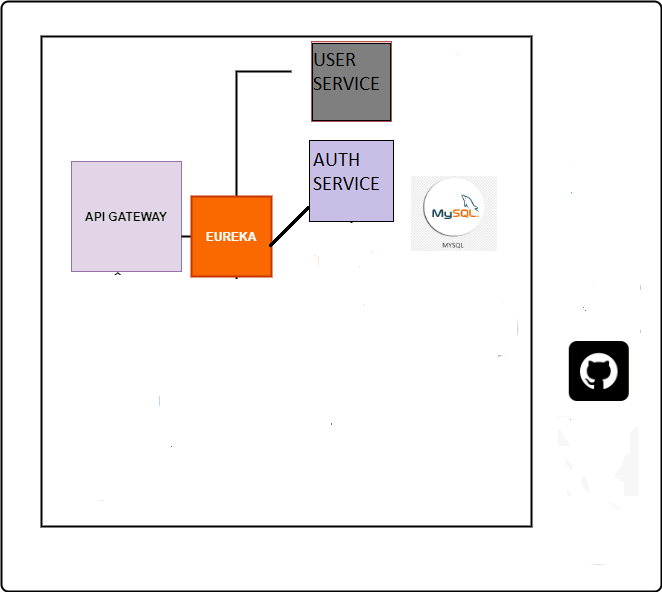
MICRO-SERVICE

TRẦN TRUNG HIẾU

Link github:

* <https://github.com/mariohieu09/micro-architect>

1. **Tổng quan kiến trúc logic micro-service**



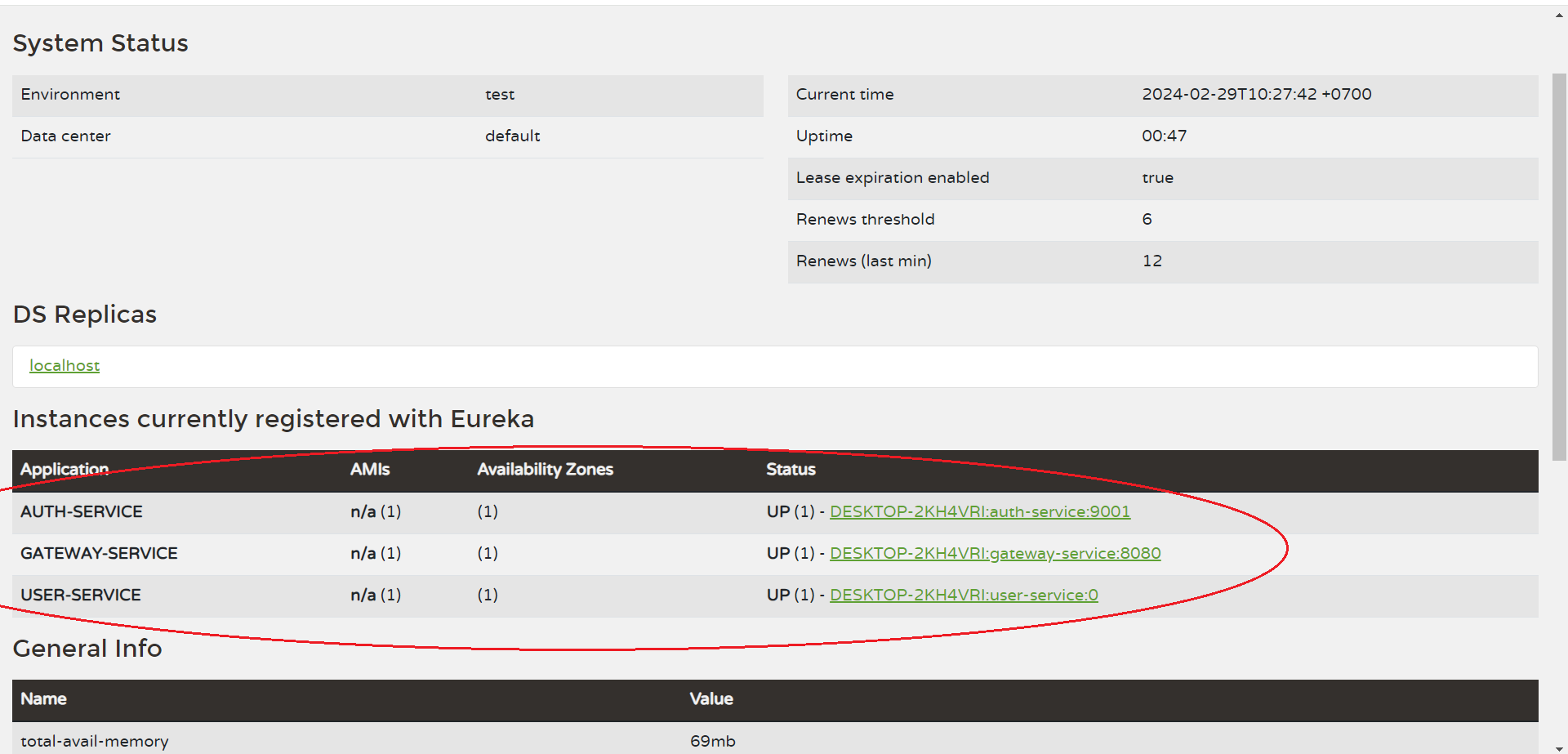
1. **Các thành phần chính và nhiệm vụ của chúng**
2. **Eureka (Discovery-server)**

