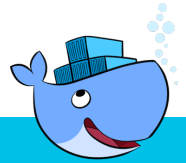
# Tự tạo ra Image của mình 😎

Hiep Nguyen
2021

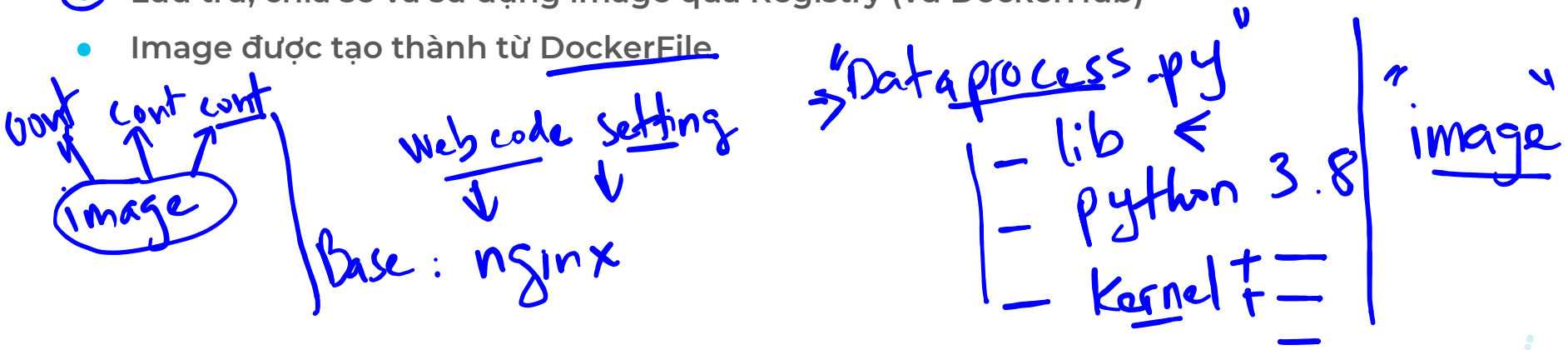
# Nội dung chương học

- Khái niệm và ý nghĩa của Image trong Docker
- Sử dụng Docker Hub registry
- Quản lý local image
- Tự tạo Image của riêng mình
- Cách sử dụng Volume và mount dữ liệu vào container



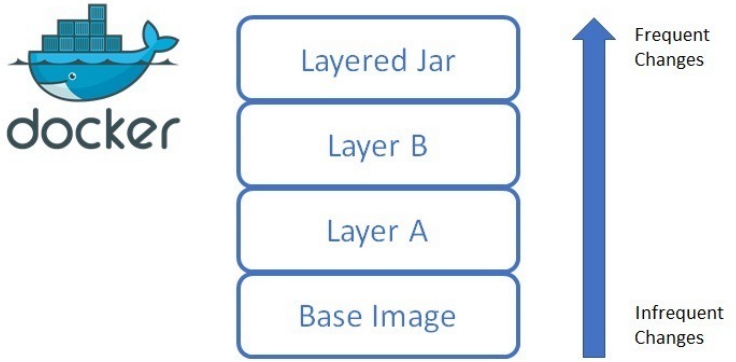
# Sơ bộ về Docker Image

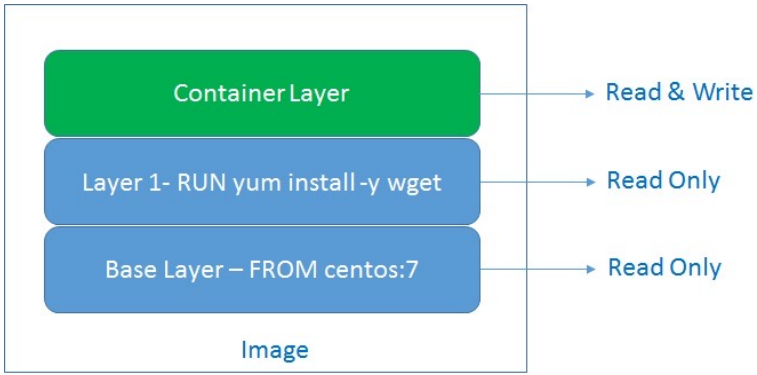
- App binary và dependency
- Metadata về việc khởi tạo một container
- KHÔNG phải là một OS image, chỉ bao gồm những thành phần cần thiết để chạy ứng dụng
- Lưu trữ, chia sẻ và sử dụng image qua Registry (vd DockerHub)
- Image được tạo thành từ DockerFile

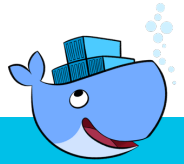


# Có những gì bên trong một Image?

- Image được tạo thành bởi nhiều Layers
- Mỗi layer chỉ được lưu MỘT lần trên host
  - Tiết kiệm bộ nhớ
  - Tiết kiệm thời gian pull/push
- Container là một Layer của Image
- Kiểm tra bằng `docker image inspect` & `docker image history`

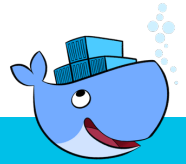


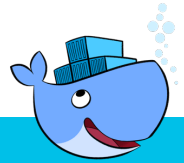
# Cách sử dụng image tag và DockerHub

- Sử dụng image tag
- Đăng nhập DockerHub trên CLI
- Pull và push với DockerHub



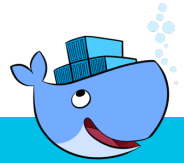
# Tạo ra Image cho riêng mình với DockerFile

- Căn bản về DockerFile
- `FROM` (base image)
- `ENV` (environment variable)
- `RUN` (chạy shell command)
- `EXPOSE` (mở port từ container đến virtual network) •
- `CMD` (chạy shell command sau khi container được tạo ra)
- `WORKDIR` (chuyển vào thư mục hiện tại)
- `COPY` (copy file từ máy tính vào thư mục trong image)
- `docker image build` (tạo image từ DockerFile)



# Bài tập 4 – Dùng DockerFile để tạo Image

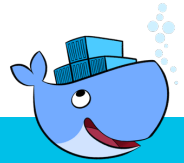
- Sử dụng một Node.js app đang chạy và Dockerize nó
- Tạo Dockerfile. Build-Test-Push-Run.
- Sử dụng node 6.x phiên bản alpine
- Kiểm tra nội dung trong http://localhost
- Tag và push image hoàn thiện lên Docker Hub account (free)
- Xóa local image và chạy lại từ DockerHub



# Bài tập 4 – Dùng DockerFile để tạo Image

- Sử dung node:6-alpine làm base image. Đây là phiên bản node 6 trên nền alpine.
- Ứng dung sẽ lắng nghe traffic trên port 3000, người dung từ ngoài kết nối container vào port 80, nên khi truy cập http://localhost:80 sẽ phải vào được ứng dung.
- Sử dụng alpine package manager để cài đặt tini: 'apk add --update tini'
- Sau đó sẽ tạo ra folder /usr/src/app để lưu app file sử dung 'mkdir'
- Copy file package.json vào thư mục app vừa tạo
- Sau đó chạy 'npm install' để cài đặt các dependency mô tả trong file package
- Clean các thành phần không cần thiết sau khi cài đặt bang 'npm cache clean --force'
- Copy tất cả các file trong thư mục hiện tại vào thư mục app trong image
- Khi chạy container thì cần chạy lệnh '/sbin/tini -- node ./bin/www' để bật dịch vụ
- Sử dung các command FROM, RUN, WORKDIR, COPY, EXPOSE, CMD trong Dockerfile

(Các files đã được đính kèm trong chương học này)



# Thank you