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

Updated on 12/19/2025

Register

Bazel Training: Build, Test, and CI/CD at Scale

3 days (21 hours)

Overview

Bazel is an open-source build system for enterprises. This industrialization platform for multi-language and multi-platform applications provides tools for incremental and reproducible compilation, testing, and packaging.

Our Bazel training: Build, Test, and CI/CD at Scale will enable you to structure your projects, manage dependencies, and accelerate your pipelines by leveraging caching and remote execution.

You will be able to obtain more information on build performance, configure a complete Bazel environment, and define CI/CD and security best practices.

Bazel makes it easy to orchestrate tests and correlate results with pipeline metrics. What's more, Bazel adapts to monorepo repositories and distributed architectures, from the developer's workstation to CI agents.

After completing this training, you will be able to install, configure, and use Bazel, as well as understand its architecture and optimization mechanisms. Like all our training courses, this one will introduce you to the latest stable version and its new features.

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

Objectives

- Understand the architecture and key concepts of Bazel
- Industrialize a multilingual build environment
- Reduce build times through caching and remote execution
- Automate testing and integrate Bazel into CI/CD

- Ensure portability, security, and traceability of artifacts

Target audience

- Software developers
- DevOps engineers
- Application architects

Prerequisites

- Proficiency in command line and Git
- Basic development skills (Java, Python, C++, or Go)
- Basic knowledge of continuous integration

Our training program Bazel: Build, Test, and CI/CD at scale

Introduction to Bazel and fundamental concepts

- Understanding Bazel's architecture and positioning
- Comparison with Maven, Gradle, Make, and other tools
- Installation and initial configuration on your workstation
- Role of WORKSPACE and BUILD files
- Managing external dependencies and Bzlmod
- Workshop: Installation and initial configuration of a Bazel project

Project structure and dependency management

- Organization into packages, targets, and multilingual monorepo
- Native rules: cc_binary, java_library, py_test, etc.
- Introduction to Starlark for creating custom rules
- Declaration/consumption of dependencies and platform constraints
- Modularization strategies and macro reuse
- Workshop: Structuring a multilingual project with Bazel

Build optimization

- Principles of incremental builds and dependency graphs
- Local cache vs. remote caching: when and how
- Remote execution and large-scale parallelization
- Performance settings (sandboxing, spawn strategies, RAM/CPU)
- Measurement and diagnostics: profiles, build event protocol
- Workshop: Setting up a remote cache and measuring gains

Test management with Bazel

- Unit, integration, and end-to-end testing with test rules
- Tightness, reproducibility, and flaky tests
- Key parameters: --test_output, --test_filter, --runs_per_test
- Integration with JUnit, pytest, Go test, etc.
- Reports, execution times, and sharding
- Workshop: Writing and running a multi-language test suite

Bazel and CI/CD

- Integrating Bazel into Jenkins, GitHub Actions, GitLab CI
- Pipeline strategies: incremental builds, branch builds, releases
- Artifact management, shared cache, and job isolation
- Optimizing cycle times and execution costs
- Reliability best practices (timeouts, retries, cache keys)
- Workshop: Creating a complete CI/CD pipeline with Bazel

Cross-platform development

- Builds for Linux, macOS, Windows
- Cross-compilation and toolchains (C/C++, Java, Go, Python, etc.)
- Native dependencies and dynamic/static links
- Integration with Docker and reproducible images
- Portability, compatibility, and multi-agent CI
- Workshop: Building and testing a project for multiple platforms

Extension and customization

- Rules_* ecosystem and extension selection
- Writing rules in Starlark: providers, actions, attrs
- Macros and build code factorization
- Internal publication of shared rules
- Dependency management, versions, and compatibility
- Workshop: Creating a custom Bazel rule

Security and governance

- Access policies, auditing, and build traceability
- Artifact validation and signing (SBOM, attestations)
- Open-source dependency and licensing management
- Attack surface analysis and hardening
- Compliance and regulatory requirements
- Workshop: Implementing security controls in the chain

Maintenance and best practices

- Structuring a monorepo for maintainability
- Naming conventions, BUILD file review
- Long-term performance monitoring and metrics
- Maintenance automation (linters, buildifier)
- Migration from other tools (Maven/Gradle/Make)
- Workshop: Optimizing an existing Bazel project

Closing and implementation plan

- Review of key learnings and accelerators
- Bazel industrialization checklist in CI/CD
- .bazelrc template and reusable patterns
- Adoption roadmap and change strategy
- Resources and monitoring (releases, LTS, roadmap)
- Workshop: Developing your Bazel adoption plan

Companies concerned

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.

Placement at the start of training

The positioning at the start of the training complies with Qualiopi quality criteria. Upon final registration, the learner receives a self-assessment questionnaire that allows us to assess their estimated level of proficiency in different types of technologies, as well as their expectations and personal objectives for 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 be problematic for the monitoring and smooth running of the training session.

Teaching methods

Practical training: 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.

Validation

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

Certification

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

Training organization registered under number 11 75 54743 75. This registration does not constitute state approval.
© Ambient IT 2015-2025. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg