|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

Институт Информационных технологий (ИТ)

|  |
| --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4** |
| **по дисциплине** |
| **«Технология разработки программных приложений»**  **Тема: «Основы работы с Docker»** |

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

|  |  |
| --- | --- |
| Выполнил студент группы ИКБО-08-21 | Гусев Т.А. |
| Приняла | Туманова М.Б. |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2023г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023 г. | *(подпись руководителя)* |

Москва 2023

# Образы

На рисунке 1 представлен просмотр имеющихся образов.

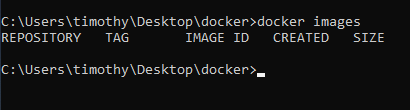


Рисунок 1 – Имеющиеся образы

На рисунке 2 представлена загрузка образа последней версии ubuntu.

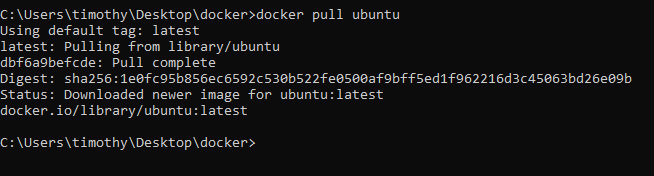


Рисунок 2 – Загрузка образа

На рисунке 3 прдеставлен просмотр образов после установки образа ubuntu.

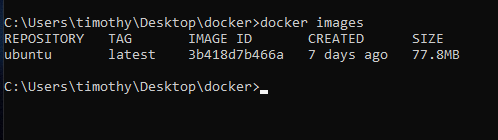


Рисунок 3 – Просмотр имеющихся образов

На рисунке 4 представлен просмотр имеющихся контейнеров.

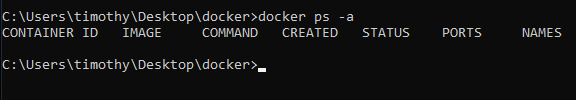


Рисунок 4 – Список контейнеров

# Изоляция

На рисунке 5 представлен просмотр информации о хостовой системе на локальной машине.

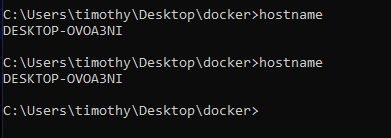


Рисунок 5 – Информация о хостовой системе

На рисунке 6 представлен просмотр информации о хостовой системе на контейнере.

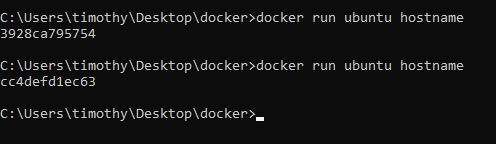


Рисунок 6 – Информация о хостовой системе в контейнерах

*Вопрос: Одинаковый ли результат получился при разных запусках? При запуске на локальной машине результат одинаковый, при выполнении в контейнере результат разный.*

На рисунке 7 представлен просмотр запущенных конейнеров.

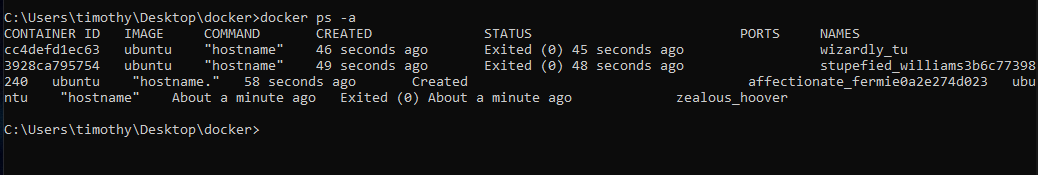


Рисунок 7 – Запущенные контейнеры

На рисунке 8 представлена команда для запуска bash в контейнере с флагами -i и -t, также неявно указана последняя версия ubuntu:latest.

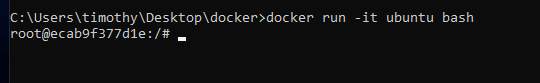


Рисунок 8 – Запуск bash в контейнере

# Работа с портами

На рисунке 9 изображена загрузка python-образа версии 3.9-slim.

