2808ICT/7623ICT Assignment 2

Automate and Build Full-stack Project

Team: XXXTeam

Members:

Yip Hing Wa , Tom (s5252235)

Table of Contents

[Dockerfiles 3](#_Toc116640325)

[frontend/Dockerfile: 3](#_Toc116640326)

[mongo-express/Dockerfile: 3](#_Toc116640327)

[mongodb/Dockerfile: 3](#_Toc116640328)

[proxy/Dockerfile: 3](#_Toc116640329)

[Content of docker-compose.yml 8](#_Toc116640330)

# Dockerfiles

## frontend/Dockerfile:

#Each instruction in this file creates a new layer

#Here we are getting our node as Base image

FROM node:latest

#Creating a new directory for app files and setting path in the container

RUN mkdir -p /usr/src/app

#setting working directory in the container

WORKDIR /usr/src/app

#copying the package.json file(contains dependencies) from project source dir to container dir

COPY package.json /usr/src/app

# installing the dependencies into the container

RUN npm install

#copying the source code of Application into the container dir

#note you must be in the source code directory and remember COPY is recursive

COPY . /usr/src/app

#container exposed network port number

EXPOSE 3000

#command to run within the container

CMD ["node", "server.js"]

## mongo-express/Dockerfile:

#Here we are getting mongo-express as Base image

FROM mongo-express

#container exposed network port number

EXPOSE 8081

#command to run within the container

CMD ["mongo-express"]

