Deploy Django on EC2

how to deploy django project into ec2 within free tier settings

Deploying a Django project on an **AWS EC2** instance within the **Free Tier** involves several steps. Here's a step-by-step guide:

Step 1: Launch an EC2 Instance

1. **Log in to AWS Console** → Go to **EC2 Dashboard**.
2. Click **Launch Instance**.
3. Choose **Ubuntu 22.04 LTS** (or any other preferred OS).
4. Select **t2.micro** (Free Tier eligible).
5. Configure:
   - Allow **HTTP (80), HTTPS (443), and SSH (22)** in security group.
6. Click **Launch** and download the key pair (`.pem` file).

Step 2: Connect to EC2

1. Open your terminal and run:

```bash
chmod 400 your-key.pem
ssh -i your-key.pem ubuntu@your-ec2-public-ip
```

Step 3: Update & Install Dependencies

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

Step 4: Clone Your Django Project

```bash
git clone https://github.com/your-repo.git
cd your-repo
```

Step 5: Create a Virtual Environment & Install Requirements

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

Step 6: Configure Django

1. **Set allowed hosts in `settings.py`:**

   ```python
   ALLOWED_HOSTS = ['your-ec2-public-ip', 'your-domain.com']
   ```

2. **Apply migrations & collect static files:**

```bash
python manage.py migrate
python manage.py collectstatic --noinput
```

3. **Create a superuser (optional):**

```bash
python manage.py createsuperuser
```

Step 7: Run Django with Gunicorn

```bash
pip install gunicorn
gunicorn --bind 0.0.0.0:8000 your_project.wsgi
```

- If you see the app running on `0.0.0.0:8000`, press `CTRL+C`.

Step 8: Configure Nginx as a Reverse Proxy

1. Open Nginx config:

```bash
sudo nano /etc/nginx/sites-available/django
```

2. Add the following:

```nginx
server {
    listen 80;
    server_name your-ec2-public-ip;
    location / {
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

3. Save and exit (`CTRL+X`, `Y`, `ENTER`).
4. Enable the config:

```bash
sudo ln -s /etc/nginx/sites-available/django /etc/nginx/sites-enabled
sudo systemctl restart nginx
```

Step 9: Configure Firewall

```bash
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

Step 10: Run Django as a Service (Optional)

1. Create a service file:

```bash
sudo nano /etc/systemd/system/django.service
```

2. Add:

```ini
```

```
[Unit]
Description=Django Service
After=network.target

[Service]
User=ubuntu
WorkingDirectory=/home/ubuntu/your-repo
ExecStart=/home/ubuntu/your-repo/venv/bin/gunicorn --workers 3 --bind
unix:/home/ubuntu/your-repo/django.sock your_project.wsgi:application

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

3. Enable and start the service:

```bash
sudo systemctl daemon-reload
sudo systemctl start django
sudo systemctl enable django
```

Step 11: Access Your Django App

- Open a browser and visit `http://your-ec2-public-ip/`.

Optional: Add a Domain & SSL (HTTPS)

1. **Point your domain to the EC2 IP** in your domain provider.
2. Install **Certbot** for **SSL**:

```bash
sudo apt install certbot python3-certbot-nginx -y
sudo certbot --nginx -d your-domain.com
```

Done! 🎉

Your Django project is now running on **AWS EC2 Free Tier**! 🚀