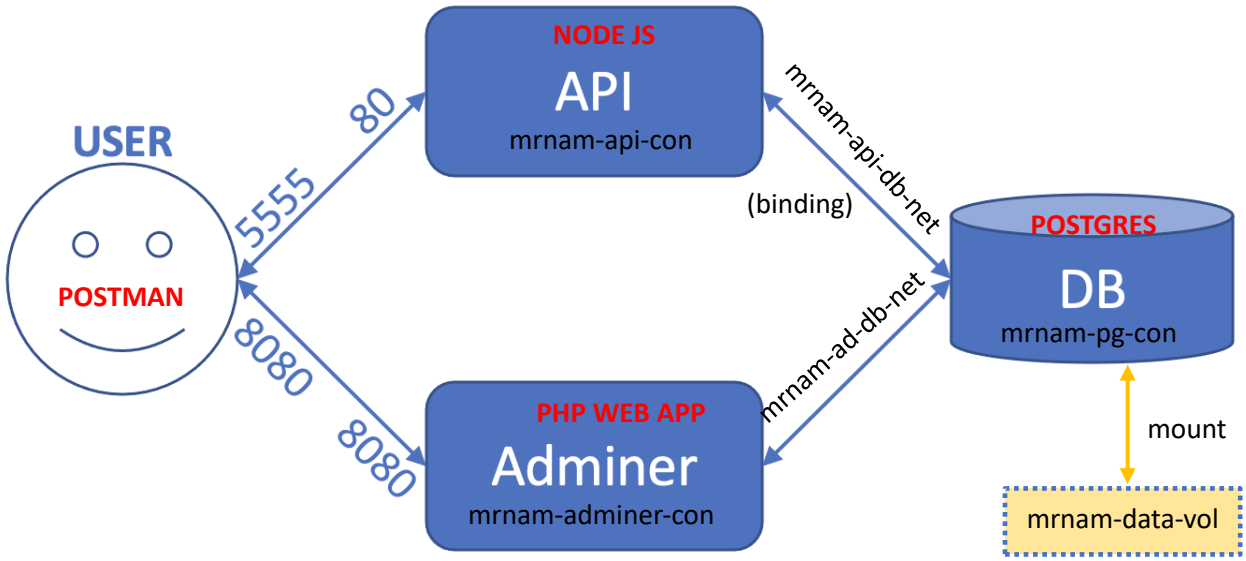
WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

THỰC HÀNH XÂY DỰNG ỨNG DỤNG DỊCH VỤ VỚI DOCKER

Mô hình



(Nếu giả lập trên máy ảo Virtual Box thì chạy Virtual Box dưới quyền Administrator)

Phần 1: Chuẩn bị hệ thống

1. Tiến hành ssh để copy thư mục backend về máy chủ server /home/osboxes

```
C:\Windows\system32>scp -r C:\backend\ osboxes@192.168.1.8:
The authenticity of host '192.168.1.8 (192.168.1.8)' can't be established.
ECDSA key fingerprint is SHA256:sI1xqq0R4zxeieoXVGVnhFtnss7/QeTdGgfk4ffMAKs.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '192.168.1.8' (ECDSA) to the list of known hosts.
osboxes@192.168.1.8's password:
.dockerignore                                 100%   30    10.0KB/s   00:00
data.js                                       100%  630   314.5KB/s   00:00
Dockerfile                                    100%  133    66.7KB/s   00:00
package-lock.json                             100%   53KB  13.3MB/s   00:00
package.json                                  100%  337   196.7KB/s   00:00
start.js                                      100%  687   480.3KB/s   00:00

C:\Windows\system32>ssh osboxes@192.168.1.8
osboxes@192.168.1.8's password:
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-14-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat Jan 27 06:59:38 AM UTC 2024

  System load:  0.0               Processes:             106
  Usage of /:   7.6% of 97.87GB   Users logged in:       1
  Memory usage: 7%                IPv4 address for enp0s3: 192.168.1.8
  Swap usage:   0%


49 updates can be applied immediately.
To see these additional updates run: apt list --upgradable


Last login: Sat Jan 27 06:47:05 2024
osboxes@osboxes:~$ ls /
bin  boot  cdrom  dev  etc  home  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  snap  srv  swap.img  sys  tmp  usr  var
osboxes@osboxes:~$ ls /home/osboxes
backend  flask_app
osboxes@osboxes:~$
```

2. Tiến hành copy thư mục database sang máy chủ ảo
   scp -r C:\database\ osboxes@192.168.1.9:

WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

```
C:\Windows\system32>scp -r C:\database\ osboxes@192.168.1.9:
osboxes@192.168.1.9's password:
init-db.sql                                                   100%  120    40.1KB/s   00:00

C:\Windows\system32>ssh osboxes@192.168.1.9
osboxes@192.168.1.9's password:
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-14-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sun Jan 28 03:30:37 AM UTC 2024

  System load:  0.01              Processes:              109
  Usage of /:   7.9% of 97.87GB   Users logged in:        1
  Memory usage: 8%                IPv4 address for enp0s3: 192.168.1.9
  Swap usage:   0%


60 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


Last login: Sun Jan 28 03:07:56 2024 from 192.168.1.95
osboxes@osboxes:~$ ls
backend  database  flask_app
```

3. Tiến hành build image với tên mrnam-api-image

```
osboxes@osboxes:~$ ls
backend  flask_app
osboxes@osboxes:~$ cd backend
osboxes@osboxes:~/backend$ ls
data.js  Dockerfile  package.json  package-lock.json  start.js
osboxes@osboxes:~/backend$
```

sudo docker build -t mrnam-api-image .

```
osboxes@osboxes:~/backend$ sudo docker build -t mrnam-api-image .
[+] Building 2.0s (11/11) FINISHED                                                  docker:default
 => [internal] load build definition from Dockerfile                                         0.0s
 => => transferring dockerfile: 172B                                                         0.0s
 => [internal] load .dockerignore                                                            0.1s
 => => transferring context: 70B                                                             0.0s
 => [internal] load metadata for docker.io/library/node:14-alpine                            1.8s
 => [auth] library/node:pull token for registry-1.docker.io                                  0.0s
 => [1/5] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e606  0.0s
 => [internal] load build context                                                            0.0s
 => => transferring context: 189B                                                            0.0s
 => CACHED [2/5] WORKDIR /usr/src/app                                                        0.0s
 => CACHED [3/5] COPY package*.json ./                                                       0.0s
 => CACHED [4/5] RUN ["npm", "ci"]                                                           0.0s
 => CACHED [5/5] COPY . .                                                                    0.0s
 => exporting to image                                                                       0.0s
 => => exporting layers                                                                      0.0s
 => => writing image sha256:02a498f2f573fd481dd302862b21dcb96ba73d84ca011229c5d51e20cd9c3a73  0.0s
 => => naming to docker.io/library/mrnam-api-image                                           0.0s
osboxes@osboxes:~/backend$ sudo docker images
REPOSITORY        TAG       IMAGE ID       CREATED         SIZE
mrnam-api-image   latest    02a498f2f573   4 minutes ago   125MB
```

4. Tạo hai bridge network trong hệ thống docker là: mrnam-api-db-net và mrnam-ad-db-net
   sudo docker network create --driver bridge mrnam-api-db-net
   sudo docker network create --driver bridge mrnam-ad-db-net
   sudo docker network ls

WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

```
osboxes@osboxes:~$ sudo docker network create --driver bridge mrnam-api-db-net
ce97cf1bf4a9160401309311434680711b2bebb329124dac179fc6bd096cffb0
osboxes@osboxes:~$ sudo docker network create --driver bridge mrnam-ad-db-net
bd20bec3d96cb506e4bd4790901f198bf7b90cbb43320c49b0973724f2e66545
osboxes@osboxes:~$ sudo docker network ls
NETWORK ID     NAME              DRIVER    SCOPE
6c22e863c6ab   bridge            bridge    local
566473afbac0   host              host      local
bd20bec3d96c   mrnam-ad-db-net   bridge    local
ce97cf1bf4a9   mrnam-api-db-net  bridge    local
d2a45d575e86   none              null      local
```

