



как посмотреть группы в которых состоит пользователь?



Для просмотра групп, в которых состоит пользователь в Linux, вы можете использовать команду `groups`. Просто укажите имя пользователя после команды. Например:



```
bash
```

[Copy code](#)

```
groups имя_пользователя
```

Замените `имя_пользователя` на фактическое имя пользователя, информацию о котором вы хотите получить. Например:

```
bash
```

[Copy code](#)

```
groups daniilkonef
```

Эта команда выведет список групп, в которых состоит указанный пользователь.

```
daniilkonef@MyUbuntu:~$ sudo usermod -aG docker ${USER}
[sudo] password for daniilkonef:
daniilkonef@MyUbuntu:~$ groups daniilkonef
daniilkonef : daniilkonef adm cdrom sudo dip plugdev lpadmin lxd sambashare roots docker
```

Если вам нужно добавить пользователя в группу `docker`, для которой вы не выполнили вход, объявите имя пользователя явно, используя следующую команду:

```
$ sudo usermod -aG docker username
```

Copy

В дальнейшем в статье подразумевается, что вы запускаете команду `docker` от имени пользователя в группе **`docker`**. В обратном случае вам необходимо добавлять к командам префикс `sudo`.

Давайте перейдем к знакомству с командой `docker`.



DigitalOcean will be at SuperComputing 2023 in Denver from Nov 13 to 16. Sign up to meet with the team →

Blog



Products ▾

Solutions ▾

Developers ▾

Businesses ▾

Pricing

Tutorials

Questions

Learning Paths

For Businesses

Product Docs

Social Impact

## OpenSSH

## CONTENTS

Шаг 1 — Вход с  
привилегиями rootШаг 2 — Создание  
нового  
пользователяШаг 3 —  
Предоставление  
административных  
прав

Шаг 4 —

Нам нужно убедиться в том, что брандмауэр разрешает подключения SSH, чтобы мы могли выполнить вход в следующий раз. Чтобы разрешить эти подключения, можно ввести следующее:

# `ufw allow OpenSSH`

Copy

После этого мы можем активировать брандмауэр с помощью следующей команды:

# `ufw enable`

Copy



```
daniilkonef@MyUbuntu:~$ ufw app list
ERROR: You need to be root to run this script
daniilkonef@MyUbuntu:~$ sudo ufw app list
[sudo] password for daniilkonef:
Available applications:
  CUPS
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
  OpenSSH
daniilkonef@MyUbuntu:~$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
daniilkonef@MyUbuntu:~$ ufw enable
ERROR: You need to be root to run this script
daniilkonef@MyUbuntu:~$ sudo ufw enable
Firewall is active and enabled on system startup
daniilkonef@MyUbuntu:~$
```

```
daniilkonef@MyUbuntu:~$ sudo ufw status
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)

```
daniilkonef@MyUbuntu:~$
```

```
daniilkonef@MyUbuntu:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-11-16 22:38:48 MSK; 28min ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 1019 (dockerd)
     Tasks: 11
    Memory: 41.4M
       CPU: 6.304s
    CGroup: /system.slice/docker.service
            └─1019 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

## Шаг 4 – Работа с образами Docker

Контейнеры Docker запускаются из образов Docker. По умолчанию Docker получает образы из хаба [Docker Hub](#), представляющего собой реестр образов, который поддерживается компанией Docker. Кто угодно может создать и загрузить свои образы Docker в Docker Hub, поэтому для большинства приложений и дистрибутивов Linux, которые могут потребоваться вам для работы, уже есть соответствующие образы в Docker Hub.

