Training Project : Product Store Spring Boot Microservice

Project Overview

This training project focuses on building a microservices-based product store using Spring Boot. The system comprises multiple services handling product management, orders, inventory, and notifications. It leverages modern technologies such as Kafka for messaging, Keycloak for authentication, and Kubernetes for deployment, ensuring a scalable and cloud-ready architecture.

Learning Objectives

- Building Spring Boot REST APIs
- Database Persistence using Spring Data JPA, MySQL, Flyway
- Event Driven Async Communication using Kafka
- Implementing OAuth2-based Security using Spring Security and Keycloak
- Implementing API Gateway using Spring Cloud Gateway
- Implementing Resiliency using Resilience4j
- Job Scheduling with ShedLock-based distributed Locking
- Using RestClient, Declarative HTTP Interfaces to invoke other APIs
- Creating Aggregated Swagger Documentation at API Gateway
- Local Development Setup using Docker, Docker Compose and Testcontainers
- Testing using JUnit 5, RestAssured, Testcontainers, Awaitility, WireMock
- Monitoring & Observability using Grafana, Prometheus, Loki, Tempo
- Kubernetes Kind cluster for development
- Kubernetes EKS for deployment

Services in Microservice projects:

Product Service for managing product information

CRUD operations for products
Stores product data in MongoDB
Exposes RESTful APIs

Order Service Handles order creation and management

Interacts with Product and Inventory services
Uses MySQL for storing order-related data
Publishes events to Kafka for asynchronous communication

Inventory ServiceManages product stock levels

Ensures stock availability before order placement
Stores inventory data in MySQL
Listens to Kafka events for stock updates

Notification Service Listens to order events and sends notifications

Uses Kafka to receive order status updates
Sends email/SMS notifications

API Gateway using Spring Cloud Gateway MVC

API gateway provide single entry point to the application, we need to apply security to the api gateway, so that only register app can interact with our application

Technologies Used

Backend Technologies Spring Boot: Core framework for microservices
Spring Cloud Gateway MVC: API Gateway for request routing
MongoDB: NoSQL database for Product and Inventory services
MySQL: Relational database for Order Service
Kafka: Event-driven communication between microservices
Keycloak: Identity and access management
Test Containers with Wiremock: Testing support for microservices
Monitoring and LoggingGrafana Stack:
Prometheus: Metrics collection
Grafana: Visualization and dashboarding
Loki: Log aggregation
Tempo: Distributed tracing
Deployment & CloudKubernetes: Orchestrating microservices
AWS Cloud: Hosting and scaling services