

# Báo cáo thực tập

## Mục lục

Báo cáo thực tập.....	1
1. Tìm hiểu về hệ điều hành linux và các thao tác lệnh cơ bản.....	2
1.1. Hệ điều hành linux .....	2
1.2. Một số câu lệnh cơ bản trong linux .....	2
2. Tìm hiểu về cơ sở dữ liệu Mysql và các thao tác lệnh.....	8
2.1. Lí thuyết.....	8
2.2. Các thao tác lệnh trên cơ sở dữ liệu MySQL.....	9
3. Tìm hiểu về network và thực hành trên packet tracer .....	13
3.1. Lí thuyết.....	13
3.2. Định tuyến.....	16
3.3. Subnet mask .....	17
3.4. Vlan .....	19
4. Docker .....	20
4.1. Lí thuyết.....	20
4.2. Một số khái niệm trong docker .....	22
4.2.1. Docker engine .....	22
4.2.2. Docker Hub.....	25
4.3. Lab tạo và sử dụng mysql trong docker.....	25

# 1. Tìm hiểu về hệ điều hành linux và các thao tác lệnh cơ bản

## 1.1. Hệ điều hành linux

Linux là một hệ điều hành mã nguồn mở được phát triển từ Unix vào năm 1991 và viết dựa trên ngôn ngữ C. Hiện nay, nó được hỗ trợ trên nhiều thiết bị như máy tính xách tay, máy tính bàn, thiết bị nhúng và server.

Cấu trúc của hệ điều hành linux như gồm 3 phần:

- Kernel (nhân của hệ điều hành) : chứa module , thư viện để giao tiếp và tương tác với phần cứng máy tính. Ngoài ra còn các nhiệm vụ khác như : quản lí hệ thống phần cứng, hỗ trợ hệ thống tệp tin, quản lí tiến trình, hỗ trợ giao tiếp mạng và bảo mật
- Shell : là một chương trình có chức năng thực thi các câu lệnh từ người dùng hoặc ứng dụng, chuyển tiếp yêu cầu cho kernel xử lí. Ngoài ra, shell còn bảo vệ kernel với những lệnh không hợp lí.
- Application : là chương trình cụ thể được thiết kế để thực hiện nhiệm vụ cụ thể của người dùng

## 1.2. Một số câu lệnh cơ bản trong linux

### 1.2.1. ls

Lệnh này có tác dụng liệt kê tất cả file và thư mục có trong folder ta chọn

Cú pháp: **ls [option] [folder]**

**ls** : liệt kê tất cả file và thư mục có trong thư mục hiện tại

**ls /home/document** : liệt kê tất cả file và thư mục có trong thư mục /home/document

**ls -a** : liệt kê tất cả file và thư mục kể cả thư mục ẩn có trong thư mục hiện tại

### 1.2.2. cd

lệnh này có tác dụng chuyển hướng tệp tin trong linux

cú pháp : **cd [đường dẫn]**

**cd /home/document** : chuyển hướng đến tệp tin /home/document

**cd** : quay về thư mục /home

**cd ..** : chuyển đến thư mục cha của thư mục hiện tại

**cd ~** : quay về thư mục đã ở trước đó

### 1.2.3. pwd

lệnh này có tác dụng trả về đường dẫn của thư mục hiện tại

### 1.2.4. mkdir

lệnh này có tác dụng tạo ra thư mục trống

cú pháp: **mkdir** [name folder]

**mkdir example** : tạo ra thư mục có tên là example trong thư mục hiện tại

**mkdir test** : tạo ra thư mục có tên là test trong thư mục hiện tại

### 1.2.5. rm

lệnh này có tác dụng xóa file, thư mục

cú pháp: **rm** [option] [file/folder]

**rm test.txt** : xóa file test.txt trong thư mục hiện tại

**rm Test/abc.txt** : xóa file abc.txt trong thư mục Test

**rm -r Test** : xóa thư mục Test và tất cả các file, thư mục con có trong đó

### 1.2.6. cp

lệnh này sẽ copy file sang thư mục khác

cú pháp : **cp** [option] [source] [destination]

**cp test.txt home/user/Test** : copy file test.txt sang thư mục home/user/Test

### 1.2.7. mv

có tác dụng di chuyển file đến vị trí khác hoặc đổi tên

cú pháp: **mv** [option] [source] [destination]

**mv test.txt home/user/Test** : di chuyển file test.txt đến thư mục home/user/Test

**mv test.txt aaa.txt** : đổi tên file test.txt thành aaa.txt

### 1.2.8. cat

đọc bản tin trên output tiêu chuẩn

cú pháp: **cat** [option] [file]

cat abc.txt : hiển thị toàn bộ file abc.txt lên output tiêu chuẩn

### **1.2.9. tail**

hiển thị 10 dòng cuối file

cú pháp tail [option] [file]

tail abc.txt : hiển thị 10 dòng cuối file abc.txt

### **1.2.10.head**

hiển thị 10 dòng đầu file

cú pháp head [option] [file]

head abc.txt : hiển thị 10 dòng đầu tiên của file abc.txt

### **1.2.11.vim**

lệnh này giúp chúng ta soạn thảo văn bản

cú pháp : vim [option] [file]

vim abc.txt : hiển thị trình soạn thảo văn bản cho phép ta thay đổi chỉnh sửa file abc.txt

có 3 chế độ:

- normal mode : đây là chế độ mặc định khi mở. Chế độ này cho phép ta điều hướng đọc văn bản
- insert mode : chế độ cho phép chỉnh sửa văn bản. Nhấn i để mở
- command line mode : chế độ này cho phép ta tìm kiếm . Nhấn : hoặc / để mở

### **1.2.12.grep**

lệnh này để tìm kiếm từ khóa nào đó trong file văn bản

cú pháp: grep [option] [từ khóa] [file]

grep home test.txt : tìm từ khóa 'home' trong file test.txt

grep home abc.txt xyz.txt : tìm kiếm từ khóa 'home' trong file abc.txt và xyz.txt

### **1.2.13.find**

lệnh này có tác dụng tìm kiếm file theo điều kiện

cú pháp : find [đường dẫn tệp] [điều kiện]

find /home/user -name "filename" : tìm kiếm trong thư mục /home/user những file và thư mục có tên là filename

find /home/user -type f : tìm kiếm trong thư mục /home/user tất cả file

find /home/user -type d : tìm kiếm trong thư mục /home/user tất cả thư mục

find /home/user -size +1M : tìm kiếm tệp lớn hơn 1MB

find /home/user -size -1M : tìm kiếm tệp nhỏ hơn 1MB

find /home/user -size -1M -type f -name "filename" : tìm kiếm tất cả file dung lượng nhỏ hơn 1M và có tên là "filename"

### **1.2.14.tar, gzip**

2 lệnh này có tác dụng quản lí file lưu trữ, nén và giải nén

Cú pháp: tar [option] [tệp lưu trữ] [file và thư mục]

Một số option như:

-c: tạo một tệp lưu trữ mới

-x : trích xuất các tệp từ tệp lưu trữ

-f : chỉ định tên tệp

-v : hiển thị quá trình thực hiện

-z : nén hoặc giải nén với gzip

-t : liệt kê nội dung tệp

tar -cvf archive.tar file1 file2 directory : tạo 1 tệp lưu trữ tên là archive.tar chứa file1, file2 và thư mục directory

tar -xvf archive.tar : trích xuất tệp archive.tar thành file1, file2, directory

tar -czvf archive.tar.gz file1 file2 directory : trích xuất và nén thành file archive.tar.gz từ file1, file2 và thư mục directory

tar -xzvf archive.tar.gz : trích xuất file archive.tar.gz thành các file ban đầu

### 1.2.15.chown

lệnh này có tác dụng thay đổi quyền sở hữu của một tệp tin

cú pháp chown [option] [chủ sở hữu] [file]

chown hai document.txt : đổi quyền sở hữu file document.txt sang hai

chown -R hai document : thay đổi quyền sở hữu thư mục document bao gồm tất cả các thư mục con sang cho hai

