



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение
высшего образования*

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных
технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4

по дисциплине

«Технология разработки программных приложений»

Тема: «Docker»

Выполнил студент группы ИКБО-25-22

Ракитин В.А.

Принял

Мельников Д. А.

Москва 2024

СОДЕРЖАНИЕ

1 Цель работы	3
2 Задание для выполнения	4
2.1 Образы.....	4
2.2 Изоляция	4
2.3 Работа с портами	5
2.4 Именованные контейнеры, остановка и удаление	6
2.5 Постоянное хранение данных	6
2.6 Переменные окружения	9
2.7 Dockerfile	9
2.8 Индивидуальные задания.....	10
3 Выводы о проделанной работе	12

1 Цель работы

Ознакомиться с основными понятиями и возможностями Docker, научиться устанавливать и использовать Docker для создания и управления контейнерами, а также понять принципы работы с образами и контейнерами в Docker.

2 Задание для выполнения

2.1 Образы

Просмотр имеющегося образа: `docker images`. Загрузка образа: `docker pull Ubuntu`. Просмотр имеющегося образа ещё раз: `docker images`. Просмотр списка контейнеров, выполнив команду: `docker ps -a`.

```
version@MSI:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix://
/var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix
/var/run/docker.sock: connect: permission denied
version@MSI:~$ sudo docker images
REPOSITORY      TAG              IMAGE ID         CREATED          SIZE
hello-world     latest          d2c94e258dcb    11 months ago   13.3kB
version@MSI:~$ docker pull ubuntu
Using default tag: latest
permission denied while trying to connect to the Docker daemon socket at unix://
/var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/images/creat
e?fromImage=ubuntu&tag=latest": dial unix /var/run/docker.sock: connect: permiss
ion denied
version@MSI:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
3c645031de29: Pull complete
Digest: sha256:1b8d8ff4777f36f19bfe73ee4df61e3a0b789caeff29caa019539ec7c9a57f95
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout qui
ckview ubuntu
version@MSI:~$ sudo docker images
REPOSITORY      TAG              IMAGE ID         CREATED          SIZE
ubuntu          latest          7af9ba4f0a47    13 days ago     77.9MB
hello-world     latest          d2c94e258dcb    11 months ago   13.3kB
version@MSI:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND          CREATED          STATUS
PORTS         NAMES
```

Рисунок 1 – Просмотр имеющегося образа, загрузка образа просмотр еще раз

2.2 Изоляция

Посмотрим информацию о хостовой системе, выполнив команду `hostname`. Выполните её ещё один раз.

Вопрос: одинаковый ли результат получился при разных запусках? Да.

Выполним два раза команду `docker run ubuntu hostname`.

Вопрос: Одинаковый ли результат получился при разных запусках? Нет.

Так происходит, потому что из одного образа ubuntu были запущены два изолированных контейнера, поэтому у них и был разный hostname.

```
version@MSI:~$ sudo hostname
MSI
version@MSI:~$ sudo hostname
MSI
version@MSI:~$ docker run ubuntu hostname
docker: permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": di
al unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
version@MSI:~$ sudo docker run ubuntu hostname
6b3fc54237fd
version@MSI:~$ sudo docker run ubuntu hostname
16e7c45efdd4
version@MSI:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND          CREATED          STATUS          PORTS          NAMES
16e7c45efdd4   ubuntu        "hostname"       25 seconds ago   Exited (0)      25 seconds ago   vigorous_kare
6b3fc54237fd   ubuntu        "hostname"       31 seconds ago   Exited (0)      31 seconds ago   ecstatic_goldberg
a5d907830081   hello-world   "/hello"         15 minutes ago   Exited (0)      15 minutes ago   tender_shamir
version@MSI:~$ sudo docker run ubuntu bash
version@MSI:~$ sudo -it: docker run -it ubuntu bash
root@213f99cbf5fe:/#
root@213f99cbf5fe:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@213f99cbf5fe:/# exit
exit
```

