

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №17
дисциплины «Основы программной инженерии»

Выполнил:
Матвеев Александр Иванович
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Работа в docker с сетью контейнеров и томами.

Цель работы: познакомить студентов с использованием Docker для управления томами и сетями.

Ход работы.

1. Задача: Создайте пользовательскую сеть в Docker с именем "my_custom_network". Запустите два контейнера, присоединенных к этой сети, например, с использованием образов Nginx и PostgreSQL. Убедитесь, что они могут взаимодействовать друг с другом.

```
d:\Docker_test\laba2>docker network create my_custom_network
9aa773e4997838169b9237cbdb38956406d542645d515df31eae0683abfb9102
```

Рисунок 1 – Создание кастомной сети

```
d:\Docker_test\laba2>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e0eec57198fc	postgres	"docker-entrypoint.s..."	7 seconds ago	Up 6 seconds	5432/tcp	postgres_container
d9f08f6d57e9	nginx	"/docker-entrypoint..."	2 minutes ago	Up 2 minutes	80/tcp	sad_carson

Рисунок 2 – Создание двух контейнеров, использующих созданную сеть

```
d:\Docker_test\laba2>docker inspect -f '{{.NetworkSettings.Networks}}' sad_carson
'map[my_custom_network:0xc0004d80c0]'
```

```
d:\Docker_test\laba2>docker inspect -f '{{.NetworkSettings.Networks}}' postgres_container
'map[my_custom_network:0xc000000240]'
```

Рисунок 3 – Оба контейнера работают в одной сети

2. Задача: Создайте Docker-контейнер с использованием тома. Запишите данные в том из одного контейнера, а затем прочитайте их из другого контейнера, используя тот же том. Обеспечьте, чтобы данные сохранялись после перезапуска контейнеров.

```
d:\Docker_test\laba2>docker run -d -v tom:/data --name first alpine
d4527a8967cda7360e474457d63006e321c48e6fc3239626a30c7069241f5143

d:\Docker_test\laba2>docker run -d -v tom:/data --name first nginx
docker: Error response from daemon: Conflict. The container name "/firs
360e474457d63006e321c48e6fc3239626a30c7069241f5143". You have to remove
hat name.
See 'docker run --help'.

d:\Docker_test\laba2>docker run -d -v tom:/data --name second nginx
ee3f5884cb6799cfe44ea4f067a7ac831016d75fd114ebb1a9c4b2f1b1a2e183
```

Рисунок 4 – Создание двух контейнеров с одним томом

```
d:\Docker_test\laba2>docker exec -it first /bin/bash
root@52b6e5dd7a1b:/# cd /data
root@52b6e5dd7a1b:/data# echo "Hello world!" > test.txt
```

Рисунок 5 – Запись данных

```
d:\Docker_test\laba2>docker exec -it second /bin/bash
root@ee3f5884cb67:/# cd /data
root@ee3f5884cb67:/data# ls
test.txt
root@ee3f5884cb67:/data# cat test.txt
Hello world!
```

Рисунок 6 – Чтение данных

3. Задача: Запустите два контейнера и присвойте им IP-адреса из одной пользовательской сети.

```
d:\Docker_test\laba2>docker run -d -it --name first --network my_custom_network --ip 172.18.0.2 nginx
f8709a80bbfcdcf19043b2276dd2623c27779424545e714670777c4018c7d8571
```

Рисунок 7 – Запуск первого контейнера

```
d:\Docker_test\laba2>docker run -d -it --name second --network my_custom_network --ip 172.18.0.3 ubuntu
4181c1da10d10a47976db9e65ce2791752997c9e1f18f09752ecf6cc08d6c616
```

Рисунок 8 – Запуск второго контейнера

4. Задача: Используя устаревшую опцию `--link`, создайте два контейнера (например, с Nginx и MySQL) и свяжите их между собой. Убедитесь, что контейнер с Nginx может успешно обращаться к контейнеру с MySQL через имя контейнера, указанное при использовании опции `--link`.

```
d:\Docker_test\laba2>docker run -d -e POSTGRES_PASSWORD=678432 --name postgres_container postgres
b8b373e46764e67d1cfeb050a0d9d74e9999b53bb562e092e09d3de7fce525a8f
```

Рисунок 9 – Запуск первого контейнера

```
d:\Docker_test\laba2>docker run -d --name nginx_container --link postgres_container:postgres nginx
d45ba10fc74b9eb3680e7b19f90526de69c05b0c7e04a117ec41756f77414e2c
```

Рисунок 10 – Запуск второго, соединение с первым

5. Задача: Используя устаревшую опцию `--link`, создайте два контейнера (например с Nginx и MySQL) и свяжите их между собой. Убедитесь, что контейнер с Nginx может успешно обращаться к контейнеру с MySQL через имя контейнера, указанное при использовании опции `--link`.

