



Research & Development Team

API Gateway with Kong in Practice

www.pnpsw.com

sommai.k@gmail.com

081-754-4663

Lineid : sommaik

Medium: @sommaikrangpanich

สมหมาย กรังพาณิช

ตำแหน่งปัจจุบัน

- กรรมการสมาคมอุตสาหกรรมซอฟต์แวร์ไทย ATSI
- กรรมการผู้จัดการบริษัท พี เอ็น พี โซลูชั่น จำกัด

ด้านความเชี่ยวชาญ

- ผู้เชี่ยวชาญด้านการออกแบบและพัฒนา Software ด้วย Framework ดังนี้
 - Java, Springboot, Go, Node.js
 - Full Stack, Angular, Vue, React, Svelte
 - Apollo GraphQL
 - Docker, Docker Swarm, K8s
 - Flutter, Dart
 - MongoDB, Oracle, MySQL, SQL Server, DB2, PostgreSQL
- ผู้เชี่ยวชาญด้านการออกแบบและประยุกต์ใช้ DevOps ในการพัฒนา Software
- ผู้เชี่ยวชาญด้านการออกแบบและประยุกต์ใช้ Cloud Native ในการพัฒนา Software
- ผู้เชี่ยวชาญด้านการออกแบบและพัฒนาระบบงานในรูปแบบ Microservice

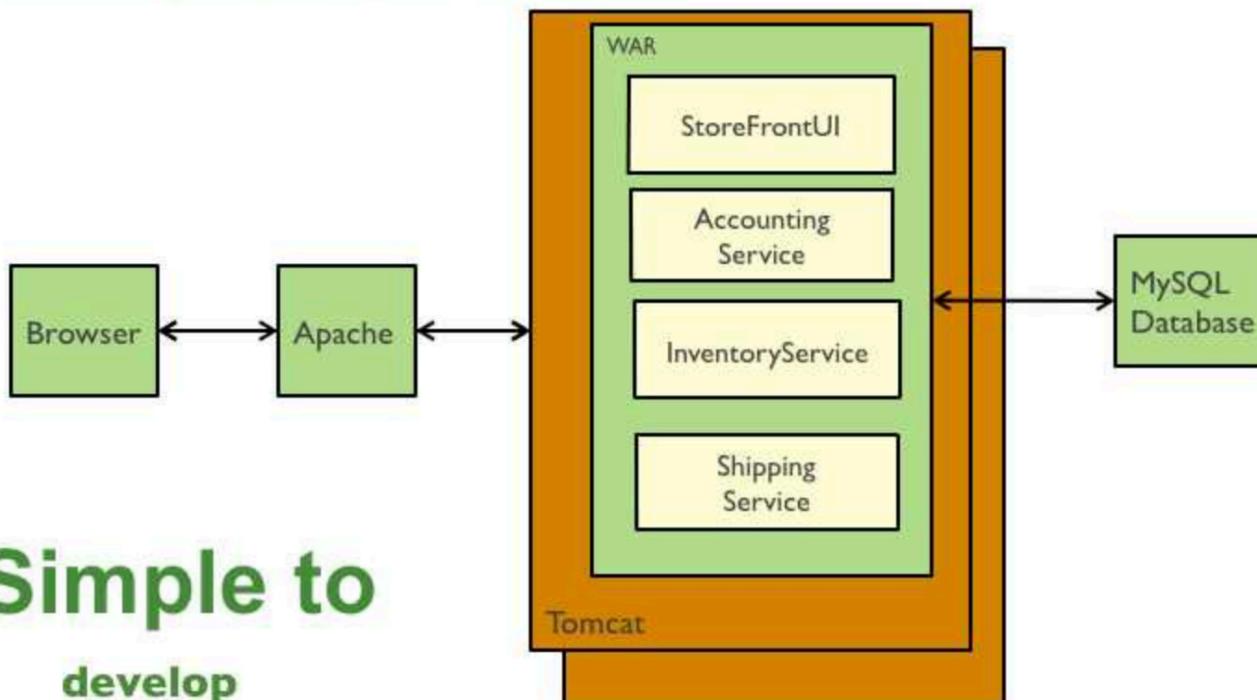


Introduction to

MICROSERVICE ARCHITECTURE

Monolithic Architecture

Traditional web application architecture



Simple to

develop
test
deploy
scale

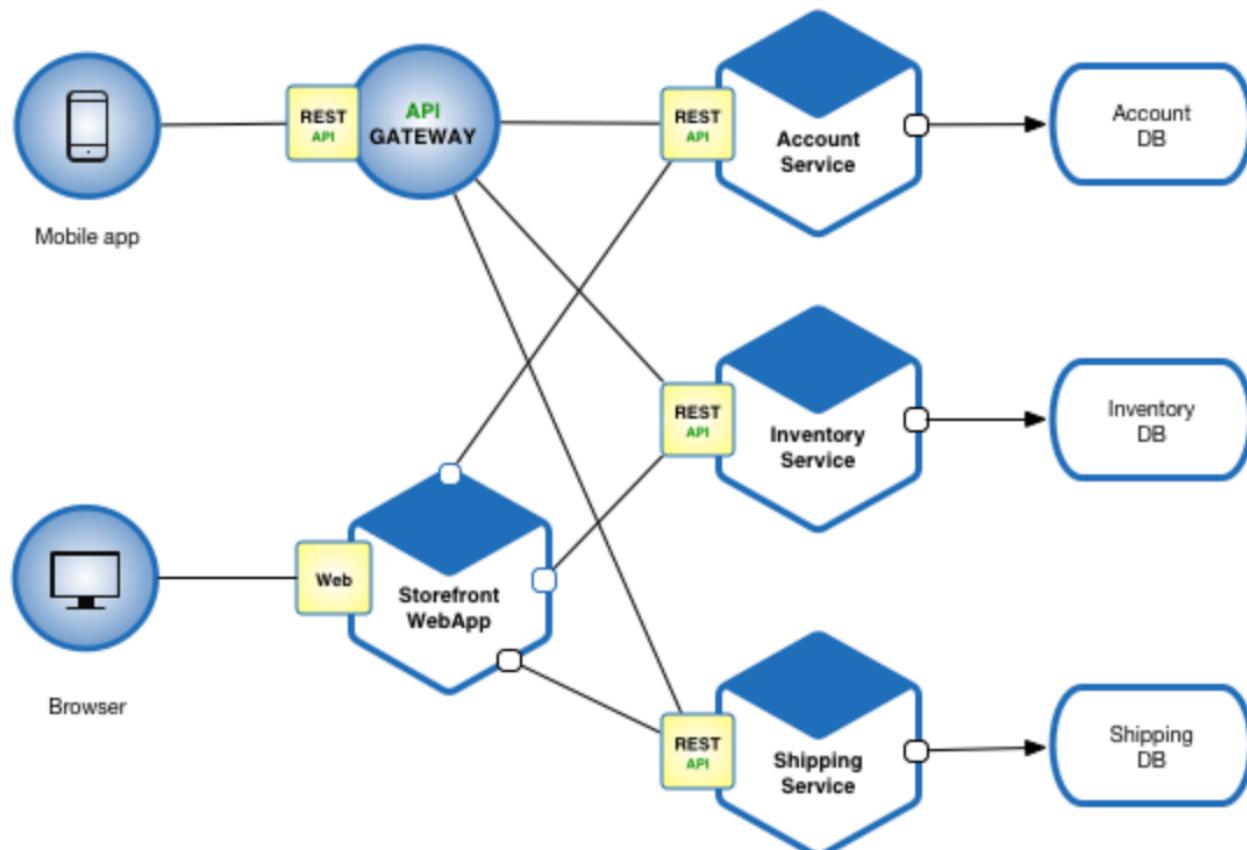
Benefits

- Simple to develop - the goal of current development tools and IDEs is to support the development of monolithic applications
- Simple to deploy - you simply need to deploy the WAR file (or directory hierarchy) on the appropriate runtime
- Simple to scale - you can scale the application by running multiple copies of the application behind a load balancer

Drawbacks

- The large monolithic code base intimidates developers, especially ones who are new to the team.
- Overloaded IDE
- Overloaded web container
- Continuous deployment is difficult
- Scaling the application can be difficult
- Obstacle to scaling development
- Requires a long-term commitment to a technology stack

Microservice Architecture



Benefits

- Enables the continuous delivery and deployment of large, complex applications.
 - Improved maintainability - each service is relatively small and so is easier to understand and change
 - Better testability - services are smaller and faster to test
 - Better deployability - services can be deployed independently
 - It enables you to organize the development effort around multiple, autonomous teams. Each (so called two pizza) team owns and is responsible for one or more services. Each team can develop, test, deploy and scale their services independently of all of the other teams.

Benefits #2

- Each microservice is relatively small:
 - Easier for a developer to understand
 - The IDE is faster making developers more productive
 - The application starts faster, which makes developers more productive, and speeds up deployments
- Improved fault isolation. For example, if there is a memory leak in one service then only that service will be affected. The other services will continue to handle requests. In comparison, one misbehaving component of a monolithic architecture can bring down the entire system.
- Eliminates any long-term commitment to a technology stack. When developing a new service you can pick a new technology stack. Similarly, when making major changes to an existing service you can rewrite it using a new technology stack.

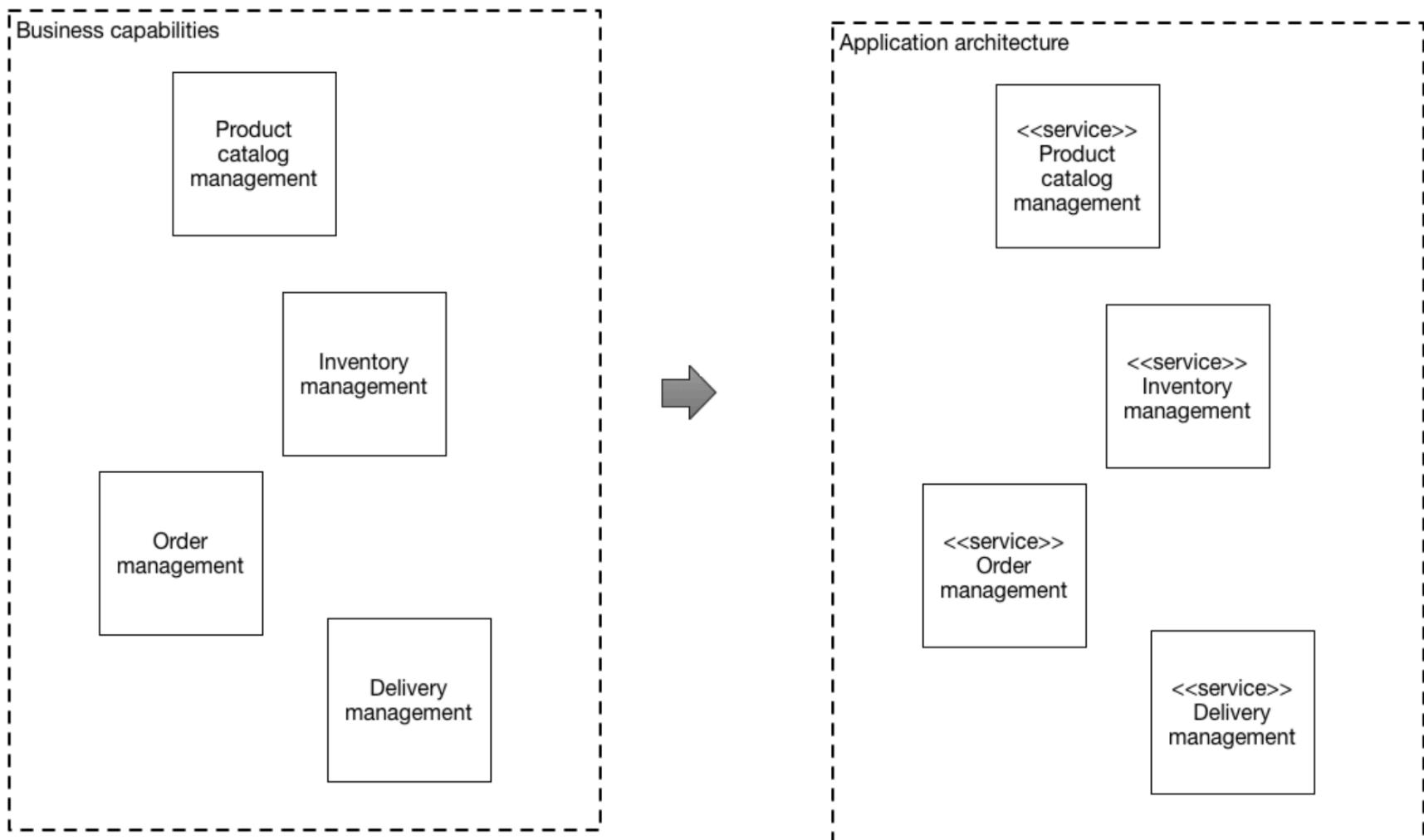
Drawbacks

- Developers must deal with the additional complexity of creating a distributed system:
 - Developers must implement the inter-service communication mechanism and deal with partial failure
 - Implementing requests that span multiple services is more difficult
 - Testing the interactions between services is more difficult
 - Implementing requests that span multiple services requires careful coordination between the teams
 - Developer tools/IDEs are oriented on building monolithic applications and don't provide explicit support for developing distributed applications.
- Deployment complexity. In production, there is also the operational complexity of deploying and managing a system comprised of many different services.

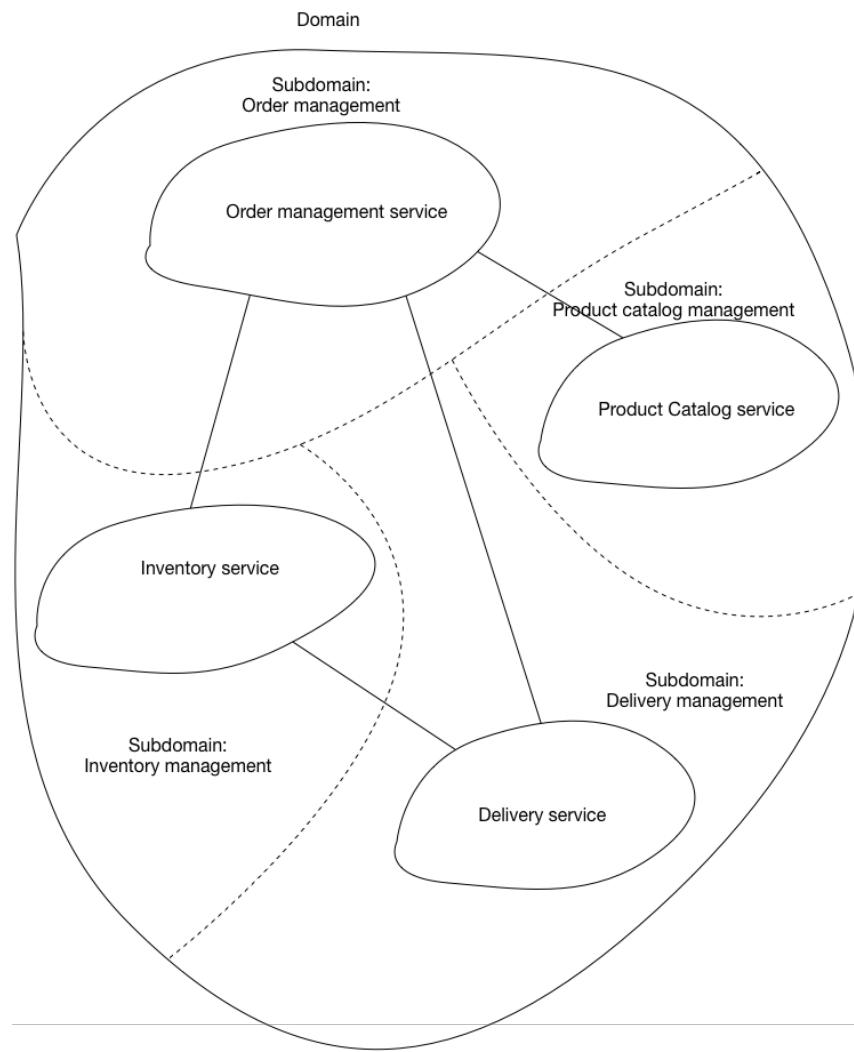
How to decompose

- Decompose by business capability and define services corresponding to business capabilities.
- Decompose by domain-driven design subdomain.
- Decompose by verb or use case and define services that are responsible for particular actions. e.g. a Shipping Service that's responsible for shipping complete orders.
- Decompose by nouns or resources by defining a service that is responsible for all operations on entities/resources of a given type. e.g. an Account Service that is responsible for managing user accounts.

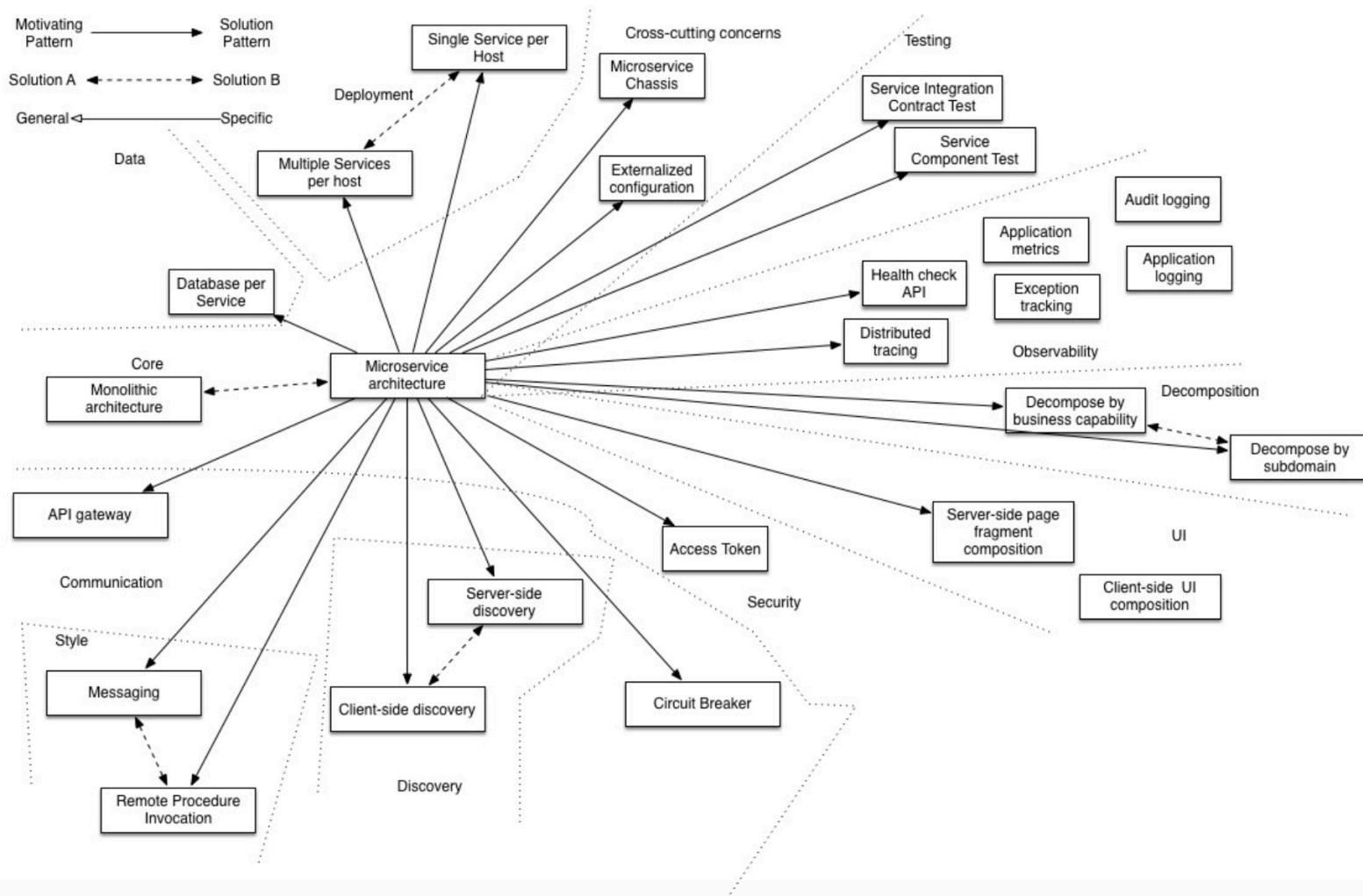
Decompose by business capability



Decompose by domain-driven design subdomain



Related patterns



Deployment

- Single Service per Host
- Multiple Service per Host

Cross-cutting concerns

- Microservice Chassis
- Externalized Configuration

Testing

- Service Integration Contract Test
- Service Component Test

Observability

- Health Check API
- Distributed Tracing
- Application Metrics
- Exception Tracking
- Audit Logging
- Application Logging

Decomposition

- Decompose by Business Capability
- Decompose by Subdomain

UI

- Server-Side Page Fragment Composition
- Client-Side UI Composition

Security

- Access Token

Discovery

- Server-Side Discovery
- Client-Side Discovery

Style

- Messaging
- Remote Procedure Invocation

Communication

- API gateway
- Circuit Breaker

Core

- Monolithic Architecture

Data

- Database per Service

Microservice with
REST API

The Anatomy Of A Request

- **The endpoint (or route)** is the url you request for.
- **The method** is the type of request you send to the server.
- **The headers** are used to provide information to both the client and server.
- **The data (or body)** contains information you want to be sent to the server. This option is only used with POST, PUT, PATCH requests.

The endpoint

- The **root-endpoint** is the starting point of the API you're requesting from.
- The **path** determines the resource you're requesting for.
- The **query parameters** give you the option to modify your request with key-value pairs.
- They always begin with a question mark (?).
- Each parameter pair is then separated with an ampersand (&).



The method

- GET
- POST
- PUT and PATCH
- DELETE
- These methods provide meaning for the request you're making. They are used to perform four possible actions: Create, Read, Update and Delete (CRUD).

GET

- This request is used to get a resource from a server.
- If you perform a `GET` request, the server looks for the data you requested and sends it back to you.
- In other words, a `GET` request performs a `READ` operation.
- This is the default request method.

POST

- This request is used to create a new resource on a server.
- If you perform a `POST` request, the server creates a new entry in the database and tells you whether the creation is successful.
- In other words, a `POST` request performs an `CREATE` operation.

PUT and PATCH

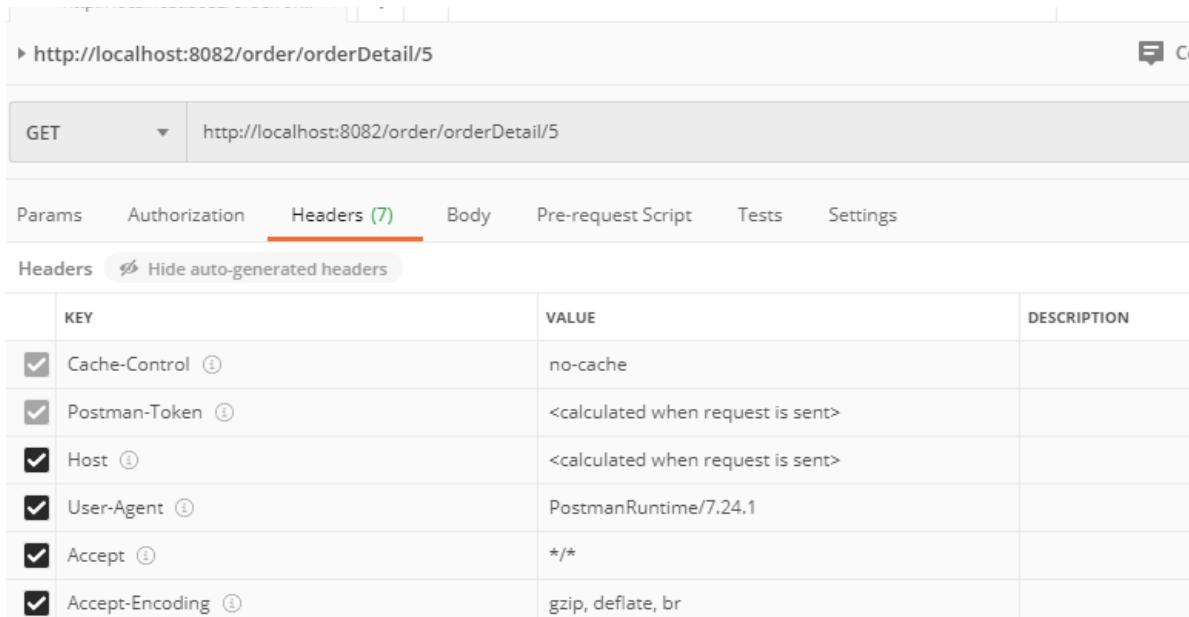
- These two requests are used to update a resource on a server.
- If you perform a `PUT` or `PATCH` request, the server updates an entry in the database and tells you whether the update is successful.
- In other words, a `PUT` or `PATCH` request performs an `UPDATE` operation.

DELETE

- This request is used to delete a resource from a server.
- If you perform a `DELETE` request, the server deletes an entry in the database and tells you whether the deletion is successful.
- In other words, a `DELETE` request performs a `DELETE` operation.

The Headers

- Headers are used to provide information to both the client and server.
- It can be used for many purposes, such as authentication and providing information about the body content.
- **HTTP Headers** are property-value pairs that are separated by a colon.



http://localhost:8082/order/orderDetail/5

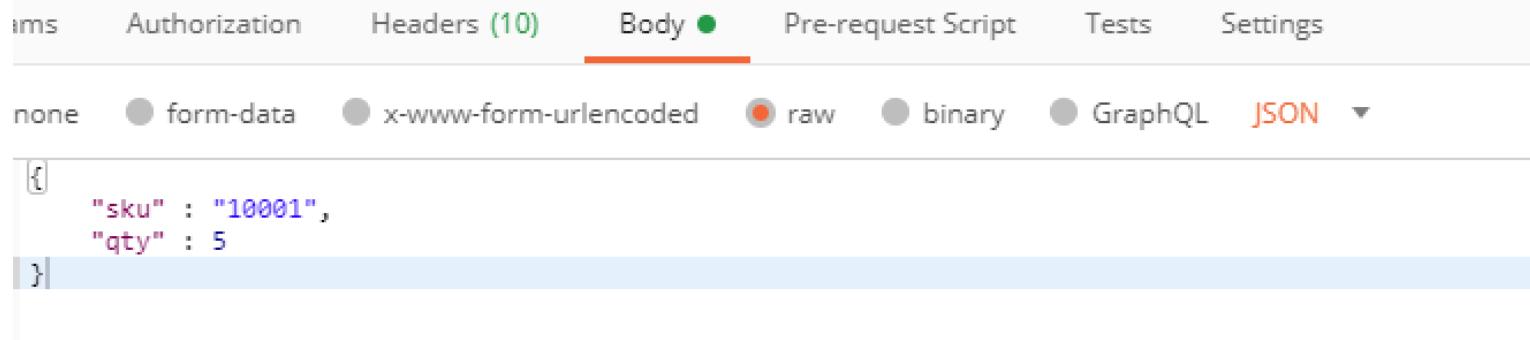
GET http://localhost:8082/order/orderDetail/5

Headers (7)

KEY	VALUE	DESCRIPTION
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.24.1	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	

The Data

- The data (sometimes called “body” or “message”) contains information you want to be sent to the server.
- This option is only used with POST, PUT, PATCH requests.



The screenshot shows the Postman interface with the 'Body' tab selected. The 'Body' tab has a green dot next to it, indicating it is active. Below the tabs, there are several options for data encoding: 'none', 'form-data', 'x-www-form-urlencoded', 'raw', 'binary', and 'GraphQL'. The 'raw' option is selected and highlighted in orange. To the right of the encoding type is a dropdown menu currently set to 'JSON'. The main body area contains the following JSON data:

```
{"sku": "10001",  
 "qty": 5}
```

Software

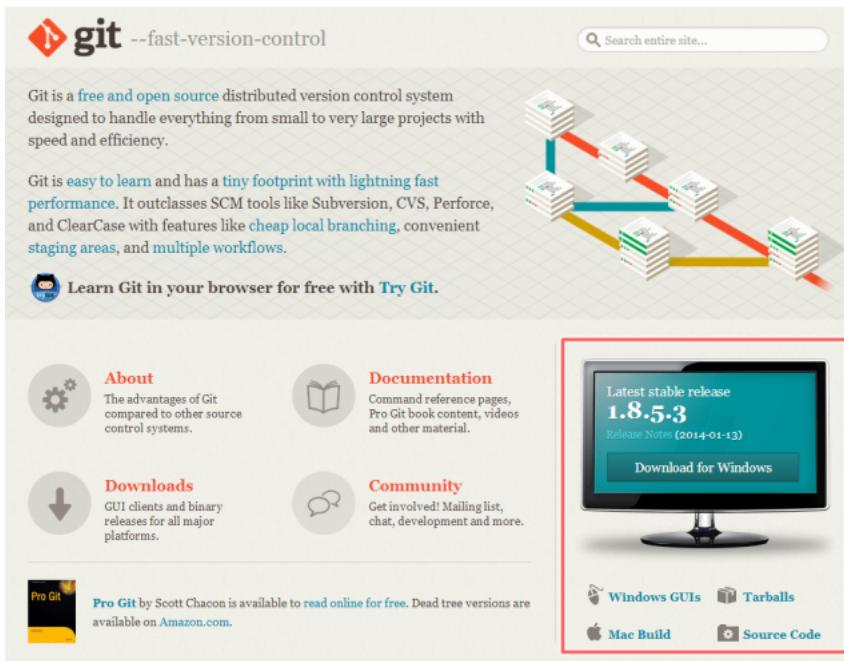
INSTALLATION

การติดตั้ง

GIT FOR WINDOWS

การติดตั้ง GIT สำหรับ Windows

- เข้า website <https://git-scm.com/downloads>
- เลือก Download Git เพื่อติดตั้งบน Windows



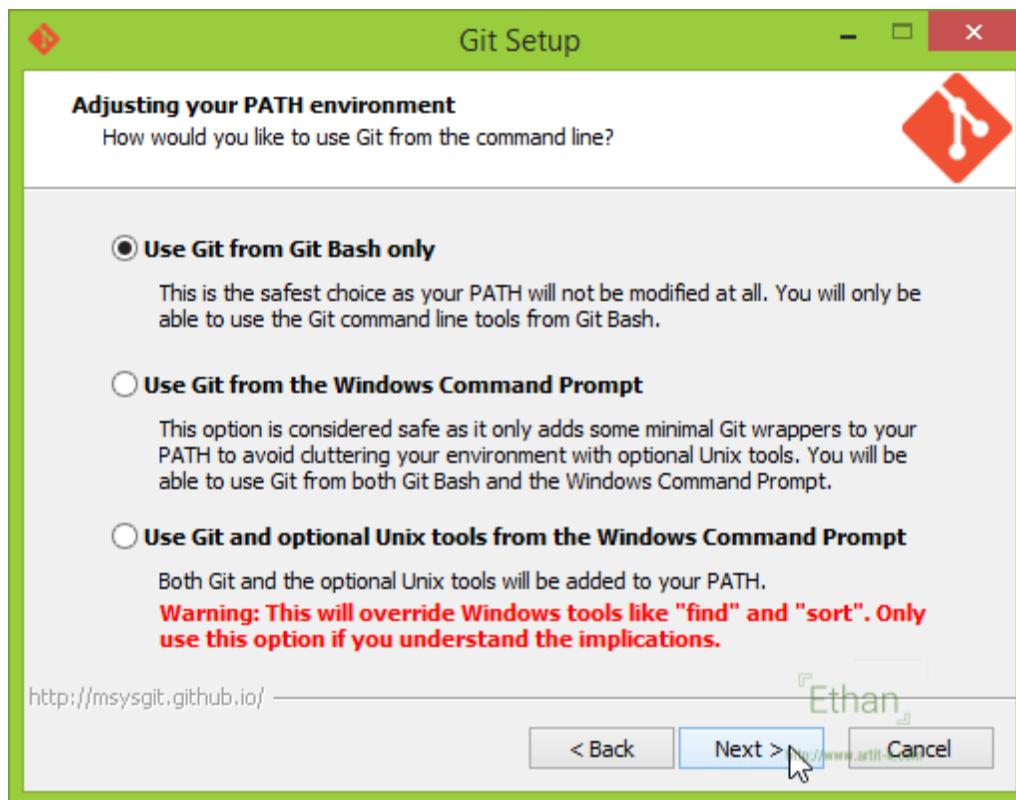
การติดตั้ง GIT สำหรับ Windows #2

- ติดตั้งโดยใช้สิทธิ Administrator ในการติดตั้ง



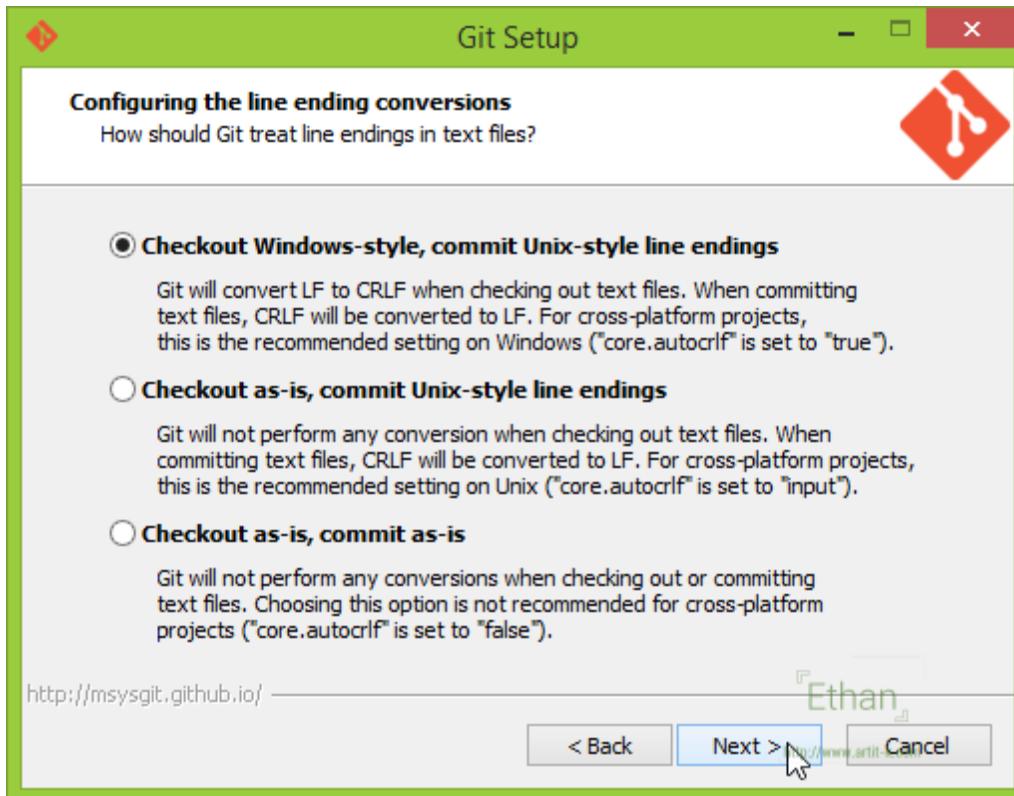
การติดตั้ง GIT สำหรับ Windows #3

- เลือกติดตั้งแบบ **Use Git from the Windows Command Prompt**



การติดตั้ง GIT สำหรับ Windows #4

- เลือกเป็น **Checkout Windows-style, commit Unix-style line endings**



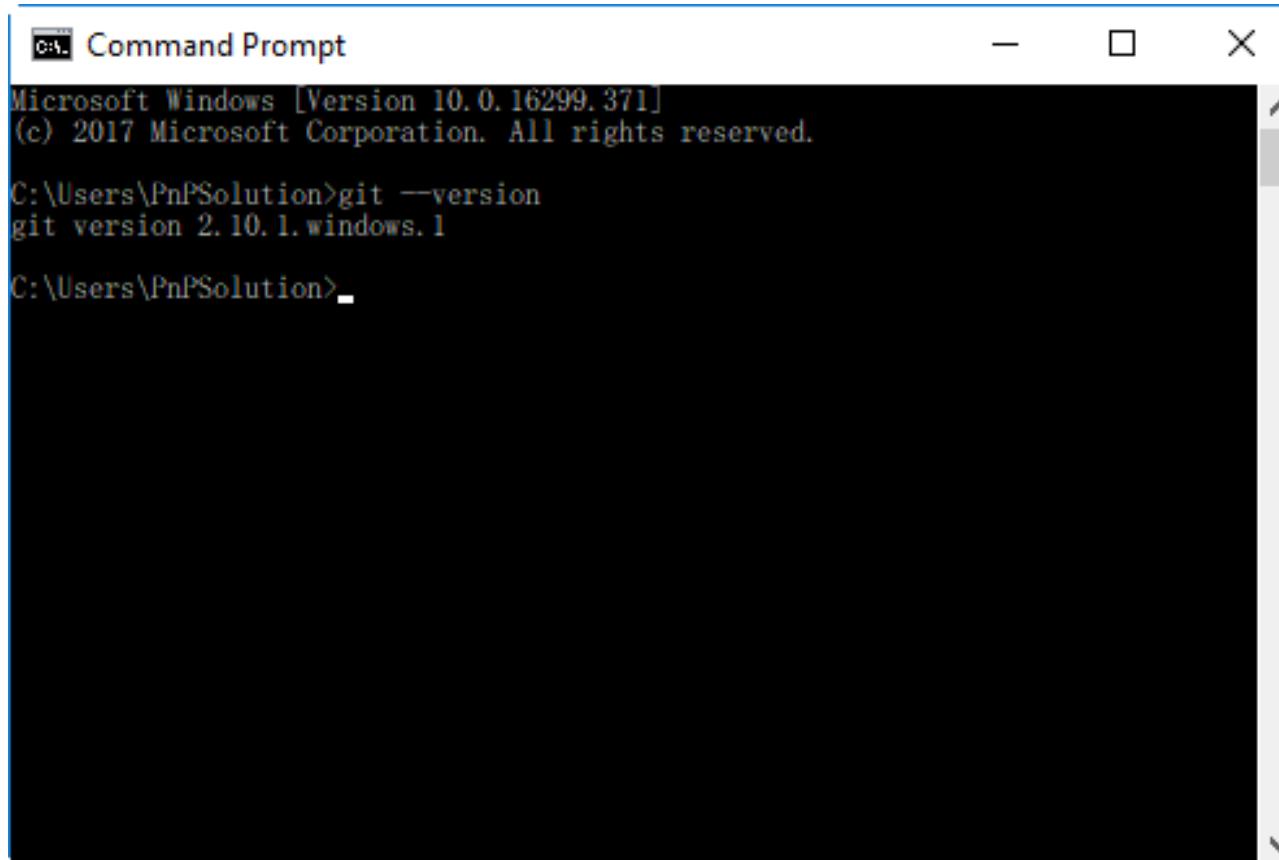
การติดตั้ง GIT สำหรับ Windows #5

- กดปุ่ม next ไปจนถึงหน้าสุดท้าย



ทดสอบหลังการติดตั้ง Git

- เปิดโปรแกรม cmd และพิมพ์คำสั่ง git --version



```
Command Prompt
Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\PnP\Documents>git --version
git version 2.10.1.windows.1

C:\Users\PnP\Documents>
```

การติดตั้ง

DOCKER FOR WINDOWS

การติดตั้ง Docker สำหรับ Windows

- ความต้องการขั้นพื้นฐานสำหรับการติดตั้ง Docker for windows
 - Windows ต้องเป็น version 64bit เท่านั้น
 - Windows 7 or higher
 - CPU ต้องมีความสามารถ Hyper-v และรองรับ BIOS

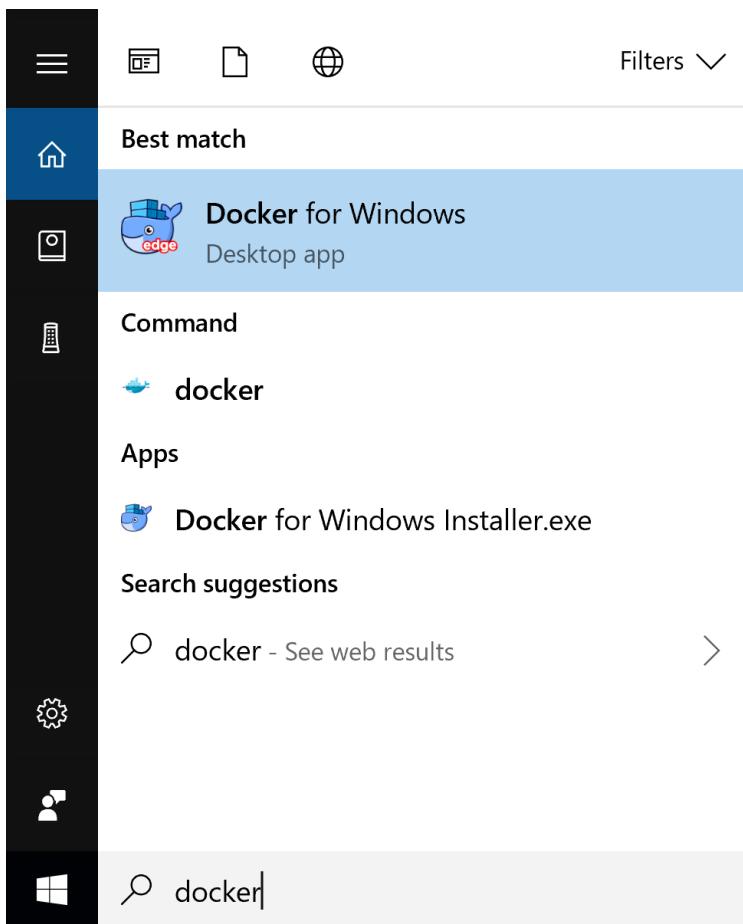


การติดตั้ง Docker สำหรับ Windows #2

- เข้า link [https://download.docker.com/win/stable/
Docker%20for%20Windows%20Installer.exe](https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe)
- กด next ไปเรื่อยๆ จนสิ้นสุดการติดตั้ง

การติดตั้ง Docker สำหรับ Windows #3

- หลังจากติดตั้งเสร็จแล้วให้ทำการเปิด docker ดังนี้



การติดตั้ง Docker สำหรับ Windows #4

- จะมี docker run อยู่ที่ taskbar ดังภาพ



- ทดสอบการติดตั้งโดยการเปิด cmd และพิมพ์คำสั่งดังนี้

```
Sommais-MacBook-Pro:~ sommaik$ docker ps
```

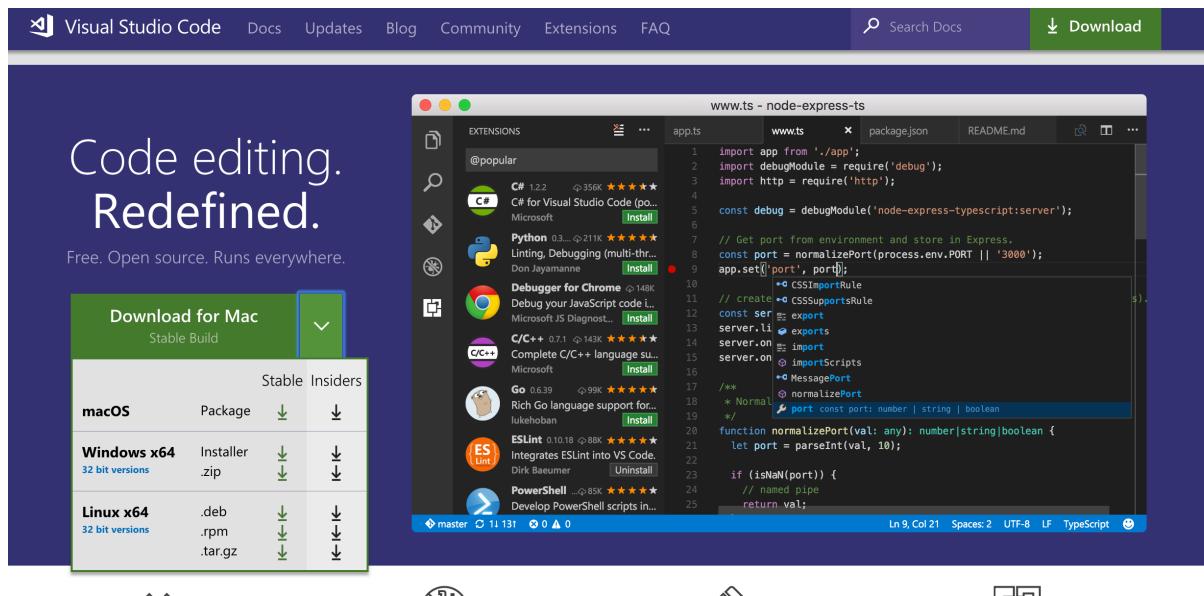
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

การติดตั้ง

VISUAL STUDIO CODE

การติดตั้ง VSC

- เข้าไปที่ website <https://code.visualstudio.com/>
- เลือก download สำหรับ windows (stable)



การติดตั้ง

KONG

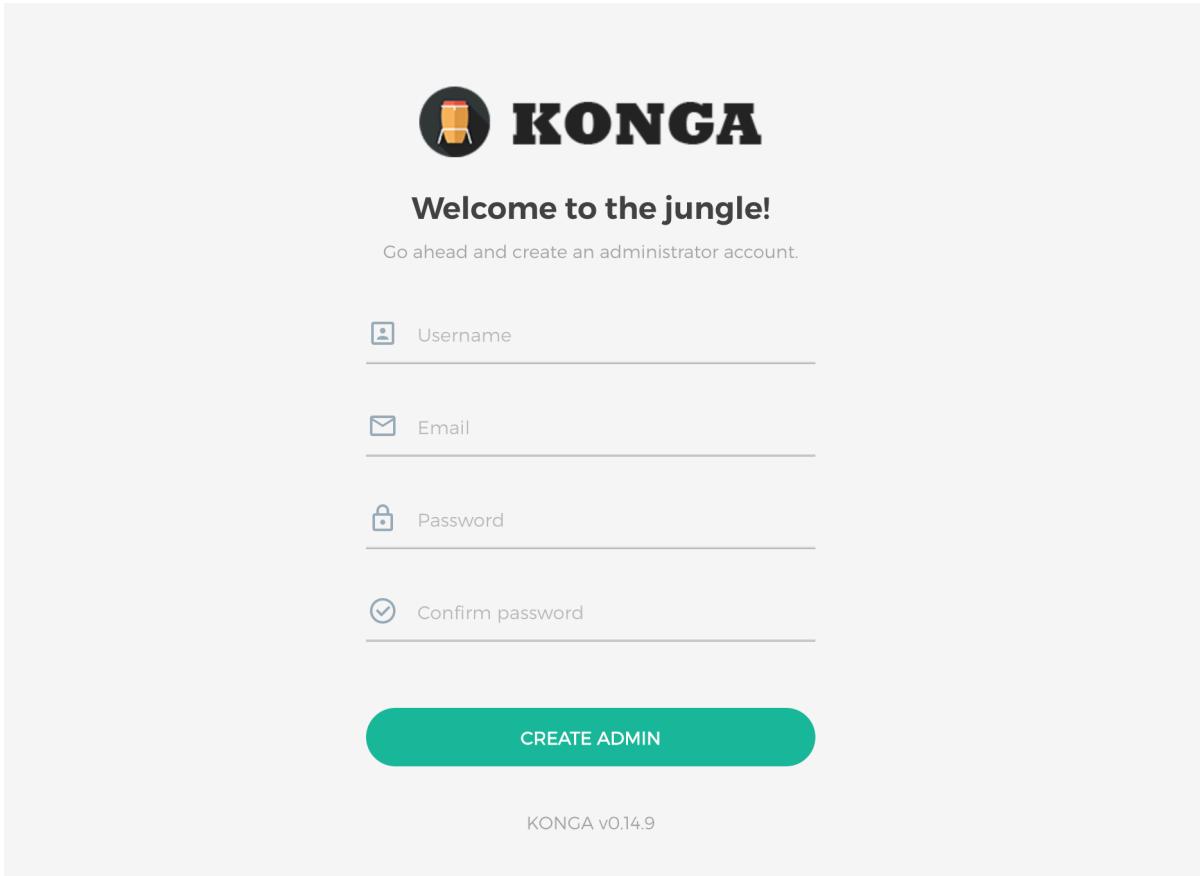
Step

- git clone https://gitlab.com/devops1174/kong.git
- cd kong
- cd kong-server
- docker-compose up -d

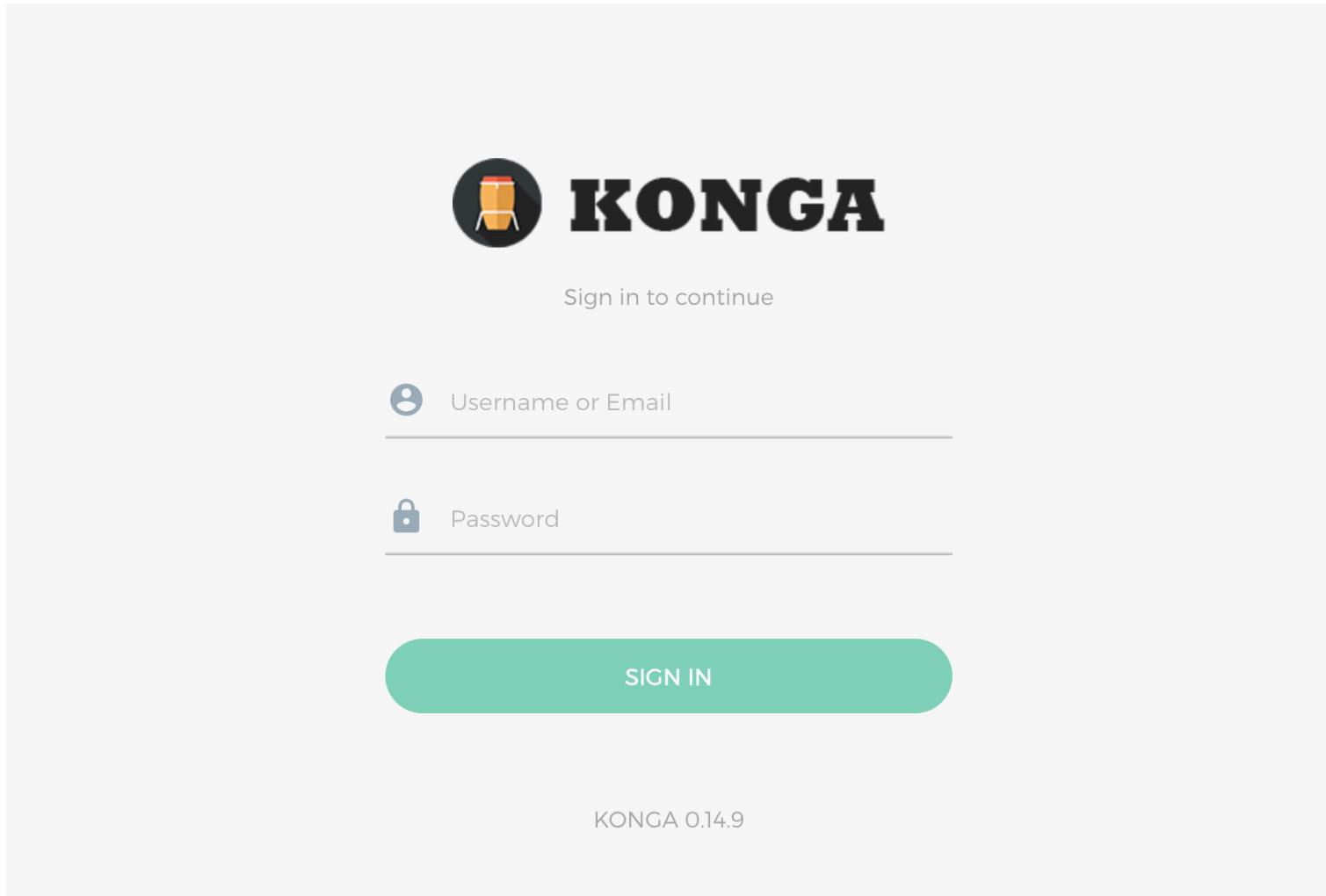
```
Creating network "kong-server_kong-net" with driver "bridge"
Creating kong-server_kong-database_1 ... done
Creating kong-server_konga_1      ... done
Creating kong-server_konga-prepare_1 ... done
Creating kong-server_kong-migration_1 ... done
Creating kong-server_kong_1        ... done
```

Konga Register Admin

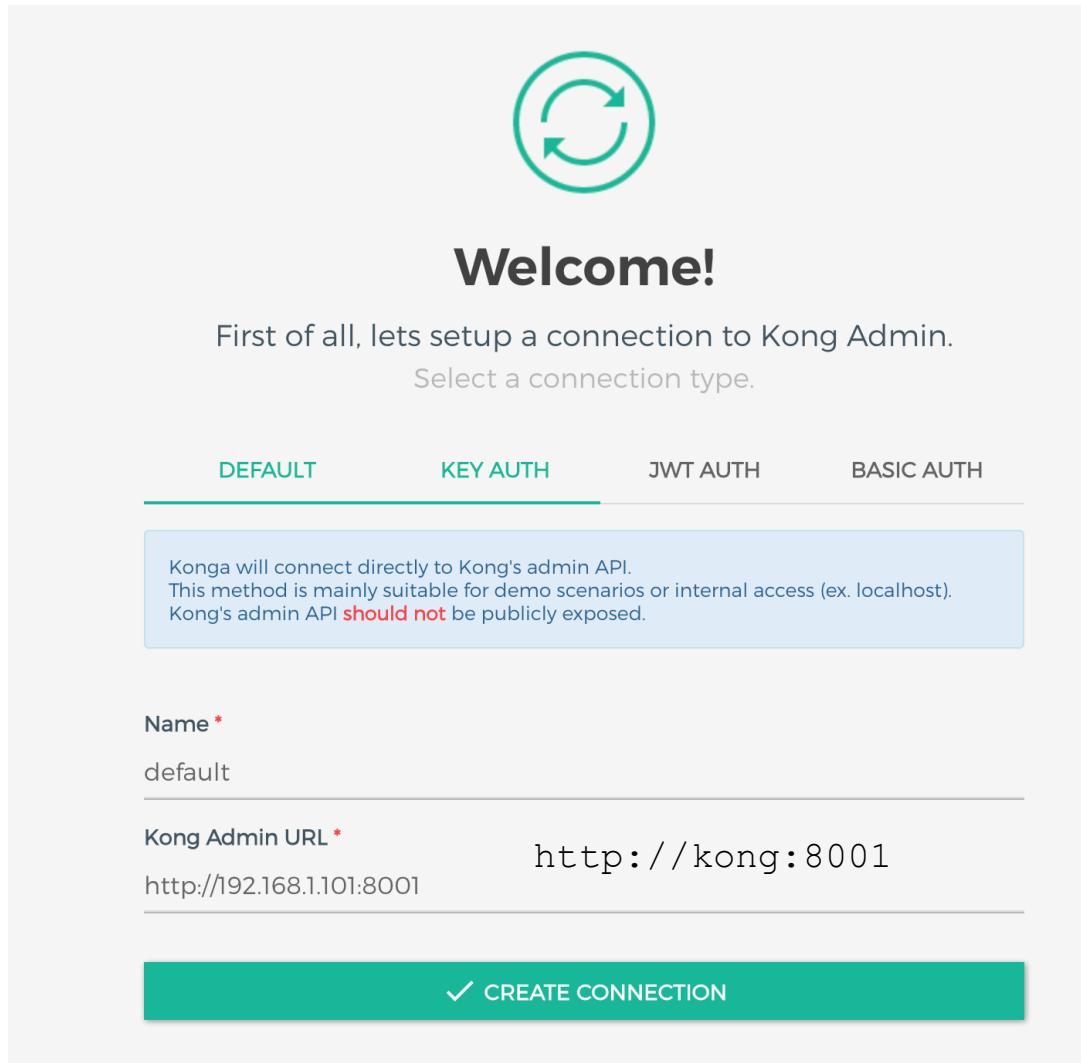
- <http://{your-server-ip}:1337/>



Konga Login



Config Kong Admin URL



The image shows a screenshot of a web application interface for configuring a Kong Admin connection. At the top center is a large teal circular icon with two arrows forming a circle. Below it, the word "Welcome!" is displayed in a large, bold, dark font. Underneath "Welcome!", there is a message: "First of all, let's setup a connection to Kong Admin." followed by "Select a connection type." Below this, there are four tabs: "DEFAULT" (which is underlined in teal), "KEY AUTH", "JWT AUTH", and "BASIC AUTH". A callout box highlights the "DEFAULT" tab, containing the text: "Konga will connect directly to Kong's admin API. This method is mainly suitable for demo scenarios or internal access (ex. localhost). Kong's admin API **should not** be publicly exposed." Below the tabs, there are two input fields: "Name *" with the value "default" and "Kong Admin URL *" with the value "http://kong:8001" (with "http://192.168.1.101:8001" visible below it). At the bottom is a teal button with a white checkmark and the text "CREATE CONNECTION".

Welcome!

First of all, let's setup a connection to Kong Admin.

Select a connection type.

DEFAULT KEY AUTH JWT AUTH BASIC AUTH

Konga will connect directly to Kong's admin API.
This method is mainly suitable for demo scenarios or internal access (ex. localhost).
Kong's admin API **should not** be publicly exposed.

Name *

default

Kong Admin URL *

http://kong:8001

http://192.168.1.101:8001

✓ CREATE CONNECTION

Introduction to

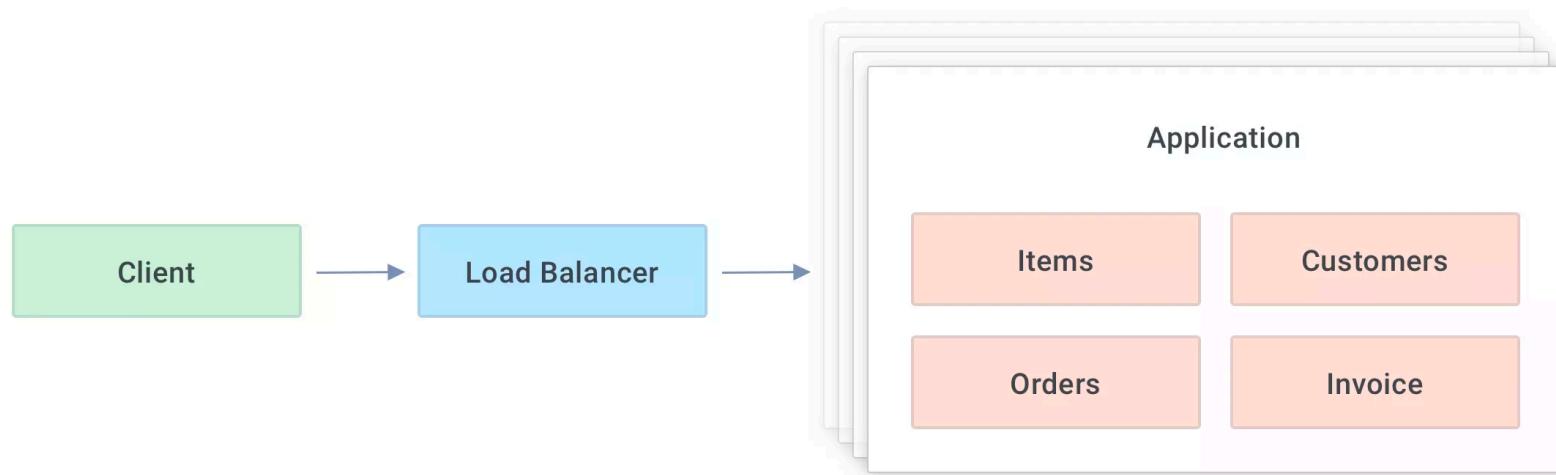
KONG API GATEWAY

Why Use Kong Gateway

- One of the reasons that companies like UnitedHealth Group/Optum deploy Kong Gateway is to modernize a monolithic application and make it more scalable.
- For example, in the application diagram below, if I wanted to scale up the “Items” service to handle more load, I would have to scale up the whole application. Instead, I want to break the application into smaller pieces called modular services or microservices. Then I can scale each service as needed, distribute the service close to its data source and focus on building that service’s business logic.

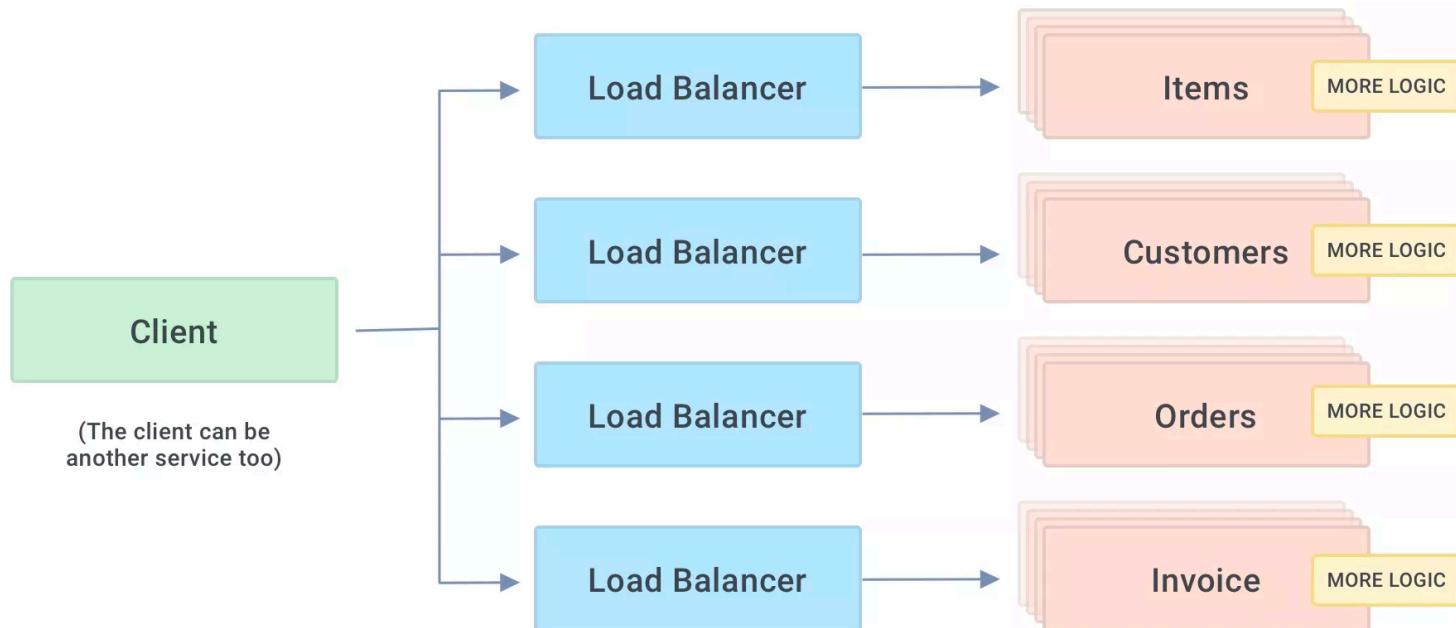
Monolith

Monolith



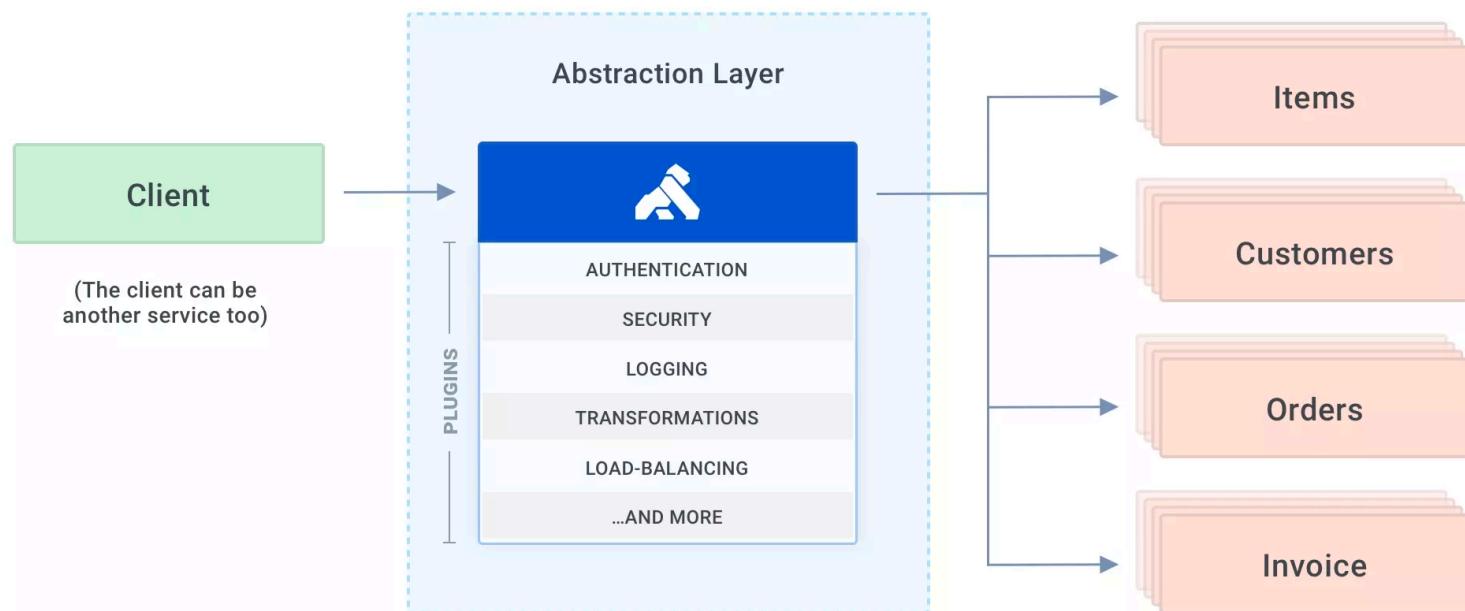
Microservices

Microservices



API Gateway

Centralizing Common Policy



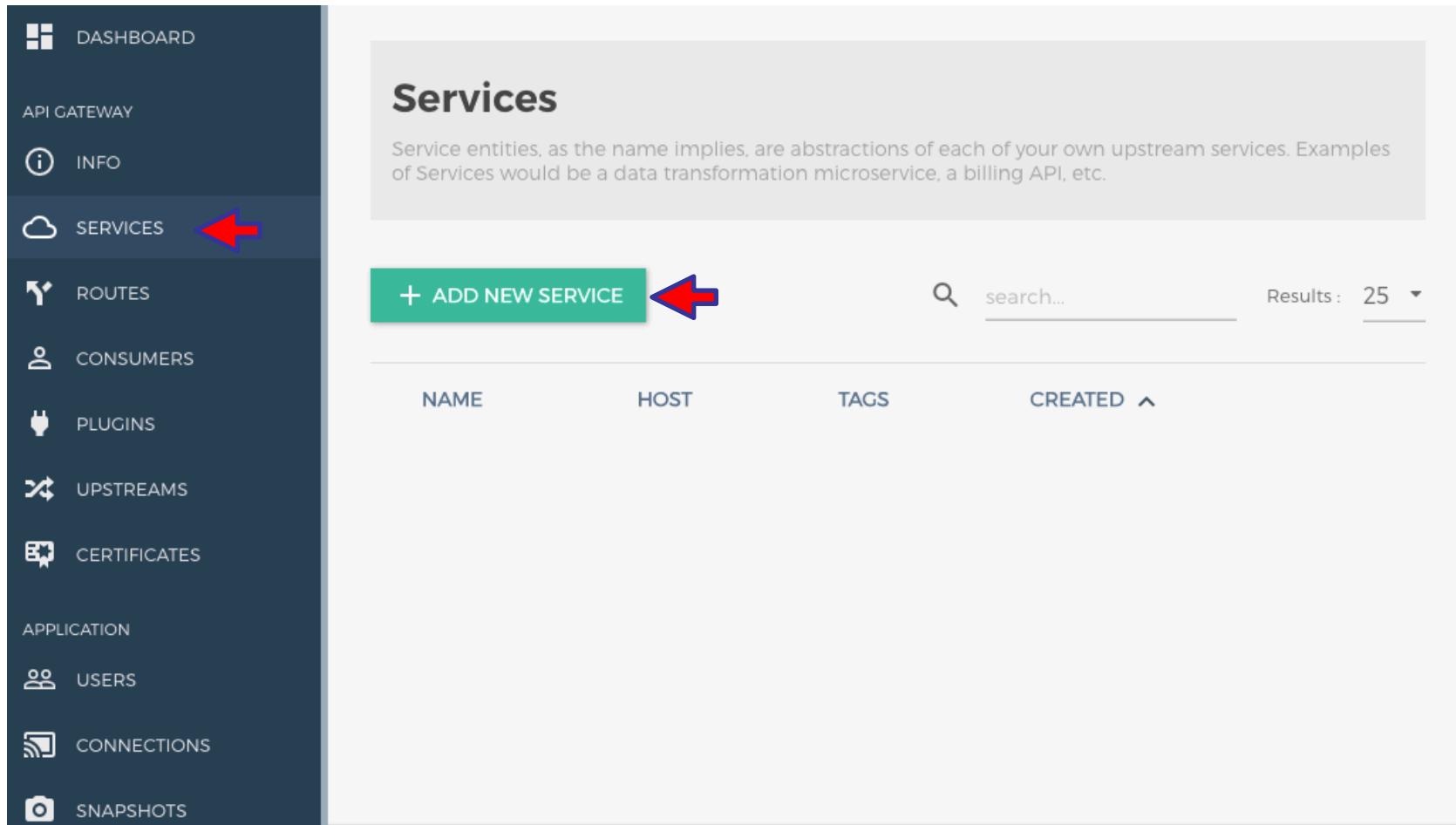
Kong API Gateway

WORKSHOP

How to add

SERVICE

Step 1 / 3



The screenshot shows the 'Services' page of a cloud-based management interface. On the left, a dark sidebar lists various service categories: DASHBOARD, API GATEWAY, INFO, SERVICES (with a red arrow pointing to it), ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, and SNAPSHOTs. The main content area is titled 'Services' and contains a brief description: 'Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.' Below this is a search bar with a magnifying glass icon and the placeholder 'search...'. To the right of the search bar is a dropdown menu showing 'Results : 25'. At the top of the main area is a green button labeled '+ ADD NEW SERVICE' with a red arrow pointing to it. Below this button is a table header with columns: NAME, HOST, TAGS, and CREATED (with an upward arrow). The table body is currently empty.

Step 2 / 3

CREATE SERVICE



Name <i>(optional)</i>	api 
The service name.	
Description <i>(optional)</i>	
An optional service description.	
Tags <i>(optional)</i>	
Optionally add tags to the service	
Url <i>(shorthand-attribute)</i>	
Shorthand attribute to set <code>protocol</code> , <code>host</code> , <code>port</code> and <code>path</code> at once. This attribute is write-only (the Admin API never "returns" the url).	
Protocol <i>(semi-optional)</i>	http 
The protocol used to communicate with the upstream. It can be one of <code>http</code> or <code>https</code> .	
Host <i>(semi-optional)</i>	192.168.1.101 
The host of the upstream server.	
Port <i>(semi-optional)</i>	3030 
The upstream server port. Defaults to <code>80</code> .	

Step 3 / 3

Services

Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.

[+ ADD NEW SERVICE](#)

Results : 25

NAME	HOST	TAGS	CREATED	
 api	192.168.1.101		Apr 16, 2022	 DELETE

How to add

ROUTE

Step 1 / 3

Services

Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.

+ ADD NEW SERVICE

search... Results : 25

NAME	HOST	TAGS	CREATED	
 api 	192.168.1.101		Apr 16, 2022	 DELETE

Step 2 / 3

Service simple

services / show

Service Details

Routes 

Plugins

Eligible consumers beta

Routes  + ADD ROUTE

search routes...

Name / ID	Hosts	Paths	Protocols	Methods	Regex priority	Created
no data found...						

Step 3 / 3

ADD ROUTE TO API



* For hosts, paths, methods and protocols, snis, sources, headers and destinations press enter to apply every value you type

Name
(optional)



The name of the Route.

Tags
(optional)

Optionally add tags to the route

Hosts
(semi-optional)

A list of domain names that match this Route. For example: example.com. At least one of hosts, paths, or methods must be set.

Paths
(semi-optional)



A list of paths that match this Route. For example: /my-path. At least one of **hosts**, **paths**, or **methods** must be set.

Headers
(semi-optional)

One or more lists of values indexed by header name that will cause this Route to match if present in the request. The **Host** header cannot be used with this attribute: hosts should be specified using the **hosts** attribute.

Field values format example: x-some-header:foo,bar

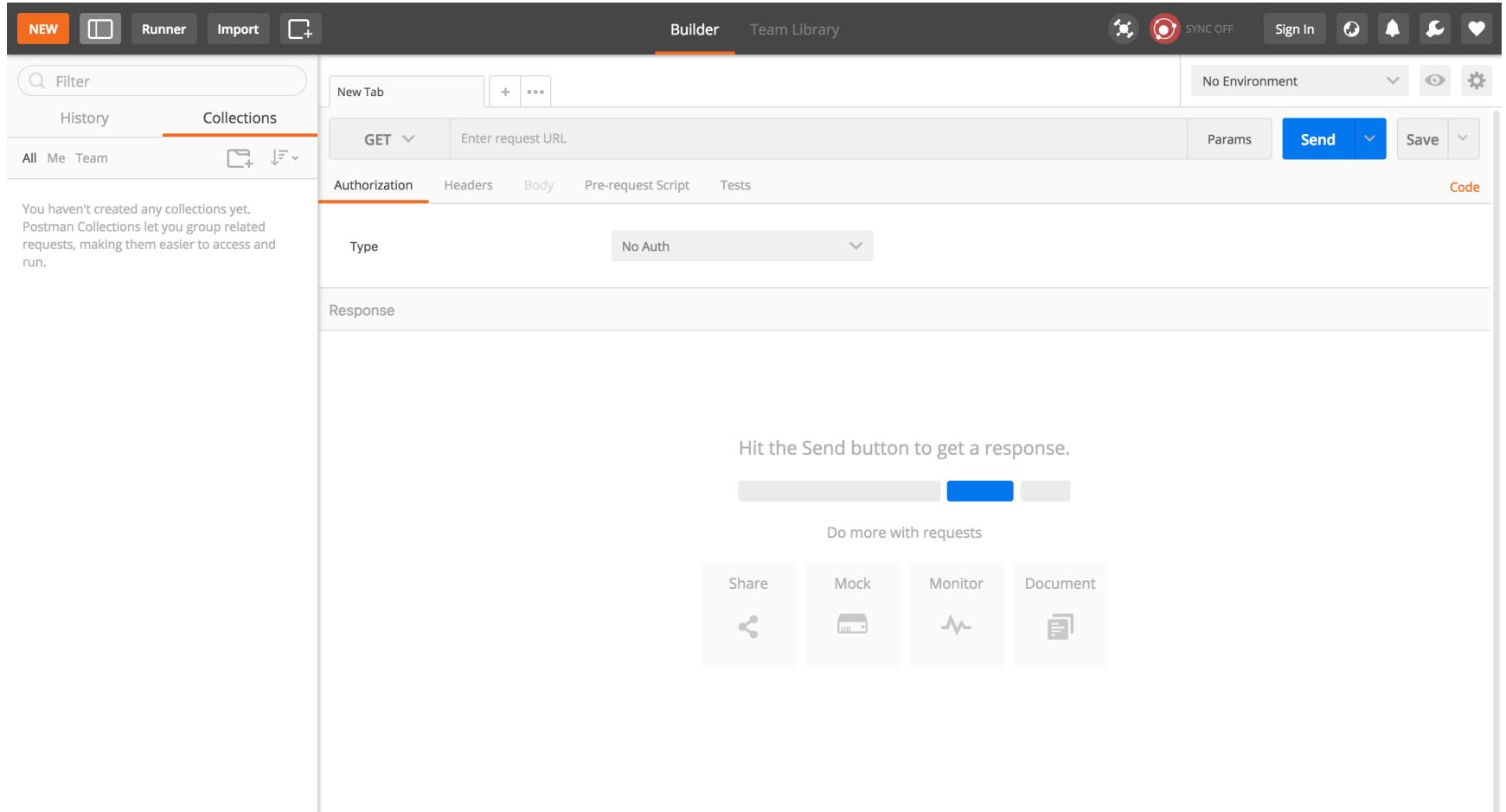
Path handling

Controls how the Service path, Route path and requested path are combined when sending a request to the upstream. See above for a detailed description of each behavior. Accepted values are: "v0", "v1". Defaults to "v1".

Test rest api with

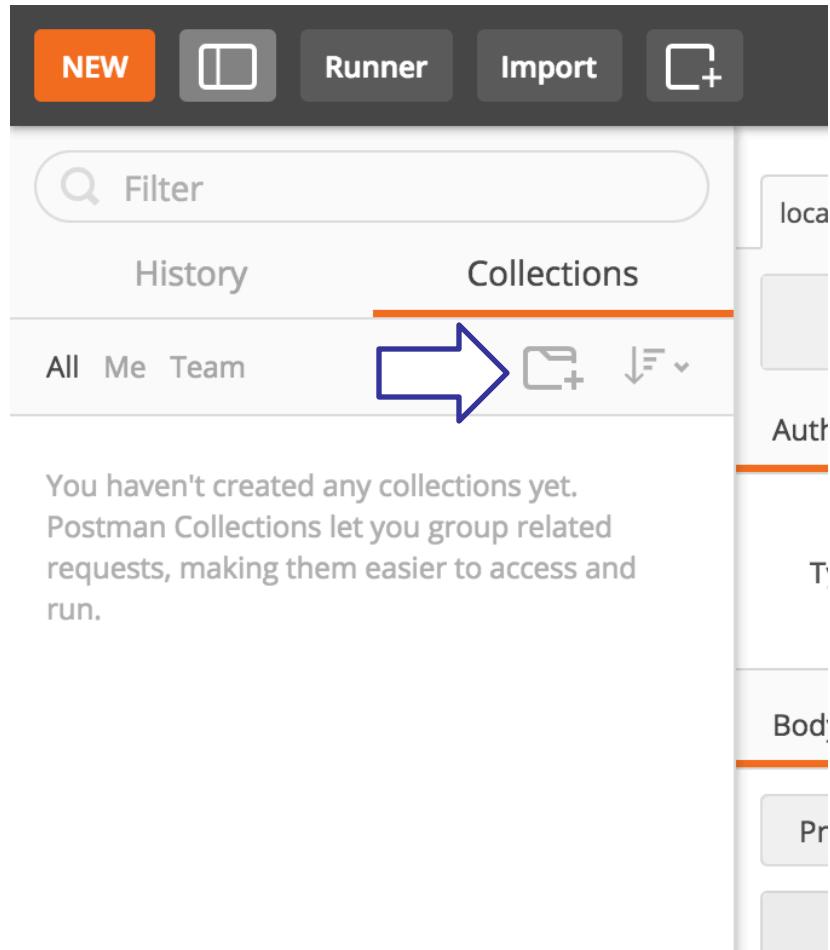
POSTMAN

POSTMAN MAIN SCREEN



The screenshot shows the Postman application interface. At the top, there is a navigation bar with buttons for NEW, Runner, Import, and a tab labeled "Builder". To the right of the navigation bar are icons for Sync (OFF), Sign In, and various notifications. Below the navigation bar, there is a search bar with a "Filter" placeholder and a "Collections" tab selected. On the left side, there is a sidebar with tabs for History, Collections, All, Me, and Team. A message indicates that no collections have been created yet. The main workspace is titled "New Tab" and contains a request builder for a "GET" method. The "Authorization" tab is selected, showing "No Auth" as the type. Below the request builder, there is a large text area with the placeholder "Hit the Send button to get a response." At the bottom of the workspace, there are buttons for "Share", "Mock", "Monitor", and "Document", each accompanied by a small icon.

