РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Основы работы с Docker»

Отчет по лабораторной работе

по дисциплине «Анализ данных»

Выполнил студент группы ИВТ-б-о-21-1

Толубаев Рамиль Ахметович

Подпись студента_____

Работа защищена « »_____20__г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: научиться использовать основные команды Docker для управления контейнерами и понимать их назначение.

Порядок выполнения работы:

Задача 1: Основы Docker

Загрузите образ Ubuntu с Docker Hub.

```
assadramil@Ramilassad:~$ docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
aece8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f
05f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

Рисунок 1 – Загрузка образа ubuntu

Создайте и запустите контейнер на основе этого образа.

```
assadramil@Ramilassad:~$ docker run -it ubuntu
```

Рисунок 2 - Запуск контейнера

Войдите в созданный контейнер и выполните команду ls , чтобы просмотреть файлы внутри контейнера.

```
root@8026d1eee6bf:/# ls
bin   dev  home  lib32  libx32  mnt  proc  run   srv  tmp  var
boot  etc  lib   lib64  media   opt  root  sbin  sys  usr
root@8026d1eee6bf:/#
```

Рисунок 3 - Выполнение команды ls внутри контейнера

Задача 2: Управление контейнерами и образами

Загрузите образ Nginx с Docker Hub.

```
assadramil@Ramilassad:~$ docker pull nginx:latest
latest: Pulling from library/nginx
1f7ce2fa46ab: Pull complete
9b16c94bb686: Pull complete
9a59d19f9c5b: Pull complete
9ea27b074f71: Pull complete
c6edf33e2524: Pull complete
84b1ff10387b: Pull complete
517357831967: Pull complete
Digest: sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d6
2ee
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

Рисунок 4 – Загрузка образа ngnix

Создайте контейнер на основе этого образа и пробросьте порт 8080 контейнера на порт 80 хоста.

```
assadramil@Ramilassad:~$ docker run -p 8080:80 -d nginx
2ebc716a01a7fd9c5cb9274f6d62ef828827de76d8da939ea4c77a81f242a54a
```

Рисунок 5 - Создание контейнера и проброс порта

Посмотрите список активных контейнеров и убедитесь, что ваш контейнер работает.

```
assadramil@Ramilassad:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS                  NAMES
2ebc716a01a7   nginx     "/docker-entrypoint.…"   About a minute ago   Up About a minute   0.0.0.0:8080->80/tcp   silly_villani
```

Рисунок 6 - Список активных контейнеров

Остановите и удалите контейнер.

```
assadramil@Ramilassad:~$ docker stop silly_villani
silly_villani
assadramil@Ramilassad:~$ docker rm silly_villani
silly_villani
```

Рисунок 7 - Остановка и удаление контейнера

Задача 3: Мониторинг и управление контейнерами

Запустите контейнер с именем "my_container".

```
assadramil@Ramilassad:~$ docker run --name my_container -d nginx
aaf102a28aff0adfe69bf10022e5d3bf4cbc0223cc4205bcc7956777356b0a1f
```

Рисунок 8 – Запуск контейнера

Используя команду docker ps , убедитесь, что контейнер запущен.

```
assadramil@Ramilassad:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS     NAMES
aaf102a28aff   nginx     "/docker-entrypoint.…"   52 seconds ago   Up 51 seconds   80/tcp    my_container
```

Рисунок 9 - Список активных контейнеров

Остановите контейнер.

```
assadramil@Ramilassad:~$ docker stop my_container
my_container
```

Рисунок 10 - Остановка контейнера

Проверьте его статус снова и убедитесь, что он остановлен.

```
assadramil@Ramilassad:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS     NAMES
aaf102a28aff   nginx     "/docker-entrypoint.…"   3 minutes ago    Exited (0) 29 seconds ago             my_container
8026d1eee6bf   ubuntu    "/bin/bash"              26 minutes ago   Exited (0) 18 minutes ago             distracted_gates
```

Рисунок 11 - Просмотр активных контейнеров

Удалите контейнер.

```
assadramil@Ramilassad:~$ docker rm my_container
my_container
```

