VIỆN CÔNG NGHỆ THÔNG TIN T3H

Đào tạo chuyên sâu - Trải nghiệm thực tế

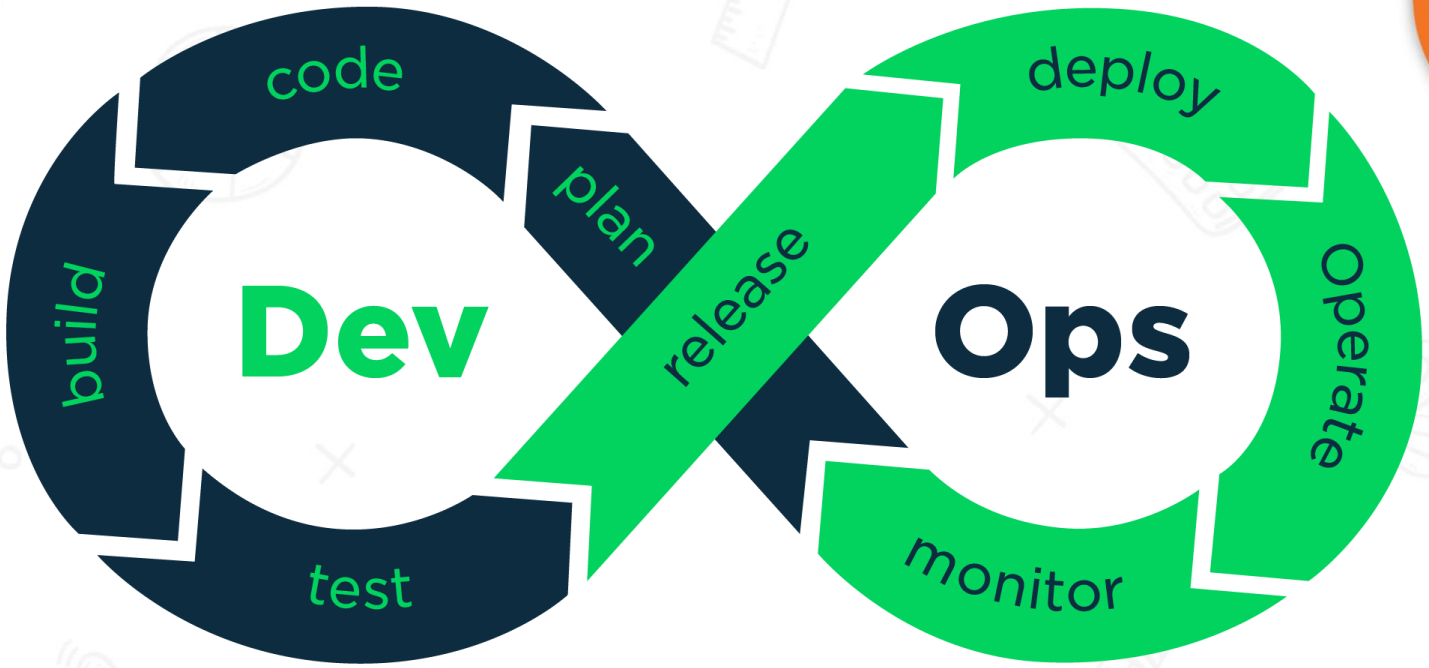
Môn học

DevOps

Giảng viên

Tan Do

0868880797



Nội dung buổi 03

1. Các lệnh cơ bản Linux
2. Kiến trúc các lớp Docker Image
3. Tạo và push image lên DockerHub
4. Các khái niệm Network cơ bản
5. Network trong Docker
6. Giao tiếp giữa các container
7. Chia sẻ và lưu trữ file cho container

Các lệnh cơ bản Linux

- cd
- ls, ls -R, ls -a, ls -al
- touch
- cat filename, cat > filename (Ctrl+D), cat >> filename
- cp
- mv
- rm
- mkdir
- rmdir

