

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Đề tài 05: Triển khai API gateway và security

Nhóm 08

Nguyễn Văn Khang Kim - 21520314

Nguyễn Vũ Anh Duy - 21520211

Đinh Minh Thiện - 21522618

CONTENT

01.

Tổng quan về Kong Gateway

02.

Kiến trúc

03.

Tính năng

04.

Demo

TỔNG QUAN VỀ KONG GATEWAY

01

Tổng quan về Kong Gateway

- Kong Gateway là một API gateway dựa trên cloud.
- Kong Gateway cho phép chúng ta quản lý, cấu hình và định tuyến các yêu cầu tới API của mình.
- Kong Gateway được đặt trước API RESTful và nó có thể mở rộng thông qua các mô-đun và plugin.
- Nó được thiết kế để chạy trên kiến trúc microservices và kiến trúc phân tán, bao gồm việc triển khai trên hybrid-cloud và multi-cloud .

Ưu điểm của Kong Gateway

- Có khả năng mở rộng theo chiều ngang để gia tăng hiệu suất.
- Có khả năng mở rộng tính năng
- Triển khai và cấu hình dễ dàng đối với hệ thống microservices vừa và nhỏ.
- Tài liệu hỗ trợ dễ hiểu.
- Khả năng tương thích tốt với các hệ thống hiện đại.
- Hỗ trợ nhiều ngôn ngữ lập trình.

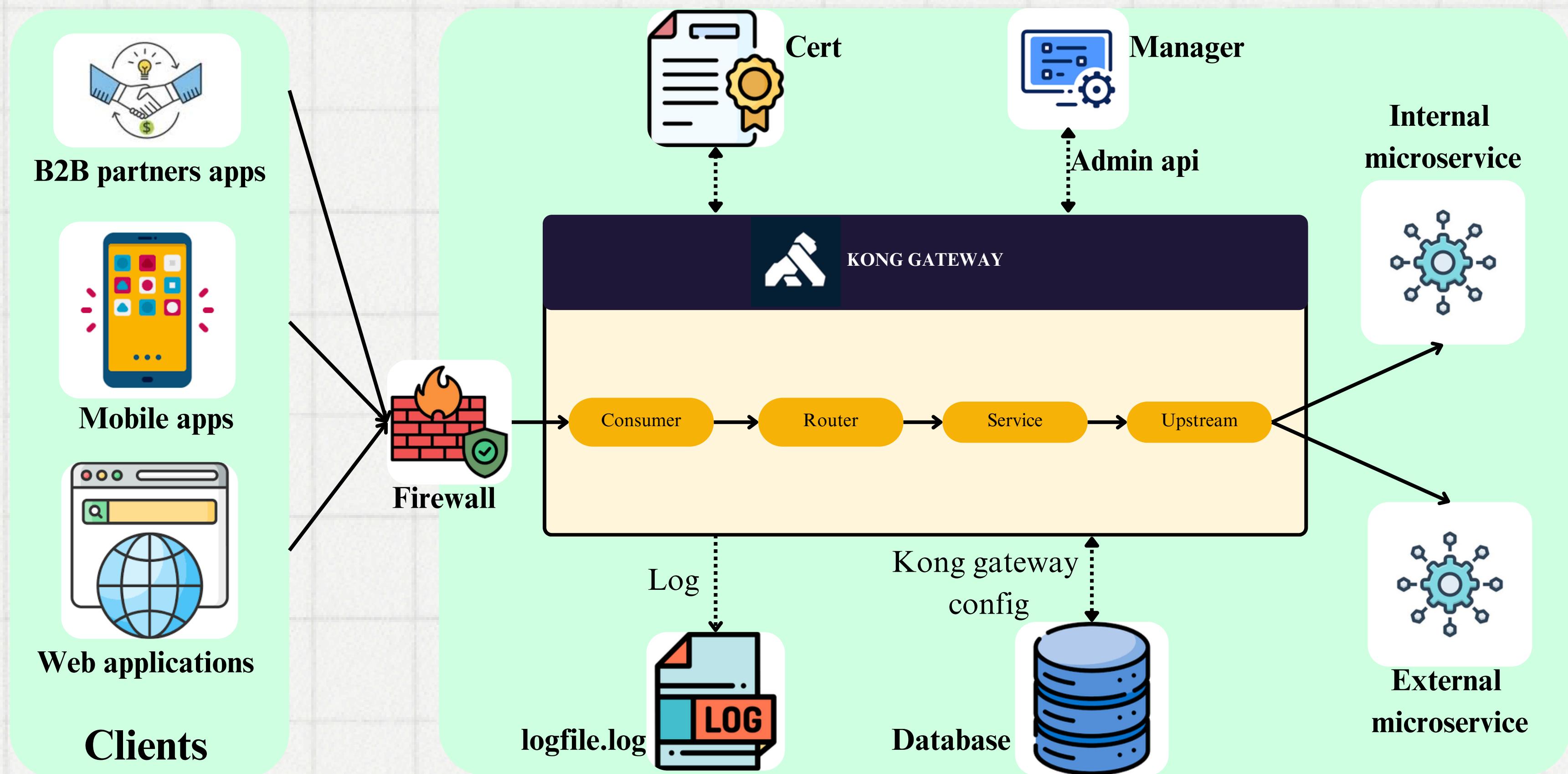
Nhược điểm của Kong Gateway

- Tiêu tốn tài nguyên hệ thống.
- Chi phí hạ tầng và phiên bản cho doanh nghiệp tối kén.
- Có thể gây độ trễ khi phải xử lý nhiều yêu cầu cùng lúc.
- Đối với các hệ thống microservices lớn, việc cấu hình sẽ trở nên khó khăn và tốn nhiều công sức.

02

KIẾN TRÚC

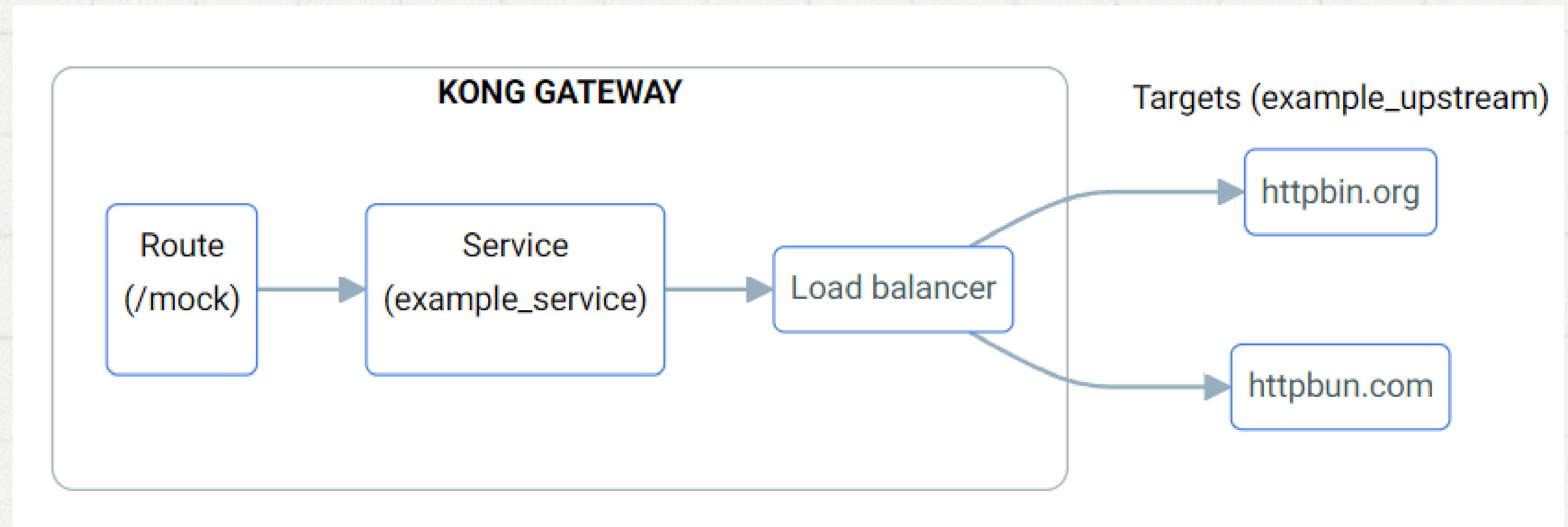
Kiến trúc



- Consumer: Đại diện cho một người dùng cuối của Kong, nó có thể kiểm soát ai có thể truy cập vào API.
- Service: Trong Kong, một service đại diện cho một upstream application. Các service có thể có mối quan hệ one-to-many với các upstream application.
- Route: Là một đường dẫn đến tài nguyên (còn được gọi là endpoint hay url) của upstream application. Các tuyến sẽ được thêm vào các dịch vụ để xác định các ứng dụng để truy cập.
- Upstream: Đại diện cho nhiều upstream application, có thể kiểm tra tình trạng, ngắt mạch và cân bằng tải đến nhiều mục tiêu.