Eureka server là một thành phần quan trọng trong Spring Cloud, một framework được sử dụng để xây dựng và quản lý các ứng dụng phân tán. Eureka server đảm nhận vai trò là một dịch vụ Registry (đăng ký) và Discovery (khám phá).

Đây là những vai trò mà Eureka đảm nhiệm:

* **Registry Service (Dịch vụ Đăng ký):** Eureka server chấp nhận các đăng ký từ các service client (các ứng dụng đang chạy) và duy trì một danh sách các service instances đang hoạt động trong hệ thống. Mỗi service instance được đăng ký sẽ được gán một tên duy nhất (ví dụ: tên của service và địa chỉ IP hoặc tên máy chủ).
* **Discovery Service (Dịch vụ Khám phá):** Eureka server cung cấp các API cho các service client để họ có thể truy vấn và khám phá các service instances khác trong hệ thống. Nhờ vào Eureka server, các service client có thể biết được các instance của service nào đang hoạt động, địa chỉ của chúng và sẵn sàng giao tiếp.
* **Fault Tolerance (Khả năng chống lỗi):** Eureka server có khả năng phát hiện và loại bỏ các service instances không hoạt động khỏi danh sách đăng ký, giúp đảm bảo rằng các service client chỉ truy vấn và tương tác với các service instances đang hoạt động.

Ví dụ:



Eureka đảm nhiệm vai trò tìm kiếm và phát hiện những dịch vụ đã được đăng kí với Eureka (client service). Như trong hình thì có 3 dịch vụ lần lượt là AUTH-SERVICE,

GATEWAY-SERVICE và USER-SERVICE đăng ký với eureka.

1. **API-Gateway (Gateway server)**

Gateway Server đảm nhiệm vai trò là một điểm truy cập (entry point) vào hệ thống phân tán. Nó là một dịch vụ có khả năng định tuyến (routing) và lọc (filtering), giúp điều hướng yêu cầu từ người dùng đến các dịch vụ (microservices) phía sau.

