## Intro

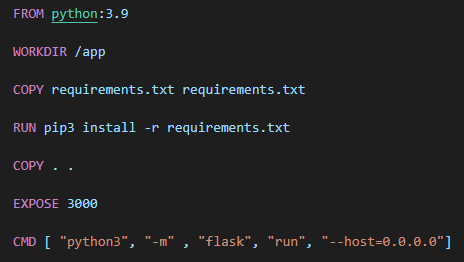
My name is Wilson Chan, thank you so much for giving me the opportunity, which I have attempted to the best of my ability. I have completed parts of this project using Windows 10 Home edition so some solutions may differ.

## Processes

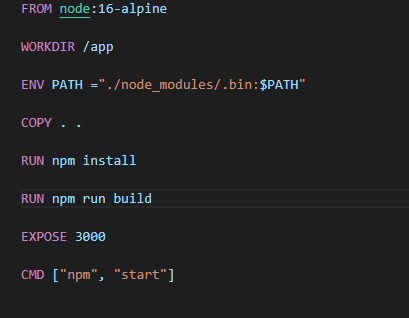
**Dockerise the application**

As presented, there are two main components to this project, the Python-Flask framework API and the React frontend. When both are running, it displays a web application that shows RAM and CPU usage on two statistics on the local system. To begin, I experimented on the containerisation of each of these components. To do this, Docker files had to be created in their respective directories.

The Dockerfile that I have set up in the Python API directory is as follows.



Working with system Python version 3.9, the command is followed by copying the container dependencies requirements text. For this instance, I included the essential packages such as flask, flask cor etc that were essential to running the application. Gunicorn is also added within the file. Then, a copy command copies the entire directory (and directory above). To end, I chose to expose the documentary port 3000 and run the flask command to initiate the server in this array.

The Dockerfile that I have set up in the React frontend directory is as follows.

I have chosen the last distribution of Alpine instead of 17 as it was a workaround *to Error: error:0308010C:digital envelope routines::unsupported*. The copy command captures the json directory file and installs everything required for the frontend to run. The env routes the environment and runs the build commands. To run the containers in tandem sync, I’ve also set up a docker-compose file that works with *docker-compose up*.



NGINX was installed locally and redirected to port 4000

**Deployment to Amazon EC2/EBS**

**Deployment to Kubernetes/Minikube**

From the completed Docker compose yaml, I used the Kompose conversion tool to convert to a Kubernetes manifest file. Initially, it did not work as the file paths had to be manually corrected. I managed, preliminarily, on Windows Docker Desktop’s Kubernetes integration as well as Kubernetes UI build with kubectl.