Load balancing

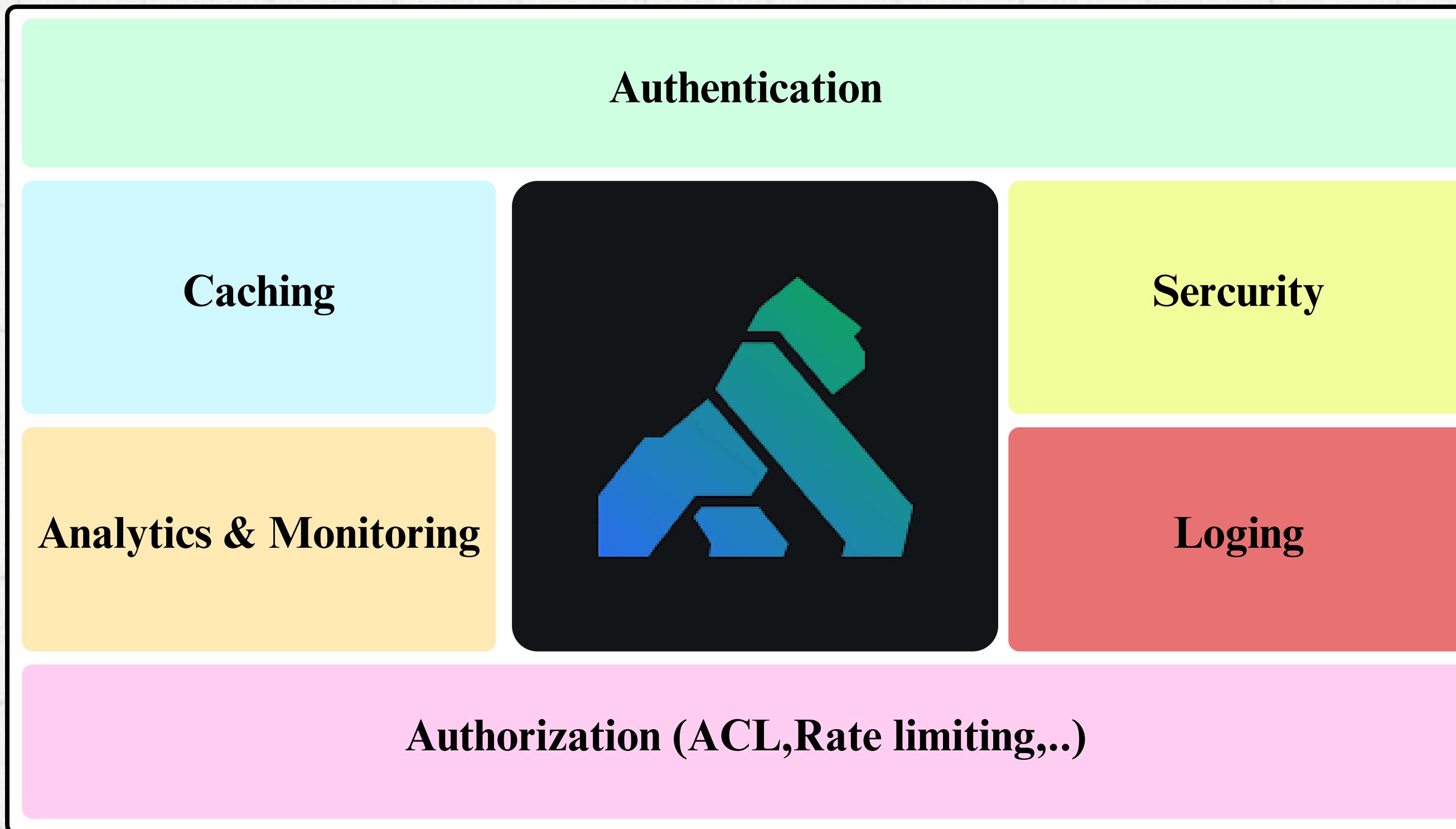
- Một service bây giờ sẽ không còn trỏ đến một upstream application cụ thể nữa mà trỏ đến một upstream.
- Upstream sẽ trỏ đến hai máy chủ chạy cùng một dịch vụ. Và nó có tác dụng cân bằng tải các yêu cầu đến dịch vụ này.



03

TÍNH NĂNG

Tính năng



Tính năng

- Authentication (xác thực) là quá trình xác minh danh tính của người dùng hoặc ứng dụng gửi yêu cầu đến. Nó đảm bảo rằng yêu cầu đến từ một nguồn hợp lệ.
- Các plugin hỗ trợ xác thực.
 - Key Authentication
 - Basic Authentication
 - JWT (JSON Web Token)
 - OAuth2 Authentication
 - HMAC Authentication
 - LDAP Authentication
 - Session

Tính năng

- Authorization (phân quyền) là quá trình kiểm tra và xác định những hành động hoặc tài nguyên nào mà một người dùng hoặc ứng dụng được phép làm và truy cập sau khi đã được xác thực.
- Các plugin hỗ trợ phân quyền.
 - ACL: Kiểm soát các nhóm người dùng có thể dùng dịch vụ.
 - CORS(Cross-Origin Resource Sharing): Cho phép tính năng chia sẻ tài nguyên gốc cho một dịch vụ hoặc một tuyến.
 - Rate Limiting và Response Rate Limiting: Giới hạn số lượng yêu cầu mà người dùng có thể gửi trong một khoảng thời gian nhất định.

Tính năng

- Caching (bộ nhớ đệm) là một tính năng quan trọng giúp cải thiện hiệu suất và giảm tải cho các dịch vụ backend bằng cách lưu các yêu cầu gần nhất vào bộ nhớ đệm và cung cấp phản hồi trực tiếp.
- Các plugin hỗ trợ việc caching.
 - Proxy Caching (miễn phí).
 - Proxy Caching advance (tiền).
 - GraphQL Proxy Caching Advanced (tiền).

Tính năng

- Giám sát và phân tích (monitoring and analytics) trong Kong Gateway là một phần quan trọng để đảm bảo rằng hệ thống hoạt động hiệu quả và hữu ích về bảo mật.
- Các plugin hỗ trợ việc giám sát và phân tích.
 - Datadog
 - OpenTelemetry
 - Prometheus
 - StatsD
 - Zipkin

Tính năng

- Trong Kong Gateway, logging (ghi log) là một chức năng quan trọng để theo dõi hoạt động của hệ thống, giám sát các yêu cầu API, và phát hiện các sự cố.
- Có 3 loại log trong Kong Gateway:
 - Access logs ghi lại tất cả các yêu cầu đến một máy chủ hoặc ứng dụng.
 - Error logs ghi lại các lỗi và cảnh báo phát sinh trong quá trình hoạt động của máy chủ hoặc ứng dụng.
 - Audit logs ghi lại các sự kiện quan trọng liên quan đến bảo mật. Bao gồm các hành động của người dùng như đăng nhập, đăng xuất, thay đổi cấu hình, truy cập dữ liệu nhạy cảm, và các hành động quản trị khác.
- Các plugin:

◦ File log	◦ Syslog	◦ Udp log
◦ Http log	◦ Logly	◦ Tcp log

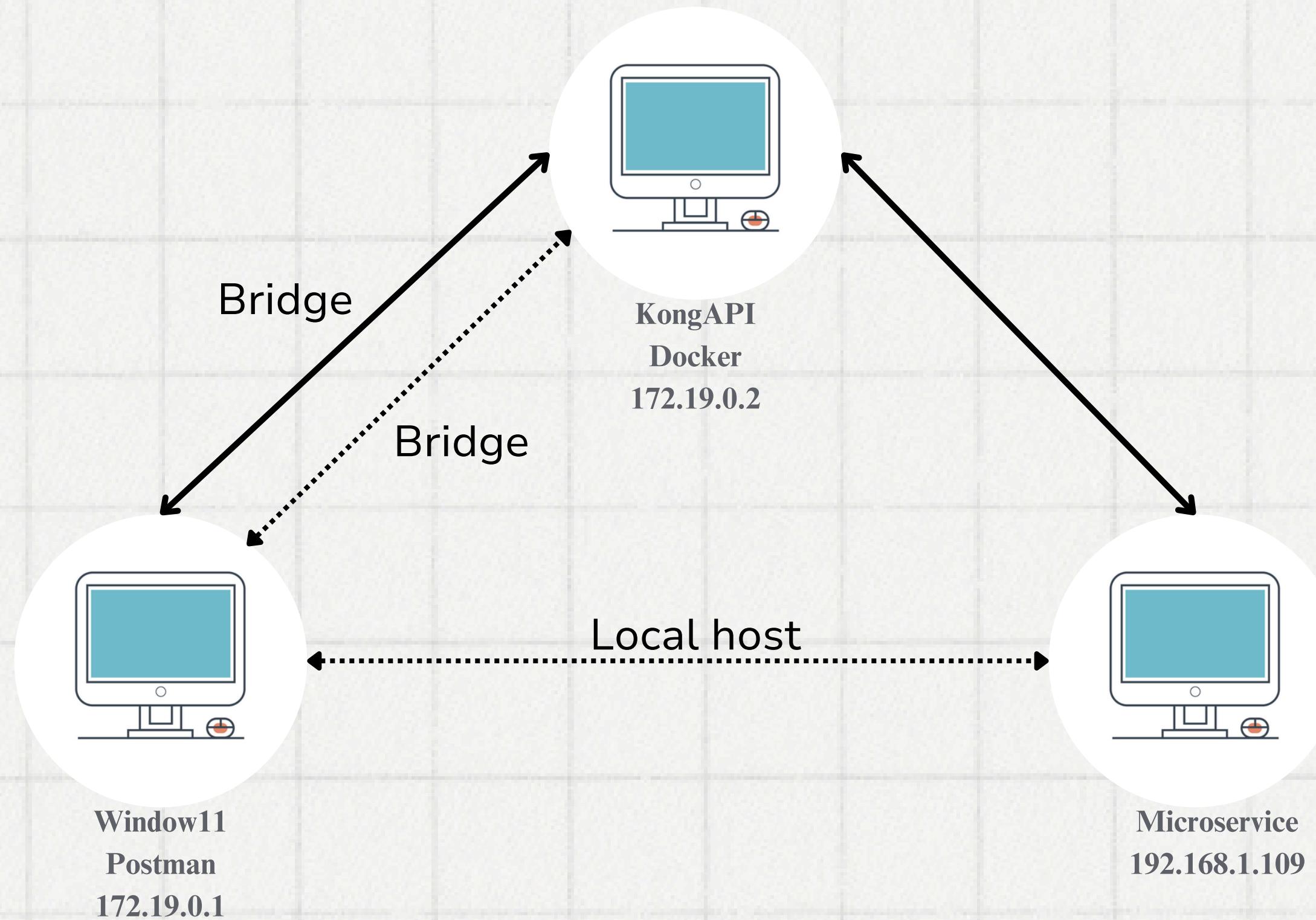
Tính năng

- Security: Kong Gateway có thể giúp tăng yếu tố bảo mật bằng các plugin sau đây:
 - ACME plugin: Cho phép Kong gateway nhận các chứng chỉ miễn phí từ Let's Encrypt hoặc bất kỳ dịch vụ ACMEv2 nào khác.
 - Bot detection plugin: Dùng plugin này để tự động phát hiện Bot, bảo vệ dịch vụ hoặc tuyến khỏi các bot phổ biến nhất, Tạo danh sách cho phép và từ chối cho khách hàng tùy chỉnh.
 - IP restriction: Hạn chế quyền truy cập vào một dịch vụ hoặc tuyến bằng cách cho phép hoặc từ chối địa chỉ IP.

04

DEMO

Deployment



Demo

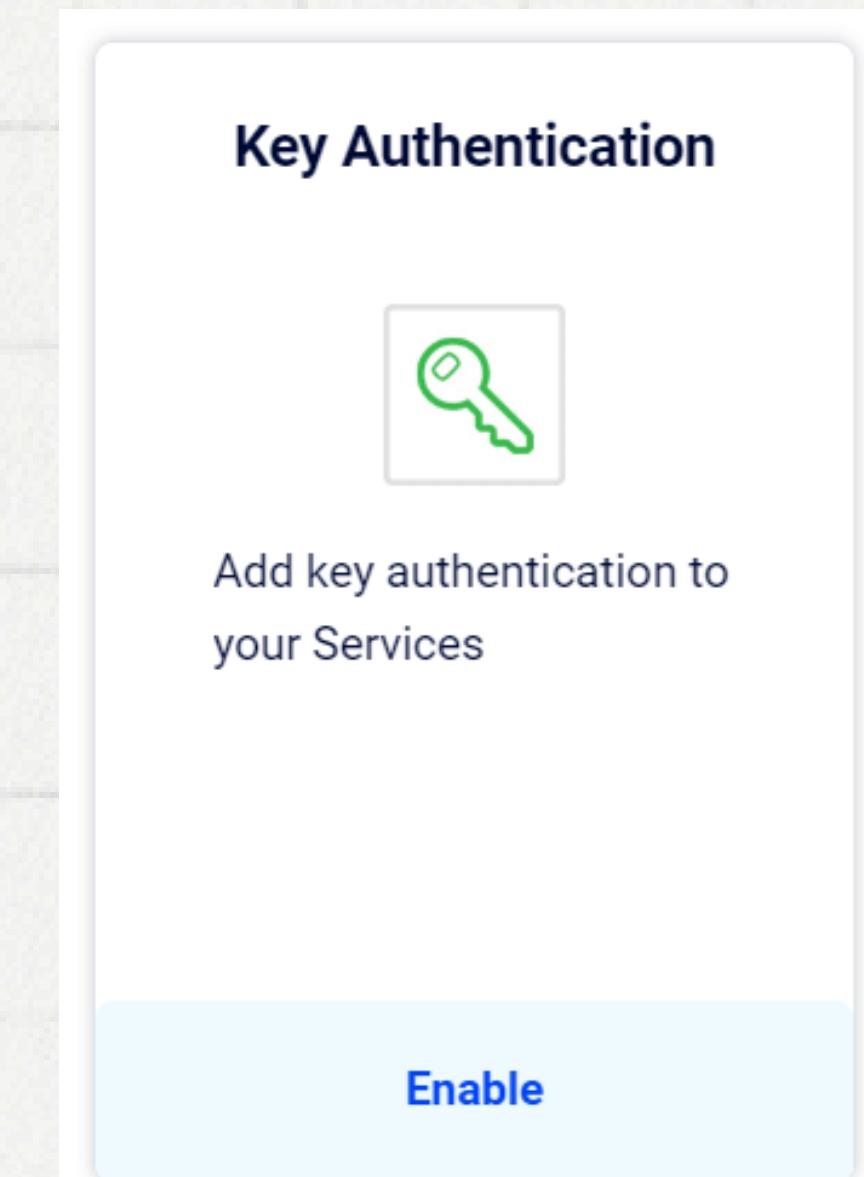
- Trước khi bắt đầu thêm các plugin để hỗ trợ tính năng, ta thêm các dịch vụ, tuyến và người dùng.

The image displays three separate views from a service management application:

- Gateway Services View:** Shows a table of services with columns: Name, Protocol, Host, Port, Path, Enabled, and Tags. Three services are listed: profile (http://10.45.134.22:8081/uit/student/profile), account (http://10.45.134.22:8081/uit), and courses (http://10.45.134.22:8081/uit). All services are currently enabled.
- Routes View:** Shows a table of routes with columns: Name, Protocols, Hosts, Methods, Paths, and Tags. Three routes are listed: account (HTTP, HTTPS) pointing to /student/account, profile (HTTP, HTTPS) pointing to /uit/student/profile, and courses (HTTP, HTTPS) pointing to /courses.
- Consumers View:** Shows a table of consumers with columns: Username, Custom ID, and Tags. Three consumers are listed: Minh Thien, Khang Kim, and Anh Duy. Each consumer has a corresponding custom ID and no explicit tags assigned.

