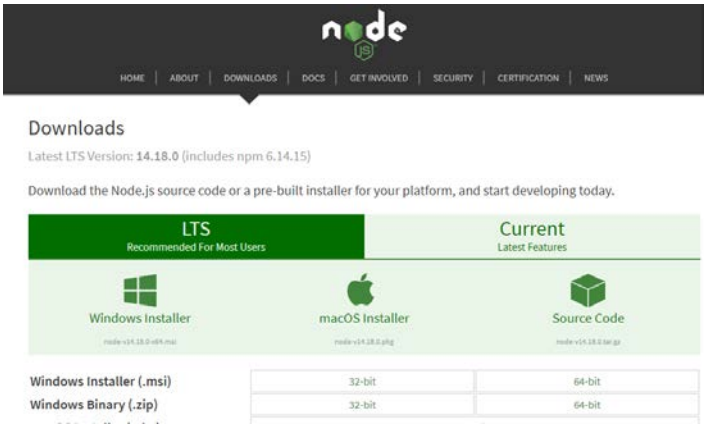
Tutorial 07: Microservice API Gateway using KONG
Date: October 2022
Architecture Pro/Con:
Clients => API Gateway => Services
Demo: Kong and Konga
OS: Windows10
Pre-requisite: NodeJS, Docker, Postgres, Kong, Konga, Insomnia

***ดู vdo ประกอบที่ https://www.mycourseville.com/?q=cvdlit/theatre/youtube/list/4867/0
***NodeJS source code ที่ใช้ประกอบข้อ 2.2 f) อยู่ที่
https://www.mycourseville.com/?q=courseville/course/35365/view_content_node_1007324_material

Step 0: Installations of NodeJS and Docker

0.1) Install NodeJS for Windows10 Home
- to create http services for testing the API gateway
- goto download site https://nodejs.org/en/download/



- download msi installer named “node-v14.18.0-x64.msi” (or the latest LTS one) and run it.
- verify the installation using command windows and type

  c:\>node -v (จะแสดง version)

  c:\>npm -v (จะแสดง version)

0.2) Install Docker Desktop for Windows10 Home
- to provide the images and containers management tools
- goto the download site https://docs.docker.com/docker-for-windows/install/

- please enable Hyper-V Windows Features or install required Windows components for WSL2 option ในหน้า Configuration page ตามที่อธิบายในขั้นตอนข้างล่างนี้

Install Docker Desktop on Windows

1. Double-click **Docker Desktop Installer.exe** to run the installer.

   If you haven't already downloaded the installer (`Docker Desktop Installer.exe`), you can get it from Docker Hub. It typically downloads to your `Downloads` folder, or you can run it from the recent downloads bar at the bottom of your web browser.

2. When prompted, ensure the **Enable Hyper-V Windows Features** or the **Install required Windows components for WSL 2** option is selected on the Configuration page.
3. Follow the instructions on the installation wizard to authorize the installer and proceed with the install.
4. When the installation is successful, click **Close** to complete the installation process.
5. If your admin account is different to your user account, you must add the user to the **docker-users** group. Run **Computer Management** as an administrator and navigate to **Local Users and Groups** > **Groups** > **docker-users**. Right-click to add the user to the group. Log out and log back in for the changes to take effect.

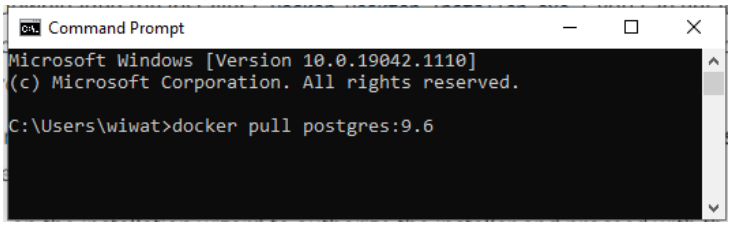
- the Docker Desktop is bundled to facilitate the images, and containers arrangement (แต่ใน tutorial นี้จะใช้งานแค่ docker ผ่านทาง command line เท่านั้น)
- to verify the docker desktop installation using command windows and type
  c:\> docker -v (แสดง version)

Step 1: Install containers of Postgres 9.6, Kong

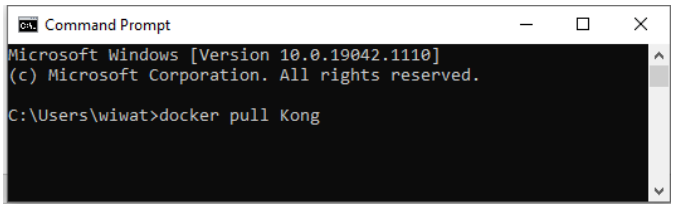
1.1) Install Postgres 9.6 image using Docker to pull Postgres (exactly version 9.6) from dockerhub (see https://hub.docker.com/_/postgres)