Create new collection



The screenshot shows the Postman application interface. At the top, there are five buttons: NEW (orange), a document icon, Runner, Import, and a folder icon with a plus sign. Below this is a search bar with a magnifying glass icon and the word 'Filter'. Underneath the search bar are tabs for 'History' and 'Collections', with 'Collections' being the active tab. Below the tabs are buttons for 'All', 'Me', and 'Team'. To the right of these buttons is a large blue arrow pointing towards the 'Create' button in the 'CREATE A NEW COLLECTION' dialog. The main area displays the message: 'You haven't created any collections yet. Postman Collections let you group related requests, making them easier to access and run.' On the far right, there is a sidebar with sections labeled 'Auth', 'Body', and 'Pr'.

bst:3000/user

CREATE A NEW COLLECTION

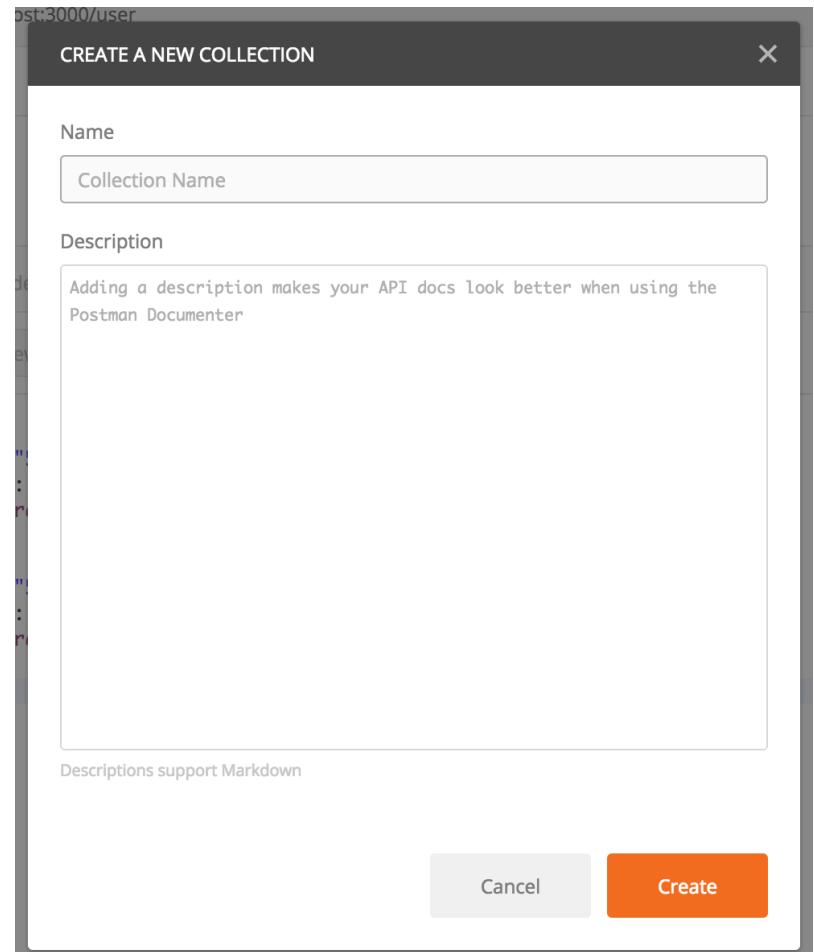
Name

Description

Adding a description makes your API docs look better when using the Postman Documenter

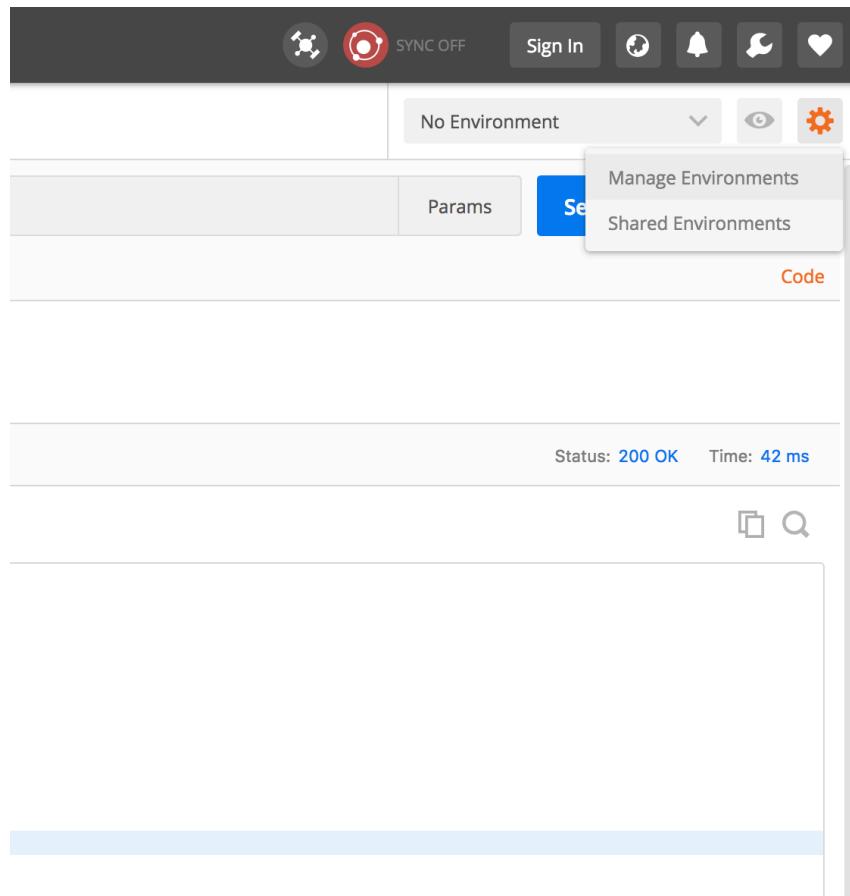
Descriptions support Markdown

Cancel **Create**

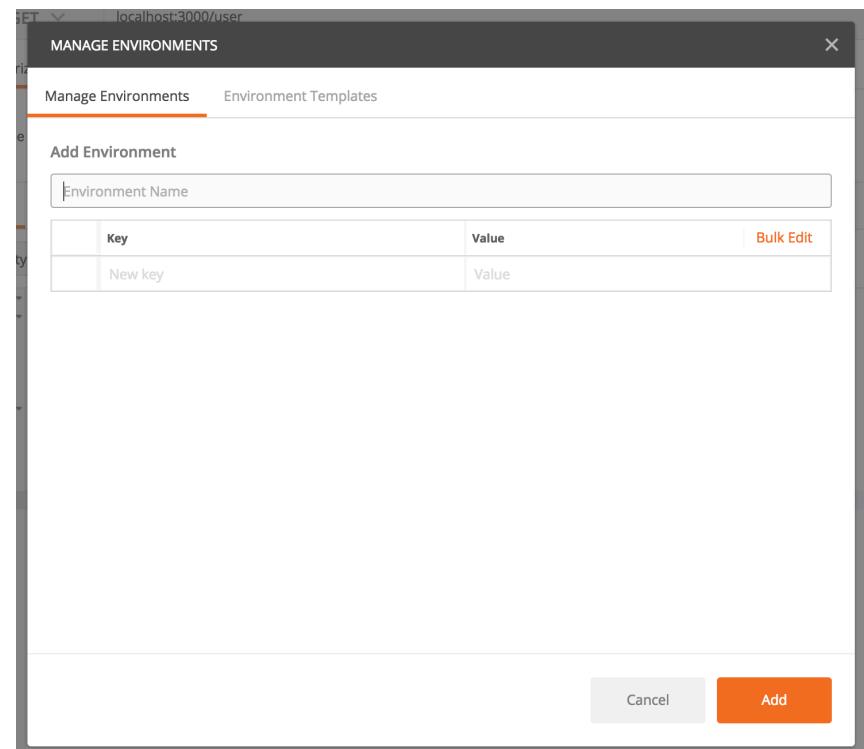


This is a modal dialog titled 'CREATE A NEW COLLECTION'. It contains two input fields: 'Name' and 'Description'. The 'Name' field is empty. The 'Description' field contains the text: 'Adding a description makes your API docs look better when using the Postman Documenter'. Below the description field is a note: 'Descriptions support Markdown'. At the bottom of the dialog are two buttons: 'Cancel' and 'Create'.

Manage Environments



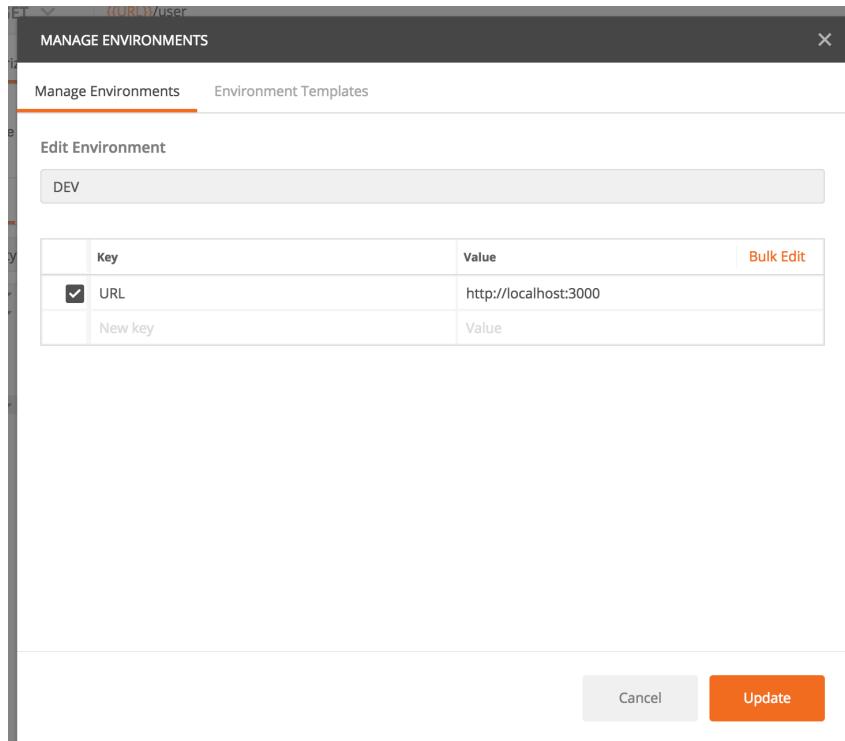
This screenshot shows a web-based application interface for managing environments. At the top, there is a navigation bar with icons for settings, sync, sign in, and notifications. Below the navigation bar, a main header displays "No Environment". A dropdown menu is open, showing options: "Manage Environments" (which is highlighted in blue), "Shared Environments", and "Code". The "Manage Environments" option is currently selected. The main content area is titled "Manage Environments" and contains a sub-section for "Add Environment". It includes a text input field for "Environment Name" and a table for defining environment variables. The table has columns for "Key" and "Value", with a single entry: "New key" and "Value". At the bottom right of the modal are "Cancel" and "Add" buttons. The status bar at the bottom of the screen indicates "Status: 200 OK" and "Time: 42 ms".



This screenshot shows a detailed view of the "Manage Environments" dialog box. The title bar says "MANAGE ENVIRONMENTS" and the URL "localhost:3000/user". The dialog is divided into two tabs: "Manage Environments" (which is active) and "Environment Templates". The "Manage Environments" tab contains a section for "Add Environment" with a text input field for "Environment Name". Below this is a table for defining environment variables. The table has columns for "Key" and "Value", with a single entry: "New key" and "Value". At the bottom right of the dialog are "Cancel" and "Add" buttons. The status bar at the bottom of the screen indicates "Status: 200 OK" and "Time: 42 ms".

Manage Environments

- อ้างถึงตัวแปรใน environment ด้วยคำสั่ง {{ชื่อตัวแปร}}

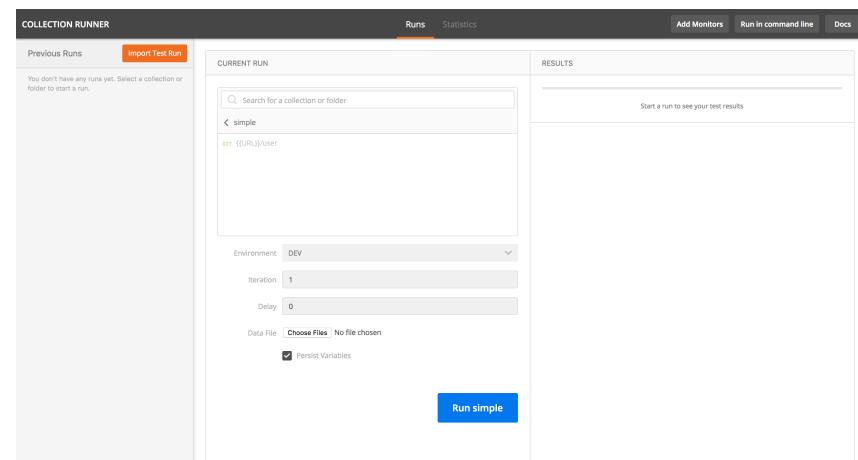
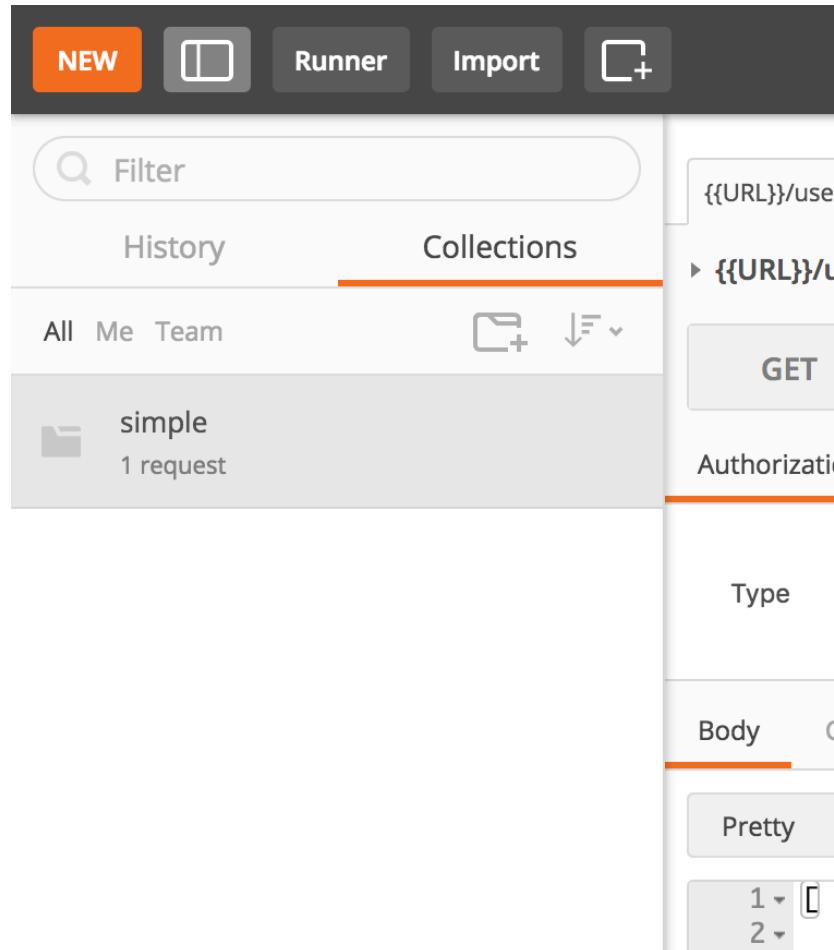


The screenshot shows the Postman 'Builder' tab. A blue arrow points from the 'Manage Environments' dialog to the 'Authorization' section of the builder. The 'Authorization' dropdown is set to 'No Auth'. The 'Body' tab shows a JSON response:

```
1 ↴ [ { "_id": "Saf42a0dd148c80016cd1ce2", "email": "xxx@test.com", "password": "admin" } ]
```

The status bar at the bottom right indicates 'Status: 200 OK'.

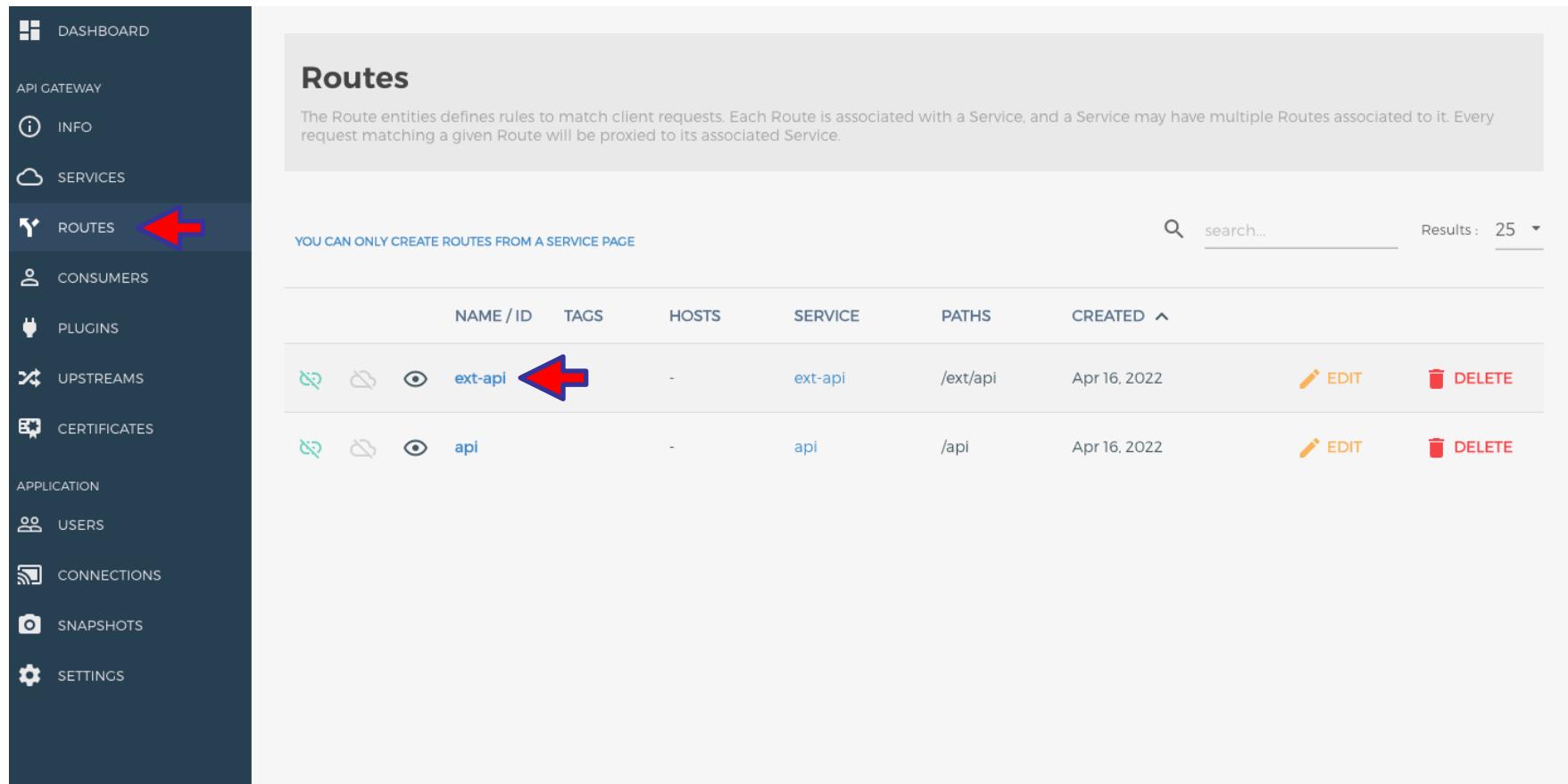
POSTMAN RUNNER



How to add

BASIC AUTH PLUGIN

Step 1 / 4



The screenshot shows the 'Routes' section of an API Gateway interface. On the left, a sidebar lists various management options like Dashboard, API Gateway, Services, and Routes. The 'Routes' option is selected and highlighted with a blue arrow. The main area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this, a message says 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table lists two routes:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4

Route ext-api

routes / route

Route Details

Plugins 

Eligible consumers   + ADD PLUGIN

Assigned plugins

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

ADD PLUGIN X

Authentication ←

Security

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

» AUTHENTICATION

Protect your services with an authentication layer

Basic Auth



Add Basic Authentication to your...

ADD PLUGIN ←

Key Auth



Add a key authentication to your...

ADD PLUGIN

Oauth2



Add an OAuth 2.0 authentication to your...

ADD PLUGIN

Hmac Auth



ADD PLUGIN

Jwt



ADD PLUGIN

Ldap Auth



ADD PLUGIN

Step 4 / 4

ADD BASIC AUTH



Add Basic Authentication to your APIs, with username and password protection. The plugin will check for valid credentials in the `Proxy-Authorization` and `Authorization` header (in this order).

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

anonymous

hide credentials

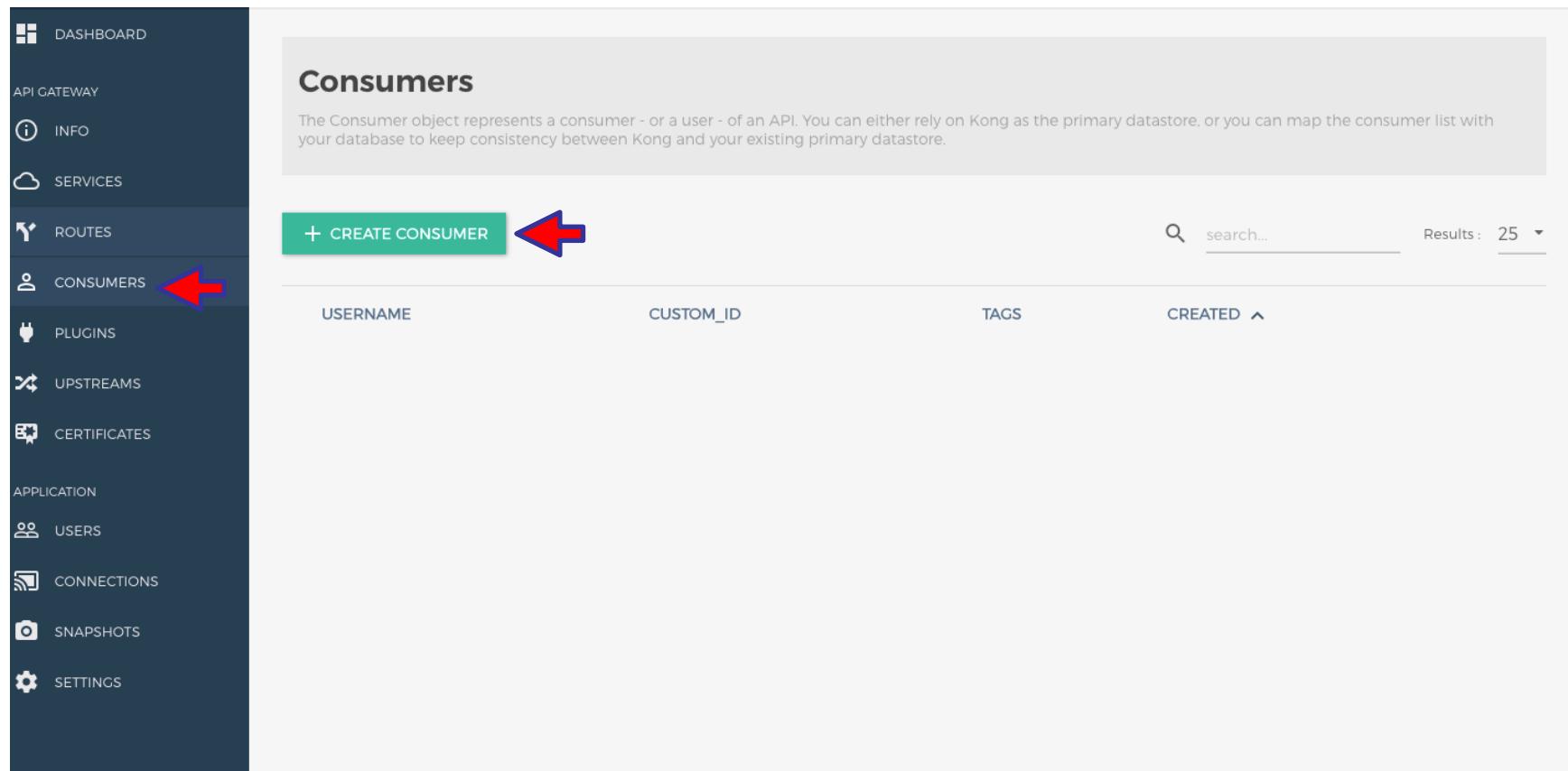
 NO

An optional boolean value telling the plugin to hide the credential to the upstream API server. It will be removed by Kong before proxying the request

 ADD PLUGIN

How to use basic auth with
CONSUMERS

Step 1 / 5



The screenshot shows the Kong API Manager interface. On the left, a dark sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS (which has a red arrow pointing to it), PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled "Consumers". It contains a brief description: "The Consumer object represents a consumer - or a user - of an API. You can either rely on Kong as the primary datastore, or you can map the consumer list with your database to keep consistency between Kong and your existing primary datastore." Below this is a search bar and a results count of "Results : 25". A green button labeled "+ CREATE CONSUMER" is highlighted with a red arrow. The table below has columns: USERNAME, CUSTOM_ID, TAGS, and CREATED (with an upward arrow). There are no rows of data in the table.

Step 2 / 5

CREATE CONSUMER



username
(semi-optional)

imc 

The username of the consumer. You must send either this field or **custom_id** with the request.

custom_id
(semi-optional)

Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or **username** with the request.

Tags
(optional)

Optionally add tags to the consumer

 SUBMIT CONSUMER

Step 3 / 5

CONSUMER: imc

[consumers](#) / edit consumer

Details Groups Credentials Accessible Routes Plugins

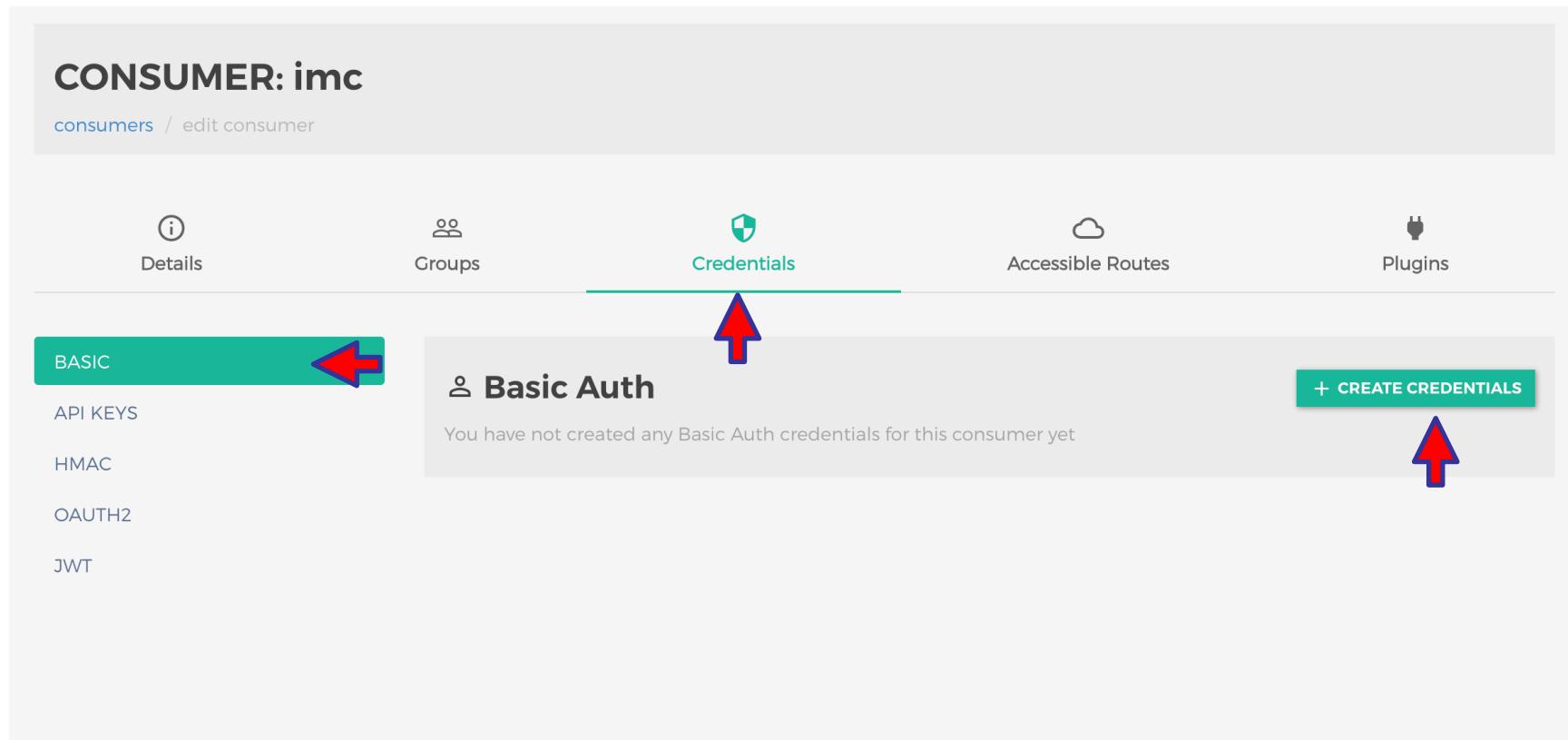
BASIC 

 Basic Auth

+ CREATE CREDENTIALS 

You have not created any Basic Auth credentials for this consumer yet

API KEYS HMAC OAUTH2 JWT



Step 4 / 5

BASIC AUTH

Manage Basic Auth credentials for **imc**

username
(required)

The username to use in the Basic Authentication

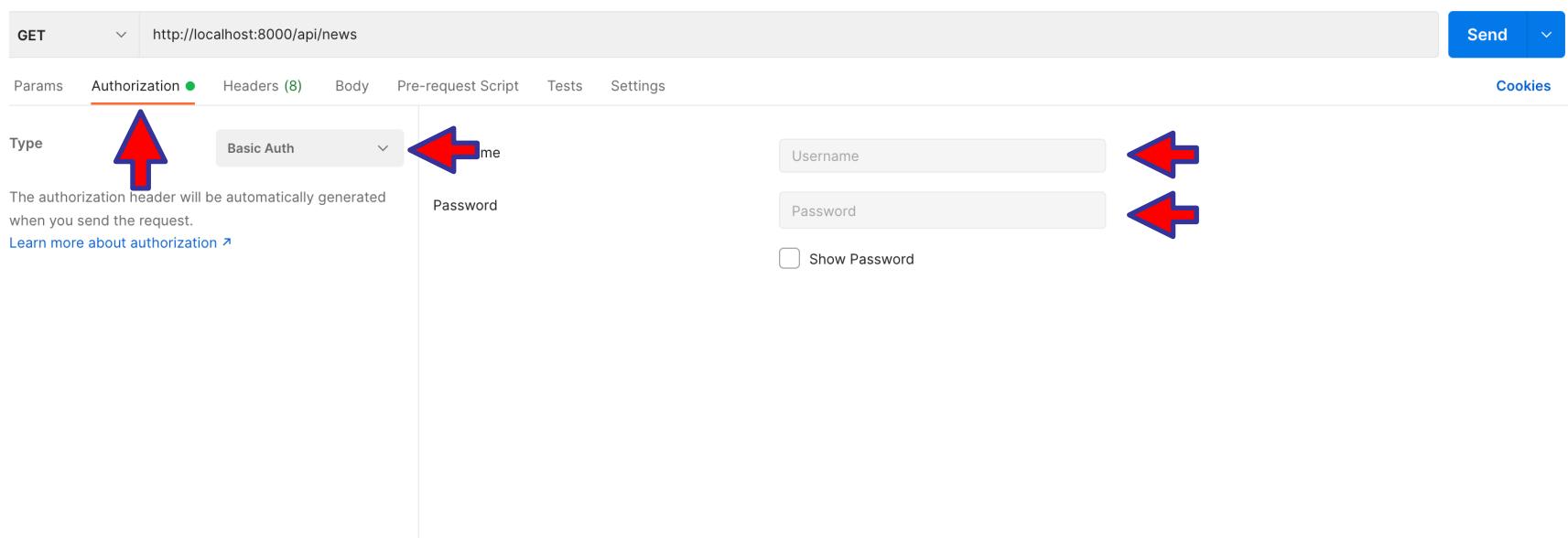
password
(optional)

The password to use in the Basic Authentication

 **SUBMIT**



Step 5 / 5



The screenshot shows the Postman interface for a GET request to `http://localhost:8000/api/news`. The 'Authorization' tab is selected. A red arrow points to the 'Type' dropdown, which is set to 'Basic Auth'. Another red arrow points to the 'Username' field, which contains 'me'. A third red arrow points to the 'Password' field. Below the fields, there is a checkbox labeled 'Show Password'.

GET http://localhost:8000/api/news

Params Authorization ● Headers (8) Body Pre-request Script Tests Settings Cookies

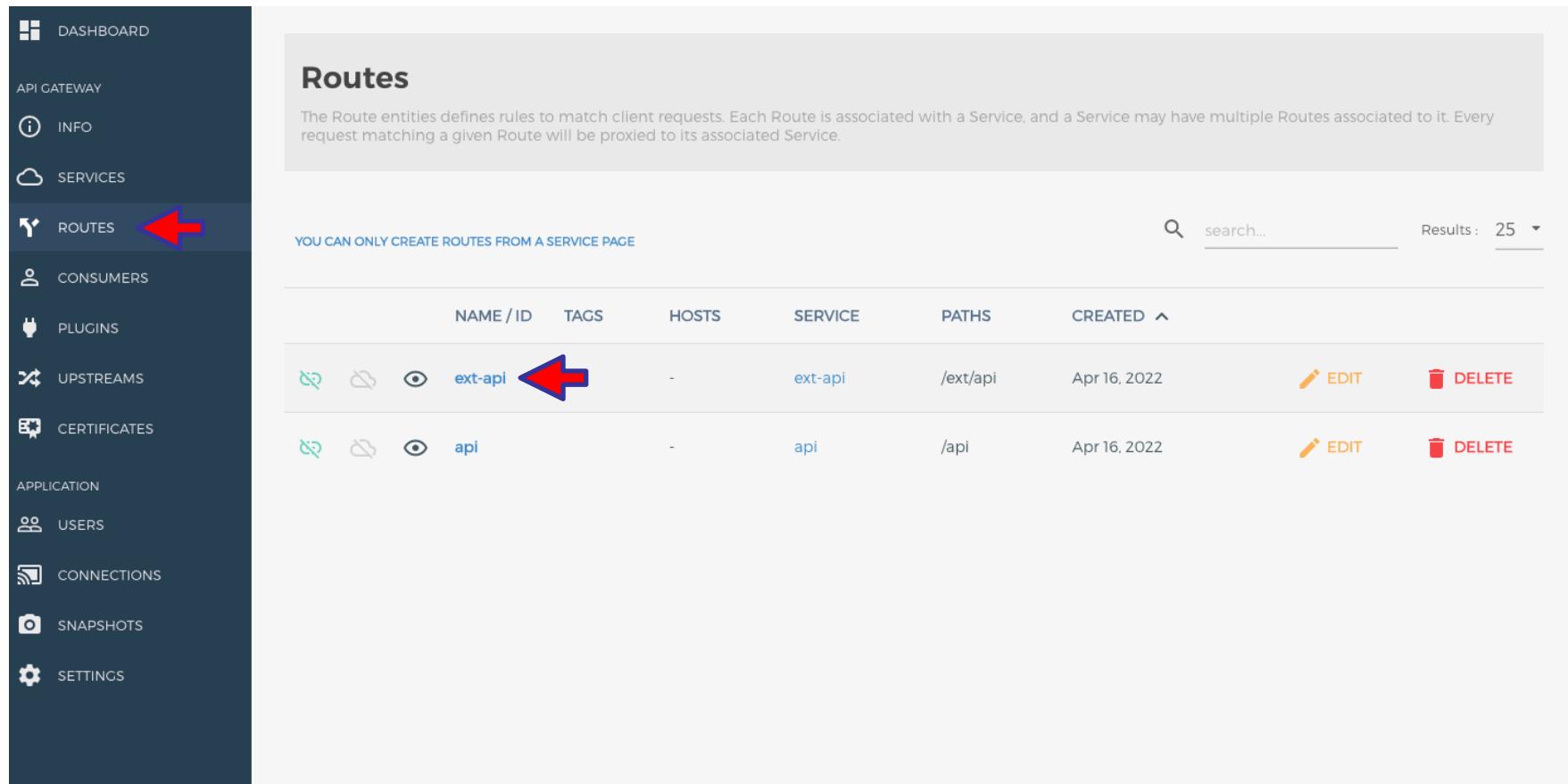
Type Basic Auth Username me Password Show Password

The authorization header will be automatically generated when you send the request.
[Learn more about authorization ↗](#)

How to add

KEY AUTH PLUGIN

Step 1 / 4



The screenshot shows the 'Routes' section of an API Gateway interface. On the left, a sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES (highlighted with a blue arrow), CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this is a message: 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table displays two rows of route information:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api (highlighted with a red arrow)	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4

Route ext-api

routes / route

Route Details

Plugins 

Eligible consumers  

Assigned plugins

+ ADD PLUGIN 

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

ADD PLUGIN X

Authentication ←

Security

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

» AUTHENTICATION

Protect your services with an authentication layer

Basic Auth



Add Basic Authentication to your...

ADD PLUGIN

Key Auth



Add a key authentication to your...

ADD PLUGIN ←

Oauth2



Add an OAuth 2.0 authentication to your...

ADD PLUGIN

Hmac Auth



Ldap Auth



Jwt



Step 4 / 4

ADD KEY AUTH X

Add Key Authentication (also referred to as an API key) to your APIs. Consumers then add their key either in a querystring parameter or a header to authenticate their requests.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

key names x-api-key X ←

Tip: Press **Enter** to accept a value.
Describes an array of comma separated parameter names where the plugin will look for a key. The client must send the authentication key in one of those key names, and the plugin will try to read the credential from a header or the querystring parameter with the same name.

hide credentials NO

An optional boolean value telling the plugin to hide the credential to the upstream API server. It will be removed by Kong before proxying the request.

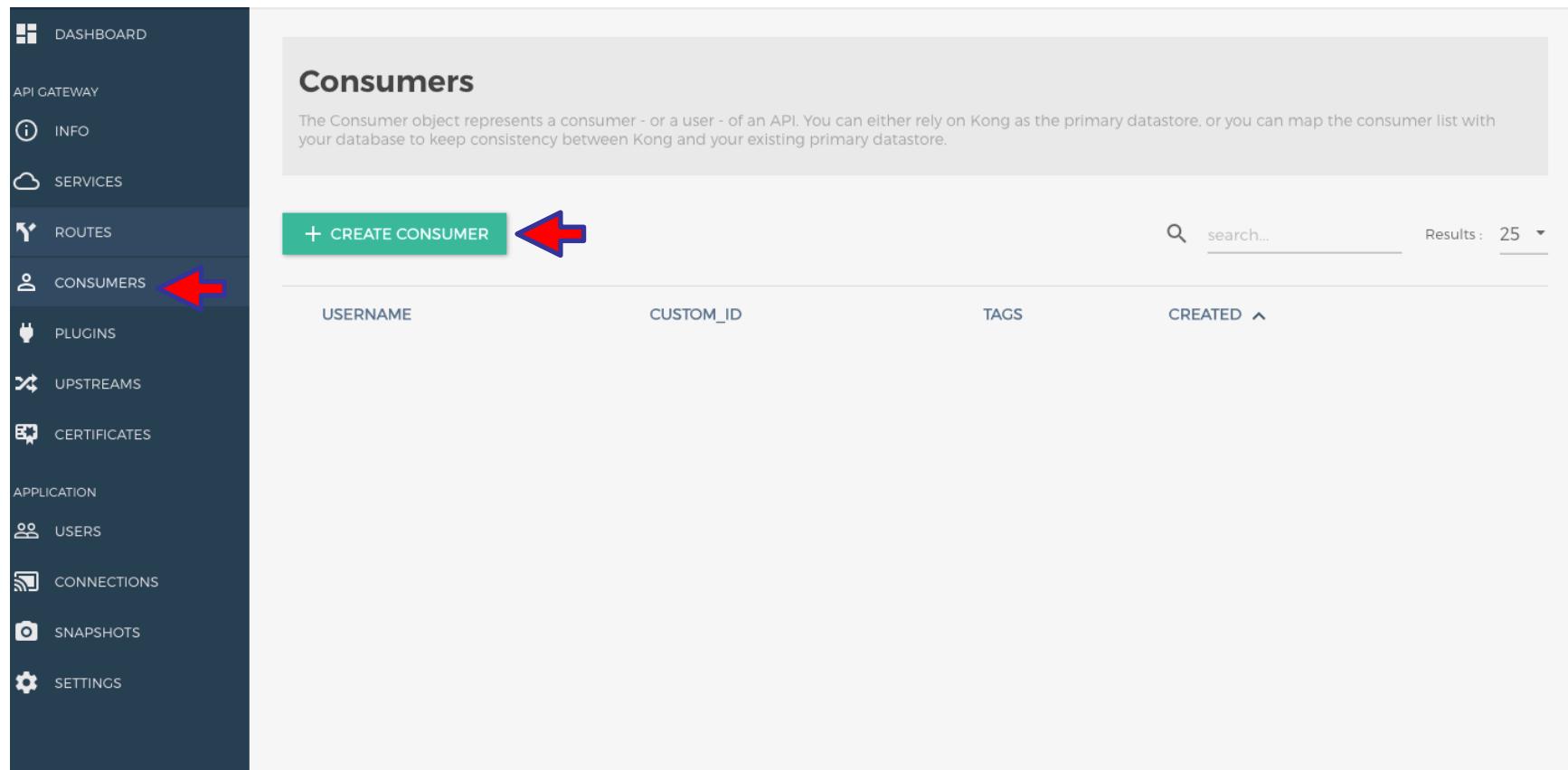
anonymous

key in header YES

key in query YES

How to use key auth with
CONSUMERS

Step 1 / 6



The screenshot shows the Kong API Manager interface. On the left, a dark sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS (which has a red arrow pointing to it), PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled "Consumers". It contains a brief description: "The Consumer object represents a consumer - or a user - of an API. You can either rely on Kong as the primary datastore, or you can map the consumer list with your database to keep consistency between Kong and your existing primary datastore." Below this is a search bar and a results count of "Results : 25". A green button labeled "+ CREATE CONSUMER" is highlighted with a red arrow. The table below has columns: USERNAME, CUSTOM_ID, TAGS, and CREATED (with an upward arrow indicating sorting). There are no rows of data in the table.

Step 2 / 6

CREATE CONSUMER



username
(semi-optional)

imc 

The username of the consumer. You must send either this field or **custom_id** with the request.

custom_id
(semi-optional)

Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or **username** with the request.

