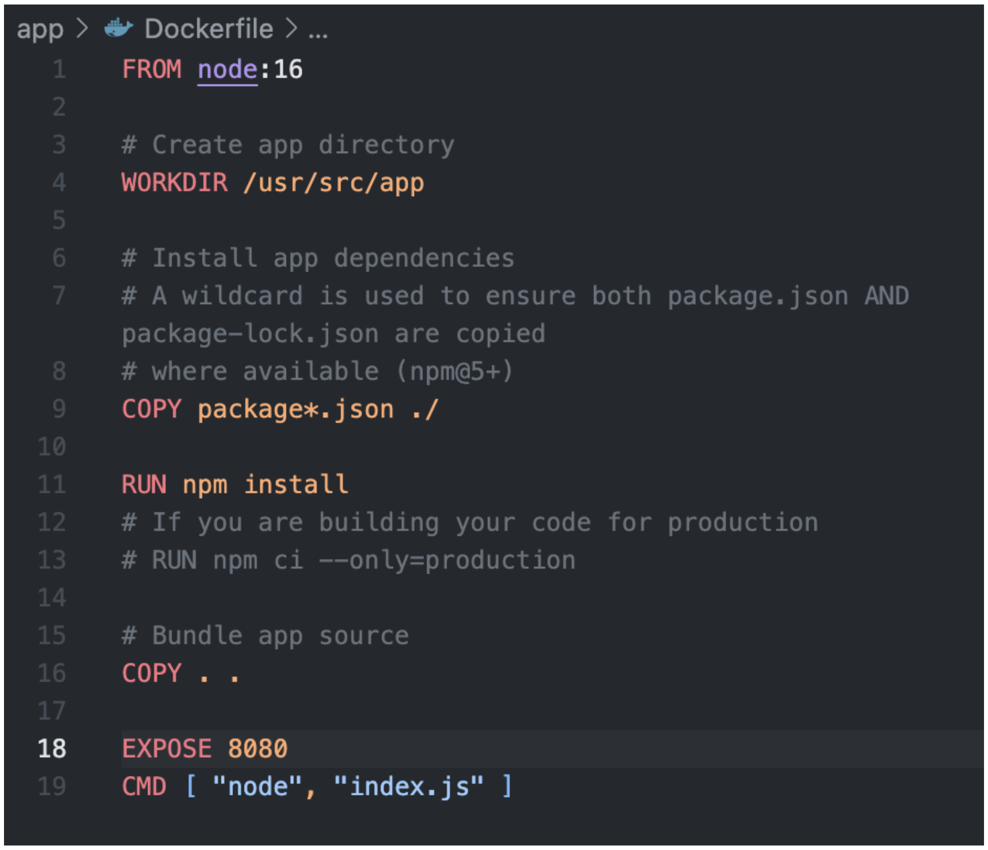
Name: Chen Hsiao Ting
Matriculation Number: A0222182R
Link to GitHub repository: https://github.com/hsiaotingluv/CS3219-OTOT-TaskA1
Link to Demo video:
https://drive.google.com/file/d/1q6xOcmwVU5KD8ske_ENULFgm_kKD_omM/view?usp=sharing

