

# Lab - Chuyển chương trình Python Flask vào Docker Container

o0o

## I. Mô tả

- Sơ đồ bài thực hành này gồm máy Workstation cài đặt Docker
- Tạo một thư mục bao gồm 3 file:

+ app.py (Chứa nội dung chương trình Python Flask).

+ requirements.txt (Chứa nội dung thư viện cần cài đặt)

+ Dockerfile (Chứa nội dung cấu hình của Docker).

## II. Yêu cầu kỹ thuật

- Học viên thực hiện viết chương trình Python Flask cơ bản in ra tên của mình.
- Học viên thực hiện viết cấu hình Dockerfile để chuyển chương trình Python Flask vào Docker Container với port 5000.
- Kiểm tra kết quả

## III. Các bước thực hiện

Bước 1: Viết chương trình Python Flask cơ bản

Viết chương trình Python Flask cơ bản đặt tên là “app.py” có nội dung như sau:

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def index():
    return "Chào mừng đến với trung tâm tin học VnPro"
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

```
root@localhost:~/Documents/cau6
File Edit View Search Terminal Help
from flask import Flask
app = Flask(__name__)
@app.route('/')
def home_page():
    return " Hello Final Devnet Lab "
if __name__ == '__main__':
    app.run(host='0.0.0.0')
~
~
```

## Bước 2: Chuẩn bị file requirements.txt

File requirements.txt dùng để cài đặt các thư viện Python sử dụng trong chương trình. Vì ở đây sử dụng mỗi thư viện Flask nên nội dung file “requirements.txt” có nội dung như sau:

```
Flask==2.0.1
```

## Bước 3: Chuẩn bị file cấu hình Docker Container với tên là Dockerfile

Đầu tiên cần xác định images để sử dụng, trong bài này sẽ sử dụng images python:3.7-alpine. Alpine là một Distro lightweight thường được sử dụng trong việc build container

Nội dung của Dockerfile như sau:

```
FROM python:3.7-alpine
RUN apk add --nocache gcc musl-dev linux-headers
```

Giải thích:

- FROM: chỉ định image gốc
- RUN: thực thi lệnh. Ở đây là apk là package manager của alpine.

Sau khi tạo xong Dockerfile, tiến hành build images. Khi đó ta cần để chung 3 file vừa tạo cùng 1 thư mục sau đó di chuyển đến thư mục chứa 3 file đó và thực thi

```
bash: l: command not found...
[root@localhost cau6]# ls
app.py Dockerfile requirements.txt
[root@localhost cau6]#
```

Sử dụng lệnh “docker build -t flask\_app:latest .” (lưu ý có dấu chấm).

```
[root@localhost cau6]# docker build -t flask_app:latest .
Sending build context to Docker daemon 4.096kB
Step 1/2 : FROM python:3.7-alpine
--> 04ffff590161
Step 2/2 : RUN apk add --no-cache gcc musl-dev linux-headers
--> Running in e4facf157ca2
fetch https://dl-cdn.alpinelinux.org/alpine/v3.16/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.16/community/x86_64/APKINDEX.tar.gz
(1/12) Installing libgcc (11.2.1_git20220219-r2)
(2/12) Installing libstdc++ (11.2.1_git20220219-r2)
(3/12) Installing binutils (2.38-r2)
(4/12) Installing libgomp (11.2.1_git20220219-r2)
(5/12) Installing libatomic (11.2.1_git20220219-r2)
(6/12) Installing gmp (6.2.1-r2)
(7/12) Installing isl22 (0.22-r0)
(8/12) Installing mpfr4 (4.1.0-r0)
(9/12) Installing mpc1 (1.2.1-r0)
(10/12) Installing gcc (11.2.1_git20220219-r2)
(11/12) Installing linux-headers (5.16.7-r1)
(12/12) Installing musl-dev (1.2.3-r0)
Executing busybox-1.35.0-r13.trigger
OK: 135 MiB in 48 packages
Removing intermediate container e4facf157ca2
--> 2893c234f941
Successfully built 2893c234f941
Successfully tagged flask_app:latest
[root@localhost cau6]#
```

Khi đó trong “docker images” sẽ có images mới vừa tạo:

```
[root@localhost cau6]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
flask_app           latest         2893c234f941   6 minutes ago  166MB
```

Khởi chạy docker images mới vừa tạo bằng lệnh “docker run -it flask\_app:latest /bin/sh”:

```
[root@localhost cau6]# docker run -it flask_app:latest /bin/sh
/#
```

Sau khi chạy lệnh trên sẽ vào được mode shell của Container flask\_app. Nhập “exit” để thoát.

