МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра информационных систем и технологий

Отчет по лабораторной работе «основы работы с Docker».

Дисциплина: «Основы программной инженерии»

Выполнил:

Студент группы ПИЖ-б-о-22-1,

направление подготовки: 09.03.04

«Программная инженерия»

ФИО: Джараян Арег Александрович

Проверил:

Воронкин Р. А.

Ставрополь 2023

Тема: Лабораторная работа 2. Работа в Docker с сетью контейнеров и томами.

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

Выполнение работы:

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.

```
C:\Users\aregd>docker network create my_custom_network
bc0f1efb7a54ed4d4133f88f5f9d6bbc4d55725616ea805d43577a880a5e531f

C:\Users\aregd>docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
af107e978371: Pull complete
85f7bca87921: Pull complete
948f1cf08e62: Pull complete
1a83ab26a0f0: Pull complete
12bab27fafd3: Pull complete
644cfda281a1: Pull complete
03299695f2b9: Pull complete
6e36bf1505f3: Pull complete
a35465a6a76a: Pull complete
83b026289c5c: Pull complete
c158e73dda41: Pull complete
264ae53c0064: Pull complete
2e3c2c5fbb6d: Pull complete
08c5357f23b5: Pull complete
Digest: sha256:e2135391c55eb2ecabaaaeef4a9538bb8915c1980953fb6ce41a2d6d3e4b5695
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```

Рисунок 1.1 – Создание сети

```
C:\Users\aregd>docker run --network=my_custom_network -d postgres:9.5.20-alpine
af014eb398e4574e8a55e198d35962886bdde8a2562cbec359e355719e31ce38

C:\Users\aregd>
```

Рисунок 1.2 – Создание контейнеров

```
C:\Users\aregd>docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS          PORTS      NAMES
af014eb398e4   postgres:9.5.20-alpine   "docker-entrypoint.s…"   51 seconds ago   Up 49 seconds   5432/tcp   lucid_lehma
nn
868c5f5c73ac   nginx                  "/docker-entrypoint.…"   10 minutes ago   Up 10 minutes   80/tcp     upbeat_curi
e
```

Рисунок 1.3 – Активные контейнеры

```
C:\Users\aregd>docker inspect -f '{{.NetworkSettings.Networks}}' lucid_lehmann
'map[my_custom_network:0xc000000000]'

C:\Users\aregd>docker inspect -f '{{.NetworkSettings.Networks}}' upbeat_curie
'map[my_custom_network:0xc000000000]'
```

Рисунок 1.4 – Оба контейнера работают на одной 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.

```
C:\Users\aregd>docker run -d -v tom:/areg --name kont2 nginx
1edd5d381579a8e504a5d3ff693a47d689559d09ad092a7b0f3b9ded6f9b16bb
```

Рисунок 2.1 – Создание контейнеров с использованием тома

```
C:\Users\aregd>docker run -d -v tom:/areg --name kont1 ubuntu
4c2cb4f567813bdd2f86f0007e7b3ca01350d6d86f37d41c181e57db304cf260
```

Рисунок 2.2 - Создание контейнеров с использованием тома

