



Kong Labs Training

 @kong

<https://konghq.com>

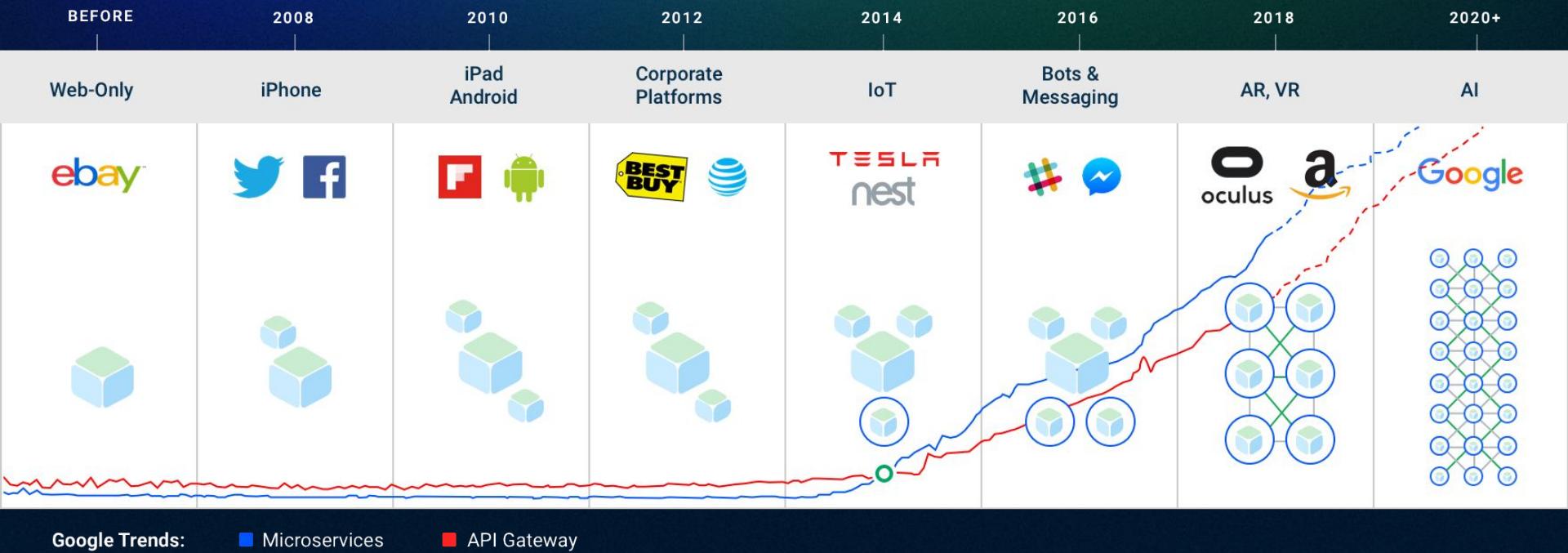
Introduction

Components, Architecture and Deployment

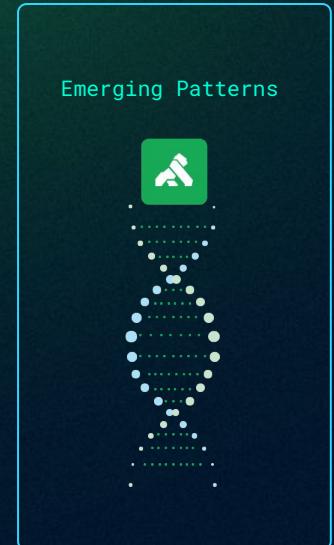
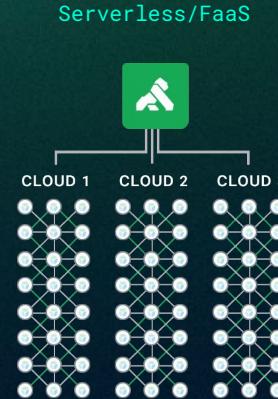
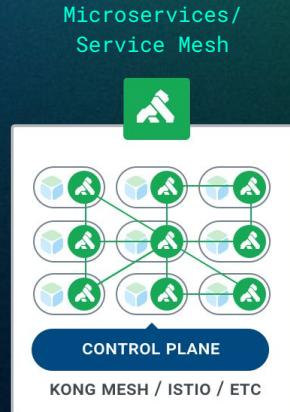
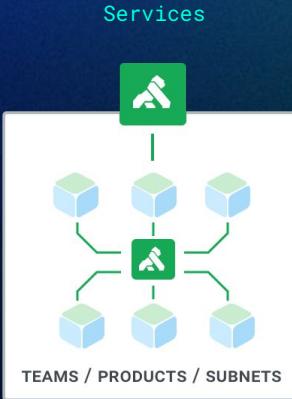
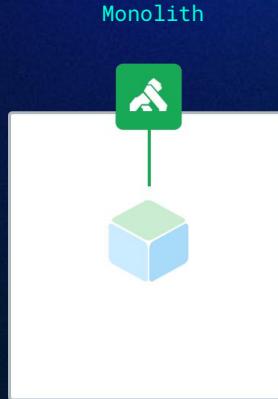
Kong Overview

Components, Architecture and Deployment

Successful businesses are Service-driven



We are on a Journey



Kong and Kong Enterprise

Millions of Kong downloads across multiple platforms.

- Open-Source
- **54M+** Downloads since 2015
- Built on top of NGINX
- 19,000+ Stars in GitHub
- 130+ Core Contributors
- 5000+ Ecosystem Contributions
- 100+ Meetups
- 38K+ Community Members

Kong is light weight



Built on NGINX
and not heavy
weight JVM.

20^{MB}

Size an order of
magnitude smaller
than ESB.



Plugin architecture
avoids bloating
and increases
simplicity.



Blazing fast
startup time: in
milliseconds.

Critical to quickly scale.
Critical for containers.



Minimal
component
dependency.

Kong has industry's best latency performance



Very low latency



Based on fast and performant NGINX.



Written in LuaJIT.

Fastest scripting language in the world.

Proven technology: 10%+ of worldwide traffic passes through NGINX/LuaJIT*.



Kong architecture is optimized for low latency.

In memory cache.

Non-blocking IO/Asynchronous request.

*<https://www.cloudflare.com/press-releases/2017/cloudflare-introduces-argo-a-virtual-backbone-for-a-faster-more-reliable-internet/>

Kong is extensible and deployment agnostic



Containers, cloud native, Kubernetes ready.



Plugin architecture makes Kong extensible



Programmable for any use-case even uncommon ones.



