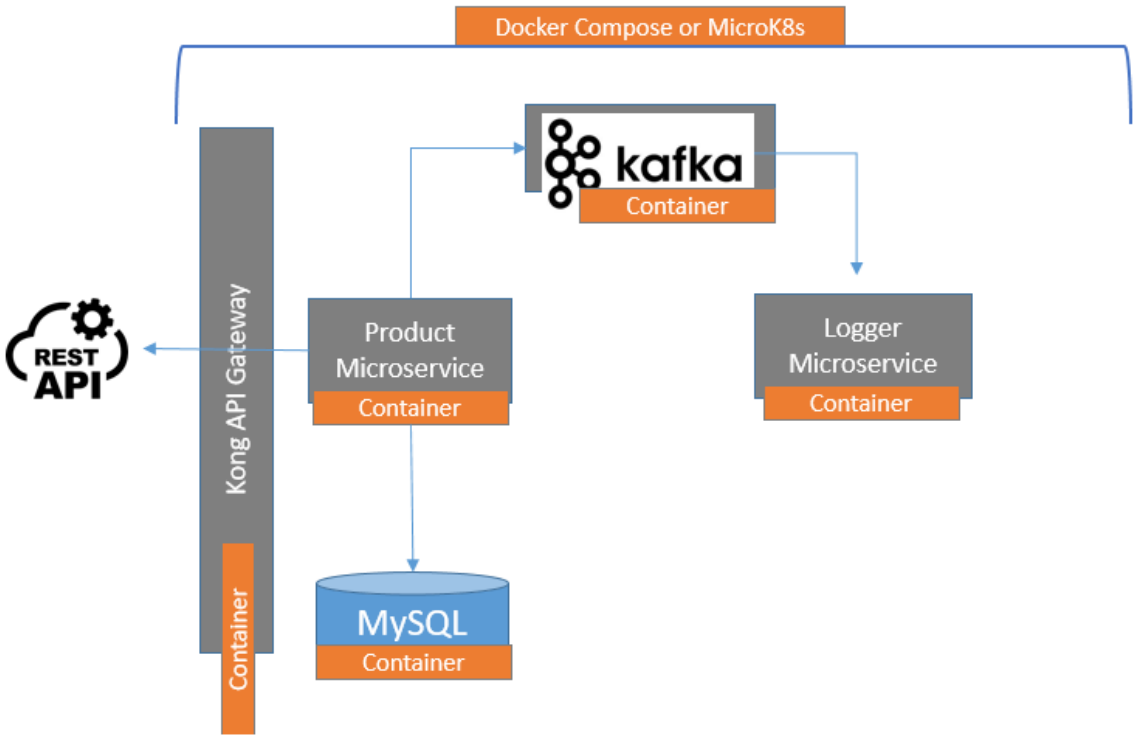
Collabera®

POC Application

Activity

Build two microservices using spring boot including java 8 / 11 features which communicates via message brocker exposing rest endpoints

Architecture



Product Microservice:

- Spring boot Microservice exposing rest endpoints for Products (listing & addition of products) & booking of Products

- Accepts orders.

-Broadcast all the activities to Kafka for next Microservice to log

Activity Logger:

- Spring boot Microservice to process all the activity log
- which listens for events and stores in nosql dB

Requirement

-Product Microservice should expose POST, GET and PUT / PATCH API's

- API's request validations should be performed and valid error/exception should be returned

-Microservices should have a proper exception and error handling across

- Microservices should include health check framework/ dependency

- Leverage swagger for API doc

- Product Microservice should leverage JPA / Hibernate as ORM along with repositories

- Microservice should follow the below structure

- Controller

-SampleController

-DTO

- RequestDTO

-ResponseDTO

- Service

SampleService

-ServiceImpl

-SampleImpService

-DTO

-SampleDTO

-Repository

-SampleRepository

-Entity

-Exceptions

- Utils

- Additions folders / substructures can be assumed

- Necessary tables can be assumed

- Use necessary implementation for logging and security (APO, Interceptors, Service mesh)

- Application should be Dockerized

- Both the Microservice, Kafka, DB, and API Gateway should be running on Docker and can be on same network using Docker Compose or deployed in MicroK8s

- Coding standards and Microservice best practice to be followed

Deliverables

- Technical / Design document (design pattern and other considerations made)
- Github link (source code along with docker composer or K8 yaml files)

Note :

- **API Gateway is optional (can run on a container for context-based routing)**
- **Install SonarLint Extention to your IDE**