Amazon API Gateway

Xây dựng Serverless API

Quản lý API một cách hiệu quả, bảo mật và mở rộng với AWS API Gateway


Bảo mật


Integration


Deployment


Monitoring

2025-11-01

API Gateway là gì?

Amazon API Gateway là một dịch vụ quản lý API đầy đủ, cho phép các nhà phát triển dễ dàng tạo, xuất bản, duy trì, giám sát và bảo mật API ở mọi quy mô. Nó hoạt động như một "cánh cổng duy nhất" cho tất cả các dịch vụ và API của bạn, cung cấp một điểm truy cập thống nhất và an toàn cho các ứng dụng client.

Fully Managed

Dịch vụ được quản lý hoàn toàn bởi AWS, không cần lo lắng về việc quản lý máy chủ hay cơ sở hạ tầng.

Lambda Integration

Kết hợp với AWS Lambda để xây dựng các ứng dụng serverless mà không cần quản lý hạ tầng, tối ưu hóa chi phí và khả năng mở rộng.

WebSocket Support

Hỗ trợ các API WebSocket cho phép giao tiếp hai chiều, liên tục giữa client và server.

API Versioning

Dễ dàng quản lý nhiều phiên bản API cùng lúc, cho phép phát triển và triển khai các bản cập nhật mà không ảnh hưởng đến các ứng dụng đang sử dụng phiên bản cũ.

Environment Management

Cung cấp khả năng tạo và quản lý các môi trường triển khai khác nhau (ví dụ: phát triển, thử nghiệm, sản xuất), giúp quy trình phát triển và kiểm thử trở nên có tổ chức hơn.

API Gateway – HTTP API vs REST API

HTTP APIs

- **Hiệu suất:** Low-latency (Độ trễ thấp)
- **Chi phí:** Cost-effective (Hiệu quả về chi phí)
- **Tính năng:** Built-in CORS, tích hợp nhanh với Lambda và HTTP backend

Trường hợp sử dụng phù hợp:

HTTP APIs là lựa chọn tối ưu khi bạn ưu tiên hiệu suất cao và chi phí thấp. Chúng rất phù hợp để tạo proxy đến các hàm Lambda hoặc các HTTP backend khác.

REST APIs

- **Hiệu suất:** Cao, nhưng có thể cao hơn HTTP APIs do nhiều tính năng hơn
- **Chi phí:** Chi phí cao hơn HTTP APIs
- **Tính năng:** Đầy đủ tính năng (Usage Plans, API Keys, Request Validation, Caching, WAF, v.v.)

Trường hợp sử dụng phù hợp:

REST APIs cung cấp bộ tính năng phong phú hơn, bao gồm các khả năng quản lý nâng cao như Usage Plans, API Keys, Request Validation, và tích hợp với AWS WAF, phù hợp cho các API yêu cầu kiểm soát và tùy chỉnh sâu rộng.

Hướng dẫn lựa chọn:

Chọn HTTP API nếu:
Bạn cần tối ưu hiệu suất và chi phí, với các API đơn giản hoặc trung gian đến Lambda.

Chọn REST API nếu:
Bạn cần toàn bộ sức mạnh và khả năng tùy chỉnh của API Gateway, các API phức tạp yêu cầu quản lý chi tiết.

API Gateway – Integrations

API Gateway đóng vai trò trung tâm trong việc tích hợp các yêu cầu từ client với các dịch vụ backend khác nhau. Nó hỗ trợ ba loại tích integration chính:

Lambda Function

Tích integration phổ biến nhất, cho phép API Gateway trực tiếp khởi chạy các hàm AWS Lambda để xử lý logic nghiệp vụ.

HTTP Endpoint

API Gateway có thể hoạt động như một proxy cho các HTTP endpoint hiện có, bao gồm máy chủ on-premise, ứng dụng EC2, hoặc Application Load Balancer.

AWS Service

Cho phép API Gateway gọi trực tiếp các API của các dịch vụ AWS khác như Amazon SQS, AWS Step Functions, hoặc Amazon S3, không cần thông qua Lambda trung gian.

