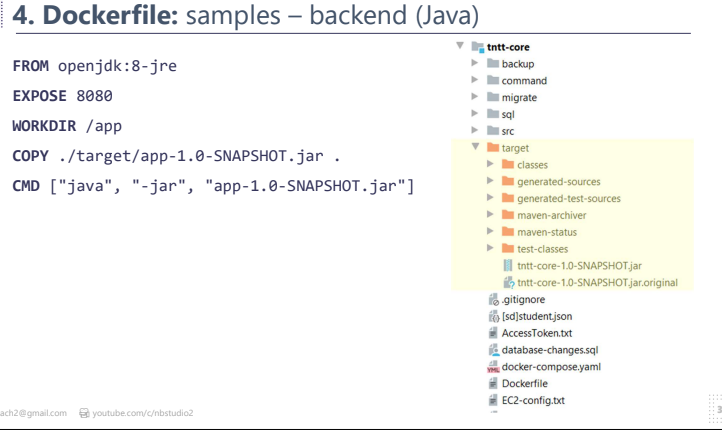
* Dockerfile: là một tập tin bao gồm các chỉ dẫn để build một image .
* Images: là một khuôn mẫu để tạo một container. Thường thì image sẽ dựa trên 1 image có sẵn với những tùy chỉnh thêm. Ví dụ bạn build 1 image dựa trên image Centos mẫu có sẵn để chạy Nginx và những tùy chỉnh, cấu hình để ứng dụng web của bạn có thể chạy được. Bạn có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẽ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile.
* Container là 1 instance của image. Nó được hiểu như 1 thùng chứa các thành phần cần thiết, các thư viện để ứng dụng của chúng ta có thể chạy được. Chúng ta có thể tạo ra nhiều container và ứng dụng của chúng ta sẽ chạy trong những container đó. Các tiến trình trong 1 container độc lập với các tiến trình của container khác trong cùng hệ thống tuy nhiên các tiến trình này đều chia sẻ kernel của host OS
* Bạn không thể chạy container ubuntu trên hệ điều hành windows được
* Liệt kêt tất cả image: **docker images**
* Liệt kê các container đang chạy: **docker ps**
* Liệt kê tất cả các container: **docker ps -a**
* Khi cần chạy container nào cần có sự tương tác với người dùng ta cần thêm cờ it vào: **docker run –it alpine**
* Thoát 1 container ta nhấn Ctr + D nó sẽ thoát hẳn, còn muốn thoát khỏi màn hình tạm thời, container đó vẫn chạy ngầm, ta dùng tổ hợp **Ctr + P + Q ,** để quay lại với container đó ta dùng câu lệnh **docker attach <container\_id>**
* Lệnh **docker exec <container\_id> <command>** dùng để chạy 1 câu lệnh bên trong container
* Câu lệnh để kết nối đến 1 container đang chạy (ssh): **docker exec –it <container\_id> sh.** Khi kết nối kiểu này thì ấn ctr + d sẽ không kết thúc cả container vì khi ta chạy câu lên trên là ta đang chạy 1 tiến trình thứ cấp chạy song song với tiến trình chủ đạo của container thì khi ta kết thúc nó cũng sẽ chẳng liên quan đến tiến trình chủ đạo, tiến trình chủ đạo vẫn chạy bình thường
* Port Mapping ta sẽ mapping port của máy chạy container với container:

Command: **docker run –p <target\_port>:< container\_port>**

* **Volume:** phần bộ nhớ mà docker dùng để lưu dữ liệu dùng cho container, vì image là bất biến. Cụ thể nó là 1 thư mục ảo do docker tạo ra và quản lí việc chúng ta cần làm là lấy cái volume đó gắn vào container(bind mount)
* **Tạo volume: docker volume create [volume name]**
* **Bind mount: docker run –v [local\_dir/volume]:[container\_dir]**
* **Ex: docker volume create pgdata**
* **Docker run –v pgdata: /var/lib/postgresql/data –p 5432:5432 postgres**
* Có thể bind trực tiếp thư mục của máy thật vào thư mục của container trường hợp này chỉ dùng khi máy local là linux, nếu là windows sẽ xảy ra lỗi chính vì vậy volume vẫn được sử dụng
* Dockerfile:
* FROM <image>: chỉ định image gốc mà chúng ta sử dụng để build image của chúng ta
* RUN <command>: giúp ta chạy các câu lệnh đó trong container
* WORKDIR <directory> : tạo một thư mục và đặt thư mục đó là thư mục mặc định của container khi mà được khởi tạo
* COPY <src> <dest> : copy file từ máy local đi vào bên trong image đó
* ADD <src/URL> <dest>: download file từ internet, nếu cần thì giải nén
* EXPOSE: thông báo cái port mà ứng dụng bên trong container đang chạy, câu lệnh này như 1 cái comment để báo cho người đọc docker file biết ứng dụng đang chạy trên port nào thôi
* CMD: câu lệnh mồi, nó chỉ thực thi khi nào chúng ta khởi chạy container
* Tạo image: **docker build –t <image\_name>:<tag> .**
* Dấu chấm tham chiếu đên thư mục chứa docker file. Tất cả nội dung trong folder đó gọi là build context.



* **Docker compose:** Cấu hình và giao tiếp giữa nhiều container với nhau, cú pháp của docker compose, cùng cách mà docker compose sẽ hỗi trợ build image và chạy container dễ dàng ntn.
* **Docker compose:** giúp cho triển khai nhiều container cùng 1 lúc, các container sẽ được khai báo trong file docker compose định dạng yaml sử dụng file này để chạy container đó
* **Yaml:** định dạng khác giống json(key – value), thụt vào thì có nghĩa value nó là 1object, có dấu – thì value nó là mảng, mỗi dấu gạch là 1 phần tử
* **Tạo file docker-compose.yaml:**
* **Version:** là version của docker engine
* **Services:** là