

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное автономное образовательное
учреждение высшего образования**
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций
по дисциплине «Анализ данных»

Лабораторная работа 2
«Работа в Docker с сетью контейнеров и томами»

Выполнил студент группы ИВТ-б-о-21-1

Пентухов С. А. «_____» ____ 20__ г.

Подпись студента _____

Работа защищена «_____» ____ 20__ г.

Проверил Воронкин Р.А

(подпись) _____

Ставрополь, 2023 г

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

Ссылка на гид - https://github.com/Pentuhov/AD_D-2

Ход работы

Задание 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.

```
system@DESKTOP-42FSGJH:~$ docker network create my_custom_network
85bd90f8b971f01d1c7627f19427c9c7ba08fd8b04fd87327b978f2b999acd5f
```

Рис. 1 – Создание сети

```
system@DESKTOP-42FSGJH:~$ docker run --network=my_custom_network -d --name db_container -e postgres_password=1 postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
1f7ce2fa46ab: Already exists
a797f18545f0: Pull complete
969823a73455: Pull complete
3cf63429c214: Pull complete
0829252d0d9a: Pull complete
b66a9305e34c: Pull complete
799c670608dc: Pull complete
870cc573d452: Pull complete
fb2a58049417: Pull complete
79e525202743: Pull complete
a3104899b78b: Pull complete
43ef4fb89a17: Pull complete
bbf10b4ac81d: Pull complete
Digest: sha256:71da05df8c4f1e1bac9b92ebfba2a0eeb183f6ac6a972fd5e55e8146e29efe9c
Status: Downloaded newer image for postgres:latest
4c19d78105b7385dac43e6db37101726689dfb4fb55f89b31c02139a553359b4
```

Рис. 2 - Проверка наличия контейнеров в сети

```

system@DESKTOP-42FSGJH:~$ docker network inspect my_custom_network
[
  {
    "Name": "my_custom_network",
    "Id": "7c28cb4f49eeafcb413bf7eb280b74ec5c8ea34c9f9874e9edacbb7465bde899",
    "Created": "2023-11-28T21:35:38.502277499Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.20.0.0/16",
          "Gateway": "172.20.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {},
    "Options": {},
    "Labels": {}
  }
]
system@DESKTOP-42FSGJH:~$

```

Рис. 3 - Проверка наличия контейнеров в сети

Задание 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.

```

system@DESKTOP-42FSGJH:~$ docker volume create vol_data
vol_data
system@DESKTOP-42FSGJH:~$ docker run -d -v vol_data:/data --name container1 nginx
b5b81e7877d6edddb3afef5ec96bd3dd7a8482133782673fd39fb972d444851c
system@DESKTOP-42FSGJH:~$ docker exec -it container1 bash
root@b5b81e7877d6:/# echo "Hello World!" > /data/data_file.txt
root@b5b81e7877d6:/# exit
exit
system@DESKTOP-42FSGJH:~$ docker run -d -v vol_data:/data --name container2 nginx
a45bc04e5b3f18aa18a2798e7c8c871fd40b588aaafc77dac935d64b6306e0d8
system@DESKTOP-42FSGJH:~$ docker exec -it container2 bash
root@a45bc04e5b3f:/# cat /data/data_file.txt
Hello World!
root@a45bc04e5b3f:/# exit
exit
system@DESKTOP-42FSGJH:~$

```

Рис. 4 – Передача данных через том

Задание 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.

```
system@DESKTOP-42FSGJH:~$ docker swarm init
Swarm initialized: current node (yrp7bcbvflj7k8teh5lit1prq) is now a manager.

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

    docker swarm join --token SWMTKN-1-538jk39dc4ic9zibmen3v214kbl8d2elbhi02bv0kzh3lurje0-6s74v0cs8il25
    27f01lj5h7z 192.168.65.3:2377

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

system@DESKTOP-42FSGJH:~$ docker network create -d overlay --attachable my_overlay_network
jmd068daeq2c9yu89l9a86py3
system@DESKTOP-42FSGJH:~$ docker run --network=my_overlay_network -d --name swarm1 nginx
837d95d1eb5877a0d986783f97eb2717c475b0ccc2e7e6ed253913f673035428
system@DESKTOP-42FSGJH:~$ docker run --network=my_overlay_network -d --name swarm2 nginx
ec6619bb5b661ae96e42eff48400ba1df629c7655be90b45bd54e06fcb2c9888
system@DESKTOP-42FSGJH:~$
```

Рис. 5 – Создание сети overlay

Задание 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.

```
system@DESKTOP-42FSGJH:~$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' d
b_container
172.18.0.3
system@DESKTOP-42FSGJH:~$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' w
eb_container
172.18.0.2
```

Рис. 6 - Связь контейнеров по IP-адресу

```
root@78826c827c22:/# 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=0.172 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.138 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=0.151 ms
^C
--- 172.18.0.3 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2110ms
rtt min/avg/max/mdev = 0.138/0.153/0.172/0.014 ms
root@78826c827c22:/#
```

Рис. 7- Связь контейнеров по IP-адресу

Задание 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`.

```
system@DESKTOP-42FSGJH:~$ 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
6b001d52aa4794015d0aa9d5e66e9d15406e490816abee58700363d5c7b78f0a
system@DESKTOP-42FSGJH:~$ docker run -d --name ng_container --link mysql_container:db -p 8080:80 nginx
2f89497ccdfad294031eb67d32346a99caf7c33360230968c7a11c3522dbe0cd
```

Рис. 8 – Связь контейнеров с помощи ссылки

Вывод: Научился способу использования 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`