Các lệnh cơ bản Linux

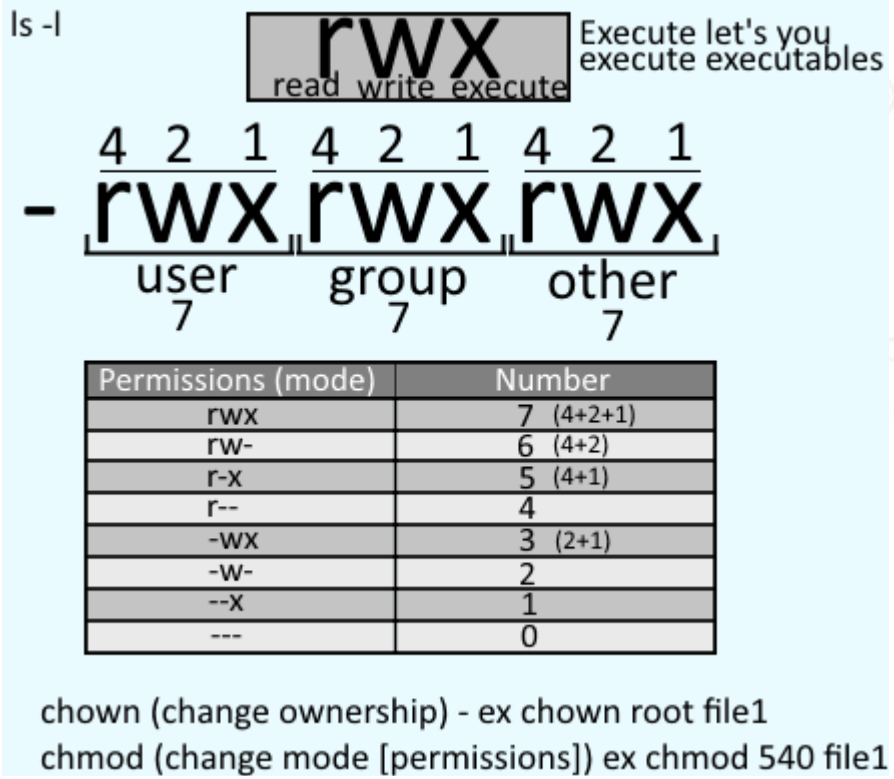
- Find:
  - find /home/ -name note.txt
  - find . -name note.txt
  - / -type d -name my-folder
- grep blue notepad.txt
- htop
- df -m # disk free
- du -sh # disk usage
- head -n 5 filename
- tail -n 5 filename

Các lệnh cơ bản Linux

- cat /etc/*release*
- Sử dụng bộ soạn thảo:
  - vi/vim
    - i: insert
    - q: quit
    - w: write
    - x: exit
  - nano

Các lệnh cơ bản Linux

- chmod 777 filename
- chown user1 filename
- chown user2:group2 directory

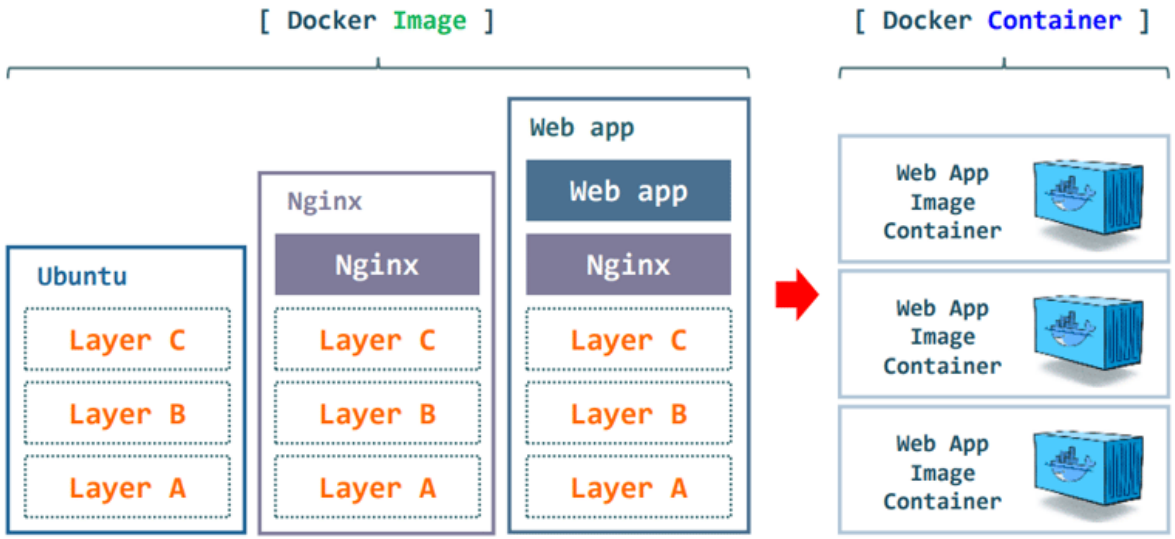


Các lệnh cơ bản Linux

- wget
- tar
  - zip -r archive.zip folder # nén
  - tar -xzvf archive.tar.gz # giải nén
- history
- echo
- zip, unzip
- useradd, userdel

