
Fiche de TP n.1 – Apache

1 Lancement et arrêt du processus démon httpd

Le processus démon correspondant au serveur HTTP d'Apache se nomme `httpd`. Dans la suite nous noterons par `<httpdDir>` le chemin du répertoire dans lequel se trouve l'exécutable `httpd` (`/usr/sbin` par exemple). La commande `httpd` admet plusieurs options, en voici quelques unes :

- `<httpdDir>/httpd -v` : pour obtenir la version d'Apache,
- `<httpdDir>/httpd -h` : pour obtenir les différentes options disponibles,
- `<httpdDir>/httpd -l` : pour obtenir la liste des différents modules compilés dans Apache,
- `<httpdDir>/httpd -k start` : pour lancer le démon `httpd`,
- `<httpdDir>/httpd -k stop` : pour stopper le démon `httpd`,
- `<httpdDir>/httpd -k restart` : pour relancer le démon `httpd`,
- `<httpdDir>/httpd -t` : pour tester la syntaxe des fichiers de configuration de `httpd`.

Le démon `httpd` peut également être manipulé, comme d'autres démons, par la commande `service` :

```
<serviceDir>/service httpd {status|start|restart|stop|configtest}
```

On indique le paramètre `status` pour connaître l'état de `httpd`, `start` pour démarrer `httpd`, `restart` pour redémarrer `httpd`, `stop` pour stopper `httpd`, `configtest` pour tester la syntaxe des fichiers de configurations de `httpd`.

2 L'arborescence du répertoire racine de httpd

Nous noterons par `<serverRoot>` le chemin du répertoire contenant entre autres les répertoires, fichiers de configuration et les journaux d'activités de `httpd`. Par exemple, `<serverRoot>` pourrait être `/etc/httpd` sur votre machine. Dans ce répertoire se trouvent en particulier les sous-répertoires suivants :

- `<serverRoot>/conf` : un sous-répertoire contenant les fichiers généraux de configuration de `httpd`. Il s'y trouve en particulier le fichier `httpd.conf` qui contient les directives générales de configuration.
- `<serverRoot>/logs` : un sous-répertoire contenant les différents journaux d'activités parmi lesquels le fichier `access.log` qui contient les requêtes traitées par le serveur et le fichier `error_log` qui contient les erreurs serveur survenues.
- `<serverRoot>/modules` : un sous-répertoire qui contient différents modules (le module PHP par exemple) susceptibles d'être chargés par le serveur.
- `<serverRoot>/conf.d` : un sous-répertoire contenant d'autres fichiers de configuration, en particulier des fichiers de configuration des modules pouvant être chargés par le serveur. Les directives de ces fichiers sont par défaut incluses dans `httpd.conf` (commande `Include conf.d/*.conf` se trouvant dans `httpd.conf`).
- `<serverRoot>/run` : un sous-répertoire dans lequel se trouvent des fichiers contenant des pid de démons s'exécutant. Par exemple, le fichier `httpd.pid` contiendra le pid de `httpd`.

Remarque : certains de ces chemins de sous-répertoires sont en fait des liens symboliques.

3 Manipulations préliminaires

M0. Connectez vous comme `root`. Localisez et notez les chemins des différents sous-répertoires et fichiers présentés précédemment. Sauvegardez le fichier `httpd.conf` en effectuant une copie. La sauvegarde se nommera `httpd.conf.ancien`.

M1. Testez les commandes de démarrage, de redémarrage et d'arrêt de `httpd`. Lancez le serveur apache. Quels sont les exécuteurs des processus du serveur `httpd`? Inspectez la directive `User` donnée dans `httpd.conf`.

M2. Sur la même machine avec votre nom d'utilisateur (pas le `root`), ouvrez avec un navigateur l'url `http://localhost/` ou `http://127.0.0.1/` et vérifiez que la page de test s'affiche bien.

M3. Insérez une erreur de syntaxe dans le fichier de configuration `httpd.conf` en modifiant sa première ligne. Testez la syntaxe des fichiers de configuration de `httpd` avec la commande adéquate. Remettez en état le fichier `httpd.conf` et testez de nouveau la syntaxe des fichiers de configurations de `httpd`.

M4. Repérez le numéro du port écouté par le serveur en notant le paramètre de la commande `Listen` se trouvant dans `httpd.conf`.

M5. Récupérez et notez les valeurs attribuées à `ServerRoot` et `DocumentRoot` dans le fichier `httpd.conf`. `ServerRoot` correspond au répertoire `<serverRoot>` dont on a parlé plus haut. `DocumentRoot` correspond au répertoire racine des documents publiés par le serveur. Nous le noterons `<documentRoot>`. Ainsi, l'url `http://localhost/` correspond au répertoire `<documentRoot>`.