Рисунок 2 – Просмотр hostname на хостовой системе и в контейнерах также
запуск контейнера и bash

2.3 Работа с портами

Запуск встроенной в Python модуль веб-сервера из корня контейнера.

```

version@MSI:~$ sudo docker run -it python python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
^C
Keyboard interrupt received, exiting.
version@MSI:~$ sudo docker run -it -p8000:8000 python python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
^C
Keyboard interrupt received, exiting.
version@MSI:~$ sudo -p hostPort:containerPort
-p: command not found
version@MSI:~$ sudo -p hostPort:containerPort
usage: sudo -h | -K | -k | -V
usage: sudo -v [-ABknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-ABknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
[command]
usage: sudo [-ABbEHknPS] [-r role] [-t type] [-C num] [-D directory] [-g group]
[-h host] [-p prompt] [-R directory] [-T timeout] [-u user]
[VAR=value] [-i|-s] [<command>]
usage: sudo -e [-ABknS] [-r role] [-t type] [-C num] [-D directory] [-g group]
[-h host] [-p prompt] [-R directory] [-T timeout] [-u user] file ...
version@MSI:~$ sudo docker run -it -p8000:8000 python python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
^C
Keyboard interrupt received, exiting.
version@MSI:~$ docker run -it -p8888:8000 python python -m http.server
docker: permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": di
al unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
version@MSI:~$ sudo docker run -it -p8888:8000 python python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
..^C

```

Directory listing for /

- [dockerenv](#)
- [bin/@](#)
- [boot/](#)
- [dev/](#)
- [etc/](#)
- [home/](#)
- [lib/@](#)
- [lib64/@](#)
- [media/](#)
- [mnt/](#)
- [opt/](#)
- [proc/](#)
- [root/](#)
- [run/](#)
- [sbin/@](#)
- [srv/](#)
- [sys/](#)
- [tmp/](#)
- [usr/](#)
- [var/](#)

Рисунок 3 – Представлен запуск сервера на Python

2.4 Именованные контейнеры, остановка и удаление

Запуск контейнера с удалением его после завершения работы командой
docker run --rm -p8000:8000 --name pyserver -d python python -m http.server

```

version@MSI:~$ sudo docker run -it -p8000:8000 python python -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
^C
Keyboard interrupt received, exiting.
version@MSI:~$ sudo docker run -p8000:8000 --name pyserver -d python python -m h
ttp.server
842d00710a22f4db8ab8acfc518a897360a5862b5346d81786ac6499de5f6821
version@MSI:~$ sudo docker ps | grep pyserver
842d00710a22 python "python -m http.serv..." 11 seconds ago Up 10 seconds
0.0.0.0:8000->8000/tcp pyserver
version@MSI:~$ sudo docker run -it -p8000:8000 --name pyserver -d python python
-m http.server
docker: Error response from daemon: Conflict. The container name "/pyserver" is
already in use by container "842d00710a22f4db8ab8acfc518a897360a5862b5346d81786a
c6499de5f6821". You have to remove (or rename) that container to be able to reus
e that name.
See 'docker run --help'.
version@MSI:~$ sudo docker rm -f pyserver
pyserver

```

Рисунок 4 – Представлен запуск сервера

2.5 Постоянное хранение данных

Запустите контейнер, в котором веб-сервер будет отдавать содержимое
 директории /mnt

Остановим контейнер: `docker stop pyserver`, а затем снова запустим: `docker run -p8000:8000 --name pyserver --rm -d python python -m http.server -d /mnt`.

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

`-p8000:8000`: Этот флаг пробрасывает порт 8000 из контейнера на порт 8000 хоста.

`--name pyserver`: Этот флаг задает имя контейнера как "pyserver".

`--rm`: Этот флаг указывает Docker на удаление контейнера после его завершения.

`-d`: Этот флаг запускает контейнер в фоновом режиме (detach mode).

