**Оглавление**

[1.Образы…. 3](#_Toc130942449)

[2. Изоляция 3](#_Toc130942450)

[3. Работа с портами 5](#_Toc130942451)

[4. Именованные контейнеры остановка и удаление 6](#_Toc130942452)

[5. Постоянное хранение данных 7](#_Toc130942453)

[6. Переменные окружения 11](#_Toc130942454)

[7. Dockerfile 11](#_Toc130942455)

[8. Индивидуальное задание 13](#_Toc130942456)

[9. Вывод… 15](#_Toc130942457)

1. Образы

На рисунке 1 представлен результат загрузки образа Ubuntu версии latest на хостовую машину.

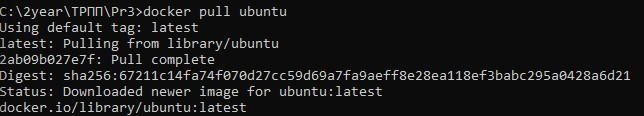


Рисунок 1 - загрузка образа Ubuntu:latest на хостовую машину

На рисунке 2 представлен вывод списка всех имеющихся образов.

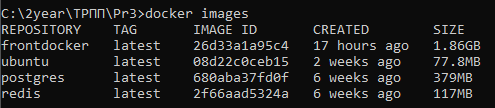


Рисунок 2 - вывод списка всех имеющихся образов

На рисунке 3 представлен вывод списка всех имеющихся контейнеров.

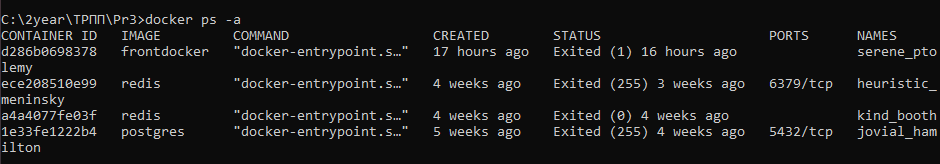


Рисунок 3 - вывод списка всех имеющихся контейнеров

1. Изоляция

На рисунке 4 представлено выполнение команды hostname дважды.

В обоих случаях команда hostname вывела одинаковый результат, так как выполнение происходит на одной и той же локальной машине.

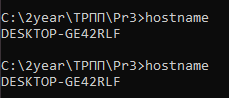


Рисунок 4 - выполнение команды hostname дважды

На рисунке 5 представлено выполнение команды для вывода имени хоста на образах внутри docker контейнера. В данном случае результат был отличен, так как каждый раз создавался новый контейнер с новым образом Ubuntu.

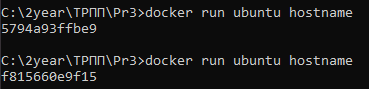


Рисунок 5 - выполнение команды hostname на образах внутри docker контейнеров

На рисунке 6 представлен вывод всех имеющихся контейнеров



Рисунок 6 - вывод всех имеющихся контейнеров

На рисунке 7 представлен запуск bash внутри одного из контейнеров.



Рисунок 7 - запуск bash внутри одного из контейнеров

На рисунке 8 представлен запуск bash внутри одного из контейнеров в интерактивном терминале при помощи флагов « -i », « -t » b и выход из него.



Рисунок 8 - запуск bash внутри одного из контейнеров в интерактивном терминале

1. Работа с портами

На рисунке 9 представлен результат загрузки образа python.

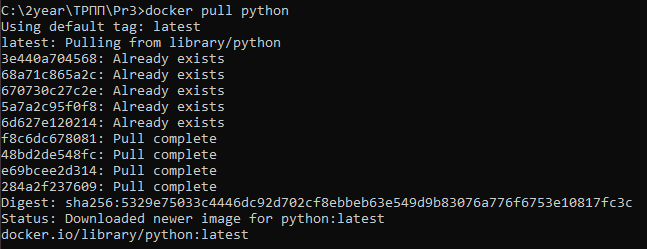


Рисунок 9 - загрузка образа python

На рисунке 10 представлен запуск веб-сервера из корня контейнера без проброса портов.