```
→ docker git:(main) * docker run --name mysql-container -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:l
atest

Unable to find image 'mysql:latest' locally
^[[Alatest: Pulling from library/mysql
f065eb68ef2e: Pull complete
4347407baabc: Pull complete
3ac2468bda24: Pull complete
83f773ef2c87: Pull complete
1dad2c7f0c6a: Pull complete
808b6b0a452f: Pull complete
1938c61da20f: Pull complete
bd13b3b553e4: Pull complete
a00ae17e153b: Pull complete
29255c8a5cf0: Pull complete
Digest: sha256:4ef30b2c11a3366d7bb9ad95c70c0782ae435df52d046553ed931621ea36ffa5
Status: Downloaded newer image for mysql:latest
2bc7ba03ec7f4e61269f52c5de734b5b9ad28f417e9422f587b3864ddbaaf2073
```

Рисунок 11 – Запуск контейнера с mysql

```
➔ docker git:(main) * docker run --name nginx-container --link mysql-container:mysql -d nginx:latest

Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
24e221e92a36: Already exists
58cc89079bd7: Pull complete
3799b53049f3: Pull complete
2a580edba2f4: Pull complete
cfe7877ea167: Pull complete
6f26751fc54b: Pull complete
c98494bb3682: Pull complete
Digest: sha256:2bdc49f2f8ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026
Status: Downloaded newer image for nginx:latest
88aaa20ac2549dc8d0b24f53288979b983de6b7978f270632f63bc64bb570564
```

Рисунок 12 – Запуск nginx с использованием --link

Контрольные вопросы:

1. Чтобы создать новый том в Docker, используйте команду ``docker volume create``. Например, ``docker volume create my_volume``.
2. Чтобы удалить существующий том в Docker, используйте команду ``docker volume rm``. Например, ``docker volume rm my_volume``.
3. Чтобы просмотреть список всех созданных томов в Docker, используйте команду ``docker volume ls``.
4. Чтобы создать том с определенным именем, просто укажите имя после команды ``docker volume create``. Например, ``docker volume create my_custom_volume``.
5. Чтобы присоединить том к контейнеру при его запуске, используйте флаг ``-v`` или ``--volume`` с командой ``docker run``. Например, ``docker run -v my_volume:/path/in/container image_name``.
6. Чтобы просмотреть подробную информацию о конкретном томе в Docker, используйте команду ``docker volume inspect``. Например, ``docker volume inspect my_volume``.
7. Чтобы создать новую сеть в Docker, используйте команду ``docker network create``. Например, ``docker network create my_network``.
8. Чтобы удалить существующую сеть в Docker, используйте команду ``docker network rm``. Например, ``docker network rm my_network``.
9. Чтобы просмотреть список всех созданных сетей в Docker, используйте команду ``docker network ls``.
10. Чтобы создать пользовательскую сеть с определенным именем, используйте команду ``docker network create``. Например, ``docker network create my_custom_network``.
11. Чтобы присоединить контейнер к пользовательской сети при его запуске, используйте флаг ``--network`` с командой ``docker run``. Например, ``docker run --network=my_custom_network image_name``.

12. Чтобы просмотреть подробную информацию о конкретной сети в Docker, используйте команду ``docker network inspect``. Например, ``docker network inspect my_custom_network``.

13. Чтобы указать определенную сеть при запуске контейнера с использованием ``docker run``, используйте флаг ``--network``. Например, ``docker run --network=my_custom_network image_name``.

14. Если не указана конкретная сеть, контейнер будет подключен к сети ``bridge`` по умолчанию.

15. Чтобы присоединить контейнер к нескольким сетям сразу при его запуске, вы должны сначала создать контейнер, а затем подключить его к каждой сети с помощью команды ``docker network connect``. Например, ``docker network connect network1 container_name && docker network connect network2 container_name``.

16. Чтобы просмотреть список сетей, доступных на хосте Docker, используйте команду ``docker network ls``.

17. Чтобы создать контейнер, подключенный к сети "bridge", используйте флаг ``--network=bridge`` с командой ``docker run``. Например, ``docker run --network=bridge image_name``.

18. Чтобы создать контейнер, подключенный к сети "host", используйте флаг ``--network=host`` с командой ``docker run``. Например, ``docker run --network=host image_name``.