



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практическим работам №13-15

по дисциплине «Системное программное обеспечение»

Выполнил:

Студент группы ИВБО-08-22

Стецюк В.В.

Проверил:

Старший преподаватель Воронцов Ю.А.

МОСКВА 2024 г.

СОДЕРЖАНИЕ

1. ПРАКТИЧЕСКАЯ РАБОТА 13	3
1.1 Формулировка задачи	3
1.2 Выполнение работы	3
2. ПРАКТИЧЕСКАЯ РАБОТА 14	6
2.1 Формулировка задачи	6
1.2 Выполнение работы	6
3. ПРАКТИЧЕСКАЯ РАБОТА 15	8
3.1 Формулировка задачи	8
3.1 Выполнение работы	9

1. ПРАКТИЧЕСКАЯ РАБОТА 13

1.1 Формулировка задачи

Цель работы: получить базовые навыки по работе с контейнеризацией с применением Docker.

Задание:

1. загрузить на машину образ и создать контейнер с его помощью, согласно варианту:

- 1) Alpine версии 3.15.4;
- 2) Ubuntu версии 20.04;
- 3) PostgreSQL версии 14;
- 4) Nginx версии 1.21;
- 5) Mysql версии 8.

Если номер варианта больше 5, отсчет начинается с начала, то есть, к примеру, 6 вариант берет 1, 12 вариант берет 2.

2. Запустить контейнер с веб-сервером, отображающий содержимое корневой директории. В хостовой системе сервер должен открываться на порту $(8800 + \text{номер варианта})$. Например, для 22-го варианта это порт 8822.

1.2 Выполнение работы

- 1) Загрузим образ Mysql версии 8 с помощью команды pull:

Создадим контейнер и посмотрим список контейнеров.

```

fake@fake:~$ sudo docker pull postgresql:14
Error response from daemon: error parsing HTTP 403 response body: invalid character '<' looking for beginning of value: "<html><body><h1>403 Forbidden</h1>\nSince Docker is a US company, we must comply with US export control regulations. In an effort to comply with these, we now block all IP addresses that are located in Cuba, Iran, North Korea, Republic of Crimea, Sudan, and Syria. If you are not in one of these cities, countries, or regions and are blocked, please reach out to https://hub.docker.com/support/contact/\n</body></html>\n"
fake@fake:~$ sudo docker pull postgres:14
Error response from daemon: error parsing HTTP 403 response body: invalid character '<' looking for beginning of value: "<html><body><h1>403 Forbidden</h1>\nSince Docker is a US company, we must comply with US export control regulations. In an effort to comply with these, we now block all IP addresses that are located in Cuba, Iran, North Korea, Republic of Crimea, Sudan, and Syria. If you are not in one of these cities, countries, or regions and are blocked, please reach out to https://hub.docker.com/support/contact/\n</body></html>\n"
fake@fake:~$ sudo docker pull mysql:8
8: Pulling from library/mysql
07bc88e18c4a: Pull complete
1a9c1668bf49: Pull complete
1021dda8eecf: Pull complete
fb61b56acac1: Pull complete
0bca83908a5b: Pull complete
165e8b3d37ca: Pull complete
3e1b086f1295: Pull complete
dba651668484: Pull complete
ed90f5355e12: Pull complete
0412f59ab2b5: Pull complete
Digest: sha256:aa021e164da6aacbefc59ed0b933427e4835636be380f3b6523f4a6c9564e1f0
Status: Downloaded newer image for mysql:8
docker.io/library/mysql:8
fake@fake:~$

```

Запустим контейнер командой run, пробросив порт 8823.

```

fake@fake:~$ sudo docker create mysql:8
a4b9fc360bb49e7016281518c4ffa4326a52597b5f76f15e58ecbad0c250fd33
fake@fake:~$ sudo docker ps -a

```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
a4b9fc360bb4	mysql:8	"docker-entrypoint.s..."	19 seconds ago	Created	
7eb1d6073c2d	python	"python -m http.serv..."	4 minutes ago	Exited (0) 45 seconds ago	

```

fake@fake:~$

