## Преамбула

**Docker**

Docker-контейнер - это Docker image (образ) который оживили. Собственно говоря Docker image - это то, из чего запускается любой Docker-container.

Каждому Docker image соответствует свой набор инструкций и файл с такими инструкция называется Dockerfile – без расширений и каких-либо точек. Из Dockerfile в дальнейшем собираются Docker images (образы).

В Dockerfile написаны инструкции по созданию образа. Каждая инструкция пишется с начала строки заглавными буквами, а после инструкции идут ее аргументы. Обработка инструкций происходит сверху вниз согласно тому порядку, как они написаны в файле. Простейший пример Dockerfile выглядит следующим образом:

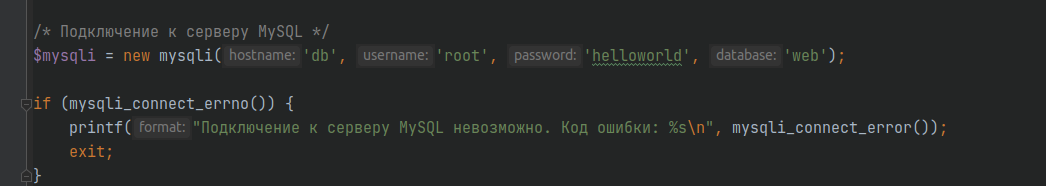
FROM ubuntu:20.04

COPY . /var/www/html

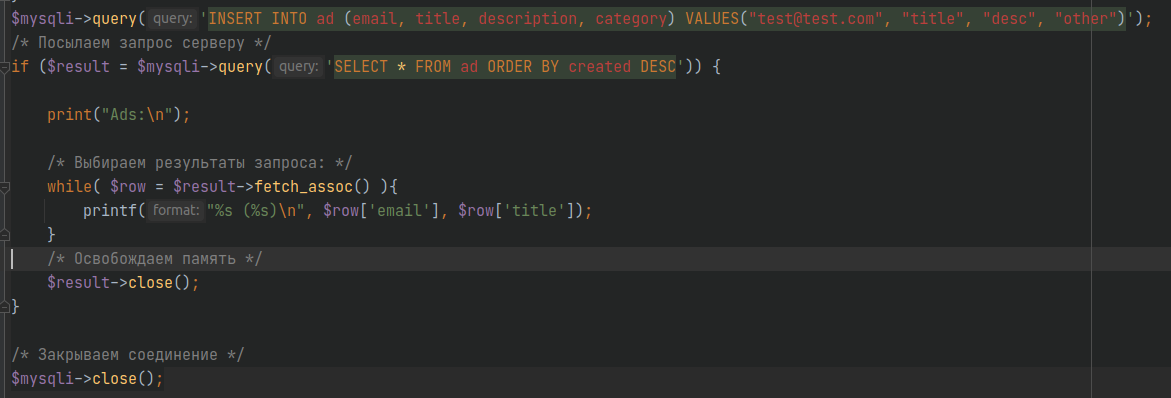
Новые слои в итоговом образе создаются только инструкциями FROM, RUN, COPY, ADD. Остальные инструкции что-то описывают, настраивают или общаются с Docker’ом говоря, например – открыть такой-то порт.

**PHP и Mysql**

Подключение к mysql

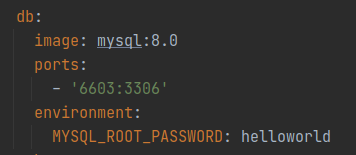
****

Запросы к mysql



## Требуется

1. Docker
   1. Добавить в docker-compose.yml новый контейнер с mysql

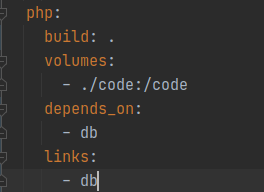


* 1. Создать файл Dockerfile (без расширения, так и называется) рядом с docker-compose.yml

Содержимое файла:



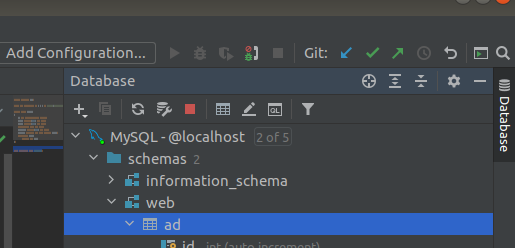
* 1. В файле docker-compose.yml изменить контейнер php. Вместо образа прописываем сборочный файл Dockerfile и добавляем link на db



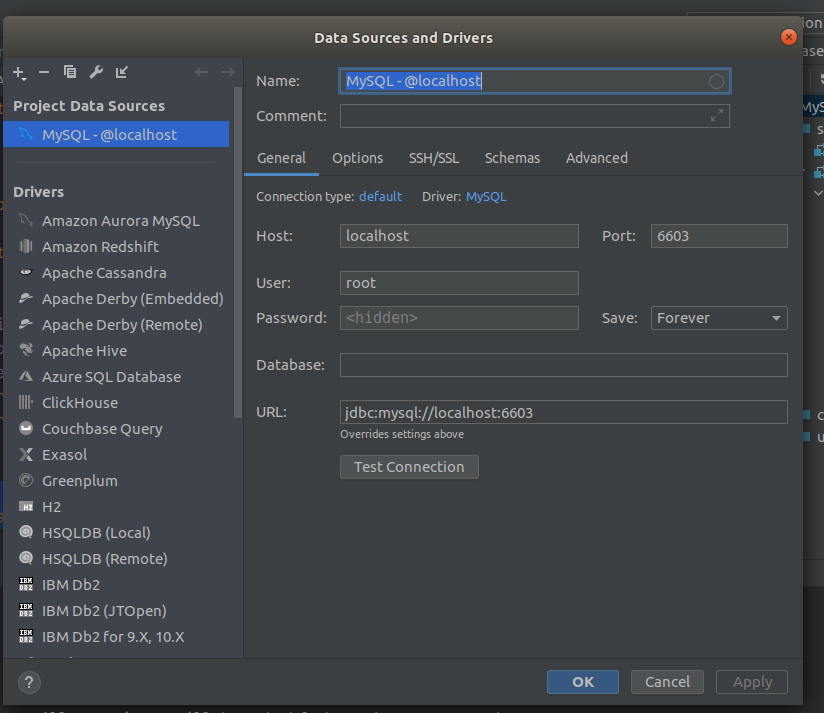
* 1. docker-compose stop

docker-compose build && docker-compose up -d

1. SQL
   1. В PHPStorm в разделе Database добавляете New - Data Source - Mysql.



Заполняете host - localhost, user - root, password - helloworld, port - 6603



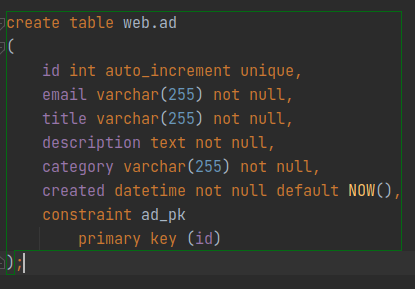
test connection - apply

* 1. По созданному подключению правая кнопка мыши - Jump to query console

1. В открывшемся окне консоли можно вводить и запускать чистый sql
   1. Создание базы



* 1. Создание таблицы объявлений



1. PHP  
   Сделать хранение объявлений в базе данных. Добавлять в базу и читать с базы.