
Exam : Installation et configuration d'un serveur Web – Apache

Objectifs du TP

Configurer, lancer et administrer le serveur Web Apache sous Linux

Outil et documentation:

- Machines virtuelles Fedora 12, XP et Slitaz
- Fichiers de configuration et de lancement du serveur Web Apache

Installation du serveur Apache

Afin de pouvoir utiliser Apache 2, il faut au préalable installer tous les fichiers dont il a besoin. En général, l'installation se fait via des modules ou paquetages dédiés appelés 'rpm' pour la distribution Fedora de Linux. L'installation d'un rpm 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.

* Pour installer un paquetage rpm : **rpm -ivh nom_du_paquetage_avec_extension_rpm**

* Pour désinstaller un paquetage rpm : **rpm -e nom_du_paquetage_sans_extension_rpm**

Fedora fournit aussi un gestionnaire de paquets « yum » donc l'interface graphique est « aptitude ». Il permet de télécharger et d'installer d'une façon très simple et dynamique, lorsqu'on est connecté à internet, les paquetages rpm cibles ainsi que leurs dépendances.

Q. 1 Procéder à l'installation d'Apache 2 ainsi que de son manuel à l'aide de la commande « yum install httpd httpd-manual ». Confirmer l'installation par « Y »

Fichiers de configuration du serveur Apache

Le fichier de configuration principale d'apache est /etc/httpd/conf/httpd.conf. Ce fichier est organisé sous forme de directives définissant les paramètres généraux du serveur web. Une directive peut être comparée à une valeur suivie de sa valeur. Par exemple, la directive DocumentRoot définit l'emplacement de la racine du serveur, c-à-d où sont placés des fichiers du site web hébergé par le serveur (ici sa valeur est /var/www/html) :

DocumentRoot /var/www/html

Bien sur, on peut changer la valeur par défaut de chaque directive (ce que nous allons faire dans ce TP)

La configuration des différents modules est placée dans /etc/httpd/conf.d. Les modules sont présents dans /etc/httpd/modules

Il n'est pas possible de lister toutes les directives du fichier httpd.conf mais quelques unes sont importantes :

- **DocumentRoot** : définit l'emplacement de la racine du serveur où sont placés les fichiers du site web hébergé par le serveur. La valeur par défaut est /var/www/html.
- **ServerRoot** : répertoire contenant les fichiers du serveur (configuration et modules). C'est généralement /etc/httpd.
- **Listen** : ports sur lesquels le serveur Apache écoute. Par défaut 80 (443 en https). On peut en spécifier plusieurs avec plusieurs directives Listen. Si le serveur dispose de plusieurs adresses IP, on peut ajouter l'IP au port associé : Listen 192.168.1.2 :80. Si on change la valeur par défaut du port, il faut spécifier au navigateur le port du serveur utilisé.
- **User** : Le compte utilisé par Apache lors du lancement (UID). On n'utilise jamais root, mais un compte créé pour l'occasion, généralement nobody.
- **Group** : idem mais pour le groupe
- **ServerAdmin** : adresse de courrier électronique de l'administrateur
- **ServerName** : nom d'hôte (et port) du serveur. Il ne correspond pas forcément au nom de la machine. Par contre, il doit être valide, par exemple www.mondomaine.org
- **UseCanonicalName** : si elle vaut on, Apache va répondre en utilisant les informations de ServerName et Port, et pas les informations envoyées par le client. Par exemple, un http://192.168.1.3 se transforme en http://www.mondomaine.org.
- **UserDir** : nom du répertoire dans le répertoire courant où chaque utilisateur de la machine peut placer ses fichiers HTML personnels. Généralement public_html. On y accède avec http://www.mondomaine.org/~login/page.html avec login est nom d'un utilisateur valide.
- **ErrorLog** : fichier où sont placées les logs d'erreur du serveur. /var/log/httpd/error_log.
- **CustomLog** : fichier journal de Apache. /var/log/httpd/access_log.
- **Timeout** : durée pendant laquelle le serveur attend des émissions/réceptions au cours d'une communication. Elle est réglée sur 300 secondes.
- **DirectoryIndex** : précise les fichiers html ou cgi par défaut lors du chargement d'une URL. Par exemple si : DirectoryIndex index.php index.html alors au chargement, sans préciser le nom du fichier html, le serveur tentera de charger index.php et, s'il est absent, index.html. Dans le cas contraire, c'est l'option Indexes qui détermine si le contenu doit être visible sous forme de répertoire.

