Ubuntu20 + nginx + Django REST framework + sqlite 21.01.2022

Трудоемкость инструкции: около 20 минут

Сервер, развернутый по данной инструкции, предназначен для создания быстрых прототипов.

Данный сервер развернут с кодом с gihub и позволяет загружать свежие релизы непосредственно с репозитария github.

Подготовка репозитария github:
1. В инструкции используется репозитарий тестового сервера DRF https://github.com/vakhnin/start-Django-REST-framework. Необходимо сделать fork проекта на github и изменить название профиля /vakhnin/ на свое в ссылке https://github.com/vakhnin/start-Django-REST-framework/settings/keys

Подготовка DRF проекта (в тестовом примере https://github.com/vakhnin/start-Django-REST-framework необходимая подготовка уже проведена):
1. Создать файл requirements.txt

```
$ pip3 freeze > requirements.txt
```

2. Проверить файл tutorial/settings.py

```
ALLOWED_HOSTS = ['*']
```

Настройка серверерa:

Устанавливаем необходимые пакеты, создаем пользователя django, становимся пользователем django

```
$ sudo apt update
$ sudo apt upgrade -y
$ sudo apt install python3-venv git-core -y

$ sudo useradd -g www-data  -s /bin/bash -m django
$ sudo --login -u django
```

Создаем пару ключей под пользователем django, выводим публичный ключ в консоль, выделяем и копируем ключ

```
$ ssh-keygen
$ cat /home/django/.ssh/id_rsa.pub
```

Идем по URL и добавляем открытый ключ из предыдущего пункта

https://github.com/vakhnin/start-Django-REST-framework/settings/keys

Не забываем заменить в ссылке /vakhnin/ на название своего профиля на github

Пользователем django клонируем себе в папку код из github репозитария, создаем виртуальное окружение, активируем виртуальное окружение.
Уже в окружении обновляем pip, устанавливаем необходимые пакеты, выполняем миграцию БД, проверяем (при помощи curl, в данной методичке не описано) тестовый сервер Django и тестовый сервер gunicorn, деактивируем виртуальное окружение.
Выходим из пользователя django

```
$ cd /home/django/
$ git clone git@github.com:vakhnin/start-Django-REST-framework.git

$ cd start-Django-REST-framework/
$ python3 -m venv env
$ source env/bin/activate
```

```
(env) pip install -U pip
(env) pip3 install -r requirements.txt
(env) pip3 install gunicorn
(env) python3 manage.py migrate
(env) python3 manage.py runserver
(env) gunicorn --bind 127.0.0.1:8000 tutorial.wsgi
(env) python3 manage.py collectstatic
(env) deactivate
```

```
$ exit
```

Создаем файл

```
$ sudo nano /etc/systemd/system/gunicorn.socket
```

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
```

Создаем файл

```
$ sudo nano /etc/systemd/system/gunicorn.service
```

```
[Unit]
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
User=django
Group=www-data
WorkingDirectory=/home/django/start-Django-REST-framework
ExecStart=/home/django/start-Django-REST-framework/env/bin/gunicorn \
        --access-logfile - \
        --workers 5 \
        --bind unix:/run/gunicorn.sock \
        tutorial.wsgi:application

[Install]
WantedBy=multi-user.target
```

Запускаем gunicorn

```
$ sudo systemctl enable gunicorn.socket
$ sudo systemctl enable gunicorn.service
$ sudo systemctl start gunicorn
```

Устанавливаем и настраиваем nginx

```
$ sudo apt install nginx -y
$ sudo nano /etc/nginx/sites-available/default
```

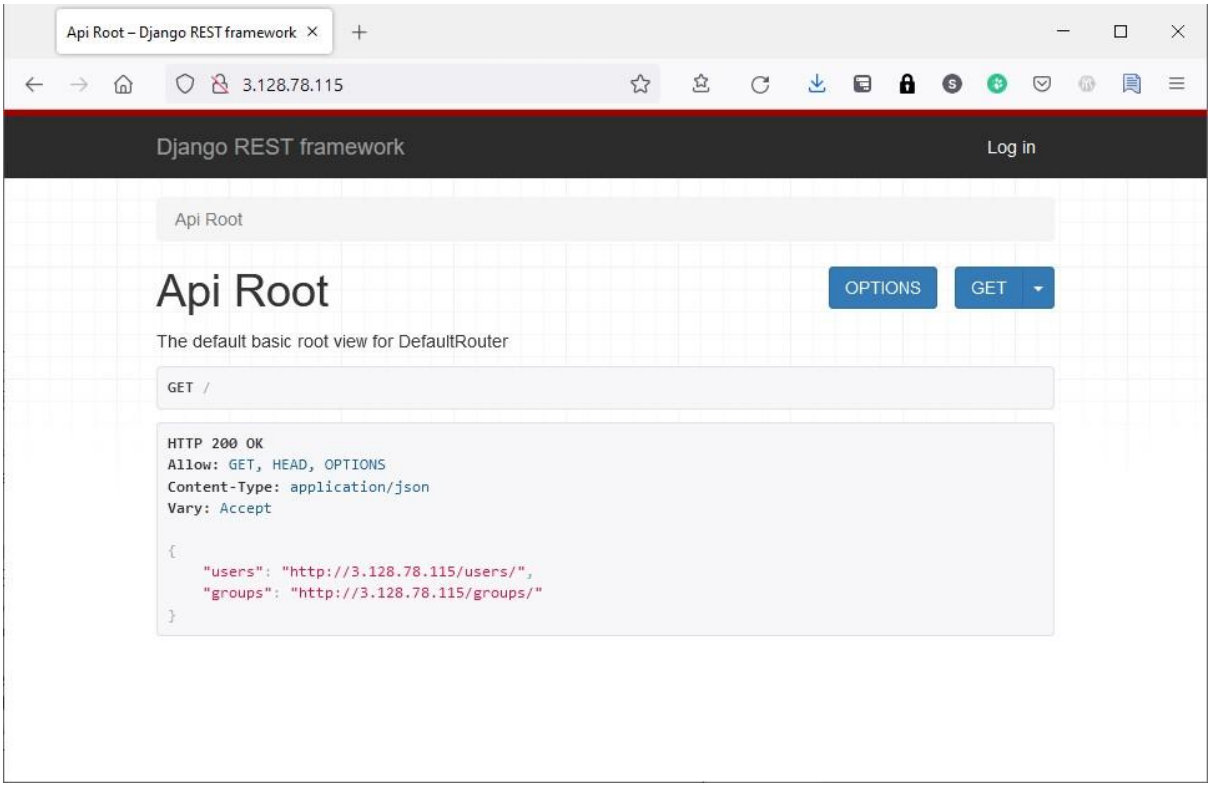
```
server {
    listen 80;
    server_name 127.0.0.1;

    location /static/ {
        root  /home/django/start-Django-REST-framework/;
    }

    location /media/ {
        root  /home/django/start-Django-REST-framework/;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/run/gunicorn.sock;
    }
}
```

```
$ sudo systemctl restart nginx
$ sudo systemctl reload nginx
```



Full success!
Scribbled by Vakh.