Docker Lab Part 2

Objective

The objective of this lab is to learn to build your own docker images to containerise applications. You will also learn to pass data into and out of containers.

Preparation

If you have not already done so, install Docker on your laptop and ensure it works properly. You may optionally use the following resources to familiarise yourself with building docker images.

1. Watch the rest of this video https://youtu.be/fqMOX6JJhGo?t=2650
2. Read this article https://stackify.com/docker-build-a-beginners-guide-to-building-docker-images/

Instructions

In the lab zip file you find four simple Node.js applications within the subdirectories. The Dockerfile under `'app1/'` demonstrates how to build and run '`app1.js`'. For some of the exercises you will need to create Dockerfiles for the rest of the applications. Put answers to essay-type questions in a plain text file called `answers.txt` at the top level.
Upload a single zip of your answers. Do not modify the directory structure or any of the files given. This is an individual assignment.

Exercises

1. Use the '`docker build`' command to build an image and run '`app1.js`'. Ensure it is built successfully by doing '`docker image ls`'. Now run it and ensure it works by accessing http://localhost:3000.
2. Now write a Dockerfile for `'app2.js'` using the Dockerfile for '`app1.js`' as a starting point. The provided Dockerfile copies files to the root directory of the image. Modify it to copy application files to the `'/app'` directory within the image.
3. `'app2.js'` can be passed two arguments, `'message'` and `'colour'` as environment variables. Show how you can set the message and colour variables when running this docker image (without changing the code given.)
4. `'app3.js'` requires the modules in `package.json` to be installed. Use the '`npm`' command to install the necessary modules when building the image. You can set the same environment variables as `app2.js` to change the message and background colour at run time like `'app2.js'`.
5. Write a Dockerfile for '`app4.js`' installing the required modules. Run the image and upload a file. Verify that the file was uploaded by visiting http://localhost:3000/upload/filename (e.g. http://localhost:3000/upload/photo.jpg.)

6. Stop the container you ran in (5) and remove the stopped container using the '`docker container rm`' command. Now start a new container of the '`app4.js`' image. Try to access the file you uploaded in (5.) Explain the reason why the file is not accessible.
7. Create a volume to hold uploads using '`docker volume create`'. Now show how '`app4.js`' can be run with the volume mounted. Verify the that the uploaded files are available after the container is removed and a new one is started (with the volume mounted.)
8. Create an account on Docker Hub. Upload the image you created for '`app4.js`' to Docker Hub and provide a URL to your image.