**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

Стригалов Дмитрий.


«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.

```
root@ReviOS:~# docker network create my_custom_network
a9de2b7c2d5959f39b022ab84b26ae2b6edc474eedad8b89541778b564f2a8da
root@ReviOS:~# docker run --network=my_custom_network -d --name web_container nginx
52524c491a050e9009a6f15a663a457224352aa37a9ecada93fc45690bcfc185
```

Рисунок 1 – Создание сети

```
root@ReviOS:~# docker run --network=my_custom_network -d --name db_container -e POSTGRES_PASSWORD=123 p
ostgres
```


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.

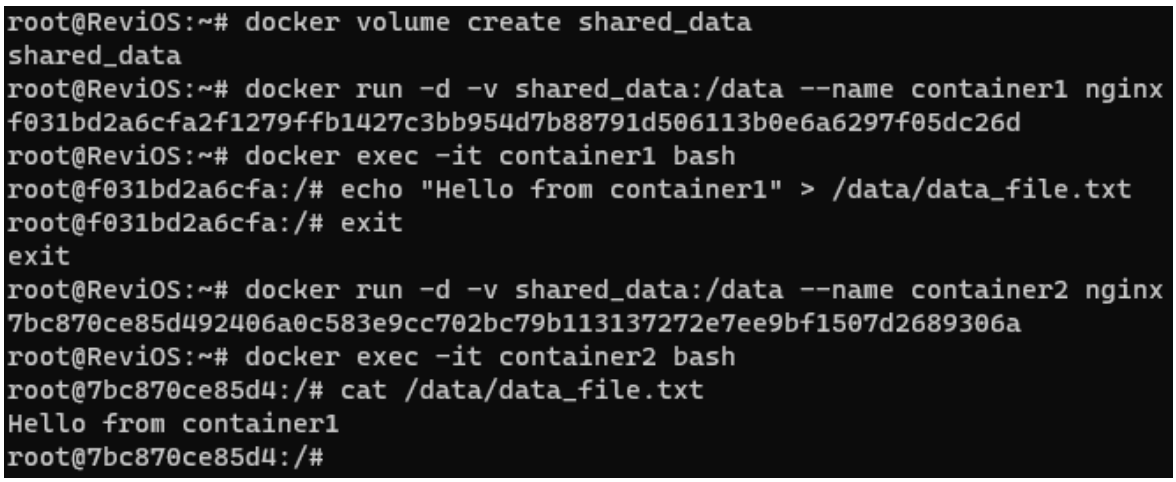



Рисунок 3 – Передача данных через том

**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.

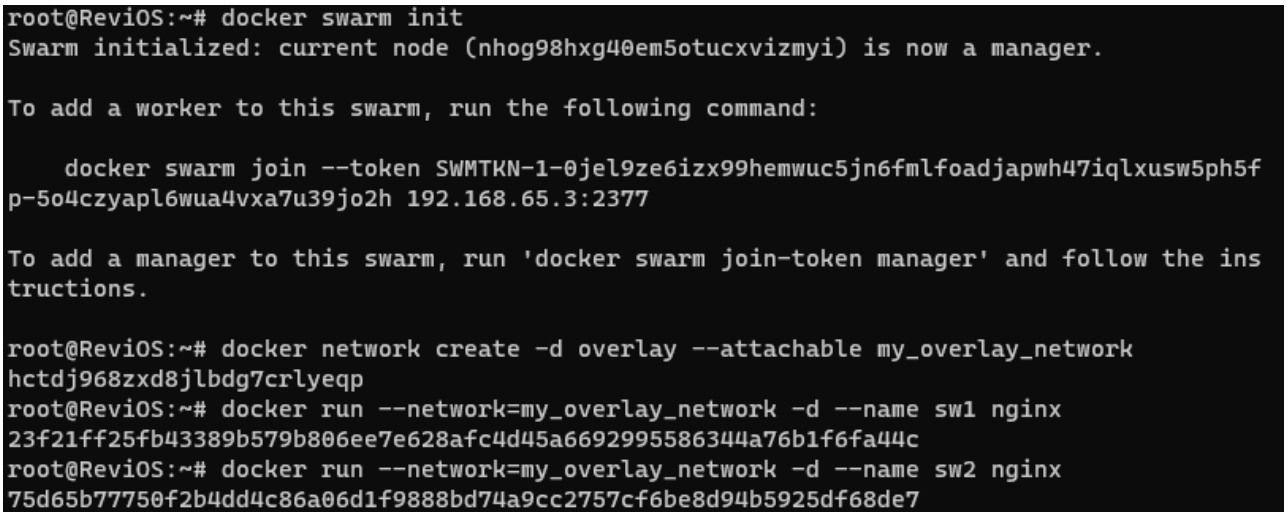
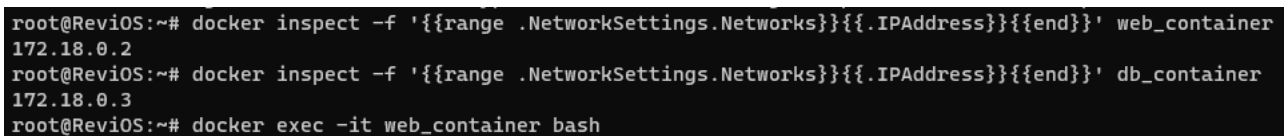



