AMBIENT-IT

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

Updated on 10/26/2026

Sign Up

Trivy Training

2 days (14 hours)

Overview

Trivy is an application security tool that detects vulnerabilities, exposed secrets, and misconfigurations. With its scanners, client-server mode, and integration with GitHub Actions, it helps teams secure application images and dependencies.

Our Trivy training will help you master application security terminology and implement controls tailored to your day-to-day DevSecOps practices.

You'll learn how to interpret CVEs and severity levels, configure scanners, and analyze Docker Hub images using a Trivy server hosted on AWS in a secure environment.

By the end of the training, you'll be able to build GitHub Actions workflows to scan applications in public and private repositories, manage permissions, and apply blocking thresholds consistent with your security requirements.

This training also covers scan reports, prioritizing fixes, documented exceptions, and validating remediation. The workshops will help you connect detection, automation, and risk mitigation within an application delivery pipeline.

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

Objectives

- Master application security terminology and Trivy's main scanners

- Configure vulnerability, secret, and misconfiguration scans with Trivy in an application project
- Deploy a Trivy server on AWS and analyze images from Docker Hub
- Automate scans of applications from public and private repositories using GitHub Actions in workflows
- Use reports to prioritize fixes and verify remediation

Target Audience

- DevSecOps Engineer
- DevOps Engineer
- Application Developer
- Application Security Engineer
- Cloud Administrator

Prerequisites

- Proficiency with the Linux command line for routine operations
- Knowledge of Docker and the principles of image building
- Experience using Git and GitHub to manage a repository
- Understanding of CI/CD pipelines and reading YAML configuration files
- Basic understanding of AWS and the network rules associated with virtual machines

Technical Prerequisites

- Computer with at least 8 GB of RAM, 20 GB of free disk space, and a native Ubuntu 24.04 LTS environment or a virtual machine
- Docker Engine, Git, and Trivy must be installed in versions validated prior to the session, with installation permissions
- A GitHub account that allows the creation of public and private repositories and the execution of GitHub Actions for the workshops
- Access to an AWS lab environment with supervised permissions on Amazon EC2 and security groups, as well as a Docker Hub account for authenticated scans
- A stable internet connection allowing access to GitHub, Docker Hub, Trivy’s vulnerability databases, and the AWS training environment

Trivy Training Program

[Day 1 - Morning]

Understanding application security and getting started with Trivy

- Application security terminology: vulnerability, threat, risk, dependency, and software supply chain

- Interpreting CVEs, CVSS scores, and severity levels; distinguishing between severity and contextualized risk
- Trivy's scope: dependency analysis, secret detection, and configuration checks, distinction from dynamic testing and source code analysis
- Installing and verifying Trivy, vulnerability databases, cache, and explicit scanner selection
- Basic commands: `image`, `fs`, and `config`; reviewing detected packages and patched versions
- Hands-on workshop: Scan an image and a demo project, identify a vulnerability, a dummy secret, and a misconfiguration, then explain the results.

[Day 1 - Afternoon]

Scanning Docker Hub with a Trivy server on AWS

- Client-server architecture: client-side image analysis, server-side vulnerability scanning, and client-side secret and configuration checks
- Deploying the Trivy server on a lab Amazon EC2 instance, caching, and updating the databases
- Securing communications via an SSH tunnel, token-based authentication, and restricting access using AWS security groups
- Accessing Docker Hub images from the client installed on EC2, registry authentication, tags, digests, and download limits
- Running scans with --server, filtering by severity, and exporting results in JSON format
- Hands-on workshop: Deploy a Trivy server on AWS, connect a client to the same instance, and analyze two Docker Hub images to compare their vulnerabilities.

[Day 2 - Morning]

Automating scans with GitHub Actions

- Structure of a GitHub Actions workflow: push and pull_request events, jobs, runners, and scan steps
- Checking out the repository with `actions/checkout` and configuring `trivy-action` to analyze project dependencies, secrets, and configurations
- Scanning public and private repositories, minimal GITHUB_TOKEN permissions, and dedicated authentication for accessing another private repository
- Securing workflows: GitHub secrets, restrictions on external contributions, and actions pinned to verified SHAs
- Building and scanning an application image, cache management, and shared configuration in trivy.yaml
- Hands-on workshop: Create and run a scan workflow on a public repository, then on a private repository
  private, analyze the project and its image, then verify the permissions and the generated reports.

[Day 2 – Afternoon]

Leverage the results and ensure the reliability of the checks

- Define blocking thresholds based on severity and exit code; distinguish between detected vulnerabilities and runtime errors
- Publishing JSON reports as artifacts and SARIF results in GitHub Code Scanning when repository permissions and the plan allow it
- Alert qualification: exposure context, availability of a fix, false positives, and detection limitations
- Management of documented exceptions, targeted exclusions, validation leads, and deadlines for re-evaluation
- Remediation by updating dependencies and base images, removing secrets, correcting configurations, and running a new scan
- Hands-on workshop: Cause a workflow to fail on a high-priority vulnerability, retain the report despite the failure, apply a fix, and validate the new scan before cleaning up AWS resources.

Target Audience

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

Pre-training 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 level across 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 operation of the training session.

Teaching Methods

Hands-On Training: 60% Hands-On, 40% Theory. Training materials distributed in digital format to all participants.

Organization

The course alternates between theoretical instruction from the trainer—supported by examples and reflection sessions—and group work.

Assessment

At the end of the session, a multiple-choice quiz is administered to verify that participants have successfully acquired the skills.

Certification

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

Training organization registered under number 11 75 54743 75. This registration does not constitute state accreditation.
© Ambient IT 2015-2026. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg