

ЭТАПЫ РЕШЕНИЯ

ЭТАП 1. Создание контейнера («Сервер»)

1. Создаём образ с Apache и PHP 7.4 с фамилией в названии.

```
user@kali: ~/Desktop/SkillFactory_гепo/Вводный курс/d...
GNU nano 7.2 Dockerfile
FROM php:7.4-apache
RUN apt-get update && \
    docker-php-ext-install mysqli pdo pdo_mysql
```

```
(user@kali)-[~/.../Вводный курс/docker/Homework/Server]
$ docker build -t server/ilyukhin .
Sending build context to Docker daemon 4.096kB
Step 1/2 : FROM php:7.4-apache
7.4-apache: Pulling from library/php
a603fa5e3b41: Pull complete
```

```
(user@kali)-[~/.../Вводный курс/docker/Homework/Server]
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
server/ilyukhin	latest	e661b8720953	46 seconds ago	471MB
php	7.4-apache	20a3732f422b	15 months ago	453MB

2. Создаём образ для второго контейнера — публичного образа mysql:5.7. Далее будем называть этот контейнер «БД».

```
user@kali: ~/Desktop/SkillFactory_гепo/Вводный курс/d...
GNU nano 7.2 Dockerfile
FROM mysql:5.7
```

```
(user@kali)-[~/.../Вводный курс/docker/Homework/Database]
$ docker build -t db/ilyukhin .
Sending build context to Docker daemon 4.096kB
Step 1/1 : FROM mysql:5.7
5.7: Pulling from library/mysql
20e4dcae4c69: Pull complete
```

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
server/ilyukhin	latest	e661b8720953	9 minutes ago	471MB
mysql	5.7	5107333e08a8	2 months ago	501MB
db/ilyukhin	latest	5107333e08a8	2 months ago	501MB
php	7.4-apache	20a3732f422b	15 months ago	453MB

ЭТАП 2. Создание образа для второго контейнера («БД»)

Создаём базу данных MySQL и наполним её в контейнере «БД» с помощью файла `init.sql`.

```
GNU nano 7.2                               init.sql
CREATE DATABASE IF NOT EXISTS reg1;
CREATE USER IF NOT EXISTS 'user' @ '%' IDENTIFIED BY 'password';
GRANT SELECT, UPDATE, INSERT ON reg1.* TO 'user' @ '%';
FLUSH PRIVILEGES;
USE reg1;
DROP TABLE IF EXISTS regions;
CREATE TABLE IF NOT EXISTS regions (
  region VARCHAR(100) NOT NULL,
  maincity VARCHAR(50) NOT NULL,
  population INT (10) NOT NULL
);
INSERT INTO regions (region, maincity, population) VALUES
('Primorskiy kraj', 'Vladivostok', 1845),
('Khabarovskiy kraj', 'Khabarovsk', 1291),
('Amurskaja oblast', 'Blagovestchensk', 767);
```

ЭТАП 3. Настройка двух контейнеров в docker-compose (контейнер «БД»)

В `docker-compose.yml` для контейнера «БД» укажем следующие параметры:

1. Запуск скрипта `init.sql` при старте контейнера командой `--init-file`.
2. Создадим директорию `dbfile` и примонтируем её в контейнер в директорию `/var/lib/mysql`.
3. Примонтируем файл `init.sql` в контейнер в директорию `/data/application/init.sql`.
4. Пробросим порт 3306 контейнера на любой порт хостовой машины.

```
GNU nano 7.2                               docker-compose.yml *
version: '3.0'
services:
  db:
    image: 'db/ilyukhin'
    container_name: db
    command: '--init-file /data/application/init.sql'
    environment:
      MYSQL_ROOT_PASSWORD: 123456
    volumes:
      - './Database/dbfile:/var/lib/mysql'
      - './Database/init.sql:/data/application/init.sql'
    ports:
      - '3308:3306'
    restart: always
```

ЭТАП 4. Настройка двух контейнеров в docker-compose (контейнер «Сервер»)

В docker-compose контейнера «Сервер» укажем следующие параметры:

1. Пробросим порт 80 контейнера на любой порт хостовой машины.
2. Создадим папку `src` и примонтируем её в контейнер в директорию `/var/www/html`.
3. Внутри папки `src` создадим файл `index.php`.

```
server:
  image: 'server/ilyukhin'
  container_name: server
  ports:
    - '8888:80'
  volumes:
    - './Server/src:/var/www/html'
  depends_on:
    - db
```

```
GNU nano 7.2                                index.php *
<html lang="en">
<head>
  <title>Russian Far East Regions</title>
  <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
  <h1>Regions</h1>
  <table>
    <tr>
      <th>region</th>
      <th>maincity</th>
      <th>population</th>
    </tr>
    <?php
      $mysqli = new mysqli("db", "user", "password", "reg1");
      $result = $mysqli->query("SELECT * FROM regions");
      foreach ($result as $row) {
        echo "<tr><td>{$row['region']}</td><td>{$row['maincity']}</td><td>{$row['population']}</td></tr>";
      }
    </table>
    <?php phpinfo(); ?>
  </body>
</html>
```

ЭТАП 5. Вывод содержимого на веб-страницу

1. Контейнер «Сервер» с помощью скрипта `index.php` должен выводить содержимое базы данных MySQL контейнера «БД» на веб-страницу. Структура веб-страницы описывается в файле `index.php`.
2. На веб-страницу также выводится информация о версии PHP.

```
(user@kali)-[~/../SkillFactory_repo/Вводный курс/docker/Homework]
$ docker-compose up
Starting db ... done
Starting server ... done
Attaching to db, server
```

Regions

region	maincity	population
Primorskij kraj	Vladivostok	1845
Khabarovskij kraj	Khabarovsk	1291
Amurskaja oblast	Blagovestchensk	767

PHP Version 7.4.33



System	Linux a51b9bb804b4 6.5.0-kali3-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.6-1kali1 (2023-10-09) x86_64
Build Date	Nov 15 2022 06:03:12
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqld' '--with-password-argon2' '--with-sodium=shared' '--with-pdo-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-iconv' '--with-openssl' '--with-readline' '--with-zlib' '--disable-phpdbg' '--with-pear' '--with-libdir=lib/x86_64-linux-gnu' '--disable-cgi' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20190902
PHP Extension	20190902

```
(user@kali)-[~/.../SkillFactory_repo/Вводный курс/docker/Homework]
```

```
$ tree
```

Database

dbfile

```

├── auto.cnf
├── ca-key.pem
├── ca.pem
├── client-cert.pem
├── client-key.pem
├── ib_buffer_pool
├── ibdata1
├── ib_logfile0
├── ib_logfile1
├── ibtmp1
├── mysql [error opening dir]
├── mysql.sock -> /var/run/mysqld/mysqld.sock
├── performance_schema [error opening dir]
├── private_key.pem
├── public_key.pem
├── reg1 [error opening dir]
├── server-cert.pem
├── server-key.pem
├── sys [error opening dir]

```

```
├── Dockerfile
```

```
├── init.sql
```

```
├── docker-compose.yaml
```

```
├── Russian Far East Regions.pdf
```

Server

```
├── Dockerfile
```

src

```

├── index.php
├── style.css

```

```
└── Практическая работа по Docker (Описание).pdf
```