Tags
(optional)

Optionally add tags to the consumer

 SUBMIT CONSUMER

Step 3 / 6

CONSUMER: imc

consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

BASIC

API KEYS 

HMAC

OAUT2

JWT

 **Credentials**

 **+ CREATE API KEY**

Api Keys

You have not created any keys for this consumer yet

Step 4 / 6

CREATE API KEY



Create Api Key for **imc**

key
(optional)



You can optionally set your own unique **key** to authenticate the client. If missing, Kong will generate one.

 SUBMIT

Step 5 / 6

CONSUMER: imc

consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

BASIC API KEYS HMAC OAUTH2 JWT

Api Keys

+ CREATE API KEY

#	key	created	
1.	97z777VvK4d1zkXlOcwArzoKCb75n3BT	Apr 16, 2022	 DELETE



Step 6 / 6

http://192.168.1.101:8000/ext/api/news Save ▾

GET ▼ http://192.168.1.101:8000/ext/api/news ←

Params	Authorization	Headers (7)	Body	Pre-request Script	Tests	Settings
<input checked="" type="checkbox"/> host					<small>(i) <calculated when request is sent></small>	
<input checked="" type="checkbox"/> User-Agent					<small>(i) PostmanRuntime/7.29.0</small>	
<input checked="" type="checkbox"/> Accept					<small>(i) */*</small>	
<input checked="" type="checkbox"/> Accept-Encoding					<small>(i) gzip, deflate, br</small>	
<input checked="" type="checkbox"/> Connection					<small>(i) keep-alive</small>	
<input checked="" type="checkbox"/> x-api-key					WZ7ZWQKdhhwaaZpujBPbD4O5Df2hdeb2	←
Key			Value			Description

Body Cookies Headers (10) Test Results Status: 200 OK Time: 215 ms Size: 315 B S

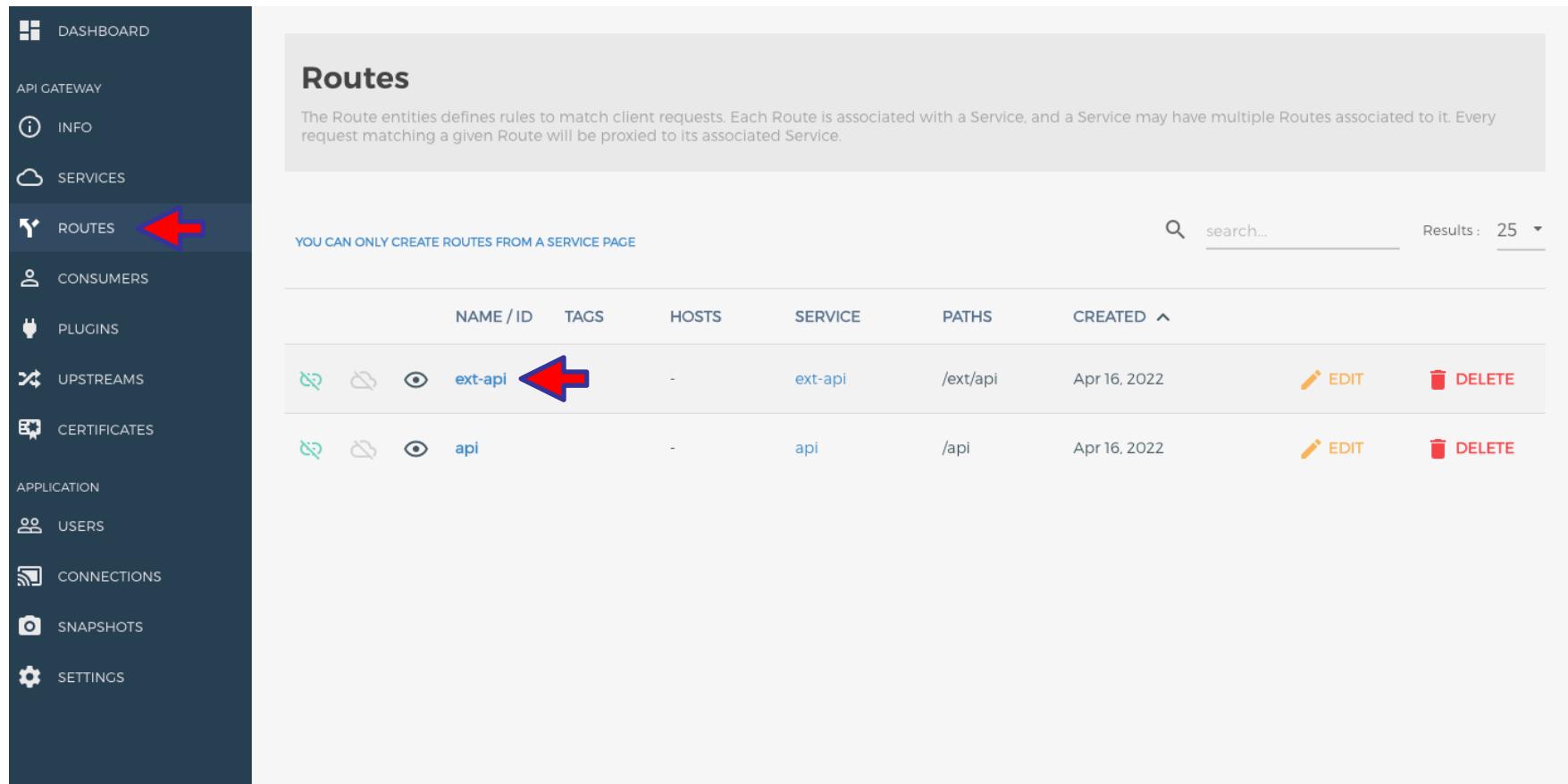
Pretty Raw Preview Visualize JSON ▾ ←

1 []

How to add

OAUTH2 PLUGIN

Step 1 / 4



The screenshot shows the 'Routes' section of an API gateway interface. On the left, a sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES (highlighted with a blue arrow), CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this is a message: 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table displays two routes:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api (highlighted with a red arrow)	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4

Route ext-api

routes / route

Route Details

Plugins 

Eligible consumers  beta

Assigned plugins

+ ADD PLUGIN 

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

ADD PLUGIN

X

Authentication 

Security

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

AUTHENTICATION

Protect your services with an authentication layer

Basic Auth

Add Basic Authentication to your...
ADD PLUGIN

Key Auth

Add a key authentication to your...
ADD PLUGIN

Oauth2

Add an OAuth 2.0 authentication to your...
ADD PLUGIN 

Hmac Auth


Jwt


Ldap Auth


Step 4 / 4

ADD OAUTH2 X

Add an OAuth 2.0 authentication layer with the [Authorization Code Grant](#), [Client Credentials](#), [Implicit Grant](#) or [Resource Owner Password Credentials Grant](#) flow. This plugin requires the [SSL Plugin](#) with the `only_https` parameter set to `true` to be already installed on the API, failing to do so will result in a security weakness.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

scopes

`user_profile X` `biometric X` `verify_acc X` ←

Tip: Press [Enter](#) to accept a value.
Describes an array of comma separated scope names that will be available to the end user

mandatory scope

`YES` ←
An optional boolean value telling the plugin to require at least one scope to be authorized by the end user

provision key

token expiration

7200

An optional integer value telling the plugin how long should a token last, after which the client will need to refresh the token. Set to 0 to disable the expiration.

enable authorization code

`YES` ←
An optional boolean value to enable the three-legged Authorization Code flow (RFC 6742 Section 4.1)

Copy Provision Key

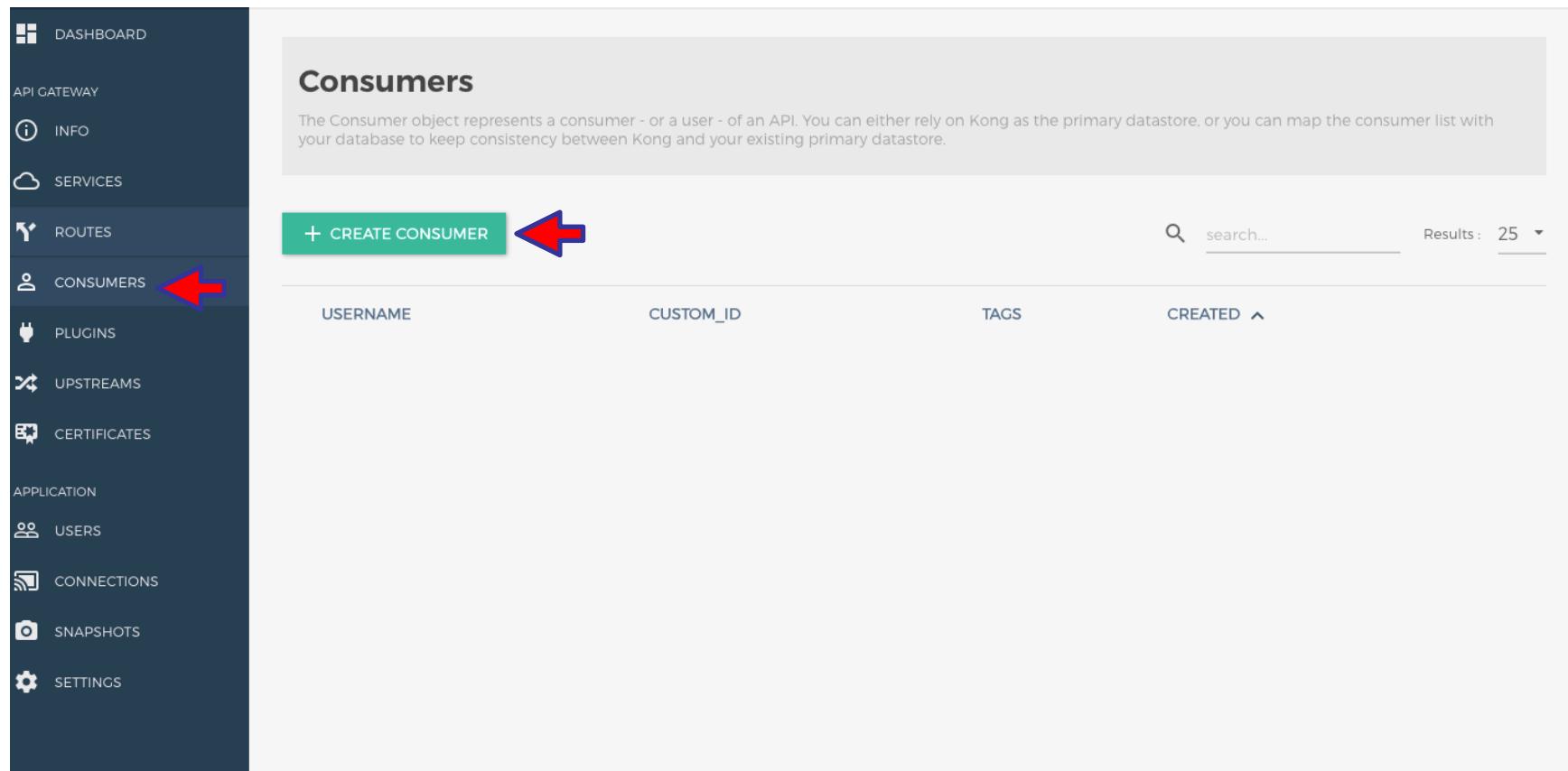
RAW VIEW X

```
{  
    "tags": null,  
    "enabled": true,  
    "created_at": 1679924402,  
    "route": null,  
    "consumer": null,  
    "service": null,  
    "name": "oauth2",  
    "id": "85fde0bf-a79a-443b-9946-d1a66c8ea5b0",  
    "instance_name": null,  
    "protocols": [  
        "grpc",  
        "grpcs",  
        "http",  
        "https"  
    ],  
    "config": {  
        "pkce": "lax",  
        "refresh_token_ttl": 1209600,  
        "global_credentials": false,  
        "reuse_refresh_token": false,  
        "accept_http_if_already_terminated": false,  
        "provision_key": "zbSHLH4MeBRd602q0utILrb2u4UMbqT2",   
        "hide_credentials": false,  
        "anonymous": null,  
        "enable_client_credentials": false  
    }  
}
```

How to use oauth2 with

CONSUMERS

Step 1 / 8



The screenshot shows the Kong API Manager interface. On the left, a dark sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS (which has a red arrow pointing to it), PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled "Consumers". It contains a brief description: "The Consumer object represents a consumer - or a user - of an API. You can either rely on Kong as the primary datastore, or you can map the consumer list with your database to keep consistency between Kong and your existing primary datastore." Below this is a search bar and a results count of "Results : 25". A green button labeled "+ CREATE CONSUMER" is highlighted with a red arrow. The table below has columns: USERNAME, CUSTOM_ID, TAGS, and CREATED (with an upward arrow indicating sorting). There are no visible rows of data.

Step 2 / 8

CREATE CONSUMER



username
(semi-optional)

imc 

The username of the consumer. You must send either this field or **custom_id** with the request.

custom_id
(semi-optional)

Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or **username** with the request.

Tags
(optional)

Optionally add tags to the consumer

 SUBMIT CONSUMER

Step 3 / 8

CONSUMER: auser

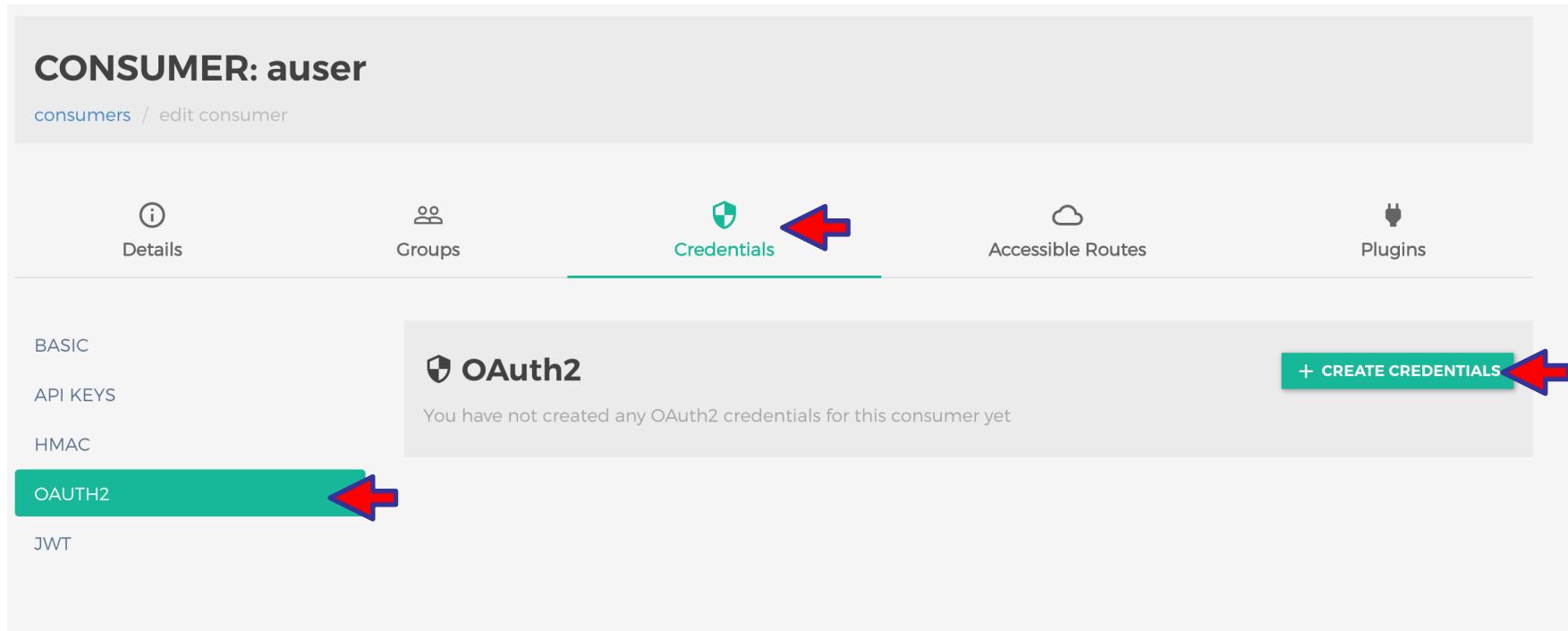
consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

BASIC API KEYS HMAC OAUTH2 JWT

OAuth2
You have not created any OAuth2 credentials for this consumer yet

+ CREATE CREDENTIALS



Step 4 / 8

CREATE OAUTH2 X

Create OAuth2 for **auser**

name (required) bob ←
The name to associate to the credential. In OAuth 2.0 this would be the application name.

client_id (optional)
You can optionally set your own unique **client_id**. If missing, the plugin will generate one.

client_secret (optional)
You can optionally set your own unique **client_secret**. If missing, the plugin will generate one.

redirect_uris (required) https://duckstore.com X ←
Press enter to apply every value you type
The URL in your app where users will be sent after authorization ([RFC 6742 Section 3.1.2](#))

✓ **SUBMIT**

Step 5 / 8

RAW VIEW



```
{  
    "hash_secret": false,  
    "client_id": "W3i5kl5YR2el2nDPBx595iJV7TdDX0kU",   
    "client_secret": "18grgHeeZc0zon0Fb1U5Hyp0ux0GftZZ",   
    "created_at": 1679924688,  
    "consumer": {  
        "id": "ef147477-210a-48f5-9cd6-e902a3e7d61c"  
    },  
    "redirect_uris": null,  
    "client_type": "confidential",  
    "id": "3570b4e1-d409-4aa1-b9b0-b617be284bf2",  
    "tags": null,  
    "name": "auser"  
}
```

Step 6 / 8

POST <https://localhost:8443/api/lb/oauth2/authorize>

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings ●

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾

```
1 {  
2   "client_id": "rBXaciPvUBAPa7JynmvxnBRYPVu9eb4l", ←  
3   "response_type": "code",  
4   "scope": "verify_acc",  
5   "provision_key": "zbSHLH4MeBRd602q0utILrb2u4UMbqT2", ←  
6   "authenticated_userid": "bob" ←  
7 }
```

Body Cookies Headers (8) Test Results  Status

Pretty Raw Preview Visualize JSON ▾

```
1 {  
2   "redirect_uri": "https://duckstore.com?code=hhFPHqM40CmRa2865oUp79MlZYr3gzak" ←  
3 }
```

Step 7 / 8 (Generate)

POST <https://localhost:8443/apilb/oauth2/token>

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings ●

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {  
2   "grant_type": "authorization_code",  
3   "code": "hhFPHqM40CmRa2865oUp79MlZYr3gzak",  
4   "client_id": "rBXaciPvUBAPa7JynmvxnBRYPVu9eb4l",  
5   "client_secret": "1F4gweZ7savk3HJhLfv8lgqfxVJjHrNt"  
6 }
```

Body Cookies Headers (8) Test Results  

Pretty Raw Preview Visualize JSON ▾

```
1 {  
2   "token_type": "bearer",  
3   "access_token": "emHCAz2oWg3Ffl4KURMRl5FD2qzLqXHo",  
4   "refresh_token": "nIdQSrCzNT4fd6QucXoIk8FC5LJKd5nl",  
5   "expires_in": 7200  
6 }
```

Step 7 / 8 (Refresh)

POST <https://localhost:8443/apilb/oauth2/token>

Params Authorization Headers (8) Body • Pre-request Script Tests Settings •

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾

```
1 {  
2   "grant_type": "refresh_token",  
3   "refresh_token": "nIdQSRcZNT4fd6QuCxoIk8FC5LJKd5nl",  
4   "client_id": "rBXaciPvUBAPa7JynmvxnBRYPVu9eb4l",  
5   "client_secret": "1F4gweZ7savk3HJhLfV8lgqfxVJjHrNt"  
6 }
```

Body Cookies Headers (8) Test Results

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {  
2   "token_type": "bearer",  
3   "access_token": "JtMaIK9hPsMT4sBaCMTjJ4a00fImb4FE",  
4   "refresh_token": "1le7UH01cD1iyZI9LiiK65wxa6y2Gg3H",  
5   "expires_in": 7200  
6 }
```

Step 8 / 8

GET ▼ https://192.168.1.101:8443/apilb/news

Params Authorization ● Headers (8) Body Pre-request Script Tests Settings

Type Bearer Token ▼ Token UrlhqhcXqTS7K7Z8ml0RrpE0t456btsV

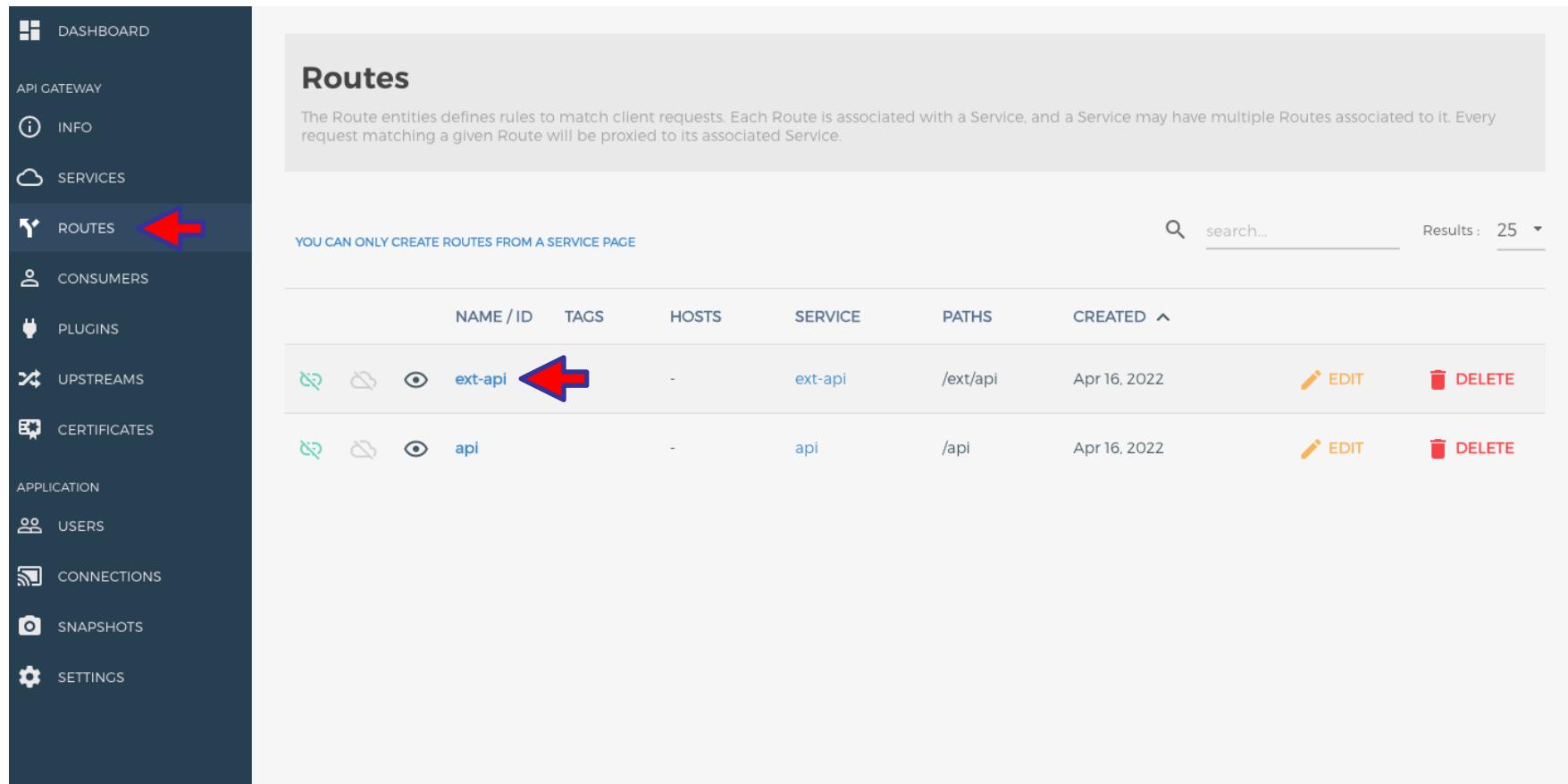
The authorization header will be automatically generated when you send the request. Learn more about [authorization ↗](#)

Body Cookies Headers (10) Test Results 🌐 Status: 200 OK Time: 13 ms Size: 447

How to add

JWT PLUGIN

Step 1 / 4



The screenshot shows the 'Routes' section of an API gateway interface. On the left, a sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES (highlighted with a blue arrow), CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this is a message: 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table displays two routes:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api (highlighted with a red arrow)	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4

Route ext-api

routes / route

Route Details

Plugins 

Eligible consumers   + ADD PLUGIN

Assigned plugins

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

Authentication 

Security

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

👤 AUTHENTICATION

Protect your services with an authentication layer

Basic Auth

Add Basic Authentication to your..
ADD PLUGIN

Key Auth

Add a key authentication to your..
ADD PLUGIN

Oauth2

Add an OAuth 2.0 authentication to your..
ADD PLUGIN

Hmac Auth

Add HMAC Authentication to your..
ADD PLUGIN

Jwt

Verify and authenticate JSON Web Tokens
ADD PLUGIN 

Ldap Auth

Integrate Kong with a LDAP server
ADD PLUGIN

Step 4 / 4

Tip: Press **Enter** to accept a value.

key claim name iss

The name of the claim in which the key identifying the secret must be passed.

NO

If true, the plugin assumes the credential's secret to be base64 encoded. You will need to create a base64 encoded secret for your consumer, and sign your JWT with the original secret.

claims to verify

Tip: Press **Enter** to accept a value.

A list of registered claims (according to RFC 7519) that Kong can verify as well. Accepted values: exp, nbf.

anonymous

run on preflight

maximum expiration 0

header names

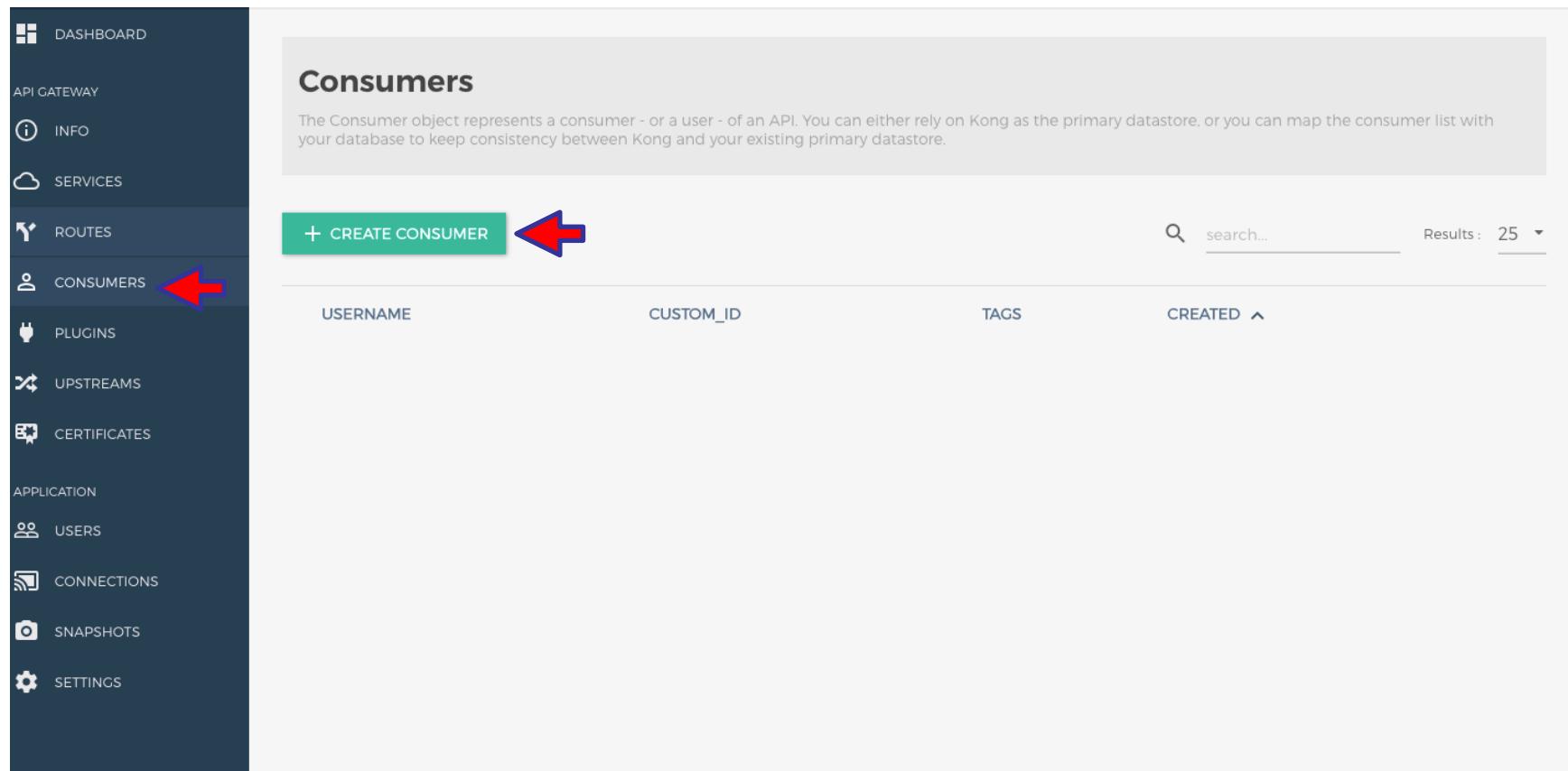
Tip: Press **Enter** to accept a value.

✓ ADD PLUGIN



How to use json with
CONSUMERS

Step 1 / 7



The screenshot shows the Kong API Manager interface. On the left, a dark sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS (which has a red arrow pointing to it), PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled "Consumers". It contains a brief description: "The Consumer object represents a consumer - or a user - of an API. You can either rely on Kong as the primary datastore, or you can map the consumer list with your database to keep consistency between Kong and your existing primary datastore." Below this is a search bar and a results count of "Results : 25". A table header includes columns for USERNAME, CUSTOM_ID, TAGS, and CREATED (with an upward arrow). At the top left of the main content area is a green button labeled "+ CREATE CONSUMER" with a red arrow pointing to it.

Step 2 / 7

CREATE CONSUMER



username
(semi-optional)

imc 

The username of the consumer. You must send either this field or **custom_id** with the request.

custom_id
(semi-optional)

Field for storing an existing ID for the consumer, useful for mapping Kong with users in your existing database. You must send either this field or **username** with the request.

Tags
(optional)

Optionally add tags to the consumer

 SUBMIT CONSUMER

Step 3 / 7

CONSUMER: imc

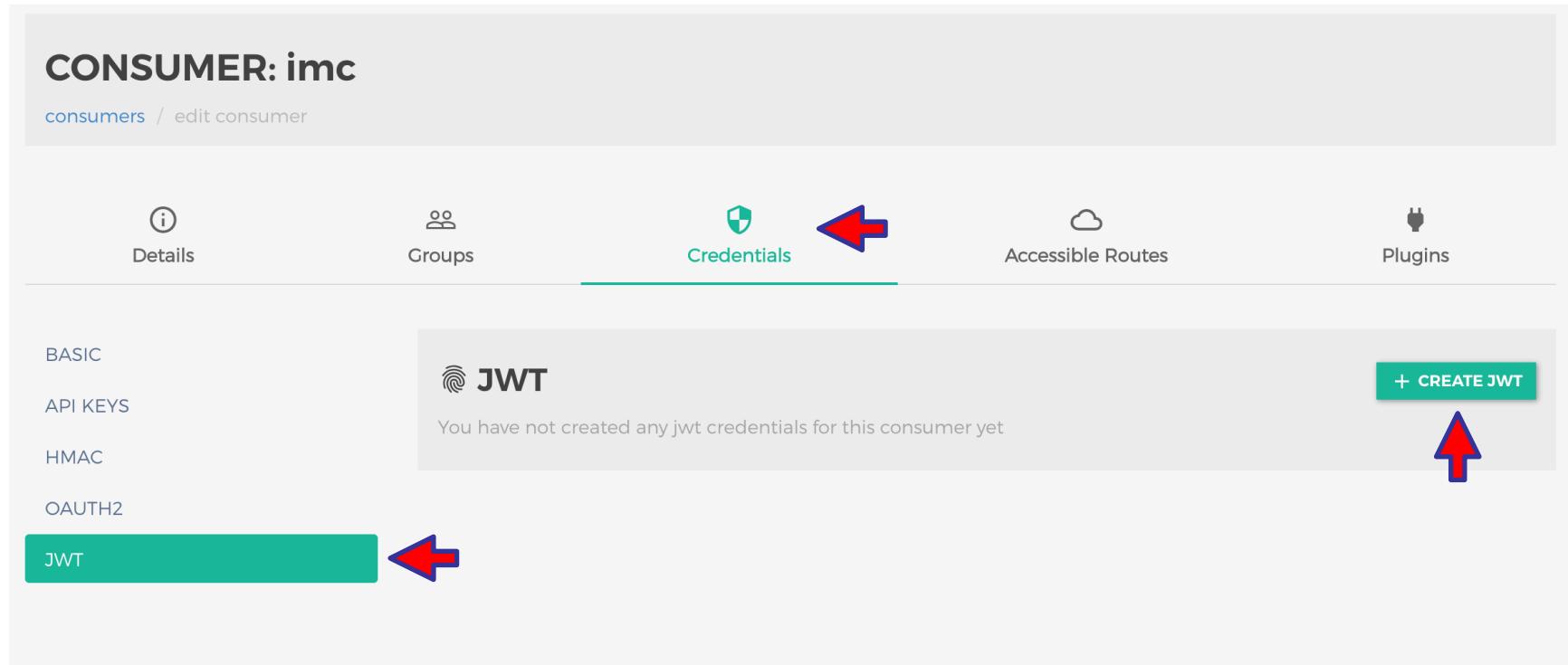
consumers / edit consumer

Details Groups **Credentials**  Accessible Routes Plugins

BASIC API KEYS HMAC OAUTH2 

 **JWT**  + CREATE JWT

You have not created any jwt credentials for this consumer yet



Step 4 / 7

CREATE JWT

Create JWT for **imc**

key
(optional)

A unique string identifying the credential. If left out, it will be auto-generated. However, usage of this key is **mandatory** while crafting your token, as specified in the next section.

algorithm
(optional)

HS256

The algorithm used to verify the token's signature. Can be **HS256** or **RS256**.

rsa_public_key
(optional)

If **algorithm** is **RS256**, the public key (in PEM format) to use to verify the token's signature.

secret
(optional)

If **algorithm** is **HS256**, the secret used to sign JWTs for this credential. If left out, will be auto-generated. If **algorithm** is **RS256**, this is the private key (in PEM format) to use to verify the token's signature.

SUBMIT 

Step 5 / 7

CONSUMER: imc

consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

BASIC API KEYS HMAC OAUTH2 JWT

JWT + CREATE JWT

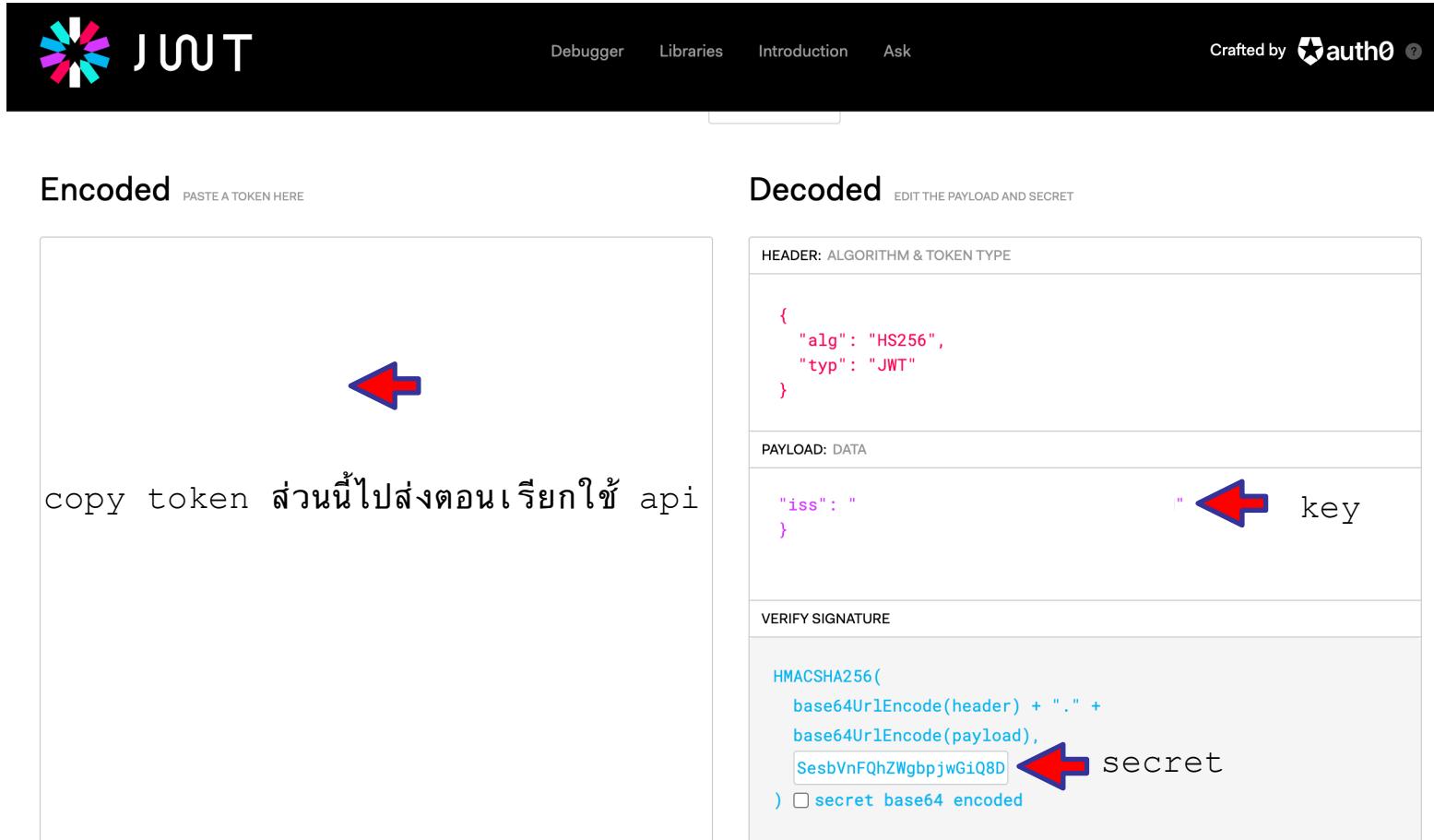
```
{  
  "id": "b7e2722a-b723-492e-b9fa-85bb19b01d33",  
  "key": "KsCI3BgXxnywgqo4zRa8sJ5TkFH0wLCY",  
  "algorithm": "HS256",  
  "created_at": 1659507770,  
  "secret": "TyN0nQXwyHBF0KvPLG61QSAhgj6EWUu",  
  "consumer": {  
    "id": "8149bd47-f188-435a-b70a-0380e254ae5e"  
  },  
  "tags": null,  
  "rsa_public_key": null  
}
```

Created Aug 3, 2022 



126

Step 6 / 7



The screenshot shows the JUUT JWT Debugger interface. On the left, under the "Encoded" section, there is a large input field with a red arrow pointing to it, indicating where to paste the token. On the right, under the "Decoded" section, the token is shown in JSON format:

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}  
  
PAYLOAD: DATA  


```
{"iss": "
"}

VERIFY SIGNATURE


```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  SesbVnFQhZWgbpjwGiQ8D  
)  secret base64 encoded
```


```



Three specific fields are highlighted with red arrows:



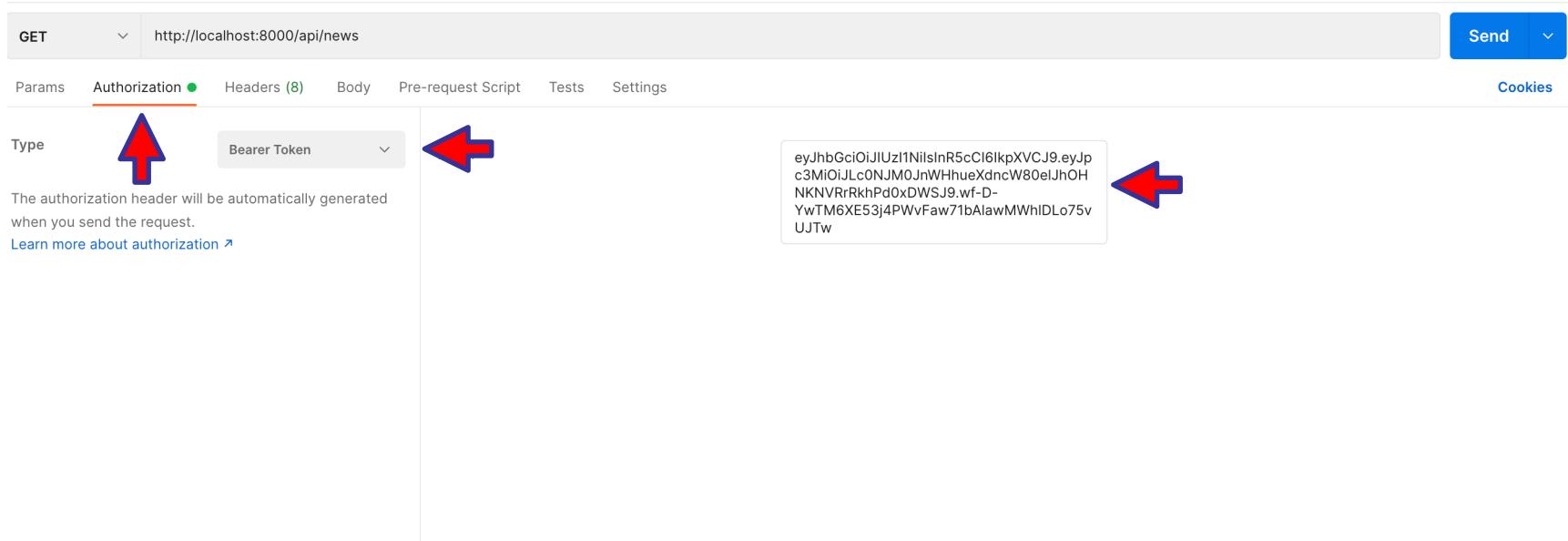
- A red arrow points to the "iss" field in the payload section, labeled "key".
- A red arrow points to the "SesbVnFQhZWgbpjwGiQ8D" value in the HMACSHA256 signature calculation, labeled "secret".
- A red arrow points to the "secret" checkbox in the HMACSHA256 signature calculation, labeled "secret".