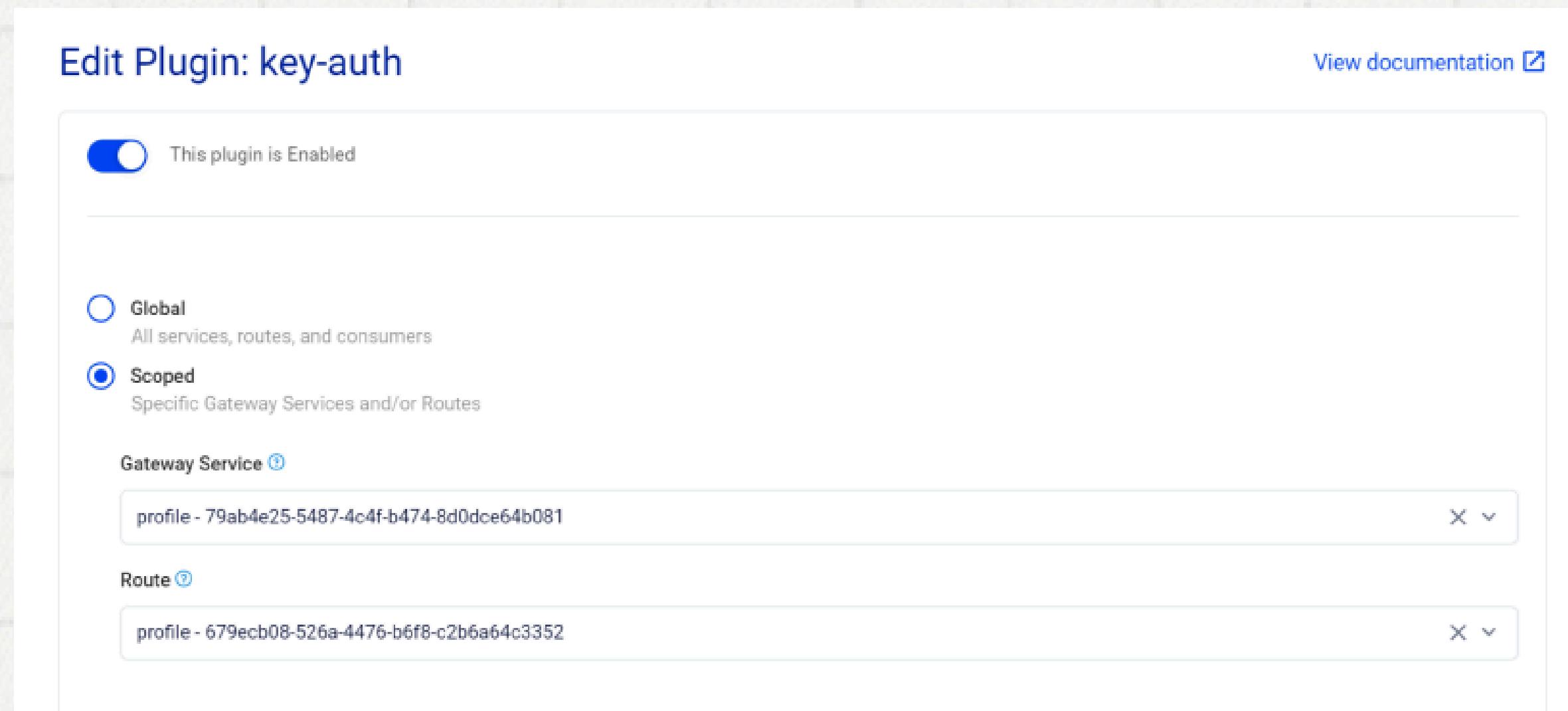
Demo: Authentication

- Khi nhận yêu cầu từ khách hàng, điều cơ bản đầu tiên phải làm đó là kiểm tra xem đây có phải là một người dùng hay không.
- Key authentication: Plugin này thêm 1 khóa xác thực cho dịch vụ của mình và người dùng được yêu cầu khóa này để truy cập vào dịch vụ.



Demo: Authentication

- Đầu tiên chúng ta sẽ chọn chọn Scoped để áp dụng cho một dịch vụ cụ thể là profile, chọn service và route tương ứng.



Demo: Authentication

- Tiếp theo ta sẽ đặt tên là keyauth-profile, Key name là apikey và bấm save.

The screenshot shows two parts of a user interface for managing authentication profiles.

Top Panel (Configuration Dialog):

- Instance Name:** keyauth-profile
- Key Names:** apikey
- Buttons:** + Add, Run On Preflight (unchecked), View Configuration, Cancel, Save

Bottom Panel (List View):

Name	Applied To	Status	Ordering	Tags
keyauth-profile Key Authentication	Route Service	Enabled <input checked="" type="checkbox"/>	Static	⋮

Demo: Authentication

- Trong phần consumer chúng ta vào Credentials và tạo khóa xác thực.
Nhập Key là khangkim và bấm save.
- Lưu ý khóa xác thực của người dùng là duy nhất. Và không có người dùng nào có khóa tương tự.

The screenshot shows a user interface for managing consumer credentials. At the top, the path 'Consumers > Khang Kim >' is visible, followed by the name 'Khang Kim'. On the right, there are 'Back' and 'Consumer actions' buttons. A sidebar on the left lists 'Configuration', 'Groups', 'Credentials' (which is highlighted in blue), and 'Plugins'. The main content area is titled 'Credentials' and 'Key Authentication'. A blue button labeled '+ New Key Auth Credential' is located in the bottom right of this section. Below this, a modal window titled 'Create Key Auth Credential' is open. It has a 'Key' input field containing 'khangkim'. A note below the input says, 'You can optionally set your own unique key to authenticate the client. If missing, it will be generated for you.' The entire interface is set against a light gray background with a subtle grid pattern.

Demo: Authentication

The screenshot shows a POST request to `http://localhost:8000/uit/student/profile`. The Headers tab is selected, showing a `Connection` header and an `apiKey` header with value `khangkim`. The response status is `401 Unauthorized`, and the response body is:

```
1 {  
2   "message": "No API key found in request",  
3   "request_id": "60470d43892ec00be0805c0075575d2a"  
4 }
```

- Thêm apikey vào phần header và thử lại.

The screenshot shows a POST request to `http://localhost:8000/uit/student/profile`. The Headers tab is selected, showing both `Connection` and `apiKey` headers. The response status is `200 OK`, and the response body is:

```
1 {  
2   "student1": {  
3     "id": "21520314",  
4     "name": "Nguyen Van Khang Kim",  
5     "age": "21",  
6     "gender": "Male",  
7     "class": "ANTN_2021",  
8     "email": "khangkim10012003@gmail.com",  
9     "phone": "0000000000",  
10    "university": "UIT",  
11    "address": {  
12      "street": "123 Nguyen Trai",  
13      "city": "Ho Chi Minh"  
14    }  
15  }  
16 }
```

- Thủ truy cập khi không có API key.

Demo: Authorization

- Sau khi đã xác thực, điều tiếp theo chúng ta cần làm là xem xét các hoạt động hay tài nguyên nào mà người dùng có thể làm và truy cập.
- Các plugin demo:
 - Rate limiting
 - CORS(Cross-Origin Resource Sharing)

