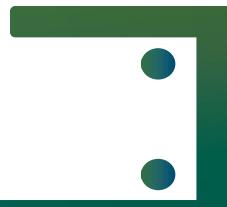




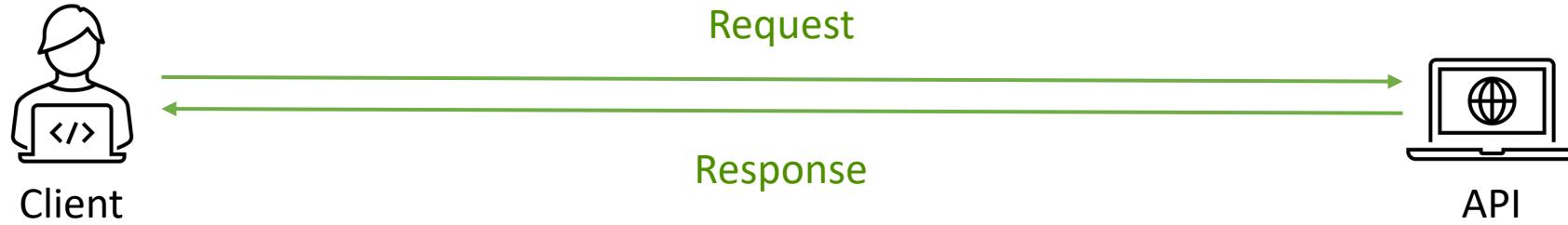
Kong API Gateway

Pleum - SDM

Prologue



Simplified API calling flow

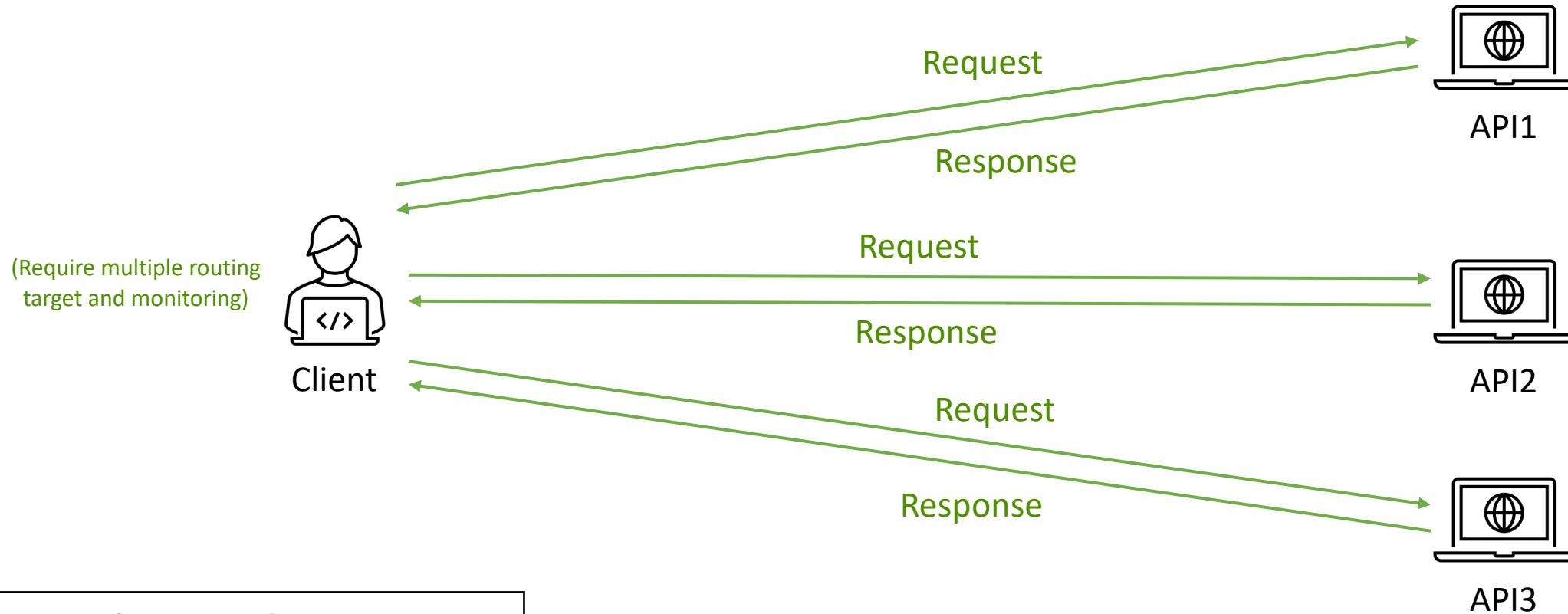


Prologue



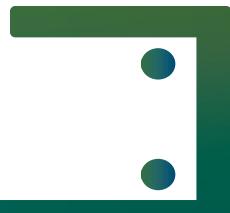
SIRISOFT
COMPANY LIMITED

Simplified API calling flow with multiple APIs

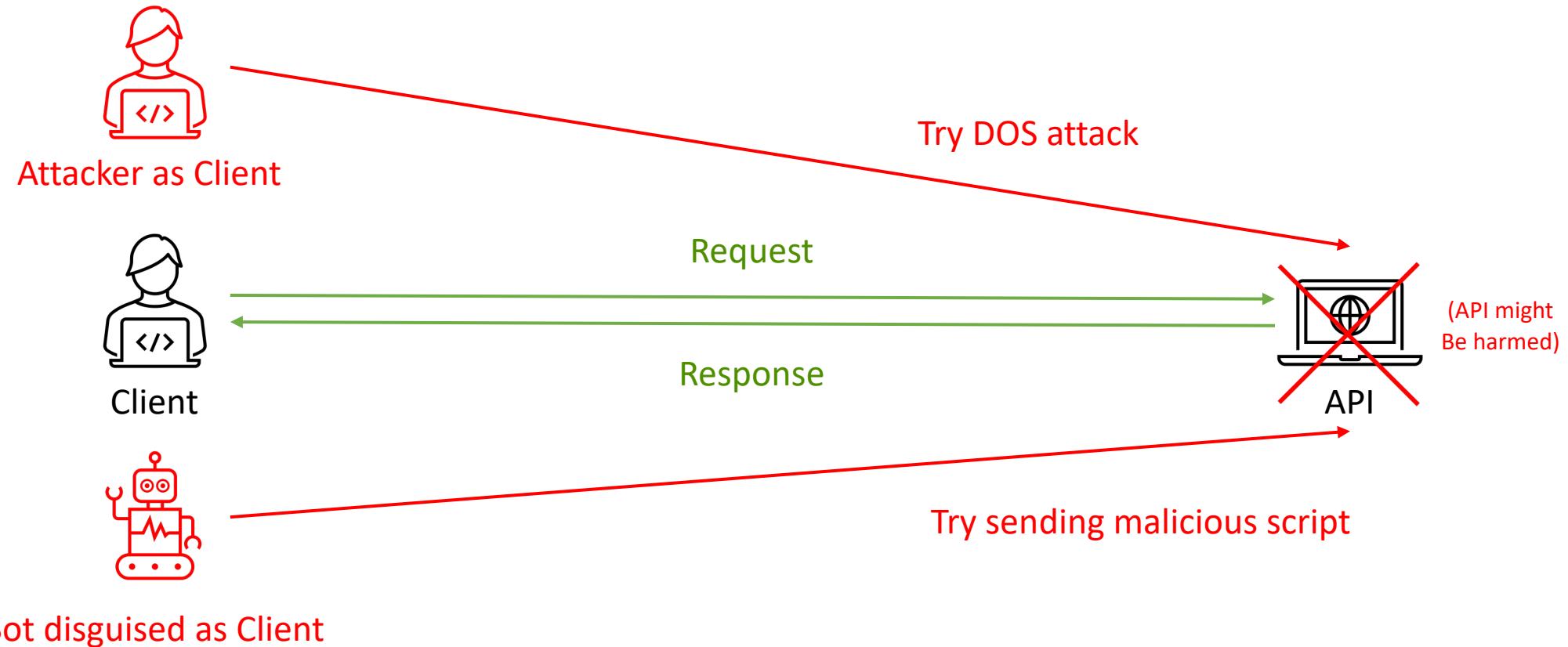


Complex and Tiresome

Prologue

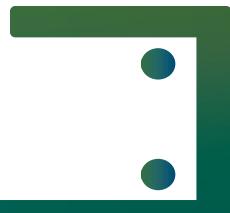


Simplified API calling flow with hostile clients



Security vulnerabilities

Prologue

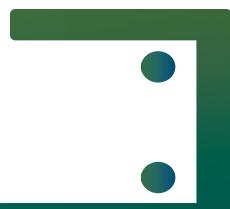


SIRISOFT
COMPANY LIMITED

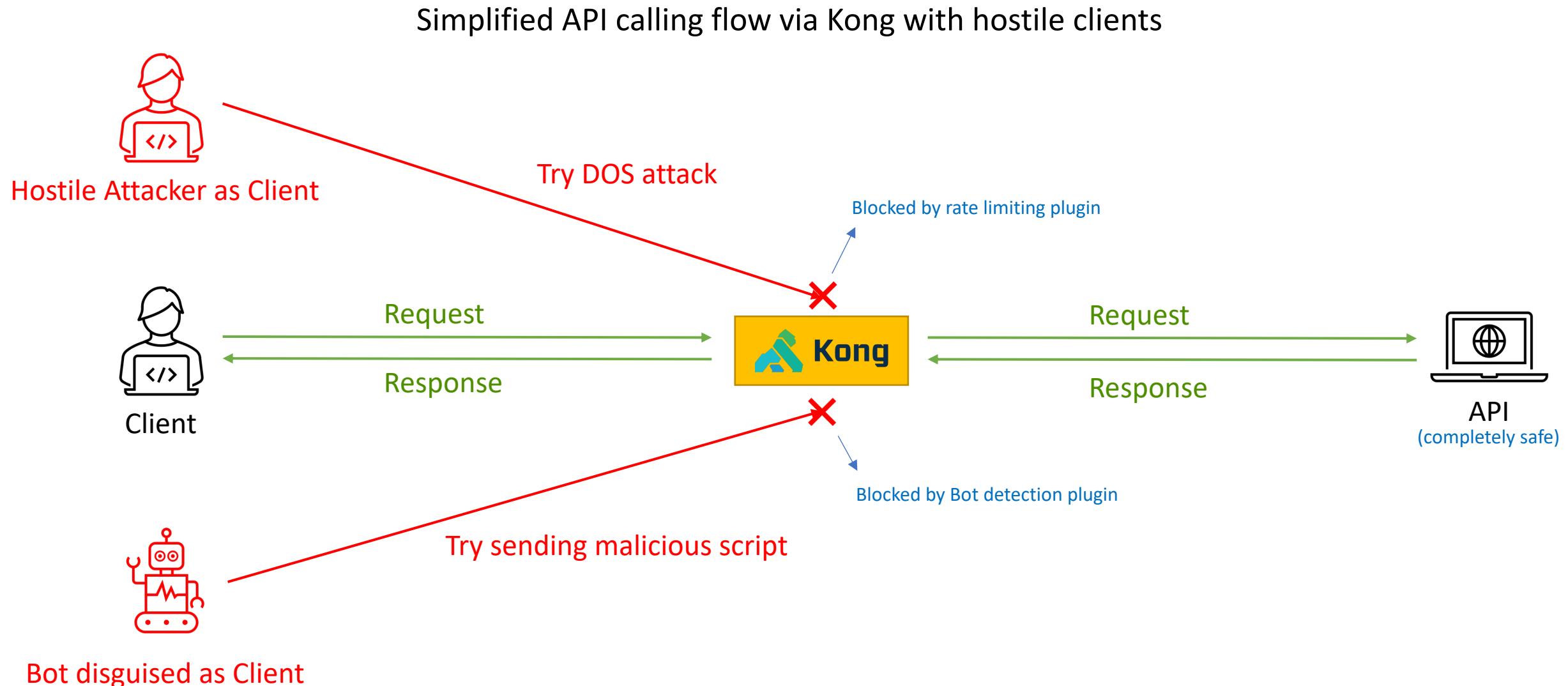
Simplified API calling flow via Kong with multiple APIs



Prologue



SIRISOFT
COMPANY LIMITED

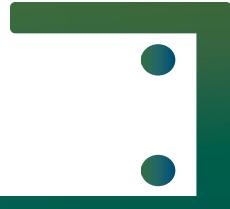


Agenda



- What is Kong? How it work? What can we do with it?
- Kong Real-life Use Case
- Konga GUI
- Installing Kong API Gateway
- Service and route creation
- Service and route testing with Postman
- Recommended plugins and installation
- Plugin testing with Postman
- HTTPS connection

What you will get?

