|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Инструментального и прикладного программного обеспечения (ИиППО) | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ**  **ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 7**  «Создание контейнера с веб сервером Nginx в Docker» | |
| **по дисциплине** | |
| **«**Настройка и администрирование сервисного программного обеспечения**»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Сидоров С.Д. |
| Принял преподаватель | Хозяинов А.В. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_ г. | *(подпись руководителя)* |

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 понадобиться.

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

Используя инструкцию, представленную в общих сведениях Docker-compose создать следующие сервисы. Работу каждого сервиса продемонстрировать преподавателю.

Задание 1: повторить пример, который был рассмотрен в разделе «Общие сведения». Проект назвать docker\_py. Обязательно продемонстрировать сервис при обсуждении с преподавателем отчёта практической работы. Результат представлен на листингах 1-3 и рисунке 1

Листинг 1 – Dockerfile для серверной части docker\_py

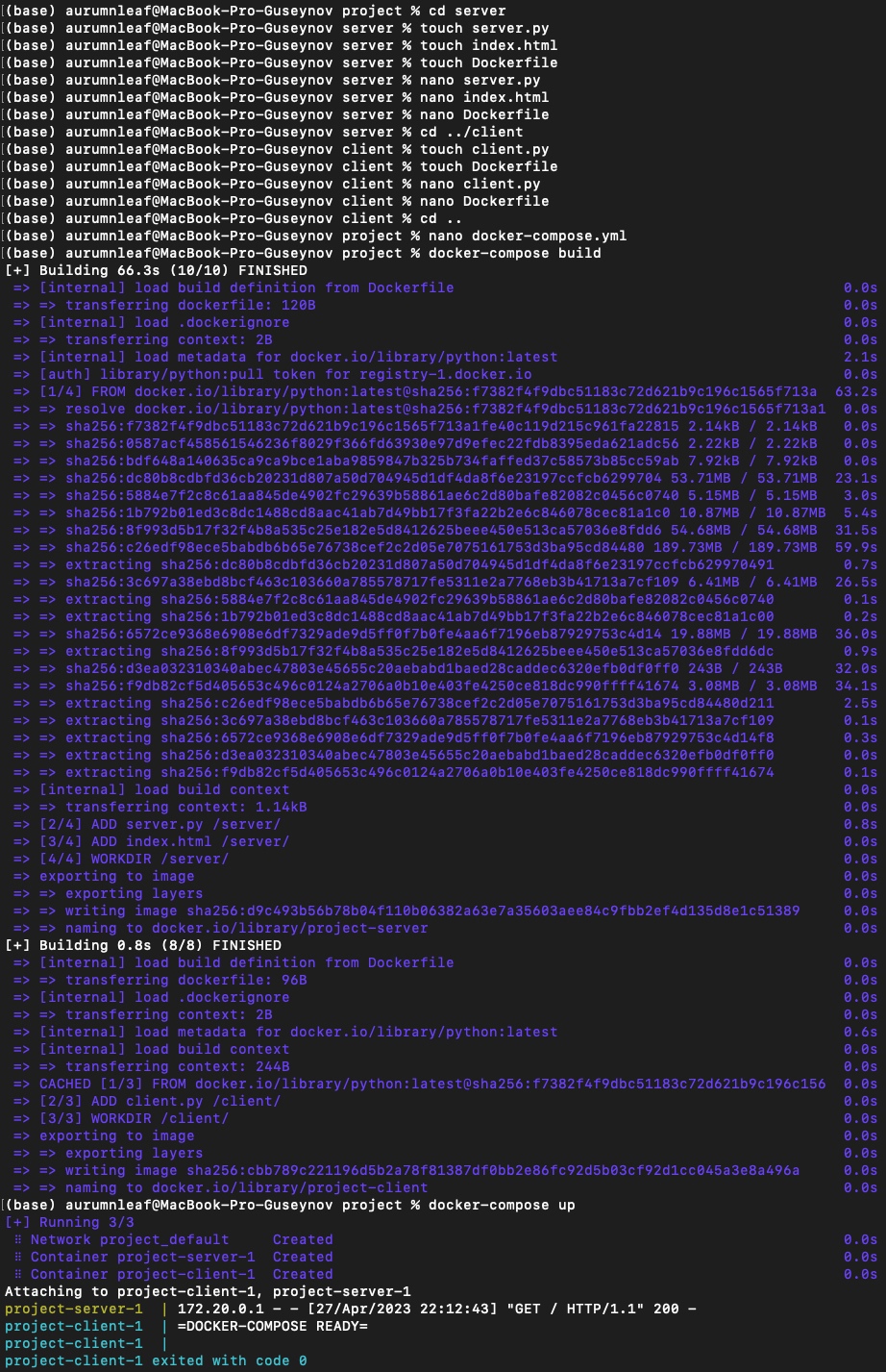
|  |
| --- |
| FROM python:latest  ADD server.py /server/  ADD index.html /server/  WORKDIR /server/ |

Листинг 2 – Dockerfile для клиентской части docker\_py

|  |
| --- |
| FROM python:latest  ADD client.py /client/  WORKDIR /client/ |

Листинг 3 – Docker-compose для docker\_py

|  |
| --- |
| version: "3.8"  services:  server:  build: server/  command: python ./server.py  ports:  - 1234:1234  client:  build: client/  command: python ./client.py  network\_mode: host  depends\_on:  - server |



