**Microservices based Hotel User Rating Application**

* **Project Start Date: 14rd November,2022**

The project is built on Spring Boot with a microservices architecture and has three different service projects, mainly user, rating and hotel services. All these services are registered with the Eureka server and use the client as a Spring-based Eureka client. We mainly used MySQL, PostgreSQL and MongoDB databases for services such as users, hotels, ratings.

These services are passed by API Gateway, i.e. Spring Cloud Gateway, and share a common port number with different service names as path endpoints.

We collected the data from the hotel and rating service and transferred it into the user service using the Fiegn client to retrieve the specific rating and hotel details for this user id present in the database.

The hotel service and rating service also operate independently as APIs for retrieving hotel and rating details. The service's configuration is stored on Github and retrieved by spring-cloud-config-server by passing the Github URL in the application.properties file.

We Monitor our services using Actuator and Resilence 4j to monitor the flow of API calls to see and manage if our services are down or degraded.

The project is currently in working status on adding more functionality to this microservice application.

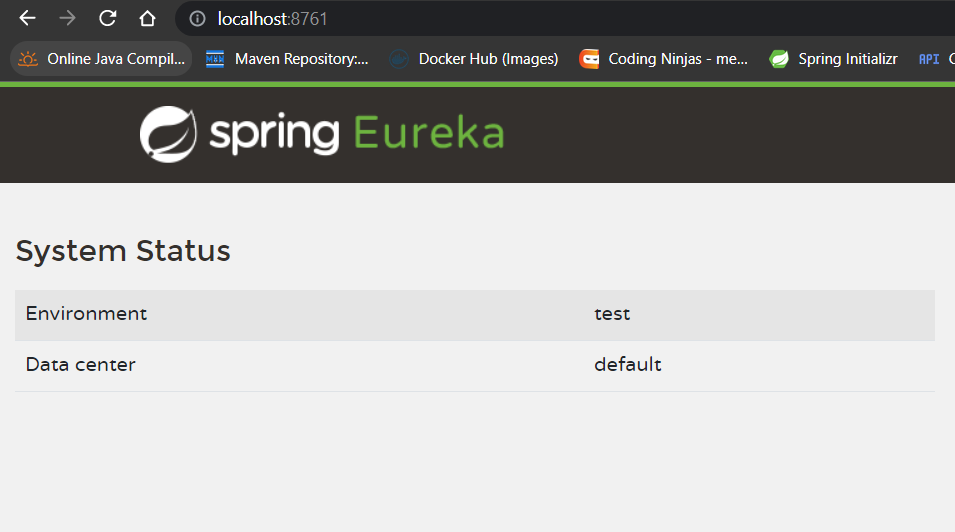
* **Dependencies used in project:**

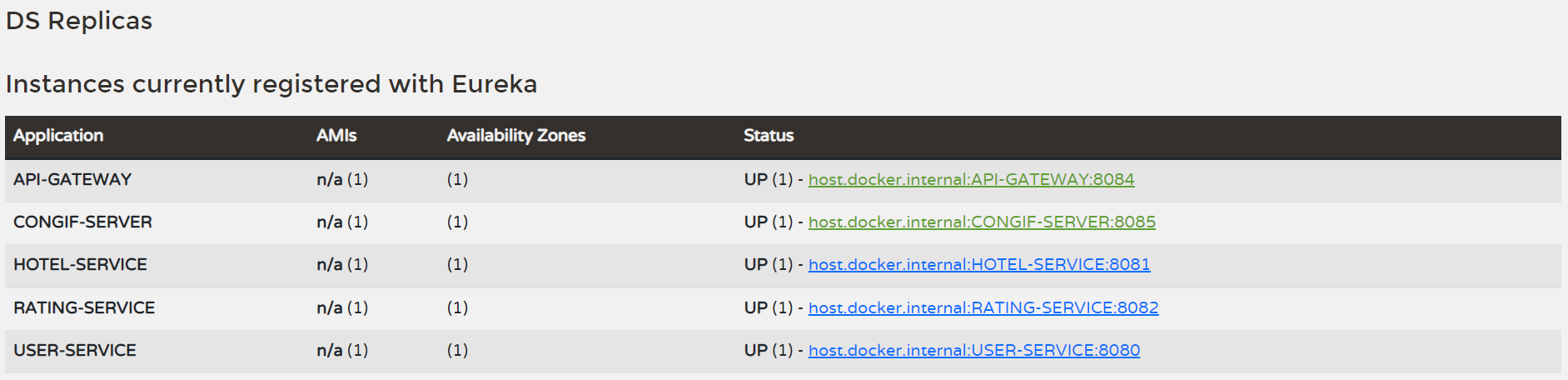
1. ***spring-boot-starter-data-jpa***
2. ***spring-boot-starter-web***
3. ***spring-boot-starter-actuator***
4. ***lombok***
5. ***spring-cloud-starter***
6. ***spring-cloud-starter-netflix-eureka-client -> common on user, hotel and rating services***
7. **mysql-connector-j -> user-service**
8. **postgresql -> hotel-service**
9. **spring-boot-starter-data-mongodb -> rating-service**
10. **\*\*\*\*we can also use RestTemplate/open feign**
11. **spring-cloud-starter-openfeign -> used for post,create for other service in present service**
12. **spring-cloud-starter -> for service registry of server**
13. **spring-cloud-starter-netflix-eureka-server -> for service registry of server**
14. **spring-cloud-starter-gateway -> API Gateway**
15. **spring-boot-starter-webflux -> used as runtime for spring cloud starter-gateway**
16. **spring-cloud-starter**
17. **spring-cloud-starter-netflix-eureka-client**
18. **spring-cloud-config-server – config server**

