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-50-23 Павлов Н.С.

Принял Степанов П.В.

Практическая работа выполнена «__»_____2025г. *(подпись студента)*

«Зачтено» «__»_____2025 г. *(подпись руководителя)*

Москва 2025

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Цель работы: Знакомство с Docker. Возможности docker.

Персональный вариант: 9 (mysql-client)

- необходимо использовать базовый образ ubuntu:20.04
- примонтировать директорию data в директорию /mnt/files/ в контейнере.

2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 ОБРАЗЫ

```
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
[node1] (local) root@192.168.0.18 ~
$ docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
d9802f032d67: Pull complete
Digest: sha256:8e5c4f0285ecbb4ead070431d29b576a530d3166df73ec44affc1cd27555141b
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
[node1] (local) root@192.168.0.18 ~
$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
ubuntu       20.04     6013ae1a63c2   4 months ago   72.8MB
[node1] (local) root@192.168.0.18 ~
$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 1 – Просмотр и добавление образов

2.2 ИЗОЛЯЦИЯ

```
[node1] (local) root@192.168.0.18 ~
$ hostname
node1
[node1] (local) root@192.168.0.18 ~
$ hostname
node1
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu:20.04 hostname
32baa3241d90
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu:20.04 hostname
165c9ddb00a4
[node1] (local) root@192.168.0.18 ~
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND      CREATED              STATUS
              PORTS     NAMES
165c9ddb00a4   ubuntu:20.04   "hostname"   27 seconds ago       Exited (0) 27 seco
nds ago                 gifted_saha
32baa3241d90   ubuntu:20.04   "hostname"   31 seconds ago       Exited (0) 30 seco
nds ago                 festive_williams
54fdaf26283e   ubuntu         "hostname"   About a minute ago   Exited (0) About a
 minute ago             trusting_hopper
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 2 – Запуск и просмотр контейнеров

```
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu:20.04 bash
[node1] (local) root@192.168.0.18 ~
$ docker run -it ubuntu:20.04 bash
root@82b8f7388b41:/# exit
exit
[node1] (local) root@192.168.0.18 ~
$ docker run ubuntu hostname
f861d36c8360
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 3 – Вариации запуска контейнеров

2.3 РАБОТА С ПОРТАМИ

```
[node1] (local) root@192.168.0.18 ~
$ docker pull python
Using default tag: latest
latest: Pulling from library/python
155ad54a8b28: Pull complete
8031108f3cda: Pull complete
1d281e50d3e4: Pull complete
447713e77b4f: Pull complete
21754c21aa78: Pull complete
854e2aed8deb: Pull complete
5e9ad5aa09b4: Pull complete
Digest: sha256:385ccb8304f6330738a6d9e6fa0bd7608e006da7e15bc52b33b0398e1ba4a15b
Status: Downloaded newer image for python:latest
docker.io/library/python:latest
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 4 – Загрузка python в docker

```
[node1] (local) root@192.168.0.18 ~
$ 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.
[node1] (local) root@192.168.0.18 ~
$ 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/) ...
```

Рисунок 5 – Запуск веб сервера

Рисунок 6 – Содержимое запущенного веб сервера

2.4 ИМЕНОВАННЫЕ КОНТЕЙНЕРЫ, ОСТАНОВКА И УДАЛЕНИЕ

```
[node1] (local) root@192.168.0.18 ~
$ docker run -p8000:8000 --name pyserver -d python python -m http.server
f576ca3467ab4650990ddad2960dde083f8a8a7d6bf67e786366968d608d6caf
[node1] (local) root@192.168.0.18 ~
$ docker ps | grep pyserver
f576ca3467ab   python    "python -m http.serv…"   25 seconds ago   Up 24 seconds
 0.0.0.0:8000->8000/tcp   pyserver
[node1] (local) root@192.168.0.18 ~
$ docker logs pyserver
[node1] (local) root@192.168.0.18 ~
$ docker stop pyserver
pyserver
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 7 – Запуск и остановка контейнера

