

 endlessnights вчера в 19:46

Реклама

Запуск Django сайта на nginx + Gunicorn + SSL

Разработка веб-сайтов, Python, Django, Nginx

Из песочницы

Предисловие

Для написания этой статьи ушло очень много сил и времени. Я наткнулся на множество русскоязычных статей, но как я понял, - они все были клонами оригинальной статьи на Digital Ocean, а все потому, что все ошибки и неточности передаются с одного ресурса на другой.

Подготовка

У нас есть обычный VPS с ОС Ubuntu, и мы уже написали в PyCharm или в блокноте свой код, который нужно лишь опубликовать, привязать домен, установить сертификат и в путь.

Первым делом необходимо обновить список репозиторий и установить сразу же пакет n

```
apt get update
apt get-install nginx
```

Я решил хранить файлы сайта в каталоге: `/var/www/`. В данном случае перемещаемся в каталог `cd /var/www/` и создаем новый каталог `mkdir geekhero` и получаем такой путь: `/var/www/geekhero/`

Переходим в новый каталог `geekhero`: `cd geekhero` и создаем виртуальное окружение: `python3 -m venv geekhero_env`

Активируем виртуальное окружение: `source geekhero_env/bin/activate` и устанавливаем в него Django: `pip install Django` и сразу же ставим `pip install gunicorn`

Создаем проект: `django-admin startproject ghproj`

Далее нужно произвести все первичные миграции; для этого прописываем: `python manage.py makemigrations`, затем `python manage.py migrate`.

После этого создаем административную учетную запись: `python manage.py createsuperuser` и следуем инструкции.

Далее уже создаем applications, но так как вы читаете данную статью, то вы уже умеете все это делать.

Заходим в `Settings.py` и прописываем, если отсутствует:

```
import os — в заголовке, остальное в самом низу текстового файла:
```

```
STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
```

Настройка Gunicorn

Идем в каталог `/etc/systemd/system/` и создаем два файла: `gunicorn.service` и `gunicorn.socket`:

Содержимое файла `gunicorn.service`:

```
[Unit]
```

```
Description=gunicorn daemon
Requires=gunicorn.socket
After=network.target

[Service]
User=root
WorkingDirectory=/var/www/geekhero #путь до каталога с файлом manage.py
ExecStart=/var/www/geekhero/geekhero_env/bin/gunicorn --workers 5 --bind unix:/run/gunicorn.sock
#путь до файла gunicorn в виртуальном окружении

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

Содержимое файла `gunicorn.socket`:

```
[Unit]
Description=gunicorn socket

[Socket]
ListenStream=/run/gunicorn.sock

[Install]
WantedBy=sockets.target
```

Для проверки файла `gunicorn.service` на наличие ошибок:

```
systemd-analyze verify gunicorn.service
```

Настройка NGINX

Далее идем в каталог: </etc/nginx/sites-available/> и создаем файл `geekhero` (название вашего сайта) без расширения:

```
server {
    listen 80;
    server_name example.com;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /var/www/geekhero;          #путь до static каталога
    }

    location /media/ {
        root /var/www/geekhero;          #путь до media каталога
    }

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

Для того, чтобы создать символическую ссылку на файл в каталоге </etc/nginx/site-enabled/> необходимо ввести следующую команду:

```
sudo ln -s /etc/nginx/sites-available/geekhero /etc/nginx/sites-enabled/
```

При любых изменениях оригинального файла, ярлык из sites-enabled нужно удалять и пересоздавать заново командой выше или выполнять команду: `sudo systemctl restart nginx`

Финальный этап

Для проверки конфигурации nginx нужно ввести команду:

```
sudo nginx -t
```

`sudo nginx -t` Далее запускаем службу unicorn и создаем socket:

```
sudo systemctl enable unicorn  
sudo systemctl start unicorn
```

Для отключения выполняем команды:

```
sudo systemctl disable unicorn  
sudo systemctl stop unicorn
```

Также, эти обе команды пригодятся, если вы будете вносить какие-либо изменения в HTML или python файлы, чтобы обновить свой сайт, но помните, что, если вносите изменения в модели, то обязательно нужно сделать `python manage.py makemigrations` <app> и `migrate <app>` из каталога с проектом.