В данной команде, команда, которая будет выполнена внутри контейнера, это `python -m http.server -d /mnt`. Это команда Python для запуска простого веб-сервера, который будет слушать порт 8000 и отображать содержимое директории `/mnt` как корневую директорию веб-сервера.

```
version@MSI:~$ sudo docker run -p8000:8000 --name pyserver --rm -d python python
-m http.server -d /mnt
937e12d543912ca37a0632820480333bd2d67ba5b265303b004416dd6d662066
version@MSI:~$ sudo docker exec -it pyserver bash
root@937e12d54391:/# cd mnt && echo "hello world" > hi.txt
root@937e12d54391:/mnt#
exit

What's next?
Try Docker Debug for seamless, persistent debugging tools in any container or
image → docker debug pyserver
Learn more at https://docs.docker.com/go/debug-cli/
version@MSI:~$ docker stop pyserver
permission denied while trying to connect to the Docker daemon socket at unix://
/var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/containers/p
yserver/stop": dial unix /var/run/docker.sock: connect: permission denied
version@MSI:~$ sudo docker stop pyserver
pyserver
```

Directory listing for /

• [hi.txt](#)

Рисунок 5 – Представлен запуск сервера и добавления файла

Directory listing for /

}

Рисунок 6 – Представлен перезапуск запуск сервера

Создание отдельного тома для хранения файлов

```
version@MSI:~$ sudo docker run -p8000:8000 --rm --name pyserver -d -v $(pwd)/myfiles:/mnt python python -m http.server -d /mnt
81e1f2564342dfb8e817b9228181214e6e42c1df3c7
version@MSI:~$ docker exec -it pyserver bash
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.45/container
s/pyserver/json": dial unix /var/run/docker.sock: connect: permission denied

What's next?
  Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug pyserver
  Learn more at https://docs.docker.com/go/debug-cli/
version@MSI:~$ sudo docker exec -it pyserver bash
root@81e1f2564342:/mnt# && echo "hello world" > hi.txt
root@81e1f2564342:/mnt# ^C
root@81e1f2564342:/mnt#
exit

What's next?
  Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug pyserver
  Learn more at https://docs.docker.com/go/debug-cli/
version@MSI:~$ sudo docker inspect -f "{{json .Mounts }}" pyserver
[[{"Type": "bind", "Source": "/home/version/myfiles", "Destination": "/mnt", "Mode": "", "RW": true, "Propagation": "rprivate"}]]
version@MSI:~$
version@MSI:~$ sudo docker volume create my_volume
my_volume
version@MSI:~$ sudo docker volume inspect my_volume
[
  {
    "CreatedAt": "2024-04-24T19:08:00Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/my_volume/_data",
    "Name": "my_volume",
    "Options": null,
    "Scope": "local"
  }
]
version@MSI:~$ sudo docker volume rm my_volume
my_volume
```

Рисунок 7 – Представлен отдельный том для хранения

Пере монтирование одного файла в другой


```

version@MSI:~$ mkdir myfiles
mkdir: cannot create directory 'myfiles': File exists
version@MSI:~$ sudo touch myfiles/host.txt
version@MSI:~$ docker run -p8000:8000 --rm --name pyserver -d -v $(pwd)/myfiles:/mnt python py
thon -m http.server -d /mnt
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/r
un/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock
: connect: permission denied.
See 'docker run --help'.
version@MSI:~$ sudo docker run -p8000:8000 --rm --name pyserver -d -v $(pwd)/myfiles:/mnt pyth
on python -m http.server -d /mnt
b769e6b1651dfd964c97039dbff59e091afc2fa147bb7e1675bec5bd76ababbe
version@MSI:~$ pwd
/home/version
version@MSI:~$ sudo docker exec -it pyserver bash
root@b769e6b1651d:/# cd /mnt
root@b769e6b1651d:/mnt# ls
hi.txt host.txt
root@b769e6b1651d:/mnt# echo "hello world" > hi.txt
root@b769e6b1651d:/mnt# exit
exit

What's next?
Try Docker Debug for seamless, persistent debugging tools in any container or image → docker
debug pyserver
Learn more at https://docs.docker.com/go/debug-cli/
version@MSI:~$ sudo ls myfiles
hi.txt host.txt
version@MSI:~$ sudo docker stop pyserver
pyserver

