

Administration et Configuration Complète du Serveur Apache2

Installation, gestion des modules, hôtes virtuels, PHP et sécurité



Introduction à Apache2

Fondation Apache

Apache est une fondation qui a développé le serveur web Apache httpd depuis plus de 20 ans.

Popularité d'Apache

C'est le serveur web le plus utilisé au monde, reconnu pour sa robustesse et sa polyvalence.

Concurrent principal

Son principal concurrent est Nginx, autre logiciel libre populaire.

Suivi des parts de marché

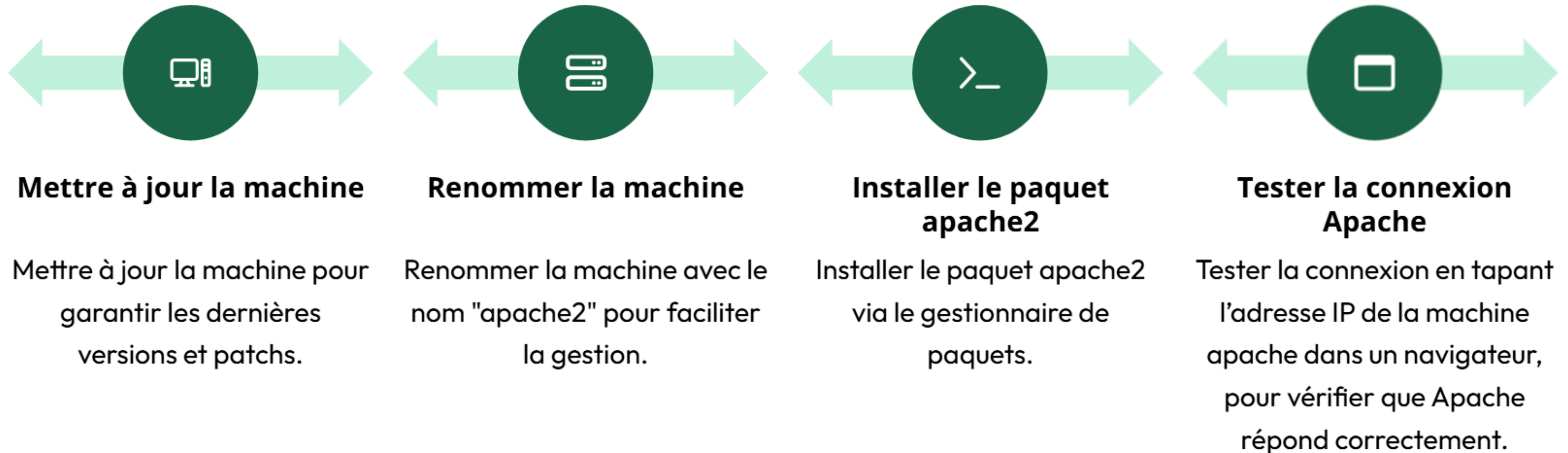
Pour suivre les parts de marché : <https://fr.hostadvice.com/marketshare/>

Site officiel

Site officiel Apache2 : <https://httpd.apache.org/>



Installation d'Apache2 - Mise à jour et Configuration initiale



Comprendre l'arborescence des fichiers Apache2

01
10

Localisation de l'exécutable principal

/usr/sbin/apache2 : exécutable principal (binaire système).



Bibliothèques d'Apache2

/usr/lib/apache2 : bibliothèques d'Apache.



Emplacement des fichiers de configuration

/etc/apache2 : fichiers de configuration (notamment apache2.conf).



Dossiers contenant les fichiers partagés

/usr/share/apache2 : fichiers partagés.



Documentation Apache2

Documentation accessible via `man apache2` (exemple :
`/usr/share/man/man8/apache2.8.gz`).



Fichier apache2.conf et contexte Directory

Le fichier `apache2.conf` présente une vue d'ensemble de l'arborescence et des configurations.

Focus sur le contexte `<Directory>` qui détermine les règles appliquées à certains répertoires.

Gestion des modules MPM dans Apache2

01

Trois modules MPM disponibles

Apache utilise trois modules MPM pour gérer les connexions : `mpm_event`, `mpm_worker`, et `mpm_prefork`.

02

`mpm_event` - Module par défaut

`mpm_event` est optimisé pour la gestion des connexions keepalive et combine processus et threads.

03

`mpm_worker` - Gestion multi-thread

`mpm_worker` est similaire à event mais moins efficace sur keepalive. Chaque processus gère plusieurs threads.

04

`mpm_prefork` - Processus par connexion

`mpm_prefork` crée un processus fils par connexion, fiable mais gourmand en mémoire et moins performant.

05

Un seul module MPM actif

Seul un module MPM peut être actif à la fois dans Apache.

06

Gestion hybride et supervision

Le processus maître (root) supervise plusieurs processus fils (`www-data`) multithreadés répondant aux requêtes clients, équilibrant performance et consommation mémoire pour les charges variables.



Activation et désactivation des modules MPM



Méthode manuelle d'activation/désactivation

Suppression ou création de liens symboliques dans `/etc/apache2/mods-enabled/`

Redémarrage du service Apache2

Vérification avec `ps aux` ou `a2query -M` pour confirmer le module actif



Utilisation de scripts dédiés

Scripts facilitant la gestion des bascules entre `mpm_event`, `mpm_worker` ou `mpm_prefork`

Activation et désactivation simplifiées des modules MPM



Suivi et dépannage avec PID

Détermination des Process ID (PID) des processus Apache avec différentes commandes

Permet un suivi précis et un dépannage efficace

Création d'hôtes virtuels - Bases



Créer un répertoire dédié

Créer un répertoire dédié au site, par exemple “Sitka”, qui contiendra les fichiers du site.



Copier la page d'accueil

Copier la page d'accueil index.html depuis /var/www/html vers ce nouveau répertoire.



Modifier la page HTML

Modifier la page HTML pour afficher “Sitka Alaska” comme titre dans le navigateur.



Organisation des sites

Cette base permet d'organiser proprement les sites par répertoire sur le serveur.

Configuration des hôtes virtuels par nom de domaine



Création et configuration du VirtualHost

Dupliquer le fichier de configuration par défaut 000-default.conf en nommant sitka.conf.
Configurer un VirtualHost avec `<VirtualHost *:80>` pour écouter sur le port 80 de toutes les interfaces.
Spécifier `ServerName` (nom d'hôte unique), `ServerAlias` (aliases complémentaires).
Définir `DocumentRoot` pointant vers le répertoire du site (ex : `/var/www/sitka`).
Ajouter `ServerAdmin` pour l'adresse email du responsable.



Gestion des logs et des erreurs

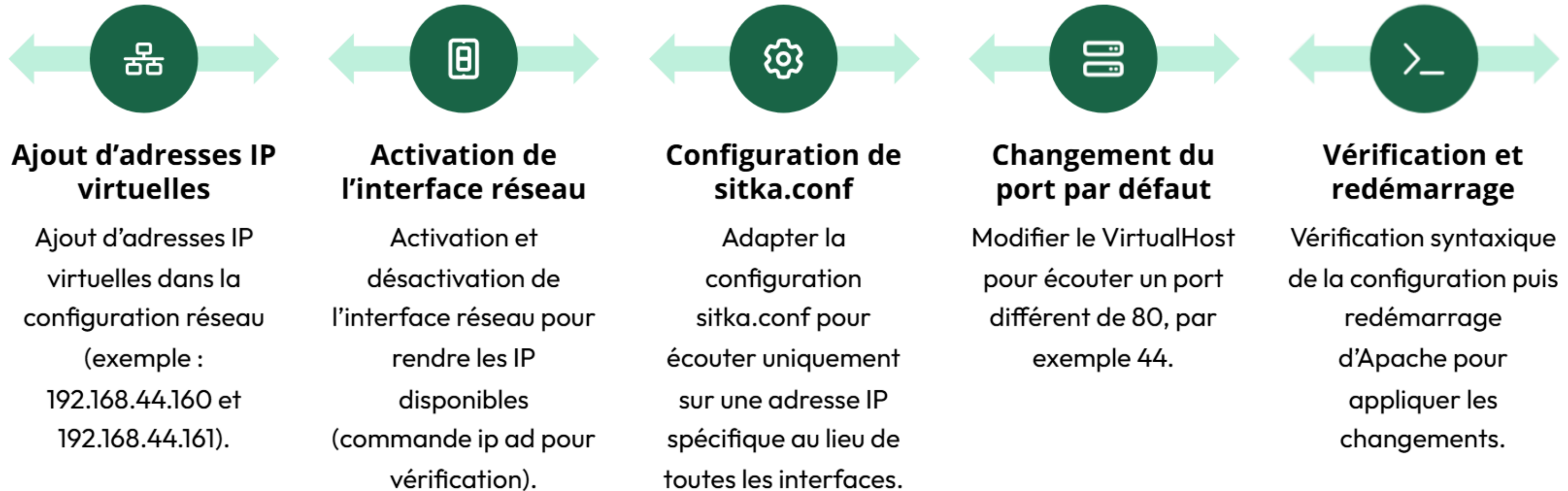
Configurer les logs séparés avec `CustomLog` (format combined) et `ErrorLog` pour une gestion claire des erreurs.