Чтобы проверить, можете ли вы осуществлять доступ и загружать образы из Docker Hub, введите следующую команду:

```
$ docker run hello-world
```

Copy

Для просмотра загруженных на компьютер образов нужно ввести:

```
$ docker images
```

Copy

Вывод должен быть похож на представленный ниже:

Output

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	113a43faa138	4 weeks ago	81.2MB
hello-world	latest	e38bc07ac18e	2 months ago	1.85kB

```
daniilkonef@MyUbuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    e4c58958181a   6 weeks ago    77.8MB
hello-world    latest    9c7a54a9a43c   6 months ago   13.3kB
daniilkonef@MyUbuntu:~$
```

В качестве примера запустим контейнер с помощью последней версии образа Ubuntu. Комбинация параметров `-i` и `-t` обеспечивает интерактивный доступ к командному процессору контейнера:

```
$ docker run -it ubuntu
```

[Copy](#)

Командная строка должна измениться, показывая, что мы теперь работаем в контейнере. Она будет иметь следующий вид:

Output

```
root@d9b100f2f636:/#
```

```
daniilkonef@MyUbuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest   e4c58958181a   6 weeks ago   77.8MB
hello-world    latest   9c7a54a9a43c   6 months ago   13.3kB
daniilkonef@MyUbuntu:~$ docker run -it ubuntu
root@56e20d9a1eb9:/#
```

Теперь можно запускать любые команды внутри контейнера. Попробуем, например, обновить базу данных пакета внутри контейнера. Здесь перед командами не нужно использовать `sudo`, поскольку вы работаете внутри контейнера как пользователь с привилегиями **root**:

```
root@d9b100f2f636:/# apt update
```

[Copy](#)

Теперь в нем можно установить любое приложение. Попробуем установить Node.js:

```
root@d9b100f2f636:/# apt install nodejs
```

[Copy](#)

```
daniilkonef@MyUbuntu:~$ docker run -it ubuntu
root@56e20d9a1eb9:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu jammy InRelease [270 kB]
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
```

Чтобы выйти из контейнера, вводим команду `exit`.

```
root@dcca183549ea:/# node -v
v12.22.9
root@dcca183549ea:/# exit
exit
daniilkonef@MyUbuntu:~$
```



Через некоторое время после начала использования Docker на вашей машине будет множество активных (запущенных) и неактивных контейнеров. Просмотр \*\* активных контейнеров \*\*:

```
$ docker ps
```

Copy

Результат получится примерно следующим:

Output

CONTAINER ID	IMAGE	COMMAND	CREATED
--------------	-------	---------	---------

По нашей инструкции вы запустили два контейнера: один из образа `hello-world`, второй из образа `ubuntu`. Оба контейнера уже не запущены, но существуют в системе.

```
daniilkonef@MyUbuntu:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
daniilkonef@MyUbuntu:~$ docker ps --all
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
dcca183549ea   ubuntu   "/bin/bash"   16 minutes ago   Exited (0) 6 minutes ago   elated_euclid
56e20d9a1eb9   ubuntu   "/bin/bash"   4 days ago       Exited (129) 3 days ago    tender_lalande
426113547e96   hello-world   "/hello"   4 days ago       Exited (0) 4 days ago      stoic_shirley
1dd68800bb32   hello-world   "/hello"   5 days ago       Exited (0) 5 days ago      fervent_jennings
```

Контейнер `hello-world`, запущенный на предыдущем этапе, является примером контейнера, который запускается и завершает работу после вывода тестового сообщения. Контейнеры могут выполнять и более полезные действия, а также могут быть интерактивными. Контейнеры похожи на виртуальные машины, но являются менее требовательными к ресурсам.

В качестве примера запустим контейнер с помощью последней версии образа Ubuntu. Комбинация параметров `-i` и `-t` обеспечивает интерактивный доступ к командному процессору контейнера:

```
$ docker run -it ubuntu
```

Copy

Командная строка должна измениться, показывая, что мы теперь работаем в контейнере. Она будет иметь следующий вид:

Output

```
root@d9b100f2f636:/#
```



## ChatGPT

Для запуска и входа в командную оболочку в интерактивном режиме внутри контейнера, вы можете использовать команду `docker exec`. В вашем случае, где контейнер использует образ Ubuntu, команда будет выглядеть следующим образом:

bash

Copy code

```
docker exec -it elated_euclid /bin/bash
```

Где:

- `-it` обеспечивает интерактивный режим и подключение к терминалу контейнера.
- `elated_euclid` - это имя вашего контейнера.
- `/bin/bash` - это команда, которую вы хотите выполнить внутри контейнера. В данном случае, это запуск интерактивной оболочки bash.