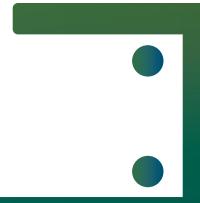


- Understand the concept of API gateway, specifically Kong
- Understand the purpose and what can be achieved with Kong
- Understand how Kong is constructed, and interaction between each entity
- Be able to install and config Kong
- Understand the concept and purpose of Konga
- Be able to manage Kong by using Konga
- Be able to utilize plugins to strengthen Kong



What is Kong?
How it works?
What can we do with it?

What is Kong?



- **Kong** คือ API gateway ที่ใช้ในการจัดการการเข้าถึงและเพิ่มความปลอดภัยให้กับการเชื่อมต่อไปยัง API โดยมีจุดเด่นคือ ความสามารถในการปรับแต่งโดยใช้ plugins ต่าง ๆ เพื่อให้ Kong สามารถรองรับการทำงานได้ตรงตามความต้องการของผู้ใช้ โดย ปัจจุบันเป็น open source ที่ถูกพัฒนาโดย Kong Inc. ในชื่อ Kong API Gateway
- API Gateway ทำหน้าที่เป็นทางผ่านสำหรับการเชื่อมต่อไปยัง API ต่าง ๆ โดยการรับ request มาจาก client และส่งต่อไปยัง API ที่ ต้องการ และจึงส่งกลับไปยัง client เมื่อได้รับ response จาก API และ ซึ่งการที่มี API มาคั่นระหว่างนั้นทำให้การเชื่อมต่อนั้น ปลอดภัยและเป็นระเบียบมากขึ้น



Why Kong?



- ▶ สิ่งที่ทำให้ Kong โดยเด่นจาก API Gateway ตัวอื่น ๆ คือความสามารถในการดึง plugin ทั้งจาก official และ third-party ที่ได้รับการ verify จาก Kong เพื่อนำมาใช้ในการเพิ่ม feature ใหม่ ๆ เพื่อให้ตรงตามความต้องการของผู้ใช้งาน เช่น ในด้าน security หรือ metric สำหรับการทำ report เป็นต้น

▶ Kong ควรถูกใช้เมื่อไร?

- เมื่อมีจำนวน API เยอะมาก
- เมื่อ Endpoint ของ API อาจมีการเปลี่ยนแปลง
- เมื่อต้องการ feature เพิ่มเติมจาก API Gateway ทั่วไป

จ่ายต่อการ monitor request

ไม่ต้องเปลี่ยนที่ client แก้ config ที่ Kong แทน

ใช้ plugins เพื่อเพิ่ม feature

▶ Kong ไม่ควรถูกใช้เมื่อไร?

- เมื่อมี API แค่ตัวเดียวหรือจำนวนน้อยโดยไม่ต้องการ feature เพิ่มเติม
- เมื่อให้ความสำคัญกับ performance มาก ๆ

ไม่มีความจำเป็นต้องใช้

การเข้ามารองรับ Kong ใช้เวลาอย่างมาก แทบไม่ effect ระบบ
แต่หากต้องการ performance จริง ๆ ก็ไม่ควรใช้ Kong



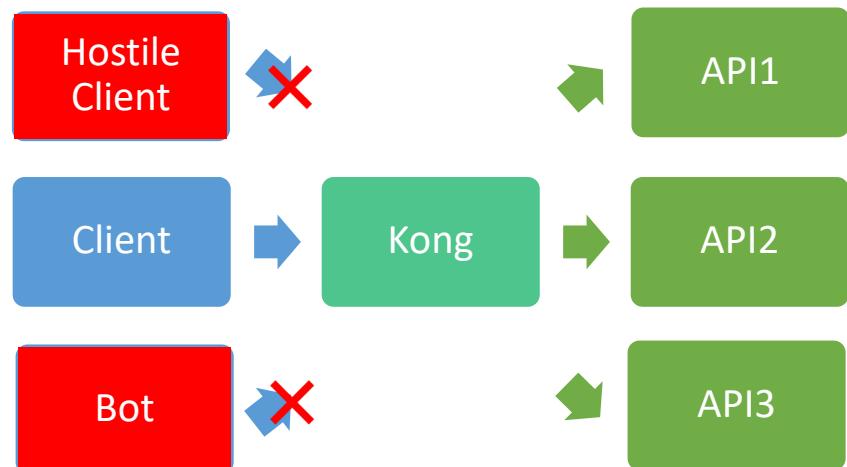
How it works?

➤ Kong จะทำหน้าที่เป็น API Gateway ที่เป็นตัวกลางระหว่าง client และ APIs โดยทุก ๆ การเชื่อมต่อไปที่ APIs จะต้องผ่านที่ Kong ก่อนเสมอ (โดยการปิดการเชื่อมต่อตรงช่องทางอื่น ๆ นอกจგาต์จาก Kong) ซึ่งการเชื่อมต่อไปที่ API ผ่าน Kong สามารถทำได้ดังนี้:

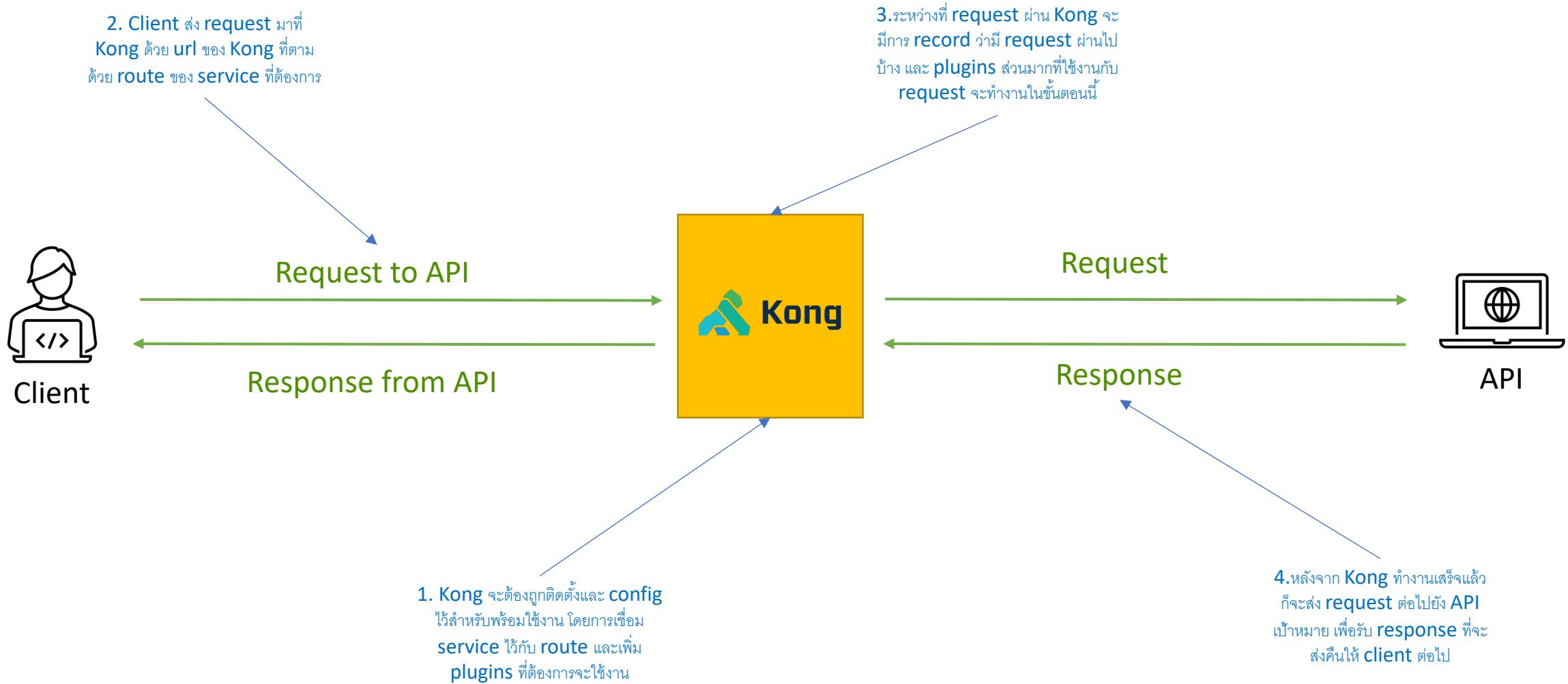
- Kong จะต้องถูกติดตั้งและ config ไว้สำหรับพร้อมใช้งาน โดยการเชื่อม service ไว้กับ route และเพิ่ม plugins ที่ต้องการจะใช้งาน
- Client ส่ง request มาที่ Kong ด้วย url ของ Kong ที่ตามด้วย route ของ service ที่ต้องการ
- ระหว่างที่ request ผ่าน Kong จะมีการ record ว่ามี request ผ่านไปบ้าง และ plugins ส่วนมากที่ใช้งานกับ request จะทำงานในขั้นตอนนี้
- หลังจาก Kong ทำงานเสร็จแล้ว ก็จะส่ง request ต่อไปยัง API เป้าหมาย เพื่อรับ response ที่จะส่งคืนให้ client ต่อไป

Keywords

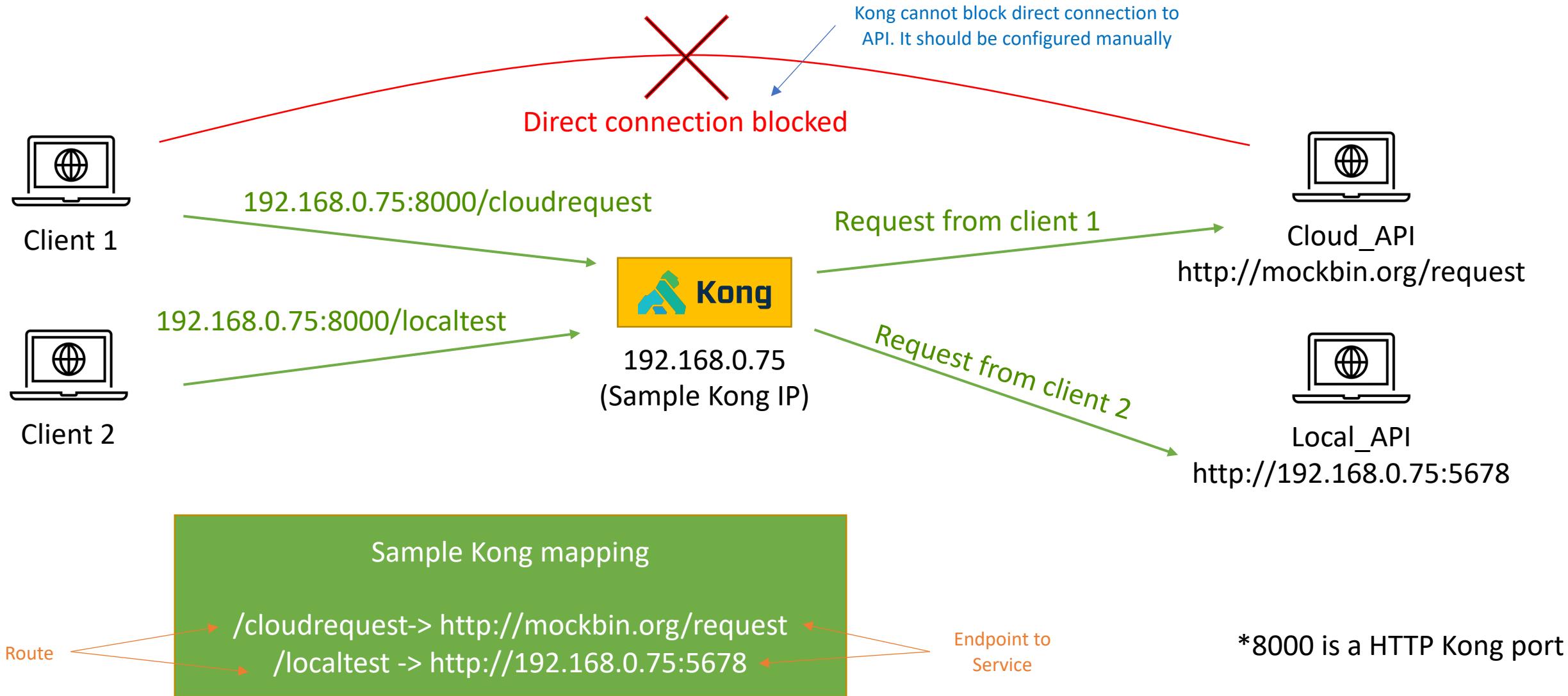
Service -> API เป้าหมายที่จะถูกเชื่อมผ่าน Kong
Route -> Path ที่ใช้ในการเรียก Service



