Django Deployment with Mod_wsgi ,Supervisor and Gunicorn Server

Django is a powerful web framework that can help you get your Python application or website off the ground. Django includes a simplified development server for testing your code locally, but for anything even slightly production related, a more secure and powerful web server is required.

Install the Packages from the Ubuntu Repositories

- sudo apt-get update
- sudo apt-get install python3-pip
- sudo apt-get install python3-dev libpq-dev
- sudo apt-get install nginx

Create a Python Virtual Environment for your Project

- sudo -H pip3 install --upgrade pip
- sudo -H pip3 install virtualenv
- mkdir ~/myproject
- cd ~/myproject
- source myprojectenv/bin/activate
- pip install django gunicorn psycopg2
- django-admin.py startproject myproject ~/myproject
- nano ~/myproject/myproject/settings.py

Add The Following Ip's In Your Settings.py file

```
[
# The simplest case: just add the domain name(s) and IP addresses of your Django server
# ALLOWED_HOSTS = [ 'example.com', '203.0.113.5']
# To respond to 'example.com' and any subdomains, start the domain with a dot
# ALLOWED_HOSTS = ['.example.com', '203.0.113.5']
ALLOWED_HOSTS = ['your_server_domain_or_IP', 'second_domain_or_IP', . . .]
]
```

Django Deployment with Mod_wsgi ,Supervisor and Gunicorn Server

Add The Static Media Directory

STATIC_URL = '/static/'

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

Complete Initial Project Setup

- ~/myproject/manage.py makemigrations
- ~/myproject/manage.py migrate
- ~/myproject/manage.py createsuperuser
- ~/myproject/manage.py collectstatic

Create an exception for port 8000 by typing:

- sudo ufw allow 8000
- ~/myproject/manage.py runserver your_ip_server:8000
- (environment) deactivate

Create a Gunicorn systemd Service File

- sudo nano /etc/systemd/system/gunicorn.service

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

[Service]
User=sammy
Group=www-data
WorkingDirectory=/home/sammy/myproject
ExecStart=/home/sammy/myproject/myprojectenv/bin/gunicorn --access-logfile - --workers 3 --bind unix:/home/sammy/myproject/myproject.sock myproject.wsgi:application

[Install]
WantedBy=multi-user.target

]

- sudo systemctl start gunicorn
- sudo systemctl enable gunicorn
- sudo systemctl status gunicorn
- ls /home/sammy/myproject
- sudo journalctl -u gunicorn
- sudo systemctl daemon-reload
- sudo systemctl restart gunicorn

Configure Nginx to Proxy Pass to Gunicorn

- sudo nano /etc/nginx/sites-available/myproject

```
[
server {
    listen 80;
    server_name server_domain_or_IP;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /home/sammy/myproject;
    }

    location / {
        include proxy_params;
        proxy_pass http://unix:/home/sammy/myproject/myproject.sock;
    }
}

]
```

- sudo ln -s /etc/nginx/sites-available/myproject /etc/nginx/sites-enabled
- sudo nginx -t
- sudo systemctl restart nginx
- sudo ufw delete allow 8000
- sudo ufw allow 'Nginx Full'

Django Deployment with Mod_wsgi ,Supervisor and Gunicorn Server

Troubleshooting Nginx and Gunicorn

- sudo tail /var/log/nginx/access.log
- sudo tail /var/log/nginx/error.log
- namei -nom /home/dell/myproject/myproject.sock
- sudo systemctl status postgresql
- sudo systemctl start postgresql
- sudo systemctl enable postgresql
- sudo systemctl restart gunicorn
- sudo systemctl daemon-reload
- sudo systemctl restart gunicorn
- sudo nginx -t && sudo systemctl restart nginx