

## 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

```
root@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
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
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
default  flask_settings
```

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_settings
/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

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

""" """
# app/ __init__.py
from flask import Flask, render_template

# initialize the app

app = Flask(__name__)

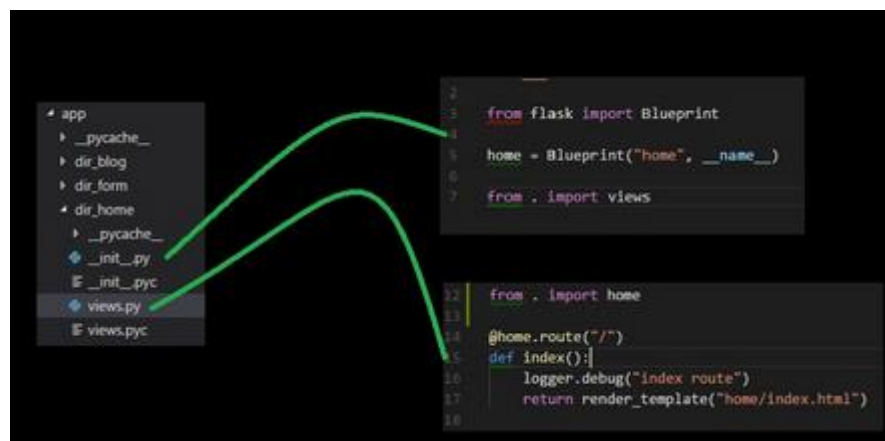
# load the views

# from app import views

from .dir_technology import technology as technology_blueprint
app.register_blueprint(technology_blueprint)

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

Folder->\_\_init\_\_, views



The app is called from run.py for testing

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

run.py
# This is the file that is invoked to start up a development server

from app import app

if __name__ == "__main__":
    # for linux
    print("\n Success")
    print("System path jekl")
    app.run(host='0.0.0.0')
```

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 app

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

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

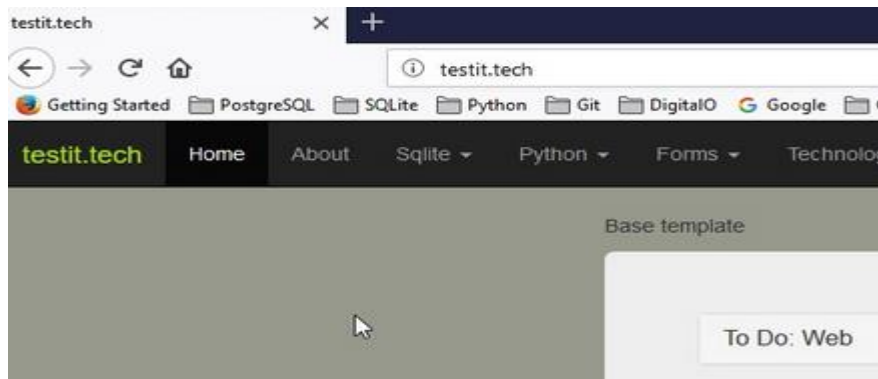
Restart and test Nginx, load activate you env, start gunicorn

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