Kong flow



Kong sample architecture





Kong Real-life
Use Case

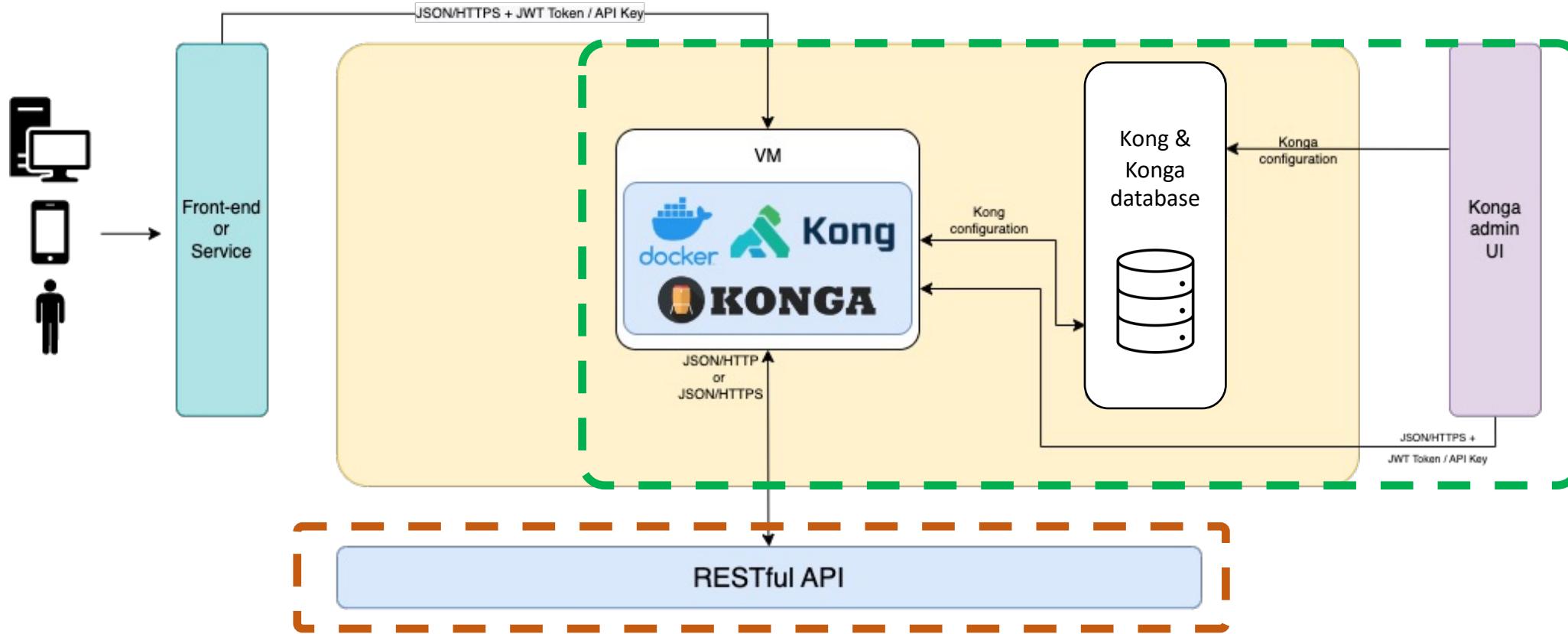
Real-life use case sample



- **Kong** ถูกนำไปใช้สำหรับโปรเจกต์ Union Auction Centralize API ซึ่งทำร่วมกับบริษัท สหการประมูล จำกัด (มหาชน) ซึ่งทางสหการประมูลมีความต้องการที่อยากรวบรวม API ทั้งหมดให้ผ่านที่ Gateway เพื่อเพิ่มความสะดวกในการเรียกใช้งาน และสามารถ monitor request ที่เข้ามาของทุก ๆ API พร้อม ๆ กันได้ รวมถึงเพิ่ม security โดยใช้ plugins เพื่อเพิ่มความปลอดภัยให้กับ API เนื่องจาก API ของสหการประมูลจะถูกต่อโดยคุณนอกอีกด้วย



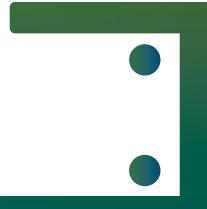
Real-life use case architecture





Konga GUI

Konga GUI



- ▶ Konga คือเครื่องมือที่ถูกสร้างขึ้นมาเพื่อควบคุมและจัดการ Kong โดยเฉพาะ โดยสามารถทำงานแทนการใช้ command line ที่ใช้ควบคุม Kong ได้เกือบทั้งหมดโดยจะทำหน้าที่เป็น user interface เพื่อเพิ่มความสะดวกในการใช้งาน Kong
- ▶ Konga มี feature ที่แยกออกจาก Kong คือการสร้าง user สำหรับใช้งาน Konga โดยเฉพาะ โดยสามารถสร้างเป็น admin ที่สามารถควบคุม Kong ได้ หรือเป็น guest ที่สามารถดู config "ได้แต่ไม่สามารถแก้ไข"ได้



KONGA

Alternative GUI



- **Kong Manager** คือเครื่องมือควบคุมและจัดการ Kong อย่างเป็นทางการซึ่งถูกสร้าง Kong Inc. ซึ่งเป็นบริษัทผู้สร้าง Kong โดย Kong Manager สามารถใช้ได้เฉพาะเมื่อใช้ Kong แบบ Enterprise เท่านั้น ทำให้ Konga ที่มี feature คล้ายกันและถูกสร้างมาจากการ third party ได้รับความนิยมมากกว่าสำหรับการใช้งานทั่วไปที่ไม่ได้ต้องการ Enterprise
- สำหรับ Training นี้ จะใช้ Konga สำหรับ GUI เป็นหลัก

The screenshot shows the Kong Manager interface. The left sidebar has a navigation menu with options: Change Workspace, Dashboard, API Gateway (Services, Routes selected), Consumers, Plugins, Upstreams, Certificates, SNI, Vitals (Status Codes), and Immunity (Alerts). The main content area is titled 'Routes' and contains a table with two rows of route definitions. The columns are: Name, Protocols, Methods, Hosts, Paths, and Id. The first row has a Name of 'kong.kong-kong-admin.00', Protocols as '["http", "https"]', Methods as '["GET"]', Hosts as '["*"]', Paths as '["/admin"]', and Id as '89b6141a...'. The second row has a Name of 'default.apip-kong.00', Protocols as '["http", "https"]', Methods as '["GET"]', Hosts as '["*"]', Paths as '["/api"]', and Id as '0e5df37...'. There are 'View', 'Edit', and 'Delete' buttons for each row.

| Name | Protocols | Methods | Hosts | Paths | Id |
|-------------------------|-------------------|---------|-------|------------|-------------|
| kong.kong-kong-admin.00 | ["http", "https"] | ["GET"] | ["*"] | ["/admin"] | 89b6141a... |
| default.apip-kong.00 | ["http", "https"] | ["GET"] | ["*"] | ["/api"] | 0e5df37... |



Installing Kong API Gateway

Lab target



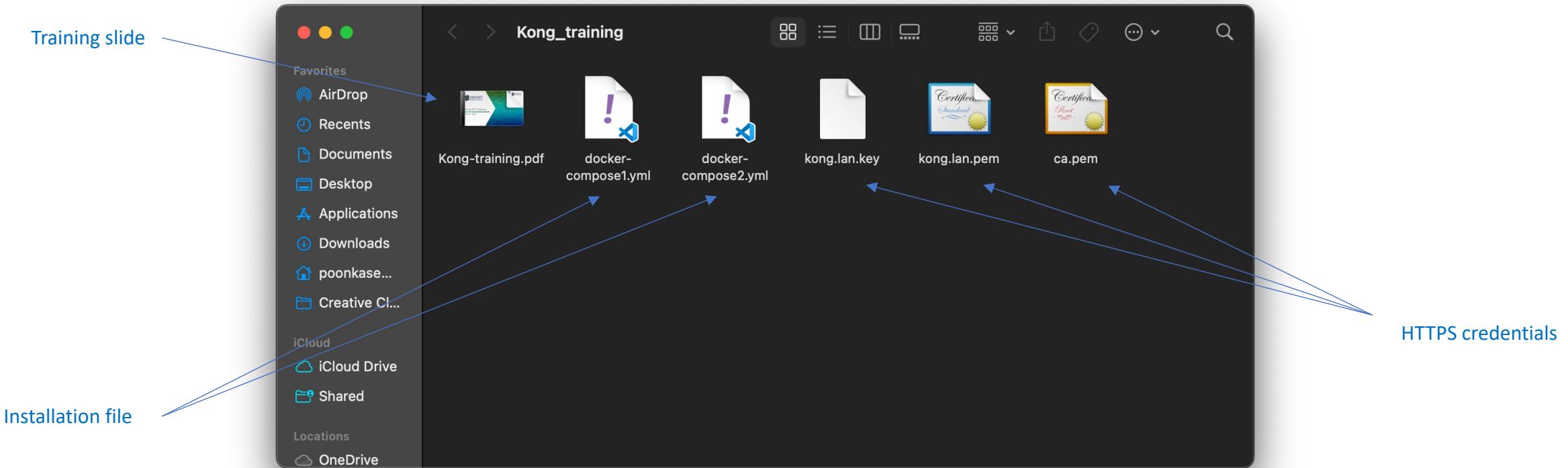
➤ To-do

- ใช้ docker-compose เพื่อติดตั้ง Kong และ Konga รวมถึง database สำหรับเก็บข้อมูลของ Kong
- Setup Konga สำหรับการใช้งาน GUI เพื่อไปต่อ กับ Kong

➤ Expected result

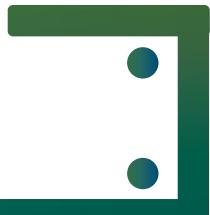
- Kong, Konga และ database รันบน docker
- Konga ที่ register user และต่อ กับ Kong พร้อมใช้งาน

Installing Kong



▶ เปิด folder Kong_training จะพบกับไฟล์ตามนี้

Installing Kong



```
Last login: Wed May 17 19:24:14 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[Poonkasem-SRS:~ poonkasemkasemsrinaayutthaya$ cd /Users/poonkasemkasemsrinaayutt]
haya/Desktop/Kong_training
Poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$ ]
```

▶ เปิด terminal (หรือ cmd) ที่ folder นี้

Installing Kong



```
Last login: Wed May 17 18:56:10 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[poonkasem-SRS:~ poonkasemkasemsrinaayutthaya$ cd /Users/poonkasemkasemsrinaayutt]
haya/Desktop/Kong_training
[poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$ docker-compose -f dock]
er-compose1.yml up -d
[+] Running 4/4
  # Network kong_training_kong-net          Created      0.1s
  # Volume "kong_training_kong-datastore"  Created      0.0s
  # Container kong_training-kong-database-1 Started      0.6s
  # Container kong_training-kong-migration-1 Started      0.9s
[poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$ docker-compose -f dock]
er-compose2.yml up -d
WARN[0000] Found orphan containers ([kong_training-kong-migration-1 kong_trainin
g-kong-database-1]) for this project. If you removed or renamed this service in
your compose file, you can run this command with the --remove-orphans flag to cl
ean it up.
[+] Running 2/2
  # Container konga                      Started      0.6s
  # Container kong_training-kong1-1       S...          0.6s
poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$
```

