Configuring custom deployment protection rules

In this article

About custom deployment protection rules

Using existing custom deployment protection rules

Prerequisites

Enabling custom deployment protection rules for the environment

---

Use GitHub Apps to automate protecting deployments with third-party systems.

> Custom deployment protection rules are available in public repositories for all plans. For access to custom deployment protection rules in private or internal repositories, you must use GitHub Enterprise. For more information, see "[GitHub's plans](#)."

> **Note:** Custom deployment protection rules are currently in public beta and subject to change.

About custom deployment protection rules 🔗

Custom deployment protection rules are powered by GitHub Apps. Once a deployment protection rule is configured and installed in a repository, it can be enabled for any environments in the repository.

After you enable a custom deployment protection rule on an environment, every time a workflow step targets that environment, the deployment protection rule will run automatically. For more information about targeting an environment for deployments, see "[Using environments for deployment](#)."

For more information about creating your own custom deployment protection rules, see "[Creating custom deployment protection rules](#)."

> **Note:** Any number of GitHub Apps-based deployment protection rules can be installed on a repository. However, a maximum of 6 deployment protection rules can be enabled on any environment at the same time.

Using existing custom deployment protection rules 🔗

You can choose to create your own custom deployment protection rules or you may use any existing custom deployment protection rules.

The following is a list of official partner implementations for deployment protection rules.

- Datadog: you can enforce protection rules on your GitHub Actions deployment workflows using Datadog monitors. For more information, see [Gating your GitHub Actions Deployments with Datadog Monitors](#) in the Datadog documentation.

- Honeycomb: you can define thresholds to reject or approve deployments based on data you are sending to Honeycomb. For more information, see [the Honeycomb app](#) in the GitHub Marketplace.
- New Relic: for more information, see [the New Relic app](#) in the GitHub Marketplace.
- NCM NodeSource: for more information, see [the NCM NodeSource app](#) in the GitHub Marketplace.
- Sentry: for more information, see [the Sentry Deployment Gate app](#) in the GitHub Marketplace.
- ServiceNow: for more information, see [GitHub integration with DevOps Change Velocity](#) in the ServiceNow documentation.

Prerequisites 🔗

In order for a custom deployment protection rule to be available to all environments in a repository, you must first install the custom deployment protection rule on the repository. For more information, see "[Installing your own GitHub App](#)."

After a custom deployment protection rule has been installed in a repository, it must be enabled for each environment where you want the rule to apply.

Enabling custom deployment protection rules for the environment 🔗

1. On GitHub.com, navigate to the main page of the repository.

2. Under your repository name, click ⚙ **Settings**. If you cannot see the "Settings" tab, select the ••• dropdown menu, then click **Settings**.
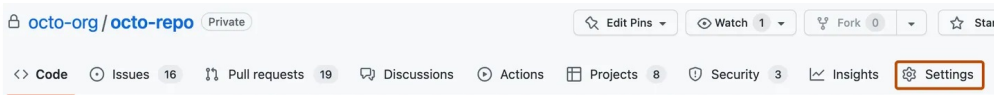
3. In the left sidebar, click **Environments**.

4. Select the environment you want to configure.

5. Under "Deployment protection rules," check the box next to each custom deployment protection rule you want to enable for the environment.

6. Click **Save protection rules**.

Once a custom deployment protection rule has been enabled for an environment, it will automatically run whenever a workflow reaches a job that references the environment. You can see the results of an approval or rejection for your deployment by reviewing the details of the deployment. For more information, see "[Reviewing deployments](#)."