Rate Limiting



Rate-limit how many HTTP requests a developer can make

Enable

CORS

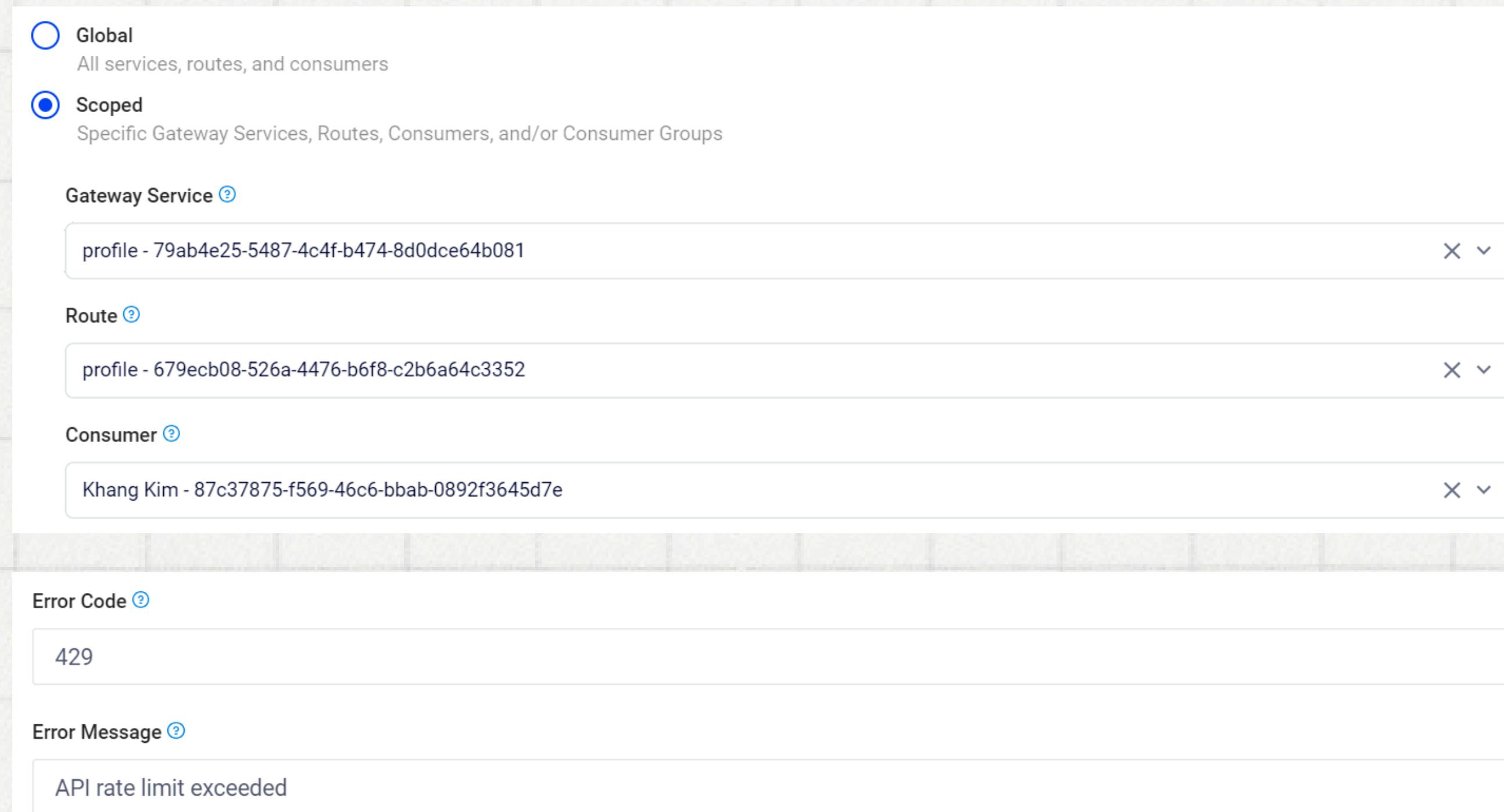


Allow developers to make requests from the browser

Enable

Demo: Authorization với Rate limiting

- Trong phần plugin , chọn Rate Limiting. Đầu tiên chọn Scoped, chọn service, route và consumer là profile.
- Chọn Error Code là 429, Error Message có thể tùy chỉnh.



Demo: Authorization với Rate limiting

- Chọn Limit By là consumer, Minute là 5, các thứ khác giữ nguyên. Bấm save.

Limit By [?](#)

consumer

Minute [?](#)

5

- Sau khi gửi request quá 5 lần trong 1 phút thì sẽ hiện mã lỗi và thông báo lỗi.

HTTP Contract testing / Test response

Save [Edit](#) [Delete](#)

GET http://localhost:8000/uit/student/profile Send

Params Authorization Headers (7) Body Pre-request Script Tests • Settings Cookies

Connection keep-alive

apiKey khangkim

Body Cookies Headers (13) Test Results (1/5) Status: 429 Too Many Requests Time: 9 ms Size: 520 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "message": "API rate limit exceeded",  
3   "request_id": "647ac24c960a317614eb250c40d70d7e"  
4 }
```

Demo: Authorization với CORS

- Chọn Scoped và lần lượt chọn service và route là profile. Đặt tên là cors-profile.
- Thêm một Origins là * (tất cả), giữ nguyên các trường khác và bấm save.

The screenshot shows a configuration interface for setting up CORS (Cross-Origin Resource Sharing) authorization. The interface includes the following sections:

- Authorization Type:** A radio button group with "Global" and "Scoped". "Scoped" is selected, indicating it applies to specific Gateway Services and/or Routes.
- Gateway Service:** A dropdown menu containing "profile - 79ab4e25-5487-4c4f-b474-8d0dce64b081".
- Route:** A dropdown menu containing "profile - 679ecb08-526a-4476-b6f8-c2b6a64c3352".
- Instance Name:** A text input field containing "cors-profile".
- Origins:** A text input field containing "*" (all origins).
- Add:** A blue "+ Add" button located at the bottom left of the Origins section.

Demo: Authorization với CORS

HTTP <http://localhost:8000/uit/student/profile>

Save

GET <http://localhost:8000/uit/student/profile> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Key	Value	Description
User-Agent	PostmanRuntime/7.37.3	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
apikey	khangkim	
Origin	http://localhost:8000/courses	
Access-Control-Request-Method	GET	

- Trong trường header của phản hồi, ta có thể thấy origin được cho phép truy cập là courses.

Status: 200 OK Time: 24 ms Size: 1.67 KB

Body Cookies Headers (13) Test Results

Header	Value
Server	Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.0.30
X-Powered-By	PHP/8.0.30
Access-Control-Allow-Origin	http://localhost:8000/courses
vary	Origin
Access-Control-Allow-Credentials	true
X-Kong-Upstream-Latency	9
X-Kong-Proxy-Latency	6
Via	kong/3.6.1.2-enterprise-edition

Find and replace Console Postbot Runner Start Proxy Cc

- Trong header của request, ta thêm một thuộc tính origin bằng http://localhost:8000/courses.

Demo: Authorization với CORS

- Kết quả ta nhận được sẽ là courses thay vì profile.

The screenshot shows the Postman interface with a request to `http://localhost:8000/uit/student/profile`. The `Headers` tab is selected, containing the following entries:

Key	Value	Description
Origin	http://localhost:8000/courses	
Access-Control-Request-Method	GET	

The response body is a JSON object:

```
{  
  "student1": {  
    "id": "21520314",  
    "name": "Nguyen Van Khang Kim",  
    "age": "21",  
    "gender": "Male",  
    "class": "ANTN_2021",  
    "email": "khangkim10012003@gmail.com",  
    "phone": "0000000000",  
    "university": "UIT",  
    "address": {  
      "street": "123 Nguyen Trai",  
      "city": "Ho Chi Minh"  
    }  
  }  
}
```

Demo: Caching

- Để tăng tốc độ xử lý và giảm tải cho các ứng dụng thì ta dùng caching.
- Mình sẽ demo proxy-cache plugin vì nó miễn phí.
- Ta sẽ chọn áp dụng cho tất cả. Chọn thời gian để lưu trữ các phản hồi là 300 giây và nơi lưu là memory.

Edit Plugin: proxy-cache [View documentation ↗](#)

This plugin is Enabled

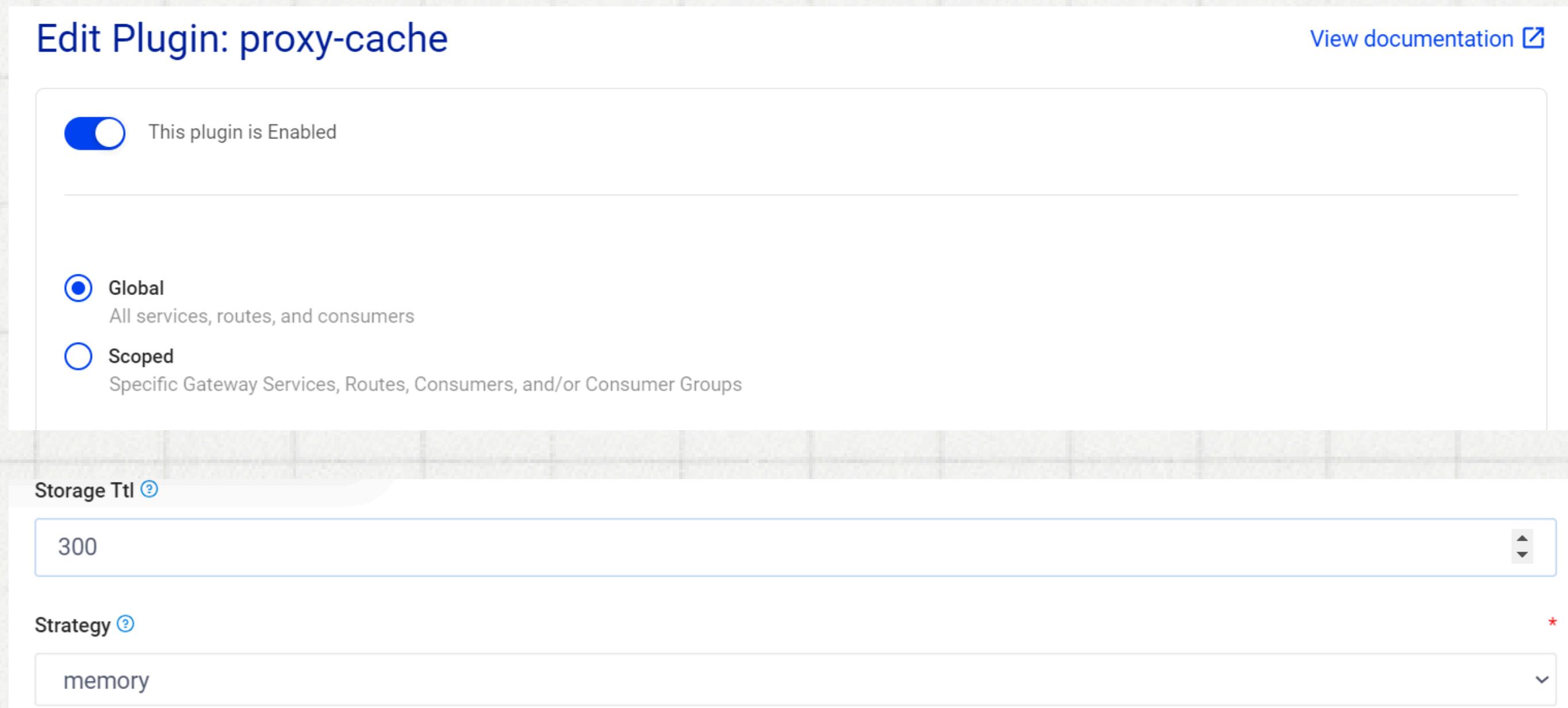
Global
All services, routes, and consumers

Scoped
Specific Gateway Services, Routes, Consumers, and/or Consumer Groups

Storage Ttl ?
300

Strategy ? *

memory



Demo: Caching

- Sau khi lưu nó thì nó đã kích hoạt sẵn. Bây giờ ta gửi một yêu cầu, trong trường header của phản hồi có trường X-Cache-Status là Miss có nghĩ là yêu cầu này được chuyển đến cho các ứng dụng thượng nguồn.

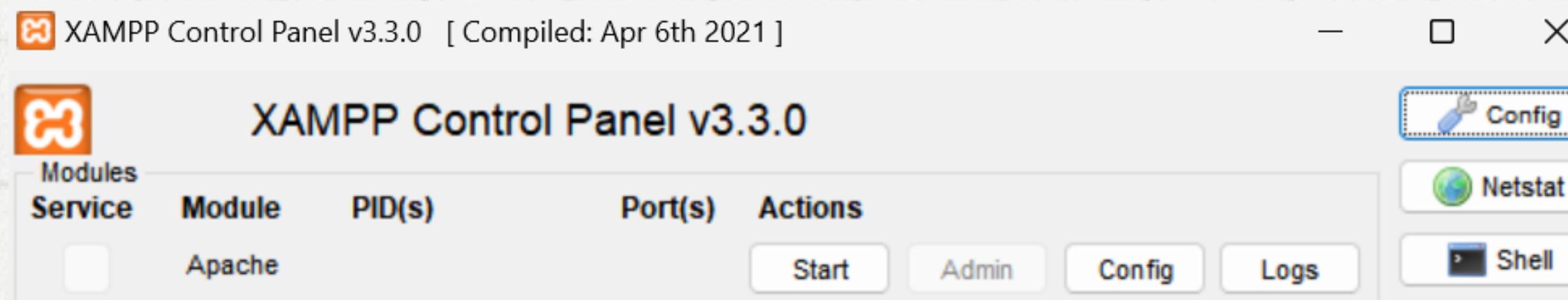
The screenshot shows the Postman interface. At the top, there is a configuration bar for "Proxy Caching" with "Global" selected, "Enabled" set to "Static", and a toggle switch. Below this is a search bar with "GET" and "http://localhost:8000/uit/student/profile". The main area shows a table of request headers. The "Headers" tab is selected, showing the following entries:

Header	Value
Content-Length	1241
Connection	keep-alive
X-Cache-Key	0a998de720bc197348e73b8470688dc49b326c03fc1231c443ffe3a54d98396d
X-Cache-Status	Miss

At the top right of the main area, there is a status summary: "Status: 200 OK Time: 15 ms Size: 1.69 KB" and a "Save as example" button.

Demo: Caching

- Bây giờ ta sẽ tắt các ứng dụng thượng nguồn.



- Sau khi gửi lại request thì nó vẫn trả về kết quả bình thường và X-Cache-status bây giờ là Hit nghĩa là phản hồi này được lấy từ cache.

A screenshot of the Postman application interface. At the top, it shows a "GET" method and the URL "http://localhost:8000/uit/student/profile". On the right, there are "Send" and "Headers" buttons. Below the URL, there are tabs for "Params", "Authorization", "Headers (7)", "Body", "Pre-request Script", "Tests", "Settings", and "Cookies". The "Headers" tab is selected. Under "Headers", there are sections for "Body", "Cookies", and "Headers (13)". The "Headers (13)" section is currently active. It lists three headers: "Connection: keep-alive", "X-Cache-Key: 0a998de720bc197348e73b8470688dc49b326c03fc1231c443ffe3a54d98396d", and "X-Cache-Status: Hit". On the right side of the interface, there are status details: "Status: 200 OK", "Time: 13 ms", "Size: 1.69 KB", and "Save as example".

Demo: Caching

- Sau khi chờ hết thời gian 5 phút. Ta gửi lại yêu cầu và nhận được thông báo lỗi vì đã hết hạn lưu các phản hồi trong cache.

The screenshot shows a POSTMAN interface with the following details:

- Method:** GET
- URL:** http://localhost:8000/uit/student/profile
- Headers:** (7)
- Body:** (Pretty, Raw, Preview, Visualize, JSON selected)
- Responses:** (Cookies, Headers (11), Test Results selected)
- Status:** Status: 502 Bad Gateway, Time: 12.24 s, Size: 569 B
- Message:** "An invalid response was received from the upstream server", "request_id": "b4c6302f093573147b87327fd1ef9a63"

Demo: Monitoring and analysis

- Prometheus plugin: Prometheus là một hệ thống giám sát và cảnh báo mã nguồn mở được thiết kế để thu thập và lưu trữ các số liệu từ các dịch vụ và ứng dụng.

Install Plugin: prometheus

This plugin is Enabled

[View documentation](#)

Global
All services, routes, and consumers

Scoped
Specific Gateway Services, Routes, and/or Consumers

• Đầu tiên chọn áp dụng cho tất cả

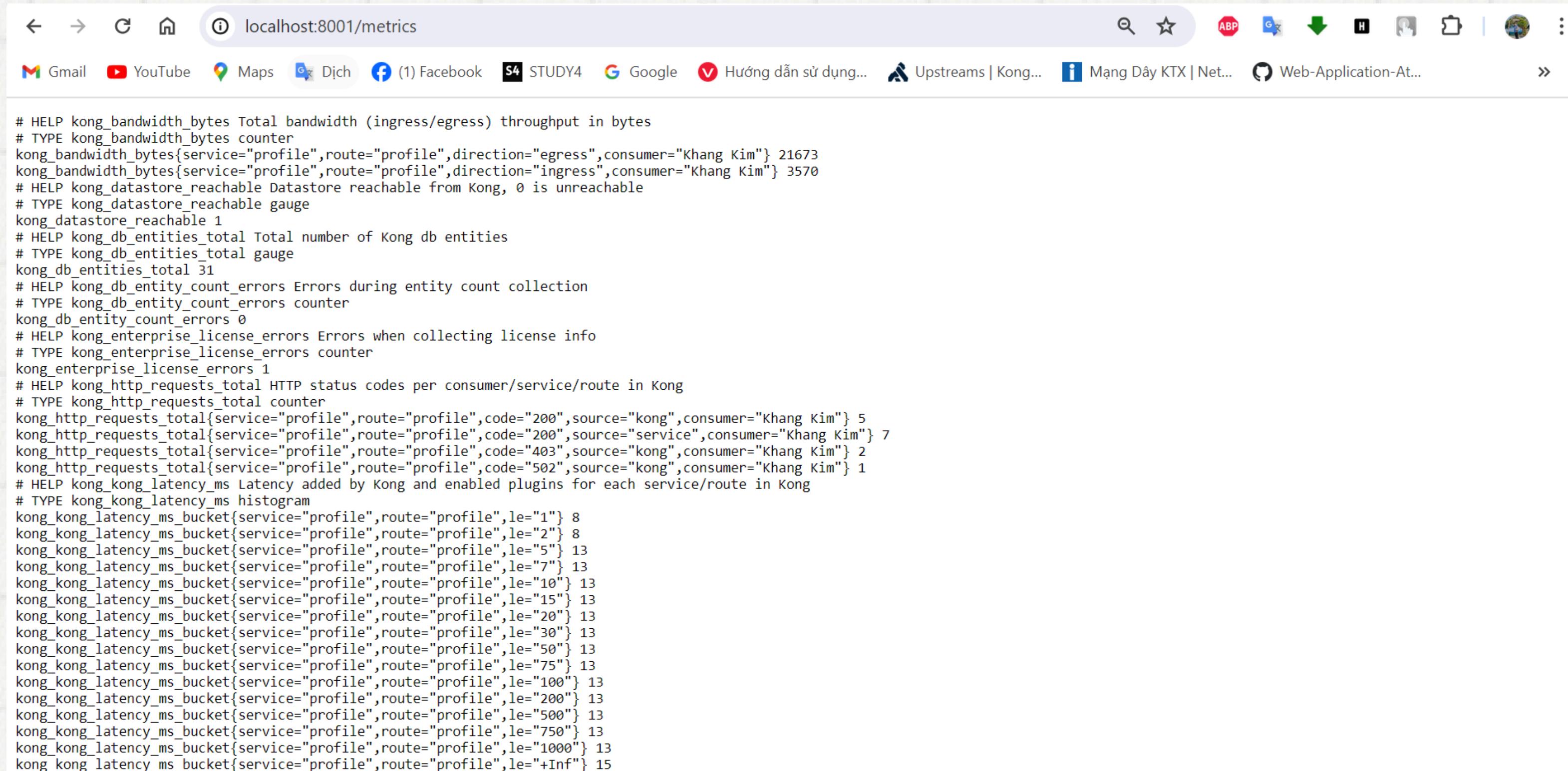
• Chọn thu thập hết tất cả metrics

Bandwidth Metrics ⓘ
 Latency Metrics ⓘ
 Per Consumer ⓘ
 Status Code Metrics ⓘ
 Upstream Health Metrics ⓘ

[View Configuration](#) [Cancel](#) [Save](#)

Demo: Monitoring and analysis

- Các số liệu thu thập được sẽ có sẵn ở <http://localhost:8001/metrics>



The screenshot shows a web browser window with the URL `localhost:8001/metrics` in the address bar. The page content is a large block of text representing system metrics. The metrics are categorized by prefix, such as `kong_bandwidth_bytes`, `kong_db_entities`, `kong_http_requests`, and `kong_kong_latency_ms`. Specific data points include bandwidth usage (e.g., 21673 bytes for egress), database entity counts (e.g., 31 total), and HTTP request counts (e.g., 5 for code 200 from Kong). Latency histograms show distribution across various time buckets.

