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

Богдашов Артём Владимирович.

«20» ноября 2023г.

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

```
root@DESKTOP-V640UPV:~# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Image is up to date for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
```

Рисунок 1 – Загрузка образа ubuntu

Создайте и запустите контейнер на

основе этого образа.

```
root@DESKTOP-V640UPV:~# docker run -it ubuntu
root@ea0028f5999e:/#
```

Рисунок2- Запуск контейнера

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

```
root@01a0ce2819ce:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
```

Рисунок 3 - Выполнение команды ls внутри контейнера

Задача 2: Управление контейнерами и образами

Загрузите образ Nginx с Docker Hub.

```
root@DESKTOP-V640UPV:~# 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: sha256:10d1f5b58f74683ad34eb29287e07dab1e90f10af243f151bb50aa5dbb4d62ee
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview nginx:latest
root@DESKTOP-V640UPV:~#
```

Рисунок 4 – Загрузка образа ngnix

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

```
root@DESKTOP-V640UPV:~# docker run -p 8080:80 -d nginx
6010ce9df25009290e335dea8017230938b9dc6c90044a1adbe5fe806b42de94
root@DESKTOP-V640UPV:~#
```

Рисунок 5 - Создание контейнера и проброс порта

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

```
root@DESKTOP-V640UPV:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED         STATUS         PORTS                  NAMES
2ad869a72a94   nginx     "/docker-entrypoint.…"   27 seconds ago   Up 25 seconds   0.0.0.0:8080->80/tcp   nervous_hofstadter
```

Рисунок 6 - Список активных контейнеров

Остановите и удалите контейнер.

```
root@DESKTOP-V640UPV:~# docker stop vibrant_engelbart
vibrant_engelbart
root@DESKTOP-V640UPV:~#

root@DESKTOP-V640UPV:~# docker rm vibrant_engelbart
vibrant_engelbart
root@DESKTOP-V640UPV:~#
```

Рисунок 7 - Остановка и удаление контейнера

Задача 3: Мониторинг и управление контейнерами Запустите контейнер с именем "my_container".

```
root@DESKTOP-V640UPV:~# docker run --name my_container -d nginx
78b143c4adf8ca3b1f4564aea43b1bd605a4fe6127b5a26d0df50563a4e6b617
root@DESKTOP-V640UPV:~#
```

Рисунок 8 – Запуск контейнера

Используя команду docker ps , убедитесь, что контейнер запущен.

```
root@DESKTOP-V640UPV:~# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS                  NAMES
04b443bc560d   nginx     "/docker-entrypoint.…"   25 seconds ago   Up 25 seconds   80/tcp                 my_container
2ad869a72a94   nginx     "/docker-entrypoint.…"   4 minutes ago    Up 4 minutes    0.0.0.0:8080->80/tcp   nervous_hofsta
dter
```

Рисунок 9 - Список активных контейнеров

Остановите контейнер.

```
root@DESKTOP-V640UPV:~# docker stop my_container
my_container
root@DESKTOP-V640UPV:~#
```

Рисунок 10 - Остановка контейнера

Проверьте его статус снова и убедитесь, что он остановлен.

```
root@DESKTOP-V640UPV:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                          PORTS                  NAMES
04b443bc560d   nginx     "/docker-entrypoint.…"   About a minute ago   Exited (0) 33 seconds ago                              my_container
2ad869a72a94   nginx     "/docker-entrypoint.…"   5 minutes ago        Up 5 minutes                    0.0.0.0:8080->80/tcp   nervous_hofstadter
ea0028f5999e   ubuntu    "/bin/bash"              9 minutes ago        Exited (129) 5 minutes ago                             priceless_proskuriakova
8758fb7ca61c   ubuntu    "/bin/bash"              23 hours ago         Exited (255) 11 minutes ago                            bold_shirley
eba0f1a70082   nginx     "/docker-entrypoint.…"   23 hours ago         Exited (255) 11 minutes ago     0.0.0.0:8080->80/tcp   some-nginx
root@DESKTOP-V640UPV:~#
```

Рисунок 11 - Просмотр активных контейнеров

Удалите контейнер.

```
root@DESKTOP-V640UPV:~# docker rm my_container
my_container
root@DESKTOP-V640UPV:~#
```

