2 Container Project
Created by: Kyle Pierrehumbert

Introduction

This project was created to satisfy the task of building a 2 container application serving static content. There are 2 NGINX containers in this environment. The first container contains security controls and proxy, the second container serves the static html content.

I have created a brief demonstration video of this project:
https://www.youtube.com/watch?v=4xi89P1rKng

Assumptions

- You are familiar with Docker, linux, .yml and other associated components.
- You have a properly configured test environment.
- You do not have anything running on ports 8081 and 443. This project relies on these ports.

Environment

- This was completed on macOS version 10.15.7
- Docker client version 20.10.6 Engine version 20.10.6

Files included

- "Kyleproject" folder containing all information and structured to work with "docker-compose up"
- docker-compose.yml
- "auth2" folder containing dockerfile, nginx.conf, certificates, and hidden password file.
- "html2" folder containing conf, dockerfile, index.html

Considerations/potential limitations

There is no docker0 bridge on macOS. Because of the way networking is implemented in Docker Desktop for Mac, you cannot see a docker0 interface on the host. This interface is actually within the virtual machine. The official Docker work around for this is using the special DNS name "host.docker.internal" which resolves to the internal IP address used by the host. This is for development purpose and will not work in a production environment outside of Docker Desktop for Mac. (REF-https://docs.docker.com/docker-for-mac/networking/).

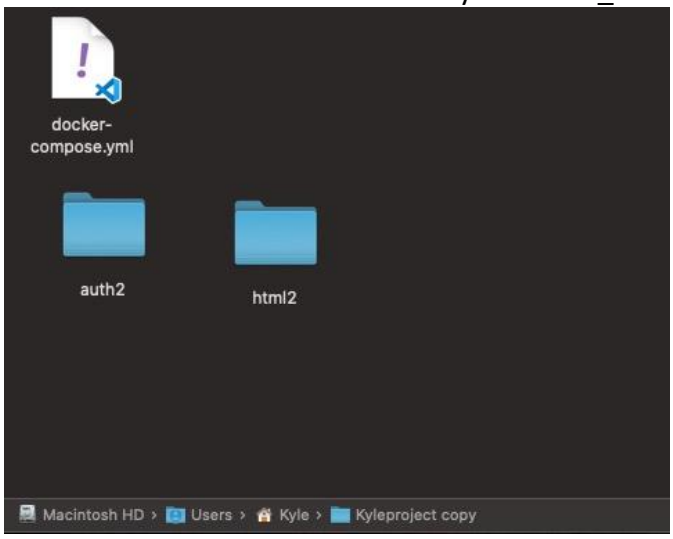
This Docker networking feature is supported in Docker Desktop for Mac and Windows. Windows (REF for Windows- https://docs.docker.com/docker-for-windows/networking/).

This may affect functionality in your testing environment. Please use Mac or properly configured Windows.

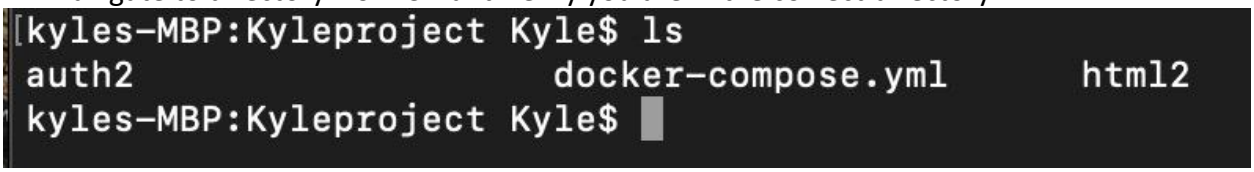
Login credentials

- Username- "admin" password- "admin"
- Secret "url token" is "kyletoken" https://localhost/**kyletoken**/index.html

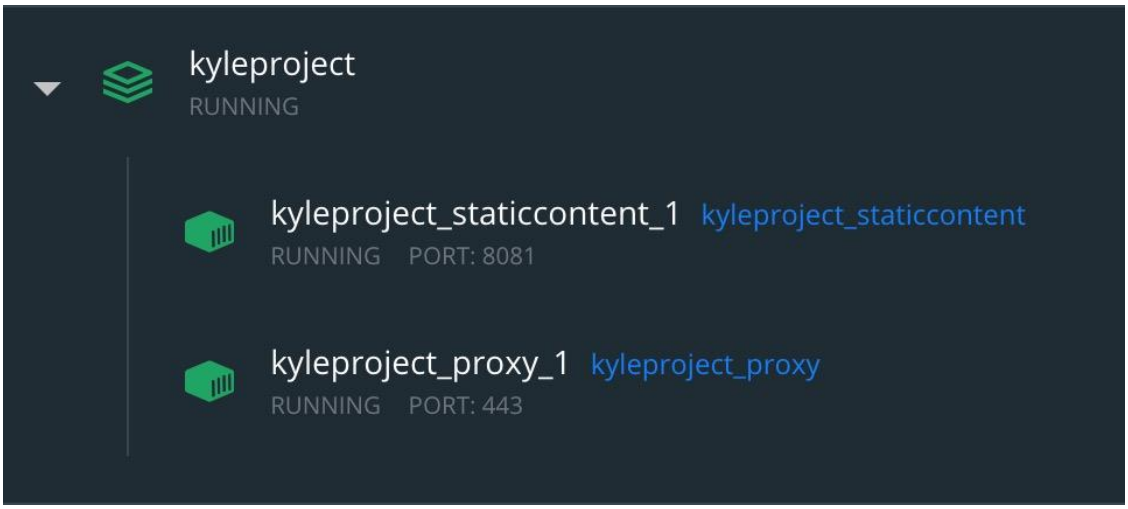
- **Directions**
- Download files and verify that the file structure is correct. When downloading zip there will be an extra file created by mac ".DS_Store" this is not needed.