5. Tạo volume mang tên mrnam-data-vol
   sudo docker volume create mrnam-data-vol
   sudo docker volume ls

```
osboxes@osboxes:~$ sudo docker volume create mrnam-data-vol
mrnam-data-vol
osboxes@osboxes:~$ sudo docker volume ls
DRIVER    VOLUME NAME
local     mrnam-data-vol
```

Phần 2: Chuẩn bị Database Postgres

1. Khởi tạo container (detached mode) mang tên mrnam-pg-con và gắn (mount) vào mrnam-data-vol
   sudo docker run --name mrnam-pg-con -p 5432:5432 -e POSTGRES_USER=admin -e POSTGRES_PASSWORD=admin -e POSTGRES_DB=dbshop -v mrnam-data-vol:/var/lib/postgresql/data -d postgres

```
osboxes@osboxes:~$ sudo docker run --name mrnam-pg-con -p 5432:5432 -e POSTGRES_USER=admin -e POSTGRES_PASSWORD
=admin -e POSTGRES_DB=dbbooks -v mrnam-data-vol:/var/lib/postgresql/data -d postgres
30d22bf8291fed3deb3591b209398b7fd511994f71f8222e102c6acb8fc66661
osboxes@osboxes:~$ sudo docker volume ls
DRIVER    VOLUME NAME
local     mrnam-data-vol
osboxes@osboxes:~$ sudo docker container ls
CONTAINER ID   IMAGE      COMMAND                  CREATED              STATUS              PORTS
                     NAMES30d22bf8291f   postgres   "docker-entrypoint.s…"   About a minute ago   Up About
a minute   0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   mrnam-pg-con
```

2. Kiểm tra xem container đã chạy và kiểm tra truy cập thành công vào postgres database với tài khoản admin

   sudo docker exec -it mrnam-pg-con psql -U admin -d dbshop

```
osboxes@osboxes:~$ sudo docker exec -it mrnam-pg-con psql -U admin -d dbshop
psql (16.1 (Debian 16.1-1.pgdg120+1))
Type "help" for help.

dbshop=# \q
osboxes@osboxes:~$
```

WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

3. Copy file init-db.sql trong thư mục database vào thư mục docker-entrypoint-initdb.d trong container mrnam-pg-con

   sudo docker cp /home/osboxes/database/init-db.sql mrnam-pg-con:/docker-entrypoint-initdb.d/init-db.sql

```
osboxes@osboxes:~$ sudo docker cp /home/osboxes/database/init-db.sql mrnam-pg-con:/docker-entrypoint-initdb.d/init-db.sql
Successfully copied 2.05kB to mrnam-pg-con:/docker-entrypoint-initdb.d/init-db.sql
```

4. Sử dụng tài khoản root của container để khởi chạy script init-db.sql để tạo bảng books và nạp dữ liệu cho bảng tblbook trong CSDL dbshop

   sudo docker exec -u root mrnam-pg-con psql dbshop admin -f docker-entrypoint-initdb.d/init-db.sql

   (Cú pháp: docker exec -u <container_user> containername psql <dbname> <postgres_user> -f /container/path/file.sql)

```
osboxes@osboxes:~$ sudo docker exec -u root mrnam-pg-con psql dbshop admin -f docker-entrypoint-initdb.d/init-db.sql
CREATE TABLE
INSERT 0 5
```

5. Kiểm tra xem dữ liệu đã được tạo thành công

   sudo docker exec -it mrnam-pg-con psql -U admin -d dbshop

```
osboxes@osboxes:~$ sudo docker exec -it mrnam-pg-con psql -U admin -d dbshop
psql (16.1 (Debian 16.1-1.pgdg120+1))
Type "help" for help.

dbshop=# select * from tblbook;
 id |       title       | author
----+-------------------+---------
  1 | DevOps            | MR. NAM
  2 | Big Data          | MR. NAM
  3 | Cloud Deployement | MR. NAM
  4 | Data Analysis     | MR. NAM
  5 | Block Chain       | MR. NAM
(5 rows)

dbshop=# \q
```

