

## HW6: Programming Assignment v11.08.2021 4PM

# WORKING WITH DOCKER CONTAINERS AND MICROSERVICES

In this assignment you will create two microservices, a server and a client, that will be deployed in a docker container. Then you will automate their deployment. Completing this assignment will give you a feel for how many companies can now run modern containerized micro web services.

Due Date: Oct 11<sup>th</sup>, 2021

## 1. Description of Task

For this assignment you will be working two build and automate the deployment of two programs (microservices) which will require the following components. For this assignment you will deploy Docker on your personal computer. Before you get started you will need to download Docker here: <https://docs.docker.com/get-docker/> Docker works with Windows, Mac and Linux.

The server will return the square root of int n.

**Server:** Basic python server. [10]

1. Minimal amount of code (~10 lines)
2. Listens on port 5000
3. Returns "The square root of n is: [value of  $n^{0.5}$ ]" when a request is received
4. This file should be inside a folder called app

**Docker Container for Server:** Docker container that contains the python server. [10]

1. Create a Dockerfile (script that specifies how container will be built) (~6 lines of text)
2. This Dockerfile will provide the commandline argument of 'n' to server.py
3. Create a requirements.txt file (specify python dependencies for server)
4. Put both of these files and the app folder with the python server code in a folder called server

**Client:** Basic python server. [10]

1. Minimal amount of code (~9 lines of text)
2. Does a GET request on <http://localhost:5000> to get data from server
3. Prints message received from the server
4. This file should be inside a folder called app

**Docker Container for Client:** Docker container that contains the python server. [10]

1. Create a Dockerfile (script that specifies how container will be built) (~6 lines of text)
2. Create a requirements.txt file that specifies the python dependencies for client (This is not necessary but depends on your implementation).
3. Put both files and the app folder with the python client code in a folder called client

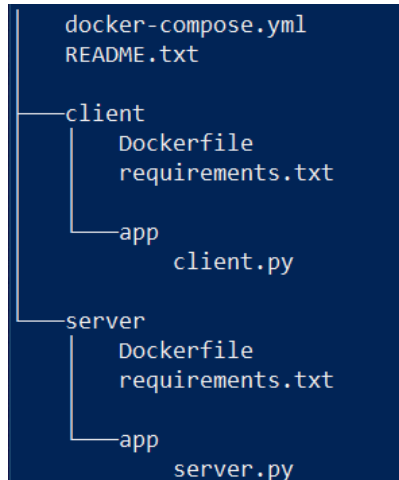
**Docker Compose:** Specifications for deploying the client and server containers. [10]

1. Create docker-compose.yml
2. Minimal amount of code (~11 lines of text)
3. Create a folder called hw6 that contains the docker-compose.yml as well as the server and client folders created earlier
4. When run with: 'docker-compose up' both client and server containers should be built and run which will cause the server to return the square-root to the client when it requests it and then the client will display the text.

**README.txt** [40]

Don't forget to implement README.txt parallel to your docker-compose.yml, the HW6 folder structure should be like the diagram below.

**HW6** folder structure:



**Hints:**

1. Answer the questions in section 4 first to get a feeling for how docker works. You can find the answers by looking through dockers website <https://docs.docker.com/>, typing 'docker' into the command line to look at the built in help docs, or googling the question.
2. This assignment will be mostly about getting comfortable with the different components of the docker ecosystem, the actual code will be minimal and many of the components can be found already completed in Dockers documentation or on different internet sites. Putting everything together is what will take a bit of work.
3. Starting points for the code in this program can be found in several places on dockers websites as well as the internet in general. Here are some search terms you might want to try besides the obvious, ex. dockerfile, docker-compose, requirements.txt, etc.
  - a. docker python server
  - b. python get request
  - c. python print response text
  - d. docker-compose network host
4. Use Google / search for any error messages to address them.

## 2. Task Requirements

1. Answer the questions in Part 4 (40% of your grade)
2. Complete the components described in Part 1 (50% of your grade)
3. When the command docker-compose up is run output similar to the following should be displayed. Because this project has some flexibility the output does not have to be identical but the following items in the output should be visible: output related to the server container being build, output related to the client being built, output related to the server running (highlighted yellow), output from the client displaying the message received from the server (highlighted red). (10% of your grade)

### 3. Files Provided

This writeup is the only file provided with this assignment.

