

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Кафедра инфокоммуникаций
Институт цифрового развития**

ОТЧЁТ

по лабораторной работе №2

Дисциплина: «Конфигурационное распределенное управление ПО»

Тема: «Работа в Docker с сетью контейнеров и томами»

Вариант 5

Выполнила: студентка 3 курса,
группы ИВТ-б-о-21-1
Диченко Дина Алексеевна

Ставрополь 2023

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

Практическая часть

1 Создание пользовательской сети:

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.

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker network create my_custom_network
27b1de69ba2668794942e7e521c796e2535b13b0e56d98fa54bdd284c42cb121
```

Рисунок 1. Создание сети

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run --network=my_custom_network -d nginx
1b3c44ba4dfd53c382eef85431373cbe962a0621d6a3b3286381a1203236082f

super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run --network=my_custom_network -d postgres
b1efede39bfba3a4e22f53e74d9e35d6ba12a372cc388df76e1180ccee1e4c63
```

Рисунок 2. Запуск контейнеров

```
$ docker network inspect my_custom_network
[
  {
    "Name": "my_custom_network",
    "Id": "e530d685ed4d479a7ba05eb09cbcd626d1c925be16a5e88dd6ed59c5106227794",
    "Created": "2023-11-28T21:07:25.028804027Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "8675f7e880be3f3d4e5fb3f6e89ae740cc7a6049eea19d5540aca638d961d843": {
        "Name": "con_post",
        "EndpointID": "55c038a198297e160459c2593f1eccbaa4970d8e5381cea760c4521999cb9303",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      },
      "fd62f274d85424a10de55d0a865d146ab1265aa50c7eb4ca37600ccda16f6f5": {
        "Name": "con_nginx",
        "EndpointID": "b88214bec86f8a373b87d4bf0ad49f4f18ef969ddbc50c1813d43c1e5b5e3c7",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    }
  }
]
```

Рисунок 3. Проверка наличия контейнеров в сети

2 Передача данных через тома:

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.

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker volume create tom
tom

super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run -d -v tom:/data --name container nginx
86fe2bc829b707e8c787f52897ce98bcb9fedafd4deaed7a4ef211759394193a
```

Рисунок 4. Создание тома

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker exec -it container bash
root@86fe2bc829b7:/# echo "some data" > /data/data_file.txt
root@86fe2bc829b7:/# exit
exit

super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run -d -v tom:/data --name container2 nginx
63d9e570c2f21b4ed52abe6dccbe252e1ccb740c6871e24fe6fff6b1b53a9738

super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker exec -it container2 bash
root@63d9e570c2f2:/# cat /data/data_file.txt
some data
root@63d9e570c2f2:/#
```

Рисунок 5. Передача данных через том

3 Создание сети overlay для распределенного приложения:

Задача: Используйте Docker Swarm или Kubernetes (в зависимости от предпочтений) для создания кластера. Создайте overlay-сеть и запустите несколько контейнеров, которые могут взаимодействовать через эту сеть.

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker swarm init --advertise-addr 10.0.2.15
Swarm initialized: current node (px76o4h0mujgyf2vq0p1w1toj) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-6562dpykbhit1yzomhvw30yv3ns4hxxvhigbqc69s449w6cjni-e
v7w1ry0xaldshceesfadp6a1 10.0.2.15:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instru
ctions.
```

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker network create -d overlay --attachable overlay_network
vwz1678iitxxzs1fxk73o2gw
```

Рисунок 6. Создание overlay-сети

```
super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run --network=overlay_network -d --name con_ov1 nginx
724ad2e0242253ae6a8e17afa3787ab62585b990f4cac427630bd972445abb93

super@DESKTOP-1B0Q56B MINGW64 /c/Program Files/Docker Toolbox
$ docker run --network=overlay_network -d --name con_ov2 nginx
a941a8ad26682891c457776a5cd6267d1232f386e9c87e1b29b762e55e3fd2bc
```

Рисунок 7. Запуск контейнеров в сети

4 Связь контейнеров по IP-адресу:

Задача: запустите два контейнера и присвойте им IP-адреса из одной пользовательской сети. Обеспечьте взаимодействие между контейнерами по их IP-адресам.

```
root@DESKTOP-1B0Q56B:~# docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' container_1
172.18.0.2
root@DESKTOP-1B0Q56B:~# docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' container_2
172.18.0.3
root@DESKTOP-1B0Q56B:~# docker exec -it container_1 sh
/data # ping 172.18.0.3
PING 172.18.0.3 (172.18.0.3): 56 data bytes
64 bytes from 172.18.0.3: seq=0 ttl=64 time=1.071 ms
64 bytes from 172.18.0.3: seq=1 ttl=64 time=0.186 ms
64 bytes from 172.18.0.3: seq=2 ttl=64 time=0.133 ms
64 bytes from 172.18.0.3: seq=3 ttl=64 time=0.089 ms
64 bytes from 172.18.0.3: seq=4 ttl=64 time=0.120 ms
64 bytes from 172.18.0.3: seq=5 ttl=64 time=0.102 ms
^C
--- 172.18.0.3 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 0.089/0.283/1.071 ms
/data #
```