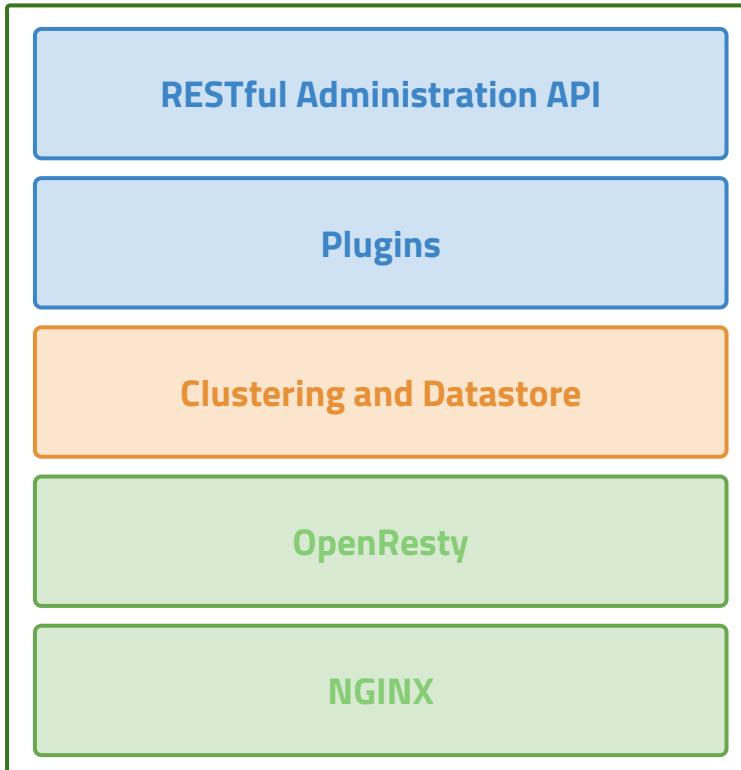
Can be deployed anywhere: on-prem, cloud, hybrid.



Future-proof deployment on any platform of the future.

Minimal moving parts and open-source community ensures Kong is ready for the next wave of platforms.

Kong OpenResty & NGINX



- JSON HTTP API
- Extendable by Plugins
- Can be integrated for automation

- Plugins created with LUA
- Intercept Request/Response lifecycle
- Can integrate with third-party services

- Cassandra or PostgreSQL
- Optionally Redis for some plugins
- Single or Multi-DC clustering

- Underlying engine of Kong
- Provides hooks for Req/Res lifecycle
- Extends underlying NGINX

- The core dependency
- Handles low-level operations
- Solid foundation and known tech





Kong and Kong Enterprise

Deployment

Docker deployment

Kong - Deployment Agnostic



Step 1

Start The

Datastore

```
docker run -d --name kong-database \
            -p 5432:5432 \
            -e "POSTGRES_USER=kong" \
            -e "POSTGRES_DB=kong" \
            postgres:9.6
```

Step 2

Run

Migrations

```
docker run --rm --name kong \
    --link kong-database:kong-database \
    -e "KONG_DATABASE=postgres" \
    -e "KONG_PG_HOST=kong-database" \
    kong-ee kong migrations up
```

Step 3

Start Kong

```
docker run -d --name kong
  --link kong-database:kong-database
  -e "KONG_DATABASE=postgres"
  -e "KONG_PG_HOST=kong-database"
  -e "KONG_ANONYMOUS_REPORTS=off"
  -e "KONG_ADMIN_LISTEN=0.0.0.0:8001"
  -e "KONG_VITALS=on"
  -e "KONG_PORTAL=on"
  -p 8000:8000
  -p 8443:8443
  -p 8445:8445
  -p 8001:8001
  -p 8002:8002
  -p 8003:8003
  -p 8004:8004
  -p 8444:8444
  kong-ee
```

Docker Compose

Docker Compose provides an easy to use configuration file to start multiple Docker instances at the same time with specific configuration and internal network.

It's the easiest and fastest way to spin up software and all the dependencies (e. g. Kong and Postgres).

Get docker-compose.yml at: <https://training.apim.eu/>

Important note: If you want to make the database persistent (otherwise a restart will wipe all previously created) uncomment the volumes-setting in your docker-compose.yml postgres section:

```
volumes:  
  - ./postgres-data:/var/lib/postgresql/data
```



Docker Compose

The provided docker-compose.yml for training also includes some more typical services like:

- Kong Enterprise
- Postgres
- httpbin (see <https://httpbin.org/>)
http://YOUR GATEWAY IP:81/anything from extern and
http://httpbin/anything from inside Docker (Gateway)
- JSONPlaceholder (see <https://jsonplaceholder.typicode.com/>)
http://YOUR GATEWAY IP:3005/posts from extern and
http://placeholder/posts from insided Docker (Gateway)
- Prometheus (not configured nor used today)



Training materials

<https://training.apim.eu/>

Username: training

Password: hongkong



Deployment Steps overview

1. Download the Kong Enterprise Docker image and import it into your local Docker installation
2. Using docker-compose
 - a. Install docker-compose
 - b. Get a docker-compose.yml (three different for groups working on the same machine)
 - c. Start Postgres and Kong
3. Check if admin API can be reached
 - a. curl <http://localhost:8001> (or http :8001 - see <https://httpie.org>)

Get help: <https://training.apim.eu/>



Lab Deployment using docker-compose.yml

1. Download Kong Enterprise and docker-compose.yml to your server
2. > curl -u training:hongkong http://(....) -o filename
3. On the server (as root)
 - a. Add Kong Docker image to internal registry:
docker load -i kong-enterprise-0.34.tar
 - b. Edit docker-compose.yml and replace "localhost" with IP-address (or FQDN) of your server (two times)
 - c. Choose one:
 - i. /usr/local/bin/docker-compose up -d
 - ii. /usr/local/bin/docker-compose -f docker-compose2.yml up -d
 - iii. /usr/local/bin/docker-compose -f docker-compose3.yml up -d
4. On client: check if admin API can be reached
 - a. curl http://YOUR_SERVER_IP:8001 (or http YOUR_SERVER_IP:8001)



Gateway

Routes, Services, Plugins, Authentication

Services and Routes

A **Service** represents the upstream service (typical URL elements)

```
{  
  "id": "0c61e164-6171-4837-8836-8f5298726d53",  
  "created_at": 1422386534,  
  "updated_at": 1422386534,  
  "name": "my-service",  
  "retries": 5,  
  "protocol": "http",  
  "host": "example.com",  
  "port": 80,  
  "path": "/some_api",  
  "connect_timeout": 60000,  
  "write_timeout": 60000,  
  "read_timeout": 60000  
}
```



Services and Routes

A **Route** represents a set of matching rules (a **Service** has multiple **Routes**)

```
{  
    "id": "173a6cee-90d1-40a7-89cf-0329eca780a6",  
    "created_at": 1422386534,  
    "updated_at": 1422386534,  
    "name": "my-route",  
    "protocols": ["http", "https"],  
    "methods": ["GET", "POST"],  
    "hosts": ["example.com", "foo.test"],  
    "paths": ["/foo", "/bar"],  
    "regex_priority": 0,  
    "strip_path": true,  
    "preserve_host": false,  
    "service": {"id": "f5a9c0ca-bdbb-490f-8928-2ca95836239a"}  
}
```

AND



Lab Gateway 1 (Routes and Services)

Create a service and route to httpbin

Try: <http://httpbin.org/anything>

Create a service and route such that

<http://localhost:8000/anything>

Will return the same, but through Kong



Plugins

Plugins add functionality on top of requests.

