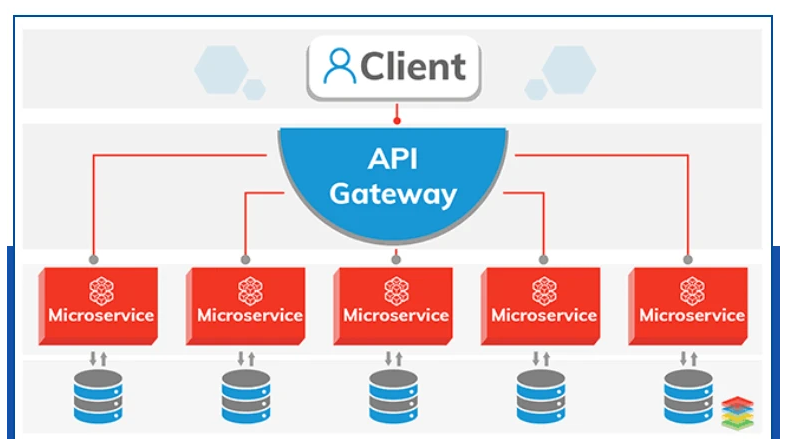
**Worksheet Pertemuan 23 - Praktik Microservices using Docker (Individu)**

NIM : 21523090

Nama : Satriya Adhi Pradana



1. **Objektif**
2. Membuat aplikasi kalkulator sederhana berbasis microservice menggunakan docker container
3. **Kebutuhan aplikasi**
4. Docker
5. Postman
6. Git Bash (untuk OS Windows)
7. Kong (API Management Gateway for Microservices)

**\*Perhatian**: sebelum memulai langkah berikutnya, pastikan aplikasi Docker sudah berjalan di perangkat Anda

1. **Instalasi Kong**

Sebelum melakukan instalasi Kong pastikan bahwa di komputer sudah terpasang aplikasi **Docker**, **Postman**, dan **Git Bash** (pada praktik ini ketiga aplikasi tersebut sudah terpasang di komputer)

**Langkah 1: Membuat Docker network**

1. Docker network ini akan digunakan untuk menghubungkan beberapa kontainer yang sudah dibuat agar dapat saling berkomunikasi
2. Jalankan aplikasi Git Bash
3. Jalankan perintah di bawah ini:

| **$ docker network create --subnet=192.168.1.0/24 --gateway=192.168.1.1 kong-net** |
| --- |

1. Perintah di atas digunakan untuk membuat network dengan:
   1. Nama jaringan: **kong-net** (boleh menggunakan nama lainnya)
   2. Jaringan: **192.168.1.0/24**
   3. Alamat gateway: **192.168.1.1**
2. Untuk melihat semua jaringan yang ada, jalankan perintah di bawah ini

**$ docker network ls**

Lakukan pengambilan tangkapan layar lalu letakkan hasilnya di bawah ini

|  |
| --- |

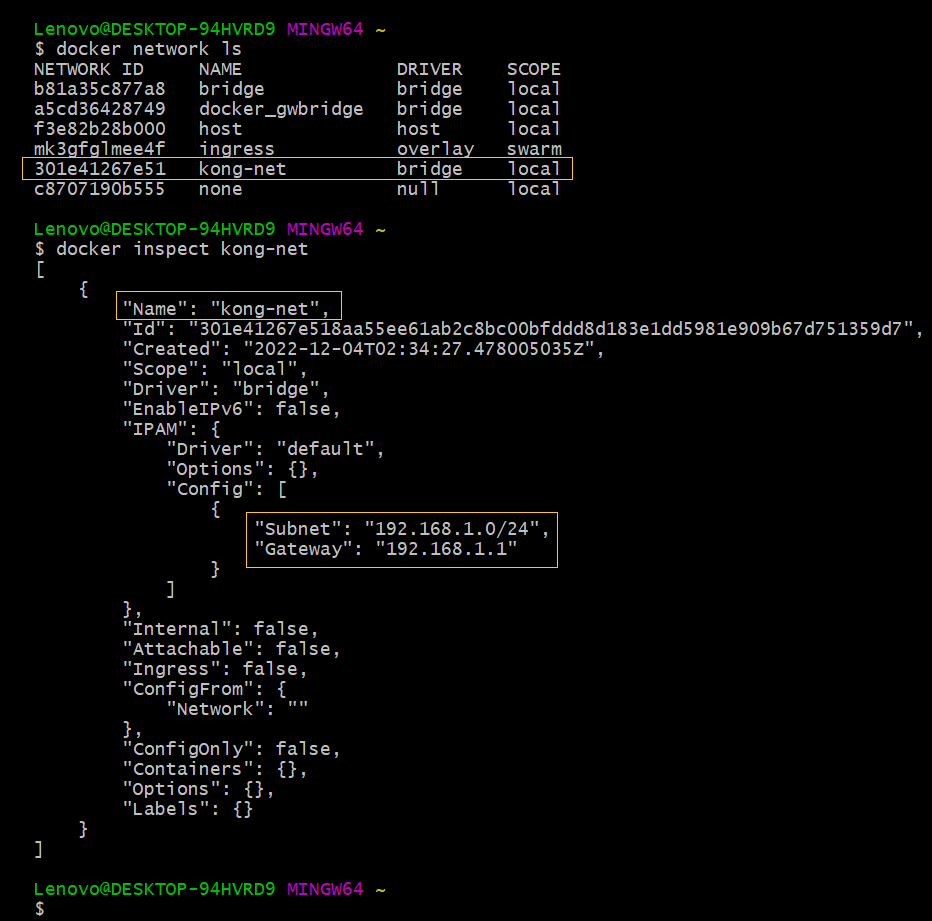
1. Untuk melihat detail informasi jaringan dapat menggunakan perintah:

