

**CI/CD with GitLab**

**Course Number:** GIT-106WA  
**Duration:** 2 days

**Overview**

Git has fast become the most popular version control system for engineering teams and is gaining popularity as a means to automate and manage infrastructure/DevOps. In addition, platforms like GitLab, GitHub, and Bitbucket aim to replace popular CI/CD tools such as Jenkins and CircleCI.

This CI/CD with GitLab training course teaches Infrastructure and DevOps teams how to implement continuous integration and continuous deployment (CI/CD) pipelines leveraging Git and GitLab.

**Prerequisites**

Some knowledge of Java is beneficial.

**Materials**

All attendees receive comprehensive course materials.

**Software Needed on Each Student PC**

* Git 2.x or later
* Internet access for all attendees and the instructor

**Objectives**

* Work with version control using Git
* Branch, merge, and work with Remotes
* Work with GitFlow Workflow
* Work with GitLab.com

**Outline**

* DevOps Fundamentals
  + Why DevOps?
  + What is DevOps?
  + Collaborative, Matrixed, and Cross-Functional Teams
  + Key Components of Successful DevOps Teams
  + DevOps-ification
  + DevOps Vocabulary
  + DevOps Goals
  + Not DevOps - Crush Buzzwords
  + Driving Business Outcomes with DevOps
  + Technology-Enabled Business
  + DevOps Key Enabler for Digital Transformation
  + Core Values and Mission
  + Communication
  + Collaboration
  + Value Stream Mapping
  + Behavioral Patterns for Success
  + DevOps Org Structures
  + DevOps Team - Separate
  + DevOps Merged Organization
  + DevOps Overlapped Organization
  + Organizational Structure Leadership
  + What Does Continuous Delivery Mean?
  + Deployment Pipelines
  + Your Organization is Doing CD if …
  + Pipelining for CD
  + Continuous Integration
  + CI Pipeline
  + CD & CI Methodologies
  + Key Tool Categories for CI/CD
* Introduction to Git
  + What is Git
  + Git's Design Goals
  + Branching and Merging
  + Centralized Version Control
  + Distributed Version Control
  + Git Basics
  + Getting Git
  + Git on the Server
  + Git Repository Managers
  + Git on Someone's Server
* Basic Git Operations
  + Using Git
  + Definitions
  + Commit
  + How to Think About Commits
  + Viewing History
  + Configuring Git
  + Configuration Scope
  + User Identification
  + GPG Signing
  + Gnu Privacy Guard
  + GPG Basics
  + GPG and Git
  + .gitignore
  + Other Useful Configurations
  + Gitattributes
  + Where Gitattributes are Set
  + How Gitattributes are Set
  + Git Hooks
  + commit-msg Hook with Jira Key
* Branching, Merging, and Remotes
  + Branching
  + Branches in Git
  + Merge
  + Fast Forward Merge
  + --no-ff
  + More Than One Repository
  + Working with Remotes
  + Fetch and Pull
  + Push
  + Pull Requests
  + Tagging a Commit
  + Lightweight Tags
  + Annotated Tags
  + Sharing Tags
  + Checking Out a Tag
* Introduction to Git Flow
  + Why Use an SCM Workflow?
  + What is Git Flow
  + The Main Branch
  + Benefits
  + What is Git Flow and how does it work?
  + Git Flow Extension
  + Initializing Git Flow
  + Features
  + Release
  + Hotfixes
  + Git Flow and Continuous Integration
  + Git Flow Alternatives
  + Trunk-based Development
  + GitHub Flow
  + GitHub Flow – Pros and Cons
* Introduction to GitLab
  + What is GitLab
  + History
  + Benefits
  + Features
  + Advantages
  + Disadvantages
  + Setting up an Account
  + Create New Project
  + Choice for Projects
  + GitLab Commands
* Introduction to CI/CD and GitLab
  + Foundation of Agile AppDev
  + Extreme Programming
  + Agile Development
  + What is Continuous Integration (CI)
  + What is Continuous Integration (cont'd)
  + Typical Setup for Continuous Integration
  + Setup Notes for Continuous Integration
  + CI with Artifact Management
  + What is Continuous Delivery (CD)?
  + Why Continuous Delivery?
  + DevOps and Continuous Delivery
  + Continuous Delivery Challenges
  + Continuous Delivery vs Continuous Deployment
  + GitLab CI/CD
  + GitLab CI/CD (Cont)
  + Running GitLab
  + Running GitLab (cont)
  + Creating a GitLab Project
  + Creating a GitLab Project (cont)
  + GitLab via YAML Templates
* Introduction to GitLab Flow
  + Environment Branches
  + Release Branches
  + Pros and Cons
* Best Practices for GitLab
  + Secure GitLab
  + Users
  + Build Directory
  + Shutdown
  + CI/CD
  + Optimize Pipeline Stages
* Conclusion