### 1.2.16.chmod

thay đổi quyền của tệp tin. Một tệp tin có 3 quyền đọc, ghi, thực thi. Giờ ta thay đổi các quyền đó

cú pháp chmod [option] [mode tệp] [tệp]

theo qui định quyền đọc đánh số 4, viết đánh số 2, thực thi số 1. Do đó, 7 (=4+2+1) là có cả 3 quyền đọc, ghi, thực thi. 4(=4+0+0) là chỉ có quyền đọc

chmod 755 document.txt : lệnh này phân quyền cho file, user(số 7 đầu tiên) nên có 3 quyền đọc, ghi và thực thi. Group(số 5 ở giữa) có quyền đọc và thực thi, Other(số 5 cuối cùng) có quyền đọc và thực thi.

chmod u=rw,g=w,o= abc.txt : cũng có thể dùng lệnh này để thực thi quyền cho tệp tin. User có quyền đọc ghi, Group có quyền ghi và Other không có quyền gì

### 1.2.17.ping

kiểm tra kết nối đến với một địa chỉ đích nào đó

cú pháp : ping [option] [destination]

ping google.com : kiểm tra kết nối đến trang web google.com

có một số option như:

-c : chỉ định một số gói tin gửi đi

-i : thiết lập thời gian ping giữa các gói tin

-s : thiết lập kích thước gói tin

ping -c 5 6.6.6.6 : gửi 5 gói tin ping đến địa chỉ 6.6.6.6

ping -i 2 7.7.7.7 : gửi gói tin cách nhau 2s

ping -s 500 8.8.8.8 gửi gói tin có kích thước 500 bytes đến địa chỉ 8.8.8.8

### **1.2.18.su**

chuyển đổi người dùng trong phiên làm việc hiện tại

cú pháp: su [option] [tên người dùng]

su root : chuyển sang người dùng root

### **1.2.19.free**

kiểm tra thông tin về bộ nhớ hệ thống

cú pháp: free [option]

có một số tùy chọn như sau :

-b : hiển thị bộ nhớ tính bằng byte

-k : hiển thị bộ nhớ tính bằng kilobyte

-m : hiển thị bộ nhớ bằng megabyte

-g : hiển thị bộ nhớ bằng gigabyte

-h : hiển thị bộ nhớ ở định dạng dễ đọc

-t : hiển thị bộ nhớ tổng

### **1.2.20.du**

kiểm tra thông tin về dung lượng của tệp và thư mục đã sử dụng

cú pháp du [option]

có một số tùy chọn sau:

-a : hiển thị dung lượng của tất cả các tệp và thư mục\

-h : hiển thị dung lượng dạng dễ đọc

-s [thư mục] : hiển thị dung lượng của thư mục chỉ định

-c: hiển thị dung lượng cuối cùng

du -s /home : hiển thị dung lượng của thư mục home

du -a : hiển thị dung lượng của tất cả các tệp và thư mục

...

## 2. Tìm hiểu về cơ sở dữ liệu Mysql và các thao tác lệnh

### 2.1. Lí thuyết

Dữ liệu là những số liệu rời rạc mô tả về sự kiện, sự vật, hiện tượng được chọn lọc để lưu trữ với một mục đích nào đó. Ví dụ như “nguyễn văn a” là tên một người, “1-1-2002” là năm sinh của một người nào đó,... Trong một số ngữ cảnh hoặc trường hợp nào đó, ta cần nhiều dữ liệu và có sự logic hơn. Ví dụ trong quản lý đào tạo sinh viên của một trường đại học bất kỳ, ta cần lưu trữ dữ liệu về thông tin sinh viên, giảng viên, khóa học, học phần, kết quả học tập,... Khi này, ta cần lưu trữ dữ liệu một cách đa dạng và liên kết với nhau. Cơ sở dữ liệu từ đó mà ra đời. Cơ sở dữ liệu là một tập các dữ liệu biểu diễn một vài khía cạnh của thế giới thực, có liên hệ logic thống nhất và được thiết kế để phục vụ một mục đích nào đó.

MÔN HỌC	Tên MH	Mã MH	Số TC	Khoa
	Khoa học máy tính	CS1310	4	CNTT
	Cấu trúc dữ liệu	CS3320	4	CNTT
	Toán rời rạc	MATH2410	3	TOÁN
	Cơ sở dữ liệu	CS3380	3	CNTT

SINH VIÊN	Tên	MSSV	Lớp	Khoa
	Trang	17	1	CNTT
	Ngọc	8	2	CNTT

HỌC PHẦN	Mã HP	Mã MH	Học Kỳ	Năm	Giáo Viên
	85	MATH2410	1	2008	Anh
	92	CS1310	1	2007	Tiến
	112	MATH2410	2	2008	Anh
	119	CS1310	2	2007	Tiến

KẾT QUẢ	MSSV	Mã HP	Điểm
	17	112	10
	17	119	7
	8	85	6
	8	92	9



ĐIỀU KIỆN	MãMH	MãMH_Trước
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Ta có một số yêu cầu về cơ sở dữ liệu như lưu trữ như thế nào để dữ liệu cần phải có sự chính xác, trích xuất dữ liệu như thế nào để nhanh chóng và hiệu quả,... Lúc này, hệ quản trị cơ sở dữ liệu ra đời. Đây là một hệ thống phần mềm giúp tạo và duy trì duy trì cơ sở dữ liệu. Nó có nhiệm vụ như sau:

- Định nghĩa – khai báo khung dữ liệu cùng với các mô tả chi tiết về dữ liệu
- Xây dựng – lưu trữ dữ liệu lên các phương tiện lưu trữ
- Xử lý – truy vấn, cập nhật và phát sinh báo cáo
- Chia sẻ - cho phép nhiều người dùng và ứng dụng truy cập đồng thời cơ sở dữ liệu
- Bảo vệ - đảm bảo dữ liệu được lưu trữ an toàn từ các sự cố, ngăn cản truy cập trái phép,...

MySQL là một hệ quản trị cơ sở dữ liệu tự do mã nguồn mở phổ biến nhất thế giới. Nó có ưu điểm như tốc độ cao, ổn định, và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. MySQL sử dụng ngôn ngữ truy vấn có cấu trúc SQL (Structure Query Language)

## 2.2. Các thao tác lệnh trên cơ sở dữ liệu MySQL

Đầu tiên ta cần tạo một số bảng dữ liệu và điền dữ liệu vào trước tiên

Ta có lệnh tạo bảng như sau:

```
CREATE TABLE sinhvien (  

Id_thanhvien int AUTO_INCREMENT PRIMARY KEY,  

maSV varchar(20) NOT NULL,
```

```
Ten varchar(255) NOT NULL,  
NgaySinh date NOT NULL,  
maKhoa varchar(10) NOT NULL,  
hocPhiDaNop int,  
);
```

```
CREATE TABLE khoa (  
maKhoa varchar(10) PRIMARY KEY,  
ten_khoa varchar(255) NOT NULL,  
hoc_phi int NOT NULL,  
);  
...
```

Sau đây, ta cần thêm dữ liệu vào các bảng với câu lệnh như sau:

```
INSERT INTO sinhvien (maSV, Ten, NgaySinh, maKhoa, hocPhiDaNop)  
VALUES ('SV001', 'Nguyen Van A', '2000-01-01', 'K01', 5000000);
```

```
INSERT INTO monhoc (maMH, TenMH, Khoa, tin_chi, nam_hoc)  
VALUES ('TEL0001', 'giải tích', 'VT', 3,6000000);
```

...

