# Лабораторная работа №7

**Реализация взаимодействия между микросервисами**

1. **Цель работы**
   1. Изучить процесс разработки микросервисной архитектуры.
   2. Изучить процесс развертывания микросервисной архитектуры.
2. **Литература**
   1. Зверева В. П., Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. Образования / В. П. Зверева, А. В. Назаров. – М. : Издательский центр «Академия», 2018. – 256 с.
3. **Подготовка к работе**
   1. Повторить теоретический материал (см. п.2).
   2. Изучить описание лабораторной работы.
4. **Основное оборудование**
   1. Персональный компьютер.
5. **Задание**
   1. Настроить решение из лабораторной работы для работы с контейнерами.
      1. В контекстном меню обозревателя решений во всех проектах выбрать опцию **Добавить > Поддержка оркестратора контейнеров**
      2. Получившаяся структура решения должна выглядеть следующим образом:

Решение/

ApiGateway/

ProductService/

OrderService/

docker-compose.yml

* 1. В файл docker-compose вписать конфигурацию для развертывания разработанной системы. В листинге представлена часть конфигурации.

|  |
| --- |
| services:  apigateway:  container\_name: "apigateway"  image: ${DOCKER\_REGISTRY-}apigateway  build:  context: .  dockerfile: ApiGateway/Dockerfile  environment:  - ASPNETCORE\_ENVIRONMENT=Development  ports:  - "5000:8080"  depends\_on:  - productservice  links:  - productservice  networks:  - net  productservice:  container\_name: "productservice"  image: ${DOCKER\_REGISTRY-}productservice  build:  context: .  dockerfile: ProductService/Dockerfile  environment:  - ASPNETCORE\_ENVIRONMENT=Development  - CONNECTION\_STRING=Server=productdb;Database=Products;User=root;Password=root;Port=3306  depends\_on:  - productdb  links:  - productdb  networks:  - net  productdb:  container\_name: "productdb"  image: mysql:latest  restart: always  environment:  MYSQL\_ROOT\_PASSWORD: root  MYSQL\_DATABASE: Products  MYSQL\_USER: root  MYSQL\_PASSWORD: root  volumes:  - productdb:/var/lib/mysql  networks:  - net  volumes:  productdb:  networks:  net: |

* 1. Поменять настройки Api-шлюза для корректной работы с docker

|  |
| --- |
| {  "Routes": [  {  "DownstreamPathTemplate": "/api/{everything}",  "DownstreamHostAndPorts": [  {  "Host": "productservice",  "Port": 8080  }  ],  "UpstreamPathTemplate": "/{everything}",  "UpstreamHttpMethod": [ "Get", "Post", "Put", "Delete" ]  }  //остальные методы  ],  "GlobalConfiguration": {  "BaseUrl": "http://localhost:5000"  }  } |

* 1. Запустить проект и проверить корректность работы API.
  2. Составить отчет по проделанной работе.

1. **Порядок выполнения работы**
   1. Повторить теоретический материал п. 3.1;
   2. Выполнить задания 5.1-5.
   3. Ответить на контрольные вопросы п. 8;
   4. Заполнить отчет п. 7.
2. **Содержание отчета**
   1. Титульный лист;
   2. Цель работы;
   3. Таблица п. 5.2
   4. Ответы на контрольные вопросы п. 6.3;
   5. Вывод по проделанной работе.