- Changing the request flow
- Modifying the request



Lab Gateway 2 (Caching)

Add caching to your API

Add the **proxy-cache** plugin to the service you created

Check the returned headers for cache activity and latency.



Lab Gateway 3 (Key Authentication)

Add the **key-auth** plugin to the service you created.

Then regain access to the service by adding a **Consumer** and a **key-auth credential**



Plugins (2)

A named plugin only runs once on a request.

Plugins can be configured on: Services, Routes, Consumers or Global. The more specific a plugin configuration gets, the higher the precedence.

- Highest: **consumer_id**, **route_id** and **service_id** set
- Lowest: none specified (= global)

NOTE: to use Consumer, an auth plugin MUST run on the request, and hence an auth plugin can NOT be configured on a Consumer.



Plugins (3)

A **Plugin** has a generic and plugin specific part

```
{  
  "id": "a3ad71a8-6685-4b03-a101-980a953544f6",  
  "name": "named-plugin",  
  "created_at": 1422386534,  
  "route": null,  
  "service": null,  
  "consumer": null,  
  "config": { ... plugin specific ... },  
  "run_on": "first",  
  "enabled": true  
}
```



Lab Gateway 4 (Rate Limiting)

1. Add a global rate-limit of 60 calls per hours, and 3 per minute
2. Make an exception for your previously added Consumer, allow 600 per hour.



Lab Gateway 5 (OpenID Connect)

Add the **OIDC Plugin** to the same **Service**

(see training portal <https://training.apim.eu> for Keycloak details)

Validate: **key-auth + OIDC**, is it AND or OR ?

Hint: If you want use the Kong GUI available at <http://localhost:8002/>



Lab Gateway 6 (Mixing authentications)

We want **key-auth** for M-2-M, but **OIDC** for humans.

We can set them up in an **OR** fashion by using [anonymous consumers](#), see
<https://docs.konghq.com/1.0.x/auth/#anonymous-access> and
<https://docs.konghq.com/1.0.x/auth/#multiple-authentication>

- Add an anonymous consumer
- Reconfigure the **key-auth** and **OIDC** plugins
- Add **request-termination**



Kong Manager & RBAC

Workspaces, Roles and Permissions

Get Global Visibility, Granular Control

Kong Manager

■ See everything

Dashboards, Views, Charts, and more

■ React instantly

One-click operations to execute any task in Kong

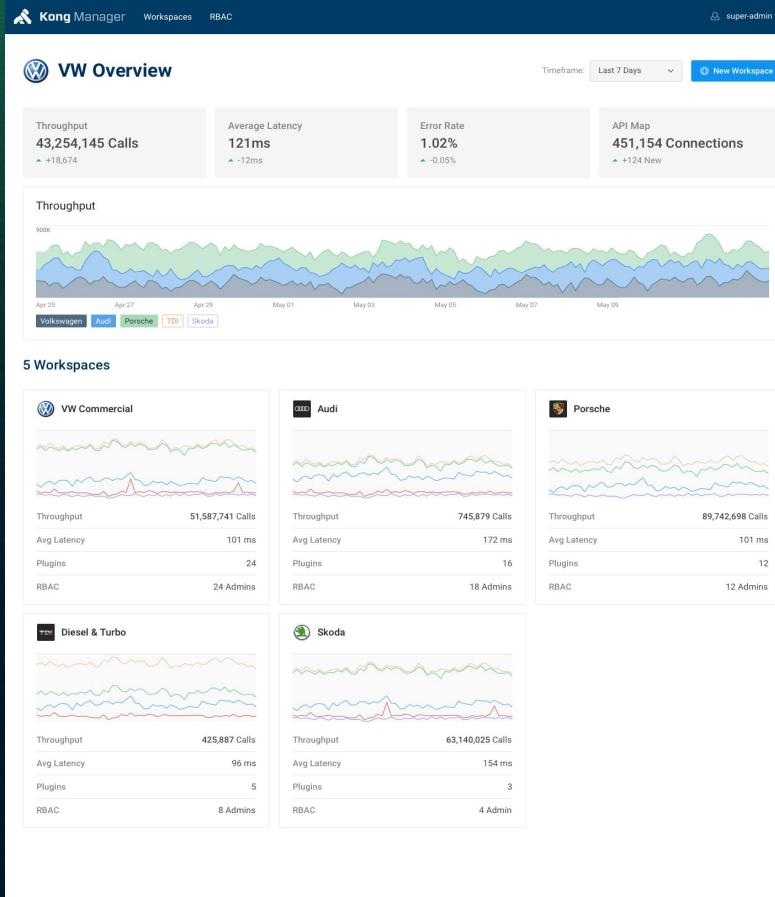
■ Optimize performance

Make dynamic changes to improve performance in real-time

■ Connect your teams

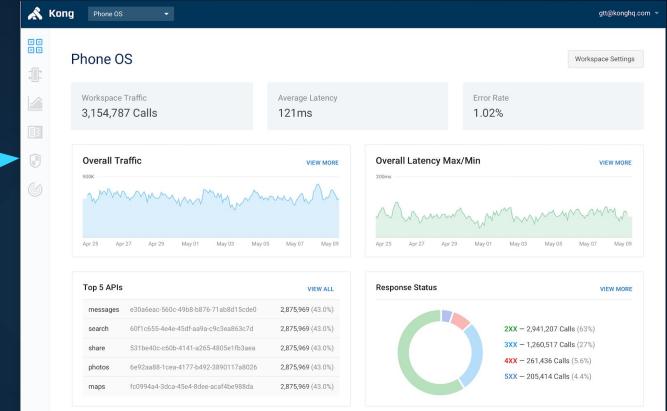
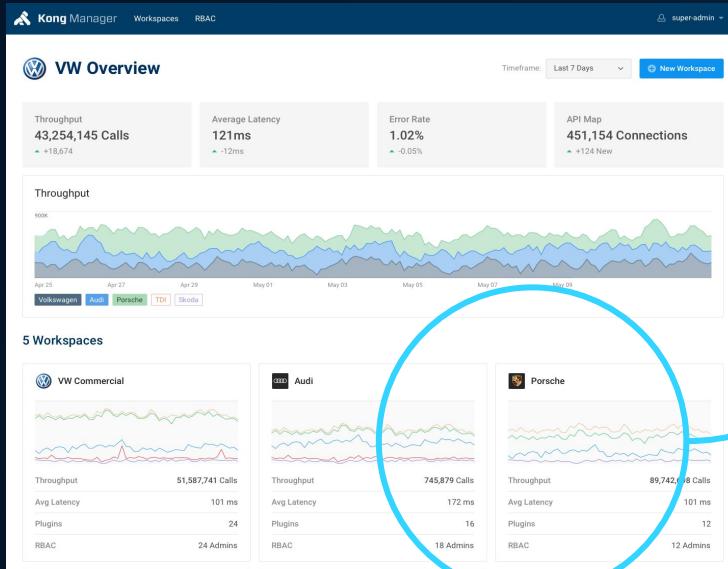
Use Workspaces to organize your teams and services