**$ docker inspect <nama\_jaringan>**

1. Misalkan kita ingin melihat detail dari jaringan **kong-net** maka jalankan perintah di bawah ini:

**$ docker inspect kong-net**

1. Beberapa hal yang perlu diperhatikan dari isian detail informasi tersebut antara lain:



* 1. Jaringan **kong-net** menggunakan alamat jaringan **192.168.1.0/24** dengan default gateway **192.168.1.1**
  2. Setiap kontainer yang tergabung dengan jaringan **kong-net** nantinya akan memiliki alamat ip yang termasuk dalam jaringan **192.168.1.0/24**

**Langkah 2: Menyiapkan dan menjalankan Postgres DB**

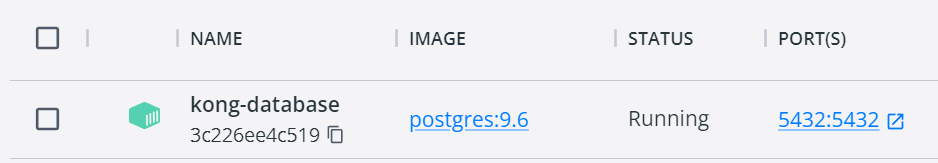
1. Kita akan menggunakan Postgres database yang digunakan untuk menyimpan konfigurasi **Kong**
2. Jalankan aplikasi **Git Bash**
3. Jalankan perintah di bawah ini

| docker run -d --name kong-database --network=kong-net -e "POSTGRES\_USER=kong" -e "POSTGRES\_DB=kong" -e "POSTGRES\_PASSWORD=kong" -p 5432:5432 postgres:9.6 |
| --- |

1. Beberapa hal yang perlu diperhatikan pada perintah di atas yaitu:
   1. **--name kong-database**: nama kontainer yang akan dibuat/dijalankan
   2. **--network=kong-net**: kontainer yang dibuat/dijalankan akan dimasukkan ke dalam jaringan **kong-net** yang sudah dibuat sebelumnya
   3. **"POSTGRES\_USER=kong"**: nama user yang digunakan pada Postgres
   4. **"POSTGRES\_DB=kong"**: nama database yang digunakan
   5. **"POSTGRES\_PASSWORD=kong"**: password yang digunakan
   6. **-p 5432:5432**: (publish a container's port(s) to the host) proses mapping port yang digunakan kontainer **kong-database** yaitu **5432** (bagian kedua)agar bisa diakses melalui PC host melaui port 5432 (bagian kesatu)
   7. **postgres:9.6**: versi Postgres yang digunakan
2. Jalankan perintah di bawah ini untuk menginstall aplikasi Kong dan menghubungkannya dengan database yang sudah dibuat sebelumnya

| 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 |
| --- |

1. Beberapa hal yang perlu diperhatikan pada perintah di atas yaitu:
   1. Opsi **--rm**: secara otomatis akan menghapus kontainer jika sudah ada sebelumnya
   2. **kong:latest** : image yang akan digunakan untuk menjalankan kontainer
   3. **migrations bootstrap**: membuat beberapa table yang dibutuhkan di database sebelum kontainernya dijalankan (Before starting Kong, you need to bootstrap the database, meaning kong needs to create some tables in the database before it starts.)
2. Jika berhasil, maka pada aplikasi Docker akan muncul tampilan seperti di bawah ini yang menandakan kontainer **kong-database** berhasil dijalankan pada port **5432** dengan versi **postgres:9.6**

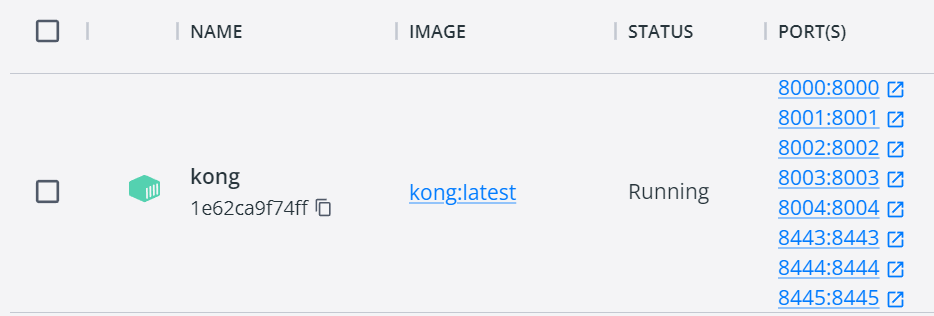


**Langkah 3: Menjalankan Kong**

1. Jalankan aplikasi Git Bash
2. Jalankan perintah di bawah ini

| 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\_ADMIN\_LISTEN=0.0.0.0:8001, 0.0.0.0:8444 ssl" \  -p 8000:8000 -p 8443:8443 -p 8001:8001 -p 8444:8444 \  -p 8002:8002 -p 8445:8445 \  -p 8003:8003 -p 8004:8004 \  kong:latest |
| --- |

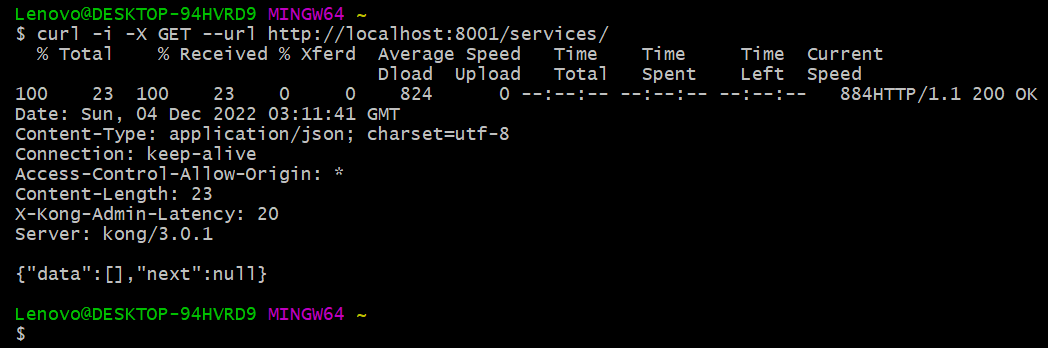
1. Beberapa hal yang perlu diperhatikan pada perintah di atas yaitu:
   1. -**-name kong**: nama kontainer yang akan dibuat/dijalankan
   2. **--network=kong-net**: kontainer yang dibuat/dijalankan akan dimasukkan ke dalam jaringan **kong-net** yang sudah dibuat sebelumnya
   3. **kong:latest**: nama image docker yang digunakan
2. Jika berhasil maka pada aplikasi Docker akan muncul tampilan seperti di bawah ini yang menandakan kontainer **kong** berhasil dengan menggunakan image docker **kong:latest**