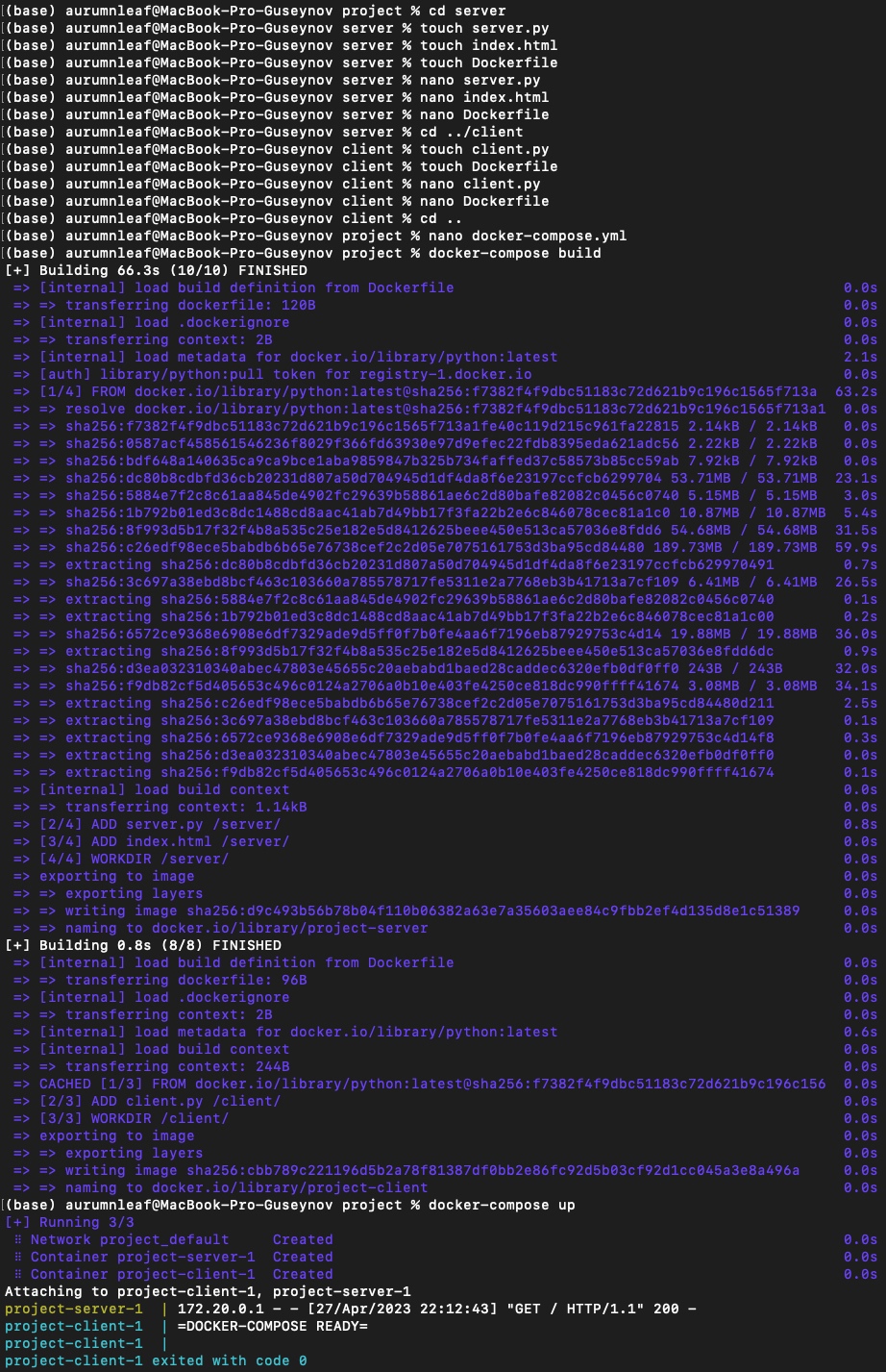


Рисунок 1 – Создание и запуск проекта

Задание 2: cоздать аналогичный сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо языка Python использовать язык JavaScript. Проект назвать docker\_js. Результат представлен на листингах 4-6 и рисунке 2

