

Updated 05/17/2024

Sign up

Git training : GitLab & GitFlow

2 days (14 hours)

Presentation

In just a few years, [Git](#) has become the world's most widely used source manager. Based on decentralized versioning, Git is a free, cross-platform software created by the no less famous Linus Torvalds. Simple yet powerful, Git's main task is to manage the evolution of tree content.

In this training course, you'll learn how to harness its full power, use the many tools available for this project, such as GitLab and GitHub, and collaborate effectively in a team thanks to Gitflow.

We'll start by getting to grips with all the subtleties of Git in development mode, through the use of basic commands, then advanced ones, ending with history management and interaction with remote repositories.

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

GitLab also enables you to automate the deployment of projects to production, and with its comprehensive tools, it will enable you to optimize integration and continuous development with GitLab CI/CD (Continuous Integration & Delivery), which will support you in your DevOps approaches to continuous integration.

And last but not least, GitFlow, the multi-branch collaborative workflow that will enable you to elaborate 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 your work as much as possible, and touch the Master branch as little as possible. This method therefore represents a highly efficient branch-based architecture to help you with your web & software development.

Having trouble migrating to Git? Our add-on module is for you: in

In this optional section, we offer to help you migrate from SVN (Subversion) to GIT.

As with all our training courses, we'll be using the latest software versions ([GitLab 17.0](#) and [Git-Flow AVH 1.12](#)).

Objectives

- Install and configure the GIT tool
- Understanding the differences between centralized and decentralized version management
- Efficiently manage local and remote depots
- Using GIT commands to manage files and branches
- Workflow and repository management with GitFlow
- Using continuous integration and deployment tools with GitLab CI
- Increase efficiency by using the most effective tools in the GIT ecosystem
- Quickly integrate and train a newcomer to your project
- Understanding the migration from SVN to GIT

Target audience

Developers, Architects, System Administrators

Prerequisites

Basic knowledge of a source manager

Technical requirements

- Have Git installed

GitLab CI & GitFlow with GIT training program

Day 1 - Git & GitFlow

Introduction

- Background and history of version control
- Use cases
- Introducing Git
- Benefits of Git

Basic controls

- Repository creation, .git folder
- Git's 3 states
- Add and commit commands
- Status, log and diff commands

Branches

- Benefits of branching
- Pointers and HEAD
- Divergence
- The different fusion methods
- Merging best practices

Remotes

- Central and remote multi-depot facilities
- Fetch and pull commands
- The push command
- Good interaction practices

Advanced controls

- 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
- Zooming in 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

Installing GitLab

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

GitLab administration

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

Best practices for working 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 collaborative workflow on GitLab

GitLab CI Advanced (SUPPLEMENTARY MODULE + 2 days) Use

the full power of GitFlow and GitLab's powerful CI chain **GitFlow**

- Features/bugfix branches
- Version branches (and tags)

Installing and deploying GitLab

- Both installation modes will be presented (docker & .deb)
- GitLab administration feedback
- At the end of the training, we will leave a working instance on their server.

GitLab & GitFlow

- Merge Request (MR) concepts
- Explore through a hands-on session the interactions and collaborations that can be made with a PM, the functions that assist code review, and the subtleties that can arise.
- The ideal workflow
- External tools that can complement GitLab in this section

GitLab Issues

- Outcome presentation
- Evolve your workflows gradually to encourage adoption, rather than being too disruptive
- A demonstration & Live demo

GitLab CI

- Do a TP CI, to at least detail the GitLab part (and avoid Docker)
- Show a simple case study with unit test CI (without Docker)
- Show a more interesting case with functional tests. (with a simple Docker case)
- Conclusion

MIGRATE FROM SUBVERSION TO GIT (ADD-ON + 1 day)

Use GIT as a multi-depository and modify your history

TERMINOLOGY GIT VS SVN (QR)

MIGRATING FROM SVN TO GIT

- Feedback
- The different methods

THE GIT-SVN COMMAND

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

MANAGE MULTIPLE REMOTE REPOSITORIES WITH GIT

- Contribute to a single deposit
- Contribute to multiple repositories
- The different modes of integration
- Example of complex repository maintenance: the GIT project
- Practical work: Managing several remote repositories

BASIC COMMANDS AND MIGRATION

- Basic GIT commands
- Practical work: SVN > GIT migration

ADVANCED GIT COMMANDS

OBJECT TYPES

- Commit
- Blob
- Tree
- Tag

PORCELAIN

- Develop
- Inspect
- Share
- Merge
- Patcher
- Debug
- Administer
- TP: Managing a complete versioning cycle

GIT HOOKS & SUB-REPOS SUBTREE VS SUBMODULES

- When to use a subtree or submodule
- Use the native git submodule command
- Making a sub-tree by hand
- Using a script: git-subrepo

GIT ATTRIBUTES

- Binary file processing
- Bi-directional filters (clean & smudge)
- Archiving attributes
- Merger strategy

GIT HOOKS

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

Companies concerned

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

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

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

Validation

At the end of the session, a multiple-choice questionnaire is used to check correct acquisition.

skills.

Sanction

A certificate will be issued to each trainee who completes the course.