

## JAVA

# Réaliser une architecture microservices avec Java

4 jours (28h00) | JAVA012 | Num form : form-319 | Perfectionnement / Avancé

INFORMATIQUE / DÉVELOPPEMENT / JAVA

## À l'issue de ce stage vous serez capable de :

- Définir une architecture basée sur les microservices
- Développer et consommer des services REST
- Mettre en place des microservices à l'aide d'un framework Java
- Déployer des microservices dans des conteneurs Docker
- Administrer et surveiller des microservices

## Niveau requis :

Avoir de bonnes connaissances du langage Java, connaissances de base de l'administration Linux.

## Public concerné :

Développeurs,

architectes.

Administrateurs Systèmes

## Programme :

### Introduction

- Regard sur l'évolution logicielle et organisationnelle.
- Avantages et inconvénients des applications monolithiques.
- Méthodologies Agiles et pratiques DevOps.
- Processus d'intégration continue et de déploiement continu.

## Les architectures microservices

- Définition d'une architecture microservices.
- Caractéristiques des microservices.
- L'émergence des architectures microservices.
- Les principaux acteurs.
- 

## Communications dans une architecture microservices

- Choix du style de collaboration : REST request/reply versus Publish-Subscribe Messaging.
- Principes de base et rappels HTTP.
- Mettre en œuvre des services REST avec Java.
- Le principe HATEOAS.
- Documenter un service REST avec Swagger.
- Tester un service REST.

## Développer un microservice en Java

- Présentation de frameworks Java "tout en un".
- Comparaison entre SpringBoot et Dropwizard.
- Développer une application avec Dropwizard ou SpringBoot.
- Packager une application production-ready.

## Exécuter un microservice à l'aide de conteneurs

- Présentation de Docker.
- Fonctionnement des conteneurs avec Docker.
- Concevoir des images personnalisées.
- Configurer et démarrer des conteneurs.

## Déploiement d'une architecture microservices

- Techniques d'optimisation de déploiement.
- Mettre en œuvre une application multi-conteneurs avec Docker Compose.
- Solutions d'infrastructure pour les microservices (Kubernetes, Mesos, Swarm).
- Solutions Cloud (Cloud Foundry, Heroku).

## Administration et Monitoring

- Centralisation des logs.
- Le monitoring de microservices.
- Métriques et Health Check.