

Microservices, Node.js, and Kafka

Advanced 5-day course outline

FA-0320 · 5-day

Software Development

Introduction

This comprehensive 5-day training program that delves into the intricacies of microservices, Node.js, and Kafka. This immersive course is designed to equip attendees with a nuanced understanding of developing and managing scalable applications using microservices architecture, with Node.js as the building block and Kafka for handling real-time data feeds. Participants will journey through the conceptual groundwork to the advanced application of these technologies, engaging with real-world scenarios and case studies.

Learning outcomes

By the end of this training, participants will:

- Understand the core concepts and principles of microservices architecture.
- Gain proficiency in developing server-side applications using Node.js.
- Acquire the skill to construct and deconstruct real-time, data-driven applications using Kafka.
- Learn the methodologies to integrate microservices, Node.js, and Kafka.
- Develop the competence to troubleshoot and solve common and complex problems that might arise within the technological triad of microservices, Node.js, and Kafka.
- Grasp best practices for monitoring and scaling applications built on this stack.
- Comprehend the security implications and preventive measures in microservices architecture.

Prerequisites

- Basic understanding of software development processes.
- Familiarity with JavaScript programming (ES6 knowledge is beneficial).

- Conceptual knowledge of web servers and RESTful APIs.
- Basic understanding of databases and message brokers (advantageous but not mandatory).
- It is recommended to have a general idea about server-side programming and distributed systems.

Training outline

1. 1. Introduction to Microservices:

- Defining microservices: Understanding the architecture.
- The transition: Monoliths to microservices.
- Advantages & challenges in implementing microservices.
- Key design principles of microservice architecture.
- Communication between microservices: protocols and strategies.
- Hands-on activity: Identifying microservices in a case study.

2. 2. Deep Dive into Node.js:

- Overview of Node.js: Non-blocking I/O, event-driven architecture.
- Setting up a Node.js environment: Tools and best practices.
- Core modules and NPM: Package management and modular coding in Node.js.
- Building a basic RESTful API with Express.js.
- Understanding asynchronous programming (callbacks, promises, and async/await).
- Hands-on activity: Developing a simple server-side application.

3. 3. Integrating Microservices with Node.js:

- Node.js in a microservices architecture: Why it's a fit.
- Creating microservices in Node.js.
- Inter-service communication: Effective strategies and tools.
- Implementing API Gateways with Node.js.
- Service discovery mechanisms in a Node.js environment.
- Hands-on activity: Building a microservices-based application using Node.js.

4. 4. Introduction to Kafka:

- Understanding Kafka: Basics and architecture.
- Kafka's role in handling real-time data streams.

- Setting up Kafka: From installation to configuration.
- Kafka Producers and Consumers: Deep dive and application.
- Understanding Kafka Streams and Kafka Connect.
- Hands-on activity: Setting up a basic Kafka cluster.

5. 5. Kafka and Microservices: Real-time Communication:

- The synergy between Kafka and microservices for real-time applications.
- Implementing event-driven microservices with Kafka.
- Configuring Kafka for data consistency and recovery: Replication and fault tolerance.
- Managing data streams with Kafka in a microservice ecosystem.
- Case study: Real-world implementation of Kafka with microservices.
- Hands-on activity: Integrating Kafka with a microservices application.

6. 6. Building Robust Applications: Error Handling and Testing:

- Strategies for effective error handling in a distributed environment.
- Implementing testing in microservices: Unit, integration, and end-to-end testing.
- Testing Node.js applications: Tools and libraries.
- Testing strategies in Kafka: Ensuring message integrity and delivery.
- Hands-on activity: Writing test cases for our ongoing project.

7. 7. Securing Your Applications:

- Security challenges in microservices architecture.
- Implementing authentication and authorization in microservices with Node.js.
- Securing Node.js applications: Best practices and common vulnerabilities.
- Kafka security mechanisms: Authentication, authorization, and encryption.
- Hands-on activity: Securing the developed microservices application.

8. 8. Deployment and Monitoring:

- Deployment strategies for microservices.
- Containerization with Docker and orchestration with Kubernetes.
- CI/CD in microservices, Node.js, and Kafka environment.
- Monitoring microservices: Tools, practices, and key metrics.

- Performance monitoring in Node.js and Kafka.
- Hands-on activity: Deploying the final application using Docker and Kubernetes; setting up a basic monitoring system.

9. 9. Best Practices and Patterns:

- Best practices in designing microservices.
- Effective patterns for microservices development.
- Optimizing Node.js performance: memory management, clustering, and more.
- Kafka optimization: Configurations for enhanced performance.
- Hands-on activity: Refactoring code — a step towards optimization.

10. 10. Course Closure:

- Recap of the learning journey: From understanding to practical implementation.
- Open discussion: Overcoming potential challenges and addressing any unresolved issues.
- Future trends: Microservices, Node.js, and Kafka in the evolving technological landscape.
- Personal development: Pathways to advanced learning and application.

This detailed, interactive, and hands-on training is formulated to ensure a complete understanding and practical knowledge of building, managing, and scaling applications using microservices, Node.js, and Kafka. Through this journey, we aim to empower you to be industry-ready and capable of handling real-world projects efficiently and effectively.