

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**"МИРЭА - Российский технологический университет"**

**РТУ МИРЭА**



Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6**

**по дисциплине**

**«**НАСТРОЙКА И АДМИНИСТРИРОВАНИЕ СЕРВИСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**»**

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-20-21 | Хитров Н. С. |
|  |  |
| Принял к.э.н. доцент | Хозяинов А.В. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | « » 2023 г. |  |
|  |  |
| «Зачтено» | « » 2023 г. |  |

Москва 2023

# Практическая работа №6. Создание контейнеров Docker

## Цель работы

Получить навыки сборки Docker-контейнера с использованием Docker-composer.

## Теоретическое введение

Docker-compose позволяет запускать несколько контейнеров,  
связывать их и определять различные свойства контейнера в одном файле.

Этот файл называется docker-compose.yml.

Docker Compose управляет контейнерами, запускает их вместе, в  
нужной последовательности.

Docker-compose написан в формате YAML который по своей сути  
похож на JSON или XML. В формате YAML имеют значения пробелы и  
табуляции, именно пробелами отделяются названия параметров от их  
значений.

Команды Docker-compose

Чтобы просмотреть команды и их действия, необходимо выполнить в  
терминале :

$ docker-compose –help

Docker Compose Up

Эта команда используется, для запуска или служб в файле docker-  
compose.yml.

Файл Docker-compose.yml определяет сервисы, их свойства,  
переменные и зависимости.

$ docker-compose up

Для просмотра запущенных контейнеров:

$ docker-compose ps

Также можно указать docker-compose запустить только один сервис,  
например

$ docker-compose up my\_container1

## Ход работы

На рисунке 1 представлен файл docker-compose.yml

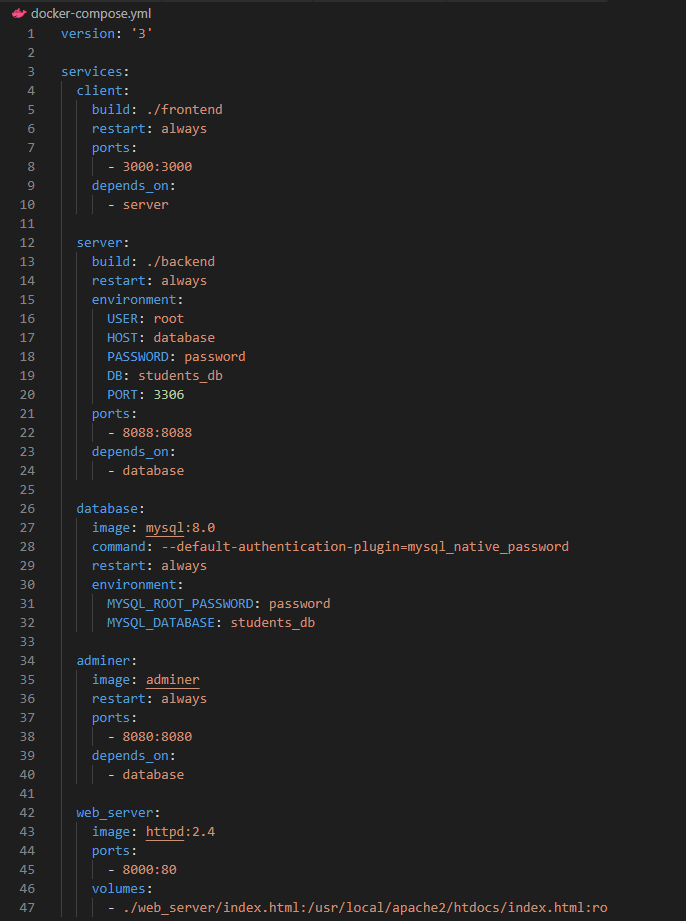


Рисунок 1 – Файл docker-compose.yml

На рисунке 2 представлен Dockerfile для серверной части приложения.

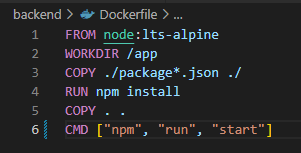


Рисунок – Dockerfile для собственного приложения

На рисунке 3 представлен Dockerfile для клиентской части приложения.

