

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Институт цифрового развития**

**ОТЧЁТ**

**по лабораторной работе**

**Дисциплина: «Анализ данных»**

Выполнил: студент 3 курса

группы ИВТ-б-о-21-1

Уланбекова Айканыш Уланбековна

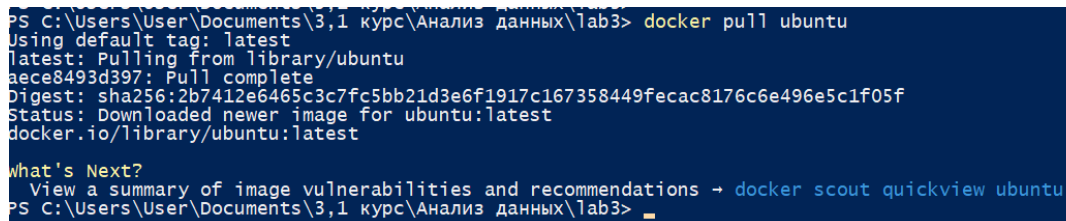
Ставрополь 2023

## Основы работы с Docker

**Цель занятия:** научить студентов использовать основные команды Docker для управления контейнерами и понимать их назначение.

### Задача 1. Основы Docker

Загрузите образ Ubuntu с Docker Hub.

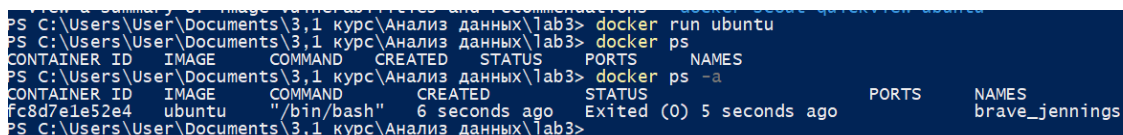


```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
a6ce8493d397: Pull complete
Digest: sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 1. Загрузка Ubuntu с Docker Hub

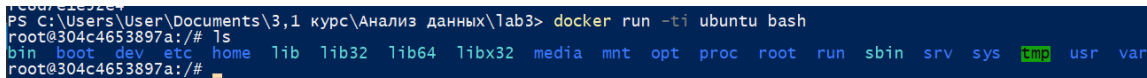
Создайте и запустите контейнер на основе этого образа.



```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run ubuntu
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS   NAMES
fc8d7e1e52e4   ubuntu    "/bin/bash"             6 seconds ago Exited (0) 5 seconds ago           brave_jennings
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 2. Запуск контейнера

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

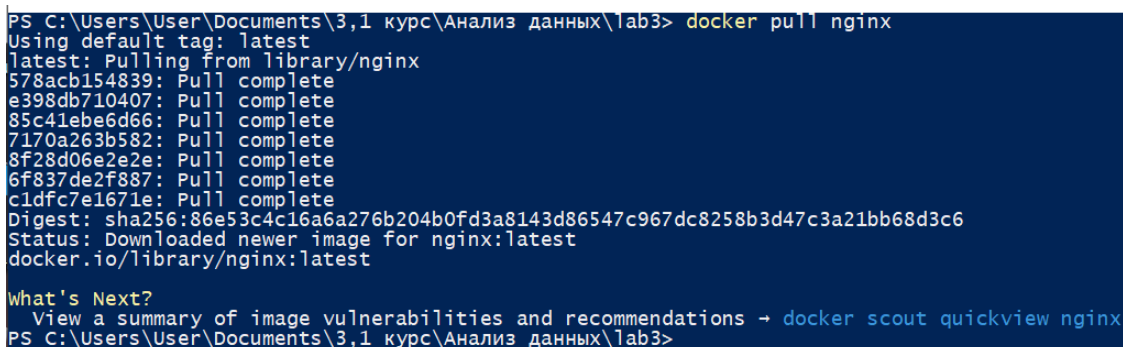


```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run -ti ubuntu bash
root@304c4653897a:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
root@304c4653897a:/#
```

Рисунок 3. Вход в контейнер

### Задача 2: Управление контейнерами и образами

Загрузите образ Nginx с Docker Hub.



```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
578acb154839: Pull complete
e398db710407: Pull complete
85c41ebe6d66: Pull complete
7170a263b582: Pull complete
8f28d06e2e2e: Pull complete
6f837de2f887: Pull complete
c1dfc7e1671e: Pull complete
Digest: sha256:86e53c4c16a6a276b204b0fd3a8143d86547c967dc8258b3d47c3a21bb68d3c6
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
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 4. Загрузка Nginx с Docker Hub

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

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run -p 8080:80 -d nginx 4fcffa3bf8e6db8f20b9839b9f8e8b9ca6d592f99251cbfa17c20f5bf93b5ee8
```

