**МИНОБРНАУКИ РОССИИ**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»  
(БГТУ им. В.Г.Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №3

Тема: «Серверное программирование. Laravel»  
Дисциплина: «Технологии web-программирования»

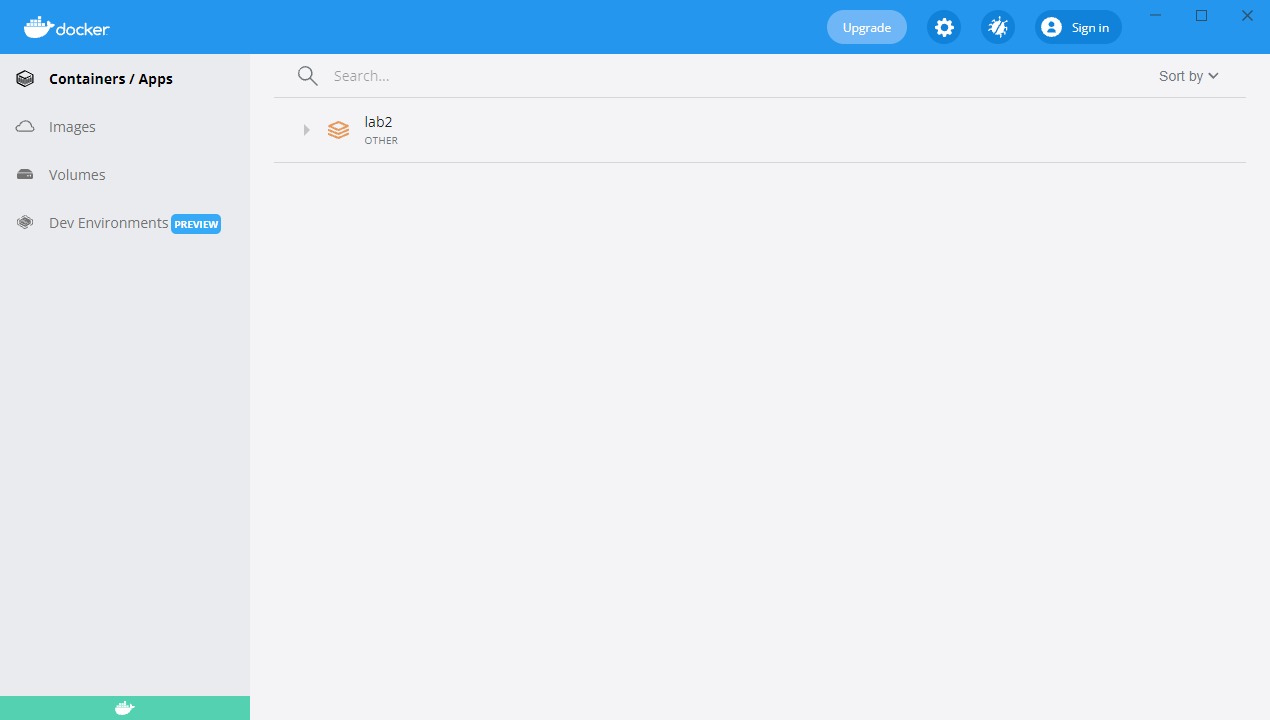
Выполнил: ст. группы ВТ-41  
Иост Евгений Вячеславович

Белгород 2021 г.

# Серверное программирование. Laravel

В ходе выполнения познакомились с основами backend разработки web-приложений. Научились писать скрипты. Познакомились с основами работы docker. Познакомились с фреймворком Laravel Lumen и научились разворачивать проект, производить его настройку. Научились работать с API в приложении Postman.

Установим сам docker с сайта docker.com



Затем добавим в проект следующие файлы из репозитория:

docker-compose.yml

version: '3'  
services:  
  
 app:  
 build:  
 context: .  
 dockerfile: docker/Dockerfile  
 image: digitalocean.com/php  
 restart: unless-stopped  
 tty: true  
 environment:  
 SERVICE\_NAME: app  
 SERVICE\_TAGS: dev  
 working\_dir: /var/www  
 volumes:  
 - ./:/var/www  
 - ./php/local.ini:/usr/local/etc/php/conf.d/local.ini  
 networks:  
 - app-network  
  
 webserver:  
 image: nginx:alpine  
 restart: unless-stopped  
 tty: true  
 ports:  
 - "80:80"  
 - "443:443"  
 volumes:  
 - ./:/var/www  
 - ./nginx/conf.d/:/etc/nginx/conf.d/  
 networks:  
 - app-network  
  