6. Thiết lập mạng cho container mrnam-pg-con vào network mrnam-api-db-net. Sau đó, kiểm tra xem network đã config thành công

   sudo docker network connect mrnam-api-db-net mrnam-pg-con
   sudo docker network inspect mrnam-api-db-net

WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

```
osboxes@osboxes:~$ sudo docker network connect mrnam-api-db-net mrnam-pg-con
osboxes@osboxes:~$ sudo docker network inspect mrnam-api-db-net
[
    {
        "Name": "mrnam-api-db-net",
        "Id": "ce97cf1bf4a9160401309311434680711b2bebb329124dac179fc6bd096cffb0",
        "Created": "2024-02-16T04:07:56.369224985Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "26e185d12ea62e092bfff6315d47669a402dcae67f7378bf4285b53336c4b13d": {
                "Name": "mrnam-pg-con",
                "EndpointID": "8e251c82da8265796994837394058d25e121d648195c0bcc93cfb77e5ac5a23",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

Phần 3: Container mrnam-api-con

1. Khởi chạy container mrnam-api-con dưới ứng dụng nền (detached) với cấu hình biến môi trường (dùng -e)

   sudo docker run --name mrnam-api-con -p 5555:80 -e PGUSER='admin' -e PGPASSWORD='admin' -e PGDATABASE='dbshop' -e PGHOST='mrnam-pg-con' -e PGPORT=5432 -d mrnam-api-image

```
osboxes@osboxes:~$ sudo docker run --name mrnam-api-con -e PGUSER='admin' -e PGPASSWORD='admin' -e
PGDATABASE='dbshop' -e PGHOST='mrnam-api-db-net' -e PGPORT=5432 -d mrnam-api-image
[sudo] password for osboxes:
4084730de977c765300858ac78e958bd0f30f6ec84a01e1085caad194c8296e3
```

2. Kiểm tra danh sách container đang chạy
   sudo docker container ls -a

WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

```
osboxes@osboxes:~$ sudo docker container ls
CONTAINER ID   IMAGE             COMMAND                  CREATED         STATUS         PORTS                                       NAMES
4084730de977   mrnam-api-image   "docker-entrypoint.s…"   8 minutes ago   Up 8 minutes   80/tcp                                      mrnam-api-con
26e185d12ea6   postgres          "docker-entrypoint.s…"   2 hours ago     Up 2 hours     0.0.0.0:5432->5432/tcp, :::5432->5432/tcp   mrnam-pg-con
```

3. Thiết lập mạng cho container mrnam-api-con vào network mrnam-api-db-net

   sudo docker network connect mrnam-api-db-net mrnam-api-con

```
osboxes@osboxes:~$ sudo docker network connect mrnam-api-db-net mrnam-api-con
```

4. Kiểm tra xem network đã config thành công

   sudo docker network inspect mrnam-api-db-net

```
"Containers": {
    "7e7215d3a521ce657bc3db196408f4e442fddc4ec93d801b6295d1de2ab4fff4": {
        "Name": "mrnam-api-con",
        "EndpointID": "2cd1ecf3b8d81753f5428f15f13b8fe5044be06aa5df9987f8ff5eaca2d507af",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
    },
    "c5df89ad88694d94e99d5cdb46766ed4283d95d6e04068b8adbfe7e2c8b1ba17": {
        "Name": "mrnam-pg-con",
        "EndpointID": "96c41ac38f1391b168c561d2e3f6baca7944a9e658279b6dd315a6ce65904fd9",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
    }
},
```

Phần 4: Test thử dịch vụ backend sử dụng curl tại máy ảo server

1. Tiến hành thực hiện request với phương thức HTTP GET:
   (tham khảo: https://linuxize.com/post/curl-rest-api/)

   sudo curl http://localhost:5555/api/books

```
osboxes@osboxes:~$ sudo curl http://localhost:5555/api/books
[{"id":1,"title":"DevOps","author":"MR. NAM"},{"id":2,"title":"Big Data","author":"MR. NAM"},{"id":3,"title":"Cloud Deployement","author":"MR. NAM"},{"id":4,"title":"Da
ta Analysis","author":"MR. NAM"},{"id":5,"title":"Block Chain","author":"MR. NAM"}]osboxes@osboxes:~$
```

   (Ctrl Z) hoặc (Ctrl C) để thoát

2. Hướng dẫn thao tác lõi container để check log của container:
   container: sudo docker exec -it mrnam-api-con sh

