



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения
(ИиППО)

ПРАКТИЧЕСКАЯ РАБОТА №3

по дисциплине «Технологии виртуализации клиент-серверных приложений»

Студент группы ИКБО-10-19

Дараган Федор Алексеевич

Руководитель практической работы

(подпись студента)
ассистент Мельников Д.А.

(подпись руководителя)

Работа представлена

«__» _____ 2022 г.

Допущен к работе

«__» _____ 2022 г.

Москва 2022

Оглавление

Практическая работа № 3: «Базовое использование Dockerfile».....	3
Цель работы.....	3
Ход работы.....	3
Вывод.....	7
Ответы на вопросы к практической работе.....	8
1. Опишите процесс запуска приложения внутри контейнера Linux, используя Docker.....	8
2. Что такое образ Docker и для чего он нужен?.....	8
3. Как соотносятся между собой файлы Dockerfile и Docker-Compose?....	8
4 Что такое Dockerfile?.....	8
5 Опишите политики перезапуска контейнера.....	8
6 Назовите все возможные состояния контейнеров.....	8
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	9

Практическая работа № 3: «Базовое использование Dockerfile»

Цель работы

Изучить и применить на практике технологию Docker.

Ход работы

Было составлено 2 Dockerfile, показанных на листингах 1 и 2.

Листинг 1 — Первый Dockerfile.

```
FROM alpine:latest

LABEL author=laefad

RUN apk update && apk upgrade

RUN addgroup -S laefadgroup && adduser -S laefad -G laefadgroup
USER laefad

CMD whoami
```

Листинг 2 — Второй Dockerfile

```
FROM alpine:latest

ENV MESSAGE "Hello from Dockerfile 2"

EXPOSE 80

RUN apk update && apk upgrade && apk add python3
WORKDIR /workdir
COPY example.txt example.txt
ADD changableTxt.txt cTxt.txt
RUN echo "new text" > cTxt.txt
RUN mkdir myvol
VOLUME /workdir/myvol
RUN echo "$MESSAGE" > myvol/greeting.txt

ENTRYPOINT python3 -m http.server
ONBUILD ADD /workdir /workdir
```

На рисунке 1 показан результат сборки и запуска 1-го Dockerfile.

```
PS D:\Учеба\Виртуалки\Практическая работа 3\part1> docker build -f Dockerfile -t df1 .
[+] Building 0.6s (7/7) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 328 0.0s
=> [internal] load .dockerignore
=> => transferring context: 28 0.0s
=> [internal] load metadata for docker.io/library/alpine:latest 0.5s
=> [1/3] FROM docker.io/library/alpine:latest@sha256:8914eb54f968791faf6a8638949e480fef81e697984fba772b3976835194c6d4 0.0s
=> CACHED [2/3] RUN apk update && apk upgrade 0.0s
=> CACHED [3/3] RUN addgroup -S laefadgroup && adduser -S laefad -G laefadgroup 0.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> writing image sha256:05044e10494f9257a3f514d3b69746a1749b81a5105d81e6b62a2d77f64fd86b 0.0s
=> naming to docker.io/library/df1 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS D:\Учеба\Виртуалки\Практическая работа 3\part1> docker run df1
laefad
```

Рисунок 1. Сборка и запуск 1-го Dockerfile

На рисунке 2 показан результат сборки и запуска 2-го Dockerfile.

```
PS D:\Учеба\Виртуалки\Практическая работа 3\part1> docker build -f Dockerfile.2 -t df2 .
[+] Building 10.9s (13/13) FINISHED
=> [internal] load build definition from Dockerfile.2 0.0s
=> => transferring dockerfile: 416B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 28 0.0s
=> [internal] load metadata for docker.io/library/alpine:latest 0.5s
=> CACHED [1/8] FROM docker.io/library/alpine:latest@sha256:8914eb54f968791faf6a8638949e480fef81e697984fba772b3976835194c6d4 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 97B 0.0s
=> [2/8] RUN apk update && apk upgrade && apk add python3 8.0s
=> [3/8] WORKDIR /workdir 0.1s
=> [4/8] COPY example.txt example.txt 0.0s
=> [5/8] ADD changableTxt.txt cTxt.txt 0.0s
=> [6/8] RUN echo "new text" > cTxt.txt 0.6s
=> [7/8] RUN mkdir myvol 0.6s
=> [8/8] RUN echo "Hello from Dockerfile 2" > myvol/greeting.txt 0.5s
=> exporting to image 0.5s
=> => exporting layers 0.5s
=> writing image sha256:a7f4865b2d7eed5d590173f484f3efdedb04a3d6a3752b32e5ff267ee28cddce 0.0s
=> naming to docker.io/library/df2 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
PS D:\Учеба\Виртуалки\Практическая работа 3\part1> docker run -p 8000:8000 df2
```

Рисунок 2. Сборка и запуск 2-го Dockerfile

На рисунке 3 показан работающий сервер в контейнере.

