David Allen

Reflection Paper 3

Date: 11/25/2025

Company: Old Dominion University – Division of Digital Transformation & Technology

Position: Cloud Developer Intern

ODU Fall 2025

Professor Teresa Duvall

Internship Reflection Paper

Third 50 Hours

During my third 50 hours as a cloud developer intern, I have gained knowledge about how to use Docker. Although they were not my most successful, I was still able to learn about the importance of Docker. Docker is an application that hosts an application within a container. This container is isolated from the host with its own port, but still able to communicate with the host system. Docker is a great platform for cloud developers to be able to develop and test their applications to see how they behave. It also comes with some security features as well. The container itself can be scanned for vulnerabilities. If you sign in, you can see recommended actions to mitigate these vulnerabilities, although I was mostly trying to get the application running on Docker.

I had two repositories, technically three because one had separate file directories, where I was tasked with running an application on Docker. The first repository was the Student360 front end and back end, which both were separated into different directories. The way Docker works is that the repository needs to be built using a Docker configuration file. This Docker configuration file defines the build image used, such as Alpine, for example. After that, the working directory for the application is defined. Package files that contain dependencies are copied from the host and into the container. After that, dependencies are installed, such as using NPM to update the repository within the container as stated in the Docker configuration file. I had some other errors, like Python not being installed, so a script is used to install necessary compilers in the same step. Once that was finished, the rest of the project files are copied from the host and into the container. If necessary, a command is used to build the project. I tried running the build command, but this isn't defined in the package files as a script based on how the repositories were developed. I removed it from the Docker configuration file to fix the error. Once that is done, the port for the container is exposed. Any number that is available can be chosen, but I chose 3000 for the front end and 3001 for the back end. After that, the serve command is installed globally. The serve command is used to serve the application to the exposed port. Once the port is exposed, the application should be accessible to the user by clicking on a link within Docker. A command can also be used to run the container by defining the port for the container and the port for the host to establish the connection.

The main problem I had was that I didn't know how to run the application, as when I built the container and accessed the link it would display the file directory. I kept having this issue with both repositories and it felt like a lost cause. I was at least able to understand it through the tutorial provided on Docker with another simple application. Learning about Docker

was a great experience for me. It helped me understand how applications could run through a container for DevOps purposes, or in my case DevSecOps. I've previously learned about DevSecOps through other courses I attended not at ODU. It really helped me make that connection of testing an application for both functionality and security during the internship.

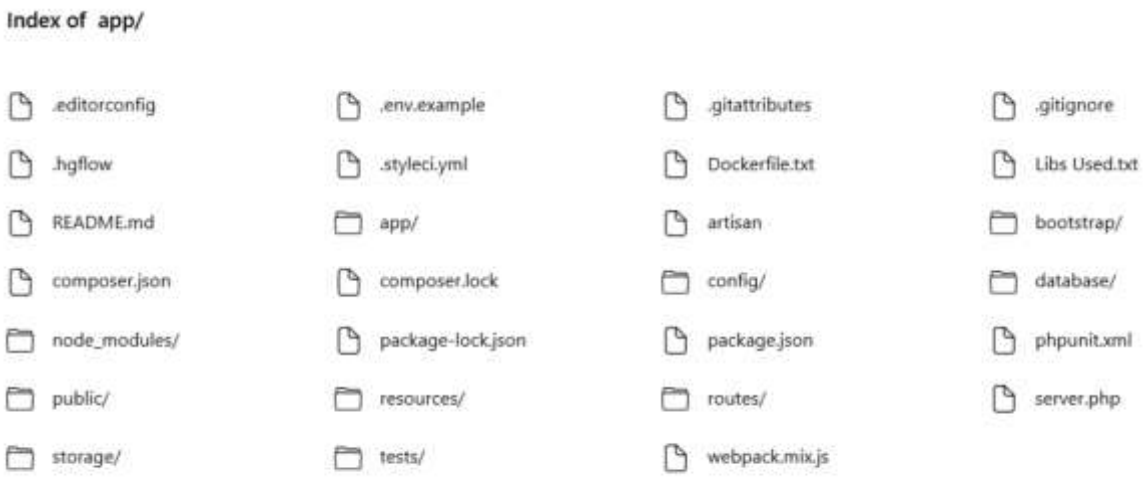
The value of patience and determination are the most important things I've learned during my third 50 hours of my internship. This has been the most difficult part in my internship, with the challenging tasks given. I've probably faced the most obstacles and discouragement with some of the tasks I've tried to perform, but sometimes things just aren't meant to be, but this is alright for me. It's better to move on to greater and more productive tasks. If one repository didn't work, I tried another. If things are going like they are, then it's better to move on to something else than to waste time doing the same thing over and over again.

Overall, I enjoyed the third 50 hours of my internship, although there were some mountains to climb. I was still able to learn despite the challenges I faced and was unable to produce. I still understood the basics of Docker in other ways, which is what I found the most important. As a cloud developer intern, I find this very useful to deploy applications for security testing. I'd highly recommend it, if the application being made is possible to deploy on Docker.

