

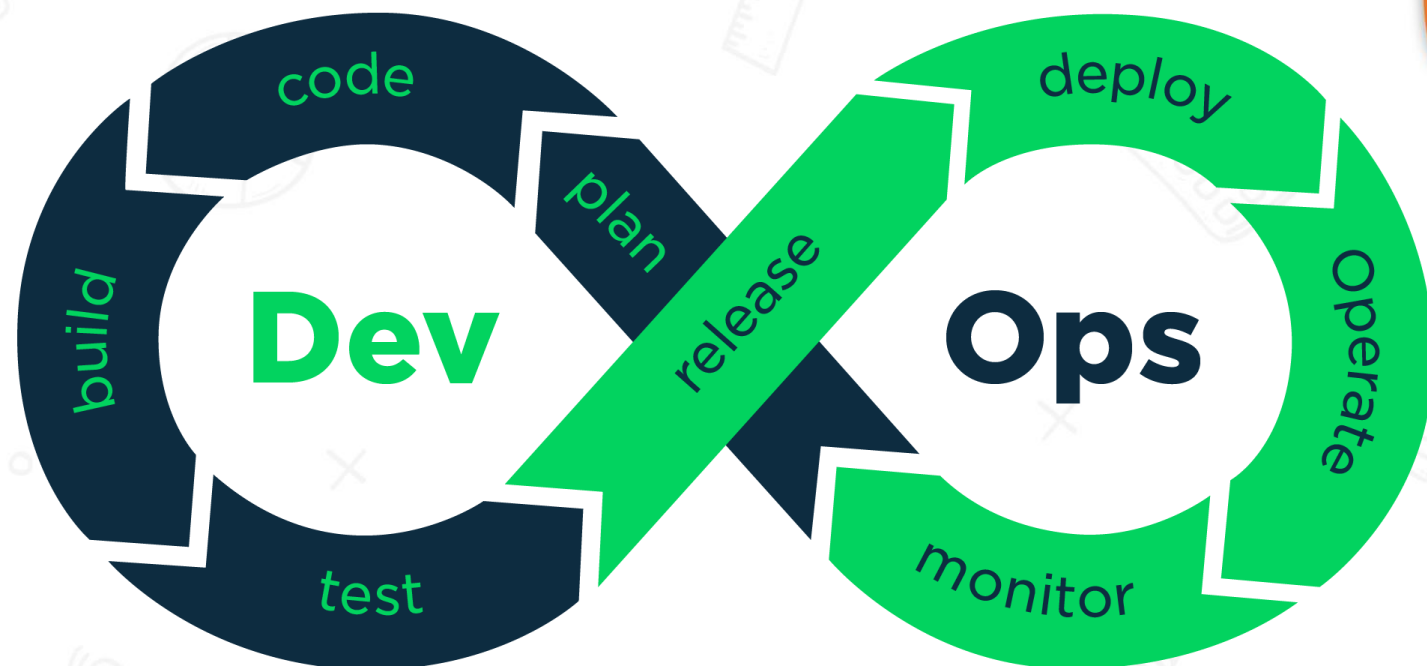
Môn học

DevOps

Giảng viên

Tan Do

0868880797



Nội dung buổi 06

- Docker Compose (phần 3)
 - Deploy app Nodejs+MySQL
 - Deploy app SpringBoot+MySQL
 - Deploy app Django+MySQL+NGINX
 - Truy cập và xem log các app

Deploy app Nodejs+MySQL

Mục tiêu là run 2 docker container:

- Nodejs container
- MySQL container
- Nodejs có thể giao tiếp với MySQL

Deploy app Nodejs+MySQL

- Tạo ứng dụng nodejs đơn giản

```
mkdir web && cd web
```

```
npm init -y
```

```
npm install express mysql
```

```
touch index.js
```

- Thực hành source code

- <https://github.com/bezkoder/docker-compose-nodejs-mysql>

Deploy app Nodejs+MySQL

- Nội dung file index.js

```
const express = require("express");
const mysql = require("mysql");
const app = express();

const connection = mysql.createPool({
  connectionLimit: 10,
  host: process.env.MYSQL_HOST || "localhost",
  user: process.env.MYSQL_USER || "root",
  password: process.env.MYSQL_PASSWORD || "password",
  database: process.env.MYSQL_DATABASE || "test",
});

app.get("/", (req, res) => {
  connection.query("SELECT * FROM Student", (err, rows) => {
    if (err) {
      res.json({
        success: false,
        err,
      });
    } else {
      res.json({
        success: true,
        rows,
      });
    }
  });
});

app.listen(5000, () => console.log("listening on port 5000"));
```

