|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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 № 6**  «Создание контейнеров 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-composer

Теоретическое введение

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.  
 Docker-compose написан в формате YAML который по своей сути похож на JSON или XML. В формате YAML имеют значения пробелы и табуляции, именно пробелами отделяются названия параметров от их значений.  
 Команды Docker-compose  
 Чтобы просмотреть команды и их действия, необходимо выполнить в терминале :  
 $ docker-compose –help  
 Docker Compose Up  
 Эта команда используется, для запуска или служб в файле docker-compose.yml.  
 Файл Docker-compose.yml определяет сервисы, их свойства, переменные и зависимости.  
 $ docker-compose up  
 Для просмотра запущенных контейнеров:  
 $ docker-compose ps  
 Также можно указать docker-compose запустить только один сервис, например  
 $ docker-compose up my\_container1

Выполнение работы

Используя Docker-compose создать контейнеры для работы

Задание 1: Веб-сервера apache, так чтобы можно было запускать самостоятельно созданный сайт. Продумайте компоновку контейнера на свое усмотрение. Результат представлен на листинге 1 и рисунках 1, 2

Листинг 1 – docker-compose для apache

|  |
| --- |
| version: "3.9"  services:  apache:  image: httpd:latest  ports:  - "8080:80"  volumes:  - ./my\_website:/usr/local/apache2/htdocs/ |

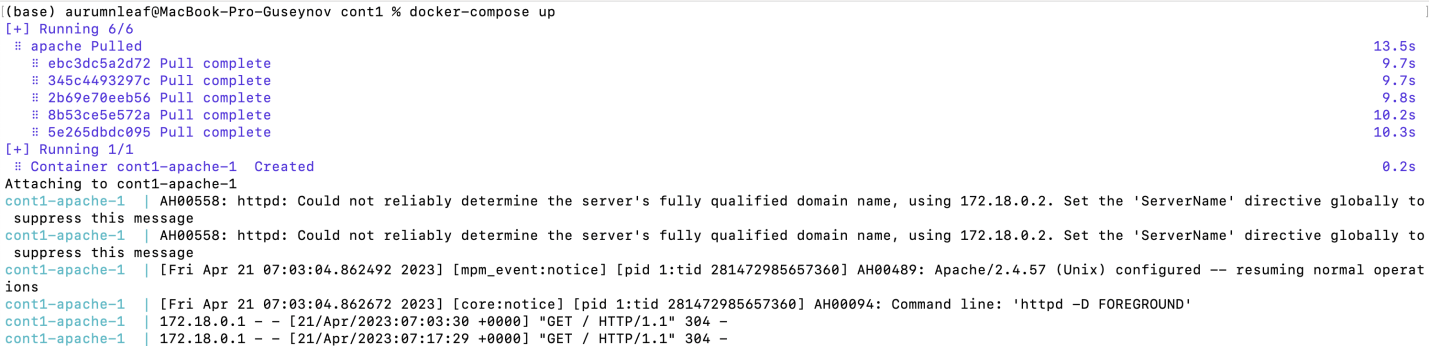


Рисунок 1 – Запуск docker-compose для apache

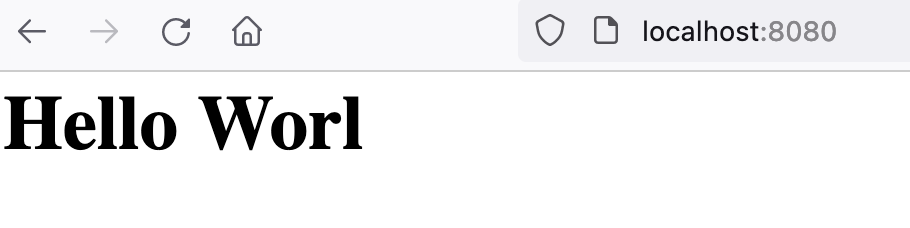


Рисунок 2 – Сайт, запущенный на apache

Задание 2: СУБД mysql, при обращении к которой из самостоятельно созданного приложения можно было помещать и читать данные в базу данных. Продумайте компоновку контейнера на свое усмотрение. Результат представлен на листинге 2 и рисунках 3, 4

Листинг 2 – docker-compose для mysql

|  |
| --- |
| version: "3.9"  services:  mysql:  image: mysql:latest  environment:  MYSQL\_ROOT\_PASSWORD: root  MYSQL\_DATABASE: Spring  volumes:  - ./data:/var/lib/mysql  ports:  - 3307:3306 |

