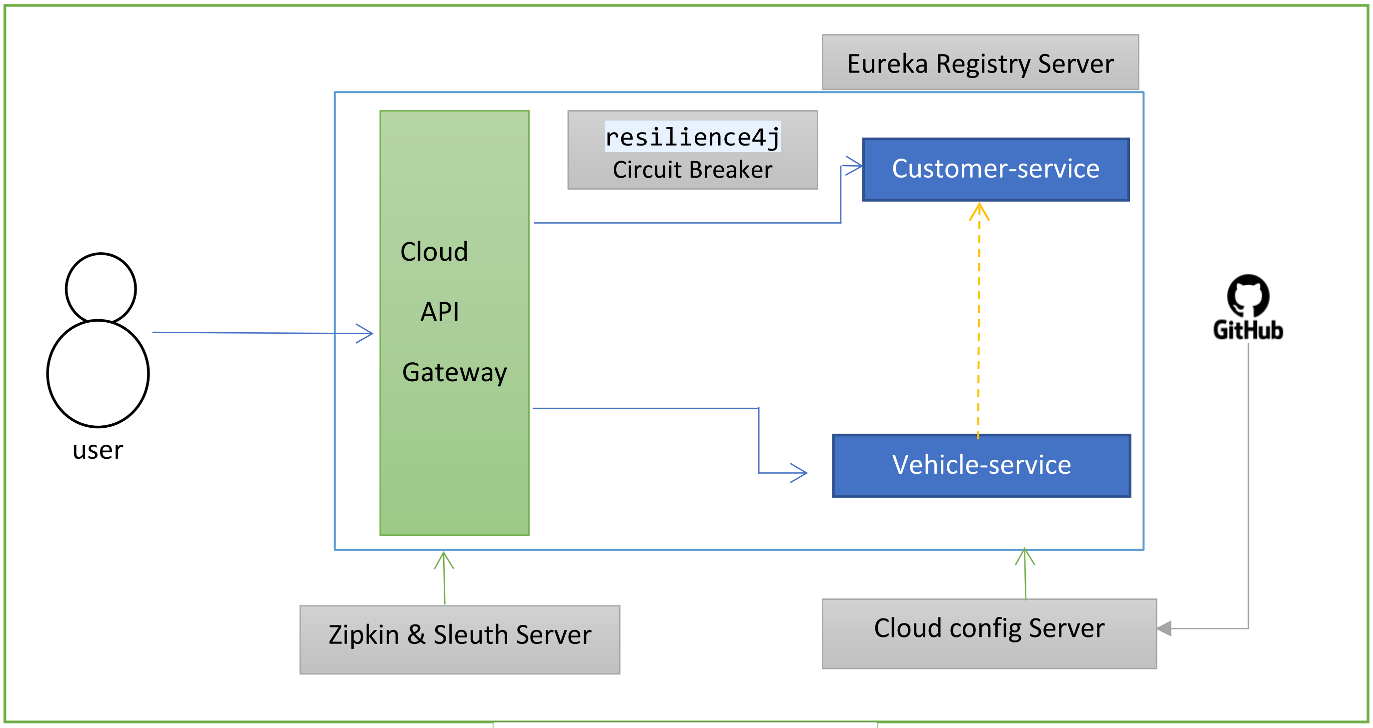
Spring boot

Micro-Service Architecture tutorial

This tutorial implement below list and shown in figure “Microservices Architecture”

1. Customer-service:
2. Vehicle-service
3. Eureka registry Server
4. Cloud API Gateway
5. Cloud config Server
6. Zipkin & Sleuth Server



Microservices Architecture

Will go through component in brief the aim of this tutorial for learning propose

Customer-service:

Spring boot contains two Endpoints saveCustomer and getCutomer

a. saveCustomer : endpoint accept customer detail as JSON and persist it in Mysql database below is customer details

Customer
Long customerId
String firstName
String lastName
String email
String phone

To test in postman client use below

URL POST : http://localhost:8003/customers/

b. getCutomer : endpoint will retrieve customer details along with vehicles which registered under his name from Mysql database, to get the vehicles that are registered under certain customer we will call vehicle service which deployed in remote host , to avoid service unavailable we implements resilience circuit breaker for fault tolerance .

