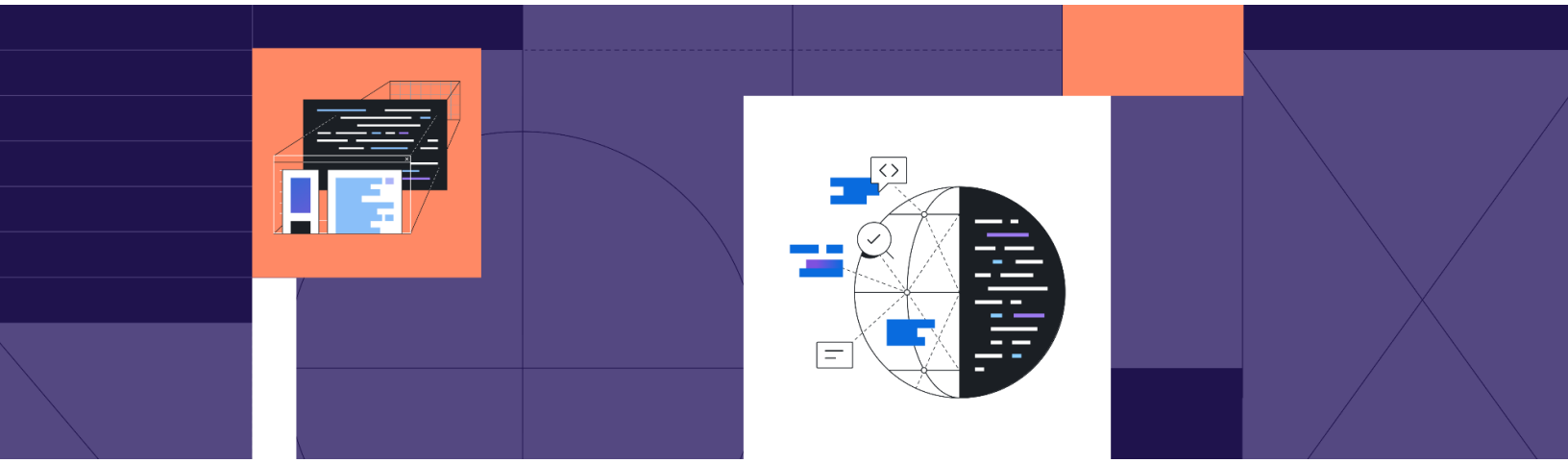
What is GitHub Actions?

Benefits and examples



GitHub Actions gives developers the ability to automate their workflows across issues, pull requests, and more—plus native CI/CD functionality. Here's everything you need to know about Actions including its benefits, how it works, popular use cases, and more.

In 2018, we launched GitHub Actions to help developers automate their workflows—all within GitHub.

Unlike other automation tools and features, GitHub Actions goes beyond the typical applications of testing, building, and deploying. Instead, it offers the flexibility to automate any webhook. It also brings CI/CD (continuous integration/continuous deployment) to the GitHub experience.

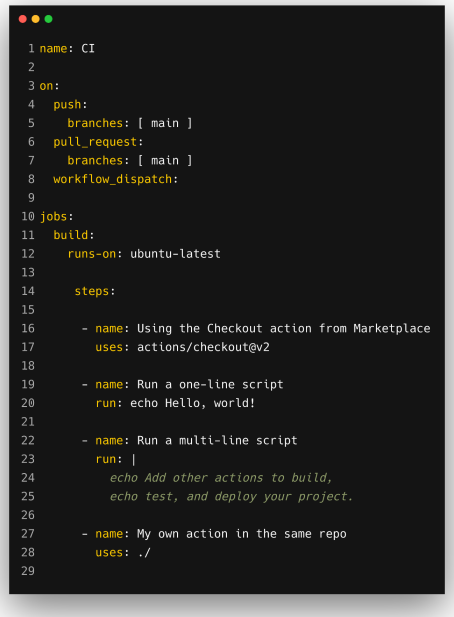
Below, we'll explain everything you need to know about GitHub Actions to get started with automating your workflows.

What's inside

What is GitHub Actions? How does it work?

At the most basic level, GitHub Actions brings automation directly into the software development lifecycle on GitHub via event-driven triggers. These triggers are specified events that can range from creating a pull request to building a new brand in a repository.

