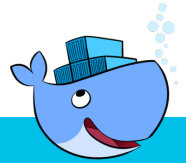
Sử dụng Docker thôi nào ☺

Hiep Nguyen
2021

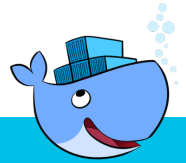
Nội dung chương học

- Học các câu lệnh quản trị Docker
- Chạy một container đầu tiên (nginx)
- Kết nối mạng trong môi trường Container



Kiểm tra hoạt động của Docker

- `Docker version`
- `Docker info`
- **Cú pháp chung:** `docker <command> <action-command> [options]`
- `docker container ls`
- `docker container ls -a`
- `docker container stop`



Container và Virtual Machine khác gì nhau?

- Container != mini Virtual Machine
- Container là Process chạy trên hệ điều hành
  - Process này được giới hạn tài nguyên sử dụng trên OS (file, network, CPU, mem...)
  - Container dừng và thoát như đóng process
- Một số câu lệnh kiểm tra
  - `docker container run --name mongo -d mongo`
  - `docker top mongo`
  - `ps aux`
  - `docker container logs mongo`

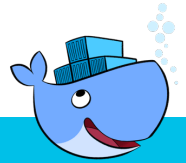
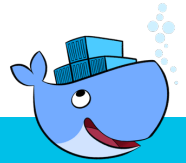


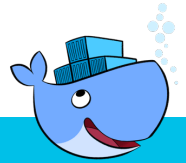
Image và Container khác gì nhau?

- Image là ứng dụng
- Container là process khi chạy ứng dụng đó
- Có thể có nhiều container chạy từ cùng một image
- Image được lưu trên một kho chung gọi là “registry”



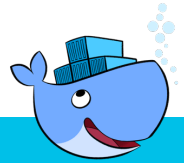
```
docker container run --name Web --publish 8080:80 -d nginx
```

- Kiểm tra xem image nginx có sẵn trên host chưa
- Nếu chưa có thì pull từ Docker Hub về
- Tạo container tên Web dựa trên image nginx
- Cấp địa chỉ IP cho container
- Mở port 8080 trên host và NAT vào port 80 của container
- Chạy container với các lệnh được định nghĩa trong image



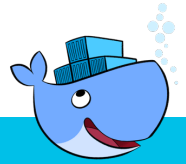
Bài tập 1: chạy cùng lúc nhiều containers

- Chạy 3 container `nginx`, `mysql` và `httpd`
- Lưu ý `detach` và đặt tên cho từng container
- Cổng: `nginx (80:80), httpd (8080:80), mysql (3306:3306)`
- Với mysql, sử dụng `--env` để thiết lập biến môi trường `MYSQL_RANDOM_ROOT_PASSWORD=yes`, sau đó xem `logs` của mysql để tìm password được sinh ra khi khởi chạy container
- Xóa hết container sau khi hoàn thành bài tập với lệnh `stop` và `rm`



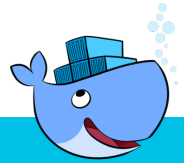
Trong Container đang chạy những gì

- `docker container top` - process list
- `docker container inspect` – xem file cấu hình của container
- `docker container stats` – performance



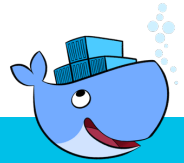
Trong Container đang chạy những gì – Vào shell của container

- `docker container run -it` – chạy mới container và vào shell
  - Container chạy bash mặc định (ubuntu, centos...)
  - Container chạy app (nginx, httpd, drupal...) thì cần thêm command `bash`
- `docker container exec -it` – vào shell của container đang chạy
- Phím tắt để thao tác với Container:
  - `Ctrl + C` để thoát interactive terminal và thoát container
  - `Ctrl + PQ` để thoát interactive terminal



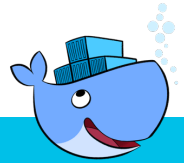
Hệ thống mạng của Docker

- Mỗi container kết nối với một mạng ảo dạng BRIDGE
- Mỗi mạng ảo sau đó được NAT ra IP của host OS
- Mỗi container được kết nối trực tiếp, ngang hang với nhau
- Có thể tạo virtual network riêng cho mỗi lớp ứng dụng



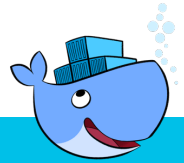
Hệ thống mạng của Docker

- Một container có thể kết nối đến nhiều virtual network khác nhau
- Container cũng có thể kết nối trực tiếp với dải mạng của host (not recommended)
- Có nhiều loại (driver) virtual network cho những mục đích khác nhau (bridge, overlay...)



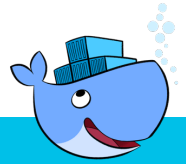
Hệ thống mạng của Docker

- **Show networks** `docker network ls`
- **Kiểm tra cấu hình network** `docker network inspect`
- **Tạo network** `docker network create –driver`
- **Kết nối network vào container** `docker network connect`
- **Bỏ kết nối network vào container** `docker network disconnect`



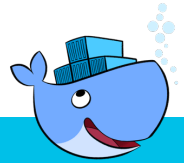
Hệ thống mạng của Docker - DNS

- Container không nói chuyện với nhau bằng IP mà bằng tên (--name)
- DNS là chức năng có sẵn của hệ thống khi container được tạo ra



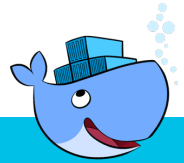
Bài tập 2: tạo và chạy lệnh trong container

- Chạy 2 container là `centos:7` và `ubuntu:16.04` với kết nối vào shell
- Cài đặt curl cho 2 container này
  - **ubuntu:** `apt-get update && apt-get install curl`
  - **centos:** `yum update curl`
- Dùng cơ chế `docker container run –rm` để khi thoát sẽ tự xóa container



Bài tập 3: DNS Round robin

- 2 containers có thể có cùng một DNS name bằng cách dùng `—network-alias`
- Tạo 2 container `elasticsearch:2`
- Đặt alias giống nhau cho 2 container này, ví dụ tên `search`
- Từ container centos ở Bài tập 2, chạy `curl -s search:9200` nhiều lần để thấy kết quả DNS phân giải đến cả 2 container `elasticsearch` và trên `centos` nhận được response từ cả 2 server.



Thank you