

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN

.....✧✧📖📖✧✧.....



HCMUTE

MÔN HỌC: ĐIỆN TOÁN ĐÁM MÂY

BÁO CÁO ĐỒ ÁN

**WEB QUẢN LÝ SINH VIÊN DÙNG KẾT NỐI
NHIỀU DOCKER VỚI NHAU**

GVHD: Huỳnh Xuân Phụng

SVTH:

MSSV:

Ngô Xuân Thắng

18110368

Nguyễn Khắc Thành Sang

18100281

Đoàn Mạnh Hiếu

18100353

Lớp thứ 6, tiết 13+14+15

Tp. Hồ Chí Minh, Tháng 12 năm 2020

PHÂN CÔNG NHIỆM VỤ

Link Trello: <https://trello.com/b/XLuBUKps/%C4%91i%E1%BB%87n-to%C3%A1n-%C4%91%C3%A1m-m%C3%A2y-k%E1%BA%BFt-h%E1%BB%A3p-nhi%E1%BB%81u-docker-v%E1%BB%9Bi-nhau>

Mục lục

Chương 1.	Giới thiệu đồ án và tìm hiểu về docker.....	1
1.1.	Giới thiệu đồ án.....	1
1.2.	Tìm hiểu về những nội dung của Docker sử dụng trong đồ án.....	1
1.2.1.	Docker là gì.....	1
1.2.2.	Docker image	1
1.2.3.	Docker container	2
1.2.4.	Dockerfile	3
1.2.5.	Docker compose	3
Chương 2.	Tiến hành xây dựng ứng dụng web quản lí sinh viên chạy bằng nhiều image docker	5
2.1.	Chuẩn bị project web	5
2.1.1.	Công nghệ sử dụng	5
2.1.2.	Chức năng	5
2.2.	Xây dựng dockerfile cho từng phần và push image lên Dockerhub.....	6
2.3.	Xây dựng file dockercompose kết nối các phần lại với nhau	8
2.4.	Hướng dẫn chia sẻ và sử dụng ứng dụng cho người khác	12
Chương 3.	Kết luận và tài liệu tham khảo	13
3.1.	Kết luận và kết quả đạt được.....	13
3.1.1.	Kết quả đạt được:.....	13
3.1.2.	Link demo:.....	13
3.2.	Tài liệu tham khảo.....	13

Chương 1. Giới thiệu đồ án và tìm hiểu về docker

1.1. Giới thiệu đồ án

Đồ án là ứng dụng web quản lý sinh viên đơn giản với các chức năng như xem danh sách sinh viên cũng như xem, sửa thông tin của sinh viên cũng như xóa khỏi danh sách. Và nó được kết hợp với việc sử dụng trên nền ảo hóa docker. Với việc chia làm ba phần frontend, backend và database và được xây dựng thành các container từ image và sau đó chạy cả một project thông qua một câu lệnh duy nhất phù hợp cho nhiều người có thể xem dù học không phải lập trình viên.

1.2. Tìm hiểu về những nội dung của Docker sử dụng trong đồ án

1.2.1. Docker là gì

Docker là một công cụ giúp cho việc tạo ra và triển khai các container để phát triển, chạy ứng dụng được dễ dàng. Các container là môi trường, mà ở đó lập trình viên đưa vào các thành phần cần thiết để ứng dụng của họ chạy được, bằng cách đóng gói ứng dụng cùng với container như vậy, nó đảm bảo ứng dụng chạy được và giống nhau ở các máy khác nhau (Linux, Windows, Desktop, Server ...)

Docker có vẻ rất giống máy ảo (nhiều người từng tạo máy ảo với công cụ ảo hóa như Virtual Box, VMWare), nhưng có điểm khác với VM: thay vì tạo ra toàn bộ hệ thống (dù ảo hóa), Docker lại cho phép ứng dụng sử dụng nhân của hệ điều hành đang chạy Docker để chạy ứng dụng bằng cách bổ sung thêm các thành phần còn thiếu cung cấp bởi container. Cách này làm tăng hiệu suất và giảm kích thước ứng dụng.

Ai dùng Docker? Docker mang lại lợi ích cho cả lập trình viên lẫn quản trị hệ thống, sử dụng Docker lập trình viên tập trung vào mà viết code chứ không lo lắng về việc triển khai, không lo lắng ở máy của lập trình viên chạy được, máy khác lại không chạy được ...

1.2.2. Docker image

Image là một gói phần mềm trong đó chứa những thứ cần như thư viện, các file cấu hình, biến môi trường để chạy mọi ứng dụng nào đó. Nó giống như cái USB chứa bộ cài đặt hệ điều hành Windows!

Chúng ta có thể sử dụng các Image được build sẵn phục vụ cho nhiều mục đích khác nhau thông qua những image của các nhà phát triển được chia sẻ. Và dockerhub là kho image được người dùng chia sẻ chúng ta hoàn toàn có thể build và tự chia sẻ image của chính mình cho những người dùng khác tương tự github.

1.2.3. Docker container

Khi một phiên bản của image chạy, phiên bản chạy đó gọi là container - (vậy muốn có container phải có image). Bất kỳ lúc nào bạn cũng có thể kiểm tra xem có bao nhiêu container đang chạy và nó sinh ra từ image nào.

Và sau khi thực hiện đối với trên container ta có thể lưu những thay đổi và tạo thành image mới khác hoàn toàn so với image đã được dùng tạo nên container đó.

Cấu trúc và một số lệnh cơ bản:

Đề tạo và chạy một container theo cú pháp:

```
docker run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

Các tham số đó là:

- [OPTIONS] các thiết lập khi tạo container, có rất nhiều thiết lập tùy mục đích tạo container, sẽ tìm hiểu các thiết lập qua từng trường hợp cụ thể.
- IMAGE tên image hoặc ID của image từ nó sinh ra container.
- [COMMAND] [ARG...] lệnh và tham số khi container chạy.

Ngoài ra còn có vài tham số khác:

- Tạo và chạy container, container tự xóa khi kết thúc thì thêm vào tham số --rm
- Tạo và chạy container - khi container chạy thì hành ngay một lệnh nào đó, ví dụ ls -la

```
docker run -it --rm debian ls -la
```

- Tạo và chạy container - ánh xạ một thư mục máy host vào một thư mục container, chia sẻ dữ liệu

```
docker run -it --rm -v path-in-host:path-in-container debian
```

- Vào container đang đang chạy

Kiểm tra xem bằng lệnh docker ps, nếu có một container với ID là containerid đang chạy, để quay quay trở lại terminal của nó dùng lệnh:

- docker container attach containerid
- Chạy một container đang dừng

```
docker container start -i containerid
```

- Nếu cần xóa bỏ hẳn một container thì dùng lệnh

```
docker container rm containerid
```

Và trong projectweb quản lí sinh viên các container đã được build thông qua file dockercompose

1.2.4. Dockerfile

Dockerfile là một file text, trong đó chứa các dòng chỉ thị để Docker đọc và chạy theo chỉ thị đó để cuối cùng bạn có một image mới theo nhu cầu. Khi đang có một Dockerfile giả sử có tên là Dockerfile để ra lệnh cho Docker chạy nó bạn có thể gõ:

```
docker build -t nameimage:version -f Dockerfile .
```

Dấu . ở cuối lệnh docker build ở trên, có nghĩa tìm file có tên Dockerfile ở thư mục hiện tại. Build và giải thích dockerfile.

-t nameimage:version là đặt tên và tag được gán cho image mới tạo ra.

Việc dùng dockerfile để build image giúp chúng ta tiết kiệm thời gian hơn trong việc xây dựng image nó tập hợp tất cả cái lệnh và thực hiện tuần tự một cách tự động giúp chúng ta không phải thực thi từng lệnh một rất tốn thời gian.

1.2.5. Docker compose

Xây dựng image mới từ iKhi phân phối các ứng hoàn thành, từng phần của ứng dụng đó ở dạng sản phẩm gọi nó là các service (dịch vụ), hãy nhớ lại ví dụ của phần trước bạn đã cài đặt một ứng dụng Wordpress, hãy xem nó để trang web đó chạy được bạn đã tạo ra những thành phần gì:

- Một container chạy HTTPD Apache là webserver.
- Một container chạy PHP để thi hành code các file PHP.
- Một container chạy MySQL làm CSDL cho trang web.

Những thành phần này khi phân phối ứng dụng chúng đều được gọi là các dịch vụ service. Như vậy các service thực chất là các container chạy đáp ứng chức năng thành phần tạo nên ứng dụng. Bạn cần hiểu sơ qua về khái niệm này vì ta sẽ sử dụng nó trong docker-compose giúp tạo ra các service trên một cách tự động.

Docker Compose với file docker-compose.yml:

File docker-compose.yml gần giống ý nghĩa với file Dockerfile đã tìm hiểu trong Sử dụng Dockerfile, là một file text, viết với định dạng YAML (Ain't Markup Language, đọc nhanh định dạng Định dạng YML) là cấu hình để từ đó lệnh docker compose sinh ra

và quản lý các service (container), các network, các ổ đĩa ... cho một ứng dụng hoàn chỉnh.

Một số lệnh thường dùng của dockercompose:

- docker-compose up: tạo và chạy các thành phần định nghĩa trong docker-compose.yml (các dịch vụ, image, container, mạng, đĩa ...).
- docker-compose down: dừng và xóa: image, container, mạng, đĩa tạo ra bởi docker-compose up
- docker-compose logs [SERVICES]: Theo dõi Logs từ các dịch vụ

Ngoài ra còn có các lệnh nhỏ khác như exec ps restart ...

Chương 2. Tiến hành xây dựng ứng dụng web quản lý sinh viên chạy bằng nhiều image docker

2.1. Chuẩn bị project web

2.1.1. Công nghệ sử dụng

Ứng dụng web quản lý sinh viên đơn giản được xây dựng nên từ 3 thành phần gồm:

- Frontend sử dụng Angular
- Backend sử dụng java Spring-boot
- Database sử dụng Mysql

2.1.2. Chức năng

Ứng dụng có những chức năng cơ bản như là xem danh sách sinh viên, thêm sinh viên, xóa khỏi danh sách cũng như là thay đổi thông tin của từng sinh viên.

Giao diện web:

Thêm sinh viên:

Danh sách sinh viên Thêm sinh viên Chỉnh sửa

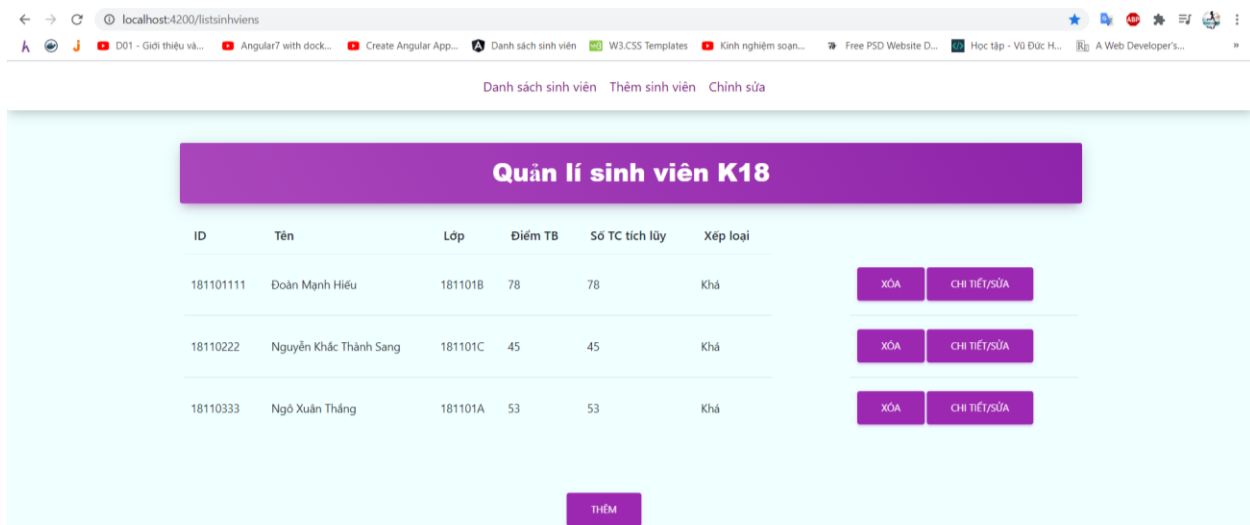
Quản lý sinh viên K18

Nhập thông tin sinh viên

ID	Họ và tên	Lớp
18110333	Ngô Xuân Thắng	181101A
Điểm trung bình	Số tin chỉ	Xếp loại
6.5	56	Khá
Thông tin chi tiết		
Bình Định		

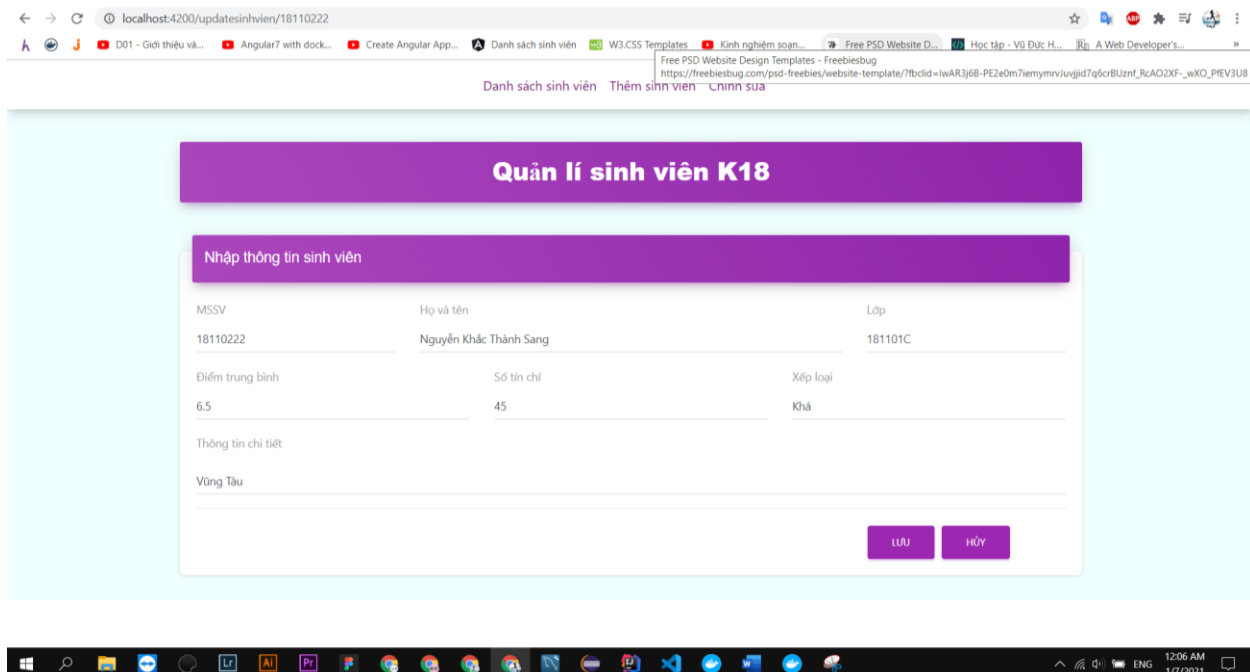
LƯU HỦY

Danh sách sinh viên:



Có các chức năng như xóa thêm sửa sinh viên.

Cập nhật thông tin sinh viên:



2.2. Xây dựng dockerfile cho từng phần và push image lên Dockerhub

Dockerfile frontend:

```

Dockerfile > ...
1 FROM node:latest as build-WebUI # Xây dựng image từ image node:latest
2 WORKDIR /app # Thiết lập thư mục làm việc
3 COPY package.json ./ # Copy file package.json vào thư mục app
4 RUN npm install # Cài đặt npm
5 RUN npm install -g @angular/cli # Cài đặt bộ công cụ của Angular
6 COPY . . # Sao chép cả thư mục webui vào trong
7 RUN npm run build --prod # Thực hiện build project
8
9
10 FROM amd64/nginx:1.18.0-alpine # Xây dựng web server
11 COPY --from=build-WebUI /app/dist/webui /usr/share/nginx/html # Copy các file
12 EXPOSE 4200 # Thiết lập cổng làm việc
13 |

```

Dockerfile backend:

```

Dockerfile > FROM
1 FROM openjdk:latest # Image sử dụng
2 ADD target/qlsv-mysql.jar qlsv-mysql.jar // thêm file qlsv-mysql # đã build trong folder target
3 #Cổng sử dụng
4 EXPOSE 8086 #Cấu hình container chạy file qlsv-mysq
5 ENTRYPOINT ["java", "-jar", "qlsv-mysql.jar"]


```

Database: ta dùng các image được build sẵn của nhà phát triển MySql nên ta không cần xây dựng dockerfile để build ra image ở đây sử dụng image 'mysql:latest'

Build dockerfile :

