

Dockerfile sử dụng Ubuntu 22.04 và cài đặt Python

 Danh mục: Lập trình

 Tác giả: duy vo

Tạo hoặc cập nhật tệp Dockerfile của bạn như sau:

```
# Sử dụng image chính thức của Ubuntu 22.04
FROM ubuntu:22.04

# Thiết lập biến môi trường để tránh các thông báo trong quá trình cài đặt
ENV DEBIAN_FRONTEND=noninteractive

# Cập nhật danh sách các gói và cài đặt các gói cần thiết
RUN apt-get update && \
    apt-get install -y python3 python3-pip && \
    apt-get clean

# Thiết lập thư mục làm việc trong container
WORKDIR /app

# Sao chép file yêu cầu vào container
COPY ./requirements.txt /app/requirements.txt

# Cài đặt các gói yêu cầu
RUN pip3 install --no-cache-dir --upgrade -r /app/requirements.txt

# Sao chép toàn bộ mã nguồn vào container
COPY . /app

# Lệnh để chạy ứng dụng FastAPI sử dụng Uvicorn
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

Multi-stage builds

Sử dụng multi-stage builds trong Dockerfile có thể giúp tối ưu kích thước của image bằng cách chỉ giữ lại các phần cần thiết cho môi trường sản xuất. Dưới đây là cách cải tiến Dockerfile của bạn bằng cách sử dụng multi-stage builds:

```
# Sử dụng image chính thức của Ubuntu 22.04
# Giai đoạn build
FROM ubuntu:22.04 AS builder

# Thiết lập biến môi trường để tránh các thông báo trong quá trình cài đặt
ENV DEBIAN_FRONTEND=noninteractive

# Cập nhật danh sách các gói và cài đặt các gói cần thiết
RUN apt-get update && \
    apt-get install -y python3 python3-pip && \
    apt-get clean

# Thiết lập thư mục làm việc trong container
WORKDIR /app

# Sao chép file yêu cầu vào container
COPY ./requirements.txt /app/requirements.txt

# Cài đặt các gói yêu cầu
RUN pip3 install --no-cache-dir --upgrade -r /app/requirements.txt

# Sao chép toàn bộ mã nguồn vào container
COPY . /app

#####

# Giai đoạn sản xuất
FROM ubuntu:22.04

# Thiết lập biến môi trường để tránh các thông báo trong quá trình cài đặt
ENV DEBIAN_FRONTEND=noninteractive

# Cập nhật danh sách các gói và cài đặt các gói cần thiết cho runtime
RUN apt-get update && \
    apt-get install -y python3 && \
    apt-get clean

# Thiết lập thư mục làm việc trong container
WORKDIR /app

# Sao chép các gói cài đặt từ giai đoạn build
COPY --from=builder /usr/local/lib/python3.10/dist-packages /usr/local/lib/python3.10/dist-packages
COPY --from=builder /usr/local/bin /usr/local/bin

# Sao chép mã nguồn vào container
COPY . /app

# Lệnh để chạy ứng dụng
CMD ["python3", "run.py"]
```

Xây dựng Docker Image

Mở terminal và điều hướng đến thư mục chứa các tệp của bạn (`main.py`, `Dockerfile`, `requirements.txt`). Sau đó chạy lệnh sau để xây dựng Docker image:

```
docker build -t myfastapiapp .
```

Chạy Docker Container

```
docker run -d --name myfastapiapp -p 8000:8000 myfastapiapp
```

Sao lưu Docker Image

Lưu Docker Image thành tệp tar:

```
docker save -o myfastapiapp_backup.tar myfastapiapp
```

- `-o myfastapiapp_backup.tar`: Chỉ định tên và đường dẫn của tệp tar mà bạn muốn lưu.
 - `myfastapiapp`: Tên của Docker image mà bạn muốn sao lưu.
- Kiểm tra tệp sao lưu:** Sau khi lệnh `docker save` hoàn thành, bạn sẽ có một tệp tar chứa Docker image của bạn. Bạn có thể di chuyển, sao chép, hoặc lưu trữ tệp này ở bất kỳ đâu bạn muốn.

Khôi phục Docker Image từ tệp sao lưu

Tải Docker Image từ tệp tar:

Sử dụng lệnh `docker load` để tải Docker image từ tệp tar. Ví dụ:

```
docker load -i myfastapiapp_backup.tar
```

Kiểm tra Docker Image đã khôi phục:

```
docker images
```