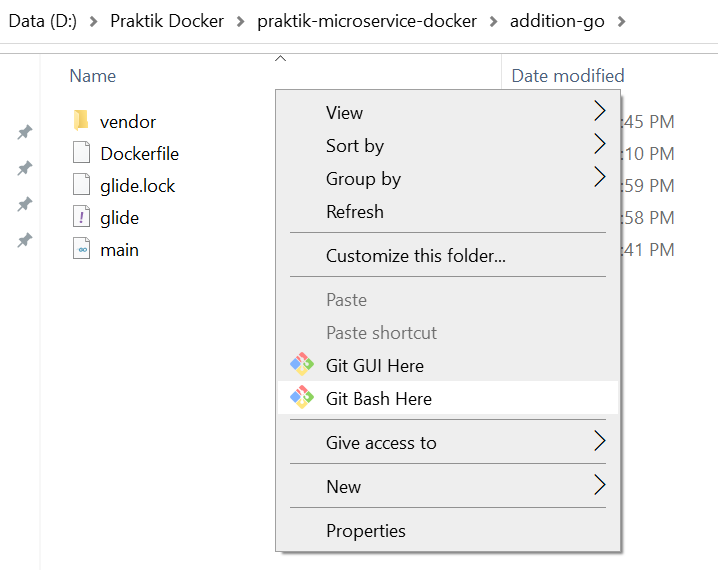
1. Lakukan verifikasi instalasi Kong dengan menjalankan perintah di bawah ini melalui **Git Bash**:

**$ curl -i -X GET --url http://localhost:8001/services/**

1. Jika instalasi berhasil maka akan muncul tampilan seperti di bawah ini



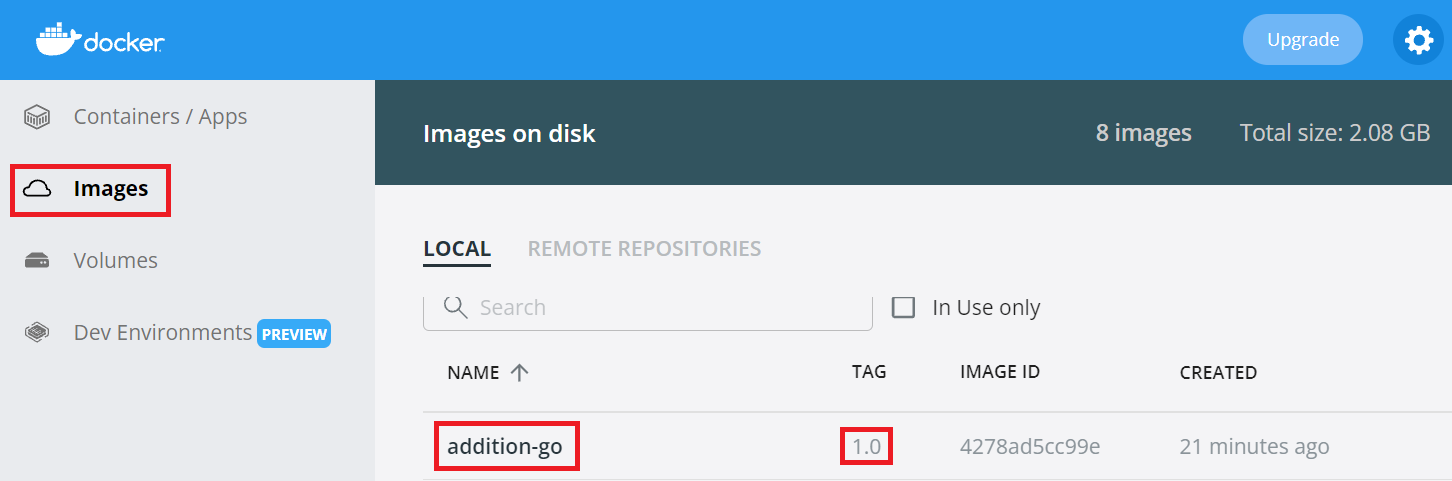
1. Kong sudah siap digunakan untuk dijadikan sebagai API Gateway
2. **Membuat dan Menjalankan Docker Kontainer addition-go**
3. **addition-go** merupakan kontainer yang memberikan layanan (service) untuk melakukan operasi penambahan yang dibuat dengan menggunakan bahasa **Go**
4. Silakan unduh dan ekstrak fail **praktik-microservice-docker.zip** yang sudah disediakan di Classroom.
5. Buka folder **praktik-microservice-docker\addition-go**
6. Klik kanan di area kosong di dalam folder tersebut lalu pilih **Git Bash Here**