Docker Image

- Có gì bên trong image
  - Image được tạo thành bởi nhiều layer
  - Mỗi layer chỉ được lưu 1 lần trên host
    - Tiết kiệm bộ nhớ
    - Tiết kiệm thời gian pull/push
  - Container là 1 layer của image
  - Kiểm tra:
    - docker image inspect
    - docker image history



Tag và Push image lên DockerHub

- Đăng ký tài khoản trên DockerHub
- Tạo mới repository trên DockerHub
  - my-namespace/my-repository
- Login tài khoản đã đăng ký vào Docker Desktop
- Đăng nhập DockerHub bằng CLI
  - docker login
- Đánh tag image
  - docker image ls
  - docker image tag container-id my-namespace/my-repository
- Push với DockerHub
  - Docker image push my-namespace/my-repository

Các khái niệm Network cơ bản

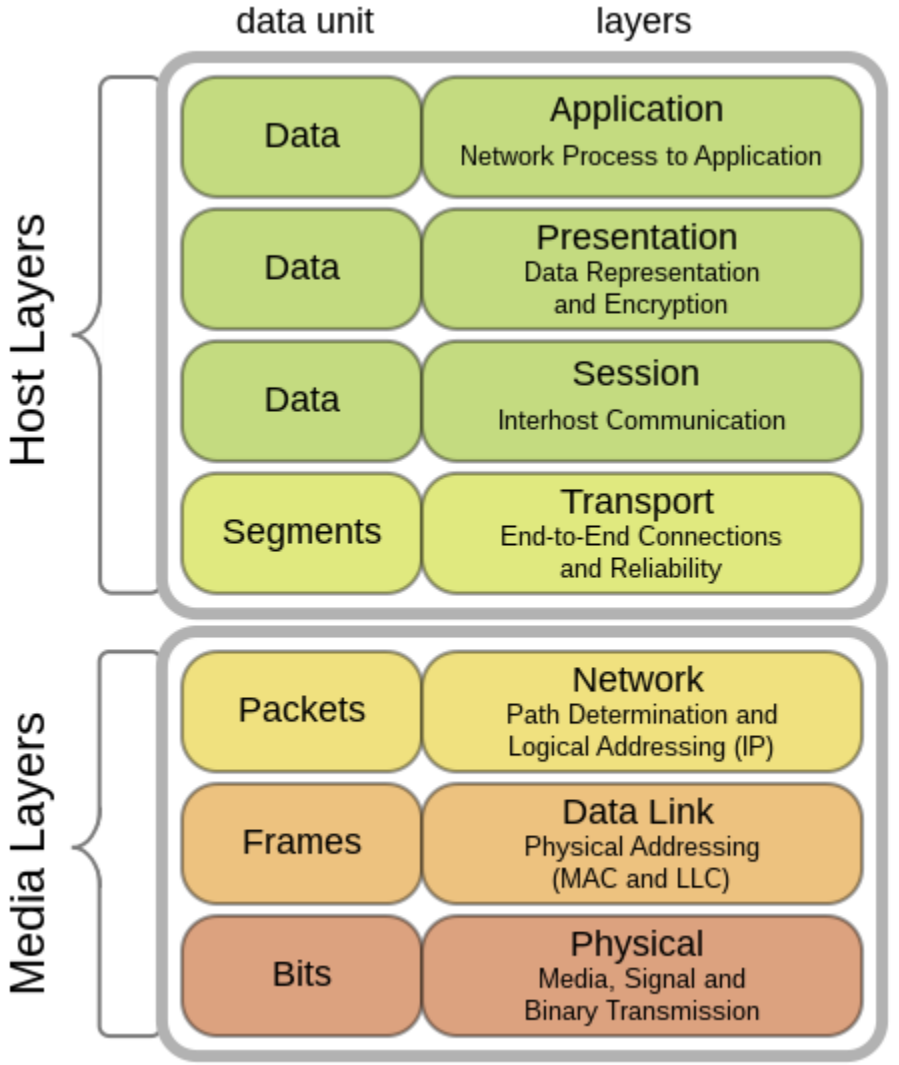
- LAN (Local Area Network): Các thiết bị trong đó được kết nối cùng một vùng địa lý nhỏ, như trong một tòa nhà hoặc một cơ sở làm việc
- WAN (Wide Area Network): Các thiết bị được kết nối qua các khu vực địa lý rộng lớn hơn, thường là thông qua các dịch vụ của các nhà cung cấp mạng
- IP Address (Internet Protocol Address): Địa chỉ giao thức Internet là một địa chỉ số dùng để xác định duy nhất một thiết bị trên mạng Internet hoặc mạng nội bộ
- Subnet: Là một phần của mạng IP lớn được chia thành các phân đoạn nhỏ hơn, giúp quản lý địa chỉ IP hiệu quả hơn
- Router: Thiết bị mạng được sử dụng để kết nối các mạng khác nhau với nhau và định tuyến gói dữ liệu giữa chúng
- Switch: Thiết bị mạng được sử dụng để kết nối các thiết bị mạng với nhau trong một mạng LAN và chuyển tiếp dữ liệu dựa trên địa chỉ MAC