# Instructions on how to run the Docker container

## Task A1.1: Dockerize the given node app in the "app" folder

https://docs.docker.com/language/nodejs/build-images/

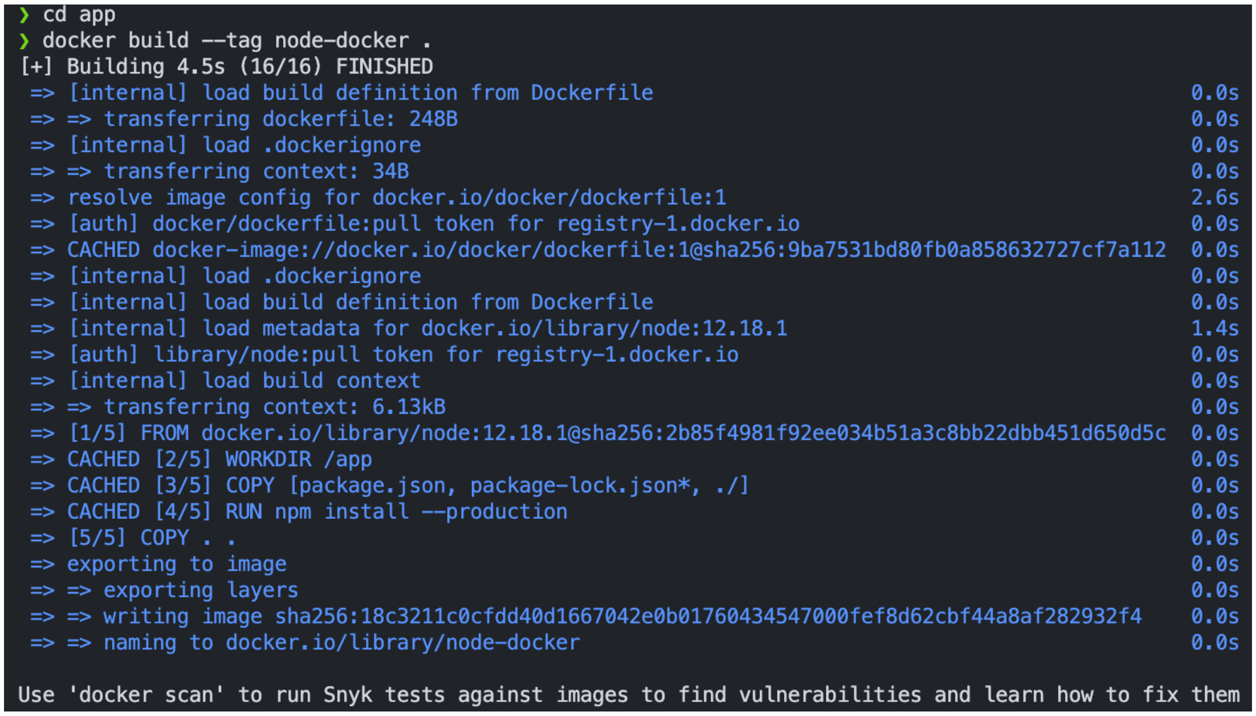
1. Create a Dockerfile in "app" folder



app > Dockerfile > ...

```
FROM node:16

# Create app directory
WORKDIR /usr/src/app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./

RUN npm install
# If you are building your code for production
# RUN npm ci --only=production

# Bundle app source
COPY . .

EXPOSE 8080
CMD [ "node", "index.js" ]
```

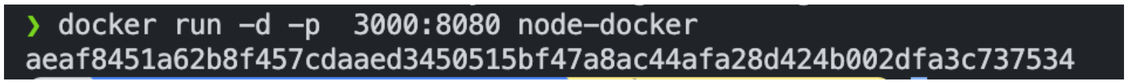
## 2. Build image



```
❯ cd app
❯ docker build --tag node-docker .
[+] Building 4.5s (16/16) FINISHED
 => [internal] load build definition from Dockerfile                                0.0s
 => => transferring dockerfile: 248B                                                0.0s
 => [internal] load .dockerignore                                                   0.0s
 => => transferring context: 34B                                                    0.0s
 => resolve image config for docker.io/docker/dockerfile:1                          2.6s
 => [auth] docker/dockerfile:pull token for registry-1.docker.io                    0.0s
 => CACHED docker-image://docker.io/docker/dockerfile:1@sha256:9ba7531bd80fb0a858632727cf7a112  0.0s
 => [internal] load .dockerignore                                                   0.0s
 => [internal] load build definition from Dockerfile                                0.0s
 => [internal] load metadata for docker.io/library/node:12.18.1                     1.4s
 => [auth] library/node:pull token for registry-1.docker.io                         0.0s
 => [internal] load build context                                                   0.0s
 => => transferring context: 6.13kB                                                 0.0s
 => [1/5] FROM docker.io/library/node:12.18.1@sha256:2b85f4981f92ee034b51a3c8bb22dbb451d650d5c  0.0s
 => CACHED [2/5] WORKDIR /app                                                       0.0s
 => CACHED [3/5] COPY [package.json, package-lock.json*, ./]                        0.0s
 => CACHED [4/5] RUN npm install --production                                       0.0s
 => [5/5] COPY . .                                                                  0.0s
 => exporting to image                                                              0.0s
 => => exporting layers                                                             0.0s
 => => writing image sha256:18c3211c0cfdd40d1667042e0b01760434547000fef8d62cbf44a8af282932f4   0.0s
 => => naming to docker.io/library/node-docker                                      0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
```

- run `docker build --tag node-docker .` on terminal
- The build command optionally takes a --tag flag. The tag is used to set the name of the image and an optional tag in the format ‘name:tag’

