
TP réseaux 4 : Installation et configuration d'un serveur Web Apache

Objectifs

- Installer, configurer, lancer et administrer le serveur Web Apache sous Linux

Données de base

- machine fonctionnant sous Linux qui hébergera le serveur Web
- fichiers de configuration et de lancement du serveur Web Apache
- la commande `grep -r chaine *` permet de rechercher la présence de la chaîne de caractères *chaine* dans tous les fichiers du répertoire courant et récursivement dans tous les répertoires rencontrés

Documentation

- Documentation complète [Apache version 2.2](#)
- Présentation succincte du [protocole HTTP](#)
- Les fichiers de configuration de Apache

Principe d'un serveur Web

Un serveur Web est une machine sur laquelle est exécutée un programme en arrière plan fournissant des ressources en utilisant le protocole HTTP. Ce programme, ou *daemon*, attend qu'un client (navigateur) envoie une demande de connexion. Lorsqu'un client est connecté, le serveur traite les requêtes du client (envoi des informations demandées : page HTML, image...). Il est à noter que les principes qui régissent le fonctionnement d'un serveur Web sont globalement les mêmes que ceux de n'importe quel autre serveur.

Installation/désinstallation de Apache

Afin de pouvoir utiliser Apache, il faut au préalable installer tous les fichiers dont il a besoin. En général, l'installation se fait à l'aide d'un gestion de paquets (*zipper*, *apt*, ...). qui place les bons fichiers aux bons endroits sur la machine cible. À l'inverse, une désinstallation supprime tous les fichiers installés et laisse le système tel qu'il était avant l'installation (sauf si vous avez rajouté des fichiers ou répertoires supplémentaires auquel cas il faut les supprimer manuellement).

Q1 : Donner la commande permettant d'installer "apache2" l'implémentation du serveur web Apache sous Linux ?

Q2 : Donner la commande permettant de dé-installer "apache2" ?

Q3 : Donner le commande permettant de savoir si le service "apache2" tourne en tâche de fond ?

Q4 : Donner le commande permettant d'arrêter un service ?

Remarque : Vous **devez** à la fin du TP désinstaller le module Apache (quel que soit votre état d'avancement) et effacer tous les fichiers et répertoires.

Étant donné que le TP est long, il est vivement conseillé de **sauvegarder** vos fichiers de configuration (de préférence sur un support personnel ou sur votre compte étudiant).

- **Avant de commencer à répondre à une question, lisez la jusqu'au bout.**
- **Dans les systèmes Unix, les fichiers de configuration se trouvent souvent à partir du répertoire /etc.**
- **Pour la plupart des questions, les réponses se trouvent souvent dans les fichiers de configuration, ces derniers ont très souvent l'extension .conf.**
- **Lorsque vous observez ou modifiez la configuration du serveur donnez la directive concernée (son nom) et sa valeur (nouvelle et/ou ancienne).**
- **Une modification dans la configuration n'est prise en compte qu'au redémarrage du serveur ou le rechargement de sa configuration.**
- **Pour l'édition des fichiers de configuration, il est fortement conseillé d'utiliser l'[éditeur vi](#).**
- **À la fin de la séance, désinstallez tous les paquets installés et supprimez tous les fichiers et répertoires créés.**

Observation de la configuration d'un serveur Web Apache

Q 3bis : Quel est le nom du répertoire racine du serveur Apache : répertoire à partir duquel se trouve la configuration du serveur ?

Quelle directive de configuration permet de changer cette valeur par défaut ?

Q4 : Quel est le nom et la localisation du fichier de configuration principal ?

Quels sont les fichiers de configuration secondaires inclus dans le fichier de configuration principal ? Lister tous les fichiers de configuration inclus dans le fichier principal et expliquer en une phrase leur contenu.

Quels sont les fichiers de configuration secondaires permettant :

- La définition des variables d'environnement d'apache2 ?
- La configuration générale du serveur ?
- La configuration des réponses en cas d'erreur (dans le répertoire conf.d) ?
- La configuration des hôtes virtuels (les sites activés) ?

Q5 : Quel est le nom et la localisation du fichier des rapports d'erreurs ?

Quelle est la directive qui permet de le spécifier ?

Est-ce que ce fichier existe à l'heure actuelle sur votre machine ? Pourquoi ?