4 Configuration du répertoire racine des document publiés et des fichiers par défaut

La manipulation suivante va consister à mettre comme répertoire `<documentRoot>` un sous-répertoire web placé dans votre répertoire principal. Examinez la manière dont les droits doivent être mis.

M6. Changez le répertoire racine des documents publiés de la manière suivante. Créez le répertoire `<homeUserDir>/web`. Donnez les droits 711 (passage pour `ugo`) à `<homeUserDir>` et les droits 755 (passage et lecture pour `ugo`) à `<homeUserDir>/web`.

Remarque importante : les futurs documents à l'intérieur de ce dernier répertoire devront être lisibles de tout le monde pour pouvoir être publiés par le serveur.

Commentez la ligne (ajout d'un `#` en début de ligne) de `httpd.conf` affectant une valeur à `DocumentRoot`. Rajoutez une nouvelle ligne affectant la valeur `<homeUserDir>/web` à `DocumentRoot` puis relancez `httpd`. Maintenant `http://localhost/` correspond à `<home_user_dir>/web`. Ouvrez l'url `http://localhost/`. Que se passe-t-il ?

Par défaut, lorsqu'une url désignant un répertoire (et non un fichier spécifique) est demandée, le fichier `index.html` est tout d'abord recherché dans ce répertoire puis, dans le cas où il n'est pas trouvé, `index.html.var` est recherché dans ce répertoire (pour la négociation de contenus ...). La directive `DirectoryIndex` de `httpd.conf` permet de spécifier les fichiers recherchés. Si l'un d'eux est trouvé il est renvoyé. Dans le cas contraire :

- Si l'option `Indexes` est mise pour le répertoire concerné, à la place du message d'erreur de type 403 s'affiche la liste des éléments du répertoire concerné.
- si l'option `Indexes` n'est pas mise pour le répertoire concerné (c'est le cas par défaut), un message d'erreur de type 403 est retourné. Généralement, pour le répertoire `<documentRoot>`, le message correspondant à cette erreur est une page de test (cela peut être par exemple `/var/www/error/noindex.html`).

M7. Commentez la ligne `DirectoryIndex` du fichier `httpd.conf` puis, rajoutez une ligne spécifiant le fichier `principal.html` comme fichier à rechercher par défaut lorsqu'une url désignant un répertoire est demandée. Relancez `httpd`.

M8. Créez un fichier `principal.html` dans le répertoire `<home_user_dir>/web`. Cette page permettra entre autres choses d'afficher votre nom de machine, votre prénom, votre nom et votre nom d'utilisateur unix (remarque : vous garderez la même machine pour tous les travaux pratiques). Ouvrez maintenant l'url `http://localhost/`.

M9. Créez les répertoires `<homeUserDir>/web/infos1` et `<homeUserDir>/web/infos2`. Ouvrez l'url `http://localhost/infos1`. Relevez l'erreur retournée. Ouvrez l'url `http://localhost/infos3`, relevez l'erreur retournée.

La directive `Directory` est utilisée pour configurer un répertoire ou un groupe de répertoires. La syntaxe de cette directive est la suivante :

```
<Directory dir>
  directive1 pour dir
  directive2 pour dir
  ...
</Directory>
```

`dir` est une expression régulière désignant un ou plusieurs répertoires : `"/home/condotta/web/infos1"` ou `/` par exemple. `directive1`, `directive2`, ... sont des directives qui s'appliqueront à `dir`, ainsi qu'à ses sous-répertoires. La directive `Options` permet de spécifier des options (pour un répertoire particulier notamment). Une ligne de commande correspondant à cette directive est de la forme :

```
Options option1 option2 ...
```

Les options spécifiées `option1`, `option2`, ... sont attribuées, celles qui ne sont pas spécifiées ne sont pas attribuées. Par exemple, avec la directive

```
Options FollowSymLinks Indexes
```

les options `FollowSymLinks` et `Indexes` sont attribuées, les autres options ne le sont pas. Une autre syntaxe possible pour la directive `Options` est de rajouter devant chaque nom d'option un modificateur : + (rajouter l'option) ou - (supprimer l'option). Les options spécifiées sont modifiées, celles qui ne sont pas spécifiées ne le sont pas. Par exemple, la directive

Option `+Indexes`

rajoutera l'option `Indexes` mais ne supprimera ou ne rajoutera pas d'autres options.

