



Microservices pour les développeurs

Durée 3 jour(s) (MICROSERVICES-DEV-03)

Développement, déploiement et administration

Description

Une architecture microservices présente des avantages, mais nécessite une grande maturité dans les pratiques de développement, de déploiement et de maintien en conditions opérationnels. Cette formation vise à couvrir les pratiques, techniques et technologies essentielles pour tirer pleinement bénéfice d'un système découpé en microservices.

Objectifs

- Sensibiliser les développeurs aux problématiques et à les résoudre dans une architecture microservices
- Aider les développeurs à créer des applications s'intégrant facilement et efficacement dans une architecture microservices

Public

- Développeur
- Architecte

Prérequis

- Programmation Java

Répartition

40% Théorie, 60% Pratique

Programme

Introduction et enjeux

REST

- Principes
- Tests automatisés
- Versionning
- HATEOAS
- Documentation
- Mise en application : Spring Boot, Spring MVC, Spring HATEOAS, Swagger, Spring REST Docs

Persistance polyglotte

- Découpage d'un monolithe ("bounded context")
- Avantages et inconvénients

Mise en application : Spring Data (JPA, ElasticSearch)

- Configuration d'une architecture microservices
- Externalisation et centralisation de la configuration
- Environnements et profils
- Sécurité
- Mise en application : Spring Boot, Spring Cloud Configuration Server

Enregistrement et découverte des services

- Le besoin d'un registre de services
- Approches (load balancer intelligent, registre de services)
- Mise en application : Spring Cloud et Netflix Eureka

Routage

- Load balancing
- Pattern API Gateway
- Mise en application : Load balancing avec Netflix Ribbon et Spring Cloud, Gateway Netflix Zuul

Fiabilisation des services

- Les appels entre services
- Problème du cascading failure
- Pattern circuit breaker
- Mise en application : Netflix Hystrix

Monitoring

- Le monitoring au centre d'un microservice
- Métriques et health check
- Mise en application : Actuator Spring Boot

Journalisation

- Centralisation des logs
- Parcours de solutions : ELK, Graylog
- Syslog
- Identifiant de corrélation
- Mise en application : journalisation via UDP dans une application, implémentation d'un ID de corrélation avec Spring Boot

Déploiement d'une architecture microservices

- Déploiement continu de dizaines/centaines d'applications
- Techniques d'optimisation de déploiement
- Introduction à Docker
- Mise en application : déploiement d'une application Spring Boot en tant que service Linux, packaging d'un microservice et de l'infrastructure (registre de services, load balancer) v Docker
- Solutions d'infrastructure pour les microservices (Kubernetes, Mesos, Swarm)
- Solutions Cloud (CloudFoundry, Heroku)