

Làm thế nào để chạy Container ?

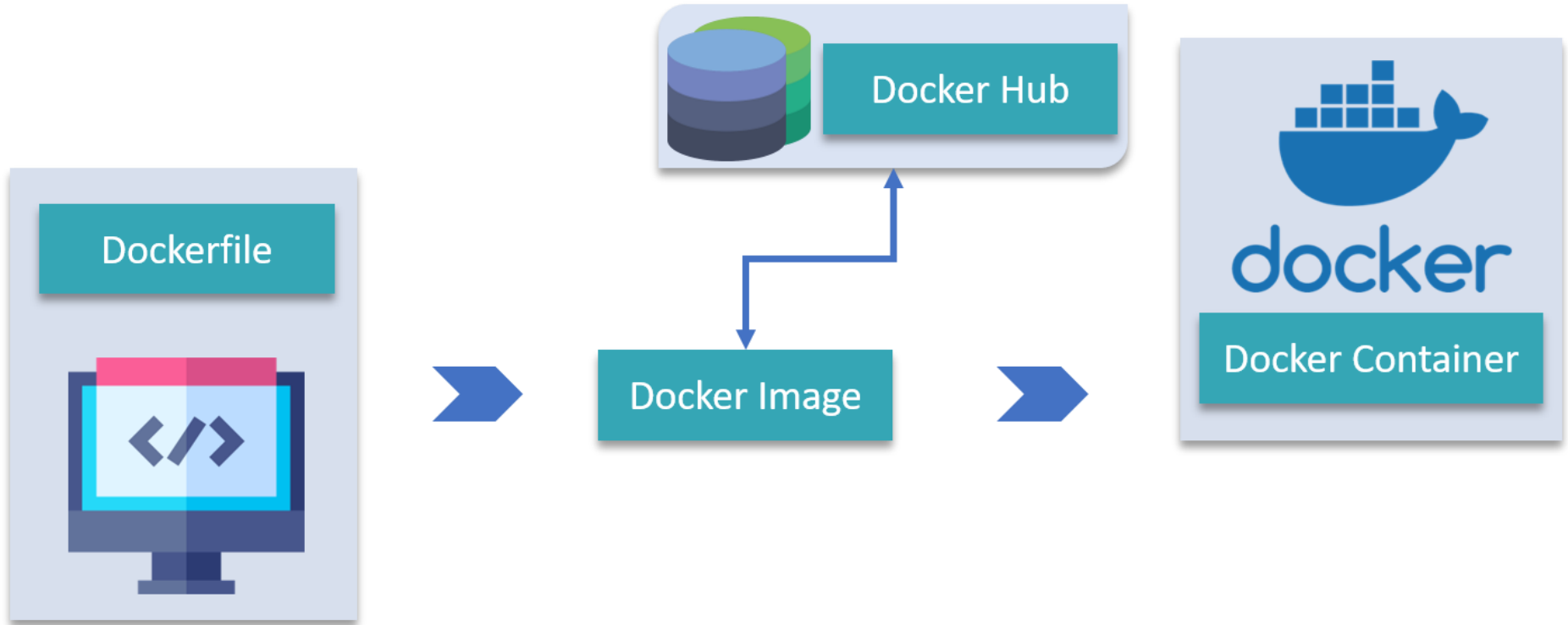


Agenda

- **Chạy một Container (nginx web server)**
- **Điều gì xảy ra khi chạy một Container ?**
- **Containerization (Docker vs VM)**
- **Vào trong Shell của Container**



Docker Concept





Container vs Virtual Machine

- Container không phải Mini-Virtual Machine, chỉ là những processes
- Giới hạn truy cập tài nguyên: file path, network, process
- Tắt ngay khi tiến trình ứng dụng dừng hoạt động

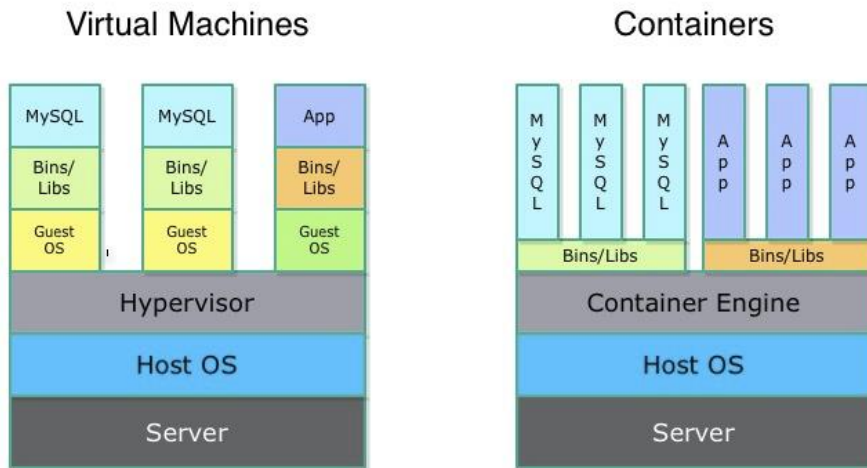
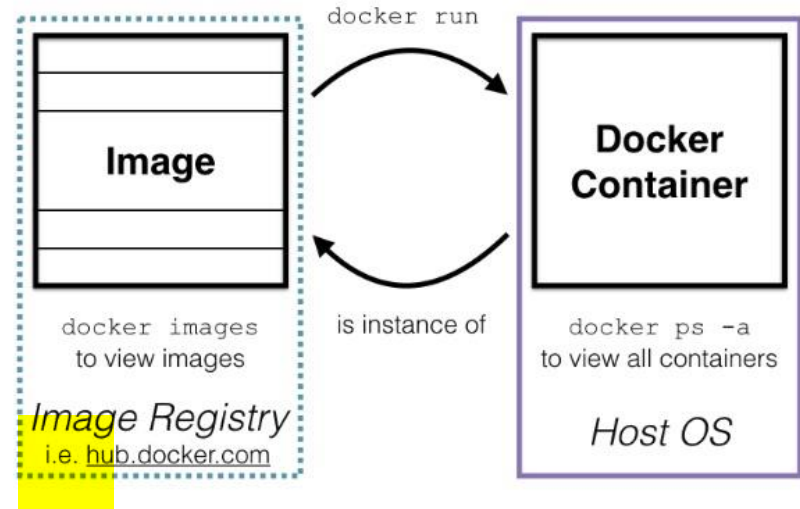




Image vs Container

- Docker Image là tập hợp các file hệ thống mà ứng dụng chúng ta muốn chạy
- Docker Container là kết quả khi ta chạy ứng dụng đó như một tiến trình
- Có thể chạy nhiều Container khác nhau trên cùng một Docker Image
- Mặc định Docker Image được lưu trên "registry": Docker Hub





Chạy một Container

```
docker container run --publish 80:80 nginx
```

1. Tải Image có tên 'nginx' từ Docker Hub
2. Chạy một Container mới từ Image trên
3. Mở port 80 trên IP máy host
4. Truyền data qua port 80 trên IP Container



Chuyện gì xảy ra khi chạy một Container ?



1. Tìm kiếm Docker Image ở trên host trong image cache và không tìm được.
2. Tìm kiếm trên image repository (mặc định Docker Hub)
3. Tải về phiên bản cuối cùng của Nginx Image (nginx:latest - mặc định)
4. Tạo Container mới trên Image vừa tải về
5. Cấp một Virtual IP trên Private Network bên trong Docker Engine
6. Mở port 80 trên host và chuyển dữ liệu đến port 80 của Container
7. Khởi động Container bằng cách chạy **lệnh CMD trong bản Dockerfile của Image.**



Thay đổi các giá trị mặc định

Đặt tên cho Container

Thay đổi phiên bản của Image

```
docker container run --publish 8080:80 --name webhost -d nginx:1.11 nginx -T
```

Thay đổi port trên host

Thay đổi CMD khi khởi động



Bài tập

- Chạy một `mysql` server
- Chạy ở chế độ `--detach` (hoặc `-d`), đặt tên Container `--name mysql` port `3306:3306`
- Khi chạy `mysql`, sử dụng tùy chọn `--env` (hoặc `-e`) để thêm vào `MYSQL_RANDOM_ROOT_PASSWORD=yes`
- Sử dụng câu lệnh `docker container logs` trên `mysql` để tìm mật khẩu được tạo ngẫu nhiên
- Dừng và xóa Container với câu lệnh `docker container stop` và `docker container rm` (cả hai câu lệnh có thể truyền nhiều tham số)
- Sử dụng `docker container ls` để kiểm tra trước khi xóa Container

Mẹo: tham khảo docs.docker.com và `--help`



Bài tập

- Chạy hai `nginx`, `httpd` server
- Chạy ở chế độ `--detach` (hoặc `-d`), đặt tên Container
- Nginx server chạy port 80, apache server chạy port 8080 trên host
- Sử dụng câu lệnh `docker container logs` để kiểm tra truy cập đến server
- Dừng và xóa Container với câu lệnh `docker container stop` và `docker container rm` (cả hai câu lệnh có thể truyền nhiều tham số)
- Sử dụng `docker container ls` để kiểm tra trước khi xóa Container

Mẹo: tham khảo docs.docker.com và `--help`



Điều gì xảy ra trong Container ?

- `docker container top`
Liệt kê các process trên Container
- `docker container inspect`
Các cấu hình của một Container
- `docker container stats`
Thống kê hiệu năng của tất cả Container





Vào trong Shell của Container

- `docker container run -it`

Tham số để tương tác với Shell khi khởi tạo Container

- `docker container exec -it`

Chạy câu lệnh mở rộng tương tác với Container khi đang hoạt động



Vào trong Shell của Container

```
docker container run -it name proxy nginx bash
```

- Tương tác bằng command "ls -la"
- Thoát ra bằng command "exit" và kiểm tra container

```
docker container exec -it mysql bash
```

- Tương tác bằng command "ls -la"
- Thoát ra bằng command "exit" và kiểm tra container



Vào trong Shell của Container

```
docker container run -it name ubuntu ubuntu
```

- Cài đặt Curl
- Sử dụng command curl với <http://google.com>
- Thoát ra bằng command "exit" và kiểm tra container

```
docker container start -ai ubuntu
```

- Chạy thử bằng command curl

```
docker container run -it alpine
```

