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

Updated on 03/12/2026

Sign up

Harness Agent Training: Master the Code with Claude

3 days (21 hours)

Overview

Our Agent Harness training teaches you how to use Claude as a true development teammate: generating reliable code, performing rapid refactoring, and automating repetitive tasks. You'll leave with reusable patterns to accelerate your
build-test-review cycles on real-world scenarios.

The training aims to design and execute agent-based workflows with Harness: defining objectives, breaking down tasks into steps, managing context, and quality control (tests, lint, security). The approach emphasizes decision traceability and reducing hallucinations through guidelines, constraints, and systematic validations.

Each session alternates between demos and workshops: creating a "code reviewer" agent, a "migration" agent, and a "bug fixer" agent. Deliverables: structured prompts, Harness scripts, validation checklists, and a mini-project integrating generation, testing, and automated review.

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

Objectives

- Configure a Harness project and execute reproducible runs.
- Write robust instructions (roles, constraints, acceptance criteria).
- Orchestrate multi-step tasks with controlled context and memory.
- Automate refactorings, fixes, and migrations with validations.
- Implement safeguards: tests, lint checks, code reviews, and actionable logs.

Target Audience

- Back-end / front-end developers
- Tech leads and quality managers
- DevOps/platform engineers
- Technical consultants and trainers

Prerequisites

- Proficiency in a programming language (JavaScript, Python, Java, etc.)
- Experience with Git and review workflows (PR/MR)
- Basic understanding of automated testing and CI
- Comfortable with the command line

Technical requirements

- Computer with at least 8 GB of RAM (16 GB recommended)
- macOS, Linux, or Windows (WSL2 recommended)
- A code editor (VS Code, IntelliJ, etc.)
- Access to Claude and installation/configuration of Harness
- Node.js or Python depending on the workshops, plus a terminal (Bash/Zsh/PowerShell)

Our Harness Agent Training Program

[Day 1 - Morning]

Claude Code Fundamentals and Agent Architecture

- Introduction to Claude Code CLI: installation, authentication, and configuration
- Agent Harness Philosophy: Why Encapsulate AI? (Security, Context, Control)
- Understanding the Agentic Workflow reasoning loop (Thought -> Action -> Observation)
- Managing token budgets and Anthropic API limits
- Overview of Capabilities: File Editing, Command Execution, and Search
- Hands-on Workshop: Setting up the environment and first "Pair Programming" session with Claude Code on a local project.

[Day 1 - Afternoon]

Mastering Context and the MCP Protocol

- The Model Context Protocol (MCP): Connecting Claude to Your Ecosystem (GitHub, Slack, DB)
- Configuring existing MCP servers and creating a custom MCP server
- Context reduction strategies: avoiding prompt window overload
- Managing the .claudecode.toml configuration file and project rules
- Using structure-parsing tools (ls, cat, grep) by the agent

- Hands-on workshop: Create an MCP connector to enable Claude to read private technical documentation.

[Day 2 - Morning]

Building a Secure Agent Harness

- Isolation and execution: Using Docker or an E2B-type sandbox
- Setting up Guardrails: Filtering dangerous commands
- Human Validation: Configuring "Architect" vs. "Agent" modes
- Managing fine-grained permissions on the file system and processes
- Architecture of a Minimalist Harness: Inputs, Outputs, and Control Loops
- Hands-on workshop: Developing a Python "Harness" that intercepts Claude's commands before execution.

[Day 2 - Afternoon]

Automated Engineering: Refactoring and Testing

- Agent-driven Test-Driven Development (TDD) strategies
- Refactoring technical debt: AI-assisted complexity analysis
- Generating unit and integration tests (Playwright, Jest)
- Code migration (e.g., JS to TS) and dependency updates
- Using the "edit_file" tool for precise modifications without a complete rewrite
- Hands-on workshop: Migrating a legacy module with test coverage generated by Claude Code.

[Day 3 - Morning]

Industrialization and CI/CD Integration

- Integrating Claude Code into GitHub Actions or GitLab CI pipelines
- Automated PR Review: Configuring an agent to comment on Pull Requests
- Automatic generation of technical documentation (Swagger, README)
- Managing secrets and environment variables in an agent-based workflow
- Automated security audit by the agent during the build
- Hands-on workshop: Creating an CI bot that automatically fixes Lint errors.

[Day 3 - Afternoon]

Agent Observability and Monitoring

- Tracing AI Decisions: Logging Chains of Thought
- Agent performance evaluation: success rates and benchmarks
- Cost and ROI: Monitoring Anthropic API Usage in Production

- Toward Full Autonomy: Exploring Current Limits and Ethics
- Checklist for deploying an autonomous agent
- Hands-on Workshop: Final Project - Resolving a Complex GitHub Issue from Start to Finish (Analysis -> Code -> Testing -> PR).

Target Companies

This training is designed for both individuals and companies—large or small—looking to train their teams in new advanced IT technologies or to acquire specific industry knowledge or modern methodologies.

Entry-level assessment

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 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 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 instruction from the instructor—supported by examples and discussion 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.