Для первичного запуска / полной остановки сервиса Unicorn:

```
service unicorn start / service unicorn stop
```

Чтобы посмотреть статус запущенного сервиса нужно ввести:

```
sudo systemctl status unicorn  
или  
sudo journalctl -u unicorn.socket  
(с последней командой правильный вывод такой: map 05 16:40:19 byfe systemd[1]: Listening on unicorn.socket)
```

Для проверки создания сокета, необходимо ввести команду:

```
file /run/unicorn.sock
```

Такой вывод считается правильным: `/run/unicorn.sock: socket`

Если внес какие-либо изменения в файл `unicorn.service` или `.socket`, необходимо выполнить команду:

```
systemctl daemon-reload
```

Если все нормально сработало, то можем запустить nginx:

```
sudo service nginx start
```

Получаем сертификат SSL для домена

Установим certbot от Let's Encrypt: `sudo apt-get install certbot python-certbot-nginx`

Произведем первичную настройку certbot: `sudo certbot certonly --nginx`

И наконец-то автоматически поправим конфигурацию nginx: `sudo certbot install --nginx`

Осталось только перезапустить сервис nginx: `sudo systemctl restart nginx`

Итог

В рамках этой статьи мы разобрали как вывести наш сайт в production, установив Django, Gunicorn, nginx и даже certbot от Let's Encrypt.

Теги: gunicorn, django, разработка веб-сайтов, разработка веб-приложений, python, cms, nginx, ssl, lets

Хэбы: Разработка веб-сайтов, Python, Django, Nginx

+8

27

1,4k

22

Поделиться



4,0

Карма

12,0

Рейтинг

Bakhti Baymukhamedov @endlessnights

Сетевой Инженер

Сайт

Github

Telegram

ПОХОЖИЕ ПУБЛИКАЦИИ

6 марта 2020 в 10:47

Разработка веб-сайта на паскале (backend)

53

24,7k

77

191

19 февраля 2019 в 22:02

Как быстро написать веб-сайт или веб-приложение и не увязнуть в сборщиках

19

11,6k

123

9

6 сентября 2018 в 10:45

Различия между компиляцией веб-сайта и веб-приложения

8

6,7k

46

0

КУРСЫ



Веб-дизайнер

16 марта 2021 • 8 месяцев • 83 000 ₽ • GeekBrains



Python QA Engineer

16 марта 2021 • 4 месяца • 60 000 ₽ • OTUS



Python-разработчик с нуля

22 марта 2021 • 7 месяцев • 69 900 ₽ • Нетология



Веб-дизайн

22 марта 2021 • 6 недель • 15 000 ₽ • Loftschool



Python для работы с данными

26 марта 2021 • 4 месяца • 31 500 ₽ • Нетология

[Больше курсов на Хабр Карьере](#)

Реклама

**Сверхмощный фонарь Quatro**

Скидка 60%. Нож в подарок. Ликвидация склада! Необходимые устройства для выживания.

comover.ru * ООО "Инструмент" ОГРН 1127746305500 ИНН

Комментарии 22

TheGodfather вчера в 19:50

-1

Только один вопрос — почему без контейнеров?

**Tanner** вчера в 21:09

+4

Контейнеры не нужны.

lampslave сегодня в 02:02

+1

Контейнеры — ладно, но хоть virtualenv-то надо, там же не один проект будет крутиться наверняка.

**Tanner** сегодня в 02:21

0

Вы правы, virtualenv и ruenv – обязательно. Даже если проект будет один, зависимости надо фиксировать.

**suffix_ixbt** вчера в 20:23

0

Прошу прощения — Вы правда считаете что установкой certbot начинается и заканчивается настройка ssl в nginx ?

**endlessnights** вчера в 22:34

0

Здравствуйте, в данной статье я рассмотрел полный порядок от начала и до конца, как быстро и просто, в том числе заполнить SSL-сертификат силами let's encrypt — файлы сертификаты автоматически создаются для домена и последующей командой в файл конфигурации nginx вносятся изменения для работы домена с https:

```
listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}server {
    if ($host = example.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name example.com;
    return 404; # managed by Certbot
```

**Tanner** вчера в 21:20

+1

Плохо в смысле безопасности.

Сейчас в systemd есть пользовательские юниты. Использовать их очень просто: нужно вместо `/etc/systemd/system` использовать `~/.config/systemd/user`, а к вызову `systemctl` добавлять ключ `--user`.

Когда пользовательских юнитов не было, принято было подменять пользователя для запуска unicorn/uwsgi/et c. вот таким образом:

```
[Service]
User=www
Group=www
```

Пускать Django из-под рута – это варварство.



endlessnights вчера в 22:35

0

Все понял, буду пробовать и так делать.

YaroslavMik вчера в 22:37

+1

Слабовато. Не безопасно. Устаревше. Пора к ASGI привыкать.