Q. 2 Procéder à une sauvegarde du fichier httpd.conf

Q. 3 Editer le fichier `httpd.conf` (par exemple avec `gedit` en tant que `root`) puis répondre aux questions suivantes à partir de ce fichier :

- a. Quel est le nom du répertoire racine du serveur, répertoire à partir duquel se trouve la configuration du serveur ?
- b. Quels sont les noms et localisations des principaux fichiers de configuration ?
- c. Quels sont les noms et la localisation des fichiers de rapport d'erreurs et de connexions ?
- d. Quel est le répertoire racine du site web hébergé par le serveur ?
- e. Quels sont le nom et la localisation du fichier où le système enregistre le numéro d'identification du processus qui exécute le programme `httpd` ?
- f. Avec quel utilisateur et groupe est lancé le serveur ?
- g. Quel est le port d'écoute par défaut du serveur ?
- h. Supposer que vous avez modifié la valeur par défaut du port du serveur et que lors d'un test avec un navigateur, le serveur n'est pas accessible. Donner aux moins deux raisons possibles et les moyens de remédier au problème.
- i. Quelle est la directive qui permet l'utilisation d'un répertoire personnel par les utilisateurs afin de déployer leurs sites web personnels ?
- j. Vous avez activé la directive du répertoire personnel dans Apache puis vous avez relancé le serveur. Lorsque vous essayez de vous connecter sur votre compte web, l'accès vous est refusé. Donner une explication et une solution pour remédier à ce problème.

Lancement/Arrêt/Test du serveur Apache

Lorsqu'il est installé, Apache est lancé comme un service de la station grâce au script de démarrage `/etc/init.d/httpd`. Le nom du service dépend de la distribution. Il est souvent intitulé `apache` ou `httpd`. Dans le cas de Fedora, il se nomme « `httpd` ». Ainsi, pour lancer Apache on peut soit utiliser la commande `service` ou invoquer directement le script de démarrage (il faut être administrateur).

- `/etc/init.d/httpd start` ou `service httpd start` : démarre ;
- `/etc/init.d/httpd stop` ou `service httpd stop` : stoppe ;
- `/etc/init.d/httpd restart` ou `service httpd restart` : redémarre ;
- `/etc/init.d/httpd reload` ou `service httpd reload` : demande à Apache de relire sa configuration sans redémarrer (lors d'un changement dans les fichiers de configurations)

Il existe plusieurs méthodes pour vérifier le bon lancement du serveur Apache :

- Utilisation de la commande `service httpd status` qui donne le statut d'apache (démarrer ou arrêter)
- Vérification de la table des processus : `ps aux | grep httpd`
- Les fichiers log d'Apache
- On tape dans un navigateur web <http://127.0.0.1> ou encore `http://localhost`

Q. 4 Vérifier que le compte utilisateur `nobody` existe (voir `/etc/passwd`). S'il n'existe pas, créer le avec la commande `useradd` ou `adduser`.

Q. 5 Vérifier que le groupe `nobody` existe (voir `/etc/group`). S'il n'existe pas, créer le.

- Q. 6** Modifier dan httpd.conf pour qu'Apache se lance avec l'utilisateur nobody et le groupe nobody.
- Q. 7** Procéder au lancement d'Apache et vérifier qu'il est bien lancé par plusieurs méthodes.
- Q. 8** Arrêter le serveur puis relancer-le. Notez le PID du serveur après chaque action. Rappelez à quel endroit il est possible de récupérer ce PID.
- Q. 9** Accéder au serveur web à partir d'un navigateur puis lancez la commande *tail /etc/httpd/logs/accesses_log* dans un terminal. Commentez le résultat.