```

Рисунок 8 – Пере монтирования одного файла в другой

2.6 Переменные окружения

Вывод всех переменных окружения с помощью утилиты env используя командой `docker run -it --rm -e MIREA="ONE LOVE" ubuntu env`.

```

version@MSI:~$ sudo docker run -it --rm -e MIREA="ONE LOVE" ubuntu env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=fc61a2bfd94f
TERM=xterm
MIREA=ONE LOVE
HOME=/root

```

Рисунок 9 – Представлен вывод всех переменных окружения

2.7 Dockerfile

Сборка образ, в который будут установлены дополнительные пакеты, примонтированы директории и установлены команды запуска. Для этого создается файл Dockerfile (без расширения).

```

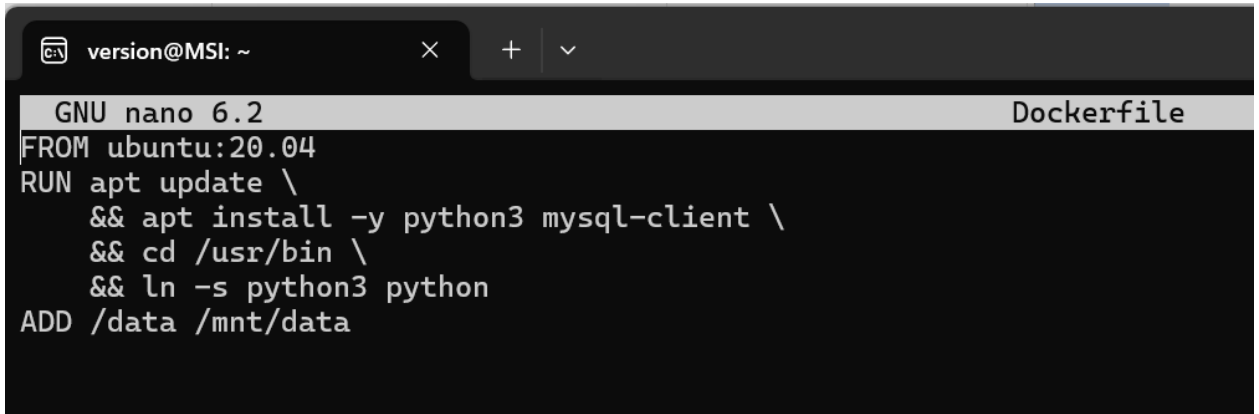
version@MSI:~$ sudo nano Dockerfile
version@MSI:~$ sudo docker build -t mycoolimage .
[+] Building 36.0s (8/8) FINISHED                                docker:default
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 309B                             0.0s
=> [internal] load metadata for docker.io/library/ubuntu:20.04  1.4s
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> CACHED [1/4] FROM docker.io/library/ubuntu:20.04@sha256:71b82b8e734f5cd0b3533a16f40 0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 2B                                    0.0s
=> [2/4] RUN apt update      && apt install -y python3 fortune    && cd /usr/bin  34.5s
=> CACHED [3/4] RUN /usr/games/fortune > /mnt/greeting-while-building.txt 0.0s

```

Рисунок 10 – Представлена установка через Dockerfile

2.8 Индивидуальные задания

Написать Dockerfile, собрать образ, запустить контейнер для 4 варианта установка доп пакета imagemagick.



