**Cloud 300**

Optum

Student Exercise Manual

# Lab 4.1: Dockerfile

In this app, we will create a Python app using flask, and deploy it using docker.

Here is the Dockerfile we are going to use

# The base image  
FROM ubuntu:latest  
  
# Install python and pip  
RUN apt-get update -y  
RUN apt-get install -y python-pip python-dev build-essential  
  
# Install Python modules needed by the Python app  
COPY requirements.txt /usr/src/app/  
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt  
  
# Copy files required for the app to run  
COPY app.py /usr/src/app/  
  
# Declare the port number the container should expose  
EXPOSE 5000  
  
# Run the application  
CMD ["python", "/usr/src/app/app.py"]

Note what we're going to do. We are going to start with ubuntu.

cd myfirstapp  
docker build -t myfirstapp

You will get a very long output, which will be Docker loading all of your Dockerfile commands onto your container.

You should look at your Dockerfile

Sending build context to Docker daemon 4.096kB  
Step 1/8 : FROM ubuntu:latest  
latest: Pulling from library/ubuntu  
a48c500ed24e: Pull complete  
1e1de00ff7e1: Pull complete  
0330ca45a200: Pull complete  
471db38bcfbf: Pull complete  
0b4aba487617: Pull complete  
Digest: sha256:c8c275751219dadad8fa56b3ac41ca6cb22219ff117ca98fe82b42f24e1ba64e  
Status: Downloaded newer image for ubuntu:latest  
 ---> 452a96d81c30  
Step 2/8 : RUN apt-get update -y  
 ---> Running in 84d9f73bfc4f

I've snipped this for clarity, but pay attention to what's going on.

## List your containers

docker ls

REPOSITORY TAG IMAGE ID CREATED SIZE  
myfirstapp latest b2959a4bbf48 Less than a second ago 462MB  
ubuntu latest 452a96d81c30 3 weeks ago 79.6MB

Notice myfirstapp is fairly large (462MB). This is because we have a full-on ubuntu system here. We may not need that.

## Run the container

docker container run -p 5000:5000 --name myfirstapp myfirstapp

This will run our app. console output should look like the following

\* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)  
172.17.0.1 - - [24/May/2018 15:55:11] "GET / HTTP/1.1" 200 -  
172.17.0.1 - - [24/May/2018 15:55:13] "GET /favicon.ico HTTP/1.1" 404 -

## Go to browser

Open your browser and go to YOURMACHINE:5000. If you are running on localhost, then go to localhost:5000

You should see something like the following in your browser:

Hello! This is my Flask app.

This indicates your Flask app is running properly. You can now close your container by typing control-c