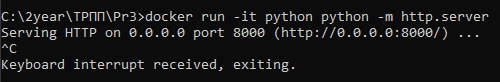


Рисунок 10 - запуск веб-сервера из корня контейнера без проброса портов

На рисунке 11 представлен запуск веб-сервера из корня контейнера с проброшенными портами.

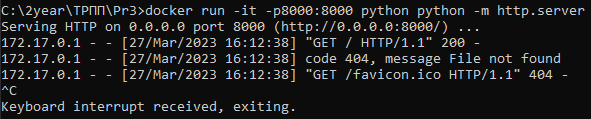


Рисунок 11 - запуск веб-сервера из корня контейнера с проброшенными портами

На рисунке 12 представлена работа веб-сервера по адресу «http://localhost:8000/».

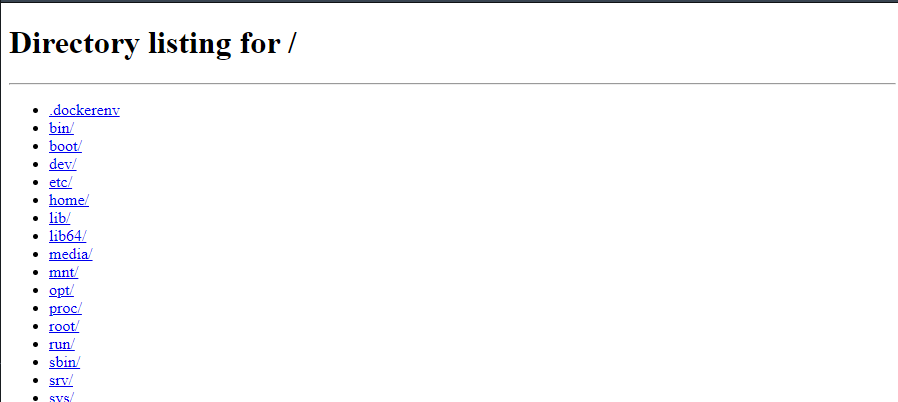


Рисунок 12 - работа веб-сервера

1. Именованные контейнеры остановка и удаление

На рисунке 13 представлен запуск контейнера в фоне (флаг « -d »), с назначением имени (флаг « --name »).



Рисунок 13 - запуск контейнера в фоне с назначением имени

На рисунке 14 представлен вывод списка запущенных контейнеров

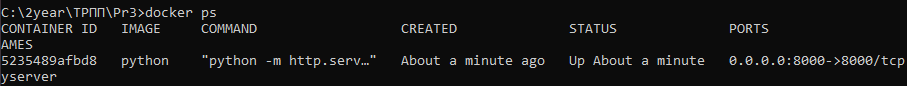


Рисунок 14 - вывод списка запущенных контейнеров

На рисунке 15 представлен вывод логов запущенного контейнера.



Рисунок 15 - вывод логов запущенного контейнера

На рисунке 16 представлена остановка запущенного контейнера.



Рисунок 16 - остановка запущенного контейнера

На рисунке 17 представлено удаление данного контейнера.



Рисунок 17 - удаление данного контейнера

На рисунке представлен запуск контейнера с опцией самоуничтожения при остановке (флаг « --rm »).



Рисунок 18 - запуск контейнера с опцией самоуничтожения при остановке

1. Постоянное хранение данных

На рисунке 18 представлен запуск контейнера с указанием корневой директории для отображения на веб-сервере (ключ « -d » c параметром «/mnt»).



Рисунок 19 - запуск контейнера с указанием корневой директории для отображения

Значения остальных флагов: « --rm » - удаление контейнера при остановке, « -p8000:8000 » - прокидывание порта 8000 на образ в контейнер, « -d » запуск контейнера в фоне, «python –m http.server» - исполняемая команда.

