

Guide d'installation d'Apache2 sous Debian et Ubuntu

Ce guide explique comment installer, démarrer et configurer rapidement **Apache HTTP Server (apache2)** sur des systèmes **Debian** et **Ubuntu** (et dérivés). Les commandes supposent un accès **sudo**.

1. Prérequis

- Une machine Debian/Ubuntu à jour.
- Un compte avec droits **sudo**.
- Accès réseau (internet) pour installer les paquets.
- (Optionnel) Un nom de domaine pointant vers le serveur si vous prévoyez HTTPS.

2. Installation d'Apache2 (APT)

1) Mettre à jour l'index APT :

```
sudo apt update
```

2) Installer Apache2 :

```
sudo apt install -y apache2
```

3) Vérifier l'installation :

```
apache2 -v
```

```
dpkg -l | grep apache2
```

3. Démarrer / arrêter le service (systemd)

- Démarrer: `sudo systemctl start apache2`
- Arrêter: `sudo systemctl stop apache2`
- Redémarrer: `sudo systemctl restart apache2`
- Recharger la config (sans couper les connexions): `sudo systemctl reload apache2`
- Activer au démarrage: `sudo systemctl enable apache2`
- Statut: `systemctl status apache2 --no-pager`

4. Test rapide (page par défaut)

Depuis le serveur, vous pouvez tester avec :

```
curl -I http://localhost
```

Depuis un autre poste, ouvrez `http://<IP_DU_SERVEUR>/` dans un navigateur : vous devriez voir la page « Apache2 Debian Default Page » (ou équivalent Ubuntu).

5. Emplacements importants (Debian/Ubuntu)

- Configuration principale : `/etc/apache2/apache2.conf`
- Ports d'écoute : `/etc/apache2/ports.conf`
- Modules : `/etc/apache2/mods-available/` et `/etc/apache2/mods-enabled/`
- Hôtes virtuels : `/etc/apache2/sites-available/` et `/etc/apache2/sites-enabled/`
- Racine web par défaut : `/var/www/html`
- Logs : `/var/log/apache2/access.log` et `/var/log/apache2/error.log`

6. Créer un site (VirtualHost) simple

Exemple pour un site `example.local` servi depuis `/var/www/example`.

1) Créer le répertoire et une page de test :

```
sudo mkdir -p /var/www/example
sudo chown -R www-data:www-data /var/www/example
echo 'OK - example' | sudo tee /var/www/example/index.html
```

2) Créer le VirtualHost :

```
sudo nano /etc/apache2/sites-available/example.conf

<VirtualHost *:80>
ServerName example.local
DocumentRoot /var/www/example

ErrorLog ${APACHE_LOG_DIR}/example_error.log
CustomLog ${APACHE_LOG_DIR}/example_access.log combined

<Directory /var/www/example>
Options -Indexes +FollowSymLinks
AllowOverride All
Require all granted
</Directory>
</VirtualHost>
```

3) Activer le site, vérifier la configuration, puis recharger Apache :

```
sudo a2ensite example.conf
sudo apache2ctl configtest
sudo systemctl reload apache2
```

(Optionnel) Désactiver le site par défaut si besoin :

```
sudo a2dissite 000-default.conf
sudo systemctl reload apache2
```

7. Activer des modules courants

Quelques modules fréquemment utiles :

- Réécriture d'URL : `sudo a2enmod rewrite`
- En-têtes HTTP : `sudo a2enmod headers`
- Support TLS (certificats) : `sudo a2enmod ssl`

Puis rechargez Apache :

```
sudo systemctl reload apache2
```

8. Ouvrir le pare-feu (UFW – Ubuntu/Debian si activé)

Si ufw est activé :

```
sudo ufw allow 'Apache'
sudo ufw status verbose
```

Selon votre besoin, vous pouvez aussi autoriser uniquement HTTP ou HTTPS via les profils Apache, Apache Full ou Apache Secure (liste avec `sudo ufw app list`).

9. (Optionnel) Activer HTTPS avec Let's Encrypt (Certbot)

Prérequis : un nom de domaine public pointant vers votre serveur, et les ports 80/443 accessibles.

Installer Certbot et l'intégration Apache :

```
sudo apt install -y certbot python3-certbot-apache
```

Générer et installer un certificat (exemple) :

```
sudo certbot --apache -d example.com -d www.example.com
```

Vérifier le renouvellement automatique :

```
sudo certbot renew --dry-run
```

10. Logs et commandes utiles

- Voir les erreurs en direct : `sudo tail -f /var/log/apache2/error.log`

- Voir les accès en direct : `sudo tail -f /var/log/apache2/access.log`
- Voir les messages systemd : `sudo journalctl -u apache2 -e`
- Vérifier la syntaxe : `sudo apache2ctl configtest`
- Voir les VirtualHosts actifs : `sudo apache2ctl -S`
- Voir les modules chargés : `sudo apache2ctl -M`

11. Dépannage (problèmes fréquents)

- **Le port 80/443 est occupé** : vérifier qui écoute avec `sudo ss -ltnp | grep -E ':80|:443'`, puis arrêter le service en conflit.
- **Erreur après modification** : lancer `sudo apache2ctl configtest` et consulter `/var/log/apache2/error.log`.
- **403 Forbidden** : vérifier les droits sur le répertoire du site, la directive `Require all granted` et le bloc `<Directory>`.
- **.htaccess ignoré** : s'assurer que `AllowOverride All` est activé (ou ajuster précisément), et que le module `rewrite` est chargé si vous utilisez des règles de réécriture.
- **Site non accessible depuis l'extérieur** : vérifier pare-feu (UFW/iptables), sécurité cloud (Security Group), et que l'IP/DNS pointe bien vers le serveur.

12. Check-list rapide

- apache2 installé et service active (running).
- Un VirtualHost défini dans `sites-available` et activé via `a2ensite`.
- `apache2ctl configtest` OK.
- Pare-feu ouvert (au minimum 80, et 443 si HTTPS).
- Logs consultables et rotation gérée par défaut (`logrotate`).