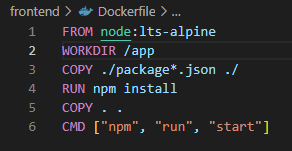


Рисунок 3 – Dockerfile для собственного приложения

На рисунке 4 представлена сборка образов.

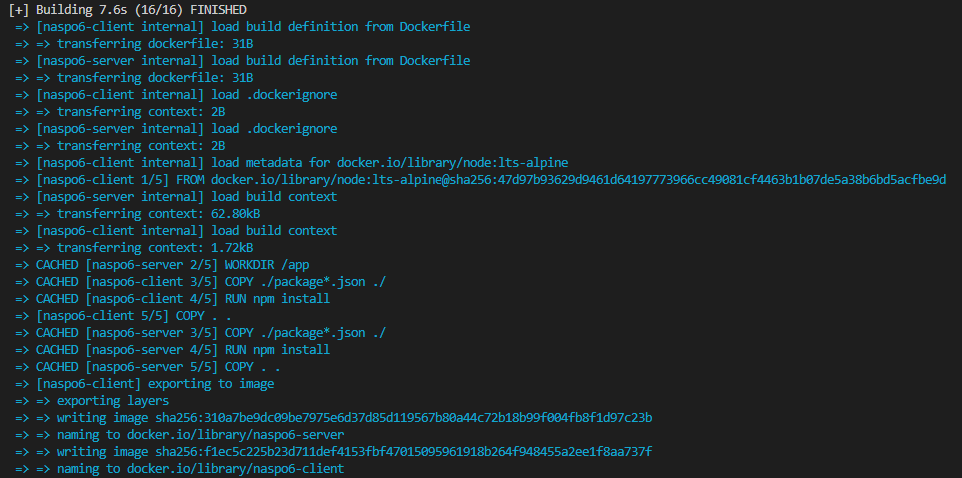


Рисунок 4 – Сборка образов

На рисунке 5 представлен создания контейнеров.

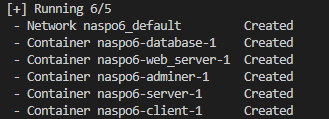


Рисунок 5 – создание контейнеров

На рисунке 6 страница из веб-сервера Apache.

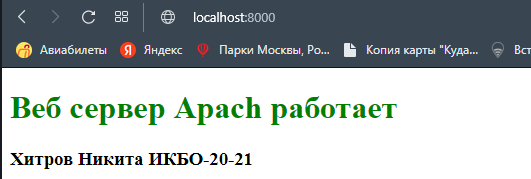


Рисунок 6 – Веб-сервер Apache

На рисунке 7 представлен страница Adminer, подключенная к базе данных MySQL.

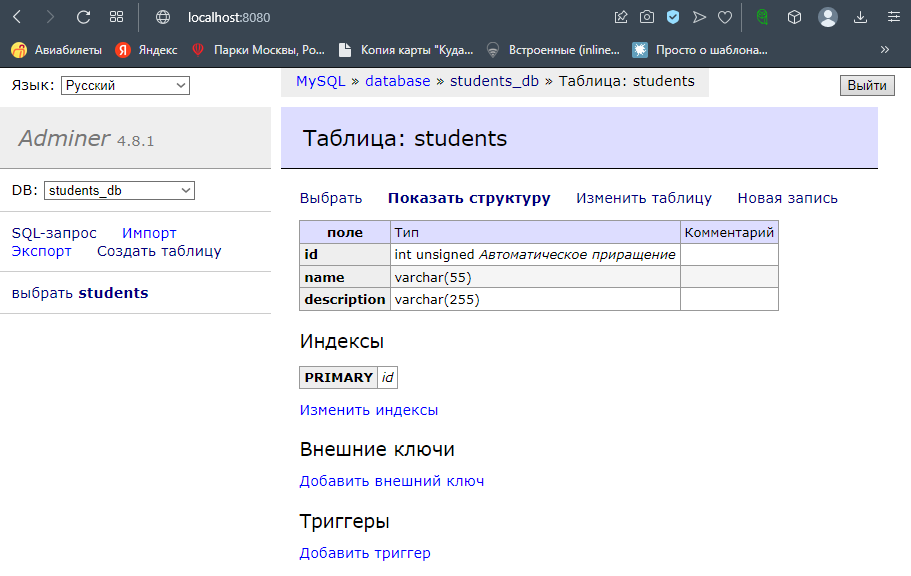


Рисунок 7 – Adminer

На рисунке 8 представлена страница собственного приложения на Express.js.