* **\*\*\*\*\*\*Ports\*\*\*\***

1. **USER\_SERVICE - 8080**
2. **HOTEL-SERVICE - 8081**
3. **RATING-SERVICE - 8082**
4. **Service Registry - 8761**
5. **API-GATEWAY - 8084**
6. **Config Server- 8085**

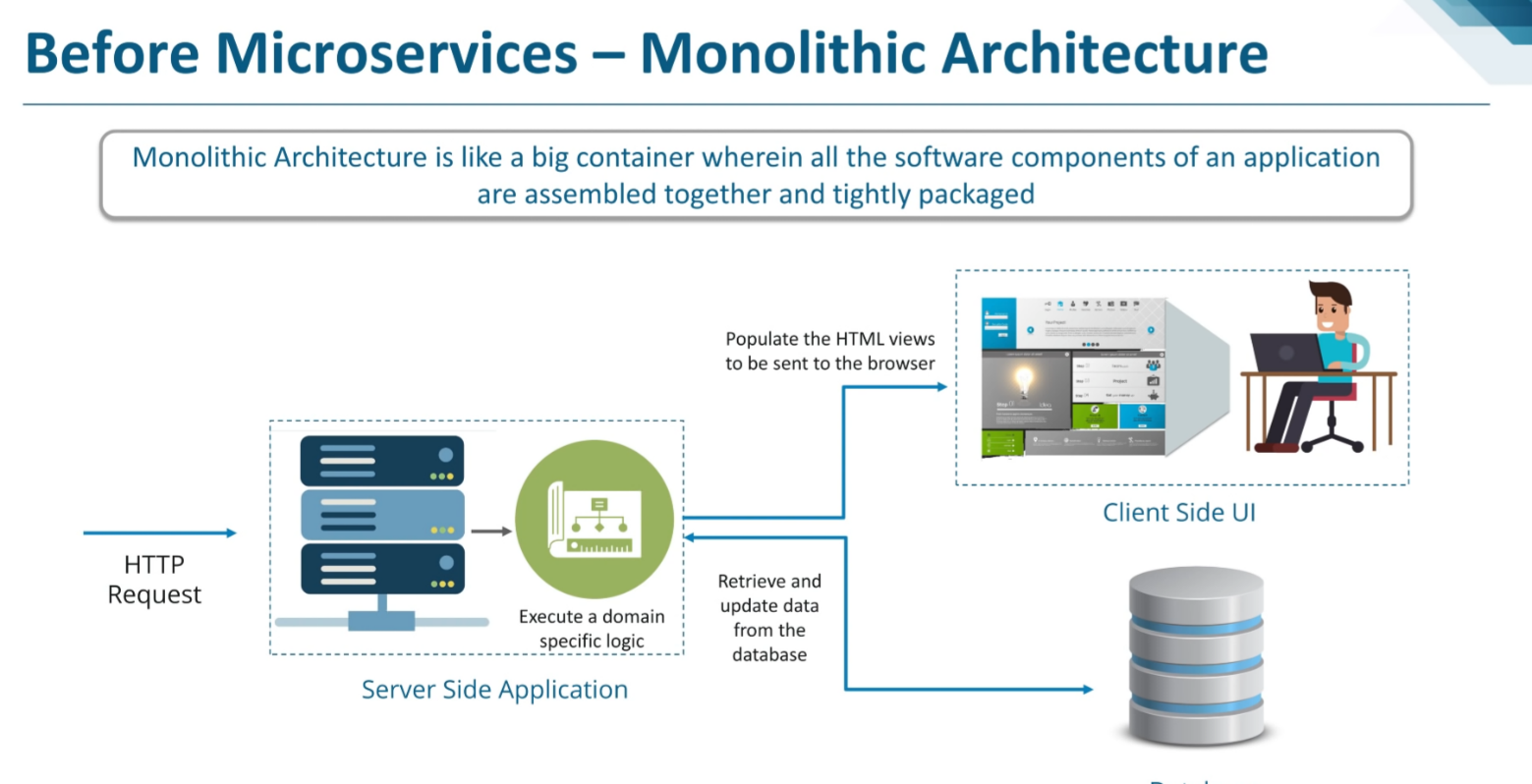
**Screenshot:**

****

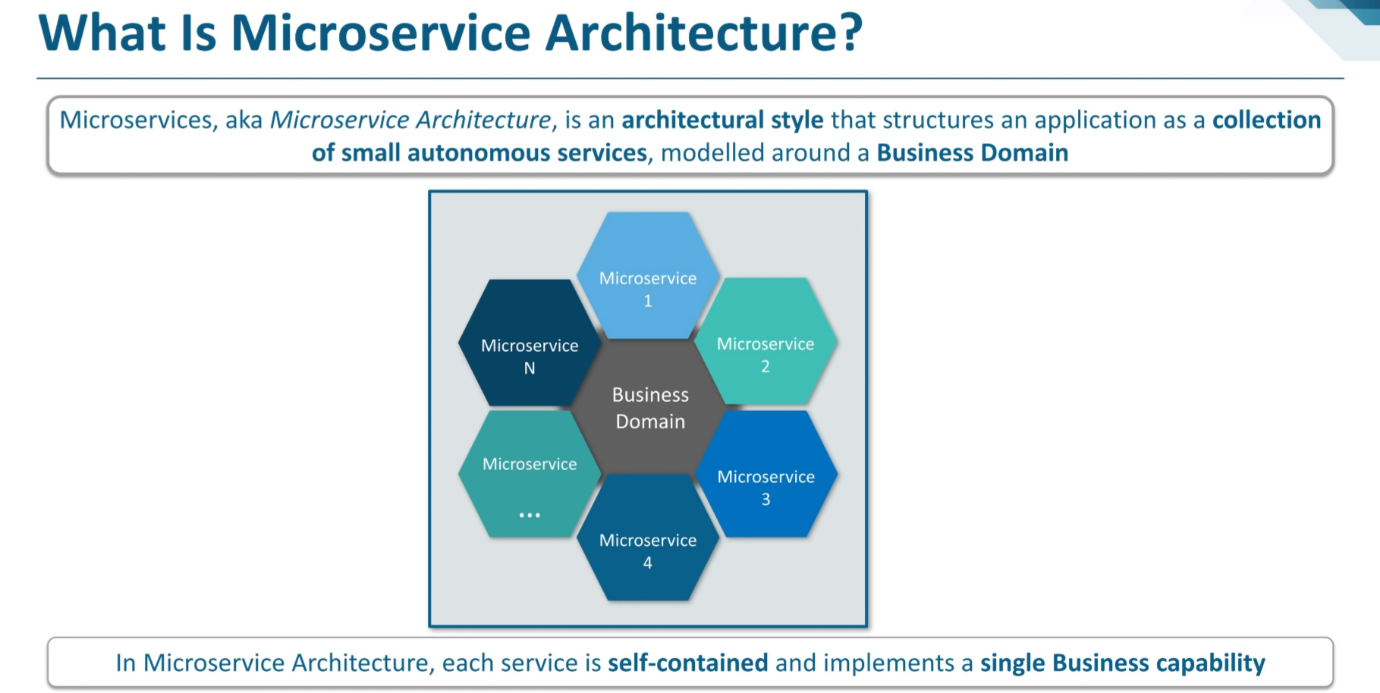
****

* **Detailed Project Parts:**

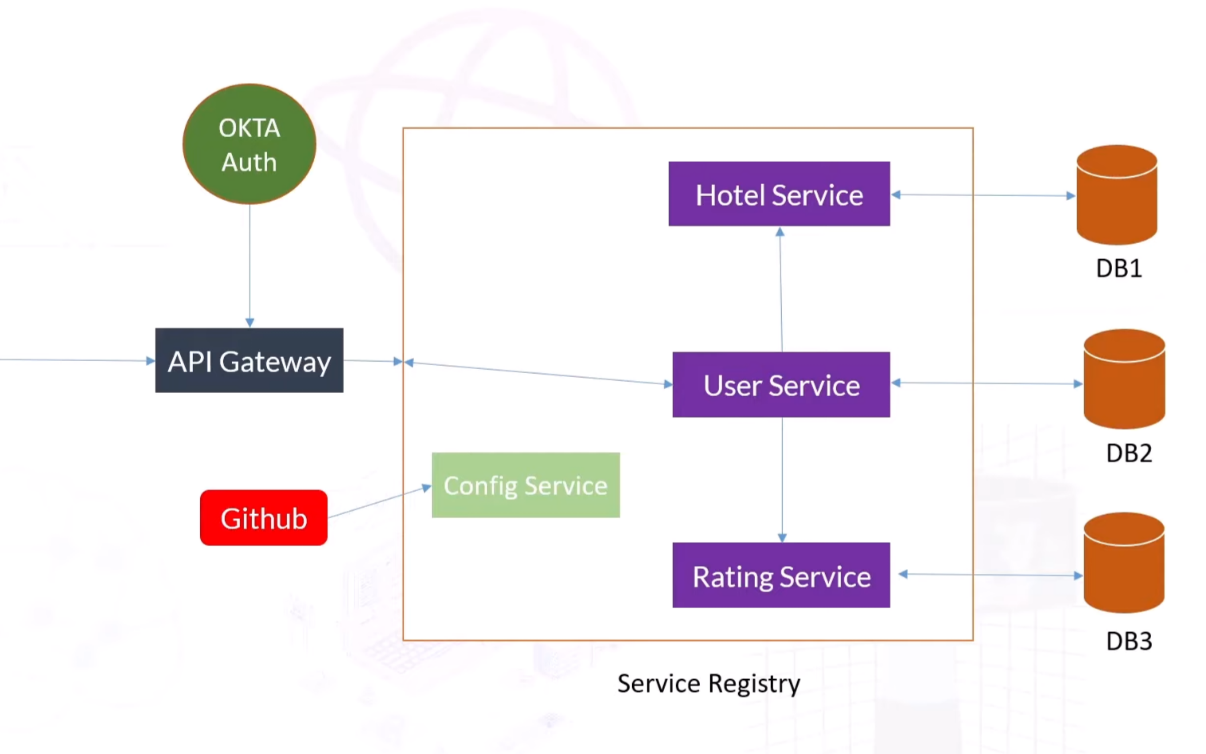