➤ รัน docker-compose -f docker-compose1.yml up -d และ docker-compose -f docker-compose2.yml up -d ตามลำดับเพื่อสร้าง Kong และ Konga พร้อม Database ที่เก็บข้อมูลของ Kong

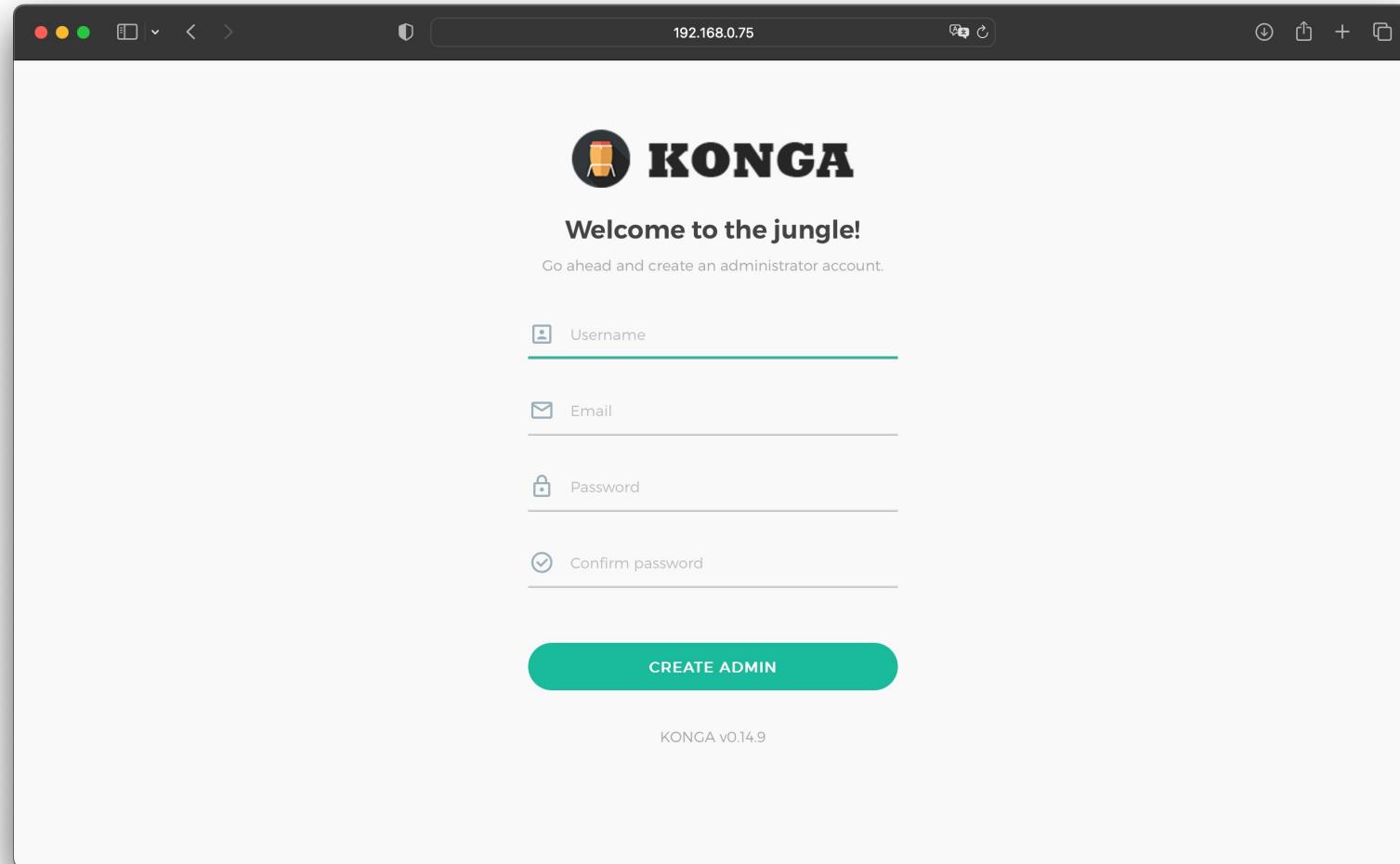
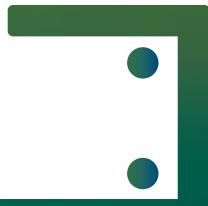
Installing Kong



```
[Poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
55c3300ee48e        kong:latest        "/docker-entrypoint..."   46 seconds ago    Up 45 seconds (healthy)
446930ac2d66        pantsel/konga     "/app/start.sh"         46 seconds ago    Up 45 seconds
e85c07c3bc76        postgres:9.6      "docker-entrypoint.s..." 53 seconds ago    Up 52 seconds (healthy)
```

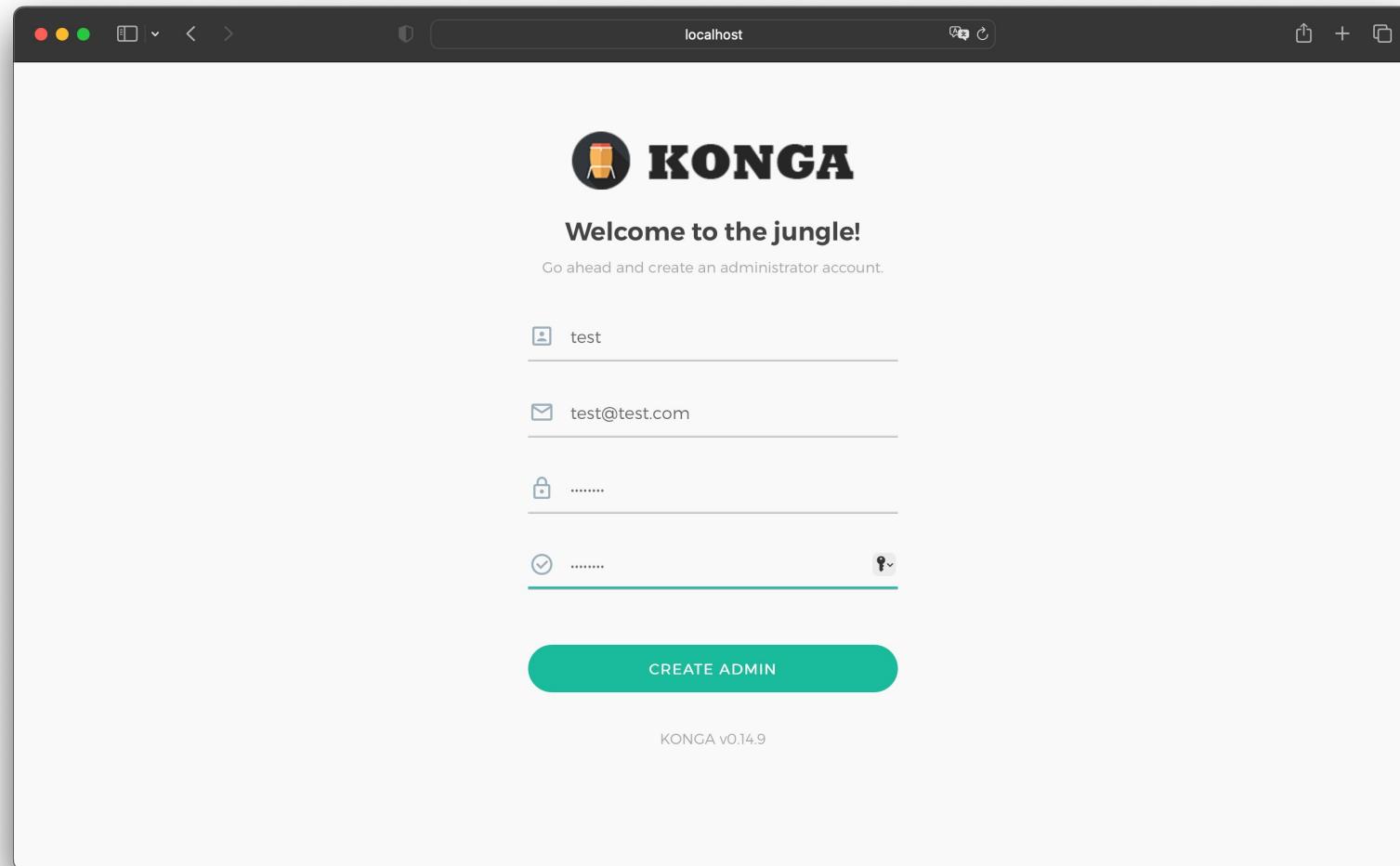
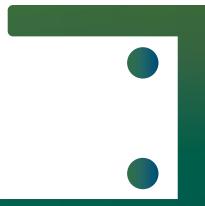
- รัน docker ps เพื่อเช็คว่า container ขึ้นมาแล้ว โดยจะมี Kong, Konga และ Postgres

Installing Kong



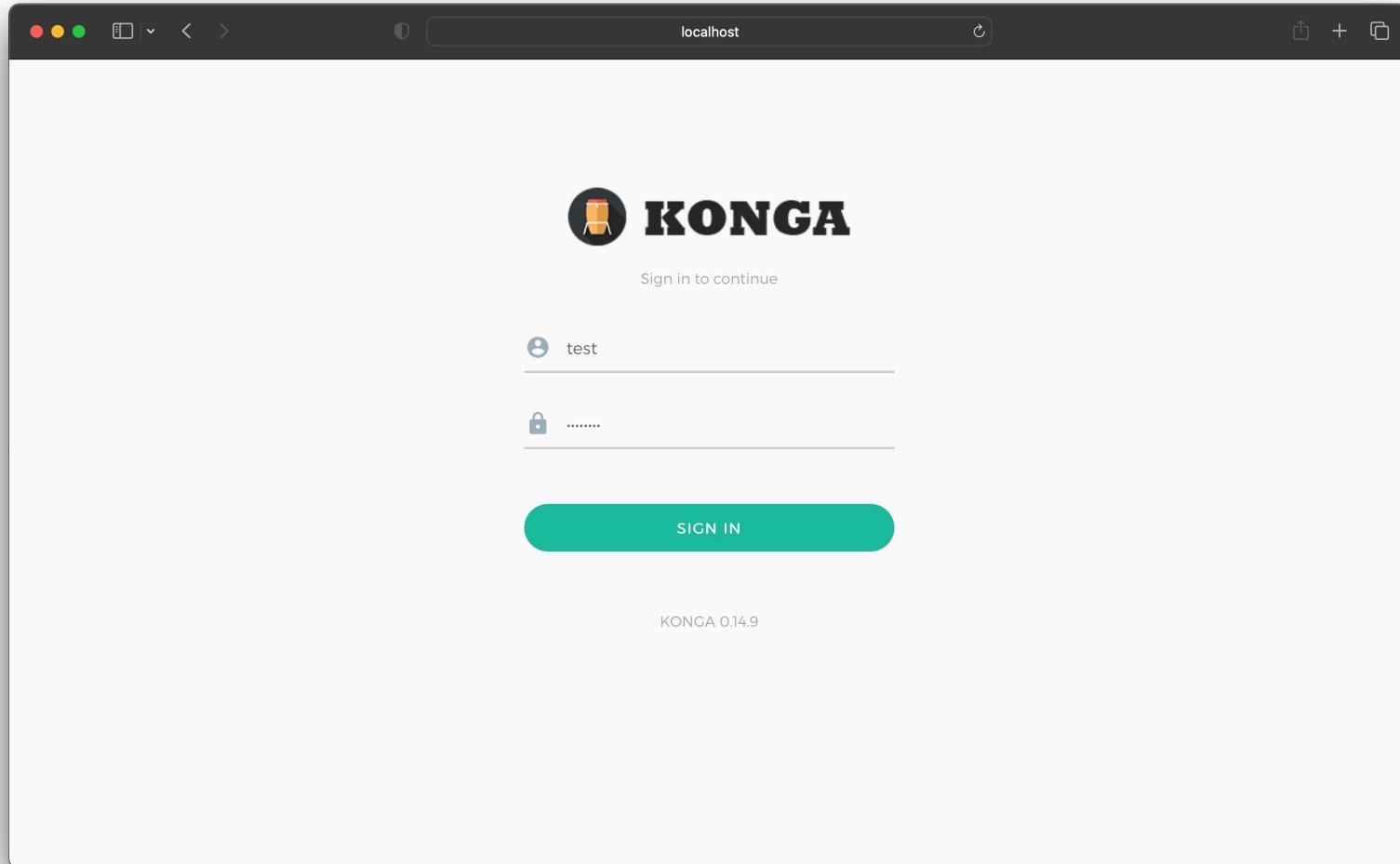
➤ เปิด <http://localhost:1337> เพื่อเปิด Konga ว่าสามารถใช้งานได้

Konga registration



➤ ใส่ข้อมูลเพื่อสร้าง admin สำหรับใช้งาน

Konga registration



➤ Login โดยใช้ account ที่เพิ่งสร้างไป

Connect Konga to Kong



The screenshot shows the Konga web application interface. On the left is a dark sidebar with the KONGA logo at the top and five menu items: DASHBOARD, APPLICATION, USERS, CONNECTIONS, and SNAPSHOT. Below these are two more items: SETTINGS and a large empty space. The main content area has a teal header bar with the text "localhost". In the center, there's a large teal circular icon with a white double-headed arrow. Below it is the word "Welcome!". Underneath "Welcome!" is the text "First of all, let's setup a connection to Kong Admin." followed by "Select a connection type.". There are four tabs: DEFAULT (which is underlined in teal), KEY AUTH, JWT AUTH, and BASIC AUTH. A callout box below the tabs contains 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." At the bottom of the form, there are fields for "Name *" (containing "kong") and "Kong Admin URL *" (containing "http://192.168.0.12:8001"). At the very bottom is a teal button with a white checkmark and the text "CREATE CONNECTION". At the very bottom of the page, there's a footer bar with links to "KONGA 0.14.9", "GitHub", "Issues", "Support the project", and a status message "Connected to N/A".