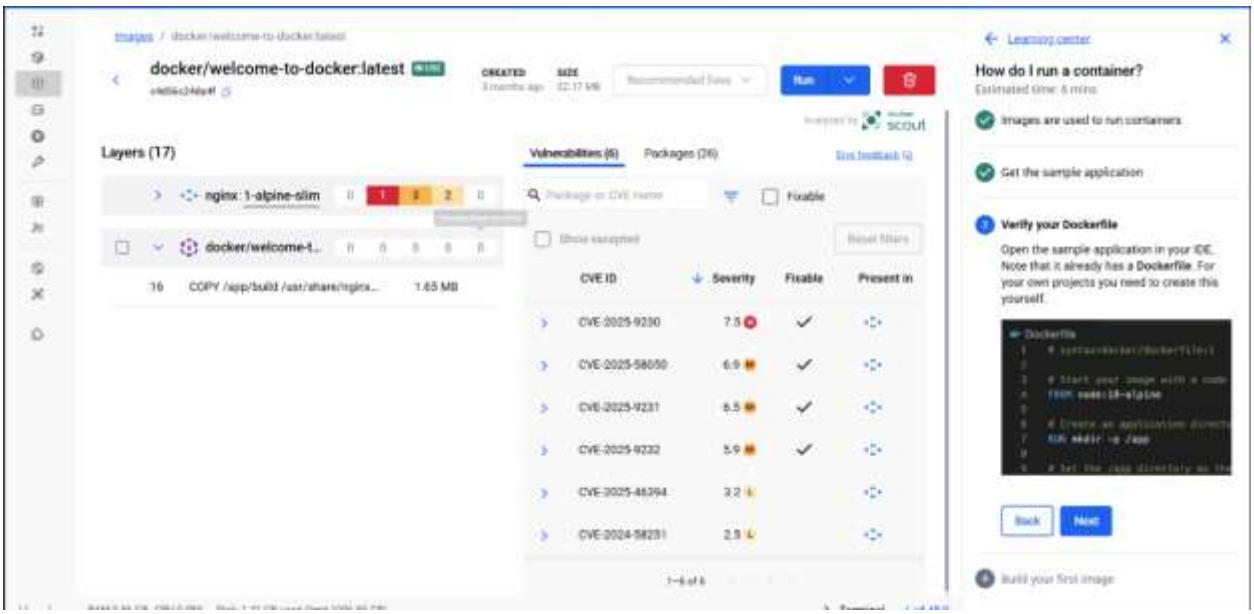
Appendix

```
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/bnqmp4ov4rmnfexmfzq9ta2c8
PS C:\Users\dalle042\Repository\student-success-web-service-personal-test> docker build -f Dockerfile.txt -t student-su
ccess-test .
[+] Building 24.3s (12/12) FINISHED                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile.txt                                                          0.0s
 => => transferring dockerfile: 538B                                                                              0.0s
 => [internal] load metadata for docker.io/library/node:22-alpine                                                 0.6s
 => [internal] load .dockerignore                                                                                 0.0s
 => => transferring context: 2B                                                                                   0.0s
 => [1/7] FROM docker.io/library/node:22-alpine@sha256:b2358485e3e33bc3a33114d2b1bdb18cdbe4df01bd2b257198eb51beb  0.0s
 => => resolve docker.io/library/node:22-alpine@sha256:b2358485e3e33bc3a33114d2b1bdb18cdbe4df01bd2b257198eb51beb  0.0s
 => [internal] load build context                                                                                 0.0s
 => => transferring context: 22.04kB                                                                              0.0s
 => CACHED [2/7] WORKDIR /app                                                                                     0.0s
 => CACHED [3/7] COPY package*.json ./                                                                            0.0s
 => CACHED [4/7] RUN apk add --update --no-cache python3 build-base gcc && ln -sf /usr/bin/python3 /usr/bin/pyth  0.0s
 => CACHED [5/7] RUN npm install --legacy-peer-deps                                                               0.0s
 => [6/7] COPY . .                                                                                                0.2s
 => [7/7] RUN npm install -g serve                                                                                4.6s
 => exporting to image                                                                                           18.6s
 => => exporting layers                                                                                          12.6s
 => => exporting manifest sha256:f50b505b2877bbe0fc6f329f2b6692681154624dd224d3d02c58b4b1458dfcce                  0.0s
 => => exporting config sha256:5742f575d7387f17122bcca9385feddfb61f2ce3b336e0ce2528a41398e39f61                   0.0s
 => => exporting attestation manifest sha256:5d67107fb8214ad48e7fa041b3b779896e7c29ba97ffa76a77434238c444aaf1     0.0s
 => => exporting manifest list sha256:22d35508e2d2db43758171b721265421fad1237a9fd42160f2e692880c3eac22            0.0s
 => => naming to docker.io/library/student-success-test:latest                                                    0.0s
 => => unpacking to docker.io/library/student-success-test:latest                                                 5.9s
```

The process of a Docker container being built



The issue I was having where only the file directory was being displayed



Scanning for vulnerabilities in Docker