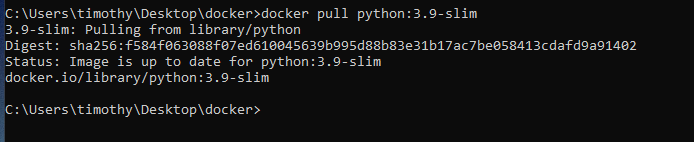


Рисунок 9 – Установка образа python

На рисунке 10 представлен запуск встроенного модуля веб-сервера python.



Рисунок 10 – Запуск сервера

На рисунке 11 представлен запуск сервера с пробросом порта 8000. Также представлен запуск сервера на порту 8888 с помощью флага -p 8888:8000.

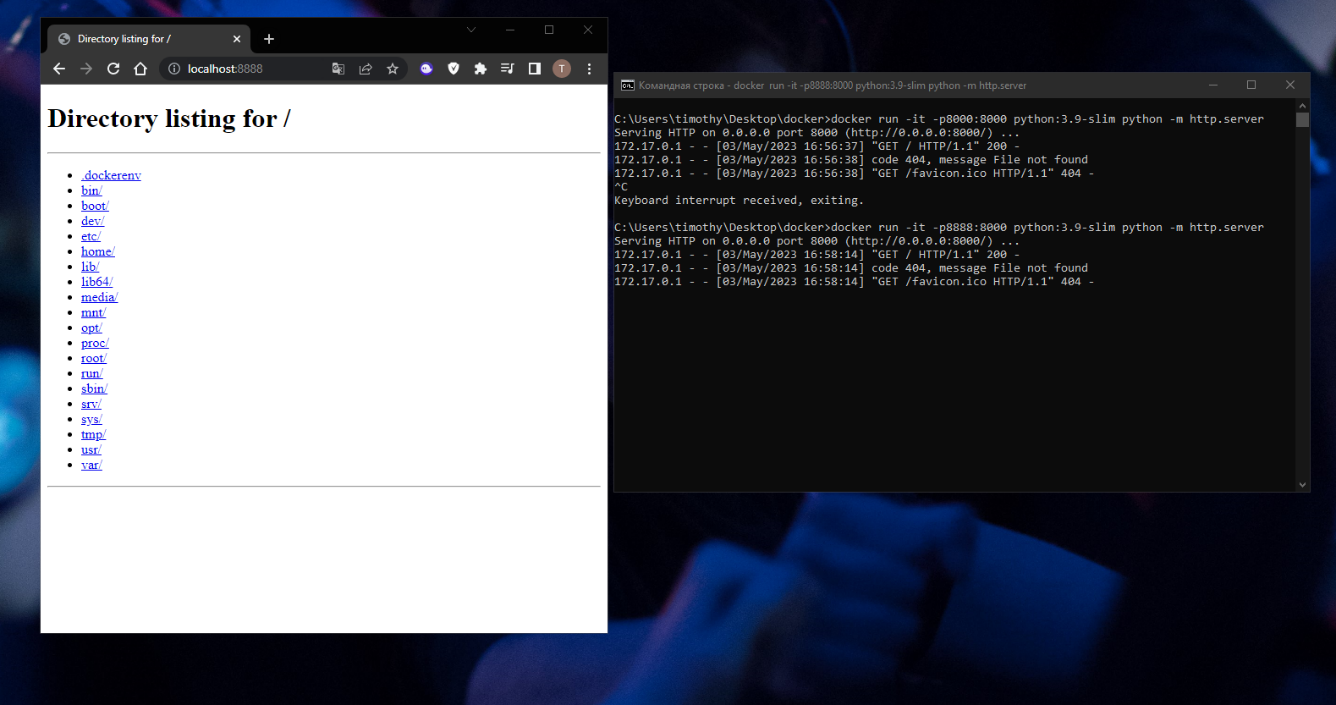


Рисунок 11 – Запуск сервера на определенном порту

# Именованные контейнеры, остановка и удаление

На рисунке 12 представлен запуск контейнера в фоновом режиме с помощью флага -d.

