

CLOUD, DEVOPS, AND CONTAINERS

CI/CD with GitHub Actions

Level: Practitioner • 2 days (expandable to 3) • Virtual, In-person

Overview

Every team says it does CI/CD; far fewer have pipelines they trust. GitHub Actions has become the default automation platform for much of the industry because the pipeline lives next to the code, but that convenience makes it easy to accumulate slow, flaky, copy-pasted workflows that nobody wants to touch. The difference between that and a pipeline that quietly ships every change is a set of learnable practices.

This is a hands-on, practitioner course. Rather than tour every feature of GitHub Actions, it builds a small number of skills that compound: first a clear mental model of workflows, events, and runners, then a real continuous integration pipeline, then continuous delivery with environments and approvals, and finally the reuse and security practices that keep pipelines maintainable at team scale. Every module includes a lab in a real repository, and each module builds on the one before.

Who Should Attend

- Developers who want to automate their build, test, and release process
- DevOps and platform engineers standardizing CI/CD on GitHub
- Teams migrating pipelines from Jenkins, Azure DevOps, or other CI systems

Prerequisites

- Comfortable with git and GitHub: branching, commits, and pull requests
- Working proficiency in at least one programming language
- No prior GitHub Actions experience required; for CI/CD concepts from first principles, see *DevOps Foundations*

What You Will Learn

- Explain how GitHub Actions works: workflows, events, jobs, runners, and the marketplace
- Build a continuous integration pipeline that builds, tests, and reports on every pull request
- Speed pipelines up with caching, artifacts, and matrix builds
- Design a continuous delivery flow with environments, approvals, and safe rollbacks
- Reuse pipeline logic with reusable workflows and composite actions
- Secure pipelines with OIDC, least-privilege permissions, and supply chain hygiene

Course Outline

Day one: continuous integration that earns trust

- How Actions Works
 - Workflows, events, jobs, steps, and runners: the mental model
 - Reading workflow runs and logs without guessing
 - The marketplace: using actions without blindly trusting them

- Lab: write a first workflow from scratch and trigger it from a push and a pull request
 - A Real CI Pipeline
 - Building and testing on every pull request
 - Status checks and branch protection: making green mean something
 - Lab: build a CI pipeline for a sample application that blocks a failing pull request
 - Faster and Broader
 - Caching dependencies and build outputs
 - Artifacts: passing results between jobs
 - Matrix builds across versions and platforms
 - Lab: cut the pipeline's run time with caching and add a matrix build
- Day two: delivery, reuse, and security**
- From CI to CD
 - What changes when pipelines deploy: environments, approvals, and secrets
 - Deploying to a real target, and rolling back when you must
 - Lab: deploy the application to staging automatically, with an approval gate to production
 - Reuse at Team Scale
 - Reusable workflows and composite actions
 - Sharing standards across repositories without copy-paste
 - Lab: extract the pipeline into a reusable workflow another repository can call
 - Securing the Pipeline
 - Secrets management and OIDC: deploying to the cloud without stored credentials
 - Least-privilege permissions for workflows and tokens
 - Supply chain hygiene: pinning actions and reviewing third-party code
 - Lab: harden the pipeline with OIDC, scoped permissions, and pinned actions

Extended Version

The three-day version keeps the same gradient and adds:

- Self-hosted runners: when you need them and how to run them safely
- Advanced deployment strategies: blue-green and canary releases from Actions
- Automating more than code: releases, versioning, and repository chores
- A capstone: build a complete, secured CI/CD pipeline for an application, from pull request to production