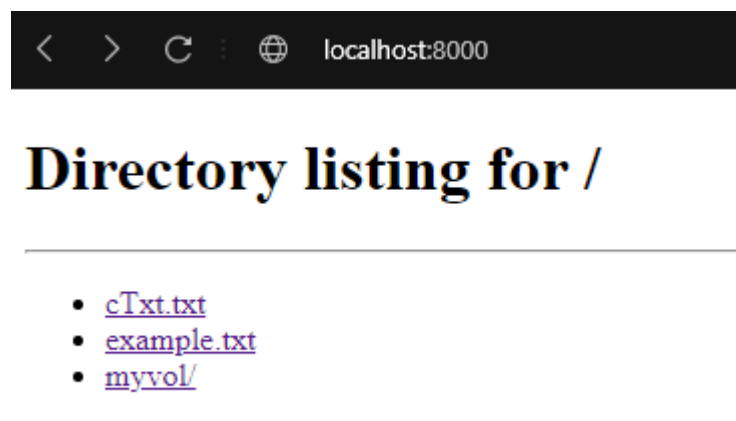


Рисунок 3. Скриншот сайта, работающего в контейнере

Затем был составлен Dockerfile для приложения с использованием нескольких стадий. Для подключения БД был использован docker-compose файл. На листинге 3 показан докерфайл для приложения.

Листинг 3 — Dockerfile для приложения

```
FROM gradle:7.5.1-jdk17-alpine AS build

LABEL CREATOR = "Дараган Федор"
LABEL GROUP = "ИКБО-10-19"

RUN apk update && apk add wget

COPY --chown=gradle:gradle . /home/gradle/src

WORKDIR /home/gradle/src

RUN wget -O mirea.png
https://www.mirea.ru/upload/medialibrary/80f/MIREA_Gerb_Colour.png && \
    mv mirea.png app/src/main/resources

RUN gradle clean build
ONBUILD RUN echo "Сборка и запуск произведены. Автор: ${CREATOR}:${GROUP}"

FROM openjdk:20-slim as prod

RUN mkdir /app

COPY --from=build /home/gradle/src/app/build/libs/app.jar /app/spring-boot-
application.jar

ENTRYPOINT ["java", "-Djava.security.egd=file:/dev/./urandom", "-jar", "-
Dspring.profiles.active=prod", "/app/spring-boot-application.jar"]
```

На листинге 4 показан docker-compose файл для подключения БД.

Листинг 4 — docker-compose файл

```
version: '3.8'

volumes:
  postgres-data:

services:
  app:
```

```

container_name: java
build:
  context: .
  dockerfile: Dockerfile
environment:
  # NOTE: POSTGRES_DB/USER/PASSWORD should match values in db container
  POSTGRES_PASSWORD: postgres
  POSTGRES_USER: postgres
  POSTGRES_DB: postgres
  POSTGRES_HOSTNAME: postgresdb2
  POSTGRES_PORT: 5432

ports:
  - 8000:8000

  # Overrides default command so things don't shut down after the process
ends.
  command: sleep infinity
  depends_on:
    - db

db:
  container_name: postgresdb2
  image: postgres:latest
  restart: unless-stopped
  volumes:
    - postgres-data:/var/lib/postgresql/data
  environment:
    # NOTE: POSTGRES_DB/USER/PASSWORD should match values in app container
    POSTGRES_PASSWORD: postgres
    POSTGRES_USER: postgres
    POSTGRES_DB: postgres

```

На рисунке 4 показано тестирование приложения посредством curl запросов.

```
PS D:\Учеба\Виртуалки\Практическая работа 3\part2> curl -d '{"price":10}' -H "Content-Type: application/json" -X POST http://localhost:8000/ticket
{"id":1,"price":10.0}
PS D:\Учеба\Виртуалки\Практическая работа 3\part2> curl -d '{"price":20}' -H "Content-Type: application/json" -X POST http://localhost:8000/ticket
{"id":2,"price":20.0}
PS D:\Учеба\Виртуалки\Практическая работа 3\part2> curl http://localhost:8000/tickets
[{"id":1,"price":10.0}, {"id":2,"price":20.0}]
PS D:\Учеба\Виртуалки\Практическая работа 3\part2> curl -o image.png http://localhost:8000/image
% Total    % Received % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 77250  100 77250    0     0  3354k      0  0 --:--:-- --:--:-- --:--:-- 3429k
```

Рисунок 4. Скриншот тестирования эндпоинтов

На рисунке 5 показан герб МИРЭА, доступный по эндпоинту.