После выполнения этой команды, вы будете находиться внутри контейнера и сможете выполнять команды в его оболочке.




```
daniilkonef@MyUbuntu:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED             STATUS              PORTS          NAMES
dcca183549ea   ubuntu   "/bin/bash"             About an hour ago   Up 36 minutes      -              elated_euclid
daniilkonef@MyUbuntu:~$ docker exec -it elated_euclid /bin/bash
root@dcca183549ea:/# ~
bash: /root: Is a directory
root@dcca183549ea:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@dcca183549ea:/# exit
exit
daniilkonef@MyUbuntu:~$
```

Для остановки запущенного контейнера используем команду `docker stop`, затем указываем идентификатор контейнера или его имя. В этот раз мы используем имя, которое назначил контейнеру Docker, то есть `sharp_volhard`:

```
$ docker stop sharp_volhard
```

Copy

Продолжить тут

 <https://www.digitalocean.com/community/tutorials/docker-ubuntu-18-04-1-ru>



You

а какой командой можно вывести размеры сразу всех контейнеров?



ChatGPT

Для вывода размеров всех контейнеров в системе, вы можете воспользоваться командой ``docker ps`` с параметром ``-s`` (или ``--size``):

bash

Copy code

```
docker ps -s
```

Эта команда выведет информацию о запущенных контейнерах, включая размеры их файловой системы.

Если вы хотите увидеть размеры всех контейнеров (включая остановленные), вы можете использовать команду ``docker ps -a -s``:

bash

Copy code

```
docker ps -a -s
```

Обратите внимание, что размеры могут быть выражены в байтах, килобайтах, мегабайтах и так далее в зависимости от их величины.



```
daniilkonef@MyUbuntu:~$ docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS              PORTS          NAMES          SIZE
33d56f1e7c36   hello-world    "/hello"                24 hours ago   Exited (0) 24 hours ago           goofy_gould    0B (virtual 13.3kB)
78ed6c26c6d5   ubuntu        "/bin/bash"             24 hours ago   Exited (0) 24 hours ago           keen_turing    5B (virtual 77.8MB)
dcca183549ea   ubuntu        "/bin/bash"             25 hours ago   Exited (137) 24 hours ago          elated_euclid  130MB (virtual 207MB)
56e20d9a1eb9   ubuntu        "/bin/bash"             5 days ago     Exited (129) 5 days ago          tender_lalande 130MB (virtual 207MB)
426113547e96   hello-world    "/hello"                5 days ago     Exited (0) 24 hours ago           stoic_shirley  0B (virtual 13.3kB)
1dd68800bb32   hello-world    "/hello"                7 days ago     Exited (0) 7 days ago             fervent_jennings 0B (virtual 13.3kB)
daniilkonef@MyUbuntu:~$
```

Для остановки запущенного контейнера используем команду `docker stop`, затем указываем идентификатор контейнера или его имя. В этот раз мы используем имя, которое назначил контейнеру Docker, то есть `sharp_volhard`:

```
$ docker stop sharp_volhard
```

Copy

Если вам контейнер больше не нужен, удаляем его командой `docker rm` с указанием либо идентификатора, либо имени контейнера. Чтобы найти идентификатор или имя контейнера, связанного с образом `hello-world`, используйте команду `docker ps -a`. Затем контейнер можно удалить.

```
$ docker rm festive_williams
```

