

# **GitHub Actions**

## **What is GitHub Actions?**

GitHub Actions is a Continuous Integration and Continuous Delivery (CI/CD) platform provided by GitHub. It allows developers to automate the build, test, and deployment processes of applications. Workflows can be configured to run automatically whenever specific events occur in a repository.

For example, GitHub Actions can:

- Build and test code whenever a pull request is created
- Deploy changes after a pull request is merged
- Automatically add labels when a new issue is opened

GitHub provides hosted virtual machines with Linux, Windows, and macOS environments to run these workflows. If needed, organizations can also set up self-hosted runners in their own infrastructure.

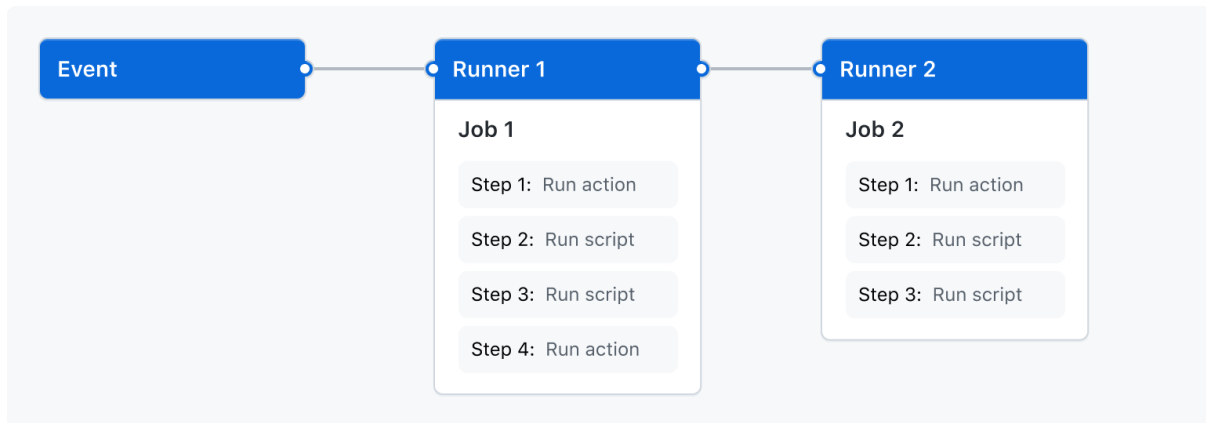
## **Why is GitHub Actions important?**

In software development, many tasks such as building, testing, and deploying need to be performed repeatedly. Doing these tasks manually increases the chances of errors and consumes time.

GitHub Actions helps automate these processes so that they run consistently and automatically whenever required. It ensures that code changes are tested properly and deployment happens in a controlled and reliable manner. It also helps automate repository management tasks beyond DevOps activities.

## Core Components of GitHub Actions

GitHub Actions works through several core components. Each component plays a specific role in the automation process.



### 1. Workflows

#### What is a Workflow?

A workflow is a configurable automated process that runs one or more jobs. It is defined in a YAML file and stored in the repository under:

`.github/workflows`

A repository can have multiple workflows, each designed for different purposes such as:

- Building and testing pull requests
- Deploying applications when a release is created
- Adding labels when a new issue is opened

Workflows can be triggered:

- Automatically when a specific event occurs
- Manually
- On a defined schedule

Workflows can also reference other workflows for reuse.

## **Why are Workflows important?**

Workflows define the automation logic of a repository. Without workflows, GitHub Actions would not know what tasks to perform.

They allow teams to organize different automation processes separately and clearly. Multiple workflows also make it easier to manage complex projects by dividing tasks into logical units.

## **2. Events**

### **What is an Event?**

An event is a specific activity that triggers a workflow to run.

Examples include:

- Creating a pull request
- Opening an issue
- Pushing a commit
- Scheduled triggers
- Manual triggers
- REST API triggers

When an event occurs, GitHub checks whether any workflow is configured to respond to that event and starts it.

### **Why are Events important?**

Events act as the starting point of automation.

Without an event, a workflow will not execute. Events ensure that automation runs exactly when needed, such as after code changes or repository updates.

### **3. Jobs**

#### **What is a Job?**

A job is a collection of steps that run on the same runner. Each job executes independently and contains multiple steps that run sequentially.

By default:

- Jobs run in parallel
- Jobs do not depend on each other

However, dependencies can be configured. If one job depends on another, it will wait until the first job completes successfully.

For example:

- Multiple build jobs can run in parallel for different architectures
- A packaging job can depend on all build jobs
- Once all builds succeed, the packaging job runs

Jobs can also use a matrix strategy to run the same job multiple times with different configurations such as operating systems or language versions.

#### **Why are Jobs important?**

Jobs help divide large workflows into manageable sections.

Running jobs in parallel improves performance and reduces execution time. Dependencies between jobs help maintain proper order when certain tasks must be completed before others.

## **4. Steps**

### **What is a Step?**

A step is an individual task within a job.

Each step can:

- Run a shell script defined by the user
- Execute an action

Steps run in sequence and depend on one another. Since they execute on the same runner within a job, data created in one step can be used in the next step.

For example:

- One step builds the application
- The next step tests the built application

### **Why are Steps important?**

Steps break a job into smaller tasks, making automation easier to understand and manage.

Sequential execution ensures that tasks are performed in the correct order, and shared data between steps allows smooth workflow execution.

## **5. Actions**

### **What is an Action?**

An action is a reusable unit of automation that performs a specific task within a workflow.

Actions can perform tasks such as:

- Pulling the repository from GitHub
- Setting up the required build environment
- Configuring authentication for cloud providers

Users can either use pre-built actions from the GitHub Marketplace or create their own custom actions.

### **Why are Actions important?**

Actions reduce repetitive code in workflow files.

Instead of writing the same configuration repeatedly, reusable actions simplify workflow creation and make automation more efficient and consistent.

## **6. Runners**

### **What is a Runner?**

A runner is a server that executes jobs when workflows are triggered. Each runner can run only one job at a time.

GitHub provides hosted runners with:

- Ubuntu Linux
- Microsoft Windows
- macOS

Each workflow run executes on a fresh virtual machine. Larger runners are also available for higher performance requirements. If specific configurations are needed, self-hosted runners can be set up in private infrastructure.

### **Why are Runners important?**

Runners provide the environment where automation actually takes place.

Using fresh virtual machines ensures a clean and consistent execution environment. Self-hosted runners allow flexibility when special hardware or operating systems are required.

## **How Everything Works Together**

### **What happens in the complete process?**

1. An event occurs in the repository
2. The event triggers a workflow
3. The workflow starts one or more jobs
4. Each job runs on a runner
5. Each job executes its steps in sequence
6. Steps run scripts or actions to complete tasks

### **Why is this structure useful?**

This structured approach ensures that automation is organized, predictable, and scalable. Each component has a defined role, which makes the entire CI/CD process reliable and easy to manage.

## **Conclusion**

GitHub Actions is an integrated automation platform within GitHub that supports CI/CD and repository-related automation.

By organizing processes into workflows, events, jobs, steps, actions, and runners, it provides a clear and structured system for automating development activities. This improves consistency, efficiency, and reliability in software development projects.