# **Đề tài: Tìm Hiểu và Triển khai Docker**

1. **Danh sách thành viên & Công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| **Họ & tên** | **MSSV** | **Công việc** | **Tiến độ** |
| Đào Ninh Thái | 175A071427 | - Cài đặt docker và triển khai web  - Hướng dẫn sử dụng | -Xong cài đặt docker trên linux |
| Nguyễn Thanh Tùng | 175A071423 | - Tìm hiểu tài tiệu về Docker.  - Phân tích Ưu điểm / Nhược điểm của Docker.  - So sánh Docker và các phần mềm tương tự khác | 100% |

1. **Nội dung nghiên cứu** 
   1. **Giới thiệu**

* Phát triển ứng dụng ngày yêu cầu không những khả năng viết code mà còn đỏi hỏi rất nhiều công cụ khác. Sử dụng nhiều ngôn ngữ lập trinh bậc cao như C#,C++,Python,…., hay những phần mềm cơ sở dự liệu như mySQL, MariaDB…, hoặc hàng tỷ khác như khung phần mềm, hay kiến trúc phần mềm được update ngày qua ngày như PHP, Nodejs, MongoDB, Laravel, Redis….., việc cài đặt cùng lúc nhiều phần mềm này để chạy cho cùng lúc nhiều dự án cũng như triển khai chúng trên nhiều máy tính khác nhau hoặc từ môi trường local lên sever là cực kỳ khó khăn, tốn thời gian, tốn thận do tốn thời gian của nhà phát triển, và thề là Docker ra đời, phần mềm này giống như một thứ tôn giáo của nhà phát triển, Docker sẽ tạo ra một môi trưởng ảo hóa cho phép chạy nhiều ứng dụng của cả Windows và Linux trên đó mà ko làm ảnh hưởng tới môi trường gốc do đó không phải sợ hiện tượng CSDL dẫm chân lên nhau rồi lăn ra chết nữa .
* Cách thức hoạt động của Docker, Docker sử dụng Docker Engine để tao ra một môi trường ảo hóa gọi là Container hoặc công-tơ-nơ cho các ứng dụng linux và Windows hoạt động trong cái Container này, môi trường trong Container ảo hóa này sẽ được khởi bẳng các Image (hình ảnh) các Image này được định nghĩa bằng các ứng dụng như mySQL, nodejs, ubuntu,….,   
  VD: ta triển khai một dự án yêu cầu có MySQL Ubuntu, PHP thì ta sẽ cần 3 Container trong 3 cái Container đó thì lại có. Image Ubuntu, Image MySQL, Image PHP.  
  Hình dung đến việc ra đảo xây nhà vậy , bạn mang nguyên vật liệu đến cái đảo đó bằng 1 cái tàu chở hàng to tướng trên cái tàu đó thì chở rất nhiều Container trong mỗi Công-te-nơ (Container) thì lại chứa các (Image) nguyên vật liệu khác nhau như Xi Măng, Gạch, Kim Cương, Vàng, Đồ nội Thất….. và bạn dùng những thứ này để xây nhà . Xây trên đảo khác thì lại mang tàu sang đó xây thôi.
* Nói sơ qua thì nhìn chung Docker khá giống với những phần mềm máy áo khác như Virtual Box

Hãy xem qua bảng so sánh sau

|  |  |
| --- | --- |
|  |  |
| Với Docker bạn có thể chạy nhiều Project với các Container ngoài ra để phần cứng để chạy Docker với máy thật là như nhau bạn không cần phải share 1 lượng lớn tài nguyên của mình. Các kích thước của các Container cũng rất nhẹ giúp chương trình chạy tốt hơn. | Với VB với mỗi Project bạn sẽ lại phải chạy 1 máy ảo bạn sẽ lại phải chia tài nguyên phần cứng dành cho máy áo ngoài ra ổ cứng của bạn cũng phải nhận thêm dung lượng của hệ điều hành máy ảo, nó rất nặng nề và chậm chạp ngay cả Core i9 chắc cũng chỉ giúp bạn chạy được 10 ứng dụng là hết đất  Ngoài ra nếu máy áo bị lỗi bạn phải xóa đi cài lại lúc đó mới thấm đau |

VD: Docker bạn ở trong một ngôi nhà có nhiều phòng cần ăn uống thì thì vào nhà bếp cần đi vệ sinh vào phòng vệ sinh cần giải trì vào phòng giải trí rất tiện lợi, nhanh chỉ trong diện tích của 1 ngôi nhà. Còn với VB bạn đang ở trong 1 dãy phố siêu to khổng lồ muốn ăn bạn phải vào nhà hang muốn đi vệ sinh thì cần vô nhà vệ sinh công cộng muốn giải trí thì tới casino việc này làm bạn tốn rất nhiều …. Tiền diện tích xây dừng thì nhiều cũng như thời gian di chuyển thì lâu.

Docker còn có rất nhiều tính năng hữu ích bắt đầu bằng chữ TỰ ĐỘNG <3

* Docker thật sự là một phần mềm của GOD :() nó nhanh tiện lợi và FREE ngoài ra cộng đồng đông đảo sẵn sàng Support tận răng người dùng mới tuy vậy Jerusalem cũng có lúc thất thủ nên thật ra Docker vẫn có một số nhược điểm sau:

+ các Container của Docker hoạt động mặc dù nhanh hơn rất nhiều so với máy ảo nhưng nó vẫn không thể chạy với 100% công suất. lý dó có thể do bị quá tải

+ hệ sinh thái của Docker vẫn chưa đủ mạnh mặc dù là một phần mềm mã nguồn mở nhưng nó không có được sự hỗ trợ từ một số phần mềm do liên quan tới các công ty đối thủ

+ lưu trự dự liệu trong Docker khá khó khăn, các dữ liệu trong Container sẽ biến mất khi bạn tắt nó đi trừ khi bạn lưu nó ở một nơi nào đó trước đó bạn có thể sử dụng Docker Volume nhưng đây vẫn là một vấn đề mà Docker chưa xử lý được triệt để

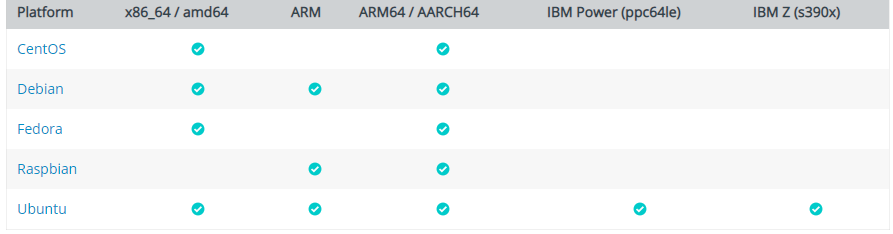
+ các ứng dụng đồ hòa không hoạt động tốt trong các Container mặc dù môi trường Docker cũng như SV không dung GUI nhưng có một số nhà sáng tạo đã làm một số cách để GUI hoạt động trong Container nhưng nhìn chung chúng khá khó hiểu và phức tạp

+ Không phải ứng đụng nào cũng hưởng lợi từ Docker nhìn chung các phần mềm với kiến trúc microservices sẽ chạy tốt trên Docker

+ Docker khá khó sự dụng và cài đặt vì ko có GUI đòi hỏi người dung có kiến thức về máy tính

* 1. **Hướng Dẫn Cài Đặt**

**Cách 1: cài đặt theo Official Repository (Kho lưu trữ chính thức của Docker)**

**B1**: Kiểm tra kiến trúc và hệ điều hành để chuẩn bị cài đặt.

**B2**: nếu máy bạn đã có Docker bản cũ thì hãy tiến hành gỡ bỏ.

sudo apt-get remove docker docker-engine docker.io containerd runc

**B3**: có nhiều cách để cài Docker ở đây sẽ cái đặt theo Repository (Kho)

Cập nhật chỉ apt và cài đặt các gói để cho phép apt sử dụng kho lưu trữ qua HTTPS:

$ sudo apt-get update

$ sudo apt-get install \

apt-transport-https \

ca-certificates \

curl \

gnupg-agent \

software-properties-common

**B4**: Thêm khóa GPG chính thức của Docker dung để mã hóa

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

**B5**: khóa GPG của Docker official là 9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88, bằng cách tìm kiếm 8 ký tự cuối cùng của dấu vân tay.

$ sudo apt-key fingerprint 0EBFCD88

pub rsa4096 2017-02-22 [SCEA]

9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88

uid [ unknown] Docker Release (CE deb) <docker@docker.com>

sub rsa4096 2017-02-22 [S]

**B6**: kiến trúc người dung hiện tại là x86\_64/amd64 sử dụng lệnh sau để thiết lập kho lưu trữ ổn định

$ sudo add-apt-repository \

"deb [arch=amd64] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) \

stable"

**B7**: cài đặt Docker Engine hoặc nếu bạn muốn cài đặt cho một phiên bản tùy ý thì tới B8

$ sudo apt-get update

$ sudo apt-get install docker-ce docker-ce-cli containerd.io

**B8**: cài đặt cho một bản tùy chỉnh

$ apt-cache madison docker-ce

docker-ce | 5:18.09.1~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages

docker-ce | 5:18.09.0~3-0~ubuntu-xenial | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages

docker-ce | 18.06.1~ce~3-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages

docker-ce | 18.06.0~ce~3-0~ubuntu | https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages

...

VD bạn dung 5:18.09.1~3-0~ubuntu-xenial thì chạy

$ sudo apt-get install docker-ce=<VERSION\_STRING> docker-ce-cli=<VERSION\_STRING> containerd.io

**B9**: kiểm tra xem bạn đã cài đặt docker thành công chưa

$ sudo docker run hello-world

**B10**: khi chạy dự án thực tế thì hầu như ta sẽ sử dụng Docker Compose để chạy project, vì thực tế hầu như ta luôn cần nhiều hơn 1 container cho 1 project, tiến hành cài đặt Docker Compose

sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

**B11**: thiết lập quyền

sudo chmod +x /usr/local/bin/docker-compose

**B12**: Kiểm tra và nâng cấp

$ docker-compose --version

docker-compose version 1.25.5, build 1110ad01

docker-compose migrate-to-labels

**B13**: Xóa compose

sudo rm /usr/local/bin/docker-compose

**Cách 2: cài đặt theo Default Repositories (Kho lưu trữ mặc định)**

**B1:** nâng cấp các ứng dụng

sudo apt-get update

**B2:** gỡ bỏ bản docker cũ

sudo apt-get remove docker docker-engine docker.io

**B3:** cài docker

sudo apt install docker.io

**B4:** Khởi động Docker

sudo systemctl start docker

sudo systemctl enable docker

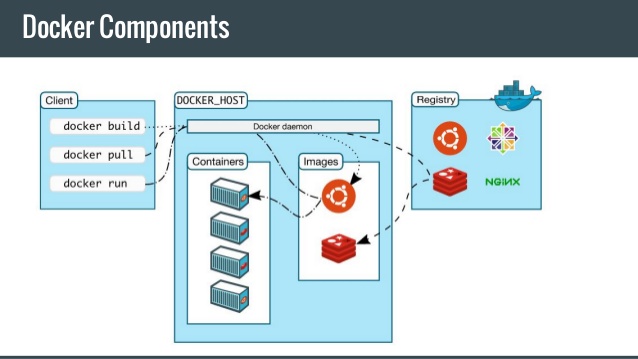
**B5:** cài đặt Docker-compose

sudo curl -L "https://github.com/docker/compose/releases/download/1.25.5/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

**B6**: xét quyền cho Docker-compose

sudo chmod +x /usr/local/bin/docker-compose

c. Hướng dẫn sử dụng / quản trị



1.các thành phần của docker

Registry => Images => Container

Registry (Docker Hub) là server trung tâm nơi chứa các images original, hoặc các bản images đã được cài đặt chỉnh sửa theo nhu cầu riêng biệt.

Images: là OS, một ứng dụng đã được cài đặt và đóng gói. Image chỉ có quyền đọc.

Container là bản thực thể của một image, được clone ra từ image, mọi người sẽ sử dụng và làm việc trên container là chính

Registry (Docker Hub) nằm ở server internet, còn images và container nằm ở máy người dùng.

2. các lệnh cơ bản thường dùng

- Để hiển thị các images ta sử dụng lệnh

docker images

- Để hiển thị các container ta sử dụng lệnh

docker ps -a (liệt kê các container)

docker ps (chỉ liệt kê container đang chạy background)

- Để khởi động / dừng các container ta sử dụng lệnh

docker start/stop <tên hoặc id của container)

- Để xóa container ta sử dụng lệnh

docker rm -f <tên hoặc id của container>

hoặc

docker rm $(docker ps -a -q) (xóa tất cả container)

- Để truy cập vào container đang chạy

docker exec -it <tên hoặc id container>

- Để xem danh sách volume

docker volume ls

- Để xóa volume

docker rm <tên volume>

- Để xóa image

docker rmi <id hoặc tên image>

- Để xóa object không sử dụng

docker system prune

- Để xóa tất cả container không sử dụng

docker container prune

- Để xem thông tin network

docker network ls

- Để chạy docker-compose

docker-compose up

- Để pull image từ docker hub

docker pull <tên image>:<tag>(tag là phiên bản muốn tải , nếu không viết thì sẽ tải bản mới nhất là latest)