L'option `all` est mise par défaut pour un répertoire (elle regroupe toutes les options, sauf `Multiviews`).

Remarque 1 : plusieurs directives de type `Directory` peuvent concerner le même répertoire. Les directives sont exécutées en partant du chemin le moins spécifique vers le plus spécifique. Par exemple, si l'option `Indexes` est mise pour `"/home/condotta/web/"` et est supprimée pour `"/home/condotta/web/infos1"`, `"/home/condotta/web/infos1"` n'aura pas l'option `Indexes`.

Remarque 2 : la directive `Location` est similaire à `Directory` mais *travaille* sur les noms des chemins spécifiés dans les url et non sur les noms des répertoires du système de fichiers.

M10. Examinez le fichier `httpd.conf` et notez les options associées au répertoire racine `/`.

M11. Rajoutez au fichier `httpd.conf` une directive permettant de rajouter l'option `Indexes` au répertoire `<homeUserDir>/web/infos2`. Relancez le serveur `httpd`. Créez les fichiers `fichier1.txt`, `fichier2.txt` et `fichier3.txt` dans `<homeUserDir>/web/infos2`. Ouvrez l'url `http://localhost/infos2/`.

5 Utilitaires pour effectuer des requêtes HTTP

Jusqu'à maintenant vous avez effectué des requêtes `http` à l'aide de votre navigateur Web préféré (le navigateur mozilla par exemple). Il est également possible d'utiliser l'utilitaire `telnet` pour envoyer des requêtes `http`. Pour cela nous devons tout d'abord nous connecter au serveur sur le port adéquat : `telnet localhost 80` par exemple, puis taper le texte de la requête `http` et deux retours à la ligne. Pour les requêtes complexes il est préférable d'utiliser un éditeur de texte pour composer la requête et de faire un copier/coller (paramétrez l'éditeur pour ne pas mettre de retour à la ligne automatique). Il existe également l'utilitaire `lwp-request` et les commandes `GET`, `POST` et `HEAD` associées.

M12. Utilisez `telnet` pour envoyer une requête `http` permettant de récupérer la page `principal.html`. Notez la réponse renvoyée par le serveur.

M13. Utilisez la commande `lwp-request` pour récupérer le document `principal.html` (utilisez la commande `man` pour vous renseigner sur `lwp-request`).

M14. Utilisez la commande `GET` pour récupérer le document `principal.html` (utilisez la commande `man` pour vous renseigner sur `GET`).

6 Contenus des messages d'erreur

Des messages d'erreur sont renvoyés par défaut pour chacune des erreurs possibles. Il est possible de définir ses propres messages d'erreur à l'aide de la directive `ErrorDocument`. Cette directive prend deux paramètres. Le premier correspond au numéro de l'erreur, le deuxième spécifie le contenu du message de l'erreur à retourner. Ce dernier paramètre peut être un simple texte (une chaîne de caractères), un document html particulier, une url correspondant à un document externe au site ou encore un programme cgi :

```
ErrorDocument 403 ''Document non accessible !''
ErrorDocument 403 /erreur/erreur403.html
ErrorDocument 403 /cgi-bin/script403
ErrorDocument 403 http://serveur/doc.html
```

La directive `ErrorDocument` peut être placée dans une section `Directory` ou `Location`.

M15. Modifiez `httpd.conf` pour faire en sorte que s'affiche un texte particulier pour les erreurs 403 et 404 qui surviendrait dans la recherche d'un document dans `/web/infos1`.

M16. Créez le répertoire `<userHomeDir>/web/erreurs`. Créez dans ce répertoire les fichiers `erreur403.html` et `erreur404.html` contenant respectivement une page pour l'erreur 403 et une page pour l'erreur 404. Associez ces pages aux erreurs 403 et 404 pouvant survenir dans le répertoire `<userHomeDir>/web/infos2`. Rajoutez un fichier `fichier4.txt` dans `<userHomeDir>` sans les droits adéquats pour être publié. Testez l'affichage de ces messages d'erreur.

7 Les journaux d'activités

Le répertoire `<serverRoot>/logs` contient différents journaux d'activités mis à jour par le serveur Apache. Parmi les fichiers d'historique se trouvent par défaut :

- `error_log` dans lequel le serveur `httpd` enregistre des informations concernant tous les problèmes rencontrés : les problèmes systèmes (problèmes de démarrage, d'arrêt, ...) et les problèmes concernant les documents publiés (document non trouvé, non accessible, ...).

La directive `ErrorLog` permet d'indiquer le fichier qui va contenir cet historique

(`<serverRoot>/logs/error_log` généralement).