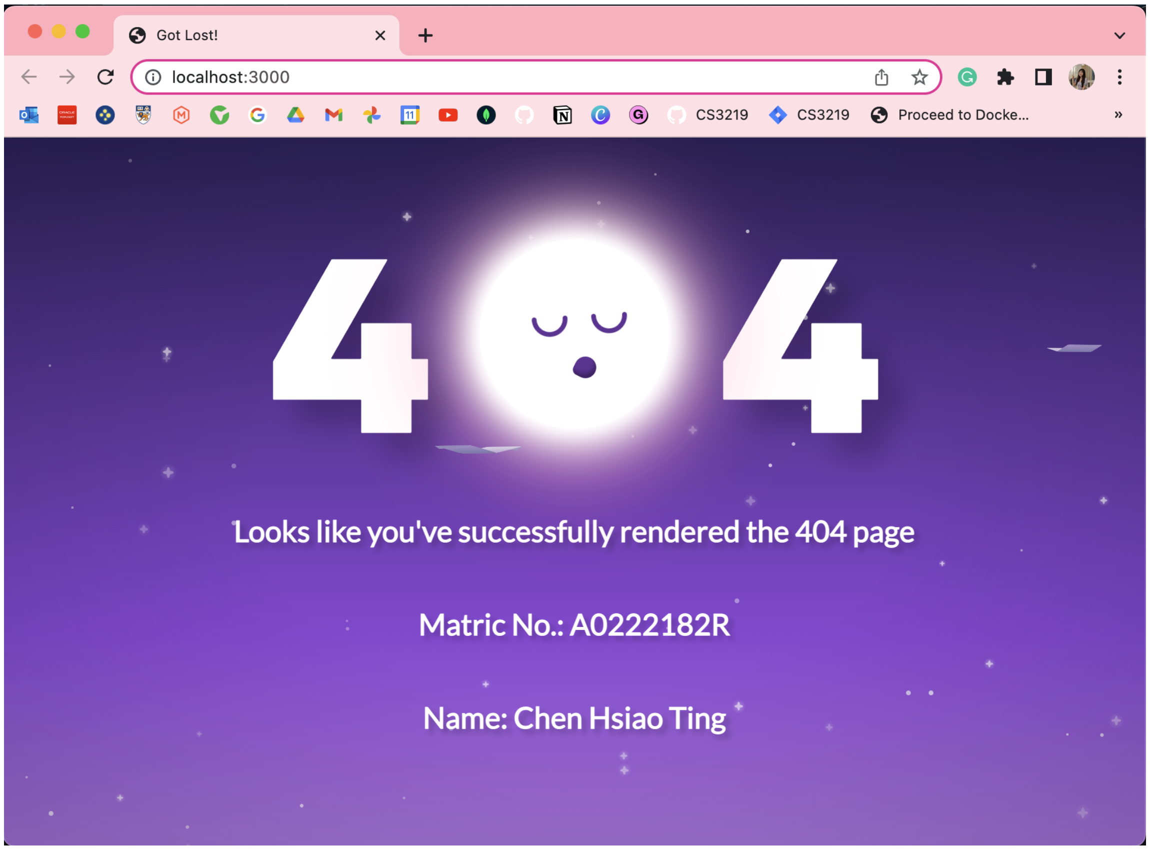
## 3. Run image as a container in detached mode



```
❯ docker run -d -p  3000:8080 node-docker
aeaf8451a62b8f457cdaaed3450515bf47a8ac44afa28d424b002dfa3c737534
```

- run `docker run -d -p  3000:8080 node-docker` on terminal
- -d or --detach run our container in detached mode or in the background
- -p or --publish to publish a port for our container. The format of the --publish command is [host port]:[container port]. So if we wanted to expose port 8080 inside the container to port 3000 outside the container, we would pass 3000:8080

4. View on localhost:3000



Looks like you've successfully rendered the 404 page

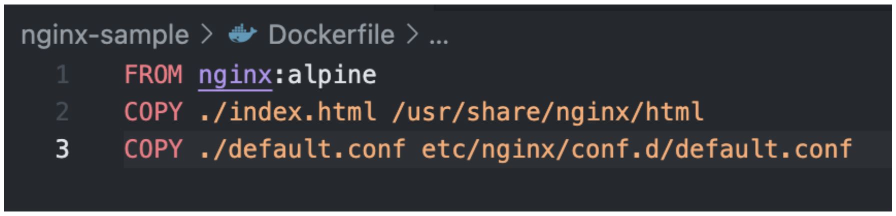
Matric No.: A0222182R

Name: Chen Hsiao Ting

## Task A1.2: Dockerize NGINX to serve the static html page in the “nginx-sample” folder

https://www.dailysmarty.com/posts/steps-for-deploying-a-static-html-site-with-docker-and-nginx