```
osboxes@osboxes:~$ sudo docker exec -it  mrnam-api-con sh
/usr/src/app #
```

   Kiểm tra xem port 80 đang chạy node: netstat -tulpn | grep :80

```
/usr/src/app # netstat -tulpn | grep :80
tcp        0      0 :::80                   :::*                    LISTEN      18/node
```

(Gõ exit để thoát)

3. Xem log của container: sudo docker container logs mrnam-api-con

```
osboxes@osboxes:~$ sudo docker container logs mrnam-api-con

> practic@1.0.0 start /usr/src/app
> node start.js

Server started listening on port 80
Getting all books
Getting all books
```

4. Tiến hành thêm dữ liệu mới với phương thức HTTP POST
   curl -X POST -H "Content-Type: application/json" -d '{"title": "Docker Lesson","author":"MR. Huynh Nam", "body": "Post body."}' http://localhost:5555/api/books

```
osboxes@osboxes:~$ curl -X POST -H "Content-Type: application/json" -d '{"title": "Docker Lesson","author":"MR. Huynh Nam", "body": "Post body."}' http://localhost:5555
/api/books
{"id":10}osboxes@osboxes:~$
```

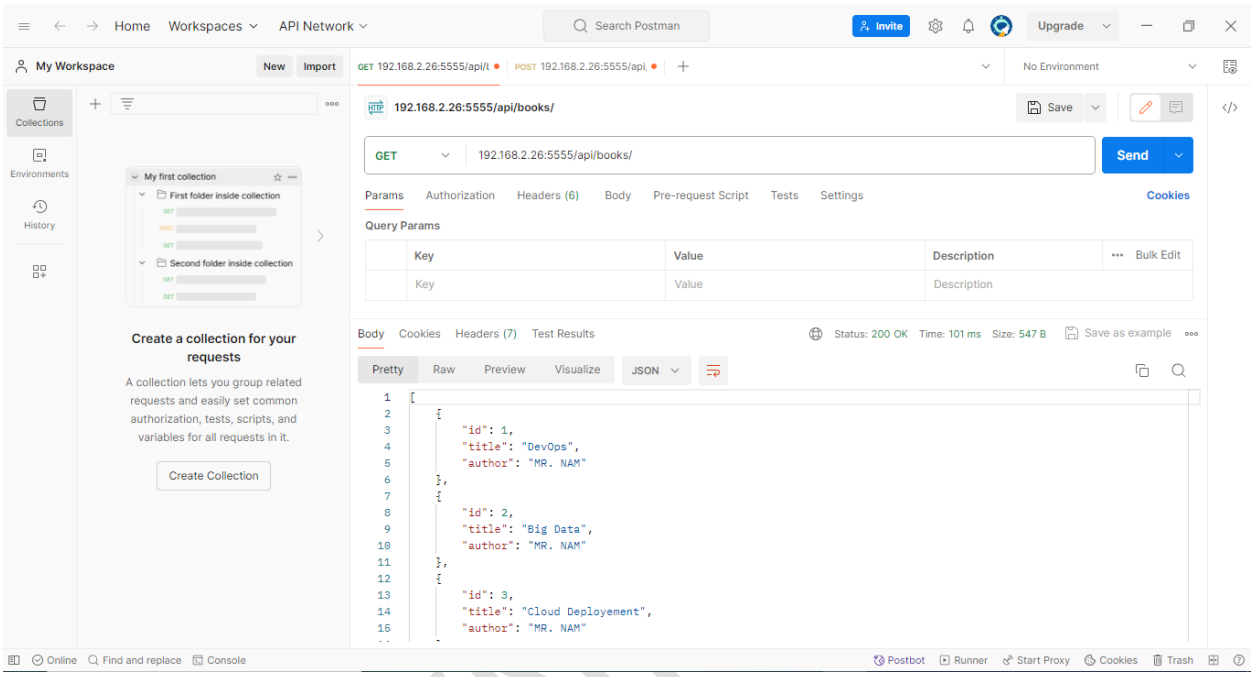
5. Kiểm tra lại dữ liệu mới với HTTP GET