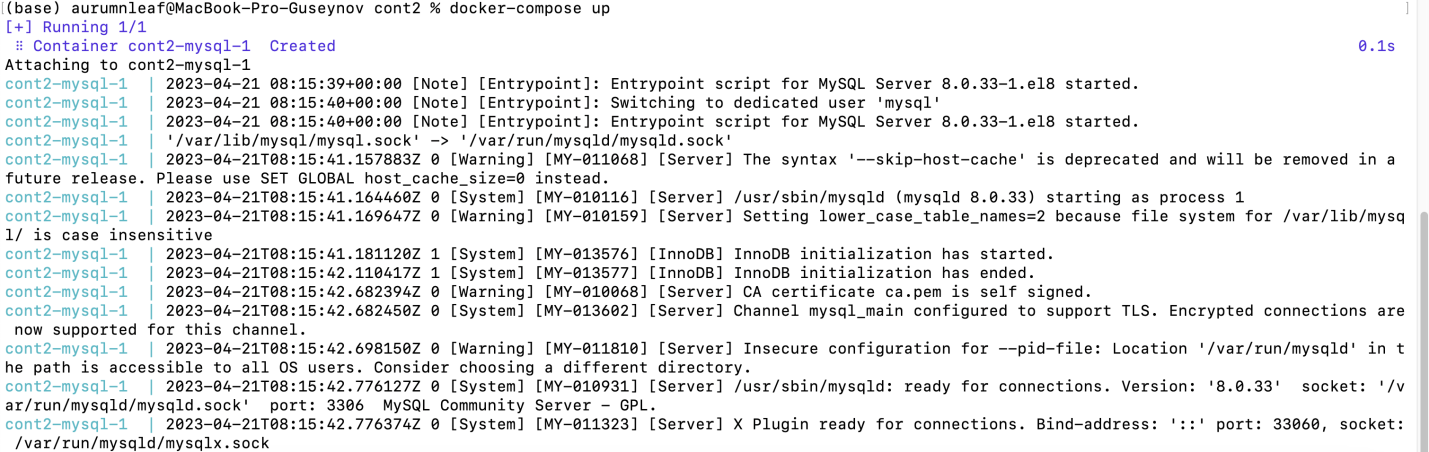


Рисунок 3 – Запуск mysql

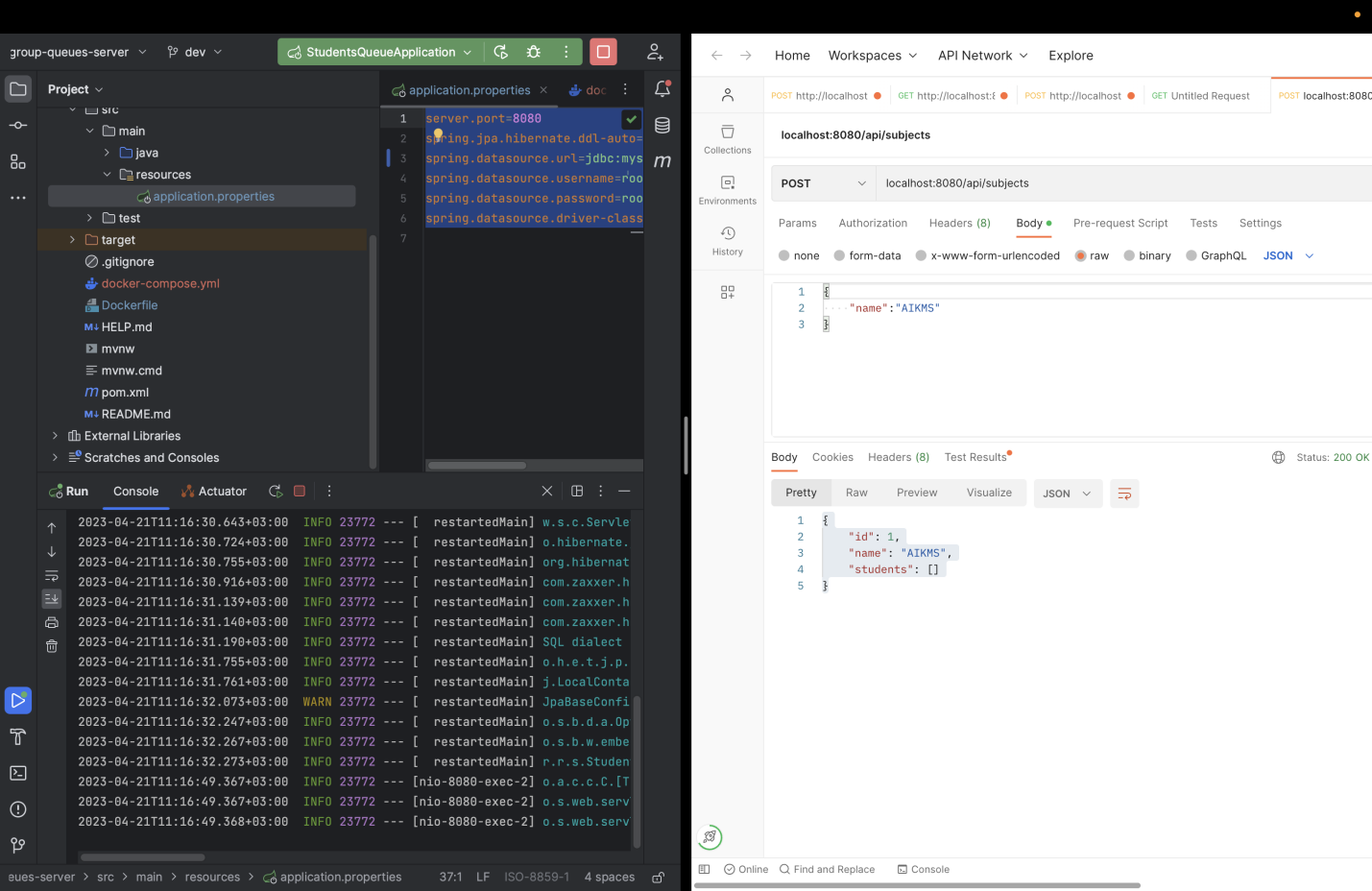


Рисунок 4 – Проверка работоспособности БД

Задание 3: поместить в контейнер собственное приложение. Продумайте компоновку контейнера на свое усмотрение. Результат представлен на листингах 3, 4 и рисунке 5.

Листинг 3 – Dockerfile для приложения tresk

|  |
| --- |
| FROM openjdk:latest COPY target/tresk-0.0.1-SNAPSHOT.jar /app.jar ENTRYPOINT ["java","-jar","/app.jar"] |

