AMBIENT-IT

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

Updated on 01/29/2026

Register

JBoss EAP Training

4 days (28 hours)

Overview

Red Hat JBoss EAP is a Jakarta EE-certified Java application server designed to host critical applications in cloud and DevOps environments.

This training will teach you how to install, configure, and administer JBoss in standalone mode, deploy applications via the CLI, automate tasks with Ansible, and integrate the server into a CI/CD pipeline.

Your team will learn how to use JBoss EAP effectively, manage deployments consistently, monitor performance, and secure production environments.

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 operation of JBoss EAP
- Install, configure, and administer a standalone server
- Deploy Java applications via CLI or console
- Automate installation and deployment with Ansible
- Integrate JBoss EAP into a basic CI/CD pipeline

Target audience

- DevOps
- Production engineers
- Back-end developers
- System administrators

Prerequisites

- Knowledge of Linux or Windows system administration
- Basic proficiency in a scripting language (Bash, YAML, etc.)
- General knowledge of Java or the JEE ecosystem appreciated
- No JBoss prerequisites required

Our JBoss EAP training program

Introduction to JBoss EAP and its ecosystem

- Origin and positioning of JBoss in the Red Hat ecosystem
- Role of the application server in an enterprise Java architecture
- Comparison with Tomcat, WebLogic, and Payara
- Typical use cases in a DevOps environment
- General architecture (Java EE/Jakarta EE, execution container)
- Supported formats: WAR, EAR, executable JAR

Local installation and server structure

- Downloading and installing JBoss EAP in standalone mode
- Overview of the directory tree (bin/, standalone/, modules/, deployments/)
- Standalone vs. domain modes: differences and uses
- Initial configuration (standalone.xml, logging, ports)
- Start/stop via CLI or shell script
- Workshop: Install JBoss EAP in standalone mode, access the console, explore the structure

Administration console and CLI interface

- Connecting to the web administration console
- Overview of the main menus (server, runtime, deployments, etc.)
- Basic syntax of jboss-cli.sh
- Add an administrator user with the add-user.sh script
- Query examples: read, modify, reload
- Reusable CLI scripts for automation

Application deployment

- Supported formats: WAR, EAR, JAR with servlet

- Hot deploy vs. CLI deploy
- Deployment management in standalone.xml
- Application context management and redeployment
- Dynamic activation/deactivation of applications
- Workshop: Deploying a WAR application via CLI and checking its availability

Configuring data sources and JDBC drivers

- Defining a data source in standalone.xml
- Deploying a JDBC driver (PostgreSQL, MySQL, etc.)
- Configuring JDBC connections and pools
- Testing database connectivity
- Monitoring and managing active connections
- Use in a Java EE application with JNDI

Logging, monitoring, and configuration files

- Configuring the logging subsystem (level, format, handlers)
- Main logs: server.log, boot.log
- Adding new application or technical loggers
- Monitoring memory consumption and GC
- Access to system metrics via CLI or HTTP
- Possible integration with Prometheus/Grafana

Automation with Ansible

- Presentation of the official redhat.eap collection
- Structure of an Ansible playbook: inventory, roles, variables
- Automated deployment of JBoss + Java app
- Using a role to configure a data source
- Workshop: Create a simple Ansible playbook to install JBoss and deploy an app with a data source

Best practices for server configuration

- Separation of environments (dev, test, prod)
- Outsourcing sensitive variables (.properties file, vault)
- Management of ports, logs, and dynamic paths
- Structuring custom modules
- Choice of profiles (full, ha, default) depending on the environment

Security and user management

- JBoss security configuration (.users.properties, .roles.properties)
- Secure console and CLI access
- JAAS, LDAP, local database authentication

- Definition of roles and access control by domain
- Best practices for hardening (ports, logs, credentials)

DevOps & CI/CD integration

- Maven integration for building and packaging applications
- Automatic deployment with a script
- Simple pipeline concept: build? deploy? restart
- Integration with Jenkins or GitLab CI
- Logs and deployment status via console or CLI

Backup, rollback, and version management

- Backup of the standalone/ directory and deployments
- Management of deployed versions via CLI or web UI
- Manual or automated rollback (scripts)
- Tips for dual-instance environments (blue/green)
- Secure deployment with connectivity verification

Post-training summary and review

- Summary of key commands
- JBoss production deployment checklist
- Recommendations for monitoring and maintenance
- Opening up to possible extensions: clustering, JBoss Operator, OpenShift
- Official resources, support, Red Hat documentation

Companies involved

This training is aimed at both individuals and companies, large or small, wishing to train their teams in new advanced IT technology or to acquire specific professional knowledge or modern methods.

Placement at the start of training

The placement test at the start of the training course complies with Qualiopi quality criteria. Once they have finalized their registration, learners receive 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 course, 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.