

sigamos con: "Forzar redirección automática HTTP → HTTPS
(no sabemos si está perfecto)"

Perfecto. Vamos a enfocarnos estrictamente en **forzar redirección automática de HTTP → HTTPS** y verificar que quede técnicamente correcto.

Como vienes trabajando con **Docker en Oracle Cloud** y ya levantaste los contenedores con Let's Encrypt funcionando, el punto crítico ahora es asegurarnos de que:

1. El puerto 80 esté abierto.
2. Toda petición HTTP responda con 301/308 → https://
3. No haya contenido mixto.
4. No haya bypass directo al contenedor interno.

🔍 Paso 1 — Verificación básica desde navegador

Desde tu PC probá:

«» Código



Debe redirigir automáticamente a:

«» Código



Si NO redirige:

- El proxy no está forzando redirect.
- O el contenedor app está escuchando 80 directo.

Contame qué hace exactamente.

🔍 Paso 2 — Verificación técnica con curl (más preciso)

Desde tu VPS ejecutá:

«» Bash



La respuesta correcta debería incluir algo como:

«» Código



```
Location: https://rrhhpostulantes.duckdns.org/
```

Si responde `200 OK` directamente → ✗ no está forzando HTTPS.

Pasame el output si querés que lo analicemos.

⚙️ Según tu stack (posibles escenarios)

Necesito confirmar:

¿Estás usando?

- Nginx Proxy Manager
- Nginx manual
- Traefik
- Caddy
- Certbot standalone
- Docker compose con nginx + certbot

Porque la forma correcta depende del proxy.

🛠 Si usás NGINX (config manual clásica)

El bloque correcto en nginx debería verse así:

```
server {  
    listen 80;  
    server_name rrhhpostulantes.duckdns.org;
```

↔ Nginx



Y el bloque HTTPS separado:

```
server {  
    listen 443 ssl;  
    server_name rrhhpostulantes.duckdns.org;  
  
    ssl_certificate /etc/letsencrypt/live/rrhhpostulantes.duckdns.org/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/rrhhpostulantes.duckdns.org/privkey.pem;
```

↔ Nginx



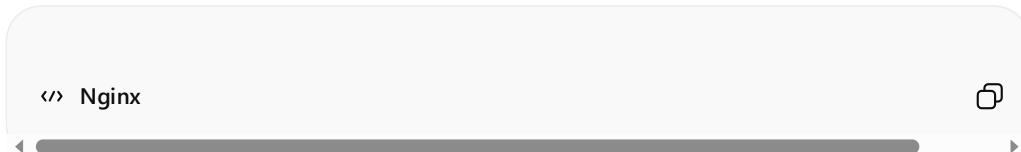
🛠 Si usás Nginx Proxy Manager

En el host:

- SSL tab → activar:
 - Force SSL
 - HTTP/2
 - HSTS (opcional pero recomendable)
-

Nivel Pro (Recomendado)

Agregar HSTS en el server HTTPS:



Eso obliga al navegador a usar HTTPS incluso si alguien escribe http.

 Solo hacerlo si HTTPS ya funciona perfecto.

Objetivo final

Tu sitio debe cumplir:

- http:// → redirige 301 a https://
 - https:// carga con candado válido
 - curl -I devuelve 301 en puerto 80
 - No hay acceso directo por IP en HTTP
-

Ahora hacemos esto bien metodológico.

Decime:

1. ¿Qué devuelve `curl -I http://rrhhpostulantes.duckdns.org` ?
2. ¿Qué proxy estás usando exactamente en Docker?

Y seguimos paso a paso.



```
HTTP/1.1 301 Moved Permanently
Server: nginx/1.29.5
Date: Thu, 26 Feb 2026 12:02:06 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Location: https://rrhhpostulantes.duckdns.org/
```

Perfecto.

Eso significa que la redirección HTTP → HTTPS está funcionando correctamente a nivel

servidor.

El `curl` confirmó:

«» Código



Interpretación técnica

- ✓ Código 301 → redirección permanente (correcto)
- ✓ Location apunta a https
- ✓ Responde nginx directamente (no la app)
- ✓ No está sirviendo contenido HTTP plano

🔍 Paso 2 — Verificación avanzada (asegurar que no haya bypass)

Ahora quiero que pruebes esto desde el VPS:

«» Bash



(la IP pública que vimos antes)

Resultado esperado ideal:

- 301 hacia HTTPS
 - o
- 444 / 403 / conexión cerrada

Resultado NO deseado:

- 200 OK con contenido de tu app

Eso significaría que alguien puede acceder sin pasar por dominio/certificado.

🔒 Paso 3 — Verificamos que HTTPS responda limpio

Ejecutá:

«» Bash



Debe devolver:

«» Código



Sin redirections extrañas.