Рисунок 5. Создание контейнера

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

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
4fcffa3bf8e6   nginx    "/docker-entrypoint..." 7 seconds ago  Up 6 seconds  0.0.0.0:8080->80/tcp
304c4653897a   ubuntu   "bash"                   3 minutes ago  Exited (130)  About a minute ago
```

Рисунок 6. Список активных контейнеров

Остановите и удалите контейнер.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker stop 4fcffa3bf8e6
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker rm 4fcffa3bf8e6
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
304c4653897a   ubuntu   "bash"                   5 minutes ago  Exited (130)  3 minutes ago
```

Рисунок 7. Удаление контейнера

### Задача 3: Мониторинг и управление контейнерами

Запустите контейнер с именем «my\_container».

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run -d --name my_container nginx e2a1f4fc5cb5eda1668e307e72d87722dd8cd95493add4a63f83c577db2d006f
```

Рисунок 8. Запуск контейнера

Используя команду `docker ps`, убедитесь, что контейнер запущен.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
e2a1f4fc5cb5   nginx    "/docker-entrypoint..." 27 seconds ago  Up 26 seconds  80/tcp       my_container
```

Рисунок 9. Команда `docker ps`

Остановите контейнер.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker stop my_container
my_container
```

Рисунок 10. Остановка контейнера

Проверьте его статус снова и убедитесь, что он остановлен.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS              NAMES
e2a1f4fc5cb5   nginx    "/docker-entrypoint..." About a minute ago   Exited (0) 39 seconds ago              my_container
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 11. Проверка статуса

Удалите контейнер.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker rm my_container
my_container
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS              NAMES
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 12. Удаление контейнера

## Задача 4: Удаление образов и оптимизация дискового пространства

Загрузите образы Ubuntu и Alpine с Docker Hub.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> 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
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> 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?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview ubuntu
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 13. Загрузка Ubuntu и Alpine с Docker Hub

Создайте контейнеры на основе обоих образов.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run -d --name my_alpine alpine
a6cd34d3dd1177564624620a3a27f29072144b88555fae3b2b01c0709ab12e27
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run -d --name my_ubuntu ubuntu
396ef8757a748ac9e640404c961bcd45b1272791771b0437082b4fd0efc24a22
```

Рисунок 14. Создание контейнера

Убедитесь, что контейнеры запущены и работают.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS              NAMES
396ef8757a74   ubuntu   "/bin/bash"             28 seconds ago      Exited (0) 27 seconds ago              my_ubuntu
a6cd34d3dd11   alpine   "/bin/sh"                40 seconds ago      Exited (0) 39 seconds ago              my_alpine
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 15. Запуск и работа контейнеров

Удалите образ Ubuntu.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker rmi ubuntu -f
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:2b7412e6465c3c7fc5bb21d3e6f1917c167358449fecac8176c6e496e5c1f05f
Deleted: sha256:e4c58958181a5925816faa528ce959e487632f4cfd192f71b32df2744b4
```

Рисунок 16. Удаление Ubuntu

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

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest    c20060033e06   7 days ago    187MB
alpine        latest    8ca4688f4f35   5 weeks ago    7.34MB
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 17. Проверка

## Задача 5: Взаимодействие с контейнером

Запустите контейнер с именем «my\_container» в фоновом режиме.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker run -d --name my_container nginx
fd2718ca0cfdb304b38c3d2e8566ed40483a8cf82ddd765131ca80bd75ee160a
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 18. Запуск контейнера в фоновом режиме

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

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker exec -ti my_container bash
root@fd2718ca0cfdf:/# ls -l /app
ls: cannot access '/app': No such file or directory
root@fd2718ca0cfdf:/# ls -l /
total 64
lrwxrwxrwx 1 root root 7 Oct 30 00:00 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Sep 29 20:04 boot
drwxr-xr-x 5 root root 340 Nov 8 21:51 dev
drwxr-xr-x 1 root root 4096 Nov 1 05:12 docker-entrypoint.d
-rwxrwxr-x 1 root root 1620 Nov 1 05:11 docker-entrypoint.sh
drwxr-xr-x 1 root root 4096 Nov 8 21:51 etc
drwxr-xr-x 2 root root 4096 Sep 29 20:04 home
lrwxrwxrwx 1 root root 7 Oct 30 00:00 lib -> usr/lib
lrwxrwxrwx 1 root root 9 Oct 30 00:00 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 Oct 30 00:00 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 Oct 30 00:00 libx32 -> usr/libx32
drwxr-xr-x 2 root root 4096 Oct 30 00:00 media
drwxr-xr-x 2 root root 4096 Oct 30 00:00 mnt
drwxr-xr-x 2 root root 4096 Oct 30 00:00 opt
dr-xr-xr-x 249 root root 0 Nov 8 21:51 proc
drwx----- 1 root root 4096 Nov 8 21:56 root
drwxr-xr-x 1 root root 4096 Nov 8 21:51 run
lrwxrwxrwx 1 root root 8 Oct 30 00:00 sbin -> usr/sbin
drwxr-xr-x 2 root root 4096 Oct 30 00:00 srv
dr-xr-xr-x 11 root root 0 Nov 8 21:51 sys
drwxrwxrwt 1 root root 4096 Nov 1 05:12 tmp
drwxr-xr-x 1 root root 4096 Oct 30 00:00 usr
drwxr-xr-x 1 root root 4096 Oct 30 00:00 var
root@fd2718ca0cfdf:/#
```