➤ ตั้งชื่อ connection ที่ต้องการและใส่ <http://<Your IP Address>:8001> ซึ่งเป็น port admin ของ Kong เพื่อเริ่มการเชื่อมต่อ



Service and
route creation

Lab target



➤ To-do

- สร้าง Cloud_API service พร้อมสร้าง Cloud_API_Route
- สร้าง Local_API service พร้อมสร้าง Local_API_Route

➤ Expected result

- ภายใน Kong มีการสร้าง Service ชื่อ Cloud_API และ Local_API
- มีการสร้าง Cloud_API_Route และ Local_API_Route เพื่อ map กับ service ตามลำดับ

Create service



The screenshot shows the KONGA interface on a Mac OS X desktop. The left sidebar has a dark blue background with white icons and text. The 'SERVICES' option is selected, highlighted with a blue bar. 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 teal button labeled '+ ADD NEW SERVICE'. To the right is a search bar with the placeholder 'search...' and a dropdown menu set to 'Results: 25'. A table header with columns 'NAME', 'HOST', 'TAGS', and 'CREATED' is visible. At the bottom of the page, there's a footer with links to 'KONGA 0.14.9', 'GitHub', 'Issues', 'Support the project', and a status message 'Connected to kong'.

➤ เลือก SERVICES ที่ด้านซ้าย จะขึ้นหน้า List ของ services ทั้งหมดที่มี พร้อมกับตัวเลือกสร้าง service ใหม่

Create service

A screenshot of the KONG API Gateway interface. On the left, a dark sidebar lists various sections: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOT, and SETTINGS. The 'SERVICES' section is currently selected. In the center, a modal window titled 'CREATE SERVICE' is open. The form fields are as follows:

- Name (optional): Cloud_API
- Description (optional): An optional service description.
- Tags (optional): Optionally add tags to the service
- Url (shorthand-attribute): http://mockbin.org/request
- Protocol (semi-optional): The protocol used to communicate with the upstream. It can be one of [http](#) or [https](#).
- Host (semi-optional): The host of the upstream server.
- Port (semi-optional): The upstream server port. Defaults to [80](#).
- Path (optional): The path to be used in requests to the upstream server. Empty by default.

The status bar at the bottom right of the modal shows 'Connected to kong'. The background of the main interface shows a search bar with 'localhost' and a results list with 25 items.

➤ กดสร้าง Service พร้อมกับใส่ข้อมูลตามนี้ เพื่อสร้าง Service ที่ชื่อ Cloud_API ที่จะต่อไปยัง API ที่ <http://mockbin.org/request>

Create service



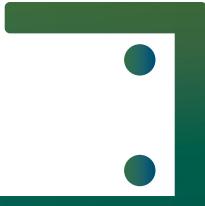
The screenshot shows the KONGA interface on a Mac OS X desktop. The left sidebar has a dark blue theme with white icons and text, listing various API gateway components: DASHBOARD, API GATEWAY, INFO, SERVICES (selected), ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. 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 teal button labeled '+ ADD NEW SERVICE'. A search bar with the placeholder 'search...' and a dropdown menu set to 'Results : 25' are on the right. A table lists one service entry:

| NAME | HOST | TAGS | CREATED |
|-----------|-------------|------|--------------|
| Cloud_API | mockbin.org | | May 20, 2023 |

Actions for this service include a red 'DELETE' button. At the bottom of the main area, there are links for 'KONGA 0.14.9', 'GitHub', 'Issues', 'Support the project', and a status message 'Connected to kong'.

➤ Cloud_API service ที่สร้างไปอยู่ใน list ของ service ทั้งหมด

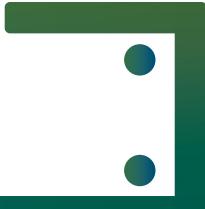
Create route



The screenshot shows the KONGA interface for managing an API service named "Service Cloud_API". The left sidebar lists various service categories: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. A red arrow points from the text "Not this" to the "INFO" button in the sidebar. Another blue arrow points from the text "This" to the "ROUTES" button, which is highlighted with a green background. The main content area displays the "Service Cloud_API" details, including "Service Details", "Routes" (which is the active tab), "Plugins", and "Eligible consumers (beta)". A search bar at the top of the routes table allows searching by "Name / ID", "Hosts", "Paths", "Protocols", "Methods", "Regex priority", and "Created". Below the search bar, a message says "no data found...". At the bottom of the page, there are links to KONGA 0.14.9, GitHub, Issues, Support the project, and a note that it is "Connected to kong".

➤ เลือกที่ชื่อ Cloud_API และเลือก Routes จะแสดงหน้า route ทั้งหมดที่ถูก map ไว้กับ service นี้

Create route



เมื่อพิมพ์ค่า Paths และ
Methods เสิร์ฟให้กด
Enter ไม่งั้นจะขึ้น error

The screenshot shows the KONGA API Gateway interface. On the left, there's a sidebar with various menu items: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. The 'ROUTES' item is highlighted. The main area is titled 'ADD ROUTE TO CLOUD_API'. It contains several input fields:

- Name (optional)**: Cloud_API_Route
- Tags (optional)**: (empty)
- Hosts (semi-optional)**: (empty)
- Paths (semi-optional)**: /cloudrequest
- Headers (semi-optional)**: (empty)
- Path handling**: v1

At the bottom right of the modal, there's a 'Created' message and a 'Connected to kong' status indicator.

➤ สร้าง route ที่ชื่อ Cloud_API_Route พร้อมกับตั้ง Paths สำหรับเข้ามายังต่อเป็น /cloudrequest และใช้ Methods เป็น GET

Create route



The screenshot shows the KONGA API Gateway interface. On the left, a sidebar menu lists various sections: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. The 'SERVICES' section is currently selected. The main content area displays a service named 'Service Cloud_API'. Below it, there are tabs for 'Service Details' (disabled), 'Routes' (selected and highlighted in green), 'Plugins' (disabled), and 'Eligible consumers' (beta). A search bar labeled 'search routes...' is present. A table titled 'Routes' lists one entry: 'Cloud_API_Route'. The table columns include Name / ID, Hosts, Paths, Protocols, Methods, Regex priority, and Created. The entry details are: Name / ID: Cloud_API_Route, Hosts: -, Paths: /cloudrequest, Protocols: http, https, Methods: GET, Regex priority: 0, Created: May 20, 2023. A green button labeled '+ ADD ROUTE' is located at the top right of the routes table. At the bottom of the page, there are links for KONGA 0.14.9, GitHub, Issues, Support the project, and a note indicating it is Connected to kong.

| Name / ID | Hosts | Paths | Protocols | Methods | Regex priority | Created |
|-----------------|-------|---------------|-------------|---------|----------------|--------------|
| Cloud_API_Route | - | /cloudrequest | http, https | GET | 0 | May 20, 2023 |

➤ Cloud_API_Route ที่สร้างไปอยู่ใน list ของ route ของ Cloud_API service

Create another service and route



ทดลองสร้าง service และ route โดยต่อไปที่ API ที่รันเป็น container ในเครื่องของตัวเอง

1. รัน docker run -d -p 5678:5678 hashicorp/http-echo -text success เพื่อสร้าง API container ด้วยย่างที่จะ return ว่า success เมื่อยิงไปถึง
2. สร้าง service ชื่อ Local_API พร้อมกับ Url ว่า http://<Your IP Address>:5678
3. สร้าง route สำหรับ service Local_API ที่ชื่อ Local_API_Route ด้วย Paths ว่า /localtest และ Method ใช้ GET

A screenshot of the KONGA API gateway dashboard. The left sidebar shows navigation options like DASHBOARD, API GATEWAY, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOT, and SETTINGS. The main content area is titled 'Services' and displays two entries: 'Local_API' with host 192.168.0.12 and 'Cloud_API' with host mockbin.org. A search bar at the top right shows 'Hello, test'. At the bottom, it says 'KONGA 0.14.9 GitHub Issues Support the project' and 'Connected to kong'.

A screenshot of the KONGA API gateway dashboard showing the details for the 'Service Local_API'. The left sidebar is identical to the previous screenshot. The main content area is titled 'Service Local_API' and shows 'Service Details' with a note about 'Eligible consumers beta'. It lists one route named 'Local_API_Route' with path '/localtest', protocol 'http.https', method 'GET', and priority '0'. At the bottom, it says 'KONGA 0.14.9 GitHub Issues Support the project' and 'Connected to kong'.



Service and route
testing with Postman

Lab target



➤ To-do

- ทดสอบการใช้งานของ Service และ Route ที่สร้างขึ้นด้วย Postman

➤ Expected result

- ทุก test case ที่ทดสอบต่อไปนี้ เป็นไปตาม expected result

Postman testing



The screenshot shows the Postman interface in 'Scratch Pad' mode. A modal dialog is open in the center, stating 'Working locally in Scratch Pad. Switch to a Workspace'. Below this, a request card is displayed for 'http://192.168.0.12:8000' with a 'GET' method. The 'Headers' tab is selected, showing '(6)' entries. The 'Body' tab is also visible. The response pane at the bottom shows a JSON object with a single key 'message': "no Route matched with those values".

GET http://192.168.0.12:8000

http://192.168.0.12:8000

GET http://192.168.0.12:8000

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

1 "message": "no Route matched with those values"

2

3

404 Not Found 11 ms 250 B Save Response

Find and Replace Console Runner Trash

Error นี้จะขึ้นเมื่อคุณไปที่ Kong และ route
ที่ส่งไปไม่เหมือนกับ route ที่เคยตั้งไว้

➤ เปิด Postman พร้อมกับยิง GET ไปที่ `http://<Your IP Address>:8000`

Postman testing



The screenshot shows the Postman interface in the Scratch Pad workspace. A blue arrow points from the text "เชื่อมต่อผ่าน Kong" to the "APIs" section in the sidebar. Another blue arrow points from the text "ผลลัพธ์เมื่ອนกับยังไงบ้าง" and "API โดยตรง" to the main request area. The request URL is `http://192.168.0.12:8000/clouddrequest`. The response body is displayed as JSON:

```
1  "startedDateTime": "2023-05-20T07:53:57.062Z",
2  "clientIPAddress": "172.23.0.1",
3  "method": "GET",
4  "url": "http://192.168.0.12/request",
5  "httpVersion": "HTTP/1.1",
6  "cookies": {},
7  "headers": {
8      "host": "mockbin.org",
9  }
```

➤ เมื่อเพิ่ม /clouddrequest ชึ่งเป็น path ที่ใช้ยิงไปที่ Cloud_API service ทำให้การเชื่อมต่อสำเร็จ

Postman testing



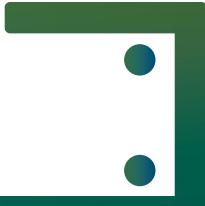
A screenshot of the Postman application interface. The left sidebar shows 'Scratch Pad' selected, along with other options like 'Collections', 'APIs', 'Environments', 'Mock Servers', 'Monitors', and 'History'. A red callout box highlights the 'APIs' section. The main workspace shows a 'Working locally in Scratch Pad. Switch to a Workspace' message. A request card for 'http://192.168.0.12:8000/localtest' is displayed, with the method set to 'GET'. The 'Headers' tab is selected, showing '(6)' entries. The 'Body' tab is also visible. The response section shows a status of '200 OK' with a response time of '11 ms' and a size of '263 B'. The response body contains the text '1 success'. Other tabs like 'Cookies', 'Test Results', and 'Pretty' are also visible.

