Réinstallation propre du tunnel Cloudflare pour YunoHost

Procédure de référence avec **pascot.ca** défini dès l'installation comme domaine principal

Contexte visé : Raspberry Pi + YunoHost fraîchement installé + domaine déjà géré par Cloudflare. Cette méthode évite l'ouverture de ports et ne nécessite pas d'IP fixe.

1. Pré-requis

- Le domaine **pascot.ca** est déjà chez Cloudflare et les *nameservers* pointent bien vers Cloudflare.
- YunoHost vient juste d'être installé et le domaine principal choisi pendant le post-install est **pascot.ca**.
- Vous avez un accès SSH au serveur avec un utilisateur sudo (dans les exemples : **dpt**).
- Aucun enregistrement **A** ou **AAAA** ne doit rester pour le site web principal quand le tunnel sera en place. Les enregistrements mail (MX, TXT, DKIM, SPF) peuvent rester.

2. Vérifier que YunoHost est bien en domaine principal pascot.ca

Commande utile :

```
sudo yunohost domain list
```

Résultat attendu :

```
domains:
  - pascot.ca
main: pascot.ca
```

3. Installer cloudflared

```
sudo apt update
sudo apt install -y curl
curl -L https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-linux-arm64.deb
sudo dpkg -i cloudflared.deb
```

Sur Raspberry Pi 64 bits, le paquet **arm64** est le bon choix.

4. Authentifier cloudflared auprès de Cloudflare

```
cloudflared tunnel login
```

Dans le navigateur, choisir la zone **pascot.ca** puis autoriser le tunnel. Une fois validé, vérifier que le certificat existe :

```
ls -la ~/.cloudflared
```

On doit y voir au minimum **cert.pem**.

5. Créer le tunnel

```
cloudflared tunnel create yunohost
```

Cette commande crée un tunnel nommé **yunohost** et un fichier JSON de credentials dans **~/.cloudflared/**. Notez l'UUID renvoyé : il sera utilisé dans la configuration.

6. Créer la configuration du tunnel

```
sudo mkdir -p /etc/cloudflared
sudo nano /etc/cloudflared/config.yml
```

```
tunnel: UUID_DU_TUNNEL
credentials-file: /home/dpt/.cloudflared/UUID_DU_TUNNEL.json

ingress:
  - hostname: pascot.ca
    service: https://localhost:443
    originRequest:
      noTLSVerify: true

  - hostname: "*.pascot.ca"
    service: https://localhost:443
    originRequest:
      noTLSVerify: true

  - service: http_status:404
```

Le bloc ***.pascot.ca** est pratique : il permet de publier plus tard de nouveaux sous-domaines d'applications sans devoir toucher à **config.yml** à chaque fois.

7. Valider la configuration

```
cloudflared tunnel ingress validate
```

8. Créer les routes DNS du tunnel

Pour le domaine principal :

```
cloudflared tunnel route dns yunohost pascot.ca
```

Pour un futur sous-domaine, par exemple **aquarelles.pascot.ca** :

```
cloudflared tunnel route dns yunohost aquarelles.pascot.ca
```

Dans Cloudflare DNS, gardez les enregistrements **Tunnel** créés par ces commandes. Supprimez les anciens enregistrements **A** ou **AAAA** qui pointaient vers une IP pour le site web principal. En revanche, gardez les enregistrements mail (MX/TXT).

9. Installer le tunnel en service permanent

```
sudo cloudflared service install
sudo systemctl enable cloudflared
sudo systemctl start cloudflared
sudo systemctl status cloudflared
```

Le service doit être en **active (running)**. Si Cloudflare affiche une erreur 1033, la première chose à vérifier est généralement l'état de ce service.

10. Test fonctionnel

- Portail YunoHost : **https://pascot.ca**
- Administration : **https://pascot.ca/yunohost/admin**
- Si l'admin affiche une erreur Cloudflare 1033, vérifier : **sudo systemctl status cloudflared**.
- Si un sous-domaine répond "site introuvable", vérifier qu'il a bien été ajouté dans YunoHost et qu'une route DNS tunnel existe pour lui.

11. Procédure standard pour chaque nouvelle application en sous-domaine

```
# 1) Ajouter le sous-domaine dans YunoHost
sudo yunohost domain add monapp.pascot.ca

# 2) Créer la route DNS Cloudflare vers le tunnel
cloudflared tunnel route dns yunohost monapp.pascot.ca

# 3) Installer l'application
sudo yunohost app install NOM_DE_L_APP -d monapp.pascot.ca

# 4) Redémarrer le tunnel si vous avez modifié config.yml
sudo systemctl restart cloudflared
```

Avec le wildcard ***.pascot.ca** déjà présent dans **config.yml**, l'étape 4 est en général inutile tant que l'on ne change pas la structure du fichier.

12. Ce qu'il vaut mieux éviter

- Changer ensuite le domaine principal YunoHost : cela peut casser le SSO ou provoquer des redirections vers un ancien domaine.
- Conserver en même temps un enregistrement DNS de type **A** et un enregistrement **Tunnel/CNAME** pour le même nom.
- Forcer un certificat Let's Encrypt côté YunoHost dans ce montage précis : le HTTPS public est déjà terminé côté Cloudflare. Avec **noTLSVerify: true**, le tunnel fonctionne sans exposer le serveur sur Internet.
- Oublier que le tunnel doit être lancé en service permanent : un test manuel avec **cloudflared tunnel run** ne suffit pas à long terme.

13. Dépannage rapide

Symptôme	Cause probable	Commande utile
Erreur Cloudflare 1033	service cloudflared arrêté	sudo systemctl status cloudflared
404 sur /yunohost/sso	domaine principal YunoHost incohérent	sudo yunohost domain list

Symptôme	Cause probable	Commande utile
Sous-domaine introuvable	route DNS tunnel absente	cloudflared tunnel route dns yunohost monapp.pascot.c
Sous-domaine ajouté mais non servi	config.yml tunnel incomplète sans wildcard	cloudflared tunnel ingress validate

Document de synthèse rédigé pour une réinstallation rapide et reproductible.