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

Updated on 04/25/2025

Sign up

DevSecOps training - Putting security at the heart of DevOps

1 day (7 hours)

PRESENTATION

Our DevSecOps training course provides architects, developers and security experts with a shared vision of the issues at stake, enabling them to speak the same language and initiate a structured approach.

Our training program is aimed at development, infrastructure and security professionals wishing to integrate concrete DevSecOps practices into their projects.

Demonstrations and feedback will illustrate how a strong SecOps culture enables architects, developers and security teams to speak the same language and work together seamlessly.

At the end of this training course, you'll have a clear view of the possibilities offered by SecOps in your company.

OBJECTIVES

- Understanding the fundamental concepts of DevSecOps
- Detect and prevent vulnerabilities (code, dependencies, containers, secrets...)
- Implement a secure CI/CD chain
- Master key tools such as Bandit, Trivy, Checkov, KubeLinter, Syft and NeuVector

TARGET AUDIENCE

- Software architects
- Cybersecurity Team
- Lead developers

- Possibly Ops/System administrators interested in security aspects

Prerequisites

- Good understanding of DevOps and CI/CD practices
- Bases under development
- Basic knowledge of Docker and Infrastructure as Code (Terraform, Kubernetes)
- Unrestricted Internet access (VPN, proxy, etc.) to TP environments

DevSecOps training program

Introduction to DevSecOps

- What is DevSecOps?
- The evolution of DevSecOps
- The importance of safety in the SDLC
- The "Shift Left" approach
- The role of automation in DevSecOps
- The benefits of DevSecOps

Git, IaC, Immutable Infrastructure and GitOps

- General Git security considerations
- How immutable infrastructure improves security
- Eliminating configuration drifts
- GitOps: the future of infrastructure management

Secret management and prevention

- The growing problem of the proliferation of secrets
- Secret leakage levels
- Pre-ordered hooks
- Practical work: Preventing secrets leaks with TruffleHog

Secure dependencies

- The growing threat of attacks via dependencies
- Common vulnerabilities and exposures (CVEs)
- Vulnerability Scoring System (CVSS)
- List of common weaknesses (CWE)
- List of common platforms (CPE)
- TP: OWASP Dependency-Check

Code quality and safety

- The importance of code quality and security
- Understanding abstract syntax trees (ASTs)
- Common security pitfalls in code
- Practical work: Bandit - Python security linter

Container safety

- Dockerfile: security linting
- Securing Dockerfiles
- Common security pitfalls in Dockerfiles
- Practical work: Hadolint - a practical example of linting
- Dockerfile linting vs. Docker image scanning
- Practical work: Scanning Docker images with Trivy
- Understanding Trivy reports

Secure collaboration with IaC (Terraform)

- Secret management
- Role-based access control (RBAC) and the principle of least privilege
- Remote status storage
- Drift mitigation and compliance

IaC code analysis

- Common security pitfalls in IaC (Terraform)
- Terraform security with Checkov
- Common security problems in Kubernetes manifests
- Practical work: Static analysis with KubeLinter

Software Supply Chain and SBOM

- Importance of supply chain security
- Understanding SBOMs (Software Bill of Materials)
- The role of SBOMs in DevSecOps
- Practical work: Using Syft for analysis
- TP: Syft and OWASP DependencyTrack

Security Policy as Code (SPaC)

- SAST, DAST, and Shift Left approach
- Compliance and audit
- Web Application Firewall (WAF)
- Data loss prevention (DLP) sensors
- Response rules and incident management
- Practical work: NeuVector - a practical example of SPaC

DevSecOps pipeline

- DevSecOps as a system
- CI/CD as a backbone
- Building a DevSecOps pipeline (GitLab CI/CD)

Measuring and improving DevSecOps

- Feedback loops, coverage, metrics and security
- Learning from failure
- Shift Left, extend right
- Success indicators (KPIs)

Companies concerned

This training course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer 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 course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.