➤ ทำแบบเดียวกันกับ /localtest เพื่อต่อไปยัง Local_API service



Recommended
plugins and installation

Plugins



- ▶ **Plugins** คือการใส่ความสามารถเพิ่มเติมลงไปใน Kong เพื่อให้ตรงตามความต้องการ โดย Kong มี plugin ให้เลือกจำนวนมาก โดยเลือกมาทั้งหมด 4 ตัวที่มีการใช้งานเยอะที่สุด เพื่อเป็นตัวอย่างในการใช้งาน

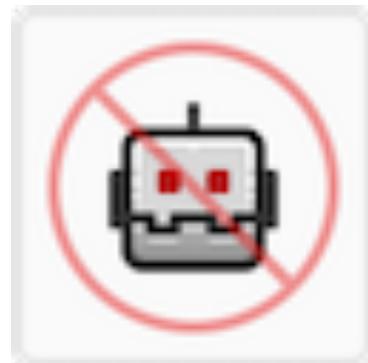
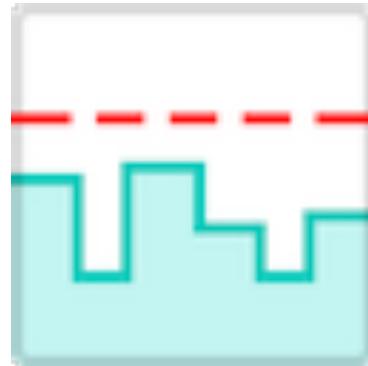
Plugins list

Key Auth -> สร้าง API key เพื่อ authenticate ผู้ใช้

Bot Detection -> ป้องกัน request ที่มี pattern ของ Bot

Rate Limiting -> limit จำนวน request ที่ client คนหนึ่งจะต่อกมาได้ต่อ 1 ช่วงเวลา

Request Size Limiting -> limit ขนาดของ request ที่เข้ามา



Lab target



➤ To-do

- ติดตั้ง Rate Limiting, Bot Detection และ Request Size Limiting สำหรับ global plugins
- ติดตั้ง Key Auth สำหรับ Cloud_API service พร้อมสร้าง consumer สำหรับใช้งานกับ plugin

➤ Expected result

- Plugin ทั้ง 4 ตัว ถูกติดตั้งและ config ไว้สำหรับพร้อมใช้งาน

Adding global plugins



The screenshot shows the Konga UI for managing an API gateway. The left sidebar has a dark blue background with white icons and text, listing various sections: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS (which is currently selected), UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area has a light gray background. At the top, it says "localhost" in the address bar, and there are user profile and notification icons. Below that, a header says "Plugins". A sub-header explains what a plugin is: "A Plugin entity represents a plugin configuration that will be executed during the HTTP request/response workflow, and it's how you can add functionalities to APIs that run behind Kong, like Authentication or Rate Limiting for example." A teal button labeled "+ ADD GLOBAL PLUGINS" is at the top left of the list area. To its right is a search bar with the placeholder "search..." and a dropdown menu showing "Results : 25". The main list area has columns: NAME, SCOPE, APPLY TO, CONSUMER, and CREATED (with an upward arrow). A message "no data found..." is centered in this area. At the bottom of the main content, there are links for "KONGA 0.14.9", "GitHub", "Issues", "Support the project", and a status indicator "Connected to kong".

➤ เลือก PLUGINS ที่แทบด้านซ้ายและเพิ่ม Plugins

Adding global plugins



The screenshot shows the KONGA interface on a Mac OS X desktop. The left sidebar has a dark blue background with white icons and text, listing various API management sections: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS (which is highlighted in teal), UPSTREAMS, and CERTIFICATES. Below these are sections for APPLICATION: USERS, CONNECTIONS, SNAPSHOTs, and SETTINGS. The main content area has a light gray header with the title 'Add Global Plugins' and a breadcrumb 'plugins / add'. A teal banner at the top of the content area contains the text 'Plugins added in this section will be applied Globally.' followed by two bullet points: '- If you need to add plugins to a specific Service or Route, you can do it in the respective section.' and '- If you need to add plugins to a specific Consumer, you can do it in the respective Consumer's page.' Below the banner is a navigation bar with tabs: Authentication (which is teal and bolded), Security, Traffic Control, Serverless, Analytics & Monitoring, Transformations, Logging, and Other. The 'Authentication' tab is active, showing a sub-section titled 'Authentication' with the sub-instruction 'Protect your services with an authentication layer'. There are four cards in this section: 'Basic Auth' (with a user icon and a box icon), 'Key Auth' (with a key icon), 'Oauth2' (with a circular icon containing a '2'), and 'Hmac Auth' (with a hash icon). At the bottom of the main content area, there is a footer with links: 'KONGA 0.14.9', 'GitHub', 'Issues', 'Support the project', and 'Connected to kong'.

➤ จะมี list ของ plugins ทั้งหมดที่แบ่งหมวดหมู่ไว้ออกมาให้เลือก

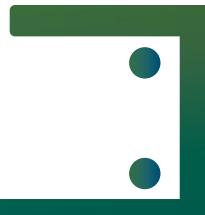
Adding global plugins



The screenshot shows the KONGA API Gateway dashboard. On the left, the sidebar includes options like DASHBOARD, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS (which is currently selected), UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHTOS, and SETTINGS. The main content area has a message bar at the top stating: "- If you need to add plugins to a specific Service or Route, you can do it in the respective section.
- If you need to add plugins to a specific Consumer, you can do it in the respective Consumer's page." Below this is a modal window titled "ADD BOT DETECTION". The modal describes the plugin's purpose: "Protects your API from most common bots and has the capability of whitelisting and blacklisting custom clients." It has sections for "consumer" (with a note about targeting a specific consumer ID) and "allow" and "deny" rules (both with a tip to press Enter to accept a value). At the bottom of the modal is a green "✓ ADD PLUGIN" button. To the right of the modal, there is a preview card for the "Bot Detection" plugin, showing its icon (a red circle with a white bot head), a brief description ("blocks bots or custom clients"), and a "ADD PLUGIN" button.

➤ ไปที่หมวด Security และเลือก Bot Detection พร้อมกับกดเพิ่ม Bot Detection plugin โดยไม่ต้องใส่ค่าเพิ่มเดิม

Adding global plugins



จำนวน 5 request
ต่อนาที ใช้เพื่อการ
ทดสอบเท่านั้น

The screenshot shows the Kong API Gateway dashboard with the 'PLUGINS' section selected. A modal window titled 'ADD RATE LIMITING' is open, allowing configuration of rate limits across different time intervals. The 'consumer' section is highlighted, showing a text input for the consumer ID. Below it, the 'second' interval is selected, with a value of '5' entered. Other intervals (minute, hour, day, month) also have their respective limit inputs. To the right of the main window, a smaller window titled 'Request Terminating' is visible, showing a note about terminating requests with a 429 status code if the limit is reached.

สามารถปรับจำนวนได้
ตามต้องการ และทุก
หน่วยเวลาสามารถใช้
งานร่วมกันได้

- ▶ ไปที่หมวด Traffic Control และเลือก Rate Limiting พร้อมกับใส่ไว้ client คนนึงสามารถยิงได้ 5 ครั้งต่อนาที (ต่อ service) และเพิ่ม plugin

Adding global plugins



ตั้งเป็น 0 bytes เพื่อการทดสอบ

KONGA

Add Global Plugins

plugins / add

ADD REQUEST SIZE LIMITING

Block incoming requests whose body is greater than a specific size in megabytes.

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.

allowed payload size

0

Allowed request payload size in megabytes, default is 128 (128000000 Bytes)

size unit

bytes

require content length

NO

✓ ADD PLUGIN

DASHBOARD

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS

SETTINGS

ADD PLUGIN

ADD PLUGIN

ADD PLUGIN

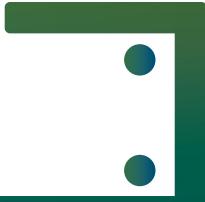
ADD PLUGIN

KONGA 0.14.9 GitHub Issues Support the project

Connected to kong

➤ เลือก Request Size Limiting พร้อมกับใช้ 0 bytes และเพิ่ม plugin

Adding service plugins



สามารถตั้งให้เป็นเฉพาะ route
หนึ่งเท่านั้นได้เมื่อกัน โดยเข้าไป
ที่ Routes และเลือก Plugins

KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS

SETTINGS

localhost

Hello, test

Service Cloud_API

services / show

Service Details

Routes

Plugins

Eligible consumers

beta

Assigned plugins

+ ADD PLUGIN

search plugins...

| Name | Consumer | Created |
|------------------|----------|---------|
| no data found... | | |

KONGA 0.14.9 GitHub Issues Support the project Connected to kong

➤ เพื่อเพิ่ม plugin สำหรับ service หนึ่งเท่านั้น เข้าไปใน Cloud_API service และเลือก plugin และเพิ่ม plugin

Adding service plugins



A screenshot of the Kong API Gateway UI. The left sidebar shows navigation options like DASHBOARD, API GATEWAY (INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES), APPLICATION (USERS, CONNECTIONS, SNAPSHOT, SETTINGS). The main area is titled 'ADD KEY AUTH'. It contains a sub-header: '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.' Below this is a 'consumer' section with a description: '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.' A 'key names' input field contains 'api-key' with a delete button. A tip below it says: 'Tip: Press Enter to accept a value.' Another description follows: '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.' Under 'hide credentials', there's a 'NO' button with a description: '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.' In the 'anonymous' section, there's a 'key in header' toggle switch set to 'YES'. A status bar at the bottom right says 'Connected to kong'.

➤ ตั้งชื่อ Key ว่า api-key และสร้าง plugin

Plugins list



The screenshot shows the Kong API Gateway management interface. On the left, a dark sidebar menu lists various components: DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS (which is currently selected), UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. The main content area is titled "Plugins". It contains a brief description: "A Plugin entity represents a plugin configuration that will be executed during the HTTP request/response workflow, and it's how you can add functionalities to APIs that run behind Kong, like Authentication or Rate Limiting for example." Below this is a table with the following data:

| | NAME | SCOPE | APPLY TO | CONSUMER | CREATED | |
|--|-----------------------|----------|--------------------------------------|---------------|--------------|--|
| | key-auth | services | 7cb62457-14ec-4a26-a1e4-e30eee0f6776 | All consumers | May 20, 2023 | |
| | request-size-limiting | global | All Entrypoints | All consumers | May 20, 2023 | |
| | bot-detection | global | All Entrypoints | All consumers | May 20, 2023 | |
| | rate-limiting | global | All Entrypoints | All consumers | May 20, 2023 | |

At the bottom of the main content area, there are links for KONGA 0.14.9, GitHub, Issues, Support the project, and Connected to kong.

➤ เมื่อเลือก PLUGINS ที่ด้านข้ายอกรัง จะเห็นว่ามี plugins ที่สร้างไว้แสดงขึ้นมาแล้ว

Key Auth plugin setup

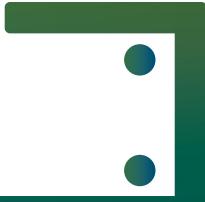


สร้าง Consumer เพื่อ
ใช้สำหรับเก็บ api-key

The screenshot shows the Kong API Gateway dashboard at `localhost`. On the left sidebar, under the 'CONSUMERS' section, there is a blue arrow pointing from the text 'สร้าง Consumer เพื่อใช้สำหรับเก็บ api-key' to the '+ CREATE' button. The main content area displays a 'CREATE CONSUMER' dialog box. It has two input fields: 'username (semi-optional)' with the value 'test' and 'custom_id (semi-optional)'. Below these fields is a 'Tags (optional)' section with a placeholder 'Optional tags for this consumer'. At the bottom right of the dialog is a green '✓ SUBMIT CONSUMER' button.

