AMBIENT-IT

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

Updated on 04/09/2026

Sign up

Dagger Training

3 days (21 hours)

Overview

Dagger is a modern software delivery automation platform designed to make pipelines more reliable, portable, and programmable. This solution allows you to build, test, and deploy code in containerized environments, whether locally, in a CI/CD tool, or in the cloud.

Our Dagger training will help you master a "pipeline as code" approach that is more robust than fragile shell scripts or proprietary YAML files.

You will learn to write workflows in actual code, leverage the incremental cache, ensure reliable execution between local machines and continuous integration, and make your build and delivery pipelines more reusable.

By the end of the course, you'll be able to design readable, maintainable, and portable Dagger pipelines, integrate them with your existing CI tools, share steps across projects, and improve visibility into your workflows using traces, logs, and metrics.

Like all our training courses, this one will introduce you to **the latest stable version** of the technology and its new features.

Objectives

- Understand Dagger's architecture and role within a CI/CD pipeline.
- Write pipelines in code rather than in shell scripts or proprietary YAML.
- Run reproducible workflows locally, in CI, and in the cloud.
- Leverage the incremental cache to speed up builds and tests.
- Integrate Dagger with an existing orchestrator such as Jenkins, GitHub Actions, or GitLab CI.
- Standardize reusable build, test, packaging, and deployment steps.

Target Audience

- DevOps engineers
- Platform engineers
- SREs
- Backend or full-stack developers
- Tech leads and software industrialization managers

Requirements

- Basic knowledge of containers and CI/CD
- Proficiency in a common development language
- Proficiency with basic command-line commands
- Understanding of builds, automated testing, and application deployment

Dagger Training

[Day 1 - Morning]

Dagger Fundamentals and Modern Automation

- Understanding what Dagger is and its role in the CI/CD ecosystem
- Identifying the limitations of YAML, shell, and proprietary logic pipelines
- Discover the "pipeline as code" approach and its operational benefits
- Presenting the architecture: Engine, CLI, SDK, system API, REPL
- Position Dagger relative to Jenkins, GitHub Actions, and GitLab CI
- Hands-on workshop: Installing Dagger, getting started with the CLI, and running your first local pipeline.

[Day 1 - Afternoon]

Containerized execution and reproducibility

- Understanding the role of containers in Dagger execution
- Mastering the concept of reproducible environments between local and CI
- Leveraging the incremental cache to speed up workflows
- Managing explicit dependencies and task isolation
- Avoiding behavior discrepancies between developer workstations and the CI pipeline
- Hands-on workshop: Run a build/test suite in a containerized environment that is identical locally and in CI.

First pipelines in code

- Structuring a Dagger pipeline with readable and reusable functions
- Using an SDK to write the pipeline's business logic
- Chaining build, test, and release steps
- Organizing inputs, outputs, and artifacts
- Adopt best practices for readability, modularity, and maintainability
- Create an initial pipeline ready for integration into an existing orchestrator

[Day 2 - Morning]

Integration with existing CI/CD tools

- Understand how Dagger integrates with an orchestrator without necessarily replacing it
- Call Dagger from Jenkins, GitHub Actions, GitLab CI, or other compatible CI tools
- Distinguish between external orchestration and internal pipeline logic
- Standardize the same workflow across multiple CI platforms
- Reducing dependency on a specific provider
- Hands-on workshop: Connecting a Dagger pipeline to an existing CI runner.

[Day 2 - Afternoon]

Managing artifacts, images, and releases

- Compiling an application and producing versioned artifacts
- Build a container image in a Dagger workflow
- Publish images and deliverables to a registry or artifact repository
- Secure delivery operations with strict input management
- Standardize packaging and deployment processes
- Implement reusable steps across multiple projects

Modularization and reuse of workflows

- Factor common steps into reusable components
- Share pipelines across teams and projects
- Design an internal library of Dagger functions
- Organize pipeline code to promote scalability
- Define structuring and versioning conventions
- Hands-on workshop: Refactoring a monolithic pipeline into reusable modules.

[Day 3 - Morning]

Observability, Debugging, and Understanding Execution

- Leveraging logs, traces, and metrics provided by Dagger
- Understanding the causes of failure in a step or pipeline
- Diagnosing performance bottlenecks related to the cache, network, or dependencies
- Implementing a pipeline-oriented debugging approach
- Make workflows more readable for delivery teams
- Manage execution with improved operational visibility

[Day 3 - Afternoon]

Security, secrets, and best practices for industrialization

- Manage secrets without exposing them in code or logs
- Securing access to registries, repositories, and target environments
- Master best practices for maintainable and portable pipelines
- Reducing anti-patterns related to shell scripts and CI hacks
- Preparing a production-ready pipeline
- Hands-on workshop: Integrating secrets and making a release workflow reliable.

Full implementation on a real-world use case

- Design a complete build, test, and deploy pipeline
- Combine modularity, caching, observability, and portability
- Industrialize a real-world project with a clear pipeline logic
- Measure gains in reliability, maintainability, and reusability
- Prepare for the gradual adoption of Dagger within an organization
- Hands-on workshop: Building a complete professional software delivery pipeline with Dagger.

Target Audience

This training is intended for 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.

Assessment upon enrollment

The pre-training assessment complies with Qualiopi quality standards. Upon final registration, the learner receives a self-assessment questionnaire that allows us to evaluate their estimated proficiency in various types of technologies, as well as their expectations and personal goals regarding 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 pose challenges for monitoring and ensuring the smooth running of the training session.

Teaching Methods

Practical Course: 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 reflection sessions, and group work.

Assessment

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

Certification

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