

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бек-энд разработка

Отчет

Лабораторная работа №3 и №4

“Разделение приложения на микросервисы”

Выполнил:
Стукалов Артем
К33392

Проверил:
Добряков Д. И.

Санкт-Петербург

2024 г.

Задание:

В рамках данной лабораторной работы необходимо разделить приложение написанное в лабораторной работе №2 на микросервисы. Также необходимо упаковать приложение в докер контейнер и настроить CI через github actions для развертывания приложения на удаленной машине.

Разделение на микросервисы:

Проведем разделение на микросервисы посредством превращения обычного репозитория в моно репозиторий. В качестве раннера будем использовать turbo. Всю общую логику связанную с инициализацией базового сервера fastify вынесем в shared модуль. Разделим приложение на 2 сервиса: сервис авторизации, отвечающий за работу с jwt и пользователем, и сервис девайсов, отвечающий за все остальное. Получим следующую структуру проекта.



Упаковка в docker

Для каждого сервиса напишем свой докер файл, каждый из которых будет отличаться только выбором конечного билда. Запускать будем все через docker-compose. Также подними внутри базу данных и брокер сообщений. Конечный файл выглядит вот так:

```
version: '3'
services:
  db:
    image: postgres:16-alpine
    container_name: postgres
    environment:
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=postgres_pwd
    volumes:
      - ./postgres:/var/lib/postgresql/data
    expose:
      - 5432
    restart: always

  rabbitmq:
    image: rabbitmq
    expose:
      - 5672

  auth:
    build:
      context: ./
      dockerfile: Dockerfile.auth
    tty: true
    env_file:
      - .env
    ports:
      - 3000:3000
    depends_on:
      - db
    command: sh -c "pnpm turbo db:migrate:deploy && cd api-services/auth && node dist/index.js"
    healthcheck:
      test: ['CMD', 'pg_isready']
      interval: 10s
      timeout: 5s
      retries: 5

  devices:
    build:
      context: ./
      dockerfile: Dockerfile.devices
    tty: true
    env_file:
      - .env
    ports:
```

CI/CD

Финально сделаем возможным автоматический деплой приложения. Напишем следующий сценарий запуска, в рамках которого новая версия посредством ssh будет доставляться на сервер и запускать повторную сборку docker-compose.

```
name: Deploy

on:
  push:
    branches:
      - astukalov/feature/lab-3

jobs:
  deploy:
    runs-on: ubuntu-latest
    env:
      VPS_USER_NAME: ${ secrets.vps_user_name }
      VPS_SERVER_NAME: ${ secrets.vps_server_name }
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-node@v1
        with:
          node-version: 20.x
      - name: Setup
        run: |
          sudo apt-get update
          sudo apt-get install rsync -y
          mkdir "$HOME/.ssh"
          echo "${ secrets.vps_private_key }" > "$HOME/.ssh/key"
          chmod 600 "$HOME/.ssh/key"
      - name: Delivery
        run: |
          rsync -e "ssh -p 23348 -i $HOME/.ssh/key -o StrictHostKeyChecking=no" --archive --compress --delete .
            $VPS_USER_NAME@$VPS_SERVER_NAME:/home/tutor03/Documents/itmo-back-ts
      - name: Building on server
        run: |
          ssh -p 23348 -i $HOME/.ssh/key $VPS_USER_NAME@$VPS_SERVER_NAME 'cd /home/tutor03/Documents/itmo-back-ts/Labs/K33392/Стуканов\
            Артём/lab-3/ && docker compose up -d --build'
```

You, 49 minutes ago • fix: changed path