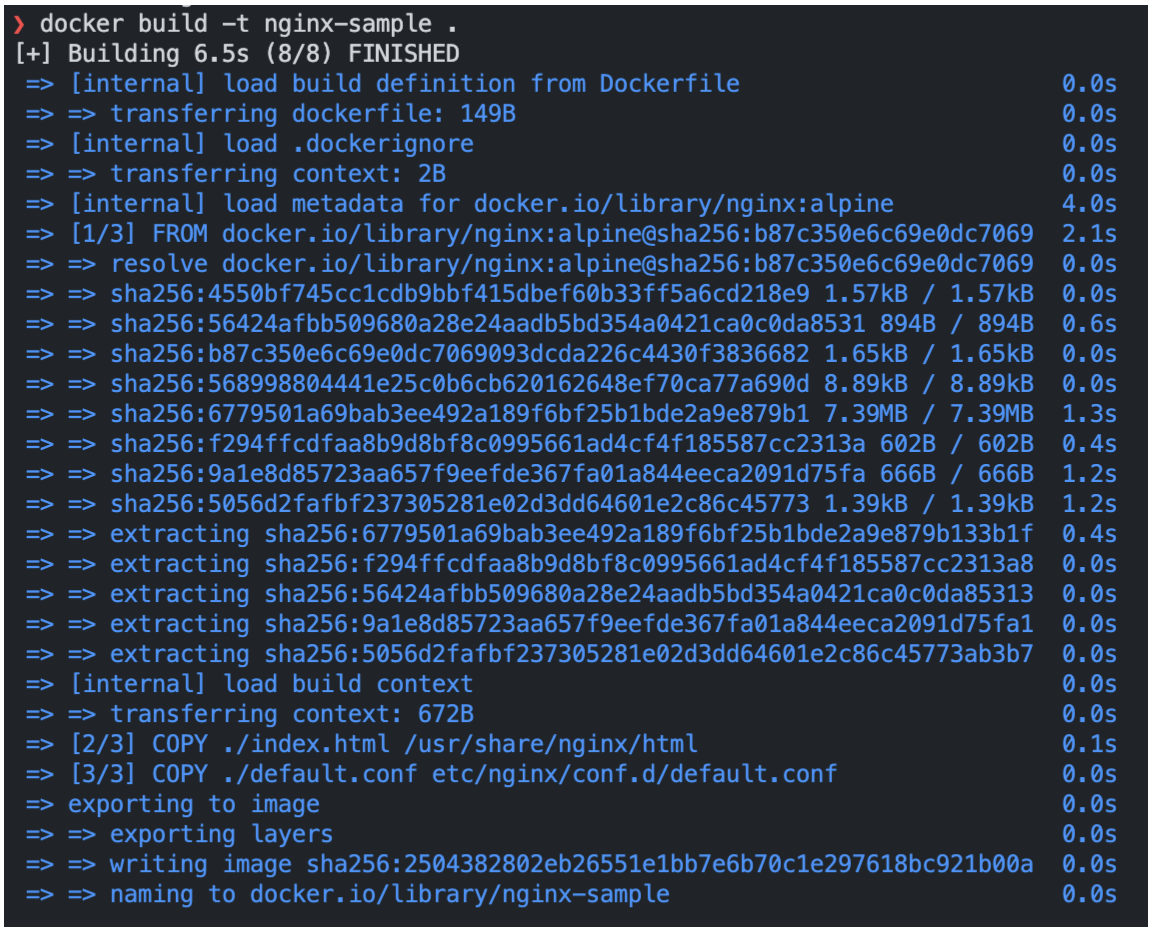
1. Create a Dockerfile in “nginx-sample” folder



nginx-sample > Dockerfile > ...

```dockerfile
FROM nginx:alpine
COPY ./index.html /usr/share/nginx/html
COPY ./default.conf etc/nginx/conf.d/default.conf
```

