

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Лабораторная работа 4

Выполнили:

Красюк Карина, Прокопец Семён

Группы К3341, К3339

Проверил:

Добряков Д. И.

Санкт-Петербург

2025 г.

Задача

- реализовать Dockerfile для каждого сервиса;
- написать общий docker-compose.yml;
- настроить сетевое взаимодействие между сервисами.

Ход работы

Докеры:

```
version: '3.8'

services:
  postgres:
    image: postgres:17
    container_name: jobboard-postgres
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: job_board
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data
    restart: unless-stopped

  api-gateway:
    build: ./api-gateway
    container_name: jobboard-api-gateway
    ports:
      - "3000:3000"
    depends_on:
      - user-resume-service
      - company-vacancy-service
      - industry-service
    restart: on-failure

  user-resume-service:
    build:
      context: ./user-resume-service
      target: development
    container_name: jobboard-user-resume
    ports:
      - "3001:3001"
    environment:
```

```

    DATABASE_URL:
"postgresql://postgres:postgres@postgres:5432/job_board?schema=public"

    NODE_ENV: development
    depends_on:
      - postgres
    restart: on-failure

company-vacancy-service:
  build:
    context: ./company-vacancy-service
    target: development
  container_name: jobboard-company-vacancy
  ports:
    - "3002:3002"
  environment:
    DATABASE_URL:
"postgresql://postgres:postgres@postgres:5432/job_board?schema=public"

    NODE_ENV: development
    depends_on:
      - postgres
    restart: on-failure

industry-service:
  build:
    context: ./industry-service
    target: development
  container_name: jobboard-industry
  ports:
    - "3003:3003"
  environment:
    DATABASE_URL:
"postgresql://postgres:postgres@postgres:5432/job_board?schema=public"

    NODE_ENV: development
    depends_on:
      - postgres
    restart: on-failure

volumes:
  postgres_data:

```

Dockerfile для каждого сервиса:

```

FROM node:20-alpine as development

WORKDIR /app

COPY package*.json ./

```

```
RUN npm install --include=dev
```

```
COPY . .
```

```
RUN npx prisma generate
```

```
RUN npm run build
```

```
EXPOSE 3001
```

```
CMD ["yarn", "start"]
```

```
FROM node:20-alpine as development
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install --include=dev
```

```
COPY . .
```

```
RUN npx prisma generate
```

```
RUN npm run build
```

```
EXPOSE 3003
```

```
CMD ["yarn", "start"]
```

```
FROM node:20-alpine as development
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm install --include=dev
```

```
COPY . .
```

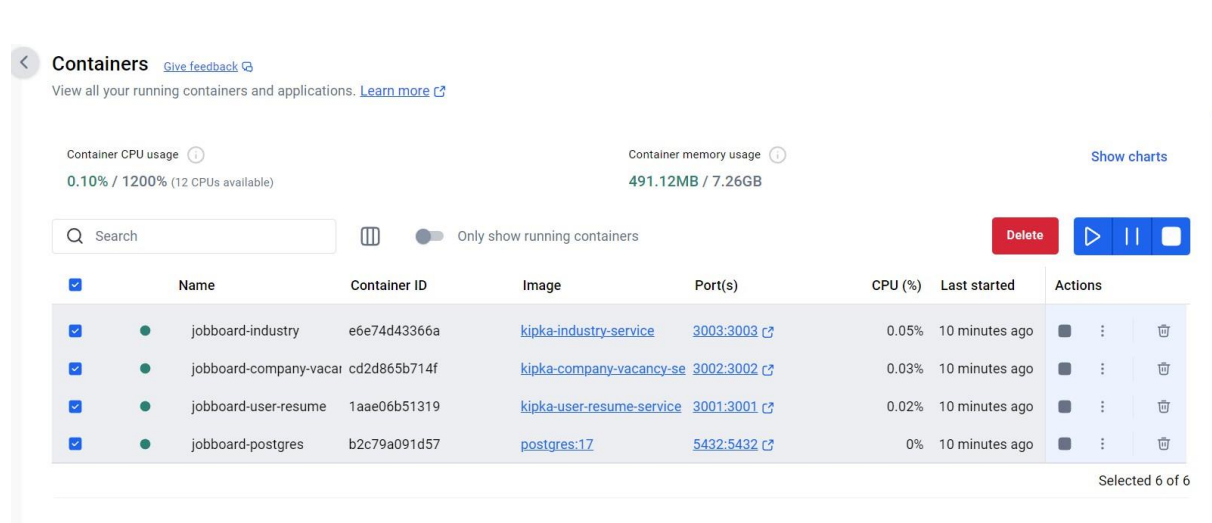
```
RUN npx prisma generate
```

```
RUN npm run build
```

```
EXPOSE 3002
```

```
CMD ["yarn", "start"]
```

Сетевое взаимодействие между сервисами:



Вывод

В данной лабораторной работе мы завернули всё в докер.