Copy

```

daniilkonef@MyUbuntu:~$ docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES          SIZE
dcca183549ea   ubuntu        "/bin/bash"            25 hours ago   Exited (137)  24 hours ago   elated_euclid   130MB (virtual 207MB)
56e20d9a1eb9   ubuntu        "/bin/bash"            5 days ago     Exited (129)  5 days ago     tender_lalande   130MB (virtual 207MB)
426113547e96   hello-world   "/hello"               5 days ago     Exited (0)    25 hours ago   stoic_shirley    0B (virtual 13.3kB)
1dd68800bb32   hello-world   "/hello"               7 days ago     Exited (0)    7 days ago     fervent_jennings 0B (virtual 13.3kB)
daniilkonef@MyUbuntu:~$ docker rm fervent_jennings
fervent_jennings
daniilkonef@MyUbuntu:~$ docker rm stoic_shirley
stoic_shirley
daniilkonef@MyUbuntu:~$ docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES          SIZE
dcca183549ea   ubuntu        "/bin/bash"            25 hours ago   Exited (137)  24 hours ago   elated_euclid   130MB (virtual 207MB)
56e20d9a1eb9   ubuntu        "/bin/bash"            5 days ago     Exited (129)  5 days ago     tender_lalande   130MB (virtual 207MB)
daniilkonef@MyUbuntu:~$ docker rm tender_lalande
tender_lalande
daniilkonef@MyUbuntu:~$ docker ps -a -s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES          SIZE
dcca183549ea   ubuntu        "/bin/bash"            25 hours ago   Exited (137)  24 hours ago   elated_euclid   130MB (virtual 207MB)

```



You

по какой схеме докер придумывает имена контейнерам?



ChatGPT

Docker генерирует имена контейнеров по схеме "прилагательное\_существительное". Эти имена создаются автоматически, если вы не явно указали имя контейнера при его создании.

Такие имена состоят из двух частей:

1. **Прилагательное:** Это первая часть имени, которая добавляется для придания уникальности. Например, это может быть что-то вроде "focused", "angry", "elastic" и так далее.
2. **Существительное:** Вторая часть, которая обычно является существительным. Это может быть, например, "turing", "einstein", "thompson" и т.д.

Таким образом, Docker генерирует уникальные имена контейнеров, чтобы избежать конфликтов и обеспечить их отличие друг от друга. Если вы хотите явно указать имя контейнера при создании, вы можете использовать опцию `--name`:

bash

Copy code

```
docker run --name мое_имя_контейнера мой_образ
```

Где `мое_имя_контейнера` - это желаемое имя контейнера, а `мой_образ` - это имя Docker образа, который вы хотите использовать.







You

а это самое "существительное" это какие то знаменитости ?



ChatGPT

Да, вы правы. "Существительные" в именах контейнеров Docker обычно выбираются из набора знаменитостей, имен, животных или других сущностей. Это может быть что угодно, от имен компьютерных ученых до названий животных, мест и т. д. Эти "существительные" добавляются для придания именам контейнеров некоторого смысла и уникальности.

Примеры "существительных" в именах контейнеров могут включать "turing", "einstein", "thompson", "penguin", "angry\_babbage" и так далее. Это просто часть процесса генерации уникальных и легко запоминающихся имен контейнеров.



Запустить новый контейнер и задать ему имя можно с помощью параметра `--name`. Параметр `--rm` позволяет создать контейнер, который самостоятельно удалится после остановки. Для более подробной информации о данных и других опциях используйте команду `docker run help`.

bash

Copy code

```
docker run --name мое_имя_контейнера мой_образ
```

Где ``мое_имя_контейнера`` - это желаемое имя контейнера, а ``мой_образ`` - это имя Docker образа, который вы хотите использовать.

```

daniilkonef@MyUbuntu:~$ docker ps -a -s
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES        SIZE
dcca183549ea   ubuntu    "/bin/bash"             5 days ago    Exited (137)  4 days ago    elated_euclid  130MB (virtual 207MB)
daniilkonef@MyUbuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    e4c58958181a   7 weeks ago    77.8MB
hello-world    latest    9c7a54a9a43c   6 months ago    13.3kB
daniilkonef@MyUbuntu:~$ docker run --name altium-server ubuntu
daniilkonef@MyUbuntu:~$ docker ps -a -s
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES        SIZE
1246ad20aded   ubuntu    "/bin/bash"             23 seconds ago    Exited (0) 18 seconds ago    altium-server  0B (virtual 77.8MB)
dcca183549ea   ubuntu    "/bin/bash"             5 days ago    Exited (137)  4 days ago    elated_euclid  130MB (virtual 207MB)
daniilkonef@MyUbuntu:~$