All GitHub Actions automations are handled via workflows, which are YAML files placed under the .github/workflows directory in a repository that define automated processes.



```yaml
name: CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest

    steps:

      - name: Using the Checkout action from Marketplace
        uses: actions/checkout@v2

      - name: Run a one-line script
        run: echo Hello, world!

      - name: Run a multi-line script
        run: |
          echo Add other actions to build,
          echo test, and deploy your project.

      - name: My own action in the same repo
        uses: ./

```

An example workflow on GitHub Actions.

Every workflow consists of several different core concepts. These include:

- **Events:** Events are defined triggers that kick off a workflow. They can be configured to look for one or more triggers and qualified as

needed by a developer. They can also be set to run on specific coding branches within a given repository on GitHub.

[Learn more about what events can trigger a workflow.]

- **Jobs:** Jobs are a set of steps that execute on the same runner. Each runs in its own VM and parallel to other jobs, unless otherwise specified.

- **Steps:** Steps are individual tasks that run commands in a job. These can be an action or a shell command. All steps in a job execute on the same runner.

- **Actions:** An action is a command that's executed on a runner—and the core element of GitHub Actions, which is named after it.

- **Runners:** A runner is a GitHub Actions server. It listens for available jobs, runs each in parallel, and reports back progress, logs and results. Each runner can be hosted by GitHub or self-hosted on a localized server. GitHub Hosted runners are based on Ubuntu Linux, Windows, and macOS.

[Learn more about the concepts that make up a GitHub Actions workflow.]

Taken together, these core concepts make workflows incredibly flexible—they can contain simple IFTTT-style (if this, then that) logic flows, or more complex use cases.

Workflows support a number of different automations ranging from pull requests and branch merging—plus, third-party integrations with your

preferred tools such as chat app notifications, testing suites, container management, and more via the GitHub Marketplace.

You can leverage automated workflows to build, test, and deploy code directly into a virtual machine or a Docker container. GitHub Actions also supports “matrix builds,” which enables you to simultaneously test builds across multiple operating systems and runtime versions.

> **Pro tip:** You can reuse workflows in GitHub Actions—which in simple terms means you can put one workflow in another workflow. This makes it easier to avoid duplication and also leverage pre-existing workflows within new automations.

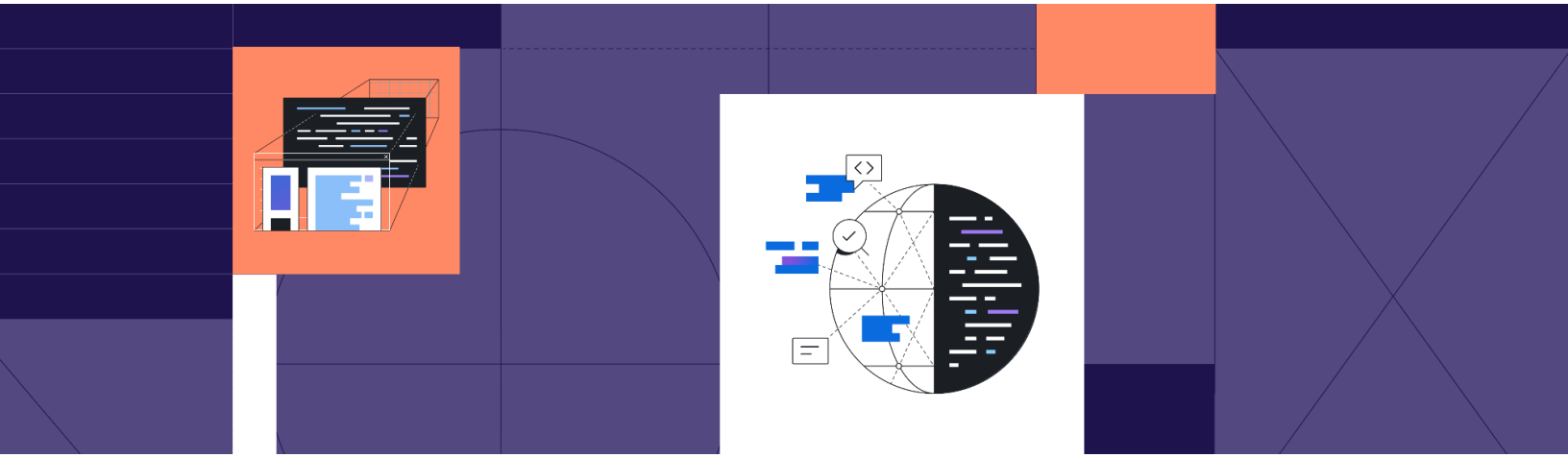
Is GitHub Actions a CI/CD tool?

