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 №17
дисциплины «Основы программной инженерии»

Выполнил:
Плугатырев Владислав Алексеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения

_____
(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г.

Тема: работа в Docker с сетью контейнеров и томами.

Цель работы: познакомиться с использованием Docker для управления томами и сетями.

Порядок выполнения работы

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.

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker network create my_custom_network
8785ec3b6280ca6f937f7cd17ac9a10ef6760627f7bb9faa87f44004614961f5
```

Рисунок 1.1 – Создание пользовательской сети

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run --network=my_custom_network -d --name web_container nginx
f29b985c09eb26ff7cf5a77cfcd43c3201e6d000edefe29816bb5b1c48bbe63e

C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run --network=my_custom_network --name db_container -e POSTGRES_PASSWORD=123 -d postgres
c115786cda5fd1909cd4f584cef728daf664f1e57a4b8b656cb6f5ea1ba5c2f3

C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED              STATUS              PORTS      NAMES
c115786cda5f   postgres   "docker-entrypoint.s…"   6 seconds ago        Up 5 seconds        5432/tcp   db_container
f29b985c09eb   nginx      "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp     web_container
```

Рисунок 1.2 – Запуск двух контейнеров, присоединенных к этой сети

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker inspect -f "{{.NetworkSettings.Networks}}" web_container
map[my_custom_network:0xc000534180]

C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker inspect -f "{{.NetworkSettings.Networks}}" db_container
map[my_custom_network:0xc0004d80c0]
```

Рисунок 1.3 – Сети контейнеров

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.

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker volume create shared_data
shared_data
```

Рисунок 2.1 – Создание общего тома

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run -itd -v shared_data:/data --name container1 ubuntu
13988c5bb4ec6642017bd6d70085ff3e9e1cb5c19679b92a9f8e875b36e7375a
```

Рисунок 2.2 – Запуск первого контейнера с указанным общим томом

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker exec -it container1 /bin/bash
root@13988c5bb4ec:/# ls
bin  boot  data  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@13988c5bb4ec:/# cd data
root@13988c5bb4ec:/data# ls
data_file.txt
root@13988c5bb4ec:/data# echo "Hello from container1" > data_file.txt
```

Рисунок 2.3 – Запись в файл из первого контейнера

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run -itd -v shared_data:/data --name container2 ubuntu
6f8a27e1c7e83dd4c770fbf7baa5495a3770dae60e796cb1018edd54e822660e
```

Рисунок 2.4 – Запуск второго контейнера

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker exec -it container2 bash
root@6f8a27e1c7e8:/# cat /data/data_file.txt
Hello from container1
```

Рисунок 2.5 – Чтение из второго контейнера

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.

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker swarm init
Swarm initialized: current node (5i4lnrbaj5v6ht2jafka4ckoz) is now a manager.

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

    docker swarm join --token SWMTKN-1-5n0qvlj1x59xgfa9fycqfk8aq9fvfshe3bi900zexqyrjxsr1h-biq1tamrp0irh6nrqm8f2yytl 192.168.65.3:2377

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

Рисунок 3.1 – Инициализация Swarm-кластера

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker network create -d overlay --attachable my_overlay_network
dox35f6t3kp4o5iresoc0whst
```

Рисунок 3.2 – Создание Ovelay-сети

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run --network=my_overlay_network -d nginx
d19591ecb1003c24122ec05aca7f084b64881b3ce19bdc7bb7b914566ca6df12

C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run --network=my_overlay_network -itd ubuntu
29c7ec799abac1aef8c911450a7c25ebff98206a0b77a441a7a99d6658388b3c
```

Рисунок 3.3 – Запуск двух контейнеров

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS     NAMES
29c7ec799aba   ubuntu    "/bin/bash"              About a minute ago   Up About a minute             compassionate_wozniak
d19591ecb100   nginx     "/docker-entrypoint.…"   2 minutes ago        Up 2 minutes        80/tcp    mystifying_jones

C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker inspect -f "{{.NetworkSettings.Networks}}" 29c7ec799aba
map[my_overlay_network:0xc000000000]

C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker inspect -f "{{.NetworkSettings.Networks}}" d19591ecb100
map[my_overlay_network:0xc000000000]
```

Рисунок 3.4 – Контейнеры находятся в одной сети

4. Связь контейнеров по IP-адресу: запустите два контейнера и присвойте им IP-адреса из одной пользовательской сети. Обеспечьте взаимодействие между контейнерами по их IPадресам.