Листинг 4 – Dockerfile для серверной части docker\_js

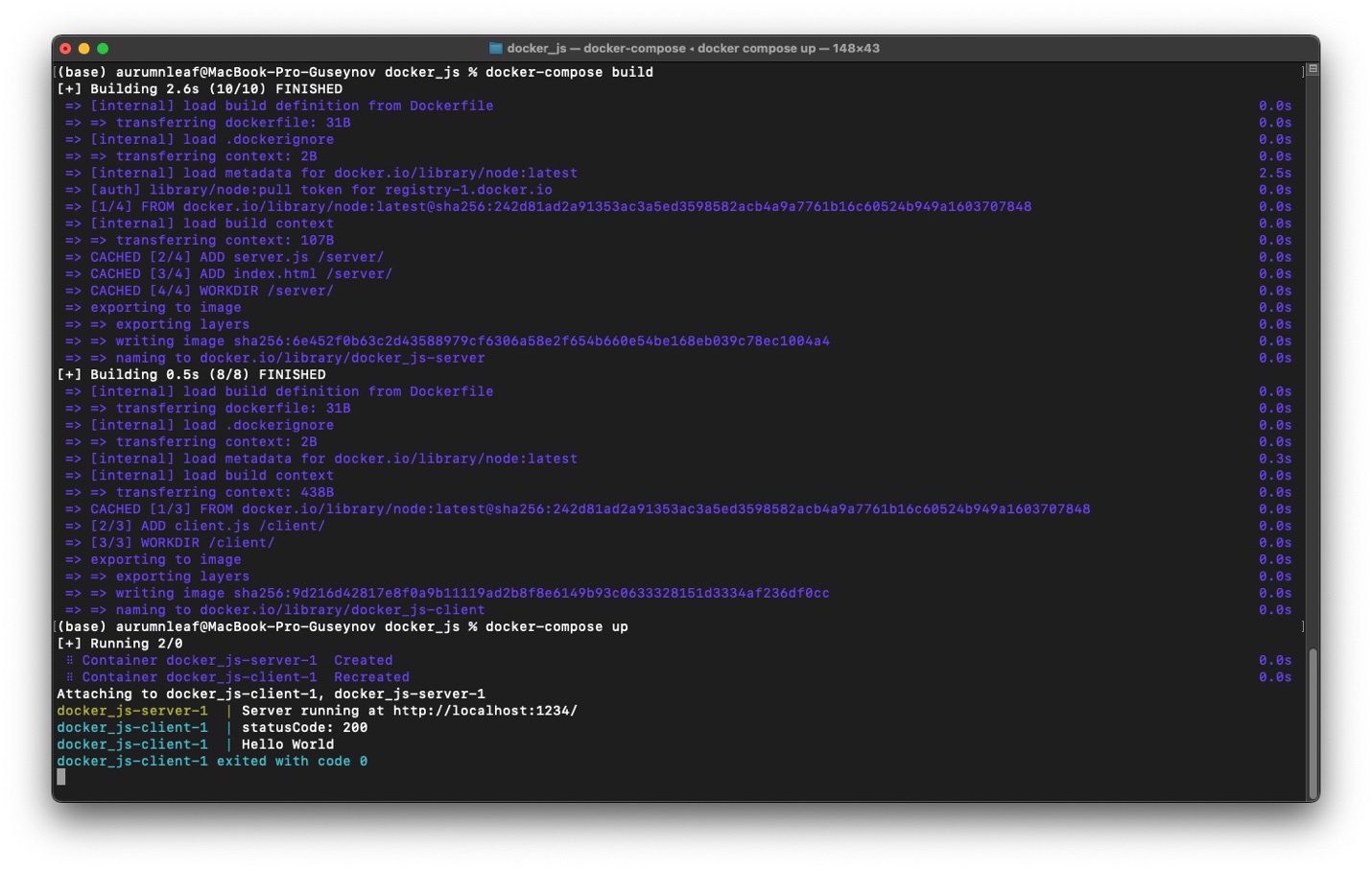
|  |
| --- |
| FROM python:latest  ADD server.py /server/  ADD index.html /server/  WORKDIR /server/ |

Листинг 5 – Dockerfile для клиентской части docker\_js

|  |
| --- |
| FROM node:latest  ADD client.js /client/  WORKDIR /client/ |

Листинг 6 – Docker-compose для docker\_js

|  |
| --- |
| version: "3.8"  services:  server:  build: server/  command: node ./server.js  ports:  - 1234:1234  client:  build: client/  command: node ./client.js  network\_mode: host  depends\_on:  - server |