Lợi ích chung của các tích integration:

- Thêm lớp bảo mật cho backend
- Caching để tối ưu hóa hiệu suất
- Rate limiting để bảo vệ backend

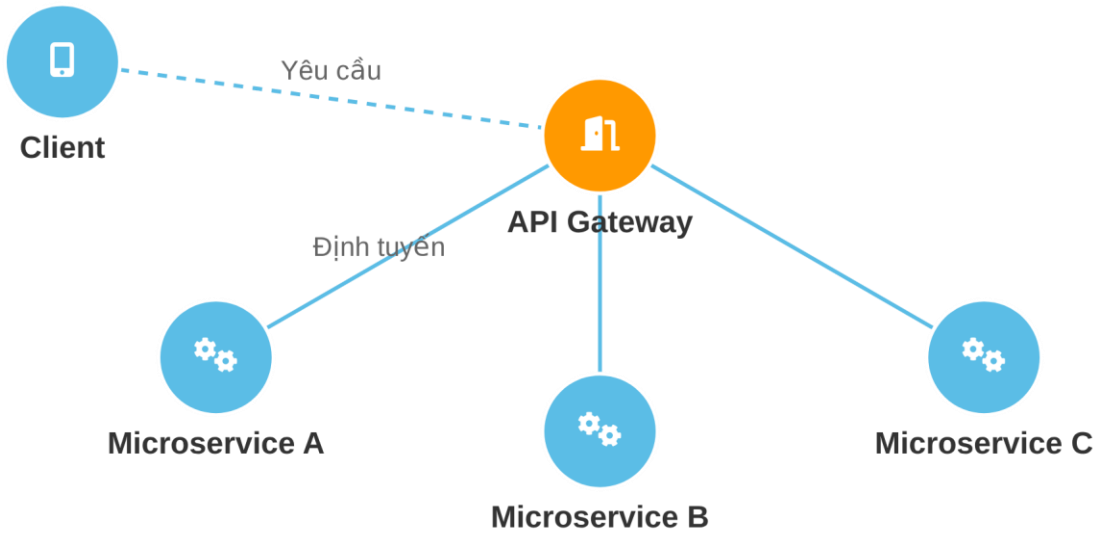
API Gateway – Architecture

Vai trò trong Microservices

Trong kiến trúc Microservices, API Gateway đóng vai trò là một "cánh cổng" duy nhất, cung cấp một giao diện thống nhất cho các client để tương tác với nhiều microservice khác nhau.

Các chức năng chính:

- Định tuyến các yêu cầu đến microservice phù hợp
- Áp dụng các quy tắc chuyển tiếp (Forwarding Rules)
- Biến đổi dữ liệu (Transformation Rules) nếu cần
- Hỗ trợ Custom Domain và SSL để đảm bảo giao tiếp an toàn



Lợi ích

API Gateway giúp che chắn các microservice khỏi client, giảm thiểu sự phụ thuộc giữa client và implementation của backend, đồng thời cung cấp một điểm truy cập thống nhất và an toàn

API Gateway – Endpoint Types

API Gateway cung cấp ba loại endpoint khác nhau để tối ưu hóa hiệu suất và khả năng truy cập tùy depending vào vị trí của người dùng và ứng dụng của bạn.

Edge-Optimized

- Sử dụng Amazon CloudFront để định tuyến các yêu cầu
- Yêu cầu được định tuyến qua các Edge Locations của CloudFront trên toàn cầu
- Giảm độ trễ cho người dùng ở xa
- CloudFront chuyển tiếp yêu cầu đến Regional API Gateway gần nhất

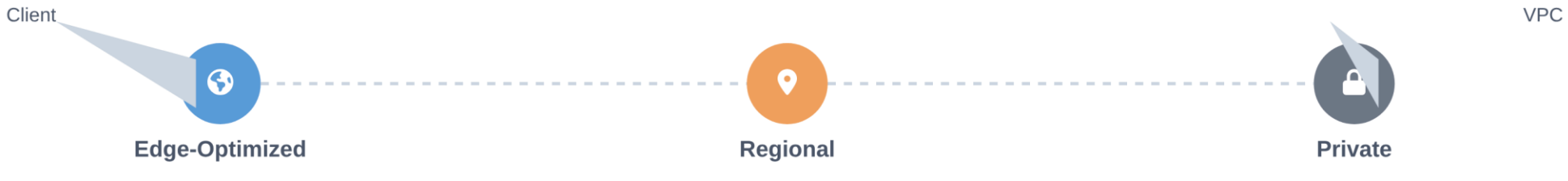
Regional

- Được triển khai trong một AWS Region cụ thể
- Client gọi trực tiếp đến API Gateway trong Region đó
- Phù hợp khi client và API Gateway nằm trong cùng một Region
- Tự quản lý việc phân phối nội dung bằng CDN của riêng mình

