

Applications Réparties TD 2

Installation et Configuration d'un Serveur HTTP Apache

Le but de ce TD est d'installer et d'étudier la configuration d'un serveur http. Nous utiliserons Apache 2.4 sous Linux. Ce TD se déroulera sur vos PCs portables, sous Linux. Vous utiliserez votre compte **root** (commande **sudo**). Pour chacune des sections du sujet et pour avoir plus de détails, vous pourrez vous reporter à la documentation officielle <http://httpd.apache.org/docs/2.4/fr/>.

1 Installation et configuration du serveur HTTP Apache 2.4

Apache peut-être composé de plusieurs familles de paquets.

Vous utiliserez un gestionnaire de paquets comme **synaptic**, **apt-get**, pour l'installation d'Apache 2.4.

En premier lieu, Apache 2.4 nécessite l'installation d'une bibliothèque, d'un serveur et d'outils (Figure 1).

Nom	Rôle
 libapr1	Apache's Portable Runtime Library, bibliothèque de fonctions standards portables.
 apache2	Ce paquet contient le serveur.
 apache2.2-common	Ce paquet contient les modules standards apache2, qui incluent le support SSL.
 apache2-utils	Outils pour serveurs web.

Figure 1 : les paquets d'Apachez

Ensuite, vous pouvez installer un Multi-Processing Module (MPM), ou module multi traitements. C'est le moteur du serveur qui définit la manière dont il intercepte les requêtes et les traite simultanément. Il en existe plusieurs à vous de choisir en fonction de vos besoins (Figure 2).

Nom	Commentaire
 apache2-mpm-prefork	Modèle traditionnel pour Apache2 version sans thread, intercepte les requêtes à la manière Apache 1.3, utile pour éviter la mise en thread pour la compatibilité avec les bibliothèques non-thread-safe. C'est le meilleur mpm pour isoler les requêtes.
 apache2-mpm-worker	Modèle à processus haute vitesse pour Apache2 version avec thread, il est considérablement plus rapide que la version traditionnelle et c'est le MPM recommandé .

Figure 2 : les paquets MPM

Dans les versions récentes d'Ubuntu les paquets sont regroupés, vous n'aurez donc pas à charger ces paquets les uns après les autres, à la main. En effet l'installation de base jusqu'à Ubuntu 6.06 nécessitait d'installer les paquets : *apache2 apache2-utils apache2-mpm-prefork libapache2-mod-chroot libapache2-mod-auth-pam libapache2-mod-auth-sys-group*.

De même pour les versions antérieures à Ubuntu 11.04, il fallait installer les paquets *libapr1 apache2 apache2.2-common apache2-utils*.

Depuis Ubuntu 11.04, il suffit d'installer le paquet *apache2* qui contient alors toutes les librairies et outils nécessaires.

Exercice 1 : Installez Apache 2.4 sur votre machine. Quels sont les paquets qui sont alors installés ?

2 Mise en route du serveur HTTP

Pour lancer le serveur Apache 2.4, vous pouvez utiliser la commande : *sudo service apache2 start/stop/restart* (Sachez qu'il existe des interfaces graphiques pour utiliser Apache telles que : *localhost-indicator, rapache et webmin*).

Applications Réparties TD 2

Installation et Configuration d'un Serveur HTTP Apache

Exercice 2 : Lancez Apache et tester son fonctionnement sur localhost. Qu'affiche-t-il ?

Exercice 3 : Utiliser la commande `apachectl -t -D DUMP_MODULES` pour lister les modules apachez chargés ? Nous aurons besoin plus tard de module `userdir`. Est-il chargé ?

Remarque : Au démarrage d'Apache, il est possible que celui-ci ne soit pas capable de déterminer un nom de domaine. Ce message sera affiché dans le terminal : `"Could not reliably determine the server's fully qualified domain name, using 127.0.1.1 for ServerName"`. Pour corriger l'erreur vous pouvez éditer le fichier `/etc/apache2/httpd.conf`. Par défaut ce fichier est vide, il faut donc ajouter la ligne suivante : `ServerName localhost`. Enregistrez alors le fichier, quittez et redémarrez Apache. Pour cela utilisez la commande : `sudo service apache2 restart` (en fait cela remplace l'ancienne commande `sudo /etc/init.d/apache2 restart`).

Remarque : Notez qu'il est aussi possible de redémarrer apachez sans "tuer" les requêtes en cours avec `apachectl graceful`, ce qui a pour avantage de ne pas perturber le fonctionnement de votre serveur durant son administration. Les vérifications du bon fonctionnement de votre serveur peuvent se faire au travers ses fichiers de « log » : `/var/log/syslog`, `/var/log/apache2/access.log`, `/var/log/apache2/error.log`.

3 Configuration du serveur http

Tous les **fichiers de configuration** de Apachez sont dans le dossier `/etc/apache2`.

