

## Create react app + django deploy

# Деплой create-react-app + django на Ubuntu server

## Минимальная настройка сервера перед установкой

1. Включить ufw firewall (опционально, но желательно)
2. Настройка ssh доступа по ключам (опционально, для удобства)
3. Создание не root пользователя для работы

```
1 | adduser --ingroup sudo <username>
```

4. Обновление пакетов и установка некоторых базовых утилит

```
1 | sudo apt update
2 | sudo apt upgrade
3 | sudo apt install vim screenfetch htop tmux # возможно, достаточно htop, а може
```

## Nginx

1. Установить нгинкс на сервер

```
1 | sudo apt install nginx
2 | sudo systemctl start nginx # запустить нгинкс
3 | sudo systemctl enable nginx # автоматически включать нгинкс после ребута
```

2. Конфигурация

Базовый конфиг нгинкса лежит в `/etc/nginx/nginx.conf`. Его трогать не надо.

Добавление своих конфигов лучше делать через директорию `/etc/nginx/conf.d/`, в случае одного домена

будет достаточно одного конфига, например `/etc/nginx/conf.d/default.conf`. Все `*.conf` из этой папки нгинкс добавит автоматически в базовый конфиг.

Вот актуальный конфиг сразу для всего, без лишних заумствований:

```
1 server {
2     server_name allday.rest www.allday.rest;
3
4     # Локейшн для фронтенда, просто проксируем на нгинкс всё
5     location / {
6         proxy_pass http://localhost:5000;
7         proxy_set_header Host $host;
8         proxy_set_header Server $host;
9         client_max_body_size 10M;
10    }
11
12    # Для запросов к вебсокетам
13    location ^~ /ws/ {
14        proxy_pass http://0.0.0.0:8001;
15        proxy_http_version 1.1;
16        proxy_set_header Upgrade $http_upgrade;
17        proxy_set_header Connection "upgrade";
18    }
19
20    # Нужно для того, чтобы джанга находила статику, иначе не пашет
21    location /static_back {
22        alias /home/user/backend/prod/static_back;
23    }
24
25    # Основные локейшны для джанги
26    location ^~ /api/v1/ {
27        proxy_pass http://0.0.0.0:8000/api/v1/;
28        proxy_set_header Host $host;
29        proxy_set_header Server $host;
30        client_max_body_size 10M;
31    }
32    location ^~ /admin/ {
33        proxy_pass http://0.0.0.0:8000/admin/;
34        proxy_set_header Host $host;
35        proxy_set_header Server $host;
36        client_max_body_size 10M;
37    }
38    location ^~ /login/ {
39        proxy_pass http://0.0.0.0:8000/login;
40        proxy_set_header Host $host;
41        proxy_set_header Server $host;
42        client_max_body_size 10M;
43    }
44    location ^~ /login {
45        proxy_pass http://0.0.0.0:8000/login;
46        proxy_set_header Host $host;
47        proxy_set_header Server $host;
48        client_max_body_size 10M;
49    }
50    location ^~ /logout {
51        proxy_pass http://0.0.0.0:8000/logout;
```

```

52     proxy_set_header Host $host;
53     proxy_set_header Server $host;
54     client_max_body_size 10M;
55 }
56 location ^~ /register/ {
57     proxy_pass http://0.0.0.0:8000/register;
58     proxy_set_header Host $host;
59     proxy_set_header Server $host;
60     client_max_body_size 10M;
61 }
62 location ^~ /register {
63     proxy_pass http://0.0.0.0:8000/register;
64     proxy_set_header Host $host;
65     proxy_set_header Server $host;
66     client_max_body_size 10M;
67 }
68
69 # Добавляется автоматически Цертботом
70 listen 443 ssl; # managed by Certbot
71 ssl_certificate /etc/letsencrypt/live/www.allday.rest/fullchain.pem; # manag
72 ssl_certificate_key /etc/letsencrypt/live/www.allday.rest/privkey.pem; # man
73 include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
74 ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
75 }
76
77 # Добавляется автоматически Цертботом
78 server {
79     if ($host = www.allday.rest) {
80         return 301 https://$host$request_uri;
81     } # managed by Certbot
82     server_name allday.rest www.allday.rest;
83     listen 80;
84     return 404; # managed by Certbot
85 }

```

## Letsencrypt/ssl/https

[Вот тут можно посмотреть, как скачать и пользоваться цертботом](#) 

1. Установить
2. Запустить что-то вроде `sudo certbot --nginx`
3. Указать в интерактиве, что просят, выбрать домены для сертификата

Certbot сам сгенерирует то что нужно в конфиге нгинкса.

## Postgresql (локально)

1. Ставим постгрес, врубаем

```
1 | sudo apt install postgresql postgresql-contrib
2 | sudo systemctl start postgresql
3 | sudo systemctl enable postgresql
```

2. Входим в `psql` и настраиваем базу:

```
1 | psql -U postgres # вход
2 |
3 | # \q - выйти из psql
4 | # \l - показать базы
5 | # \du - показать роли
6 |
7 | create role django with password 'adminlikeadmin';
8 | create database scientist_bd encoding 'UTF-8';
9 | grant all privileges on database scientist_bd to django;
10 | # может нужно будет разрешить роли django логиниться
11 | \q
```

3. Для настройки доступа к базе из докера, поменяем конфиги и ребутнём постгрес:

```
1 | sudo find / -type f -name postgresql.conf
2 | nano .../postgresql.conf
3 | # add line: listen_addresses = '*'
4 | # save
```

```
1 | sudo find / -type f -name pg_hba.conf
2 | nano .../pg_hba.conf
3 | # add line: host scientist_bd django 172.17.0.0/16 trust
4 | # save
```

[Инфа отсюда](#) 

## Docker (Установка и базовая настройка)

1, [Просто шаг 1 отсюда](#) 

2.

```
1 | sudo systemctl start docker
2 | sudo systemctl enable docker
```

## create-react-app in docker (nginx)

1. Нужно поставить ноду нужной версии

```
1 | sudo apt install nodejs npm
2 | sudo npm cache clean -f
3 | sudo npm install -g n
4 | sudo n 15.3.0
5 | sudo npm install -g npm@6.14.8
```

## 2. Первый запуск проекта с гита

```
1 | git clone https://github.com/FSWL98/scientists
2 | mv scientists frontend
3 | cd frontend
4 | chmod +x run.sh
5 | sudo ./run.sh
```

## 3. При обновлении кода в репозитории, для пересборки:

```
1 | cd frontend
2 | git pull # Можно включить в run.sh и стартовать только его
3 | sudo ./run.sh
```

# django in docker (python, gunicorn)

Не всё делал я, надо бы дополнить, опишу в общих чертах

## 1. первый запуск с гита

```
1 | git clone https://github.com/nupak/mastersite.git
2 | mv mastersite backend
3 | cd backend
4 | mkdir -p {static_back,media_back,static}
5 | python3 -m venv venv
6 | source venv/bin/activate
7 | pip install -r requirements.txt
8 | python manage.py collectstatic
9 | chmod +x run.sh
10 | sudo ./run.sh
```

## 2. после обновления

```
1 | cd backend
2 | . venv/bin/activate
3 | git pull
4 | pip install -r requirements.txt
```

```
5 | python manage.py collectstatic  
6 | sudo ./run.sh
```

Powered by [Wiki.js](#)