

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Бэк-энд разработка

Отчет

Лаб № 4: Docker, docker compose, docker swarm

Выполнил:

Вали Насибулла

Группа К33402

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

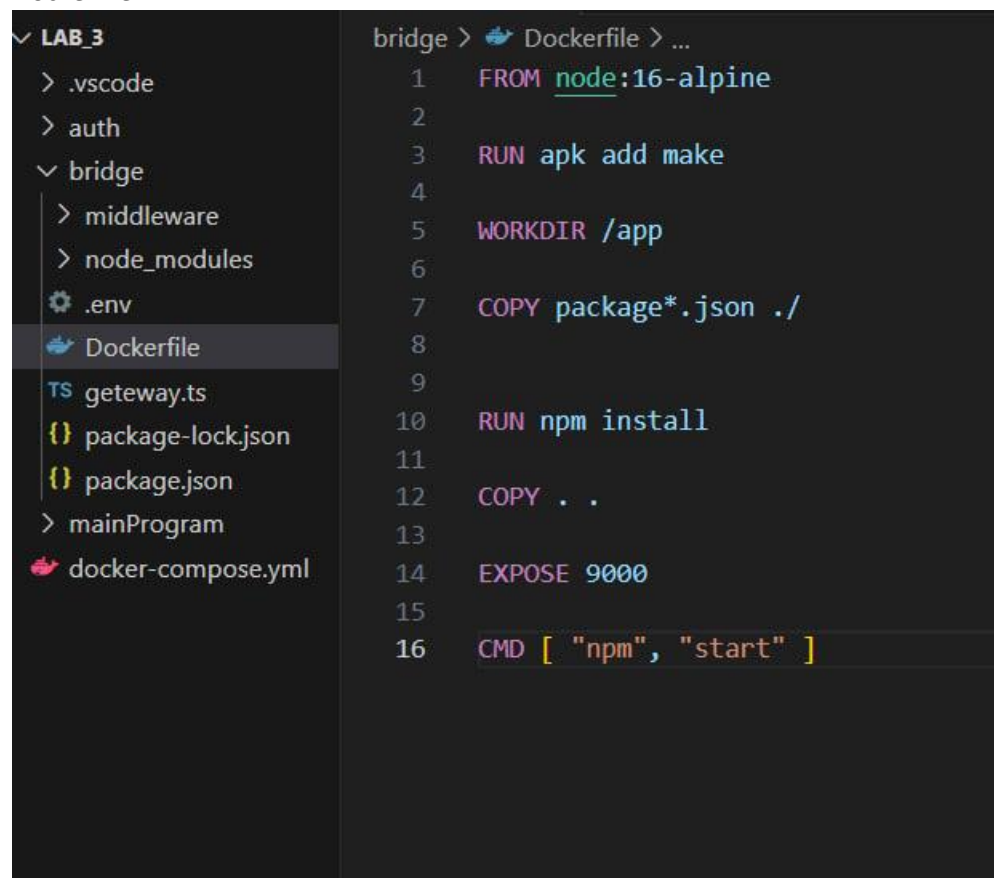
## Задание:

Необходимо упаковать ваше приложение в docker-контейнеры и обеспечить сетевое взаимодействие между различными частями вашего приложения. Делать это можно как с помощью docker-compose

## Ход работы

[https://github.com/Naseebullah-Wali/ITMO-ICT-Backend-2023/tree/master/labs/K33402/Naseebullah\\_Wali/Lab\\_4](https://github.com/Naseebullah-Wali/ITMO-ICT-Backend-2023/tree/master/labs/K33402/Naseebullah_Wali/Lab_4)

Dockerfile:



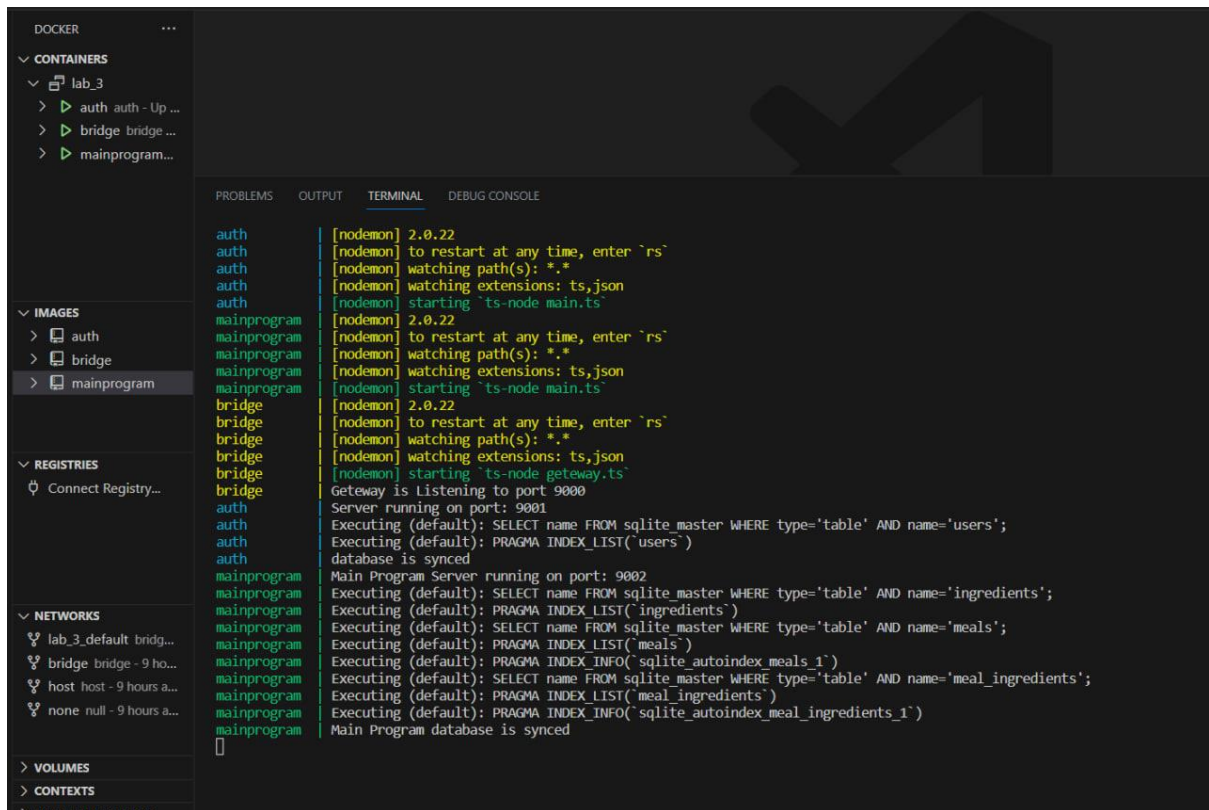
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a directory structure for 'LAB\_3' with files like '.vscode', 'auth', 'bridge', 'middleware', 'node\_modules', '.env', 'Dockerfile', 'getaway.ts', 'package-lock.json', 'package.json', 'mainProgram', and 'docker-compose.yml'. The 'Dockerfile' file is selected. The code editor shows the content of the 'Dockerfile' file, which is a Dockerfile for a Node.js application. The code is as follows:

```
bridge > Dockerfile > ...
1 FROM node:16-alpine
2
3 RUN apk add make
4
5 WORKDIR /app
6
7 COPY package*.json ./
8
9
10 RUN npm install
11
12 COPY . .
13
14 EXPOSE 9000
15
16 CMD [ "npm", "start" ]
```