➤ เลือก CONSUMERS ที่ด้านข้ายพร้อมกับกดสร้าง consumer ด้วยชื่อ test

Key Auth plugin setup



KONGA

DASHBOARD

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS

SETTINGS

localhost

Hello, test

CONSUMER: test

consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

BASIC

API KEYS

HMAC

OAUTH2

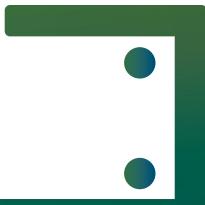
JWT

+ CREATE API KEY

KONGA 0.14.9 GitHub Issues Support the project Connected to kong

➤ เลือก Credentials พร้อมกับเลือกไปที่ API KEYS

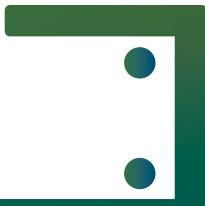
Key Auth plugin setup



The screenshot shows the Konga API Gateway interface. On the left, a sidebar menu includes options like DASHBOARD, API GATEWAY, INFO, SERVICES, ROUTES, CONSUMERS, PLUGINS, UPSTREAMS, CERTIFICATES, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. The main content area is titled "CONSUMER: test" and shows a sub-menu for "consumers / edit consumer". A modal window titled "CREATE API KEY" is open, prompting the user to "Create Api Key for test". The "API KEYS" tab is selected under the "BASIC" section. An input field labeled "key (optional)" contains the value "testkey". Below the input field, a note states: "You can optionally set your own unique key to authenticate the client. If missing, Kong will generate one." At the bottom of the modal is a green "SUBMIT" button. In the top right corner of the main interface, there is a user profile placeholder with the text "Hello, test".

➤ ใส่ key ว่า testkey พร้อมกับสร้าง API key

Key Auth plugin setup



KONGA

localhost

Hello, test

CONSUMER: test

consumers / edit consumer

Details Groups Credentials Accessible Routes Plugins

BASIC API KEYS + CREATE API KEY

| # | key | created |
|----|---------|--------------|
| 1. | testkey | May 20, 2023 |

HMAC OAUTH2 JWT

KONGA 0.14.9 GitHub Issues Support the project Connected to kong

➤ API key จะถูกสร้างขึ้น สำหรับพร้อมไปใช้งานกับ Key Auth plugin



Plugin testing
with Postman

Lab target



➤ To-do

- ทดสอบการใช้งานของ plugins ที่สร้างขึ้นด้วย Postman

➤ Expected result

- ทุก test case ที่ทดสอบต่อไปนี้ เป็นไปตาม expected result

Plugins testing



The screenshot shows the Postman application interface. On the left, there's a sidebar with options like Home, Workspaces, Explore, Scratch Pad, Collections, APIs (which is selected), Environments, Mock Servers, Monitors, and History. A prominent message in the center says "Switch to a workspace to use this feature. This is an online feature and is only available in workspaces." Below this is a red "Switch to Workspaces" button. The main area displays a request to "http://192.168.0.12:8000/clouddrequest" with a GET method. The Headers tab shows 7 items. The Body tab is selected, showing the raw JSON response:

```
1 "message": "No API key found in request"
```

. The response section shows a 401 Unauthorized status with 4 ms and 286 B. At the bottom, there are buttons for Find and Replace, Console, Runner, Trash, and Help.

➤ ทดสอบยิ่ง /clouddrequest อีกครั้งจะพบว่าไม่สามารถต่อถึงได้ เนื่องจากติดเรื่อง API Key

Plugins testing



The screenshot shows the Postman application interface. On the left, there's a sidebar with options like Home, Workspaces, Explore, Scratch Pad, Collections, APIs (selected), Environments, Mock Servers, Monitors, and History. A prominent message in the center says "Switch to a workspace to use this feature". Below it, another message states "This is an online feature and is only available in workspaces." A red button labeled "Switch to Workspaces" is visible. The main area displays a request configuration for "http://192.168.0.12:8000/clouдрequest". The "Headers" tab is selected, showing the following configuration:

| Key | Value | Description |
|-----------------|-----------------------------------|-------------|
| Postman-Token | <calculated when request is sent> | |
| Host | <calculated when request is sent> | |
| User-Agent | PostmanRuntime/7.31.1 | |
| Accept | /* | |
| Accept-Encoding | gzip, deflate, br | |
| Connection | keep-alive | |
| api-key | testkey | |

The "Body" tab is selected, showing a JSON response:

```
1  "startedDateTime": "2023-05-20T09:04:44.114Z",
2  "clientIPAddress": "172.23.0.1",
3  "method": "GET",
4  "url": "http://192.168.0.12/request",
5  "httpVersion": "HTTP/1.1",
6  "cookies": {},
7  "headers": {
8    "host": "mockbin.org",
9  }
```

➤ ไปที่แท็บ Headers พร้อมกับเพิ่ม api-key ด้วยค่า testkey และยัง จะสามารถยิงได้แล้วเนื่องจาก API Key ถูกส่งไปพร้อม request

Plugins testing

A screenshot of the Postman application interface. On the left, there's a sidebar with icons for Collections, APIs (which is selected), Environments, Mock Servers, Monitors, and History. A red box highlights the 'APIs' icon. The main area shows a 'Scratch Pad' tab with a 'New' and 'Import' button. A yellow banner at the top says 'Working locally in Scratch Pad. Switch to a Workspace'. Below it, a request card for 'http://192.168.0.12:8000/localtest' is shown with a 'GET' method and the URL. The 'Headers' tab is selected, displaying the following headers:

| KEY | VALUE | DESCRIPTION | Bulk Edit | Presets |
|-----------------|-----------------------------------|-------------|-----------|---------|
| Postman-Token | <calculated when request is sent> | | | |
| Host | <calculated when request is sent> | | | |
| User-Agent | PostmanRuntime/7.31.1 | | | |
| Accept | */* | | | |
| Accept-Encoding | gzip, deflate, br | | | |
| Connection | keep-alive | | | |

The 'Body' tab shows a simple JSON response:

```
1 success
```

▶ เมื่อ слับไปยัง /localtest จะพบว่าไม่ต้องใช้ API Key ในการยิง เนื่องจาก Key Auth plugin ถูกติดตั้งไว้ให้แล้วกับ Cloud_API service เท่านั้น

Plugins testing



The screenshot shows the Postman application interface. On the left, the sidebar has tabs for Collections, APIs (which is selected), Environments, Mock Servers, Monitors, and History. A yellow banner at the top says "Working locally in Scratch Pad. Switch to a Workspace". In the main area, there's a request card for "http://192.168.0.12:8000/localtest" with a status of "429 Too Many Requests". The "Body" tab is selected, showing the response JSON: "1", "2", "3", "message": "API rate limit exceeded". Other tabs include Body, Cookies, Headers (12), and Test Results.

- ▶ ทดสอบยิ่ง /localtest หรือ /cloudrequest (ที่มี api-key) 6 ครั้งติดกัน ในครั้งที่ 6 จะ error เนื่องจาก Rate Limiting plugin ตั้งไว้ว่าให้ยิ่งได้มากที่สุด 5 request ต่อนาที

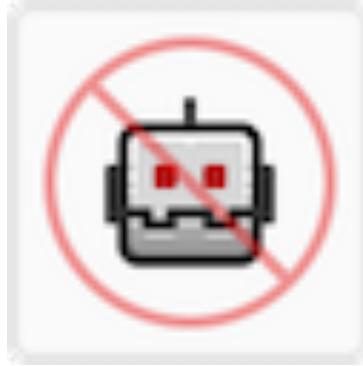
Plugins testing



The screenshot shows the Postman application interface. On the left, there's a sidebar with options like Collections, APIs (which is selected), Environments, Mock Servers, Monitors, and History. The main area displays a request configuration for a GET method to the URL `http://192.168.0.12:8000/clouddrequest`. The 'Body' tab is active, showing the value `a`. Below the request, the response pane shows an error: `413 Request Entity Too Large` with the message `"message": "Request size limit exceeded"`.

- ▶ ทดสอบยิง `/localtest` หรือ `/clouddrequest` (ที่มี api-key) แต่เพิ่ม body ใน request เข้าไปจะทำให้ error ว่า size เกินจาก Request Size Limiting plugin ที่ตั้งไว้ที่ 0 bytes

Plugins testing



- ▶ Bot Detection ของ Kong สามารถไปเช็ค list ของ pattern ที่จะ block ได้ตามนี้ (<https://github.com/Kong/kong/blob/master/kong/plugins/bot-detection/rules.lua>)

โดยจะทำงานอัตโนมัติเมื่อมี pattern เดียวกับที่มีเข้ามา



HTTPS
connection

Lab target



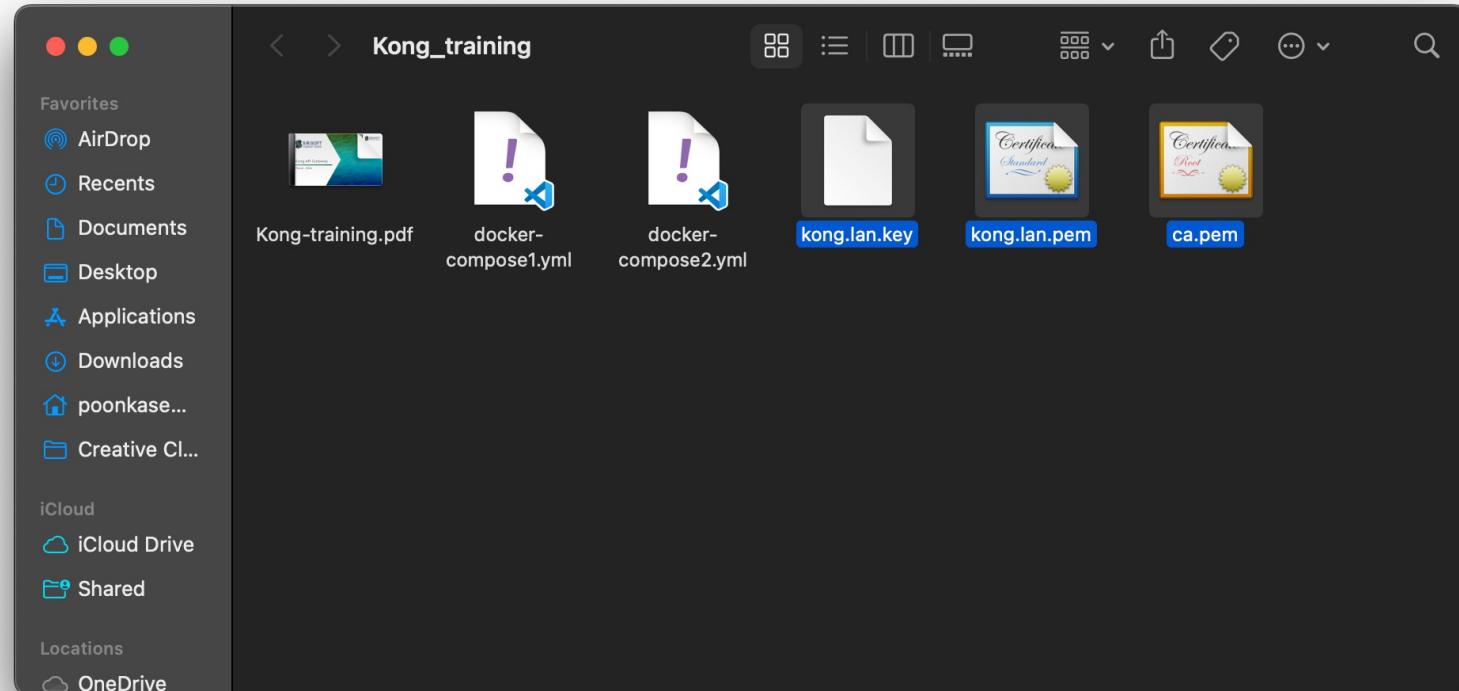
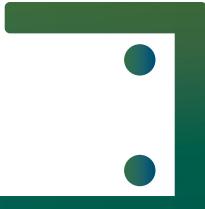
➤ To-do

- Config การเชื่อมต่อแบบ HTTPS กับ Kong
- ทดสอบการเชื่อมต่อแบบ HTTPS

➤ Expected result

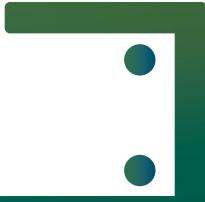
- Kong สามารถเชื่อมต่อแบบ HTTPS ได้

HTTPS setup



➤ Kong สามารถใช้การเชื่อมต่อแบบ HTTPS ได้โดยการเพิ่ม certificate เข้าไป

HTTPS setup



The screenshot shows the Kong API Gateway dashboard at `localhost`. The left sidebar menu is visible with various options like Dashboard, API Gateway, Info, Services, Routes, Consumers, Plugins, Upstreams, Certificates (which is currently selected), Applications, Users, Connections, Snapshots, and Settings. The main content area is titled "Certificates". It contains a brief description: "A certificate object represents a public certificate/private key pair for an SSL certificate. These objects are used by Kong to handle SSL/TLS termination for encrypted requests. Certificates are optionally associated with SNI objects to tie a cert/key pair to one or more hostnames." Below this is a button labeled "+ ADD CERTIFICATE". A search bar and a results count of "Results : 25" are also present. A message "No data found..." is displayed below the table header. The table has columns for ID, TAGS, SNIS, and CREATED. At the bottom of the page, there are links to KONGA 0.14.9, GitHub, Issues, Support the project, and a note "Connected to kong".