```
C:\Users\aregd>docker run -it -v tom:/areg --name kont1 ubuntu
root@7b1216dfba27:/# cd /
root@7b1216dfba27:/# ls
areg  bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@7b1216dfba27:/# cd areg
root@7b1216dfba27:/areg# cat areg.txt
cat: areg.txt: No such file or directory
root@7b1216dfba27:/areg# touch areg.txt
root@7b1216dfba27:/areg# cat areg.txt
root@7b1216dfba27:/areg# echo "areg" > areg.txt
root@7b1216dfba27:/areg# cat areg.txt
areg
root@7b1216dfba27:/areg#
```

Рисунок 2.3 – Создание текстового файла с содержанием

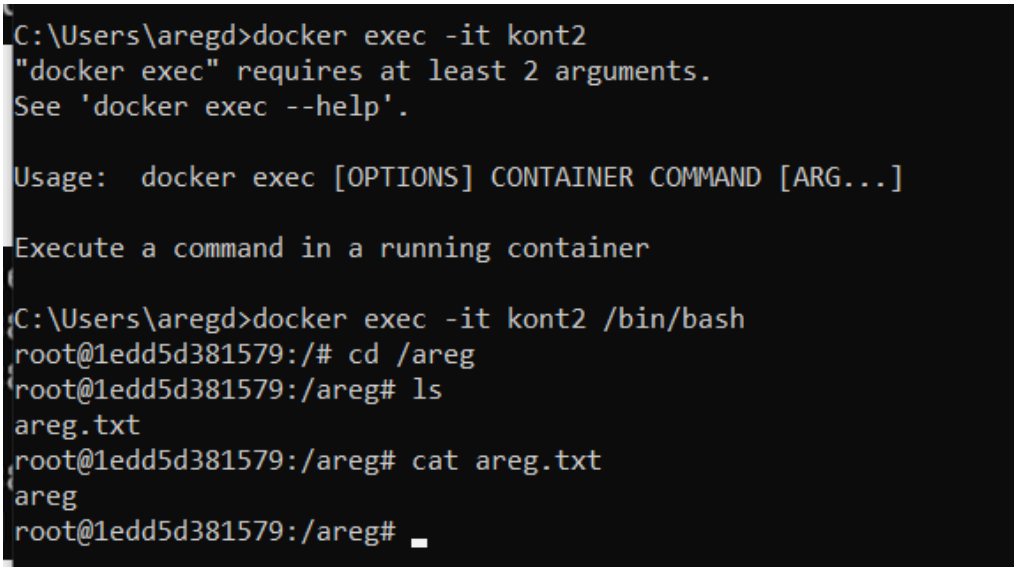

Рисунок 2.4 – Чтение файла из другого контейнера

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.

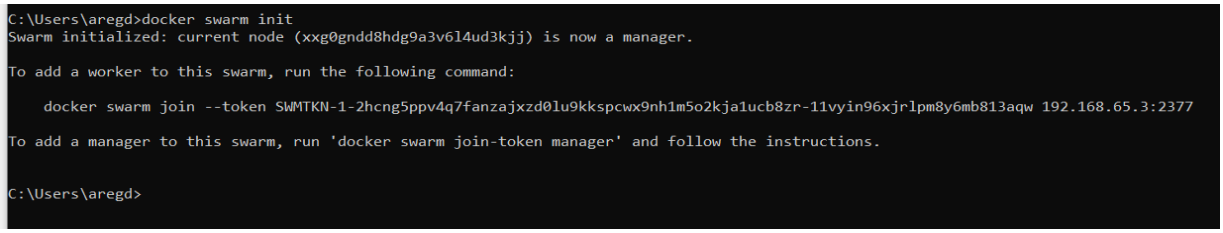

Рисунок 3.1 – Инициализация Swarm-кластера

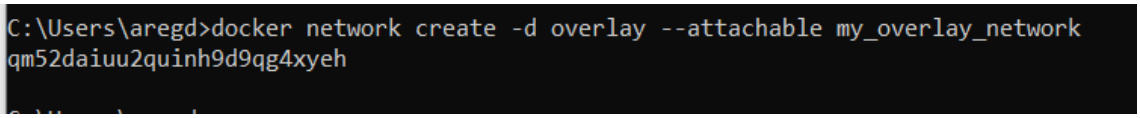

Рисунок 3.2 – Создание Ovelay-сети

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.

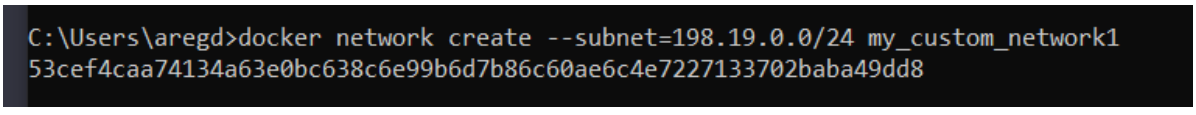

Рисунок 4.1 – Создание пользовательской сети

Рисунок 4.2 - Запуск первого контейнера

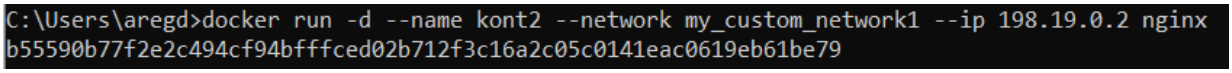

Рисунок 4.3 - Запуск второго контейнера

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 .

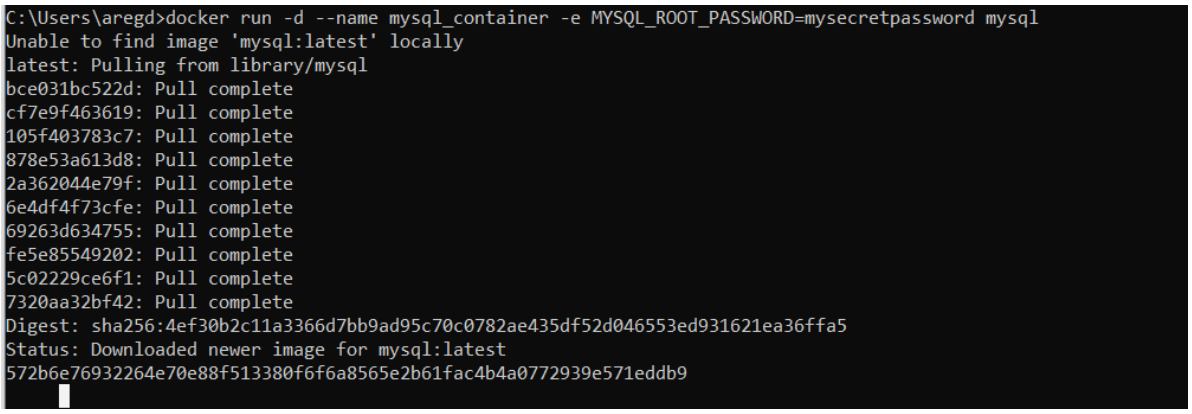

Рисунок 5.1 – Создание первого контейнера

Рисунок 5.2 – Создание второго контейнера и установка связи с первым контейнером

Контрольные вопросы:

1. Как создать новый том в Docker?

Используйте команду docker volume create <имя_тома> для создания нового тома.

2. Как удалить существующий том в Docker?

Выполните команду docker volume rm <имя_тома> для удаления существующего тома.

3. Как просмотреть список всех созданных томов в Docker?

Воспользуйтесь командой docker volume ls для просмотра списка всех созданных томов.

4. Как создать том с определенным именем?

Используйте команду docker volume create --name <имя_тома> для создания тома с определенным именем.

5. Как присоединить том к контейнеру при его запуске?