Рисунок 12 - Удаление контейнера

Задача 4: Удаление образов и оптимизация дискового пространства Загрузите образы Ubuntu и Alpine с Docker Hub.

```
root@DESKTOP-V640UPV:~# docker pull alpine
Using default tag: latest

latest: Pulling from library/alpine
96526aa774ef: Pull complete
Digest: sha256:eece025e432126ce23f223450a0326fbebde39cdf496a85d8c016293fc851978
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview alpine
root@DESKTOP-V640UPV:~#
```

Рисунок 13 - Загрузка образа alpine

Создайте контейнеры на основе обоих образов.

```
root@DESKTOP-V640UPV:~# docker run --name comtainer_1 -d ubuntu
6a2c7093369d0ff25734764737ebfd3967f6af850d2f74a336119dc6b4fd75b6
root@DESKTOP-V640UPV:~# docker run --name comtainer_2 -d ubuntu
62ce4c751aadc51f089fbe1465f23d06333c3b7e05a69f73135380ab826ca6e3
root@DESKTOP-V640UPV:~#
```

Рисунок 14 - Создание контейнеровУбедитесь, что контейнеры запущены и работают.

```
root@DESKTOP-V640UPV:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                          PORTS     NAMES
a2a88352c4ad   alpine    "/bin/sh"                About a minute ago   Exited (0) About a minute ago             mscontainer_2
5a09e894ed43   ubuntu    "/bin/bash"              About a minute ago   Exited (0) About a minute ago             mscontainer_1
2ad869a72a94   nginx     "/docker-entrypoint.…"   11 minutes ago       Exited (0) 9 seconds ago                  nervous_hofstadter
root@DESKTOP-V640UPV:~#

root@DESKTOP-V640UPV:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS                          PORTS                  NAMES
62ce4c751aad   ubuntu    "/bin/bash"              About a minute ago   Exited (0) 19 seconds ago                              comtainer_2
6a2c7093369d   ubuntu    "/bin/bash"              About a minute ago   Exited (0) 14 seconds ago                              comtainer_1
6010ce9df250   nginx     "/docker-entrypoint.…"   15 minutes ago       Up 15 minutes                   0.0.0.0:8080->80/tcp   busy_pasteur
01a0ce2819ce   ubuntu    "/bin/bash"              20 minutes ago       Up 22 seconds                                          lucid_payne
root@DESKTOP-V640UPV:~#
```

Рисунок 15 - Список контейнеровУдалите образ Ubuntu.

```
root@DESKTOP-V640UPV:~# docker rmi -f ubuntu
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Deleted: sha256:e4c58958181a5925816faa528ce959e487632f4cfd192f8132f71b32df2744b4
root@DESKTOP-V640UPV:~#
```

Рисунок 16 - Удаление образа ubuntu

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

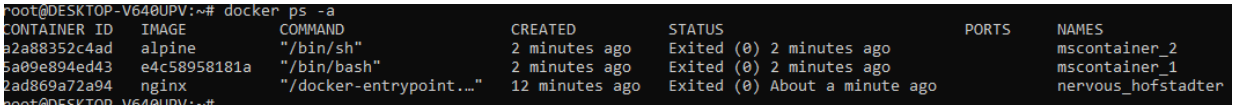

Рисунок 17 - Просмотр контейнеров

Задача 5: Взаимодействие с контейнером

Запустите контейнер с именем "my_container" в фоновом режиме.

