

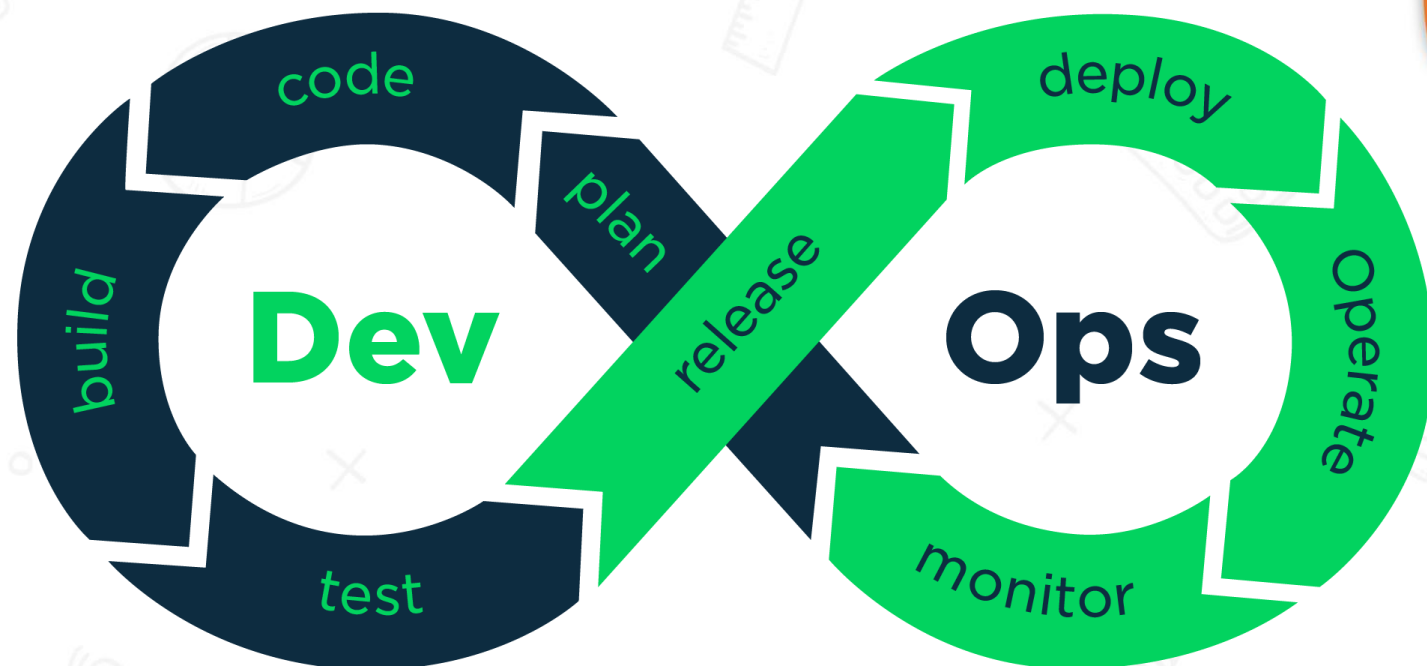
Môn học

DevOps

Giảng viên

Tan Do

0868880797



Nội dung buổi 05

- Docker Compose (phần 2)

Tạo ứng dụng my-node-app

Sử dụng nodejs express

Tạo app my-node-app init

- mkdir my-node-app
- cd my-node-app
- npm init -y
- npm install express

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

Tạo app my-node-app file Dockerfile

```
# Sử dụng image chính thức của Node.js làm base image
FROM node:14

# 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"]
```


Tạo app my-node-app file .dockerignore

node_modules

npm-debug.log

Dockerfile

.dockerignore

.git

.gitignore

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
```


Tạo app my-node-app Multi-stage Builds

Stage 1: Build

```
FROM node:14 AS build
```

```
WORKDIR /usr/src/app
```

```
COPY package*.json ./
```

```
RUN npm install
```

```
COPY . .
```

```
RUN npm run build
```

Stage 2: Production

```
FROM node:14-alpine
```

```
WORKDIR /usr/src/app
```

```
COPY --from=build /usr/src/app .
```

```
EXPOSE 3000
```

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

Tạo docker-compose Nodejs MySQL

Định nghĩa docker file để tạo ra các thành phần sau:

- Container MySQL
- Container Nodejs
- Network (bridge) để các service trên kết nối vào mạng này

Tạo docker-compose Networks

version: "3.9"

networks:

my_app:

Tạo docker-compose MySQL

services:

mysql:

image: mysql

environment:

- MYSQL_USER=tando
- MYSQL_PASSWORD=password

networks:

- my_app

volumes:

- ./data:/var/lib/mysql

Tạo docker-compose Nodejs

services:

node:

image: my-node-app

environment:

- MYSQL_HOST: mysql
- MYSQL_USER: tando
- MYSQL_PASSWORD: password

ports:

- "3000:3000"

networks:

- my_app

depends_on:

- mysql

Tạo docker-compose PHP Httpd MySQL

Định nghĩa 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 9999 của máy host vào cổng 80 của máy chủ HTTP

Tạo docker-compose Networks

version: "3.9" #là phiên bản docker compose

#Tạo mạng tên là my-network

networks:

my-network:

driver: bridge

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

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"

Tạo docker-compose MySQL

#Tạo container my-mysql từ image mysql:latest có kết nối với mạng my-network, config các biến môi trường

my-mysql:

container_name: myql-product

image: "mysql:latest"

hostname: mysql

restart: always

networks:

- my-network

environment:

- MYSQL_ROOT_PASSWORD=123abc
- MYSQL_DATABASE=db_site
- MYSQL_USER=sites
- MYSQL_PASSWORD=123abc

Tạo docker-compose PHP Httpd MySQL

Run docker compose:

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

THANK YOU