При запуске контейнера используйте опцию -v или --volume, например: docker run -v <имя_тома>:<путь_в_контейнере> <имя_образа>.

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

Используйте команду docker volume inspect <имя_тома> для получения подробной информации о конкретном томе.

7. Как создать новую сеть в Docker?

Запустите команду docker network create <имя_сети> для создания новой сети.

8. Как удалить существующую сеть в Docker?

Выполните команду docker network rm <имя_сети> для удаления существующей сети.

9. Как просмотреть список всех созданных сетей в Docker?

Используйте команду docker network ls для просмотра списка всех созданных сетей.

10. Как создать пользовательскую сеть с определенным именем?

При создании новой сети укажите опцию --driver bridge и имя сети, например: docker network create --driver bridge <имя_сети>.

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

Используйте опцию --network при выполнении команды docker run, например: docker run --network=<имя_сети> <имя_образа>.

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

Запустите команду docker network inspect <имя_сети> для получения подробной информации о конкретной сети.

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

Используйте опцию --network, например: docker run --network=<имя_сети> <имя_образа>.

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

По умолчанию контейнер будет подключен к сети "bridge".

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

Используйте опцию --network с указанием нескольких сетей, например: docker run --network=<сеть1> --network=<сеть2> <имя_образа>.

16. Как просмотреть список сетей, доступных на хосте Docker?

Используйте команду docker network ls для просмотра списка всех доступных сетей.

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

Контейнер по умолчанию будет подключен к сети "bridge". Просто выполните docker run <имя_образа>.

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

Используйте опцию --network host, например: docker run --network host <имя_образа>.