

Prosit 3 : Apache

Mots clés :

- Serveur Apache
- Hôtes virtuels
- Http 404/500
- Développeur frontend/backend
- Log apache
- Document d'apache
- Méthodes GET et POST
- Ressources statiques
- HTTPS
- Outils de développement

DEFINITION

- **Serveur Apache** : Un serveur web open-source qui permet d'héberger des sites web et de gérer les requêtes HTTP. Apache HTTP Server est l'un des serveurs web les plus populaires et il peut être configuré pour servir des pages HTML, PHP, ou d'autres types de contenu web.
- **Hôtes virtuels** : Une configuration permettant de faire fonctionner plusieurs sites web sur un seul serveur. Chaque hôte virtuel correspond à un nom de domaine différent et permet de servir des sites distincts depuis une seule machine physique ou virtuelle, en fonction du nom de domaine demandé.
- **HTTP 404/500** : Ce sont des codes d'état HTTP utilisés pour indiquer des erreurs.
- **404** : Page non trouvée, signifiant que la ressource demandée n'existe pas sur le serveur.
- **500** : Erreur interne du serveur, ce qui indique un problème sur le serveur empêchant la demande d'être traitée.
- **Développeur frontend/backend** :
- **Frontend** : Un développeur frontend s'occupe de la partie visible d'une application web, c'est-à-dire de l'interface utilisateur avec laquelle les utilisateurs interagissent.
- **Backend** : Un développeur backend travaille sur le côté serveur d'une application web, gérant les bases de données, les serveurs, les API et la logique de traitement des données.
- **Log Apache** : Un fichier ou un ensemble de fichiers utilisés pour enregistrer les actions et événements générés par le serveur Apache. Ces logs contiennent des

informations telles que les requêtes HTTP reçues, les erreurs survenues et d'autres informations utiles pour le dépannage et l'analyse de la performance.

- **Document d'Apache** : En général, cela se réfère à la documentation officielle du serveur Apache, qui détaille la configuration, les modules, et l'administration du serveur. Cette documentation est une ressource essentielle pour les administrateurs de serveurs.
- **Méthodes GET et POST** : Ce sont deux des principales méthodes HTTP utilisées pour envoyer des données entre le client et le serveur.
- **GET** : Utilisé pour récupérer des données du serveur. Les informations envoyées avec une requête GET sont généralement visibles dans l'URL.
- **POST** : Utilisé pour envoyer des données au serveur, comme lors de la soumission d'un formulaire. Contrairement à GET, les données sont envoyées dans le corps de la requête et non dans l'URL.
- **Ressources statiques** : Ce terme fait référence à des fichiers qui ne changent pas dynamiquement et sont directement envoyés au client par le serveur. Par exemple : des images, des fichiers CSS, des fichiers JavaScript ou des documents HTML.
- **HTTPS** : HyperText Transfer Protocol Secure, une version sécurisée du protocole HTTP. HTTPS chiffre les données échangées entre le client (navigateur) et le serveur pour garantir la confidentialité et la sécurité des informations, comme les mots de passe ou les données de carte bancaire.
- **Outils de développement** : Des logiciels ou des ensembles d'outils utilisés pour faciliter le développement de sites ou d'applications web. Cela inclut des éditeurs de code, des débogueurs, des outils de gestion de versions comme Git, des navigateurs avec des outils de développement intégrés (comme Chrome DevTools), et d'autres outils d'automatisation et de test.

Contexte :

L'équipe doit configurer un environnement local, efficace et sécurisé

Problématique

Comment configurer un serveur apache avec plusieurs hôtes virtuels

Contraintes :

Créer 2 hôtes virtuels

Activer HTTPS

Utiliser les outils de dev

Maîtriser GET et POST

Utilisation des logs apache

Généralisation

Développement web

Livrable

Fiche de configuration du serveur

Site principale et site pour les ressources statique

Pistes de solution

Utiliser la directive document root d'apache

Utiliser les méthodes http

Configurer les hôtes virtuels

Plan d'action

Documentation Apache/HTTP/HTTPS/hôtes virtuels

Configurer le serveur apache

Utiliser les outils de développement disponible dans le navigateur

Explication

Introduction à Apache

Qu'est-ce qu'Apache ?