```
osboxes@osboxes:~$ sudo curl http://localhost:5555/api/books
[{"id":1,"title":"DevOps","author":"MR. NAM"},{"id":2,"title":"Big Data","author":"MR. NAM"},{"id":3,"title":"Cloud Deployeme
nt","author":"MR. NAM"},{"id":4,"title":"Data Analysis","author":"MR. NAM"},{"id":5,"title":"Block Chain","author":"MR. NAM"}
osboxes@osboxes:~$
```

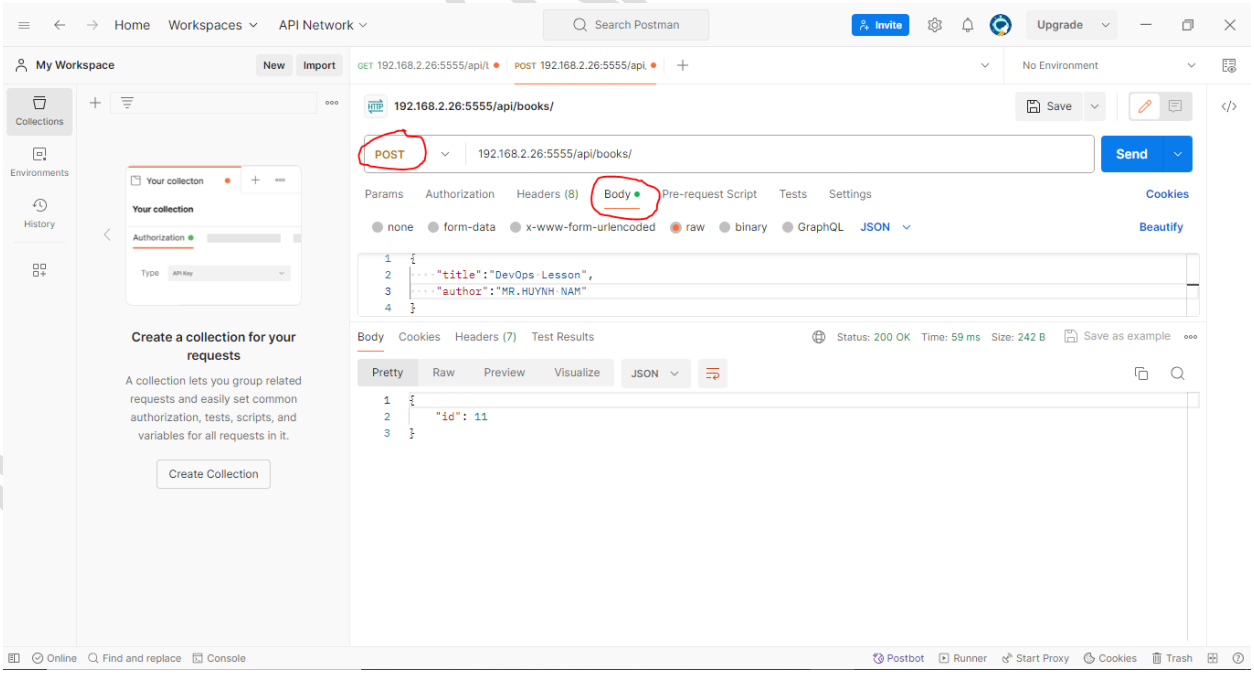
WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

Phần 6: Sử dụng POST MAN để kiểm tra dịch vụ (máy thật)

1. Tiến hành kiểm tra thử dịch vụ HTTP GET



2. Tiến hành kiểm tra thử dịch vụ HTTP POST



WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com

Phần 7: Quản trị CSDL với Adminer

Sinh viên tự tải image Adminer về và khởi chạy container với tên mrnam-adminer-con, cấu hình theo mô hình trên

1. Thêm mrnam-pg-con vào đường mạng mrnam-ad-db-net
   sudo docker network connect mrnam-ad-db-net mrnam-pg-con

