Microservices for developers

Duration 3 day(s) (MICROSERVICES-DEV-03)

Development, deployment and administration

Description

A microservice architecture has advantages, but requires a great maturity in the practices of development, deployment and maintenance in operational conditions. This training aims to cover the practices, techniques and technologies essential to take full advantage of a system cut into microservices.

Goals

- Sensitize developers to problems and solve them in a microservices architecture
- Help developers to create applications integrating easily and efficiently into a microservices architecture

Public

- Developer - Architect

Prerequisites

- Java programming

Structure

40% Theory, 60% Practice

Program

Introduction and issues

REST

- Principles
- Automated tests
- Versionning
- HATEOAS
- Documentation
- Implemented: Spring Boot, Spring MVC, Spring HATEOAS, Swagger, Spring REST Docs

Persistence polyglot

- Cutting a monolith ("bounded context")
- Advantages and disadvantages

Enforcement: Spring Data (JPA, ElasticSearch)

- Configuration of a microservices architecture
- Outsourcing and centralization of the configuration
- Environments and profiles
- Security
- Implementation: Spring Boot, Spring Cloud Configuration Server

Registration and discovery of services

- The need for a service registry
- Approaches (intelligent load balancer, service registry)
- Enforcement: Spring Cloud and Netflix Eureka

Routing

- Load balancing
- Pattern API Gateway
- Implementation: Load balancing with Netflix Ribbon and Spring Cloud, Gateway Netflix Zuul

Reliability of services

- Calls between services
- Problem of cascading failure
- Pattern circuit breaker
- Implementation: Netflix Hystrix

Monitoring

- Monitoring at the center of a microservice
- Metrics and health check
- Implemented: Spring Boot Actuator

Logging

- Centralization of the logs
- Solution course: ELK, Graylog
- Syslog
- Correlation identifier
- Enforcement: UDP logging in an application, implementing correlation ID with Spring Boot

Deploying a microservices architecture

- Continuous deployment of tens / hundreds of applications
- Deployment optimization techniques
- Introduction to Docker
- Implementation: deployment of a Spring Boot application a Linux service, packaging of a microservice and infrastructure (service registry, load balancer) via Docker
- Infrastructure solutions for microservices (Kubernetes, Mesos, Swarm)
- Cloud Solutions (CloudFoundry, Heroku)