... and More.



Drill down to understand your teams

Kong Workspaces



- Organize your services
- Monitor performance with detailed analytics
- Define granular RBAC
- Quickly see anomalies

Empower developers, spur innovation

Your API documentation at your fingertips

The screenshot shows the Kong Developers API documentation interface. At the top, there's a navigation bar with the Kong logo, 'Kong Developers', 'API Reference', 'About', and 'Guides'. A user profile icon is also present.

The main content area displays the 'PATCH /plugins/{id_or_name}' endpoint. It includes:

- A title 'Update details about a plugin'.
- A 'Parameters' section with the following fields:
 - name**: string (formData) - Required.
 - consumer_id**: string (formData)
 - config.{property}**: string (formData)
- A note: 'The configuration properties for the Plugin which can be found on the plugins documentation page in the Plugin Gallery'.
- An 'Example Request' section showing code snippets in javascript, shell, python, and ruby.
- A 'Responses' section showing a 200 status with the message 'Plugin information updated'.
- An 'Example Value' section showing a JSON object with fields: id, service_id, consumer_id, name, and config.

■ Streamline discovery

Enable internal and external developers to find and interact with documentation

■ Onboard seamlessly

Grant access to developers with just a few clicks

■ Standardize specs

Drive consistency with support for the most popular spec formats

■ Customize your experience

Make the portal reflect your company brand, colors, and more

... and More.

Kong Enterprise: Security (RBAC)

■ More security for Admin operations

Create admin users and groups to protect the entire cluster.

■ Grow inside the organization

Multiple teams and departments can now be supported natively by Kong.

■ Self-Service resource provisioning

Teams can directly access their resources, while being blocked from using anything else.

■ Future developments

More security checks, auditing logs, reports, etc.

Actions
<pre>apis:create apis:read apis:update apis:delete apis:* plugins:create plugins:read plugins:update plugins:delete plugins:*</pre> <p>...</p>

Groups
<pre>Team1: [{ actions:[plugins:create plugins:read], targets: ["api_123"] }]</pre> <p>...</p>

Users
<pre>John: Team1 Luke: Team1 Bob: Team2 Joe: Team2 ...</pre>

Lab Management 1 (Workspaces and RBAC)

1. Create two workspaces
2. Create a new portal within one of workspace
3. Create a new super-admin user

Get help: <https://training.apim.eu/>



How to apply settings in Docker container

In order to enable RBAC we need change a setting and restart Kong.

As we are running inside a Docker container

1. > docker ps
2. Find the instance of the Kong Gateway and get its ID
3. > docker exec -ti ID_OF_KONG /bin/sh
4. # KONG_ENFORCE_RBAC=on kong reload
5. # exit
6. Try to access admin API on port 8001

See also: <https://docs.konghq.com/enterprise/0.34-x/installation/docker/#enable-rbac>

Get help: <https://training.apim.eu/>



Lab Management 2 (Workspaces and RBAC)

1. Create a user within your first workspace
2. Create a new workspace admin role
3. Add a user to this role in your first workspace
4. Test if your created user is allowed to work in second workspace

When done: disable RBAC (hence not to have pass tokens all the time...)

Get help: <https://training.apim.eu/>



Enterprise Portal

CMS, OpenAPI

Kong Portal

■ Interactive API Documentation

Feature rich reference editor and automatic API explorer.

■ Developer Management

Delight your devs with a seamless onboarding experience.

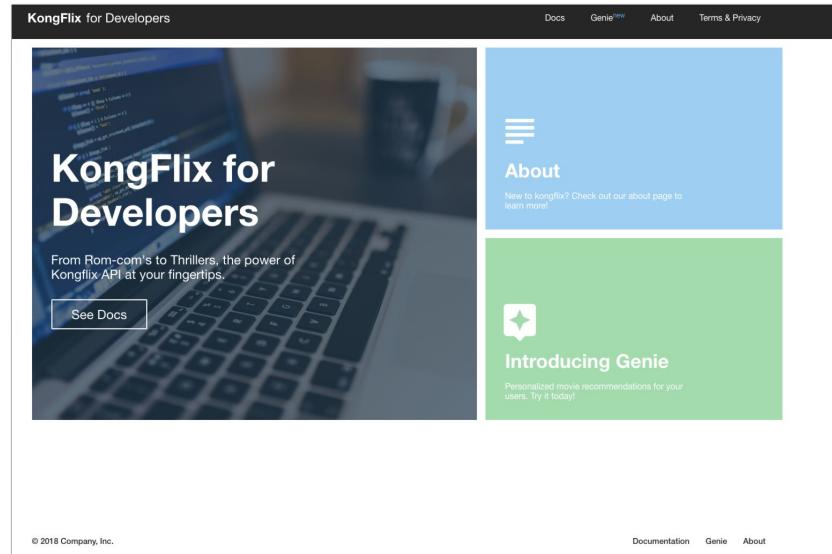
■ Flexible Customization

Make the portal truly yours, branding, colors and more.

■ Swagger & OpenAPI Spec

Native support for the most popular spec formats.

... and More.



Developer Lifecycle Management

Developers

[+ Invite Developers](#)

Approved	Revoked	Rejected	Requested Access
----------	---------	----------	------------------

Developers with permission to use all the resources offered in your Dev Portal.