Листинг 4 – docker-compose для приложения tresk

|  |
| --- |
| version: "3.9" services:  web:  build: .  ports:  - "8080:8080" |

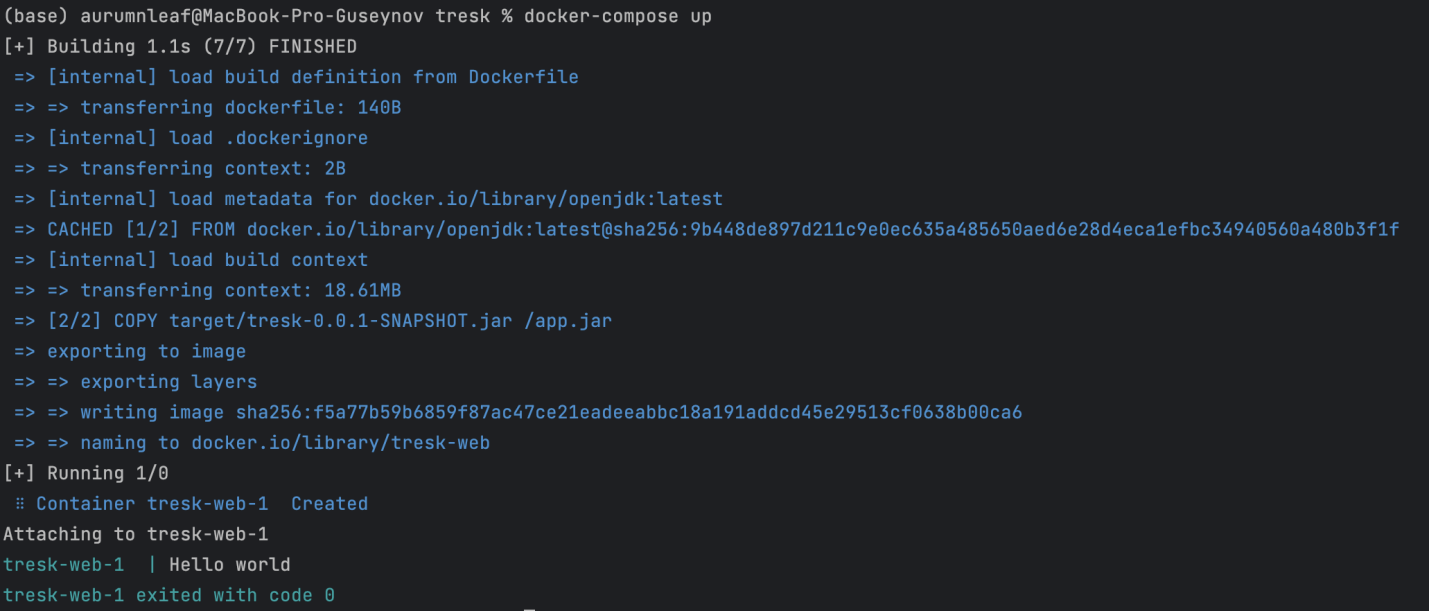


Рисунок 5 – Запуск docker-compose

Задание 4: самостоятельно придумать содержимое контейнера, которое принесет практическую пользу. Была использована серверной часть приложения для очередей (проект трпп). Результат представлен на листингах 5, 6 и рисунке 6.

