

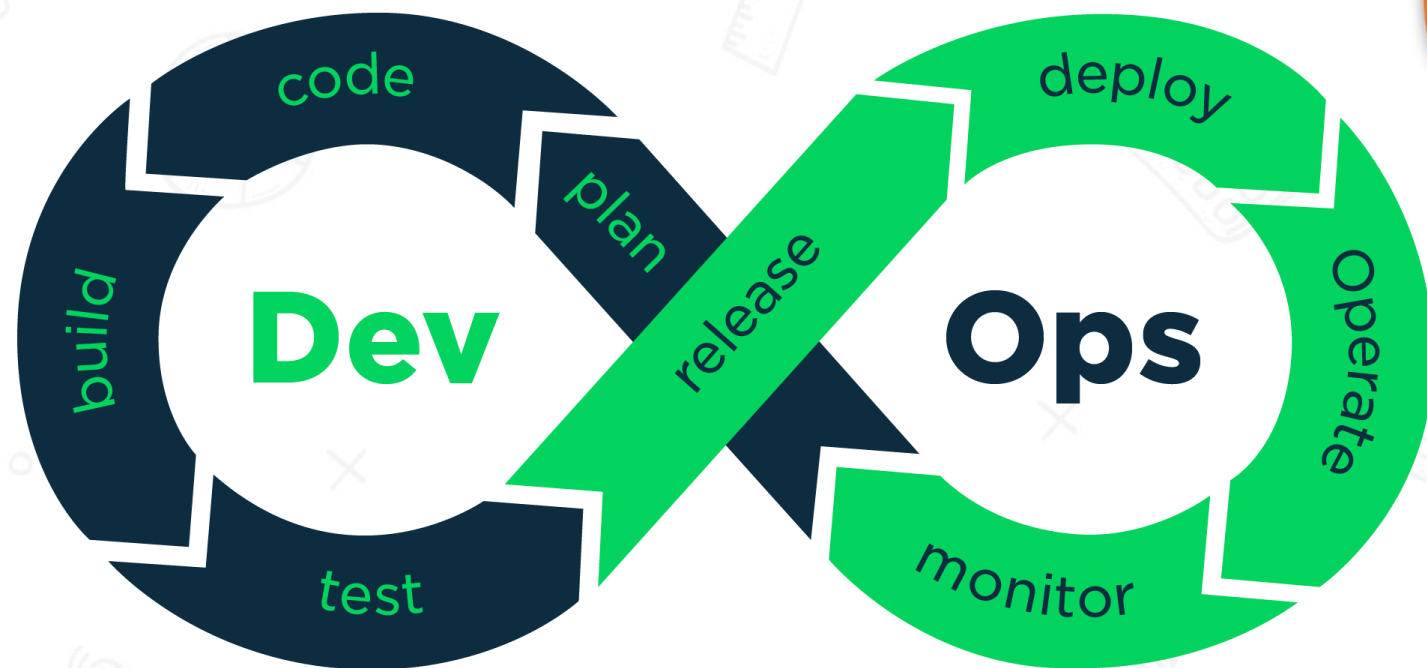
Môn học

DevOps

Giảng viên

Tan Do

0868880797



Nội dung buổi 05

- Docker Compose (phần 2)
 - Deploy ứng dụng webserver đơn giản theo 3 cách
 - Chỉ sử dụng Dockerfile
 - Chỉ sử dụng docker-compose
 - Sử dụng cả Dockerfile và docker-compose
 - Deploy app Nodejs
 - Deploy app Reactjs
 - Deploy app Multi-stage builds
 - Deploy app Nodejs+MySQL
 - Deploy app Reactjs+Nodejs+MySQL
 - Deploy app PHP+Apache+MySQL

Kết hợp Dockerfile và Docker Compose

- Deploy ứng dụng webserver đơn giản sử dụng Dockerfile
- Dockerfile

```
FROM nginx:alpine
```

```
COPY . /usr/share/nginx/html
```

