

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 № 4
«Работа в Docker с сетью контейнеров и томами»

Выполнила:
Кузнецова Алена Валерьевна
3 курс, группа ИВТ-б-о-21-1,
09.03.01 «Информатика
и вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»,
очная форма обучения

(подпись)

Ставрополь, 2023 г.

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

Выполнение работы:

1. Создание пользовательской сети:

Создайте пользовательскую сеть в Docker с именем "my_custom_network".

```
alena@LAPTOP-ARKAM4ED:~$ docker network create my_custom_network
fef897fc5366672148f400529169a8ab9cabebf754e037f52c4ff279a8f458cc
```

Рисунок 1 – Создание сети

Запустите два контейнера, присоединенных к этой сети, например, с использованием образов Nginx и PostgreSQL

```
alena@LAPTOP-ARKAM4ED:~$ docker run --network=my_custom_network -d --name con_ng nginx
5d35cd71ab5803ce56d7f25e2956283c599c0539a4c2d2c022615ef37be9981f

alena@LAPTOP-ARKAM4ED:~$ docker run --name con_psql -e POSTGRES_PASSWORD=con --network=my_custom_network -d postgres:13.3
1d5812af436367c24cb34a3e583c8eb6f5ed1b7935a5a20e3fd98770af70b35b
```

Рисунок 2 – Запуск контейнеров

```
alena@LAPTOP-ARKAM4ED:~$ docker network inspect my_custom_network
[
  {
    "Name": "my_custom_network",
    "Id": "fef897fc5366672148f400529169a8ab9cabebf754e037f52c4ff279a8f458cc",
    "Created": "2023-11-22T14:48:31.900595453Z",
    "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": {
      "1d5812af436367c24cb34a3e583c8eb6f5ed1b7935a5a20e3fd98770af70b35b": {
        "Name": "con_psql",
        "EndpointID": "7444aab3cdda4cb1a0b8ee6cc276db4328633321c7a9cca680843c2cb3cbbece",
        "MacAddress": "02:42:ac:12:00:03",
        "IPv4Address": "172.18.0.3/16",
        "IPv6Address": ""
      },
      "556b5c6e6cbef7b18d2bcb2450e77d16e28c4f5371f5dc10b445e09de9a7e3f0": {
        "Name": "con_ng",
        "EndpointID": "1cb15b9b8d25d77ee57c55e0902b4057e498a93583ef97019209cfd618c179db",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      }
    }
  }
]
```

Рисунок 3 – Оба контейнера в одной сети

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.

```
alena@LAPTOP-ARKAM4ED:~$ docker volume create my_volume
my_volume
```

Рисунок 4 – Создание тома

```
alena@LAPTOP-ARKAM4ED:~$ docker run -d -v my_volume:/data --name container1 nginx
7538030ce804d6015379b49fac05b1eabc655631816554c5fe0937a62ee1f146
alena@LAPTOP-ARKAM4ED:~$ docker run -d -v my_volume:/data --name container2 nginx
f1bb58997d42e6f75eb7e47c42072bbbb7297a45ca69d39005315681f967037a
```

Рисунок 5 – Создание контейнеров

```
alena@LAPTOP-ARKAM4ED:~$ docker exec -it container1 bash
root@7538030ce804:/# echo "lb4 docker" > /data/file.txt
root@7538030ce804:/# exit
exit
alena@LAPTOP-ARKAM4ED:~$ docker exec -it container2 bash
root@f1bb58997d42:/# cat /data/file.txt
lb4 docker
root@f1bb58997d42:/# exit
exit
alena@LAPTOP-ARKAM4ED:~$
```

Рисунок 6 – Запись данных и чтение

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.

```
alena@LAPTOP-ARKAM4ED:~$ docker swarm init
Swarm initialized: current node (t1wtdh9qt5m4wsk1304oxqkn) is now a manager.

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

    docker swarm join --token SWMTKN-1-1091w96u8qlet6j9bvtzn pazlim4prb3yq1mr11hv0r2k9r-20dpyw5ey4ppa
kxyc8bql8l 192.168.65.3:2377

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

Рисунок 7 – Инициализация Swarm

```
alena@LAPTOP-ARKAM4ED:~$ docker network create -d overlay my_overlay_network
vxnymlaimdhlyf9a3i9tkv01i
```

Рисунок 8 – Создание Overlay-сети

```
alena@LAPTOP-ARKAM4ED:~$ docker run --network=my_overlay_network -d --name con_over1 nginx
b8b2c1736d564e9b6307201585dda61b793f0a61ad8bb6c3a4ca2b8439aff97b
```

Рисунок 9 – Запуск контейнеров

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.

```
alena@LAPTOP-ARKAM4ED:~$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' con_ng
172.18.0.2
alena@LAPTOP-ARKAM4ED:~$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' con_psql
172.18.0.3
```

Рисунок 10 – IP-адреса контейнеров

```
alena@LAPTOP-ARKAM4ED:~$ docker exec -it con_psql bash
root@1d5812af4363:/# ping 172.18.0.2
PING 172.18.0.2 (172.18.0.2) 56(84) bytes of data.
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=0.185 ms
64 bytes from 172.18.0.2: icmp_seq=2 ttl=64 time=0.126 ms
64 bytes from 172.18.0.2: icmp_seq=3 ttl=64 time=0.107 ms
64 bytes from 172.18.0.2: icmp_seq=4 ttl=64 time=0.139 ms
64 bytes from 172.18.0.2: icmp_seq=5 ttl=64 time=0.077 ms
64 bytes from 172.18.0.2: icmp_seq=6 ttl=64 time=0.084 ms
^C
--- 172.18.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 185ms
rtt min/avg/max/mdev = 0.077/0.119/0.185/0.038 ms
root@1d5812af4363:/# exit
exit
```

Рисунок 11 – Проверка связи контейнеров

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`.

```

alena@LAPTOP-ARKAM4ED:~$ docker run -d --name mysql-container -e MYSQL_ROOT_PASSWORD=my-secret-pw mysql:latest
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
82094d60af40dbb0dfa9a3c87a2d85475fa4e7645da5c8d210e6a5fc159184f6
alena@LAPTOP-ARKAM4ED:~$ docker run -d --name nginx-con --link mysql-container:mysql -p 80:80 nginx:latest
3eb3e77f5d81c95d89fc1057de6749cbfd27847e1ce3aadc5e778ce487f1ca0e

```

Рисунок 12 – Связь контейнеров с помощью `--link`

Вывод: мы познакомились с использованием Docker для управления томами и сетями.

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
```