## mongodb/Dockerfile:

#Here we are getting mongo as Base image

FROM mongo

#container exposed network port number

EXPOSE 27017

## proxy/Dockerfile:

#Here we are getting nginx as Base image

FROM nginx

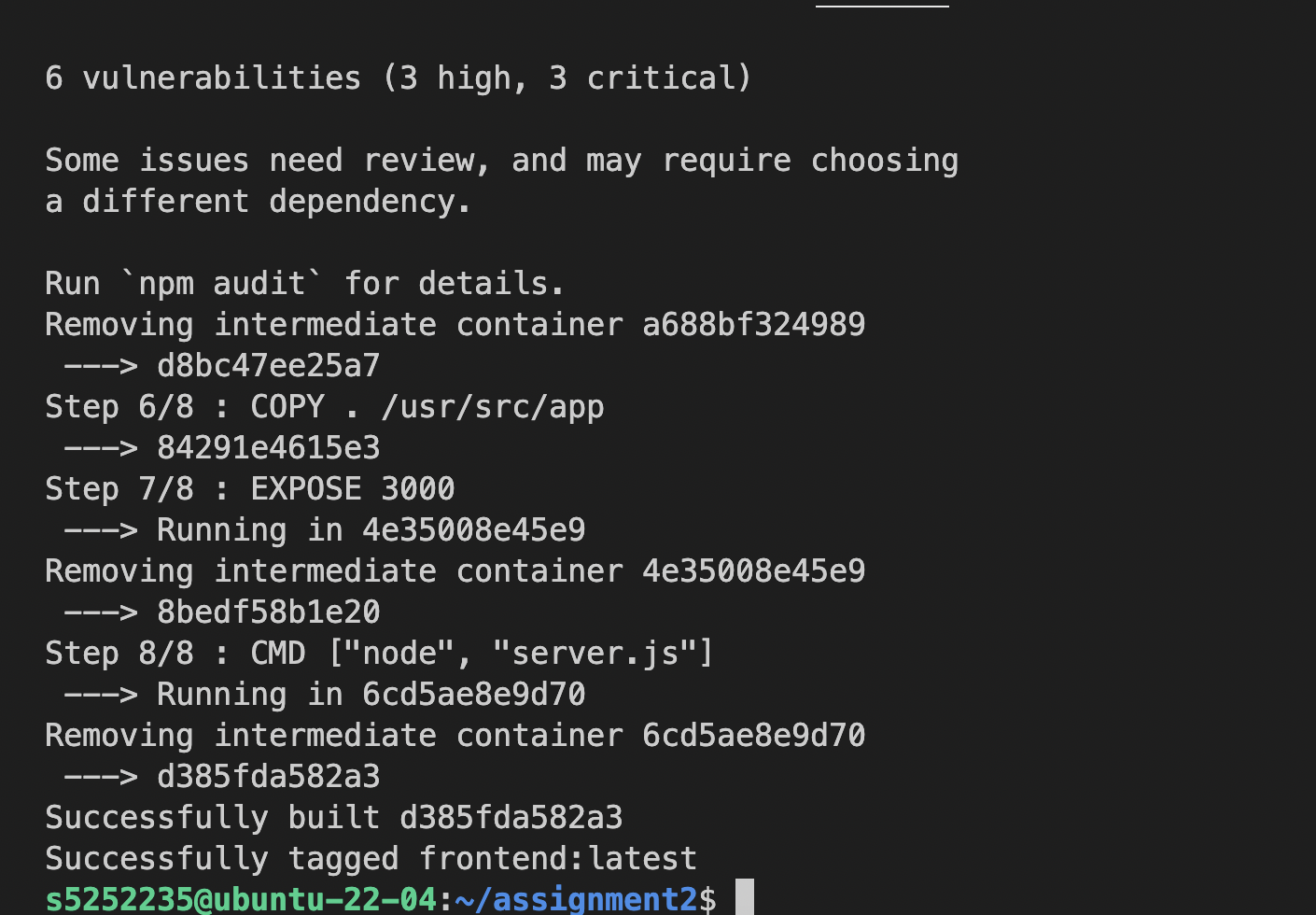


Fig 1a. Successful built of frontend images

Text

Description automatically generated

Fig 1b. Successful built of mongo-express images

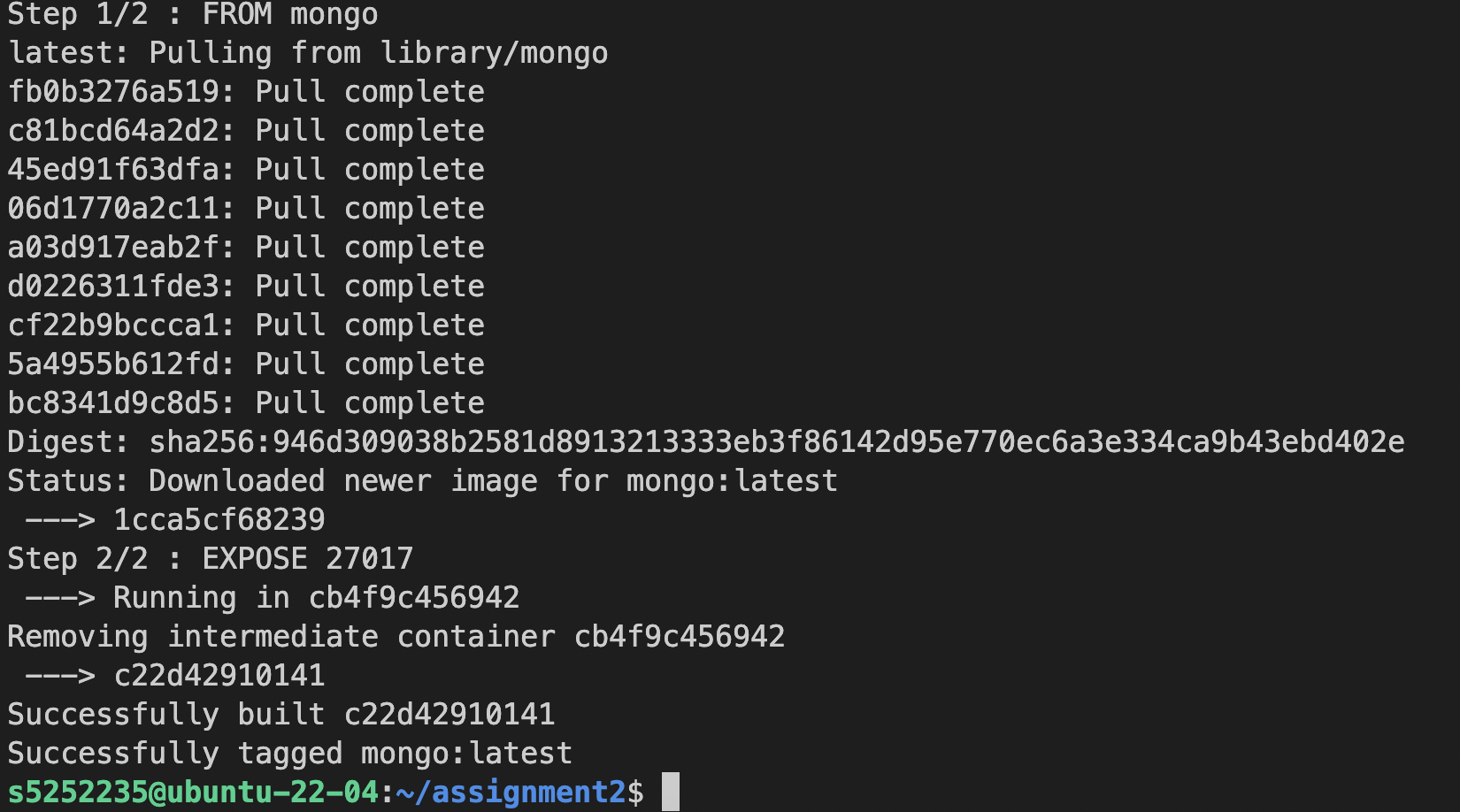


Fig 1c. Successful built of mongoDb images

Text

Description automatically generated