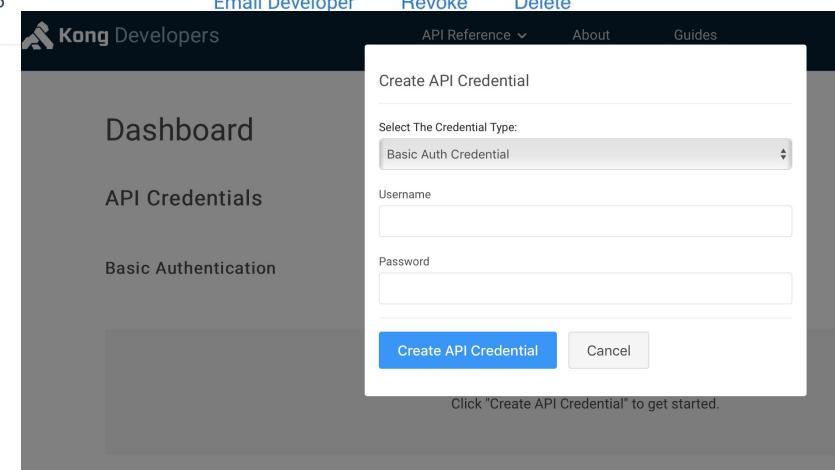
 email Press enter to search

email	meta	id	Email Developer	Revoke	Delete
user1@myemail.com	{"full_name": "luigi"}	1539c5e7-db87-4ffc-8a57-0d4b97f05bc5			

Supported Authentication Methods:

- Basic Auth
- Key Auth
- OpenIdConnect

Create API credentials through Kong Authentication plugins



Lab Portal 1 (Authentication)

1. Enable authentication
 - a. Basic Auth or
 - b. OpenID Connect with your Google Account or
 - c. OpenID Connect with Keycloak
2. Sign up a new developer
3. Enroll a new developer



Lab Portal 2 (OpenAPI)

1. Upload an OpenAPI definition
(for example <https://petstore.swagger.io/v2/swagger.json>)
2. Change the default theme (e. g. add some text to footer, change some colors)

Get help: <https://training.apim.eu/>



How the File Types Work Together

about.hbs
devportal.com/about



Partials:

- about-content.hbs
- sidebar.hbs
- header.hbs
- footer.hbs

Portal File Types

- **Specifications / Specs**- An API specification, in OpenAPI (formerly known as Swagger) format.
- **Partials**- These are Handlebar files made up of HTML, JS, and CSS content that define the look, feel, functionality, and structure of your Dev Portal.
- **Pages**- Pages are Handlebars templates that bring together the previously described Partial files and result in pages in your Dev Portal.
- **Loader**- The mechanism which compiles and serves HTML and JavaScript files to the browser when a visitor visits any Dev Portal page.
 - The Loader requests Pages, Partials and Specifications from Kong, which it uses to render your Dev Portal in the visitor's browser.
 - The Loader is not modifiable by Admins - instead, customization is performed by modifying Specifications, Partials, and Pages.



Lab Portal 3 (Themes)

1. Change the default theme (e. g. add some text to footer, change some colors)
2. Change the “Login with Google” to “Login with OpenID Connect”

Get help: <https://training.apim.eu/>



Vitals

Reports and use of external monitoring systems

What are Vitals?

The Vitals feature in Kong's Admin API and GUI provides useful metrics about the health and performance of your Kong nodes, as well as metrics about the usage of your Kong-proxied APIs.



VITALS

How to Activate Vitals

Vitals are activated in kong.conf

This can be done directly in the configuration file (requires a restart)

`vitals = on # vitals is enabled`

This can also be done through an environment variable (also requires a restart)

`export KONG_VITALS=on`

Request Counts

Total Requests- This metric is the count of all API proxy requests received. This includes requests that were rejected due to rate-limiting, failed authentication, etc.

Requests Per Consumer- This metric is the count of all API proxy requests received from each specific consumer. Consumers are identified by credentials in their requests (e.g., API key, OAuth token, etc) as required by the Kong Auth plugin(s) in use

Latency

Proxy Latency (Request)- These metrics are the min, max, and average values for the time, in milliseconds, that the Kong proxy spends processing API proxy requests that were rejected due to rate-limiting, failed authentication, etc.

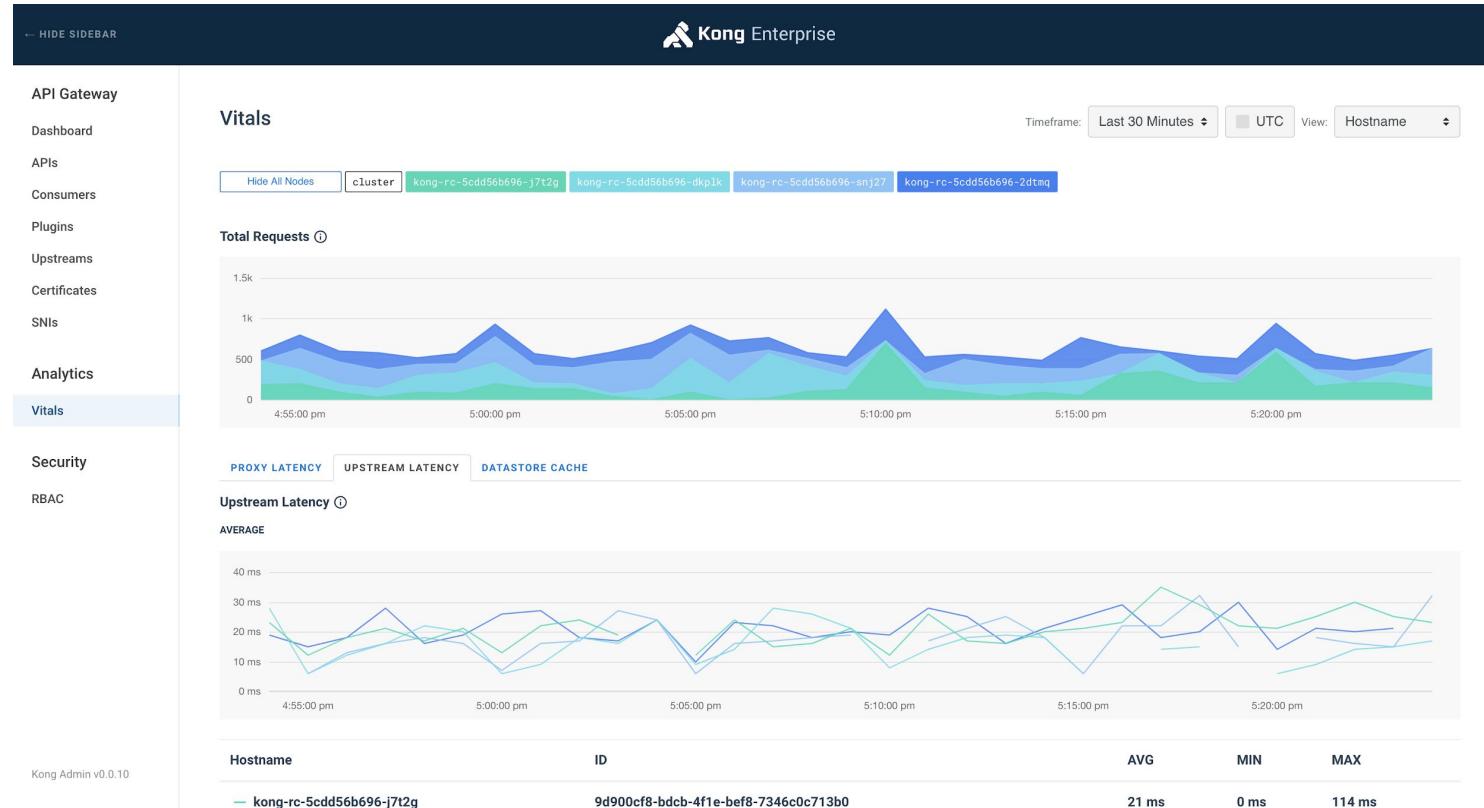
Upstream Latency- These metrics are the min, max, and average values for the time elapsed, in milliseconds, between Kong sending requests upstream and Kong receiving the first bytes of responses from upstream.