```

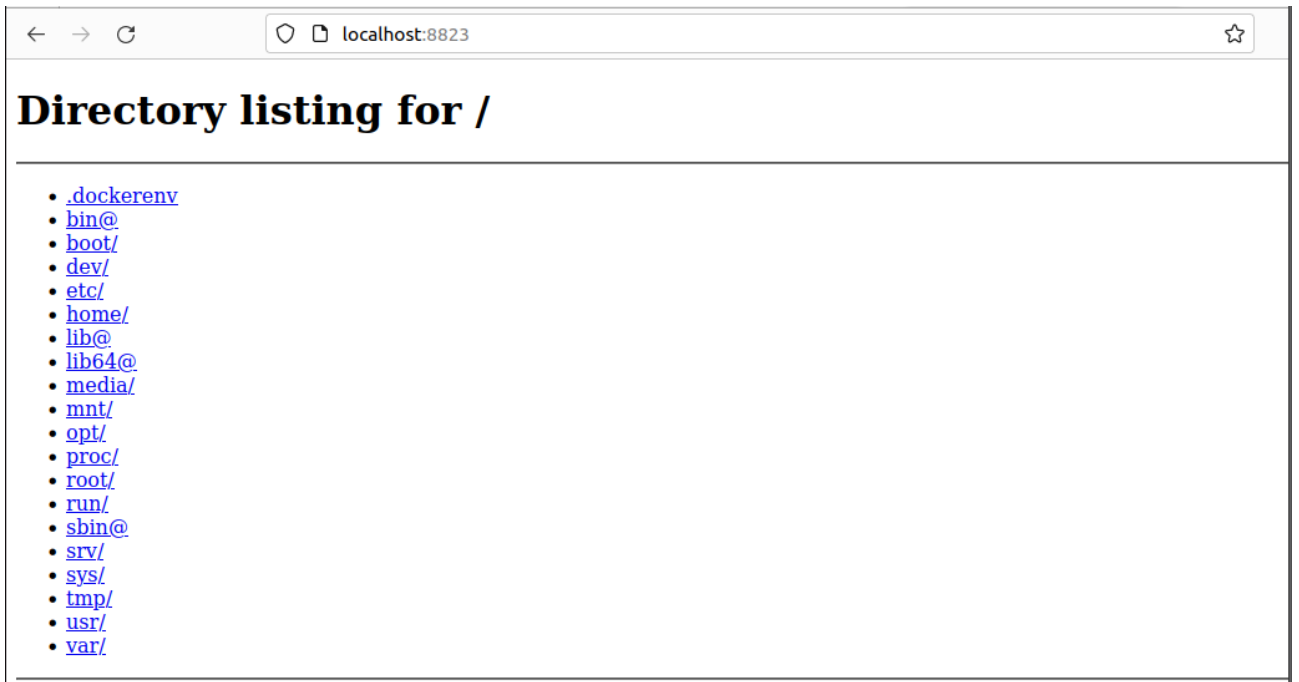
Запустим контейнер командой run, пробросив порт 8823.

```

fake@fake:~$ sudo docker run -it -p8823:8000 python python -m http.server
Unable to find image 'python:latest' locally
latest: Pulling from library/python
c6cf28de8a06: Pull complete
891494355808: Pull complete
6582c62583ef: Pull complete
bf2c3e352f3d: Pull complete
a99509a32390: Pull complete
d46a03def8d9: Pull complete
4429b810e09e: Pull complete
2a4ca5af09fa: Pull complete
Digest: sha256:3966b81808d864099f802080d897cef36c01550472ab3955fdd716d1c665acd6
Status: Downloaded newer image for python:latest
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
172.17.0.1 - - [02/Jun/2024 13:04:21] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [02/Jun/2024 13:04:21] code 404, message File not found
172.17.0.1 - - [02/Jun/2024 13:04:21] "GET /favicon.ico HTTP/1.1" 404 -

```

По адресу `http://localhost:8823/` открывается содержимое корневой директории в контейнере.



2. ПРАКТИЧЕСКАЯ РАБОТА 14

2.1 Формулировка задачи

Цель работы: получить расширенные навыки создания контейнеров в средстве контейнеризации Docker.