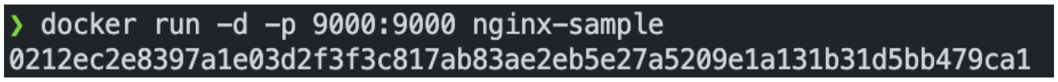
2. Build the Docker Image for the HTML Server



```
❯ docker build -t nginx-sample .
[+] Building 6.5s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                       0.0s
 => => transferring dockerfile: 149B                                       0.0s
 => [internal] load .dockerignore                                          0.0s
 => => transferring context: 2B                                            0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine            4.0s
 => [1/3] FROM docker.io/library/nginx:alpine@sha256:b87c350e6c69e0dc7069  2.1s
 => => resolve docker.io/library/nginx:alpine@sha256:b87c350e6c69e0dc7069  0.0s
 => => sha256:4550bf745cc1cdb9bbf415dbef60b33ff5a6cd218e9 1.57kB / 1.57kB  0.0s
 => => sha256:56424afbb509680a28e24aadb5bd354a0421ca0c0da8531 894B / 894B  0.6s
 => => sha256:b87c350e6c69e0dc7069093dcda226c4430f3836682 1.65kB / 1.65kB  0.0s
 => => sha256:568998804441e25c0b6cb620162648ef70ca77a690d 8.89kB / 8.89kB  0.0s
 => => sha256:6779501a69bab3ee492a189f6bf25b1bde2a9e879b1 7.39MB / 7.39MB  1.3s
 => => sha256:f294ffcdfaa8b9d8bf8c0995661ad4cf4f185587cc2313a 602B / 602B  0.4s
 => => sha256:9a1e8d85723aa657f9eefde367fa01a844eeca2091d75fa 666B / 666B  1.2s
 => => sha256:5056d2fafbf237305281e02d3dd64601e2c86c45773 1.39kB / 1.39kB  1.2s
 => => extracting sha256:6779501a69bab3ee492a189f6bf25b1bde2a9e879b133b1f  0.4s
 => => extracting sha256:f294ffcdfaa8b9d8bf8c0995661ad4cf4f185587cc2313a8  0.0s
 => => extracting sha256:56424afbb509680a28e24aadb5bd354a0421ca0c0da85313  0.0s
 => => extracting sha256:9a1e8d85723aa657f9eefde367fa01a844eeca2091d75fa1  0.0s
 => => extracting sha256:5056d2fafbf237305281e02d3dd64601e2c86c45773ab3b7  0.0s
 => [internal] load build context                                          0.0s
 => => transferring context: 672B                                          0.0s
 => [2/3] COPY ./index.html /usr/share/nginx/html                          0.1s
 => [3/3] COPY ./default.conf etc/nginx/conf.d/default.conf                0.0s
 => exporting to image                                                     0.0s
 => => exporting layers                                                    0.0s
 => => writing image sha256:2504382802eb26551e1bb7e6b70c1e297618bc921b00a  0.0s
 => => naming to docker.io/library/nginx-sample                            0.0s
```

- run `docker build -t nginx-sample .` on terminal

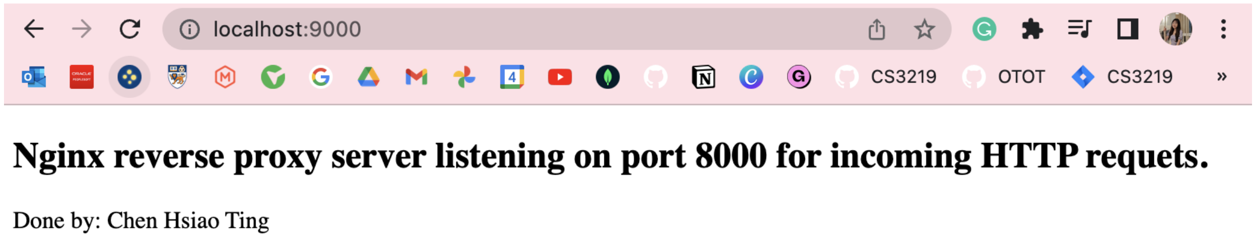
3. Run the Docker Container



```
❯ docker run -d -p 9000:9000 nginx-sample
0212ec2e8397a1e03d2f3f3c817ab83ae2eb5e27a5209e1a131b31d5bb479ca1
```

- run `docker run -d -p 9000:9000 nginx-sample` on terminal

4. View on localhost:80



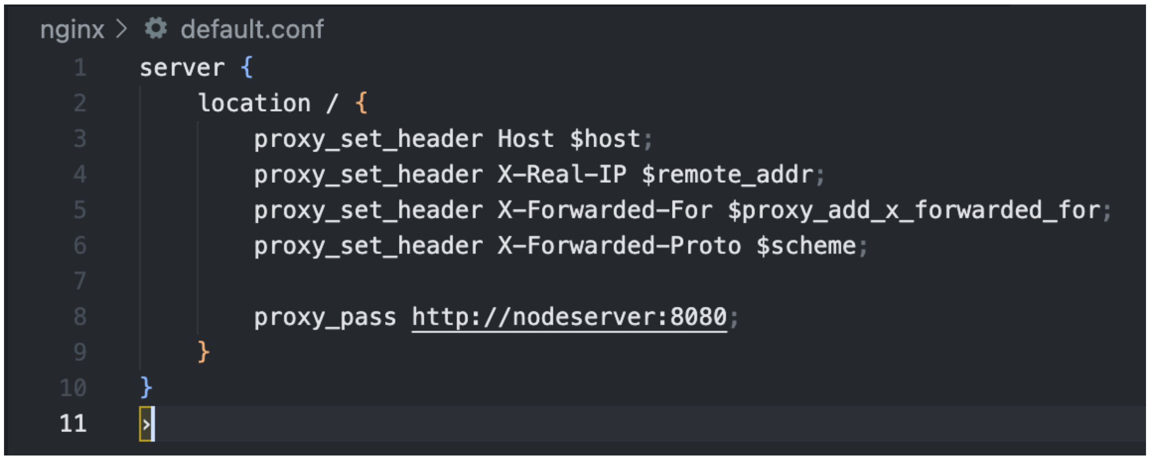
**Nginx reverse proxy server listening on port 8000 for incoming HTTP requets.**

