

## Install Nginx, remove default site, create your site

Pre:

Deploy flask app Digital Ocean with Nginx / Gunicorn Ubuntu 16.04.3 LTS is done:

<https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-gunicorn-and-nginx-on-ubuntu-16-04>

## Install Nginx, remove default site, create your site

Deploying Flask Apps to an Ubuntu Server:

<https://www.youtube.com/watch?v=kDRRtPO0YPA>

sudo apt-get update

Python 2, sudo apt-get install python-pip python-dev nginx (or just install Nginx)

Python 3, sudo apt-get install python3-pip python3-dev nginx

```
t@ubuntu-512mb-fra1-01:~$ sudo apt-get install python-dev python-pip nginx
Reading package lists... Done
Building dependency tree
```

Remove default Nginx (I had one more folder to remove)

```
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-enabled# rm webprod
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-enabled# rm default
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-enabled# cd
root@ubuntu-512mb-fra1-01:~# cd /etc/nginx/sites-available/
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# ls
back_default  default  webprod
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# rm default
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# rm webprod
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# ls
back_default
```

Create a new file for you flask app

```
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# sudo touch flask_settings
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# ls
back_default  flask settings
```

Edit flask settings:

```
base.html  about.html  flask_settings x  index.html  pdf_files

1  server {
2
3      location / {
4          # where the request is coming from, nginx will hit here
5          # gunicorn is on port 8000
6          # passing the same host
7          proxy_pass http://127.0.0.1:8000;
8          proxy_set_header Host $host;
9          proxy_set_header X-Real-IP $remote_addr;
10
11      }
12
13 }
```

Make symlink

```
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-available# cd
root@ubuntu-512mb-fra1-01:~# sudo ln -s /etc/nginx/sites-available/flask_setting
/etc/nginx/sites-enabled/flask_settings
root@ubuntu-512mb-fra1-01:~# cd /etc/nginx/sites-enabled/
root@ubuntu-512mb-fra1-01:/etc/nginx/sites-enabled# ls
flask_settings
```

Test Nginx configuration before / after etc

```
root@ubuntu-512mb-fra1-01:~/web# sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu-512mb-fra1-01:~/web#
```

Python 2, sudo pip install virtualenv

Python 3, sudo pip3 install virtualenv

```
root@ubuntu-512mb-fra1-01:~# sudo pip install virtualenv
Collecting virtualenv
```

Create you Flask app:

I have used blueprints in \_\_init\_\_.py

```
def create_app(config_name):
    app = Flask(__name__, instance_relative_config=True)
    app.config.from_object(app_config[config_name]) #object config
    app.config.from_pyfile("config.conf") # instance config

    Bootstrap(app)
    db.init_app(app)
```

Code omitted

```

from .auth import auth as auth_blueprint
app.register_blueprint(auth_blueprint)

from .home import home as home_blueprint
app.register_blueprint(home_blueprint)

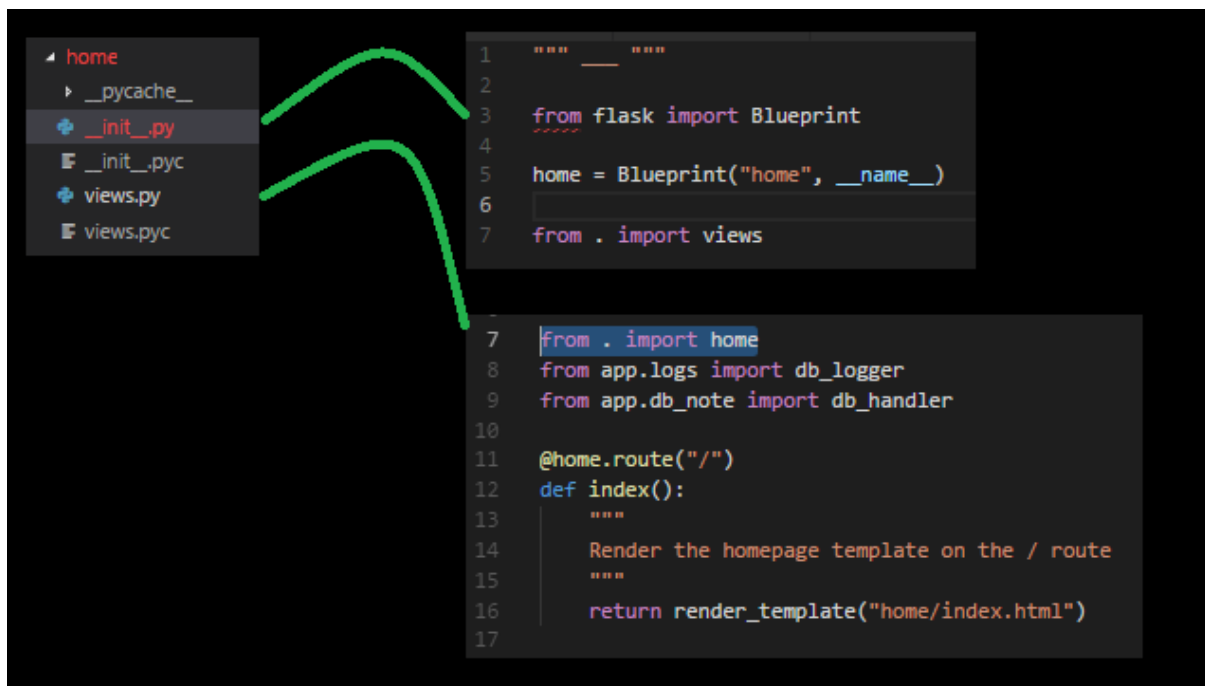
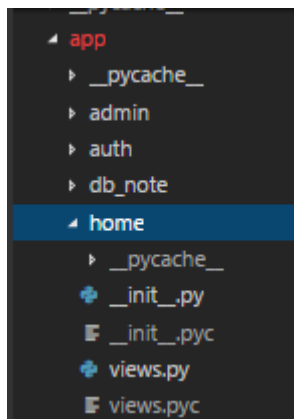
from .note import note as note_blueprint
app.register_blueprint(note_blueprint)

from .tech import tech as tech_blueprint
app.register_blueprint(tech_blueprint)

return app

```

Folder: \_\_init\_\_, views



The app is called from run.py for testing on Linux

Windows is the code that is commented out

```
root@ubuntu-512mb-fra1-01: ~/web
GNU nano 2.5.3 File: run.py

# run.py

import os
from flask import render_template

from app import create_app

dv = "development"
pr = "production"
[]
app = create_app(dv)

@app.errorhandler(404)
def page_not_found(error):
    return render_template("/error/400.html")

if __name__ == '__main__':
    # for linux
    app.run(host='0.0.0.0')
    # app.run(port=5100)
    # debug should come from config
```

The app is called from wsgi.py for production (gunicorn)

```
root@ubuntu-512mb-fra1-01: ~/web
GNU nano 2.5.3 File: wsgi.py

from app import create_app
from flask import render_template

dv = "development"
pr = "production"

app = create_app(dv)

@app.errorhandler(404)
def page_not_found(error):
    return render_template("/error/400.html")

if __name__ == "__main__":
    print("\n Success, system path jekl")
    app.run()
```

When you have tested and are ready to host / run the app:

Restart and test Nginx, load activate you env, start gunicorn

sudo service nginx status / stop / start

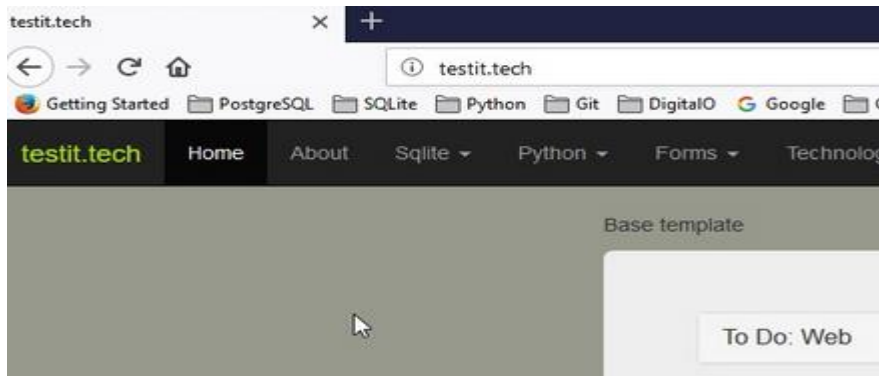
Here, gunicorn is started with 2 workers

```

ok ] Restarting nginx (via systemctl): nginx.service.
root@ubuntu-512mb-fra1-01:~/web# source webenv/bin/activate
webenv) root@ubuntu-512mb-fra1-01:~/web# gunicorn --workers=2 wsgi:app
2017-12-17 14:32:56 +0000] [1513] [INFO] Starting gunicorn 19.7.1
2017-12-17 14:32:56 +0000] [1513] [INFO] Listening at: http://127.0.0.1:8000 (1513)
2017-12-17 14:32:56 +0000] [1513] [INFO] Using worker: sync
2017-12-17 14:32:56 +0000] [1518] [INFO] Booting worker with pid: 1518
2017-12-17 14:32:56 +0000] [1519] [INFO] Booting worker with pid: 1519

```

Now visit your IP or domain



## Virtual envs

Start in test mode: `source webenv/bin/activate`

Stop test mode: `deactivate`

## Python

Start in test mode: `source webenv/bin/activate`

Stop test mode: `deactivate`

## Gunicorn

Create a `wsgi.py` file with the code

```

from app import app

if __name__ == "__main__":
    app.run()

```

The you can perform the following:

Start your env first

(myprojectenv) \$ `gunicorn --bind 0.0.0.0:port wsgi:app`

(myprojectenv) \$ `deactivate`

Templates / Static not loading after SFTP, stop gunicorn and start it again

### Linux

`sudo lsof -i:port / sudo lsof -i:8000`

`kill pid`

Not finding pid?, `pkill gunicorn`

### More on Gunicorn

```
(webenv) root@ubuntu-512mb-fral-01:~/web# gunicorn -w 1 --threads 10 wsgi:app
[2018-07-14 14:40:17 +0000] [3079] [INFO] Starting gunicorn 19.7.1
[2018-07-14 14:40:17 +0000] [3079] [INFO] Listening at: http://127.0.0.1:8000 (3079)
[2018-07-14 14:40:17 +0000] [3079] [INFO] Using worker: threads
[2018-07-14 14:40:17 +0000] [3084] [INFO] Booting worker with pid: 3084
```

<https://stackoverflow.com/questions/35837786/how-to-run-flask-with-gunicorn-in-multithreaded-mode>

Hang in Gunicorn, showing old data from old views?

Some cach issue, reboot server

To reboot immediately you have to specify either of the following commands:

`sudo shutdown -r 0`

`sudo shutdown -r now`