Docker Commands for Lab 4

Richard Bradt

Build Docker Container

$ docker build -t lab4_hello:v6 .

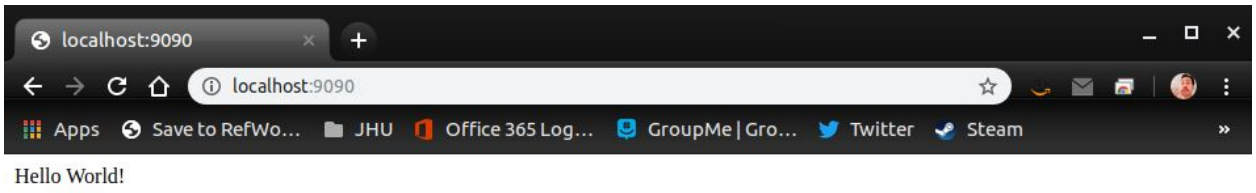
```
dix@dix-ubuntu:cloud-sec-2019-04$ docker build -t lab4_hello:v6 .
Sending build context to Docker daemon  4.608kB
Step 1/7 : FROM python
 ---> 02d2bb146b3b
Step 2/7 : RUN mkdir /app
 ---> Using cache
 ---> 1b50f2e49e77
Step 3/7 : COPY server.py /app/
 ---> 6a0fa4ab7884
Step 4/7 : WORKDIR /app
 ---> Running in 1871e67b49d9
Removing intermediate container 1871e67b49d9
 ---> a4d9986907e2
Step 5/7 : RUN pip install flask
 ---> Running in 6bdd18b58023
Collecting flask
  Downloading https://files.pythonhosted.org/packages/9b/93/628509b8d5dc749656a9641f4caf13540e2cdec852769
64ff8f43bbb1d3b/Flask-1.1.1-py2.py3-none-any.whl (94kB)
Collecting Jinja2>=2.10.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/1d/e7/fd8b501e7a6dfe492a433deb7b9d833d39ca74916fa8b
c63dd1a4947a671/Jinja2-2.10.1-py2.py3-none-any.whl (124kB)
Collecting click>=5.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/fa/37/45185cb5abbc30d7257104c434fe0b07e5a195a684750
6c074527aa599ec/Click-7.0-py2.py3-none-any.whl (81kB)
Collecting Werkzeug>=0.15 (from flask)
  Downloading https://files.pythonhosted.org/packages/ce/42/3aeda98f96e85fd26180534d36570e4d18108d62ae36f
87694b476b83d6f/Werkzeug-0.16.0-py2.py3-none-any.whl (327kB)
Collecting itsdangerous>=0.24 (from flask)
  Downloading https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a
4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10.1->flask)
  Downloading https://files.pythonhosted.org/packages/98/7b/ff284bd8c80654e471b769062a9b43cc5d03e7a615048
d96f4619df8d420/MarkupSafe-1.1.1-cp37-cp37m-manylinux1_x86_64.whl
Installing collected packages: MarkupSafe, Jinja2, click, Werkzeug, itsdangerous, flask
Successfully installed Jinja2-2.10.1 MarkupSafe-1.1.1 Werkzeug-0.16.0 click-7.0 flask-1.1.1 itsdangerous-
1.1.0
Removing intermediate container 6bdd18b58023
 ---> 742a6f6bd5f2
Step 6/7 : EXPOSE 8080
 ---> Running in 7b5455b65490
Removing intermediate container 7b5455b65490
 ---> 162064fc8da3
Step 7/7 : CMD ["python", "server.py"]
 ---> Running in e54a4cae4bb2
Removing intermediate container e54a4cae4bb2
 ---> 8e4578a9ef99
Successfully built 8e4578a9ef99
Successfully tagged lab4_hello:v6
dix@dix-ubuntu:cloud-sec-2019-04$
```

Run image binding external port 9090 to container port 8080. Server code is executed when container is ran.

$ docker run -d --name=lab4_sub -p 9090:8080 lab4_hello:v6

```
dix@dix-ubuntu:cloud-sec-2019-04$ docker run -d --name=lab4_sub -p 9090:8080 lab4_hello:v6
8f25545838ff01945e43b07c995d11f7de53869ed036f5c33570a7f7ab480978
dix@dix-ubuntu:cloud-sec-2019-04$
```

Contact server from my host browser (Using port 9090).



Allow kernel to bind port to container port 8080. Use bash script to get the assigned port.

$ docker run -d --name=lab4_script -p 0:8080 lab4_hello:v6
$ ./getport.sh lab4_script

```
dix@dix-ubuntu:cloud-sec-2019-04$ docker run -d --name=lab4_script -p 0:8080 lab4_hello:v6
74e7af3101123bd40aa58a32a57ce16323e12cdedd6eebe43ee143ac809ade2b
dix@dix-ubuntu:cloud-sec-2019-04$ ./getport.sh lab4_script
Got container name: lab4_script
IP: localhost
Port: 32771
URL is http://localhost:32771
dix@dix-ubuntu:cloud-sec-2019-04$
```

Contact server using the provided URL.