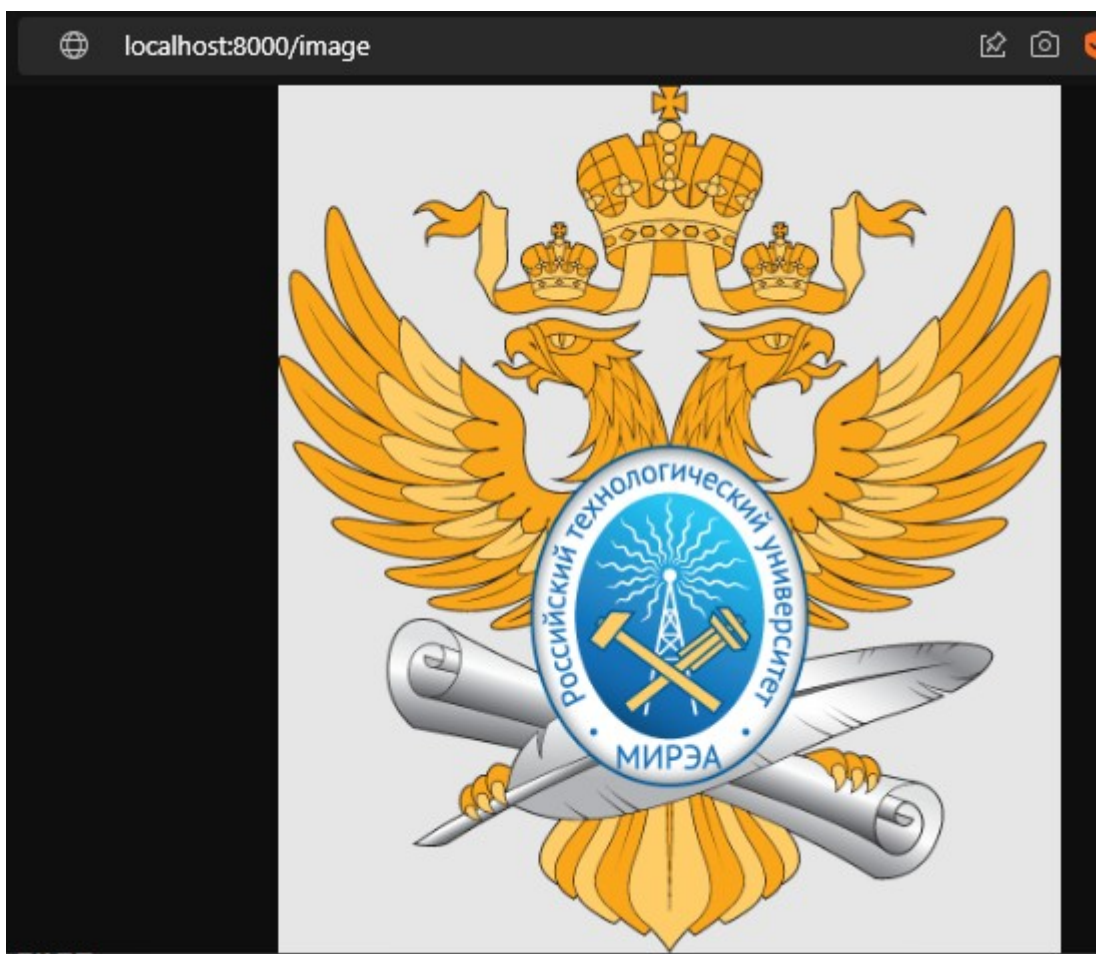


Рисунок 5. Скриншот страницы с гербом МИРЭА

Вывод

Была изучена и применена на практике технология Docker. Составлен Dockerfile с веб-приложением, в котором каждая из команд использована хотя бы 1 раз. А так же был составлен Dockerfile для приложения с БД.

Ответы на вопросы к практической работе

1. Опишите процесс запуска приложения внутри контейнера Linux, используя Docker.

Docker скачивает оригинальный контейнер, указанный в инструкции FROM, применяет к нему требуемые изменения, приложение запускается изолированно внутри контейнера как отдельный процесс.

2. Что такое образ Docker и для чего он нужен?

Образ Docker (Docker Image) - это неизменяемый файл, содержащий исходный код, библиотеки, зависимости, инструменты и другие файлы, необходимые для запуска приложения.

3. Как соотносятся между собой файлы Dockerfile и Docker-Compose?

Файл Docker-compose позволяет задать взаимодействие между множеством образов Docker, создаваемых файлами Dockerfile.

4 Что такое Dockerfile?

Dockerfile — это текстовый файл с инструкциями, необходимыми для создания образа контейнера.

5 Опишите политики перезапуска контейнера.

- No – не перезапускать контейнер;
- on-failure – перезапускать контейнер при ошибке;
- always – всегда перезапускать контейнер, если он останавливается;
- unless-stopped – перезапускать контейнер, если он не был остановлен.

6 Назовите все возможные состояния контейнеров.

Created, restarting, running, removing, paused, exited, dead.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- Виртуализация — Википедия — URL: <https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F> (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.
- Open Virtualization Format — Википедия — URL: https://ru.wikipedia.org/wiki/Open_Virtualization_Format (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.
- Файл VMX – чем открыть, описание формата — URL: <https://open-file.ru/types/vmx> (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.
- ISO-образ — Википедия — URL: <https://ru.wikipedia.org/wiki/ISO-%D0%BE%D0%B1%D1%80%D0%B0%D0%B7> (дата обращения: 05.09.2022). — Режим доступа: для неавториз. пользователей.