

РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
«Работа в Docker с сетью контейнеров и томами»

Отчет по лабораторной работе
по дисциплине «Анализ данных»

Выполнил студент группы ИВТ-б-о-21-1

Шайдеров Дмитрий Викторович.

«20» ноября 2023г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: научиться использовать Docker для управления томами и сетями.

Порядок выполнения работы:

Задача 1: Создание пользовательской сети:

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.

```
user@Shayderov:~$ docker network create my_custom_network
a239f442259725433af3cadcb7fb5063d525dd5b161b5c071b64343815adf8c8
user@Shayderov:~$ docker run --network=my_custom_network -d --name web_container nginx
88b30666264a474ad6b8cbcf346ebffa84368e444e33548d1bfbf8523c559747
```

Рисунок 1 – Создание сети

```
user@Shayderov:~$ docker run --network=my_custom_network -d --name db_container -e POSTGRES_PASSWORD=123
postgres
685e232ede56eb7124c1f9919af77e120d6f3769001cfddedbbf1227d97d0de8c
user@Shayderov:~$ docker network inspect my_custom_network
[
  {
    "Name": "my_custom_network",
    "Id": "a239f442259725433af3cadcb7fb5063d525dd5b161b5c071b64343815adf8c8",
    "Created": "2023-11-25T07:27:51.698716321Z",
    "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": {
      "685e232ede56eb7124c1f9919af77e120d6f3769001cfddedbbf1227d97d0de8c": {
        "Name": "db_container",
        "EndpointID": "70b2e857dcd59a4a9f5cba68d47799b2a14ff3a4feac193639c427d650c27b14",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      },
      "88b30666264a474ad6b8cbcf346ebffa84368e444e33548d1bfbf8523c559747": {
        "Name": "web_container",
        "EndpointID": "fac2e9c4ba53bfa458e5ceddbb396e20ba8cc007242e255c943de476294ae8de",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    }
  }
]
```

Рисунок 2 - Проверка наличия контейнеров в сети

Задача 2: Передача данных через тома:

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.

```
user@Shayderov:~$ docker volume create shared_data
shared_data
user@Shayderov:~$ docker run -d -v shared_data:/data --name container1 nginx
753aa441a8b6d034d4195df58a23d5b82d05eab46276e941f6f643621739cb39
user@Shayderov:~$ docker exec -it container1 bash
root@753aa441a8b6:/# echo "Hello from container1" > /data/data_file.txt
root@753aa441a8b6:/# exit
exit
user@Shayderov:~$ docker run -d -v shared_data:/data --name container2 nginx
1a8ccc8bd091c40e1d2931757444e4a1f8c860a39ec4ff224e4a8b61ee53dfe0
user@Shayderov:~$ docker exec -it container2 bash
root@1a8ccc8bd091:/# cat /data/data_file.txt
Hello from container1
root@1a8ccc8bd091:/#
```

Рисунок 3 – Передача данных через том

Задача 3: Создание сети overlay для распределенного приложения:

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.

```
user@Shayderov:~$ docker swarm init
Swarm initialized: current node (pgfb5qdk9846c2do3myf9k7xh) is now a manager.

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

    docker swarm join --token SWMTKN-1-02cwbfl14t6ukn0rrh7ct8puitmes2h7hdjrl28v2e3j6wc5ef-9o4bj8z4oq2hi3v8bvmtdkq5w 192.168.65.3:2377

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

user@Shayderov:~$ docker network create -d overlay --attachable my_overlay_network
z32y83ylwf22nyy1bz30gfh4n
user@Shayderov:~$ docker run --network=my_overlay_network -d --name sw1 nginx
docker: Error response from daemon: Conflict. The container name "/sw1" is already in use by container "f6f651b2d60d1a1e08ec66c2f1a827ee427025f5809b984a5c5872f89b4f9ce4". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
user@Shayderov:~$ docker run --network=my_overlay_network -d --name swarm1 nginx
b1e7e4f2ea3f948c547a4965ffcbc269144c83202539de52ff40234713e8b210
user@Shayderov:~$ docker run --network=my_overlay_network -d --name swarm2 nginx
c473e8e36f09493332516de207a44324a9fa56192a70bf58f427979572ba4473
```

Рисунок 4 – Создание сети overlay

Задача 4: Связь контейнеров по IP-адресу:

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

```

user@Shayderov:~$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' web_container
172.18.0.2
user@Shayderov:~$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' db_container
172.18.0.3
user@Shayderov:~$ docker exec -it web_container bash
root@88b30666264a:/# ping 172.18.0.3
PING 172.18.0.3 (172.18.0.3) 56(84) bytes of data:
64 bytes from 172.18.0.3: icmp_seq=1 ttl=64 time=0.154 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.089 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=0.084 ms
64 bytes from 172.18.0.3: icmp_seq=4 ttl=64 time=0.086 ms
^C
--- 172.18.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3155ms
rtt min/avg/max/mdev = 0.084/0.103/0.154/0.029 ms
root@88b30666264a:/#

```

Рисунок 5 - Связь контейнеров по IP-адресу

Задача 5: Использование ссылок для связи контейнеров:

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`.

```

user@Shayderov:~$ docker run --name mysql_container -e MYSQL_ROOT_PASSWORD=my -d mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
8e0176adc18c: Pull complete
2d2c52718f65: Pull complete
d88d03ce139b: Pull complete
4a7d7f11aa1e: Pull complete
ce5949193e4c: Pull complete
f7f024dfb329: Pull complete
5fc3c840facc: Pull complete
509068e49488: Pull complete
cbc847bab598: Pull complete
942bef62a146: Pull complete
Digest: sha256:1773f3c7aa9522f0014d0ad2bbdaf597ea3b1643c64c8ccc2123c64afd8b82b1
Status: Downloaded newer image for mysql:latest
bab4b8c1d4c247d51e9573d78eb9b23b0853d62eb229460eafa8418a1f062d52
user@Shayderov:~$ docker run -d --name ng_container --link mysql_container:db -p 8080:80 nginx
6adbef03b56244cd6c1dfbd2b52b37a77814ba75c11916721e27e66de978ff11
user@Shayderov:~$

```

Рисунок 5 – Связь контейнеров с помощи ссылки

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?

`docker volume create my_volume`

5. Как присоединить том к контейнеру при его запуске?

`docker run -v /путь/на/хосте:/путь/в/контейнере -d image_name`

6. Как просмотреть подробную информацию о конкретном томе в Docker?

`docker volume inspect my_volume`

7. Как создать новую сеть в Docker?

`docker network create my_custom_network`

8. Как удалить существующую сеть в Docker?

`docker network rm my_custom_network`

9. Как просмотреть список всех созданных сетей в Docker?

`docker network ls`

10. Как создать пользовательскую сеть с определенным именем?

`docker network create my_custom_network`

11. Как присоединить контейнер к пользовательской сети при его запуске?

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

12. Как просмотреть подробную информацию о конкретной сети в Docker?

`docker network inspect my_network`

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, host и none.

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: были изучены способы использования Docker для управления томами и сетями.