 db:  
 image: mysql:5.7  
 restart: unless-stopped  
 tty: true  
 ports:  
 - "3306:3306"  
 environment:  
 MYSQL\_ROOT\_PASSWORD: 'superpass321!'  
 MYSQL\_DATABASE: 'web-db'  
 MYSQL\_USER: 'user'  
 MYSQL\_PASSWORD: 'pass321!'  
 networks:  
 - app-network  
  
networks:  
 app-network:  
 driver: bridge

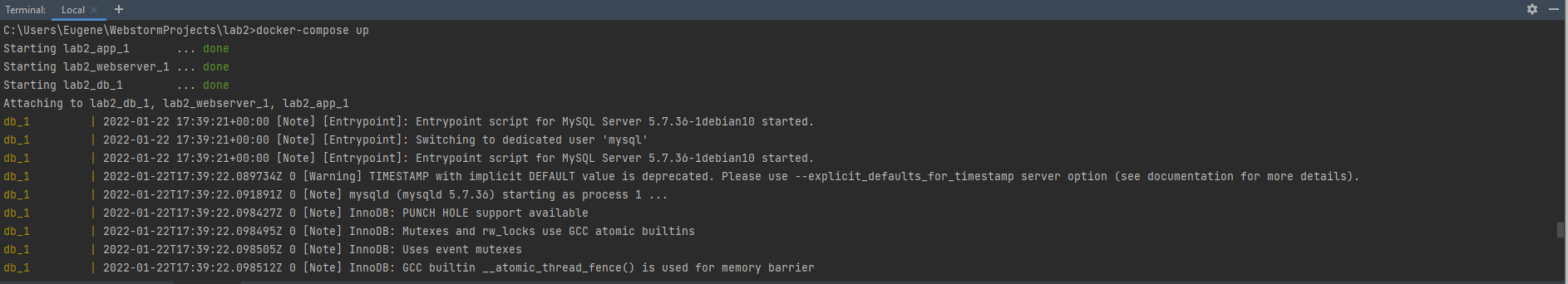
docker/Dockerfile

FROM php:7.4-fpm  
  
# Copy composer.lock and composer.json  
# COPY composer.json /var/www/  
  
# Set working directory  
WORKDIR /var/www  
  
# Install dependencies  
RUN apt-get update && apt-get install -y \  
 build-essential \  
 libpng-dev \  
 libjpeg62-turbo-dev \  
 libfreetype6-dev \  
 locales \  
 libzip-dev \  
 zip \  
 jpegoptim optipng pngquant gifsicle \  
 vim \  
 unzip \  
 git \  
 curl  
  
# Clear cache  
RUN apt-get clean && rm -rf /var/lib/apt/lists/\*  
  
# Install extensions  
RUN docker-php-ext-install pdo\_mysql zip exif pcntl  
RUN docker-php-ext-configure gd --with-freetype=/usr/include/ --with-jpeg=/usr/include/  
RUN docker-php-ext-install gd  
  
# Install composer  
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --filename=composer  
  
# Add user for laravel application  
RUN groupadd -g 1000 www  
RUN useradd -u 1000 -ms /bin/bash -g www www  
  
# Copy existing application directory contents  
COPY . /var/www  
  
# Copy existing application directory permissions  
COPY --chown=www:www . /var/www  
  
# Change current user to www  
USER www  
  
# Expose port 9000 and start php-fpm server  
EXPOSE 9000  
CMD ["php-fpm"]

nginx/conf.d/app.conf

server {  
 listen 80;  
 index index.php index.html;  
 error\_log /var/log/nginx/error.log;  
 access\_log /var/log/nginx/access.log;  
 root /var/www/public;  
  
 client\_max\_body\_size 128M;  
  
 location / {  
 try\_files $uri $uri/ /index.html;  
 }  
  
 location /api {  
 try\_files $uri $uri/ /index.php?$query\_string;  
 gzip\_static on;  
 }  
  
 location ~ \.php$ {  
 try\_files $uri =404;  
 fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;  
 fastcgi\_pass app:9000;  
 fastcgi\_index index.php;  
 include fastcgi\_params;  
 fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;  
 fastcgi\_param PATH\_INFO $fastcgi\_path\_info;  
 }  
}

Запустим docker при помощи команды docker-compose up



Далее установим Laravel. При переходе на <http://localhost/> получаем следующий результат:

