

Votre feuille de route LAMP prête à suivre sur Debian 13

Installez une pile Apache, MariaDB et PHP-FPM dans le bon ordre, avec un contrôle après chaque composant. Utilisez cette fiche comme guide de réalisation et de validation avant de déployer une application.

Au sommaire

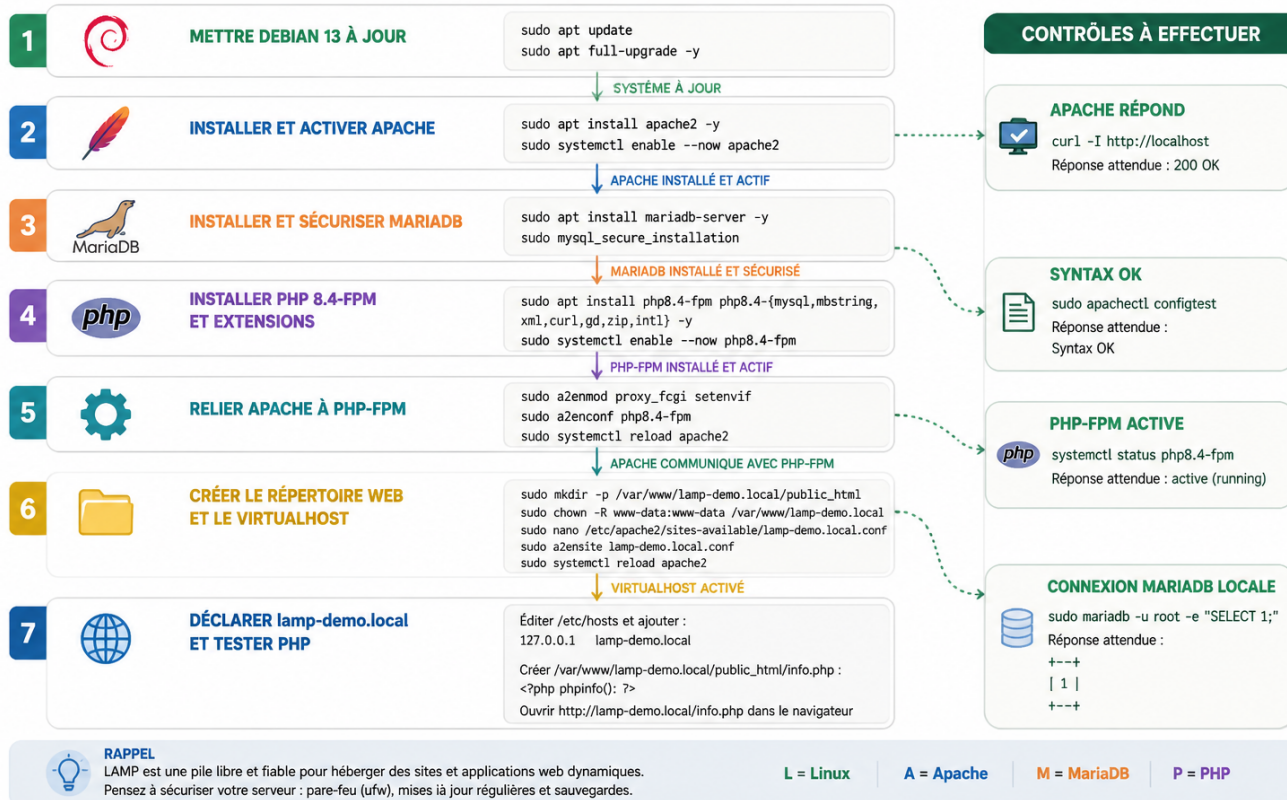
- 1 Avant de commencer
- 2 1. Préparer Debian et installer Apache
- 3 2. Installer et sécuriser MariaDB
- 4 3. Installer PHP 8.4-FPM et relier Apache
- 5 4. Créer le site PHP de validation
- 6 Contrôle final : les trois preuves à obtenir