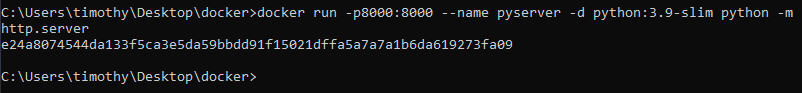


Рисунок 12 – Запуск контейнера в фоновом режиме

На рисунке 13 показан список контейнеров, а также наш ранее запущенный в фоновом режиме контейнер.

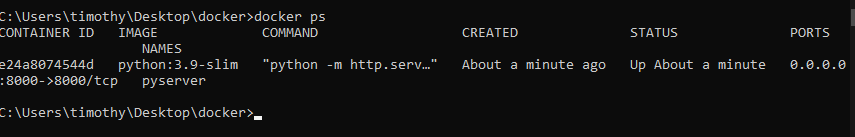


Рисунок 13 – Контейнеры

На рисунке 14 представлена остановка ранее запущенного в фоновом режиме контейнера.

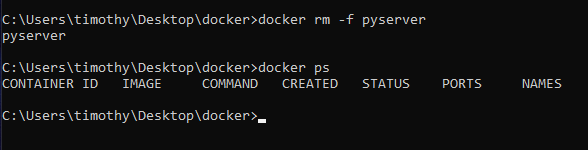


Рисунок 14 – Остановка контейнера

На рисунке 15 представлен запуск контейнера с его последующем удалением с помощью флага –rm.

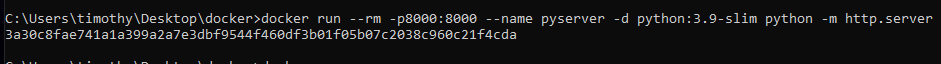


Рисунок 15 – Запуск контейнера с удалением

# Постоянное хранение данных

На рисунке 16 представлен запуск контейнера, в который веб-сервер будет отдавать содержимое директории /mnt.

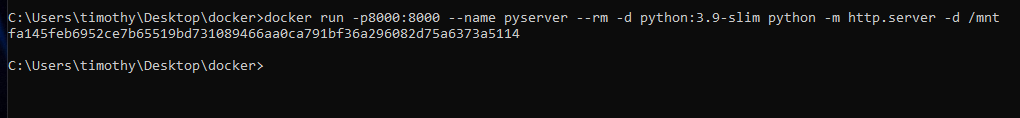


Рисунок 16 – Запуск контейнера

*Вопрос: Что значат остальные флаги запуска? Где здесь команда, которая выполнится в контейнере?*

Флаги запуска: -p – пробрасывание порта, --name – задание имени для

контейнера, --rm – автоматическое удаление после установки, -d – запуск в

фоновом режиме.

На рисунке 17 представлен вход в контейнер с последующем мозданием файла hi.txt в нем.