Задание:

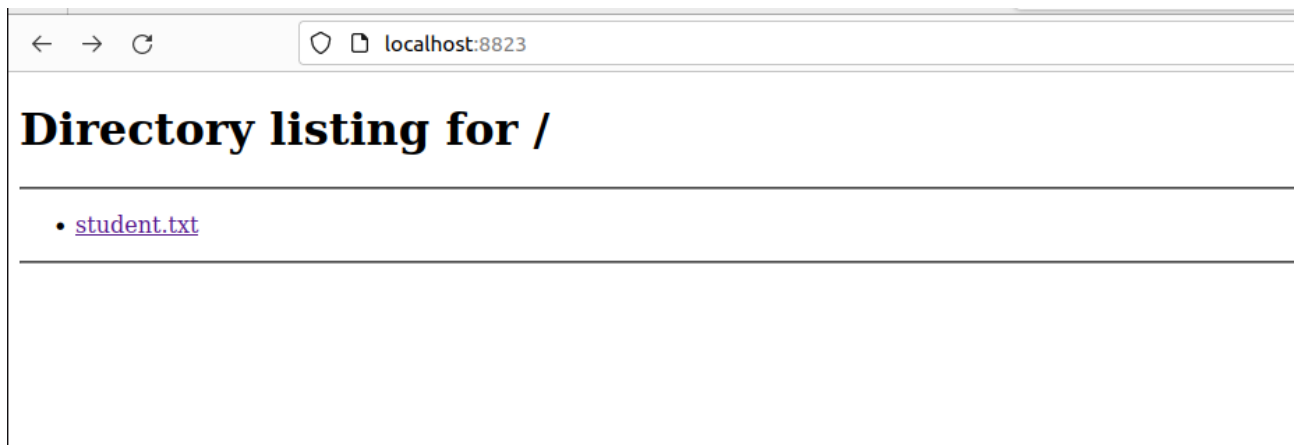
1. Создать директорию data и в ней файл student.txt, содержащий ФИО, название группы и номер варианта.
2. Запустить контейнер с веб-сервером и примонтированной директорией data в директорию /mnt/files/ в контейнере. Контейнер должен отображать содержимое директории /mnt/files и открываться на порту (8800 + номер варианта).

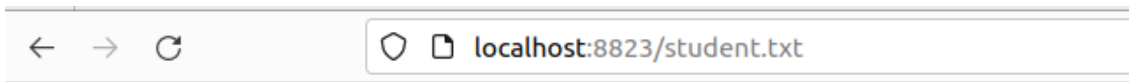
1.2 Выполнение работы

- 1) Создадим директорию data и в ней файл student.txt, содержащий «StetsyukVVIVBO-08-22Var23». Запустим контейнер с веб-сервером и примонтированной директорией data в директорию /mnt/files/ в контейнере.

```
fake@fake:~$ mkdir data
fake@fake:~$ echo "StetsyukVVIVBO-08-22Var23" > data/student.txt
fake@fake:~$ sudo docker run -p 8823:8000 --rm -d --name pyserver -v $(pwd)/data:/mnt/files/ python
python -m http.server -d /mnt/files
4e11d6ab3649f0d62e8f42509d2b1408fc6d0f1244f014a439d01b5966a652c1
fake@fake:~$
```

По адресу <http://localhost:8823/> открывается содержимое папки data.





StetsyukVVIVB0-08-22Var23

3. ПРАКТИЧЕСКАЯ РАБОТА 15

3.1 Формулировка задачи

Цель работы: изучить основные конструкции формирования Dockerfile'ов для сборки образов Docker, посредством написания собственного Dockerfile'a.

Задание:

Написать Dockerfile, собрать образ, запустить контейнер (и записать команду для его запуска). Для монтирования создайте директорию data и в ней файл student.txt, содержащий ФИО, название группы и номер варианта.

Для установки пакетов использовать команду `apt install -у название-пакета`. В качестве примера можно использовать рассмотренный пример Dockerfile'a.

