

Red Hat Cloud-native Microservices Development with Quarkus

Durée: 90 Jours **Réf de cours: DO378** **Méthodes d'apprentissage: E-learning**

Résumé:

Apprenez à développer des applications basées sur des microservices avec Quarkus et OpenShift

Aujourd'hui, les entreprises se tournent vers des architectures de microservices cloud-native. Quarkus est une nouvelle technologie intéressante aussi fiable, familière et mature que Java Enterprise, qui se déploie extrêmement vite dans des conteneurs. Le cours Développement de microservices cloud-native Red Hat avec Quarkus (DO378) met l'accent sur l'apprentissage des principes architecturaux et la mise en œuvre de microservices basés sur la version Red Hat de Quarkus et Red Hat OpenShift. Vous exploiterez les bases du développement d'applications pour réfléchir aux méthodes à appliquer pour développer, surveiller, tester et déployer des applications de microservices modernes.

Ce cours repose sur la version 4.14 d'OpenShift et la version Red Hat de Quarkus 3.8.

Remarque : ce module est proposé en autonomie pendant 90 jours.

Mis à jour 28 05 2024

Public visé:

Ce cours s'adresse aux développeurs d'applications Java.

Objectifs pédagogiques:

- A l'issue de cette formation, les participants auront étudié les points suivants :
- Déploiement d'applications de microservices sur la plateforme Red Hat® OpenShift Container Platform
- Création d'une application de microservices avec Quarkus
- Mise en œuvre des tests unitaires et d'intégration pour les microservices
- Utilisation de la spécification config pour injecter des données dans un microservice
- Sécurisation de microservices avec OAuth
- Mise en œuvre des contrôles d'intégrité, ainsi que du suivi et de la surveillance des microservices
- Création d'applications réactives et asynchrones à l'aide de Quarkus

Pré-requis:

- Posséder une expérience du développement d'applications Java ou avoir suivi le cours Développement d'applications Red Hat I : programmation en Java EE (AD183)
- Maîtriser un IDE, tel que Visual Studio Code
- Posséder une expérience de Maven et du contrôle des versions (recommandé, mais pas obligatoire)
- Posséder une expérience d'OpenShift ou avoir suivi le cours Introduction aux applications OpenShift (DO101) (recommandé, mais pas obligatoire)
- Passer l'évaluation gratuite Skills Assessment pour s'assurer que l'offre est adaptée aux compétences

Test et certification

■

Après cette formation, nous vous conseillons le(s) module(s) suivant(s):

- Développement avec Red Hat OpenShift I : introduction aux conteneurs avec Podman (DO188)
- Développement avec Red Hat OpenShift II : création d'applications Kubernetes (DO288)
- Création de microservices résilients avec Istio et Red Hat OpenShift Service Mesh (DO328)
- Pipelines et processus DevOps Red Hat : CI/CD avec Jenkins, Git et développement par les tests (DO400)

Contenu:

Présentation de la version Red Hat de Quarkus : Décrire les composants et les modèles des architectures d'applications basées sur des microservices, ainsi que les fonctions de la version Red Hat de Quarkus

Développement de microservices cloud-native avec Quarkus : Mettre en œuvre des applications basées sur des microservices à l'aide de la version Red Hat de l'environnement d'exécution Quarkus et des outils de développement associés

Tests de microservices Quarkus : Mettre en œuvre des tests unitaires et d'intégration pour les microservices

Développement de microservices réactifs et asynchrones : Décrire les fonctions des architectures réactives et mettre en œuvre des services réactifs à l'aide de Quarkus

Sécurisation de microservices Quarkus : Sécuriser les communications entre les microservices en appliquant la validation de l'origine, l'authentification des demandes et l'autorisation

Mise en œuvre de microservices Quarkus sur Red Hat OpenShift Container Platform : Développer et déployer des applications cloud-native sur Red Hat OpenShift Container Platform

Mise en œuvre de la tolérance aux pannes dans des microservices : Mettre en œuvre un système de tolérance aux pannes dans une architecture de microservices

Surveillance de microservices Quarkus : Surveiller le fonctionnement d'un microservice à l'aide de journaux, d'indicateurs de mesure et du traçage distribué