1. Kiểm tra xem hệ thống docker engine vận hàng thành công với ví dụ mẫu hello world container

Lệnh : **sudo docker container run hello-world**

Run thành công sẽ hiển thị dòng hello world

A screenshot of a computer

Description automatically generated

2. Thao tác với container Alpine. Trước tiên ta sẽ tải image cỉa Alpine trên docker Registry

Lệnh : **sudo docker image pull alpine**

A screenshot of a computer

Description automatically generated

3. Kiểm tra tết acr những image trong hệ thống docker hiện tại đang có

Lệnh : **sudo docker images**

A black screen with white text

Description automatically generated

4. Khởi chạy và xem các docker container và thư viện bên trong alpine

Lệnh : **sudo docker container run alpine ls -l**

A screenshot of a computer

Description automatically generated

5. Thâm nhập vào lõi alpine để tương tác lên container alpine (để thoát ra gõ exit )

Lệnh : **sudo docker run -it alpine**

A screenshot of a computer screen

Description automatically generated

Sử dụng các lệnh linux shell để tương tác với lõi container : ls -l,

6. Đưa container vào tiến trình chạy ngầm

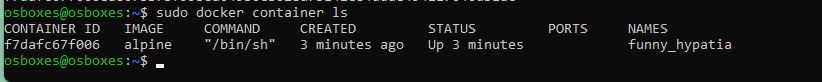
Lệnh : **sudo docker run -d -it alpine**

A screen shot of a computer

Description automatically generated

7. Xem ác container đang chạy :

Lệnh : **sudo docker container ls**



8. Truy cập container

Lệnh : **sudo docker attach [mã contaner id]**

A black screen with white text

Description automatically generated

9. Dừng container đnag chạy

Lệnh : **sudo docker stop [mã docker container]**

A screen shot of a computer

Description automatically generated

10. Xem danh sách các container đang chạy

Lệnh : **sudo docker container ls**



11. Xóa container

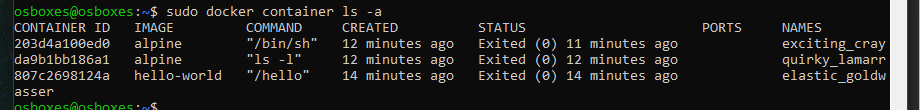
Lệnh : **sudo docker rm [mã id container]**

A black screen with white text

Description automatically generated

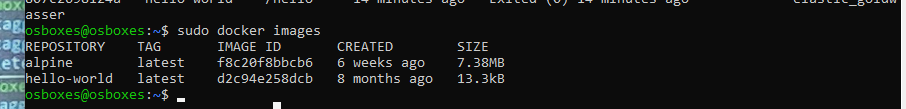
12. Xem danh sách các container hiện có

Lệnh : **sudo docker container ls -a**



13. Xem danh sách các images

Lệnh : **sudo docker images**



14. Xóa image

Lệnh : **sudo docker image rm -f [image ID]**

A screen shot of a computer

Description automatically generated

1. Coppy thư mục flask\_app vào máy ảo :

Lưu vào ổ C:\ , Mở một cmd khác để chạy lệnh coppy

Lệnh : scp -r C:\flask\_app\ [osboxes@192.168.xx.xx](mailto:osboxes@192.168.xx.xx):

A screen shot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

2. Chuển thư mục hiện hành về trong thư mục flask\_app trên máy ảo :

Lệnh : **cd flask\_app**

A black screen with white text

Description automatically generated

3. Chỉnh sử file Dockerfile thành :

Lệnh : **sudo nano Dockerfile**

Sửa thêm break system

A close up of a screen

Description automatically generated

A screenshot of a computer

Description automatically generated

4. Chỉnh sửa file requirement.txt :

Lệnh : sudo nano requirement.txt

A screenshot of a computer program

Description automatically generated

Sửa lại :

Flask==2.0.3

Werkzeug==2.2.2

A screenshot of a computer

Description automatically generated

6. Tiến hành buil image docker cho flask\_app

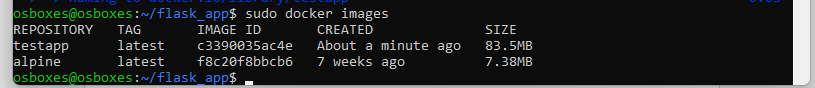
Lệnh : sudo docker build -t testapp .