Sau khi insert dữ liệu ta được các bảng như sau:

**SCHEMAS**

Filter objects

- ▼ **baitaplonsdl**
  - ▼ Tables
    - hocphi
    - khoa
    - monhoc
    - sinhvien
    - taikhoan
  - Views
  - Stored Procedures
  - Functions
- ▼ btlicsdl
- ▼ phpmyadmin
- ▼ test
- ▼ testmysql

1 • `select * from sinhvien;`

Limit to 1000 rows

Result Grid

	id_ThanhVien	maSV	Ten	NgaySinh	maKhoa	hocPhiDaNop
▶	2	B200CVT166	Nguyễn Văn Hiếu	2002-08-15	VT	30000000
	3	B200CVT111	Trần Văn Chiến	2002-08-11	VT	30000000
	8	B200CVT001	Trần Đức An	2002-03-03	VT	25000000
	9	B200CVT112	Nguyễn Văn A	2002-02-02	CN	27000000
	10	B190CVT119	Nguyễn Tiến Kiên	2002-10-10	VT	28000000
	13	B200CVT777	Nguyễn Tiến Q	2002-12-12	VT	0
*	NULL	NULL	NULL	NULL	NULL	NULL

**SCHEMAS**

Filter objects

- ▼ **baitaplonsdl**
  - ▼ Tables
    - hocphi
    - khoa
    - monhoc
    - sinhvien
    - taikhoan
  - Views
  - Stored Procedures
  - Functions
- ▼ btlicsdl
- ▼ phpmyadmin
- ▼ test
- ▼ testmysql

1 • `select * from hocphi;`

Result Grid

	role	gia
▶	kĩ thuật	535000
	kinh tế	515000
*	NULL	NULL

Thực hiện một số câu lệnh SQL tương tác với dữ liệu :

**Select \* from sinhvien** : lấy tất cả các thông tin của bảng sinh viên

The screenshot shows a database management interface. On the left, the 'SCHEMAS' pane displays the 'baitaplonsdl' database structure, including tables like 'hocphi', 'khoa', 'monhoc', 'sinhvien', and 'taikhoan'. The top right pane shows the SQL query 'select \* from sinhvien;'. The bottom right pane shows the 'Result Grid' with 7 columns: 'id\_ThanhVien', 'maSV', 'Ten', 'NgaySinh', 'maKhoa', and 'hocPhiDaNop'. It displays 6 rows of data, including a row with NULL values.

	id_ThanhVien	maSV	Ten	NgaySinh	maKhoa	hocPhiDaNop
▶	2	B20DCVT166	Nguyễn Văn Hiếu	2002-08-15	VT	30000000
	3	B20DCVT111	Trần Văn Chiến	2002-08-11	VT	30000000
	8	B20DCVT001	Trần Đức An	2002-03-03	VT	25000000
	9	B20DCVT112	Nguyễn Văn A	2002-02-02	CN	27000000
	10	B19DCVT119	Nguyễn Tiến Kiên	2002-10-10	VT	28000000
	13	B20DCVT777	Nguyễn Tiến Q	2002-12-12	VT	0
*	NULL	NULL	NULL	NULL	NULL	NULL

**Select maSV, Ten, NgaySinh, maKhoa from sinhvien Where maKhoa = 'VT'** : lấy các cột mã sinh viên, tên, ngày sinh và mã khoa của bảng sinh viên với điều kiện mã khoa = VT

The screenshot shows a database management tool interface. On the left, a 'SCHEMAS' tree lists databases: baitaploncsdl, btlicsdl, phpmyadmin, test, and testmysql. The 'baitaploncsdl' database is expanded, showing tables: hocphi, khoa, monhoc, sinhvien, and taikhoan. The main area displays a SQL query: `Select maSV, Ten, NgaySinh, maKhoa from sinhvien where maKhoa = 'VT';`. Below the query, a 'Result Grid' shows the following data:

maSV	Ten	NgaySinh	maKhoa
B20DCVT166	Nguyễn Văn Hiếu	2002-08-15	VT
B20DCVT111	Trần Văn Chiến	2002-08-11	VT
B20DCVT001	Trần Đức An	2002-03-03	VT
B19DCVT119	Nguyễn Tiến Kiên	2002-10-10	VT
B20DCVT777	Nguyễn Tiến Q	2002-12-12	VT

**Select count(\*) from sinhvien :** đếm số bản ghi có trong bảng sinhvien

The screenshot shows the same database management tool interface. The SQL query is now: `select count(*) from sinhvien;`. The 'Result Grid' shows the following data:

count(*)
6

....

### 3. Tìm hiểu về network và thực hành trên packet tracer

#### 3.1. Lí thuyết

Network được dịch ra là mạng lưới, và ở đây sẽ đề cập đến vấn đề kết nối. Tại đây, network được hiểu là một mạng gồm nhiều máy tính khác nhau được kết nối với nhau và cùng chia sẻ tài nguyên. Ngoài máy tính, có thể có nhiều thiết bị khác như máy chủ, các thiết bị mạng và thiết bị ngoại vi. Mục tiêu cuối cùng của mạng là tạo ra một môi trường tải tài nguyên

và thông tin cho các thành viên trong mạng một cách nhanh chóng, bảo mật và đáng tin cậy.

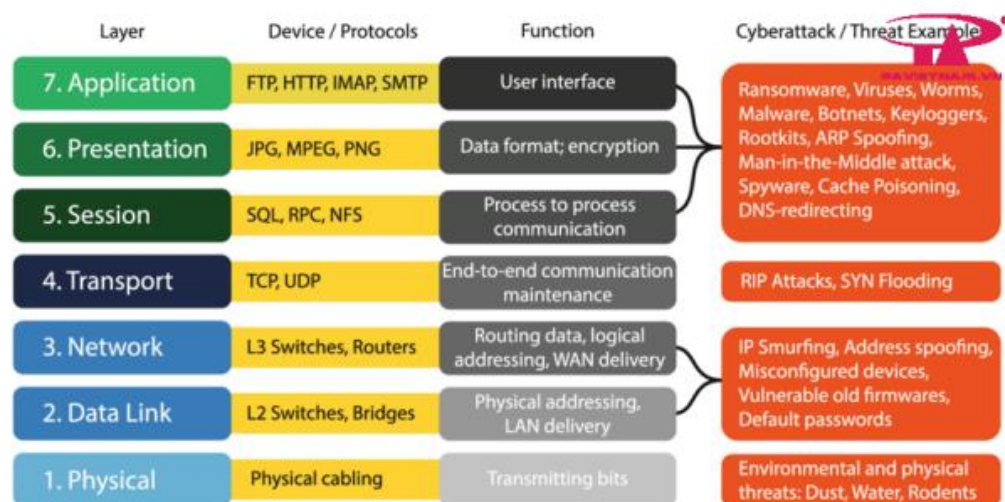
Phân loại mạng bao gồm :

- LAN (Local Area Network) mạng cục bộ, có phạm vi từ vài mét đến vài kilomet thường sử dụng kết nối bus hoặc ring
- MAN (Metropolitan area network) : mạng đô thị có phạm vi trong một thành phố
- WAN (wide area network) : mạng diện rộng, thường là nhiều LAN kết nối với nhau và có phạm vi trong một khu vực địa lí rộng lớn như quốc gia hay lục địa

Mô hình lớp mạng:

- TCP/IP
- OSI

Mô hình OSI là mô hình phân lớp mạng chia thành 7 lớp



Lớp 1 là lớp vật lí. Lớp này có nhiệm vụ là truyền các tín hiệu 0 hoặc 1 qua đường truyền. Ta có thể truyền qua môi trường dây điện (điện áp 5V tương ứng với bit 1 và điện áp 0V tương ứng với bit 0), môi trường quang (truyền qua cáp quang với bit 1 tương ứng với việc có ánh sáng và bit 0 tương đương với việc không có ánh sáng truyền qua môi trường),...

Lớp 2 là lớp liên kết dữ liệu (data link). Lớp này có nhiệm vụ như sau:

- Tạo khung (frame), gửi chúng đến môi trường truyền thông qua lớp vật lí
- Nhận khung, kiểm tra lỗi và chuyển khung không có lỗi lên lớp 3
- Xử lí lỗi : xử lí trong trường hợp các khung bị lặp, bị mất, lỗi
- Điều khiển luồng : giữ cho phát hoặc truyền khung không bị quá nhanh hoặc quá chậm

- Điều khiển truy nhập : cách điều khiển truy nhập vào kênh truyền dùng chung

Lớp 3 là lớp mạng (Network) có nhiệm vụ định tuyến, lựa chọn tuyến đường phù hợp nhất để chuyển tiếp gói tin đến đích. Ngoài ra lớp này còn có nhiệm vụ điều khiển tắc nghẽn và tính phí

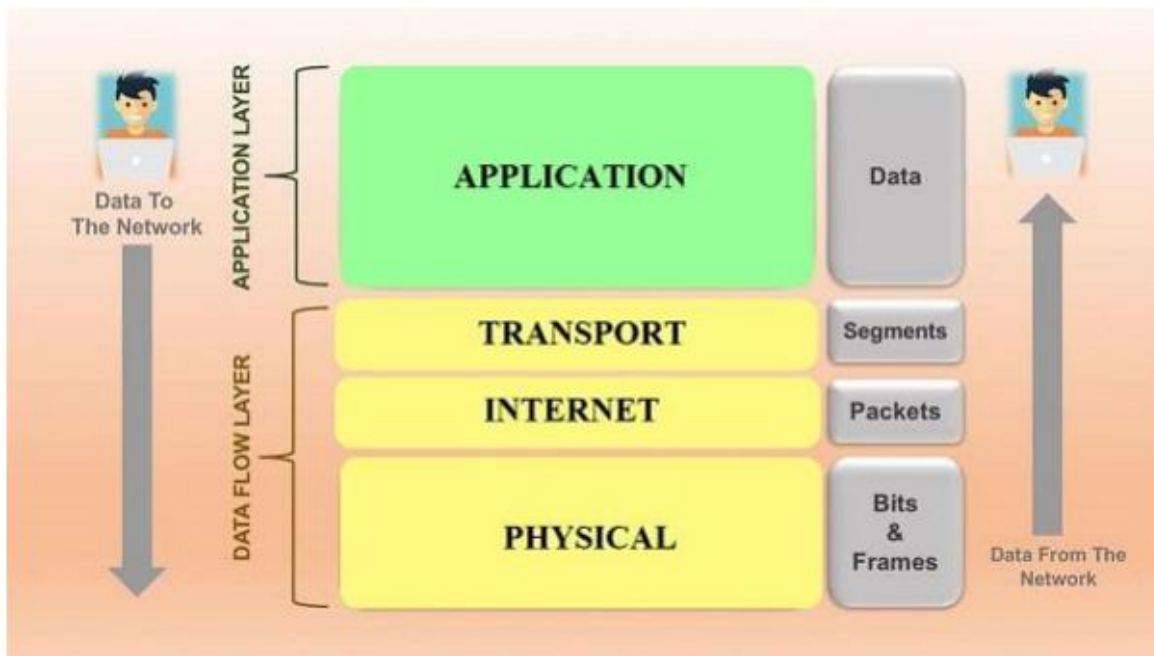
Lớp 4 là lớp giao vận (Transport) vận động như một lớp giao diện giữa lớp cao (dành cho các dịch vụ ứng dụng) và lớp thấp (kết nối mạng). Đảm bảo việc truyền dẫn từ đầu đến cuối không có lỗi và các gói tin không bị mất. Sử dụng giao thức UDP hoặc TCP để truyền.

Lớp 5 là lớp phiên (Session) : lớp phiên cho phép tạo các phiên kết nối giữa các máy khác nhau và từ đó dễ dàng trong giao tiếp và truyền giữ liệu giữa các máy

Lớp 6 là lớp trình diễn (Presentation) liên quan đến việc truyền đi theo format nào. Nó còn mã hóa dữ liệu được cấu trúc theo các định dạng của máy tính thành luồng dữ liệu phù hợp cho truyền dẫn

Lớp 7 là lớp ứng dụng (Application) : gồm các giao thức lớp ứng dụng như HTTP, FTP, SMTP,...

Khác với mô hình OSI, mô hình TCP/IP chỉ có 4 lớp



Lớp vật lý và liên kết dữ liệu trong mô hình OSI sẽ được gộp thành lớp vật lý trong mô hình TCP/IP. Lớp phiên (session), lớp trình diễn (presentation), lớp ứng dụng (application) trong mô hình OSI sẽ được gộp thành lớp Application trong mô hình TCP/IP. Lớp mạng và lớp giao vận được giữ nguyên

### 3.2. Định tuyến

Định tuyến trong mạng được chia làm 2 loại:

- Định tuyến tĩnh
- Định tuyến động

Định tuyến tĩnh là việc quản trị viên trực tiếp thiết lập một tuyến đường cố định để chuyển tiếp gói tin. Loại định tuyến này có ưu điểm là không tốn băng thông trong quá trình trao đổi thông tin định tuyến, Tuy nhiên có rất nhiều nhược điểm. Kể như là việc không thích ứng được sự thay đổi của mạng, các tuyến đường tĩnh cố định này được các quản trị viên cập nhật một cách thủ công. Do vậy sẽ không nhanh chóng khắc phục được sự thay đổi trong mạng khi có sự cố mạng.

Định tuyến động là việc sau khi người quản trị nhập lệnh cấu hình để khởi tạo định tuyến động, thông tin về tuyến sẽ được cập nhật tự động mỗi khi nhận được một thông tin mới từ liên mạng. Các thay đổi về topo mạng sẽ được các route trao đổi với nhau. Do vậy định tuyến động có sự linh hoạt hơn so với định tuyến tĩnh khi mạng có sự thay đổi trạng thái. Ngoài ra các giao thức định tuyến động có thể thực hiện load balancing (cân bằng tải) bằng cách chuyển tải qua các tuyến khác thay vì cố định qua một tuyến hoặc chỉ đổi tuyến khi mạng có sự cố. 2 giao thức định tuyến phổ biến hiện nay có thể kể đến là RIP và OSPF.

RIP(Routing Information Protocols) là giao thức định tuyến được sử dụng trong các mạng IP nhỏ và trung bình. RIP là một trong những giao thức định tuyến lâu đời nhất và dễ dàng cấu hình, nhưng có cũng có nhiều hạn chế về qui mô và hiệu suất. RIP có một số đặc điểm như sau:

- Loại giao thức : RIP là giao thức định tuyến bằng vector khoảng cách (distance vector)
- Thuật toán : Bellman-Ford để tính số bước nhảy ít nhất
- Metric : số bước nhảy (hop count) là metric duy nhất để đo lường khoảng cách đến đích. Giá trị tối đa là 15 nghĩa là nếu số bước nhảy lớn hơn hoặc bằng 16 thì đích đó coi như là không đến được.
- Cập nhật định kì : RIP gửi bảng định tuyến đầy đủ đến tất cả các bộ định tuyến láng giềng cứ mỗi 30s (ripv1) hoặc 180s (ripv2)

Ưu điểm của RIP là dễ dàng triển khai và cấu hình với các mạng nhỏ và tương thích tốt do được hỗ trợ bởi các phần mềm và thiết bị. Nó có một vài nhược điểm như số bước nhảy tối đa là 15 nên giới hạn kích thước mạng. Hiệu suất và khả năng mở rộng kém làm cho rip không phù hợp với các mạng lớn và phức tạp.



OSPF (Open Shortest Path First) là giao thức định tuyến trạng thái liên kết (Link State Routing Protocol) được sử dụng rộng rãi trong các mạng IP lớn và phức tạp . OSPF có một số đặc điểm như sau:

- Loại giao thức : OSPF là một giao thức định tuyến theo trạng thái liên kết (Link State)
- Thuật toán : Dijkstra để tính toán đường đi ngắn nhất
- Metric : OSPF sử dụng cost là metric, chi phí này được xác định dựa trên băng thông của các liên kết
- Cập nhật trạng thái liên kết : OSPF gửi thông tin về trạng thái liên kết của các router trong mạng. Thông tin này được sử dụng và duy trì một cơ sở dữ liệu trạng thái liên kết

Ưu điểm của OSPF có thể kể đến như OSPF có khả năng hội tụ rất nhanh khi có sự thay đổi trong mạng, nhờ việc cập nhật trạng thái liên kết, giúp sử dụng băng thông hiệu quả. Thay vì gửi toàn bộ bảng định tuyến, OSPF chỉ gửi các thay đổi về trạng thái liên kết, giúp sử dụng băng thông hiệu quả hơn. OSPF tính toán đường đi tối ưu dựa trên chi phí của các liên kết, giúp chọn đường đi tốt nhất. Ngoài những ưu điểm trên thì OSPF cũng tồn tại một số hạn chế như: so với RIP thì OSPF có cấu trúc phức tạp hơn, đòi hỏi kiến thức chuyên sâu về mạng. OSPF sử dụng nhiều tài nguyên hệ thống hơn như CPU và bộ nhớ để duy trì cơ sở dữ liệu trạng thái liên kết và thực hiện tính toán đường đi

### 3.3. Subnet mask

Trong lớp 3 của mô hình OSI (lớp Network), muốn định tuyến được gói tin đến đúng địa chỉ thì ta phải có được địa chỉ IP của mạng do đó, nó có một vai trò vô cùng quan trọng. Địa chỉ IP là một chuỗi 32 bit được chia làm 4 phần, mỗi phần 8 bit.

