Red Hat Cloud-native Microservices Development with Quarkus

Durée: 5 Jours **Réf de cours: DO378** **Méthodes d'apprentissage: Virtual Learning**

Résumé:

Develop microservice-based applications with Quarkus and OpenShift.
Many enterprises are looking for a way to take advantage of cloud-native architectures, but many do not know the best approach. Quarkus is an exciting new technology that brings the reliability, familiarity, and maturity of Java Enterprise with a container-ready lightning fast deployment time. Red Hat Cloud-native Microservices Development with Quarkus (DO378) emphasizes learning architectural principles and implementing microservices based on Quarkus and OpenShift. You will build on application development fundamentals and focus on how to develop, monitor, test, and deploy modern microservices applications..
This course is based on OpenShift 4.5 and Quarkus 1.7
Course content summary
- Deploy microservice applications on Red Hat® OpenShift Container Platform.
- Build a microservice application with Quarkus.
- Implement unit and integration tests for microservices.
- Use the config specification to inject data into a microservice.
- Secure a microservice using OAuth.
- Build and deploy native Quarkus applications.

Public visé:

This course is designed for application developers.

Objectifs pédagogiques:

- Impact on the organization
- - Many organizations are striving to make the move from monolithic applications to applications based on microservices, as well as how to reorganize their development paradigm to reap the benefits of microservice development in a DevOps economy. With Quarkus, developers can more quickly build, test, and deploy their applications, improving application time to market.
- - Organizations are also invested in the familiarity of Java™ programming frameworks as well as the stability and benefits Red Hat OpenShift Container Platform. This course teaches developers how to leverage microservice application development with Quarkus for streamlined deployment on OpenShift clusters.
- Impact on the individual
- - As a result of attending this course, you will understand how to develop, monitor, test, and deploy microservice-based applications using Quarkus and Red Hat OpenShift.
- - Design a microservices-based architecture for an enterprise application.
- - Quickly build and test microservices with Quarkus and deploy on to OpenShift Container Platform.
- - Implement fault tolerance and health checks for microservices.
- - Secure microservices to prevent unauthorized access.

Pré-requis:

- Experience with application development or Red Hat Application Development I: Programming in Java EE (JB183)

- Be proficient in using an IDE such as Red Hat® Developer Studio or VSCode

- Recommended, but not required: experience with Maven and version control.

- Recommended, but not required: experience with OpenShift or Introduction to OpenShift Applications (DO101)

- Take our free assessment to gauge whether this offering is the best fit for your skills.

Technology considerations: Internet access required

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

Introduction to Containers, Kubernetes, and Red Hat OpenShift (DO180)

Red Hat OpenShift Development I: Containerizing Applications (DO288)

Building Resilient Microservices with Red Hat OpenShift Service Mesh (DO328)

Contenu:

Describe microservice architectures

Describe components and patterns of microservice-based application architectures.

Implement a microservice with Quarkus

Describe the specifications in Quarkus, implement a microservice with some of the specifications, and deploy it to an OpenShift cluster.

Build microservice applications with Quarkus

Build a persistent and configurable distributed quarkus microservices application.

Implement fault tolerance

Implement fault tolerance in a microservice architecture.

Build and deploy native Quarkus applications

Describe Quarkus in native mode and describe its deployment on OpenShift Container Platform.

Test microservices

Implement unit and integration tests for microservices.

Create application health checks

Create a health check for a microservice.

Secure microservices

Secure microservice endpoints and communication.

Monitor microservices

Monitor the operation of a microservice using metrics and distributed tracing.

Autres moyens pédagogiques et de suivi:

• Compétence du formateur : Les experts qui animent la formation sont des spécialistes des matières abordées et ont au minimum cinq ans d'expérience d'animation. Nos équipes ont validé à la fois leurs connaissances techniques (certifications le cas échéant) ainsi que leur compétence pédagogique.
• Suivi d'exécution : Une feuille d'émargement par demi-journée de présence est signée par tous les participants et le formateur.
• Modalités d'évaluation : le participant est invité à s'auto-évaluer par rapport aux objectifs énoncés.
• Chaque participant, à l'issue de la formation, répond à un questionnaire de satisfaction qui est ensuite étudié par nos équipes pédagogiques en vue de maintenir et d'améliorer la qualité de nos prestations.

Délais d'inscription :

•Vous pouvez vous inscrire sur l'une de nos sessions planifiées en inter-entreprises jusqu'à 5 jours ouvrés avant le début de la formation sous réserve de disponibilité de places et de labs le cas échéant.
•Votre place sera confirmée à la réception d'un devis ou ""booking form"" signé. Vous recevrez ensuite la convocation et les modalités d'accès en présentiel ou distanciel.
•Attention, si vous utilisez votre Compte Personnel de Formation pour financer votre inscription, vous devrez respecter un délai minimum et non négociable fixé à 11 jours ouvrés.