GitHub Actions (Zero to Advanced)

Course Overview

Title: CI/CD Automation with GitHub Actions
Level: Beginner → Advanced
Duration: 15–25 Hours
Outcome:
Design, build, and operate **production-grade CI/CD pipelines** using GitHub Actions for modern cloud-native applications.

Learning Outcomes

By the end of this course, you will be able to:

- Understand GitHub Actions architecture
- Create CI/CD pipelines using YAML
- Automate build, test, security, and deployment workflows
- Integrate GitHub Actions with AWS, Docker, and Terraform
- Apply DevOps best practices and GitOps workflows
- Prepare for DevOps interviews

Module 1: CI/CD & GitHub Actions Fundamentals

Objective: Understand CI/CD concepts and GitHub Actions basics

Topics

- What is CI/CD?
- CI vs CD vs CD (Delivery vs Deployment)
- Why GitHub Actions?
- GitHub Actions vs Jenkins vs GitLab CI
- GitHub Actions use cases in real organizations

Hands-On

- Explore GitHub Actions UI
- Enable Actions for a repository

Module 2: GitHub Actions Architecture

Objective: Learn how GitHub Actions works internally

Topics

- Workflows
- Jobs
- Steps
- Runners:
    - GitHub-hosted
    - Self-hosted

- Events:
    - push
    - pull_request
    - workflow_dispatch

Hands-On

- Create first workflow YAML
- Trigger workflow on push

Module 3: Workflow Syntax (YAML Deep Dive)

Objective: Master workflow configuration

Topics

- Workflow structure
- name, on, jobs
- Job dependencies (needs)
- Steps & shell commands
- Environment variables

- Contexts & expressions

Hands-On

- Build a basic CI pipeline
- Use environment variables

Module 4: Actions & Marketplace

Objective: Reuse and create actions

Topics

- What is an Action?
- Types of actions:
  - JavaScript actions
  - Docker actions
  - Composite actions
- GitHub Marketplace
- Version pinning

Hands-On

- Use official actions: actions/checkout, actions/setup-node
- Pin action versions securely

Module 5: Jobs, Runners & Parallelism

Objective: Optimize workflow execution

Topics

- Multiple jobs in workflows
- Parallel vs sequential jobs
- Matrix builds
- Self-hosted runners
- Runner security

Hands-On

- Configure matrix builds
- Run jobs in parallel

Module 6: Secrets & Environment Management

Objective: Secure pipelines

Topics

- GitHub Secrets
- Environment-level secrets
- Repository vs Organization secrets
- Masking secrets in logs
- GitHub Environments & approvals

Hands-On

- Store AWS credentials securely
- Use secrets in workflows

Module 7: Build & Test Automation

Objective: Implement CI pipelines

Topics

- Build pipelines for: Node.js, Java, Python
- Unit testing
- Test reports & artifacts
- Caching dependencies

Hands-On

- Build & test application
- Upload test artifacts

Module 8: Docker & GitHub Actions

Objective: Automate container workflows

Topics

- Docker build & push
- Docker Hub vs Amazon ECR
- Image tagging strategies
- Docker layer caching

Hands-On

- Build Docker image
- Push image to ECR using GitHub Actions

Module 9: GitHub Actions with AWS

Objective: Deploy applications to AWS

Topics

- AWS authentication methods:
    - Access keys
    - OIDC (recommended)
- Deploy to:
    - EC2
    - ECS
    - Lambda
- IAM least privilege for CI/CD

Hands-On

- Deploy app to EC2
- Configure OIDC with AWS IAM

Module 10: Infrastructure as Code with GitHub Actions

Objective: Automate infrastructure deployment

Topics

- Terraform in GitHub Actions
- Remote backend handling
- Plan & apply workflows
- Manual approvals for prod
- GitOps strategy

Hands-On

- Terraform CI pipeline: fmt, validate, plan, apply

Module 11: Advanced Workflow Features

Objective: Build enterprise-grade pipelines

Topics

- Reusable workflows
- Composite actions
- Workflow inputs & outputs
- Conditional execution
- Scheduled workflows (cron)

Hands-On

- Create reusable workflow
- Trigger scheduled job

Module 12: Security & Compliance

Objective: Secure CI/CD pipelines

Topics

- Least privilege pipelines
- Secret scanning
- Dependency scanning
- CodeQL overview
- SAST in GitHub Actions

Hands-On

- Enable CodeQL scanning
- Integrate security tools

Module 13: Monitoring & Troubleshooting

Objective: Debug workflows efficiently

Topics

- Workflow logs
- Debug mode
- Common failures
- Rerun jobs
- Timeout handling

Hands-On

- Fix failing workflows
- Enable debug logs

Module 14: Performance & Cost Optimization

Objective: Optimize pipeline execution

Topics

- Caching strategies
- Reduce workflow runtime
- Self-hosted runners use cases
- Cost considerations

Hands-On

- Implement dependency caching
- Measure execution time

Module 15: Real-World Use Cases

Objective: Apply knowledge to real scenarios

Topics

- PR validation workflows
- Release pipelines
- Blue-green deployments
- Multi-environment pipelines
- Monorepo strategies

Module 16: Capstone Project

Objective: Build a production-ready CI/CD system

Project: End-to-End DevOps Pipeline

- CI pipeline (build & test)
- Docker image build & push
- Terraform infrastructure provisioning
- AWS deployment
- Manual approval gates
- Security scanning