```
C:\Users\vladi\OneDrive\Рабочий стол\Основы программной инженерии\Docker>docker run -itd --name cont1 --network=my_custom_network --ip 172.18.0.2 ubuntu
590a5258b003655200e0a57a383d995aaccae869d80b14b2757ee6c79e7ee34a
```

Рисунок 4.1 – Запуск первого контейнера

Рисунок 4.2 – Запуск второго контейнера

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 5.1 – Запуск первого контейнера

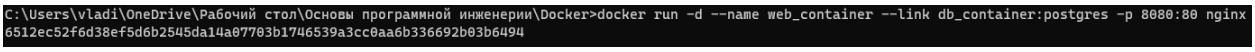

Рисунок 5.2 – Запуск второго контейнера

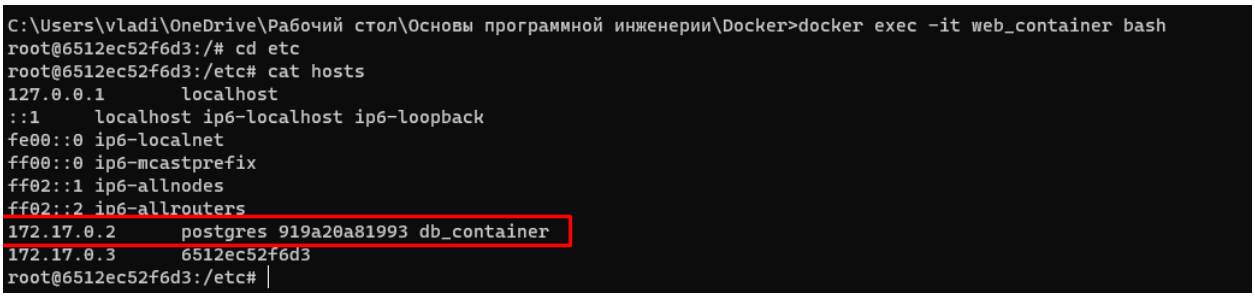

Рисунок 5.3 – Проверка связи контейнеров

Ответы на контрольные вопросы

1. Чтобы создать новый том в Docker, нужно выполнить команду docker volume create <имя_тома>. Например, чтобы создать том с именем my_volume, нужно выполнить команду docker volume create my_volume.

2. Чтобы удалить существующий том в Docker, нужно выполнить команду docker volume rm <имя_тома>. Например, чтобы удалить том с именем my_volume, нужно выполнить команду docker volume rm my_volume.

3. Чтобы просмотреть список всех созданных томов в Docker, нужно выполнить команду docker volume ls.

4. Чтобы создать том с определенным именем, нужно выполнить команду docker volume create <имя_тома>. Например, чтобы создать том с

именем my_volume, нужно выполнить команду docker volume create my_volume.

5. Чтобы присоединить том к контейнеру при его запуске, нужно использовать опцию -v или --mount при запуске контейнера. Например, чтобы присоединить том с именем my_volume к контейнеру, нужно выполнить команду docker run -v my_volume:/path/to/mount <имя_образа>.

6. Чтобы просмотреть подробную информацию о конкретном томе в Docker, нужно выполнить команду docker volume inspect <имя_тома>. Например, чтобы просмотреть информацию о томе с именем my_volume, нужно выполнить команду docker volume inspect my_volume.

7. Чтобы создать новую сеть в Docker, нужно выполнить команду docker network create <имя_сети>. Например, чтобы создать сеть с именем my_network, нужно выполнить команду docker network create my_network.

8. Чтобы удалить существующую сеть в Docker, нужно выполнить команду docker network rm <имя_сети>. Например, чтобы удалить сеть с именем my_network, нужно выполнить команду docker network rm my_network.

9. Чтобы просмотреть список всех созданных сетей в Docker, нужно выполнить команду docker network ls.

10. Чтобы создать пользовательскую сеть с определенным именем, нужно выполнить команду docker network create <имя_сети>. Например, чтобы создать сеть с именем my_network, нужно выполнить команду docker network create my_network.

11. Чтобы присоединить контейнер к пользовательской сети при его запуске, нужно использовать опцию --network при запуске контейнера. Например, чтобы присоединить контейнер к сети с именем my_network, нужно выполнить команду docker run --network my_network <имя_образа>.

12. Чтобы просмотреть подробную информацию о конкретной сети в Docker, нужно выполнить команду docker network inspect <имя_сети>.

Например, чтобы просмотреть информацию о сети с именем my_network, нужно выполнить команду docker network inspect my_network.

13. Чтобы указать определенную сеть при запуске контейнера с использованием docker run, нужно использовать опцию --network. Например, чтобы запустить контейнер на сети с именем my_network, нужно выполнить команду docker run --network my_network <имя_образа>.

14. Если не указана конкретная сеть, то контейнер будет подключен к сети “bridge” по умолчанию.

15. Чтобы присоединить контейнер к нескольким сетям сразу при его запуске, нужно использовать опцию --network несколько раз. Например, чтобы присоединить контейнер к сетям с именами my_network1 и my_network2, нужно выполнить команду docker run --network my_network1 --network my_network2 <имя_образа>.

16. Чтобы просмотреть список сетей, доступных на хосте Docker, нужно выполнить команду docker network ls.

17. Чтобы создать контейнер, подключенный к сети “bridge”, нужно выполнить команду docker run <имя_образа>. По умолчанию, контейнер будет подключен к сети “bridge”.

18. Чтобы создать контейнер, подключенный к сети “host”, нужно выполнить команду docker run --network host <имя_образа>. При использовании сети “host”, контейнер использует сетевые настройки хоста, а не свои собственные.