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

Updated on 06/05/2026

Sign up

OpenBao Training

4 days (28 hours)

Overview

OpenBao is an open-source secret management solution for enterprises. This platform enables the storage, distribution, and control of access to secrets, certificates, and cryptographic keys used by modern applications.

Our OpenBao training will enable you to master the installation, configuration, and operation of a secret vault in a DevSecOps environment. You will be able to create policies, enable authentication methods, manage tokens, and secure application access.

You'll also learn to configure KV engines, dynamic secrets, automatic rotation, and integration with Kubernetes. OpenBao enables centralized secret governance while reducing risks associated with static credentials.

Furthermore, OpenBao integrates with cloud-native environments, CI/CD pipelines, and high-availability architectures.

Following this training, you will be able to install, administer, and monitor OpenBao, as well as understand its architecture.

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

Objectives

- Understand OpenBao's architecture and components
- Install, configure, and secure an OpenBao server
- Manage tokens, policies, and authentication methods

- Administer static and dynamic secret engines
- Integrate OpenBao with Kubernetes and CI/CD pipelines
- Deploying OpenBao with high availability and applying governance best practices

Target Audience

- System and cloud administrators
- DevOps, SRE, and platform engineers
- Security and DevSecOps engineers
- Technical and cloud-native architects
- Developers who use application secrets

Prerequisites

- Basic knowledge of Linux administration
- Basic understanding of application security and access management
- General understanding of cloud or DevOps architectures
- Basic Kubernetes knowledge is a plus for the integration modules

Technical prerequisites

- Have a computer running Linux, macOS, or Windows with WSL2
- Install a compatible terminal and code editor, such as Visual Studio Code
- Install Docker for demo environments and hands-on workshops
- Install kubectl and have a local or remote Kubernetes cluster for the Kubernetes modules
- Ensure a stable internet connection to download the necessary images, dependencies, and tools

OpenBao Training

[Day 1 - Morning]

Fundamentals of secret management

- Understand the role of OpenBao in a DevSecOps architecture
- Identify the challenges related to secrets, tokens, certificates, and application keys
- Discover OpenBao’s positioning as an open-source secret management solution
- Compare traditional storage practices with the use of a secret vault
- Understand the concepts of seal, unseal, root token, policy, and secret engine
- Hands-on workshop: Install OpenBao, initialize the server, and use the first CLI commands

[Day 1 - Afternoon]

Server architecture and configuration

- Understand the architecture of an OpenBao server and its main components
- Configure listeners, storage, TLS settings, and environment variables
- Examining how data-at-rest encryption works
- Explore storage backends: file, Raft, Consul, and compatible solutions
- Analyze logs, server status, and basic administration commands
- Set up a clean local configuration for the following exercises

KV engine and secret structuring

- Enable and configure the KV secrets engine
- Compare KV v1 and KV v2: versioning, rollback, soft delete, and destroy
- Organize secret paths by application, team, and environment
- Define naming conventions suitable for production
- Performing common operations: read, write, delete, and restore
- Hands-on workshop: Create a versioned secret tree for multiple environments

[Day 2 - Morning]

Policies, tokens, and the authorization model

- Understanding the policy-based authorization model
- Creating policies tailored to applications, teams, and environments
- Manage tokens, lifetimes, renewals, revocations, and permissions
- Applying the principle of least privilege to access secrets
- Testing and debugging permission errors with the CLI and API
- Hands-on workshop: writing precise policies and validating access rights with different tokens

[Day 2 - Afternoon]

Application authentication methods

- Discover the authentication methods available in OpenBao
- Configure userpass authentication for training and administration purposes
- Set up AppRole for automated applications and services
- Understand the concepts of RoleID, SecretID, TTL, and limited usage
- Securing application authentication in a CI/CD or server environment
- Hands-on workshop: Connecting a test application to OpenBao with AppRole.

Dynamic secrets and rotation

- Understanding the difference between static and dynamic secrets
- Explore use cases with databases, temporary accounts, and application access
- Configure TTL, lease, renewal, and revocation settings
- Implement a secret rotation strategy
- Identify security benefits: reduced exposure, traceability, and rapid revocation
- Hands-on workshop: Generate temporary credentials and test their expiration.

[Day 3 - Morning]

Kubernetes Integration

- Understand the limitations of native Kubernetes Secrets
- Configure Kubernetes authentication with service accounts
- Associate Kubernetes workloads with dedicated OpenBao policies
- Exploring integration patterns: init container, sidecar, injection, and CSI driver
- Securing access to secrets for cloud-native applications
- Hands-on workshop: authenticate a Kubernetes workload and retrieve a secret from OpenBao.

[Day 3 - Afternoon] Application

integration and APIs

- Use the CLI, HTTP API, and application clients
- Consuming secrets from Node.js, Python, Go, or Java applications
- Handling common errors: expired token, access denied, incorrect path, secret destroyed
- Implement application caching and secure renewal strategies
- Avoiding anti-patterns: secrets in logs, persistent tokens, overly broad permissions
- Structure a clean integration between OpenBao and business applications

CI/CD and DevSecOps automation

- Integrate OpenBao into a CI/CD pipeline
- Retrieve secrets securely in GitLab CI, GitHub Actions, or Jenkins
- Automate configuration using scripts, APIs, and Infrastructure as Code principles
- Manage development, testing, staging, and production environments

- Implement safeguards against secret exposure in Git repositories
- Hands-on workshop: Connect a CI/CD pipeline to OpenBao to retrieve a temporary secret.

[Day 4 - Morning]

High Availability and Operations

- Understanding the challenges of high availability for a secret vault
- Deploy OpenBao with built-in Raft storage
- Understanding node roles, quorum, the leader, and failover mechanisms
- Manage backup, restore, and secure restart operations
- Monitoring cluster status, logs, and key metrics
- Hands-on workshop: Deploy a simplified OpenBao cluster and simulate a node failure.

[Day 4 - Afternoon]

Security, auditing, and hardening

- Enable and use audit devices
- Analyze access, errors, tokens, and sensitive operations
- Apply hardening best practices: TLS, minimal policies, rotation, and role separation
- Set up alerts for abnormal behavior
- Define a strategy for periodically reviewing access and secrets
- Prepare a security checklist before going live

Governance, migration, and final case

- Define an organization-wide secret governance strategy
- Prepare a migration from Vault or an existing secret management solution
- Build an operations runbook: incident response, compromise response, emergency rotation, and recovery
- Define responsibilities among security, DevOps, SRE, and development teams
- Develop a roadmap for the gradual adoption of OpenBao in production
- Hands-on workshop: complete a comprehensive end-to-end case study combining policies, secrets, AppRole, audit, and remediation.

To learn more

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 domain knowledge or modern methodologies.

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.