Below the "Encoded" section, there is a text instruction: "copy token ล้วนๆไปส่งต่อนเรียกใช้ api".


```

Step 7 / 7

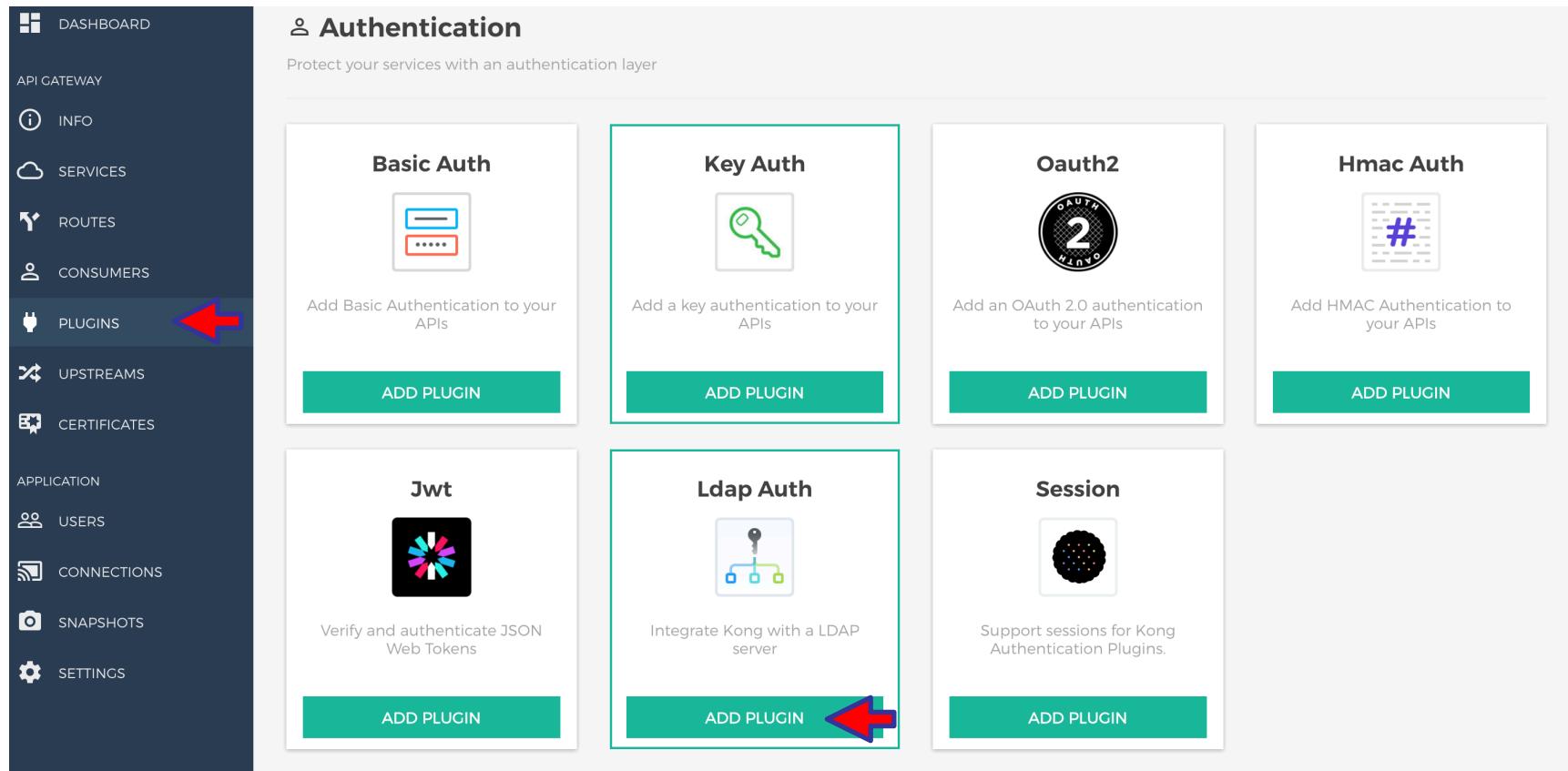


The screenshot shows the Postman application interface for making an API request. The request method is set to GET, and the URL is http://localhost:8000/api/news. The 'Authorization' tab is selected, showing a dropdown menu with 'Bearer Token'. A red arrow points to this dropdown. Below it, a text box contains a long JWT token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJLc0NJM0JnWHhueXdnW80elJhOHNKNVRrRkhPd0xDWSJ9.wf-D-YwTM6XE5j4PWvFaw71bAlawMWhlDLo75vUJTtw. Another red arrow points to the right side of this token box. The 'Params', 'Headers (8)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' tabs are also visible at the top. The 'Cookies' tab is located on the far right. A 'Send' button is at the top right, and a 'Send' dropdown is next to it. A 'Learn more about authorization' link is present below the dropdown.

How to add

LDAP AUTH PLUGIN

Step 1/3



The screenshot shows the Kong API Gateway dashboard under the 'Authentication' section. The left sidebar has a red arrow pointing to the 'PLUGINS' option. The main area displays seven authentication plugins: Basic Auth, Key Auth, OAuth2, Hmac Auth, Jwt, Ldap Auth, and Session. Each plugin has a small icon, a brief description, and a green 'ADD PLUGIN' button. A second red arrow points to the 'ADD PLUGIN' button for the Ldap Auth plugin.

Plugin	Description	Action
Basic Auth	Add Basic Authentication to your APIs	ADD PLUGIN
Key Auth	Add a key authentication to your APIs	ADD PLUGIN
OAuth2	Add an OAuth 2.0 authentication to your APIs	ADD PLUGIN
Hmac Auth	Add HMAC Authentication to your APIs	ADD PLUGIN
Jwt	Verify and authenticate JSON Web Tokens	ADD PLUGIN
Ldap Auth	Integrate Kong with a LDAP server	ADD PLUGIN
Session	Support sessions for Kong Authentication Plugins.	ADD PLUGIN

Step 2/3

applies to all consumers.

ldap host	192.168.1.101	
		Host on which the LDAP server is running.
ldap port	1389	
		TCP port where the LDAP server is listening.
ldaps	<input type="checkbox"/> NO	
start tls	<input type="checkbox"/> NO	
		Set it to true to issue StartTLS (Transport Layer Security) extended operation over ldap connection.
verify ldap host	<input type="checkbox"/> NO	
		Set it to true to authenticate LDAP server. The server certificate will be verified according to the CA certificates specified by the <code>lua_ssl_trusted_certificate</code> directive.
base dn	ou=users,dc=example,dc=org	
		Base DN as the starting point for the search.
attribute	cn	
		Attribute to be used to search the user.

Step 3/3

	Params	Authorization	Headers (9)	Body	Pre-request Script	Tests	Settings
<input checked="" type="checkbox"/>	User-Agent					<small>i</small> PostmanRuntime/7.29.0	
<input checked="" type="checkbox"/>	Accept					<small>i</small> */*	
<input checked="" type="checkbox"/>	Accept-Encoding					<small>i</small> gzip, deflate, br	
<input checked="" type="checkbox"/>	Connection					<small>i</small> keep-alive	
<input type="checkbox"/>	x-api-key					82IO6v0crI2KFUHIG9hhqnywe7nyU4rf	
<input checked="" type="checkbox"/>	Authorization					ldap dXNlcjAxOnBhc3N3b3JkMQ==	
	Key					Value	