Рисунок 4 – Создание сети overlay

**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.



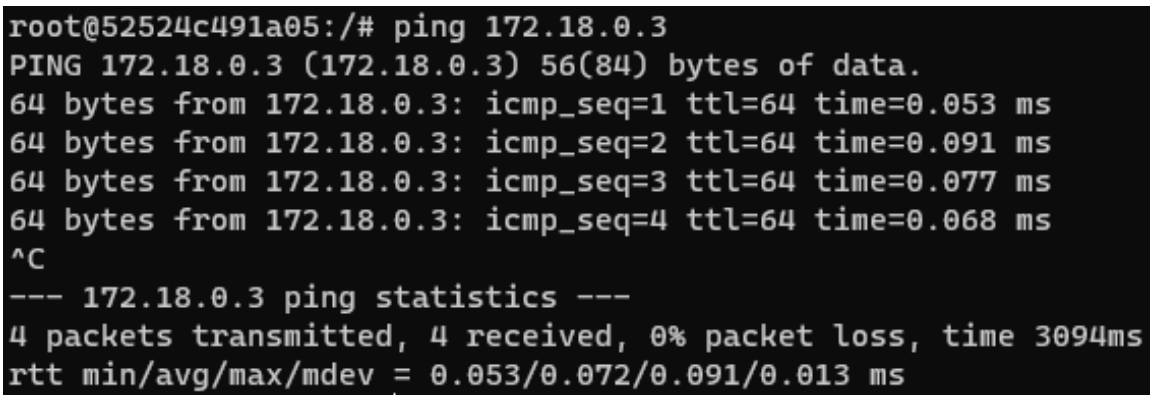



Рисунок 5 - Связь контейнеров по IP-адресу

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 .

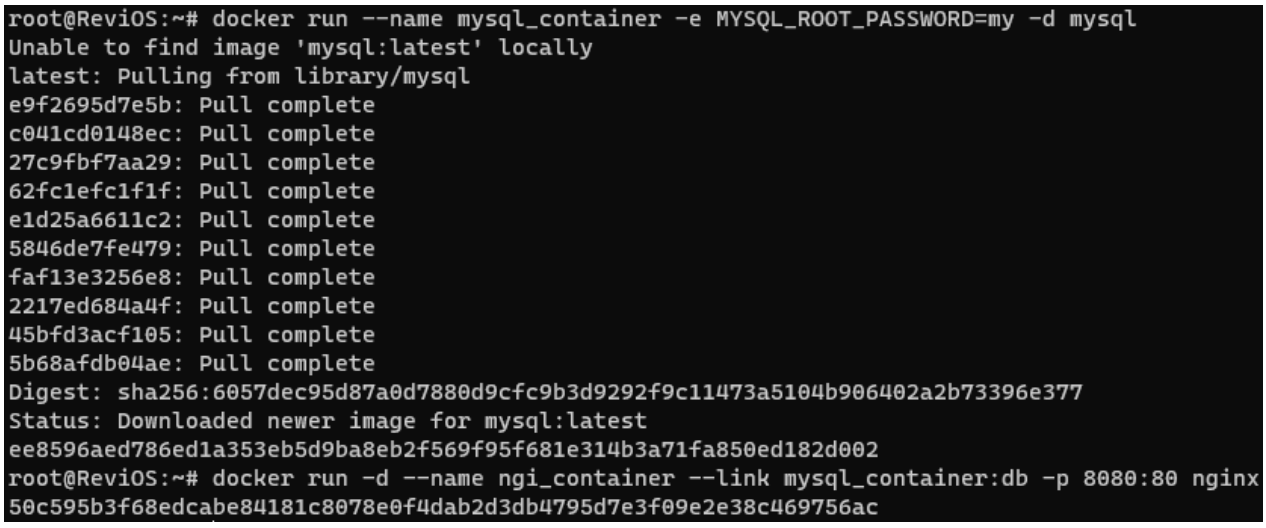
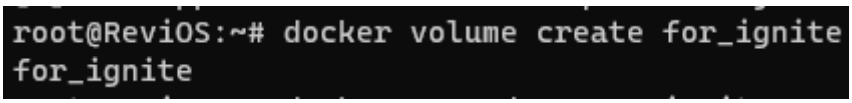