La plupart de ces fichiers sont spécifiques à Ubuntu et nous permettent de séparer la configuration d'Apache 2.4 en plusieurs parties :

- `httpd.conf` est le fichier utilisé par apache1, il est conservé vide dans Apache2 pour assurer la rétrocompatibilité. Il ne nous servira pas ;
- `envvars` est utilisé pour définir des variables d'environnement propres à Apache ;
- `ports.conf` contient la directive `listen` qui spécifie les adresses et les ports d'écoutes ;
- `apache2.conf` est le fichier principal de configuration c'est à partir de lui que tous les autres fichiers sont chargés ;
- `conf.d` est un répertoire qui contient plusieurs petits fichiers qui seront analysés par apache. Le seul fichier pour le moment est `charset`, qui spécifie l'encodage à utiliser par défaut ;
- `mods-available` contient la liste des modules d'apache installés ;
- `mods-enabled` celle des modules utilisés ;
- `sites-available` contient la liste des vhosts installés ;
- `sites-enabled` celle des vhosts utilisés.

Exercice 4 : Dans le répertoire `/etc/apache2/` éditez les fichiers : `apache2.conf` `conf.d` `envvars` `httpd.conf` `mods-available` `mods-enabled` `ports.conf` `sites-available` `sites-enabled` et parcourez les.

3.1 La page d'accueil et les pages par défaut

Exercice 5 : Renommez `index.bak` la page d'accueil du serveur. Pouvez-vous l'obtenir comme précédemment ? Pourquoi ?

Exercice 6 : Qu'obtenez-vous en passant l'URL : `http://serveur/index.bak` ? Expliquez pourquoi.

Exercice 7 : Créez rapidement une petite page HTML portant le nom `accueil.html`. L'objectif est qu'elle devienne la page d'accueil du serveur. Rechercher avec `grep` la ligne de `DirectoryIndex` et la modifier convenablement. Vérifier.

Applications Réparties TD 2

Installation et Configuration d'un Serveur HTTP Apache

3.2 Suivi du journal des accès

Exercice 8 : Examiner les journaux :

- *access.log* pour les requêtes traitées (qu'elles soient réussies ou non)
- *error.log* pour noter les erreurs de fonctionnement du serveur

Exercice 9 : 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" ...)

3.3 Modifications de paramètres d'exécution

Supposons que le serveur d'établissement soit hébergé par une machine aux ressources limitées. On peut modifier le nombre de processus http.

Exercice 10 : Agissez sur les clauses du paragraphe *prefork MPM*. Quel est le nombre actuel de serveurs WEB en exécution lors du démarrage d'Apache ? le nombre maximum en exécution simultanément ?

Exercice 11 : Adresser des requêtes dans votre navigateur, que remarquez-vous ?

3.4 Les types mimes

Exercice 12 : Où se trouve le fichier qui donne les correspondances entre types de fichiers et extension ?

Exercice 13 : Où peut-on déclarer, si nécessaire, de nouvelles associations ?

3.5 Fixer le nom public du serveur

Exercice 14 : Renseigner la directive **ServerName** dans *ports.conf* avec un nom comme *test*. Tester à partir d'un client, après avoir renseigné le fichier */etc/hosts* (<votre IP publique> pour *test*).

3.6 Modifier l'accessibilité à un répertoire

Il s'agit de régler, pour chaque répertoire du site principal (et aussi nous le verrons pour les sites virtuels), le droit d'accéder aux pages que ce répertoire contient, notamment en fonction de la machine cliente. Le fichier dans lequel ce paramétrage s'effectue est : */etc/apache2/sites-available/default*

On peut spécifier explicitement des adresses IP, des réseaux, des noms de machine ou de réseau ou all, dans la directive <Directory /...>, avec les clauses *order*, *allow* *from* *et* *deny* *from*
Voir les détails dans <http://httpd.apache.org/docs/2.4/fr/>.

Exercice 15 : Modifier l'accès au site racine (*/var/www/*) pour autoriser l'accès de toutes les machines du sous-réseau 192.168.X.X sauf celles qui vous sont voisines. Testez et vérifiez.

Exercice 16 : Interdire l'accès de tous les hôtes sauf localhost. Testez et vérifiez.

3.7 Site web personnel

Apache permet l'utilisation des répertoires personnels des utilisateurs comme racine d'un site.

Ceci grâce au module *userdir*. Le module et son fichier de configuration sont dans le dossier */etc/apache2/mods-available*.

Exercice 17 : Activer *userdir* par la commande : *sudo azenmod userdir* (N'oubliez de recharger la configuration d'apache). Créons ensuite un nouvel utilisateur : *userN* (*mot de passe 123*). Comme tous les utilisateurs, vous pouvez constater qu'il possède un dossier personnel */home/userN* et qu'il est le seul à y posséder des droits en écriture (*ls -l*).

