

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 практической и прикладной информатики (ППИ)

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №7**

**по дисциплине**

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

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

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

Москва 2023

**Цель работы:** получить навыки сборки Docker-контейнера с использованием Docker-compose.

# Теоретическое введение

Docker применяется для управления отдельными контейнерами (сервисами), из которых состоит приложение.

Применяя Docker, на пользовательской системе появляется возможность отделить компоненты разработки от системы и друг от друга. Таким образом, можно не устанавливать веб-сервер в операционную систему, а сделать несколько серверов в отдельных каталогах и потом в эти каталоги складывать проекты, которые будут работать в готовом окружении, которое можно запускать, когда понадобится.

Docker Compose используется для одновременного управления несколькими контейнерами, входящими в состав приложения. Этот инструмент предлагает те же возможности, что и Docker, но позволяет работать с более сложными приложениями.

Пример. Требуется перевести проект на другую машину.

Без применения Docker Compose придётся переносить и перенастраивать сервисы по одному.

Если применяется Docker Compose, то перенос проекта на новый сервер происходит достаточно быстро. Для того чтобы завершить перенос проекта на новое место, нужно выполнить настройки и загрузить на новый сервер резервную копию базы данных.

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.

# Выполнение работы

1. **Повторить пример, который был рассмотрен в разделе «Общие сведения». Проект назвать docker\_py**

На рисунке 1.1 показана структура проекта.

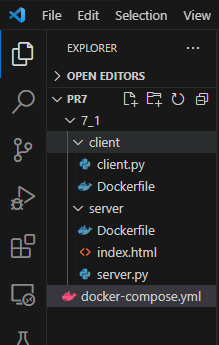


Рисунок 1.1 – Структура проекта

На рисунке 1.2 показан листинг файла client.py. На рисунке 1.3 показан Dockerfile для создания образа клиентской части.

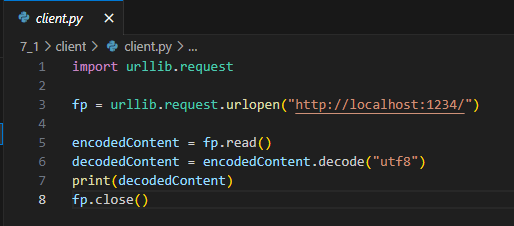


Рисунок 1.2 – Листинг файла client.py

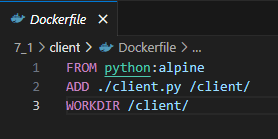


Рисунок 1.3 – Dockerfile для создания клиентской части

На рисунке 1.4 показан листинг файла server.py. На рисунке 1.5 показан Dockerfile для создания образа серверной части.

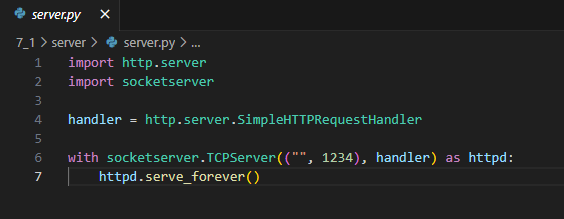


Рисунок 1.4 – Листинг файла server.py

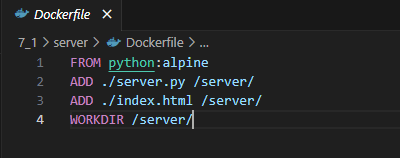


Рисунок 1.5 – Dockerfile для создания образа серверной части

На рисунке 1.6 показан листинг файла docker-compose.yml.

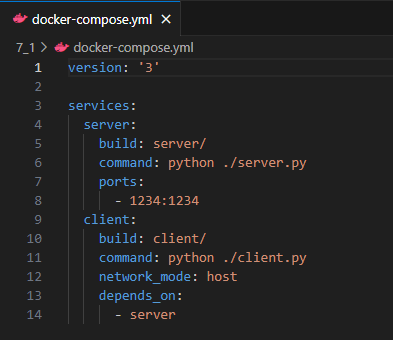


Рисунок 1.6 – Листинг файла docker-compose.yml

На рисунке 1.7 показан работающий веб-сервер.