Deploy app Nodejs+MySQL

- Tạo Dockerfile

```
FROM node:22-alpine
```

```
WORKDIR /app
```

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

```
RUN npm install
```

```
COPY . .
```

```
EXPOSE 5000
```

```
CMD ["npm", "start"]
```


Deploy app Nodejs+MySQL

- Tạo Dockerfile cho MySQL

```
mkdir db && cd db  
touch testdump.sql  
touch Dockerfile
```

Deploy app Nodejs+MySQL

- Nội dung testdump.sql

```
CREATE TABLE Student(  
    student_id INT PRIMARY KEY AUTO_INCREMENT,  
    student_name VARCHAR(60),  
    student_age INT  
);  
INSERT INTO Student(  
    student_name,  
    student_age)  
VALUES(  
    "Shubham verma",  
    21  
);  
INSERT INTO Student(  
    student_name,  
    student_age)  
VALUES(  
    "Utkarsh verma",  
    23  
);  
ALTER USER 'root' IDENTIFIED WITH mysql_native_password BY 'password';  
flush privileges;
```


Deploy app Nodejs+MySQL

- Nội dung Dockerfile

```
FROM mysql
```

```
EXPOSE 3306
```

```
COPY ./testdump.sql /docker-entrypoint-initdb.d/
```

Deploy app Nodejs+MySQL

- Tạo file docker-compose.yml

```
name: myapp
services:
  db:
    build: ./db
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: test
    restart: always
  web:
    build: ./web
    environment:
      MYSQL_DATABASE: test
      MYSQL_USER: root
      MYSQL_PASSWORD: password
      MYSQL_HOST: db
    ports:
      - "5000:5000"
    depends_on:
      - db
    restart: on-failure
```

Deploy app Nodejs+MySQL

- Build & run
 - docker-compose up
 - http://localhost:5000/

Deploy app SpringBoot+MySQL

- Khởi tạo source code
 - <https://start.spring.io>
- Hoặc download example
 - <https://github.com/amilairoshan/SpringBoot-MySql-With-Docker-Compose>

Deploy app SpringBoot+MySQL

- Tạo Dockerfile

```
# Build stage
```

```
#
```

```
FROM maven:3.8.3-openjdk-17 AS build
```

```
COPY src /home/app/src
```

```
COPY pom.xml /home/app
```

```
RUN mvn -f /home/app/pom.xml clean package
```

```
EXPOSE 8080
```

```
ENTRYPOINT ["java", "-  
jar", "/home/app/target/spring_rest_docker.jar"]
```

Deploy app SpringBoot+MySQL

- Tạo docker-compose

```
name: myapp
services:
  api_service:
    build: .
    restart: always
    ports:
      - 8080:8080
    networks:
      - springapimysql-net
    environment:
      - spring.datasource.url=jdbc:mysql://mysqlb:3306/basics?allowPublicKeyRetrieval=true
    depends_on:
      - mysqlb

    volumes:
      - .m2:/root/.m2

  mysqlb:
    image: "mysql:8.0"
    restart: always
    ports:
      - 3306:3306
    networks:
      - springapimysql-net
    environment:
      MYSQL_DATABASE: basics
      MYSQL_USER: amila_one
      MYSQL_PASSWORD: Amila_pw
      MYSQL_ROOT_PASSWORD: Amila_Rpw
    networks:
      springapimysql-net:
```


Deploy app SpringBoot+MySQL

- Khởi database

```
CREATE TABLE IF NOT EXISTS persons(  
    `id` bigint(20) NOT NULL AUTO_INCREMENT,  
    `name` varchar(255) DEFAULT NULL,  
    PRIMARY KEY (`id`)  
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
INSERT INTO persons(name) VALUES('Amila');  
INSERT INTO persons(name) VALUES('Iroshan');
```

Deploy app Django+MySQL+NGINX

- Download source code
 - <https://github.com/yoshitakameguro/docker-django-uwsgi-nginx-mysql>

Deploy app Django+MySQL+NGINX

```
- docker-compose
  name: myapp
  services:
    web:
      container_name: web
      build: ./docker/nginx
      ports:
        - 8888:80
      volumes:
        - ./app:/var/www/html
      working_dir: /etc/nginx
      links:
        - python
    python:
      container_name: python
      build: ./docker/python
      volumes:
        - ./app:/var/www/html
      working_dir: /var/www/html
      depends_on:
        mysql:
          condition: service_healthy
    mysql:
      build: docker/mysql
      container_name: mysql
      ports:
        - 3306:3306
      volumes:
        - data-volume:/var/lib/mysql
      environment:
        - MYSQL_ROOT_PASSWORD=password
        - MYSQL_DATABASE=django
      healthcheck:
        test: "exit 0"
  volumes:
    data-volume:
```

Truy cập và xem log các app

- Show log a container
 - docker logs <tên container>
 - docker logs --follow <tên container>

THANK YOU