Рисунок 19. Выполнение команд

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

```
root@a33a32c04f67:/# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  4624  3856 pts/0    Ss   21:59   0:00 /bin/bash
root        10  0.0  0.0   7060  1584 pts/0    R+   22:01   0:00 ps aux
root@a33a32c04f67:/#
```

Рисунок 20. Список запущенных процессов

Остановите и удалите контейнер.

```
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
fd2718ca0cfd   nginx     "/docker-entrypoint..." 9 minutes ago  Up 9 minutes  80/tcp       my_container
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker stop my_container
my_container
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker rm my_container
my_container
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
PS C:\Users\User\Documents\3,1 курс\Анализ данных\lab3>
```

Рисунок 21. Остановка и удаление контейнера

### Контрольные вопросы:

1. Что делает команда `docker pull`?

Загружает Docker-образ из Docker Hub или другого реестра образов.

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

`docker pull <имя_образа>`

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

`docker images`

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

`docker images --format table`

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

`docker run <имя_образа>`

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

`docker run -p <локальный_порт>:<контейнерный_порт> <имя_образа>`

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

`docker run --name <новое_имя> <имя_образа>`

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

`docker run -d <имя_образа>`

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`?

Запускает остановленный контейнер.

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

```
docker start <ID_или_имя_контейнера>
```

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

```
docker start -d <ID_или_имя_контейнера>
```

15. Что делает команда `docker stop`?

Останавливает работу контейнера.

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

```
docker stop <имя_контейнера>
```

17. Как принудительно остановить контейнер с `docker stop`?

```
docker stop -f <ID_или_имя_контейнера>
```

18. Что делает команда `docker rm`?

Удаляет контейнер.

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

```
docker rm <ID_контейнера>
```

20. Как удалить несколько контейнеров сразу с `docker rm`?

```
docker rm <ID_контейнера1> <ID_контейнера2> ...
```

21. Что делает команда `docker rmi`?

Удаляет Docker-образ.

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

```
docker rmi <имя_образа:тег>
```

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

`docker rmi <имя_образа1:тег1> <имя_образа2:тег2> ...`

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

`docker exec <ID_или_имя_контейнера> <команда>`

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

`docker exec -it <ID_или_имя_контейнера> <команда>`

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

`docker exec -u <пользователь> <ID_или_имя_контейнера> <команда>`

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

`docker exec -d <ID_или_имя_контейнера> <команда>`

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

`docker exec <имя_контейнера> <команда>`

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

`docker exec <ID_или_имя_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 <переменная>=<значение> <имя_образа>`

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

`docker run -d <имя_образа>`

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

`docker ps -s`

34. Как завершить выполнение контейнера без его удаления?

Нажмите комбинацию клавиш `Ctrl+C`.



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

```
docker container prune
```

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

Отображает все контейнеры, включая остановленные.

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

Выводит только ID контейнеров вместо полного формата.

38. Как принудительно удалить контейнер с флагом -f?

```
docker rm -f <ID_или_имя_контейнера>
```

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

Можно использовать официальный Docker-образ postgres и запустить его командой `docker run --name my-postgres -e POSTGRES_PASSWORD=<пароль> -p 5432:5432 -d postgres`

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

```
docker exec -it <ID_или_имя_контейнера> <команда>
```

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

```
docker exec -u <пользователь> <ID_или_имя_контейнера> <команда>
```