Smartfuture
Certifications | Training | Technology solutions

Formation

Développer des microservices

Formation pour concevoir, développer et déployer des applications en microservices, incluant le découpage de systèmes monolithiques, la création d'API, la gestion des données distribuées, la sécurité, la résilience, l'observabilité et l'industrialisation.

Formation 100% pratique | Exercices & cas d'usage | Formateurs expérimenté

4 jours 28 h	**Format** Ateliers & labs guidés
Modalité Présentiel ou en ligne	**Niveau** Intermédiaire à avancé

INSCRIPTION / RÉSERVATION

Je m'inscris maintenant

OBJECTIFS PÉDAGOGIQUES

- Comprendre les microservices et leur différence avec les monolithes.
- Découper un domaine en microservices cohérents et faiblement couplés.
- Concevoir des API REST sécurisées et versionnées.
- Gérer la communication entre services et les données distribuées.
- Découvrir le déploiement conteneurisé et l'orchestration (Docker, Kubernetes).
- Assurer résilience, supervision, observabilité et intégration continue.

PUBLIC CIBLE

- Développeurs visant la conception et le développement de microservices.
- Architectes et responsables techniques souhaitant moderniser leur SI.
- Leads techniques, DevOps et ingénieurs backend sur des projets distribués.
- Équipes en migration d'un monolithe vers une architecture microservices.

PRÉREQUIS

- Maîtrise d'un langage backend (Java, .NET, Python, Node.js, etc.).
- Connaissance des API REST, HTTP et des bases de données relationnelles/noSQL.
- Notions de conteneurs (Docker) et/ou pipelines CI/CD appréciées.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Introduction aux architectures microservices

Du monolithe aux microservices

- Rappels sur les architectures monolithiques : avantages et limites.
- Évolution vers les SOA puis les microservices.
- Définition d'un microservice : caractéristiques, granularité, autonomie.
- Bénéfices et défis : scalabilité, résilience, complexité, culture DevOps.

Principes clés d'une architecture microservices

- Indépendance de déploiement et de cycle de vie.
- Responsabilité unique, faible couplage et forte cohésion.
- Conformité aux principes Twelve-Factor App.
- Organisation des équipes autour des produits et des domaines métiers.

Modélisation du domaine & découpage en microservices

Domain-Driven Design (DDD) – notions essentielles

- Ubiquitous language, Entities, Value Objects, Aggregates.
- Bounded Contexts et Context Maps.
- Identifier les frontières naturelles pour les microservices.

Stratégies de découpage

- Découpage par fonctionnalités métier, par flux, par capacités.
- Anti-patterns à éviter : découpage par couches techniques, microservices trop petits.
- Atelier pratique : exemple de découpage d'un système monolithique en microservices.

Communication entre microservices

Communication synchrone

- API REST, conventions HTTP, statuts, verbes, idempotence.
- Conception d'API : ressources, DTO, formats de données (JSON, XML).
- Versioning, compatibilité, contrats d'API.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Communication asynchrone & évènements

- Échanges asynchrones : queues, topics, publish/subscribe.
- Introduction aux brokers de messages (RabbitMQ, Kafka, etc.).
- Architecture orientée événements, Event Sourcing (aperçu).

API Gateway & gestion des appels

- Rôle d'une API Gateway : routage, agrégation, transformation.
- Cross-cutting concerns : authentification, throttling, observabilité, caching.

Gestion des données dans une architecture microservices

Base de données par microservice

- Principe & avantages : indépendance, scalabilité.
- Compatibilité avec différents types de bases (SQL, NoSQL).

Cohérence et transactions distribuées

- Limiter les transactions distribuées, principes CAP.
- Patterns de cohérence : Saga, orchestration vs chorégraphie.
- View personnalisées, CQRS (Command Query Responsibility Segregation – aperçu).

Migration des données

- Stratégies de migration progressive d'un monolithe vers des microservices.
- Strangler pattern (pattern d'étranglement).

Sécurité des microservices

Authentification & autorisation

- Rappels sur les concepts d'authentification / autorisation.
- Tokens, JWT, OAuth2/OpenID Connect (aperçu).
- Gestion des rôles, scopes, permissions.

Sécuriser les API & les échanges

- Communication sécurisée (HTTPS, certificats).
- Protection contre les attaques courantes (injection, XSS, CSRF – côté API).
- Meilleures pratiques de gestion des secrets et des clés (vault, variables d'environnement).

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Conteneurisation et déploiement des microservices

Introduction à Docker

- Concepts de base : image, container, registry.
- Dockerfile : construction d'une image de microservice.
- Docker Compose : démarrer plusieurs services pour un environnement de développement.

Orchestration (aperçu de Kubernetes ou équivalent)

- Concepts : pods, services, déploiements, scaling.
- Stratégies de déploiement (rolling, blue/green, canary – notions).

Observabilité, supervision & résilience

Logs, métriques, traces

- Centralisation des logs, corrélation des requêtes entre microservices.
- Métriques techniques et métiers : disponibilité, latence, taux d'erreurs.
- Notions de tracing distribué.

Résilience et tolérance aux pannes

- Timeouts, retries, backoff, bulkheads.
- Pattern Circuit Breaker, fallback, limitation de charge.
- Tests de résilience et chaos engineering (notions).

Tests et qualité dans un environnement microservices

Stratégies de tests

- Tests unitaires, tests d'intégration, tests end-to-end.
- Tests de contrat entre consommateurs et producteurs d'API.
- Tests de performance et de charge (notions).

CI/CD pour microservices

- Pipelines d'intégration continue : build, tests, analyse de code.
- Déploiement continu ou fréquent de services indépendants.
- Feature flags, toggles et rollback.

Atelier pratique : de la théorie à la mise en oeuvre

Projet fil rouge

- Étude de cas : à partir d'un monolithe simplifié, conception d'une architecture microservices.
- Découpage en services, définition des API, choix des modèles de données.
- Implémentation de quelques microservices représentatifs (API REST, persistance, communication).
- Mise en conteneur des services et exécution via Docker Compose.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Clôture et bonnes pratiques

- Checklist des bonnes pratiques microservices.
- Erreurs fréquentes à éviter et retours d'expérience.
- Plan d’action pour démarrer ou poursuivre un projet microservices dans votre organisation.