

# Инструкция по развертыванию сервиса на любом компьютере

Требуется ОС linux

Для начала установим python3, pip3, virtualenv, git

```
sudo apt install openssh-server  
sudo apt-get clean  
sudo apt-get install python3  
sudo apt-get install python3-pip  
sudo apt install python3-venv  
sudo apt install git  
sudo pip3 install virtualenv
```

Чтобы хранить данные, будем использовать [mysql](#)

```
sudo apt install mysql-server  
sudo mysql_secure_installation
```

При установке, потребуется создать пароль от учетной записи в БД, нажимать "N"

Далее заходим в mysql  
`sudo mysql`

**Прописываем данные параметры, чтобы можно было создать простой пароль "Test11"**

```
SET GLOBAL validate_password_length = 6;  
SET GLOBAL validate_password_number_count = 2;
```

**Создаем пароль**

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Test11';
```

**Создаем схему**

```
CREATE DATABASE new_schema;
```

**В задании есть ограничения на размер полей**

Но так как это ограничение касается только входящих данных, мы вправе хранить эти данные так, как захотим  
relatives - храним в MEDIUMTEXT, а это 16 МБ целых  
town,street,building,person\_name,birth\_date, gender - VARCHAR

primary key - пара из двух полей, так как из задания следует, что для каждой выгрузки import\_id свой.

```
CREATE TABLE new_schema.data (  
    import_id INT NOT NULL,  
    citizen_id INT NOT NULL,  
    town VARCHAR(512) NOT NULL,  
    street VARCHAR(512) NOT NULL,  
    building VARCHAR(512) NOT NULL,  
    apartment INT NOT NULL,  
    person_name VARCHAR(512) NOT NULL,  
    birth_date VARCHAR(100) NOT NULL,  
    gender VARCHAR(100) NOT NULL,  
    relatives MEDIUMTEXT NOT NULL,  
    PRIMARY KEY (import_id,citizen_id));
```

### Далее создадим дефолтную строку в нашей таблице.

В процессе решения задачи, каждый раз я должен генерить уникальный import\_id.

Поэтому создал дефолтный import\_id из одной строки, далее чтобы сгенерить новый

import\_id уже для настоящих выгрузок, я просто прибавляю к наибольшему import\_id из базы 1, тем самым получая новый import\_id

```
insert into new_schema.data values (0,0,'LA','Unknown','1b',0,'Default Name','01.01.1900','male','[]');
```

### Чтобы кириллица читалась, пропишем следующие запросы.

```
ALTER DATABASE new_schema CHARACTER SET = utf8mb4 COLLATE = utf8mb4_unicode_ci;  
ALTER TABLE new_schema.data CONVERT TO CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

**Далее в качестве референса бралась данная [статья](#) на medium**

**Установим nginx, supervisor**

```
sudo apt-get install nginx  
sudo apt-get install supervisor
```

**Клонирую свой репозиторий с кодом**

```
git clone https://github.com/mkhamzanov/backendschool.git
```

**Проходим в папку app**

1. Создаем virtualenv  
virtualenv -p python3 .env
2. Активируем  
source .env/bin/activate
3. Устанавливаем все необходимые внешние python-библиотеки (зави

симости)

```
pip3 install -r requirements.txt
```

4. Устанавливаем Flask (скорее всего он уже установлен)

```
pip3 install Flask
```

## Устанавливаем gunicorn, supervisor

```
pip3 install gunicorn
```

```
pip3 install supervisor
```

## Запустим

1. threads - отвечает за мультипоточность

```
gunicorn3 app:app --bind 127.0.0.1:5000 --threads=4 &
```

2. Проверим, запустим команду ниже, на экране должен появиться json с дефолтной строкой из БД

```
curl 127.0.0.1:5000/imports/0/citizens
```

## Далее проходим в данную директорию

```
cd /etc/supervisor/conf.d/
```

**Создаем файл app.conf с данным содержанием, заметим, что autostart=true - тот самый параметр, который будет запускать наш сервис автоматически.**

```
[program:app]
directory=/home/entrant/backendschool/app
command=/home/entrant/backendschool/app/.env/bin/gunicorn app:ap
p -b localhost:5000
autostart=true
autorestart=true
stderr_logfile=/home/entrant/backendschool/app/app.err.log
stdout_logfile=/home/entrant/backendschool/app/app.out.log
```

### Запускаем данные команды

```
sudo supervisorctl reread
sudo service supervisor restart
sudo supervisorctl status
```

### Далее открываем данный файл

```
sudo nano /etc/nginx/conf.d/virtual.conf
```

Добавляем следующее проху\_pass <http://0.0.0.0:8080> - новый адрес

```
server {
    listen      80;
    server_name your_public_dnsname_here;
```

```
        location / {  
            proxy_pass http://0.0.0.0:8080;  
        }  
    }
```

### **Запускаем команды**

```
sudo nginx -t  
sudo service nginx restart
```

### **Проверка, данный запрос должен выдать json с дефолтной строчкой из БД**

```
curl 0.0.0.0:8080/imports/0/citizens
```