Về cấu tạo thì 32 bit đó được chia làm 2 phần NetID (số duy trì xác định một mạng, các máy tính trong cùng một mạng có phần NetID giống nhau) và HostID (số được gán riêng cho từng máy tính trong mạng). Cụ thể, địa chỉ IP phân làm 5 phân lớp địa chỉ A,B,C,D,E như sau

- Lớp A: có định dạng NetID.HostID. HostID. HostID trong đó 8 bit đầu sẽ là địa chỉ mạng và 24 bit sau sẽ là địa chỉ host. Dạng nhị phân của địa chỉ mạng sẽ là 0xxxxxxx do vậy lớp A có 126 địa chỉ mạng (trừ đi 2 trường hợp đặc biệt 0 và 127) từ 1.0.0.0 đến 126.0.0.0. Phần host có 24 bit vậy nên có 16777214 host (trừ đi 2 trường hợp đặc biệt toàn bit 0 và toàn bit 1)

- Lớp B: có định dạng NetID.NetID. HostID. HostID trong đó 16 bit đầu sẽ là địa chỉ mạng và 16 bit sau sẽ là địa chỉ host. Dạng nhị phân của địa chỉ mạng sẽ là 10xxxxxx do vậy lớp B có 16384 địa chỉ mạng từ 128.0.0.0 đến 191.255.0.0. Phần host có 16 bit vậy nên có 65534 host (trừ đi 2 trường hợp đặc biệt toàn bit 0 và toàn bit 1)

- Lớp C: có định dạng NetID.NetID.NetID.HostID trong đó 24 bit đầu sẽ là địa chỉ mạng và 8 bit sau sẽ là địa chỉ host. Dạng nhị phân của địa chỉ mạng sẽ là 110xxxxx do vậy lớp C có địa chỉ từ 192.0.0.0 đến 223.255.255.0 Phần host có 8 bit vậy nên có 254 host (trừ đi 2 trường hợp đặc biệt toàn bit 0 và toàn bit 1)

Trong phần trên, ta đã biết cách xác định địa chỉ mạng và host từ địa chỉ IP. Vậy ngược lại nếu được yêu cầu thiết lập một hay nhiều mạng cho một công ty và đã biết trước số máy thì sao. Ví dụ công ty có 3 phòng ban, phòng IT có 30 máy, phòng kế toán có 10 máy và phòng sales có 10 máy thì ta nên làm thế nào. Ta có thể dùng 3 địa chỉ mạng khác nhau cho mỗi phòng. Địa chỉ 192.168.1.0/24 cho phòng IT, 192.168.2.0/24 cho phòng kế toán, 192.168.3.0/24 cho phòng sales. Đúng vậy, ta có thể làm được thế này. Tuy nhiên nếu xét về kinh phí phải trả thì sẽ tốn kém rất nhiều do phải thuê cả 3 dải địa chỉ mạng và cả 3 đều không dùng hết số host có thể có của dải mạng (192.168.2.0/24 là địa chỉ lớp C do đó có thể chứa 254 host trong khi phòng IT cần 30 host như vậy sẽ thừa 224 host gây lãng phí. Ngoài ra thì địa chỉ IP không được nhiều lắm do có 32 bit nên nếu làm vậy sẽ không tối ưu). Do vậy subnet mask ra đời để giải quyết vấn đề trên. Mục đích của việc này là chia nhỏ địa chỉ mạng ra làm nhiều địa chỉ hơn với số host phù hợp.

Như ở ví dụ ở trên, có 3 phòng với một số máy tính cụ thể. Giờ đây, với dải địa chỉ 192.168.1.0/24 (đây là địa chỉ lớp C với 24 bit đầu là địa chỉ NetID và 8 bit HostID - 11000000. 10101000.00000001.00000000) ta mượn 1 bit từ địa chỉ host. Khi này, địa chỉ mạng NetID sẽ là 25 bit đầu và hostID sẽ là 7 bit cuối (11000000. 10101000.00000001.00000000). Khi này, ta có 2 dải mạng như sau 192.168.1.0/25 và 192.168.1.128/25 với các thông số như sau:

- 192.168.1.0/25 : địa chỉ host từ 192.168.1.1/25 – 192.168.1.126/25 (từ 192.168.1.0 do là địa chỉ mạng và 192.168.1.127 do là địa chỉ broadcast) tổng cộng chứa 126 host

- 192.168.1.128/25 : địa chỉ host từ 192.168.1.129 – 192.168.1.254 (từ 192.168.1.128 do là địa chỉ mạng và 192.168.1.255 do là địa chỉ broadcast) và chứa 126 host

Tuy nhiên đây mới chỉ có 2 dải mạng, ta có 3 phòng ban nên cần thêm 1 dải mạng nữa. Vì thế ta có thể mượn 2 bit thay vì mượn 1 bit. Khi này địa chỉ mạng như sau: 11000000. 10101000.00000001.00000000. Lúc này ta có 4 dải mạng (thỏa mãn điều kiện  $\geq 3$  dải mạng) và 62 host mỗi dải (thỏa mãn điều kiện). Các dải mạng như sau:

- 192.168.1.0/26 : địa chỉ host từ 192.168.1.1 – 192.168.1.62 (192.168.1.0 là địa chỉ mạng và 192.168.1.63 là địa chỉ broadcast) có 62 host

- 192.168.1.64/26 : địa chỉ host từ 192.168.1.65 – 192.168.1.126 (192.168.1.64 là địa chỉ mạng và 192.168.1.127 là địa chỉ broadcast) có 62 host

- 192.168.1.128/26 : địa chỉ host từ 192.168.1.129 – 192.168.1.190 (192.168.1.128 là địa chỉ mạng và 192.168.1.191 là địa chỉ broadcast) có 62 host

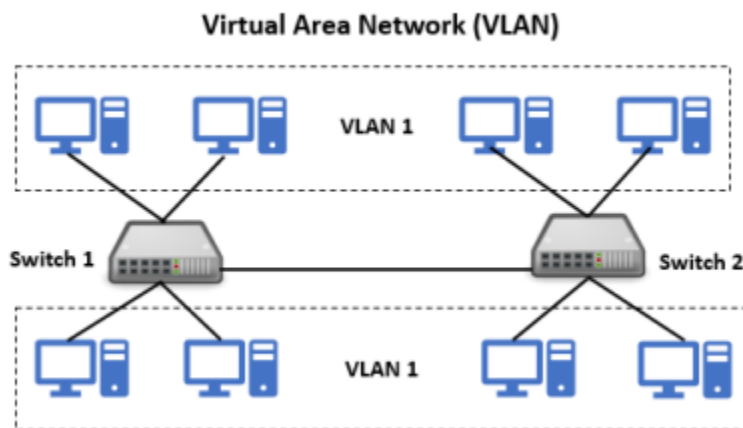
- 192.168.1.192/26 : địa chỉ host từ 192.168.1.193 – 192.168.1.254 (192.168.1.192 là địa chỉ mạng và 192.168.1.255 là địa chỉ broadcast) có 62 host

Như vậy thay vì sử dụng 3 dải mạng thì ta chỉ cần sử dụng 1 dải duy nhất tuy nhiên vẫn đảm bảo được yêu cầu đưa ra. Và sẽ tiết kiệm được khá khá kinh phí.