Datastore Cache

Proxy Latency (Request)- This metric contains the ratio of datastore cache hits to the total count of datastore cache requests.

A “Hit” indicates that an entity was retrieved from the data store cache. A “Miss” indicates that the record had to be fetched from the datastore.

Visually Through the Kong Admin GUI



Visually Through the Kong Admin GUI



The Vitals API Resources

`http :8001/vitals`

Cluster

- Get aggregated stats for the cluster
- Get count of requests for the given consumer across entire cluster

Nodes-

- Get individual stats for all nodes
- Get stats for a specific node by UUID
- Get count of requests for the given consumer on each node

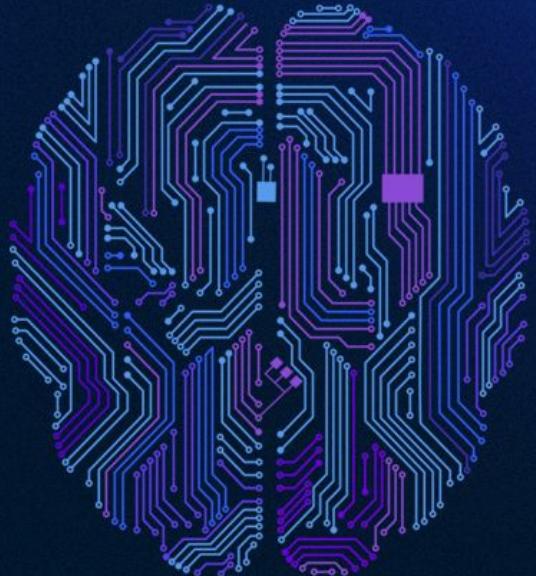
Vitals

- Get general information about Kong vitals

Plugins Development

Roadmap

What's coming next...



Automate your architecture

Kong Brain

- **Map your connections**

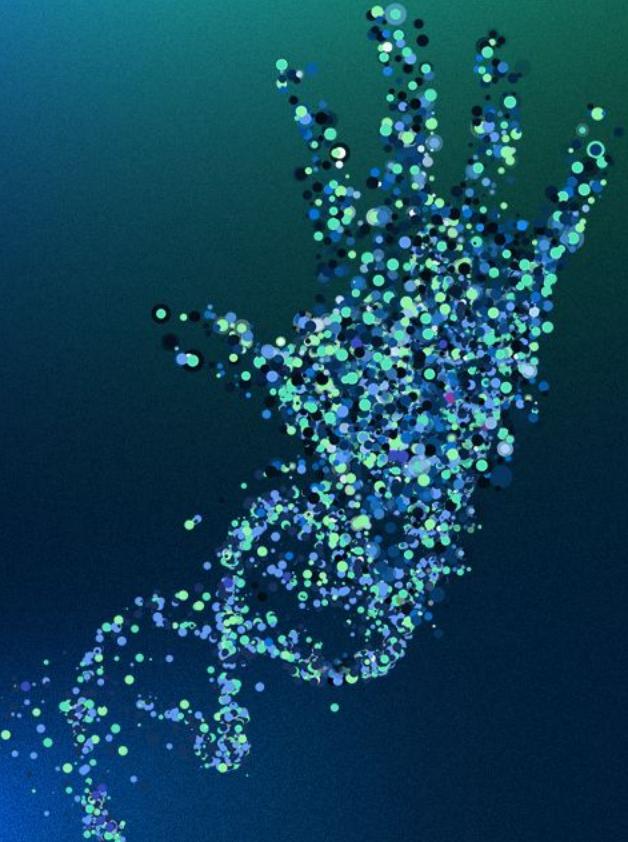
Create a visual map of all your services

