

Пользователь, именем которого выполняются действия, должен входить в группу sudoers. Далее предполагается, что имя пользователя — elenatteam.

Сервер должен иметь доступ в Интернет.

Перед началом выполнения инструкций файлы дистрибутива должны быть скопированы в домашнюю директорию пользователя, она же должна быть текущей директорией.

Python

```
sudo apt-get update

sudo apt-get install python3 // установка Python 3

python3 -V // проверка установленной версии Python 3
```

Redis

```
// установка Redis

sudo apt install redis-server

sudo nano /etc/redis/redis.conf

добавьте/отредактируйте параметр supervised:

supervised systemd

// перезапуск сервера Redis

sudo systemctl restart redis.service

// тестирование Redis

sudo systemctl status redis
```

PostgreSQL. Установка и создание базы данных

```
sudo apt-get install postgresql // Установка PostgreSQL

psql --version // проверка установленной версии PostgreSQL

sudo -u postgres psql postgres // Вход в клиент PostgreSQL
```

```
postgres=# CREATE ROLE elenatteam LOGIN PASSWORD elenatteam; // Создать
пользователя elenatteam с паролем elenatteam

postgres=# CREATE DATABASE elenatteam_db WITH OWNER elenatteam; // Создать
базу данных elenatteam_db

postgres=# GRANT ALL PRIVILEGES ON DATABASE "elenatteam_db" to elenatteam; //
Определить права доступа

postgres=# \du // Показать пользователей

postgres=# \l // Показать базы данных

postgres=# \q // Выйти из клиента

systemctl status postgresql.service // Проверить статус

sudo systemctl start postgresql.service // Запустить сервис

sudo systemctl enable postgresql.service // Добавить в автозагрузку
```

Бэкенд. Установка и подключение к базе данных

```
unzip elenatteam.zip // распаковать архив с файлами дистрибутива

cd elena_team_backend // Перейти в директорию бэкенда

sudo apt-get install python3-pip // Установка pip

pip install virtualenv // Установка virtualenv

python3 -m virtualenv env // Создание виртуального окружения в директории env

source env/bin/activate // Активировать виртуальное окружение

pip install -r requirements.txt // Установка пакетов проекта

cd elena_team_backend // Перейти в корень проекта

nano elena_team_backend/local_settings.py // Создать файл настроек local_settings.py

Содержимое файла local_settings.py:
```

```
-----
DEBUG = False
ALLOWED_HOSTS = ['127.0.0.1',]
DATABASES = {
'default': {
'ENGINE': 'django.db.backends.postgresql',
'NAME': 'elenatteam_db',
'USER': 'elenatteam',
'PASSWORD': 'elenatteam',
'HOST': 'localhost',
```

```
'PORT': '5432', } }
```

В параметре ALLOWED_HOSTS необходимо указать список доменов и/или IP-адресов, по которым будет осуществляться доступ к системе ElenaTeam

```
// Создать миграции
```

```
python manage.py makemigrations registration
```

```
python manage.py makemigrations kladr
```

```
python manage.py migrate // Применить миграции
```

```
python manage.py load_kladr_object_types // загрузить в БД данные типов адресных объектов КЛАДР
```

```
python manage.py load_kladr // загрузить в БД адресные объекты КЛАДР
```

```
python manage.py createsuperuser // Создать супер пользователя (username, email, password)
```

```
sudo ufw allow 8000 // Открыть порт 8000 для тестирования
```

```
python manage.py runserver 0.0.0.0:8000 // Тестирование работы Django. Нажать Ctrl+C для завершения тестирования
```

```
# supervisor и celery
```

```
sudo apt-get update
```

```
sudo apt-get install supervisor // установка supervisor
```

```
sudo service supervisor restart // проверка установки
```