A computer screen shot of a program

Description automatically generated

7. Kiểm tra lại xem image đã được tạo thành công hay chưa :

Lệnh : sudo docker images



8. Kiểm tra rõ hơn thông tin về image đã được tạo

Lệnh : sudo docker image inspect testapp

A screenshot of a computer screen

Description automatically generated

9. Chạy ứng dụng trên image trong docker :

Lệnh : sudo docker container run -p 8888:5000 testapp

A screen shot of a computer

Description automatically generated

10. Đăng nhập vào dockerhub

Lệnh : sudo docker login

A computer screen with white text

Description automatically generated

11. Tạo tag đánh dấu image đã tạo lên dockerhub registry

Lệnh : sudo docker tag testapp nhatlinhdev201/ismartio:example



12. Push image đã tạo lên dockerhub

Lệnh : sudo docker push nhatlinhdev201/ismartio:example

A screen shot of a computer

Description automatically generated

Cài git clien vào máy ảo, clone từ git về ứng dụng đó.

Docker giữa kỳ :

**Phần 1 : Chuẩn bị hệ thống**

1. Kết nối với máy ảo :

Lệnh : ssh [osboxes@192.168.1.13](mailto:osboxes@192.168.1.13)

A computer screen shot of a black screen

Description automatically generated

1. Copy thư mục backend và database vào máy ảo

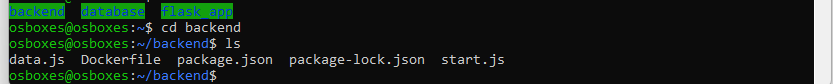
Lệnh : scp -r C:\datacnm\database\ osboxes@192.168.1.13:

A screenshot of a computer

Description automatically generated



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

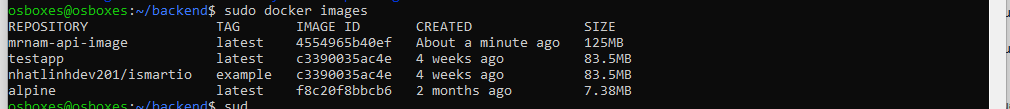


Lệnh : sudo docker build -t mrnam-api-image .

A computer screen shot of a computer screen

Description automatically generated

Xem lại các image hiện đã build : sudo docker images



Xóa đi các image không cần : sudo docker rmi testapp

Xóa các container không cần : sudo doker rm <id>

Xem các container đang chạy : sudo docker ps -a

A computer screen shot of a black screen

Description automatically generated

1. Tạo hai bridge network trong hệ thống docker là : mrnam-api-db-net và mrnam-ad-db-net

Lệnh :

sudo docker network create --driver bridge mrnam-api-db-net

sudo docker network create --driver bridge mrnam-ad-db-net

sudo docker network ls

A computer screen shot of a program

Description automatically generated

1. Tạo volume tên mrnam-data-vol

Lệnh :

sudo docker volume create mrnam-data-vol

Sudo docker volume ls

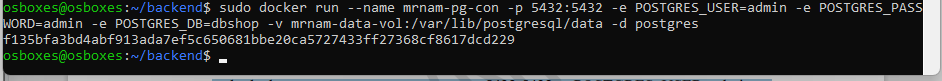
A screen shot of a computer

Description automatically generated

Phần 2 : Chuẩn bi database postgres

1. Khởi tạo container (detached mode) tên mrnam-pg-con và gắn (mount) vào mrnam-data-vol

Lệnh :