Apache HTTP Server, communément appelé Apache, est un serveur web open-source développé et maintenu par la Apache Software Foundation. Il permet de servir des pages web en répondant aux requêtes HTTP des clients, tels que les navigateurs web. Apache est très modulable et peut être configuré pour différentes utilisations grâce à ses nombreux modules.

Historique et évolution

Apache a été initialement lancé en 1995 en tant que successeur du serveur NCSA HTTPd. Son nom provient d'un jeu de mots entre "A Patchy Server", en raison de son développement basé sur des correctifs du serveur HTTPd d'origine. Depuis sa création, Apache a connu plusieurs versions majeures et a été adopté massivement dans le monde entier. Il est resté pendant longtemps le serveur web le plus utilisé avant de voir la montée en popularité de solutions comme Nginx et des services cloud.

Importance d'Apache dans le monde des serveurs web

Apache joue un rôle crucial dans l'hébergement web. Il est largement utilisé par les entreprises, les organisations et les développeurs pour déployer des sites et des applications web. Voici quelques raisons expliquant son importance :

- **Open-source et communautaire** : Apache est un projet librement accessible avec une communauté active qui contribue à son amélioration continue.

- **Fiabilité et stabilité** : Il est reconnu pour sa robustesse et sa capacité à gérer un grand nombre de connexions simultanées.
- **Flexibilité** : Apache offre une large gamme de modules permettant d'ajouter des fonctionnalités comme le support SSL/TLS, la réécriture d'URL et l'authentification.
- **Compatibilité multiplateforme** : Il fonctionne sur différents systèmes d'exploitation, notamment Linux, Windows et macOS.
- **Support étendu** : De nombreux outils et ressources sont disponibles pour aider les administrateurs à configurer et optimiser leurs serveurs.

Installation d'Apache

Prérequis

Avant d'installer Apache, assurez-vous d'avoir :

- Un accès administrateur ou root sur votre système.
- Une connexion Internet active pour télécharger les paquets nécessaires.
- Un système d'exploitation compatible (Linux, Windows...).

Installation sur différents systèmes d'exploitation

Linux (Ubuntu/Debian)

1. Mettez à jour la liste des paquets :

```
sudo apt update
```

2. Installez Apache :

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

3. Démarrez le service Apache :

```
sudo systemctl start apache2
```

4. Activez Apache au démarrage :

```
sudo systemctl enable apache2
```

Windows

1. Téléchargez la version Windows d'Apache depuis le site officiel (Apache Lounge).
2. Extrayez les fichiers dans C : \Apache24\.
3. Ouvrez une invite de commande en mode administrateur et exécutez :

```
httpd.exe -k install
```

4. Démarrez Apache via le service Windows ou la ligne de commande :

```
httpd.exe -k start
```

Vérification de l'installation

Une fois Apache installé, ouvrez un navigateur et entrez <http://localhost> dans la barre d'adresse. Si Apache fonctionne correctement, une page de test devrait s'afficher indiquant que le serveur est bien en service.

Vous pouvez aussi utiliser la commande suivante pour vérifier que le service est actif :

- Sur Linux :

```
sudo systemctl status apache2
```

- Sur Windows :

```
sc query Apache2.4
```

Configuration de base d'Apache

1. Introduction

Une fois Apache installé, il est important de comprendre et de configurer correctement ses paramètres pour assurer un bon fonctionnement, optimiser ses performances et garantir

sa sécurité. Cette section couvre les fichiers de configuration, la gestion des hôtes virtuels et la gestion des modules.

2. Fichiers de configuration principaux

Apache repose sur plusieurs fichiers de configuration situés généralement dans `/etc/apache2/` (Debian/Ubuntu) ou `/etc/httpd/` (CentOS/RHEL). Voici les fichiers et dossiers principaux :

2.1. Fichiers généraux

- **httpd.conf** : Fichier principal de configuration sur certaines distributions comme CentOS.
- **apache2.conf** : Utilisé par Debian et Ubuntu comme fichier de configuration principale.
- **envvars** : Définit les variables d'environnement pour Apache.