На рисунке 20 представлен запуск bash в созданном контейнере в интерактивном терминале. В bash происходит создание файла hi.txt в директории «/mnt» с записью в него «hello world».

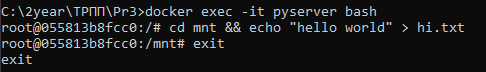


Рисунок 20 - запуск bash в запущенном контейнере в интерактивном терминале

На рисунке 21 представлена работа веб-сервера. (файл hi.txt в корневой директории отображения)

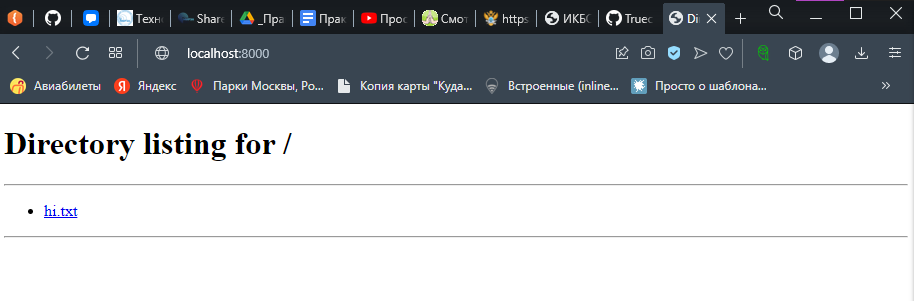


Рисунок 21 - работа веб-сервера

На рисунке 22 представлен запуск контейнера с примонтированием каталога «myfiles» на хостовой машине к каталогу «/mnt» в контейнере.



Рисунок 22 - запуск контейнера с примонтированием тома

На рисунке 23 представлен запуск bash в созданном контейнере в интерактивном терминале. В bash происходит создание файла hi.txt в директории «/mnt» с записью в него «hello world».

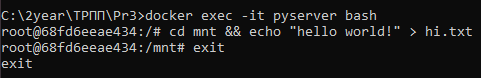


Рисунок 23 - запуск bash в созданном контейнере в интерактивном терминале

На рисунке 24 представлен вывод информации о привязанном томе.

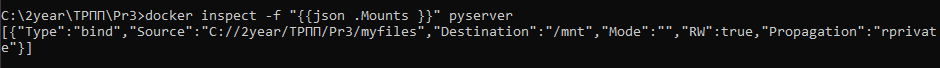


Рисунок 24 - информация о привязанном томе

На рисунке 25 представлены сохранившиеся в привязанном томе данные сохранившиеся после удаления контейнера.

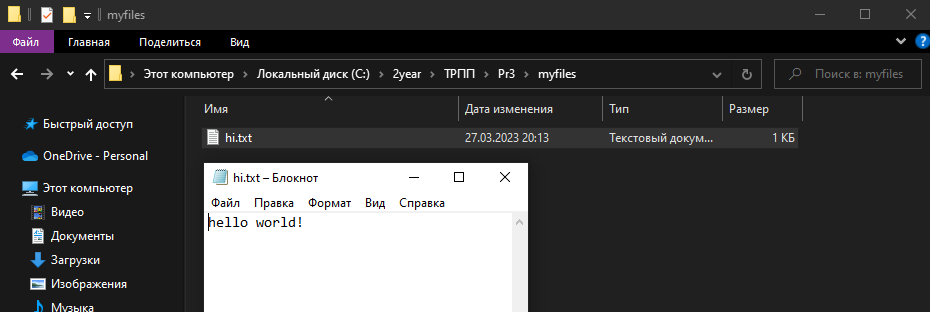


Рисунок 25 - данные в привязанном томе

На рисунке 26 представлен список всех имеющихся томов.

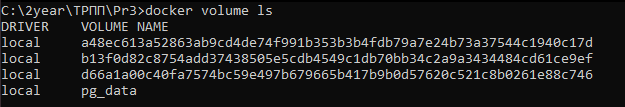


Рисунок 26 - список всех имеющихся томов

На рисунке 27 представлено создание файла host.txt в директории myfiles.

