

Updated on 04/11/2025

Register

# Git Training: GitLab & GitFlow

2 days (14 hours)

## Overview

In just a few years, [Git](#) has become the most widely used source control system in the world. Based on decentralized version control, Git is free, cross-platform software created by the equally famous Linus Torvalds. Designed to be simple and powerful, Git's main task is to manage changes to the content of a directory tree.

In this training course, you will learn how to harness its full power, use the many tools associated with this project, such as GitLab and GitHub, and collaborate effectively as a team using Gitflow.

We will begin by understanding all the subtleties of Git in development mode through the use of basic and then advanced commands, ending with history management and interaction with remote repositories.

We will also take an in-depth look at GitLab, which since its creation has developed a modern, simple, and clear infrastructure to become an essential web project management tool. More than just a source code manager, GitLab allows you to manage access rights, bug reports, and sprints in your software lifecycle.

GitLab also automates project deployment to production, and with its comprehensive tools, it enables optimal integration and continuous development with GitLab CI/CD (Continuous Integration & Delivery), which will support you in your DevOps continuous integration efforts.

Finally, GitFlow is a collaborative multi-branch workflow that allows you to develop and control your branches in order to better manage customer requests or their documents. Git Flow is a method, a Git architecture that allows you to separate work as much as possible and touch the Master branch as little as possible. This method therefore represents a very efficient branch architecture to help you with your web and software development.

Two à la carte modules are available for your private training courses:

- Having trouble migrating to Git? Our intra-company add-on module is made for you: in this optional section, we offer to assist you with migrating from SVN (Subversion) to GIT.
- Want to get started with GitLab administration? We offer an add-on module add-on module (intra-company only): in this optional section, we teach you how to install, deploy, and administer the GitLab solution.

As with all our training courses, we will use the latest version of the software ([GitLab 17.8](#) and [Git-Flow AVH 1.12](#)).

## Objectives

- Master the basic commands and fundamental concepts of Git
- Effectively manage local and remote repositories as well as collaborative interactions with Git
- Use continuous integration and deployment (CD) tools with GitLab CI
- Use advanced Git commands and solve complex problems
- Adopt and use GitFlow and GitLab for collaborative workflows and continuous integration (CI)

## Target audience

Developers, Architects, System Administrators

## Prerequisites

Basic knowledge of a source manager

## Technical prerequisites

- Git tool installed

## Our GitLab CI & GitFlow with GIT training program

### Day 1 - Git & GitFlow

#### Introduction

- Context and history of version management
- Use cases
- Presentation of Git
- Advantages of Git

## Basic commands

- Creating the repository, .git folder
- The 3 states of Git
- Add and commit commands
- Status, log, and diff commands

## Branches

- Benefits of branching
- Pointers and HEAD
- Divergence
- Different merging methods
- Best practices for merging

## Remotes

- Central repository and remote multi-repository
- Fetch and pull commands
- The push command
- Best practices for interaction

## Advanced commands

- Reset and reflog
- Rebasing, Squashing
- Stash management
- Blame
- Error Search with Bisect
- Cherry-pick

## Day 2 - Git Flow and GitLab

### Collaborative Workflows

- Why use a workflow
- The main workflows
- Focus on Git Flow and putting it into practice
- Github Flow and GitLab Flow
- Introduction to GitLab
- Software forges
- GitLab features
- How to choose a software forge
- CE vs EE

### GitLab & GitFlow

- Feature/bugfix branches
- Release branches (and tags)
- Merge Request (MR) concepts
- Explore through a practical exercise the interactions and collaborations that can be done with an MR, the functions that assist code review, and the subtleties that can arise
- The ideal workflow
- External tools that can complement GitLab in this area

## GitLab Issues

- Introduction to issues
- Gradually evolve your workflows to encourage adoption, rather than being too disruptive
- A demonstration & live demo

## GitLab CI

- Do a CI practical exercise, at least detailing the entire GitLab part (and avoiding Docker)
- Show a simple case with a unit test CI (without Docker)
- Show a more interesting case with functional tests. (with a simple Docker case)

## Best practices for collaboration with Git and GitLab CI

- Merge requests
- Rebase (+squash, +fast-forward)
- Protected branches
- Wiki and documentation
- Continuous integration with GitLab CI
  - GitLab CI/CD configuration
  - CI/CD pipelines
  - Runner management
- Continuous Deployment and GitLab Runner
- Setting up a collaboration workflow on GitLab

## **GITLAB ADMINISTRATION (ADD-ON MODULE + 2 days)**

Harness the full power of GitFlow as an administrator

## Installation and deployment of GitLab

- Omnibus vs. Docker
- Installation on GCP
- GitLab SMTP configuration
- Low-level configuration

## GitLab Administration

- Overview of the management interface
- Project administration
- Commit and source code management
- Access management
- Assigning permissions
- Use of issues
- Milestones for project management

## **MIGRATING FROM SUBVERSION TO GIT (ADD-ON MODULE + 1 day)**

Use GIT in multi-repository mode and modify your history

### GIT VS SVN TERMINOLOGY (QR)

### MIGRATING FROM SVN TO GIT

- Feedback
- The different methods

### THE GIT-SVN COMMAND

- Migration in 2 steps
- Advantages and disadvantages
- Practical work: Migrating an SVN repository to GIT

### MANAGING MULTIPLE REMOTE REPOSITORIES WITH GIT

- Contributing to a single repository
- Contributing to multiple repositories
- Different integration modes
- Example of maintaining a complex repository: the GIT project
- Practical work: Managing multiple remote repositories

### BASIC COMMANDS AND MIGRATION

- Basic GIT commands
- Practical work: Migration from SVN to GIT

### ADVANCED GIT COMMANDS

### OBJECT TYPES

- Commit
- Blob
- Tree
- Tag

## PORCELAINE

- Develop
- Inspect
- Share
- Merge
- Patch
- Debug
- Administer
- TP: Manage a complete versioning cycle

## GIT HOOKS & SUB-REPOS SUBTREE VS SUBMODULES

- When to use a subtree or a submodule
- Use the native git submodule command
- Create a sub-tree manually
- Use a script: git-subrepo GIT

## ATTRIBUTES

- Handling binary files
- Bi-directional filters (clean & smudge)
- Archiving attributes
- Merge strategy

## GIT HOOKS

- Hook environments
- Installing client-side hooks
- Install server-side hooks

## Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in new advanced IT technology or to acquire specific professional knowledge or modern methods.

## Placement at the start of training

The positioning at the start of the training complies with Qualiopi quality criteria. Upon

final registration, learners receive a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for the upcoming training, within the limits imposed by the selected format. This questionnaire also allows us to anticipate certain connection or internal security issues within the company (intra-company or virtual classroom) that could be problematic for the monitoring and smooth running of the training session.

## Teaching methods

Practical training: 60% practical, 40% theory. Training materials distributed in digital format to all participants.

## Organization

The course alternates between theoretical input from the trainer, supported by examples and discussion sessions, and group work.

## Assessment

At the end of the session, a multiple-choice questionnaire is used to verify that the skills have been correctly acquired.

## Certification

A certificate will be issued to each trainee who has completed the entire training course.