

Activez HTTPS sur Ubuntu avec Certbot : le parcours de contrôle

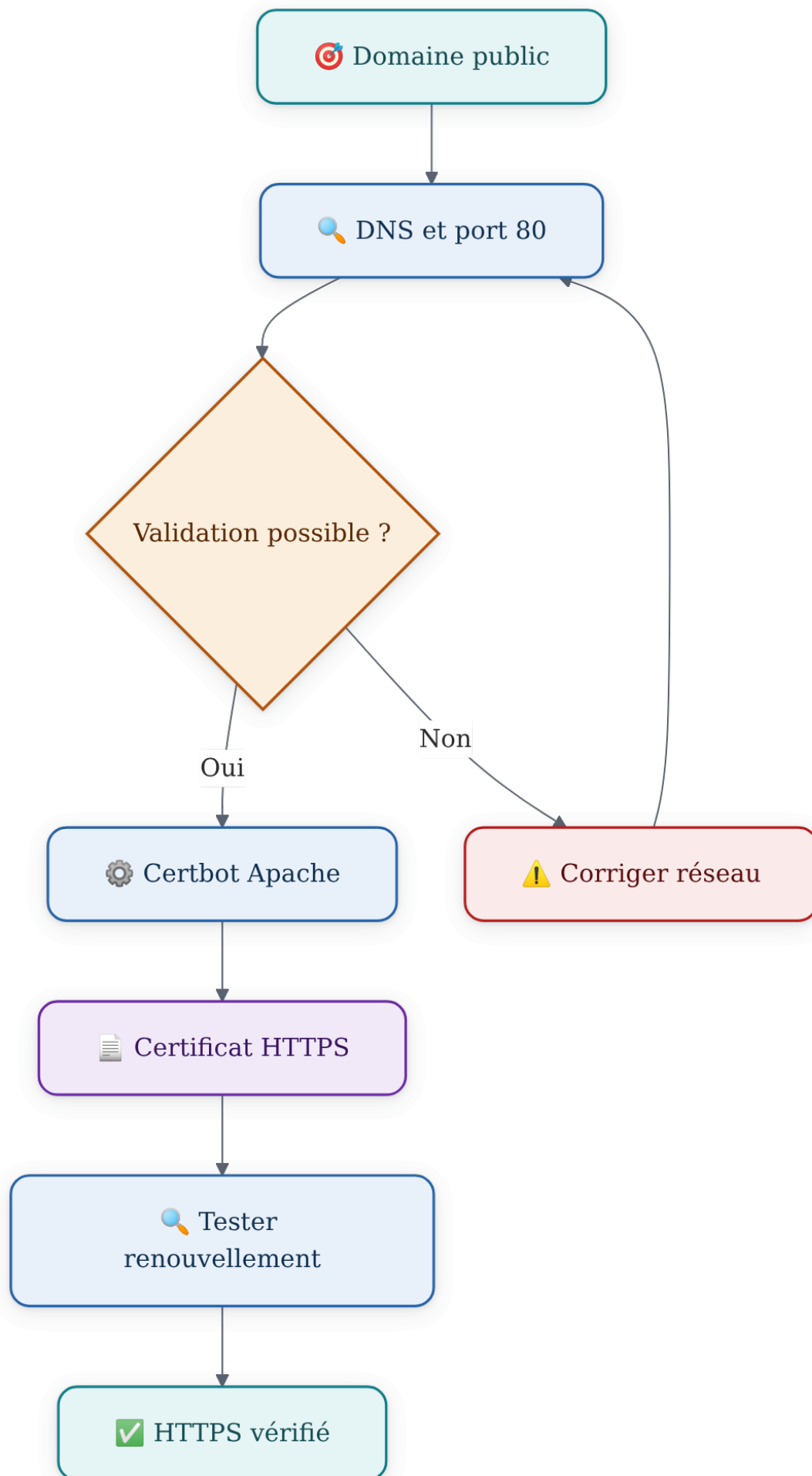
Suivez cette feuille de route pour demander, installer et vérifier un certificat Let's Encrypt sur un serveur Ubuntu avec Apache. Cochez les prérequis avant de lancer Certbot : ils conditionnent la validation du domaine.

Au sommaire

1. Prérequis à valider avant toute installation
2. Préparer Ubuntu, Apache et le pare-feu
3. Installer Certbot et demander le certificat
4. Contrôles juste après l'émission
5. Repères de diagnostic rapide

1. Prérequis à valider avant toute installation

- ☐ **Disposer d'un accès administrateur avec sudo**
Vérifier avec : `sudo -v`
- ☐ **Faire pointer le domaine vers l'adresse IP publique du serveur**
Contrôler avec : `getent hosts domain-name.com`
- ☐ **Utiliser un nom de domaine, et non l'adresse IP du serveur**
Remplacer `domain-name.com` par le domaine réel dans les commandes.
- ☐ **Vérifier qu'Apache est installé ou prévoir son installation**
Contrôler avec : `apache2 -v`
- ☐ **Rendre le port HTTP 80 accessible depuis Internet**
La validation HTTP de Let's Encrypt en dépend.
- ☐ **Autoriser HTTP et HTTPS dans UFW et dans le pare-feu de l'hébergeur, s'il existe**
Les ports 80 et 443 doivent être joignables.



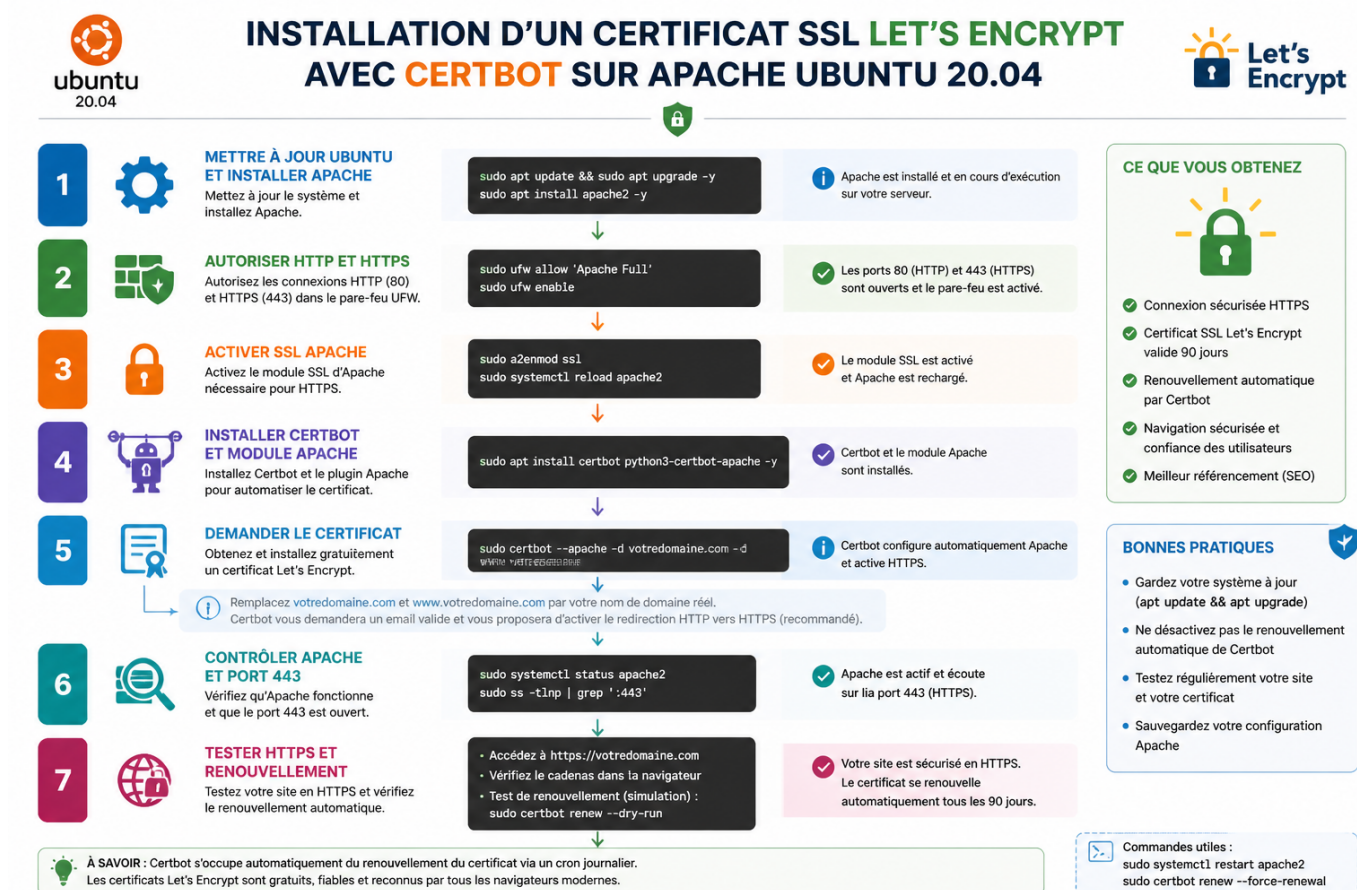
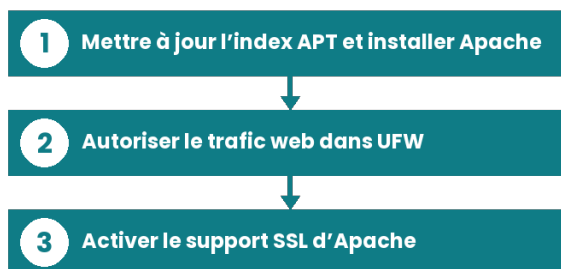