Tiếp theo bổ sung Dockerfile:

```
FROM python:3.7-alpine
RUN apk add --no-cache gcc musl-dev linux-headers
WORKDIR /app
COPY ./app
```

Giải thích:

- WORKDIR: Sẽ tạo ra một thư mục và cd tới đó
- COPY: Sẽ copy các file trong thư mục build vào trong Docker Container.

Sau khi bổ sung xong, tiến hành lưu lại và chạy lệnh “docker build -t flask\_app:latest .” (lưu ý có dấu chấm)

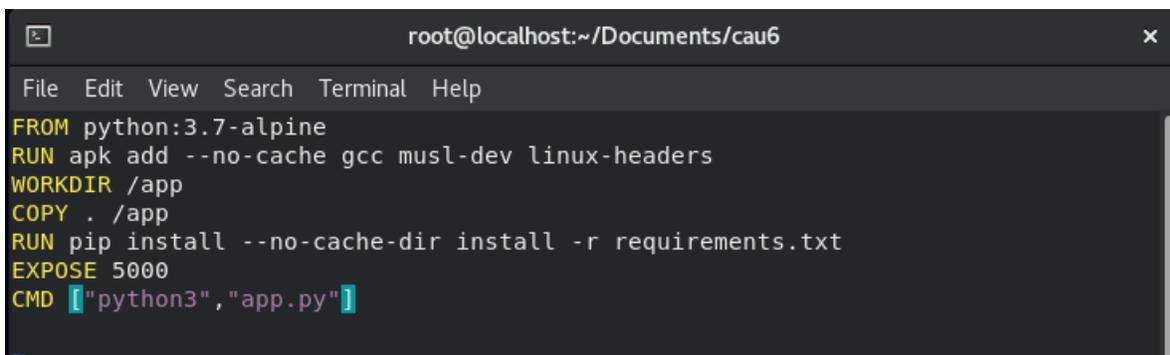
```
[root@localhost cau6]# ls
app.py Dockerfile requirements.txt
[root@localhost cau6]# vim Dockerfile
[root@localhost cau6]# docker build -t flask_app:latest .
Sending build context to Docker daemon 4.096kB
Step 1/4 : FROM python:3.7-alpine
--> 04ffff590161
Step 2/4 : RUN apk add --no-cache gcc musl-dev linux-headers
--> Using cache
--> 2893c234f941
Step 3/4 : WORKDIR /app
--> Running in 40ebf07c14af
Removing intermediate container 40ebf07c14af
--> e39611d80f58
Step 4/4 : COPY ./app
--> a9c45039db23
Successfully built a9c45039db23
Successfully tagged flask_app:latest
[root@localhost cau6]#
```

Tiếp theo, vào mode shell của flask\_app bằng lệnh “docker run -it flask\_app:latest /bin/sh” để kiểm tra:

```
[root@localhost cau6]# docker run -it flask_app:latest /bin/sh
/app # ls
Dockerfile          app.py              requirements.txt
/app # exit
[root@localhost cau6]#
```

Khác với khi nãy, bây giờ container flask\_app có thêm thư mục app và chứa 3 file ban đầu khởi tạo. Nhập “exit” để thoát.

Tiếp đến bổ sung Dockerfile để cài đặt thư viện cần thiết và chạy chương trình app.py:



```
root@localhost:~/Documents/cau6
File Edit View Search Terminal Help
FROM python:3.7-alpine
RUN apk add --no-cache gcc musl-dev linux-headers
WORKDIR /app
COPY . /app
RUN pip install --no-cache-dir install -r requirements.txt
EXPOSE 5000
CMD ["python3", "app.py"]
```

Giải thích:

- EXPOSE: Khai báo port lắng nghe của images.
- CMD: Cung cấp một lệnh và đối số cho container thực thi.

Sau khi bổ sung xong, tiến hành lưu lại và chạy lệnh “docker build -t flask\_app:latest .” (lưu ý có dấu chấm)

Tiến hành khởi chạy bằng lệnh “docker run -it -d -p 5000:5000 flask\_app:latest”

```
[root@localhost cau6]# docker run -it -d -p 5000:5000 flask_app:latest
828a6fef9848c110bb2f60c1462aa4735fad97fc85d79ad9a2e183bcdcf937ee
[root@localhost cau6]#
```

Kiểm tra:

Sau khi chạy thành công, sử dụng browser để kiểm tra. Truy vào địa chỉ localhost:5000



CÔNG TY TNHH TƯ VẤN VÀ DỊCH VỤ CHUYÊN VIỆT  
TRUNG TÂM TIN HỌC VNPRO

ĐC: 276 - 278 Ung Văn Khiêm, P. Thanh Mỹ Tây, Tp. Hồ Chí Minh  
ĐT: (028) 35124257 | Hotline: 0933427079 Email: vnpro@vnpro.org