Done by: Chen Hsiao Ting

## Task A1.3: Combining knowledge of docker and NGINX reverse proxy, use NGINX (in “nginx” folder) to serve the node app (“app” folder) using docker-compose

https://ashwin9798.medium.com/nginx-with-docker-and-node-js-a-beginners-guide-434fe1216b6b

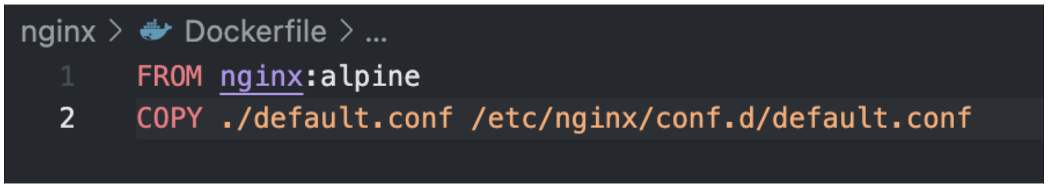
1. Create a default.conf file in “nginx” folder



nginx > default.conf

```
server {
    location / {
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;

        proxy_pass http://nodeserver:8080;
    }
}
```

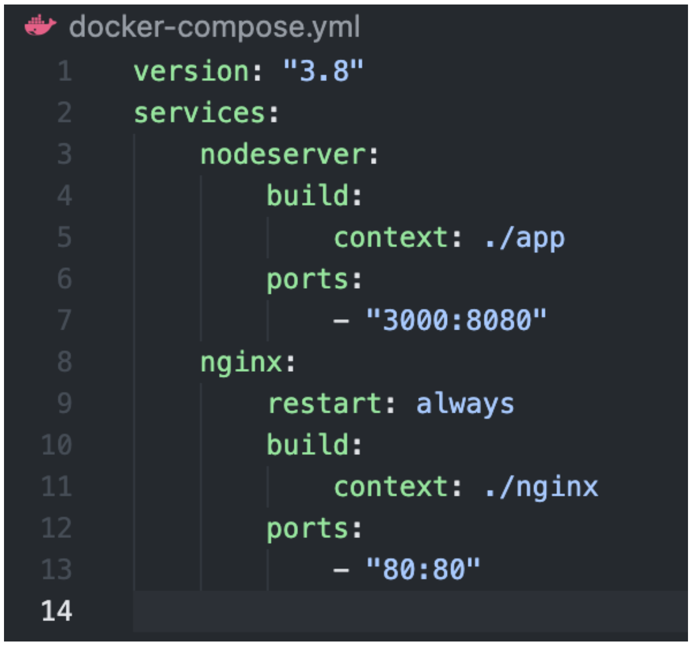
2. Create a Dockerfile in the “nginx” folder



nginx > Dockerfile > ...

```
FROM nginx:alpine
COPY ./default.conf /etc/nginx/conf.d/default.conf
```