Рисунок 6 – Связь контейнеров с помощи ссылки

**Индивидуальное задание**

Создать контейнер ignite_container_1 с образом apacheignite, таблицу в базе данных и несколько записей в ней. Сделать возможным доступ к созданной таблице в новом контейнере ignite_container_2, а также выполнить к ней запрос после удаления контейнера ignite_container_1 и образа apacheignite.




```
root@ReviOS:~# docker run -d --name ignite_container_2 --network ignite_network \
  -v ignite_data:/opt/ignite/work \
  apacheignite/ignite
53841308d9f173fcaa8e8174ffad9382d77539d1eb3a45d89939f692a44c6346
root@ReviOS:~# docker exec -it ignite_container_2 /opt/ignite/apache-ignite/bin/sqlline.sh --color=true -u jdbc:ignite:t
hin://127.0.0.1/ -n ignite -p ignite
sqlline version 1.9.0
```

```
0: jdbc:ignite:thin://127.0.0.1/> create table users (Id LONG PRIMARY KEY, name VARCHAR) WITH "template=replicated";
No rows affected (0.032 seconds)
0: jdbc:ignite:thin://127.0.0.1/> INSERT INTO users (id, name) VALUES (1, 'Alex');
1 row affected (0.029 seconds)
0: jdbc:ignite:thin://127.0.0.1/> INSERT INTO users (id, name) VALUES (2, 'Jackson');
1 row affected (0.002 seconds)
0: jdbc:ignite:thin://127.0.0.1/> select * from users;
+----+---------+
| ID |  NAME   |
+----+---------+
| 1  | Alex    |
| 2  | Jackson |
+----+---------+
2 rows selected (0.026 seconds)
```

```
root@ReviOS:~# docker rmi -f apacheignite/ignite
Untagged: apacheignite/ignite:latest
Untagged: apacheignite/ignite@sha256:43c9273e85682f62ca0d2a2ae00bb21f13ccbc9c6a3dd17bbc2097ee262fd0a2
root@ReviOS:~# docker rm -f ignite_container_2
ignite_container_2
```

Рисунок 7 – Работа с первым контейнером

```
root@ReviOS:~# docker run -d --name ignite_container_3 --network ignite_network \
  -v ignite_data:/opt/ignite/work \
  apacheignite/ignite
Unable to find image 'apacheignite/ignite:latest' locally
latest: Pulling from apacheignite/ignite
Digest: sha256:43c9273e85682f62ca0d2a2ae00bb21f13ccbc9c6a3dd17bbc2097ee262fd0a2
Status: Downloaded newer image for apacheignite/ignite:latest
272991350f9f0dbc125c7dc7663190a1b422a393b459c3c221c291c99cb73a53
root@ReviOS:~# docker exec -it ignite_container_3 /opt/ignite/apache-ignite/bin/sqlline.sh --color=true -u
 jdbc:ignite:thin://127.0.0.1/ -n ignite -p ignite
sqlline version 1.9.0
0: jdbc:ignite:thin://127.0.0.1/> SELECT * FROM users;
+----+---------+
| ID |  NAME   |
+----+---------+
| 1  | Alex    |
| 2  | Jackson |
+----+---------+
2 rows selected (0.062 seconds)
```

Рисунок 8 - Работа со вторым контейнером

**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.