# Kong API Gateway Docs (.net)

**Kong API Gateway :**

API gateway built for hybrid and multi-cloud, optimized for microservices and distributed architectures

**Introducing Kong Gateway:**

Kong Gateway is a lightweight, fast, and flexible cloud-native API gateway. An API gateway is a reverse proxy that lets you manage, configure, and route requests to your APIs.

Kong Gateway runs in front of any RESTful API and can be extended through modules and plugins. It’s designed to run on decentralized architectures, including hybrid-cloud and multi-cloud deployments.

**Flow Diagram**

**A diagram of a company

Description automatically generated**

**Environment Setup:**

**step 1 :** install docker in your system

[https://desktop.docker.com/win/main/arm64/Docker%20Desktop%20Installer.exe?utm\_source=docker&utm\_medium=webreferral&utm\_campaign=dd-smartbutton&utm\_location=module&\_gl=1\*8gefju\*\_gcl\_au\*MTc0NTMwMDYyOC4xNzI5NjY5Nzky\*\_ga\*MTMzNjExNTQyLjE3MjU5NTQ1MjE.\*\_ga\_XJWPQMJYHQ\*MTcyOTY2OTc5Mi4yLjEuMTcyOTY2OTc5OC41NC4wLjA.](https://desktop.docker.com/win/main/arm64/Docker%20Desktop%20Installer.exe?utm_source=docker&utm_medium=webreferral&utm_campaign=dd-smartbutton&utm_location=module&_gl=1*8gefju*_gcl_au*MTc0NTMwMDYyOC4xNzI5NjY5Nzky*_ga*MTMzNjExNTQyLjE3MjU5NTQ1MjE.*_ga_XJWPQMJYHQ*MTcyOTY2OTc5Mi4yLjEuMTcyOTY2OTc5OC41NC4wLjA.)

**login to your emailid in Docker desktop**

**step 2 :** install progress to kong

use below the comment code using any command tools

*docker run -d --name kong-database*

*-e POSTGRES\_USER=kong*

*-e POSTGRES\_DB=kong*

*-e POSTGRES\_PASSWORD=kong*

*-p 5432:5432*

*postgres:13*

**step 3 :** install kong to postgress

use below the comment code using any command tools

*docker run --rm*

*--link kong-database:postgres*

*-e "KONG\_DATABASE=postgres"*

*-e "KONG\_PG\_HOST=postgres"*

*-e "KONG\_PG\_PASSWORD=kong"*

*-e "KONG\_CASSANDRA\_CONTACT\_POINTS=kong-database"*

*kong:latest kong migrations bootstrap*

**step 4 :** connect to both kong and postgress

use below the comment code using any command tool

*docker run -d --name kong --link kong-database:postgres*

*-e "KONG\_DATABASE=postgres"*

*-e "KONG\_PG\_HOST=postgres"*

*-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\_LOG\_LEVEL=info"*

*-e "KONG\_ADMIN\_LISTEN=0.0.0.0:8001"*

*-e "KONG\_ADMIN\_GUI\_URL=http://localhost:8002"*

*-p 8000:8000*

*-p 8443:8443*

*-p 8001:8001*

*-p 8444:8444*

*-p 8002:8002*

*-p 8445:8445*

*kong:latest*

**step 5 :** explain all kong port

**8000 --** fetch data from kong gateway and proxy port

**8001 --** Admin API: Port 8001 is used to interact with Kong's Admin API, which is a RESTful API for managing Kong's configurations.

**8002 --** kong manger port

**8443 --** kong proxy SSL Port

**8444 --** This is a secure version of the Admin API, similar to the one on port 8001 but using TLS/SSL for encrypted communication**.**

**8445 --** kong manager SSL port

**step 6 : project flow**

1. create a new api web application project or already existing project

2. add the dockerfile inside the project

3. open the powershell then locate your project folder

4. built your project use this comment ( docker build -t username/imagename:v1 . ) using any command tool

Username : your docker username

Imagename : any name

5. in local we don’t use push commad in cloud projects we use push comment like this (docker push username/yourimagename:v1) using any command tool

Username : your docker username

Imagename : any name

**step 7:** set up to kong services and kong route

1. open your kong 8002 port in your browers

2. first create the Services click the gateway Services below this png for your requirement

A screenshot of a computer

Description automatically generated

3.Then create the route of this service below the png for your requirement

A screenshot of a computer

Description automatically generated

4. add the both kong services and route you can test the localhost:8000 port in your apis

A screenshot of a computer

Description automatically generated

5.**manually add the service and route**

Use this API postman or swagger

**service request endpoint:**

**Post:** [**http://localhost:8001/services**](http://localhost:8001/services)

**Host :** If you have an app services url replace the localhost

**Body:**

{

"name": "your services",

"url": "your docker image url"

}

**routes request endpoint**

**Post:** http//localhost:8001/services/ your-services /routes

**Body:**

{

"name":"your-route-name",

"paths":["your-api-path"],

"strip\_path":false,

"service":

{

"name": "your-service-name"

}

}

**Notes:**

**"paths":["your-api-path"] :** like this /api/v1/employee