```
osboxes@osboxes:~$ sudo docker network connect mrnam-ad-db-net mrnam-pg-con
```

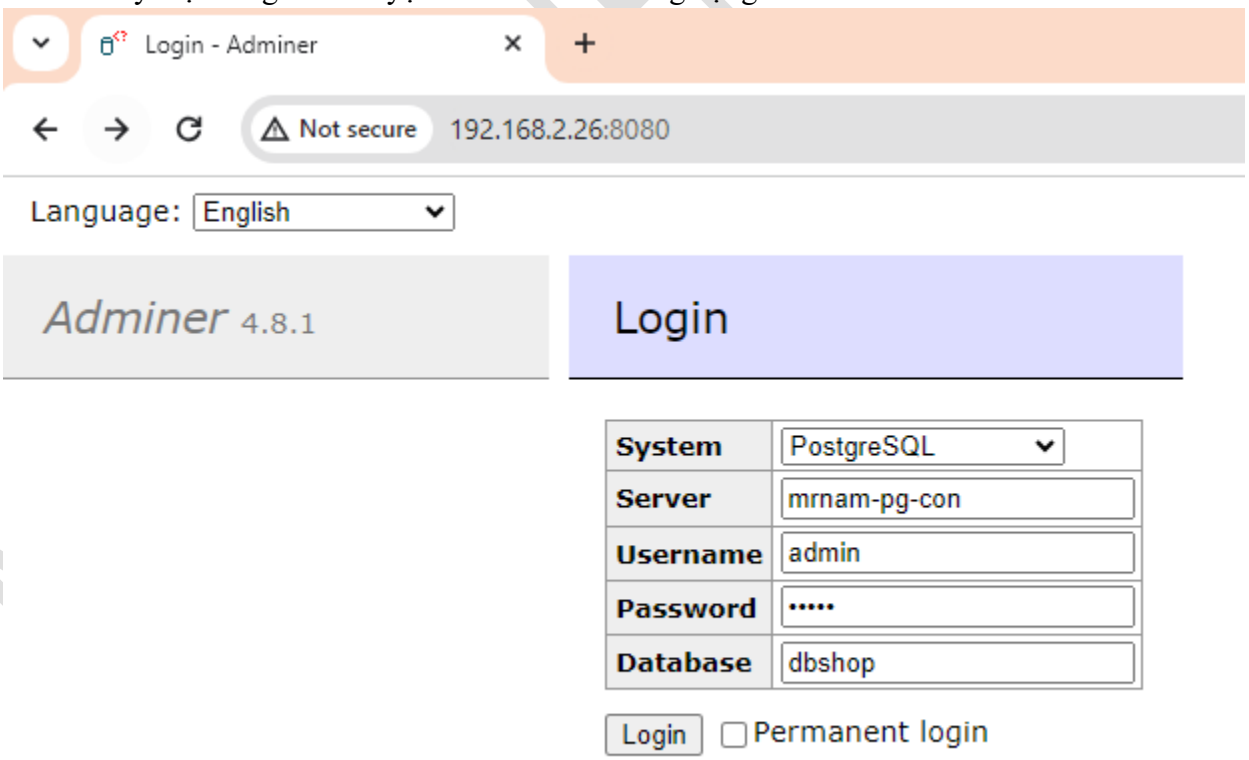
2. Khởi chạy container mrnam-adminer-con với cấu hình
   sudo docker run --name mrnam-adminer-con -p 8080:8080 -e PGUSER='admin' -e PGPASSWORD='admin' -e PGDATABASE='dbshop' -e PGHOST='mrnam-pg-con' -e PGPORT=5432 -d adminer

