Docker Tutorial (For Students)

Docker enables individuals to create, build, share, and run their web projects on any platform. We use Docker to ensure that graders are able to replicate the exact environment students use to develop their projects.

Install Docker

Install Docker according to your OS.

https://docs.docker.com/docker-for-mac/ (Mac)
https://docs.docker.com/docker-for-windows/ (Windows)
https://docs.docker.com/engine /getstarted/ (Linux)

Submitting Assignments using Docker

Prepare Submission Folder

You will submit a zipped submission folder that contains your project.

Your submission folder needs to have ONLY the following:

- `project/`
    - This folder will contain all of your project files.
- `Dockerfile`
    - This file ensures that both you and the grader build your project the same way. See next page for more information on how to configure this file.
- `docker-compose.yml`
    - This file ensures that both you and the grader run your project the same way. See next page for more information on how to configure this file.

Test Your Docker Image

Before submitting your assignment to be graded, test your Docker setup to ensure it works in the same way that the grader will be using it.

The grader will perform the following actions. (Make sure the following works!):

1. Unzip your zipped submission folder.
2. Change directories to the root of your submission folder (which should contain the Dockerfile and the docker-compose.yml file).
3. Run:
   ```
   docker-compose up --build
   ```
4. Open a browser and navigate to `localhost:8080`.
5. Start interacting with your project.

Brief Explanation of Dockerfile and docker-compose.yml files

Dockerfile

The Dockerfile describes how to build your Docker image.

1. First you'll start your image based on an existing docker image that Docker distributes. Most of these Docker images are some variant of a Linux operating system with pre-installed dependencies. You can search for images on the Docker Hub.
   e.g. `FROM php:7.0-apache`

2. Then (optional), you'll need to install any software onto the image that your project needs. Ideally, a Docker image would already exist with everything you need so that you wouldn't have to do this step.
   e.g. `RUN apt-get update && apt-get install ...`

3. Then you'll need to copy any of your project files that you want to run inside the image from your project folder onto the image. Where you need to copy them depends on how the image is configured and what software you are trying to run on that image.
   e.g. `COPY project/ /var/www/html`

This is sufficient for basic use cases, but there is much more that you can configure. See the Dockerfile reference for more information.

docker-compose.yml

The docker-compose.yml file describes how your Docker image should be run.

For your project to work correctly with the Grader, ensure you have the following pieces:

1. A section for your project.
```
project:
```

2. Specify that your project will be built from the Dockerfile in this directory.
```
project:
  build: .
```

3. Specify that your project will be served on localhost on port 8080.
```
project:
  build: .
  ports:
  - 8080:xx
```

NOTE: xx represents the exposed port on your image that you want to forward to localhost. Keep in mind that if you building off another Docker image using `FROM`, that image may be exposing ports too (images with apache typically have port 80 exposed).

This is sufficient for basic use cases, but there is much more that you can configure. See the docker compose reference for more information.

Useful Command Line Tools

Cleaning up

Docker images are large files (+200 MB). Building many unique docker images will take space on your computer. You can cleanup some space using the following commands.

```
docker stop $(docker ps –q)      # Stops all containers
docker rm $(docker ps –aq)       # Removes all containers
docker rmi $(docker images –q)   # Removes all images
```

Sometimes you might not want to delete all your images, since it can take time to rebuild them. To remove images that aren't used, run:

```
docker rmi $(docker images –q --filter "dangling=true")
```