GitHub Actions is a CI/CD tool for the GitHub flow. You can use it to integrate and deploy code changes to a third-party cloud application platform as well as test, track, and manage code changes. GitHub Actions also supports third-party CI/CD tools, the container platform Docker, and other automation platforms.

Why having a GitHub-native CI/CD tool is helpful: The most basic answer is simplicity—if you’re already hosting a project on GitHub, you have a built-in CI/CD tool that works right alongside your code.

> GitHub Actions is built by and for developers and is designed to make it easy to build a CI/CD pipeline without a dedicated DevOps engineer.

CI/CD pipelines are typically complex and have a lot of tools that range from testing applications to integration tests to container platforms and application platforms, among other things. GitHub Actions simplifies the process with Node and Docker integrations and allows you to specify which version you want to use and then connect your code to a target environment and application platform.



What coding languages does GitHub Actions support?

The short answer: A lot. The long answer: Actions supports the core languages of C, C++, C#, Java, JavaScript, PHP, Python, Ruby, Scala, and TypeScript. This is by design with the intention of making GitHub Actions simple to use for any developer in their preferred coding language. You can also leverage QEMU to use any of your preferred languages with GitHub Actions.

Who can use GitHub Actions?

GitHub Actions is free and available for use on any public repository and self-hosted runner. If you use GitHub-hosted runners or Actions on a private repository, you can still test Actions with the standard Actions allocations before switching to a paid model.

By using the GitHub Free plan, you can store 500MB of workflows on a GitHub-hosted runner and execute 2,000 minutes worth of tasks per month for private repositories.

GitHub Actions is also included in GitHub Pro, GitHub Team, GitHub Enterprise Cloud, GitHub Enterprise Server and GitHub AE with more available storage and minutes by product tier.

> **Pro tip:** GitHub Actions for enterprises are bundled with the standard enterprise package and include ready-to-use actions templates in addition to third-party actions— with configurable permissions to manage what enterprise users can do with Actions.

[Learn more about using GitHub Actions on enterprise accounts.]

What are GitHub Actions minutes?

Actions minutes is the time it takes to compute and execute specific automated tasks on GitHub-hosted runners, which host all workflow executions. If a specific job takes two minutes to execute, for example,

that counts toward the total included minutes per month a developer has per their plan.

A GitHub Free account comes with 2,000 minutes a month that can be used on private repositories (developers have unlimited use of GitHub Actions on a public repository with a self-hosted runner).

> The minutes it takes to execute a task are counted differently by what operating system (OS)—Ubuntu Linux, Windows, or macOS—a developer chooses for their GitHub-hosted runner.

While a developer's chosen OS doesn't impact the length of time it takes to run a workflow, it does impact how much each workflow counts against their minutes per month. This is called the **minute multiplier**.