Directives et validation de la configuration

Gérer les directives dans `<Directory />` telles que `ExecCGI`, `FollowSymLinks`, `Includes`, `Indexes`, et `AllowOverride` pour permettre ou restreindre les options.
Vérifier la syntaxe de la configuration (`apachectl -t` ou `apache2ctl -t`) avant activation.
L'activation se fait via `a2ensite` et redémarrage d'Apache.

Création d'hôtes virtuels par adresse IP et changement de ports



Utilisation du PHP dans Apache2 - mod_php versus FPM

mod_php : Intégration directe dans Apache

- Module Apache permettant d'intégrer PHP directement.
- Nécessite mpm_prefork, crée un processus par requête, ce qui impacte les performances.
- Partage de mémoire entre PHP et Apache, ce qui pose des risques de sécurité.
- Supporte une seule version PHP à la fois, tout fonctionne sous l'utilisateur www-data.

FPM (FastCGI Process Manager) : Service indépendant et performant

- Service indépendant d'Apache, utilisant le protocole FastCGI.
- Supporte plusieurs versions PHP simultanément via pools séparés avec utilisateurs et permissions distincts.
- Plus performant et flexible, s'adapte dynamiquement à la charge.

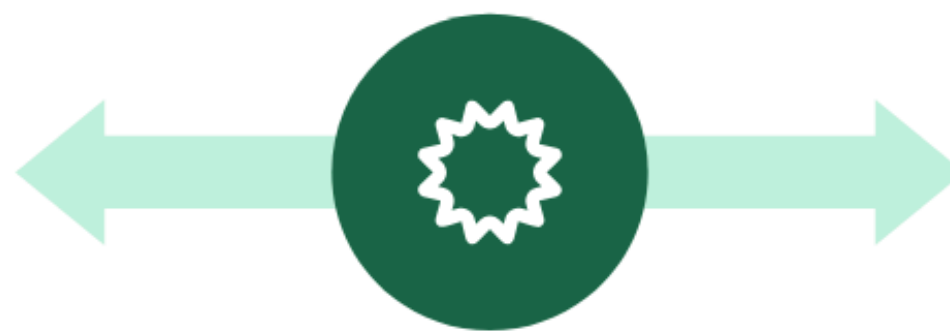
Sécurisation d'Apache2



Masquage des informations sensibles

Par défaut, Apache révèle sa version, modules, et OS via les en-têtes HTTP.

Modifier les directives `ServerTokens` et `ServerSignature` dans `/etc/apache2/conf-enabled/security.conf` ou `apache2.conf` pour limiter cette divulgation.



Sécurisation par SSL

Vérifier la présence du paquet `ssl-cert` et activer le mode SSL.

Créer un certificat autosigné (.pem) incluant clé privée et certificat.

Intégrer la configuration SSL dans le VirtualHost.

Tester la connexion sécurisée via navigateur (https).



Sécurisation des répertoires

Méthode recommandée : utiliser `VirtualHosts` pour appliquer les règles d'accès, nécessite droits root.

Alternative : fichiers `.htaccess` permettant aux non-root de gérer certaines options, mais impactant les performances.

Mise en place d'authentification avec des fichiers de mots de passe et directives `<Directory>` dans les `VirtualHosts`.