Các khái niệm Network cơ bản

- Firewall: Thiết bị hoặc phần mềm được sử dụng để kiểm soát lưu lượng mạng vào và ra khỏi mạng, bảo vệ mạng khỏi các mối đe dọa và xâm nhập
- Protocol: Giao thức là một tập hợp các quy tắc và quy định được sử dụng để truyền thông giữa các thiết bị trên mạng, như TCP/IP, UDP, HTTP, và FTP
- DNS (Domain Name System): Hệ thống tên miền là một dịch vụ cho phép chuyển đổi tên miền dễ nhớ thành địa chỉ IP để có thể truy cập các trang web và dịch vụ trên Internet
- DHCP (Dynamic Host Configuration Protocol): Giao thức cấu hình máy chủ động là một giao thức mạng được sử dụng để tự động cấp phát địa chỉ IP và các cài đặt mạng khác cho các thiết bị trên mạng

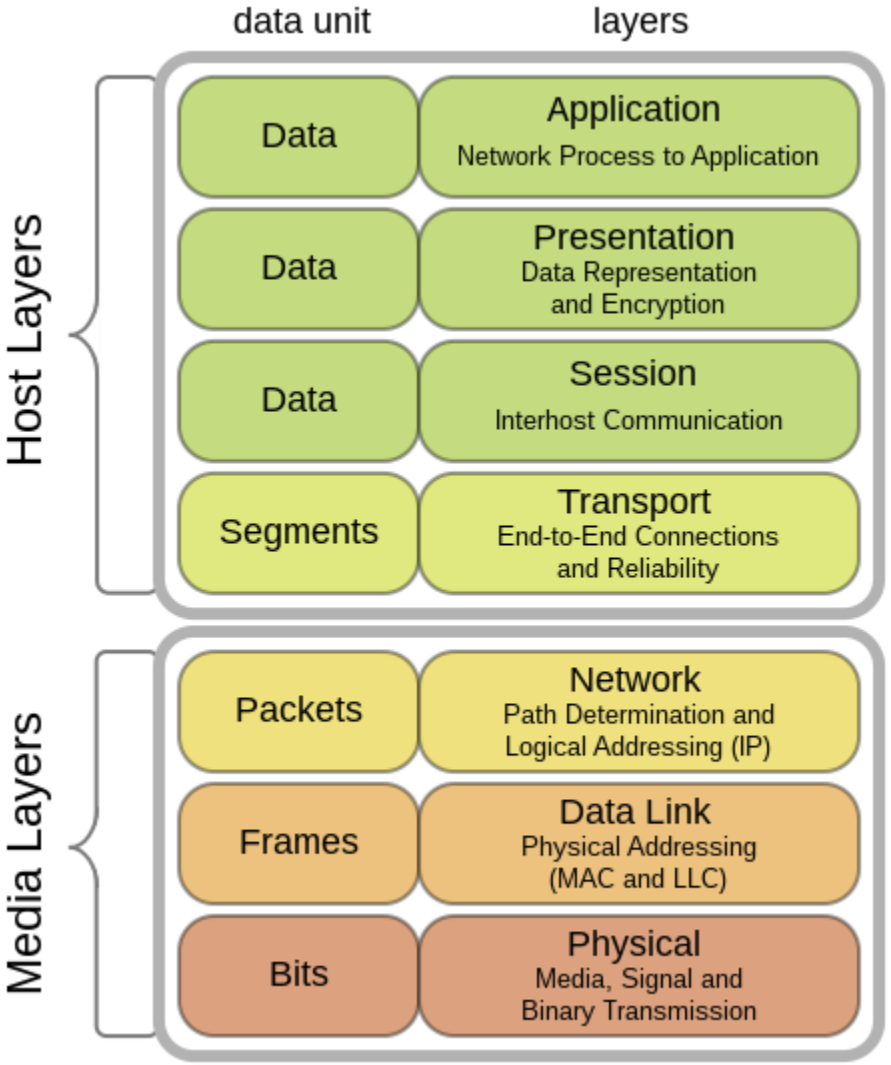
Các khái niệm Network cơ bản

- OSI:
  - Open Systems Interconnection
  - Là một mô hình tham chiếu được phát triển bởi ISO (International Organization for Standardization)
  - Mô tả cách mà các thiết bị mạng giao tiếp với nhau
  - Phân chia quá trình truyền thông thành 7 tầng, mỗi tầng đều có chức năng riêng biệt và tương tác với các tầng khác thông qua giao diện chuẩn