Dưới đây là một số vai trò quan trọng mà Gateway Server thường thực hiện:

* **Routing (Định tuyến):** Gateway Server nhận các yêu cầu từ người dùng và dựa vào các quy tắc định tuyến được cấu hình để định tuyến các yêu cầu đó đến các dịch vụ (microservices) tương ứng trong hệ thống phân tán. Điều này giúp ẩn đi sự phức tạp của cấu trúc hạ tầng và cung cấp một giao diện đơn giản, dễ sử dụng cho người dùng cuối.
* **Authentication and Authorization (Xác thực và Ủy quyền):** Gateway Server thường là nơi xử lý xác thực (authentication) và ủy quyền (authorization) cho các yêu cầu. Nó có thể kiểm tra các thông tin đăng nhập, token hoặc các header để xác định quyền truy cập của người dùng và quyết định xem họ có quyền truy cập vào các dịch vụ nào.
* **Logging and Monitoring (Ghi log và Giám sát):** Gateway Server có thể thêm thông tin log và gửi các metric liên quan đến yêu cầu truy cập vào các hệ thống giám sát (monitoring systems). Điều này giúp theo dõi hiệu suất và sự kiện xảy ra trong hệ thống.
* **SSL Termination:** Gateway Server có thể chịu trách nhiệm cho việc kết thúc kết nối SSL (SSL termination), giúp mã hóa và giải mã dữ liệu giữa người dùng và dịch vụ phía sau.
* **Protection from Attacks:** Gateway Server có thể cung cấp các bộ lọc và biện pháp bảo vệ để ngăn chặn các cuộc tấn công như DDoS hoặc các lỗ hổng bảo mật.
* **Pre-assigned permissions and roles: Gateway Server thực hiện lưu và cập nhật những quyền hạn và vai trò được chỉ định sẵn trong hệ thống phân tán.**

1. **Auth-service (Authentication and Authorization server)**

Auth-service đảm nhiệm vai trò phân quyền và xác thực người dùng trong hệ thống phân tán. Nó thực hiện nhiệm vụ kiểm tra và xác thực thông tin người dùng, đồng thời phân quyền cho người dùng trong hệ thống. Auth-service cũng được tích hợp đăng kí người dùng.

* **Authenticate**: Auth-service xác thực người dùng dựa theo thông tin xác thực (ví dụ: tên đăng nhập và mật khẩu). Sau khi xác thực người dùng thành công sẽ trả về một JWT (Json web token) chứa thông tin như username, credential, permission, user id, và một số những thông tin thiết yếu của token.
* **Register**: Auth-service cũng đảm nhiệm việc đăng ký người dùng trong hệ thống, người dùng sẽ nhập thông tin tên đăng nhập và mật khẩu để đăng ký với tài khoản trong hệ thống. Mật khẩu người dùng sẽ được mã hoá với thuật toán AES. Mỗi một mật khẩu người dùng sẽ có tạo một bản khi chứa khoá riêng để giải mã mật khẩu.
* **Authorize**: Auth-service sẽ đảm nhiệm vai trò xác thực người dùng trên đường dẫn tài nguyên mà người dùng muốn truy cập (phân quyền trên đường dẫn). Auth-service sẽ nhận một yêu cầu đến từ api-gateway để kiểm tra xem quyền hạn của người dùng có được phép truy cập vào tài nguyên đó không. Auth-serivce sẽ thực hiện việc kiểm tra sau đó trả về kết quả cho gateway.

1. **User-service**

User-service đảm nhiệm vai trò cung cấp các api để thao tác với người dùng ví dụ như: thêm, sửa, xoá, chỉnh sửa thông tin người dùng,…

* **Get user:** API được cung cấp để lấy về thông tin một người dùng được chỉ định. Chỉ user có quyền GET\_USER mới được truy cập vào api này.
* **Post user:** API được cung cấp để tạo mới một người dùng trong hệ thống. Chỉ user có quyền POST\_USER mới được truy cập vào api này.
* **Put user:**  API được cung cấp để chỉnh sửa một người dùng trong hệ thống. Chỉ user có quyền PUT\_USER mới được truy cập vào api này.
* **Delete user:** API được cung cấp để xoá một người dùng trong hệ thống. Chỉ user có quyền DELETE\_USER mới được truy cập vào api này.

1. **Source code management**

Source code của chương trình được lưu trữ sử dụng Github.