Q6 : Quel est le répertoire racine du site web hébergé par le serveur ?

Quelle directive de configuration définit ce chemin ?

Dans quel fichier de configuration se trouve elle ?

Manipulation

Q7 : Démarrez, arrêtez puis démarrez à nouveau le serveur web. Notez le PID du serveur (qui peut être récupéré à partir de `/var/run/apache2.pid`) après chaque action.

Donnez la directive qui permet de modifier le chemin et le nom du fichier contenant le PID.

Q8 : Lancez la commande `ps -ef | grep apache2`. La commande `ps` affiche les informations sur les processus actifs, l'option `e` permet d'afficher les informations sur tous les processus et l'option `f` permet d'afficher les informations complètes sur les processus. Le résultat de la commande `ps -ef` est envoyé en entrée de la commande `grep apache2`. La commande `grep` permet d'extraire les lignes contenant la chaîne de caractères `apache2`. Le résultat de la commande affiche un tableau contenant la description des processus. La première colonne donne le nom de l'utilisateur qui a lancé le processus rattachés au processus, la seconde le PID (numéro du processus), la troisième le PPID (PID du processus père), et la dernière le nom du programme.

Notez et commentez le résultat que vous obtenez, dites notamment à quoi correspond, selon vous, chaque processus (excepté celui dédié au `grep`).

Déduisez-en avec quels droits (`nom_utilisateur`) sont accédées et/ou exécutées les pages web sur le serveur et vérifiez que cet utilisateur est bel et bien déclaré dans le système.

Quelle directive permet de spécifier cet utilisateur, celle qui permet de définir son groupe et dans quel fichier de configuration elles sont définies ?

Q9 : Créez une page HTML avec un contenu minimal (vos noms, prénoms et groupe en gras) et placez la au bon endroit dans l'arborescence du site afin qu'elle soit visible en accédant à votre serveur.

Testez le fonctionnement du serveur de 2 manières différentes et **surtout** décrivez vos tests.

Q10 : Lancez la commande `tail /var/log/apache2/access_log`. Commentez le résultat.

Quelle directive définit cette valeur ?

Configuration

Pour toutes les configurations demandées ci-dessous, **donnez les directives concernées, leur valeur (courantes ou nouvelles) ainsi que le fichier de configuration dans lequel elles sont déclarées.**

Q11 : Quels sont les noms des fichiers servant d'index – pages par défaut – à un répertoire (quand l'URL demandée est un répertoire, le serveur retournera le fichier index qu'il trouvera dans le répertoire en guise de page par défaut) ?

Rajoutez dans la liste des fichiers index « `perso.html` », effectuez les tests adéquats et **surtout** décrivez vos tests.

Q12 : Quel est le nom du répertoire permettant à un utilisateur du système de publier des informations sur le serveur web directement à partir de son compte ? Ce type d'hébergement web est appelé « `public_html` ». En cherchant cette chaîne de caractères trouvez quelle directive gère cette configuration et dans quel fichier de configuration elle est définie ?

Testez que votre serveur web configuré par défaut pour publier les pages personnelles des utilisateurs du système et décrivez votre test. Il vous sera nécessaire de créer un utilisateur (nom d'utilisateur : `test` ; mot de passe : `test`) puis de le supprimer une fois le TP terminé :

- `useradd -m test` (déclaration d'un nouvel utilisateur nommé « `test` » dont le répertoire racine sera `/home/test/`)

- `passwd test` (permet de saisir le mot de passe de l'utilisateur « `test` » et d'activer son compte)

- `userdel -r test` (suppression de l'utilisateur « `test` » et de son répertoire racine)

Changez la configuration du serveur web pour que seul l'utilisateur « `invitetest` » puisse publier ses pages personnelles sur le site web. Donnez la nouvelle configuration et **surtout** décrivez vos tests.

Q13 : Modifiez le serveur pour qu'il ne prenne en compte que les demandes effectuées sur le port 1150 (et non plus sur le port 80, le port HTTP par défaut). Testez, décrivez votre test puis revenez à la configuration précédente.

Q14 : Lorsqu'une erreur survient lors d'une requête http, le serveur retourne un code d'erreur et éventuellement le message correspondant. Il est également possible que le serveur transmette une page HTML adéquate avec le message d'erreur. Donnez le chemin où se situent ces pages. Remplacez les pages par défaut des erreurs 404 et 403 par vos propres pages (les plus simples possibles mais contenant au moins le nom du binôme en haut de la page) que vous nommerez respectivement `404.html` et `403.html`. Modifiez la configuration du serveur en conséquence. Testez et surtout décrivez votre test.