Configuration de base du serveur Apache

- Q. 10** Quels sont les noms des fichiers servant d'index – pages par défaut – à un répertoire (quand l'URL est un répertoire le serveur retournera le fichier index qu'il trouvera dans le répertoire en guise de page par défaut). Indication : DirectoryIndex
- Q. 11** Rajoutez dans la liste des fichiers index le fichier « perso.html », testez.
- Q. 12** Est-ce que votre serveur supporte php par défaut? Pour cela créer une page index.php dans le répertoire du site du serveur et insérer les lignes suivantes dans :

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

Lancer votre navigateur avec le lien <http://127.0.0.1/index.php> ? Conclure

- Q. 13** Procéder à l'installation du module php . Le package se nomme php. N'oublier pas de relancer apache. Retester maintenant le lien <http://127.0.0.1/index.php>
- Q. 14** Modifier le serveur pour qu'il ne prenne en compte que les demandes effectuées sur le port 1150. Testez.
- Q. 15** Quel est le nom du répertoire par défaut permettant à un utilisateur du système de publier des informations sur le serveur web directement à partir de son compte ? Indication : Userdir. Pour réaliser les tests, suivez les instructions suivantes :
- Activer l'option qui permet à un utilisateur de publier son site web personnel.
 - Relancer le serveur
 - Créer un nouveau utilisateur test à appartenant au groupe test à l'aide de l'interface graphique de Fedora
 - Changer d'utilisateur en se connectant maintenant sous l'utilisateur test.
 - Créer un répertoire public_html dans le répertoire courant.
 - Mettre dedans une page index.html contenant Bonjour
 - Vérifier que vous avez assez de droit sur les répertoires /home/test, /home/test/public_html et aussi sur le fichier /home/test/public_html/index.html
 - Tester en lançant votre navigateur sur <http://localhost/~test>

Q. 16 Modifier le nom du serveur (donnez lui comme nouveau nom « igainfo »). Indication :
ServerName

Q. 17 Pour qu'un client s'y connecte avec le nouveau nom, il faut déclarer que le serveur correspondant à ce nom se trouve sur votre machine. Pour cela éditer le fichier `/etc/hosts` et modifier la ligne suivante ainsi :

```
127.0.0.1 localhost.localdomain    localhost igainfo
Tester en ouvrant le lien http://igainfo dans un navigateur
```

Vitural Host

Les hosts virtuelles permettent d'héberger sur le même serveur plusieurs sites. Le site servi au client dépend du nom utiliser par le client pour invoquer le serveur web. Q20 : Configurez votre serveur pour qu'il puisse héberger un autre site appelé « secondsite » en suivant les étapes suivantes :

```
<VirtualHost *:80>
#serveur principale qui répond a tous
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
ServerName localhost
ErrorLog logs/error_log
CustomLog logs/access_log common
</VirtualHost>
```

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /home/invite/public
ServerName secondsite
ErrorLog logs/invite-error_log
CustomLog logs/invite-access_log common
</VirtualHost>
```

Que doit-on ajouter au niveau du fichier `/etc/hosts`. Tester et décrire vos tests.

Authentification

Q. 18 Configurez le serveur de façon à ce qu'il puisse authentifie l'accès à un répertoire : Pour cela, on suppose qu'on souhaite protéger le répertoire « `/var/www/html/protege` » avec un mot de passe

- Créer d'abord le répertoire `"/var/www/html/protege"`
- Placer dedans un fichier `index.html` contenant bonjour
- Modifier le fichier de configuration d'apache en ajoutant les lignes suivantes :

```
<Directory "/var/www/html/protège">
```

```
AuthName Protected
```

```
AuthType basic
```

```
AuthUserFile /etc/httpd/conf/users
```

```
<Limit GET POST>
```

```
    require valid-user
```

```
</Limit>
```

```
</Directory>
```

- Créer un fichier d'authentification /etc/httpd/conf/users à l'aide de la commande
htpasswd -c /etc/httpd/conf/users test
test étant un utilisateur qu'il faut renseigner le mot de passe
Tester l'accès au site <http://localhost/protège>