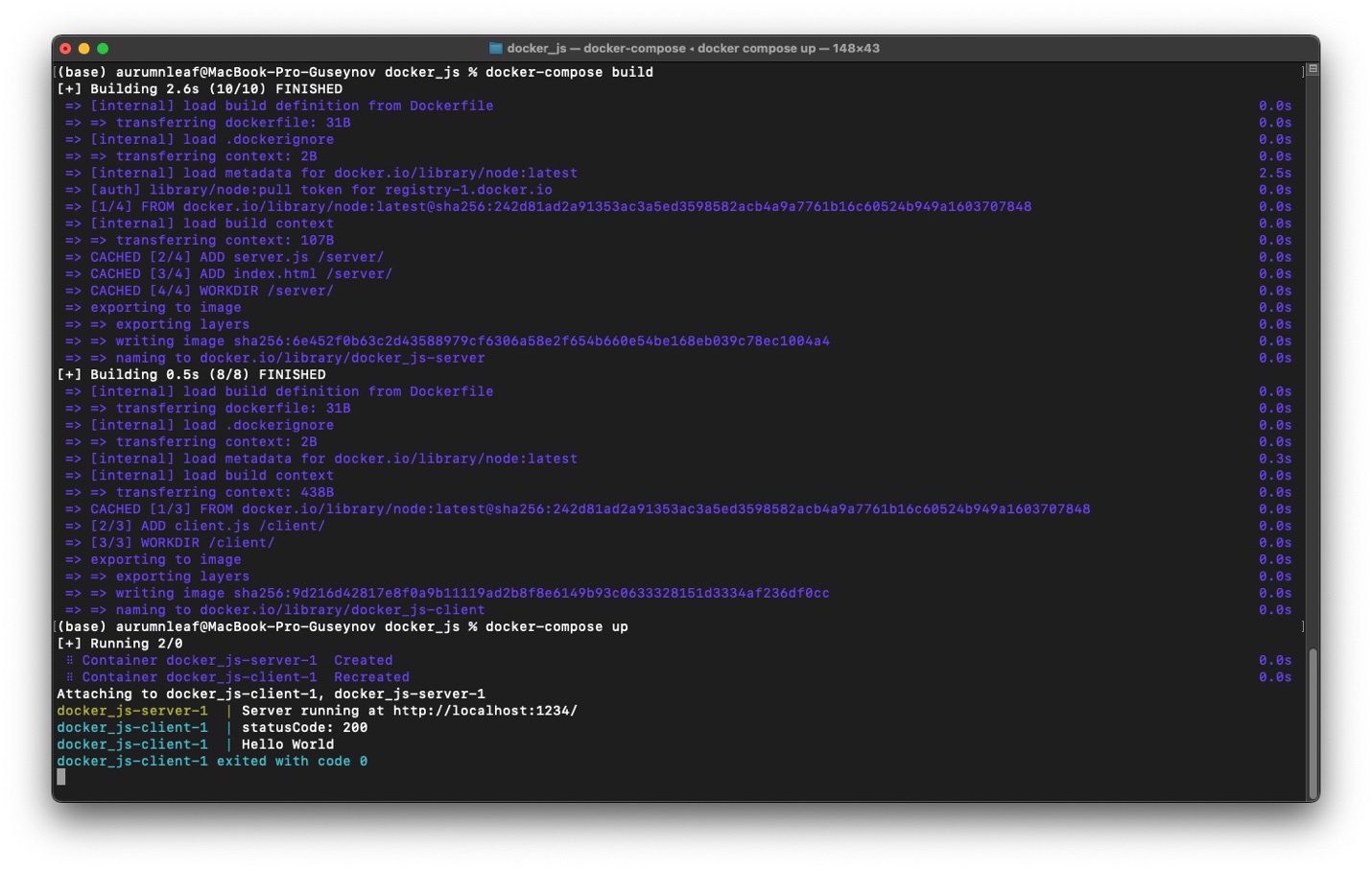
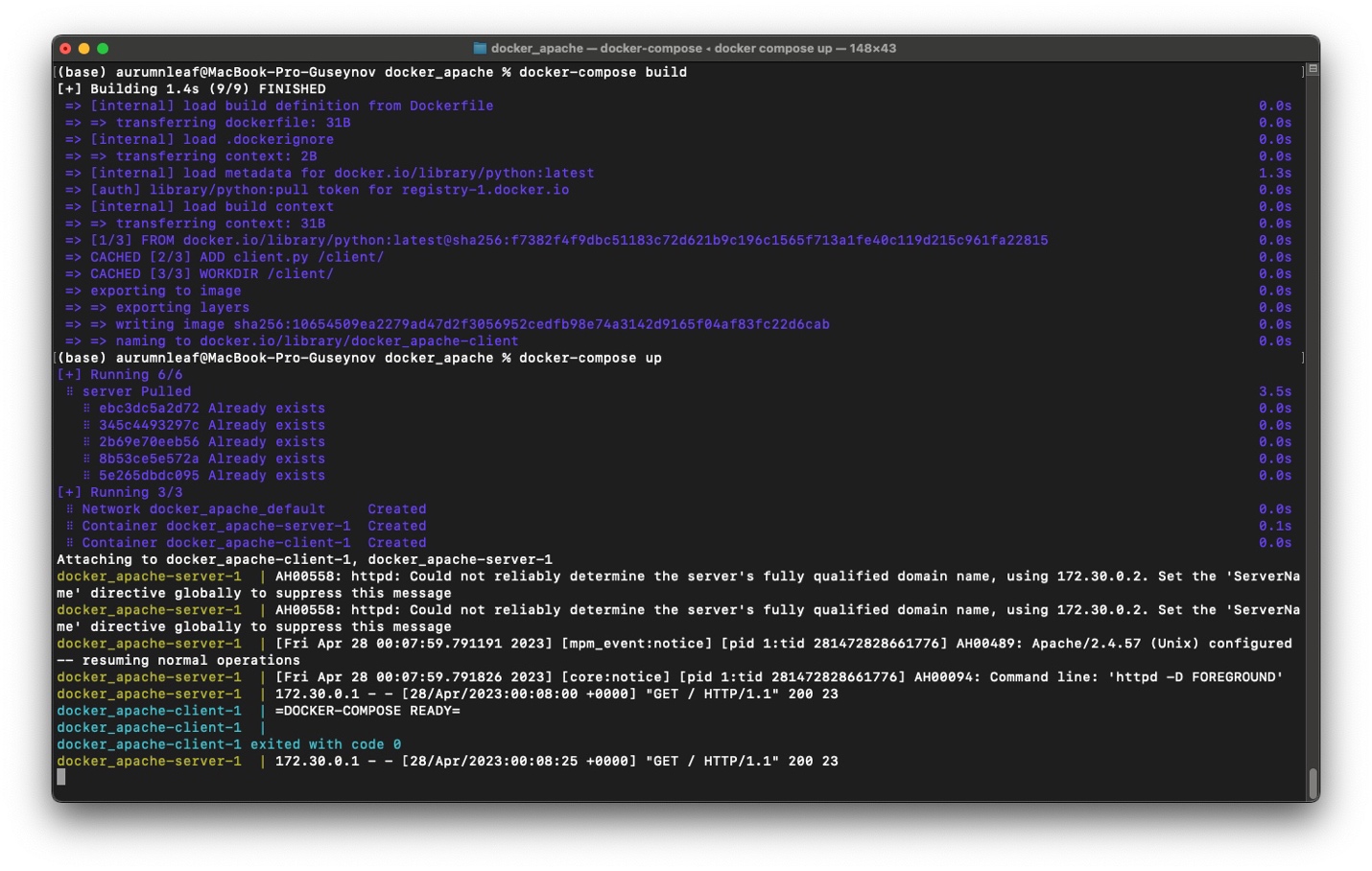


Рисунок 2 – Сборка и запуск docker\_js

Задание 3: Создать сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо сервера на Python использовать веб сервер Apache. Проект назвать docker\_apache. Результат представлен на листинге 6 и рисунке 3.