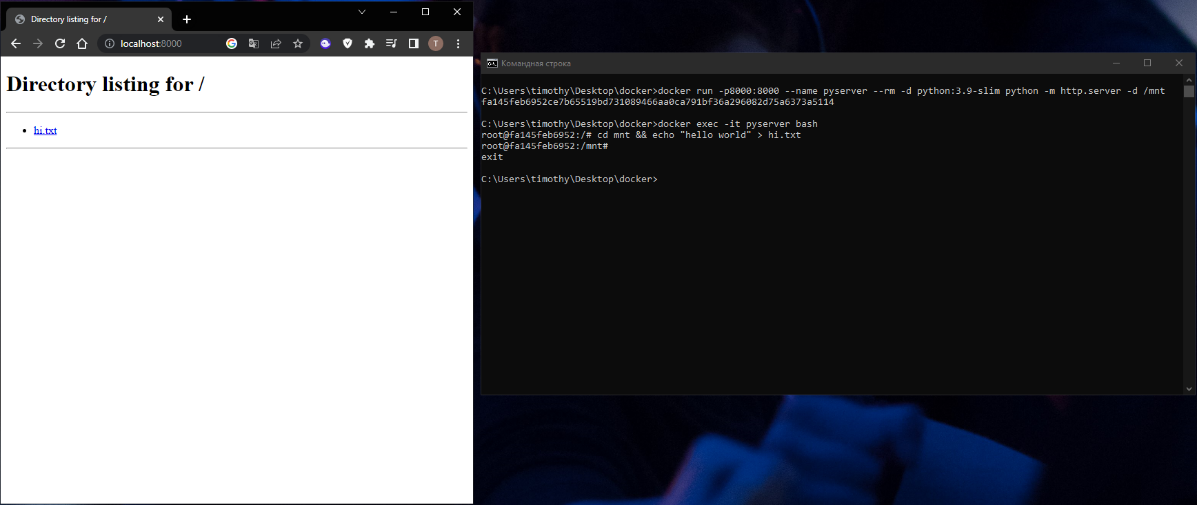


Рисунок 17 – Вход в контейнер

На рисунке 18 представлена остановка контейнера и запуск пустого контейнера с таким же именем.

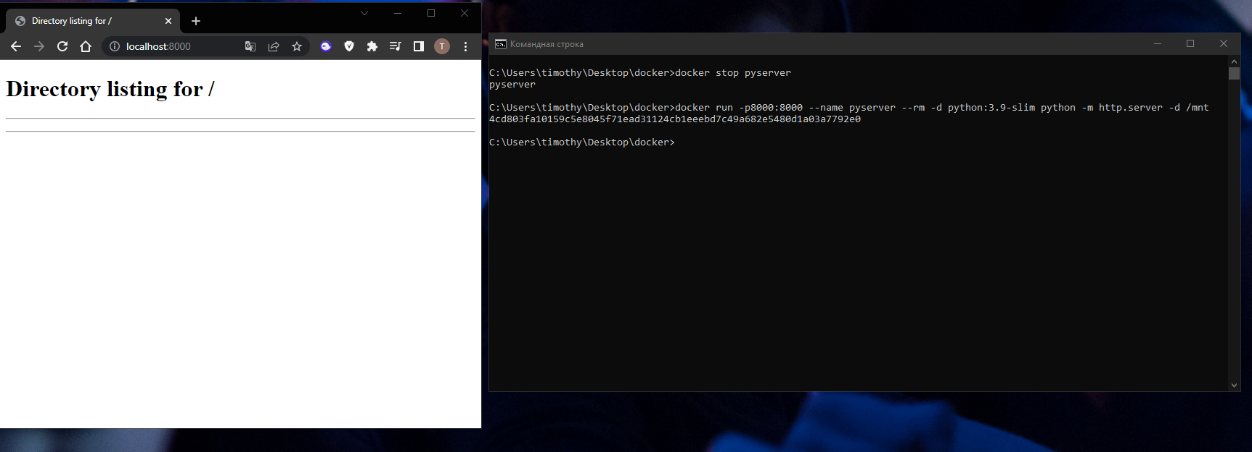


Рисунок 17 – Запуск нового контейнера

# Тома

На рисунке 18 представлено создание контейнера с уже примонтированным томом.

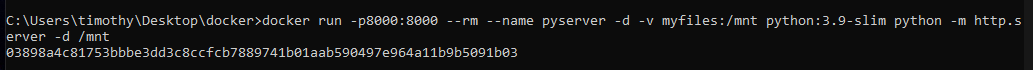


Рисунок 18 – Создание контейнера с примонтированным томом

На рисунке 19 представлено создание файла внутри контейнера запущенного на рисунке 18.

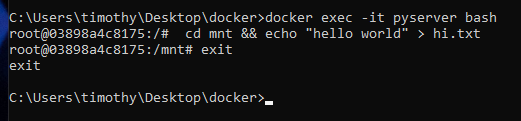


Рисунок 19 – Создание файла внутри контейнера

На рисунке 20 представлено использование команды inspect для того, чтобы узнать, где хранятся данные.

