

Using starter workflows

In this article

About starter workflows

Choosing and using a starter workflow

Further reading

GitHub Enterprise Cloud provides starter workflows for a variety of languages and tooling.

About starter workflows

Starter workflows are templates that help you to create your own GitHub Actions workflows for a repository. They offer an alternative to starting from a blank workflow file and are useful because some of the work will already have been done for you.


GitHub Enterprise Cloud offers starter workflows for a variety of languages and tooling. When you set up workflows in your repository, GitHub Enterprise Cloud analyzes the code in your repository and recommends workflows based on the language and framework in your repository. For example, if you use Node.js, GitHub Enterprise Cloud will suggest a starter workflow file that installs your Node.js packages and runs your tests. You can search and filter to find relevant starter workflows.

GitHub provides ready-to-use starter workflows for the following high level categories:

- **Deployment (CD)**. For more information, see "[About continuous deployment](#)."
- **Security**. For more information, see "[Configuring advanced setup for code scanning](#)."
- **Continuous Integration (CI)**. For more information, see "[About continuous integration](#)."
- **Automation**. Automation starter workflows offer solutions for automating workflows, such as triaging pull requests and applying a label based on the paths that are modified in the pull request, or greeting users who are first time contributors to the repository.

You can also create your own starter workflow to share with your organization. These starter workflows will appear alongside the GitHub Enterprise Cloud-provided starter workflows. Anyone with write access to the organization's `github` repository can set up a starter workflow. For more information, see "[Creating starter workflows for your organization](#)."

Choosing and using a starter workflow

- 1 On GitHub.com, navigate to the main page of the repository.
- 2 Under your repository name, click  **Actions**.

- 3 If you already have a workflow in your repository, click **New workflow**.
- 4 The "Choose a workflow" page shows a selection of recommended starter workflows. Find the starter workflow that you want to use, then click **Configure**. To help you find the starter workflow that you want, you can search for keywords or filter by category.
- 5 If the starter workflow contains comments detailing additional setup steps, follow these steps.

There are guides to accompany many of the starter workflows for building and testing projects. For more information, see "[Automating builds and tests](#)."
- 6 Some starter workflows use secrets. For example, `${{ secrets.npm_token }}`. If the starter workflow uses a secret, store the value described in the secret name as a secret in your repository. For more information, see "[Using secrets in GitHub Actions](#)."
- 7 Optionally, make additional changes. For example, you might want to change the value of `on` to change when the workflow runs.
- 8 Click **Start commit**.
- 9 Write a commit message and decide whether to commit directly to the default branch or to open a pull request.

Further reading [↗](#)

- "[About continuous integration](#)"
- "[Managing workflow runs](#)"
- "[About monitoring and troubleshooting](#)"
- "[Managing billing for GitHub Actions](#)"

Legal