```
credentials := [ldap | LDAP] base64(username:password)
```

How to add

ACL PLUGINS

Step 1 / 6

Services

Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.

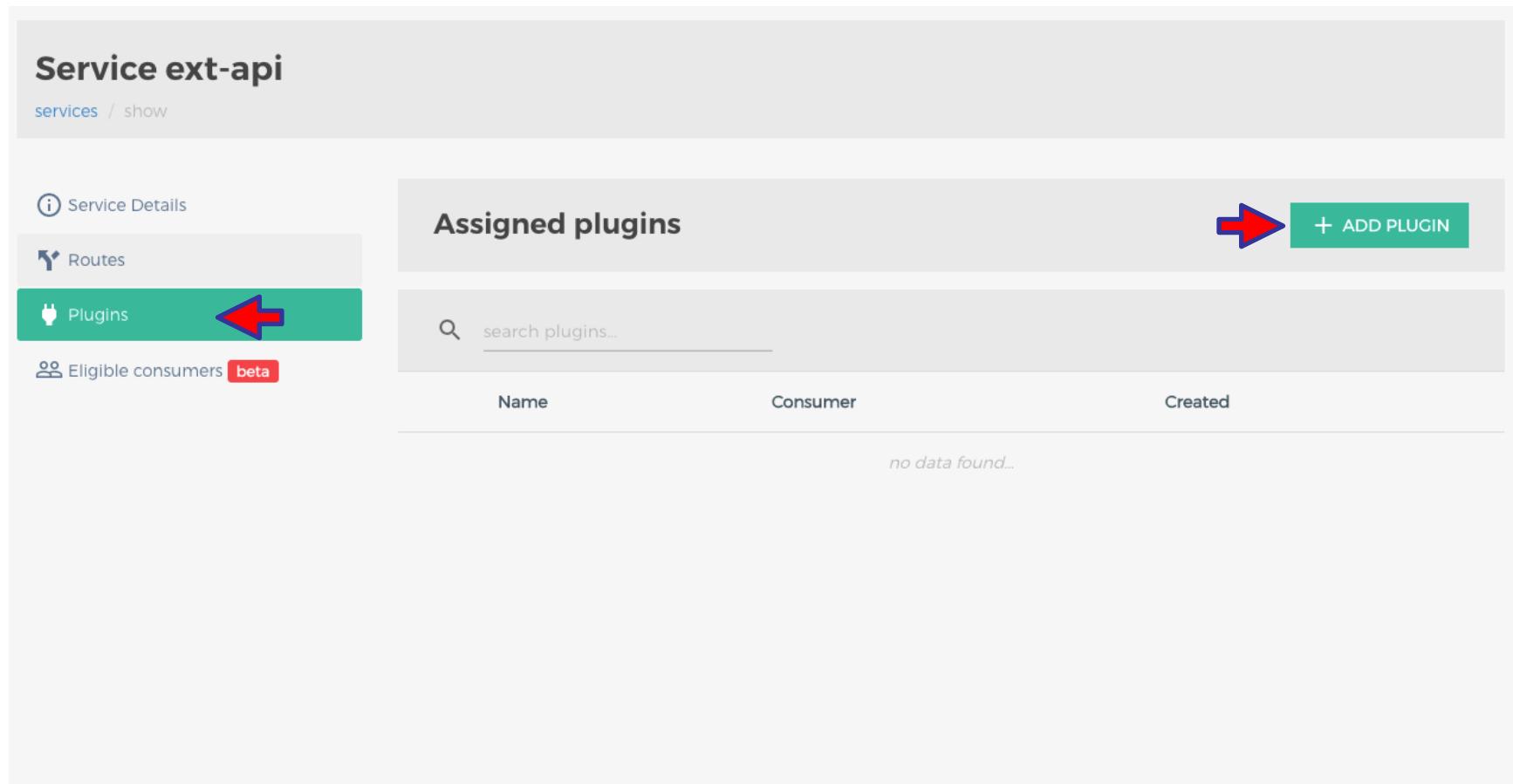
Services			
NAME	HOST	TAGS	CREATED
ext-api	192.168.1.101		Apr 16, 2022
api 	192.168.1.101		Apr 16, 2022

+ ADD NEW SERVICE

search... Results : 25

NAME	HOST	TAGS	CREATED	Actions
ext-api	192.168.1.101		Apr 16, 2022	 DELETE
api	192.168.1.101		Apr 16, 2022	 DELETE

Step 2 / 6



Service ext-api

services / show

Service Details

Routes

Plugins 

Eligible consumers 

Assigned plugins

+ ADD PLUGIN 

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 6

ADD PLUGIN X

Authentication

Security ←

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

SECURITY
Protect your services with additional security layers

Acl



Control which consumers can access...

ADD PLUGIN ←

Cors



Allow developers to make requests from th...

ADD PLUGIN

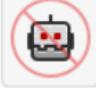
Ip Restriction



Whitelist or blacklist IPs that can make requests

ADD PLUGIN

Bot Detection



Detects and blocks bots or custom clients

ADD PLUGIN

Acme



Let's Encrypt and ACMEv2 integration...

ADD PLUGIN

Step 4 / 6

ADD ACL

Restrict access to an API by whitelisting or blacklisting consumers using arbitrary ACL group names. This plugin requires an authentication plugin to have been already enabled on the API.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

allow ← **ระบุชื่อกลุ่มที่ยอมให้เรียกใช้งาน**
Tip: Press **Enter** to accept a value.

deny ← **ระบุชื่อกลุ่มที่ไม่ยอมให้เรียกใช้งาน**
Tip: Press **Enter** to accept a value.

hide groups header NO

✓ ADD PLUGIN

Step 5 / 6

CONSUMER: imc

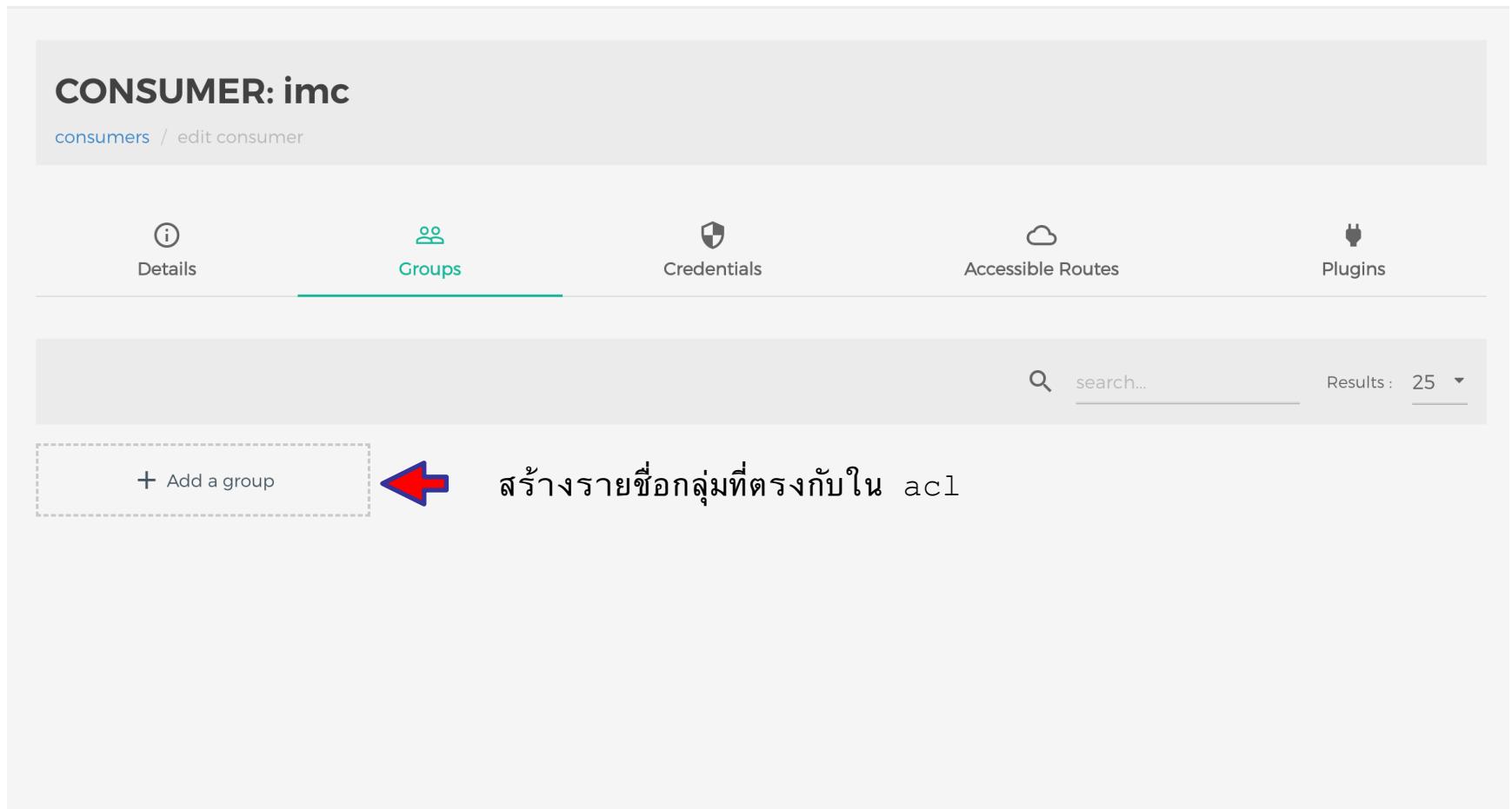
consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

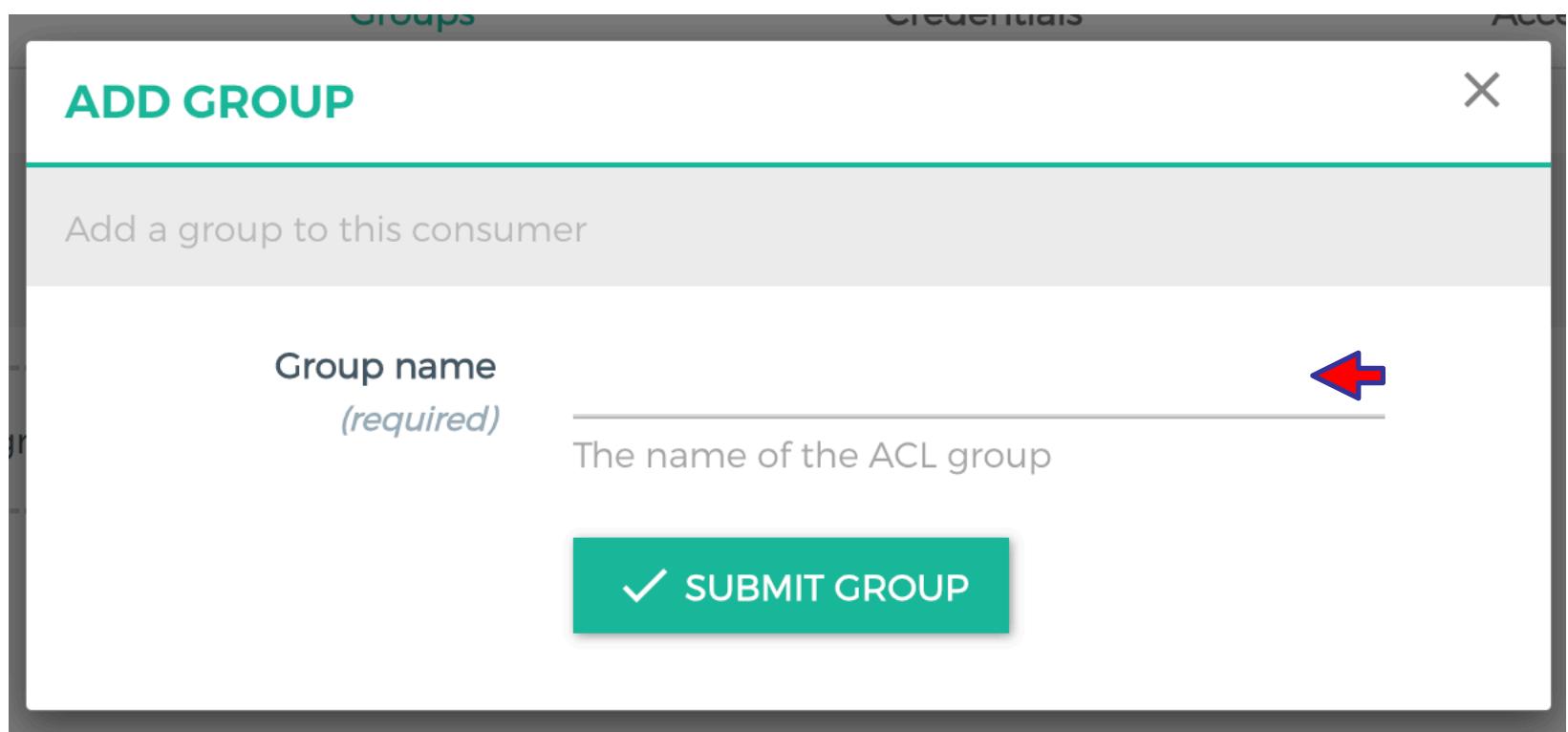
+ Add a group 

สร้างรายชื่อกลุ่มที่ตรงกับใน acl

search... Results : 25



Step 6 / 6



The screenshot shows a modal window titled "ADD GROUP". The window has a light gray header bar with the title and a close button (X). Below the header is a sub-header "Add a group to this consumer". The main content area contains a "Group name" field with the placeholder "The name of the ACL group". A red arrow points to the right side of this input field. At the bottom is a teal "SUBMIT GROUP" button with a white checkmark icon.

Groups Credentials Accounts

ADD GROUP

Add a group to this consumer

Group name
(required)

The name of the ACL group

✓ SUBMIT GROUP

How to add

CORS PLUGINS

Step 1 / 4

Services			
<p>Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.</p>			
+ ADD NEW SERVICE		search...	Results : 25 ▾
NAME	HOST	TAGS	CREATED ▾
 ext-api	192.168.1.101		Apr 16, 2022
 api	192.168.1.101		Apr 16, 2022

Step 2 / 4

Service ext-api

services / show

Service Details

Routes

Plugins 

Eligible consumers 

Assigned plugins 

+ ADD PLUGIN

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

ADD PLUGIN X

Authentication

Security 

Traffic Control

Serverless

Analytics & Monitoring

Transformations

Logging

Other

SECURITY
Protect your services with additional security layers

Acl



Control which consumers can access...

ADD PLUGIN

Cors



Allow developers to make requests from th...

ADD PLUGIN 

Ip Restriction



Whitelist or blacklist IPs that can make requests

ADD PLUGIN

Bot Detection



Detects and blocks bots or custom clients

ADD PLUGIN

Acme



Let's Encrypt and ACMEv2 integration...

ADD PLUGIN

Step 4 / 4

ADD CORS X

Easily add Cross-origin resource sharing (CORS) to your API by enabling this plugin.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

origins * ←

Tip: Press **Enter** to accept a value.

headers

Tip: Press **Enter** to accept a value.
Value for the Access-Control-Allow-Headers header, expects a comma delimited string (e.g. Origin, Authorization).

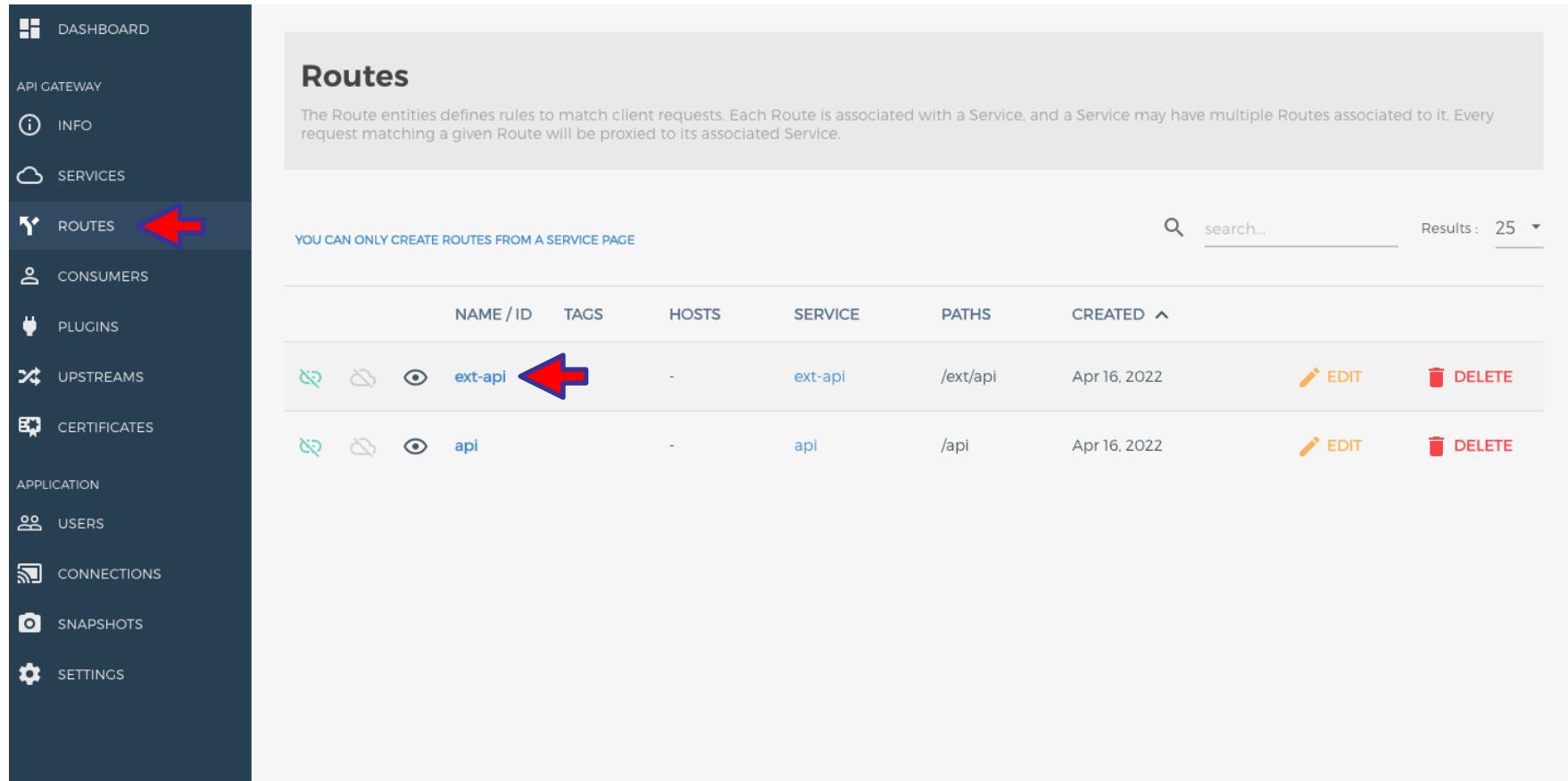
exposed headers

Tip: Press **Enter** to accept a value.
Value for the Access-Control-Expose-Headers header, expects a comma delimited string (e.g. Origin, Authorization). If not specified, no custom headers are exposed.

How to add

PROXY CACHE PLUGIN

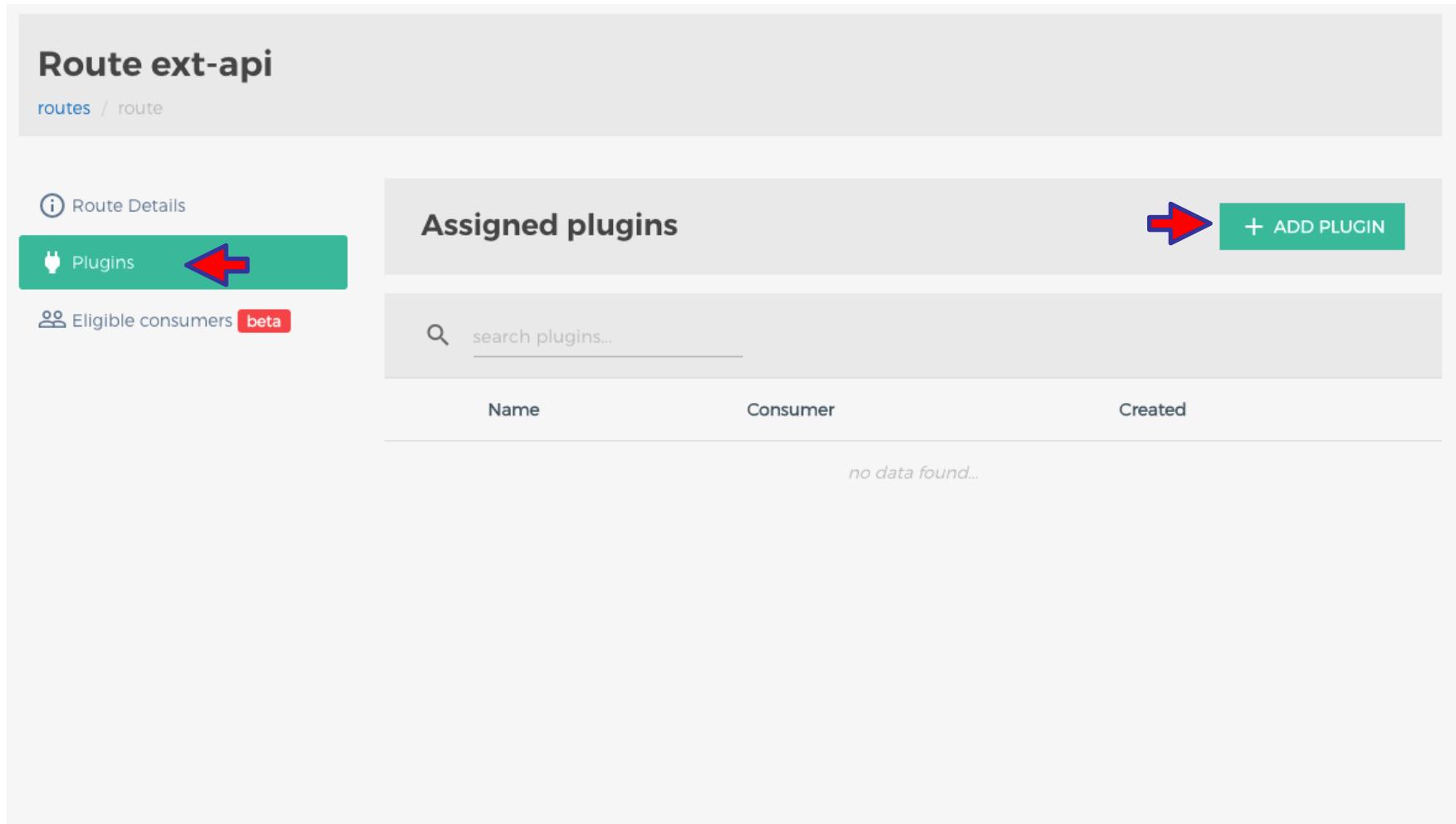
Step 1 / 4



The screenshot shows the 'Routes' section of an API gateway interface. On the left, a sidebar lists various management options like Dashboard, API Gateway, Services, and Routes. The 'Routes' option is selected and highlighted with a blue arrow. The main area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this, a message says 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table lists two routes:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4



The screenshot shows the 'Route ext-api' configuration page. At the top left, there are navigation links: 'routes' and 'route'. On the left side, there are three sections: 'Route Details' (with an info icon), 'Plugins' (with a plug icon, highlighted with a red arrow pointing to it), and 'Eligible consumers' (with a people icon and a 'beta' badge). In the center, the 'Assigned plugins' section is displayed. It includes a search bar labeled 'search plugins...' and a table with columns 'Name', 'Consumer', and 'Created'. A large red arrow points to the '+ ADD PLUGIN' button at the top right of this section. Below the table, the text 'no data found...' is visible.

Name	Consumer	Created
no data found...		

Step 3 / 4

ADD PLUGIN X

Authentication
Security
Traffic Control 
Serverless
Analytics & Monitoring
Transformations
Logging
Other

⌘ TRAFFIC CONTROL

Manage, throttle and restrict inbound and outbound API traffic



Rate Limiting

Rate-limit how many HTTP requests a...

ADD PLUGIN



Response Ratelimiting

Rate-Limiting based on a custom response...

ADD PLUGIN



Request Size Limiting

Block requests with bodies greater than a...

ADD PLUGIN



Request Termination

This plugin terminates incoming requests wit...

ADD PLUGIN



Proxy Cache

Cache and serve commonly requested...

ADD PLUGIN 

Step 4 / 4

ADD PROXY CACHE X

Configure the Plugin.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

response code

Tip: Press **Enter** to accept a value.

request method

Tip: Press **Enter** to accept a value.

content type application/json; charset=utf-8 

Tip: Press **Enter** to accept a value.

cache ttl 300

strategy memory 

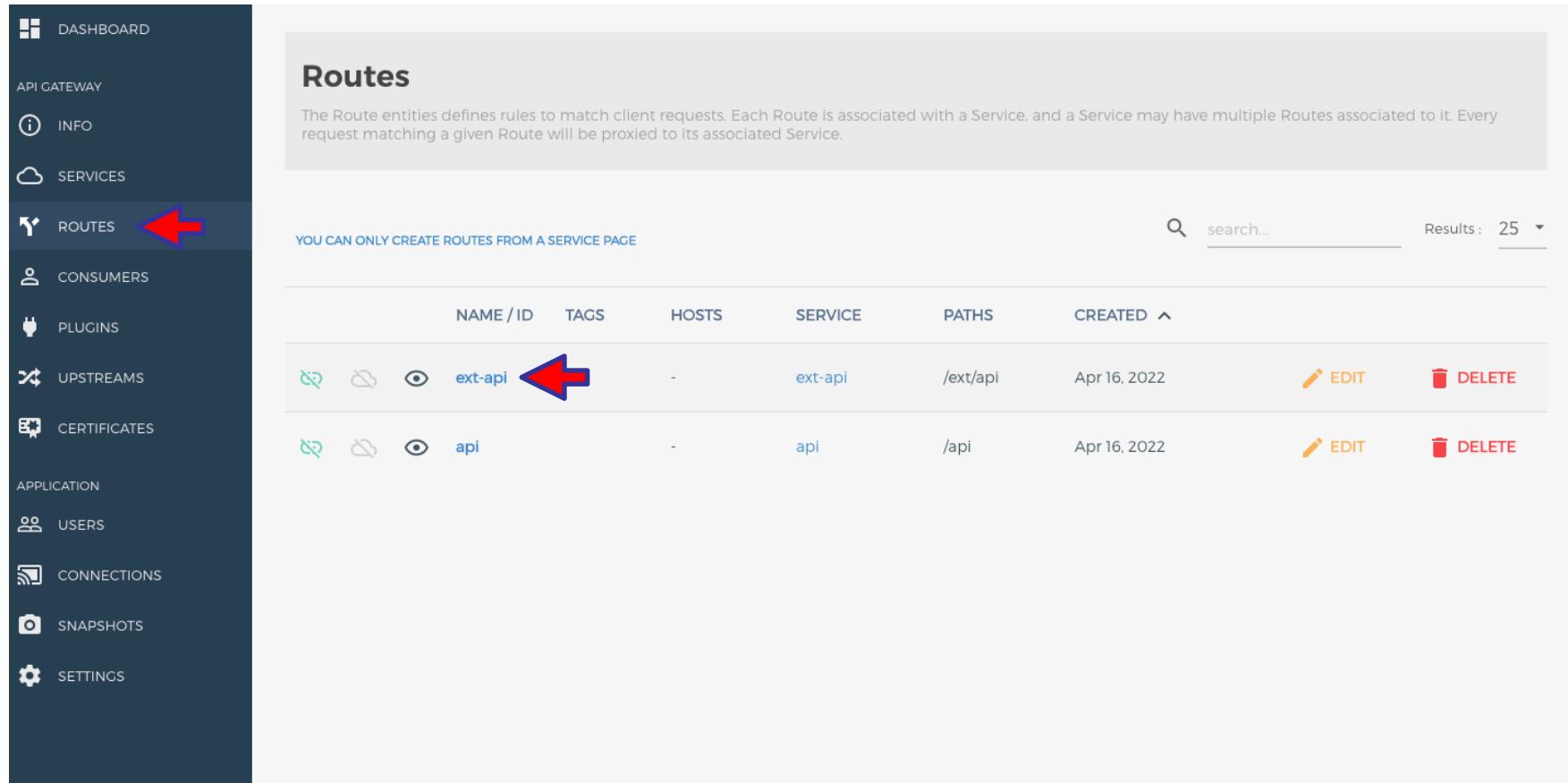
cache control NO

storage ttl

How to add

RATE LIMITING PLUGIN

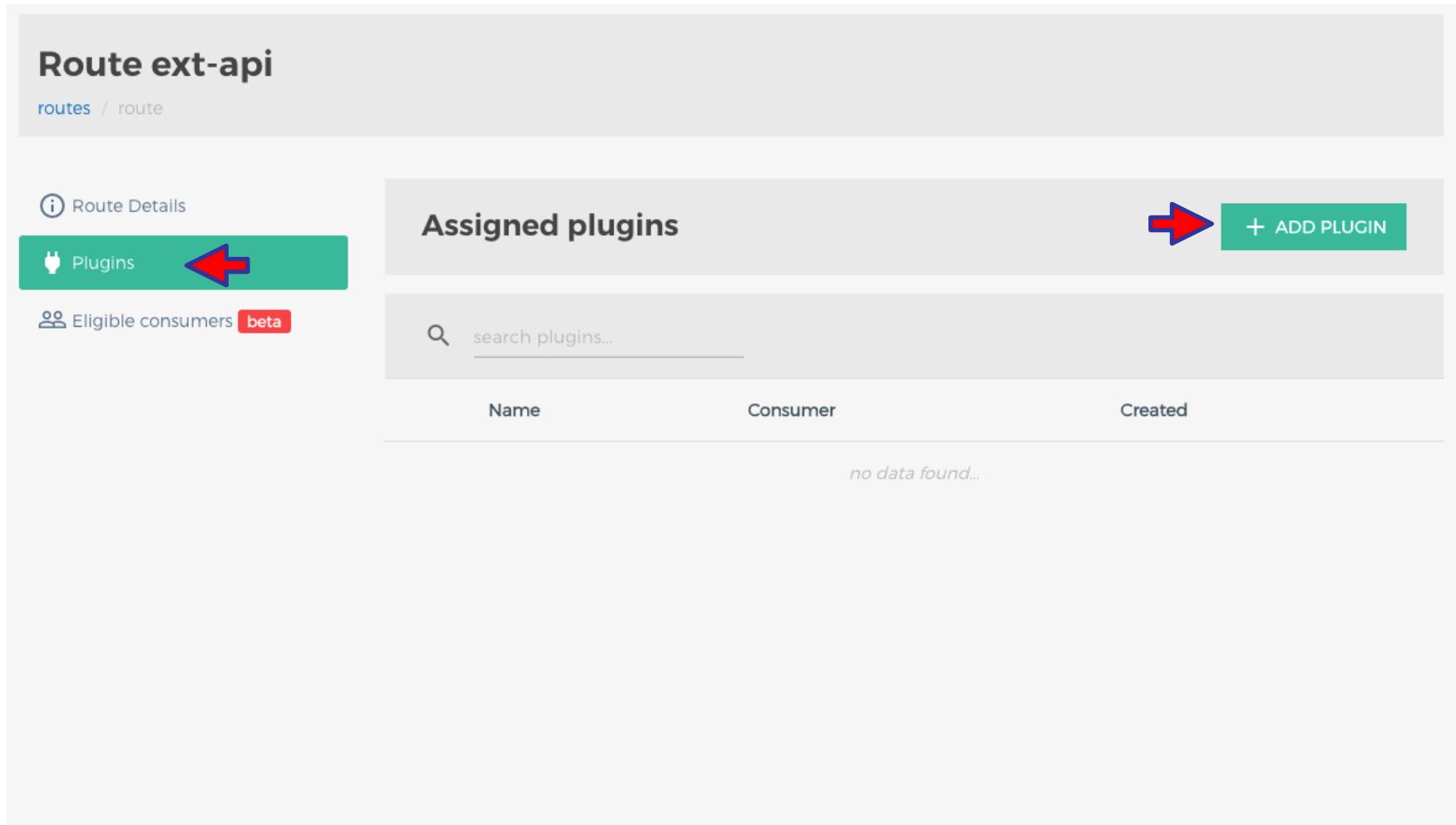
Step 1 / 4



The screenshot shows the 'Routes' section of an API gateway interface. On the left, a sidebar lists various management options like Dashboard, API Gateway, Services, and Routes. The 'Routes' option is selected and highlighted with a blue arrow. The main area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this, a message says 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table lists two routes:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4



The screenshot shows the 'Route ext-api' configuration page. At the top left, there are navigation links: 'routes' and 'route'. On the left side, there are three sections: 'Route Details' (with an info icon), 'Plugins' (with a plug icon, highlighted with a red arrow pointing to it), and 'Eligible consumers' (with a people icon and a 'beta' badge). In the center, the 'Assigned plugins' section is displayed. It includes a search bar labeled 'search plugins...', a table header with columns 'Name', 'Consumer', and 'Created', and a message 'no data found...'. A large red arrow points to the '+ ADD PLUGIN' button at the top right of this section.

Route ext-api

routes / route

Route Details

Plugins 

Eligible consumers 

Assigned plugins 

+ ADD PLUGIN

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

ADD PLUGIN

[Authentication](#)[Security](#)[Traffic Control](#) [Serverless](#)[Analytics & Monitoring](#)[Transformations](#)[Logging](#)[Other](#)

TRAFFIC CONTROL

Manage, throttle and restrict inbound and outbound API traffic

Rate Limiting



Rate-limit how many HTTP requests a...

[ADD PLUGIN](#) 

Response Ratelimiting



Rate-Limiting based on a custom response...

[ADD PLUGIN](#)

Request Size Limiting



Block requests with bodies greater than a...

[ADD PLUGIN](#)

Request Termination



Proxy Cache



Step 4 / 4

ADD RATE LIMITING



Rate limit how many HTTP requests a developer can make in a given period of seconds, minutes, hours, days, months or years. If the API has no authentication layer, the Client IP address will be used, otherwise the Consumer will be used if an authentication plugin has been configured.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

second

1 

The amount of HTTP requests the developer can make per second. At least one limit must exist.

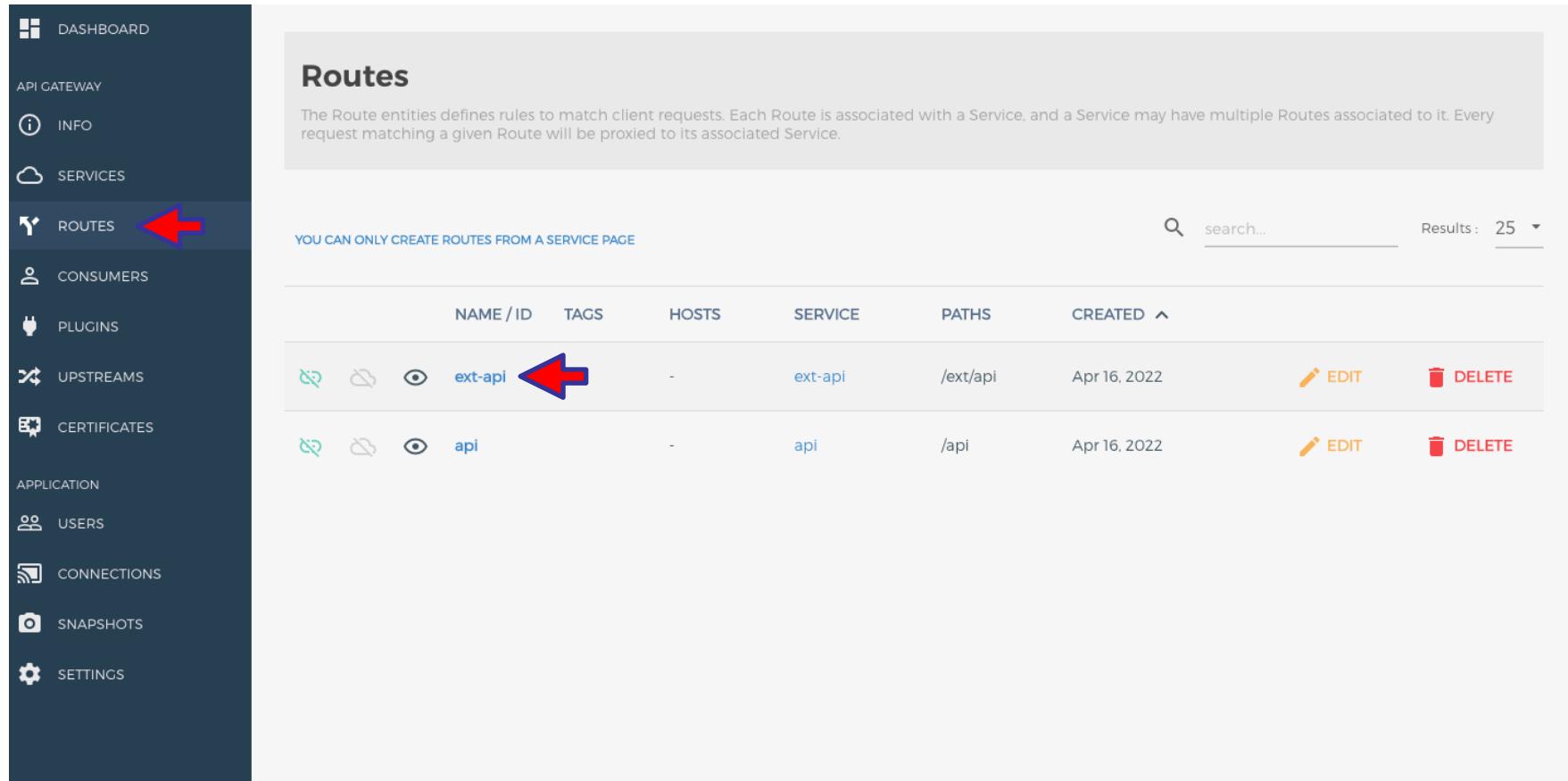
minute

The amount of HTTP requests the developer can make per minute. At least one limit must exist.

How to add

RESPONSE TRANSFORMER

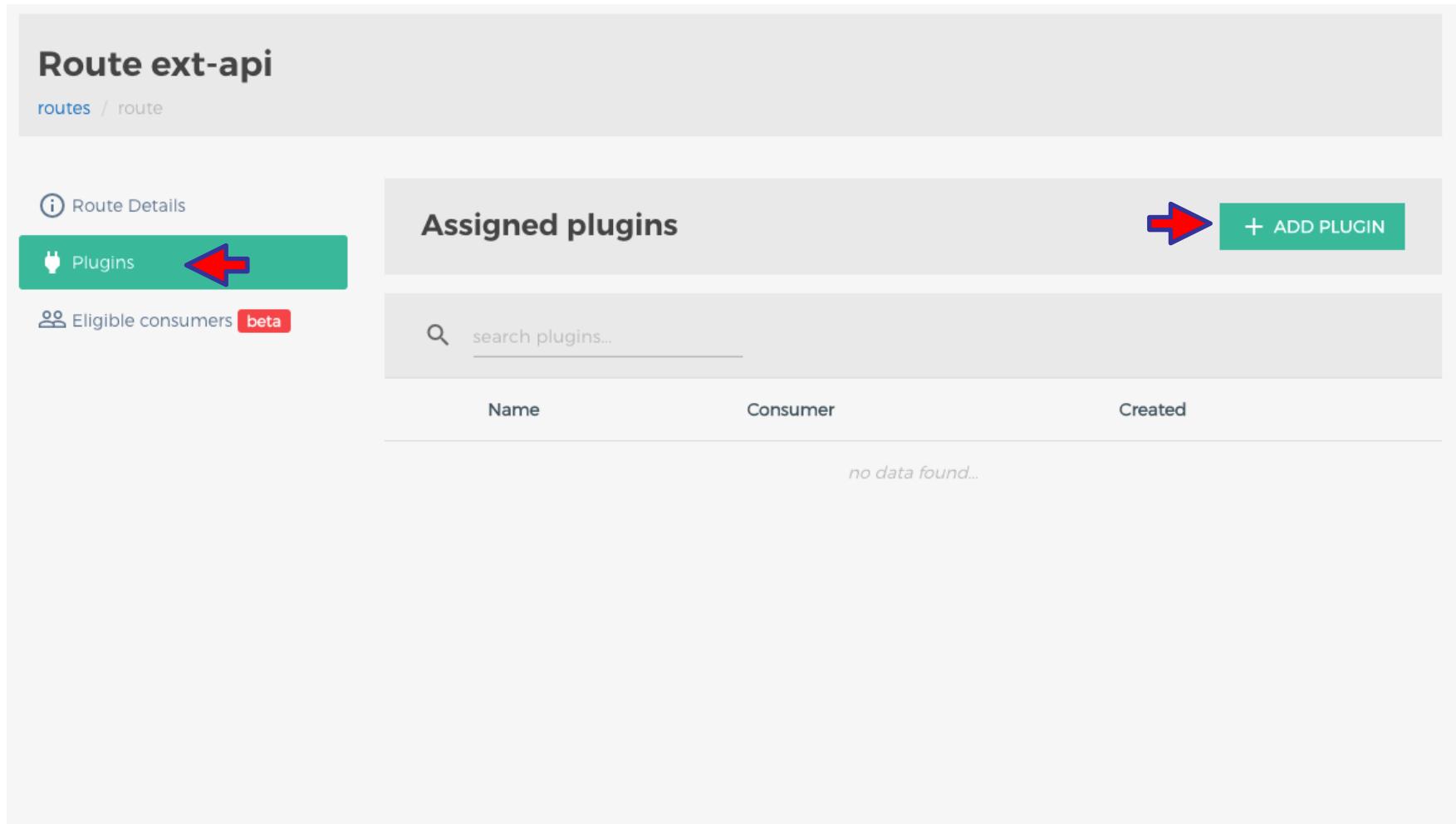
Step 1 / 4



The screenshot shows the 'Routes' section of an API gateway interface. On the left, a sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES (highlighted with a blue arrow), CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled 'Routes' and contains a brief description: 'The Route entities defines rules to match client requests. Each Route is associated with a Service, and a Service may have multiple Routes associated to it. Every request matching a given Route will be proxied to its associated Service.' Below this is a message: 'YOU CAN ONLY CREATE ROUTES FROM A SERVICE PAGE'. A search bar and a results count of '25' are also present. The table displays two routes:

	NAME / ID	TAGS	HOSTS	SERVICE	PATHS	CREATED	Actions	
			ext-api (highlighted with a red arrow)	-	ext-api	/ext/api	Apr 16, 2022	EDIT DELETE
			api	-	api	/api	Apr 16, 2022	EDIT DELETE

Step 2 / 4



The screenshot shows the 'Route ext-api' configuration page. At the top left, there are navigation links: 'routes' and 'route'. On the left side, there are three sections: 'Route Details' (with an info icon), 'Plugins' (with a plug icon, highlighted with a red arrow pointing to it), and 'Eligible consumers' (with a people icon and a 'beta' badge). In the center, the 'Assigned plugins' section is displayed. It includes a search bar labeled 'search plugins...', a table header with columns 'Name', 'Consumer', and 'Created', and a message 'no data found...'. A large red arrow points to the '+ ADD PLUGIN' button at the top right of this section.

Route ext-api

routes / route

Route Details

Plugins 

Eligible consumers 

Assigned plugins 

+ ADD PLUGIN

search plugins...

Name	Consumer	Created
no data found...		

Step 3 / 4

Authentication Security Traffic Control Serverless Analytics & Monitoring **Transformations** Logging



Transformations

Transform request and responses on the fly on Kong



Request Transformer

Modify the request before hitting the upstream server

ADD PLUGIN



Response Transformer

Modify the upstream response before returning it to the client



ADD PLUGIN



Correlation Id

Correlate requests and responses using a unique ID

ADD PLUGIN

Step 4 / 4

ADD RESPONSE TRANSFORMER

Transform the response sent by the upstream server on the fly on Kong, before returning the response to the client.

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

remove

> json

headers



rename

> headers

replace

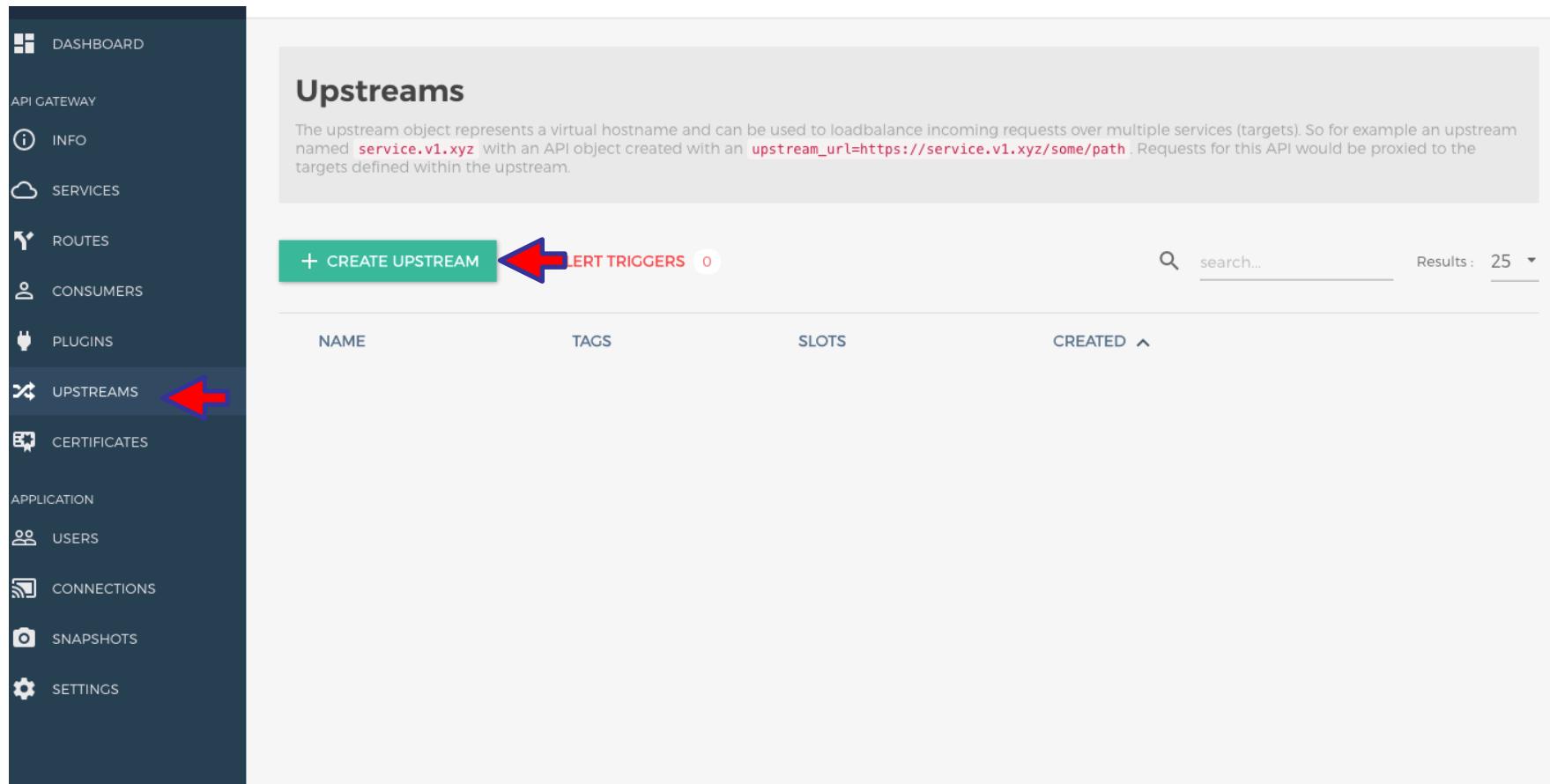
> json

> json_types

How to add

UPSTREAMS

Step 1 / 7



Upstreams

The upstream object represents a virtual hostname and can be used to loadbalance incoming requests over multiple services (targets). So for example an upstream named `service.v1.xyz` with an API object created with an `upstream_url=https://service.v1.xyz/some/path`. Requests for this API would be proxied to the targets defined within the upstream.

+ CREATE UPSTREAM  ALERT TRIGGERS 0

search... Results : 25

NAME	TAGS	SLOTS	CREATED

Step 2 / 7

CREATE UPSTREAM



Name <i>(required)</i>	api_stream 
This is a hostname like name that can be referenced in an <code>upstream_url</code> field of an <code>api</code> or the <code>host</code> of a service.	
Tags <i>(optional)</i>	
Optionally add tags to the Upstream	
Hash on <i>(optional)</i>	none 
What to use as hashing input: <code>none</code> , <code>consumer</code> , <code>ip</code> , <code>header</code> or <code>cookie</code> (defaults to <code>none</code> resulting in a weighted-round-robin scheme).	

Step 3 / 7

Upstreams

The upstream object represents a virtual hostname and can be used to loadbalance incoming requests over multiple services (targets). So for example an upstream named `service.v1.xyz` with an API object created with an `upstream_url=https://service.v1.xyz/some/path`. Requests for this API would be proxied to the targets defined within the upstream.

NAME	TAGS	SLOTS	CREATED	
api_stream		1000	Apr 16 2022 @16:56	 DETAILS  DELETE

Step 4 / 7

Edit Upstream

api_stream
upstreams / edit

Manage

- Details
- Targets
- Alerts

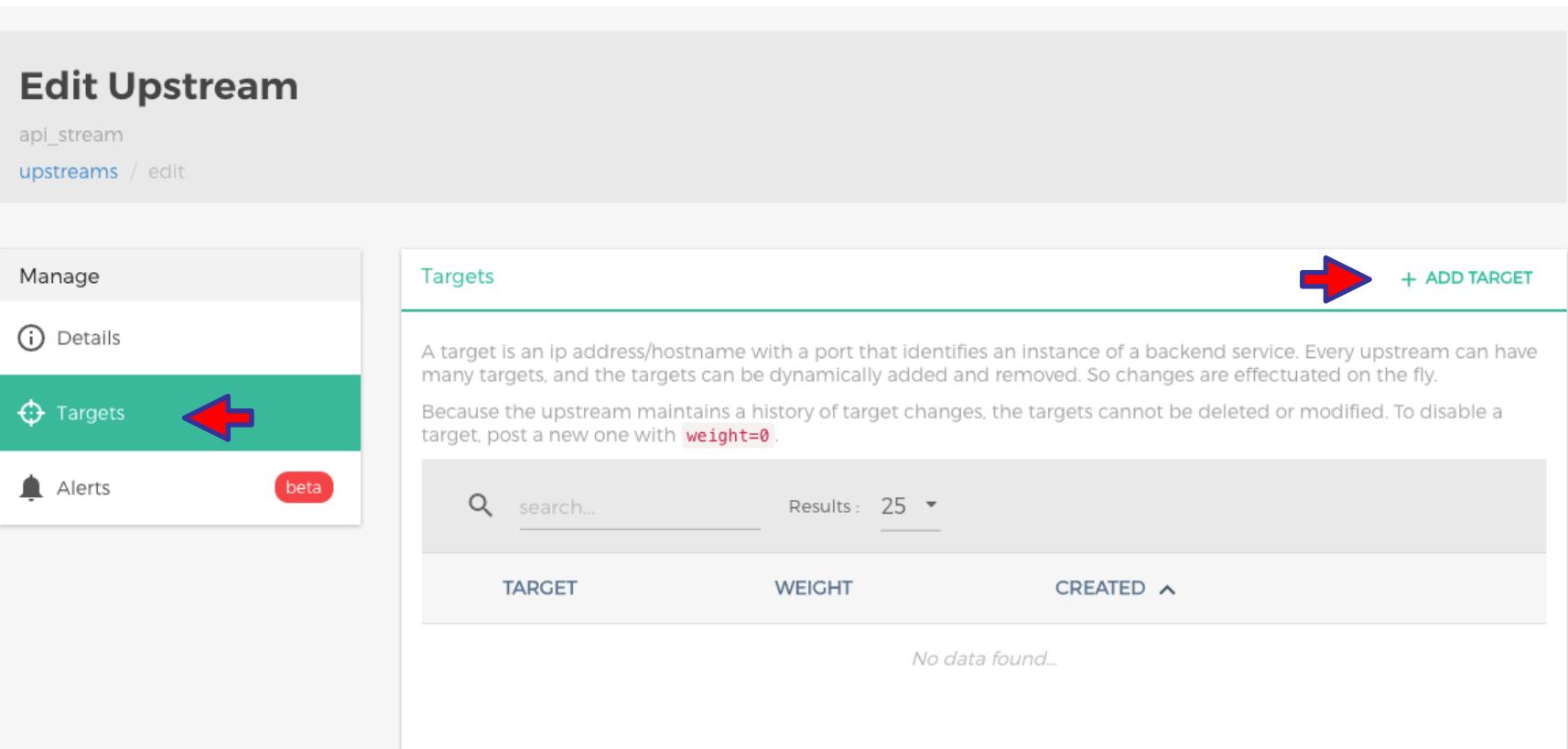
beta

Targets

A target is an ip address/hostname with a port that identifies an instance of a backend service. Every upstream can have many targets, and the targets can be dynamically added and removed. So changes are effectuated on the fly.

Because the upstream maintains a history of target changes, the targets cannot be deleted or modified. To disable a target, post a new one with `weight=0`.

TARGET	WEIGHT	CREATED
No data found...		



Step 5 / 7

ADD NEW TARGET



Target
(required)

192.168.1.101:3030



The target address (ip or hostname) and port.
If omitted the **port** defaults to **8000**. If the
hostname resolves to an SRV record, the **port**
value will overridden by the value from the
dns record.

Weight
(optional)

100

The weight this target gets within the
upstream loadbalancer (**0 - 1000**, defaults to
100). If the hostname resolves to an SRV
record, the **weight** value will overridden by
the value from the dns record.

 SUBMIT TARGET

Step 6 / 7

Services

Service entities, as the name implies, are abstractions of each of your own upstream services. Examples of Services would be a data transformation microservice, a billing API, etc.

Services			
NAME	HOST	TAGS	CREATED
ext-api	192.168.1.101		Apr 16, 2022
api 	192.168.1.101		Apr 16, 2022

+ ADD NEW SERVICE

search... Results : 25

NAME	HOST	TAGS	CREATED	Actions
ext-api	192.168.1.101		Apr 16, 2022	 DELETE
api	192.168.1.101		Apr 16, 2022	 DELETE

Step 7 / 7

Service Details

Service details

Name
(optional)
api

The service name.

Description
(optional)
An optional service description.

Tags
(optional)
Optionally add tags to the service

Protocol
(semi-optional)
http

The protocol used to communicate with the upstream. It can be one of [http](#) or [https](#).

Host
(semi-optional)
api_stream 

The host of the upstream server.

Port
(semi-optional)
8000 

The upstream server port. Defaults to [80](#).

Path
(optional)
/ 

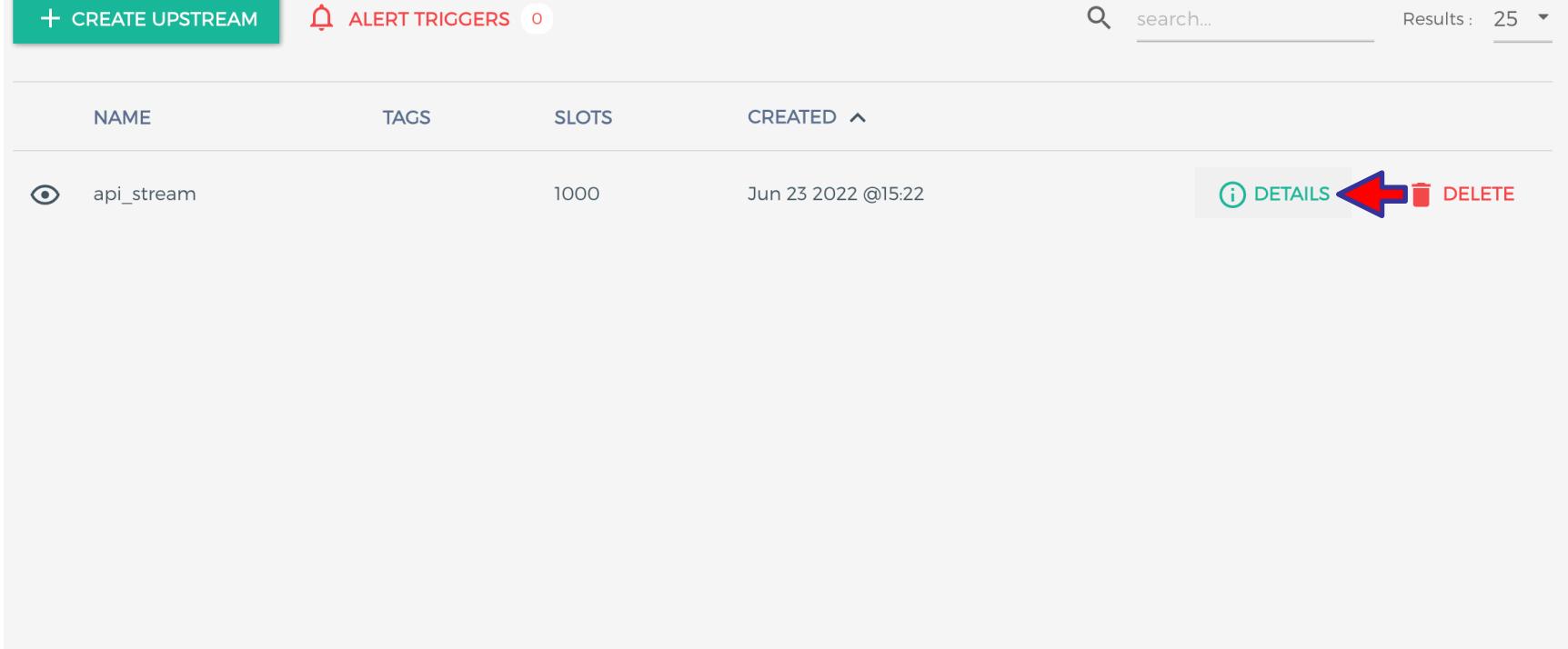
How to config server

HEALTH CHECK

Step 1/2

Upstreams

The upstream object represents a virtual hostname and can be used to loadbalance incoming requests over multiple services (targets). So for example an upstream named `service.v1.xyz` with an API object created with an `upstream_url=https://service.v1.xyz/some/path`. Requests for this API would be proxied to the targets defined within the upstream.



NAME	TAGS	SLOTS	CREATED	
api_stream		1000	Jun 23 2022 @15:22	 DETAILS  DELETE

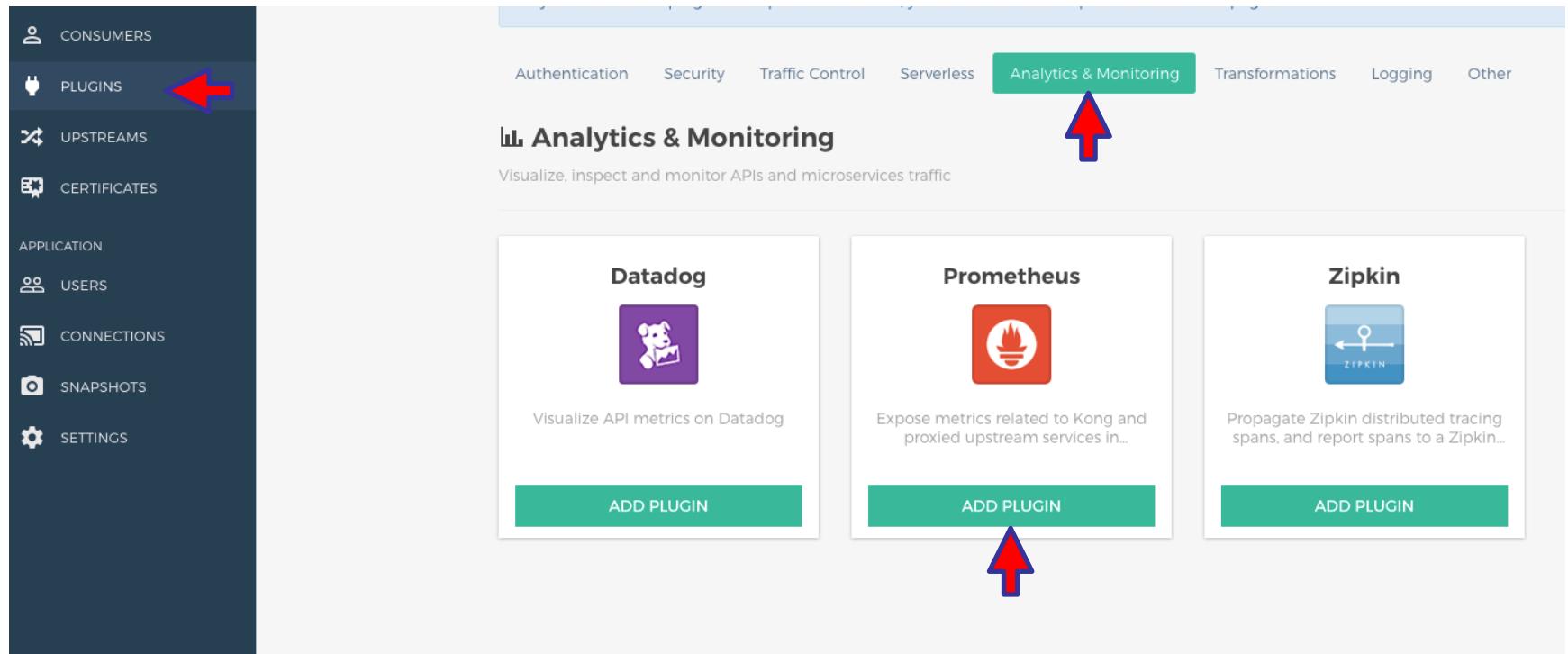
Step 2/2

	<p>default value is 1 second.</p>
Http statuses (optional)	<p>200 X 302 X</p> <hr/>
	<p>Tip: Press Enter to accept a value.</p>
	<p>An array of HTTP statuses to consider a success, indicating healthiness, when returned by a probe in active health checks. Defaults to [200, 302].</p>
Healthy interval (optional)	<p>5</p> <hr/> <p>Interval between active health checks for healthy targets (in seconds). A value of zero indicates that active probes for healthy targets should not be performed.</p>
Healthy successes (optional)	<p>5</p> <hr/> <p>Number of successes in active probes (as defined by healthchecks.active.healthy.http_statuses) to consider a target healthy.</p>
Https SNI (optional)	<hr/> <p>The hostname to use as an SNI (Server Name Identification) when performing active health checks using HTTPS. This is particularly useful when Targets are configured using IPs, so that the target host's certificate can be verified with the proper SNI.</p>
Concurrency (optional)	<p>10</p> <hr/> <p>Number of targets to check concurrently in active health checks.</p>

Monitor with

GRAFANA

Step 1 / 12



The screenshot shows the Kong Admin UI interface. On the left, a dark sidebar lists various management sections: CONSUMERS, PLUGINS (highlighted with a red arrow), UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. At the top, a navigation bar includes tabs for Authentication, Security, Traffic Control, Serverless, Analytics & Monitoring (highlighted with a red arrow), Transformations, Logging, and Other. The main content area is titled "Analytics & Monitoring" and describes visualizing, inspecting, and monitoring APIs and microservices traffic. It features three cards for Datadog, Prometheus, and Zipkin, each with a "ADD PLUGIN" button. The Prometheus card's "ADD PLUGIN" button is also highlighted with a red arrow.

Step 2 / 12

ADD PROMETHEUS

Configure the Plugin.

consumer

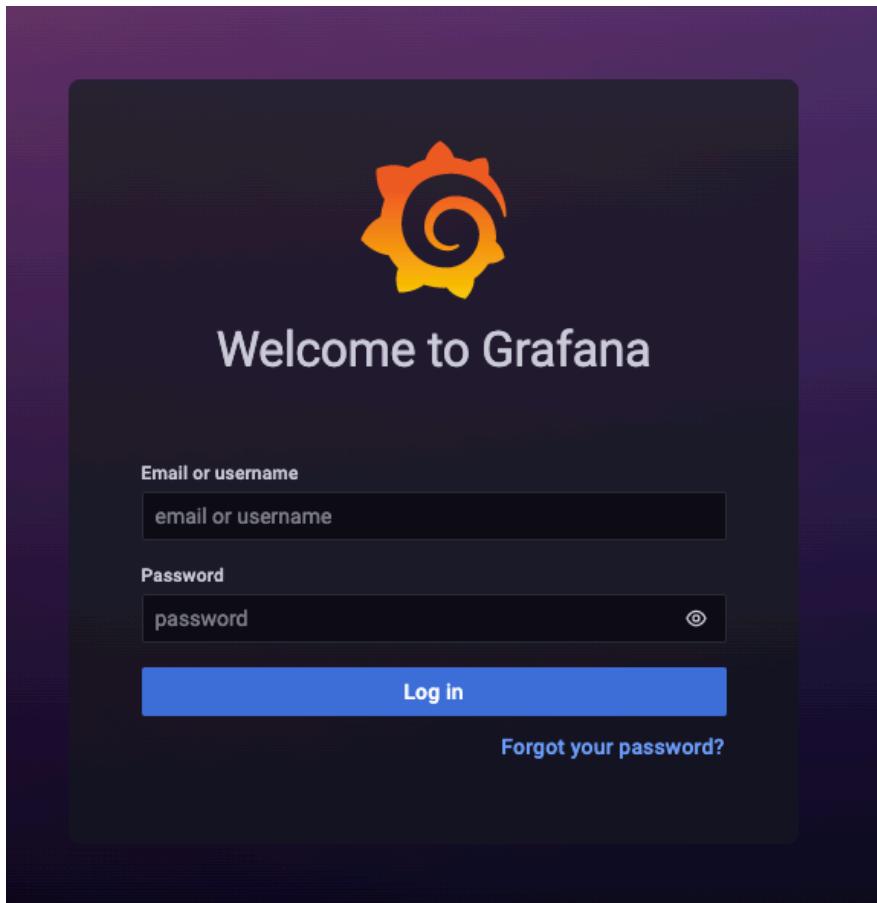
The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

per consumer

NO

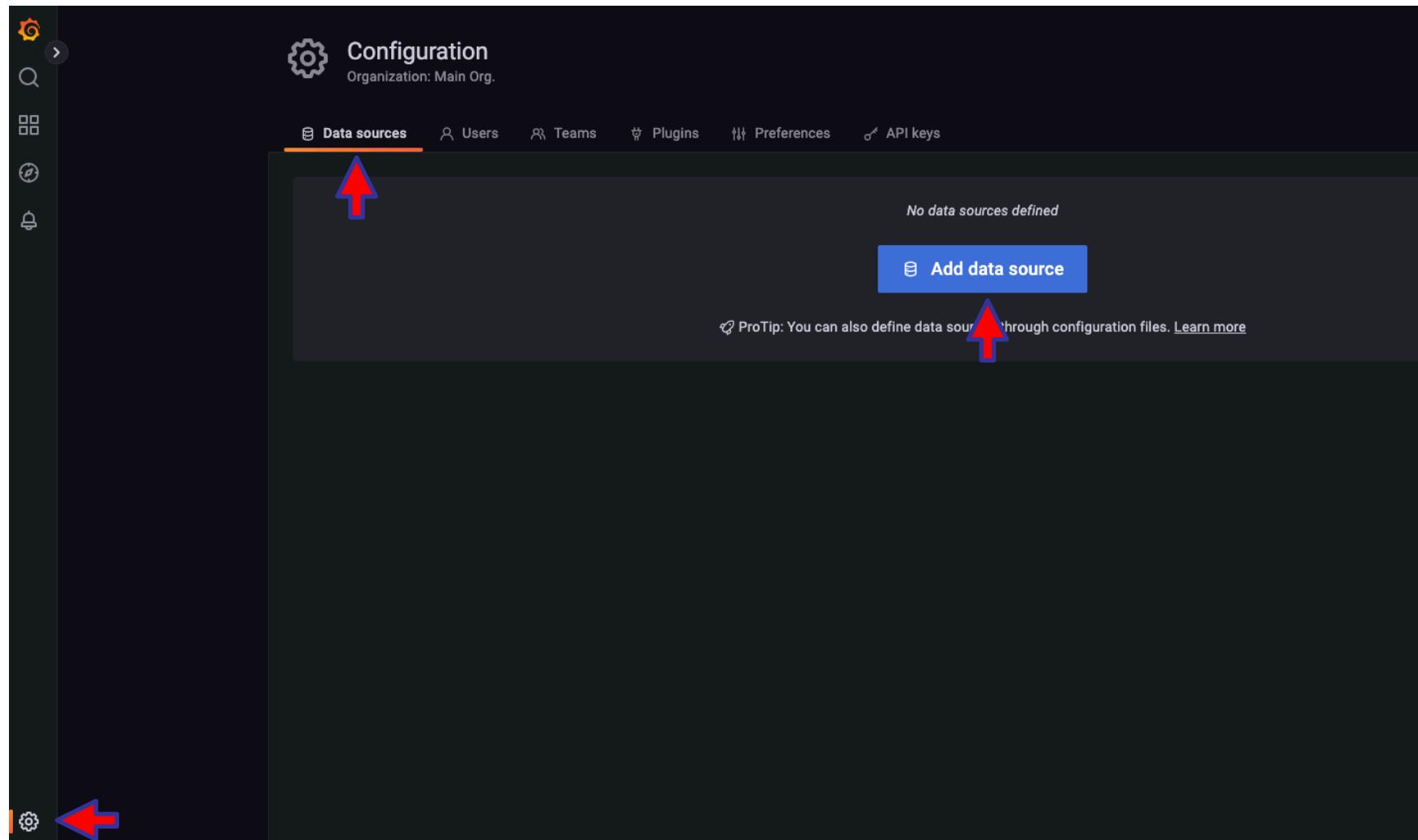
ADD PLUGIN

Step 3 / 12

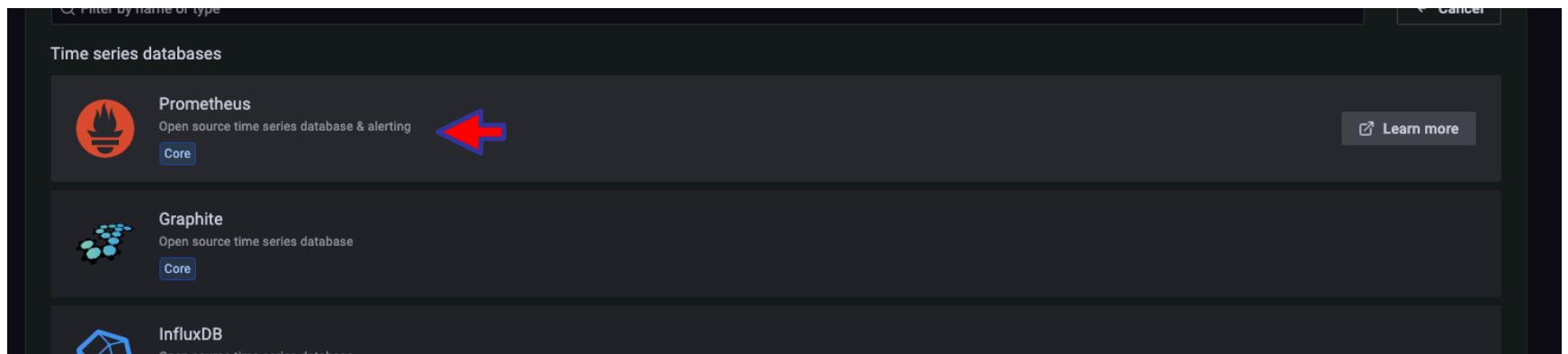