**4. Questions:** Include the answers to the following short answer questions in a file called README.txt that you will create and included in your hw6 folder.

1. What is a docker container? [2]
2. What is the difference between a container and a virtual machine? [2]
3. What is the purpose of a Dockerfile? [2]
4. What is the purpose of a requirements.txt file? [2]
5. What is the purpose of a docker-compose.yml file? [2]
6. What is the difference between a docker image and docker container? [2]
7. What command can be used to create an image from a Dockerfile? [2]
8. What command will start a docker container? [2]
9. What command will stop a docker container? [2]
10. What command will remove a docker container? Image? [5]
11. What command will list all running docker containers? all containers? [5]
12. What command will list all docker images? [2]
13. What command do you use to deploy docker containers using information in the docker-compose.yml file? [2]
14. How can you specify in the docker-compose.yml file that you want docker containers to use the hosts network? [5]
15. How can you specify in the docker-compose.yml file where the Dockerfile for a particular container is found? [5]

**5. Example Outputs** (Note:- The names of servers and client may be different, the ids of containers can be different on your system, However the IMPORTANT line required for output are highlighted.)

```
C:\Users\user\hw6$ docker-compose up --build
```

[+] Building 26.6s (15/15) FINISHED

**A**

The output lines here are removed, because it displays code.

The outputs here contain container ids and each step from your docker file.

Successfully tagged docker\_client:latest

docker\_server\_1 is up-to-date

Starting docker\_client\_1

Attaching to docker\_server\_1, docker\_client\_1

server\_1 | Reading value 5

server\_1 | \* Serving Flask app "server" (lazy loading)

server\_1 | \* Environment: production

server\_1 | WARNING: This is a development server. Do not use it in a production deployment.

server\_1 | Use a production WSGI server instead.

server\_1 | \* Debug mode: off

server\_1 | \* Serving Flask app "server" (lazy loading)

server\_1 | \* Environment: production

server\_1 | WARNING: This is a development server. Do not use it in a production deployment.

server\_1 | Use a production WSGI server instead.

server\_1 | \* Debug mode: off

server\_1 | \* Running on http://172.18.0.2:5000/ (Press CTRL+C to quit)

server\_1 | 172.18.0.1 - - [29/Oct/2021 21:29:54] "GET / HTTP/1.1" 200

client\_1 | The square root of 5 is 2.23606797749979

client\_1 |

docker\_client\_1 exited with code 0

C:\Users\user\hw6\$ docker-compose down

Removing docker\_client\_1 ... done

Removing docker\_server\_1 ... done

Removing network docker\_default

## 6. What to Submit

Use the CS370 Canvas to submit a single .zip or .tar file that contains:

- The folder hw6 with the contents and structure specified in part 1

For this and all other assignments, ensure that you have submitted a valid .zip/.tar file. After submitting your file, you can download it and examine to make sure it is indeed a valid zip/tar file, by trying to extract it.

**Filename Convention:** The archive file must be named as: <FirstName>-<LastName>-HW6.<tar/zip>. E.g. if you are John Doe and submitting for assignment 1, then the tar file should be named John-Doe-HW1.tar

## 7. Grading

For this assignment your code should run on your computer (or any computer) that is running the latest version on Docker.

The grading will be done on a 100 point scale. The points are broken up as follows:

| Objective  | Points    |
|--|-----------|
| Task 2.1 (answering Questions from Part 4)                                       | 40 points |
| Task 2.2 (completing the files specified in Part 1)                              | 50 points |
| Task 2.3 (running docker-compose up from your hw6 folder creates correct output) | 10 points |
|  |           |

## 8. Late Policy

Click here for the class policy on submitting [late assignments](#).

**Revisions:** Any revisions in the assignment will be noted below.

11/1/2021: Example output updated.

11/8/2021: Docker Compose specification should be "...will cause the server to return the square root value to the client ..." instead of "...will cause the server to ~~serve the 'Hello World!'~~ to the client.."