Magikan вчера в 23:40

0

я конечно не спец в 3й джанге (давно сменил стек), но небольшой эксперимент все же проводил. и производительность джанго под asgi у меня проседала на 30% относительно wsgi варианта. а где обещанные бонусы ?))



Tanner сегодня в 01:01

0

На ASGI придётся переходить, если понадобится вебсокет.



Magikan сегодня в 01:09

0

мой опыт показывает, что вебсокеты делаются на чем угодно aio* за 5 минут без джанги. да и зачем джанго для пуш уведомлений? или вы апи фигачите через сокеты...?



Tanner сегодня в 01:34

0

1. «Что угодно aio*» – это плюс один фреймворк, то есть для собственного резюме – здорово, а для предприятия – не то чтобы.
2. Лично я, скажу честно, не горю желанием перейти на асинхронное программирование, и не встречал ещё таких питонистов, которые бы мечтали об этом. Я видел, как самые простые вещи с `async/await` превращаются в инкубатор неотлаживаемых багов, и не хочу быть крайним в разгребании этих авгиевых конюшеней.
3. Что плохого в API через вебсокеты?



andreymal сегодня в 01:53

0

Я видел, как самые простые вещи с `async/await` превращаются в инкубатор неотлаживаемых багов


Вот все вокруг почему-то об этом говорят, а я сколько ни программировал асинхронщину — почему-то ни разу не сталкивался с какими-то специфическими проблемами



Tanner сегодня в 02:24


0

А что вы делали с неспецифическими проблемами, когда отладчики в 3.4 и 3.5 ещё не умели останавливаться в соседнем треде?

 andreymal сегодня в 02:26


0

А я отладчиками не пользуюсь) Как-то ни разу не возникало необходимости

 Magikan вчера в 23:34


-1

я что-то не понял. а что вам дает unicorn в systemd с таким конфигом, кроме автостарта при рестарте виртуалки? supervisord не осилили ?)

 andreymal вчера в 23:43

+1


А зачем ставить лишнюю сущность, если systemd и так есть по умолчанию почти в любом линуксе? А ещё supervisord всё равно будет запускаться через тот же самый systemd)

 Magikan вчера в 23:46

-1

вы все верно написали, но не по делу.

супервизор решает одну оч простую задачу — рестарт приложения при краше. именно про это и был мой вопрос тк конфиг автора для systemd этого не делает. и если приложение ляжет — само не поднимется

 arheops сегодня в 00:42

0

А systemd тоже умеет в рестарт при краше.

Достаточно добавить что-то вроде


```
StartLimitIntervalSec=500
```

```
StartLimitBurst=5
```

```
[Service]
```


```
Restart=on-failure
```

```
RestartSec=5s
```

 Magikan сегодня в 01:00

0

а кто говорит, что он этого не умеет? я написал, что конфиг автора этого не делает

 Aeckmann сегодня в 02:34

0

Не делает.

Неясно, правда, зачем нужно ставить аж целый supervisord, если можно добавить одну строчку в конфиг systemd.

Только [полноправные пользователи](#) могут оставлять комментарии. [Войдите](#), пожалуйста.

САМОЕ ЧИТАЕМОЕ

[Сутки](#)[Неделя](#)[Месяц](#)

Заключённый использовал одиночную камеру для изучения математики. Сегодня он решает самые трудные уравнения в мире

+93

42,8k

129

86

Почему у авокадо большая косточка

+100	26,2k	51	60
------	-------	----	----

«Группа смерти» изнутри: люди, которые играют в опасные игры

+257	47,8k	211	281
------	-------	-----	-----

Судьба предателя, угнавшего новейший МиГ-25 в Японию

+97	925k	132	2737
-----	------	-----	------

Куда же плывут облака? Подбиваем итоги опроса и даём аналитику

[Мегапост](#)

Ваш аккаунт

[Войти](#)[Регистрация](#)

Разделы

[Публикации](#)[Новости](#)[Хабы](#)[Компании](#)[Пользователи](#)[Песочница](#)

Информация

[Устройство сайта](#)[Для авторов](#)[Для компаний](#)[Документы](#)[Соглашение](#)[Конфиденциальность](#)

Услуги

[Реклама](#)[Тарифы](#)[Контент](#)[Семинары](#)[Мегaproекты](#)[Мерч](#)

© 2006 – 2021 «Habr»

[Настройка языка](#)[О сайте](#)[Служба поддержки](#)[Мобильная версия](#)