NGINX · LET'S ENCRYPT DIFICULTAD ●●●●● TIEMPO · ~30 MIN

Proxy inverso con Nginx y HTTPS paso a paso

Nginx es la puerta de entrada de tu servidor: recibe todas las visitas por el 443, les pone el candado de HTTPS y las reparte a cada aplicación según el subdominio. Es exactamente lo que uso en producción.

PILA	TIEMPO	PASOS	PROBADO EN
Nginx · Let's Encrypt	~30 min	4	Ubuntu 24.04 LTS

SI PREFIERES QUE LO HAGA YO
desde 90 €

Necesitas un VPS con Ubuntu 24.04, los puertos 80 y 443 abiertos y un subdominio (por ejemplo `app.tudominio.com`) con un registro `A` apuntando a la IP del servidor. La aplicación debe estar escuchando en `127.0.0.1`, por ejemplo en el puerto 8080.

> **Antes de copiar**
> En los comandos verás `tudominio.com`: sustitúyelo siempre por tu dominio real.

PASO 01 Instalar Nginx y Certbot

El plugin `python3-certbot-nginx` permite que Certbot edite la configuración de Nginx por ti al pedir el certificado.

install.sh

```
sudo apt update && sudo apt install -y nginx certbot python3-certbot-nginx
sudo ufw allow "Nginx Full"
sudo systemctl enable --now nginx
```

PASO 02

Crear el sitio para tu subdominio

Cada aplicación tiene su archivo en `sites-available`. Las cabeceras `X-Forwarded-*` le dicen a la aplicación la IP real del visitante y que la conexión original era HTTPS.

/etc/nginx/sites-available/app.tudominio.com

```
server {
    listen 80;
    server_name app.tudominio.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
        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;
    }
}
```

PASO 03

Activar el sitio y comprobar la sintaxis

Un enlace simbólico en `sites-enabled` activa el sitio. `nginx -t` revisa la configuración antes de recargar: si hay un error, Nginx sigue funcionando con la anterior.

activar.sh

```
sudo ln -s /etc/nginx/sites-available/app.tudominio.com /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx
```

PASO 04

Pedir el certificado HTTPS

Certbot valida que el dominio es tuyo, descarga el certificado y añade a Nginx el bloque del puerto 443 y la redirección de HTTP a HTTPS.

https.sh

```
sudo certbot --nginx -d app.tudominio.com --redirect
# Prueba que la renovación automática funcionará
sudo certbot renew --dry-run
```

Renovación automática

Certbot instala un temporizador de systemd que revisa los certificados dos veces al día y renueva los que caducan en menos de 30 días. Compruébalo con `systemctl list-timers | grep certbot`.

¿Prefieres que lo haga por ti?

Lo implemento por ti por **desde 90 €**, con la configuración de seguridad inicial y esta guía aplicada a tu servidor.