PROJET APACHE2

Installation, Configuration et Sécurisation d'un Serveur Web

Formation	BTS SIO Option SISR - 2ème année
Contexte	Entreprise StadiumCompany
Environnement	Debian 11 / Ubuntu Server
Compétences visées	B1.3 - Gérer le patrimoine informatique

SOMMAIRE

1. INTRODUCTION ET CONTEXTE

Dans le cadre de ma formation BTS SIO option SISR, j'ai réalisé ce projet de mise en place d'un serveur web Apache2 pour l'entreprise StadiumCompany. L'objectif était de comprendre le fonctionnement d'un serveur HTTP, sa configuration et sa sécurisation dans un environnement professionnel.

Apache HTTP Server (httpd) est le serveur web le plus utilisé au monde, développé par la fondation Apache. Il existe depuis plus de 20 ans et reste une référence malgré la concurrence de solutions comme Nginx. Ce projet m'a permis d'appréhender concrètement l'administration d'un tel service.

Points clés à retenir :

- Apache httpd domine le marché des serveurs web open source
- La configuration est modulaire et répartie dans /etc/apache2
- Les modules MPM définissent la gestion des connexions
- Les Virtual Hosts permettent d'héberger plusieurs sites sur une même IP

2. INSTALLATION DU SERVEUR APACHE2

2.1 Préparation de l'environnement

Avant toute installation, j'ai commencé par mettre à jour le système et configurer le nom d'hôte de la machine pour faciliter son identification sur le réseau.

Mise à jour des paquets :

```
apt update && apt upgrade -y
```

Configuration du hostname :

```
hostnamectl set-hostname apache2
```

2.2 Installation du paquet Apache2

```
apt install apache2 -y
```

Après l'installation, j'ai vérifié le bon fonctionnement en accédant à l'adresse IP du serveur depuis un navigateur. La page par défaut d'Apache confirme que le service est opérationnel.

3. ARCHITECTURE ET ARBORESCENCE D'APACHE

La commande whereis apache2 m'a permis d'identifier les différents répertoires utilisés par Apache. Cette compréhension de l'architecture est essentielle pour l'administration du service.

Chemin	Description
`/usr/sbin/apache2`	`Exécutable principal du service`
`/usr/lib/apache2`	`Bibliothèques et modules`
`/etc/apache2`	`Fichiers de configuration`
`/var/www/html`	`Racine web par défaut`
`/var/log/apache2`	`Journaux d'accès et d'erreurs`

3.1 Structure du répertoire /etc/apache2

Le fichier principal apache2.conf orchestre l'ensemble de la configuration. Il inclut d'autres fichiers via des directives Include, permettant une organisation modulaire.

- sites-available/ : configurations des sites disponibles
- sites-enabled/ : liens symboliques vers les sites actifs
- mods-available/ : modules disponibles
- mods-enabled/ : modules activés
- conf-available/ et conf-enabled/ : configurations additionnelles

4. GESTION DES MODULES MPM

Les modules MPM (Multi-Processing Modules) déterminent la façon dont Apache gère les connexions entrantes. Un seul module MPM peut être actif à la fois.

4.1 Les trois modules MPM principaux

Module	Caractéristiques
mpm_event	Module par défaut, optimisé pour les connexions keepalive. Gestion hybride processus/threads très performante.
mpm_worker	Multi-threadé, chaque processus gère plusieurs connexions. Bon compromis performance/stabilité.
mpm_prefork	Un processus par connexion, compatible mod_php. Consommation mémoire plus importante.

4.2 Vérification et changement de module MPM

Vérifier le module actif :

```
a2query -M
```

Changer de module (exemple : passer à prefork) :

```
a2dismod mpm_event && a2enmod mpm_prefork && systemctl restart apache2
```

J'ai analysé les processus Apache avec la commande ps aux | grep apache. Le processus maître tourne avec les droits root, tandis que les processus enfants utilisent le compte www-data pour des raisons de sécurité.