Ngoài ra còn rất nhiều lệnh liên quan đến docker khác

Muốn sử dụng docker có nhiều hướng , trong bài này nhóm em sẽ sử dụng docker-compose để build một web server bao gồm apache , php , mysql

Bước 1 : Để sử dụng docker-compose , chúng ta cần tải docker-compose xuống

sudo apt install docker docker-compose

Bước 2: Thêm group để không cần sudo

sudo groupadd docker

sudo usermod -aG docker $USER

newgrp docker

Bước 3: tạo một thư mục trong home đặt tên là web\_dev

mkdir web\_dev

Bước 4: di chuyển vào thư mục vừa tạo và tạo file docker-compose.yml

cd web\_dev

nano docker-compose.yml

Bước 5 :Trong file docker-compose.yml ta viết

version: ‘3.3’ # chọn viết theo bản 3.3 của docs.docker.com/compose/compose-file/

services: # các container trong services

web # tạo dịch vụ đầu tiên là php

image: php:7.3-apache #sử dụng image php:7.3

container\_name: php73 # tên của container

volumes:

- ./php:/var/www/html/ # ánh xạ thư mục vào container

ports:

- 8000:80 # mở cổng 8000 ánh xạ vào cổng 80

# cần chú ý khoảng trắng

Bước 6 : chạy lệnh để docker tiến hành thực thi

docker-compose up (chạy lệnh này trong thư mục chứa file docker-compose.yml

, ở đây là thư mục web\_dev)

Bước 7: kiểm tra xem apache đã chạy chưa bằng cách lên trình duyệt gõ 127.0.0.1:8000

Bước 8: sau khi đã thành công ở bước 7 , ta sẽ tiếp tục di chuyển vào thư mục php và tạo file index.php , từ web\_dev ta gõ lệnh

cd php

nano index.php

- trong file này ta gõ

<?php

echo ”hello”

?>

Bước 9: chạy lại bước 6 xem kết quả , (nếu gặp lỗi ta tiến hành phân quyền cho thư mục php bằng cách :sudo chown daoninhthai:daoninhthai php/ -R)

Nếu không có lỗi , ta sẽ thấy chữ “hello” xuất hiện trên trình duyệt

Bước 10 : thử lại bước 8 nhưng thay nội dung từ “hello” chuyển thành “Thai”,

sau đó load lại trình duyệt , nếu thành công thì trình duyệt sẽ trở thành “Thai”

nếu không thì phải xem lại volume đã đúng chưa vì nếu không có volume sẽ không thể reload file index.php được

Bước 11: Tại thư mục php , ta tiến hành tạo dockerfile

cd php

nano Dockerfile

Bước 12 : trong file này ta viết

FROM php:7.3.3-apache

RUN apt-update && apt-get upgrade -y

RUN docker-php-ext-install mysqli

EXPOSE 80

Bước 13: ta tiến hành sửa file docker-compose.yml để có thể chạy mysql

version: '3.3'

networks: #network

my-network: # đặt tên là my-network

driver: bridge

services:

web:

build: #build từ ...

context: ./

dockerfile: ./php/Dockerfile # file dockerfile(là một file cấu hình)

container\_name: php73

networks:

- my-network

depends\_on:

- db

volumes:

- ./php:/var/www/html

#tham chiếu filhttpd.conf đến /usr/local/apache2/conf/httpd.conf

- ./httpd.conf:/usr/local/apache2/conf/httpd.conf

ports:

- 8000:80

- 443:443

db:

container\_name: mysql8

image: mysql:8.0

command: --defaut-authentication-plugin=mysql\_native\_password

restart: always

networks:

- my-network

volumes:

- ./db:/var/lib/mysql

- ./my.cnf:/etc/mysql/my.cnf

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: test\_db

MYSQL\_USER: devuser

MySQL\_PASSWORD: devpass

ports:

- 6033:3306

Bước 14: viết lại file index.php và chạy lại chương trình

<?php

echo "ddd";

$servername = "localhost";

$username = "devuser";

$password = "devpass";

$db = "test\_db";

// Create connection

$conn = new mysqli($servername, $username, $password, $db);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

echo "Connected successfully";

?>

Bước 15: chạy lại docker-compose (nếu xảy ra lỗi cần xóa container rồi chạy lại)

d. Hướng dẫn phát triển