At cmd prompt:

docker pull postgres:9.6



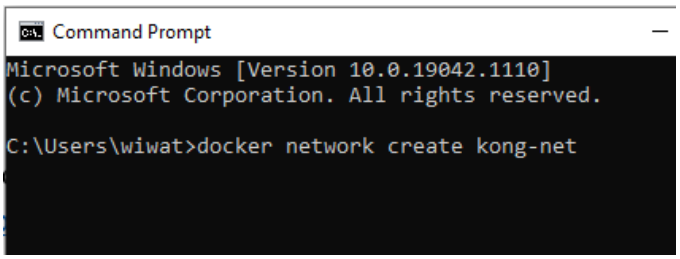
1.2) Install Kong image using Docker from dockerhub (see https://hub.docker.com/_/kong/)
At cmd prompt:
docker pull kong



1.3) Create Containers of Postgres and Kong using Docker Run

a) Create a docker network named kong-net to deploy kong and PostgresSQL with connectivity

At cmd prompt:
docker network create kong-net



b) Start the Postgres DB docker into the "kong-net" network

At cmd prompt:
docker run -d --name kong-database --network=kong-net -p 5432:5432 -e "POSTGRES_USER=kong" -e "POSTGRES_DB=kong" -e "POSTGRES_PASSWORD=kong" postgres:9.6

c) Run migration script on the Postgres DB and get it ready for Kong

At cmd prompt:
docker run --rm --network=kong-net -e "KONG_DATABASE=postgres" -e "KONG_PG_HOST=kong-database" -e "KONG_PG_PASSWORD=kong" kong:latest kong migrations bootstrap

d) Start the actual Kong docker

At cmd prompt:
docker run -d --name kong --network=kong-net -e "KONG_DATABASE=postgres" -e "KONG_PG_HOST=kong-database" -e "KONG_PG_PASSWORD=kong" -e "KONG_PROXY_ACCESS_LOG=/dev/stdout" -e "KONG_ADMIN_ACCESS_LOG=/dev/stdout" -e "KONG_PROXY_ERROR_LOG=/dev/stderr" -e "KONG_ADMIN_ERROR_LOG=/dev/stderr" -e "KONG_ADMIN_LISTEN=0.0.0.0:8001, 0.0.0.0:8444 ssl" -e "KONG_PROXY_LISTEN= 0.0.0.0:8000, 0.0.0.0:9080 http2, 0.0.0.0:9081 http2 ssl" -p 8000:8000 -p 8443:8443 -p 127.0.0.1:8001:8001 -p 127.0.0.1:8444:8444 -p 127.0.0.1:9080:9080 -p 127.0.0.1:9081:9081 kong

e) Check if Kong is up and running with

At cmd prompt:

curl -i http://localhost:8001/

f) Add service

At cmd prompt:

curl -i -X POST --url http://localhost:8001/services/ --data name=exampleservice1 --data url=http://mockbin.org

g) Add route

At cmd prompt:

curl -i -X POST --url http://localhost:8001/services/exampleservice1/routes --data paths[]=/mock

h) Verify Kong's methods and paths mapping

At browser:

http://localhost:8000/mock (ซึ่ง Kong จะ map ไปที่ http://mockbin.org แทน)

Step 2: Install Konga

2.1) Install Konga image

(see https://hub.docker.com/r/pantsel/konga/)

At cmd prompt:

docker pull pantsel/konga

2.2) Create Containers of Konga using Docker Run (Konga uses our installed Postgres DB)

a) Prepare Konga

At cmd prompt:

docker run --rm –-network=kong-net pantsel/konga -c prepare -a postgres -u postgresql://kong:kong@kong-database:5432/konga

b) Run Konga

At cmd prompt:

docker run -p 1337:1337 --network=kong-net -e "DB_ADAPTER=postgres" -e "DB_HOST=kong-database"
-e "DB_URI=postgresql://kong:kong@kong-database:5432/konga" -e "KONGA_HOOK_TIMEOUT=120000" -e "NODE_ENV=production" --name konga pantsel/konga

