

CTR-811: Git GitHub and GitLab

Overview

Course Duration: 5 Days

Audience

developers, people who work with code in one way or another

Course Objective

By the end of the course, participants will:

- Master Git as a distributed version control system for solo and collaborative development.
- Gain hands-on experience with GitHub and GitLab, focusing on real-world workflows and industry standards.
- Understand how to manage, review, and integrate code in multi-developer environments.
- Build, automate, and secure CI/CD pipelines using both GitHub Actions and GitLab CI/CD.
- Learn best practices for permissions, access control, secrets management, and DevSecOps.

Course Outline

Part 1 – Foundations of Git, GitHub, and GitLab

Introduction to Git and Version Control Principles

- What is Git, and why is it essential in modern development?
- Core concepts: repositories, commits, branches, mergers, and staging.
- Understanding Git internals: working directory, index, local vs. remote.
- Managing and rolling back changes: revert, reset, checkout, and stash.

Working with Remote Repositories

- Setting up GitHub and GitLab accounts and connecting remote repositories.
- Cloning, forking, and maintaining synchronization with origin.
- Fetch, pull, push: understanding the data flow.
- Visualizing history and collaboration with graphical tools.

Branching Strategies and Collaborative Development

- Effective use of branches for features, bug fixes, and experiments.
- Introduction to Git workflows: centralized, feature branch, GitFlow, trunk-based.
- Collaborating through Pull Requests (GitHub) and Merge Requests (GitLab).
- Code review processes: inline comments, discussions, approval requirements.

Issue Tracking, Documentation, and Project Boards

- Managing development tasks with GitHub Issues and GitLab Issues.
- Milestones, labels, assignments, and workflows.
- Creating structured README files, contributing guidelines, and changelogs.
- Using GitHub Projects and GitLab Boards for agile task tracking.

Part 2 – Advanced Git Usage, CI/CD, and DevOps Automation

Advanced Git Features and Workflows

- Submodules, tags, and release management.
- Working with rebase, squashing commits, and resolving complex conflicts.
- Using hooks and custom Git aliases to improve productivity.
- Signing commits and enforcing security policies.

CI/CD Pipeline Fundamentals

- Introduction to continuous integration and continuous deployment principles.
- Comparing GitHub Actions vs. GitLab CI/CD.
- Anatomy of a pipeline: jobs, stages, runners, triggers, and artifacts.
- YAML configuration basics and best practices.

Automated Testing and Deployment

- Integrating testing frameworks into the pipeline.
- Generating test coverage reports and using them as deployment gates.
- Building and pushing Docker images automatically.
- Deploying to live environments: Heroku, AWS, Kubernetes, and custom servers.

Secrets Management and Access Control

- Managing environment variables and sensitive data securely.
- Role-based access control in GitHub and GitLab.
- Using organizational repositories, teams, and protected branches.
- Secrets injection, masked variables, and secure pipeline execution.

DevSecOps and Security Practices

- Static application security testing (SAST) with built-in tools.
- Dependency scanning (e.g., GitHub Dependabot, GitLab Dependency Scanning).
- Mitigating common threats: exposed credentials, misconfigured pipelines.
- Enforcing policies and audit trails in team environments.

Advanced Collaboration and Open Source Best Practices

- Working with forks and upstream contributions.
- Managing contributions in open-source projects.
- Community engagement via Discussions, Wikis, and Actions Marketplace.
- Governance models and contributor agreements.

Final Project and Course Completion

- Participants will work individually or in teams to deliver a real-world Git-based project.
- The project will include version control setup, branching strategy, collaboration simulation, and automated CI/CD pipeline.
- Final presentations include demonstrating version control best practices, pipeline execution, and deployment flow.

Instruction Methodology

- Hands-on labs and real-life scenarios.
- Peer collaboration exercises simulating real software teams.
- Instructor-led walkthroughs and guided challenges.
- Access to shared repositories and sandbox environments.

Course Outcomes

Participants will finish the course with:

- Strong Git fundamentals and advanced collaboration skills.
- Practical experience in managing and contributing to repositories on GitHub and GitLab.
- Hands-on understanding of DevOps pipelines, secrets, and deployment.
- Readiness to implement Git-centric CI/CD workflows in their own work environments.