

# HƯỚNG DẪN CHI TIẾT LAB 03 MLOPS (DOCKER)

## BƯỚC 0: CHUẨN BỊ THƯ MỤC (LÀM ĐẦU TIÊN)

Để tránh lỗi, hãy tạo cấu trúc thư mục chuẩn ngay từ đầu.

1. Vào ổ đĩa **E:** (hoặc D, C tùy máy bạn).
2. Tạo thư mục tổng tên là: **Lab03\_MLOps**
3. Bên trong đó, tạo 3 thư mục con tên là:
  - **Bai3**
  - **Bai4**
  - **Bai5**

*(Bài 1 và Bài 2 chỉ gõ lệnh, không cần thư mục riêng)*

## BÀI 1: CÀI ĐẶT VÀ KIỂM TRA

**Nhiệm vụ:** Minh chứng Docker đã chạy.

1. Mở **PowerShell**.
2. Copy và chạy lệnh sau:  
PowerShell  
`docker --version`
3. Copy và chạy tiếp lệnh:  
PowerShell  
`docker run hello-world`
4. **Hành động:** Chụp màn hình kết quả (chứng minh có dòng "Hello from Docker!").

## BÀI 2: QUAY VIDEO THAO TÁC CONTAINER

**Nhiệm vụ:** Thao tác cơ bản với Nginx.

**Bật quay video màn hình và thực hiện lần lượt các lệnh sau trong PowerShell:**

1. Tải Image Nginx:

```
PowerShell  
docker pull nginx
```

2. Chạy Container (Map port 80 ra 8080):

```
PowerShell  
docker run -d -p 8080:80 --name my-web-server nginx  
(Mở trình duyệt, vào localhost:8080 để chứng minh nó đang chạy - tùy chọn)
```

3. Kiểm tra container đang chạy:

```
PowerShell  
docker ps
```

4. Dừng container:

```
PowerShell  
docker stop my-web-server
```

5. Xóa container:

```
PowerShell  
docker rm my-web-server
```

## BÀI 3: VIẾT VÀ CHẠY DOCKERFILE

**Nhiệm vụ:** Chạy một ứng dụng Python đơn giản.

### Bước 1: Tạo file

Vào thư mục **Lab03\_MLOps** -> vào tiếp **Bai3**. Tạo 2 file sau:

**File 1: app.py** (Mở Notepad dán vào, lưu lại tên app.py)

## Python

```
print("Hello, World!")  
print("This is a simple Python script.")  
print("It prints multiple lines to the console.")  
print("Goodbye, World!")
```

### File 2: Dockerfile (Lưu ý: Không có đuôi .txt)

## Dockerfile

```
FROM python:3.9-slim  
WORKDIR /app  
COPY app.py .  
CMD ["python", "app.py"]
```

## Bước 2: Thực hiện lệnh (Quay video)

Mở PowerShell tại thư mục Bai3 và chạy:

1. Build Image:  
PowerShell  
`docker build -t my-python-app .`
2. Chạy Container:  
PowerShell  
`docker run my-python-app`

(Kết quả phải in ra 4 dòng chữ tiếng Việt không dấu).

## BÀI 4: DOCKER COMPOSE

**Nhiệm vụ:** Chạy WordPress và MySQL.

## Bước 1: Tạo file

Vào thư mục **Lab03\_MLOps** -> vào tiếp **Bai4**.

**Tạo file:** docker-compose.yml

YAML

```
version: '3'
services:
  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: wordpress
    volumes:
      - db_data:/var/lib/mysql

  wordpress:
    image: wordpress:latest
    depends_on:
      - db
    ports:
      - "8000:80"
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: root
      WORDPRESS_DB_PASSWORD: password

volumes:
  db_data:
```

## Bước 2: Thực hiện lệnh (Quay video)

Mở PowerShell tại thư mục Bai4:

1. Chạy dịch vụ:  
PowerShell  
docker compose up -d
2. Kiểm tra:  
PowerShell  
docker compose ps
3. Mở trình duyệt (Chrome/Edge), vào địa chỉ: localhost:8000 (Thấy màn hình WordPress là thành công).
4. Tắt dịch vụ:  
PowerShell  
docker compose down

## BÀI 5: YOLOv7 DỰ ĐOÁN ẢNH (QUAN TRỌNG NHẤT)

**Nhiệm vụ:** Tải 128 ảnh và dùng AI nhận diện.

### Bước 1: Chuẩn bị dữ liệu tự động

Vào thư mục **Lab03\_MLOps** -> vào tiếp **Bai5**.

Tạo file: **get\_data.py**

Python

```
import os, zipfile, shutil
```

```

import urllib.request

print("--- DANG TAI DU LIEU COCO128 (Vui long cho)... ---")
url = 'https://github.com/ultralytics/yolov5/releases/download/v1.0/coco128.zip'
urllib.request.urlretrieve(url, 'coco128.zip')

print("--- DANG GIAI NEN... ---")
with zipfile.ZipFile('coco128.zip', 'r') as zip_ref:
    zip_ref.extractall(".")

print("--- DANG SAP XEP FILE... ---")
if not os.path.exists('images'):
    os.makedirs('images')

source_dir = os.path.join('coco128', 'images', 'train2017')
files = os.listdir(source_dir)

for f in files:
    shutil.move(os.path.join(source_dir, f), os.path.join('images', f))

# Dọn dẹp
shutil.rmtree('coco128')
os.remove('coco128.zip')
print("--- XONG! DA CO THU MUC IMAGES ---")

```

Chạy file này để lấy ảnh:

Tại PowerShell ở thư mục Bai5, gõ:

```
PowerShell
```

```
python get_data.py
```

(Sau khi chạy xong, bạn sẽ thấy thư mục images xuất hiện).

## Bước 2: Tạo Dockerfile cho YOLO

Vẫn ở thư mục Bai5, **tạo file Dockerfile:**

Dockerfile

# 1. Base Image Pytorch

FROM pytorch/pytorch:1.13.1-cuda11.6-cudnn8-runtime

# 2. Cài thư viện hệ thống

RUN apt-get update && apt-get install -y libgl1 libglib2.0-0 git wget

WORKDIR /app

# 3. Tải source code YOLOv7

RUN git clone https://github.com/WongKinYiu/yolov7.git

# 4. Cài thư viện Python

WORKDIR /app/yolov7

RUN pip install -r requirements.txt

# 5. Tải Weights (Model đã học)

RUN wget https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt

# 6. Điểm bắt đầu

ENTRYPOINT ["python", "detect.py"]

### Bước 3: Build và Run (Quay video kỹ phần này)

1. **Build Image** (Gõ lệnh và chờ khoảng 5-10 phút):

PowerShell

`docker build -t yolo-lab5 .`

2. Run Container (Dự đoán ảnh):

Copy chính xác lệnh dưới đây vào PowerShell (Lệnh này dùng \${PWD} nên máy nào cũng chạy được, không cần sửa đường dẫn):

PowerShell

`docker run --rm -v "Directory:/app/data" yolo-lab5 --source /app/data/images --project`

```
/app/data/output --name ket_qua --save-txt
```

## Bước 4: Kiểm tra kết quả (Minh chứng)

1. Chờ lệnh chạy xong (khoảng vài phút).
2. Vào thư mục trên máy tính: **Bai5** -> **output** -> **ket\_qua**.
3. Mở các ảnh lên để xem các khung hình nhận diện (người, xe, v.v...).
4. Chụp ảnh thư mục này đưa vào báo cáo.