Листинг 7 – docker-compose для приложения docker\_apache

|  |
| --- |
| version: "3.8"  services:  server:  image: httpd:latest  ports:  - "1234:80"  volumes:  - ./server:/usr/local/apache2/htdocs/  client:  build: client/  command: python ./client.py  network\_mode: host  depends\_on:  - server |



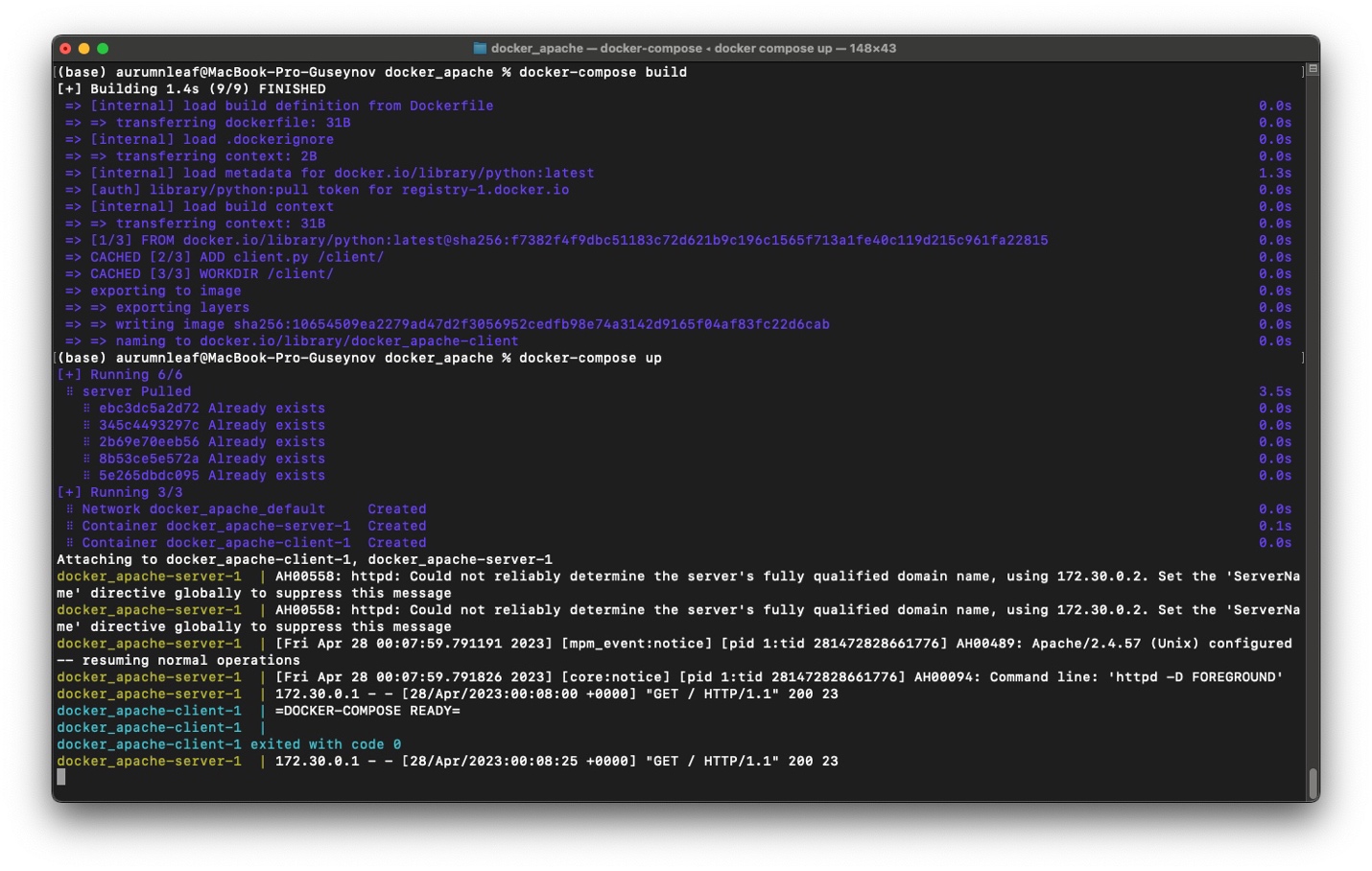
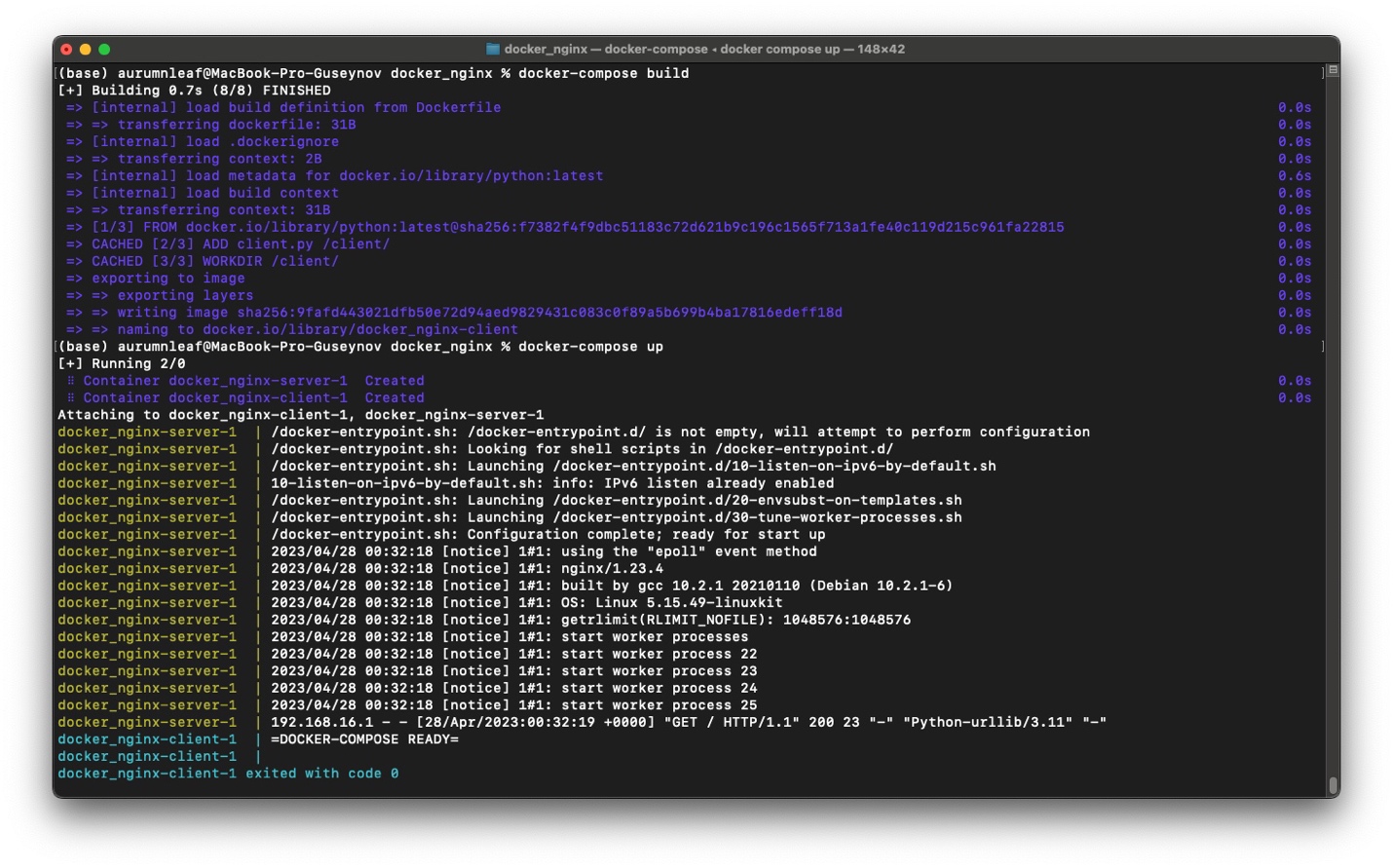


