Practical DevOps Training: Docker, CI/CD and Application Deployment

Introduction to the practical DevOps training course

Presentation of the course objectives.

Understanding the difference between DevOps as a culture and DevOps as a technical practice.

Identifying the steps of a modern software delivery cycle.

Understanding common issues: manual configuration, inconsistent environments, risky deployments, and lack of visibility.

Review of DevOps principles

Collaboration between development, operations, and quality assurance.

Automation of the delivery cycle.

Fast feedback.

Continuous integration.

Continuous deployment.

Continuous improvement.

Preparing an application for delivery

Source code organization.

Configuration by environment.

Environment variables.

Dependency management.

Automated tests.

Reproducible builds.

Hands-on workshop:

Preparing an existing application for automation.

Identifying elements that need to be externalized.

Validating the local build.

Docker for application delivery

Review of images and containers.

Creating a Dockerfile.

Containerization best practices.

Multi-stage builds.

Ports, environment variables, and logs.

Hands-on workshop:

Creating a Docker image for an application.

Running the application in a container.

Changing configuration without modifying the code.

Complete environment with Docker Compose

Why use Docker Compose.

Application and database.

Internal networks.

Persistent volumes.

.env file.

Health checks.

Dependencies between services.

Hands-on workshop:

Creating a complete environment with Docker Compose.

Connecting the application to a database.

Validating the environment.

Diagnosing a configuration issue.

Introduction to CI/CD

Understanding continuous integration.

Understanding continuous delivery.

Difference between build, test, artifact, image, and deployment.

Pipeline structure.

Triggers: push, merge request, pull request, tags, and branches.

Creating a CI/CD pipeline

Presentation of a pipeline tool: GitLab CI/CD, GitHub Actions, or Azure DevOps.

Structure of a YAML file.

Pipeline stages.

Pipeline variables.

Automated build execution.

Automated test execution.

Hands-on workshop:

Creating a continuous integration pipeline.

Automatically compiling the application.

Running tests.

Making a pipeline step fail and fixing it.

CI/CD with Docker

Building a Docker image in the pipeline.

Tagging an image.

Publishing an image to a registry.

Managing variables and secrets.

Separating environments.

Hands-on workshop:

Adding a Docker build step to the pipeline.

Publishing an image to a registry.

Using a secret for authentication.

Application deployment

Difference between delivery and deployment.

Manual deployment and automated deployment.

Deployment to a test environment.

Version update.

Simple rollback.

Post-deployment validation.

Hands-on workshop:

Deploying an application with Docker Compose.

Updating the image used by the environment.

Validating the new version.

Rolling back to a previous version.

Configuration and secrets

Configuration by environment.

Environment variables.

Configuration files.

Secrets in pipelines.

Risks related to secrets in source code.

Best practices for managing sensitive information.

Hands-on workshop:

Replacing hardcoded configuration.

Using secure variables.

Fixing a poor practice related to secrets.

Logs, health checks, and diagnostics

Reading container logs.

Understanding application logs.

Adding or using a health endpoint.

Validating service status.

Diagnosing a startup error.

Hands-on workshop:

Reading application logs.

Identifying an error.

Fixing the configuration.

Validating service health.

Introduction to observability

Why monitor an application.

Logs, metrics, and traces.

Health checks.

Dashboards.

Introduction to tools such as Prometheus, Grafana, or equivalents.

Hands-on workshop:

Enabling basic metrics.

Reading the application status.

Interpreting a simple issue.

Basic DevOps security

Common risks in a CI/CD chain.

Exposed secrets.

Vulnerable images.

Outdated dependencies.

Excessive permissions.

Security best practices for Docker and pipelines.

Hands-on workshop:

Identifying a poor practice.

Analyzing an image or dependency.

Fixing an insecure configuration.

Project Example that will be developped

Application to containerize.

Docker Compose environment.

CI/CD pipeline.

Image build and publication.

Deployment.

Validation using logs and health checks.

Simple incident correction.