3. Using docker-compose to coordinate the containers

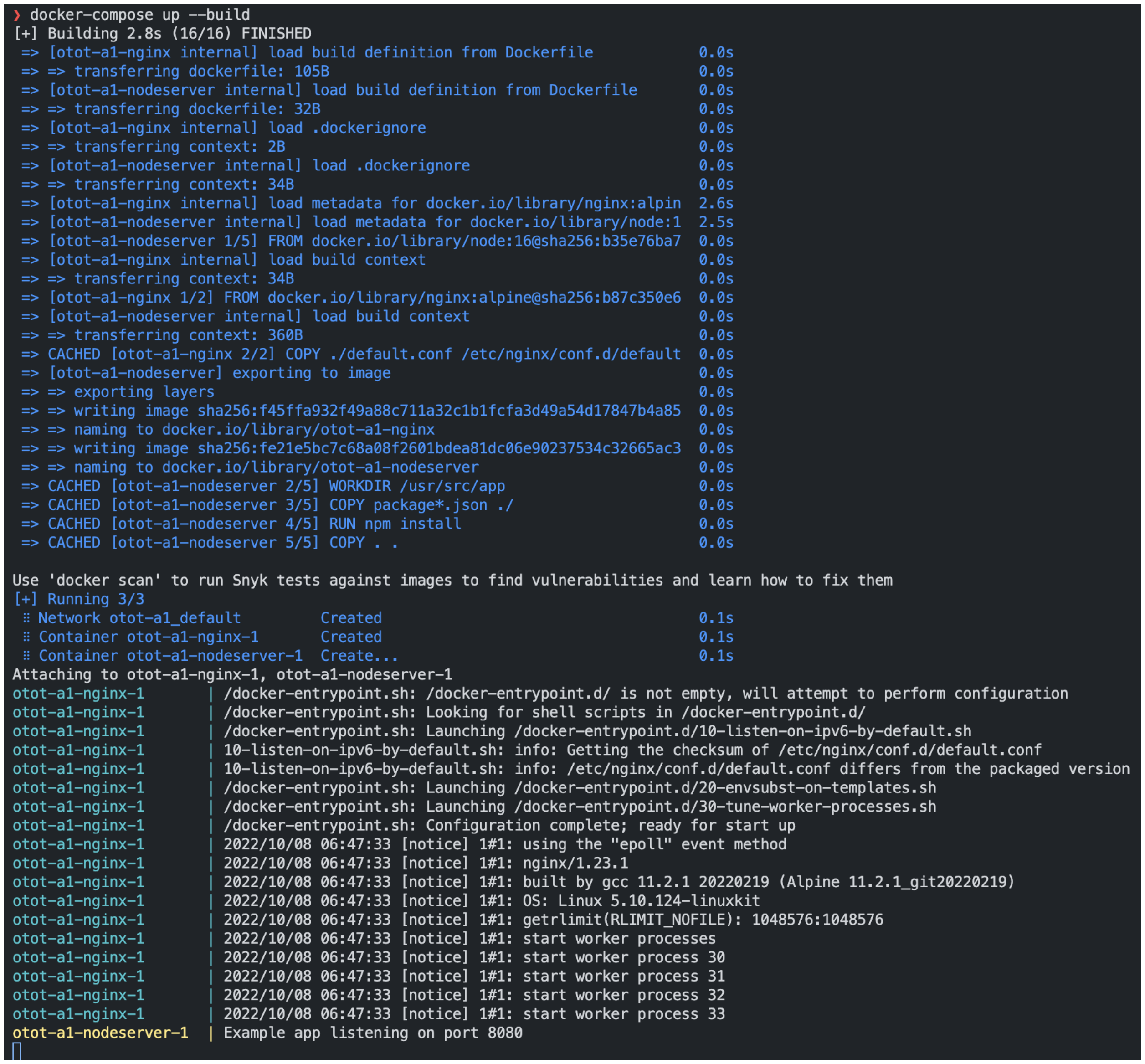


docker-compose.yml

```yaml
version: "3.8"
services:
    nodeserver:
        build:
            context: ./app
        ports:
            - "3000:8080"
    nginx:
        restart: always
        build:
            context: ./nginx
        ports:
            - "80:80"
```