Avant de commencer

- ☐ **Travailler sur une machine Debian 13 neuve, de test ou disposant d'une sauvegarde ou d'un instantané.**
Les noms de paquets et fichiers de configuration peuvent différer sur d'autres distributions ou versions.
- ☐ **Vérifier l'accès administrateur avec sudo.**
- ☐ **Vérifier l'accès réseau nécessaire au téléchargement des paquets.**
- ☐ **Prévoir un accès SSH si la machine est distante.**
Éviter d'exposer inutilement une connexion administrateur directe.
- ☐ **Ne pas installer MySQL en parallèle de MariaDB sur la même machine.**
- ☐ **Prévoir un nom de domaine uniquement si le serveur doit être publié sur Internet.**

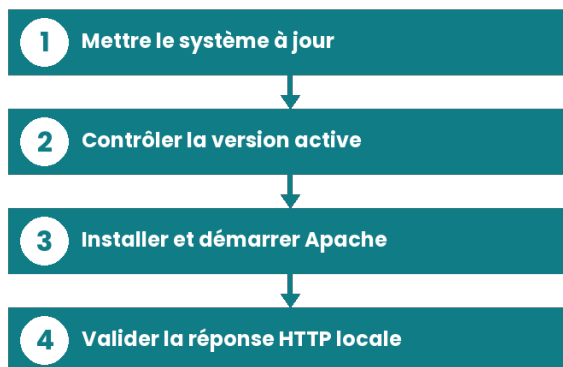
INSTALLER UN SERVEUR LAMP SUR DEBIAN 13

LAMP = Linux (Debian 13) + Apache + MariaDB + PHP



INFOGRAPHIE Schéma des étapes d'installation d'un serveur LAMP Debian 13 avec Apache MariaDB et PHP-FPM

1. Préparer Debian et installer Apache



1 Mettre le système à jour

Exécutez « `sudo apt update` && `sudo apt upgrade -y` », redémarrez avec « `sudo reboot` », puis reconnectez-vous.

2 Contrôler la version active

Utilisez « `cat /etc/debian_version` » et « `uname -r` ». Le premier résultat doit indiquer une version Debian 13.x ; le second affiche le noyau démarré.

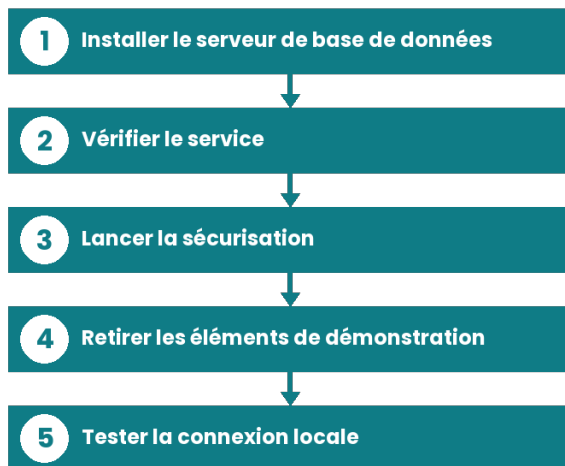
3 Installer et démarrer Apache

Exécutez « `sudo apt install -y apache2` », puis « `sudo systemctl enable --now apache2` ».

4 Valider la réponse HTTP locale

Contrôlez le service avec « `sudo systemctl status apache2 --no-pager` », puis testez « `curl -I http://127.0.0.1` ». Une réponse HTTP 200 ou 403 confirme qu'Apache répond localement.

2. Installer et sécuriser MariaDB



1 Installer le serveur de base de données

Exécutez « `sudo apt install -y mariadb-server` », puis activez le service avec « `sudo systemctl enable --now mariadb` ».

2 Vérifier le service

Lancez « `sudo systemctl status mariadb --no-pager` » et confirmez l'état « active (running) ».

3 Lancer la sécurisation

Exécutez « `sudo mysql_secure_installation` » et lisez chaque question de l'assistant avant de répondre.