Рисунок 12 - Удаление контейнера

Задача 4: Удаление образов и оптимизация дискового пространства

Загрузите образы Ubuntu и Alpine с Docker Hub.

```
assadramil@Ramilassad:~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
96526aa774ef: Pull complete
Digest: sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851
978
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
```

Рисунок 13 - Загрузка образа alpine

Создайте контейнеры на основе обоих образов.

```
assadramil@Ramilassad:~$ docker run --name container_1 -d ubuntu
175d5fe47fa5e6cd212d9319a04e0bb6bbc54253d1c6038959f57c8051d87809
assadramil@Ramilassad:~$ docker run --name container_2 -d ubuntu
abd151c913273ac3ce66423064d629e7f431f59f6ff0099d93b26f44e5e9ee68
```

Рисунок 14 - Создание контейнеров

Убедитесь, что контейнеры запущены и работают.

```
assadramil@Ramilassad:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS
        PORTS     NAMES
abd151c91327   ubuntu    "/bin/bash"   30 seconds ago   Exited (0) 29 seconds ago           container_2
175d5fe47fa5   ubuntu    "/bin/bash"   34 seconds ago   Exited (0) 33 seconds ago           container_1
```

Рисунок 15 - Список контейнеров

Удалите образ Ubuntu.

```
assadramil@Ramilassad:~$ docker rmi -f ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6
e496e5c1f05f
Deleted: sha256:e4c58958181a5925816faa528ce959e487632f4cfd192f8132f71b32df27
44b4
```

Рисунок 16 - Удаление образа ubuntu

Проверьте, что образ Ubuntu больше не существует, но образ Alpine остался на системе.

```
assadramil@Ramilassad:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS
        PORTS     NAMES
abd151c91327   ubuntu    "/bin/bash"   30 seconds ago   Exited (0) 29 seconds ago           container_2
175d5fe47fa5   ubuntu    "/bin/bash"   34 seconds ago   Exited (0) 33 seconds ago           container_1
```

Рисунок 17 - Просмотр контейнеров

Задача 5: Взаимодействие с контейнером

Запустите контейнер с именем "my_container" в фоновом режиме.

```
assadramil@Ramilassad:~$ docker run --name my_container -d nginx
6d05a382a318f91bce976fbc46cd6ae0126312807e21f1921292d16e606510e3
```

Рисунок 18 – Запуск контейнера

Используя команду docker exec , выполните команду ls -l /app внутри контейнера.