- create docker-compose.yml file in the root directory

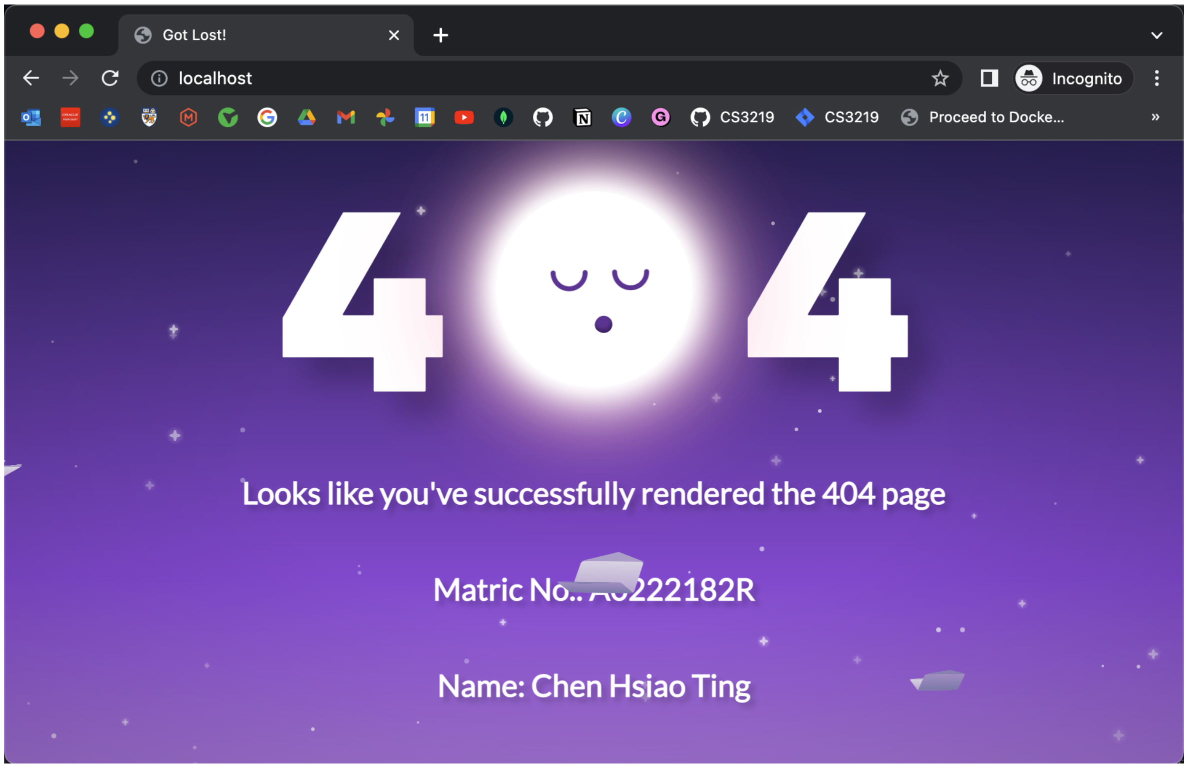
## 4. Build and run Docker



```
❯ docker-compose up --build
[+] Building 2.8s (16/16) FINISHED
 => [otot-a1-nginx internal] load build definition from Dockerfile            0.0s
 => => transferring dockerfile: 105B                                          0.0s
 => [otot-a1-nodeserver internal] load build definition from Dockerfile       0.0s
 => => transferring dockerfile: 32B                                           0.0s
 => [otot-a1-nginx internal] load .dockerignore                               0.0s
 => => transferring context: 2B                                               0.0s
 => [otot-a1-nodeserver internal] load .dockerignore                          0.0s
 => => transferring context: 34B                                              0.0s
 => [otot-a1-nginx internal] load metadata for docker.io/library/nginx:alpin  2.6s
 => [otot-a1-nodeserver internal] load metadata for docker.io/library/node:1  2.5s
 => [otot-a1-nodeserver 1/5] FROM docker.io/library/node:16@sha256:b35e76ba7  0.0s
 => [otot-a1-nginx internal] load build context                               0.0s
 => => transferring context: 34B                                              0.0s
 => [otot-a1-nginx 1/2] FROM docker.io/library/nginx:alpine@sha256:b87c350e6  0.0s
 => [otot-a1-nodeserver internal] load build context                          0.0s
 => => transferring context: 360B                                             0.0s
 => CACHED [otot-a1-nginx 2/2] COPY ./default.conf /etc/nginx/conf.d/default  0.0s
 => [otot-a1-nodeserver] exporting to image                                   0.0s
 => => exporting layers                                                       0.0s
 => => writing image sha256:f45ffa932f49a88c711a32c1b1fcfa3d49a54d17847b4a85  0.0s
 => => naming to docker.io/library/otot-a1-nginx                              0.0s
 => => writing image sha256:fe21e5bc7c68a08f2601bdea81dc06e90237534c32665ac3  0.0s
 => => naming to docker.io/library/otot-a1-nodeserver                         0.0s
 => CACHED [otot-a1-nodeserver 2/5] WORKDIR /usr/src/app                      0.0s
 => CACHED [otot-a1-nodeserver 3/5] COPY package*.json ./                     0.0s
 => CACHED [otot-a1-nodeserver 4/5] RUN npm install                           0.0s
 => CACHED [otot-a1-nodeserver 5/5] COPY . .                                  0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[+] Running 3/3
 ⠿ Network otot-a1_default         Created                                    0.1s
 ⠿ Container otot-a1-nginx-1       Created                                    0.1s
 ⠿ Container otot-a1-nodeserver-1  Create...                                  0.1s
Attaching to otot-a1-nginx-1, otot-a1-nodeserver-1
otot-a1-nginx-1       | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
otot-a1-nginx-1       | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
otot-a1-nginx-1       | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
otot-a1-nginx-1       | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
otot-a1-nginx-1       | 10-listen-on-ipv6-by-default.sh: info: /etc/nginx/conf.d/default.conf differs from the packaged version
otot-a1-nginx-1       | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
otot-a1-nginx-1       | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
otot-a1-nginx-1       | /docker-entrypoint.sh: Configuration complete; ready for start up
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: using the "epoll" event method
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: nginx/1.23.1
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: built by gcc 11.2.1 20220219 (Alpine 11.2.1_git20220219)
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: OS: Linux 5.10.124-linuxkit
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: start worker processes
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: start worker process 30
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: start worker process 31
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: start worker process 32
otot-a1-nginx-1       | 2022/10/08 06:47:33 [notice] 1#1: start worker process 33
otot-a1-nodeserver-1  | Example app listening on port 8080
```

- run `docker-compose up --build` on terminal

5. View on localhost:8080



Looks like you've successfully rendered the 404 page

Matric No.: A0222182R

Name: Chen Hsiao Ting