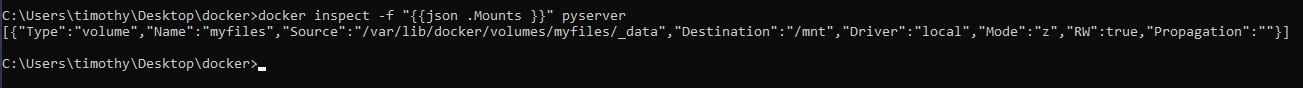


Рисунок 20 – Хранение данных

# Монтирование директорий и файлов

На рисунке 21 представлена остановка контейнера, созданного в предыдущем задании.

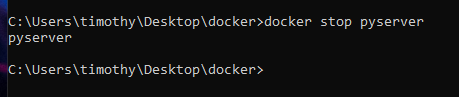


Рисунок 21 – Остановка контейнера

На рисунке 22 представлено создание директории и файла внутри нее.

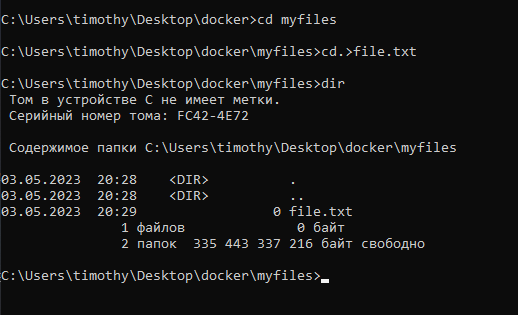


Рисунок 22 – Создание файлов

На рисунке 23 представлен запуск контейнера с примонтированной директорией myfiles.

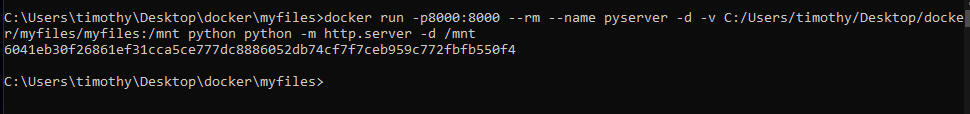


Рисунок 23 – Запуск контейнера

На рисунке 24 представлен вход в контейнер и вывод спеисок файлов внутри него.



Рисунок 24 – Вход в контейнер

На рисунке 25 представлено создание нового файла с hi.txt внутри запущенного ранее контейнера.

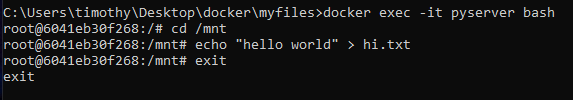


Рисунок 25 – Создание нового файла

Далее была выпалена остановка ранее созданного контейнера с помощью команды: docker stop pyserver.

# Переменные окружения

На рисунке 26 представлено добавление пременной окружения «MIREA» со значением «ONE LOVE».

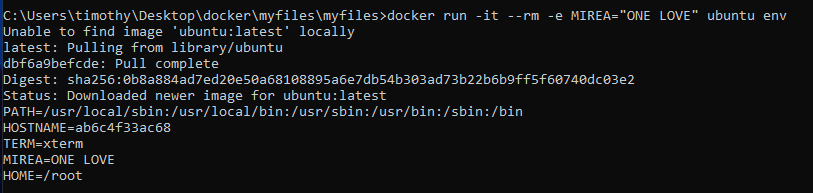


Рисунок 26 – Переменные окружения

# Dockerfile

На рисунке 27 представлено содержимое Dockerfile, из которого в последствии будет собран образ.

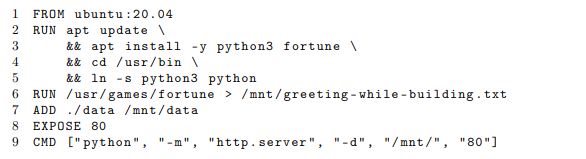


Рисунок 27 – Содержимое Dockerfile

На рисунке 28 представлена сборка образа из вышеупомянутого файла и запуск образа.