**Before Microservices are used:**

****

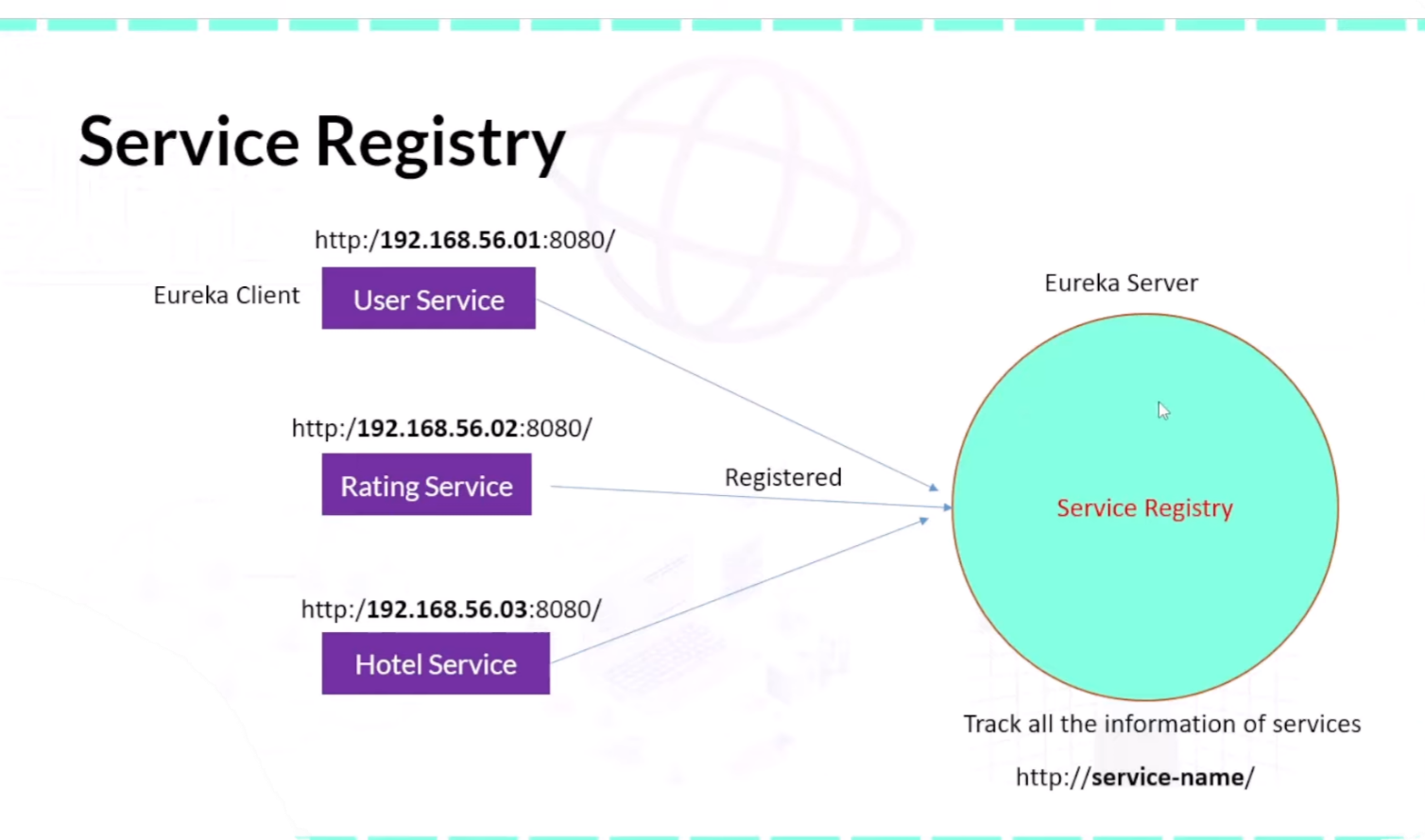
**What is Microservices:**

****

**Design Flow of Hotel Rating Application:**

****

**Additional Area of Service Registry and Client:**

****

**For Config Server:**

**Register the service of config server on service registry.**

**After that go to ->** [**http://host.docker.internal:8085/application/default**](http://host.docker.internal:8085/application/default) **via service registry**

**And you will see that there is "propertySources": [] due to no content in github for application.yml(by default).**

**After adding in git for application.properties/ application.yml, it will show like this :**

****

**Adding this dependency :**

**<dependency>**

**<groupId>org.springframework.cloud</groupId>**

**<artifactId>spring-cloud-starter-config</artifactId>**

**</dependency>**

**To 3 services and then add in application.properties:**

**spring.config.import**= **configserver:http://localhost:8085**

**and comment:**

*#eureka.instance.prefer-ip-address=true  
#eureka.client.fetch-registry=true  
#eureka.client.register-with-eureka=true  
#eureka.client.service-url.defaultZone=http://localhost:8761/eureka*

You will see the changes in the services