
Relancer le **BACKEND (Django + Gunicorn)**

1 Relancer Gunicorn

```
sudo systemctl restart gunicorn-christlandtech
```

2 Vérifier qu'il tourne bien

```
sudo systemctl status gunicorn-christlandtech --no-pager
```

Tu dois voir :

```
Active: active (running)
```

Relancer **NGINX** (recommandé après un souci backend)

```
sudo systemctl restart nginx
```

Vérifier :

```
sudo systemctl status nginx --no-pager
```

Relancer le **FRONTEND**

 Ça dépend **comment ton frontend est servi**.

◇ Cas 1 : Frontend = fichiers build (React/Vite) servis par Nginx

 **Aucune commande spéciale** Il suffit juste de recharger nginx :

```
sudo systemctl reload nginx
```

◇ Cas 2 : Frontend lancé avec PM2 / Node (SSR, Next.js, etc.)

Si tu utilises PM2 :

```
pm2 restart all
```

ou plus précis :

```
pm2 restart christlandtech-frontend
```

Voir l'état :

```
pm2 status
```

◇ Cas 3 : Frontend en mode dev (rare en prod)

Exemple :

```
npm run dev
```

🔗 À éviter en production

☑ Commande "tout relancer" (safe)

Si tu veux être sûr :

```
sudo systemctl restart gunicorn-christlandtech  
sudo systemctl restart nginx
```

🔗 90 % du temps, ça suffit.

🔍 Test rapide après redémarrage

```
curl -i https://christland.tech/api/dashboard/stats/?lang=fr
```

- 403 → normal (auth)
 - 200 → parfait
 - ✖ 502 → problème backend à nouveau
-

🧠 Astuce pro (mémo à garder)

```
# Backend
sudo systemctl restart gunicorn-christlandtech

# Voir les erreurs backend
sudo journalctl -u gunicorn-christlandtech -n 50 --no-pager

# Nginx
sudo systemctl restart nginx
sudo nginx -t
```
