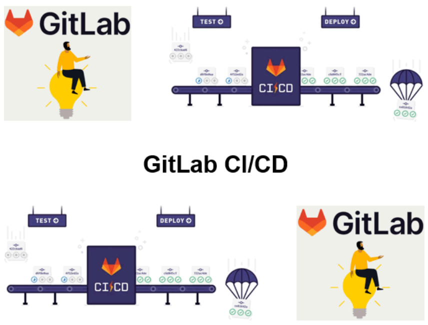
GitLab CI/CD

Duration: 2 days **Price: € 1499**

Open Schedule



Audience course GitLab CI/CD

The course GitLab CI/CD is intended for DevOps engineers, Software Developers and QA engineers who want to learn pipelining with GitLab.

Prerequisites GitLab CI/CD Course

To participate in the course, basic knowledge of Git, version control and software workflows is required. Familiarity with containers is useful.

Realization training GitLab CI/CD

The course is conducted under guidance of the trainer and theory and practice are interchanged. Real world case studies are used for explanations.

GitLab CI/CD Certificate

After successfully completing the course, participants will receive a certificate of participation in GitLab CI/CD.

Content Course GitLab CI/CD

In the course GitLab CI/CD, you will learn how to design, implement, and troubleshoot GitLab pipelines to automate building, testing, and deploying applications. The course covers writing pipelines, automating builds, handling deployments, integrating Docker, and applying best practices for performance and security.

Introduction to GitLab CI/CD

This module introduces the fundamentals of CI/CD and explains GitLab's role in automating software delivery. Participants learn about GitLab Runners, key components like jobs, stages, artifacts, and caching. The differences between shared and specific runners are discussed, along with best practices for setting up a reliable CI/CD environment.

Writing GitLab Pipelines

Participants learn how to structure and write efficient GitLab pipelines using the `.gitlab-ci.yml` file. Topics include defining jobs and stages, setting dependencies, using variables, and implementing parallel or sequential job execution. Security practices such as dependency scanning and pipeline maintainability are also covered.

Automating Builds

This module focuses on automating build processes within GitLab CI/CD. Participants learn how to run unit tests, integration tests, and implement linting. Emphasis is placed on handling failures, setting up notifications, integrating code quality and static analysis tools, and automating both build and testing stages in the pipeline.

Troubleshooting Pipelines

Participants learn techniques for diagnosing and fixing pipeline issues. Topics include understanding GitLab logs, artifacts, and common CI/CD errors. Strategies for retrying failed jobs, conditional execution, use of job tokens, and debugging broken pipelines are explored in depth.

Deployments with GitLab

This module teaches how to manage deployments using GitLab environments and jobs. Participants explore deployment strategies like blue-green and canary deployments, adding security checks, and managing secrets and environment variables to ensure safe and efficient deployment processes.

Docker in GitLab CI/CD

The course concludes with using Docker within GitLab CI/CD pipelines. Participants learn how to build, push, and deploy Docker images to Kubernetes and cloud providers, optimize performance through caching, work with distributed runners, and integrate Docker containers seamlessly into GitLab pipelines.

SpiralTrain developer courses

Modules Course GitLab CI/CD

Module 1: Intro to GitLab CI/CD	Module 2: Writing GitLab Pipelines	Module 3: Automating Builds
What is CI/CD? Overview of GitLab Continuous Integration Continuous Deployment GitLab Runners Shared vs. Specific GitLab Jobs Key GitLab Components Stages Artifacts Caching	.gitlab-ci.yml Structure Defining Jobs Defining Stages Dependencies Parallel Job Execution Sequential Execution Using Variables Maintainable pipelines Multi-stage CI/CD Pipeline Implementing Security Scans Dependency Scanning	Automating Builds Running Unit Tests Integration tests Linting in CI/CD Handling Failures Notifications Tools Integration Code Quality Tools Static Analysis Tools Build Automation in Pipeline Test Automation in Pipeline
Module 4: Troubleshooting Pipelines	**Module 5: Deployments with GitLab**	**Module 6: Docker in GitLab CI/CD**
Understanding GitLab Logs Understanding Artifacts Common CI/CD errors Fixing Errors CI/CD Job Tokens Retry Strategies Conditional Execution Debugging Broken Pipeline	Deployment Strategies Using environments Deployment jobs Blue-Green Deployments Canary Deployments Adding Security Checks Managing Secrets Managing Environment Variables	Deploying to Docker Deploying to Kubernetes Cloud Providers Pushing GitLab Container Registry Docker Images in Pipelines Jobs inside Docker Containers Optimizing Performance with Caching Distributed Runners