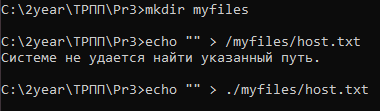


Рисунок 27 - создание файла host.txt в директории myfiles

На рисунке 28 представлен запуск контейнера с примонтированием тома myfiles.

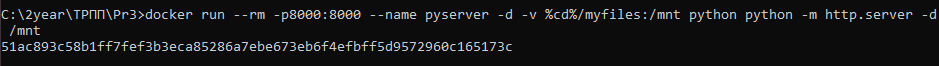


Рисунок 28 - запуск контейнера с примонтированием тома myfiles

На рисунке 29 На рисунке 23 представлен запуск bash в созданном контейнере в интерактивном терминале. В bash происходит вывод уже имеющихся файлов и создание файла hi.txt в директории «/mnt» с записью в него «hello world».

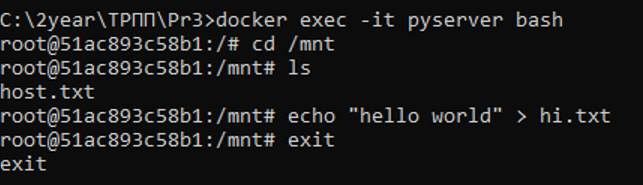


Рисунок 29 - запуск bash в созданном контейнере

На рисунке 30 представлен вывод всех файлов контейнера в томе на хостовой машине.

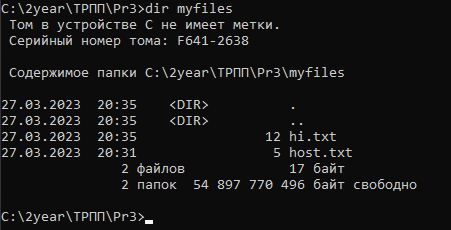


Рисунок 30 - список файлов на локальном томе

1. Переменные окружения

На рисунке 31 представлено добавление переменной MIREA=ONE\_LOVE в окружение среды и вывод всех переменных среды.

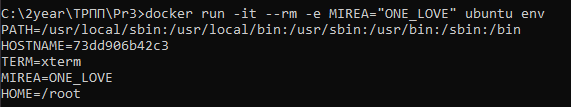


Рисунок 31 - добавление переменной MIREA=ONE\_LOVE в окружение контейнера и вывод всех переменных среды

1. Dockerfile

На рисунке 32 представлено создание Dockerfile для контейнера.

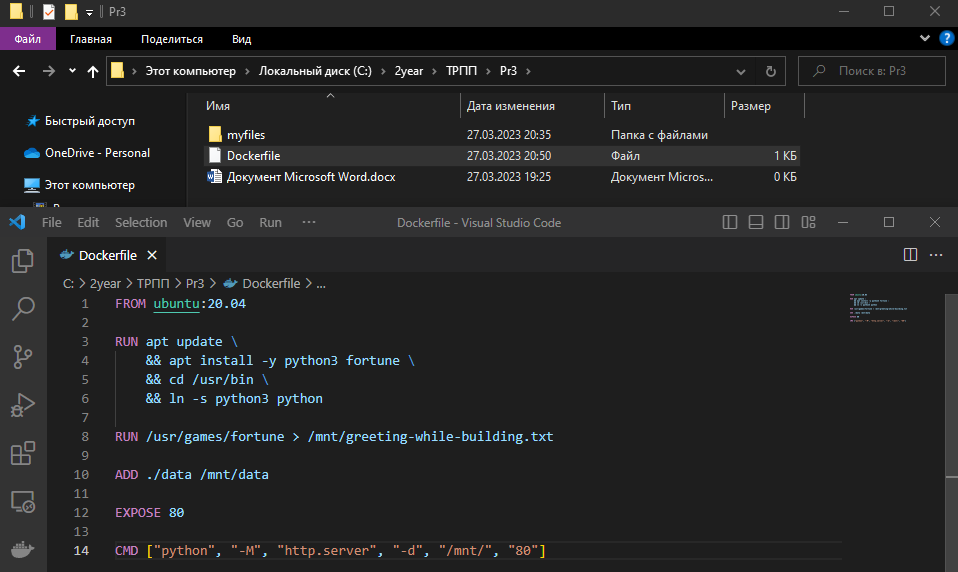


Рисунок 32 - создание Dockerfile для контейнера

На рисунке 33 представлено построение контейнера согласно Dockerfile.