1. Jalankan perintah di bawah ini untuk membuat image docker dengan nama **addition-go:1.0**

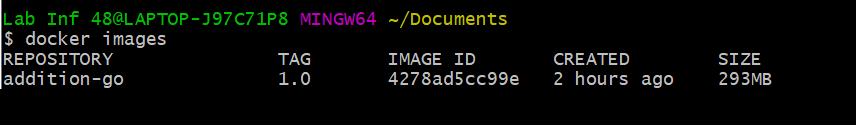
**$ docker build -t addition-go:1.0 .**

1. Jika sudah berhasil maka akan muncul tampilan seperti di bawah ini pada aplikasi Docker



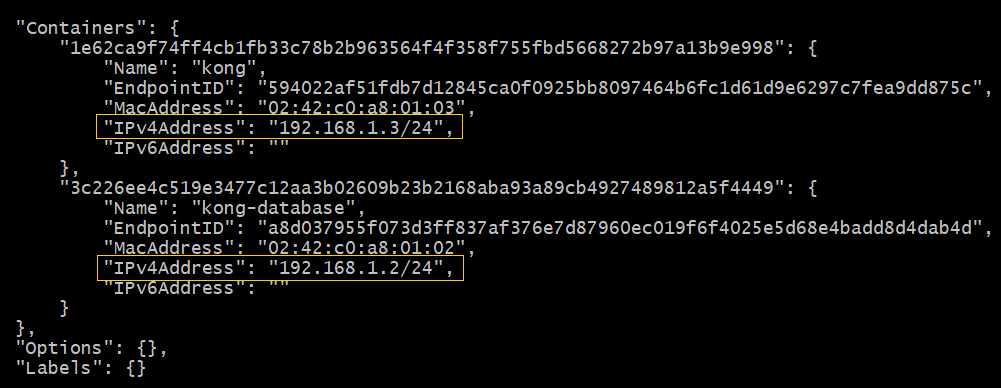
1. Atau bisa juga dengan mengecek melalui aplikasi Git Bash dengan menjalankan perintah di bawah ini:

**$ docker images**



1. Sebelum membuat dan menjalankan kontainer **addition-go**, cek terlebih dahulu alamat ip yang sudah digunakan di jaringan **kong-net**. Hal ini bertujuan mencari alamat ip yang belum digunakan dan alamat tersebut akan kita gunakan sebagai alamat ip pada kontainer **addition-go.**
2. Jalankan perintah di bawah ini

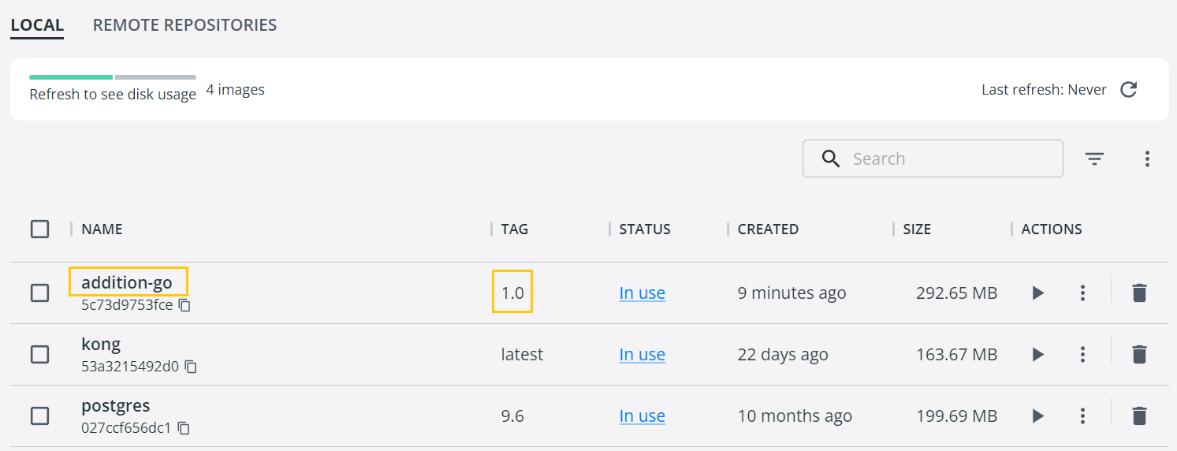
**$ docker inspect kong-net**

****

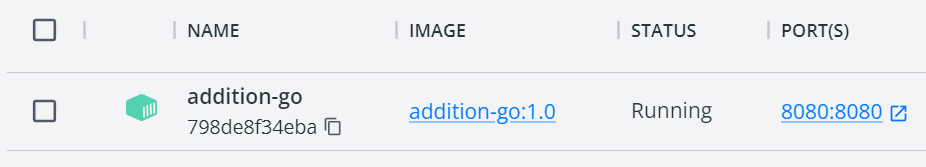
1. Misalkan pada gambar di atas sudah terdapat dua buah kontainer yaitu **kong** dengan alamat ip **192.168.1.3/24** dan **kong-database** dengan alamat ip **192.168.1.2/24.**
2. Dengan demikian, maka alamat ip yang bisa digunakan pada kontainer **addition-go** nantinya mulai dari **192.168.1.4/24, 192.168.1.5/24, 192.168.1.6/24,** dan seterusnya
3. Misalkan kita akan menggunakan alamat ip **192.168.1.4/24** sehingga untuk membuat dan menjalankan kontainer **addition-go** dapat menggunakan perintah di bawah ini:

| docker run -d --name addition-go --network kong-net --ip 192.168.1.4 -p 8080:8080 addition-go:1.0 |
| --- |