- **Automate documentation**

Create and update documentation automatically

- **Enforce standards**

Ensure that docs are consistent across all your devs



Recognize Threats, Respond Instantly

Kong Immunity

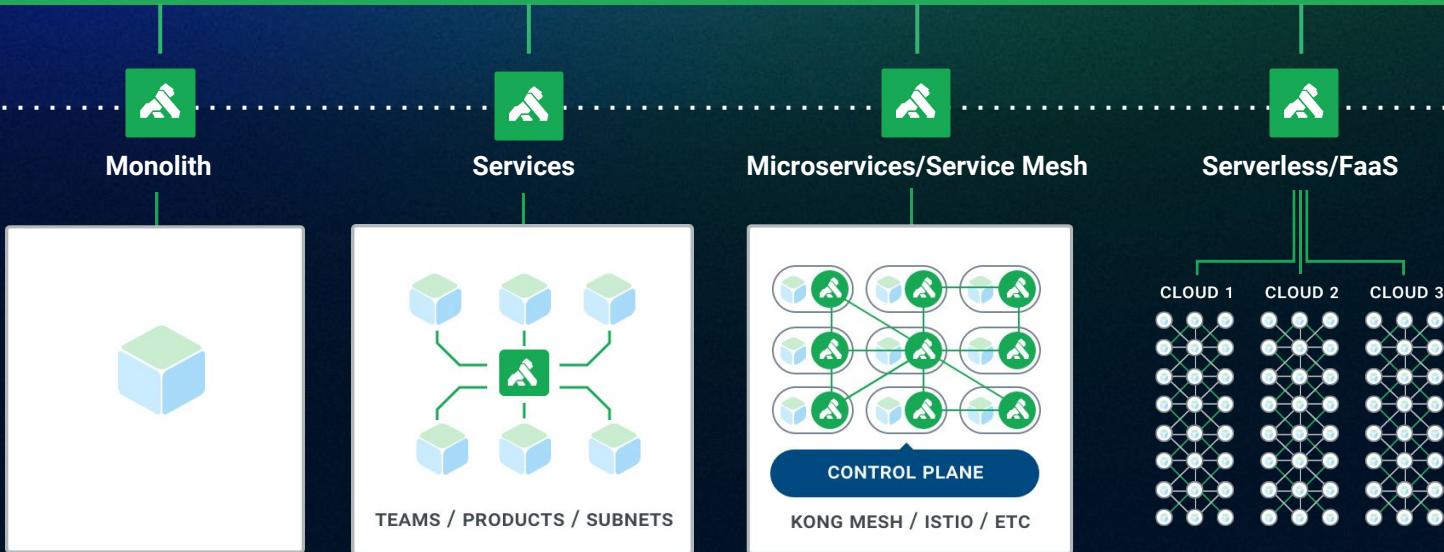
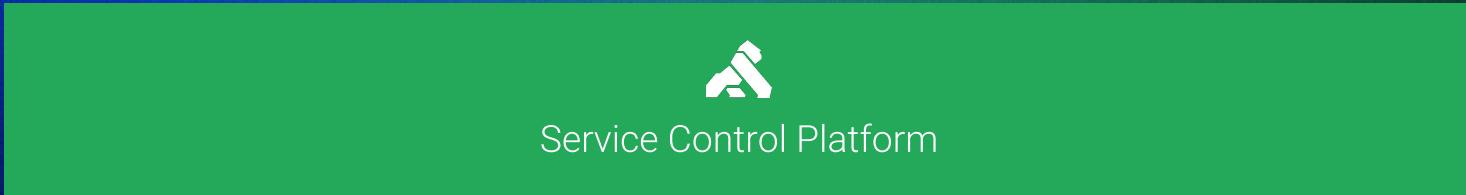
- **Identify threats autonomously**
Proactively identify traffic anomalies
- **Analyze anomalies instantly**
Track anomalies throughout their lifecycle
- **Take action against threats**
Quickly enact new policies to minimize risk

Overflow

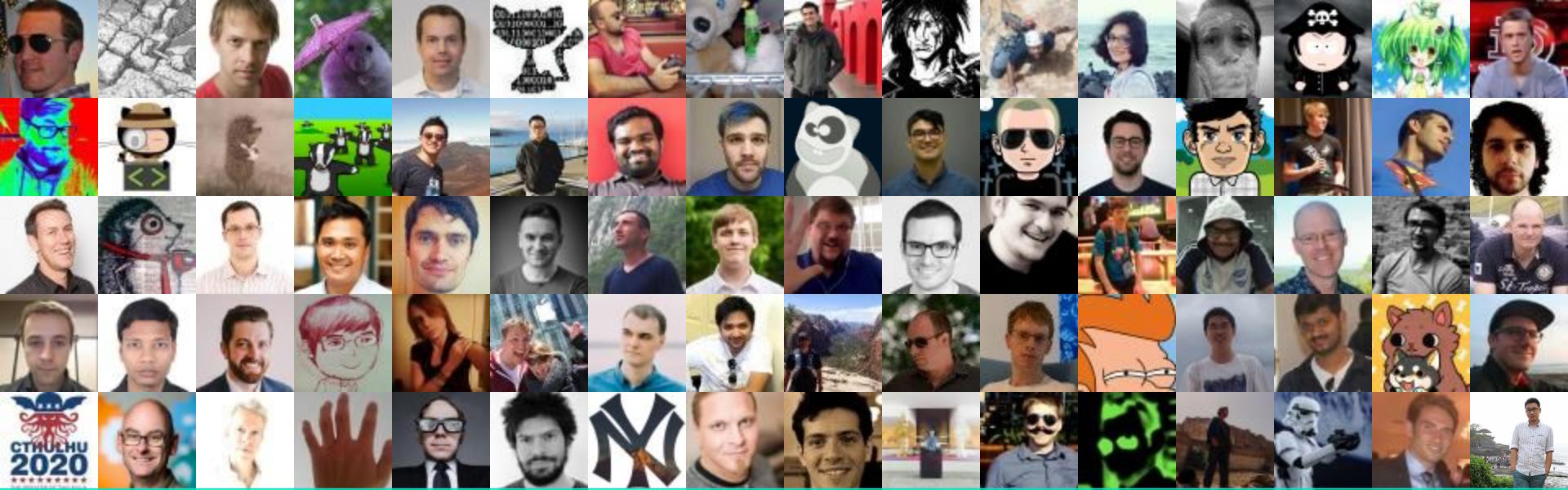
Slides if needed

Service Control Platform

To intelligently broker the flow of information across all services



Infrastructure



130+ Core Contributors

38,000+ Community Members

100+ Meetups around the world

Open-Core

54M
downloads



100

Enterprise Customers

Top Global 5000 and Top Fortune 500

OpenID Connect

- **Authenticate and Authorize Consumers**

With existing third-party OIDC Providers.

- **Supports all the most popular OIDC Cloud providers**

Auth0, Okta, Google, Microsoft Live Connect, Microsoft Azure AD, Yahoo!, Salesforce, Paypal, etc.

- **And On-Premise providers too**

Connect2id, PingFederate, IdentityServer4, OpenAM, Gluu, Keycloak, Dex, etc

- **Enterprise Governance applied to Kong**

Enabling and disabling consumers on Kong as easy as managing any other user in your organization.

