

DEVOPS

INTERMEDIATE

3 DAYS

GitHub Fundamentals for Enterprise Security and Automation

Build a strong foundation in GitHub automation at enterprise scale, from core workflows to reusable pipeline standards and governance. Learn how to implement secure CI/CD automation with traceability, approvals, compliance-ready controls, and developer-friendly practices for high-performing teams.

Why GitHub Enterprise?

Standardization across many repos

Reusable workflows reduce duplication and enforce consistent CI rules.

Compliance and audit support

Traceability from PR to release simplifies reviews and reporting.

Secure development at scale

Required checks and governance prevent risky merges and improve security posture.

Operational efficiency

Shared pipeline templates speed onboarding and reduce maintenance cost.

Enterprise delivery readiness

Environment approvals and controlled releases reduce deployment risk.

What you'll be able to do

- ✓ Understand GitHub fundamentals in an enterprise environment and adopt secure collaboration workflows.
- ✓ Apply governance controls including least privilege permissions, branch protection rules, CODEOWNERS, and required checks.
- ✓ Build robust GitHub Actions workflows with artifacts, caching, and environment approval gates.
- ✓ Enable supply chain security capabilities including secret scanning patterns, Dependabot-based dependency security, and code scanning concepts integrated into PR checks.
- ✓ Standardize automation at scale using reusable workflows and organization-wide pipeline patterns.
- ✓ Understand self-hosted runner use cases and security posture for enterprise environments.
- ✓ Apply best practices for auditing, compliance, and traceability across PRs, checks, and releases.
- ✓ Deliver an end-to-end workflow: commit to PR to checks to approvals to secure merge to CI and controlled deployment.

Who this is for

- Developers and backend engineers
- DevOps, Platform, and SRE engineers
- Security teams supporting developer workflows
- Tech leads and engineering managers

- Enterprise teams migrating from Bitbucket, GitLab, or Azure Repos to GitHub

Curriculum

01 GitHub Enterprise Fundamentals (Secure Collaboration Workflows)

- GitHub in enterprise environments (repos, orgs, teams, permissions model)
- Secure collaboration workflows (branches, PRs, reviews, merge strategies)
- Commit hygiene and safe practices (signing concept, meaningful history)
- Enterprise workflow expectations (auditability, consistency, traceability)
- **Hands-on:** Lab: Create a secure repo workflow with branches, PR reviews, and merge rules

02 Governance Controls (Least Privilege, Branch Protection, CODEOWNERS)

- Least privilege access (read, triage, write, maintain, admin)
- Branch protection rules (required reviews, required status checks)
- CODEOWNERS setup and enforcement patterns
- Required checks strategy (lint, test, scan, approvals)
- **Hands-on:** Lab: Configure branch protection + CODEOWNERS and validate enforcement via PR

03 GitHub Actions Workflows (Artifacts, Caching, Approval Gates)

- Workflow structure (triggers, jobs, steps, runners)
- Artifacts handling (build outputs, test reports, release packages)
- Caching strategies (deps cache, build cache) for faster pipelines
- Environment approvals and manual gates for controlled deployments
- **Hands-on:** Lab: Build Actions workflow with caching + artifacts + environment approval gate

04 Supply Chain Security in GitHub (Secrets, Dependencies, Code Scanning)

- Secret scanning patterns (prevent leaks, rotate, enforce PR checks)
- Dependabot security basics (alerts, updates, PR workflows)
- Code scanning concepts (static analysis signals and triage flow)
- Integrating security checks into PR requirements and quality gates
- **Hands-on:** Lab: Enable supply chain checks and wire them into required PR status checks

05 Standardizing Automation at Scale (Reusable Workflows)

- Reusable workflows and composite actions (when to use each)
- Organization-wide pipeline patterns (central templates)
- Versioning workflow templates and controlled rollout strategy
- Consistency and governance across multiple repositories
- **Hands-on:** Lab: Build a reusable workflow template and consume it from 2 repositories

06 Self-Hosted Runners (Enterprise Use Cases and Security Posture)

- When to use self-hosted runners (network, compliance, performance)
- Runner security risks (token exposure, repo trust boundaries)
- Isolation strategies (ephemeral runners, least privilege, sandboxing concepts)
- Hardening checklist (patching, secrets management, restricted permissions)
- **Hands-on: Lab:** Design a secure self-hosted runner strategy with isolation and governance controls

07 Auditing, Compliance, and Traceability (PRs to Releases)

- Audit trails across PRs, reviewers, checks, and merges
- Release traceability (commit → build → artifact → deployment)
- Tagging and release practices (semantic versioning, changelogs concept)
- Evidence collection for compliance (who approved what, when, and why)
- **Hands-on: Workshop:** Create an audit-ready release workflow checklist for an enterprise team

08 End-to-End Capstone (Commit → PR → Checks → Approval → Secure Merge → Controlled Deploy)

- Capstone goal: Deliver a complete enterprise-grade GitHub workflow
- Commit and open PR with CODEOWNERS-based review routing
- Run required checks (build, test, lint, supply chain scans)
- Use environment approvals for controlled deployments
- Secure merge strategy with full traceability and audit evidence
- **Hands-on: Capstone Lab:** Build and demonstrate the full end-to-end pipeline with governance + security gates