- Build & run
 - docker build -t myapp1 .
 - docker run -it -d -p 81:80 --name web1 myapp1

Kết hợp Dockerfile và Docker Compose

- Deploy ứng dụng webserver đơn giản sử dụng docker compose
- docker-compose.yml

services:

web2:

image: nginx:alpine

volumes:

- `./usr/share/nginx/html`

ports:

- `82:80`

- Run

- `docker-compose up -d`

Kết hợp Dockerfile và Docker Compose

- Deploy ứng dụng webserver đơn giản sử dụng Dockerfile & docker compose

- Dockerfile

```
FROM nginx
```

```
COPY . /usr/share/nginx/html
```

- docker-compose.yml

```
services:
```

```
  web3:
```

```
    build:
```

```
      context: .
```

```
      dockerfile: Dockerfile
```

```
    ports:
```

```
      - 83:80
```

- Run

```
  - docker-compose up -d --build
```


Deploy app Nodejs

- Tạo ứng dụng my-node-app sử dụng nodejs express
 - Khởi tạo source code
 - Cài đặt nodejs
 - Init, install express
 - Tạo file app.js
 - Hoặc sử dụng source code có sẵn
 - <https://github.com/brandoncaulfield/node-api-docker>
- Tạo Dockerfile
- Tạo file .dockerignore
- Build & run

Deploy app Nodejs

- Khởi tạo source code my-node-app
 - `mkdir my-node-app`
 - `cd my-node-app`
 - `npm init -y`
 - `npm install express`
 - `code .`

Deploy app Nodejs

- Tạo app my-node-app file app.js

```
const express = require("express");  
const app = express();  
const port = 3000;
```

```
app.get("/", (req, res) => {  
  res.send("Hello, Docker!");  
});
```

```
app.get("/:name", (req, res) => {  
  res.send(`Hello, ${req.params.name}!`);  
});
```

```
app.listen(port, () => {  
  console.log(`App running at http://localhost:${port}`);  
});
```


Deploy app Nodejs

- Tạo app my-node-app file Dockerfile

Sử dụng image chính thức của Node.js làm base image

FROM node:22

Tạo thư mục làm việc trong container

WORKDIR /usr/src/app

Sao chép package.json và package-lock.json vào thư mục làm việc

COPY package*.json ./

Cài đặt các phụ thuộc của ứng dụng

RUN npm install

Sao chép mã nguồn ứng dụng vào thư mục làm việc

COPY . .

Mở cổng 3000 để truy cập ứng dụng

EXPOSE 3000

Chạy ứng dụng khi container khởi động

CMD ["node", "app.js"]

Deploy app Nodejs

- Tạo app my-node-app file .dockerignore

node_modules

npm-debug.log

Dockerfile

.dockerignore

.git

.gitignore

Deploy app Nodejs

- Tạo app my-node-app Build & run
 - `docker build -t my-node-app .`
 - `docker run -d -p 3000:3000 --name my-node-container my-node-app`
- Sử dụng docker-compose và Dockerfile
 - `docker-compose.yml`

```
services:
  my-node-container:
    build:
      context: .
    ports:
      - 3001:3000
```
 - `docker compose up -d --build`

Deploy app Multi-stage builds

Stage 1: Build

```
FROM node:22 AS build
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . .
RUN npm run build
```

Stage 2: Production

```
FROM node:22-alpine
WORKDIR /usr/src/app
COPY --from=build /usr/src/app .
EXPOSE 3000
CMD ["node", "app.js"]
```

Deploy app Reactjs Multi-stage builds

- Thực hành deploy source code
 - <https://github.com/zupzup/multi-stage-docker-react>
 - <https://github.com/brandonwie/practice-dockerize-reactjs>

