Deploying a Django Application on Ubuntu 22.04 with Gunicorn and Nginx

Tools & Technologies Used:

- **Operating System:** Ubuntu 22.04
- **Programming Language:** Python (Django Framework)
- **Web Server:** Nginx
- **Application Server:** Gunicorn
- **Database:** SQLite (can be replaced with PostgreSQL/MySQL in production)
- **Virtual Environment:** venv (Python Virtual Environment)
- **Process Manager:** systemd (for managing Gunicorn service)
- **Networking & Security:** UFW (Uncomplicated Firewall) for managing access
- **Version Control:** Git (optional for project versioning)

Prerequisites:

- Ubuntu 22.04 installed
- Root or sudo access
- Docker & Docker Compose installed
- Port 80/443 open for web traffic

Step 1: Install Required Packages

Update your system and install necessary dependencies:

```
apt update && sudo apt upgrade -y
apt install -y python3 python3-pip python3-venv nginx gunicorn git
```

Step 2: Clone git repo & Navigate to Django Project

```
cd /dj/social_media

git clone https://github.com/sunilkumar0633/social_media.git

rm -rf venv    ( remove corrupt env )

python3 -m venv venv

source venv/bin/activate
```

Step 3: Install Dependencies

Install the required Python packages:

```
pip install --upgrade pip
pip install -r requirements.txt
```

Step 4: Run Migrations & Collect Static Files

```
python manage.py migrate


python manage.py collectstatic --noinput
```

Step 5: Test Your Application

Run the Django development server:

NOTE: Change server ip or server name in settings.py

```
vim social/settings.py


python manage.py runserver 0.0.0.0:8000
```

Verify it's accessible by visiting:
http://your-server-ip:8000

Step 6: Set Up Gunicorn as the Application Server

Install Gunicorn:

```
pip install gunicorn


Test running Gunicorn:

gunicorn --bind 0.0.0.0:8000 social.wsgi



Now, create a systemd service for Gunicorn:

vim /etc/systemd/system/gunicorn.service
```

Paste the following:

```
[Unit]
Description=Gunicorn daemon for Django app
After=network.target

[Service]
User=www-data
Group=www-data
WorkingDirectory=/dj/social_media
ExecStart=/dj/social_media/venv/bin/gunicorn --workers 3 --bind
unix:/dj/social_media/social.sock social.wsgi:application

[Install]
WantedBy=multi-user.target
```

```
Enable and start Gunicorn:

systemctl daemon-reload
systemctl enable gunicorn
systemctl start gunicorn
systemctl status gunicorn
```

Step 7: Configure Nginx as a Reverse Proxy

Create an Nginx configuration file:

```
Vim /etc/nginx/sites-available/social_media
```

Add the following:

```
server {
    listen 80;
    server_name www.example.com; #replace with your domain

    location / {
        include proxy_params;
        proxy_pass http://unix:/dj/social_media/social.sock;
    }

    location /static/ {
        root /dj/social_media;
    }

    location /media/ {
        root /dj/social_media;
    }
}
```

Save and exit.

Create a symbolic link:

```
ln -s /etc/nginx/sites-available/social_media /etc/nginx/sites-
enabled/
```

Check for errors:

`nginx -t`

Restart Nginx:

```
systemctl restart nginx
systemctl enable nginx
```

Step 8: Secure the server with SSL

```
apt install certbot python3-certbot-nginx
certbot --nginx -d your_domain
```

Restart services if needed:

```
systemctl restart gunicorn
systemctl restart nginx
```