Рисунок 8. Взаимодействие между контейнерами по IP-адресам

5 Использование ссылок для связи контейнеров:

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`.

```
root@DESKTOP-1B0Q56B:~# docker run --name c_mysql -e MYSQL_ROOT_PASSWORD=my -d mysql
2073d4443d60fd9ebba1edbf0b21f97ed14e1a60da6d9aa19434cde478ca4bb
root@DESKTOP-1B0Q56B:~# docker run -d --name c_nginx --link c_mysql:db -p 8080:80 nginx
6ba66f77167833aa73f5f23fdd4207423112cb46a84f64ad9092774ad8ea8b5
root@DESKTOP-1B0Q56B:~#
```

Рисунок 9. Взаимодействие контейнеров с помощью `--link`

Индивидуальное задание. Создать контейнер с образом redis, таблицу в базе данных и несколько записей в ней. Сделать возможным доступ к созданной таблице в новом контейнере, а также выполнить к ней запрос после удаления первого контейнера и образа redis.

```
root@DESKTOP-1B0Q56B:~# docker run --name my-redis-container -v my-redis-data:/data --network redis-network -d redis:alpine
68ffe66316b7477c9a03faf52016387c804c871d46513d9ab1fbe8fd511ed3f3
root@DESKTOP-1B0Q56B:~# docker exec -it my-redis-container redis-cli
127.0.0.1:6379> SET mytable:key1 value1
OK
127.0.0.1:6379> GET mytable:key1
"value1"
127.0.0.1:6379> exit
root@DESKTOP-1B0Q56B:~# docker run --name new-redis-container -v my-redis-data:/data --network redis-network -d redis
Unable to find image 'redis:latest' locally
^Croot@DESKTOP-1B0Q56B:~# docker run --name new-redis-container -v my-redis-data:/data --network redis-network -d redis:alpine
50dddabe70c09b3c4bf84a9c67a22b63858c4f2b2223fa785138e10f1fd30403
root@DESKTOP-1B0Q56B:~# docker exec -it new-redis-container redis-cli
127.0.0.1:6379> GET mytable:key1
"value1"
127.0.0.1:6379> exit
root@DESKTOP-1B0Q56B:~# docker stop my-redis-container
my-redis-container
root@DESKTOP-1B0Q56B:~# docker rm my-redis-container
my-redis-container
root@DESKTOP-1B0Q56B:~# docker exec -it new-redis-container redis-cli
127.0.0.1:6379> GET mytable:key1
"value1"
127.0.0.1:6379> _
```

Рисунок 10. Индивидуальное задание

Контрольные вопросы:

1 Как создать новый том в Docker?

docker volume create

2 Как удалить существующий том в Docker?

docker volume rm .

3 Как просмотреть список всех созданных томов в Docker?

docker volume ls

4 Как создать том с определенным именем?

Том можно создать с помощью команды docker volume create и указанием имени тома.

5 Как присоединить том к контейнеру при его запуске?

С помощью опции -v

6 Как просмотреть подробную информацию о конкретном томе в Docker?

docker volume inspect с именем или идентификатором тома

7 Как создать новую сеть в Docker?

`docker network create`

8 Как удалить существующую сеть в Docker?

`docker network rm`

9 Как просмотреть список всех созданных сетей в Docker?

`docker network ls`

10 Как создать пользовательскую сеть с определенным именем?

`docker network create имя_сети`

11 Как присоединить контейнер к пользовательской сети при его запуске?

После создания сети контейнер можно запустить, указав параметр `--network` и имя созданной сети.

12 Как просмотреть подробную информацию о конкретной сети в Docker?

13 Как указать определенную сеть при запуске контейнера с использованием `docker run` ?

`docker run --network=my_custom_network -d nginx`

14 Какие сети будут доступны по умолчанию для контейнера, если не указана конкретная сеть?

- `bridge`, сеть, к которой подключаются контейнеры, если не указано иначе.
- `none`, сетевой стек без наличия сетевого интерфейса в контейнере.
- `host`, подключение к сетевому стеку сервера.

15 Как присоединить контейнер к нескольким сетям сразу при его запуске?

`docker run --network=my_custom_network -d nginx`

16 Как просмотреть список сетей, доступных на хосте Docker?

`docker network ls`

17 Как создать контейнер, подключенный к сети "bridge"?

`docker run --network=bridge -d nginx`

18 Как создать контейнер, подключенный к сети "host"?

```
docker run --network=host -d nginx
```

Вывод: в результате выполнения работы мы познакомились с использованием Docker для управления томами и сетями.