

30. Docker

- Настройка и запуск Docker images
- Работа с контейнерами
- Параметры Dockerfile и Docker Compose

Основной функционал

Запуск нового контейнера в интерактивном режиме, на основе ubuntu (bash) с заданием параметров `--name` и `--hostname`:

```
C:\>docker run -it --name myApp --hostname myApp ubuntu bash
```

Создание образа из текущего контейнера после выполнения каких либо работ, и отправка образа на репозиторий Docker-Hub:

```
C:\>docker commit myApp linar468/myApp_
```

Пример запуска tomcat сервера в фоновом режиме `-d`, и переопределением номера порта `-p`:

```
C:\>docker run -d -p 1234:8080 tomcat
```

Запуск ubuntu контейнера, с использованием своих данных на файловой системе, `-v`:

```
C:\>docker run -d -it -v /home/linar/mydir:/home/newdir ubuntu
```

Соединение контейнеров (установление связи)

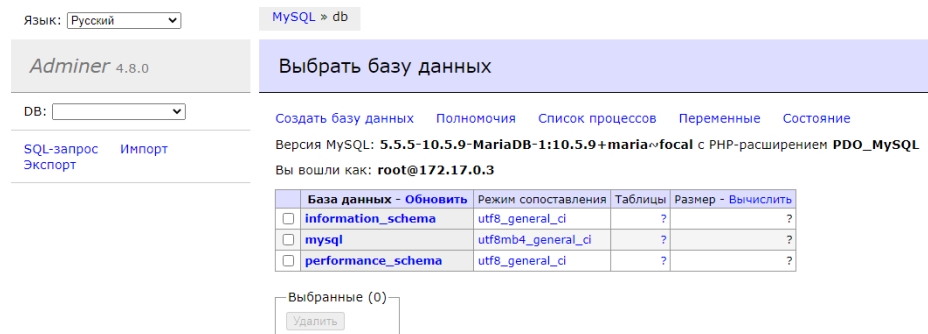
Создание MySQL соединения на основе mariadb:

```
C:\>docker run --name mysqlServer -e MYSQL_ROOT_PASSWORD=123456 -d mariadb
```

Запуск средства adminer для работы с БД, с указанием ссылки на mariadb `--link`:

```
C:\>docker run --link mysqlServer:db -p 8080:8080 adminer
```

Вход в adminer под пользователем root:



Язык: Русский MySQL > db

Adminer 4.8.0

DB:

[SQL-запрос](#) [Импорт](#) [Экспорт](#)

Выбор базы данных

[Создать базу данных](#) [Полномочия](#) [Список процессов](#) [Переменные](#) [Состояние](#)

Версия MySQL: 5.5.5-10.5.9-MariaDB-1:10.5.9+maria~focal с PHP-расширением PDO_MySQL

Вы вошли как: root@172.17.0.3

База данных - Обновить	Режим сопоставления	Таблицы	Размер - Вычислить
<input type="checkbox"/> information_schema	utf8_general_ci	?	?
<input type="checkbox"/> mysql	utf8mb4_general_ci	?	?
<input type="checkbox"/> performance_schema	utf8_general_ci	?	?

Выбранные (0)

[Удалить](#)

Настройка Dockerfile

Создание образа из Dockerfile, на основе Ubuntu, с указанием владельца (maintainer), выполнением shell-команд, и указанием точки входа (entrypoint). Структура файла:

```
FROM ubuntu:latest
MAINTAINER linar<ironman523@mail.ru>
RUN apt-get update && apt-get install -y cowsay
RUN ln -s /usr/games/cowsay /usr/bin/cowsay
ENTRYPOINT ["cowsay"]
```

Создание образа из Dockerfile, на основе Alpine, и установкой http-сервера. Конфигурация рабочей директории, подключение тома из файловой системы, и определение порта, запуск cmd команд. Структура файла:

```
FROM alpine:latest
RUN apk add npm && npm i -g http-server
VOLUME /home/server
WORKDIR /home/server
COPY ./ /home/server
EXPOSE 8080
CMD http-server
```

Настройка Docker Compose

Dockerfile(mariadb):

```
FROM mariadb:latest
```

Dockerfile(adminer):

```
FROM adminer:latest
```

Связывание с помощью docker-compose.yml:

```
version: '3.1'
services:
  db:
    build: ./db
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD:123456
  adminer:
    build: ./adminer
    restart: always
    ports:
      -6080:8080|
```

Создание MySQL соединения на основе docker-compose.yml:

```
version: '3'
services:
  mysql-dev:
    image: mysql:8.0.2
    environment:
      MYSQL_ROOT_PASSWORD:password
      MYSQL_DATABASE: blogapp
    ports:
      - "3308:3306"
  client:
    image: mysql:8.0.2
    depends-on:
      -mysql-dev
    command: mysql -u root -p password -h mysql-dev blogapp
```

Контейнеризация Java-приложения. Структура Dockerfile:

```
FROM openjdk:11
ADD target/docker-spring-boot.jar docker-spring-boot.jar
EXPOSE 8085
ENTRYPOINT ["java", "-jar", "docker-spring-boot.jar"]|
```