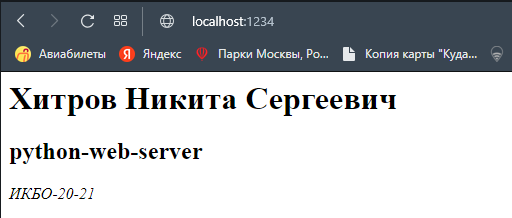


Рисунок 1.7 – Работающий веб-сервер

На рисунке 1.8 показаны созданные контейнеры.

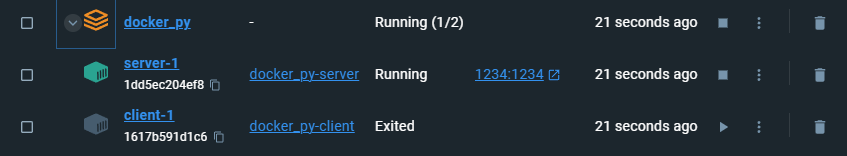


Рисунок 1.8 – Созданные контейнеры

1. Создать аналогичный сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо языка Python использовать язык JavaScript. Проект назвать docker\_js

На рисунке 2.1 показан веб-сервер на языке JavaScript.



Рисунок 2.1 – Веб-сервер на языке JavaScript

На рисунке 2.2 показан Dockerfile для создания образа приложения.

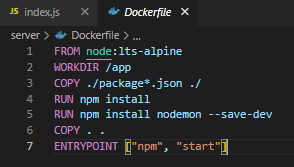


Рисунок 2.2 – Dockerfile для создания образа

На рисунке 2.3 показан файл docker-compose.yml.

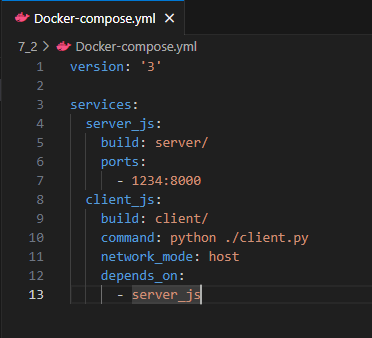


Рисунок 2.3 – Листинг файла docker-compose

На рисунке 2.4 показаны созданные контейнеры.



Рисунок 2.4 – Созданные контейнеры

На рисунке 2.5 показана работа клиентской части.

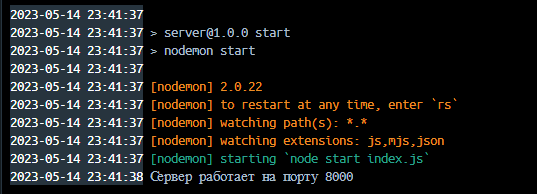


Рисунок 2.5 – Работа клиентской части

На рисунке 2.6 показан работающий веб-сервер.

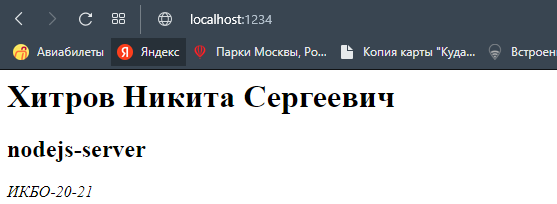


Рисунок 2.6 – Работающий веб-сервер

1. Создать сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо сервера на Python использовать веб сервер Apache. Проект назвать docker\_apache

На рисунке 3.1 показан Dockerfile для создания образа приложения с веб-сервером apache.

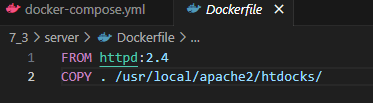


Рисунок 3.1 – Dockerfile для создания образа

На рисунке 3.2 показан листинг файла docker-compose.yml.

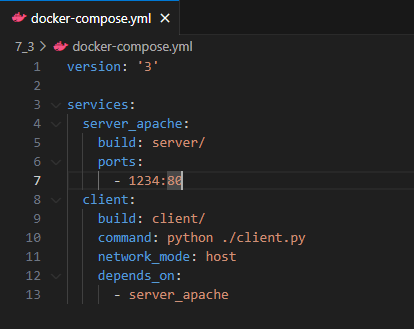


Рисунок 3.2 – Файл docker-compose

На рисунке 3.3 показаны созданные контейнеры.

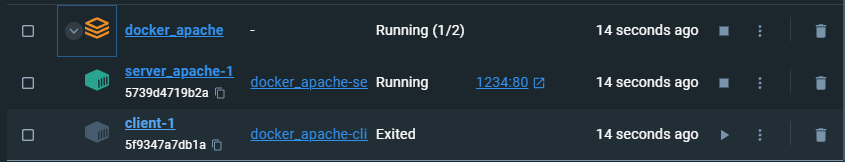


Рисунок 3.3 – Созданные контейнеры

На рисунке 3.4 показан работающий контейнер с клиентской частью.

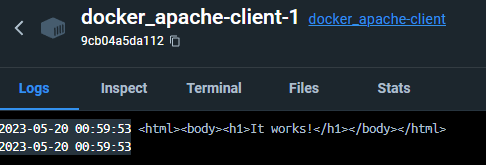


Рисунок 3.4 – Работающий контейнер с клиентской частью

На рисунке 3.5 показан работающий веб-сервер.

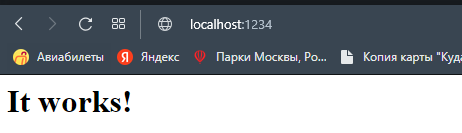


Рисунок 3.5 – Работающий веб-сервер

1. Создать сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо сервера на Python использовать веб сервер Nginx

На рисунке 4.1 показан файл Dockerfile для создания образа приложения с веб-сервером Nginx.

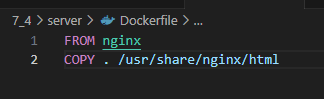


Рисунок 4.1 – Dockerfile

На рисунке 4.2 показан листинг файла docker-compose.

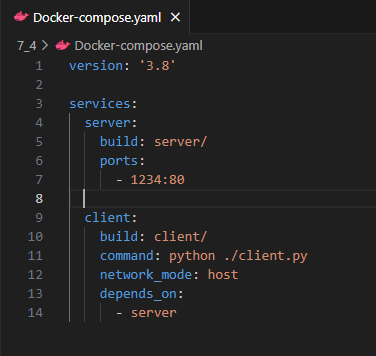


Рисунок 4.2 – Листинг файла docker-compose

На рисунке 4.3 показаны созданные контейнеры.



Рисунок 4.3 – Созданные контейнеры

На рисунке 4.4 показана работа контейнера с клиентской частью.

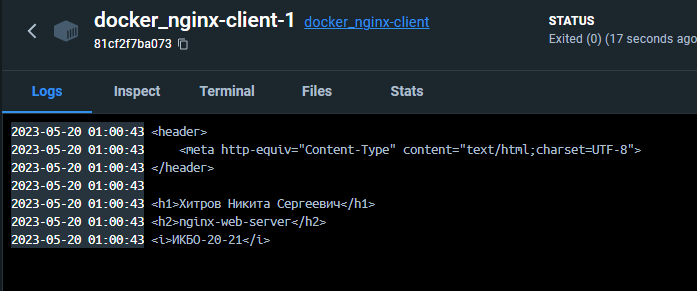


Рисунок 4.4 – Работа контейнера с клиентской частью

На рисунке 4.5 показан работающий веб-сервер.

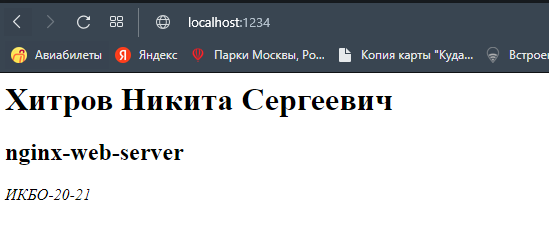


Рисунок 4.5 – Работающий веб-сервер

1. Создать сервис, в состав которого входят следующие компоненты apache, mysql, phpMyAdmin. Проект назвать docker\_phpmyadmin.

На рисунке 5.1 показан листинг файла docker-compose.