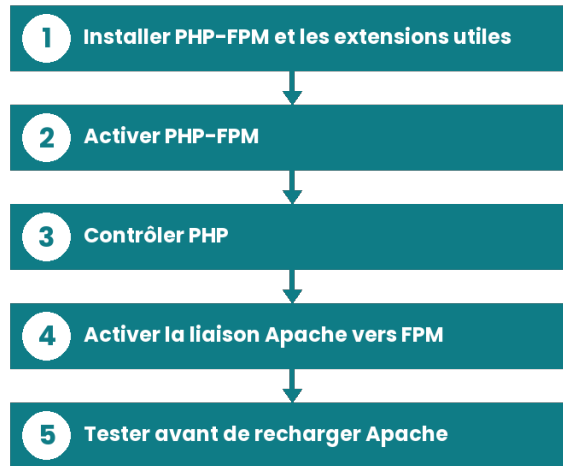
4 Retirer les éléments de démonstration

Refusez les comptes anonymes et la base de test lorsqu'ils sont proposés.

5 Tester la connexion locale

Exécutez « `sudo mariadb -e "SELECT VERSION();"` ». La commande doit retourner une version MariaDB.

3. Installer PHP 8.4-FPM et relier Apache



1 Installer PHP-FPM et les extensions utiles

Installez « php8.4-fpm », « php8.4-mysql », « php8.4-cli », « php8.4-curl », « php8.4-gd » et « php8.4-zip » avec APT.

2 Activer PHP-FPM

Exécutez « sudo systemctl enable --now php8.4-fpm ».

3 Contrôler PHP

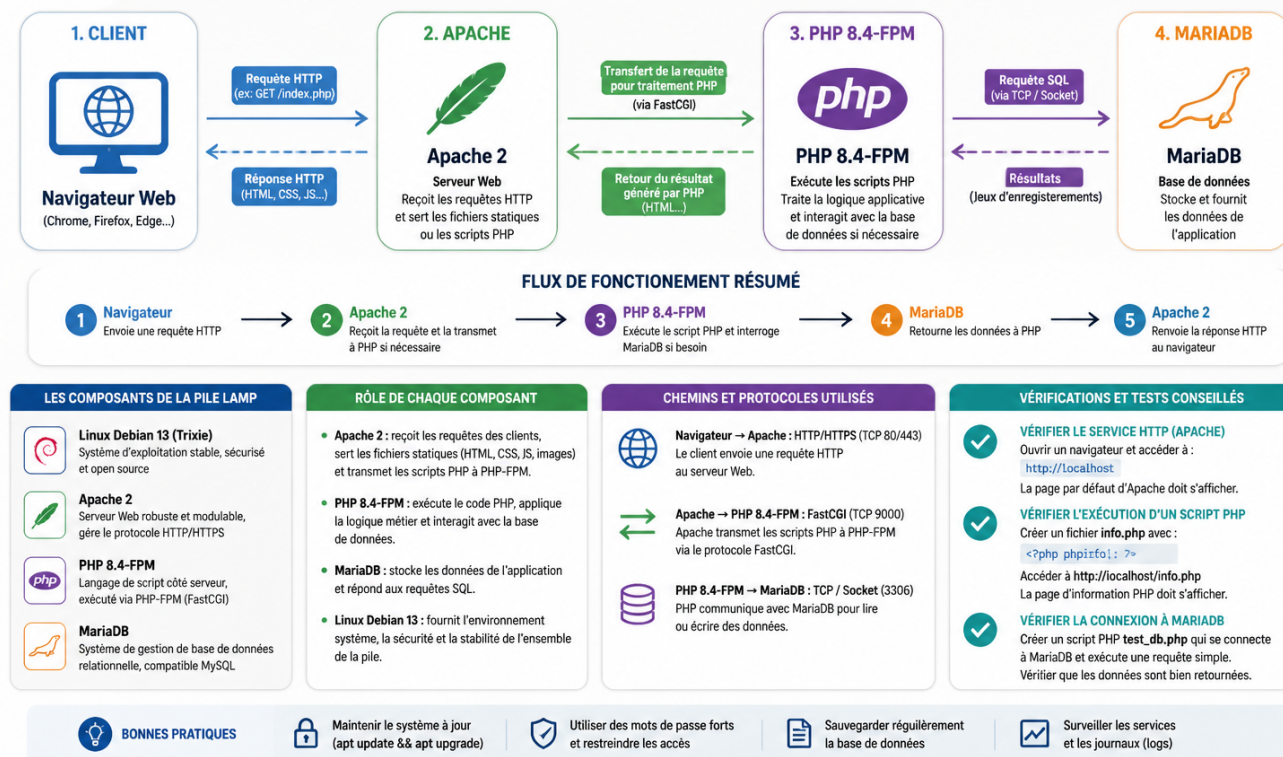
Utilisez « php -v » pour identifier PHP 8.4, puis « sudo systemctl status php8.4-fpm --no-pager » pour confirmer que le service est actif.