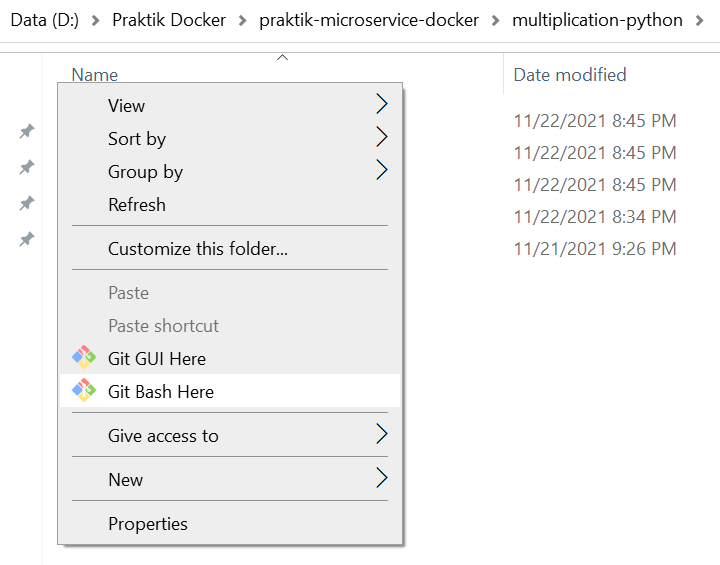
1. Beberapa hal yang perlu diperhatikan dari perintah di atas yaitu:
   1. --name **addition-go**: nama kontainer yang akan dibuat
   2. --network **kong-net**: kontainer yang dibuat akan dimasukkan ke dalam jaringan **kong-net** yang sudah dibuat sebelumnya
   3. --ip **192.168.1.4**: kontainer yang dibuat akan diberikan alamat ip tersebut
   4. **addition-go:1.0**: merupakan nama image yang digunakan untuk membuat kontainer tersebut yang sudah dibuat sebelumnya



1. Jika berhasil maka akan muncul tampilan seperti di bawah ini pada aplikasi Docker



1. **Membuat dan Menjalankan Docker Kontainer multiplication-python**
2. **multiplication-python** merupakan kontainer yang memberikan layanan (service) untuk melakukan operasi perkalian yang dibuat dengan menggunakan bahasa **Python**
3. Buka folder **praktik-microservice-docker\multiplication-python**
4. Klik kanan di area kosong di dalam folder tersebut lalu pilih **Git Bash Here**



1. Jalankan perintah di bawah ini untuk membuat image docker dengan nama **multiplication-python:1.0**

**$ docker build -t multiplication-python:1.0 .**

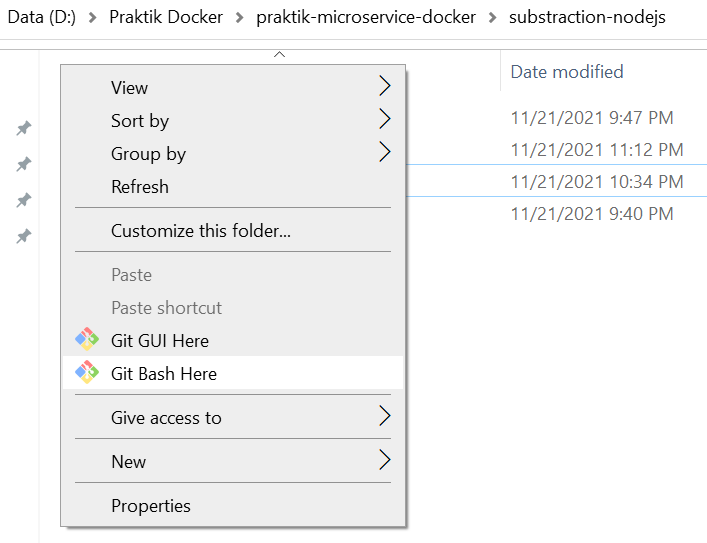
1. Misalkan kita akan menggunakan alamat ip **192.168.1.5/24** yang masih tersedia **(periksa kembali poin D.8 sampai D.11 jika Anda belum paham)** sehingga untuk membuat dan menjalankan kontainer **multiplication-python** dapat menggunakan perintah di bawah ini:

| docker run -d --name multiplication-python --network kong-net --ip 192.168.1.5 -p 5000:5000 multiplication-python:1.0 |
| --- |

1. Lakukan verifikasi apakah pembuatan kontainer **multiplication-python** sudah berhasil.

|  |
| --- |

1. **Membuat dan Menjalankan Docker Kontainer subtraction-nodejs**
2. **subtraction-nodejs** merupakan kontainer yang memberikan layanan (service) untuk melakukan operasi pengurangan yang dibuat dengan menggunakan **Nodejs**
3. Buka folder **praktik-microservice-docker\subtraction-nodejs**
4. Klik kanan di area kosong di dalam folder tersebut lalu pilih **Git Bash Here**



1. Jalankan perintah di bawah ini untuk membuat image docker dengan nama **subtraction-nodejs:1.0**

**$ docker build -t subtraction-nodejs:1.0 .**

1. Misalkan kita akan menggunakan alamat ip **192.168.1.6/24** yang masih tersedia **(periksa kembali poin D.8 sampai D.11 jika Anda belum paham)** sehingga untuk membuat dan menjalankan kontainer **subtraction-nodejs** dapat menggunakan perintah di bawah ini:

| docker run -d --name subtraction-nodejs --network kong-net --ip 192.168.1.6 -p 3000:3000 subtraction-nodejs:1.0 |
| --- |

1. Lakukan verifikasi apakah pembuatan kontainer **subtraction-nodejs** sudah berhasil.

|  |
| --- |

1. **Membuat dan Menjalankan Docker Kontainer division-php**
2. **division-php** merupakan kontainer yang memberikan layanan (service) untuk melakukan operasi pembagian yang dibuat dengan menggunakan bahasa **PHP**
3. Buka folder **praktik-microservice-docker\division-php**
4. Klik kanan di area kosong di dalam folder tersebut lalu pilih **Git Bash Here**
5. Jalankan perintah di bawah ini untuk membuat image docker dengan nama **division-php:1.0**

**$ docker build -t division-php:1.0 .**

1. Misalkan kita akan menggunakan alamat ip **192.168.1.7/24** yang masih tersedia **(periksa kembali poin D.8 sampai D.11 jika Anda belum paham)** sehingga untuk membuat dan menjalankan kontainer **division-php** dapat menggunakan perintah di bawah ini:

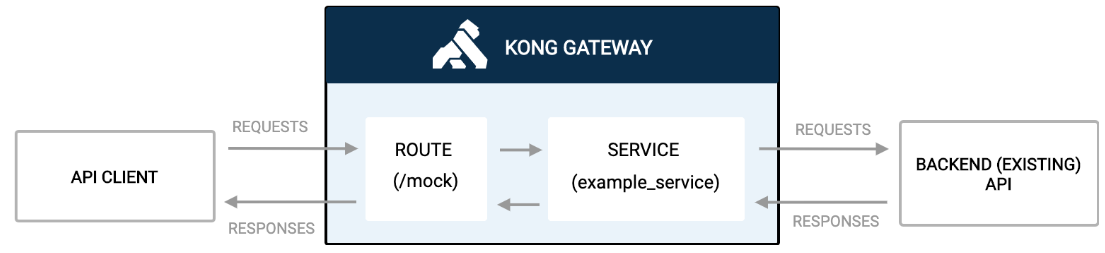
| docker run -d --name division-php --network kong-net --ip 192.168.1.7 -p 80:80 division-php:1.0 |
| --- |

1. Lakukan verifikasi apakah pembuatan kontainer **division-php** sudah berhasil.

|  |
| --- |

1. **Membuat Service dan Routes pada Kong**

* **Service** yaitu entitas untuk layanan API yang mendefinisikan upstream service kita.
* **Route** yaitu *endpoint* untuk masuk ke **Kong** yang kemudian akan mengarahkan ke *service* tertentu.



* + - 1. **Langkah 1: Membuat Service**

Service yang akan kita buat berjumlah **empat buah** sesuai dengan **jumlah kontainer** yang sudah dibuat sebelumnya melalui **Git Bash**.

* 1. Membuat service **addition\_go** (jalankan perintah di bawah ini melalui **Git Bash**)

| curl -i -s -X POST http://localhost:8001/services/ --data name='addition\_go' --data url='http://192.168.1.4:8080' |
| --- |

Perintah di atas akan membuat service dengan nama **addition\_go** dengan url upstream **http://192.168.1.4:8080** (alamat IP dari kontainer addition-go)

* 1. Membuat service **multiplication\_python**

| curl -i -s -X POST http://localhost:8001/services/ --data 'name=multiplication\_python' --data 'url=http://192.168.1.5:5000' |
| --- |

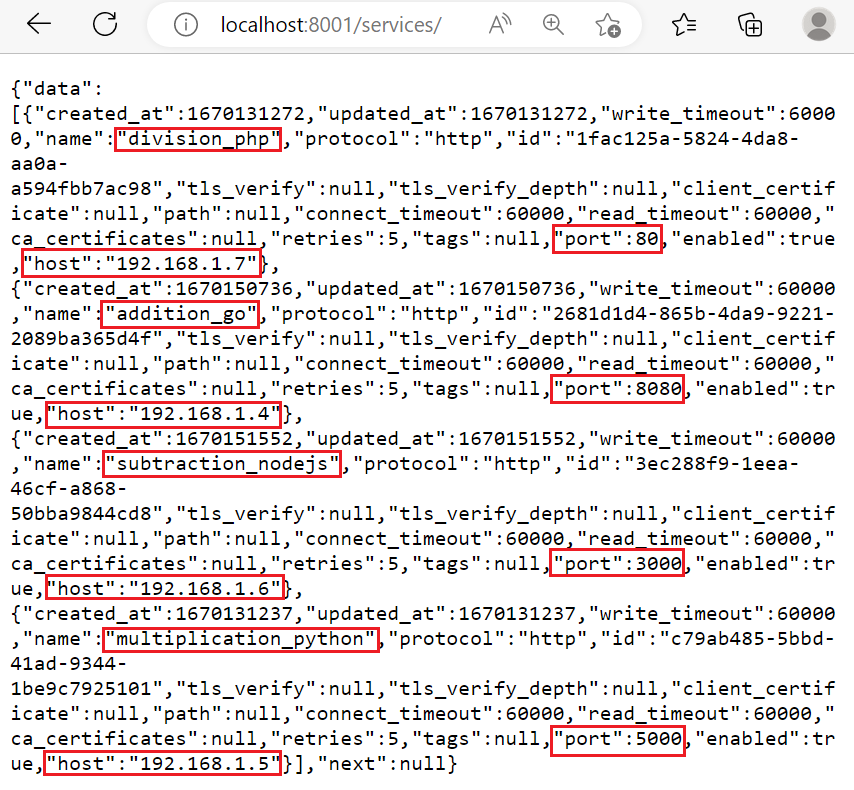
* 1. Membuat service **subtraction\_nodejs**

| curl -i -s -X POST http://localhost:8001/services/ --data 'name=subtraction\_nodejs' --data 'url=http://192.168.1.6:3000' |
| --- |

* 1. Membuat service **division\_php**

| curl -i -s -X POST http://localhost:8001/services/ --data 'name=division\_php' --data 'url=http://192.168.1.7:80' |
| --- |

Untuk mengecek service yang sudah ada/dibuat dapat dengan mengetikkan URL ini di **browser**: **http://localhost:8001/services/**



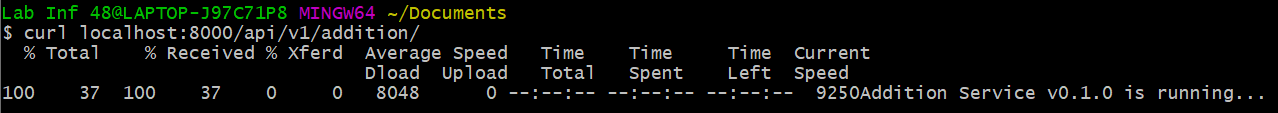
1. **Langkah 2: Membuat Route**
2. Membuat route **/api/v1/addition** (jalankan perintah di bawah ini melalui **Git Bash**)

| curl -i -X POST --url http://localhost:8001/services/**addition\_go**/routes --data 'paths[]=**/api/v1/addition**' |
| --- |