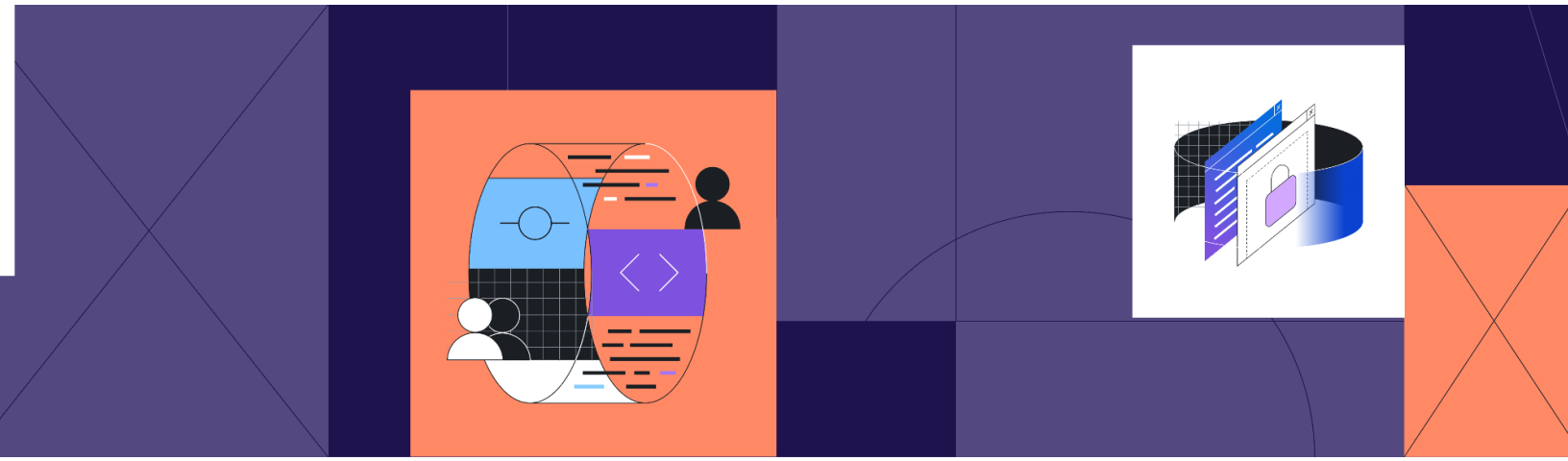
Workflows executed on a Windows or macOS hosted GitHub runner consume minutes at 2 and 10 times the rate that jobs on Linux consume.

Operating system	**Minute multiplier**
Linux	1
MacOS	10
Windows	2

GitHub Actions storage and minute tiers

Product	Storage	Minutes per month
GitHub Free	500 MB	2,000
GitHub Pro	1 GB	3,000
GitHub Free for organizations	500 MB	2,000
GitHub Team	2 GB	3,000
GitHub Enterprise Cloud	50 GB	50,000

[Learn more about billing and pricing for GitHub Actions in our documentation.]



What can you do with GitHub Actions? Popular use cases and examples

At a high level, GitHub Actions can automate any webhook on GitHub—and that makes it a powerful and flexible platform feature that's able to handle complex workflows and operations or do something as simple as send a Slack message to someone when a pull request is ready to review.

Some of the of the most common use cases for GitHub Actions include:

- **Build, test, and deploy within the GitHub flow**: Continuous Integration (CI) and continuous deployment (CD) (aka CI/CD) automations are typically the easiest way for someone to understand the full functionality of GitHub Actions. From automating tests to deploying code, Actions enables you to run CI/CD workflows in containers and virtual machines directly from your repository. You can also integrate your preferred tools third-party CI/CD tools directly into your repositories with Actions.

- **Automate repetitive tasks**: GitHub Actions can be used to automate an almost endless number of steps in the software development lifecycle. Whether it's the creation of a pull request, a new contributor joining your repository, a pull request being merged, or a web hook from a third-party application that is integrated with a given repository, you can introduce an automated response including sorting an issue, or assigning a reviewer to a pull request.

- **Manage users easily at scale:** Maintainers often use GitHub Actions to set organization rules including assigning developer permissions, notifying reviewers of new pull requests, and more. This makes it easier to manage a repository and all of the contributors in a given project.

- **Easily add preferred tools and services to your project:** From testing tools to CI/CD platforms, container management platforms to issue tracking platforms and chat applications, GitHub Actions gives you the ability to connect and integrate your preferred third-party tools and services directly into your repository. This is designed to make it simpler to manage typical workflows and build, test, and deploy code all within the GitHub flow.

- **Quickly review & test code on GitHub:** GitHub Actions lets you integrate any number of third-party testing tools directly into your workflow in your repo—at any step. Moreover, GitHub Actions enables multi-container testing and "matrix builds," which lets you run multiple tests on Linux, Windows, and macOS at the same time.

- **Keep track of your projects:** You can use GitHub Actions to monitor application builds, measure performance, track errors and more via integrations with third-party tools. GitHub Actions also produces live logs, which lets you watch your workflows run in real time. Live logs also give you the ability to copy a link from a failed step to identify and solve potential issues (they support color and emojis, too).

Get started using GitHub Actions

Go to GitHub Actions Tutorial on the Learning Lab to get started using GitHub Actions today.

Additional resources

Check out the following tutorials and resources to learn more about GitHub Actions:

- Simple GitHub Actions Examples: GitHub Docs
- GitHub Actions Kubernetes Deployments: GitHub Docs
- How to manage GitHub Actions permissions in your organization: GitHub Docs