

About continuous integration

In this article

About continuous integration

About continuous integration using GitHub Actions

Starter workflow

Further reading

You can create custom continuous integration (CI) workflows directly in your GitHub repository with GitHub Actions.

About continuous integration

Continuous integration (CI) is a software practice that requires frequently committing code to a shared repository. Committing code more often detects errors sooner and reduces the amount of code a developer needs to debug when finding the source of an error. Frequent code updates also make it easier to merge changes from different members of a software development team. This is great for developers, who can spend more time writing code and less time debugging errors or resolving merge conflicts.

When you commit code to your repository, you can continuously build and test the code to make sure that the commit doesn't introduce errors. Your tests can include code linters (which check style formatting), security checks, code coverage, functional tests, and other custom checks.

Building and testing your code requires a server. You can build and test updates locally before pushing code to a repository, or you can use a CI server that checks for new code commits in a repository.

About continuous integration using GitHub Actions

CI using GitHub Actions offers workflows that can build the code in your repository and run your tests. Workflows can run on GitHub-hosted virtual machines, or on machines that you host yourself. For more information, see "[Using GitHub-hosted runners](#)" and "[About self-hosted runners](#)."

You can configure your CI workflow to run when a GitHub event occurs (for example, when new code is pushed to your repository), on a set schedule, or when an external event occurs using the repository dispatch webhook.

GitHub runs your CI tests and provides the results of each test in the pull request, so you can see whether the change in your branch introduces an error. When all CI tests in a workflow pass, the changes you pushed are ready to be reviewed by a team member or merged. When a test fails, one of your changes may have caused the failure.

When you set up CI in your repository, GitHub analyzes the code in your repository and recommends CI workflows based on the language and framework in your repository. For example, if you use [Node.js](#), GitHub will suggest a starter workflow that installs your Node.js packages and runs your tests. You can use the CI starter workflow suggested by GitHub, customize the suggested starter workflow, or create your own custom workflow

file to run your CI tests.

In addition to helping you set up CI workflows for your project, you can use GitHub Actions to create workflows across the full software development life cycle. For example, you can use actions to deploy, package, or release your project. For more information, see "[Learn GitHub Actions](#)."

For a definition of common terms, see "[Understanding GitHub Actions](#)."

Starter workflow

GitHub offers CI starter workflow for a variety of languages and frameworks.

Browse the complete list of CI starter workflow offered by GitHub in the [actions/starter-workflows](#) repository.

Further reading

- "[Managing billing for GitHub Actions](#)"

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)