**ОГЛАВЛЕНИЕ**

[1 Постановка задачи (Вариант 21) 1](#_Toc1640)

[1 Задание 1](#_Toc20069)

[2 Проектирование и реализация 2](#_Toc1696)

[2.1 Образы 2](#_Toc14609)

[2.2 Изоляция 3](#_Toc11590)

[2.3 Работа с портами 4](#_Toc5950)

[2.4 Именнованные контейнеры, остановка и удаление 5](#_Toc2561)

[2.5 Постоянное хранение данных 6](#_Toc21089)

[2.5.1 Тома 7](#_Toc16183)

[2.5.2 Монтирование директорий и файлов 8](#_Toc28424)

[2.6 Переменные окружения 9](#_Toc25610)

[2.7 Dockerfile 10](#_Toc336)

[2.8 Индивидуальное задание 11](#_Toc22412)

[Выводы 12](#_Toc8674)

# Постановка задачи (Вариант 21)

## 1 Задание

В практической работе необходимо выполнить все шаги из разделов 1–7. В отчёт должны быть включены ответы на вопросы, выделенные курсивом, результаты выполнения команд из разделов 1–7, а также выполненное индивидуальное задание (раздел 8): листинг Dockerfile, а также команды сборки и запуска контейнера.

# Проектирование и реализация

## Образы

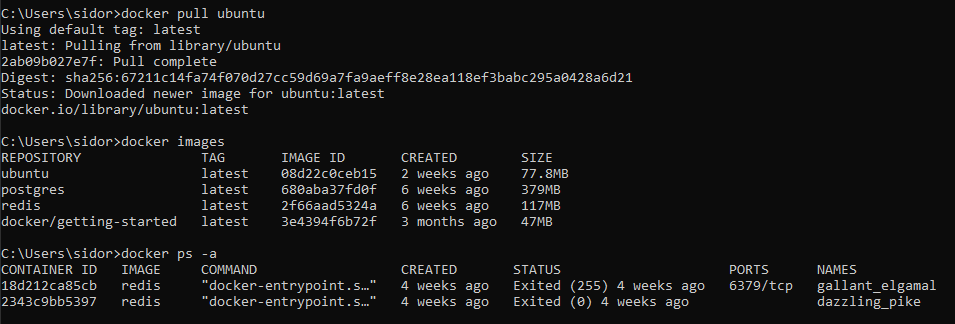


Рисунок 1 - команды для работы с образами

## Изоляция

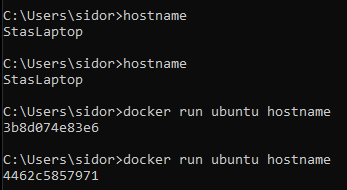


Рисунок 2 - Результат выполнения команды hostname

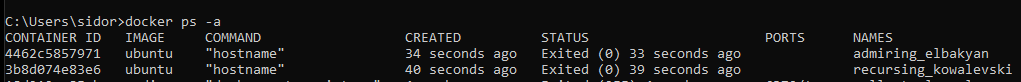


Рисунок 3 - Результат выполнения команды ps -a

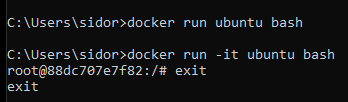


Рисунок 4 - Результат выполнения команды docker run с -it и без

## Работа с портами

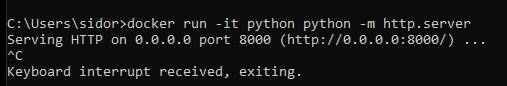


Рисунок 5 - Результат запуска веб сервера

3_2Ports

Рисунок 6 - Результат запуска веб сервера с указанием порта

## Именнованные контейнеры, остановка и удаление

4_1NamedContainers

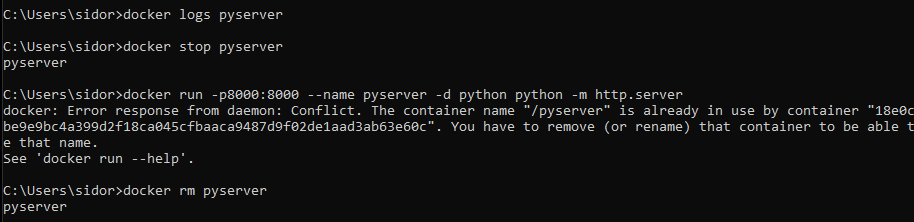


Рисунок 7 - Результат запуска контейнера в фоне

## Постоянное хранение данных

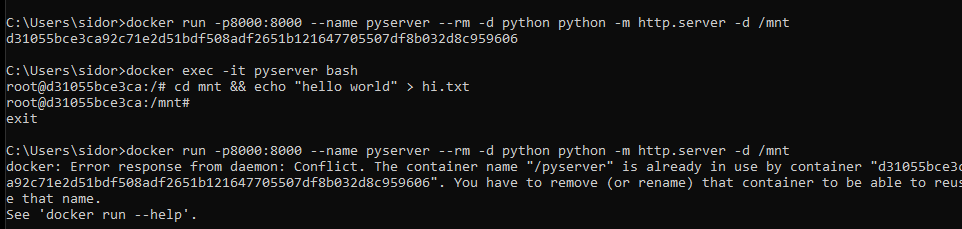


Рисунок 8 - Результат ошибочного запуска контейнера

5_0_2

Рисунок 9 - Данные после перезапуска контейнера

*Ответ на вопрос:*

В данной команде запуска контейнера с помощью Docker используются следующие флаги:

* -p8000:8000: пробрасывает порт 8000 из контейнера в порт 8000 на хосте, таким образом веб-сервер, запущенный в контейнере, будет доступен по адресу [http://localhost:8000](http://localhost:8000/)
* --name pyserver: задает имя контейнеру, в данном случае - pyserver;
* --rm: автоматически удаляет контейнер после его остановки;
* -d: запускает контейнер в фоновом режиме;
* python python -m http.server -d /mnt: команда, которая выполнится внутри контейнера и запускает веб-сервер на порту 8000 и с корневой директорией "/mnt".