```
docker build -t i-firstserver:version1 -f Dockerfile .
```

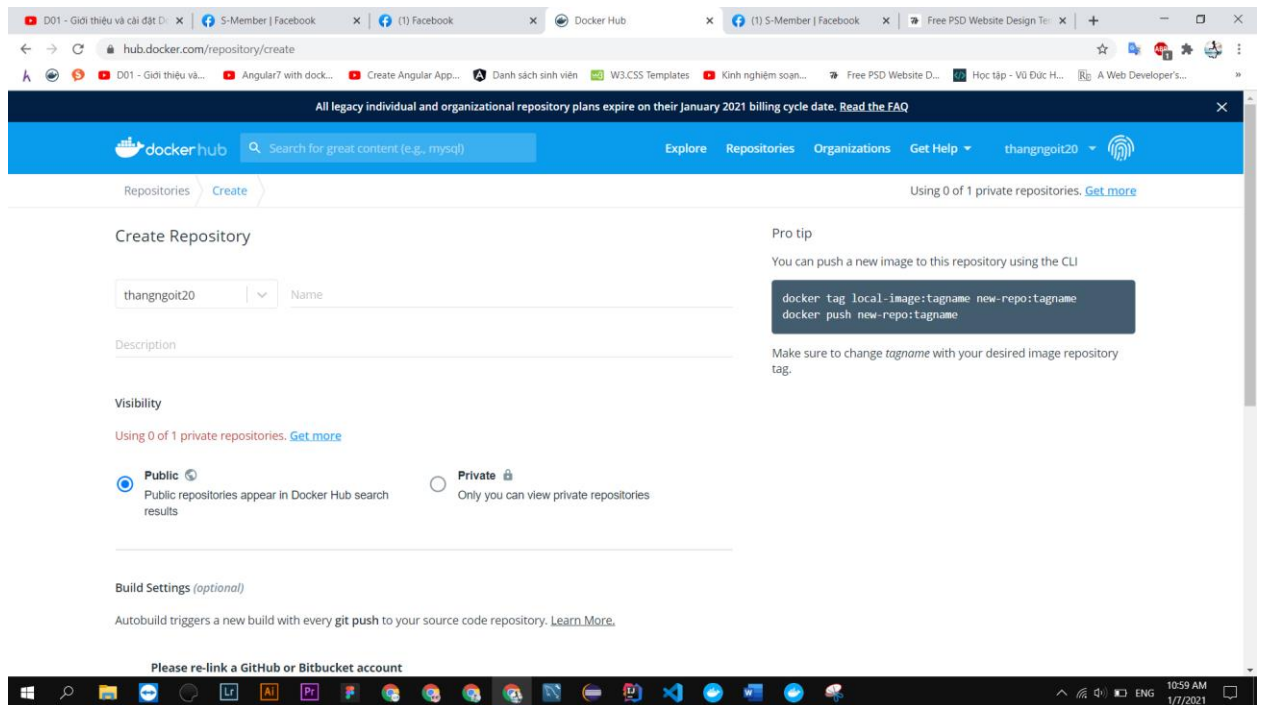
-Ở đây -i-firstserver:version1 là tên của image được xây dựng nên sau khi build thì Ta được 2 image như sau:

dmhieu/qlsv-frontend-ang...	IN USE	latest	41f27cbcdacb	about 14 hours ago	35.29 MB	 RUN ▶
dmhieu/qlsv-spring-api	IN USE	latest	25a8bb6e836e	about 19 hours ago	536.32 MB	

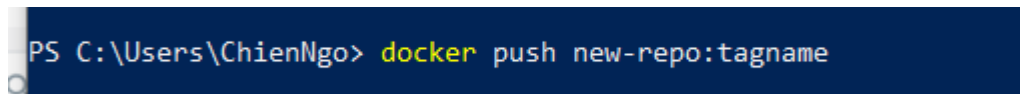
Push image lên dockerhub:

Việc đưa image lên dockerhub nhằm mục đích lưu trữ tái sử dụng và chia sẻ. Cách bước

B1: Tạo repository và phải trùng tên với image

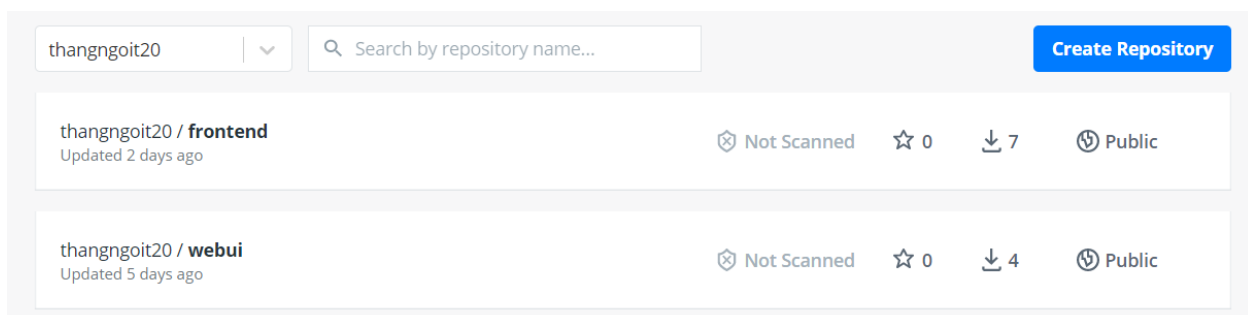


B2: Push image bằng lệnh



- New-repo là tên image
- Tagname là phiên bản

Kết quả:



2.3. Xây dựng file dockercompose kết nối các phần lại với nhau

Là file .yaml được xây dựng nhưng sau