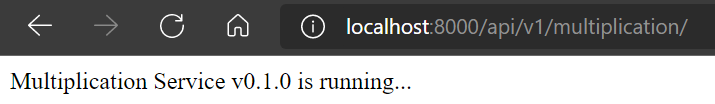
Perintah di atas membuat route pada **service addition\_go** dengan *path* dari route-nya yaitu **/api/v1/addition**

Untuk melakukan verifikasi apakah route tersebut berhasil dibuat dan berjalan dengan benar dapat dengan menjalankan perintah di bawah ini melalui **Git Bash**:

**$ curl localhost:8000/api/v1/addition/**

****

Atau dengan menjalankan URL **localhost:8000/api/v1/addition/** pada **browser**



1. Membuat route **/api/v1/multiplication**

| curl -i -X POST --url http://localhost:8001/services/**multiplication\_python**/routes --data 'paths[]=**/api/v1/multiplication**' |
| --- |

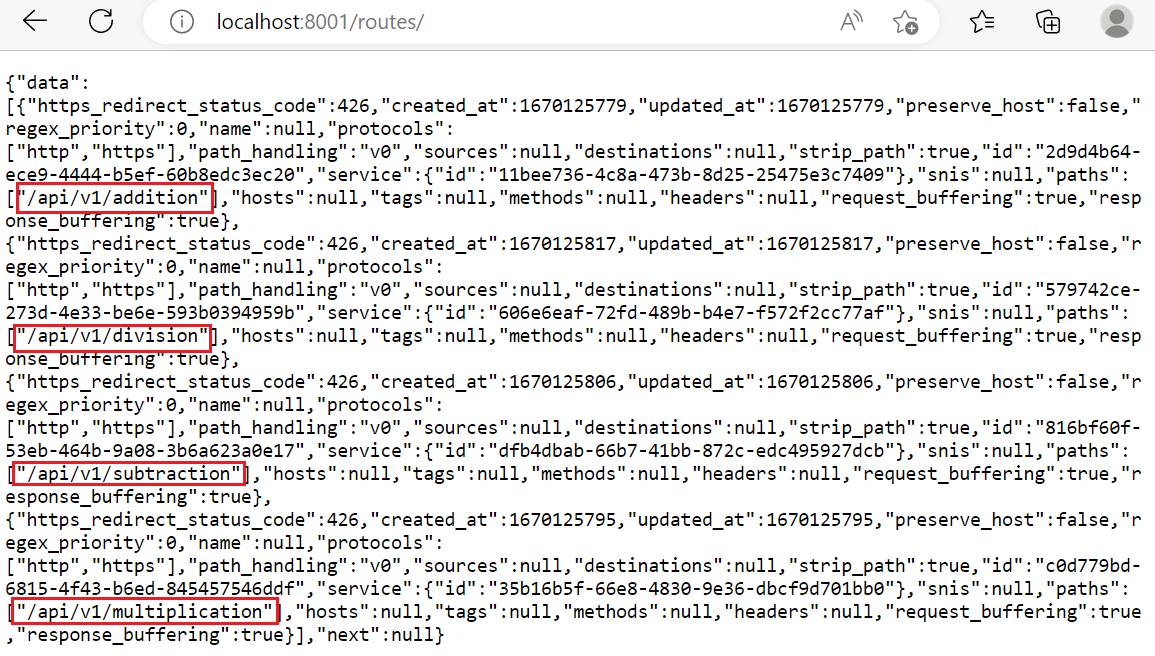
1. Membuat route **/api/v1/subtraction**

| curl -i -X POST --url http://localhost:8001/services/**subtraction\_nodejs**/routes --data 'paths[]=**/api/v1/subtraction**' |
| --- |

1. Membuat route **/api/v1/division**

| curl -X POST --url http://localhost:8001/services/**division\_php**/routes --data 'paths[]=**/api/v1/division**' |
| --- |

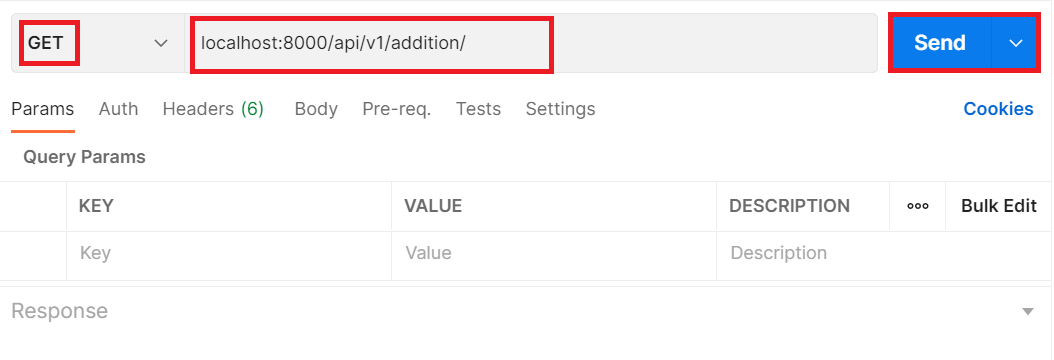
Untuk mengecek route yang sudah ada/dibuat dapat dengan mengetikkan URL ini di **browser**: **http://localhost:8001/routes/**



* Kong adalah sebuah API Gateway, membutuhkan permintaan/request dari klien dan mencarikan rute ke aplikasi upstream yang sesuai berdasarkan konfigurasi yang ada.
* Menggunakan service dan route yang sudah dibuat sebelumnya, Anda sekarang dapat mengakses kontainer **addition-go** yang mempunyai alamat IP **http://192.168.1.4** dengan menggunakan URL **http://localhost:8000/api/v1/addition**
* Secara default, Admin API pada Kong Gateway mendengarkan permintaan administratif pada port 8001 yang disebut sebagai **control plane**.
* Klien menggunakan port 8000 untuk membuat request dan hal ini disebut sebagai **data plane**.