Private

- Chỉ có thể truy cập được từ bên trong Amazon VPC của bạn
- Truy cập thông qua một VPC Endpoint
- API của bạn không thể truy cập được từ internet công cộng
- Tăng cường bảo mật cho các ứng dụng nội bộ

Sơ đồ hoạt động của các endpoint:



API Gateway – Security Overview

API Gateway cung cấp nhiều lớp bảo mật mạnh mẽ để bảo vệ các API của bạn khỏi truy cập trái phép và các mối đe dọa khác. Các cơ chế bảo mật này được tích hợp sâu vào dịch vụ, cho phép bạn kiểm soát chặt chẽ ai có thể truy cập API và cách họ tương tác với chúng.

IAM (Identity and Access Management)

Sử dụng các chính sách IAM để cấp quyền truy cập chi tiết cho người dùng và vai trò trong AWS, đảm bảo chỉ những đối tượng được ủy quyền mới có thể gọi API.

Amazon Cognito

Tích hợp với Cognito để quản lý danh tính người dùng bên ngoài (ví dụ: người dùng ứng dụng di động hoặc web), cung cấp khả năng xác thực và ủy quyền mà không cần viết mã phức tạp.

Lambda Authorizer

Cho phép bạn sử dụng một hàm Lambda tùy chỉnh để thực hiện logic xác thực và ủy quyền phức tạp, hỗ trợ các loại token tùy chỉnh (như JWT) và có khả năng lưu trữ kết quả xác thực (cache) để cải thiện hiệu suất.

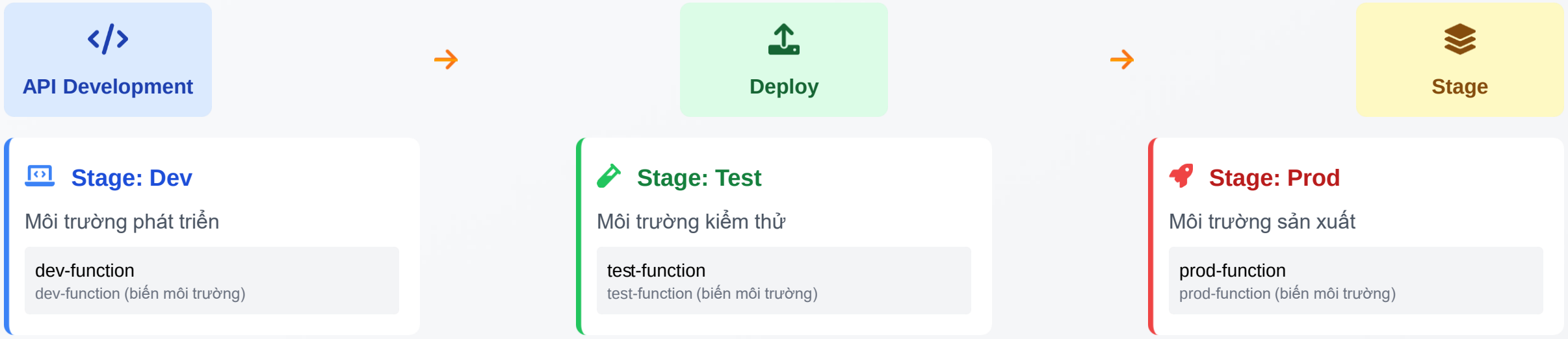
Custom Domain & HTTPS

Hỗ trợ cấu hình tên miền tùy chỉnh và sử dụng chứng chỉ SSL/TLS (HTTPS) để mã hóa dữ liệu truyền tải, đảm bảo an toàn cho các giao tiếp giữa client và API Gateway.

Deployment Stages & Biến môi trường

AWS API Gateway cho phép quản lý các phiên bản API khác nhau thông qua các Deployment Stages. Mỗi stage có thể đại diện cho một môi trường phát triển (Dev), kiểm thử (Test) hoặc sản xuất (Prod), giúp quản lý vòng đời của API một cách hiệu quả.