Рисунок 3 – Сборка и запуск docker\_apache

Задание 4: cоздать сервис как в примере, рассмотренном в разделе «Общие сведения», но вместо сервера на Python использовать веб сервер Nginx. Проект назвать docker\_nginx. Результат представлен на листинге 8 и рисунке 4.

Листинг 8 – docker-compose для приложения docker\_nginx

|  |
| --- |
| version: "3.8"  services:  server:  image: nginx:latest  ports:  - "1234:80"  volumes:  - ./server:/usr/share/nginx/html/  client:  build: client/  command: python ./client.py  network\_mode: host  depends\_on:  - server |



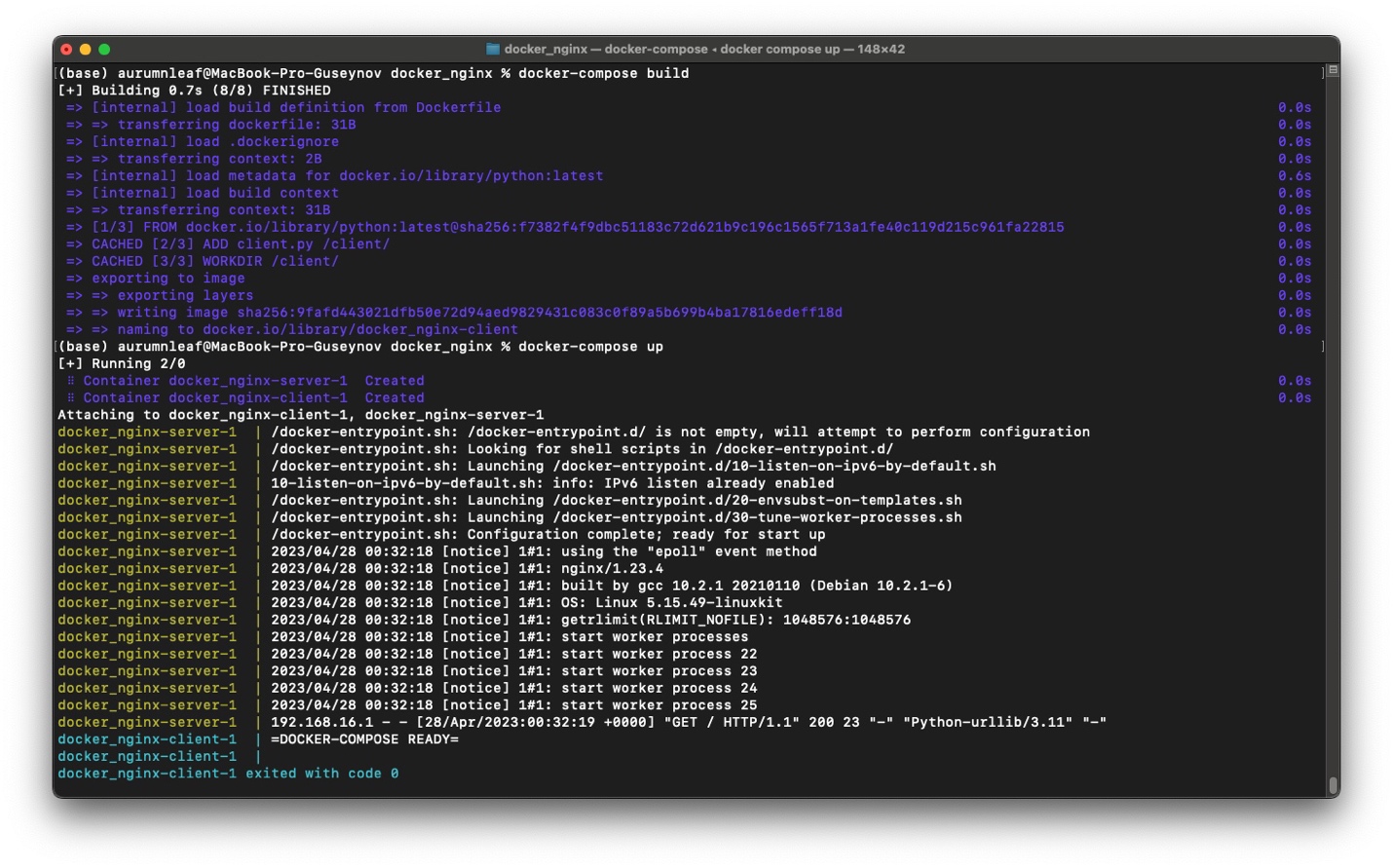
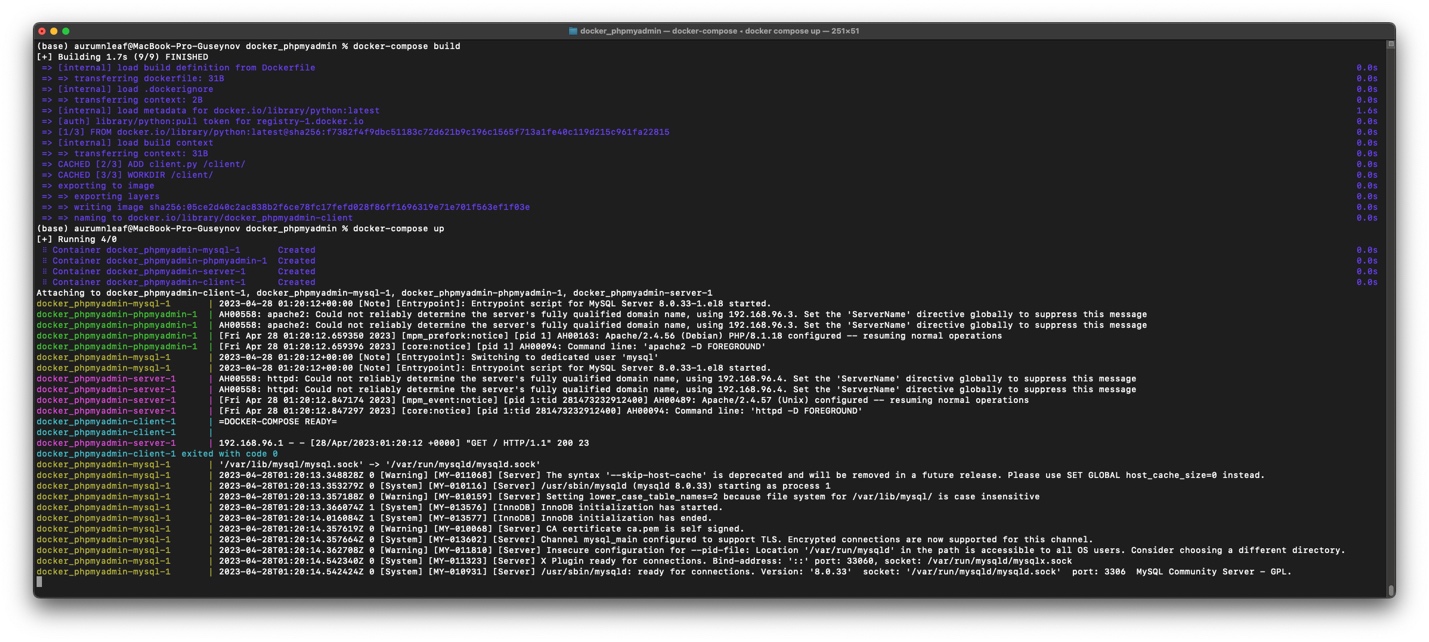


Рисунок 4 – Сборка и запуск docker\_nginx

Задание 5: cоздать сервис в состав которого входят следующие компоненты apache, mysql, phpMyAdmin. Проект назвать docker\_phpmyadmin. Результат представлен на листинге 9 и рисунках 5, 6.