```

## Шаг 7 – Сохранение изменений в контейнере в образ Docker

## Шаг 8 – Отправка контейнеров Docker в репозиторий Docker

В данном разделе показано, как отправлять образы Docker в Docker Hub. Научиться создавать собственный Docker-реестр можно с помощью статьи [How To Set Up a Private Docker Registry on Ubuntu 14.04](https://www.digitalocean.com/community/tutorials/docker-ubuntu-18-04-1-ru).

🔗 <https://www.digitalocean.com/community/tutorials/docker-ubuntu-18-04-1-ru>

// Tutorial //

# Установка Docker Compose в Ubuntu 18.04

<https://www.digitalocean.com/community/tutorials/how-to-install-docker-compose-on-ubuntu-18-04-ru>



ChatGPT

Для установки MySQL в контейнере Docker, вы можете воспользоваться официальным образом MySQL, предоставляемым Docker Hub. Вот пример команды, которую вы можете использовать:

```
bash Copy code
docker run --name my-mysql-container -e MYSQL_ROOT_PASSWORD=my-secret-pw
```

Где:

- `my-mysql-container` - это имя, которое вы выбираете для вашего контейнера.
- `my-secret-pw` - это пароль для пользователя root. Вы можете заменить его на свой уникальный пароль.
- `mysql:latest` - это тег образа MySQL, который указывает на последнюю версию. Вы можете использовать конкретный тег, если хотите установить определенную версию.

После выполнения этой команды, контейнер MySQL будет запущен, и вы сможете подключиться к нему с помощью клиента MySQL.

Если вы хотите использовать определенную версию MySQL, укажите тег образа соответствующий нужной версии, например:

```
bash Copy code
ame my-mysql-container -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:5.7
```

`docker run --name my-mysql-container -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:latest`  
`docker run --name my-mysql-container -e MYSQL_ROOT_PASSWORD=my-secret-pw -d mysql:5.7`

```
daniilkonef@MyUbuntu:~$ docker start my-mysql-container
my-mysql-container
daniilkonef@MyUbuntu:~$ docker exec -it my-mysql-container mysql -uroot -p
Enter password:
```

```
mysql> exit
Bye
daniilkonef@MyUbuntu:~$
```

```
daniilkonef@MyUbuntu:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
8e1a1e4d5254	mysql:latest	"docker-entrypoint.s..."	3 days ago	Up 7 minutes	3306/tcp, 33060/tcp	my-mysql-container



You

ок. как теперь мне обратно войти в сессию ?



ChatGPT

Чтобы вернуться в интерактивную сессию контейнера, вы можете использовать команду ``docker exec``. В вашем случае:

```
bash
```

Copy code

```
docker exec -it my-mysql-container mysql -uroot -p
```

Где:

- ``my-mysql-container`` - это имя вашего контейнера.
- ``mysql -uroot -p`` - это команда для входа в MySQL. Она указывает на то, что вы хотите войти с пользователем "root" и будет запрашивать пароль.

После выполнения этой команды, вы снова войдете в интерактивную сессию MySQL внутри контейнера.





**You**

ок. а как теперь мне установить контейнер phpmyadmin?




**ChatGPT**

Для установки контейнера phpMyAdmin вместе с контейнером MySQL вы можете использовать docker-compose. Docker Compose позволяет вам определить и запустить множество контейнеров в одном файле конфигурации.

Вот пример `docker-compose.yml`, который запустит контейнеры MySQL и phpMyAdmin:

yaml

 Copy code

```
version: '3'

services:
  mysql:
    image: mysql:latest
    environment:
      MYSQL_ROOT_PASSWORD: my-secret-pw
    ports:
      - "3306:3306"

  phpmyadmin:
    image: phpmyadmin/phpmyadmin:latest
    links:
      - mysql
    ports:
      - "8080:80"
    environment:
      PMA_HOST: mysql
      PMA_PORT: 3306
```

version: '3'

services:

mysql:

image: mysql:latest

environment:

MYSQL\_ROOT\_PASSWORD: my-secret-pw

ports:

- "3306:3306"

phpmyadmin:

image: phpmyadmin/phpmyadmin:latest

links:

- mysql

ports:

- "8080:80"

environment:

PMA\_HOST: mysql

PMA\_PORT: 3306



You


ошибка была не в том что ты использовал другое имя а не мое, мое имя ты не мог знать бро. однако ты ранее не указывал что надо запустить приложение docker-compose и передать ему файл конфигурации, ранее ты писал что якобы надо просто стартовать файл конфигурации



ChatGPT

Прошу прощения за путаницу и недоразумение. Действительно, для запуска файла конфигурации ``docker-compose.yml`` нужно использовать команду ``docker-compose``. Поэтому правильная команда для запуска с вашим файлом конфигурации будет следующей:

bash

 Copy code

```
docker-compose -f docker-comp1.yml up -d
```

Благодарю за ваше терпение и понимание. Если у вас есть еще вопросы или что-то непонятно, не стесняйтесь спрашивать.





```
daniilkonef@MyUbuntu:~/homework7$ docker-compose -f docker-comp1.yml up -d
Creating network "homework7_default" with the default driver
Pulling phpmyadmin (phpmyadmin/phpmyadmin:latest)...
latest: Pulling from phpmyadmin/phpmyadmin
faef57eae888: Pull complete
989a1d6c052e: Pull complete
0705c9c2f22d: Downloading [=====> ] 98.35MB/104.3MB
621478e043ce: Download complete
98246dcca987: Download complete
bfed8c155cb6: Download complete
7a7c2e908867: Download complete
d176994b625c: Download complete
2d8ace6a2716: Download complete
c70df516383c: Download complete
15e1b44fe4c7: Download complete
65e50d44e95a: Download complete
77f68910bc0a: Download complete
605dd3a6e332: Download complete
99ce27188f07: Download complete
74d64e32c5d5: Download complete
ef5fc9928b9f: Download complete
163f3256e112: Download complete
```