4 Activer la liaison Apache vers FPM

Exécutez « sudo a2enmod proxy_fcgi setenvif », puis « sudo a2enconf php8.4-fpm ».

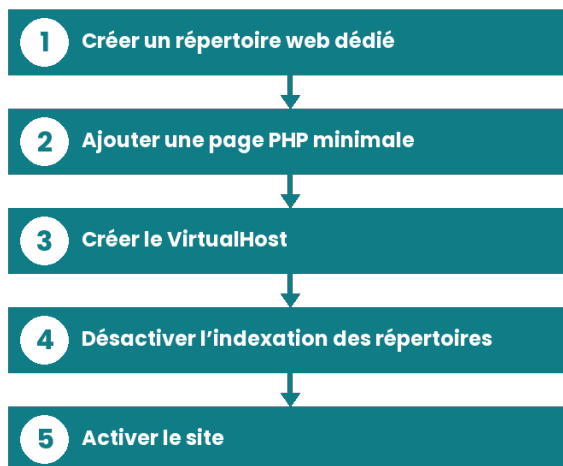
5 Tester avant de recharger Apache

Lancez « sudo apachectl configtest ». Rechargez Apache avec « sudo systemctl reload apache2 » seulement si le résultat est « Syntax OK ».



SCHEMA Chemin d'exécution : le navigateur contacte Apache, qui transmet les scripts à PHP-FPM ; PHP peut ensuite interroger MariaDB.

4. Créer le site PHP de validation



1 Créer un répertoire web dédié

Utilisez le répertoire « `/var/www/lamp-demo/public` » avec des fichiers détenus par root et lisibles par Apache.

2 Ajouter une page PHP minimale

Créez « `index.php` » dans ce répertoire avec un message confirmant que le script est transmis d'Apache à PHP-FPM.

3 Créer le VirtualHost

Créez un site Apache dédié avec « `ServerName lamp-demo.local` », « `DocumentRoot /var/www/lamp-demo/public` » et un bloc `Directory` autorisant l'accès au répertoire.

4 Désactiver l'indexation des répertoires

Conservez l'option « `Options -Indexes +FollowSymLinks` » dans la configuration du répertoire.

5 Activer le site

Désactivez le site par défaut, activez le VirtualHost dédié, puis exécutez à nouveau « `sudo apachectl configtest` » avant de recharger Apache.

Contrôle final : les trois preuves à obtenir

- ☐ **Apache répond en local.**
La requête vers `http://127.0.0.1` renvoie une réponse HTTP.
- ☐ **PHP-FPM est actif et PHP 8.4 est identifié.**
Le service `php8.4-fpm` est « `active (running)` » et « `php -v` » indique PHP 8.4.
- ☐ **Le VirtualHost passe le test de syntaxe Apache.**
La commande « `apachectl configtest` » retourne « `Syntax OK` ».
- ☐ **La page `index.php` du site affiche le message attendu.**
Cela confirme qu'Apache transmet bien le script à PHP-FPM.
- ☐ **MariaDB accepte une connexion locale.**
La requête « `SELECT VERSION();` » retourne une version MariaDB.
- ☐ **Aucun fichier de test sensible n'est laissé exposé avant une mise en production.**

! À ne pas confondre avec une mise en production complète

Une pile LAMP fonctionnelle ne suffit pas à rendre un serveur prêt pour Internet. Traitez séparément l'exposition réseau, les comptes de base de données, les accès administrateur et la suppression des fichiers de test.

Cette procédure est conçue pour Debian 13 et les paquets officiels Apache 2, MariaDB et PHP 8.4-FPM.