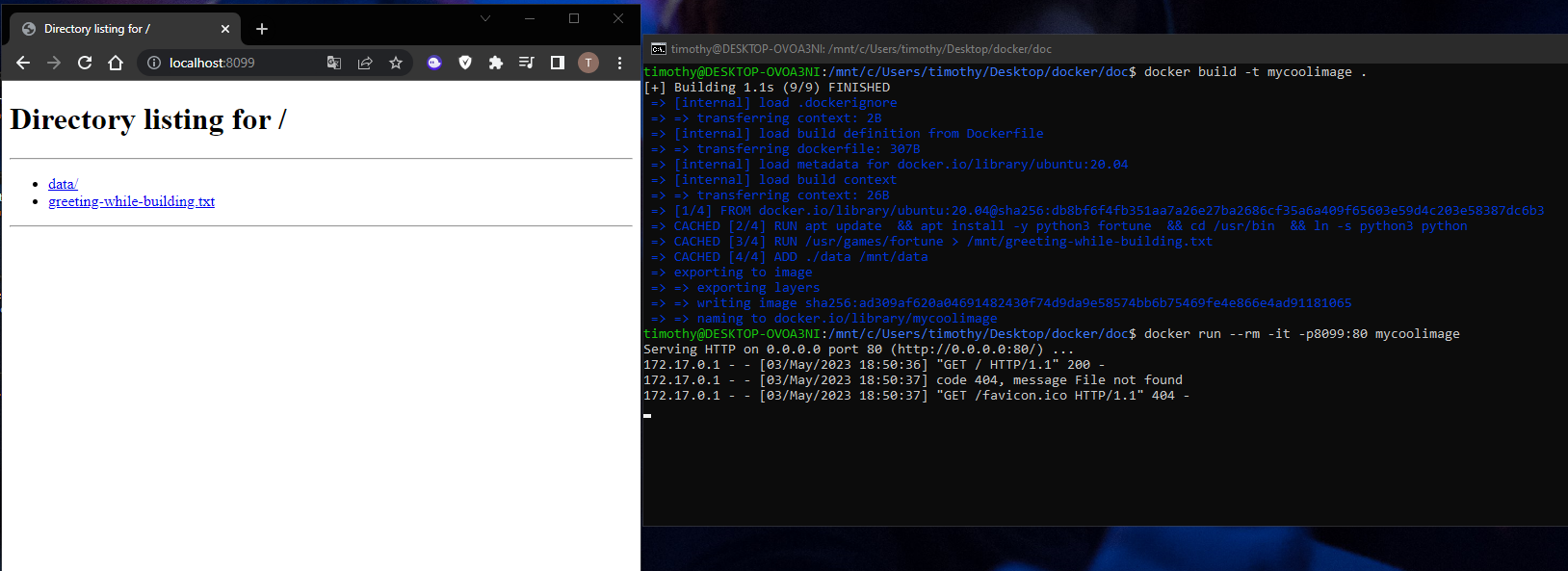


Рисунок 28 – Сборка и запуск образа

# Индивидуальное задание

На рисунке 29 представлено создание директории и файла с ФИО для монтирования.

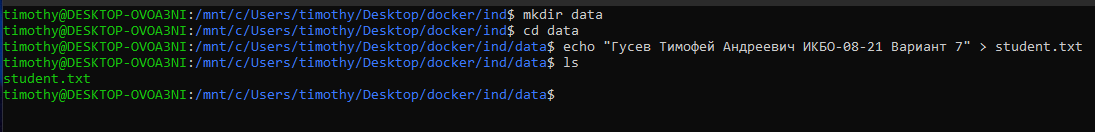


Рисунок 29 – Создание директории для монтирования

На рисунке 30 представлено содержимое Dockerfile в соответствии с индивидуальным вариантом №7.

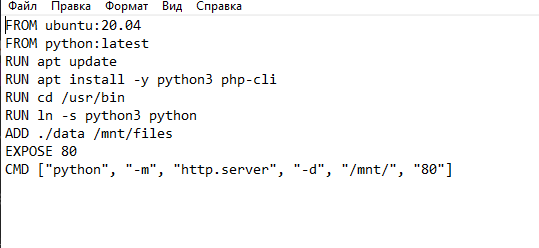


Рисунок 30 – Содержимое Dockerfile

На рисунке 31 представлена сборка и запуск образа на порту 8007 из файла, представленного на рисунке 30.