```

docker-compose.yml X
docker-compose.yml
1  version: '3'                                # Chọn viết theo phiên bản 3 của dockercompose
2
3  services:                                    # Các dịch vụ (container) nằm trong services
4
5    db:                                         # (((1))) Bắt đầu tạo dịch vụ chạy Mysql
6      image: 'mysql:latest'                    # Dùng image mysql:latest để tạo container
7      environment:                             # Thêm các biến môi trường của database
8        - MYSQL_ROOT_PASSWORD=password        # Thiết lập password
9        - MYSQL_DATABASE=qlsv
10       - MYSQL_USER=sa
11       - MYSQL_PASSWORD=password
12      ports:
13        - 33061:3306                           # Ánh xạ cổng 33061 của máy host vào port 3306 của container
14
15    web:                                        # (((2))) Bắt đầu tạo dịch vụ chạy Spring-boot
16      image: 'dmhieu/qlsv-spring-api:latest'   # Dùng dmhieu/qlsv-spring-api:latest để tạo container
17      restart: on-failure                      # Chính sách khởi động
18      depends_on:                             # Chỉ ra sự phụ thuộc của services web với services db
19        - db                                   # Tức là services db phải chạy và tạo ra trước
20      ports:
21        - 8086:8086                             # Mở cổng 8086 public, ánh xạ vào 8086
22      environment:
23        - DATABASE_HOST=db
24        - DATABASE_USER=sa
25        - DATABASE_PASSWORD=password
26        - DATABASE_NAME=qlsv
27        - DATABASE_PORT=33061
28
29    ui:                                         # (((3))) Bắt đầu tạo dịch vụ chạy Angular
30      image: 'dmhieu/qlsv-frontend-angular:latest' # Dùng dmhieu/qlsv-frontend-angular:latest để tạo container
31      ports:
32        - 4200:80                               # Mở cổng 4200 public, ánh xạ vào 80

```

Đây là file dockercopmose được xây dựng lên từ những image đã được buil lên từ trước và đã được đẩy lên dockerhub. Nếu không sử dụng các image có sẵn chúng ta hoàn toàn có thể viết dockercompose dựa theo các dockerfile như sau:

```

30    ui:
31      image: fontend                            # (((3))) Bắt đầu tạo dịch vụ chạy Angular
32      build:                                    # Dùng image build từ dockerfile
33        dockerfile: Dockerfile
34      ports:
35        - 4200:80                               # Mở cổng 4200 public, ánh xạ vào 80
36

```

Sau khi xây dựng xong dockercompose.yml bây giờ ta tiếp hành chạy chương trình

B1: Mở cmd và duy chuyển và thư mục chứa file dockercompose:

```

PS C:\Users\ChienNgo> cd C:\Users\ChienNgo\Desktop\DTDM\

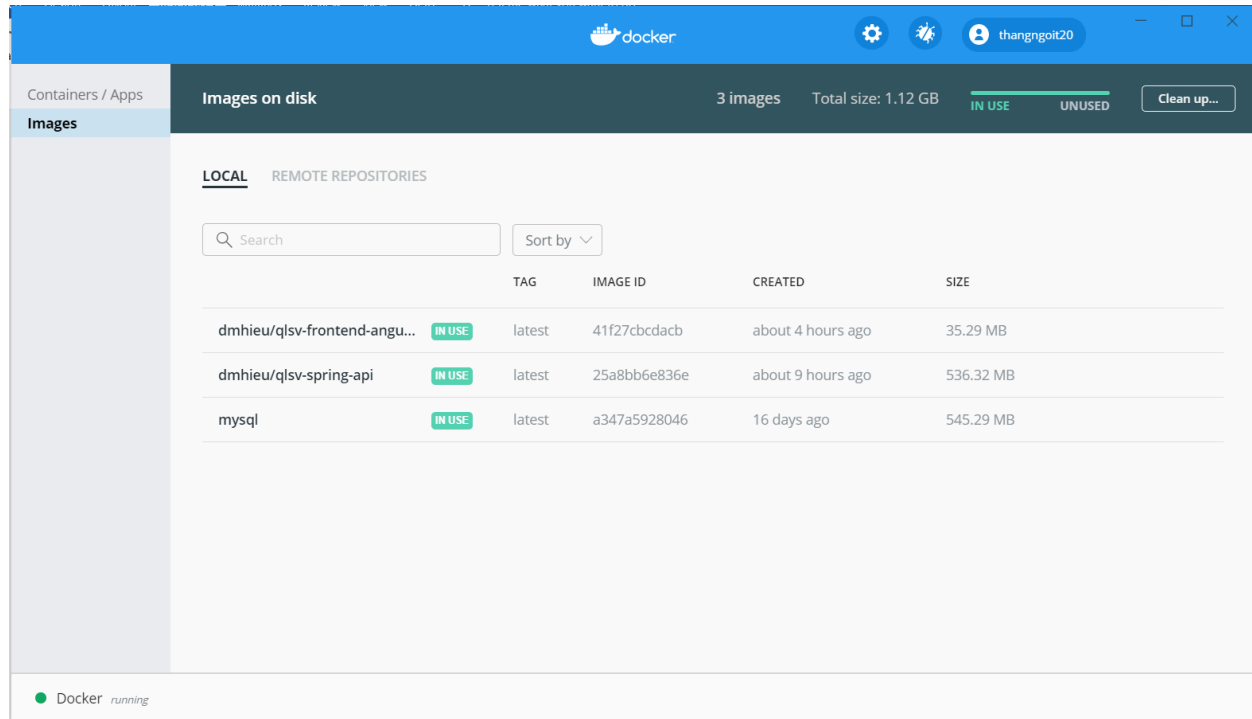
```

B2: Tiến hành chạy lệnh docker-compose up để tiến hành build file .yaml

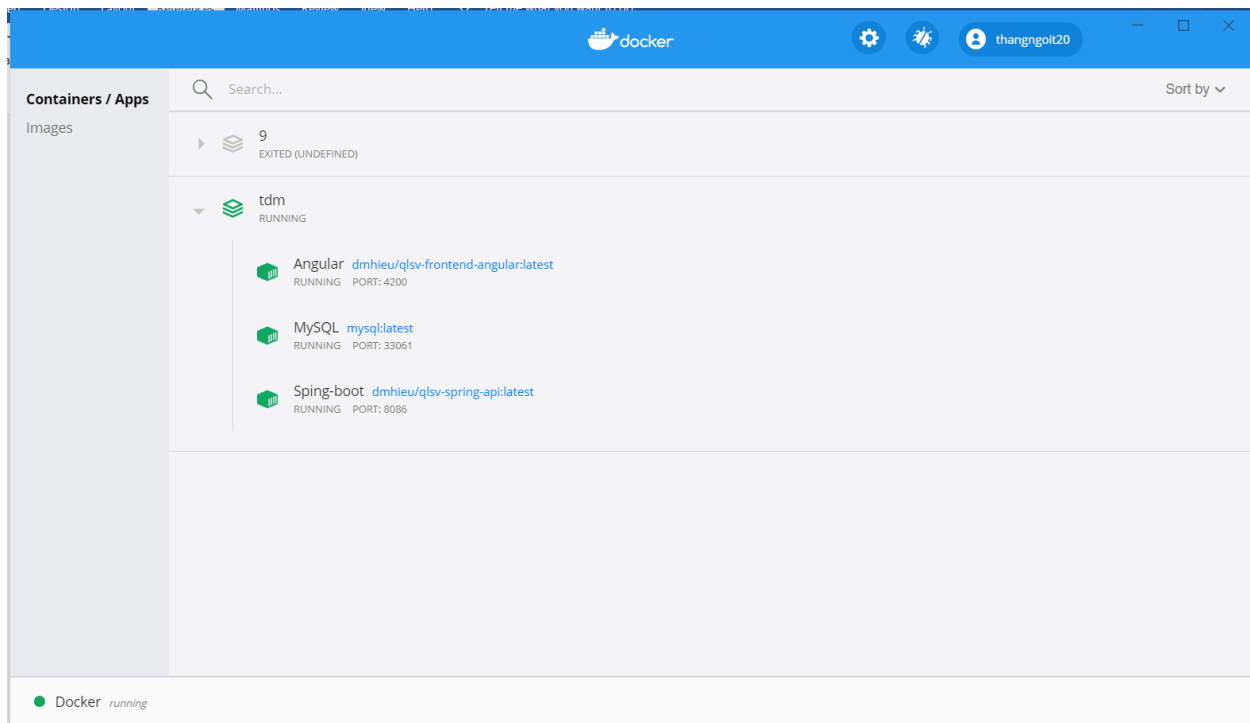
```
PS C:\Users\ChienNgo\Desktop\ĐTDM> docker-compose up
Creating network "tdm_default" with the default driver
```

Kết quả sau khi chạy xong ta được

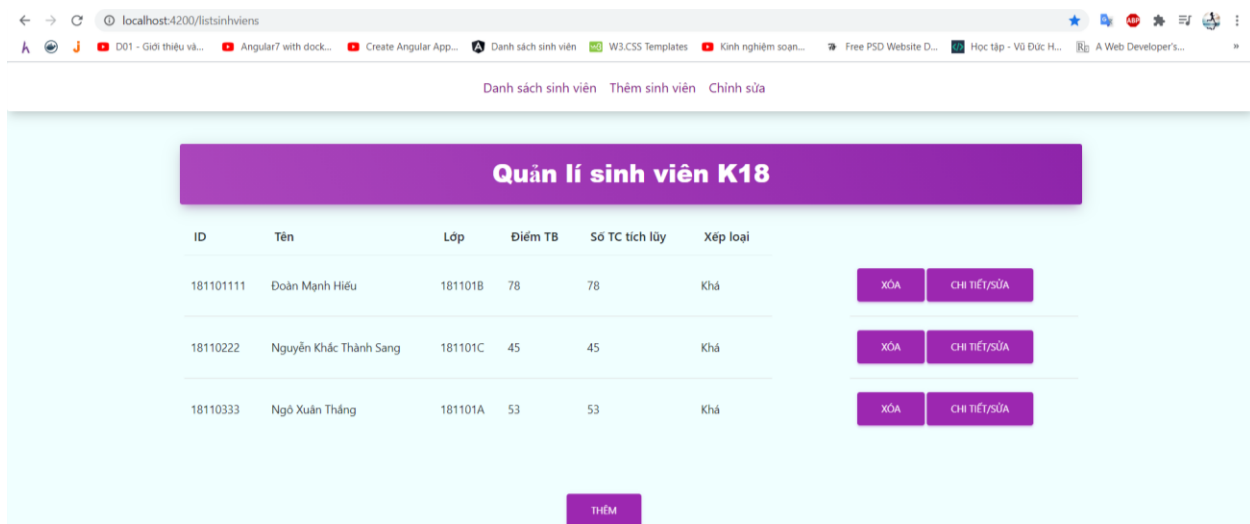
3 image:



Và các dịch vụ (container) được xây dựng bởi image



Bây giờ ta hoàn toàn có thể truy cập vào port 4200 để xem trang web



Để tiến hành dừng và xóa các container ta dùng lệnh:

```
PS C:\Users\ChienNgo\Desktop\ĐTDM> docker-compose down
```

Kết quả:

```
PS C:\Users\ChienNgo\Desktop\ĐTDM> docker-compose down
Removing Sping-boot ... done
Removing MySQL ... done
Removing Angular ... done
Removing network tdm_default
PS C:\Users\ChienNgo\Desktop\ĐTDM>
```

2.4. Hướng dẫn chia sẻ và sử dụng ứng dụng cho người khác

Bởi vì các image đã được lưu trên dockerhub nên ta chỉ cần đưa cho họ file dockercompose.yml để cho họ thực hiện các lên đơn giản của dockercompose họ hoàn toàn có thể xem được đề án đối với người không phải các lập trình viên(như khách hàng, phòng kinh doanh,...). Còn đối với người muốn tìm hiểu sâu về project thì có thể tải project về và thực hiện từng bước như ở trên.

Link github: <https://github.com/ThangNgoXuan/DoAnDienToanDamMay>

Chương 3. Kết luận và tài liệu tham khảo

3.1. Kết luận và kết quả đạt được

3.1.1. Kết quả đạt được:

Sau khi thực hiện đồ án thì nhóm chúng em đã đạt được những kết quả như là vai trò của docker đối với lập trình viên, hiểu được các thành phần cơ bản của docker như là image, container, dockerfile hay là dockercompose. Thực hiện xây dựng được ứng dụng web đơn giản chạy trên nền docker. Và sự tiện lợi của docker đối với từng dự án. Và sau khi thực hiện đồ án không tránh khỏi những sai sót mong thầy thông cảm. Và cuối cùng xin cảm ơn thầy đã hướng dẫn nhóm em thực hiện đề tài này.

3.1.2. Link demo: <https://youtu.be/OlSZP8l-Yxk>

3.2. Tài liệu tham khảo

*<https://xuanthulab.net/lenh-docker-compose-tao-va-chay-cac-dich-vu-docker.html>

*<https://aws.amazon.com/vi/docker/>

*https://www.youtube.com/watch?v=J2YhWG994Iw&t=90s&fbclid=IwAR3HcuMAGKc1wQhb-kYtEBBA7clGm_GcHU6NxiRIGkmizT5vb0ErsvAOLSQ

*<https://www.youtube.com/watch?v=J2YhWG994Iw&t=90s&fbclid=IwAR0VYCNdl2KcC-jDUcM7wsGQgse14rQTZFsedGBzxL9UTjj4kLtmUO5TYEI>