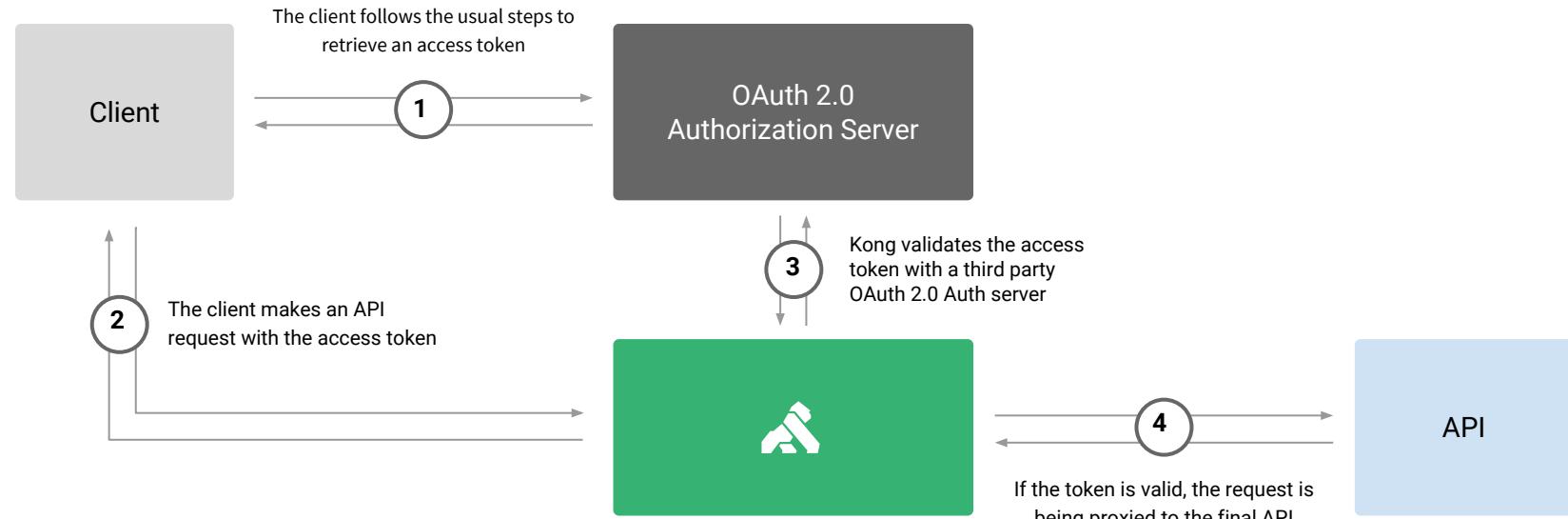
Supported Features:

- JWA / JWT / JWS
- Authentication Flows
- Introspection
- Revocation
- Token Requesting
- Refreshing the Tokens
- Discovery
- User Info Endpoint Support
- WebFinger
- Encoding and Decoding
- Bearer Token Support
- Proof-Key for Code Exchange (PKCE)
- Dynamic Client Registration
- Back-Channel Logout Support
- Claims Verification

OAuth 2.0 Introspection Endpoint

■ Validate third-party Access Tokens

With external OAuth 2.0 Authorization servers.



Kong Enterprise: Scalability

- **Enterprise Rate-Limiting**

Higher performance with eventual consistency for multi-region setups and accuracy. Redis HA Sentinel Support.

- **Proxy Caching**

Cache responses at the edge for higher speed and network optimizations

- **Backups**

Import/Export your Kong config and keep them up to date and stored on S3 or anywhere

- **Advanced Routing & Transformations**

Combine both Regex variables and dynamic transformations for sophisticated routing

Support your success

- **Enterprise-Level Support**

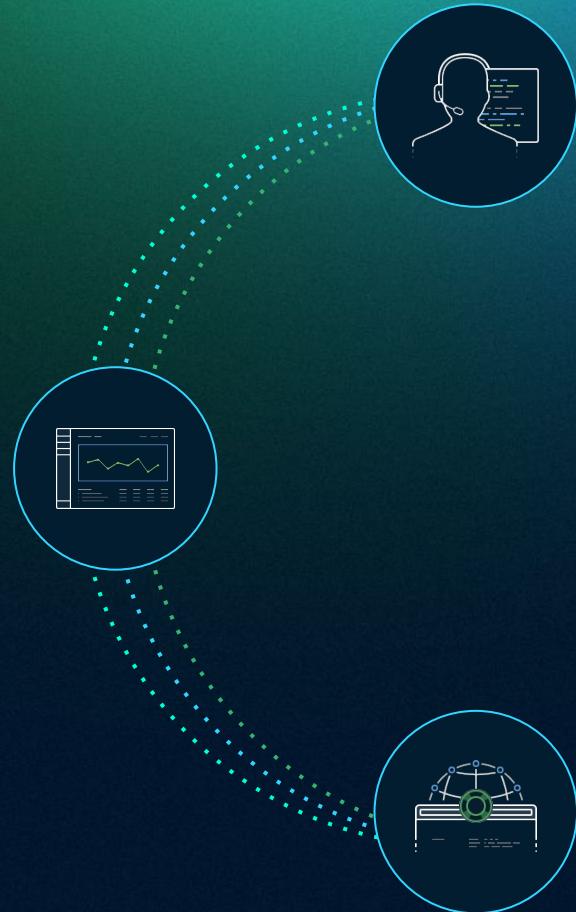
24x7 Phone and email support with 2-hour Service Level Agreements for Severity Level 1 issues.

- **Onboarding support and Health checks**

On-boarding and regular health checks with Kong CSEs throughout pre- and post-production use.

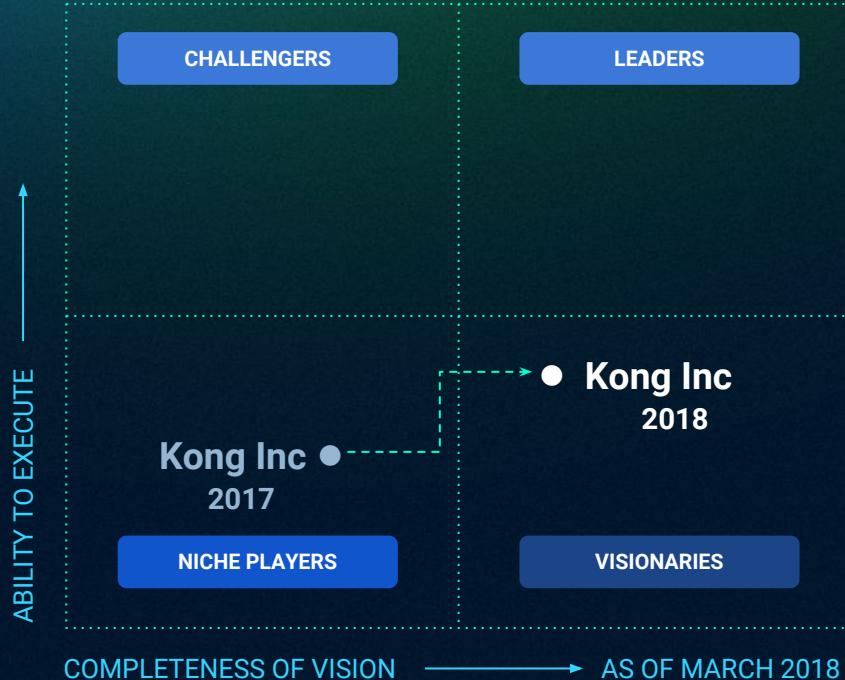
- **Professional Services (Additional Cost)**

Access to Professional Services for integration, architecture, and custom plugin development work



Industry Validation - Visionary by Gartner

Magic Quadrant for Full Life Cycle API Management



Named a **Visionary** by Gartner in
only **two years**

According to Gartner, “Kong has a differentiating focus on use cases involving microservices and serverless functions... Kong’s product strategy is comprehensive and heading in the right direction.”