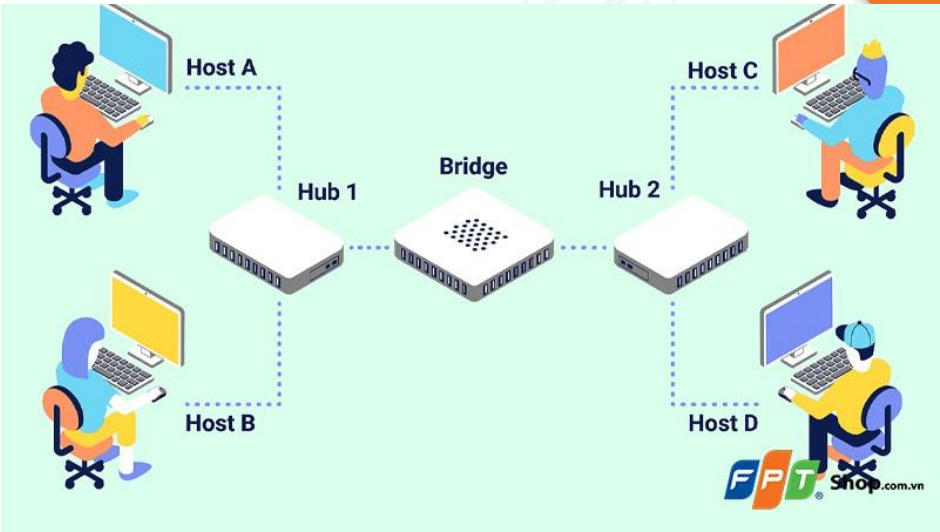
Các khái niệm Network cơ bản

- OSI:
  - Physical Layer: Tầng này định nghĩa các đặc điểm vật lý của phương tiện truyền thông, chẳng hạn như dây cáp, sóng radio, và kích thước tín hiệu
  - Data Link Layer: Tầng này quản lý việc truyền dữ liệu trên các phương tiện truyền thông cụ thể, đồng thời kiểm soát lỗi và quản lý truy cập vào phương tiện truyền thông
  - Network Layer: Tầng này điều hợp việc truyền dữ liệu qua mạng, bao gồm việc định tuyến dữ liệu giữa các mạng con khác nhau
  - Transport Layer: Tầng này cung cấp các dịch vụ truyền dữ liệu đáng tin cậy giữa các máy tính, bao gồm phân đoạn và lắp ghép dữ liệu, kiểm tra lỗi, và kiểm soát lưu lượng
  - Session Layer: Tầng này quản lý các phiên giao tiếp giữa các ứng dụng trên các máy tính khác nhau, bao gồm thiết lập, duy trì và kết thúc phiên
  - Presentation Layer: Tầng này chịu trách nhiệm về việc định dạng dữ liệu để có thể truyền qua mạng, bao gồm mã hóa, nén và mã hóa giải dữ liệu
  - Application Layer: Tầng này cung cấp giao diện cho người dùng và các ứng dụng để truy cập các dịch vụ mạng như truyền tệp, email và web browsing



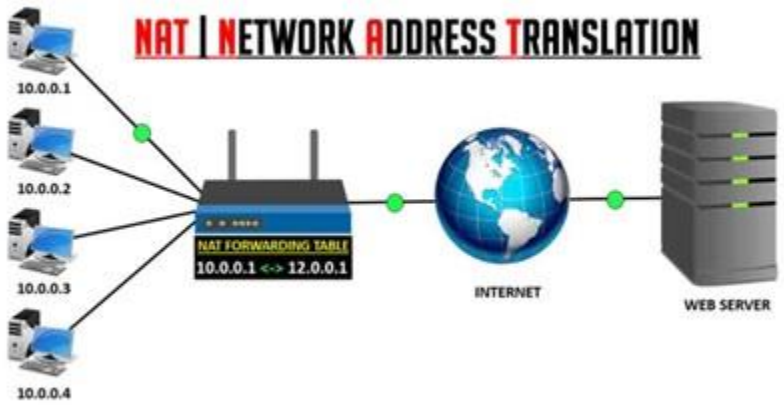
Các khái niệm Network cơ bản

- Bridge (cầu nối):
  - Có chức năng kết nối nhiều mạng LAN (mạng cục bộ) lại với nhau để tạo thành một mạng LAN lớn hơn
  - Việc này được gọi là quá trình kết nối mạng
  - Hoạt động ở tầng liên kết dữ liệu của mô hình OSI, thường được gọi là bộ chuyển mạch Layer 2
  - Chức năng của nó giống như việc xây dựng một liên kết để các thành phần khác nhau trở thành một phần của cùng một mạng



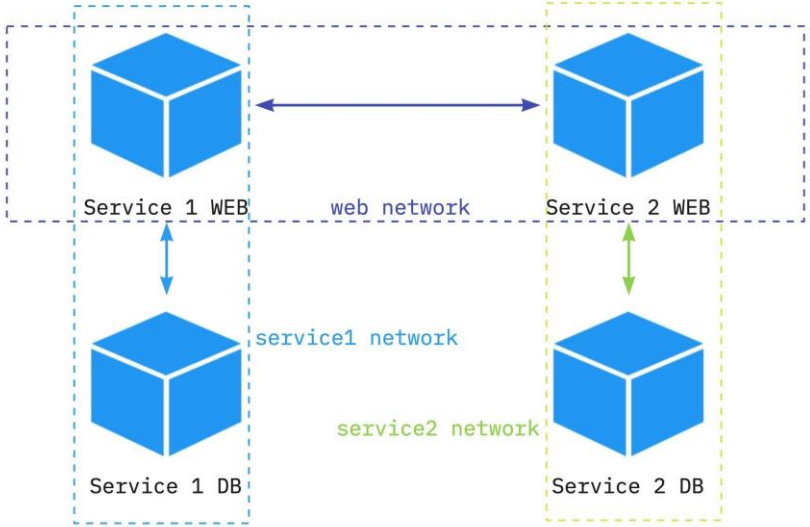
Các khái niệm Network cơ bản

- NAT (Network Address Translation)
  - Là một kỹ thuật được sử dụng trong mạng máy tính để chuyển đổi địa chỉ IP của các thiết bị trong mạng nội bộ sang một địa chỉ IP công cộng trước khi gửi gói dữ liệu ra mạng Internet
  - Kỹ thuật này thường được sử dụng trong các mạng gia đình hoặc doanh nghiệp nhỏ để chia sẻ một địa chỉ IP công cộng cho nhiều thiết bị trong mạng nội bộ



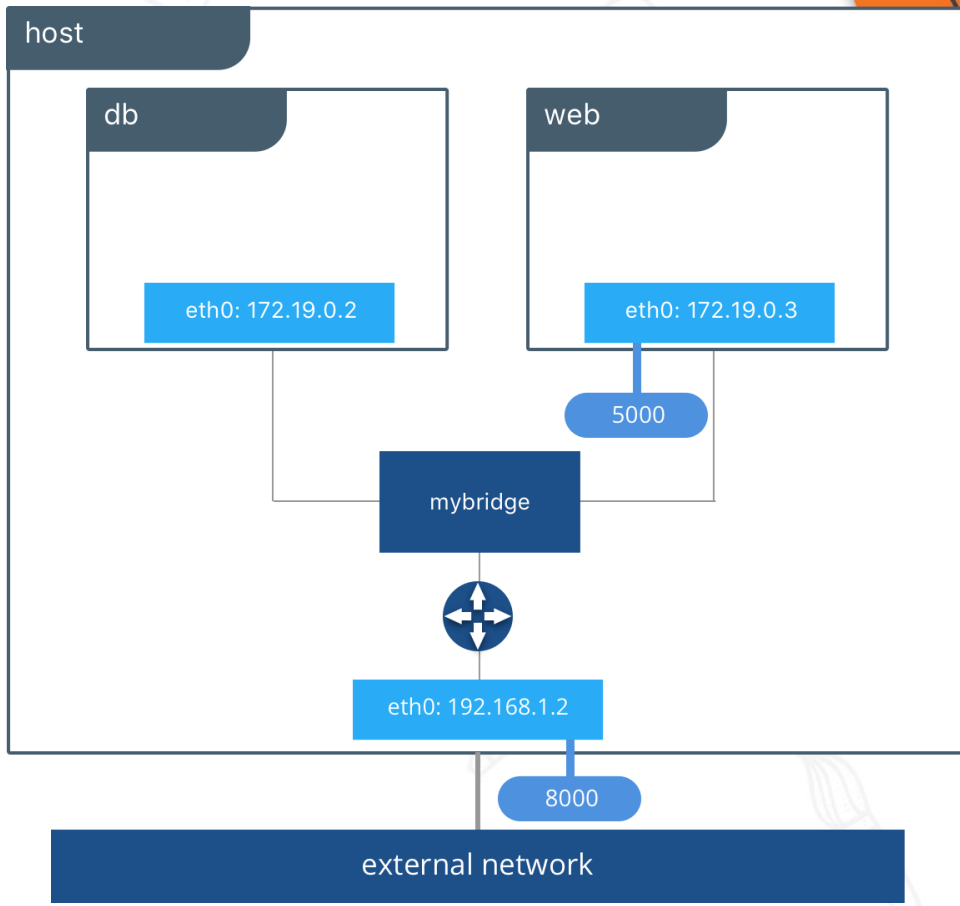
Hệ thống Network của Docker

- 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 hàng với nhau
- Có thể tạo virtual network riêng cho mỗi lớp ứng dụng