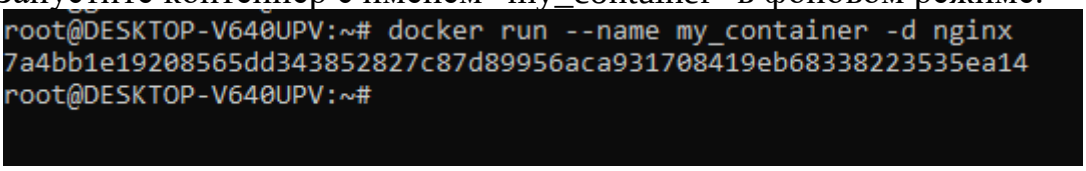

Рисунок 18 – Запуск контейнера

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

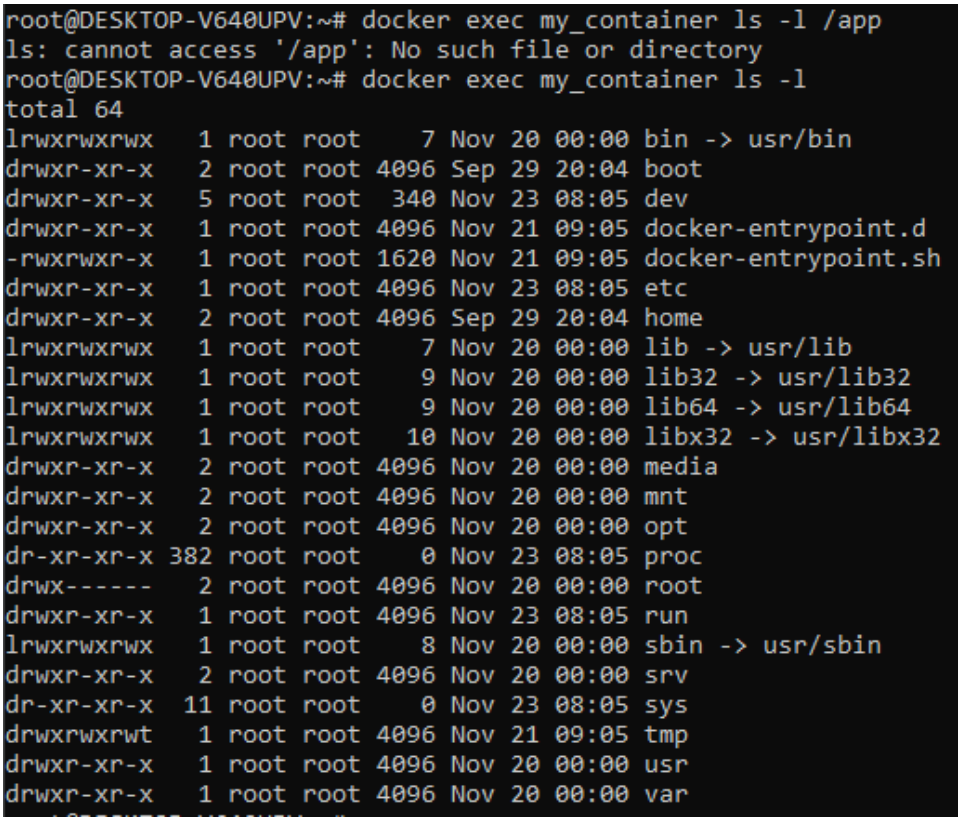

Рисунок 19 - Выполнение команды ls -l

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

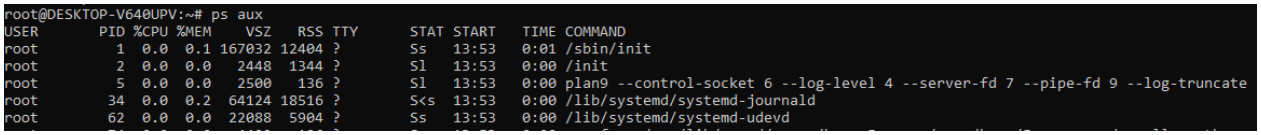

Рисунок 20 - Выполнение команды ps aux

Остановите и удалите контейнер.

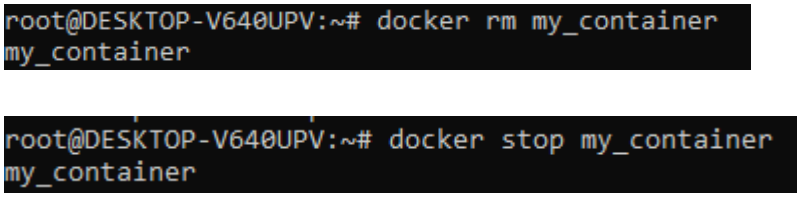

Рисунок 21 - Остановка и удаление контейнера

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 фоновомрежиме с командой docker run?

docker run -d nginx

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

docker ps -s

34. Как завершить выполнение контейнера без его удаления?

docker stop my_container

35. Каким образом можно удалить все остановленные контейнерыс 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 длясоздания контейнера с базой данных PostgreSQL?

docker run –name postgres_container postgres

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

docker exec -it my_container <команда>

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

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

Вывод: в ходе данной лабораторной работы были изучены основные команды Docker для управления контейнерами.