Module 10 - Task 2

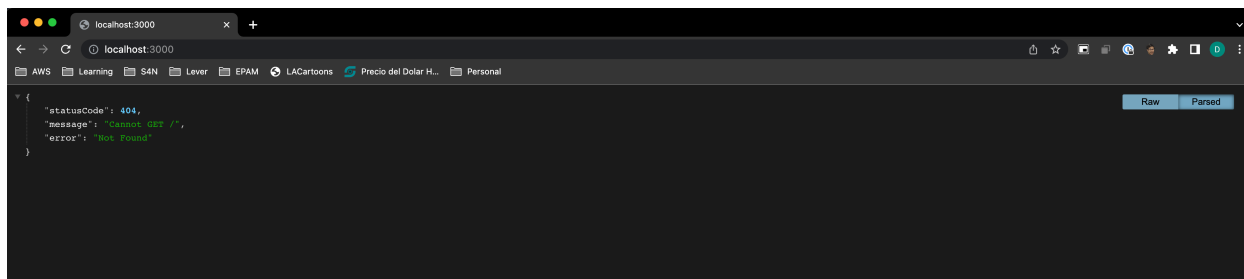1. Creating Dockerfile for the nestjs-rest-api application.

```
Dockerfile
1 FROM node:12.19.0-alpine3.9 AS development
2
3 WORKDIR /usr/src/app
4
5 COPY package*.json ./
6
7 RUN npm install glob rimraf
8
9 RUN npm install
10
11 COPY . .
12
13 RUN npm run build
14
15 FROM node:12.19.0-alpine3.9 as production
16
17 ARG NODE_ENV=production
18 ENV NODE_ENV=${NODE_ENV}
19
20 WORKDIR /usr/src/app
21
22 COPY package*.json ./
23
24 RUN npm install --only=production
25
26 COPY . .
27
28 COPY --from=development /usr/src/app/dist ./dist
29
30 CMD ["node", "dist/main"]
```

2. Creating docker-compose.yml

```
docker-compose.yml
1 version: '3.8'
2
3 services:
4     dev:
5         container_name: nestjs_api_dev
6         image: nestjs-api-dev:1.0.0
7         build:
8             context: .
9             target: development
10            dockerfile: ./Dockerfile
11        command: npm run start:dev
12        ports:
13            - 3000:3000
14            - 9229:9229
15        networks:
16            - nesjs-network
17        volumes:
18            - .:/usr/src/app
19            - /usr/src/app/node_modules
20        restart: unless-stopped
21    prod:
22        container_name: nestjs_api_prod
23        image: nestjs-api-prod:1.0.0
24        build:
25            context: .
26            target: production
27            dockerfile: ./Dockerfile
28        command: npm run start:prod
29        ports:
30            - 3000:3000
31            - 9229:9229
32        networks:
33            - nesjs-network
34        volumes:
35            - .:/usr/src/app
36            - /usr/src/app/node_modules
37        restart: unless-stopped
38
39 networks:
40     nesjs-network:
```

3. Runing the container with *docker-compose.*

Publishing image in private registry (locally)

1. Running the Docker registry image

2. Pushing image to the registry

3. Pulling image from the registry and running it

```
~/Desktop/Learning/devops-fundamentals-course/tasks/module_10/nestjs-rest-api  main ?4  docker ps
CONTAINER ID   IMAGE        COMMAND             CREATED        STATUS        PORTS                                         NAMES
d209d488e7f0   registry:2   "/entrypoint.sh /etc…"   4 minutes ago   Up 4 minutes   0.0.0.0:5001->5000/tcp, :::5001->5000/tcp   registry
~/Desktop/Learning/devops-fundamentals-course/tasks/module_10/nestjs-rest-api  main ?4  docker pull localhost:5001/daniel-nestjs-api-prod
Using default tag: latest
latest: Pulling from daniel-nestjs-api-prod
Digest: sha256:35ba8e3c930839f3d9bc34c9d2cc57bb71c27a0ce23cdeecdb9363362bcb1bde
Status: Image is up to date for localhost:5001/daniel-nestjs-api-prod:latest
localhost:5001/daniel-nestjs-api-prod:latest
~/Desktop/Learning/devops-fundamentals-course/tasks/module_10/nestjs-rest-api  main ?4  docker run -d -p 3000:3000 localhost:5001/daniel-nestjs-api-prod
3ce89996ec33532f190456672e8707f195725279ed787a561d31f39ac2ffc60c
~/Desktop/Learning/devops-fundamentals-course/tasks/module_10/nestjs-rest-api  main ?4  docker ps
CONTAINER ID   IMAGE                                  COMMAND             CREATED        STATUS        PORTS                                         NAMES
3ce89996ec33   localhost:5001/daniel-nestjs-api-prod   "docker-entrypoint.s…"   2 seconds ago   Up 1 second    0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   vigilant_bhaskara
d209d488e7f0   registry:2                             "/entrypoint.sh /etc…"   6 minutes ago   Up 6 minutes   0.0.0.0:5001->5000/tcp, :::5001->5000/tcp   registry
~/Desktop/Learning/devops-fundamentals-course/tasks/module_10/nestjs-rest-api  main ?4 
```