Đường ống CI/CD với Deployment Stages



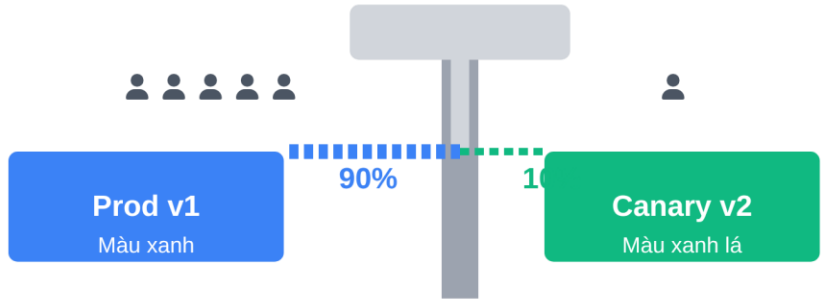
Biến môi trường (Stage Variables)

Stage Variables được sử dụng để trỏ đến các tài nguyên backend tương ứng, ví dụ như các phiên bản khác nhau của hàm Lambda. Điều này giúp dễ dàng chuyển đổi giữa các môi trường mà không cần thay đổi cấu hình API.

`${stageVariables.lambdaAlias}` → Được thay thế bởi giá trị biến môi trường khi triển khai

Blue/Green Deployment với Canary

Canary Deployment là một chiến lược triển khai cho phép giới thiệu một phiên bản mới của API cho một phần nhỏ người dùng trước khi triển khai rộng rãi, giúp giảm thiểu rủi ro bằng cách giám sát hiệu suất và lỗi của phiên bản mới trong môi trường thực tế.



Phân chia lưu lượng

- 90% lưu lượng truy cập được chuyển đến phiên bản ổn định (Prod v1 - Màu xanh)
- 10% lưu lượng truy cập được chuyển hướng đến phiên bản mới (Canary v2 - Màu xanh lá)

Quá trình triển khai

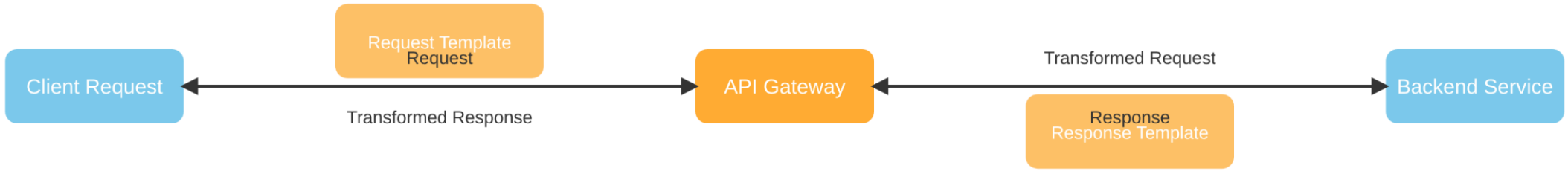
- Cả hai phiên bản đều được triển khai và giám sát riêng biệt
- Nếu phiên bản Canary hoạt động tốt, lưu lượng truy cập có thể dần dần được chuyển hoàn toàn sang phiên bản mới
- Giảm thiểu rủi ro và ảnh hưởng đến người dùng cuối

Lợi ích của Canary Deployment

- Giảm thiểu rủi ro khi triển khai phiên bản mới
- Theo dõi hiệu suất thực tế trong môi trường production
- Dễ dàng rollback nếu phát hiện vấn đề

Mapping Templates & Transformations

Mapping Templates trong API Gateway cho phép biến đổi cấu trúc của request từ client trước khi gửi đến backend và response từ backend trước khi trả về client. Điều này rất hữu ích để đảm bảo tính tương thích giữa các hệ thống và tùy chỉnh dữ liệu.



Velocity Template Language

Ngôn ngữ template mạnh mẽ được sử dụng để định nghĩa các transformation rules.

```
## Ví dụ đơn giản:
${method} == "GET"
```

Ứng dụng thực tế

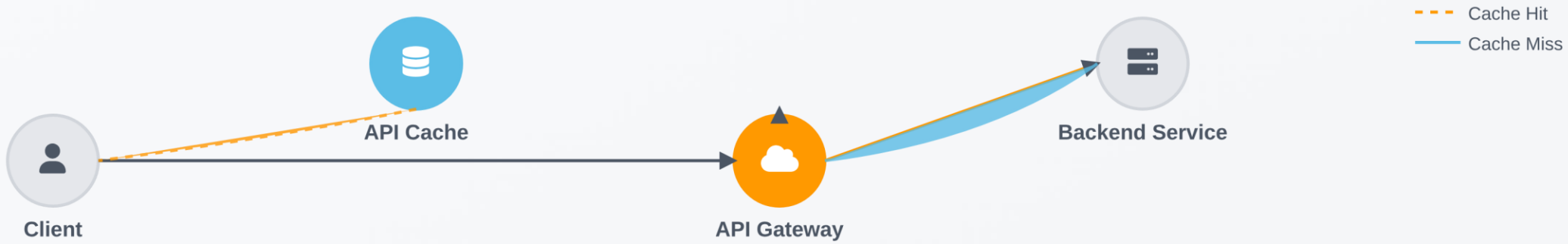
- Thêm/Xóa/Sửa headers và query string parameters
- Chuyển đổi định dạng (XML ↔ JSON)
- Lọc dữ liệu nhạy cảm trước khi trả về client
- Ánh xạ (map) các trường dữ liệu

Hạn chế:

Mapping Templates chỉ khả dụng với REST API và integration type AWS/AWS_PROXY (không dùng được với HTTP_PROXY).

Caching API Responses

API Gateway cung cấp khả năng caching các phản hồi API để giảm tải cho backend và cải thiện độ trễ cho người dùng. Khi caching được kích hoạt, API Gateway sẽ lưu trữ các phản hồi từ backend và trả về chúng trực tiếp cho các request tiếp theo mà không cần gọi lại backend.



Cải thiện độ trễ

Trả về phản hồi trực tiếp từ cache, giảm thiểu thời gian chờ đợi cho người dùng.

Giảm tải cho backend

Số lượng request đến backend được giảm thiểu, giúp tối ưu hóa tài nguyên và chi phí.

Cache Invalidation

Khả năng xóa các mục đã lưu trong cache, thường được thực hiện thủ công hoặc theo cấu hình.

Kiểm soát truy cập với Usage Plans

API Gateway cung cấp tính năng Usage Plans và API Keys để kiểm soát và quản lý việc sử dụng API của bạn, giúp bạn định nghĩa các mức truy cập khác nhau cho các đối tượng người dùng hoặc ứng dụng khác nhau.

API Key Customer A

Được sử dụng để xác định và theo dõi các yêu cầu từ Customer A

API Key Customer B

Được sử dụng để xác định và theo dõi các yêu cầu từ Customer B

API Key Customer C

Được sử dụng để xác định và theo dõi các yêu cầu từ Customer C

Usage Plan

Throttling Limit

Giới hạn số lượng yêu cầu mà một khách hàng có thể gửi đến API trong một khoảng thời gian nhất định (ví dụ: 100 request/giây)

Quota

Giới hạn tổng số yêu cầu mà một khách hàng có thể gửi trong một khoảng thời gian dài hơn (ví dụ: 10.000 request/ngày)

Lưu ý: Usage Plans được liên kết với một hoặc nhiều API Stage (ví dụ: dev, test, prod). Khi một yêu cầu đến API Gateway với một API Key, hệ thống sẽ kiểm tra Usage Plan tương ứng để đảm bảo yêu cầu đó tuân thủ các giới hạn đã đặt ra.

API Gateway

Kiểm tra Usage Plan tương ứng để đảm bảo yêu cầu tuân thủ các giới hạn đã đặt ra

Logging, Tracing & Metrics

API Gateway tích hợp chặt chẽ với các dịch vụ giám sát của AWS như CloudWatch và X-Ray để cung cấp tầm nhìn đầy đủ về hiệu suất và lỗi của API.

Logs

Ghi lại các sự kiện và lỗi của API vào CloudWatch Logs.

- Lưu trữ nhật ký API
- Log các request và response
- Phân tích nhật ký để khắc phục sự cố

Traces

Theo dõi các request xuyên suốt các dịch vụ backend bằng AWS X-Ray.

- Visual hóa luồng request
- Phân tích chuỗi gọi API

Metrics

Thu thập các chỉ số hiệu suất như số lượng request, độ trễ, và lỗi vào CloudWatch Metrics.

- Monitor các chỉ số hiệu suất
- Tạo alarms để cảnh báo
- Visual hóa dữ liệu theo thời gian

Luồng giám sát API Gateway

"Tầm nhìn đầy đủ về hiệu suất và lỗi của API"

Errors & Throttling

Các loại lỗi phổ biến

Lỗi 4XX (Client Error)

Lỗi xảy ra do yêu cầu không hợp lệ từ client.

- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 404 Not Found

Lỗi 5XX (Server Error)

Lỗi xảy ra do vấn đề từ phía server.

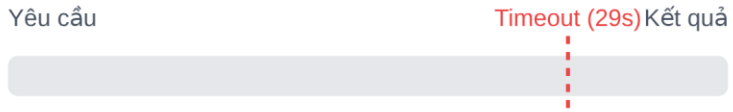
- 500 Internal Server Error
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout

Timeout & Giới hạn tài nguyên

Timeout trong API Gateway

29

29 giây - Giới hạn thời gian chờ mặc định cho một yêu cầu API trước khi API Gateway kết thúc kết nối.



Cách xử lý hiệu quả

- Thiết lập retry logic cho client
- Cấu hình Dead Letter Queue (DLQ) cho Lambda
- Sử dụng CloudWatch Metrics để theo dõi lỗi
- Thiết lập alarm cho các loại lỗi phổ biến

CORS là gì?

Cross-Origin Resource Sharing (CORS) là một cơ chế bảo mật cho phép các ứng dụng web truy cập tài nguyên từ nhiều origin (domain) khác nhau. Trong API Gateway, CORS giúp kiểm soát việc truy cập các API từ các domain khác nhau một cách an toàn.

CORS là gì?

- Là một tiêu chuẩn cho phép truy cập tài nguyên từ nhiều origin khác nhau
- API Gateway hỗ trợ cấu hình CORS thông qua header Access-Control-Allow-Origin
- Cho phép hoặc chặn truy cập từ các domain cụ thể

Cấu hình CORS

Cách cấu hình header Access-Control-Allow-Origin:

```
Access-Control-Allow-Origin: https://yourdomain.com
Access-Control-Allow-Methods: GET, POST, PUT, DELETE
Access-Control-Allow-Headers: Content-Type, Authorization
```

Luồng hoạt động CORS

Security – Tóm tắt

API Gateway cung cấp ba phương thức chính để bảo mật API, mỗi phương thức có những ưu điểm và trường hợp sử dụng riêng biệt.

IAM

- Tốt cho user/role trong AWS
- Sử dụng Sig v4
- Bảo mật API nội bộ
- Truy cập từ các dịch vụ AWS khác

Trường hợp sử dụng điển hình: Bảo vệ API nội bộ, truy cập từ các dịch vụ AWS khác hoặc người dùng AWS có vai trò cụ thể.

Cognito

- Quản lý user bên ngoài (Mobile)
- Không cần viết mã
- Xác thực người dùng ứng dụng
- Tích hợp với nhà cung cấp danh tính bên thứ ba

Trường hợp sử dụng điển hình: Xác thực người dùng ứng dụng di động/web, tích hợp với các nhà cung cấp danh tính bên thứ ba.

Lambda Authorizer

- Linh hoạt nhất
- Hỗ trợ token tùy chỉnh (JWT)
- Có cache
- Xác thực tùy chỉnh

Trường hợp sử dụng điển hình: Xác thực tùy chỉnh, tích hợp với hệ thống xác thực hiện có, xử lý logic ủy quyền phức tạp.

Tóm tắt: Sức mạnh của API Gateway

Cánh cổng thống nhất

Hợp nhất các điểm cuối API khác nhau, đơn giản hóa việc quản lý và truy cập cho các nhà phát triển và người dùng cuối.

Bảo mật, mở rộng và hiệu suất cao

Với các tính năng bảo mật tích hợp, khả năng tự động mở rộng và hỗ trợ caching, API Gateway đảm bảo các API của bạn luôn hoạt động an toàn và nhanh chóng.

Giảm thiểu chi phí vận hành

Là một dịch vụ được quản lý hoàn toàn bởi AWS, API Gateway giúp bạn loại bỏ gánh nặng quản lý cơ sở hạ tầng, cho phép tập trung vào việc phát triển tính năng cốt lõi.

Công cụ thiết yếu cho Serverless

API Gateway là cầu nối lý tưởng giữa các ứng dụng client và các hàm Lambda hoặc các microservices, tạo điều kiện thuận lợi cho việc xây dựng các hệ thống phân tán hiệu quả.

Amazon API Gateway là một cánh cổng mạnh mẽ và linh hoạt, là thành phần không thể thiếu trong việc xây dựng các ứng dụng hiện đại, đặc biệt là trong kiến trúc Serverless và Microservices.