Запустить веб-сервер, отображающий содержимое `/mnt/files`, в хостовой системе должен открываться на порту (8800 + номер варианта). Например, для 22-го варианта это порт 8822. Установить пакет, согласно варианту:

1. cowsay
2. figlet
3. zip
4. imagemagick
5. git
6. patch
7. php-cli
8. postgresql-client
9. mysql-client
10. jq
11. gpg
12. wget
13. nginx
14. nano
15. emacs-nox

3.1Выполнение работы

Напишем Dockerfile, согласно варианту, требуется установить nano, использовать базовый образ ubuntu:20.04 и добавить файл data/student.txt как /mnt/files/student.txt в контейнере.

```
fake@fake:~$ nano Dockerfile
fake@fake:~$ cat Dockerfile
FROM ubuntu:20.04
RUN apt-get update && apt-get install -y mysql-client python3
ADD data/student.txt /mnt/files/student.txt
EXPOSE 80
CMD ["python3", "-m", "http.server", "-d", "/mnt/files", "80"]
fake@fake:~$
```

```
fake@fake:~$ sudo docker build -t mysqlfile .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
             Install the buildx component to build images with BuildKit:
             https://docs.docker.com/go/buildx/
```

```
Sending build context to Docker daemon 112.4MB
```

```
Step 1/5 : FROM ubuntu:20.04
```

```
---> 2abc4dfd8318
```

```
Step 2/5 : RUN apt-get update && apt-get install -y mysql-client python3
```

```
---> Running in 3d44b4a84619
```

```
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
```

```
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
```

```
Get:3 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3675 kB]
```

```
Get:4 http://archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
```

```
Get:5 http://archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
```

```
---> b8a075a65367
```

```
Step 5/5 : CMD ["python3", "-m", "http.server", "-d", "/mnt/files", "80"]
```

```
---> Running in 508652334cea
```

```
Removing intermediate container 508652334cea
```

```
---> 8803621a75cc
```

```
Successfully built 8803621a75cc
```

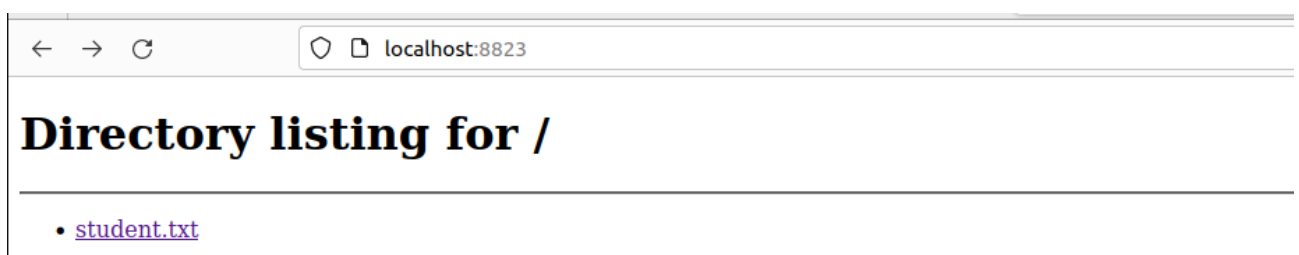
```
Successfully tagged mysqlfile:latest
```

```
fake@fake:~$
```

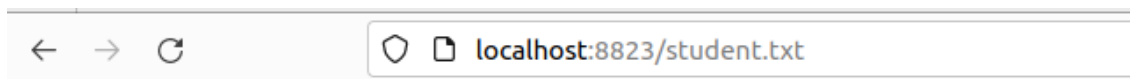
```
fake@fake:~$ sudo docker run --rm -d --name spo15-sql -p 8823:8000 mysqlfile
9c4cb487c56b5716872c2f4f866cf0eea53e668322446cfdcfad17ed6f9d277a6
```

```
fake@fake:~$
```

Соберем образ и запустим контейнер.



MySQL установлен.



StetsyukVVIVB0-08-22Var23

```
fake@fake:~$ sudo docker exec -it spo15-sql bash
root@9c4cb487c56b:/# ls
bin    dev    home  lib32  libx32  mnt    proc  run    srv    tmp    var
boot  etc    lib   lib64  media   opt    root  sbin   sys    usr
```

```
root@9c4cb487c56b:/home# mysql --version
mysql Ver 8.0.36-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
root@9c4cb487c56b:/home# mysql --help
mysql Ver 8.0.36-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Usage: mysql [OPTIONS] [database]
  -?, --help                Display this help and exit.
  -I, --help                Synonym for -?.
  --auto-rehash              Enable automatic rehashing. One doesn't need to use
                             'rehash' to get table and field completion, but startup
                             and reconnecting may take a longer time. Disable with
                             --disable-auto-rehash.
                             (Defaults to on; use --skip-auto-rehash to disable.)
  -A, --no-auto-rehash      No automatic rehashing. One has to use 'rehash' to get
```