```
[node1] (local) root@192.168.0.18 ~
$ docker rm pyserver
pyserver
[node1] (local) root@192.168.0.18 ~
$ docker run --rm -p8000:8000 --name pyserver -d python python -m http.server
4c4ebf4d748bb99c56d9e838b39ffcbbe0ec90863b78c4eb31686c4cce658613
[node1] (local) root@192.168.0.18 ~
$ docker stop pyserver
pyserver
```

Рисунок 8 – Удаление контейнера

2.5 ПОСТОЯННОЕ ХРАНЕНИЕ ДАННЫХ

```
[node1] (local) root@192.168.0.18 ~
$ docker run -p8000:8000 --name pyserver --rm -d python python -m http.server -d /
mnt
d29ab9bd842d8fa2c83ab42b72bb3d4336b3cd076d196b5bfdb3940877729a93
[node1] (local) root@192.168.0.18 ~
$ docker exec -it pyserver bash
root@d29ab9bd842d:/# cd mnt && echo "hello world" > hi.txt
root@d29ab9bd842d:/mnt# exit
exit
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 9 – Изменения в удаляемом контейнере

Directory listing for /

- hi.txt

Рисунок 10 – Содержимое контейнера

2.5.1 Тома

```
[node1] (local) root@192.168.0.18 ~
$ docker run -p8000:8000 --rm --name pyserver -d \-v $(pwd)/myfiles:/mnt python py
thon -m http.server -d /mnt
7e96ac0e3160a2e7907a87abd2264fa8c90724bc042b6bddf31340d2f7d080a4
[node1] (local) root@192.168.0.18 ~
$ docker exec -it pyserver bash
root@7e96ac0e3160:/# cd mnt && echo "hello world" > hi.txt
root@7e96ac0e3160:/mnt# exit
exit
[node1] (local) root@192.168.0.18 ~
$ docker inspect -f "{{json .Mounts }}" pyserver
[{"Type":"bind","Source":"/root/myfiles","Destination":"/mnt","Mode":"","RW":true,
"Propagation":"rprivate"}]
[node1] (local) root@192.168.0.18 ~
$
```

Рисунок 11 – Контейнер с примонтированным томом

2.5.2 Монтирование директорий и файлов

```
[node1] (local) root@192.168.0.18 ~
$ touch myfiles/host.txt
[node1] (local) root@192.168.0.18 ~
$ docker run -p8000:8000 --rm --name pyserver -d -v $(pwd)/myfiles:/mnt python pyt
hon -m http.server -d /mnt
856e01a3c93d035b5398d95a1e0f9f6f730c62df74395f1428e3fa872dbe616e
```

Рисунок 12 – Запуск докера с примонтированным томом

```
[node1] (local) root@192.168.0.18 ~
$ docker exec -it pyserver bash
root@856e01a3c93d:/# cd /mnt
root@856e01a3c93d:/mnt# ls
hi.txt  host.txt
root@856e01a3c93d:/mnt# echo "hello world" > hi2.txt
root@856e01a3c93d:/mnt# exit
exit
[node1] (local) root@192.168.0.18 ~
$ cd myfiles
[node1] (local) root@192.168.0.18 ~/myfiles
$ ls
hi.txt    hi2.txt   host.txt
[node1] (local) root@192.168.0.18 ~/myfiles
$
```

Рисунок 13 – Проброс файлов в докер и из него

2.6 ПЕРЕМЕННЫЕ ОКРУЖЕНИЯ

```
[node1] (local) root@192.168.0.18 ~/myfiles
$ docker run -it --rm -e MIREA="ONE LOVE" ubuntu env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=7916aaca4b89
TERM=xterm
MIREA=ONE LOVE
HOME=/root
[node1] (local) root@192.168.0.18 ~/myfiles
$
```

Рисунок 14 – Передача переменной окружения внутрь контейнера

2.7 DOCKERFILE

```
1 FROM ubuntu:20.04
2 RUN apt update && apt install -y python3 fortune && cd /usr/bin && ln -s pytho
  n3 python
3 RUN /usr/games/fortune > /mnt/greeting-while-building.txt
4 ADD ./data /mnt/data
5 EXPOSE 80
6 CMD ["python", "-m", "http.server", "-d", "/mnt/", "80"]
```

