Step-by-Step: Deploy Django Project on AWS EC2

Prerequisites

- AWS account
- Django project ready (locally)
- Basic understanding of Linux commands

---

1. Launch EC2 Instance

1. Go to EC2 Dashboard
2. Click **"Launch Instance"**
3. Set:
    - **Name**: `MyDjangoServer`
    - **AMI**: Select **Ubuntu Server 22.04 LTS (Free Tier eligible)**
    - **Instance Type**: `t2.micro`
    - **Key Pair**: Create new or use existing (you'll download a `.pem` file)
    - **Security Group**:
        - Allow **SSH (port 22)** from **Your IP**
        - Allow **HTTP (port 80)** and **HTTPS (port 443)** from **Anywhere**
    - Launch instance

---

2. Connect to EC2

```
chmod 400 your-key.pem
ssh -i "your-key.pem" ubuntu@<your-ec2-public-ip>
```

---

3. Install Required Packages

```
sudo apt update && sudo apt upgrade -y
sudo apt install python3-pip python3-dev libpq-dev nginx curl git -y
sudo apt install python3-venv -y
```

---

4. Clone Your Django Project or Upload Code

If using Git:

```
git clone <your-repo-url>
cd <your-project-directory>
```

Or use **SCP** from your machine:

```
scp -i your-key.pem -r your_project/ ubuntu@<ec2-ip>:/home/ubuntu/
```

---

5. Setup Virtual Environment

```
python3 -m venv venv
source venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

---

6. Django Configurations

1. **Allow EC2 IP in `ALLOWED_HOSTS`** in `settings.py`

```
ALLOWED_HOSTS = ['<your-ec2-public-ip>', 'localhost']
```

2. **Migrate DB, Create Superuser, Collect Static Files:**

```
python manage.py migrate
python manage.py createsuperuser
python manage.py collectstatic
```

---

7. Install and Setup Gunicorn

```
pip install gunicorn
gunicorn --workers 3 --bind 0.0.0.0:8000 myproject.wsgi:application
```

(To test it manually)

---

8. Setup NGINX

Create config file:

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

Paste:

```
nginx
server {
```

```
    listen 80;
    server_name <your-ec2-public-ip>;

    location = /favicon.ico { access_log off; log_not_found off; }
    location /static/ {
        root /home/ubuntu/<your-project-folder>;
    }

    location / {
        include proxy_params;
        proxy_pass http://127.0.0.1:8000;
    }
}
```

Then:

```
sudo ln -s /etc/nginx/sites-available/myproject /etc/nginx/sites-enabled
sudo nginx -t
sudo systemctl restart nginx
```

---

9. (Optional) Setup Firewall

```
sudo ufw allow 'Nginx Full'
```

---

10. Access Your Django App

Open browser:

`http://<your-ec2-public-ip>`

---

11. Run Gunicorn as a Service (so it stays running)

Create a systemd service file:

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

Paste:

```
ini

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

[Service]
User=ubuntu
Group=www-data
WorkingDirectory=/home/ubuntu/<your-project-folder>
ExecStart=/home/ubuntu/<your-project-folder>/venv/bin/gunicorn --access-logfile
- --workers 3 --bind unix:/home/ubuntu/<your-project-folder>/gunicorn.sock
<your_project_name>.wsgi:application

[Install]
```

```
WantedBy=multi-user.target
```

Then

```
sudo systemctl start gunicorn
sudo systemctl enable gunicorn
```

Update NGINX config to use the Unix socket:

```
proxy_pass http://unix:/home/ubuntu/<your-project-folder>/gunicorn.sock;
```

Reload NGINX:

```
sudo systemctl restart nginx
```

---

12. Test Everything

- Run: `sudo systemctl status gunicorn`
- Visit your app in browser