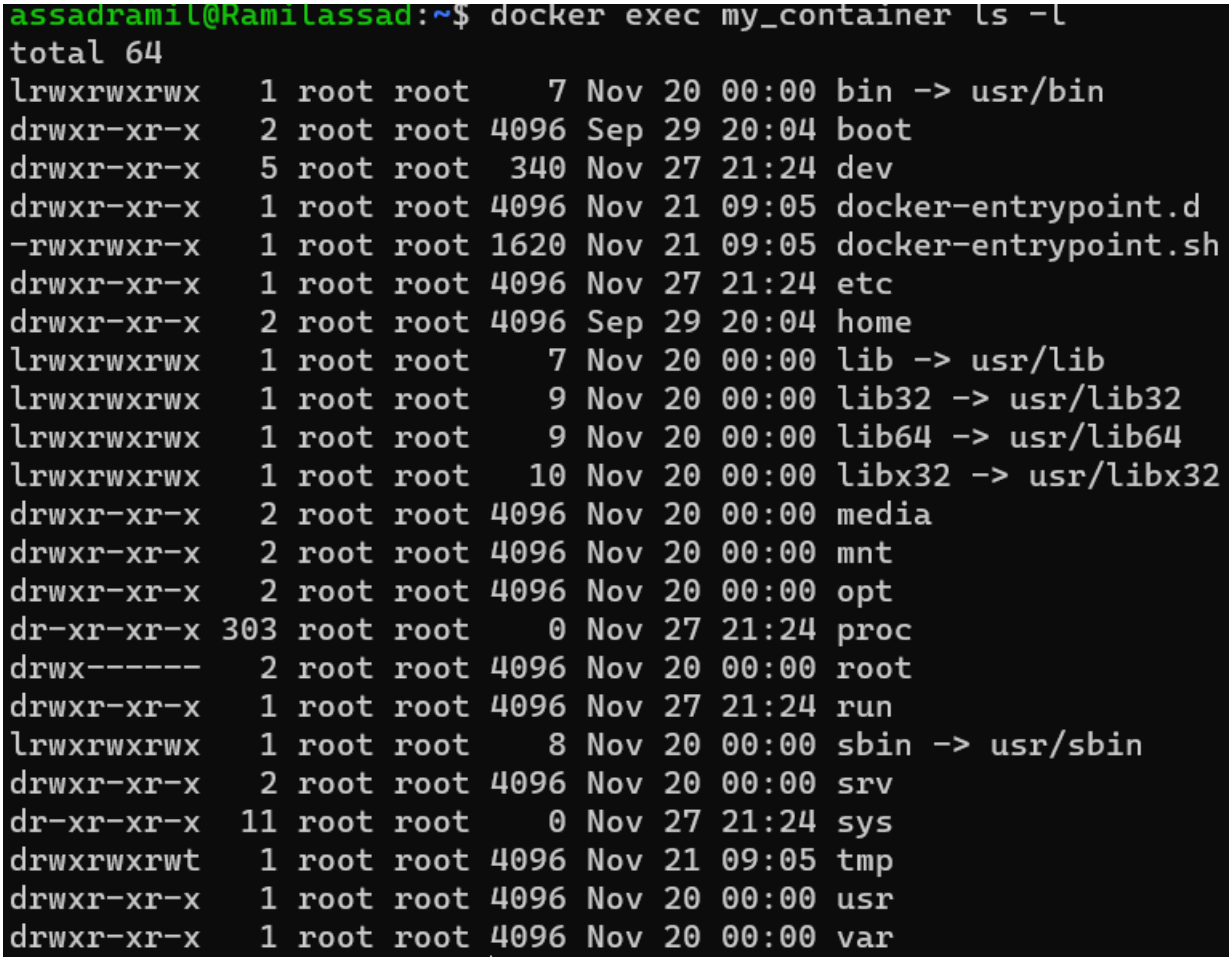

Рисунок 19 - Выполнение команды ls -l

Выполните команду ps aux внутри контейнера, чтобы увидеть список запущенных процессов.

Рисунок 20 - Выполнение команды ps aux

Остановите и удалите контейнер

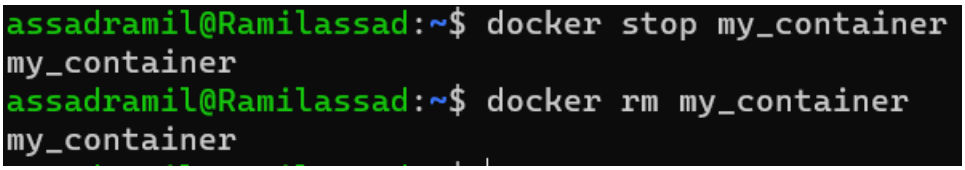

Рисунок 21 - Остановка и удаление контейнера

Контрольные вопросы:

1. Что делает команда docker pull?

Команда docker pull в Docker используется для загрузки образа контейнера с Docker Hub или другого репозитория.

2. Какой синтаксис используется для загрузки образа с Docker Hub с помощью docker pull?

docker pull <имя_образа>:<тег>

3. Как можно просмотреть список всех доступных образов на системе с помощью docker images?

docker images

Эта команда выведет список всех образов, которые находятся на вашей системе, включая их имена, теги, размер и ID.