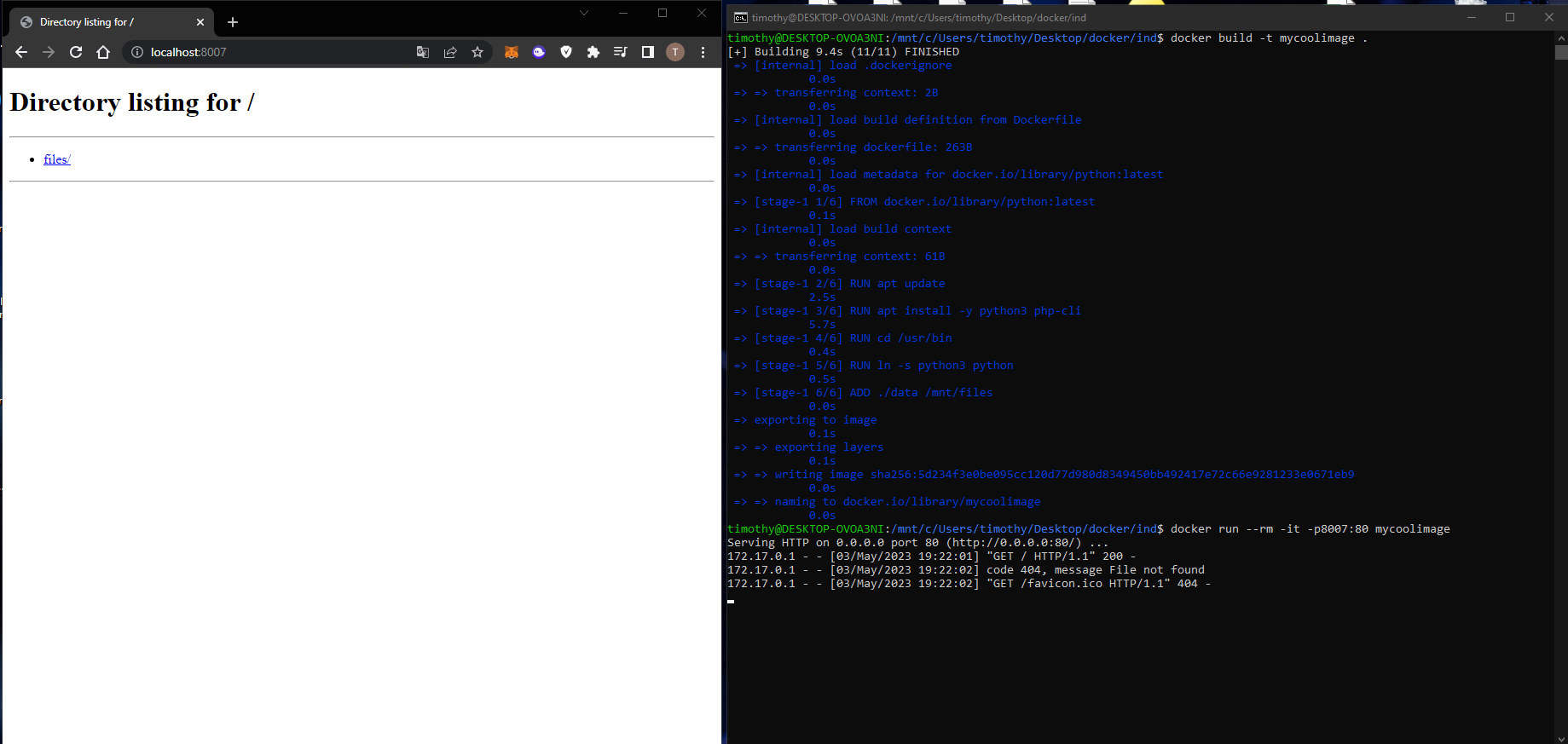


Рисунок 31 – Сборка и запуск

# Вывод

В ходе работы были освоены базовые навыки работы с Docker. Также был реализован запуск собственного образа из Dockerfile в соответствии с индивидуальным вариантом.