

INSTALLATION DU SERVEUR WEB APACHE SOUS LINUX



Objectif:

L'objectif du TP est d'installer Apache sur un système Linux, de s'assurer le bon fonctionnement des différents éléments et de modifier quelques éléments de configuration.

Présentation :

La configuration de l'ensemble Apache, PHP, MySQL n'était pas spécialement aisée avec les distributions plus anciennes, aujourd'hui les paquetages facilitent la tâche

Rappelons brièvement que:

- Apache est un serveur Web (le logiciel qui ``envoie`` des pages html à un navigateur). Il s'agit du serveur le plus utilisé actuellement sur le Web puisqu'il représente environ 2/3 des serveurs installés.
- PHP est un langage de programmation interprété. Correctement interfacé avec Apache il permet au serveur de fournir des pages dynamiques gérées en fonction des besoins du client. En clair, la page n'est plus un document statique mais peut évoluer, afficher des informations différentes selon les souhaits de l'utilisateur.
- MySQL est un gestionnaire de bases de données assez puissant et rapide

Partie 1 : Installation des paquetages logiciels

Pour installer le serveur Web Apache2, se connecter root dans un terminal (grâce à la commande su):

```
apt-get update  
apt-get install apache2 apache2-doc apache2.2-common apache2-mpm-  
prefork (apache2-mpm-prefork est requis pour le bon fonctionnement de  
PHP)
```

Ce qui installera Apache ainsi que sa documentation. D'autres paquets sont prérequis qu'il est nécessaire d'installer et d'autres sont suggérés

- apache2 écoute sur le port 80 (pour changer il suffit d'intervenir dans le fichier /etc/apache2/ports.conf)
- Après divers paramétrages, et un avertissement pas grave (il manque un nom de domaine pour le serveur) il y a démarrage de apache2

Partie 2 : Premier test d'Apache

Le fonctionnement d'Apache est matérialisé par la mise en route d'un démon (programme qui tourne en tâche de fond) nommé **httpd**.

Pour mettre en route, arrêter, vérifier etc. ce service tapez la commande `apache2ctl` (voir page de manuel de `apache2ctl`).

Exemple : `apache2ctl start`, `apache2ctl restart`, `apache2ctl stop`

En principe l'installation d'Apache a modifié les fichiers de configuration du lancement de LINUX. Le démon httpd devrait donc démarrer systématiquement et automatiquement à chaque boot. L'installation a par ailleurs démarré immédiatement ce démon, il n'est donc pas nécessaire de rebooter comme on le ferait avec un autre système d'exploitation.

1. Où les fichiers ont-ils été installés ?

Après une mise à jour par la commande `updatedb`, observer les résultats de :

```
locate apache2 | less.
```

En déduire la localisation des principaux fichiers qui concernent Apache2 :

- exécutable : `/usr/..`
- configuration : `/etc/..`
- logs : `/var/log/..`
- racine du site WEB : `/var/..`
- modules : `/usr/lib/..`
- documentation `/usr/share/ ...`

2. les répertoires de configuration

Parcourir les répertoires de configuration.

En particulier, parcourir le répertoire `/etc/apache2/mods-available`. Peut-on comprendre les rôles attribués à ces fichiers. Constatez qu'aucun ne concerne PHP.

3. Examiner les processus en exécution en mémoire :

Lancer `ps aux | grep apache2`, puis `pstree`. Comment interpréter les processus obtenus ?

Utiliser les commandes permettant d'arrêter et de relancer le serveur Apache. Vérifier la liste des processus à chaque fois.

4. Accès au site local

Accéder à la page d'accueil de votre installation :

`http://localhost` ou `http://(adresse IP)` ou `http://dellx` (si dellx est le nom donné à votre machine)

Observer le contenu du répertoire `/var/www/`

Conclure sur la page par défaut

5. Consultation de la documentation

Elle a été installée mais n'est pas accessible à partir de la page d'accueil. Elle est disponible ici : <http://localhost/manual/>

L'accès à la documentation est configurée avec le fichier :

`/etc/apache2/conf.d/apache2-doc :`

Observer le contenu de ce fichier:

- En déduire l'emplacement dans votre arborescence de la documentation
- Qui a de droit de consulter la documentation d'apache ?

Modifier la langue par défaut de la documentation.

<u>Partie 3 : Quelques éléments de configuration :</u>

Introduction :

Les fichiers de configurations d'Apache2 se trouvent dans les répertoires `/etc/apache2/`, `/etc/apache2/sites-available/` et `/etc/apache2/conf.d/`. Les fichiers de logs ou d'erreurs se trouvent dans le répertoire : `/var/log/apache2`.