Hệ thống Network của Docker

- Mỗi 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 Network của Docker

- BRIDGE
  - Là driver mạng default của Docker
  - Nếu không chỉ định driver thì bridge sẽ là driver mạng mặc định khi khởi tạo
  - Khi cài đặt Docker, virtual bridge docker0 sẽ được tạo ra, docker tìm một subnet chưa được dùng trên host và gán một địa chỉ cho docker0
- HOST
  - Đối với các container độc lập, KHÔNG có nhu cầu giao tiếp với bên ngoài thì ta có thể remove phần network của container & sử dụng network của Docker host luôn
- Overlay
  - Kết nối nhiều docker daemon lại với nhau để bật swarm service giao tiếp với nhau
  - Cũng có thể sử dụng overlay network để cho phép giao tiếp giữa 1 swarm service & 1 container standalone hoặc giữa 2 standalone container trên các docker daemon khác nhau
  - Cho phép loại bỏ việc phải sử dụng khả năng định tuyến của HĐH giữa các container này

Hệ thống Network của Docker

- Danh sách network:
  - docker network ls
- Tạo network:
  - docker network create --driver bridge network1
- Xóa network:
  - docker network rm network1
- Kiểm tra cấu hình:
  - docker network inspect network1
  - docker network inspect container1
- Kết nối network vào container:
  - docker network connect network1 container1
  - docker run -it --name container1 --network network1 ubuntu
- Bỏ kết nối network:
  - docker network disconnect network1 container1

Giao tiếp giữa các container

- Các container không nói chuyện với nhau bằng địa chỉ IP mà bằng tên (--name)
- DNS là chức năng sẵn có của hệ thống khi container được tạo ra và kết nối với custom network
- Sử dụng --network-alias để đặt tên alias cho container

Giao tiếp giữa các container

- docker container run --name server1 --network nw1 -it centos
  - ping centos2

- docker container run --name server2 --network nw1 -it centos
  - ping centos1

Giao tiếp giữa các container

- docker run --name server3 --network nw1 -d alpine ping localhost
  - ping server4
  - docker exec server3 ping server4

- docker run --name server4 --network nw1 -d alpine ping localhost
  - ping server3
  - docker exec server4 ping server3

Giao tiếp giữa các container

- docker run --name server5 --network nw1 -d alpine sleep infinity
  - ping server6
  - docker exec server5 ping server6

- docker run --name server6 --network nw1 -d alpine sleep infinity
  - ping server5
  - docker exec server6 ping server5

Giao tiếp giữa các container

- docker run --name server7 --network nw1 -d alpine top -b
  - ping server8
  - docker exec server7 ping server8

- docker run --name server8 --network nw1 -d alpine top -b
  - ping server7
  - docker exec server8 ping server7

Giao tiếp giữa các container

- docker run --name server09 --network nw1 -d ubuntu bash -c ‘tail -f /dev/null’ # -c command
  - ping server10
  - docker exec server09 ping server10

- docker run --name server10 --network nw1 -d ubuntu bash -c ‘tail -f /dev/null’
  - ping server09
  - docker exec server10 ping server09

Giao tiếp giữa các container

- docker run --name db-server --network nw1 -e MYSQL_ROOT_PASSWORD=root -d mysql
- docker run --name app-server-1 --network nw1 -d alpine sleep infinity
  - ping db-server
  - docker exec app-server-1 apk update
  - docker exec app-server-1 apk add busybox-extras
  - docker exec app-server-1 telnet db-server 3306
- docker run --name app-server-2 --network nw1 -d ubuntu sleep infinity
  - ping db-server
  - docker exec app-server-2 apt update
  - docker exec app-server-2 apt install telnet
  - docker exec app-server-2 telnet db-server 3306

Giao tiếp giữa các container

- docker run --name webserver1 --network-alias webserver --network nw1 -d nginx
- docker run --name webserver2 --network-alias webserver --network nw1 -d nginx
- docker run --name webserver3 --network-alias webserver --network nw1 -d nginx

- docker run --name alb_server --network nw1 -d alpine sleep infinity
  - ping webserver

