

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 №9**

**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 создания сетевого соединения и передачи данных между docker-контейнерами посредством localhost.

**Теоретическое введение**

Docker — это платформа, которая позволяет упаковывать и запускать приложения в контейнерах. Контейнеры позволяют запускать приложения в изолированной среде, что повышает безопасность и обеспечивает консистентность окружения.

Одним из способов обеспечения взаимодействия между контейнерами является использование механизма сетевых интерфейсов Docker. Контейнеры могут быть подключены к одному или нескольким сетевым интерфейсам, которые могут быть настроены для работы в различных режимах, например, мостовом или режиме host.

В режиме моста каждый контейнер имеет свой IP-адрес и сетевой интерфейс, но они могут свободно обмениваться данными через локальную сеть Docker. В режиме host, с другой стороны, контейнеры используют сетевой стек хост-машины, что позволяет им использовать все сетевые интерфейсы и настройки, которые есть на хост-машине.

Еще один способ обеспечения взаимодействия между контейнерами в Docker — это использование локального хоста (localhost). Контейнеры могут взаимодействовать друг с другом через локальный хост, используя IP-адрес 127.0.0.1 или доменное имя "localhost". Это может быть полезно, когда нужно взаимодействовать с контейнером, запущенным на той же машине, например, когда необходимо обеспечить доступ к базе данных, запущенной в отдельном контейнере.

В целом, межконтейнерное взаимодействие через localhost — это простой и удобный способ обеспечить связь между контейнерами в Docker, который может быть использован в различных сценариях.

**Выполнение работы**

*Задание:* создать два докер-контейнера, которые будут одновременно "слушать" localhost на разных портах. Пользователь делает запрос к контейнеру №1. Далее контейнер №1 делает запрос к контейнеру №2. Контейнер №2 обработает запрос, передаст результат контейнеру №1. Контейнер №1 передаст результат пользователю. API может быть реализовано на любой технологии. В итоге необходимо настроить систему из нескольких контейнеров, связанных друг с другом. Рисунок 1.

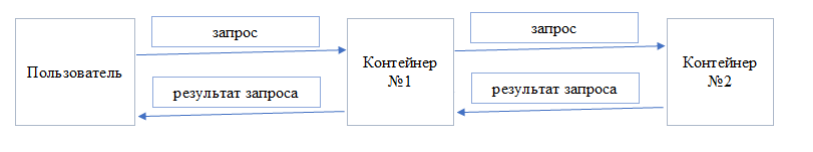


Рисунок 1 - Схема взаимодействия пользователя и контейнеров.

Для реализации данного задания необходимо создать 2 Docker контейнера и связать их при помощи link.

Для начала необходимо создать Dockerfile для каждого контейнера. Рисунки 2 – 3.

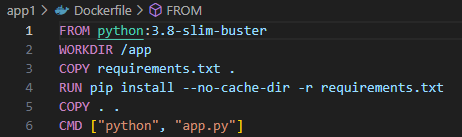


Рисунок 2 – Dockerfile для 1 контейнера

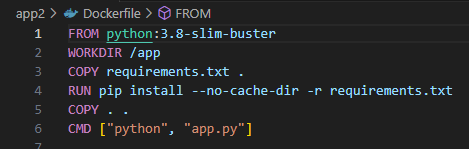


Рисунок 3 – Dockerfile для 2 контейнера

Теперь напишем 2 простых приложения на python, которые будут осуществлять взаимодействие друг с другом. Первое приложение будет отправлять POST запрос на адрес 2 docker – контейнера и выводить полученный от него ответ. Рисунок 4.

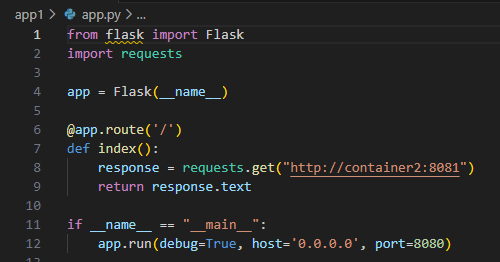


Рисунок 4 – Python код приложения для 1 контейнера

Приложение из 2 контейнера будет обрабатывать запрос и формировать ответ. Рисунок 5.

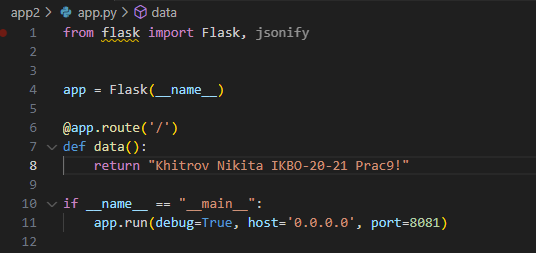


Рисунок 5 – Python код приложения для 2 контейнера

Также напишем requirements.txt файл для загрузки зависимостей, которые необходимы для работы приложения. Рисунки 6 - 7.

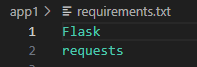


Рисунок 6 – Содержимое requirements.txt для 1 приложения



Рисунок 7 – Содержимое requirements.txt для 2 приложения

Теперь необходимо написать docker-compose файл. Готовый docker-compose.yml для данного задания показан на Рисунке 7.

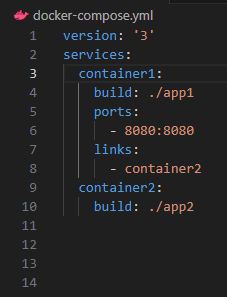


Рисунок 7 – Содержимое docker-compose

Протестируем корректность работы контейнеров, для этого отправим GET запрос на localhost порт 8080(1 контейнер) от лица пользователя, после чего обработчик обработает запрос и отправит GET запрос на порт 8081 (2 контейнер), ответом на GET запрос будет является строка «Hello, Pablo!», что мы и видим при переходе на <http://localhost:8080/>. Рисунок 8.

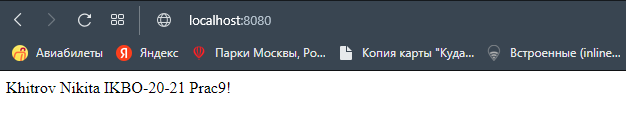


Рисунок 8 – Тестирование связи контейнеров

**Вывод**

В ходе выполнения практической работы были получены навыки создания сетевого соединения и передачи данных между docker-контейнерами посредством localhost.