## 🚀 Triển khai Django trên EC2 với Domain riêng và HTTPS (SSL)

### ✅ \*\*Bước 1: Gán Elastic IP cho EC2\*\*

IP công khai mặc định của EC2 (ví dụ: `54.255.88.116`) sẽ thay đổi khi khởi động lại instance. Để cố định IP:

1. Vào \*\*AWS Management Console\*\* > \*\*EC2\*\* > \*\*Elastic IPs\*\*.

2. Nhấn \*\*Allocate Elastic IP address\*\* > \*\*Allocate\*\*.

3. Chọn Elastic IP vừa tạo → \*\*Associate Elastic IP address\*\* → Gán vào EC2 instance (ví dụ: `ip-172-31-29-111`).

4. Tick chọn \*\*Allow this Elastic IP address to be reassociated\*\* để cho phép gán lại nếu cần.

\*\*✅ Ghi chú:\*\* Elastic IP ví dụ: `47.130.87.247`.

---

### 🌐 \*\*Bước 2: Trỏ Domain về Elastic IP\*\*

#### 🔸 Đăng ký domain

Bạn có thể đăng ký domain thông qua:

\* AWS Route 53

\* GoDaddy

\* Namecheap

\* Hoặc bất kỳ nhà cung cấp nào khác

#### 🔸 Cấu hình DNS (A Record)

Tạo hai bản ghi \*\*A record\*\* trỏ về Elastic IP:

| Mục tiêu | Type | Host | Value (Elastic IP) | TTL |

| ------------- | ---- | ----- | ------------------ | --------- |

| Gốc domain | A | `@` | `47.130.87.247` | 5 minutes |

| Subdomain www | A | `www` | `47.130.87.247` | 5 minutes |

#### 🔸 Kiểm tra DNS propagation

Truy cập [https://dnschecker.org](https://dnschecker.org) và kiểm tra domain `tothichcau.shop`.

---

### 🧰 \*\*Bước 3: Cập nhật hệ thống\*\*

```bash

sudo dnf update -y

```

---

### 🖥️ \*\*Bước 4: Cài đặt Nginx\*\*

```bash

sudo dnf install -y nginx

sudo systemctl start nginx

sudo systemctl enable nginx

```

Kiểm tra trạng thái:

```bash

sudo systemctl status nginx

```

---

### 🔐 \*\*Bước 5: Cài đặt Certbot (Let's Encrypt)\*\*

```bash

sudo dnf install -y python3 python3-pip

sudo pip3 install certbot certbot-nginx

```

Kiểm tra phiên bản:

```bash

certbot --version

```

---

### ⚙️ \*\*Bước 6: Cấu hình Nginx để phục vụ Django\*\*

Tạo file cấu hình mới:

```bash

sudo nano /etc/nginx/conf.d/stripe.conf

```

Dán nội dung sau:

```nginx

# Redirect HTTP sang HTTPS

server {

listen 80;

server\_name tothichcau.shop www.tothichcau.shop;

return 301 https://$host$request\_uri;

}

# Server HTTPS sử dụng SSL từ Let's Encrypt

server {

listen 443 ssl;

server\_name tothichcau.shop www.tothichcau.shop;

ssl\_certificate /etc/letsencrypt/live/tothichcau.shop/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/tothichcau.shop/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf;

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem;

location / {

proxy\_pass http://localhost: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;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

location /webhook/stripe/ {

proxy\_pass http://localhost:8000/webhook/stripe/;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header Stripe-Signature $http\_stripe\_signature;

}

}

```

Kiểm tra và reload Nginx:

```bash

sudo nginx -t

sudo systemctl reload nginx

```

---

### 🔒 \*\*Bước 7: Cấp phát SSL bằng Certbot\*\*

Chạy lệnh sau để tự động cấu hình SSL:

```bash

sudo certbot --nginx -d tothichcau.shop -d www.tothichcau.shop

```

Sau khi thành công:

```bash

sudo systemctl reload nginx

```

---

### 🛡️ \*\*Bước 8: Cấu hình bảo mật cho Django\*\*

Trong `settings.py`, đảm bảo có các dòng sau:

```python

ALLOWED\_HOSTS = ['www.tothichcau.shop', 'tothichcau.shop', 'localhost', '127.0.0.1']

CSRF\_TRUSTED\_ORIGINS = [

'https://www.tothichcau.shop',

'https://tothichcau.shop'

]

```

✅ Mặc dù không bắt buộc, \*\*3 dòng bảo mật cuối cùng nên được thêm vào khi đã triển khai HTTPS\*\*, giúp đảm bảo cookie và CSRF token chỉ truyền qua kết nối an toàn.

---

### 🔓 \*\*Bước 9: Kiểm tra Security Group\*\*

Vào EC2 > Security Groups > Inbound Rules:

\* ✅ Mở port 80 (HTTP)

\* ✅ Mở port 443 (HTTPS)