HƯỚNG DẪN THỰC HÀNH DOCKER

1. Copy thư mục flask_app vào máy ảo
   scp -r C:\flask_app\ osboxes@192.168.1.28:

```
C:\>scp -r C:\flask_app\ osboxes@192.168.1.28:
osboxes@192.168.1.28's password:
app.py                        100%  900   451.0KB/s   00:00
Dockerfile                    100%  277   137.2KB/s   00:00
requirements.txt              100%   13     4.3KB/s   00:00
index.html                    100%  509   252.8KB/s   00:00
```

2. Tiến hành ssh vào máy ảo

```
C:\>ssh osboxes@192.168.1.28
osboxes@192.168.1.28's password:
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-10-generic x86_64)
```

3. Chuyển thư mục hiện hành về flask_app trên máy ảo

```
osboxes@osboxes:~$ cd /home/osboxes
osboxes@osboxes:~$ ls
flask_app
osboxes@osboxes:~$ cd flask_app
osboxes@osboxes:~/flask_app$ ls
app.py  Dockerfile  requirements.txt  templates
```

   Khi đó, cấu trúc thư mục là

```
$ tree
.
├── app.py
├── requirements.txt
└── templates
    └── index.html
```

4. Chỉnh sửa file Dockerfile thành

```
osboxes@osboxes:~/flask_app$ ls
app.py  Dockerfile  requirements.txt  templates
osboxes@osboxes:~/flask_app$ sudo nano Dockerfile
```

```
FROM alpine:edge

RUN apk add --update py3-pip

COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt --break-system-packages

COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

EXPOSE 5000

CMD ["python3", "/usr/src/app/app.py"]
```

--break-system-packages

5. Chỉnh sửa file requirements.txt thành

```
osboxes@osboxes:~/flask_app$ sudo nano requirements.txt
```

```
Flask==2.0.3
Werkzeug==2.2.2
```

6. Tiến hành build image docker cho flask_app

```
osboxes@osboxes:~/flask_app$ osboxes@osboxes:~/flask_app$ sudo docker build -t testapp .
[+] Building 13.2s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 341B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/alpine:edge
 => [1/6] FROM docker.io/library/alpine:edge@sha256:9f867dc20de5aa9690c5ef6c2c81ce35a918c0007f6eac27df90d3166eaa5cc0
 => [internal] load build context
 => => transferring context: 155B
 => CACHED [2/6] RUN apk add --update py3-pip
 => [3/6] COPY requirements.txt /usr/src/app/
 => [4/6] RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt --break-system-packages
 => [5/6] COPY app.py /usr/src/app/
 => [6/6] COPY templates/index.html /usr/src/app/templates/
 => exporting to image
 => => exporting layers
 => => writing image sha256:d5daa95115085564825c939fb7d5ba84d7d602b9b05fabf31cfb7fa07c81606e
 => => naming to docker.io/library/testapp
```

7. Kiểm tra xem image đã được tạo thành công chưa

```
osboxes@osboxes:~/flask_app$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
testapp       latest    5bbce1ac50ba   7 minutes ago    83.4MB
```