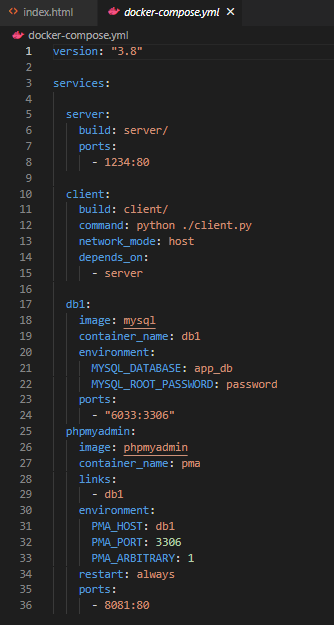


Рисунок 5.1 – Листинг файла docker-compose

На рисунке 5.2 показаны созданные контейнеры.

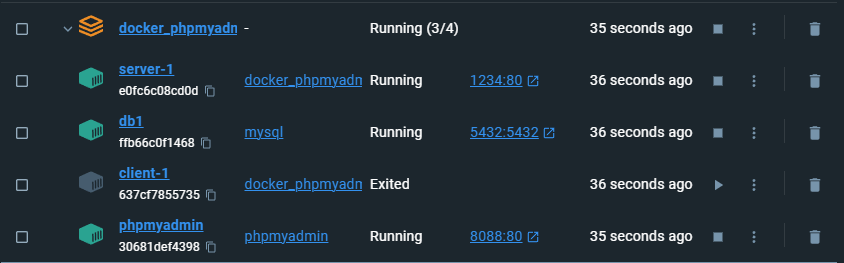


Рисунок 5.2 – Созданные контейнеры

На рисунке 5.3 показан работающий сервис.

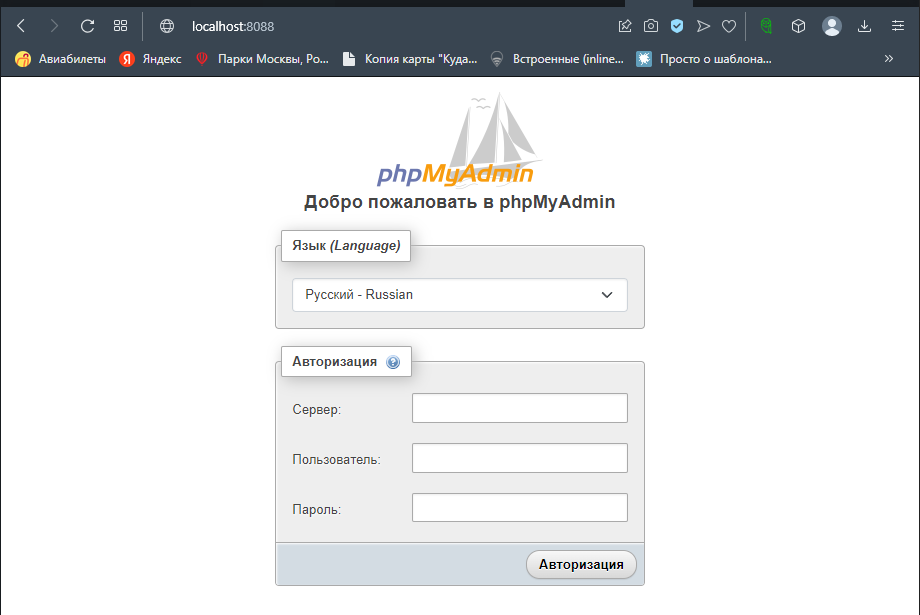


Рисунок 5.3 – Работающий сервис

# Вывод

В ходе выполнения практической работы были получены навыки сборки Docker-контейнеров с помощью docker-compose.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. DOCKER + DOCKER COMPOSE – ЗАПУСК КОНТЕЙНЕРОВ ДЛЯ ВЕБРАЗРАБОТКИ. Раздел: Docker - конфигурирование отдельного проекта – URL: https://atlogex.com/docker-start/ (дата обращения: 17.04.2023)
2. Официальная документация Dockerfile reference – URL: https://docs.docker.com/engine/reference/builder/ (дата обращения: 17.04.2023)
3. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/131719 (дата обращения: 17.04.2023). — Режим доступа: для авториз. пользователей.