The screenshot shows a terminal window with a title bar indicating the user is 'version@MSI' in the home directory. The terminal is running 'GNU nano 6.2' and editing a file named 'Dockerfile'. The content of the Dockerfile is as follows:

```

FROM ubuntu:20.04
RUN apt update \
    && apt install -y python3 mysql-client \
    && cd /usr/bin \
    && ln -s python3 python
ADD /data /mnt/data

```

Рисунок 12 – Представлен Dockerfile

```

version@MSI:~$ sudo docker build -t variant .
[+] Building 135.8s (8/8) FINISHED          docker:default
=> [internal] load build definition from Dockerfile      0.0s
=> => transferring dockerfile: 188B                      0.0s
=> [internal] load metadata for docker.io/library/ubuntu  5.3s
=> [internal] load .dockerignore                        0.0s
=> => transferring context: 2B                            0.0s
=> [1/3] FROM docker.io/library/ubuntu:20.04@sha256:874  78.3s
=> => resolve docker.io/library/ubuntu:20.04@sha256:874a  0.0s
=> => sha256:874aca52f79ae5f8258faff03e1 1.13kB / 1.13kB  0.0s
=> => sha256:cc9cc8169c9517ae035cf293b15f069 424B / 424B  0.0s
=> => sha256:2abc4dfd83182546da40dfae3e3 2.29kB / 2.29kB  0.0s
=> => sha256:d4c3c94e5e10ed15503bda7e 27.51MB / 27.51MB  77.3s
=> => extracting sha256:d4c3c94e5e10ed15503bda7e145a3652  0.9s
=> [internal] load build context                        0.0s
=> => transferring context: 106B                          0.0s
=> [2/3] RUN apt update && apt install -y python3 m  51.7s
=> [3/3] ADD /data /mnt/data                          0.0s
=> exporting to image                                  0.5s
=> => exporting layers                                    0.4s
=> => writing image sha256:fa8235168bd48891e007d48a21092  0.0s
=> => naming to docker.io/library/variant                  0.0s

```

What's Next?

View a summary of image vulnerabilities and recommendations →
[docker scout quickview](#)

```

version@MSI:~$ sudo docker run --rm -it -p8824:8000 --name var24
-d variant python -m http.server -d /mnt
3e5fdcecdabd79e73c47b311812d1ac19ebeed0afaf1d4154277a2fa1c7c9aa1
version@MSI:~$ |

```

Рисунок 13 – Представлена установка через Dockerfile

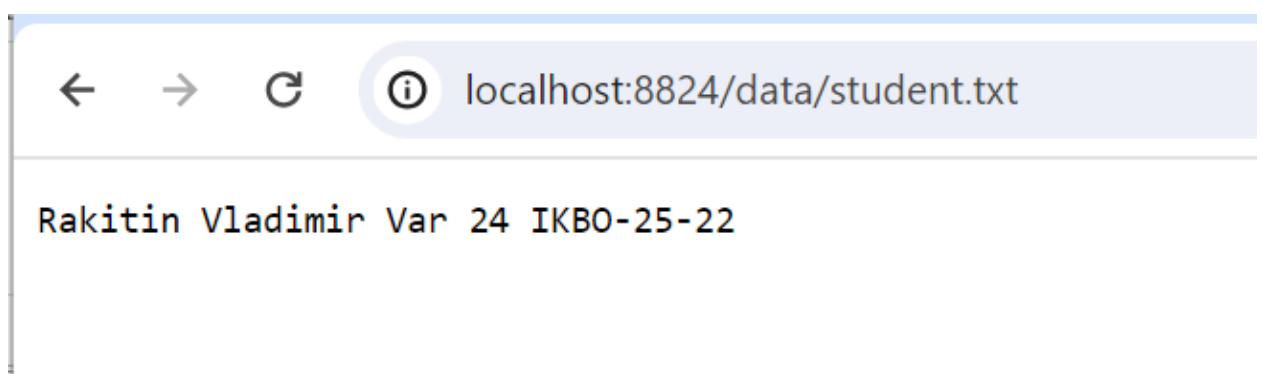


Рисунок 14 – Запуск сервера

3 Выводы о проделанной работе

В ходе выполнения работы по изучению Docker были изучены основные понятия Docker, такие как образы, контейнеры, репозитории и Dockerfile.