+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 29/09/2025

Sign up

Jujutsu Training

3 days (21 hours)

Jujutsu Training

Our training course introduces you to Jujutsu (jj), a new version management system designed to overcome the limitations of Git. Based on the concept of mutable changesets, it enables history to be rewritten and restructured with greater flexibility, while remaining interoperable with Git.

Designed for modern teams, it promotes collaboration, implicit branch management and seamless integration into CI/CD pipelines.

This Jujutsu training course will enable you to master this new tool, adopt more efficient workflows than Git, and integrate it into your DevOps projects.

You'll learn how to manage changes, collaborate as a team, secure your CI/CD processes and manage your development projects with greater visibility.

At the end of this course, you'll be able to deploy Jujutsu in your environments, train your teams and improve your versioning practices.

Like all our training courses, this one is based on the latest stable version of Jujutsu and its new features.

Objectives

- Understand the fundamental concepts of Jujutsu and changesets.
- Migrate and interoperate with existing Git repositories.
- Integrate Jujutsu into DevOps and CI/CD workflows.
- Efficiently manage history, branches and merges.
- Manage agile projects with Jujutsu as a versioning engine.

Target audience

- DevOps engineers
- Backend and fullstack developers
- Technical project managers

Prerequisites

- Basic knowledge of Git and version control
- Notions of CI/CD and DevOps practices

Jujutsu training program

[Day 1 - Morning]

Introduction to Jujutsu and version control systems (VCS)

- Why Jujutsu: Git's limitations and change-centric approach
- Key concepts: changesets, mutable history, Git interoperability
- Installation and initial configuration of jj
- Tour of basic commands: jj init, jj status, jj commit
- Import/export with Git repositories
- Practical workshop: Creating a jj repository and managing its first changes.

[Day 1 - Afternoon]

Managing history and changesets

- Git commits vs jj changesets
- Fluid rewriting: amend, reword, reorganize
- Multiple heads and history navigation (jj log)
- Secure restoration and rollback
- Best practices for describing changes
- Practical workshop: Restructuring a complex history.

Branches, merges and collaboration with Git

- Implicit jj branches: principles and implications
- jj - Git synchronization
- Intelligent merge and rebase, conflict resolution
- GitFlow workflows vs. jj approaches
- Incremental migration from Git
- Practical workshop: Migrating an existing Git workflow to jj.

[Day 2 - Morning]

Collaborative workflows and reviews

- Publishing and sharing changesets
- Strategies for distributed teams
- Interactive conflict resolution
- Proofreading, validation and continuous integration
- Traceability and auditing of operations
- Practical workshop: Simulated multi-developer collaboration.

[Day 2 - Afternoon] CI/CD

integration

- Jenkins / GitLab CI / GitHub Actions: triggering on changesets
- Artifact and version management with jj
- Performance testing in the pipeline
- Supervised blue/green and canary strategies
- Reporting and continuous quality
- Practical workshop: CI/CD pipeline based on jj.

Project management

- Branch governance and deliverable tracking
- Progress indicators and historical visibility
- Documentation and auditability
- Agile & DevOps alignment
- Cadences, rituals and synchronization
- Practical workshop: Building a project workflow with jj.

[Day 3 - Morning]

Customization and advanced cases

- Aliases, scripts and productivity
- IDE, Docker and Kubernetes integrations
- Squash, amend, multi-changeset rebase
- Backing up and restoring repositories
- Open source and enterprise use cases
- Practical workshop: Customizing jj for the target environment.

[Day 3 - Afternoon] Optimization

and best practices

- Efficient branching strategies with jj
- Keeping a clear and consistent repository
- Performance on large repositories
- Monorepo: Git vs jj comparisons
- Pitfalls to avoid and checklists
- Practical workshop: Audit and optimization of a DevOps workflow.

Production roll-out and adoption

- Deploying jj in an existing team
- Progressive migration from Git
- Feedback from early adopters
- Integration with SRE and DevSecOps
- Roadmap and change management
- Practical workshop: Team migration plan.

Companies concerned

This 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 enabling 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 forthcoming course, 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 training: 60% hands-on, 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 that skills have been correctly acquired.

Certification

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