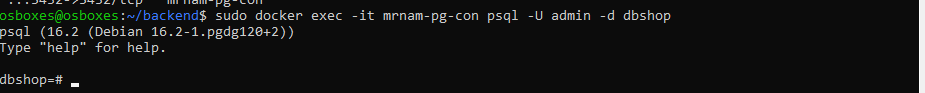
sudo docker run --name mrnam-pg-con -p 5432:5432 -e POSTGRES\_USER=admin -e POSTGRES\_PASSWORD=admin -e POSTGRES\_DB=dbshop -v mrnam-data-vol:/var/lib/postgresql/data -d postgres

A screen shot of a computer

Description automatically generated

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

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



Gõ \q enter để thoát

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

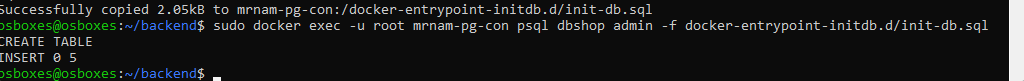
Lệnh : sudo docker cp /home/osboxes/database/init-db.sql mrnam-pg-con:/docker-entrypoint-initdb.d/init-db.sql

A screen shot of a computer

Description automatically generated

1. 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



1. Kiểm tra xem dữu liệu đã được tạo thành công chưa

sudo docker exec -it mrnam-pg-con psql -U admin -d dbshopA black screen with white text

Description automatically generated

1. 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

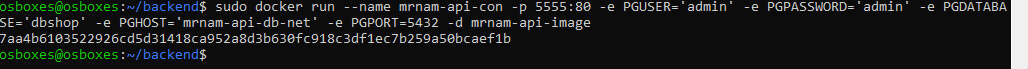
A computer screen shot of white and blue text

Description automatically generated

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



2. Kiểm tra danh sách container đang chạy

sudo docker network inspect mrnam-api-db-net

A computer screen shot of white text

Description automatically generated

3. Thiết lập mạng

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

Phần 4: Test thử dịch vụ backend

1. Cài dặt post man trên máy thệt
2. Tiến hành thực hiện request API GET POST vào dịch vụ server đang chạy

A screenshot of a computer

Description automatically generated

Docker compose

1. Xóa hết các container, giữ lại image

sudo docker container ls -a

sudo docker rm 5c2680f32a50

1. Xóa hết các network, trừ những cái mặc định của docker

sudo docker network ls

sudo docker network rm 7eadd953311b

1. Xóa hết các volume

sudo docker volume ls

sudo docker volume rm mrnam-data-vol

1. Prune lại hệ thống

sudo docker system prune

1. Cài đặt docker-compose

sudo apt install docker-compose

1. Kiểm tra version của docker-compose

sudo docker compose version

1. Coppy file docker compose sang máy ảo file-docker-compose-1.yml

scp -r C:\datacnm\file-docker-compose-1.yml [osboxes@192.168.1.13](mailto:osboxes@192.168.1.13):

1. Tiến hành compose hệ thống :

sudo docker-compose -f file-docker-compose.yml up

1. Bấm ctrl c để quay lại cmd nhưng sẽ tắt hết container đang chạy.

A screenshot of a computer

Description automatically generated

LAB 4 :

Chuẩn bị :

1. Xóa hết các container, giữ lại image

sudo docker container ls -a

sudo docker rm 5c2680f32a50

1. Xóa các image trùng tên

sudo docker rmi d1fbc0bb9669

1. Xóa hết các network, trừ những cái mặc định của docker

sudo docker network ls

sudo docker network rm 7eadd953311b

1. Xóa hết các volume

sudo docker volume ls

sudo docker volume rm mrnam-data-vol

1. Prune lại hệ thống

sudo docker system prune

Dô làm lab 4

1. Sửa lại code backend để sử dụng mysql lưu tên backend\_msql

* Thêm thư viện mysql vào package.json
* Sửa lại data

A screen shot of a computer program

Description automatically generated

1. Copy thư mục backend\_msql và database vào máy ảo

scp -r C:\datacnm\database\ [osboxes@192.168.1.10](mailto:osboxes@192.168.1.10):

1. Build image api-image cho nodejs server :