Name: Chan Ming En Minern

Matric No.: A0164749N

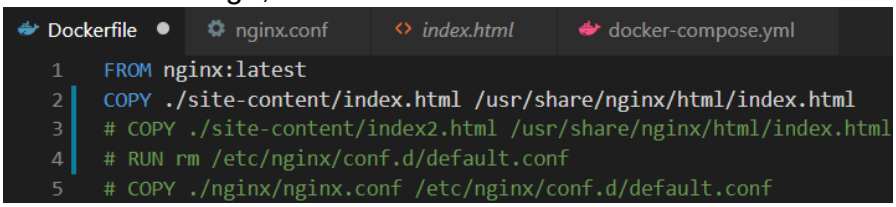
Github repo: https://github.com/minernchan/OTOTassignmentA

How to run the docker container:

1) Navigate to the assignment folder
2) Create the necessary docker images:

   To create the different docker images, we will be using the 'docker build' command multiple times with the Dockerfile in the folder. This Dockerfile should be edited according to the docker image we want to create.

   a. mywebsite
   For the first image, edit and save the Dockerfile as below:

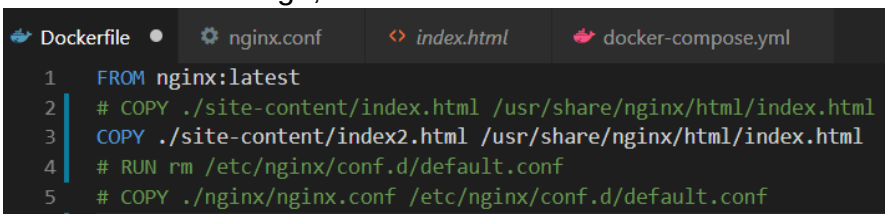


   Run in the command line the command "docker build -t mywebsite ."

   Alternatively, the image is also available on Docker Hub, so the command "docker pull mnrnn/mywebsite" should also work.

   This image simply serves a static html page replaced with the contents in index.html.

   b. mywebsite2
   For the second image, edit and save the Dockerfile as below:

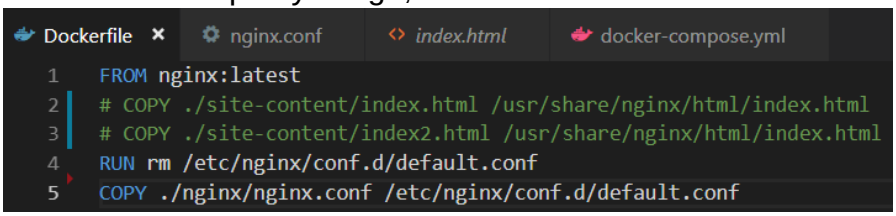


   Run in the command line the command: "docker build -t mywebsite2 ."

   Alternatively, the image is also available on Docker Hub, so the command "docker pull mnrnn/mywebsite2" should also work.

   This image simply serves a static html page replaced with the contents in index2.html.

c. myproxy
For the reverse proxy image, edit and save the Dockerfile as below:

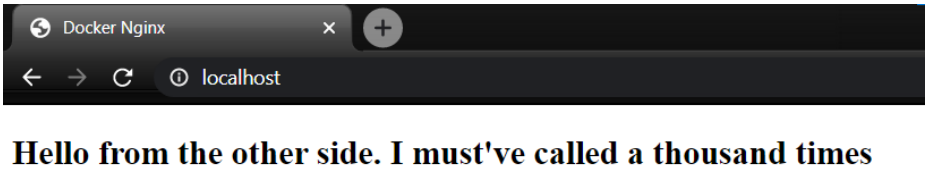


Run in the command line the command “docker build -t myproxy .”

Alternatively, the image is also available on Docker Hub, so the command “docker pull mnrnn/myproxy” should also work.

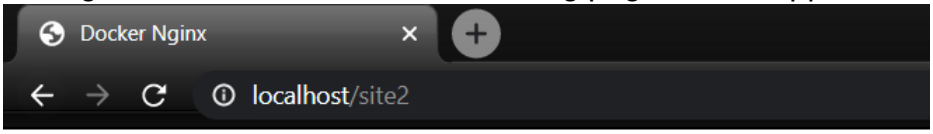
3) Run “docker-compose up” in the command line.

4) Open an internet browser e.g. Chrome, and go to “localhost”. A page like this should appear:



Hello from the other side. I must've called a thousand times

If we go to “localhost/site2”, the following page should appear:



To tell you I'm sorry for everything that I've done

Finally, if we go back to “localhost/site1”, we should be rerouted back to the same html page as when we were at “localhost”.

This is because of how the nginx.conf file was configured, and in doing so, we are able to use the different applications from different containers without knowing which ports the applications are using.

Screenshot of Docker Hub