➤ “ไปที่ CERTIFICATES ที่แถบด้านซ้าย และกดเพิ่ม certificate

HTTPS setup



ชื่อของ domain ที่จะใช้

ทดสอบ

KONGA

DASHBOARD

API GATEWAY

INFO

SERVICES

ROUTES

CONSUMERS

PLUGINS

UPSTREAMS

CERTIFICATES

APPLICATION

USERS

CONNECTIONS

SNAPSHOTS

SETTINGS

KONGA 0.14.9

ADD CERTIFICATE

Certificate

```
SLM3WxqOSUj0GcSoQmnii26EljVHHW52sXmHOsvtjjBEXCjtX6T3ueBEIxFpW0S
/x/CIHUUUBSMG+nCl1FUJvw8vSNTwdczrBERBbc7pD9OALA5xDT2/IHoRb/h9
KPqXipevd4QclBuGivYVUdCjTeYqM+JrEt6Dvx/s5j4g01Xgd3MUj4HfPb4A9
/rhqxa0ApuyEFFwoTidCn9Az0aJ2VKQLOBuX4ROS6BhjyjnfwiFJCD3bUrh/k
Byt75U7We0Zot25Rx+ICNlW3HLvk8lTD1yvt1QjIKlSy6Gtcwjl8RUxzugI
lgRAOh7qmBOIkJoo6DMovOj6eVI7n4Y2eBnp20J+mRIgmMqAgf5yOIN2a77EiDb
X9d6INRxCRyBfCq10tY5qmpjqL5fsWf+CCWMUXiGBvkJ+Asfg5koI7eWMW2CB+
suyH+tEXCOWbJemVlckRfABk0o9jCojalfo8fjklD0Q+nrTVhblYrlUEUISajWp
NYqjcewnSIKI+GJITsaBRRhQsB7+7uZROpp+
```

-----END CERTIFICATE-----

Key

```
0fBdVosD8CoEPfgWvzXyfpCmf5zrGRlnW7d5mTMcgYBn7yZpT2qYLo3dFJSJ9Cx
hufqivgf497UIAbqjNIAj8G07EqTB+vzhuHf9OclgzvLQLiJX2Es2Wf487Ji
DmOM-D06el5uxzujHAVuTctTRkqADNszyTrflf52tgU4dQyX48zaVLPud8svP/y
7QfxGiCoAktb9LBUea+CKQKBgAG8BmEIWQQlet7EuEjduL24RnTH/dDsCbE79g+
GuDOn+QNoeOvqMVlQDAgrWCBnb4vKfQfLWULwzVXdQsTLdr1ckYtmPpiPrxSDN
CT/yVINhLrhoTm2UckSL1Ek/AxtSTZalw8fpFxfv8QfVYIWcTCDyvYSA5P
46x1AoGBAl8BacQ0AeCITDkRPB6aYjlbkB979WZODnvUtsE4Vmrlke7DLsYr5wJ
stIBPAhdca78/SFJ9atZlG3hSCUgu/J3s6PWs5la0RmvHHhbqlvpYts9CXvnJEU2
MpnE/AAOrm6BM0MCLbe4gWnInUFeflpL5jzJLKcNTFOVUO69uFe
-----END RSA PRIVATE KEY-----
```

Server Name Indications

A comma separated list of Server Name Indications. ex: `ssl-example.com,other-ssl-example.com`

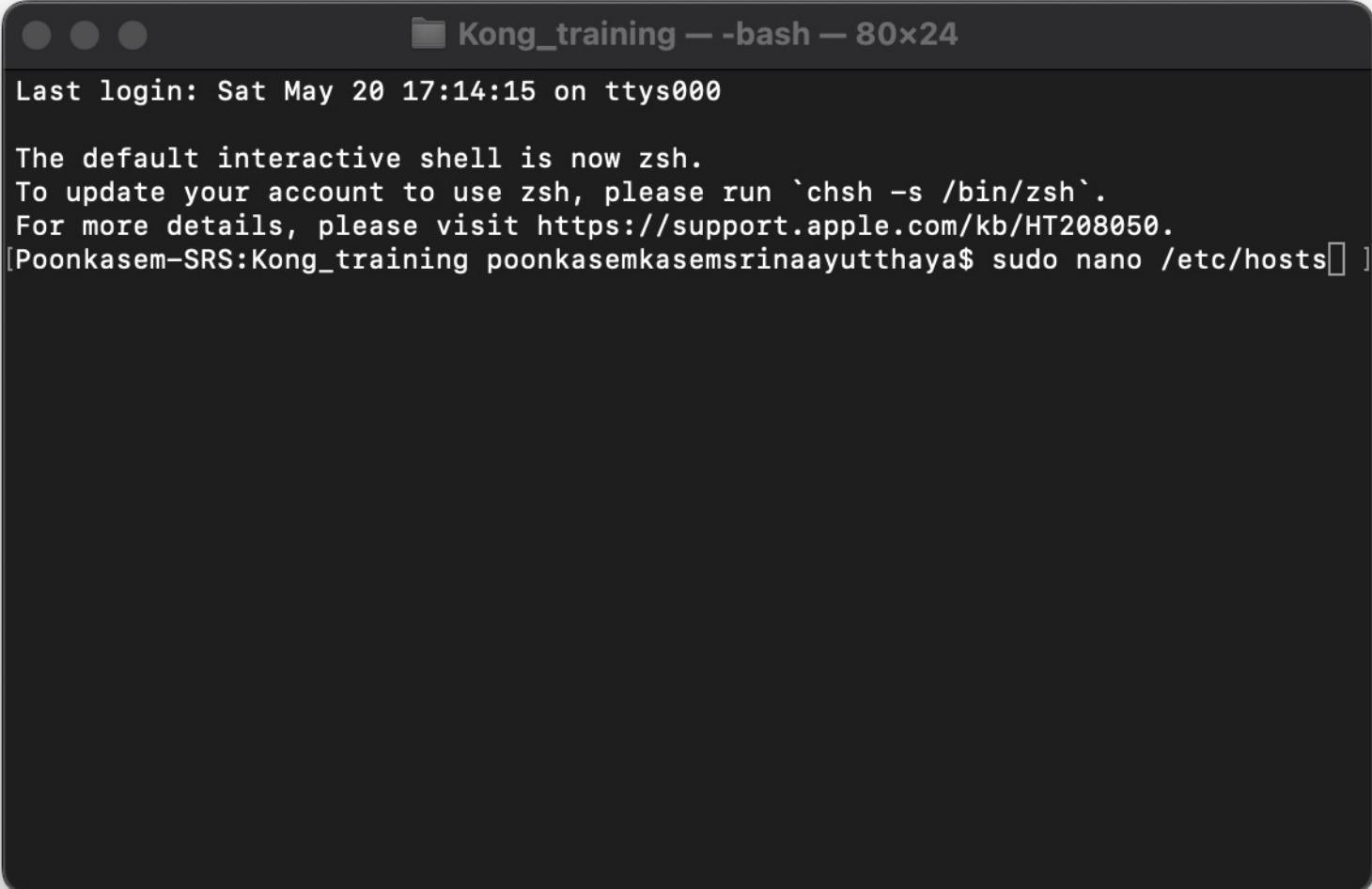
kong.lan|

Tags
(optional)

Connected to kong

- นำเนื้อหาจาก kong.lan.pem และ kong.lan.key ไปใส่ในช่อง Certificate และ Key ตามลำดับ พร้อมกับใส่ Server Name Indication ว่า kong.lan และสร้าง certificate

HTTPS setup

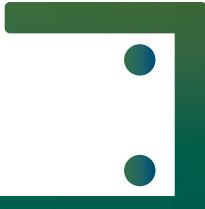


A screenshot of a macOS terminal window titled "Kong_training — -bash — 80x24". The window shows the following text:

```
Last login: Sat May 20 17:14:15 on ttys000
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[Poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$ sudo nano /etc/hosts]
```

▶ เปิด terminal / cmd ไปที่ kong_training อีกครั้ง และรัน command sudo nano /etc/hosts

HTTPS setup



```
UW PICO 5.09                                         File: /etc/hosts

##                                     # Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting. Do not change this entry.
##
127.0.0.1      localhost
255.255.255.255 broadcasthost
::1            localhost
192.168.0.85    kafka
# Added by Docker Desktop
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section

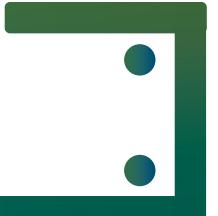
192.168.0.12 kong.lan

^G Get Help   ^O WriteOut   ^R Read File  ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit       ^J Justify    ^W Where is   ^V Next Pg   ^U UnCut Text^T To Spell
```

ทำเพื่อให้สามารถต่อไปที่ชื่อ Server Name
Indication ที่ตั้งไว้ก่อนหน้านี้ได้

- เพิ่ม hosts kong.lan ที่ map ด้วย ip ของเครื่อง (ตัวอย่างบรรทัดล่างสุด) และกด Ctrl+X และ y เพื่อออก

HTTPS testing



8443 คือ port HTTPS ของ Kong

```
[Poonkasem-SRS:Kong_training poonkasemkasemsrinaayutthaya$ curl --cacert ./ca.pem  
-v https://kong.lan:8443/localtest  
* Trying 192.168.0.12:8443...  
* Connected to kong.lan (192.168.0.12) port 8443 (#0)  
* ALPN: offers h2  
* ALPN: offers http/1.1  
* CAfile: ./ca.pem  
* CApth: none  
* (304) (OUT), TLS handshake, Client hello (1):  
* (304) (IN), TLS handshake, Server hello (2):  
* (304) (IN), TLS handshake, Unknown (8):  
* (304) (IN), TLS handshake, Certificate (11):  
* (304) (IN), TLS handshake, CERT verify (15):  
* (304) (IN), TLS handshake, Finished (20):  
* (304) (OUT), TLS handshake, Finished (20):  
* SSL connection using TLSv1.3 / AEAD-AES256-GCM-SHA384  
* ALPN: server accepted http/1.1  
* Server certificate:  
* subject: CN=kong.lan  
* start date: May 20 09:48:36 2023 GMT  
* expire date: May 18 09:48:36 2028 GMT  
* subjectAltName: host "kong.lan" matched cert's "kong.lan"  
* issuer: CN=kong.lan  
* SSL certificate verify ok.
```

➤ รัน curl --cacert ca.pem -v https://kong.lan:8443/localtest เพื่อทดสอบการเชื่อมต่อไปแบบ HTTPS ไปที่ Kong



THANK YOU