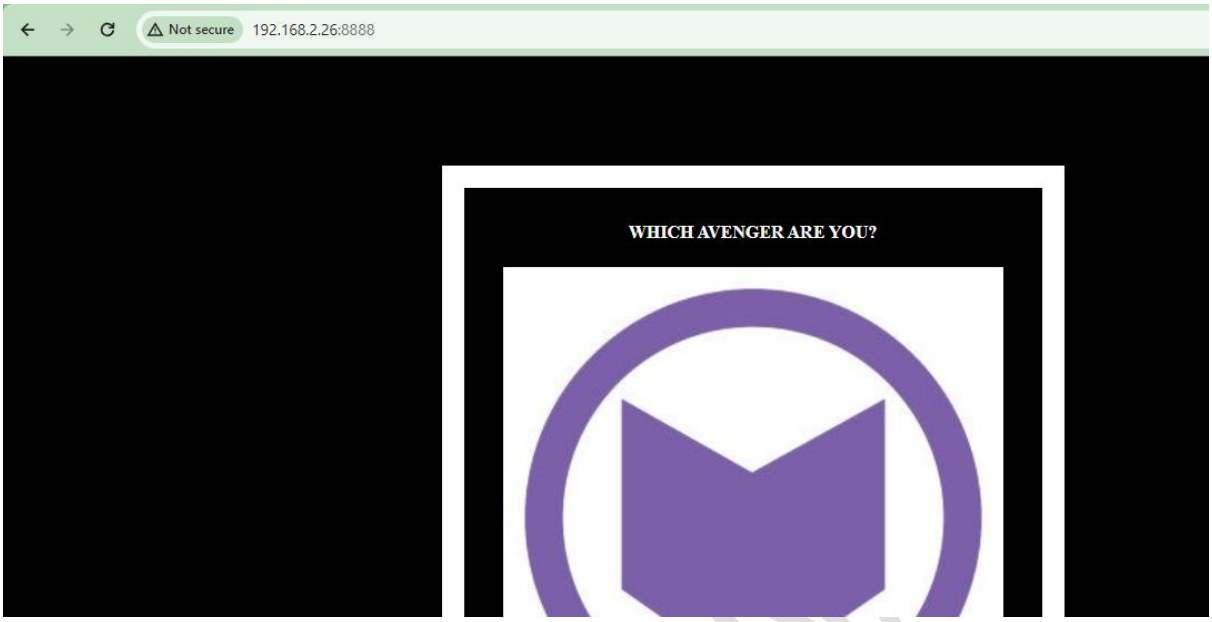
8. Kiểm tra thông tin rõ hơn về image

```
osboxes@osboxes:~/flask_app$ sudo docker image inspect testapp
```

9. Chạy thử ứng dụng trên image trong Docker

```
osboxes@osboxes:~/flask_app$ sudo docker container run -p 8888:5000 testapp
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
```

10. Kiểm tra kết quả ứng dụng trên máy thật



11. Đăng nhập vào Docker Hub

```
osboxes@osboxes:~$ sudo docker login
[sudo] password for osboxes:
Log in with your Docker ID or email address to push and pull images from Docker Hub
te one.
You can log in with your password or a Personal Access Token (PAT). Using a limited
. Learn more at https://docs.docker.com/go/access-tokens/

Username: ismartio
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

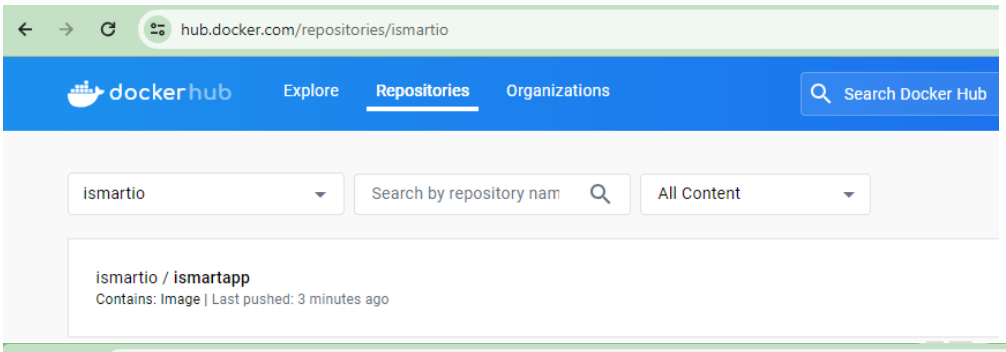
12. Tag (đánh dấu) image đã tạo trong máy local lên Docker Hub Registry

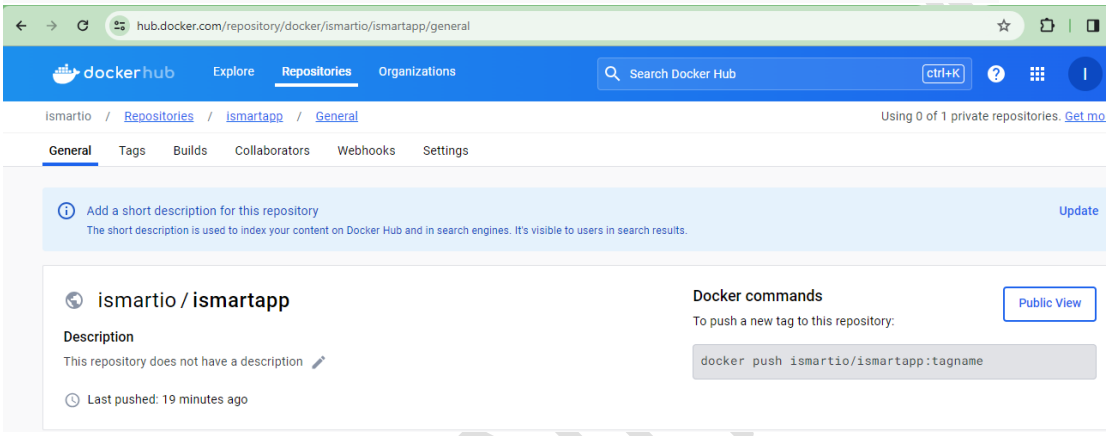
```
osboxes@osboxes:~$ sudo docker tag testapp ismartio/ismartapp:example
```

13. Tiến hành Publishing image đã tạo lên Docker Hub Registry

