

# Семинар № 3

## Введение в Docker

Установите докер

<https://docs.docker.com/engine/install/ubuntu/>

### Задание 1:

Запустите контейнер. Установите внутри утилиту ip (пакет `iproute2`). Запустите контейнер с другим именем и хостнеймом. Читайте help к командам.

### Пример решения:

```
docker run -it ubuntu:22.10 bash
ps -aux
не работают:
ip a
ifconfig
```

```
apt update
apt install iproute2
ip a
exit
```

```
docker ps -a -показать, что контейнер остановлен и вот он - в списке
```

```
docker run -it -h GeekBrains --name GB_cont ubuntu:22.10 bash
ip a(не работает, потому что контейнер другой из того же имейджа)
exit
docker ps -a
```

### Запуск конкретного контейнера:

```
docker start <name of container>
docker ps -a
docker exec -it <name of container> bash
```

### любую команду запускать без входа в контейнер:

```
docker exec -it <name of container> COMMAND
остановить контейнер:
docker stop <name of container>
```

Обратите внимание на код возврата. Когда он не 0. Значит контейнер (или приложение в нём) некорректно завершилось.

удалить все, что накопилось

```
docker rm $(docker ps -a -q)
docker rmi $(docker images -aq)
docker rmi $(docker images -aq) -force
```

## Задание 2:

Теперь давайте добавим внешнее хранилище. Прежде чем его смонтировать, давайте создадим.

```
mkdir --parents /test/folder - хранилище на хосте
docker run -it -h GB --name gb-test2 -v
/test/folder:/otherway ubuntu:22.10
```

Тут показать, что директория примонтировалась в путь, который мы и указали, хотя изначально в контейнере и не было никаких данных об этом пути.

Теперь давайте добавим данных

```
echo "$HOSTNAME" >> /otherway/test.txt
etc
```

Ну и остановим контейнер. Следующий момент: данные доступны с локальной системы:

```
cat /test/folder/test.txt
```

Удалим контейнер

```
docker rm gb-test2
```

и пересоздадим, подмонтировав папку.

Все работает! Это и есть идеальный вариант хранения данных при использовании контейнеров: не хранить никаких важных данных внутри контейнера!

## Задание 3:

Необходимо запустить в контейнере базу данных MariaDB версии 10.10.2, используя инструкции, представленные в dockerhub: [https://hub.docker.com/\\_/mariadb](https://hub.docker.com/_/mariadb)

## Пример решения:

Это примерный вариант решения, главное – чтобы логика была верной.

```
docker run --name test-mariadb -e MARIADB_ROOT_PASSWORD=test123
-d mariadb:10.10.2
docker ps -проверить, что работает
docker exec -it test-mariadb bash
```

```
mariadb -u root -p
show databases;
```

#### Задание 4:

Далее необходимо запустить второй контейнер. Теперь запускаем контейнер phpmyadmin, используя инструкции: [https://hub.docker.com/\\_/phpmyadmin](https://hub.docker.com/_/phpmyadmin)

В данном случае, при запуске этого контейнера необходимо передать данные контейнера mariadb: имя контейнера, которое вы дали ранее.

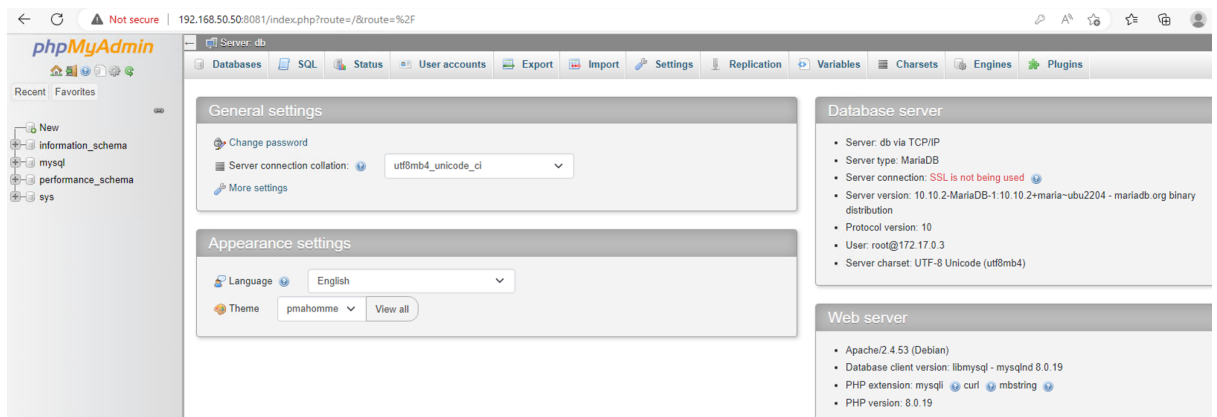
Почитать мануал про --link. Посмотреть какие переменные создаются в контейнере с phpmyadmin с помощью echo \${tab}

Почитать мануал про -p . Вспомнить проброс портов из сетей (или если не знаете что это, спросить у преподавателя).

#### Пример решения:

Это примерный вариант решения, главное – чтобы логика была верной.

```
docker run --name my-phpmyadmin -d --link test-mariadb:db -p
8081:80 phpmyadmin
```



## 1. Блок 9. Задача 5\*

#### Задание:

Удалить все, что было сделано:

```
docker rm test-mariadb
docker rm my-phpmyadmin
docker ps -a
```

Предлагается использовать контейнер с БД mariaDB. Необходимо создать новый контейнер, смонтировав вовнутрь папку таким образом, чтобы БД хранилась локально на машине с Linux.

Последовательность действий:

- Войти в контейнер
- Создать БД через интерфейс mariaDB
- Удалить контейнер
- Инициализировать новый
- Проверить, что созданные ранее данные доступны

Для решения задачи потребуются инструкции с сайта:  
[https://hub.docker.com/\\_/mariadb](https://hub.docker.com/_/mariadb)

Также потребуется этот мануал, если вдруг ученики не работали с БД ранее:  
<https://mariadb.com/kb/en/create-database/>

## Пример решения:

Это примерный вариант решения, главное – чтобы логика была верной.

```
mkdir /test-db
docker run --name test-mariadb -e MARIADB_ROOT_PASSWORD=test123
-v /test-db:/var/lib/mysql -d mariadb:10.10.2
docker ps -проверить, что работает
docker exec -it test-mariadb bash
mariadb -u root -p
CREATE DATABASE test;
show databases;
ll /test-db
docker stop test-mariadb
docker rm test-mariadb
docker ps -проверить, что контейнер удален
-создать заново контейнер с БД, проверить, что созданная ранее БД доступна и с
ней все ОК
```

## **Домашнее задание**

Условие:

**Формат сдачи ДЗ:** предоставить доказательства выполнения задания посредством ссылки на google-документ с правами на комментирование/редактирование.

Результатом работы будет: текст объяснения, логи выполнения, история команд и скриншоты (важно придерживаться такой последовательности).

В названии работы должны быть указаны ФИ, номер группы и номер урока.

**Задание:**

- 1) запустить контейнер с БД, отличной от mariaDB, используя инструкции на сайте: <https://hub.docker.com/>
- 2) \*добавить в контейнер hostname такой же, как hostname системы через переменную
- 3) заполнить БД данными через консоль
- 4) запустить phpmyadmin (в контейнере) и через веб проверить, что все введенные данные доступны