Red Hat Cloud-native Microservices Development with Quarkus (DO378) Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Overview

Build and deploy modern microservices using Quarkus and Red Hat OpenShift.

The {*training} teaches developers how to design, develop, test, and deploy**microservice-based applications** using the **Red Hat Build of Quarkus** and **Red Hat OpenShift Container Platform**. This course helps you transition from monolithic application development to a**cloud-native microservices architecture**, enabling faster deployment, scalability, and resilience.

Through hands-on labs, participants learn how to implement essential microservices principles such as **reactive programming, fault tolerance, security, and health monitoring**, using real-world examples built on Quarkus.

> *This course is based on Red Hat OpenShift 4.14 and Red Hat Build of Quarkus 3.8.*

Who Should Attend

This course is intended for:

- **Java Developers** building enterprise microservices.
- **Software Architects** designing modern distributed systems.
- **DevOps Engineers** integrating Quarkus-based microservices into OpenShift pipelines.

Red Hat Cloud-native Microservices Development with Quarkus (DO378) Training

Duration: **5 Day**

BILGINÇ IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

What You Will Learn

By the end of this {*training}, you will have gained knowledge and skills in the following areas:

- Deploy microservices on **Red Hat OpenShift Container Platform**.
- Build **Quarkus-based applications** optimized for containers.
- Write **unit and integration tests** for microservices.
- Inject configuration data using the **MicroProfile Config specification**.
- Secure microservices using **OAuth2 and authentication mechanisms**.
- Implement **health checks, tracing, and observability** for production readiness.
- Build **reactive and asynchronous services** for better scalability.
- Apply **fault tolerance and recovery mechanisms** in distributed systems.

Impact on the Organization

Organizations are rapidly shifting from **monolithic architectures to microservices** to achieve agility and scalability.
By adopting **Quarkus** with **Red Hat OpenShift**, teams can:

- Build and deploy applications faster, improving time-to-market.
- Use familiar Java frameworks while embracing modern cloud-native principles.
- Ensure consistent performance and reliability across environments.

This course helps organizations modernize their development workflows and optimize microservice delivery pipelines for DevOps environments.

Impact on the Individual

By attending this course, you will gain the skills to:

- Design and build microservices architectures for enterprise systems.
- Rapidly develop and test applications using **Quarkus**.
- Deploy and scale services on **Red Hat OpenShift**.
- Implement **fault tolerance, monitoring, and health checks**.
- Secure and manage microservices in production environments.

You'll be able to confidently build efficient, reactive, and secure microservices for modern hybrid-cloud platforms.

Red Hat Cloud-native Microservices Development with Quarkus (DO378) Training

Duration: **5 Day**

BILGINÇ IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Outline

1. Introduction to the Red Hat Build of Quarkus
Understand the microservices architectural patterns and explore the key features of Red Hat Build of Quarkus.

2. Developing Cloud-native Microservices with Quarkus
Build, package, and deploy microservices using Quarkus development tools and runtime.

3. Testing Quarkus Microservices
Implement unit and integration tests to validate microservice functionality.

4. Building Reactive and Asynchronous Applications
Develop non-blocking services using reactive programming models and Quarkus event-driven APIs.

5. Securing Quarkus Microservices
Protect microservices through OAuth2 authentication, authorization, and request validation.

6. Deploying Quarkus Applications on OpenShift
Develop, containerize, and deploy Quarkus applications seamlessly to OpenShift clusters.

7. Implementing Fault Tolerance in Microservices
Add resilience to services using retry patterns, circuit breakers, and fallback mechanisms.

8. Monitoring Quarkus Microservices
Monitor performance and availability with logging, metrics, and distributed tracing tools.