged.
- An existing branch protection rule for the `my-feature` branch requires a linear commit history, and two reviews on pull requests before they can be merged.
- An administrator of the `octo-org` organization has also set up a ruleset targeting the `my-feature` branch of the `octo-repo` repository. The ruleset blocks force pushes, and requires one review on pull requests before they can be merged.

The rules from each source are aggregated, and all rules apply. Where multiple different versions of the same rule exist, the result is that the most restrictive version of the rule applies. Therefore, the `my-feature` branch requires signed commits and a linear commit history, force pushes are blocked, and pull requests targeting the branch will require three reviews before they can be merged.

