|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-14 «Интеллектуальные системы информационной безопасности»

### Администрирование баз данных

### Практическая работа № 3

# ОТЧЁТ

# Выполнил студент группы БСБО-07-20 Любовский С.В.

# Выполнение задания.

Ссылка на репозиторий с проектом - <https://github.com/wallseat/MIREA_database_administration/tree/main/practice3>

1.­­ Создадим приложение, которое будет иметь 6 ручек (получения списка сущностей, создание и удаление сущности) для взаимодействия с сущностями Employee и Product. Будем использовать фреймворк FastAPI для реализации API с функций само-документации, Nginx в качестве балансировщика нагрузки и reverse-proxy, базу данных MongoDB для хранения информации о сущностях Products и базу данных PostgreSQL для хранения информации о сущностях Employee.



Рисунок 1. Интерфейс Swagger для описанной API

2. Опишем структуру деплоя нашего приложения в файле docker-compose.yml. Будем использовать .env файл для хранения секретов, а так же volume для персистентной информации. Для изменения количества реплик нашего сервера воспользуемся директивой deploy:replicas.

version: '3.9'

services:

  postgres:

    image: postgres:15.2-alpine

    container\_name: ${POSTGRES\_HOST}

    restart: unless-stopped

    environment:

      POSTGRES\_USER: ${POSTGRES\_USER}

      POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD}

      POSTGRES\_DB: ${POSTGRES\_DB}

    volumes:

      - ./data/postgres:/var/lib/postgresql/data

  mongo:

    image: mongo:4.4.4-bionic

    container\_name: ${MONGO\_HOST}

    restart: unless-stopped

    environment:

      MONGO\_INITDB\_ROOT\_USERNAME: ${MONGO\_USER}

      MONGO\_INITDB\_ROOT\_PASSWORD: ${MONGO\_PASSWORD}

      MONGO\_INITDB\_DATABASE: ${MONGO\_DB}

    volumes:

      - ./data/mongo:/data/db

  app:

    build: .

    restart: unless-stopped

    environment:

      POSTGRES\_USER: ${POSTGRES\_USER}

      POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD}

      POSTGRES\_DB: ${POSTGRES\_DB}

      POSTGRES\_HOST: ${POSTGRES\_HOST}

      POSTGRES\_PORT: ${POSTGRES\_PORT}

    expose:

      - 8080

    depends\_on:

      - postgres

    command: uvicorn src.main:app --host 0.0.0.0 --port 8080

    deploy:

      mode: replicated

      replicas: 3

  nginx:

    image: nginx:latest

    volumes:

      - ./nginx.conf:/etc/nginx/nginx.conf:ro

    depends\_on:

      - app

    ports:

      - "8080:8080"

3. Для запуска всех сервисов воспользуемся командой **docker compose up -d**

Листинг 1. Листинг файла docker-compose.yml

4. Для обновления всех сервисов воспользуемся командой **docker compose up -d –force-recreate –build**

5. Для остановки сервисов и освобождения системных ресурсов воспользуемся командой **docker compose down -v && sudo rm -rf ./data**