Publication date : 05/10/2024

Course : Quarkus, developing Java microservices in the cloud

Practical course - 3d - 21h00 - Ref. QRK
Price : 2150 CHF E.T.

4,4 / 5

Developed by Red Hat, Quarkus is a Java framework designed for Java virtual machines (JVMs) and native compilation. You'll learn how to develop Java applications broken down into microservices, and create images of them for containers that you'll deploy via a cloud orchestrator like Kubernetes.

Teaching objectives

At the end of the training, the participant will be able to:

- Designing a microservices-based architecture
- Setting up message-oriented communication between two microservices
- Implement fault tolerance systems
- Securing microservices to prevent unauthorized access
- Testing and deploying a microservices application
- Supervising an application in production

Intended audience

Java developers, Java/Java EE project managers.

Prerequisites

Good knowledge of Java/Java EE.

Course schedule

1 Presentation

- A new Java framework?
- The comparison with Spring Boot.
- What about Java EE and Jakarta EE?
- Microservices architectures.
- The MicroProfile standard

PARTICIPANTS

Java developers, Java/Java EE project managers.

PREREQUISITES

Good knowledge of Java/Java EE.

TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.
Participants also complete a placement test before and after the course to measure the skills they've developed.

TEACHING AIDS AND TECHNICAL RESOURCES

• The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
• At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
• A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

2 Getting started with Quarkus

- Project creation "Hello world".
- Tools for developers (Dev Services).
- Development, debugging and build processes.
- The Quarkus testing framework.
- The Docker environment.
- An overview of Quarkus extensions.

Hands-on work
Getting to grips with the environment and developing a first microservice, experimenting with live coding, debugging and continuous testing.

3 HTTP/RESTful communication and GraphQL

- REST principles.
- Using Jakarta RESTful.
- GraphQL contributions.
- Endpoint documentation (Open API).
- Writing an HTTP client.
- Security and authentication.
- Implementation of "Long Running Actions".

Hands-on work
Creation of RESTful and GraphQL services, querying from a second Quarkus microservice.

4 Fault tolerance

- The principle of resilience.
- Why is this important?
- Defensive programming.
- Circuit Breaker, Bulkhead.
- Other patterns.

Hands-on work
Application of the 6 fault tolerance patterns of the MicroProfile standard.

5 Message-oriented communication

- Reactive programming.
- Benefits and main difficulties.
- How to manage transactions
- The Saga pattern.
- Asynchronous communication (ActiveMQ, Kafka...).

Hands-on work
Implementation of message-oriented communication with Kafka between two microservices.

TERMS AND DEADLINES
Registration must be completed 24 hours before the start of the training.

ACCESSIBILITY FOR PEOPLE WITH DISABILITIES
Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

6 Production start-up

- Configuration mechanism designed for containers.
- Image construction strategy.
- HotSpot and GraalVM.
- AOT compilation, a radical change.
- Deployment on Kubernetes.

Hands-on work

Building Open Container Initiative (OCI) Docker images for deployment on a cloud orchestrator. Docker images: classic (OpenJDK) and Ahead Of Time (AOT, GraalVM).

7 Production supervision

- Definition of Health Check.
- Scattered log management.
- OpenTelemetry the new standard.
- System and custom metrics.

Hands-on work

Define a customized Endpoint Health check, collect and display telemetry data in Prometheus.

Dates and locations

REMOTE CLASS
2026 : 22 June, 2 Dec.