BÁO CÁO THỰC HÀNH LAB 6

API Gateway Pattern & Security

1. Mục tiêu bài lab

Mục tiêu của bài thực hành Lab 6 là:

- Hiểu được vai trò của **API Gateway** trong kiến trúc Microservices.
- Triển khai API Gateway như một **điểm vào duy nhất (Single Entry Point)** cho client.
- Áp dụng **Security ASR** bằng cách thực hiện xác thực (authentication) tập trung tại Gateway.
- Thực hành cơ chế **Reverse Proxy** để chuyển tiếp request từ client đến backend service.
- Kiểm tra khả năng xử lý lỗi và đảm bảo tính sẵn sàng (Availability) của hệ thống.

2. Cơ sở lý thuyết

2.1. API Gateway Pattern

API Gateway là một mẫu kiến trúc đóng vai trò như một lớp trung gian đứng giữa client và các microservices. Thay vì client phải gọi trực tiếp từng service với các địa chỉ và cổng khác nhau, toàn bộ request sẽ được gửi đến Gateway.

API Gateway có các chức năng chính như:

- Định tuyến request đến đúng backend service.
- Xác thực và phân quyền người dùng.
- Giới hạn lưu lượng truy cập (rate limiting).
- Thu thập log và monitoring.

2.2. Security ASR

Security ASR (Architectural Significant Requirement) yêu cầu hệ thống phải đảm bảo các cơ chế bảo mật được thực thi nhất quán. Việc đặt logic bảo mật tại API Gateway giúp:

- Ngăn chặn request không hợp lệ trước khi vào hệ thống.

- Giảm độ phức tạp cho các backend service.
- Tránh việc lặp lại code bảo mật ở nhiều nơi.

3. Mô hình kiến trúc hệ thống

Hệ thống trong bài lab bao gồm:

- **Client**: Gửi HTTP request.
- **API Gateway** (cổng 5000):
    - Kiểm tra token xác thực.
    - Chuyển tiếp request hợp lệ đến backend.
- **Product Service** (cổng 5001):
    - Xử lý nghiệp vụ liên quan đến sản phẩm.

Mọi request từ client đều phải đi qua API Gateway trước khi đến Product Service.

4. Triển khai hệ thống

4.1. Khởi tạo môi trường

Các bước thiết lập project API Gateway:

- Tạo thư mục project api_gateway.
- Khởi tạo môi trường ảo Python.
- Cài đặt các thư viện cần thiết: **Flask** và **requests**.
- Tạo file gateway.py để viết mã nguồn.

4.2. Cấu hình Gateway

API Gateway được cấu hình chạy trên cổng 5000 và biết địa chỉ của Product Service chạy trên cổng 5001. Gateway sử dụng Flask để tiếp nhận request từ client.

4.3. Triển khai kiểm tra bảo mật

Gateway sử dụng hàm validate_token để kiểm tra header Authorization trong request. Token hợp lệ phải có dạng:

Authorization: Bearer valid-user-token

Nếu token không tồn tại, sai định dạng hoặc không hợp lệ, Gateway sẽ trả về mã lỗi **401 Unauthorized**.

4.4. Cơ chế định tuyến và chuyển tiếp request

Sau khi request vượt qua bước kiểm tra bảo mật, Gateway sẽ:

- Xây dựng URL của backend service.
- Chuyển tiếp request với đầy đủ HTTP method, header và body.
- Nhận response từ backend và trả nguyên vẹn lại cho client.

Trong trường hợp backend service không phản hồi hoặc bị dừng, Gateway sẽ trả về mã lỗi **503 Service Unavailable**.

5. Kiểm thử hệ thống

5.1. Kiểm thử truy cập không xác thực

- **Mô tả:** Gửi request không kèm header Authorization.
- **Kết quả mong đợi:** Gateway từ chối request.
- **Kết quả thực tế:** Trả về mã lỗi 401 Unauthorized.

5.2. Kiểm thử truy cập hợp lệ

- **Mô tả:** Gửi request kèm token hợp lệ.
- **Kết quả mong đợi:** Request được định tuyến đến Product Service.
- **Kết quả thực tế:** Trả về mã 200 OK cùng dữ liệu sản phẩm.

5.3. Kiểm thử khi backend không hoạt động

- **Mô tả:** Dừng Product Service và gửi lại request hợp lệ.
- **Kết quả mong đợi:** Gateway xử lý lỗi một cách an toàn.
- **Kết quả thực tế:** Trả về mã 503 Service Unavailable.

6. Đánh giá và nhận xét

6.1. Ưu điểm

- Bảo mật được triển khai tập trung và hiệu quả.
- Client không cần biết chi tiết các backend service.

- Dễ dàng mở rộng và quản lý hệ thống.

6.2. Hạn chế

- API Gateway có thể trở thành **Single Point of Failure**.
- Cần triển khai thêm cơ chế load balancing và dự phòng để đảm bảo độ sẵn sàng cao.

7. Kết luận

Thông qua bài thực hành Lab 6, em đã hiểu rõ cách áp dụng **API Gateway Pattern** trong kiến trúc Microservices cũng như vai trò quan trọng của Gateway trong việc đảm bảo bảo mật và tính sẵn sàng của hệ thống. Việc triển khai xác thực tập trung giúp hệ thống an toàn hơn và dễ bảo trì hơn trong các hệ thống phân tán quy mô lớn.