```
osboxes@osboxes:~$ sudo docker push ismartio/ismartapp:example
The push refers to repository [docker.io/ismartio/ismartapp]
46b804d69a6b: Pushed
e75b6dba0ae3: Pushed
d8e0e31bc648: Pushed
92597a8c1666: Pushed
02fb3eeac15a: Pushed
1bb475414a7e: Mounted from library/alpine
example: digest: sha256:5128e49558f83f511c195bd2c2eef52b2a9ad2ef7f768e244c94029d50338e09 size: 1572
```

14. Kiểm tra xem image đã được publishing thành công trên Docker Hub chưa





15. Sau khi đã thấy image đã Publishing thành công, ta có thể kéo (pull) image về chạy trên mọi host hay server hay cloud system. Để thực hiện bài test này, chúng ta sẽ làm các bước sau
    a. Kiểm tra xem image ismartapp cũ có tồn tại trên docker container không, nếu có thì xóa image đó đi

```
osboxes@osboxes:~$ sudo docker images
[sudo] password for osboxes:
REPOSITORY          TAG       IMAGE ID       CREATED         SIZE
testapp             latest    829438d0a175   10 hours ago    83.5MB
ismartio/ismartapp  example   829438d0a175   10 hours ago    83.5MB
```

```
osboxes@osboxes:~$ sudo docker image rm -f 829438d0a175
Untagged: ismartio/ismartapp:example
Untagged: ismartio/ismartapp@sha256:5128e49558f83f511c195bd2c2eef52b2a9ad2ef7f768e244c94029d50338e09
Untagged: testapp:latest
Deleted: sha256:829438d0a1751926e6107f9e0475bbdfca51d136a6873f1346c8dca8cbdb6ef0
osboxes@osboxes:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
```

    b. Tiến hành pull image từ Docker Hub Registry về và run

```
osboxes@osboxes:~$ sudo docker run -p 8888:5000 ismartio/ismartapp:example
Unable to find image 'ismartio/ismartapp:example' locally
example: Pulling from ismartio/ismartapp
dcccee43ad5d: Already exists
1b84e8deb04a: Already exists
755f0a79b188: Already exists
c15a69b38c41: Already exists
6875c8331407: Already exists
7cb5b2fd892d: Already exists
Digest: sha256:5128e49558f83f511c195bd2c2eef52b2a9ad2ef7f768e244c94029d50338e09
Status: Downloaded newer image for ismartio/ismartapp:example
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
```