**Tugas: Kerjakan bagian I dan J dibawah ini.**

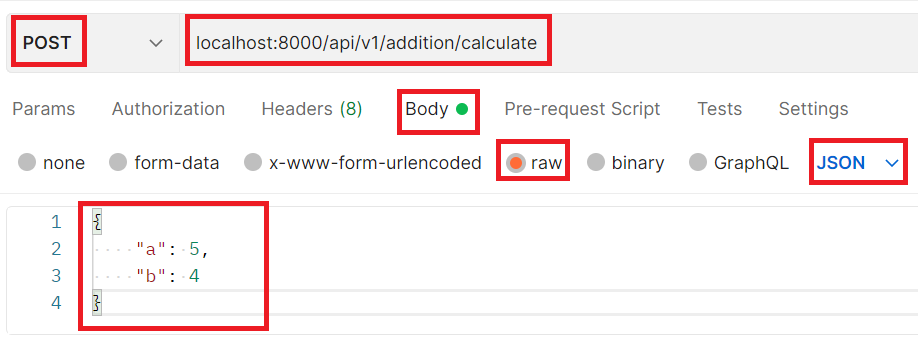
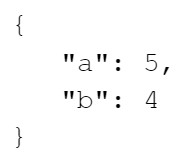
1. **Mengecek endpoint yang sudah dibuat menggunakan aplikasi Postman**
2. Jalankan aplikasi Postman
3. Mengecek endpoint **addition**
   1. Perhatikan gambar di bawah ini



* 1. Pada bagian method pilihlah **GET**
  2. Pada bagian URL silakan diisi dengan http://localhost:8000/api/v1/addition/
  3. Kemudian tekan tombol **SEND** lalu amati hasilnya.
  4. Letakkan hasil tangkapan layarnya di bawah ini

|  |
| --- |

* 1. Lakukan langkah yang sama dengan opsi seperti di bawah ini:
* Method: **POST**
* URL: **localhost:8000/api/v1/addition/calculate**
* Klik opsi **Body**, pilih opsi **raw**, opsi **JSON** dan masukkan isian seperti di bawah ini:

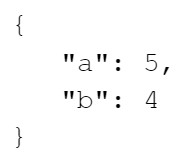


* **Catatan**: data yang dimasukkan harus **a** dan **b** dengan huruf kecil karena program yang dibuat menggunakan variabel **a** dan **b** secara statis (***untuk mempermudah***)
* Lalu tekan tombol **SEND** dan amati hasilnya
* Anda bisa mengganti nilai **a** dan **b** dengan angka lainnya
  1. Letakkan hasil tangkapan layarnya di bawah ini

|  |
| --- |

1. Mengecek endpoint **multiplication**
2. Lakukan langkah yang sama dengan opsi seperti di bawah ini:

* Method: **POST**
* URL: **localhost:8000/api/v1/multiplication/calculate**
* Klik opsi **Body**, pilih opsi **raw**, opsi **JSON** dan masukkan isian seperti di bawah ini:



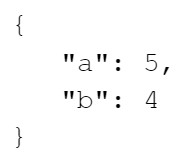
* **Catatan**: data yang dimasukkan harus **a** dan **b** dengan huruf kecil karena program yang dibuat menggunakan variabel **a** dan **b** secara statis (untuk mempermudah)
* Lalu tekan tombol SEND dan amati hasilnya
* Anda bisa mengganti nilai **a** dan **b** dengan angka lainnya

1. Letakkan hasil tangkapan layarnya di bawah ini

|  |
| --- |

1. Mengecek endpoint **subtraction**
2. Lakukan langkah yang sama dengan opsi seperti di bawah ini:

* Method: **POST**
* URL: **localhost:8000/api/v1/subtraction/calculate**
* Klik opsi **Body**, pilih opsi **raw**, opsi **JSON** dan masukkan isian seperti di bawah ini:



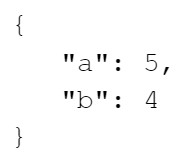
* **Catatan**: data yang dimasukkan harus **a** dan **b** dengan huruf kecil karena program yang dibuat menggunakan variabel **a** dan **b** secara statis (untuk mempermudah)
* Lalu tekan tombol SEND dan amati hasilnya
* Anda bisa mengganti nilai **a** dan **b** dengan angka lainnya

1. Letakkan hasil tangkapan layarnya di bawah ini

|  |
| --- |

1. Mengecek endpoint **division**
2. Lakukan langkah yang sama dengan opsi seperti di bawah ini:

* Method: **POST**
* URL: **localhost:8000/api/v1/division**
* Klik opsi **Body**, pilih opsi **raw**, opsi **JSON** dan masukkan isian seperti di bawah ini:



* **Catatan**: data yang dimasukkan harus **a** dan **b** dengan huruf kecil karena program yang dibuat menggunakan variabel **a** dan **b** secara statis (untuk mempermudah)
* Lalu tekan tombol SEND dan amati hasilnya
* Anda bisa mengganti nilai **a** dan **b** dengan angka lainnya

1. Letakkan hasil tangkapan layarnya di bawah ini

|  |
| --- |

1. **Kesimpulan**

Tulislah kesimpulan yang Anda dapatkan dari praktik kali ini!

| Dockers menyediakan kemampuan untuk mengemas dan menjalankan aplikasi dalam suatu lingkungan terisolasi yang disebut dengan container. |
| --- |