To test in postman client use below
URL GET : http://localhost:8003/customers/1

source: https://github.com/waliedAfro/spring-boot-customer-service

Vehicle-service :

Spring boot contains two Endpoints `saveVehicle` and `getVehilce ,` `getVehilceByCustomerID`

a. `saveVehicle` : endpoint accept Vehicle detail along with customer id as JSON and persist it in database below is vehicle details

Vehicle
Long vehicleId
String chassiNo
String bran
String model
Long customerId

To test in postman client use below

URL POST : http://localhost:8002/vehicles/

b. `getVehilceByCustomerID:` endpoint will retrieve list of vehicles by customer id

To test in postman client use below
URL GET : http://localhost:8002/vehicles/customer/1

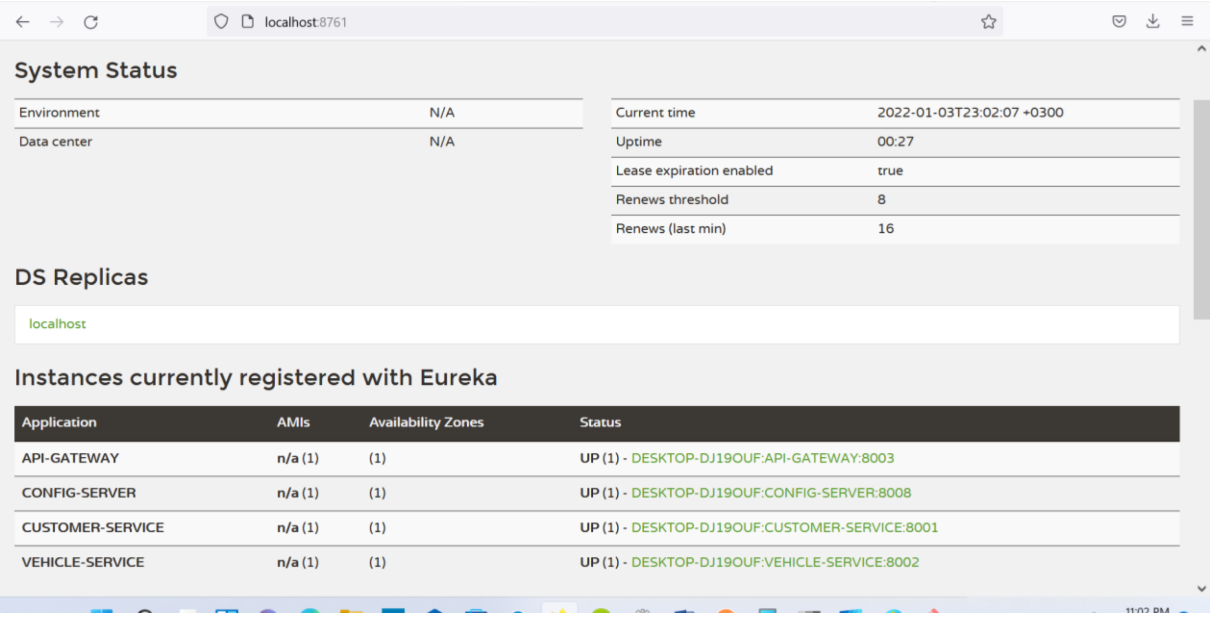
source: https://github.com/waliedAfro/spring-boot-vehicle-service

Eureka registry Server

Server registry will register below services

- customer-service
- vehicle-service
- Cloud API Gateway
- Cloud config Server

Can access from browser URL : http://localhost:8761



System Status

Environment	N/A
Data center	N/A

Current time	2022-01-03T23:02:07 +0300
Uptime	00:27
Lease expiration enabled	true
Renews threshold	8
Renews (last min)	16

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - DESKTOP-DJ19OUF:API-GATEWAY:8003
CONFIG-SERVER	n/a (1)	(1)	UP (1) - DESKTOP-DJ19OUF:CONFIG-SERVER:8008
CUSTOMER-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-DJ19OUF:CUSTOMER-SERVICE:8001
VEHICLE-SERVICE	n/a (1)	(1)	UP (1) - DESKTOP-DJ19OUF:VEHICLE-SERVICE:8002

source: https://github.com/waliedAfro/spring-boot-server-registry

Cloud API Gateway

Gateway Receiving all request from client and redirect it to corresponding service and return response .

source: https://github.com/waliedAfro/spring-boot-API-GATEWAY

Cloud config Server

Cloud config server read central config from github

source: https://github.com/waliedAfro/spring-boot-cloud-config-server

Zipkin & Sleuth Server

Center log server for our services we can download the zipkin server from

https://search.maven.org/remote_content?g=io.zipkin&a=zipkin-server&v=LATEST&c=exec

the default server port is 9411 add below config in API gateway >>resource>> bootstrap.yml

```
zipkin:
   base-url : http://127.0.0.1:9411
```

Thank you

Elwalied Elnour

Senior Java Developer

Wad393@gmail.com