```
# HELP kong_bandwidth_bytes Total bandwidth (ingress/egress) throughput in bytes
# TYPE kong_bandwidth_bytes counter
kong_bandwidth_bytes{service="profile",route="profile",direction="egress",consumer="Khang Kim"} 21673
kong_bandwidth_bytes{service="profile",route="profile",direction="ingress",consumer="Khang Kim"} 3570
# HELP kong_datastore_reachable Datastore reachable from Kong, 0 is unreachable
# TYPE kong_datastore_reachable gauge
kong_datastore_reachable 1
# HELP kong_db_entities_total Total number of Kong db entities
# TYPE kong_db_entities_total gauge
kong_db_entities_total 31
# HELP kong_db_entity_count_errors Errors during entity count collection
# TYPE kong_db_entity_count_errors counter
kong_db_entity_count_errors 0
# HELP kong_enterprise_license_errors Errors when collecting license info
# TYPE kong_enterprise_license_errors counter
kong_enterprise_license_errors 1
# HELP kong_http_requests_total HTTP status codes per consumer/service/route in Kong
# TYPE kong_http_requests_total counter
kong_http_requests_total{service="profile",route="profile",code="200",source="kong",consumer="Khang Kim"} 5
kong_http_requests_total{service="profile",route="profile",code="200",source="service",consumer="Khang Kim"} 7
kong_http_requests_total{service="profile",route="profile",code="403",source="kong",consumer="Khang Kim"} 2
kong_http_requests_total{service="profile",route="profile",code="502",source="kong",consumer="Khang Kim"} 1
# HELP kong_kong_latency_ms Latency added by Kong and enabled plugins for each service/route in Kong
# TYPE kong_kong_latency_ms histogram
kong_kong_latency_ms_bucket{service="profile",route="profile",le="1"} 8
kong_kong_latency_ms_bucket{service="profile",route="profile",le="2"} 8
kong_kong_latency_ms_bucket{service="profile",route="profile",le="5"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="7"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="10"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="15"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="20"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="30"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="50"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="75"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="100"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="200"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="500"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="750"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="1000"} 13
kong_kong_latency_ms_bucket{service="profile",route="profile",le="+Inf"} 15
```

Demo: Logging

- Để tiện cho việc theo dõi các hoạt động của hệ thống và phát hiện sự cố kịp thời thì ta sẽ ghi log.
- Plugin dùng để ghi log có rất nhiều, ở đây mình sẽ dùng file log plugin để ghi các log vào file.
- Đầu tiên ta sẽ chọn ghi toàn bộ nhật ký và đường dẫn file chứa log.

Edit Plugin: file-log [View documentation ↗](#)

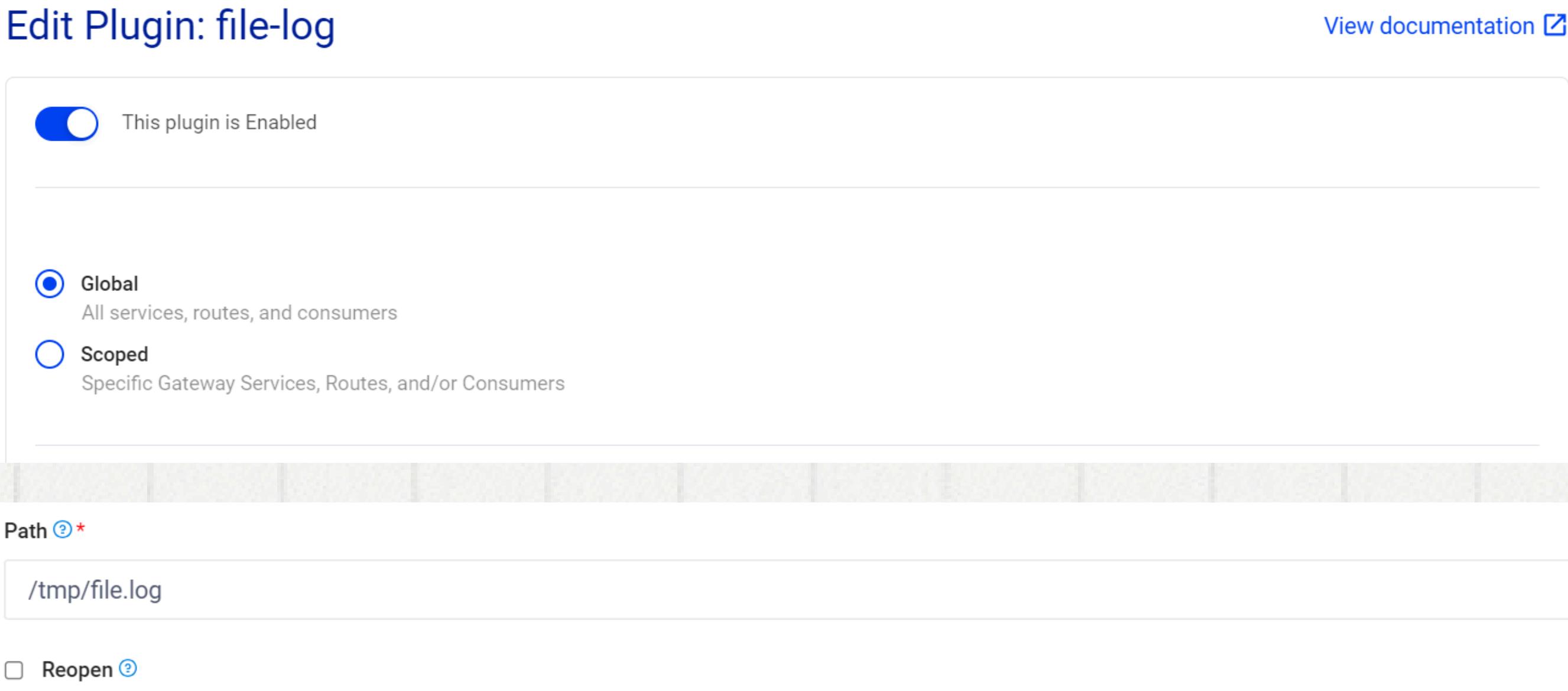
This plugin is Enabled

Global
All services, routes, and consumers

Scoped
Specific Gateway Services, Routes, and/or Consumers

Path ? *

Reopen ?



Demo: Logging

- Gửi request và kiểm tra thì thấy log đã được đẩy về file.

The screenshot illustrates a demonstration of logging. At the top, a Postman request is shown for a GET operation to `http://localhost:8000/uit/student/profile`, returning a `200 OK` status with a JSON response:

```
1 {
2   "student1": {
3     "id": "21520314",
4     "name": "Nguyen Van Khang Kim",
5     "age": "21",
6     "gender": "Male",
7     "class": "ANTN_2021",
8     "email": "khangkim10012003@gmail.com",
9     "phone": "0000000000",
10    "university": "UIT",
11    "address": {
12      "street": "123 Nguyen Trai",
13      "city": "Ho Chi Minh"
14    }
}
```

Below the request, a file browser window shows the directory structure under `/tmp`:

Path	Action	Time	Permissions
<code>tmp</code>	MODIFIED	3 minutes ago	<code>dtrwxrwxrwx</code>
<code>tmp/file.log</code>	ADDED	7.2 kB 2 minutes ago	<code>-rw-r--r--</code>
<code>tmp/usr</code>	MODIFIED	3 months ago	<code>drwxr-xr-x</code>
<code>tmp/var</code>		3 months ago	<code>drwxr-xr-x</code>

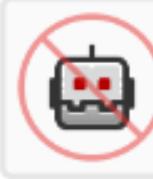
At the bottom, a terminal window displays the contents of `/tmp/file.log`:

```
1 {"route":{"tags":[],"preserve_host":false,"id":"679ecb08-526a-4476-b6f8-c2b6a64c3352","path_handling":"v0","service
2 {"route":{"tags":[],"preserve_host":false,"id":"679ecb08-526a-4476-b6f8-c2b6a64c3352","path_handling":"v0","service
3 {"route":{"tags":[],"preserve_host":false,"id":"679ecb08-526a-4476-b6f8-c2b6a64c3352","path_handling":"v0","service
4
```

Demo: Security

- Để tăng cường mức độ bảo mật thì Kong Gateway có khá là ít plugin free, ở đây mình sẽ demo về bot detection plugin và ip restriction plugin.

Bot Detection



Detect and block bots or custom clients

Enable

IP Restriction



Whitelist or blacklist IPs that can make requests

Enable

Demo: Security với Bot Detection Plugin

- Plugin này sẽ tự động chặn các yêu cầu được cho là của bot dựa vào user-agent dựa trên cơ sở dữ liệu của nó. Ta có thể cho phép hoặc từ chối các user-agent mà chúng ta muốn. Ở đây mình sẽ từ chối một user-agent.

Edit Plugin: bot-detection

[View documentation](#)

This plugin is Enabled

Global
All services, routes, and consumers

Scoped
Specific Gateway Services and/or Routes

Deny [?](#)

Mozilla/5.0 (compatible; MyBot/1.0)

 Bot Detection Global Enabled Static - [...](#)

Demo: Security với Bot Detection Plugin

- Sau khi thay user-agent thành cái bị chặn và gửi yêu cầu thì sẽ nhận được thông báo lỗi.

The screenshot shows a POST request in Postman to `http://localhost:8000/uit/student/profile`. The request method is `GET`. The `Headers` tab is selected, showing the following configuration:

Key	Description
Accept	<code>/*</code>
Accept-Encoding	<code>gzip, deflate, br</code>
Connection	<code>keep-alive</code>
apikey	khangkim
User-Agent	<code>Mozilla/5.0 (compatible; MyBot/1.0)</code>

The `Body` tab shows the response body:

```
{  
  "message": "Forbidden",  
  "request_id": "60fdcd06cccc5b2359db667af65c340f"  
}
```

The status bar at the bottom indicates `Status: 403 Forbidden`, `Time: 11 ms`, and `Size: 386 B`.

Demo: Security với IP Restriction Plugin

- Plugin này chặn các người dùng dựa trên IP.
- Đầu tiên ta sẽ chọn áp dụng cho toàn cầu, đặt tên là ip-restriction.

Edit Plugin: ip-restriction [View documentation](#)

This plugin is Disabled

Global
All services, routes, and consumers

Scoped
Specific Gateway Services, Routes, Consumers, and/or Consumer Groups

Instance Name [?](#)

ip-restriction

Demo: Security với IP Restriction Plugin

- Thêm ip là 172.19.0.1 vào danh sách bị chặn. Chỉnh sửa thông báo lỗi.
Nhập mã trạng thái là 403 và bấm save.

Deny ②

172.19.0.1 Delete

[+ Add](#)

Message ②

IP 172.19.0.1 da bi chan

Status ②

403

[View Configuration](#) Cancel Save

Demo: Security với IP Restriction Plugin

- Khi chưa kích hoạt thì các yêu cầu vẫn được phản hồi bình thường.

The screenshot shows the Postman interface with the following details:

- Header:** Global setting is "Disabled".
- Request:** Method is "GET", URL is "http://localhost:8000/uit/student/profile".
- Body:** Body tab is selected, showing a JSON response:

```
1 {  
2   "student1": {  
3     "id": "21520314",  
4     "name": "Nguyen Van Khang Kim",  
5     "age": "21",  
6     "gender": "Male",  
7     "class": "ANTN_2021",  
8     "email": "khangkim10012003@gmail.com",  
9     "phone": "0000000000",  
10    "university": "UIT",  
11    "address": {  
12      "street": "123 Nguyen Trai",  
13      "city": "Ho Chi Minh"  
14    }  
15  }  
16 }
```
- Response:** Status: 200 OK, Time: 15 ms, Size: 1.59 KB.

Demo: Security với IP Restriction Plugin

- Khi kích hoạt thì ta sẽ nhận được thông báo lỗi và IP này đã bị chặn.

The screenshot shows a Postman request to `http://localhost:8000/uit/student/profile` via GET. The request is successful, returning a JSON response:

```
1 {  
2   "message": "IP 172.19.0.1 da bi chan",  
3   "request_id": "a7c1114f71d1f71e9bbe1353ca3bc239"  
4 }
```

At the top, the 'ip-restriction' plugin configuration is shown as 'Enabled' (blue switch) and 'Static'. The status bar at the bottom indicates a `Status: 403 Forbidden`.

Demo

- Ngoài phần demo của mình thì các bạn có thể tham khảo thêm về cách cài đặt, cấu hình kong API gateway cơ bản, cách triển khai load balancer tại link :

https://www.youtube.com/playlist?list=PLBm4OGt1_S7aKTFWL2nKbyLcu6H2pRi2Q

- Cách dùng plugin kong ACME để nhận chứng chỉ miễn phí tại link:

Tại link: <https://www.youtube.com/watch?v=0gBFm1mw8TU&t=420s>

**Thank you
very much!**