```
osboxes@osboxes:~$ sudo docker run --name mrnam-adminer-con -p 8080:8080 -e PGUSER='admin' -e PGPASSWORD='admin' -e PGDATABASE='dbshop' -e PGHOST='mrnam-pg-con' -e PGPORT=5432 -d adminer
ac42b79b27d4151da06830346424ae55f5f86976e581908c998bb0f679686db4
```

3. Thêm container mrnam-adminer-con và mạng mrnam-ad-db-net
   sudo docker network connect mrnam-ad-db-net mrnam-adminer-con

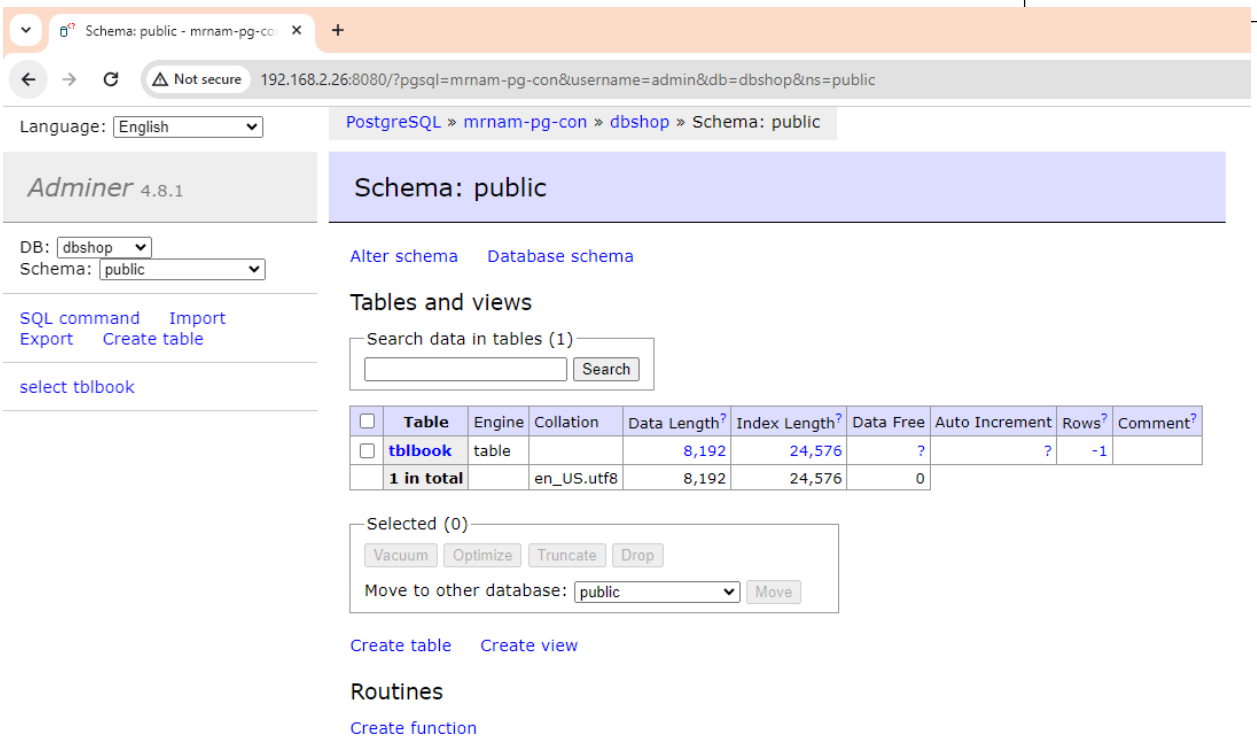
```
osboxes@osboxes:~$ sudo docker network connect mrnam-ad-db-net mrnam-adminer-con
```

4. Trở về máy thật dùng trình duyệt để kiểm tra thử ứng dụng adminer



5. Đăng nhập với tài khoản admin và mật khẩu admin

WRITTEN BY MR. HUYNH NAM
www.flex.edu.vn
Email: giangdayit@gmail.com



Phần 8: RESET DOCKER (để chuyển qua học Docker Compose)

sudo docker stop mrnam-api-con

sudo docker stop mrnam-adminer-con

sudo docker stop mrnam-pg-con

sudo docker container ls -a

sudo docker system prune

```
osboxes@osboxes:~$ sudo docker system prune
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all dangling images
  - all dangling build cache

Are you sure you want to continue? [y/N] y
```

sudo docker network prune

sudo docker volume prune