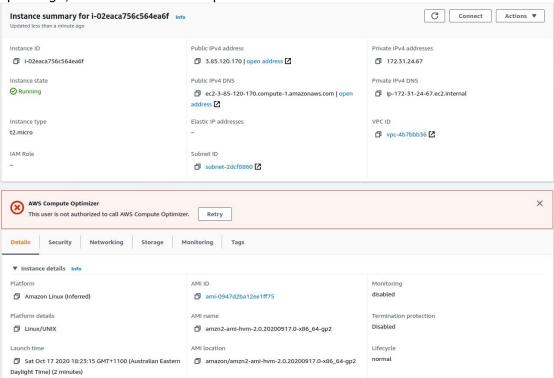
AWS Deployment

AWS deployment was done manually. As part of the setup, dockerfiles were created for the fronted and backend with a docker-compose written to pull them together into one deployable service.

```
FROM maven: 3.5.2-jdk-8-alpine AS MAVEN BUILD
MAINTAINER Rylee Randall
COPY pom.xml /build/
COPY src /build/src/
WORKDIR /build/
RUN mvn package -DskipTests
FROM openjdk:8-jre-alpine
WORKDIR /app
COPY --from=MAVEN BUILD /build/target/appointmentservicebackend-0.0.1-SNAPSHOT.jar /app/app.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "app.jar"]
# build environment
FROM node: 12.4.0-alpine as build
WORKDIR /app
ENV PATH /app/node modules/.bin:$PATH
COPY package.json /app/package.json
RUN npm install --silent
RUN npm install react-scripts@3.0.1 -g --silent
COPY . /app
ENV REACT APP PROD true
RUN npm run build
# production environment
FROM nginx:1.16.0-alpine
COPY --from=build /app/build /usr/share/nginx/html
RUN rm /etc/nginx/conf.d/default.conf
COPY ./nginx/nginx.conf /etc/nginx/conf.d
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

```
version: "3"
services:
 backend:
   image: "sept/backend"
   build:
      context: "./BackEnd/appointment-service-backend"
   container name: "backend"
   ports:
      - "8080:8080"
 frontend:
   image: "sept/frontend"
   build:
      context: "./FrontEnd/appointment-service"
   container name: "frontend"
   ports:
      - "80:80"
   depends on:
      - backend
```

An EC2 instance was created on Amazon AWS with ports 80 and 443 for web opened, and set up with git, docker and docker compose over ssh.



The repository was cloned and the docker images deployed with the docker-compose script, resulting in a running production deployment of both frontend and backend on AWS.