### 3.4. Vlan

VLAN là viết tắt của Virtual Local Area Network - nghĩa là mạng LAN ảo, là một mạng tùy chỉnh, được hình thành từ một hoặc nhiều mạng LAN, cho phép các nhóm thiết bị khả dụng kết nối cùng với một mạng dù không đặt cạnh nhau. Từ đó tạo nên mạng VLAN được quản lý tương tự với mạng LAN vật lý.

VLAN hay Virtual LAN giúp sử dụng tài nguyên mạng hiệu quả hơn và hữu ích khi kết nối quá nhiều thiết bị cùng một mạng.



Thông thường Router sẽ có vai trò là miền quảng bá. Tuy nhiên, trong mạng VLAN, thiết bị chuyển mạch Switch sẽ có chức năng tương tự tạo nên miền quảng bá. Vậy nên xét về mặt kỹ thuật, mạng VLAN được xem là một miền quảng bá được tạo ra bởi Switch. Hiểu đơn giản là việc ta nhóm một số port của switch lại thành một mạng nhỏ và chỉ cho những máy tính trong cùng một nhóm đó giao tiếp với nhau. Nhờ VLAN ta có thể thực hiện điều này và giúp giảm chi phí hơn so với việc sử dụng LAN thường.

## 4. Docker

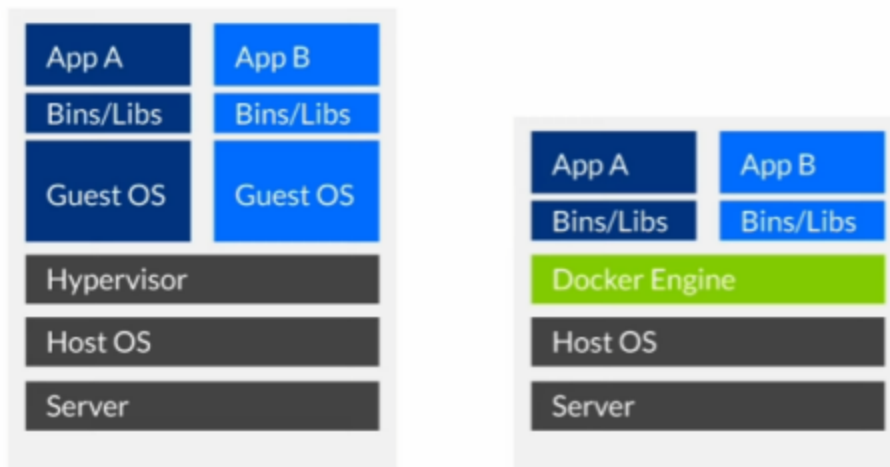
### 4.1. Lí thuyết

Docker là một nền tảng để cung cấp cách để building, deploying và running ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền tảng ảo hóa). Ban đầu viết bằng Python, hiện tại đã chuyển sang Golang.



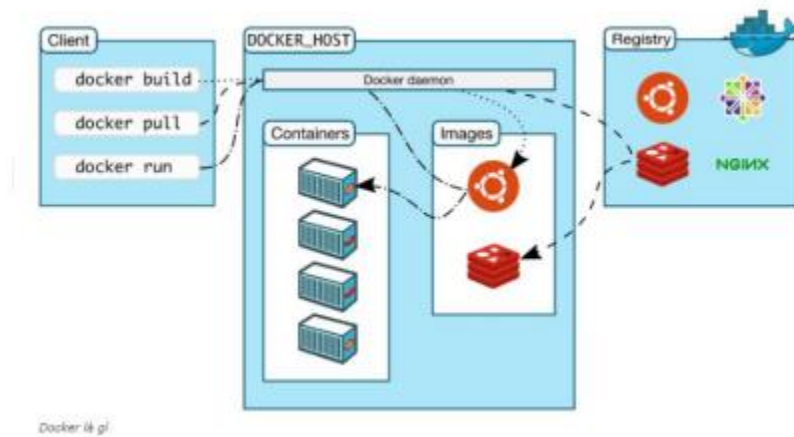
Hình trên là tổng quan của docker. Có thể hiểu đơn giản docker là một ứng dụng giúp tạo ra các môi trường độc lập (container) để chạy các ứng dụng đơn lẻ. Các ứng dụng này chạy độc lập với nhau và không có tương tác qua lại với nhau (trừ khi ta kết nối chúng lại với nhau). Về mặt cấu trúc, ta có thể thấy đây là một nền tảng ảo hóa. Từ một nền tảng cơ sở hạ tầng (Infrastructure) đã được cài hệ điều hành (Host OS) và cài đặt Docker, ta có thể tạo ra nhiều container. Mỗi container này riêng biệt và chạy một ứng dụng cụ thể nào đó. Container này đóng gói lại tất cả các công cụ cần thiết hay môi trường cụ thể nào đó cho việc chạy ứng dụng. Giờ ứng dụng có thể khởi động và hoạt động một cách dễ dàng mà không gặp bất cứ lỗi nào liên quan đến môi trường không tương thích hay thiếu công cụ nào đó. Việc này làm cho quá trình cài đặt ứng dụng trở nên dễ dàng và tiết kiệm thời gian đi rất nhiều.

## VMs vs Docker



Đây là mô hình ảo hóa sử dụng Docker và việc sử dụng phần mềm ảo hóa virture machine (VM). Mặc dù cũng là ảo hóa nhưng Docker lại có những phần khác hơn so với VM. Khi sử dụng các phần mềm ảo hóa, từ một máy chủ server ban đầu có một hệ điều hành riêng, thông qua hypervisor ta có thể chia sẻ tài nguyên và tạo ra được các máy ảo khác nhau. Mỗi máy ảo này có đặc điểm tương đương với máy chủ ban đầu, cũng có hệ điều hành riêng (có thể khác hệ điều hành so với máy chủ vật lí). Điều này giúp chạy và sử dụng các máy ảo như là một máy chủ vật lí, chạy các ứng dụng mong muốn. Tuy nhiên có một nhược điểm là các máy ảo có hệ điều hành riêng biệt nên thành ra tốc độ xử lí vì đó mà trở nên chậm lại (có thể nhận thấy rằng máy ảo xử lí chậm hơn máy vật lí nhiều). Docker thì giải quyết được nhược điểm trên. Docker tạo ra các vùng độc lập để chạy các ứng dụng riêng biệt. Các ứng dụng này được chạy trên hệ điều hành giống máy chủ vật lí. Điều này làm cho việc sử dụng docker sẽ tăng tốc độ xử lí lên tuy nhiên lại không linh hoạt được như sử dụng máy ảo. Và một điều nữa làm cho việc sử dụng docker ngày càng phổ biến hơn là sự linh động. Các không gian chạy các ứng dụng riêng biệt này có thể đóng gói và gửi đi một cách dễ dàng giống như việc ta đóng gói tất cả hàng hóa lại vào container sau đó gửi đi. Khách hàng chỉ cần mở container ra và sau đó sử dụng các ứng dụng trong đó. Nhanh chóng và tiện lợi. Hiện nay hầu hết các ứng dụng đã sử dụng Docker.

## 4.2. Một số khái niệm trong docker



Đây là mô hình hoạt động của Docker, giờ sẽ làm rõ công dụng của từng phần

### 4.2.1. Docker engine

Docker engine là phần quan trọng nhất của Docker có tác dụng như một công cụ để đóng gói ứng dụng được xây dựng theo cấu trúc client-server và được cài trên máy host

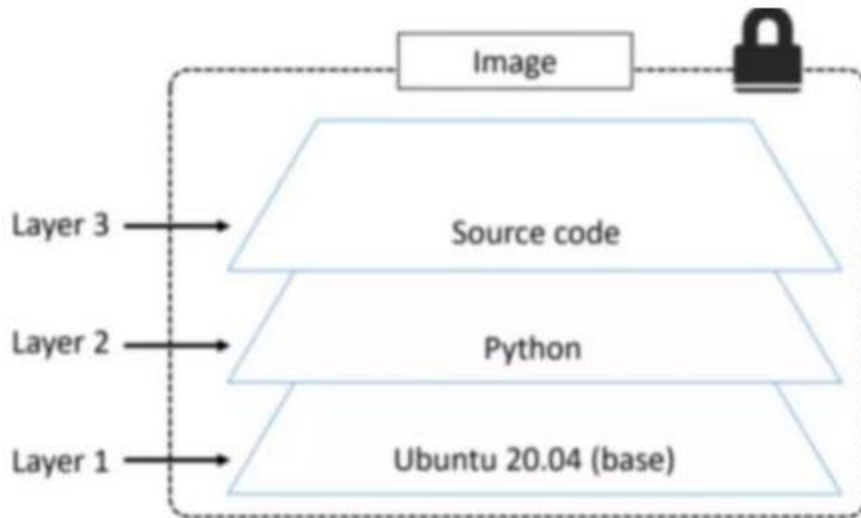
Docker engine được chia làm 3 phần:

- Server : còn có tên gọi khác là docker daemon, có tác dụng quản lí các images, container, volumes và network
- Rest API: controller của docker daemon, sẽ chỉ ra những công việc mà docker daemon sẽ làm.
- Client: Là một công cụ giúp người dùng giao tiếp với Docker host. Người dùng tương tác với docker thông qua command trong terminal (CLI). Docker Client sẽ sử dụng API gửi lệnh tới Docker Daemon.

#### a. Image

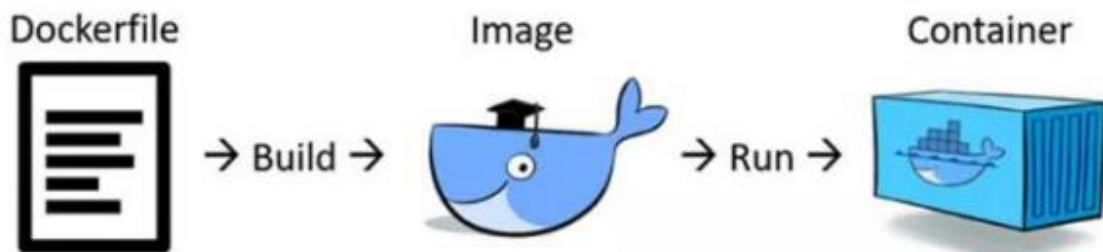
Images là một khuôn mẫu để tạo một container. Thường thì image sẽ dựa trên 1 image có sẵn với những tùy chỉnh thêm. Ví dụ bạn build 1 image dựa trên image Centos mẫu có sẵn để chạy Nginx và những tùy chỉnh, cấu hình để ứng dụng web của bạn có thể chạy được.

Ngoài ra, có thể tự build một image riêng cho mình hoặc sử dụng những image được chia sẻ từ cộng đồng Docker Hub. Một image sẽ được build dựa trên những chỉ dẫn của Dockerfile. Từ những câu lệnh đó, Docker sẽ build ra Docker image (thường có dung lượng nhỏ từ vài MB đến lớn vài GB).



### b. Dockerfile

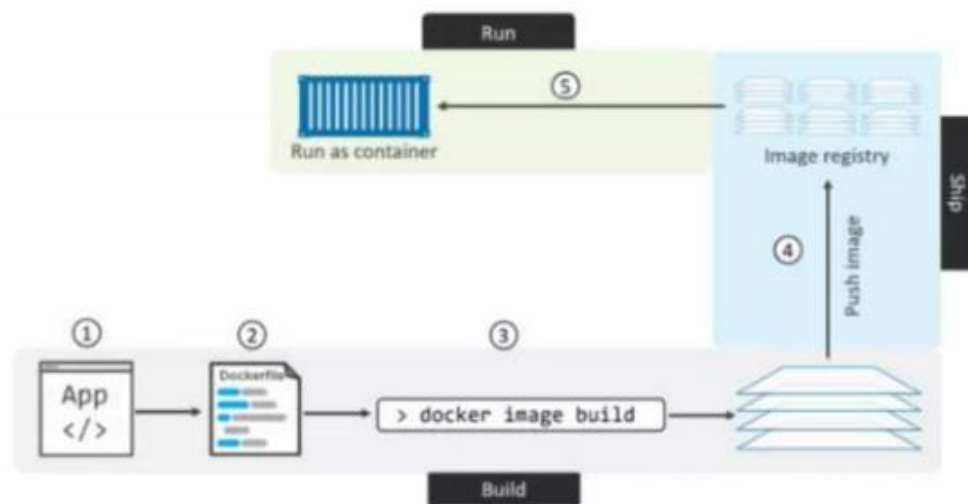
Dockerfile chứa các dòng lệnh để build nên một Images.



### c. Container

Một image có thể run ra một hoặc nhiều container. Ta có thể create, start, stop, delete,... các container dựa trên Docker API hoặc Docker CLI. Mỗi container bao gồm tất cả mọi thứ để nó có thể hoạt động một cách độc lập như: code, runtime, system tools, system libraries, setting,... Bên trong nó sẽ có một ứng dụng cụ thể nào đó.

Quá trình đưa một ứng dụng chạy trong container như sau:



- Bước 1: ta sẽ viết ứng dụng và tất cả các phụ thuộc cần thiết của ứng dụng
- Bước 2: Sau khi ta viết xong ứng dụng, ta cần viết Dockerfile để chuẩn bị cho quá trình build ứng dụng thành Images
- Bước 3: build ứng dụng thành Image. Ta sử dụng docker build để làm
- Bước 4: lưu image tại docker hub (có thể có hoặc không)
- Bước 5 : sử dụng lệnh docker run để biến image thành container và ta đã thành công đưa một ứng dụng chạy trong container

#### d. Network

Docker network có nhiệm vụ cung cấp private network (VLAN) để các container trên một host có thể liên lạc được với nhau, hoặc các container trên nhiều hosts có thể liên lạc được với nhau (multi-host networking).

#### e. Volume

Docker volume là cơ chế tạo và sử dụng dữ liệu của docker, có nhiệm vụ lưu trữ dữ liệu độc lập với vòng đời của container.

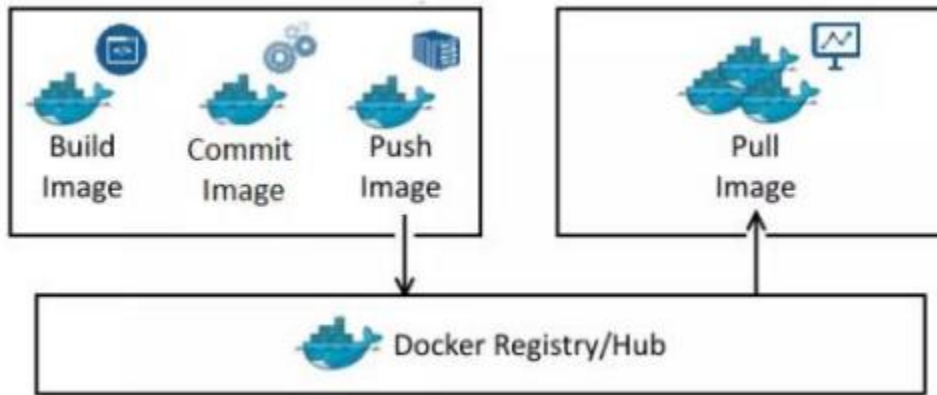
Có 3 trường hợp sử dụng Docker Volume:

- Giữ lại dữ liệu khi một Container bị xóa.
- Để chia sẻ dữ liệu giữa máy chủ vật lý và Docker Container.
- Chia sẻ dữ liệu giữa các Docker Container.



#### 4.2.2. Docker Hub

Docker Hub là một dịch vụ máy chủ cho phép lưu trữ các docker image của cá nhân, công ty, team,... Dịch vụ Docker Hub có thể được cung cấp bởi tổ chức thứ 3 hoặc là dịch vụ nội bộ được xây dựng riêng nếu bạn muốn.



Ta có thể build image và push nó lên docker hub để lưu trữ. Sau đó, khi muốn sử dụng tại nơi khác, ta có thể pull image về và run để tạo container.

### 4.3. Lab tạo và sử dụng mysql trong docker

Các bước thực thi:

- Tải image Mysql
- Dùng các lệnh để tạo ra một cơ sở dữ liệu
- Kết nối cơ sở dữ liệu đó với mysql workbench

#### Bước 1: Tải image mysql

Ta sử dụng lệnh `docker pull mysql:latest` để kéo về để sử dụng

```
PS D:\documents\STUDY\TT viettel\lab> docker pull mysql:latest
latest: Pulling from library/mysql
d9a40b27c30f: Pull complete
fe4b01031aab: Pull complete
aa72c34c4347: Pull complete
473ade985fa2: Pull complete
cc168a9482de: Pull complete
3ca3786815dd: Pull complete
3e3fac98ea83: Pull complete
10e5505c3ae4: Pull complete
a79ade39aab9: Pull complete
ae34d51c6da2: Pull complete
Digest: sha256:d8df069848906979fd7511db00dc22efeb0a33a990d87c3c6d3fcdafd6fc6123
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview mysql:latest
```

## Bước 2: Dùng lệnh để tạo một cơ sở dữ liệu

Sau khi pull thành công mysql, ta thực hiện câu lệnh

**docker run -it -p 3307:3306 -e MYSQL\_ROOT\_PASSWORD=123456 -d mysql**

Trong đó:

**-p 3307:3306** là ánh xạ địa chỉ cổng từ ứng dụng sang máy host vật lí

**-e MYSQL\_ROOT\_PASSWORD=123456** là đặt mật khẩu cho cơ sở dữ liệu này là 123456

**-d mysql** là tên image ta chạy

```
PS D:\documents\STUDY\TT viettel\lab> docker run -it -p 3307:3306 -e MYSQL_ROOT_PASSWORD=123456 -d mysql
24b89ce4bf39175b8e163fc14486d3b68e550e54df2e489fe21b0df8660383a9
```

Ta dùng câu lệnh docker ps để kiểm tra

```
PS D:\documents\STUDY\TT viettel\lab> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	N
24b89ce4bf39	mysql	"docker-entrypoint.s..."	34 seconds ago	Up 32 seconds	33060/tcp, 0.0.0.0:3307->3306/tcp	a

Hiện lên như này có nghĩa là đang có một container mysql đang chạy

**B3: sau khi đã xong 2 bước trên ta quay lại terminal và viết thêm 2 câu lệnh để kết nối thành công mysql**

**Docker exec -it 24b bash**

**mysql -u root -p**

Điền password và ta đã kết nối mysql thành công. Bây giờ ta chỉ cần sử dụng câu lệnh sql là được.

```
PS D:\documents\STUDY\TT viettel\lab> docker exec -it 24b bash
```

```

bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 9.0.1 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

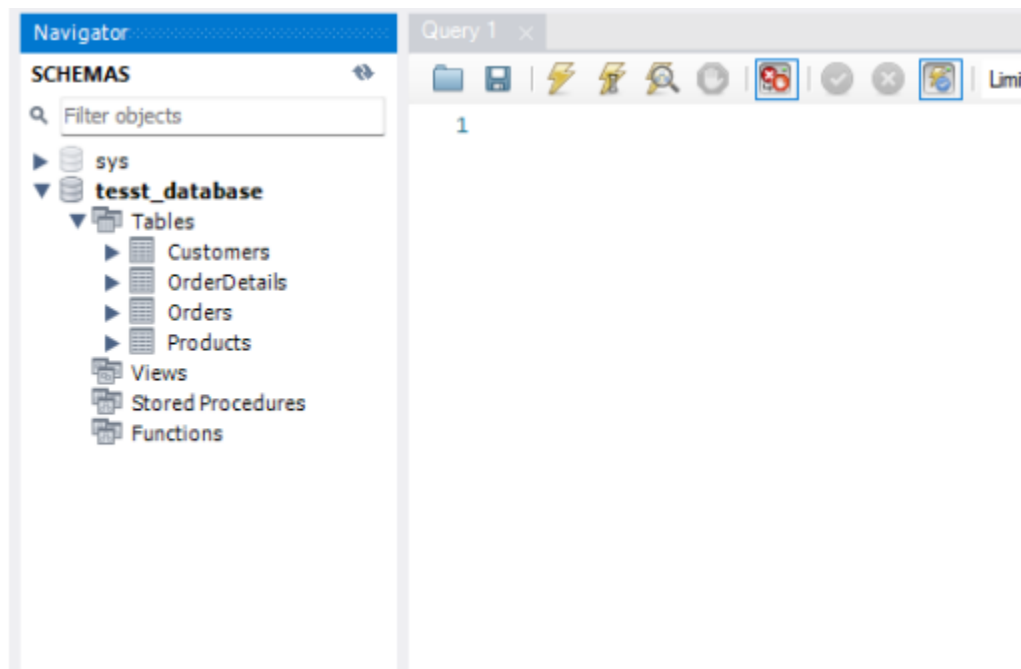
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show table;

```

ta mở mysql workbench, tạo một connection đến cổng 3307, điền mật khẩu và đã kết nối thành công mysql workbench đến với mysql trong container mysql trong docker

ta tạo một schema và chuyển sang schema mới đó.



Tạo và chạy đoạn script như sau:

**-- Tạo bảng Customers**

```

CREATE TABLE Customers (
  CustomerID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),

```

**Email VARCHAR(100)**

**);**

**-- Tạo bảng Orders**

**CREATE TABLE Orders (**

**OrderID INT PRIMARY KEY,**

**OrderDate DATE,**

**CustomerID INT,**

**FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)**

**);**

**-- Tạo bảng Products**

**CREATE TABLE Products (**

**ProductID INT PRIMARY KEY,**

**ProductName VARCHAR(100),**

**Price DECIMAL(10, 2)**

**);**

**-- Tạo bảng OrderDetails**

**CREATE TABLE OrderDetails (**

**OrderDetailID INT PRIMARY KEY,**

**OrderID INT,**

**ProductID INT,**

**Quantity INT,**

**FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),**

**FOREIGN KEY (ProductID) REFERENCES Products(ProductID)**

**);**

**-- Chèn dữ liệu vào bảng Customers**

**INSERT INTO Customers (CustomerID, FirstName, LastName, Email)**

**VALUES**

```
(1, 'John', 'Doe', 'john.doe@example.com'),  
(2, 'Jane', 'Smith', 'jane.smith@example.com'),  
(3, 'Michael', 'Johnson', 'michael.johnson@example.com');
```

-- Chèn dữ liệu vào bảng Orders

```
INSERT INTO Orders (OrderID, OrderDate, CustomerID)  
VALUES
```

```
(1, '2024-08-01', 1),  
(2, '2024-08-02', 2),  
(3, '2024-08-03', 1);
```

-- Chèn dữ liệu vào bảng Products

```
INSERT INTO Products (ProductID, ProductName, Price)  
VALUES
```

```
(1, 'Laptop', 999.99),  
(2, 'Smartphone', 499.99),  
(3, 'Tablet', 299.99);
```

-- Chèn dữ liệu vào bảng OrderDetails

```
INSERT INTO OrderDetails (OrderDetailID, OrderID, ProductID,  
Quantity)
```

```
VALUES
```

```
(1, 1, 1, 1),  
(2, 1, 2, 2),  
(3, 2, 3, 1),  
(4, 3, 1, 1),  
(5, 3, 3, 3);
```

Tiếp theo chuyển sang terminal và thử một số câu lệnh liên quan đến dữ liệu ta vừa import

```
mysql> use tesst_database;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Database changed

```
mysql> show tables;
```

Tables_in_tesst_database
Customers
OrderDetails
Orders
Products

4 rows in set (0.01 sec)

```
mysql> select * from Customers;
```

CustomerID	FirstName	LastName	Email
1	John	Doe	john.doe@example.com
2	Jane	Smith	jane.smith@example.com
3	Michael	Johnson	michael.johnson@example.com

3 rows in set (0.00 sec)

```
mysql> show databases;
```

Database
information_schema
mysql
performance_schema
sys
tesst_database

5 rows in set (0.00 sec)

```
mysql>
```