```
url: http://localhost:3000
username : admin
password : admin
```

Step 4 / 12



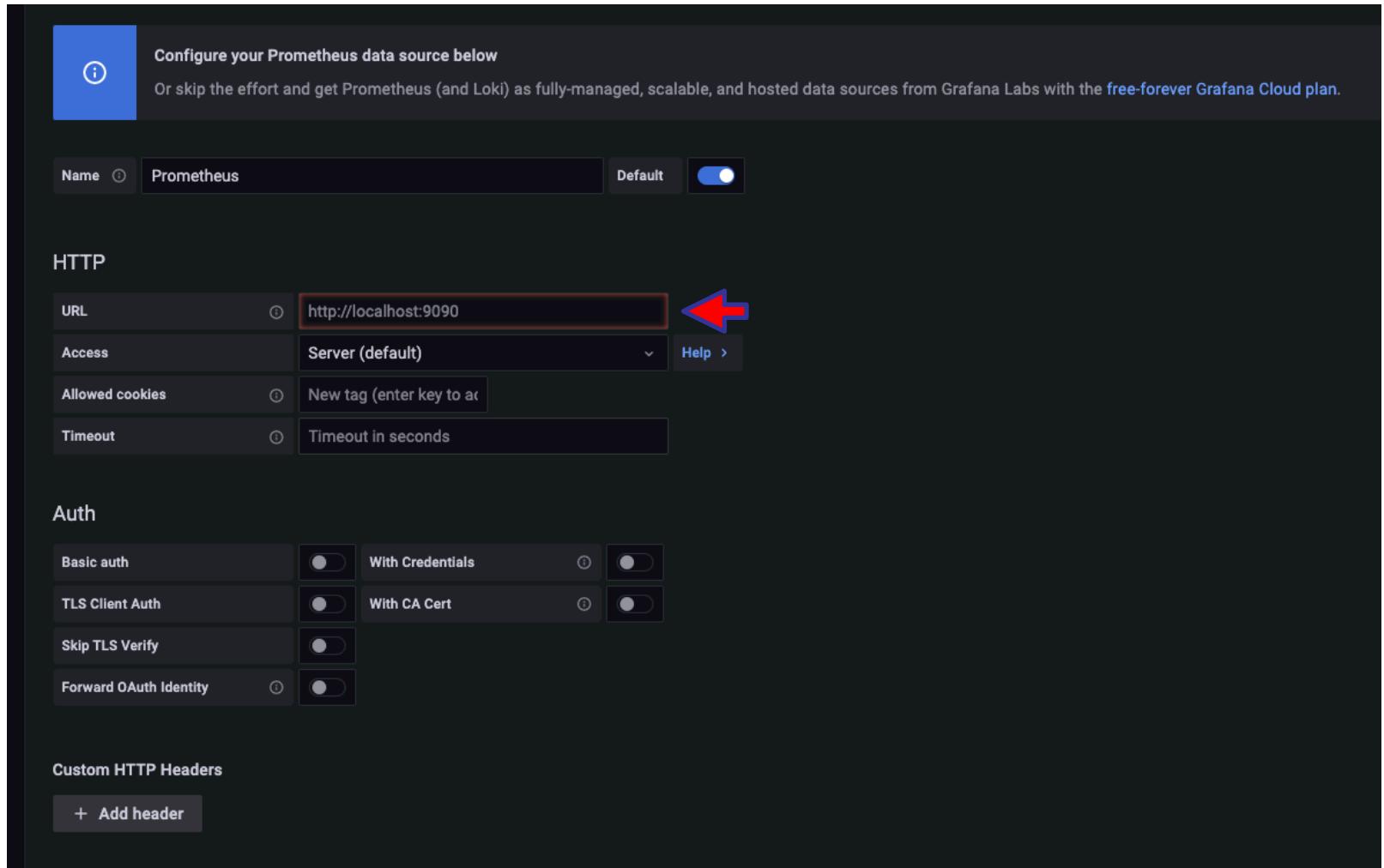
Step 5 / 12



The screenshot shows a list of time series databases. The first item, "Prometheus", is highlighted with a large red arrow pointing to its name. The Prometheus entry includes a red icon of a flame, a brief description ("Open source time series database & alerting"), a "Core" button, and a "Learn more" button.

Time series databases	
	Prometheus Open source time series database & alerting Core
	Graphite Open source time series database Core
	InfluxDB Open source time series database

Step 6 / 12



Configure your Prometheus data source below
Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the [free-forever Grafana Cloud plan](#).

Name (i) Prometheus Default (on)

HTTP

URL	<input type="text" value="http://localhost:9090"/>	(i)
Access	Server (default)	v Help >
Allowed cookies	New tag (enter key to ac	(i)
Timeout	Timeout in seconds	(i)

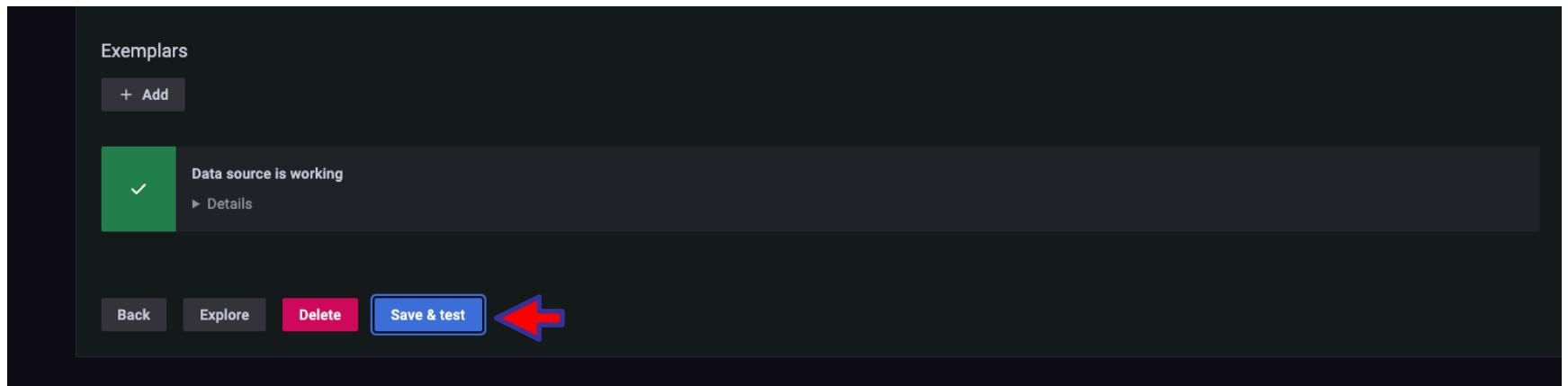
Auth

Basic auth	(on)	With Credentials	(i)	(on)
TLS Client Auth	(on)	With CA Cert	(i)	(on)
Skip TLS Verify	(on)			
Forward OAuth Identity	(i)	(on)		

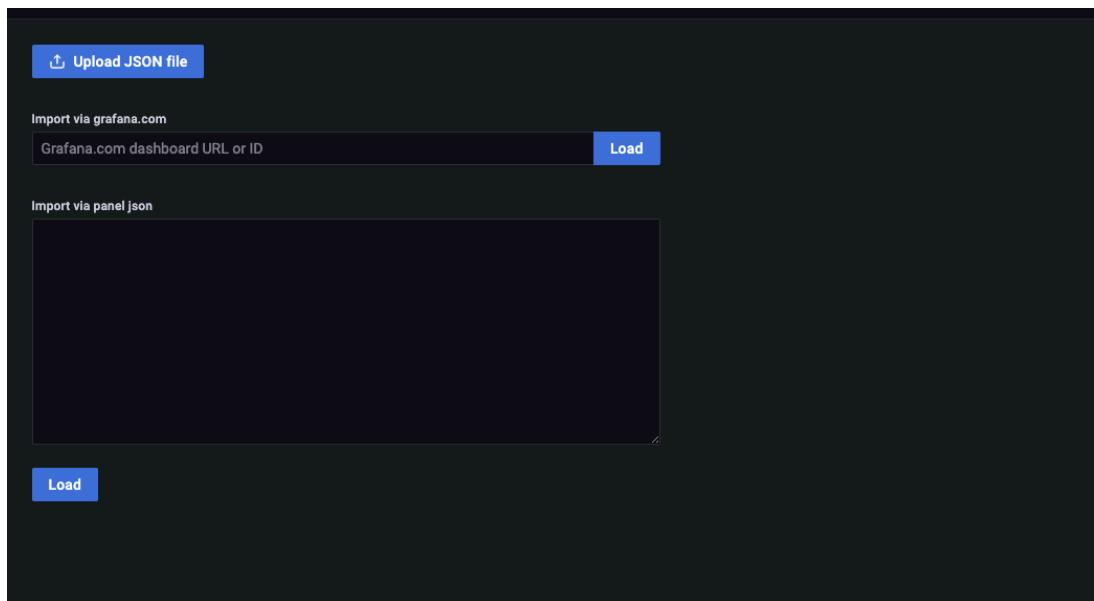
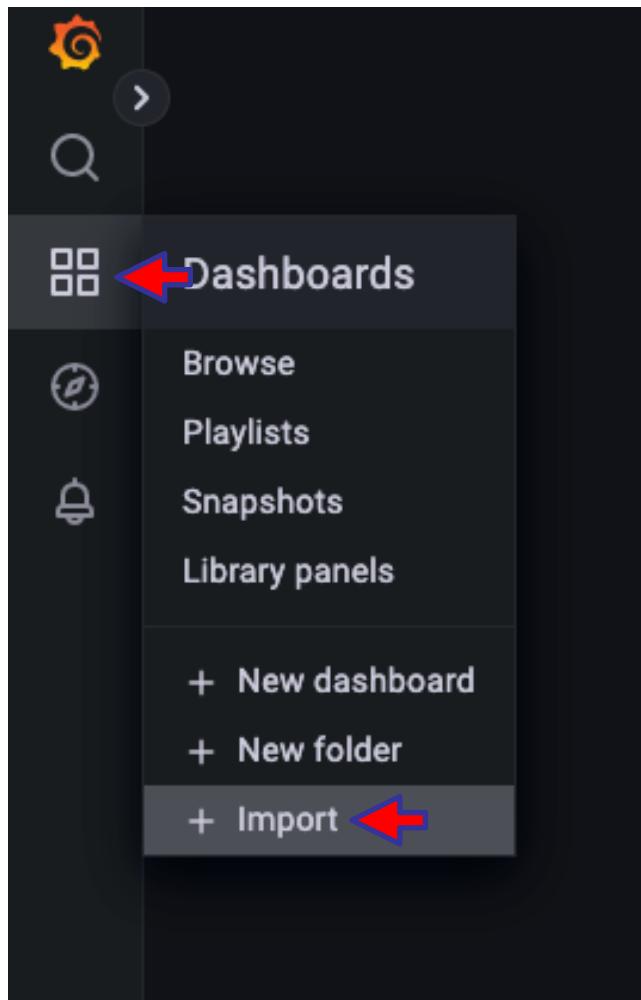
Custom HTTP Headers

+ Add header

Step 7 / 12



Step 8 / 12



The screenshot shows the 'Import' dialog box. It has two main sections:

- Import via grafana.com**: A text input field labeled 'Grafana.com dashboard URL or ID' with a 'Load' button to its right.
- Import via panel json**: A large empty text area with a 'Load' button at the bottom.

Step 9 / 12

grafana.com/grafana/dashboards/7424

GrafanaLabs Products Open source Solutions Learn Company Downloads Contact us Sign in

All dashboards » Kong (official)

 **Kong (official)** by konghq

 DASHBOARD

Dashboard that graphs metrics exported via Prometheus plugin in Kong (<http://github.com/kong/kong-plugin-prometheus>)

Last updated: a year ago

Start with Grafana Cloud and the new FREE tier. Includes 10K series Prometheus or Graphite Metrics and 50gb Loki Logs

Downloads: 37283
Reviews: 4

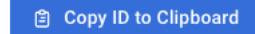


Add your review!

Overview Revisions Reviews

Get this dashboard:

7424

 Copy ID to Clipboard

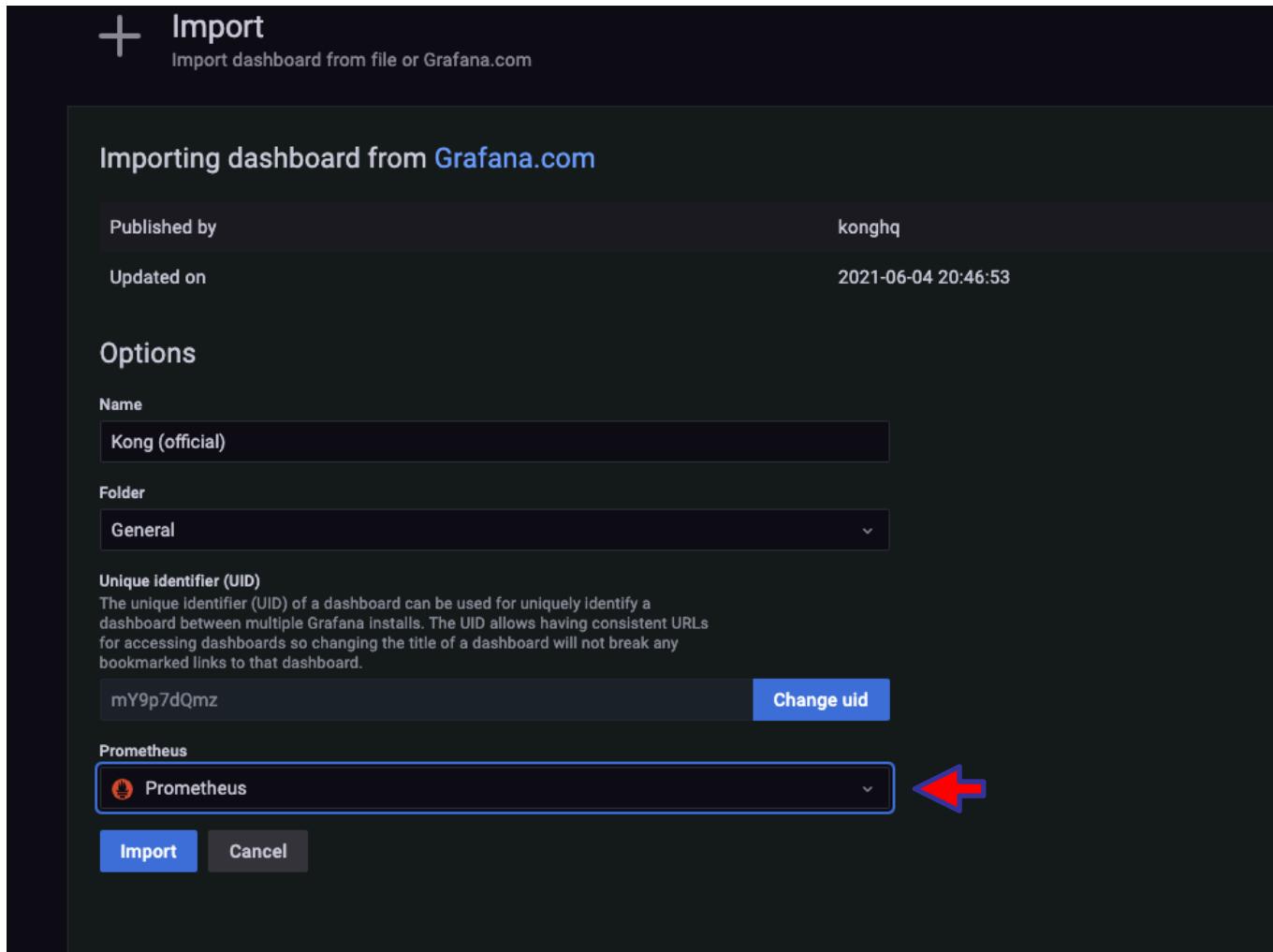
Dashboard graphing metrics collected via <http://github.com/kong/kong-prometheus-plugin>

Download JSON
How do I import this dashboard?

Dependencies:

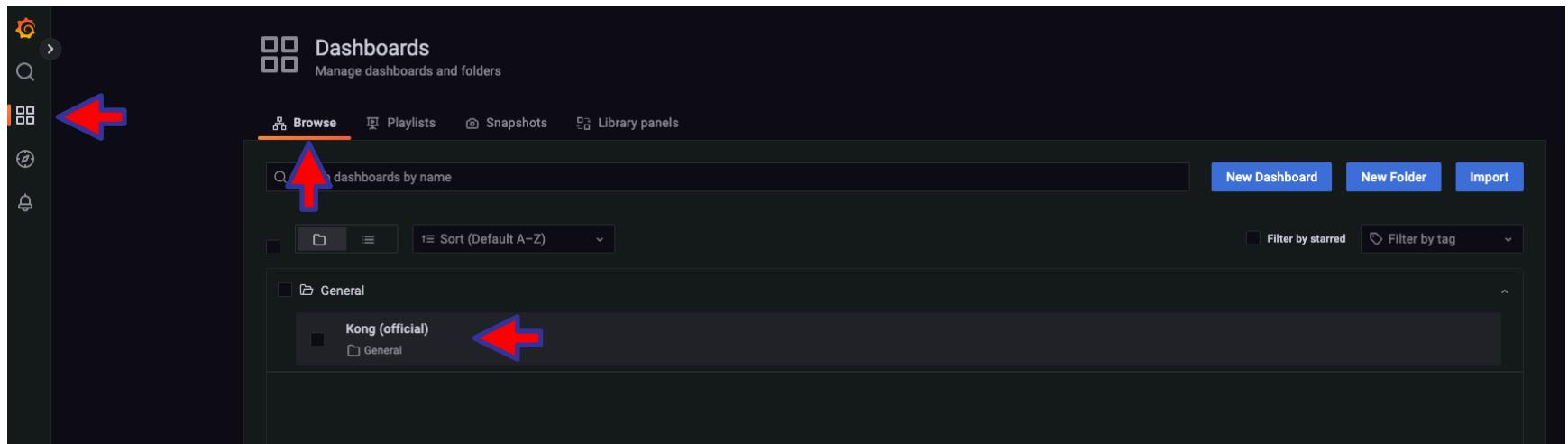


Step 10 / 12

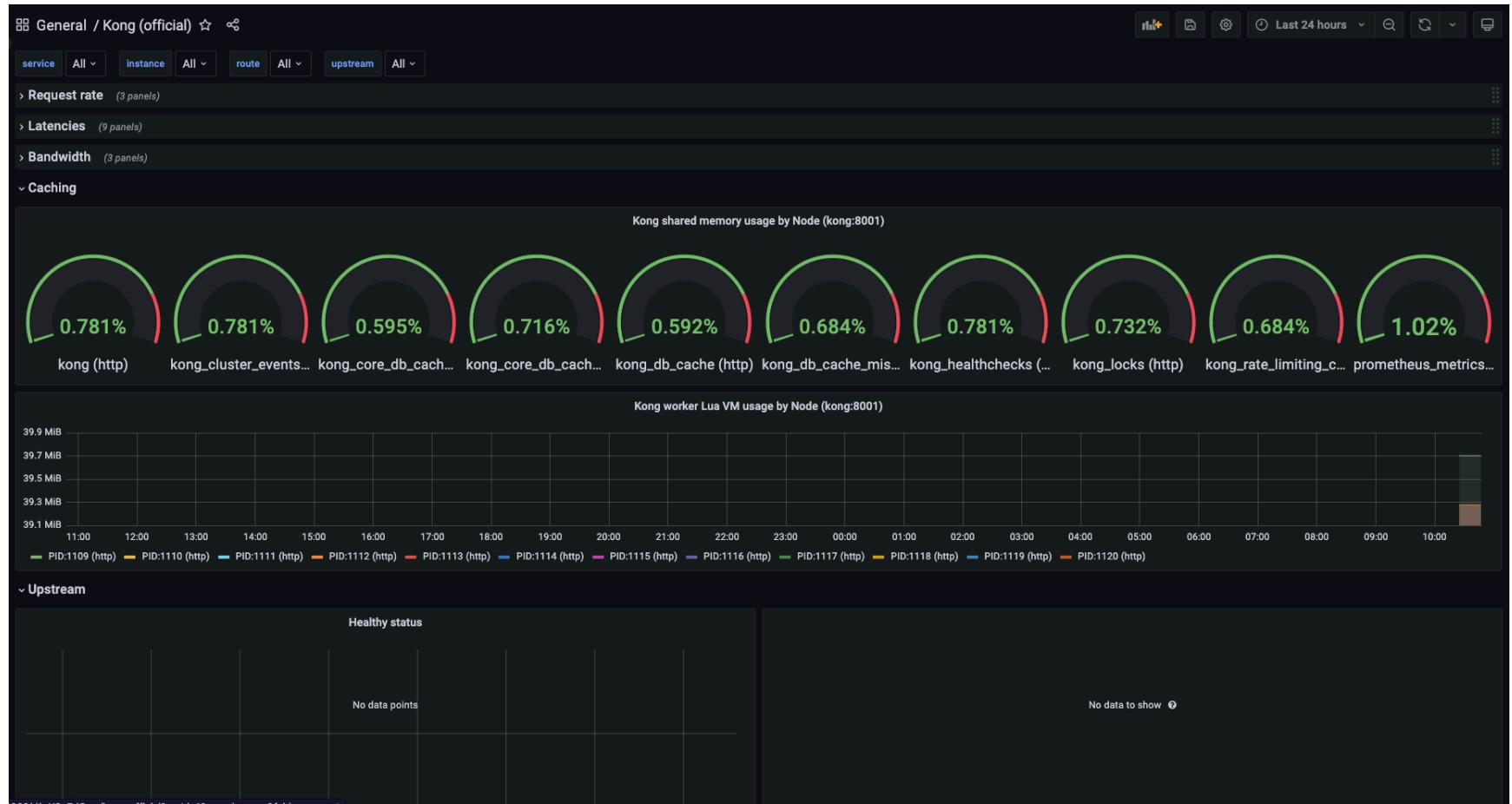


The screenshot shows the 'Import' dialog in Grafana. At the top, there's a plus icon and the word 'Import'. Below that, it says 'Import dashboard from file or Grafana.com'. The main title is 'Importing dashboard from Grafana.com'. It shows details: 'Published by konghq' and 'Updated on 2021-06-04 20:46:53'. Under 'Options', there are fields for 'Name' (set to 'Kong (official)') and 'Folder' (set to 'General'). There's also a section for 'Unique identifier (UID)' with a text input 'mY9p7dQmz' and a 'Change uid' button. At the bottom, there's a 'Prometheus' dropdown set to 'Prometheus', which has a red arrow pointing to it. Finally, there are 'Import' and 'Cancel' buttons at the bottom.

Step 11 / 12



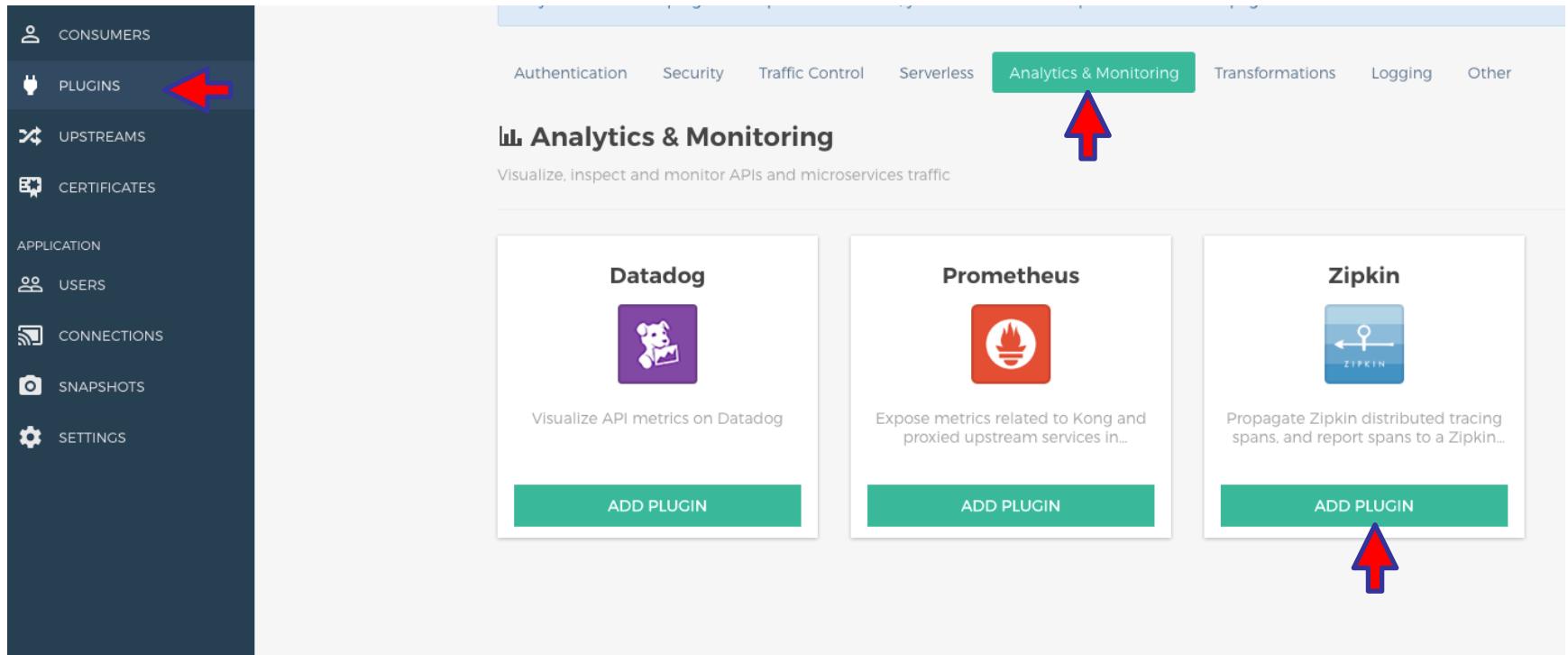
Step 12 / 12



Monitor with

ZIPKIN

Step 1 / 2



The screenshot shows the Kong Admin UI interface. On the left, a dark sidebar lists various management sections: CONSUMERS, PLUGINS (highlighted with a red arrow), UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area is titled "Analytics & Monitoring" (also highlighted with a red arrow) and includes a sub-header: "Visualize, inspect and monitor APIs and microservices traffic". Below this, three plugin cards are displayed: Datadog, Prometheus, and Zipkin. Each card features a logo, a brief description, and a green "ADD PLUGIN" button. A large red arrow points specifically to the "ADD PLUGIN" button for the Zipkin card.

Step 2 / 2

can identify the user making the request. If left blank, the plugin will be applied to all consumers.

local service name kong

http endpoint http://zipkin:9411/api/v2/spans

sample ratio 1

default service name

include credential YES

traceid byte count 16

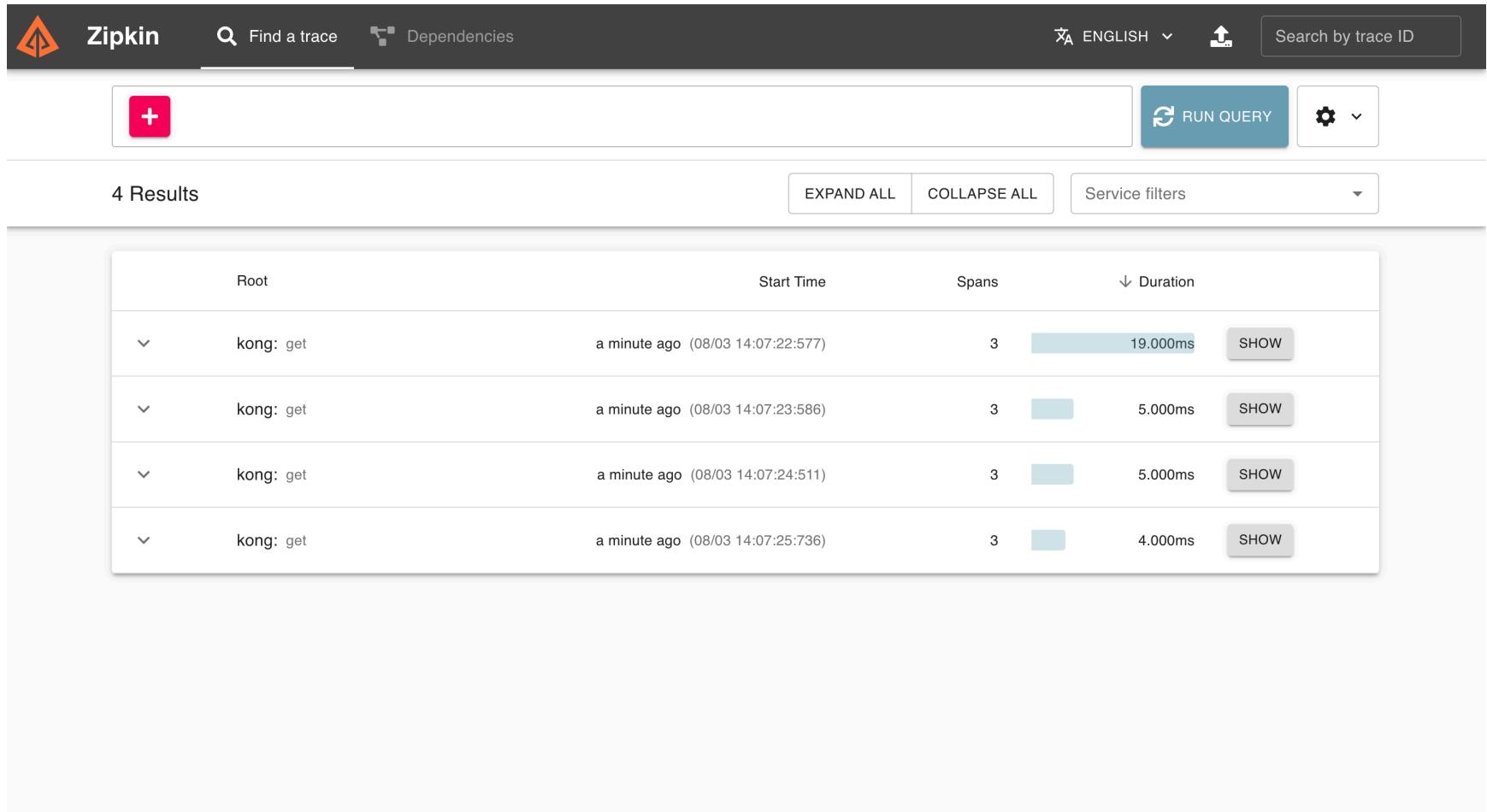
header type preserve

default header type b3

tags header Zipkin-Tags



<host>:9411/zipkin



The screenshot shows the Zipkin web interface with the following details:

Zipkin Navigation: Find a trace, Dependencies, English, Search by trace ID.

Search Bar: A red button with a white plus sign (+), a search icon, and a run query button.

Results Summary: 4 Results, EXPAND ALL, COLLAPSE ALL, Service filters.

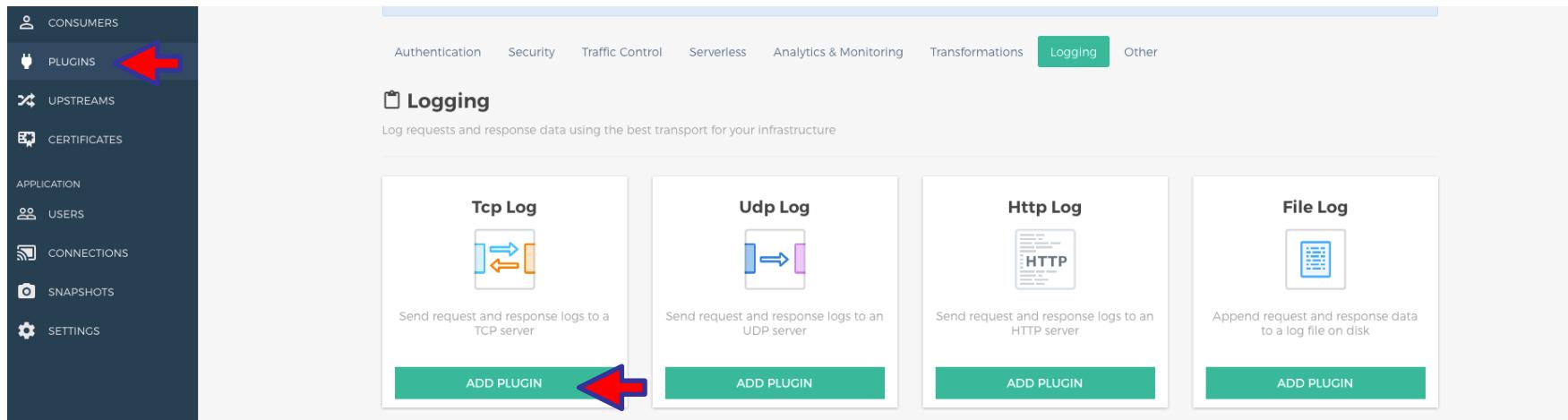
Table Headers: Root, Start Time, Spans, Duration.

Table Data:

Root	Start Time	Spans	Duration	Action
kong: get	a minute ago (08/03 14:07:22:577)	3	19.000ms	SHOW
kong: get	a minute ago (08/03 14:07:23:586)	3	5.000ms	SHOW
kong: get	a minute ago (08/03 14:07:24:511)	3	5.000ms	SHOW
kong: get	a minute ago (08/03 14:07:25:736)	3	4.000ms	SHOW

Logging with
ELK STACK

Step 1 / 6



The screenshot shows the API Management interface with the 'Logging' tab selected. On the left sidebar, the 'PLUGINS' option is highlighted with a red arrow. The main content area displays four logging options: 'Tcp Log', 'Udp Log', 'Http Log', and 'File Log'. Each option has a corresponding icon and a brief description. Below each icon is a green 'ADD PLUGIN' button, which is also highlighted with a red arrow.

Log Logging

Log requests and response data using the best transport for your infrastructure

Tcp Log

Send request and response logs to a TCP server

Udp Log

Send request and response logs to an UDP server

Http Log

Send request and response logs to an HTTP server

File Log

Append request and response data to a log file on disk

ADD PLUGIN

ADD PLUGIN

ADD PLUGIN

ADD PLUGIN

Step 2 / 6

EDIT TCP LOG X

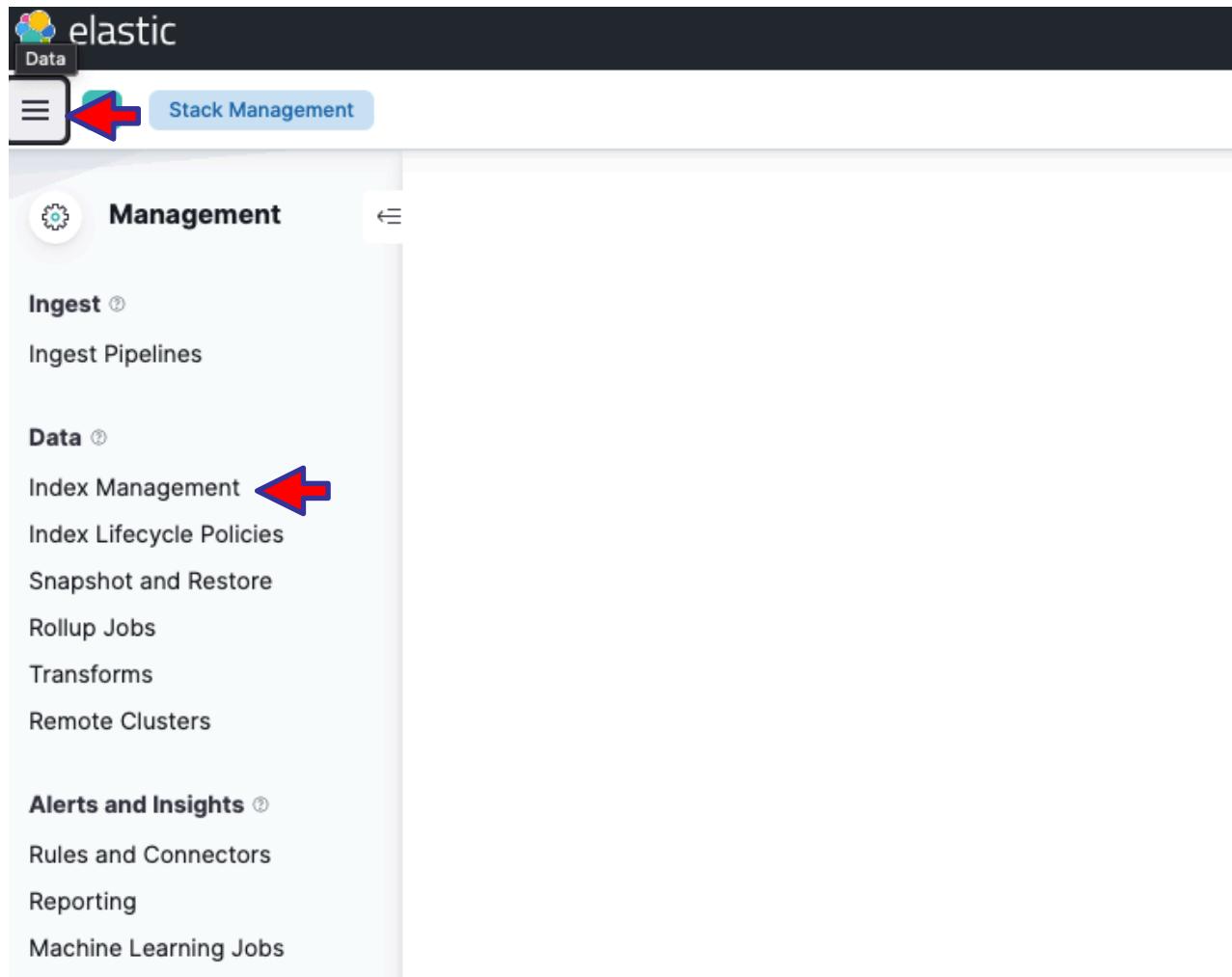
Configure the Plugin. ENABLED

consumer

The CONSUMER ID that this plugin configuration will target. This value can only be used if authentication has been enabled so that the system can identify the user making the request. If left blank, the plugin will be applied to all consumers.

host	elk	
port	5044	
timeout	10000	
keepalive	60000	
tls	<input type="radio"/> NO	
tls sni		
custom fields by lua		
 SUBMIT CHANGES		

Step 3 / 6 [<host>:5601]



Step 4 / 6

Index Management

Indices Data Streams Index Templates Component Templates

Update your Elasticsearch indices individually or in bulk. [Learn more.](#)

Include

<input type="checkbox"/>	Name	Health	Status	Primaries	Replicas	Docs count	St
<input type="checkbox"/>	elk-2022.06.22	● yellow	open	1	1	3	19

Rows per page: 10 ▾



Step 5 / 6

Kibana ⓘ

- Data Views 
- Saved Objects
- Tags
- Search Sessions
- Spaces
- Advanced Settings

Stack ⓘ

- License Management

You have data in Elasticsearch.
Now, create a data view.

Kibana requires a data view to identify which data streams, indices, and index aliases you want to explore. A data view can point to a specific index, for example, your log data from yesterday, or all indices that contain your log data.

 + Create data view



Want to learn more? [Read documentation ↗](#)

Step 6 / 6

Create data view

Name 

Enter an index pattern that matches one or more data sources. Use an asterisk (*) to match multiple characters. Spaces and the characters , /, ?, ", <, >, | are not allowed.

Timestamp field 

Select a timestamp field for use with the global time filter.

Show advanced settings

[Create data view](#) 

[Close](#)

✓ Your index pattern matches 1 source.

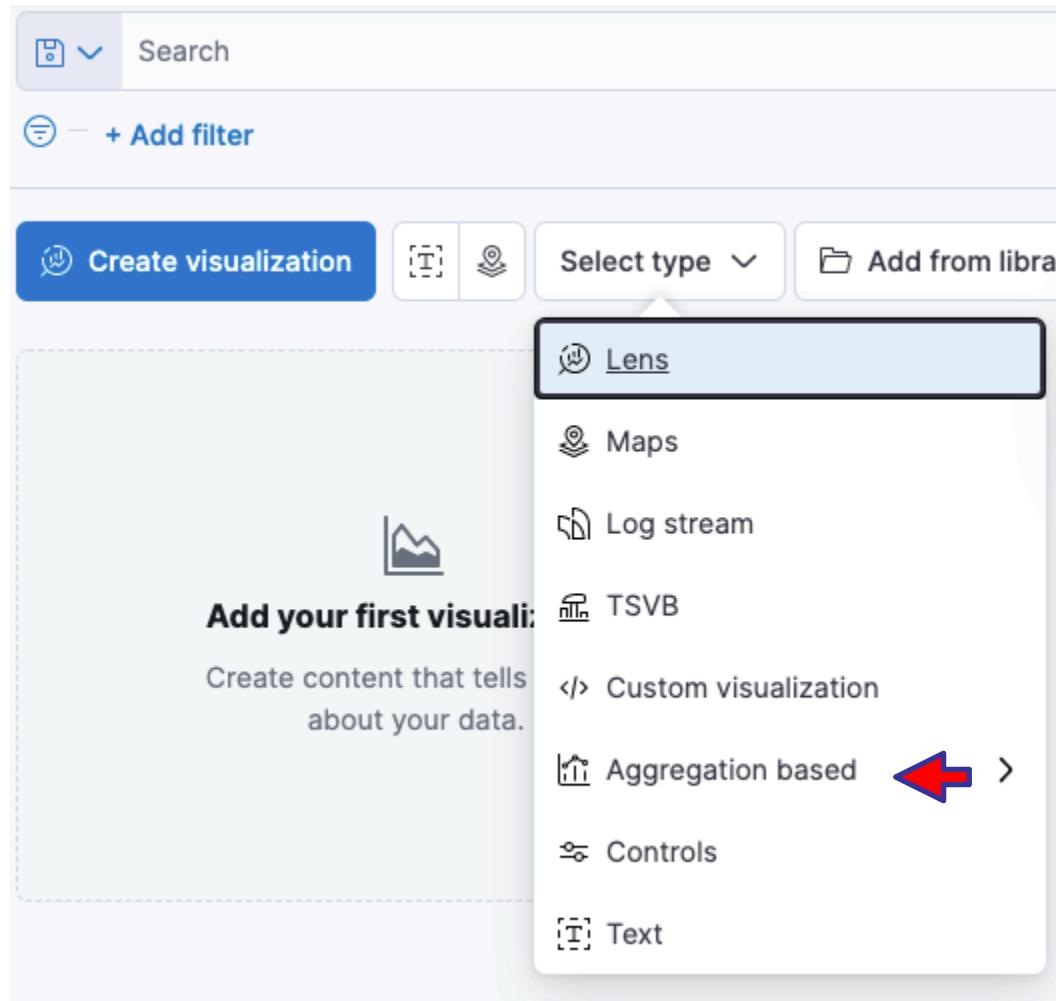
elk-2022.06.22 Index

Rows per page: 10 

Create Dashboard with

KIBANA

Step 1 / 7



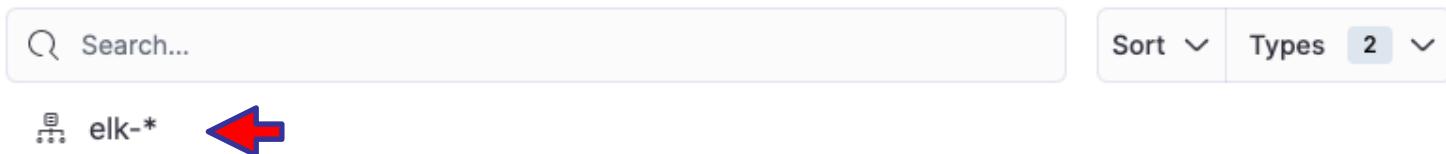
The screenshot shows a user interface for creating visualizations. At the top, there is a search bar and a filter section with a '+ Add filter' button. Below the search bar are three buttons: 'Create visualization' (blue), 'Select type' (grey), and 'Add from library' (grey). A dropdown menu is open under 'Select type', listing several options: 'Lens' (highlighted with a black border), 'Maps', 'Log stream', 'TSVB', 'Custom visualization', 'Aggregation based' (highlighted with a blue arrow pointing to it), 'Controls', and 'Text'. On the left side of the interface, there is a dashed box containing the text 'Add your first visualization' and 'Create content that tells about your data.' with a small chart icon.

Step 2 / 7

New Vertical bar / Choose a source