SCHÉMA Vue d'ensemble de la séquence Certbot avec Apache sur Ubuntu.

2. Préparer Ubuntu, Apache et le pare-feu



1 Mettre à jour l'index APT et installer Apache

Exécuter : `sudo apt update` puis `sudo apt install apache2`. Vérifier ensuite la présence d'Apache avec : `apache2 -v`.

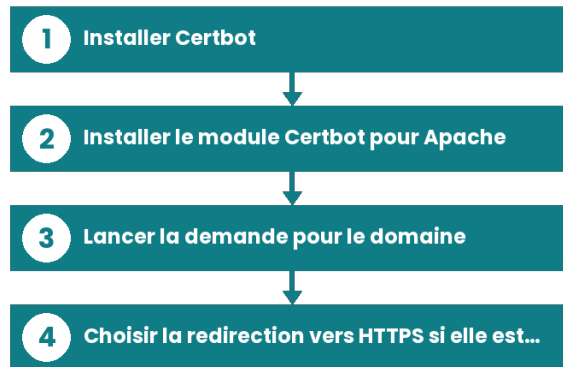
2 Autoriser le trafic web dans UFW

Si UFW est utilisé, exécuter : `sudo ufw allow "Apache Full"`. Contrôler les règles avec : `sudo ufw status`.

3 Activer le support SSL d'Apache

Exécuter : `sudo a2enmod ssl`, puis `sudo systemctl restart apache2`. Vérifier que le service est actif avec : `sudo systemctl status apache2`.

3. Installer Certbot et demander le certificat



1 Installer Certbot

Sur l'environnement Ubuntu 20.04 décrit, installer le paquet letsencrypt avec APT.

2 Installer le module Certbot pour Apache

Installer python3-certbot-apache afin de permettre à Certbot de demander le certificat et de configurer Apache.

3 Lancer la demande pour le domaine

Exécuter Certbot avec le plugin Apache pour le nom de domaine concerné. Répondre aux invites uniquement après avoir confirmé que le DNS et le port 80 sont corrects.

4 Choisir la redirection vers HTTPS si elle est proposée

Certbot peut configurer la redirection de HTTP vers HTTPS dans Apache.

4. Contrôles juste après l'émission

☐ Vérifier la syntaxe de la configuration Apache

Ne poursuivre que si la configuration est valide.

☐ Vérifier que le service Apache fonctionne

Le statut attendu est active (running).

☐ Ouvrir le site avec `https://` suivi du domaine

Contrôler que le navigateur ne signale pas d'erreur de certificat.

☐ Tester l'accès HTTP

Si la redirection a été activée, vérifier qu'il bascule vers HTTPS.

☐ Simuler le renouvellement

Exécuter : `sudo certbot renew --dry-run`

Repères de diagnostic rapide

Symptôme	Point à contrôler en priorité
La demande de certificat échoue avant l'émission	Le domaine ne résout pas vers le bon serveur, ou le port 80 n'est pas accessible.

Symptôme	Point à contrôler en priorité
Apache ne redémarre pas	Lire l'erreur de configuration et corriger le problème avant de relancer Certbot.
Le site n'est pas accessible en HTTPS	Vérifier Apache, l'autorisation du port 443 dans UFW et les règles réseau de l'hébergeur.
Le renouvellement simulé échoue	Vérifier que le domaine, le serveur web et l'accès HTTP restent cohérents.

! Ce que le certificat protège — et ce qu'il ne protège pas
HTTPS chiffre les échanges et authentifie le nom de domaine pour la connexion concernée. Il ne remplace pas les mises à jour du serveur, une configuration applicative sûre ni la protection des identifiants.

Cette procédure est distillée pour Ubuntu 20.04 avec Apache 2.4.x et les paquets APT mentionnés. Ne mélangez pas ces commandes avec un guide destiné à une autre version d'Ubuntu ou à Nginx.