5. CONFIGURATION DES VIRTUAL HOSTS

Les Virtual Hosts permettent d'héberger plusieurs sites web sur un même serveur. J'ai configuré un site pour l'entreprise StadiumCompany afin de mettre en pratique ces concepts.

5.1 Création de la structure du site

```
mkdir -p /var/www/stadiumcompany
cp /var/www/html/index.html /var/www/stadiumcompany/
```

5.2 Configuration du Virtual Host

J'ai créé le fichier de configuration /etc/apache2/sites-available/stadiumcompany.conf avec les directives essentielles :

- ServerName : nom de domaine principal du site
- ServerAlias : noms alternatifs
- DocumentRoot : répertoire racine des fichiers web
- ErrorLog et CustomLog : chemins des fichiers de logs

5.3 Activation et test

```
a2ensite stadiumcompany.conf
apache2ctl configtest
systemctl reload apache2
```

5.4 Virtual Host par adresse IP

J'ai également configuré un virtual host basé sur l'adresse IP plutôt que le nom de domaine. Cette méthode nécessite des adresses IP virtuelles configurées dans /etc/network/interfaces.

6. INTÉGRATION DE PHP

Pour exécuter des applications web dynamiques, j'ai intégré PHP au serveur. Deux méthodes existent : mod_php et PHP-FPM.

6.1 Comparaison des méthodes

Critère	mod_php	PHP-FPM
Performance	Moyenne	Excellente
Sécurité	Processus partagé	Isolation des pools
Compatibilité	Apache uniquement	Multi-serveurs
Module MPM	Prefork requis	Event compatible

6.2 Installation de PHP-FPM

```
apt install php-fpm -y
a2enmod proxy_fcgi setenvif
a2enconf php8.2-fpm
```

PHP-FPM communique avec Apache via un socket Unix. Cette configuration offre de meilleures performances et une meilleure isolation de sécurité.

7. SÉCURISATION DU SERVEUR

7.1 Masquage des informations serveur

Par défaut, Apache révèle sa version et le système d'exploitation dans les en-têtes HTTP. J'ai modifié le fichier /etc/apache2/conf-enabled/security.conf pour masquer ces informations.

```
ServerTokens Prod
ServerSignature Off
```

7.2 Configuration HTTPS avec SSL/TLS

J'ai sécurisé les communications en configurant HTTPS avec un certificat auto-signé.

Activation du module SSL :

```
a2enmod ssl
```

Génération du certificat :

```
make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/private/stadiumcompany.pem
```

Le fichier .pem généré contient à la fois le certificat et la clé privée. J'ai ensuite modifié la configuration du virtual host pour écouter sur le port 443 et référencer ce certificat.

7.3 Protection des répertoires

J'ai mis en place une authentification par mot de passe pour protéger certains répertoires sensibles, en utilisant les directives AuthType et AuthUserFile dans la configuration du virtual host.

8. COMPÉTENCES ACQUISES ET BILAN

8.1 Compétences techniques développées

- Installation et configuration d'un serveur web sous Linux
- Compréhension de l'architecture modulaire d'Apache
- Gestion des processus et modules MPM
- Configuration de virtual hosts (par nom et par IP)
- Intégration de PHP via mod_php et PHP-FPM
- Sécurisation : masquage d'informations, SSL/TLS, authentification

8.2 Lien avec le référentiel BTS SIO

Bloc	Compétence
B1	Gérer le patrimoine informatique - Administration de serveurs
B2	Répondre aux incidents - Analyse des logs Apache
B3	Développer la présence en ligne - Hébergement web

8.3 Conclusion

Ce projet m'a permis de maîtriser l'installation et la configuration complète d'un serveur web Apache2. J'ai pu appréhender les enjeux de performance avec les différents modules MPM, les bonnes pratiques d'hébergement avec les virtual hosts, et les aspects critiques de la sécurisation d'un service exposé sur le réseau.

Ces compétences sont directement transférables en environnement professionnel, que ce soit pour l'hébergement d'applications métier ou la mise en place d'infrastructures web sécurisées.