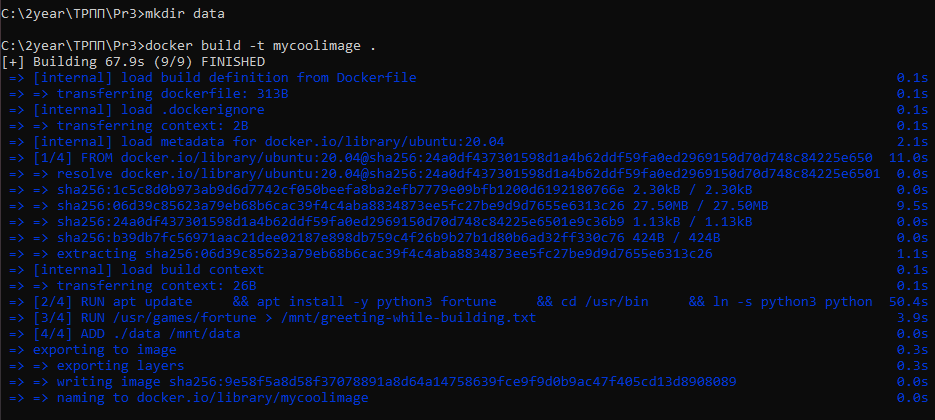


Рисунок 33 - построение контейнера согласно Dockerfile

На рисунке 34 представлен запуск созданного контейнера.



Рисунок 34 - запуск созданного контейнера

На рисунках 35-36 представлена работа веб-сервера.

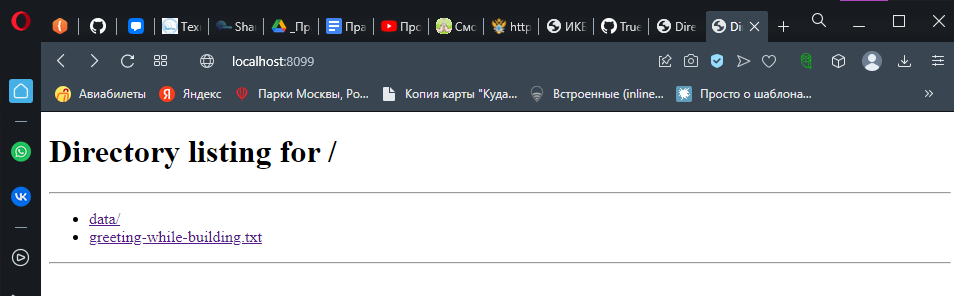


Рисунок 35 - работа веб-свервера

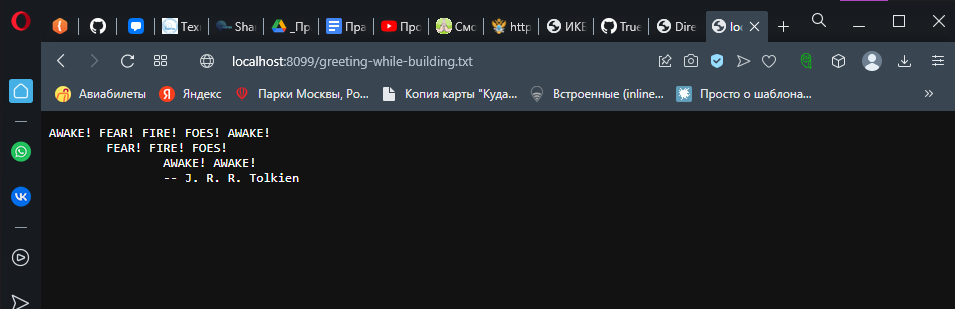


Рисунок 36 - работа веб-сервера

1. Индивидуальное задание

На рисунке 37 представлено создание файла student.txt с записью в него строки «Хитров Н.С. ИКБО-20-21 №25(10)».