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

```
// создание конфигурационного файла автозапуска celery
```

```
sudo nano /etc/supervisor/conf.d/elenateam-celery.conf
```

```
// добавьте следующее содержимое в конфигурационный файл:
```

```
[program:celery_elenateam]
directory=/home/elenateam/elena-team-backend
environment=DJANGO_SETTINGS_MODULE="elena_team_backend.settings"
command=/home/elenateam/elena-team-backend/env/bin/celery -A elena_team_backend
worker -l INFO
autostart=true
autorestart=true
```

```
// запустите celery
```

```
sudo supervisorctl start celery_elenateam
```

Gunicorn. Установка и настройка

```
pip install django gunicorn // Установка Gunicorn
```

```
gunicorn --bind 0.0.0.0:8000 config.wsgi // Тестирование работы Gunicorn. Нажать Ctrl+C для завершения тестирования
```

```
deactivate // Деактивировать виртуальное окружение
```

```
mkdir /home/elenateam/elena_team_backend/run // создать директорию для файла сокета соединения
```

```
chmod 777 /home/elenateam/elena_team_backend/run // задать права доступа для созданной директории
```

```
sudo nano /etc/systemd/system/gunicorn.socket // Создать файл сокета systemd для Gunicorn
```

Содержимое файла gunicorn.socket:

```
-----  
[Unit]  
Description=gunicorn socket  
  
[Socket]  
ListenStream=/home/elenateam/elena_team_backend/run/gunicorn.sock  
  
[Install]  
WantedBy=sockets.target  
-----
```

```
sudo nano /etc/systemd/system/gunicorn.service // Создать служебный файл systemd для Gunicorn
```

Содержимое файла gunicorn.service:

```
-----  
[Unit]  
Description=gunicorn daemon  
Requires=gunicorn.socket  
After=network.target  
  
[Service]  
User=elenateam  
Group=www-data  
WorkingDirectory=/home/elenateam/elena_team_backend  
  
ExecStart=/home/elenateam/elena_team_backend/env/bin/gunicorn --workers 3 --bind unix:/home/elenateam/elena_team_backend/run/gunicorn.sock config.wsgi:application
```

[Install]

WantedBy=multi-user.target

Параметр User - имя пользователя

Параметр WorkingDirectory - путь до каталога с файлом manage.py

Параметр ExecStart - путь до файла gunicorn в виртуальном окружении

sudo systemctl start gunicorn // Запустить сервис

sudo systemctl enable gunicorn // Добавить в автозагрузку

sudo systemctl status gunicorn // Проверить статус

sudo journalctl -u gunicorn // Проверка журнала процесса Gunicorn

file /run/gunicorn.sock // Проверить наличие файла gunicorn.sock в каталоге /run

systemd-analyze verify gunicorn.service // Проверка файла gunicorn.service на наличие ошибок

Если внесены изменения в файлы gunicorn.service или gunicorn.socket, необходимо выполнить команды:

sudo systemctl daemon-reload

sudo systemctl restart gunicorn

NGINX. Установка и настройка

sudo apt-get install nginx // Установка NGINX

nginx -v // проверка версии NGINX

sudo nano /etc/nginx/sites-available/elena-team // Создание файла настройки NGINX

Содержимое elena-team:

```
server {
listen 80;
server_name example.com;
location = /favicon.ico { access_log off; log_not_found off; }
location/ {
try_files $uri $uri/ /index.html; # перенаправлять все запросы на SPA приложение
фронта
root /home/elenateam/front; # путь до каталога фронта }
location /api/ {
include proxy_params;
proxy_pass http://unix:/home/elenateam/elena-team-backend/run/gunicorn.sock; } }
```

Параметр server_name - публичный домен, по которому будет осуществляться доступ к системе ElenaTeam

```
sudo ln -s /etc/nginx/sites-available/elena-team /etc/nginx/sites-enabled // Активировать файл настроек, создать ссылку в sites-enabled
```

```
sudo nginx -t // Протестируйте конфигурацию Nginx на ошибки синтаксиса
```

```
sudo systemctl restart nginx // перезапустить Nginx
```

```
sudo ufw delete allow 8000 // Закрыть порт 8000 для тестирования
```

```
sudo ufw allow 'Nginx Full'
```