Deploy app Nodejs+MySQL

- Tạo Dockerfile để tạo ra các thành phần sau:
 - Container MySQL
 - Container Nodejs
 - Network (bridge) để các service trên kết nối vào mạng này
- Thực hành source code
 - <https://github.com/bezkoder/docker-compose-nodejs-mysql>

Deploy app Nodejs+MySQL

- Tạo docker-compose Networks

name: myapp

networks:

my_app:

Deploy app Nodejs+MySQL

- Tạo docker-compose MySQL
services:

db:

image: `mysql`

environment:

`MYSQL_ROOT_PASSWORD=root`

`MYSQL_USER=app`

`MYSQL_PASSWORD=Aa@222`

networks:

- `my_app`

volumes:

- `C:/data:/var/lib/mysql`

Deploy app Nodejs+MySQL

- Tạo docker-compose Nodejs services:

node:

image: my-node-app

environment:

- MYSQL_HOST: db
- MYSQL_USER: app
- MYSQL_PASSWORD: Aa@222

ports:

- 3000:3000

networks:

- my_app

depends_on:

- db

Deploy app Reactjs+Nodejs+MySQL

- Tạo Dockerfile để tạo ra các thành phần sau:
 - Container MySQL
 - Container Nodejs
 - Container Reactjs
 - Network (bridge) để các service trên kết nối vào mạng này
- Thực hành source code
 - <https://github.com/bezkoder/docker-compose-react-nodejs-mysql>
 - Xử lý lỗi CORS

```
app.use((req, res, next) => {
  res.setHeader("Access-Control-Allow-Origin", "*");
  res.setHeader(
    "Access-Control-Allow-Methods",
    "GET, POST, PUT, DELETE, OPTIONS"
  );
  res.setHeader(
    "Access-Control-Allow-Headers",
    "Origin, X-Requested-With, Content-Type, Accept, Authorization"
  );
  next();
});
```


Deploy app PHP+Apache+MySQL

Tạo docker file để tạo ra các thành phần sau:

- Container MySQL
- Container HTTP APACHE
- Container PHP-FPM
- Network (bridge) để các service trên kết nối vào mạng này
- Ánh xạ cổng của máy host vào cổng 80 của máy chủ HTTP
- Thực hành với source code
 - <https://github.com/yogesh174/docker-project>

Deploy app PHP+Apache+MySQL

- Tạo docker-compose Networks

name: myapp #là tên project

#Tạo mạng tên là my-network

networks:

my-network:

driver: bridge

Deploy app PHP+Apache+MySQL

- Tạo docker-compose PHP

```
# Tạo các dịch vụ (container)  
services:
```

```
#Tạo container my-php từ imgae php:latest có kết nối với mạng  
my-network
```

```
my-php:
```

```
    container_name: php-product
```

```
    image: php:latest
```

```
    hostname: php
```

```
    restart: always
```

```
    networks:
```

```
        - my-network
```

Deploy app PHP+Apache+MySQL

- Tạo docker-compose Httpd

#Tạo container my-httpd từ image httpd:latest có kết nối với mạng my-network, ánh xạ cổng 9999 của máy host vào cổng 80

my-httpd:

container_name: c-httpd01

image: httpd:latest

hostname: httpd

restart: always

networks:

- my-network

ports:

- "9999:80"
- "443:443"

Deploy app PHP+Apache+MySQL

- Tạo container my-mysql từ imgae mysql có kết nối với mạng my-network, config các biến môi trường

my-mysql:

container_name: mysql-product

image: mysql:latest

hostname: mysql

restart: always

networks:

- my-network

environment:

MYSQL_ROOT_PASSWORD=root

MYSQL_DATABASE=db_site

MYSQL_USER=sites

MYSQL_PASSWORD=Aa@222

Deploy app PHP+Apache+MySQL

- Build & run
 - docker-compose up
 - docker ps
 - docker network ls
 - docker-compose stop
 - docker-compose down
 - docker-compose logs [SERVICES]

THANK YOU