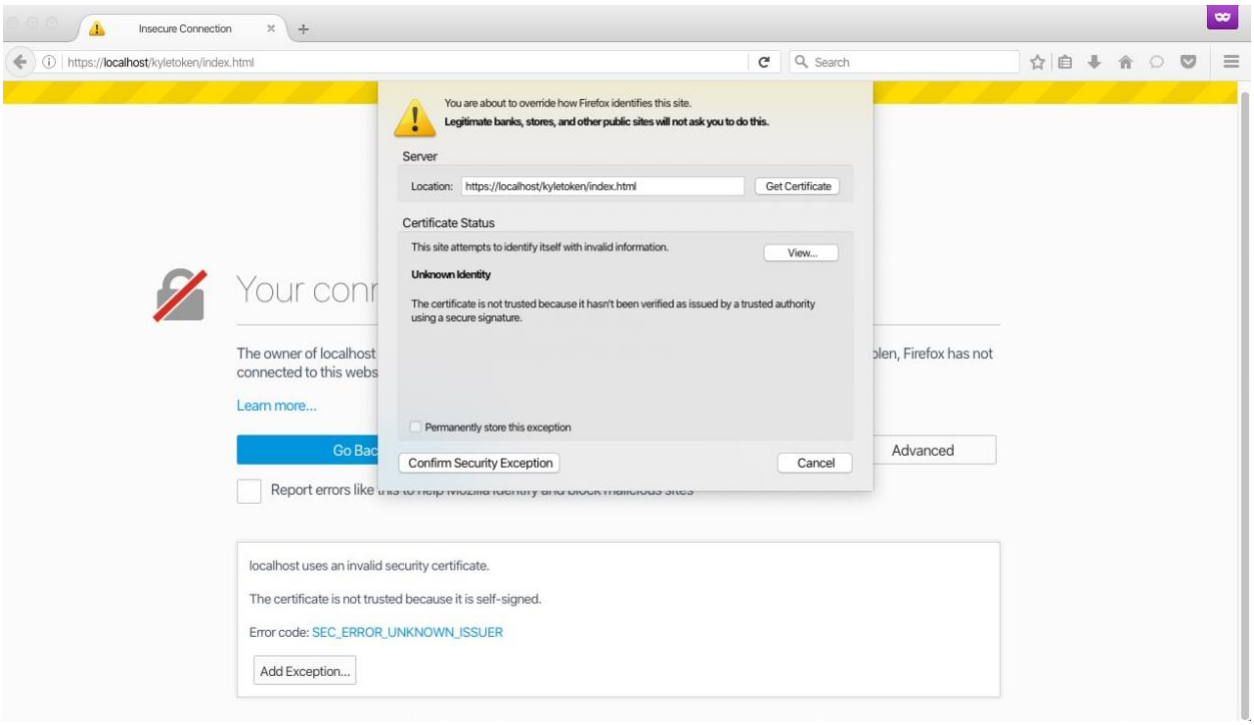
- Navigate to directory from CLI and verify you are in the correct directory.



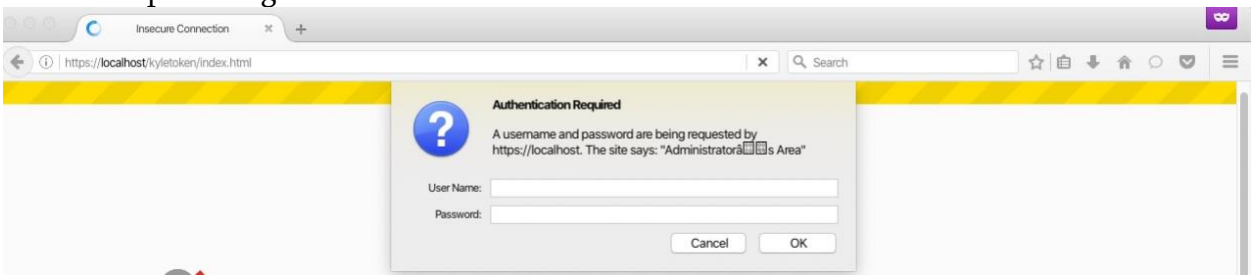
- Run "docker-compose up -d" to deploy in one command. We see the containers are running.



- Navigate to "https://localhost/kyletoken/index.html" using Firefox private window. You will receive a secure connection warning due to using self-signed certs. Click "advanced" "add exception" and "confirm security exception"



- You will be prompted to enter the login credentials. “admin” “admin” this is providing basic authentication.



- You will then be directed to the static “top secret” html content.
- You can verify encryption with the “lock”