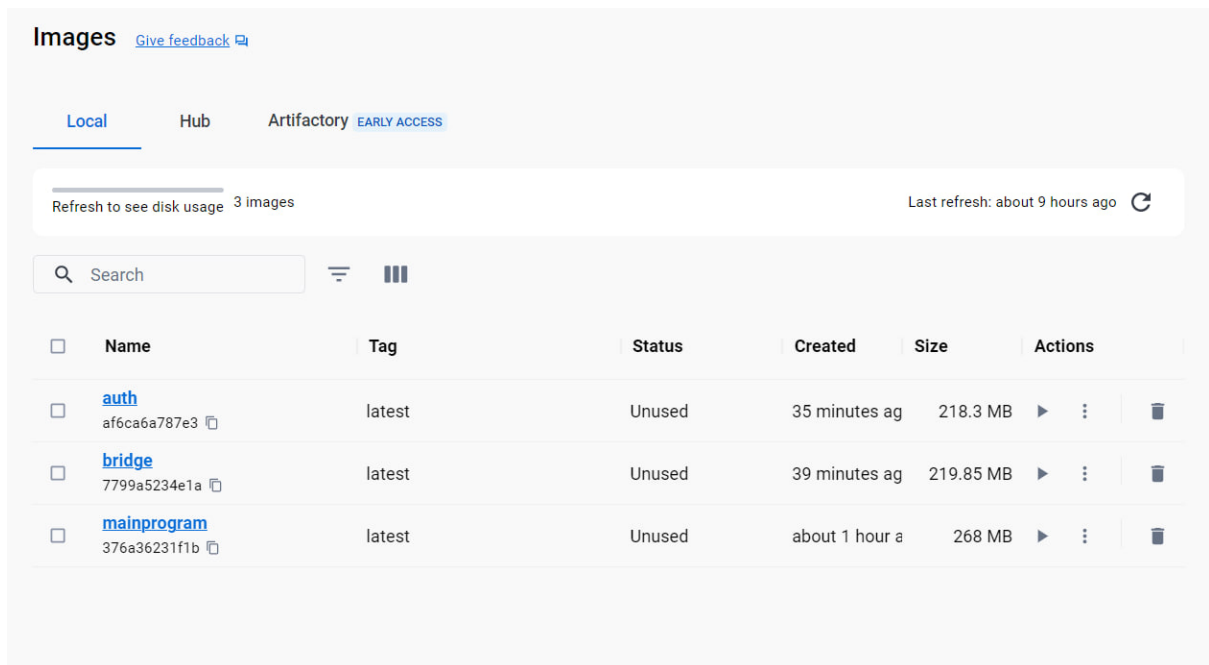
Docker-compose.yml

```
version: '3.4'
services:
  mainprogram:
    container_name: mainprogram
    image: mainprogram
    build:
      context: mainProgram
    environment:
      NODE_ENV: production
    command: npm start
    ports:
      - 9002:9002
    restart: always
  auth:
    env_file:
      - ./auth/.env
    container_name: auth
    image: auth
    build:
      context: auth
    environment:
      NODE_ENV: production
    ports:
      - 9001:9001
    restart: always











  bridge:
    container_name: bridge
    image: bridge
    build:
      context: bridge
    environment:
      NODE_ENV: production
    ports:
      - 9000:9000
    restart: always
```



## Images dashboard



## Container dashboard

<input type="checkbox"/>	<div><div></div><div><div>lab_3</div></div></div>		Running (3/3)	2 minutes ago	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>	
<input type="checkbox"/>	<div><div></div><div><div>bridge</div><div>084fe016318f </div></div></div>	<div><div>bridge</div></div>	Running	<div><div>9000:9000</div><div></div></div>	4 minutes ago	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>auth</div><div>20d15acb65ac </div></div></div>	<div><div>auth</div></div>	Running	<div><div>9001:9001</div><div></div></div>	4 minutes ago	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<div><div></div><div><div>mainprogram</div><div>ef938789c1b6 </div></div></div>	<div><div>mainprogram</div></div>	Running	<div><div>9002:9002</div><div></div></div>	2 minutes ago	<div><div></div><div></div><div></div></div>	<div><div></div><div></div><div></div></div>

## Вывод:

после выполнения этой лабораторной работы мы познакомились с докером и изучили некоторые основные команды, которые используются при создании контейнеров и image докеров. и узнать, использовать его.