Exercice 18 : Ouvrez le contenu du fichier *userdir.conf*, que constatez-vous ?

Applications Réparties TD 2

Installation et Configuration d'un Serveur HTTP Apache

Exercice 19 : Créez un dossier portant ce nom dans le dossier personnel de notre utilisateur. Dans ce dossier créez une page *index.html*. Modifiez les droits de cette page pour que tous les utilisateurs y accèdent en lecture. Testez votre page en l'appelant avec l'adresse : `<votre IP publique>/~userN`.

3.8 Protection de notre site

Protégeons maintenant l'accès à ce site. Pour cela créez dans le dossier *public_html* de votre utilisateur un fichier : *.htaccess*, avec le contenu suivant :

```
AuthName "Ce Site est sous protection"  
AuthType Basic  
AuthUserFile "/home/userN/passwords"  
Require valid-user
```

La directive *AuthName* est un Message sans importance

La directive *AuthType* indique le niveau d'authentification

La directive *AuthUserFile* indique l'emplacement du fichier des mots de passe

La directive *Require valid-user* impose la vérification du mot de passe

Exercice 20 : Créez le fichier des mots de passe *passwords* (positionnez-vous dans le dossier de votre utilisateur et saisissez la commande : `htpasswd -c passwords userN`). Saisissez un mot de passe facile à retenir (par exemple 123). Fermez votre navigateur, ouvrez le nouveau et rechargez la page: `192.168.10.1/~userN`. Que se passe-t'il ?

3.9 Installer des serveurs virtuels

Il s'agit d'héberger et de gérer plusieurs sites, dits "virtuels", par un seul et même serveur.

Ces sites seront appelés par les clients sous différentes adresses IP ou différents noms.

Trois solutions peuvent être envisagées :

- sites virtuels basés sur l'IP. Dans ce cas, le serveur écoute plusieurs adresses IP, à chacune est associé un nom de site
- sites virtuels attachés à des ports différents sur une même adresse IP
- sites virtuels basés sur le nom. C'est la solution la plus simple que nous allons mettre en œuvre : le serveur écoute une seule adresse IP, à laquelle sont associés plusieurs noms de sites.

Nous allons configurer un site *testvirtuel* à l'accès du site "privé" (précédemment protégé par mot de passe). On devrait y accéder par les URL `http://testvirtuel`, pour l'accès à la racine du site.

Exercice 21 : A la fin du fichier `/etc/apache2/apache2.conf`, insérer `NameVirtualHost [votre adresse IP]:80`, avant le ligne d'inclusion du site virtuel. Commentez la 1ère ligne du fichier `/etc/apache2/sites-available/`. A défaut de DNS local, renseignez le fichier `/etc/hosts` sur les stations autorisées à accéder aux sites :

```
127.0.0.1 localhost  
[votre adresse IP] testvirtuel
```

Les paramétrages de chaque site virtuel est placé de préférence dans un fichier de configuration spécifique (autant de sites virtuels), lequel est obligatoirement enregistré dans `/etc/apache2/sites-available/`.

Exercice 22 : Ecrivez le fichier de configuration `testvirtuel.conf`

```
# service virtuel pointant vers testvirtuel  
<VirtualHost *:80>
```

Applications Réparties TD 2

Installation et Configuration d'un Serveur HTTP Apache

```
DocumentRoot /var/testvirtuel
ServerName testvirtuel
</VirtualHost>
```

Pour que le fichier précédent, décrivant le site virtuel, soit intégré dans `apache2.conf` il faut placer les liens vers le répertoire `/etc/apache2/sites-enabled/`. Cette mise en service se fait facilement grâce à la commande `azensite`, puis on recharge la configuration.

Dans `/etc/apache2/sites-available`, lancez `azensite testvirtuel.conf` puis relancer `apache2`. Testez et vérifiez <http://testvirtuel>.

3.10 Activer les scripts CGI

Pour ce faire, rien de plus simple, ajoutez les lignes suivantes à votre "apache2.conf" :

```
ScriptAlias /cgi-bin/ /var/www/votre_site/cgi-bin/
<Directory /var/www/votre_site/cgi-bin/>
Options ExecCGI
AddHandler cgi-script cgi pl py
</Directory>
```

Exercice 23 : Déclarez un répertoire de scripts `cgi-bin`. En créer un et testez-le.

3.11 Et bien d'autres configurations ...

Exercice 24 : Comment désactiver la signature numérique qui permet d'identifier le serveur ?

Exercice 25 : Comment Empêcher le listing des répertoires ?

Exercice 26 : Comment limiter l'accès au réseau local (LAN) ?

Exercice 27 : Comment protéger votre serveur contre les attaques DOS (denial-of-service) en limitant le poids des fichiers en upload ?

....