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

Updated on 23/06/2025

Sign up

Screwdriver.CD training

3 days (21 hours)

Presentation

Our Screwdriver.cd training course will enable you to deploy an open source CI/CD platform designed for cloud-native and microservices environments. Designed by Yahoo and now incubated by the Continuous Delivery Foundation, it orchestrates tens of thousands of containerized builds every day, from commit to automated deployment.

Our Screwdriver.cd training course will teach you how to industrialize your build, test and deployment pipelines, connect the platform to GitHub or GitLab, secure your keys and tokens, use reusable templates and publish your images to a Docker registry.

At the end of our program, you'll be able to install and configure the latest stable version of Screwdriver.cd, understand its modular architecture and, above all, build highly scalable CI/CD workflows for your cloud-native applications as well as your more traditional workloads.

Like all our training courses, it will run on the latest version of the tool: Screwdriver 4.1

Objectives

- Understand Screwdriver architecture
- Define complete as-code pipelines
- Manage secrets and security
- Integrate Screwdriver.cd into your DevOps system

Target audience

- DevOps / SRE engineers
- Full-Stack developers
- Cloud-Native architects

- Technical project managers

Prerequisites

- Basic knowledge of Linux, command line
- Understanding of Git workflows (branch, PR)
- Basic knowledge of Kubernetes and Docker

Technical prerequisites

- Git installed and access to a GitHub or GitLab account

Our Screwdriver.CD training program

Genesis and vision of Screwdriver.cd

- Historical background: from legacy CI tools to Yahoo!
- Key objectives: simplicity, scalability, pipelines as code
- Place in CDF and open source governance
- Key use cases (microservices, cloud-native)
- Overview of production successes (60 k builds/day at Yahoo)

Platform anatomy

- Macro view: API, UI, Launcher, Queue, Store
- Executor plug-in: Docker, Kubernetes, Jenkins, Nomad
- Datastore and state management (MySQL, PostgreSQL, SQLite)
- Build lifecycle from A to Z
- Security: container isolation & encrypted secrets

Setting up your CI/CD forge

- System requirements: Docker, Git, OAuth SCM
- Choice of deployment : Docker Compose vs Helm Chart
- Initial configuration (config/local.yaml)
- OAuth connection (GitHub/GitLab/Bitbucket)
- Validating the instance via the UI
- Workshop: Deploying Screwdriver.cd locally
- Start Docker Compose stack, connect, check API

Pipelines as Code

- Structure of screwdriver.yaml file
- Jobs, Steps, Stages: fan-in / fan-out, manual jobs
- Triggers: commit, PR, remote pipeline, cron
- Inter-job variables & metadata
- Viewing and restarting a pipeline in the UI
- Good naming and cutting practices
- Practical workshop: Creating a complete Node.js pipeline
- Lint, tests, Docker build image, artifact publication

Secrets & dynamic configuration

- Concepts: scope, allowInPR, rotation
- Adding via UI / CLI / API
- Injection into build containers
- Access security and auditing
- Usage models (API keys, kubeconfig, Slack tokens)

Advanced execution and artifact management

- Choosing your executor: decision criteria
- Isolation, performance and scaling
- Shared build cache between jobs
- Store: logs, files, coverage reports
- Debugging: hooks, restarts, configurable timeouts

Reusability: templates & commands

- Pipeline templates (sd-template.yaml)
- Shared commands (sd-command.yaml)
- Template publication and versioning
- Parameter inheritance and overloading
- Spreading DevOps best practices at scale

Integrations & notifications

- SCM: PR checks, commit status
- Notifications: Slack, Discord, e-mail, custom webhooks
- Docker registries: automated push (ECR, ACR, Docker Hub)
- SonarQube, Snyk: quality & security scans in the chain
- Cross triggers: synchronize multiple microservices
- Practical workshop: Pipeline with Slack notification and push image
- Adding secret webhook, image publication, success/failure message

Continuous deployment & extensions

- Canary / blue-green deployment on Kubernetes

- Multi-environment pipelines (Dev ? Staging ? Prod)
- Executor VM for sensitive workloads
- Platform monitoring and metrics
- Migration strategies from other CI/CD tools
- Project roadmap & open source contribution

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced IT 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 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 verifies the correct acquisition of skills.

Certification

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