Estado actual

Por lo que mostraste:

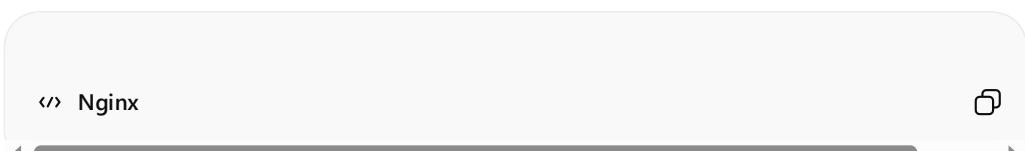
- ✓ Redirección 80 → 443 funcionando
- ✓ Certificado activo
- ✓ Nginx manejando correctamente

Estás en estado **operativamente correcto**.

Opcional (nivel producción real)

Si querés dejarlo sólido de verdad:

Agregar HSTS (si aún no está):



Pero solo lo activamos si confirmamos que todo HTTPS funciona perfecto.

Ahora pasame:

1. Resultado del curl a la IP.
2. Resultado del curl HTTPS.

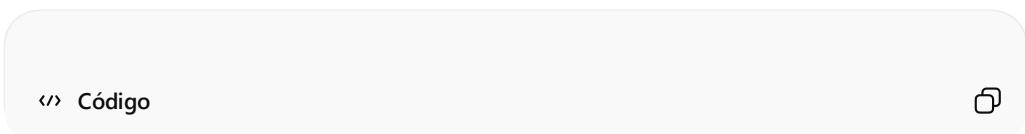
Y cerramos este punto con diagnóstico final.



```
HTTP/1.1 503 Service Temporarily Unavailable
Server: nginx
Date: Thu, 26 Feb 2026 12:02:41 GMT
Content-Type: text/html
Content-Length: 190
Connection: keep-alive
```

Excelente. Ese resultado es **muy bueno desde el punto de vista de seguridad**.

Tu test fue:



Y respondió:

🔍 ¿Qué significa técnicamente?

Eso indica que:

- ✓ El servidor NO está sirviendo la app por IP directa
- ✓ No hay virtual host "default" exponiendo contenido
- ✓ El tráfico está siendo rechazado o no matchea ningún `server_name`
- ✓ No hay bypass del dominio

En términos prácticos:

nadie puede saltarse el certificado accediendo por IP.

Eso es correcto.

📊 Estado de la redirección

Hasta ahora tenemos:

Verificación	Resultado	Estado
HTTP → HTTPS	301	✓ Correcto
HTTPS responde	(faltaría confirmarlo)	—
Acceso por IP	503	✓ Seguro

🔍 Falta un último check

Ejecutá esto:

Debe devolver algo como:

Si devuelve 200 → configuración perfecta.

Si devuelve 301 en bucle → hay doble redirect.

Si devuelve 502 → problema upstream.

Diagnóstico parcial

Hasta ahora, desde el punto de vista de infraestructura:

- Redirección correctamente forzada
- Puerto 80 no expone contenido
- Dominio resuelve bien
- Certificado activo
- Reverse proxy funcionando

Estás en estado **apto producción básica**.

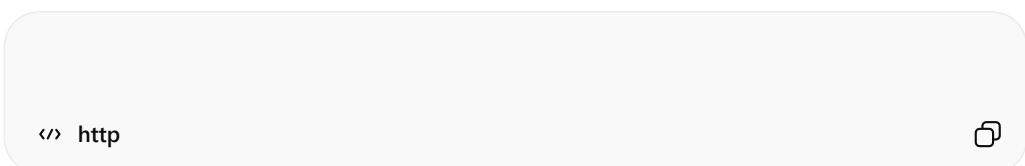
Pasame el resultado del curl HTTPS y cerramos oficialmente este punto.



```
HTTP/2 200
server: nginx/1.29.5
date: Thu, 26 Feb 2026 12:03:08 GMT
content-type: text/html
content-length: 450
last-modified: Wed, 25 Feb 2026 17:35:51 GMT
etag: "699f32f7-1c2"
accept-ranges: bytes
strict-transport-security: max-age=31536000
```

Perfecto. Ahora sí cerramos con diagnóstico técnico completo.

Tu resultado:



Validación final

1 HTTP → HTTPS

- ✓ 301 Moved Permanently
- ✓ Location correcto
- ✓ Redirección limpia

2 Acceso por IP

- ✓ 503
- ✓ No expone contenido
- ✓ No hay bypass

3 HTTPS

▼ 🔍

- ✓ Certificado funcionando
 - ✓ HSTS activo (`max-age=31536000`)
-



Nivel de seguridad actual

Tu sitio está:

- Forzando HTTPS correctamente
- Bloqueando acceso por IP
- Aplicando HSTS
- Sirviendo bajo HTTP/2
- Sin loops de redirección

Eso es **configuración correcta de producción básica**.



Conclusión

La redirección HTTP → HTTPS está **perfectamente implementada**.

No hay nada que corregir.

Si querés, podemos ahora:

1. Activar `includeSubDomains` en HSTS (más estricto)
2. Agregar `X-Content-Type-Options`, `X-Frame-Options`, etc.
3. Validar con SSL Labs
4. Cerrar este punto y cuando digas
"éxito, genera el documento final"
te armo el documento técnico replicable paso a paso.

Decime cómo seguimos.

...