2.2. Dossiers de configuration

- **conf-available/** et **conf-enabled/** : Contiennent les fichiers de configuration optionnels et ceux qui sont activés.
- **mods-available/** et **mods-enabled/** : Permettent d'activer ou désactiver des modules Apache.
- **sites-available/** et **sites-enabled/** : Gèrent les fichiers de configuration des hôtes virtuels.

3. Configuration des hôtes virtuels (Virtual Hosts)

Les hôtes virtuels permettent d'héberger plusieurs sites web sur un même serveur Apache.

3.1. Création d'un hôte virtuel

1. Créer un répertoire pour le site :

```
sudo mkdir -p /var/www/example.com/public_html
```

2. Définir les permissions :

```
sudo chown -R $USER:$USER /var/www/example.com/public_html
```

3. Créer un fichier de configuration dans `/etc/apache2/sites-available/` :

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

Ajouter la configuration suivante :

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /var/www/example.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

4. Activer l'hôte virtuel :

```
sudo a2ensite example.com.conf
sudo systemctl restart apache2
```

4. Gestion des fichiers et des répertoires

Apache utilise une structure de répertoires spécifique pour organiser les fichiers de configuration et les fichiers web.

4.1. Structure des répertoires d'Apache

- **/etc/apache2/** (Debian/Ubuntu) : Contient les fichiers de configuration.
- **/var/www/** : Répertoire par défaut pour les fichiers web.
- **/var/log/apache2/** (Debian/Ubuntu) : Contient les logs d'Apache.

Dépannage

Malgré une bonne configuration, des erreurs peuvent survenir avec Apache. Cette section aborde les techniques de diagnostic et de résolution des problèmes courants.

Vérification des journaux

Les journaux d'Apache permettent d'identifier les erreurs et problèmes de fonctionnement :

- **Journaux des erreurs :**

```
sudo tail -3 /var/log/apache2/error.log # Debian/Ubuntu
```

- **Journaux des accès :**

```
sudo tail -3 /var/log/apache2/access.log # Debian/Ubuntu
```

Vérification de la syntaxe de configuration

Avant de redémarrer Apache après une modification, testez la syntaxe des fichiers de configuration :

```
apachectl configtest Debian/Ubuntu
```

Si une erreur est détectée, Apache indiquera le fichier et la ligne concernés.

Vérification du statut du service

Vérifiez si Apache est bien en cours d'exécution :

```
sudo systemctl status apache2 Debian/Ubuntu
```

Redémarrer ou recharger Apache

Après une modification de configuration, appliquez les changements sans interrompre le service :

```
sudo systemctl reload apache2 Debian/Ubuntu
```

En cas de problème, un redémarrage complet peut être nécessaire :

```
sudo systemctl restart apache2
```

 Debian/Ubuntu

Problèmes courants et solutions

Apache ne démarre pas	Erreur de syntaxe dans la configuration	Vérifier avec <code>apachectl configtest</code> ou <code>httpd -t</code>
Page blanche ou erreur 500	Mauvaise configuration de permissions	Vérifier les permissions avec <code>ls -la</code>
Erreur 403 Forbidden	Mauvais droits d'accès aux fichiers	Vérifier <code>Require all granted</code> dans la configuration
Site non accessible	Mauvaise configuration DNS ou <code>ServerName</code> incorrect	Vérifier <code>/etc/hosts</code> et <code>ServerName</code> dans la config
Apache écoute sur le mauvais port	Mauvaise configuration de <code>Listen</code>	Modifier <code>Listen 80</code> ou <code>Listen 443</code> dans <code>ports.conf</code>

Outils de diagnostic

- **curl** : Tester les réponses HTTP :

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

- **netstat** : Vérifier si Apache écoute sur les bons ports :

```
netstat -tulnp | grep apache
```

- **ss** : Alternative à netstat pour voir les ports ouverts :

```
ss -tulnp | grep apache
```