

# **Cloud Based Project**

## **Building Microservices with Spring Cloud and Docker**

### **Demonstrated concepts:**

- Integration testing using Docker
- Polyglot persistence
- Microservice architecture
- Service discovery
- API gateway

### **Docker**

Each service is built and deployed using Docker. End-to-end integration testing can be done on a developer's machine using Docker compose.

### **Polyglot Persistence**

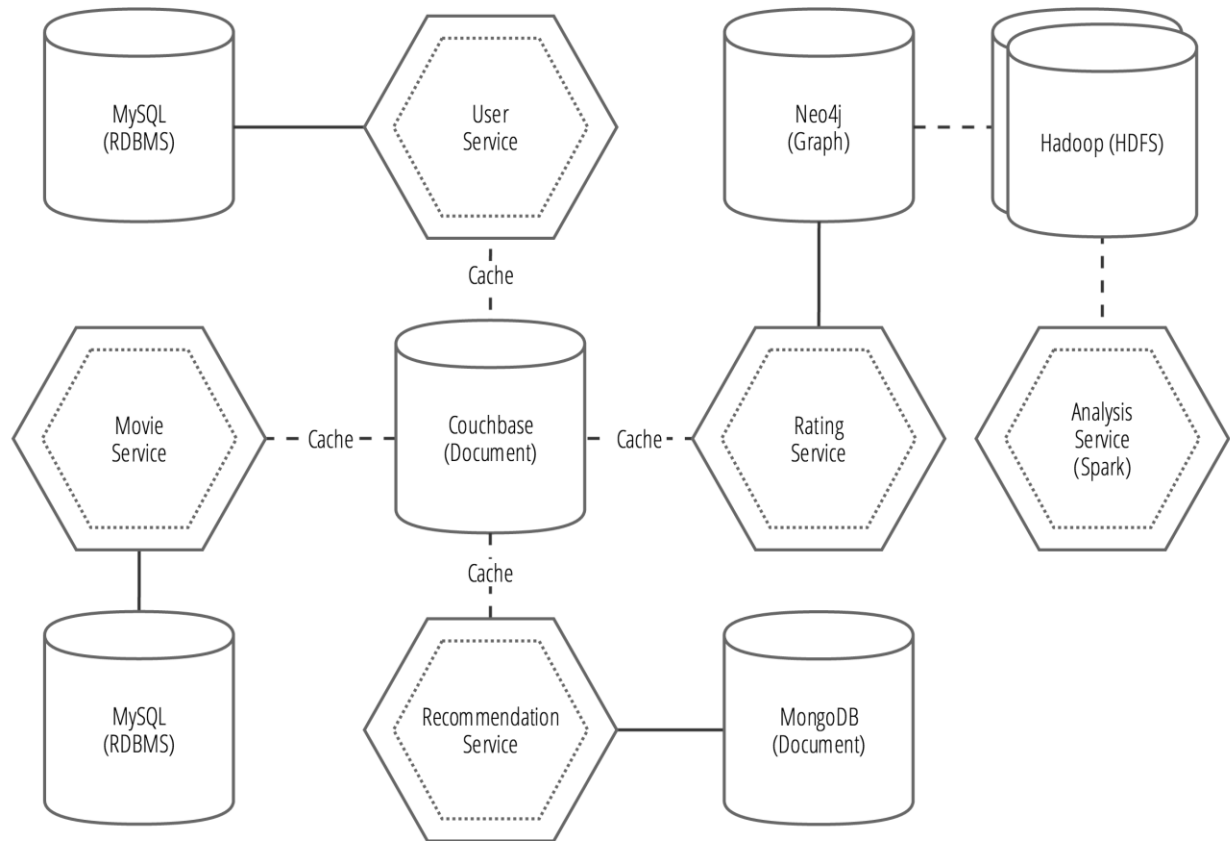
One of the core concepts of this example project is how polyglot persistence can be approached in practice. Microservices in the project use their own database, while integrating with the data from other services through REST or a message bus.

- Neo4j (graph)
- MongoDB (document)
- MySQL (relational)

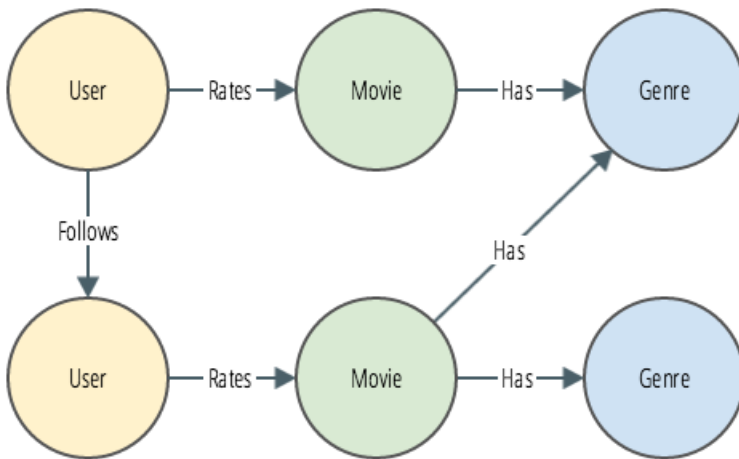
### **Movie Recommendations**

This example project focuses on movies and recommendations.

## Data Services



## Domain Data



## Microservice architecture

This example project demonstrates how to build a new application using microservices, as opposed to a monolith-first strategy. Since each microservice in the project is a module of a single parent project, developers have the advantage of being able to run and develop with each microservice running on their local machine. Adding a new microservice is easy, as the discovery microservice will automatically discover new services running on the cluster.

## Service discovery

This project contains two discovery services, one on Netflix Eureka, and the other uses Consul from Hashicorp. Having multiple discovery services provides the opportunity to use one (Consul) as a DNS provider for the cluster, and the other (Eureka) as a proxy-based API gateway.

## API gateway

Each microservice will coordinate with Eureka to retrieve API routes for the entire cluster. Using this strategy each microservice in a cluster can be load balanced and exposed through one API gateway. Each service will automatically discover and

route API requests to the service that owns the route. This proxying technique is equally helpful when developing user interfaces, as the full API of the platform is available through its own host as a proxy.

## **License**

This project is an open source product licensed under GPLv3.