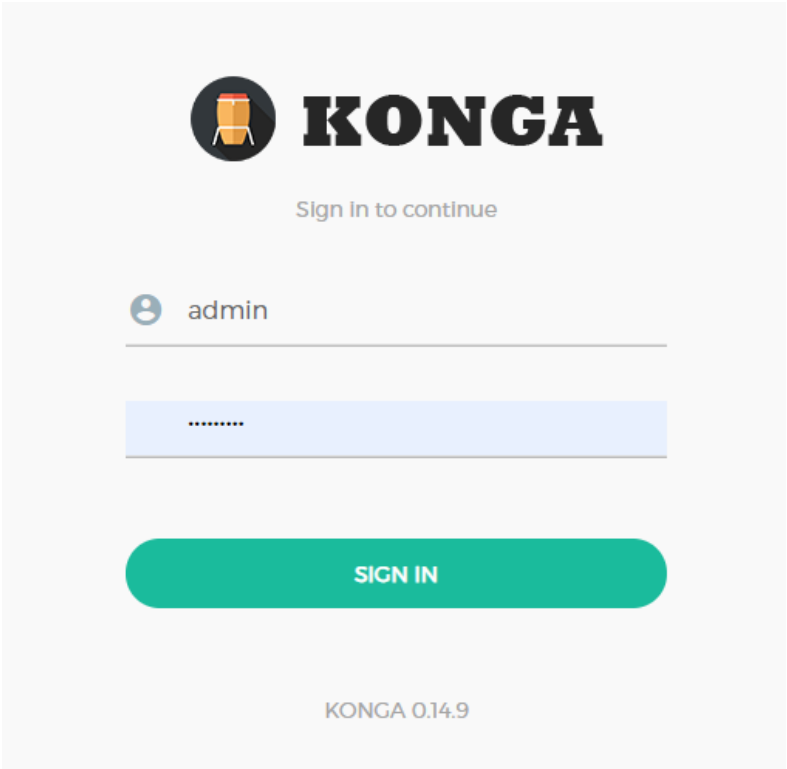
**จากนั้น ให้ค้างจอ cmd prompt ไว้สำหรับ Konga

c) Open web app of Konga

At web browser:

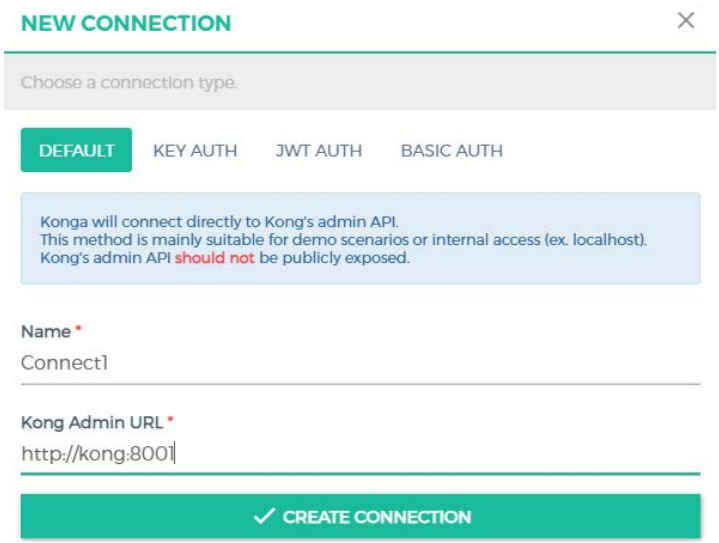
http://localhost:1337

d) Setup Konga User



e) Setup connection to Kong

Name = “Connect1” and Kong Admin URL = http://kong:8001 as shown



f) Create local http/REST/gRPC services using nodeJS
   - ให้ download source code ของ nodeJS จาก link ที่กำหนดให้
   - ทำการ start services ทั้งสามแบบตามที่ดูใน vdo
   - โดยระวังว่าต้องเปิด command windows ค้างไว้เสมอสำหรับ service ที่ start ด้วย
   - ทดสอบโดยใช้ browser เรียก http/REST services
   - ติดตั้ง Insomnia (from https://insomnia.rest/download) เพื่อทดสอบ gRPC service ด้วย

Step3: Using Konga to create service and route for http://localhost:8081

Step4: Using Konga to create service and route and upstream for load balancer to http://localhost:8081 and http://localhost:8082

Step 5: Using Konga to create REST service and route

Step 6: Using Konga to create gRPC service and route

Step 7: Basic Authentication of KONG's consumers port 8000

3.1) Using Konga, setup consumers and credentials (passwords) as shown in VDO clip
3.2) add plugin Basic Authen to a particular service

Step 8: Key Authentication for KONG's consumers at port 8000

Assignment:

1) ติดตั้งและใช้งาน KONG เป็น API Gateway เพื่อไปใช้กับ REST/gRPC Services ในโจทย์ที่ผ่านมา และทำการ Capture หน้าจอแสดง services และ routes ที่ใช้ (กรณีที่ Services อยู่ที่ localhost ให้ใช้ ip address ของตนแทน โดยตรวจดู ip address ได้ใช้คำสั่ง ipconfig ที่ cmd window)