```
163f3256e112: Pull complete
Digest: sha256:67ba2550fd004399ab0b95b64021a88ea544011e566a9a1995180a3decb6410d
Status: Downloaded newer image for phpmyadmin/phpmyadmin:latest
Creating homework7_mysql_1 ... done
Creating homework7_phpmyadmin_1 ... done
daniilkonef@MyUbuntu:~/homework7$
```



You

команда `docker-compose -f docker-comp1.yml up -d` успешно выполнилась. что теперь дальше делать чтобы запустить phpmyadmin?



ChatGPT

После успешного выполнения команды `docker-compose -f docker-comp1.yml up -d`, вы можете открыть phpMyAdmin в веб-браузере, перейдя по адресу <http://localhost:8081> (или другому порту, если вы изменяли настройки в файле `docker-compose.yml`).

1. Откройте ваш веб-браузер.
2. Введите адрес <http://localhost:8081> и нажмите Enter.

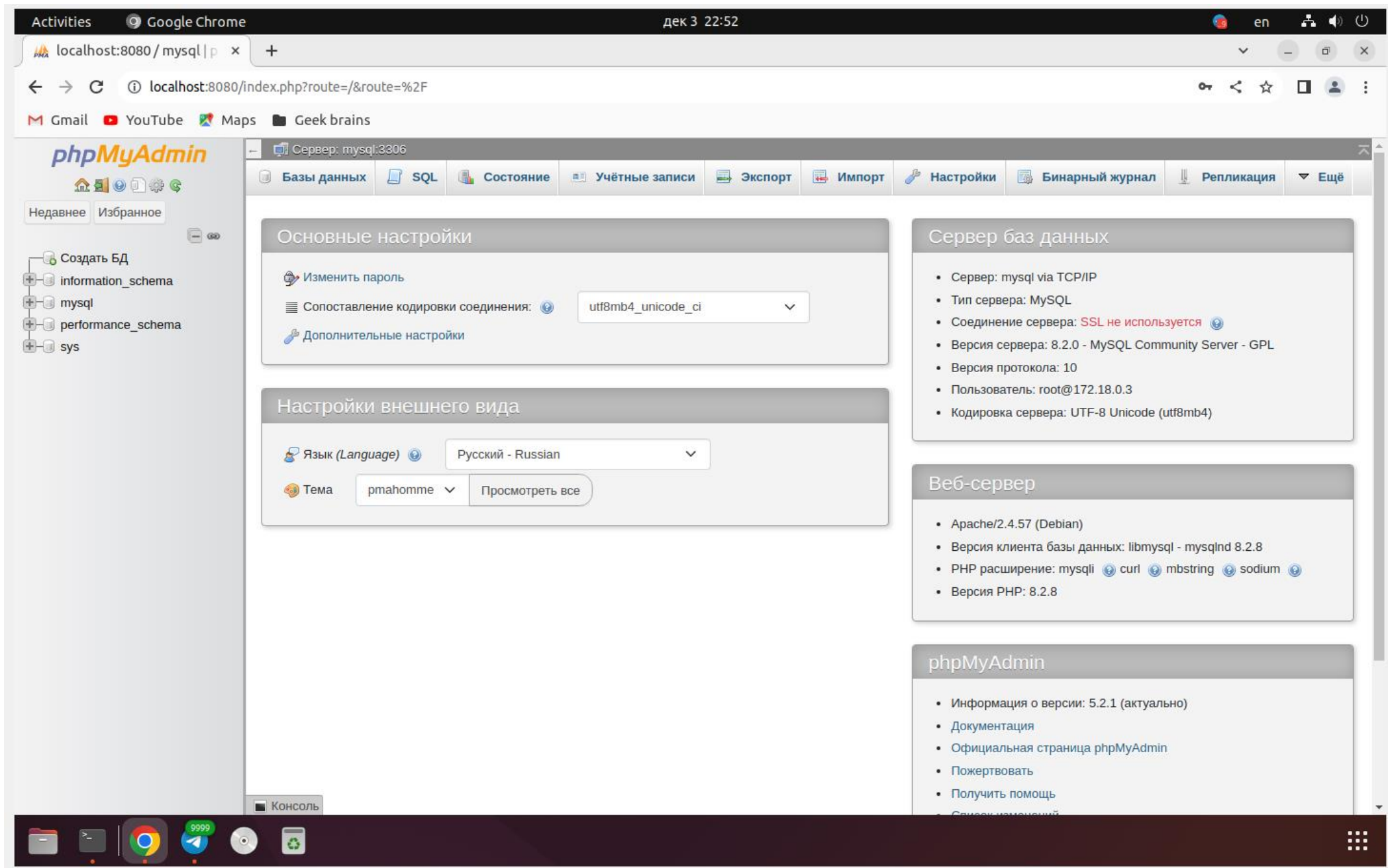
Вы должны увидеть страницу входа в phpMyAdmin. Войдите с использованием учетных данных, которые вы указали в файле ``docker-compose.yml``. По умолчанию, вы можете использовать следующие:

- Сервер: ``localhost`` (или IP-адрес вашего хоста)
- Имя пользователя: ``root``
- Пароль: ``my-secret-pw``

После входа вы должны быть подключены к вашему серверу MySQL с помощью phpMyAdmin, где вы можете управлять базами данных и выполнить различные операции.

Если у вас возникнут проблемы или у вас есть дополнительные вопросы, не стесняйтесь сообщить мне.





успешно