About packaging with GitHub Actions

In this article

You can set up workflows in GitHub Actions to produce packages and upload them to GitHub Packages or another package hosting provider.

Note: GitHub-hosted runners are not currently supported on GitHub Enterprise Server. You can see more information about planned future support on the GitHub public roadmap.

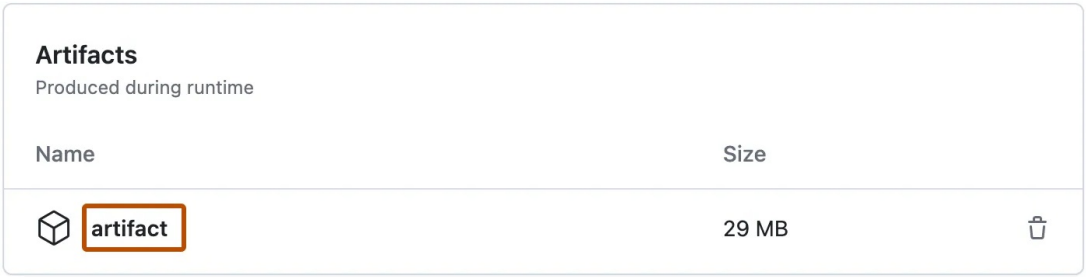
Packaging in continuous integration workflows

A packaging step is a common part of a continuous integration or continuous delivery workflow. Creating a package at the end of a continuous integration workflow can help during code reviews on a pull request.

After building and testing your code, a packaging step can produce a runnable or deployable artifact. Depending on the kind of application you're building, this package can be downloaded locally for manual testing, made available for users to download, or deployed to a staging or production environment.

For example, a continuous integration workflow for a Java project may run `mvn package` to produce a JAR file. Or, a CI workflow for a Node.js application may create a Docker container.

Now, when reviewing a pull request, you'll be able to look at the workflow run and download the artifact that was produced.



This will let you run the code in the pull request on your machine, which can help with debugging or testing the pull request.

Workflows for publishing packages

In addition to uploading packaging artifacts for testing in a continuous integration workflow, you can create workflows that build your project and publish packages to a package registry.

- **Publish packages to GitHub Packages** GitHub Packages can act as a package hosting service for many types of packages. You can choose to share your packages with all of GitHub, or private packages to share with collaborators or an organization. For more information, see "Introduction to GitHub Packages."

  You may want to publish packages to GitHub Packages on every push into the default branch. This will allow developers on your project to always be able to run and test the latest build from the default branch easily, by installing it from GitHub Packages.

- **Publish packages to a package registry** For many projects, publishing to a package registry is performed whenever a new version of a project is released. For example, a project that produces a JAR file may upload new releases to the Maven Central repository. Or, a .NET project may produce a nuget package and upload it to the NuGet Gallery.

  You can automate this by creating a workflow that publishes packages to a package registry on every release creation. For more information, see "Managing releases in a repository."

Further reading

- "Publishing Node.js packages"