Листинг 5 – Dockerfile приложения

|  |
| --- |
| FROM openjdk:latest VOLUME /tmp ARG JAR\_FILE=target/\*.jar COPY ${JAR\_FILE} app.jar ENTRYPOINT ["java","-jar","/app.jar" |

Листинг 6 – Dockerfile приложения

|  |
| --- |
| version: '3' services:  db:  image: mysql:latest  environment:  - MYSQL\_ROOT\_PASSWORD=root  - MYSQL\_DATABASE=SpringQueue  ports:  - "8889:3306"   app:  build:  context: .  dockerfile: Dockerfile  ports:  - "8080:8080"  depends\_on:  - db  environment:  - SPRING\_DATASOURCE\_URL=jdbc:mysql://db:3306/SpringQueue  - SPRING\_DATASOURCE\_USERNAME=root  - SPRING\_DATASOURCE\_PASSWORD=root |

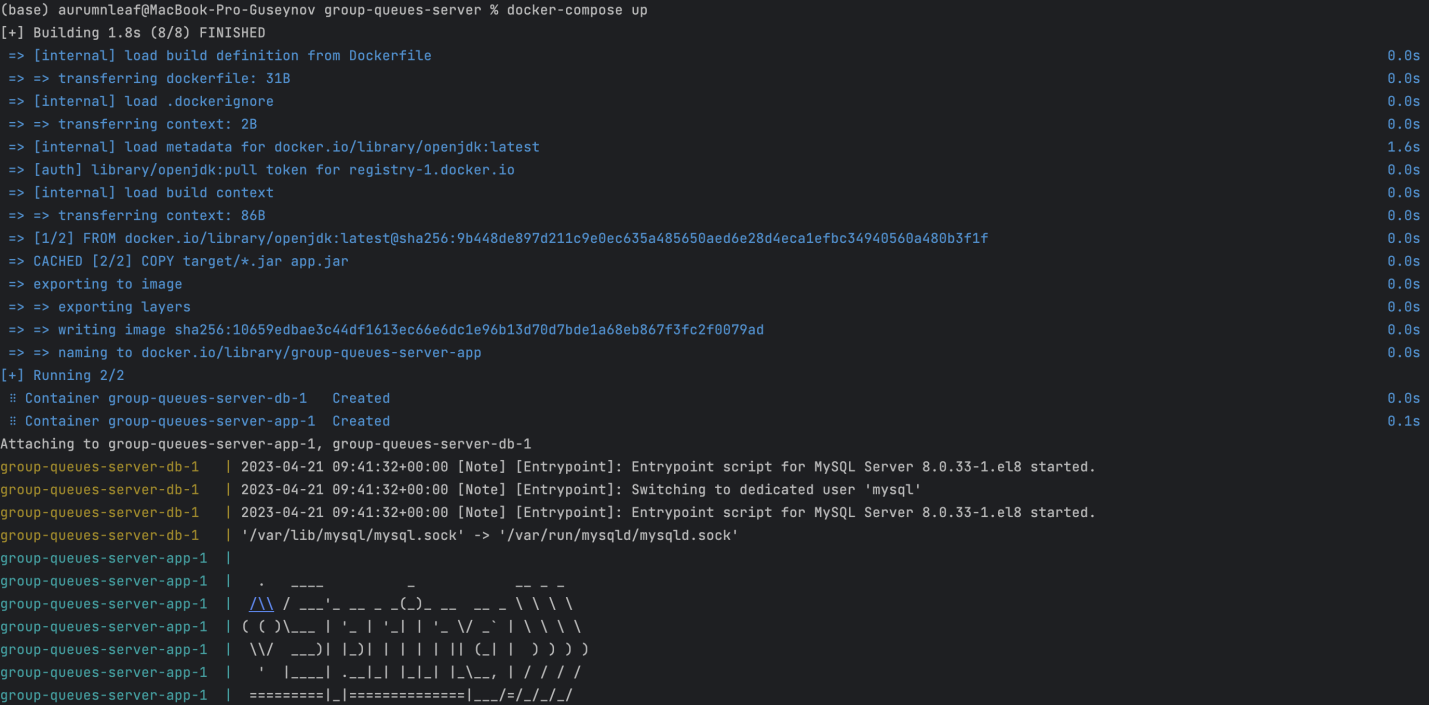


Рисунок 6 – Запуск полезного приложения

**ВЫВОД**

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