Deploy Django Project On Nginx with Gunicorn and Virtual Environment

Ubuntu 18.04.3 LTS

1. Before install any package we should update all packages by using this command-

```
$ sudo apt-get update
```

2. Install Nginx server-

```
$ sudo apt-get install nginx
```

3. Install Virtual Environment-

```
$ pip3 install virtualenv
```

4. After install virtual environment we have to create and move into directory-

```
$ mkdir myproject
$ cd myproject
```

5. Create python virtual environment

```
$ virtualenv -p python3 myprojectenv
```

6. Activate python virtual environment

```
$ source myprojectenv/bin/activate
```

7. Install Django and Gunicorn in virtual environment

```
(myprojectenv)$ sudo pip install django gunicorn
```

8. Create Django project

(myprojectenv)$ django-admin startproject myproject

9. Allow host in settings.py

(myprojectenv)$ sudo vi myproject/myproject/settings.py
ALLOWED_HOSTS = ['.example.com', '203.0.113.5']

10. Move bottom in settings.py and add settings where static files should be placed. This is necessary because Nginx handle requests for these items.

STATIC_URL = ‘/static/’
SATATIC_ROOT = os.path.join(BASE_DIR, ‘static/’)

11. Now we can migrate database schema by using -

(myprojectenv)$ /myproject/python manage.py makemigrations
(myprojectenv)$ /myproject/python manage.py migrate

12. Create superuser for project-

(myprojectenv)$/myproject/python manage.py createsuperuser

13. We can collect all the static content into the static directory location by using -

(myprojectenv)$ /*myproject*/python manage.py collectstatic

14. Create a exception for port 8000 by using -

(myprojectenv)$ sudo ufw allow 8000

15. Finally, You can test your project by usning this command -

(myprojectenv)$ /*myprojec/python manage.py runserver 0.0.0.0:8000*

16. Testing Gunicorn's Ability to Serve the Project By Using -

(myprojectenv)$ myproject/gunicorn --bind 0.0.0.0:8000 myproject.wsgi

17. Deactivate Virtual Environment

(myprojectenv)$ Deactivate

18. Create a gunicorn systemd service file by using -

$ sudo vi /etc/systemd/system/gunicorn.service

19. Add [Install] section into gunicorn.service file

[Unit]
Description = gunicorn daemon
After = network.target

[Service]
User = your user name here
Group = www-data
WorkingDirectory = Path of your Django root directory
ExacStart = Path of virtual Environment/bin/gunicorn -- access - logfile - -- workers 3 --bind unix:Path of your django project/projectname.sock projectname.wsgi:application

[Install]
WantedBy = multi-user.target

20. After saving and close this file. We will start the gunicorn.service file and enable it by using -

$ sudo systemctl start gunicorn
$ sudo systemctl enable gunicorn

21. Check Gunicorn socket file

a. Check status of gunicorn

$ sudo systemctl status gunicorn

b. Check the existence of myproject.sock file

$ ls /path of django project/

22. If you make changes in /etc/systemd/system/gunicorn.service file , reload the daemon to reread the gunicorn.service file and restart the gunicorn.service file by using -

$ sudo systemctl daemon-reload
$ sudo systemctl restart gunicorn

Configure the Nginx to proxy pass to gunicorn

23. Creating and opening a new server block in nginx's **sites-available** directory

$ sudo vi /etc/nginx/sites-available/projectname

24. Add following in sites-available

```
server {

    listen 80;

    server_name server_domain_or_IP;


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

    location /static/ {

        root path of django root project;

    }


    location / {
```

```
        include proxy_params;

        proxy_pass http://unix:path of socket file;

    }

}
```

25. Save and close the file when you are finishing.Now we can enable the file by linking it to sites-enabled directory.

$ sudo ln s /etc/nginx/sites-available/projectname /etc/nginx/sites-enabled

26. Testing Nginx Configuration by using-

```
$ sudo nginx -t
```

27. if no errors occurre, restart the Nginx by using -

```
$ sudo systemctl restart nginx
```

28. We no longer need to access development server,we can remove the rule to open port 8000 by using -

```
$ sudo ufw delete allow 8000
$ sudo ufw allow ‘Nginx Full’
```

Run you project with your IP Address