Рисунок 15 – Содержимое Dockerfile

```
[node1] (local) root@192.168.0.18 ~/myfiles
$ docker build -t mycoolimage .
[+] Building 14.3s (9/9) FINISHED                                docker:default
 => [internal] load build definition from Dockerfile                      0.0s
 => => transferring dockerfile: 291B                                      0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04           0.0s
 => [internal] load .dockerignore                                         0.0s
 => => transferring context: 2B                                           0.0s
 => CACHED [1/4] FROM docker.io/library/ubuntu:20.04                      0.0s
 => [internal] load build context                                         0.0s
 => => transferring context: 26B                                          0.0s
 => [2/4] RUN apt update && apt install -y python3 fortune && cd /usr/bin  12.8s
 => [3/4] RUN /usr/games/fortune > /mnt/greeting-while-building.txt       0.6s
 => [4/4] ADD ./data /mnt/data                                            0.0s
 => exporting to image                                                    0.8s
 => => exporting layers                                                   0.8s
 => => writing image sha256:c488f81db1648bccd4a5ac92415c47ef9efd2d71c40ae6c  0.0s
 => => naming to docker.io/library/mycoolimage                            0.0s
[node1] (local) root@192.168.0.18 ~/myfiles
$ docker run --rm -it -p8099:80 mycoolimage
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Рисунок 16 – Сборка образа и запуск

2.8 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

```
[node1] (local) root@192.168.0.18 ~/personal
$ mkdir data
[node1] (local) root@192.168.0.18 ~/personal
$ echo "Pavlov Nikita Sergeevich IKBO-50-23. Variate: 9" > data/student.txt
[node1] (local) root@192.168.0.18 ~/personal
$
```

Рисунок 17 – Подготовка файла и директории для проброса в контейнер

```
1 FROM ubuntu:20.04
2 RUN apt-get update
3 RUN apt-get install -y python3 mysql-client
4 RUN ln -s /usr/bin/python3 /usr/bin/python
5 ADD ./data /mnt/files/data
6 EXPOSE 24
7 CMD ["python", "-m", "http.server", "-d", "/mnt/files/", "24"]
```

Рисунок 18 – Содержимое Dockerfile

```
[node1] (local) root@192.168.0.18 ~/personal
$ docker build -t mypersonalvariate .
[+] Building 9.4s (10/10) FINISHED                                                                                  docker:default
 => [internal] load build definition from Dockerfile                                                                         0.0s
 => => transferring dockerfile: 263B                                                                                         0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                                              0.0s
 => [internal] load .dockerignore                                                                                            0.0s
 => => transferring context: 2B                                                                                              0.0s
 => [1/5] FROM docker.io/library/ubuntu:20.04                                                                                0.0s
 => [internal] load build context                                                                                            0.0s
 => => transferring context: 61B                                                                                             0.0s
 => CACHED [2/5] RUN apt-get update                                                                                          0.0s
 => [3/5] RUN apt-get install -y python3 mysql-client                                                                        7.8s
 => [4/5] RUN ln -s /usr/bin/python3 /usr/bin/python                                                                         0.6s
 => [5/5] ADD ./data /mnt/files/data                                                                                         0.0s
 => exporting to image                                                                                                       0.9s
 => => exporting layers                                                                                                      0.9s
 => => writing image sha256:68d0d162379e47325b5e2b5134cd9c9715b565ac34287b3071284345f1fcba44                                 0.0s
 => => naming to docker.io/library/mypersonalvariate                                                                         0.0s
[node1] (local) root@192.168.0.18 ~/personal
$ docker run --rm -it -p8824:24 mypersonalvariate
Serving HTTP on 0.0.0.0 port 24 (http://0.0.0.0:24/) ...
```

Рисунок 19 – Сборка образа и запуск контейнера

Directory listing for /

- data/

Рисунок 20 – Содержимое веб сервера

```
Pavlov Nikita Sergeevich IKBO-50-23. Variate: 9
```

Рисунок 21 – Проверка содержимого файла на сервере

3 ВЫВОДЫ

В ходе выполнения практической работы были получены навыки работы с Docker. Изучены функционал и возможности docker.