Pour les différents tests et réglages demandés, il est possible de travailler entièrement en ligne de commande

Faites dès le début une sauvegarde des fichiers de configurations d'origine que vous allez modifier :

Exemple : `cp /etc/apache2/apache2.conf /etc/apache2/apache2.conf.old`

A chaque intervention sur le fichier de configuration :

- vérifier la syntaxe de ce fichier en passant la commande : `apache2ctl -t`
- faire prendre en compte immédiatement les modifications : `/etc/init.d/apache2 reload` (ce qui force la relecture par Apache de la configuration)

Pour chercher l'emplacement de clauses particulières, il est commode d'interroger les fichiers de configuration avec l'utilitaire `grep`. Par exemple pour chercher les numéros de lignes où se trouvent les mots `DirectoryIndex`, puis `alias` (sans tenir compte de la casse), on écrira :

```
grep -n DirectoryIndex /etc/apache2/apache2.conf
grep -ni alias /etc/apache2/apache2.conf
```

1. Fixer le nom du serveur

Ajouter la directive `ServerName` avec un nom du genre `dellx`, où `x` est le numéro de la machine hôte dans la section 1 du fichier

`/etc/apache2/apache2.conf`.

Tester à partir d'un client, après avoir renseigné le fichier `/etc/hosts`

2. La page d'accueil et les pages par défaut

Lors de l'installation d'apache un site par défaut a été installé :

Observer le contenu du répertoire `/etc/apache2/sites-enabled/`

Ouvrir le fichier dans ce répertoire :

Observer la directive `DocumentRoot`

Observer l'alias `/doc/` défini

le tester

Interpréter et tester les autorisations liées à l'alias `/doc/`

Renommez `index.bak` la page d'accueil du site par défaut.

Tester l'accès à cette page par <http://dellx>. Pouvez vous l'obtenir comme précédemment ? Pourquoi ?

Qu'obtenez-vous en passant l'URL : `http://dellx/index.bak` ?

Créez rapidement une petite page HTML portant le nom `accueil.html`

L'objectif est qu'elle devienne la page d'accueil du serveur.

Rechercher la directive `DirectoryIndex` dans l'un des fichiers de configuration des modules activés et la modifier convenablement. Vérifier votre configuration.

3. Une configuration décentralisée

La configuration d'Apache est réalisée à partir de plusieurs fichiers. Le fichier principal étant `/etc/apache2/apache2.conf`, des directives `Include` permettent d'inclure d'autres fichiers de configuration.

Lister les fichiers et répertoires de configuration qui sont inclus.

Lister avec `ls -l` le contenu du répertoire `/etc/apache2/sites-enabled/`

En déduire l'endroit exact des fichiers de configuration des sites validés

Les commandes `a2ensite` et `a2dissite` permettent de créer ou détruire les liens symboliques dans le répertoire `sites-enabled` :

A l'aide de la page de manuel des commandes précédentes désactiver le site par défaut.

Lister avec `ls -l` le contenu du répertoire `/etc/apache2/sites-enabled/`

Tester la connexion à votre serveur Web. On ne devrait plus avoir accès au site par défaut.

Réactiver le site par défaut

Vérifier que votre site refonctionne

4. L'arborescence du site

La directive `DocumentRoot` permet de fixer le répertoire racine du site Web, c'est-à-dire le répertoire qui contient la page par défaut d'un site WEB.

On souhaite accéder sur votre serveur Web au fichier `toto.html` qui contient « coucou » grâce à l'URL suivante : <http://dellx/rep/toto.html> »

Créer ce qu'il faut dans votre arborescence pour répondre à la contrainte.

Que se passe-t-il si on fait référence à l'URL <http://dellx/rep/> ?

Renommer le fichier `toto.html` en `index.html` et reprendre la question précédente.

Modifier la directive `DocumentRoot` pour que la racine de votre serveur Web soit le répertoire `REP`.

Partie 4 : Suivi du journal des accès

`access.log` pour les requêtes traitées (qu'elles soient réussies ou non)

`error.log` pour noter les erreurs de fonctionnement du serveur

Visualiser les logs de connexion à votre serveur. Identifier les derniers accès

Visualiser les logs d'erreurs de votre serveur WEB. Justifier les erreurs.

Quel est le rôle de la commande `tail` (man `tail`) ?

Lancer la commande : `tail -f /var/log/apache2/access.log > /dev/tty2 &`
Observer le contenu de la console tty2 .
Demander aux voisins de jouer les rôles de clients WEB et d'interroger votre serveur `http://dellx/..`
Observer les lignes de access.log générées "en direct"
Interpréter les champs de chaque ligne, en particulier repérer les renseignements qui concernent le client et les codes de retour des diverses requêtes (200 réussie, 404 "not found" ...)

Partie 5 : Analyse Réseau

On souhaite observer les trames qui circulent sur le réseau lorsqu'un navigateur se connecter à votre serveur WEB.

Pour cette partie, on utilisera un navigateur Internet et un analyseur de trames Ethernet (Ethereal/wireshark).

Remarque : On rappelle que les applications communiquant à travers le protocole TCP (et UDP) sont identifiées par un numéro de port TCP.

Installer l'analyseur Ethereal sur votre poste (`apt-get install ethereal`)
Lancer une capture de trames Ethernet afin de pouvoir visualiser l'ensemble des trames échangées entre le client et le serveur pour la consultation d'une page Web.
Identifier les trames propres au protocole TCP et leur sens de transfert
Identifier les trames liées au protocole http et leur sens de transfert
Identifier les requêtes et les réponses http
Noter les ports TCP associés au client et au serveur.

Après avoir ouvert un nouveau navigateur :

Recommencer les étapes précédentes
Noter les ports TCP associés au client et au serveur.
Conclusion