Django 2.1 & Python | The Ultimate Web Development Bootcamp

udemy.com/the-ultimate-beginners-guide-to-django-django-2-python-web-dev-website/learn/v4/questions/5451068

RD

How I got gunicorn and nginx working, including finding portfolio.sock & successfully running site w/o :8000

Raymond · 7 months ago
Sorry this isn't a question, but more of a process of how I managed to finally get gunicorn/nginx working.

I've been going over videos 48-53 of the VPS section and noticed some slight changes from when the video was recorded to DigitalOcean updating their servers. Here are some updates/modifications for getting gunicorn/nginx code to work right.

Following the Digital Ocean Tutorial and cross-referencing with details in the video helped me to finally get the desired result of checking my website without :8000.

https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu-18-04

Here's my breakdown in 7 steps:

....

1. Create a gunicorn.socket file before the gunicorn.service file (not in vid, see tutorial):

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

```
//

[Unit]

Description=gunicorn socket

[Socket]

ListenStream=/run/portfolio.sock

[Install]

WantedBy=sockets.target
```

....

....

2. Create the gunicorn service with the 2nd parameter added to the Unit metadata:

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

```
//

[Unit]

Description=gunicorn daemon

Requires=gunicorn.socket

After=network.target
```

```
[Service]

User=djangodeploy

Group=www-data

WorkingDirectory=/home/djangodeploy/portfolio-project

ExecStart=/home/djangodeploy/myvenv/bin/gunicorn --access-logfile - --workers 3 --bind


unix:/home/djangodeploy/portfolio-project/portfolio.sock portfolio.wsgi:application


[Install]

WantedBy=multi-user.target
```

....

....

3. Run gunicorn.socket systemctl code as stated in video/blog:

```
//

sudo systemctl start gunicorn.socket

sudo systemctl enable gunicorn.socket

sudo systemctl status gunicorn.socket
```

**** Note** the portfolio.sock file **WILL** be created, but **NOT** in the portfolio-projects directory. Instead it will be created in your **home/run directory.**

....

....

4. Nginx wasn't installed on my Ubuntu 18.04 Digital Ocean server.

Luckily they have a step-by-step guide for installing it:

https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04

I went up to step 4 just to get it up and running. Of course you can keep going to get the most out of your nginx installation and setup.

....

....

5. Setting up your nginx portfolio file:

```
$ sudo nano /etc/nginx/sites-available/portfolio

//

server {

    listen 80;

    server_name 'ip_address_goes_here';


    location = /favicon.ico { access_log off; log_not_found off; }

    location /static/ {

        root /home/djangodeploy/portfolio-project;

    }


    location / {

        include proxy_params;
        proxy_pass http://unix:/run/portfolio.sock;

    }

}
```

- Please note the last line for proxy_pass, the default folder has changed to /run/portfolio.sock, and not in the djangodeploy/portfolio-project if your .sock file was created in the /run directory (if you're trying to get this to work from scratch after October 2018, my guess is your .sock file will be in the /run directory).

....

....

6. Continue to follow the video/blog with these commands:

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

sudo nginx -t

sudo systemctl restart nginx


sudo ufw delete allow 8000

sudo ufw allow 'Nginx Full'
```

....

- Now when you try your IP address in the browser, it should work. However for me, it STILL didn't work due to the .socket issue.

I got this error:

```
[error] 21260#21260: *1 connect() to unix:/run/portfolio.sock failed (111: Connection refused)
```

This is why it's important to create

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

before gunicorn.service. And if you do, but just copy it from the blog, make sure your parameter inside gunicorn.socket is like this:

```
[Socket]

ListenStream=/run/portfolio.sock
```

I had it on the default 'gunicorn.sock' which is why the 111: Connection refused error kept persisting.

....

7. Luckily Digital Ocean's Troubleshooting at the bottom of their tutorial covers this problem and how to fix the rest of it up:

https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu-18-04

Django Is Displaying: "could not connect to server: Connection refused"

First try these 3 commands:

```
$ sudo systemctl status postgresql

$ sudo systemctl start postgresql

$ sudo systemctl enable postgresql
```

Then these:

```
$ sudo systemctl restart gunicorn
```

```
$ sudo systemctl daemon-reload
```

```
$ sudo systemctl restart gunicorn.socket gunicorn.service
```

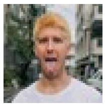
And finally for good measure:

```
$ sudo nginx -t && sudo systemctl restart nginx
```

....

- With all this tweaked, I was able to finally get the IP address running without :8000 just like Nick did in the video.

Hope this helps!



Nick — Instructor · 7 months ago
Thanks!

HS