Giao tiếp giữa các container

- Tạo 3 server client, app_server và db
  - docker run --name client -d nginx
  - docker run --name server -d ubuntu sleep infinity
  - docker run --name db -e MYSQL_ROOT_PASSWORD=root -d -p 3306:3306 mysql
- Cài lệnh ping cho nginx
  - apt-get update
  - apt-get install -y iputils-ping
- Ping giao tiếp giữa các server
- Tạo 2 mạng riêng client_server và server_db
- Kết nối các server tương ứng vào mạng
- Ping lại giao tiếp giữa các server
- Cài đặt telnet & mysql-client cho server
  - apt-get update -y
  - apt-get install telnet -y
  - apt-get install mysql-client -y
  - Kết nối db:
    - mysql -h db -u root -p

Giao tiếp giữa các container

- Tạo 3 server client, app_server và db
  - docker run --name client -d nginx
  - docker run --name server -d centos:7 sleep infinity
  - docker run --name db -e MYSQL_ROOT_PASSWORD=root -d -p 3306:3306 mysql
- Cài lệnh ping cho nginx
  - apt-get update
  - apt-get install -y iputils-ping
- Ping giao tiếp giữa các server
- Tạo 2 mạng riêng client_server và server_db
- Kết nối các server tương ứng vào mạng
- Ping lại giao tiếp giữa các server
- Cài đặt telnet & mysql-client cho server
  - yum update -y
  - yum install telnet -y
  - yum install mysql -y
  - Kết nối db:
    - mysql -h db -u root -p

Chia sẻ và lưu trữ file cho container

- Docker Volume:
  - Là một tính năng trong Docker cho phép lưu trữ dữ liệu bên ngoài các container
  - Tạo ra 1 vùng nhớ riêng cho container trên host
  - Khi tạo một volume trong Docker, thì có thể gắn nó vào container và dữ liệu được lưu trữ trong volume này sẽ tồn tại và không bị mất khi container được xóa hoặc cập nhật
  - docker volume ls
  - Thực hành:
    - Tạo 1 container database mount với volume1
    - Kết nối và thêm dữ liệu
    - Xóa container
    - Tạo mới container và mount volume1

Chia sẻ và lưu trữ file cho container

- Docker Volume:
  - Thực hành 1:
    - docker container run -d --name mysql1 -e MYSQL_ALLOW_EMPTY_PASSWORD=True -v mysql-db:/var/lib/mysql mysql
    - Docker container rm -f mysql1
    - docker container run -d --name mysql2 -e MYSQL_ALLOW_EMPTY_PASSWORD=True -v mysql-db:/var/lib/mysql mysql

Chia sẻ và lưu trữ file cho container

- Docker Volume:
  - Thực hành 2:
    - docker container run -d --name mysql3 -e MYSQL_ALLOW_EMPTY_PASSWORD=True -v C:/data:/var/lib/mysql mysql
    - Docker container rm -f mysql3
    - docker container run -d --name mysql4 -e MYSQL_ALLOW_EMPTY_PASSWORD=True -v C:/data:/var/lib/mysql mysql

Chia sẻ và lưu trữ file cho container

- Bind mount
  - Link 1 folder path trên container với 1 thư mục trên host
  - Thực hành
    - Tạo 1 webserver với source code đơn giản
      - ls
      - pwd
      - docker container run -d --name nginx -p 80:80 -v $(pwd):/usr/share/nginx/html nginx

Chia sẻ và lưu trữ file cho container

So sánh Docker Volume và Bind Mount

- Volume:
  - Command volume trong Dockerfile
  - Có thể override khi run container bằng
    - docker container run -v /path/in/container
  - Volume là 1 vùng nhớ trên ổ đĩa cứng của host
  - Quản lý volume bằng lệnh docker volume
  - Volumne có thể độc lập hoặc kết nối đến 1 hoặc nhiều container
  - Có thể đặt tên cho Volume để dễ quản lý

Chia sẻ và lưu trữ file cho container

So sánh Docker Volume và Bind Mount

- Bind Mount:
  - Map file hoặc folder trên host với file hoặc folder trên container
  - 2 locations trỏ đến cùng 1 file/folder
  - Không tạo ra file trong Dockerfile mà chỉ được sử dụng khi chạy container
    - docker container run -v /Users/admin/data:/path/container (macOS, Linux)
    - docker container run -v //c/users/admin/data:/path/container (Windows)

T3H
IT-Institute

VIỆN CÔNG NGHỆ THÔNG TIN T3H

Đào tạo chuyên sâu - Trải nghiệm thực tế

THANK YOU

www.t3h.edu.vn