Q15 : Modifiez le nom du serveur (donnez lui comme nouveau nom « `polytechtours` »).

Pour qu'un client s'y connecte, il faut déclarer que le serveur correspondant à ce nom se trouve sur votre machine. Pour cela éditez le fichier `/etc/hosts` et modifiez la ligne adéquate ainsi : `127.0.0.1 localhost.localdomain localhost polytechtours`

Testez et décrivez votre test.

Q16 : Configurez votre serveur pour qu'il puisse héberger un autre site (site virtuel) appelé «test» dont l'arborescence (répertoire racine du site) se trouve dans le répertoire /home/test/public_html, les fichiers journaux d'erreur et d'accès (test-error_log et test-access_log) dans /var/log/apache2 et l'administrateur du site est test@localhost. Donnez le nom, le contenu et le chemin du fichier de configuration créé à cet effet. Vous créez un fichier spécifique pour cet hôte virtuel /etc/apache2/sites-enabled/virtualhosttest. Testez et décrivez votre test.

Q17 : Vérifiez que le serveur virtuel est configuré pour pouvoir exécuter des programmes dans le répertoire /var/www/cgi-bin/ dont vous donnerez le chemin d'accès.

Écrivez le programme suivant en personnalisant le dernier « printf » et sauvez-le dans le fichier testcgi.c :

```
#include<stdio.h>

int main() {
    printf("Content-Type: text/html\n\n") ;
    printf("<CENTER>Ca marche!!!</CENTER>") ;
    printf("<CENTER><B>Nom Pr&eacute;nom
Groupe</B></CENTER>") ;
    return 0 ;
}
```

Compilez ce programme avec gcc testcgi.c -o testcgi.exe, déplacez-le dans le répertoire /var/www/cgi-bin/ de l'arborescence de votre site, testez et décrivez votre test.

Quels sont les droits de testcgi.exe ? Que ce passe-t-il si ces droits sont changés en 644 ?

Q18 : Le PHP est un langage de programmation permettant de générer dynamiquement des pages HTML sur le serveur. Modifiez le serveur pour qu'il interprète les fichiers php. Vous aurez besoin d'installer le paquetage libapache2-mod-php5 ainsi que le paquetage php5. Redémarrer le serveur apache. Testez le bon fonctionnement du module PHP en créant une page phpinfo.php. Cette page contiendra le code suivant :

```
<?php print_r (phpinfo()); ?>.
```

Montrer que cette page est bien interprétée ?

Authentification

Q19 : Créez dans l'arborescence du site Web un répertoire /var/www/privé avec les droits 755 et placez dedans le fichier perso.html utilisé précédemment (Q11). Vérifiez qu'avec la configuration actuelle cette page est accessible (donnez notamment l'URL qui permet de la visualiser).

Ensuite, configurez le serveur pour que les pages mises dans le répertoire privé ne soient accessibles qu'après authentification. Pour y parvenir, vous pouvez soit utiliser un fichier .htaccess soit un bloc <Directory> en y mettant les bonnes directives portant sur le répertoire à protéger. Le bloc <Directory> est présent dans votre fichier «000-default». Vous devez créer les données d'authentification (définition d'un nom d'utilisateur : TP4 et d'un mot de passe : authTP4 en utilisant la commande htpasswd2) et les placer dans un endroit inaccessible depuis le Web

(dans le répertoire de configuration du serveur par exemple.
Décrivez la configuration réalisée.

Voici un exemple de bloc <directory> :

```
<Directory "/var/www/privé">  
  AuthType Basic  
  AuthName "Pages protégées"  
  AuthUserFile /etc/apache2/TP4.auth  
  Require user TP4  
</Directory>
```

Voici un exemple d'utilisation de htpasswd2 :

Création du fichier avec le mot de passe : htpasswd2 -bc /etc/apache2/tp4.auth TP4
authTP4.

Q20 : Testez l'authentification et vérifiez que la configuration que vous avez fait ne concerne pas la visualisation des pages perso (Q12) ni des pages du répertoire principal du site (Q6)