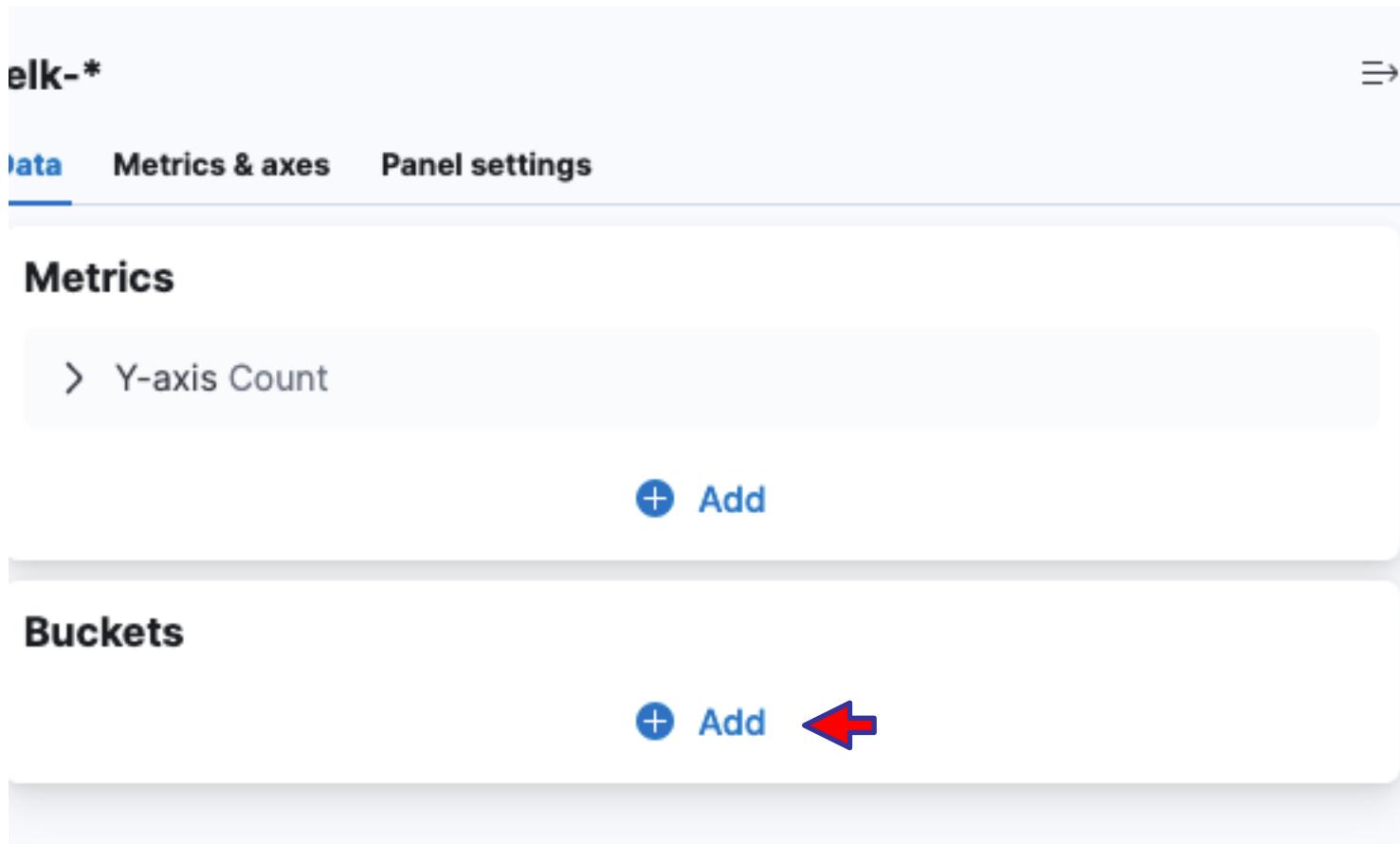
[Select a different visualization](#)

Sort ▾ Types 2 ▾



X

Step 3 / 7



The screenshot shows a user interface for configuring a visualization, likely a dashboard or report. At the top, there is a header with the text "elk-*" and a three-dot menu icon. Below the header, there are three tabs: "Data" (which is selected), "Metrics & axes", and "Panel settings".

The main area is divided into two sections: "Metrics" and "Buckets".

- Metrics Section:** Contains a list item "Y-axis Count" preceded by a right-pointing arrow icon.
- Buckets Section:** Contains a blue "Add" button with a plus sign icon. A red arrow points to this "Add" button.

Step 4 / 7

Buckets

Split chart ( )

Rows Columns

Aggregation Terms help 

Terms

Field

service.name.keyword

Order by

Metric: Count

Order Size

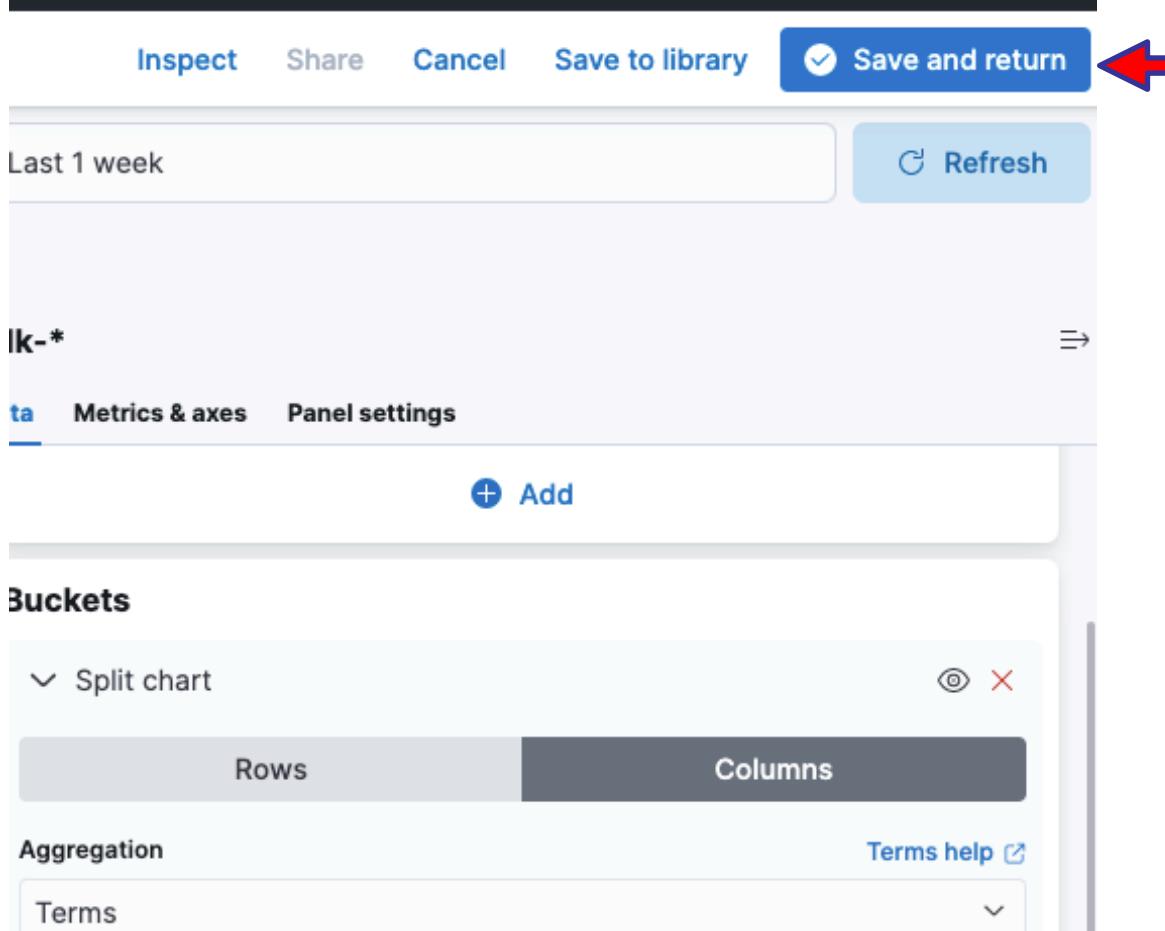
Descending 5

Group other values in separate bucket

Show missing values

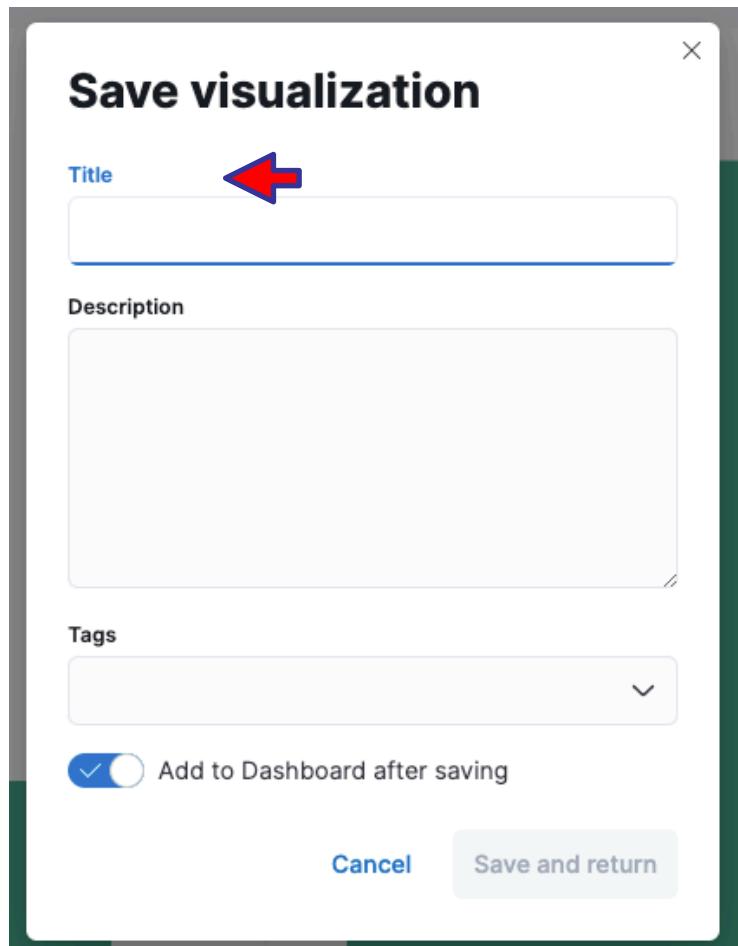
Custom label

Step 5 / 7

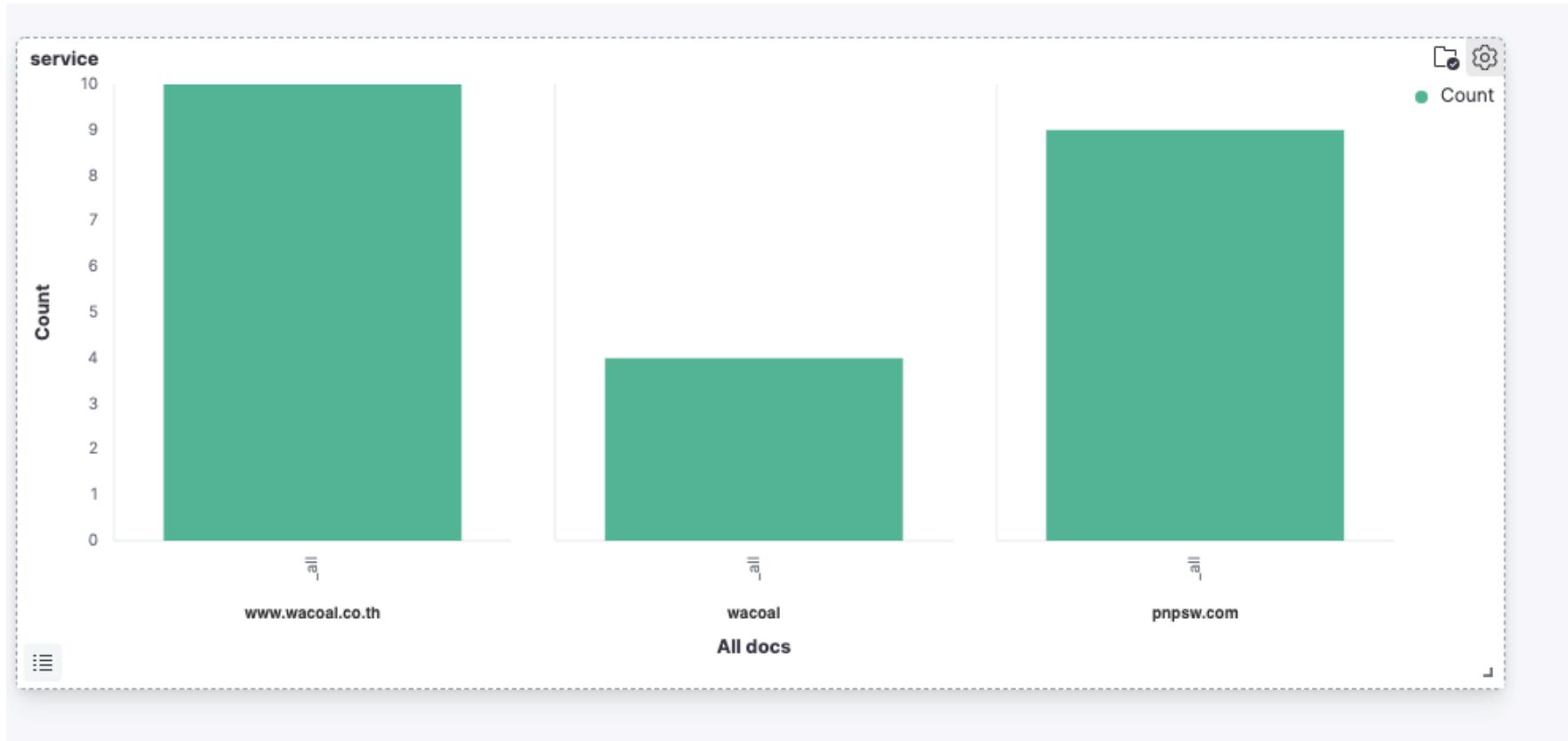


The screenshot shows a user interface for a data visualization tool. At the top, there is a navigation bar with the following buttons: Inspect, Share, Cancel, Save to library, and a prominent blue button labeled "Save and return" with a checkmark icon. A red arrow points to the "Save and return" button. Below the navigation bar, there is a search bar containing the text "Last 1 week" and a "Refresh" button. The main area of the interface is titled "lk-*". It contains three tabs: "Metrics & axes" (which is selected), "Panel settings", and "Add". Below these tabs, there is a section titled "Buckets" with a "Split chart" option. Under "Split chart", there are two buttons: "Rows" and "Columns", with "Columns" being the active tab. At the bottom, there is an "Aggregation" section with a "Terms" dropdown menu and a "Terms help" link.

Step 6 / 7



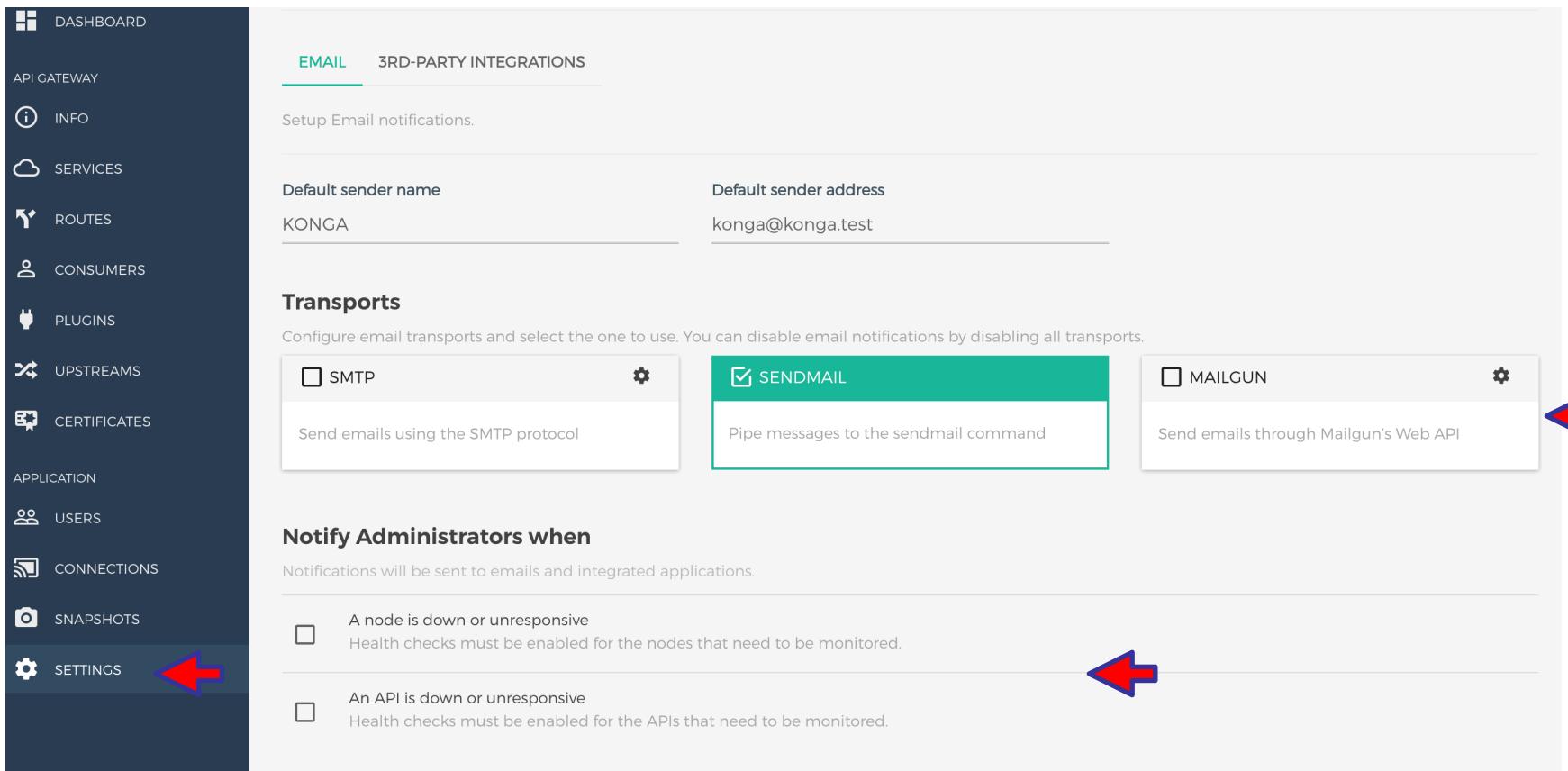
Step 7 / 7



Notify with

EMAIL

Config step

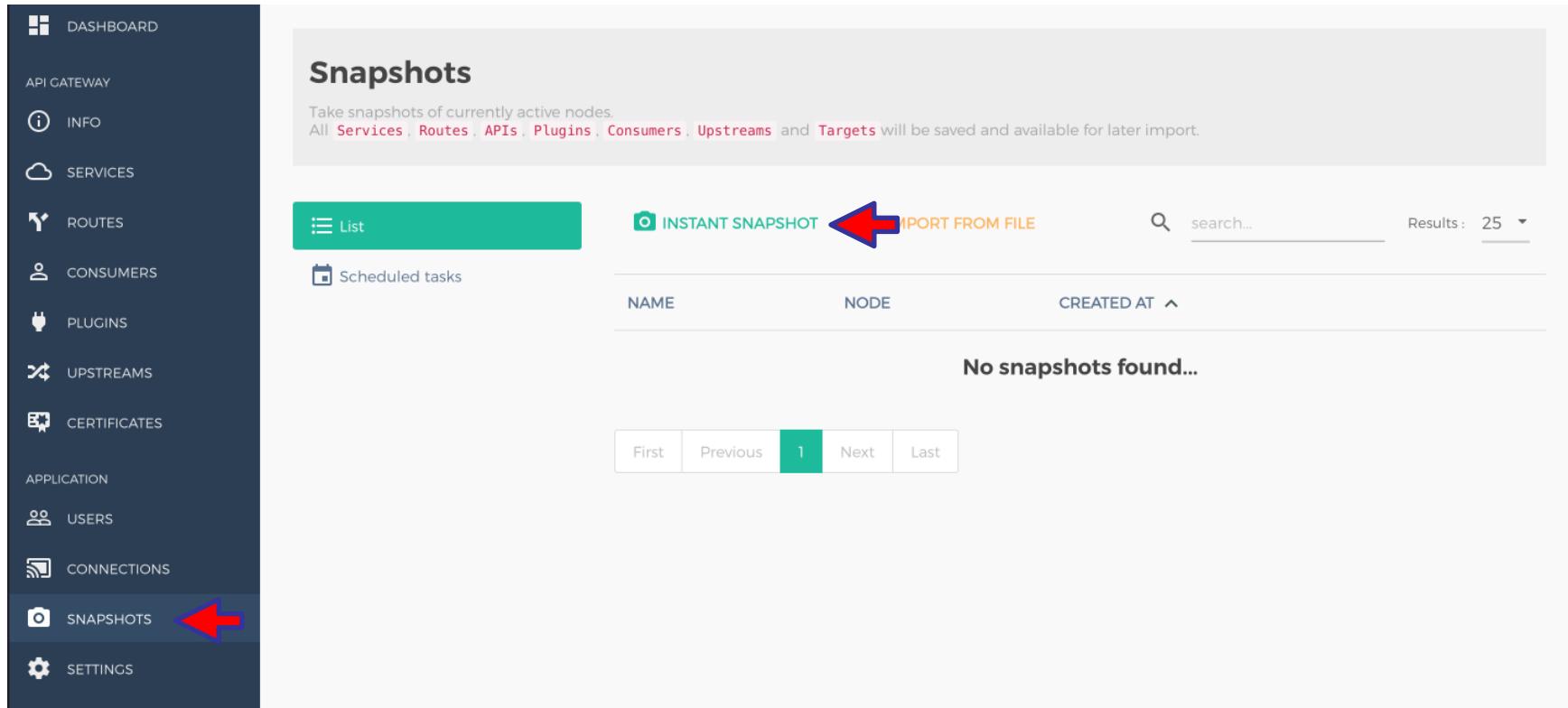


The screenshot shows the Kong configuration interface under the 'EMAIL' tab of the '3RD-PARTY INTEGRATIONS' section. On the left sidebar, the 'SETTINGS' option is highlighted with a red arrow. The main area displays the following configuration:

- Default sender name:** KONGA
- Default sender address:** konga@konga.test
- Transports:** Configure email transports and select the one to use. You can disable email notifications by disabling all transports.
 - SMTP:** Send emails using the SMTP protocol
 - SENDMAIL:** Pipe messages to the sendmail command
 - MAILGUN:** Send emails through Mailgun's Web API
- Notify Administrators when:** Notifications will be sent to emails and integrated applications.
 - A node is down or unresponsive
 - Health checks must be enabled for the nodes that need to be monitored.
 - An API is down or unresponsive
 - Health checks must be enabled for the APIs that need to be monitored.

Backup and recovery with
SNAPSHOTS

Manual Snapshot Step 1/3

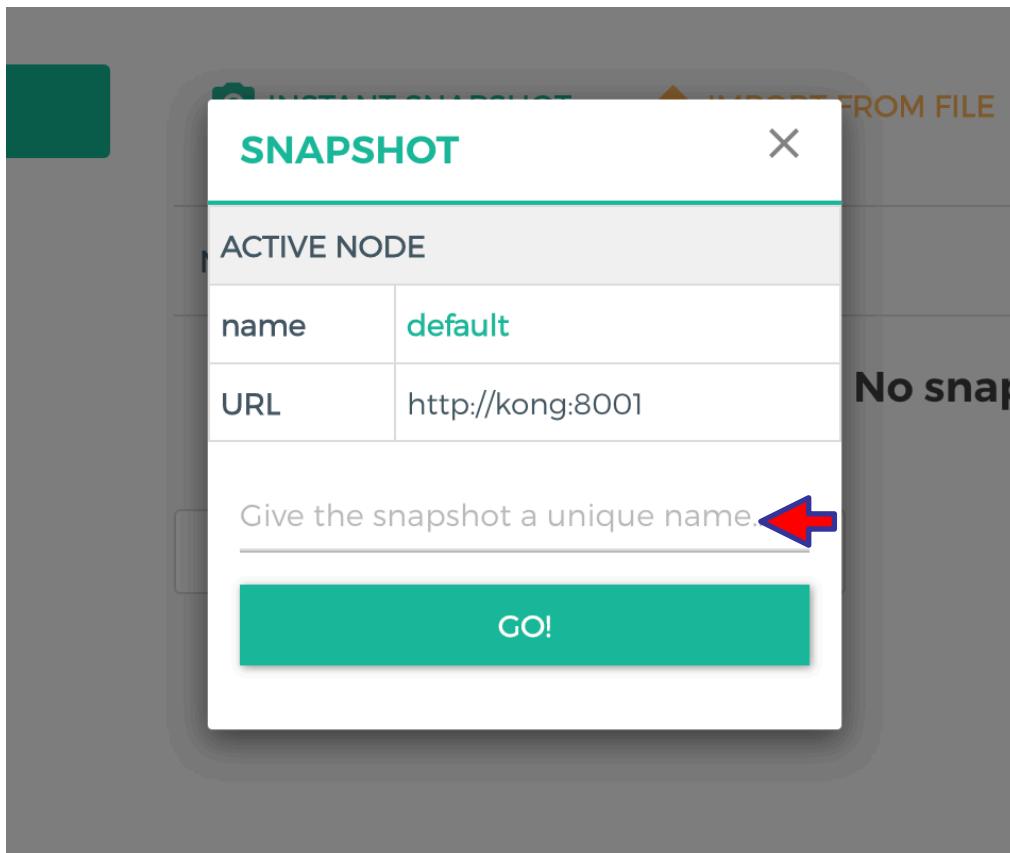


The screenshot shows the API Gateway dashboard with the 'SNAPSHOTS' section selected. The left sidebar lists various management options: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS (which has a red arrow pointing to it), and SETTINGS. The main content area is titled 'Snapshots' and contains instructions: 'Take snapshots of currently active nodes. All Services, Routes, APIs, Plugins, Consumers, Upstreams and Targets will be saved and available for later import.' Below this, there are two buttons: 'INSTANT SNAPSHOT' (highlighted with a red arrow) and 'IMPORT FROM FILE'. A search bar and a results dropdown are also present. The table below shows no snapshots found, with columns for NAME, NODE, and CREATED AT. Navigation links at the bottom include First, Previous, 1 (highlighted with a red box), Next, and Last.

NAME	NODE	CREATED AT
No snapshots found...		

First Previous 1 Next Last

Manual Snapshot Step 2/3



Manual Snapshot Step 3/3

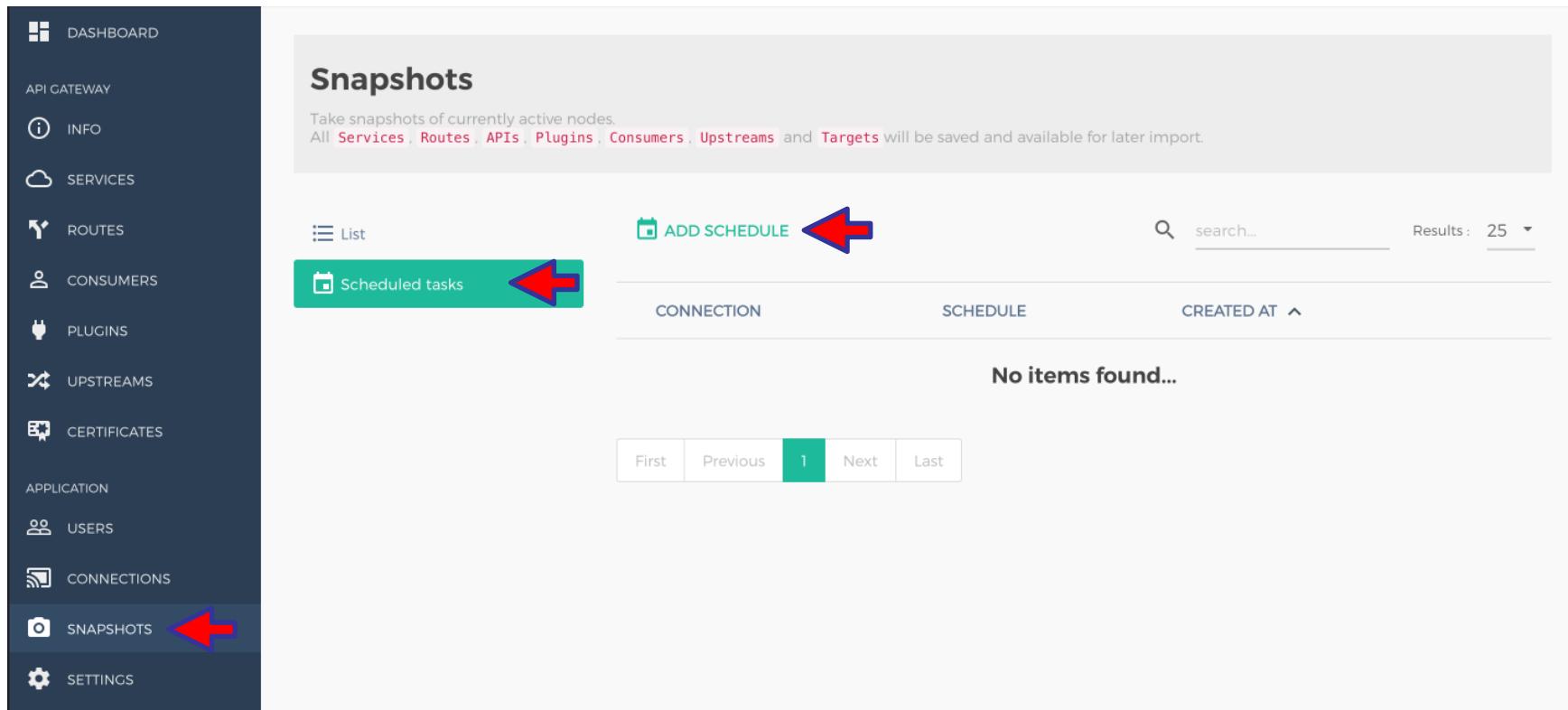
Snapshots

Take snapshots of currently active nodes.
All [Services](#), [Routes](#), [APIs](#), [Plugins](#), [Consumers](#), [Upstreams](#) and [Targets](#) will be saved and available for later import.

	NAME	NODE	CREATED AT	
 Scheduled tasks	2022-Aug	default v2.8.1	Aug 3, 2022 @10:55	 DETAILS  DELETE

First Previous 1 Next Last

Scheduled Task Step 1/3



The screenshot shows the API Gateway dashboard with the 'SNAPSHOTS' section highlighted. A red arrow points to the 'SNAPSHOTS' button in the sidebar. Another red arrow points to the 'ADD SCHEDULE' button in the main content area. The main content area also features a green 'List' button, a search bar, and a results count of 25 items.

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS ←

Snapshots

Take snapshots of currently active nodes.
All Services, Routes, APIs, Plugins, Consumers, Upstreams and Targets will be saved and available for later import.

List ADD SCHEDULE ←

Scheduled tasks

search... Results: 25

CONNEXION SCHEDULE CREATED AT ^

No items found...

First Previous 1 Next Last

Scheduled Task Step 2/3

NEW SCHEDULE

Cron

Schedule your snapshots by defining cron parameters ⓘ

*	*	*
Minute	Hour	Day of Month
*	*	
Month	Day of Week	

* Will run @ Every minute

Kong Node

Select the Kong Node / Connection to target

--- Please select a connection ---

Start immediately

CREATE SCHEDULE

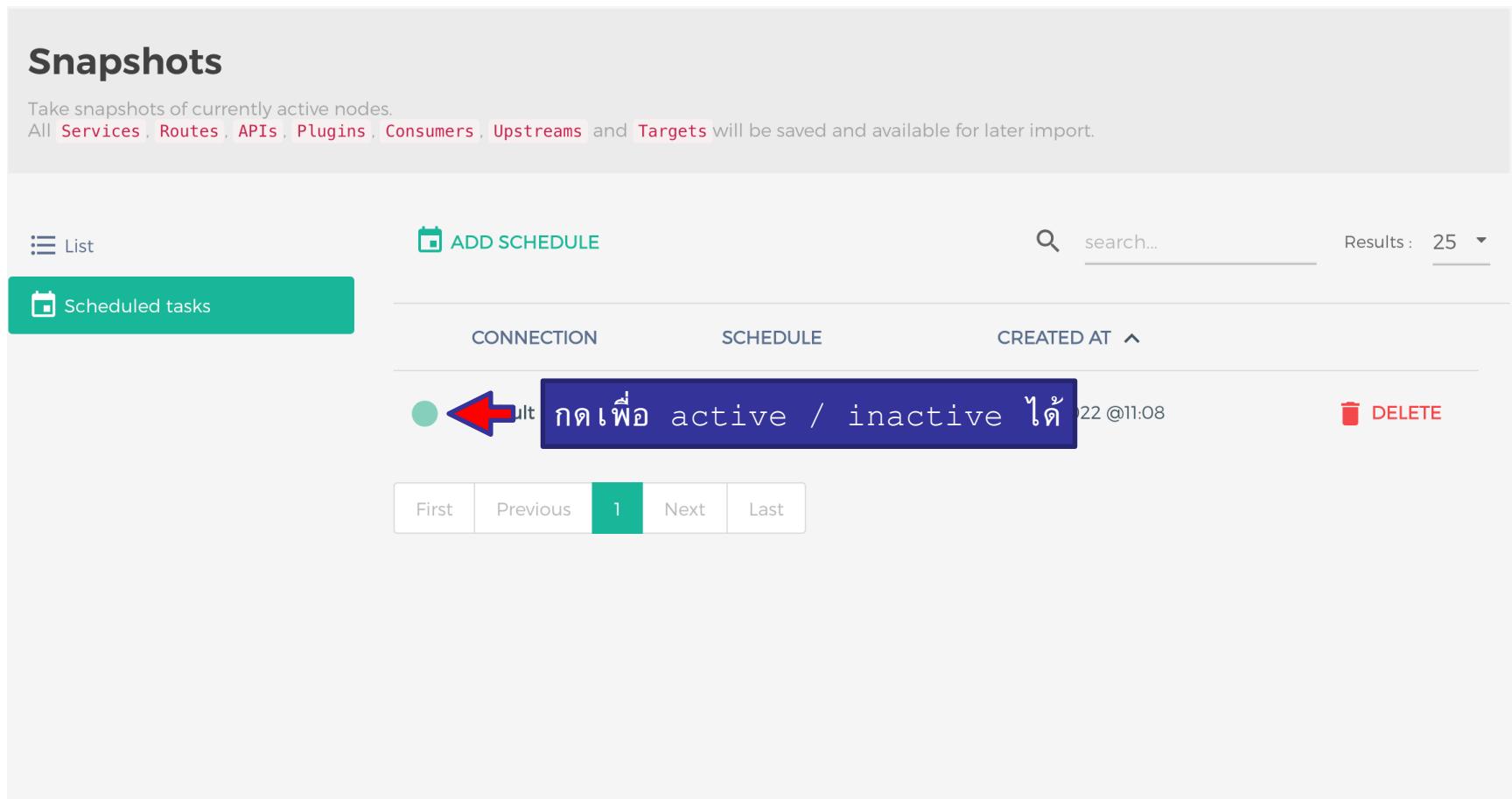


Scheduled Tasks Step 3/3

Snapshots

Take snapshots of currently active nodes.

All Services, Routes, APIs, Plugins, Consumers, Upstreams and Targets will be saved and available for later import.



The screenshot shows a web-based interface for managing scheduled tasks. At the top, there are navigation links: 'List' (grey), 'Scheduled tasks' (green, selected), and 'ADD SCHEDULE' (blue). A search bar and a results count of 'Results : 25' are also at the top. The main area displays a table with three columns: 'CONNECTION', 'SCHEDULE', and 'CREATED AT'. The first row in the table is highlighted with a blue background and contains the following data:

CONNECTION	SCHEDULE	CREATED AT
● 	กดเพื่อ active / inactive ได้	22 @11:08

Below the table is a navigation bar with buttons for 'First', 'Previous', '1' (highlighted in green), 'Next', and 'Last'.

Restore from snapshot step 1/3

Snapshots

Take snapshots of currently active nodes.
All [Services](#), [Routes](#), [APIs](#), [Plugins](#), [Consumers](#), [Upstreams](#) and [Targets](#) will be saved and available for later import.

NAME	NODE	CREATED AT	DETAILS	DELETE
snap@1659499800413	default v2.8.1	Aug 3, 2022 @11:10	 DETAILS	 DELETE

First Previous **1** Next Last

Restore from snapshot step 2/3

Snapshot Details

snapshots / snapshot details

snap@1659499800413@default

Kong 2.8.1 Aug 3, 2022

RESTORE EXPORT

SERVICES ROUTES CONSUMERS PLUGINS ACLS UPSTREAMS CERTIFICATES SNIS

 Search items...

port	3000
name	api
id	7c905e10-8774-4fed-8c8d-be3d8123d06f
connect_timeout	60000
enabled	true
tags	(Empty List)
ca_certificates	
path	
host	localhost



Restore from snapshot step 3/3

RESTORE SNAPSHOT

X

Select objects to import

Select the Objects you want to import to **default** or [select another node](#).

services routes consumers plugins acls upstreams certificates snis 

 **IMPORT OBJECTS** 

Export snapshots step 1/2

Snapshots

Take snapshots of currently active nodes.
All **Services**, **Routes**, **APIs**, **Plugins**, **Consumers**, **Upstreams** and **Targets** will be saved and available for later import.

[List](#) [!\[\]\(fc66381e8947fbc4f8abf8242228945b_img.jpg\) INSTANT SNAPSHOT](#) [!\[\]\(98e932f2e88bcaed2a809359e219f601_img.jpg\) IMPORT FROM FILE](#) Results : 25

[!\[\]\(4e152d709af5a92e3d14e70a52a4616b_img.jpg\) Scheduled tasks](#)

NAME	NODE	CREATED AT	
2022-Aug	default v2.8.1	Aug 3, 2022 @10:55	  DETAILS  DELETE

First Previous [1](#) Next Last

Export snapshot step 2/2

Snapshot Details

snapshots / snapshot details

2022-JUN-13@default

Kong 2.8.1 Aug 3, 2022

↻ RESTORE EXPORT 

[SERVICES](#) [ROUTES](#) [CONSUMERS](#) [PLUGINS](#) [ACLS](#) [UPSTREAMS](#) [CERTIFICATES](#) [SNIS](#)

 Search items...

Import snapshot

Snapshots

Take snapshots of currently active nodes.
All [Services](#), [Routes](#), [APIs](#), [Plugins](#), [Consumers](#), [Upstreams](#) and [Targets](#) will be saved and available for later import.

[List](#)  [INSTANT SNAPSHOT](#)  [IMPORT FROM FILE](#)  Results : 25

[Scheduled tasks](#)

NAME	NODE	CREATED AT
No snapshots found...		

First Previous [1](#) Next Last



QUESTION AND ANSWERS