Fig 1d. Successful built of proxy images

Text

Description automatically generated

Fig 1e. Successful launch of docker-compose & proxy image

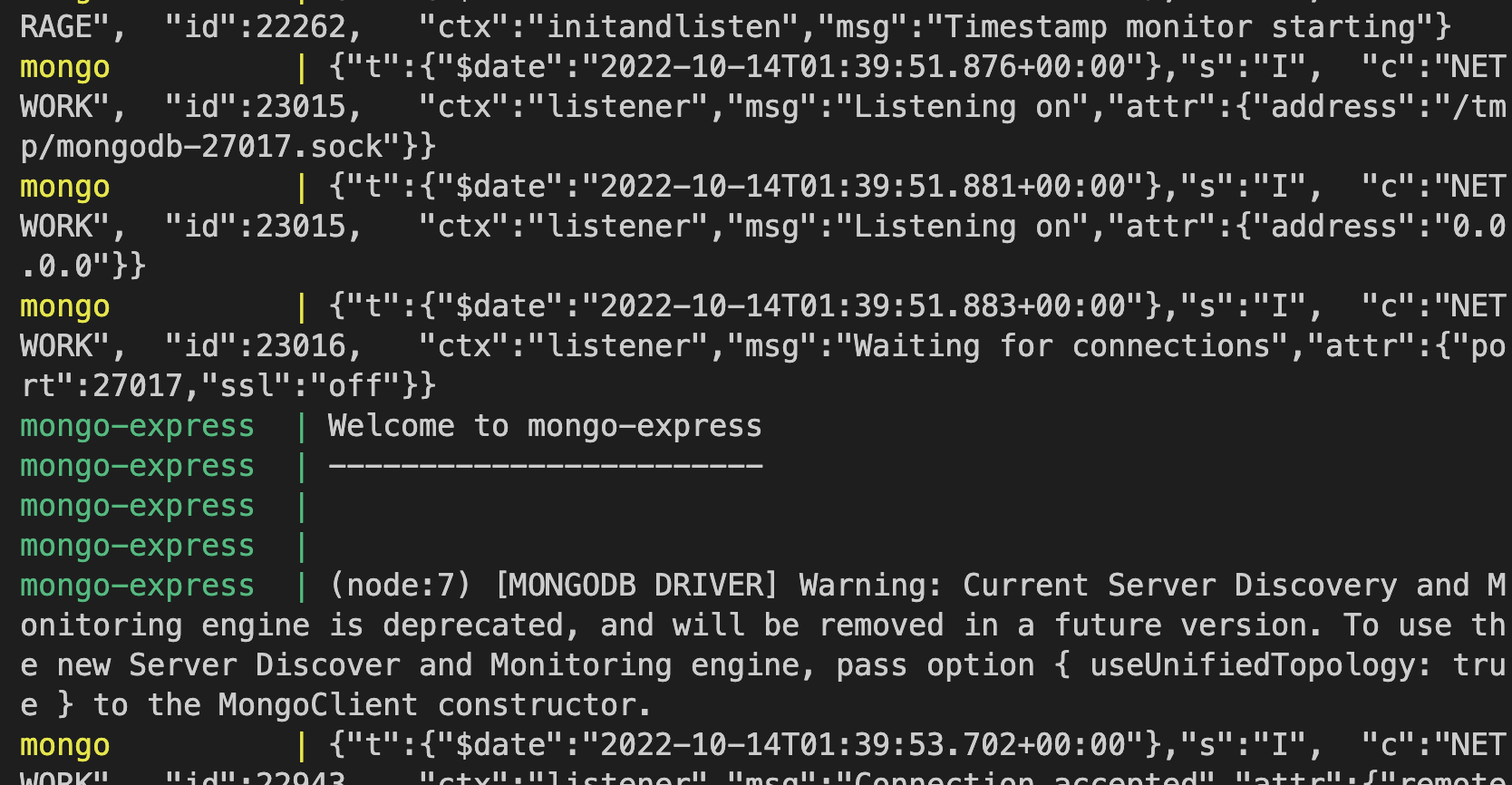


Fig 1f. Successful launch of mongo image & mongo-express images

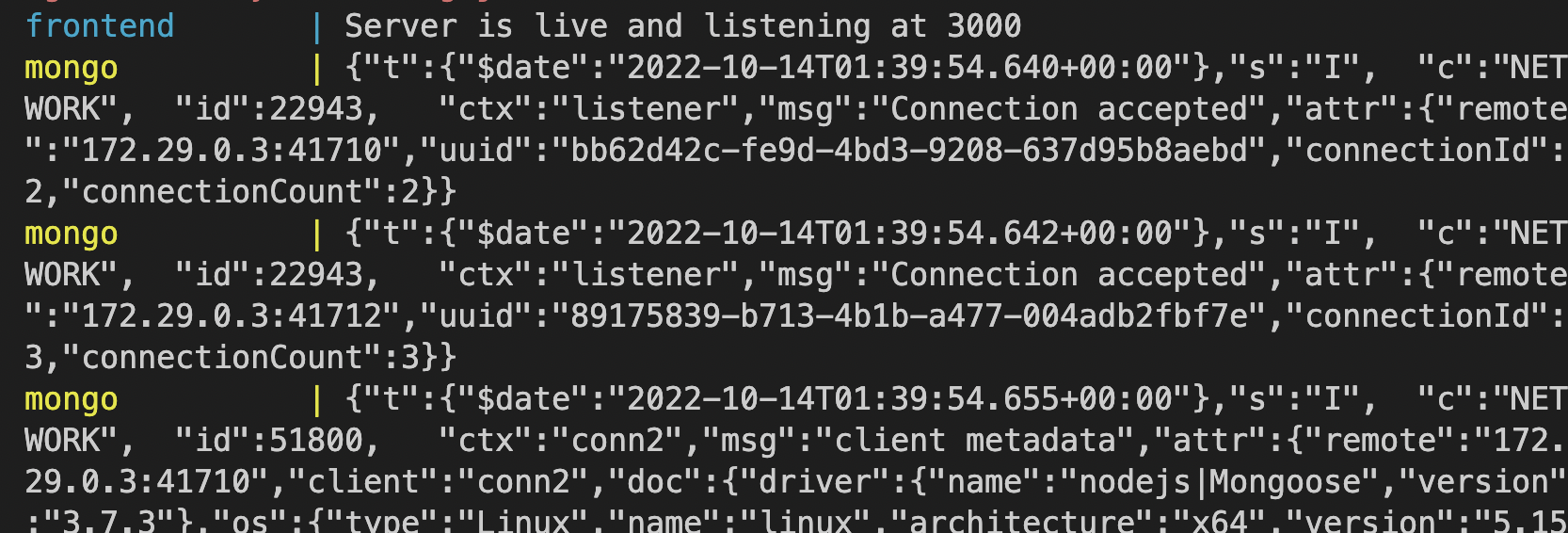


Fig 1g. Successful launch of frontend images

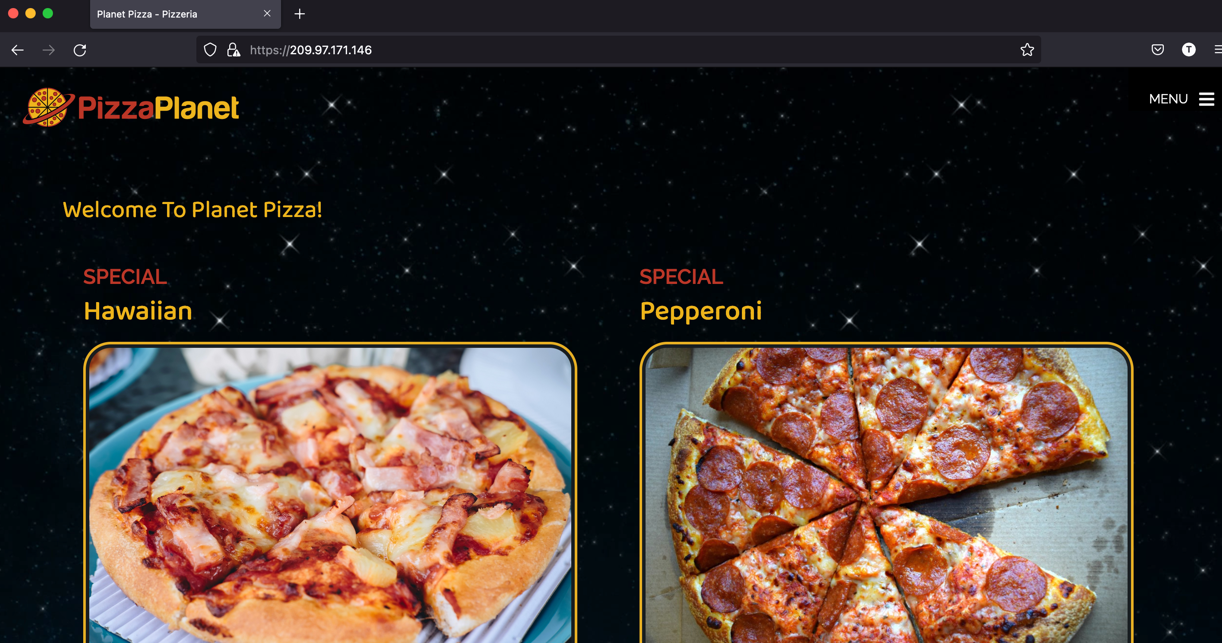


Fig 1h. Successful launch of frontend application



Fig 1g. Successful launch of mongo-express application

# Content of docker-compose.yml

version: "3.9"

services:

frontend:

container\_name: frontend

image: frontend

build: ./frontend

environment:

- MONGODB\_URI=mongodb://mongo:27017/pizzeria

- PORT=3000

- SECRET=aslbljwerljwer

depends\_on:

- mongo

networks:

- frontend

- backend

mongo:

container\_name: mongo

image: mongo

build: ./mongodb

ports:

- "27017:27017"

volumes:

- mongodata:/data/db

networks:

- backend

mongo-express:

container\_name: mongo-express

image: mongo-express

build: ./mongo-express

environment:

- ME\_CONFIG\_MONGODB\_SERVER=mongo

- ME\_CONFIG\_MONGODB\_PORT=27017

links:

- mongo

networks:

- frontend

- backend

proxy:

container\_name: proxy

image: nginx

build: .

volumes:

- ./nginx.conf:/etc/nginx/nginx.conf:ro

- ./ssl/localhost.crt:/etc/ssl/certs/localhost.crt

- ./ssl/localhost.key:/etc/ssl/private/localhost.key

ports:

- "3000:3000"

- "8081:8081"

- "443:443"

networks:

- frontend

volumes:

mongodata:

networks:

frontend:

backend:

internal: true