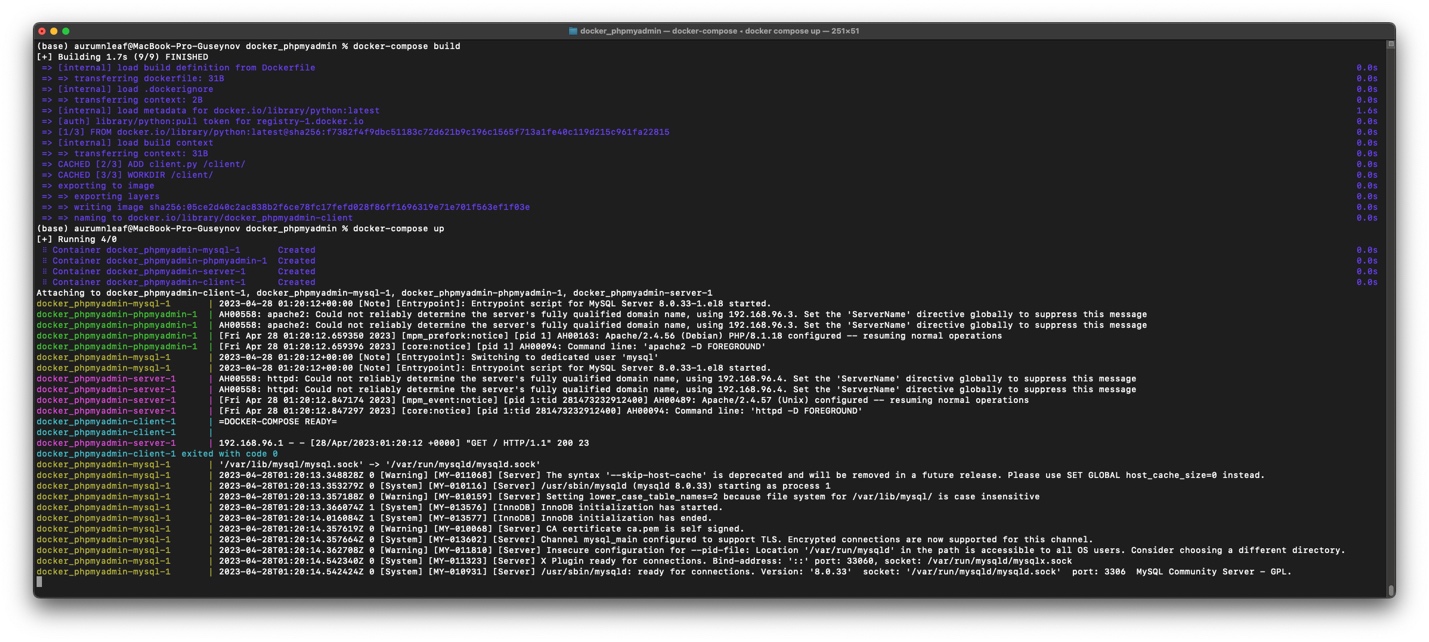


Рисунок 5 – Сборка и запуск docker\_phpmyadmin

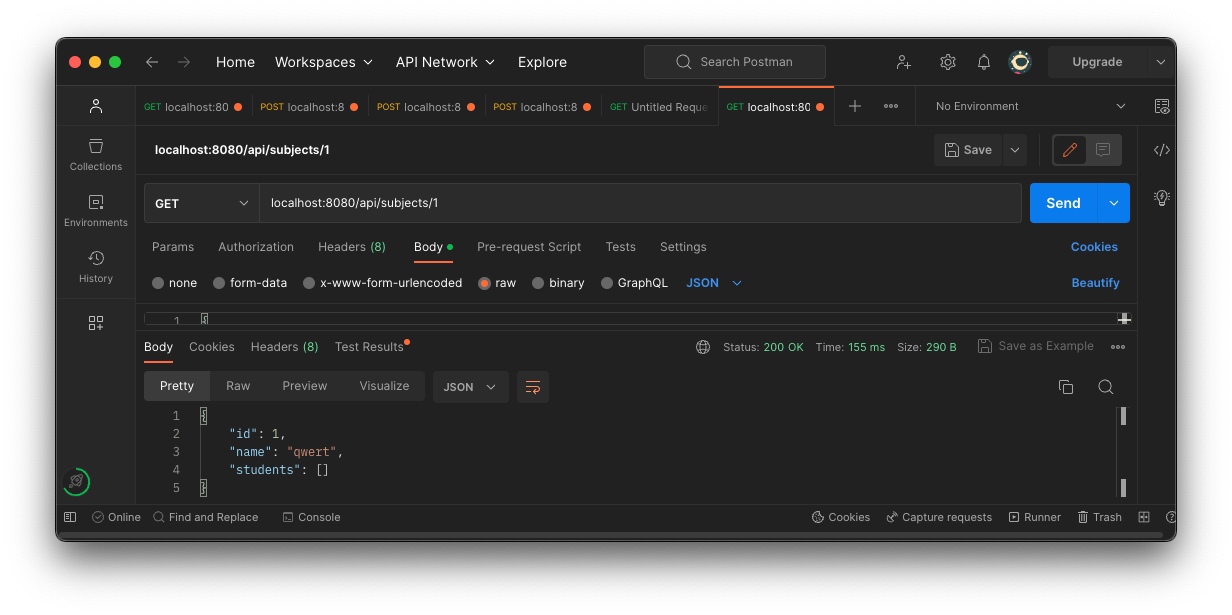


Рисунок 6 – Проверка работоспособности БД

Листинг 9 – docker-compose для приложения docker\_phpmyadmin

|  |
| --- |
| version: "3.8"  services:  server:  image: httpd:latest  ports:  - "1234:80"  volumes:  - ./server:/usr/local/apache2/htdocs/  depends\_on:  - phpmyadmin  - mysql  mysql:  image: mysql:latest  environment:  MYSQL\_ROOT\_PASSWORD: 1111  MYSQL\_DATABASE: SpringQueue  volumes:  - ./data:/var/lib/mysql  ports:  - 3307:3306  phpmyadmin:  image: phpmyadmin:latest  ports:  - "1235:80"  environment:  PMA\_HOST: mysql  PMA\_PORT: 3307  MYSQL\_ROOT\_PASSWORD: 1111  depends\_on:  - mysql  client:  build: client/  command: python ./client.py  network\_mode: host  depends\_on:  - server |

**ВЫВОД**

В ходе данной работы был получен опыт создания докер контейнеров, были изучены docker-compose и Dockerfile.