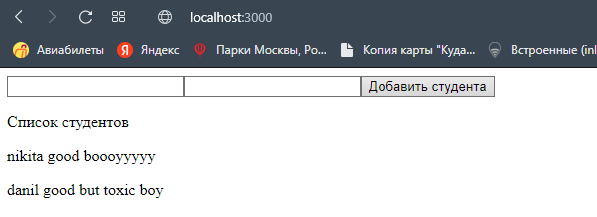


Рисунок 8 – Приложение Express.js

На рисунках 9–10 представлен код серверной части приложения на Express.js

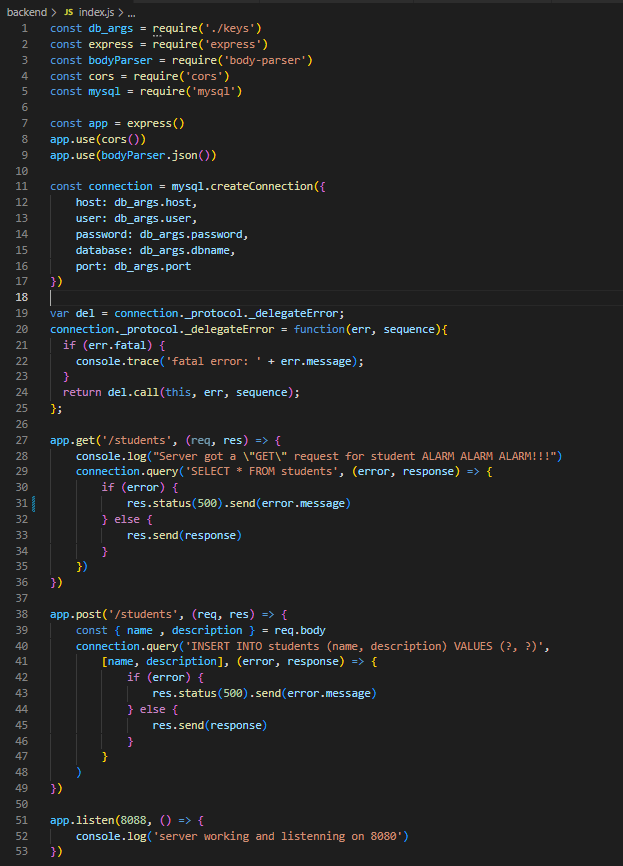


Рисунок 9 – Код серверной части приложения (часть 1)

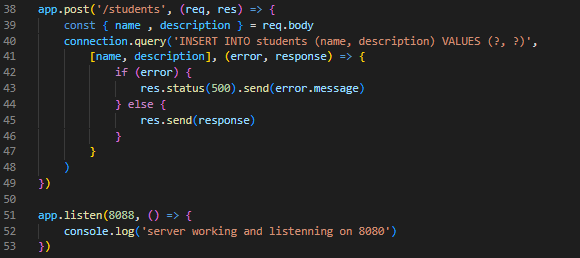


Рисунок 10 – Код серверной части приложения (часть 2)

На рисунке 11 представлен код клиентской части приложения на React.



Рисунок 11 – Код клиентской части приложения

Задание на последующую практическую: REST API приложение для сервиса аренды автомобилей.

# Вывод

В данной практической работе мы получили навыки сборки Docker-контейнера с использованием Docker-composer.

# Список использованных источников

1. Docker Documentation: сайт. – URL: <https://docs.docker.com/>
2. Express JS: сайт. – URL: https://expressjs.com/ru/