2.5.1 Тома

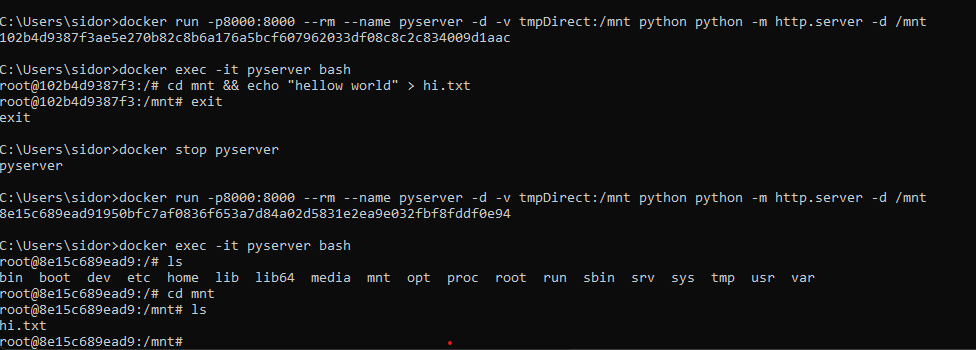


Рисунок 10 - Данные сохранненые монтировкой

5_1_2

Рисунок 11 - Местоположение данных

2.5.2 Монтирование директорий и файлов

5_2_1

Рисунок 12 - Запуск с монтированием файла

2.6 Переменные окружения

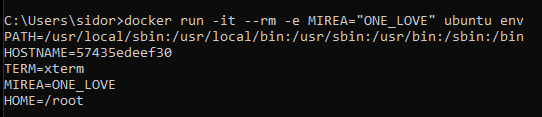


Рисунок 13 - Использование переменных окружения

2.7 Dockerfile

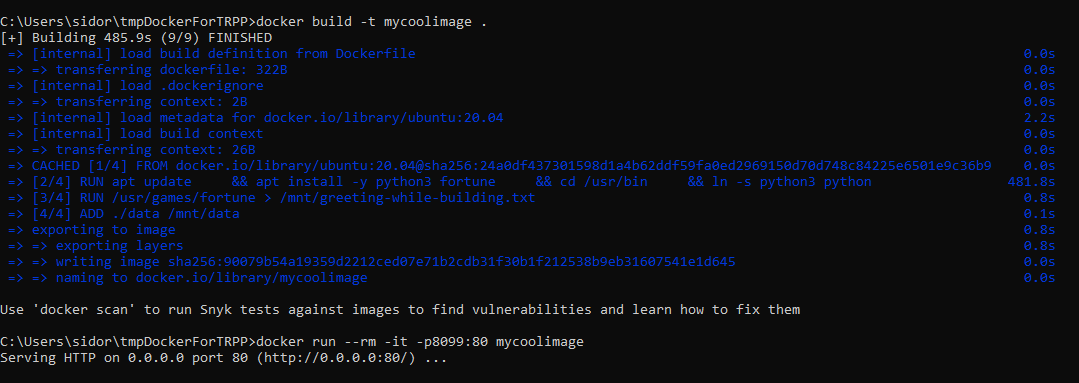


Рисунок 14 - Сборка образа описанного в задании

2.8 Индивидуальное задание

Вариант: 21

Пакет: patch

Ubuntu 20.04

Примонтировать директорию data в директорию /mnt/files/ в контейнере

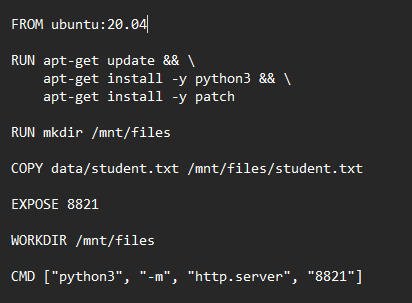


Рисунок 15 - Содержимое Dockerfile

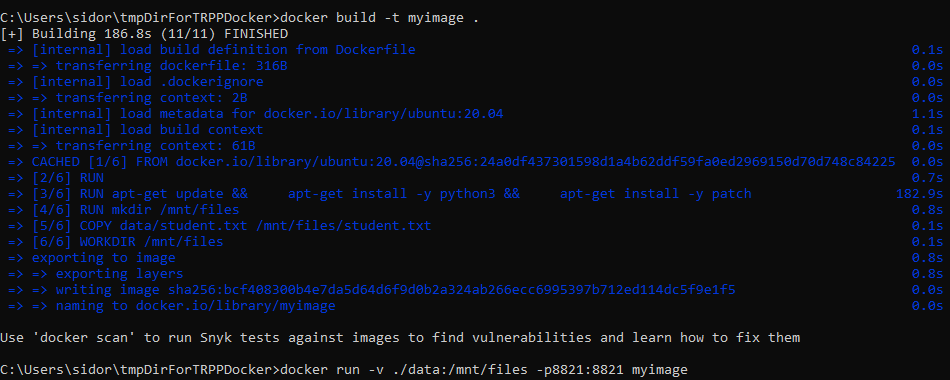


Рисунок 16 - Сборка образа созданного по заданию

**Выводы**

В ходе выполнения данной практической работы были изучены основные команды Docker, а также были получены практические навыки по работе с Docker.