4. Какой ключ используется для просмотра образов в формате таблицы с docker images?

docker images --format "table {{.Repository}}\t{{.Tag}}\t{{.Size}}"

5. Как создать и запустить контейнер с использованием docker run?

docker run [опции] <имя_образа> [команда] [аргументы]

6. Как пробросить порт при запуске контейнера с docker run?

docker run -p 8080:80 nginx

7. Как изменить имя контейнера при его создании с помощью docker run?

docker run --name my_container -d nginx

8. Как создать контейнер в фоновом режиме с docker run?

docker run -d nginx

9. Какая команда используется для просмотра активных контейнеров на системе?

docker ps

10. Какие опции могут использоваться с docker ps для отображения остановленных контейнеров?

docker ps -a

11. Как можно просмотреть список всех контейнеров, включая остановленные, с docker ps?

docker ps -a

12. Что делает команда docker start?

Команда docker start в Docker используется для запуска остановленных контейнеров.

13. Какой синтаксис используется для запуска остановленного контейнера с docker start?

docker start [опции] <имя_или_ID_контейнера>

14. Как запустить контейнер в фоновом режиме с docker start?

docker start -d my_container

15. Что делает команда docker stop?

Команда docker stop в Docker используется для остановки работающего контейнера.

16. Как остановить контейнер по его имени с помощью docker stop?

docker stop my_container

17. Как принудительно остановить контейнер с docker stop?

docker stop -f my_container

18. Что делает команда docker rm?

Команда docker rm в Docker используется для удаления контейнера, который был остановлен.

19. Как удалить контейнер по его ID с использованием docker rm?

docker rm 1234567890

20. Как удалить несколько контейнеров сразу с docker rm?

docker rm container1 container2

21. Что делает команда docker rmi?

Команда docker rmi в Docker используется для удаления образов контейнеров с вашей системы.

22. Как удалить Docker-образ по его имени и тегу с помощью docker rmi?

docker rmi ubuntu:20.04

23. Как удалить несколько Docker-образов сразу с docker rmi?

docker rmi image1 image2

24. Как выполнить команду внутри работающего контейнера с docker exec?

docker exec [опции] <имя_или_ID_контейнера> <команда> [аргументы]

25. Как выполнить команду внутри контейнера в интерактивном режиме с docker exec?

docker exec -it my_container /bin/bash

26. Как выполнить команду с использованием определенного пользователя внутри контейнера с docker exec?

docker exec -u 1000 my_container whoami

27. Какой ключ используется для запуска команды в фоновом режиме с docker exec?

docker exec -d my_container my_command

28. Как выполнить команду внутри контейнера с именем вместо ID с docker exec?

docker exec -it $(docker ps -q -f "name=my_container") /bin/bash

29. Как передать аргументы при выполнении команды с docker exec?

docker exec [опции] <имя_или_ID_контейнера> <команда> [аргументы]

30. Как проверить список доступных команд и опций для docker exec?

docker exec --help

31. Как передать переменную окружения в контейнер при его запуске?

docker run -e MYSQL_ROOT_PASSWORD=my-secret-pw mysql

32. Какой ключ используется для запуска контейнера в фоновом режиме с командой docker run?

docker run -d nginx

33. Как проверить статус выполнения контейнеров на системе с помощью docker ps?

docker ps -s

34. Как завершить выполнение контейнера без его удаления?

docker stop my_container

35. Каким образом можно удалить все остановленные контейнеры с системы?

docker rm $(docker ps -aq)

36. Что делает опция -a при использовании docker ps?

Добавление опции -a позволяет просматривать все контейнеры, включая те, которые были остановлены.

37. Что означает опция -q при выполнении docker ps?

Добавление опции –q выводит только ID контейнеров.

38. Как принудительно удалить контейнер с флагом -f?

docker rm -f my_container

39. Какой Docker-образ и какую команду можно использовать для создания контейнера с базой данных PostgreSQL?

docker run –name postgres_container postgres

40. Какой ключ используется для выполнения команды внутри контейнера в интерактивном режиме?

docker exec -it my_container <команда>

41. Какой ключ можно использовать для передачи ID пользователя при выполнении команды внутри контейнера?

С опцией -u мы указываем ID пользователя, от имени которого будет выполнена команда.

Вывод: были изучены основные команды Docker для управления контейнерами.