Рисунок 37 - создание файла student.txt

На рисунке 38 представлено создание Dockerfile для контейнера.

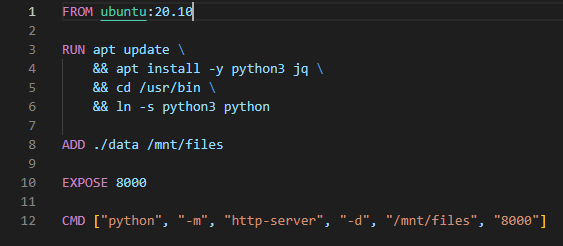


Рисунок 38 - создание Dockerfile для работы контейнера

На рисунке 39 представлено построение контейнера согласно Dockerfile.

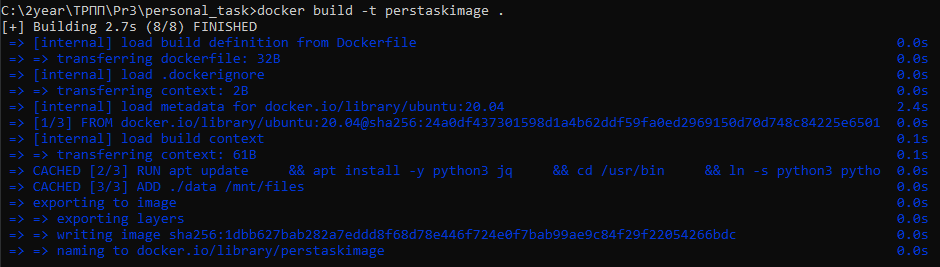


Рисунок 39 - построение контейнера согласно Dockerfile\

На рисунке 40 представлен запуск созданного контейнера.

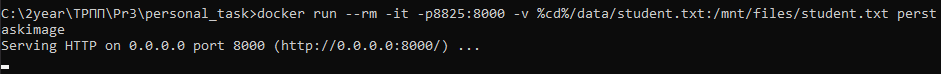


Рисунок 40 - запуск созданного контейнера

На рисунке 41 представлена работа веб-сервера.

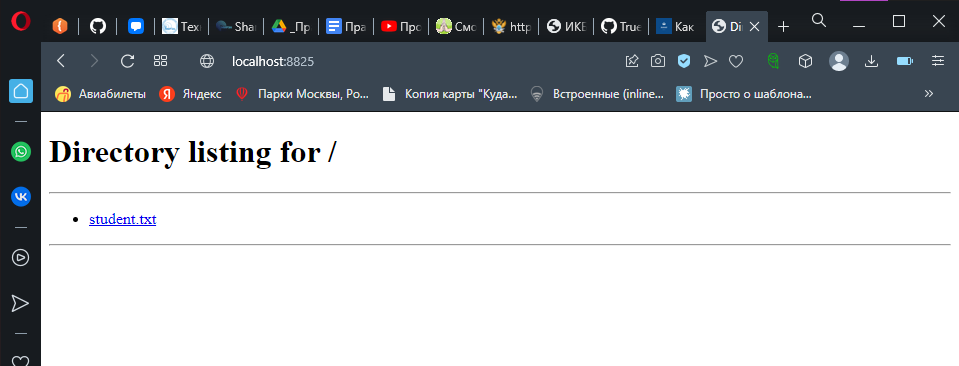


Рисунок 41 - работа веб-сервера

1. Вывод

В ходе выполнения работы были получены навыки по сборке приложения с подгрузкой всех необходимых зависимостей в контейнере Docker, настройки контейнеров, работе с Dockerfile, запуске контейнера.