La directive `LogLevel` permet de fixer le type des problèmes à enregistrer en indiquant un des niveaux suivants (du type le plus spécifique au type le plus large) : `emerg`, `alert`, `crit`, `error`, `warn`, `notice`, `info`, `debug`. En indiquant `emerg`, seuls les problèmes de la plus haute importance (ceux rendant le système inutilisable) seront reportés. À l'autre bout de la chaîne, le niveau `debug` permet de reporter des problèmes qui ne sont pas vraiment des problèmes mais des messages utiles à un débogage.

- `access_log` contient des informations concernant toutes les requêtes traitées par le serveur.

La directive `CustomLog` permet d'indiquer le fichier utilisé (`<serverRoot>/logs/access_log` par défaut) et le format des informations enregistrées. La syntaxe de cette directive est :

`CustomLog désignationDuFichierLog formatUtilisé`

Un format est défini par la directive `LogFormat` qui prend en paramètre une chaîne de caractères entourée de '' spécifiant ce que l'on veut afficher et le nom du format. Dans la chaîne spécifiant le format peut se trouver `%h` pour afficher l'adresse IP du client, `%u` pour afficher le `userId` du client, `%r` pour afficher la requête entière du client, `%m` pour afficher la méthode de la requête, `%U` pour afficher le chemin du document, `%q` pour afficher la chaîne paramètre de la requête, `%H` pour afficher le protocole de la requête, `%t` pour afficher le moment auquel la requête a été complètement traitée, `%>s` pour afficher le status retourné par le serveur, `%b` pour afficher la taille du contenu retourné au client, `%unNom` pour afficher la valeur du champ d'en-tête portant le nom `unNom`,...

Remarque : dans le cas où le serveur ne peut fournir l'information demandée, un - est affiché. Par exemple, nous pouvons définir le format `monFormat` qui permettrait d'afficher uniquement l'adresse IP du client par :

```
LogFormat ''Adresse IP du client -> \%h'' monFormat
```

M17. Localisez les directives `ErrorLog` et `LogLevel` dans le fichier `httpd.conf`, assurez vous que `error_log` est le fichier utilisé et notez également le niveau des messages pris en compte. Visualisez le contenu de `error_log` (avec un `cat` par exemple), effectuez des requêtes en provoquant des erreurs de type 403 par exemple et visualisez de nouveau le contenu de `error_log`. Effectuez de nouveau la manipulation en modifiant le niveau des erreurs reportées. Remettez le niveau initial.

M18. Localisez les directives `CustomLog` et `LogFormat` dans le fichier `httpd.conf`, assurez vous que `access_log` est le fichier utilisé. Effectuez des requêtes à votre serveur (vous pouvez utiliser `HEAD` et `GET`) et visualisez les mises à jour effectuées dans le contenu de `access_log`.

M19. Commentez la ligne `CustomLog`. Définissez un nouveau format appelé `monFormat` permettant notamment l'affichage de l'adresse IP du client, la méthode de la requête, le chemin du document demandé, les paramètres de la requête, la valeur de l'entête `From`. Utilisez la directive `CustomLog` pour que ce format soit utilisé.

8 Les programmes CGI

La directive `ScriptAlias` permet d'indiquer au serveur où se trouvent les scripts CGI. La syntaxe est `ScriptAlias cheminCGI repertoireCGI`

où `cheminCGI` indique le chemin utilisé dans l'url pour atteindre les scripts CGI et `repertoireCGI` indique le répertoire où se trouvent les scripts CGI. Ainsi, la requête `http://serveur/cheminCGI/script` fera s'exécuter le script `repertoireCGI/script`.

Remarque : pour construire une réponse, un script CGI écrit dans son flux de sortie les lignes d'en-têtes. Il faut au moins

- mettre une en-tête `content-type`, `"content-type : text/html"` par exemple,
- une ligne vide,
- puis le contenu de la réponse.

M20. Repérez la directive `ScriptAlias` dans le fichier `httpd.conf`. Commentez là. Rajoutez une ligne indiquant que le répertoire contenant les scripts est `<homeUserDir>/web/programmes` et le chemin d'accès `/cgi-bin/`. Créez le répertoire `<homeUserDir>/web/programmes`. Créez dans ce répertoire un script shell nommé `bonjour` permettant d'afficher chez le client *Bonjour!* en gras. Testez ce programme.

M21. Créez un programme similaire au précédent mais réalisé en langage C et qui portera le nom `bonjourC`. Il sera également placé dans le sous répertoire `programmes`.

M22. Réalisez un programme en shell appelé `afficheEnv` permettant d'afficher les variables d'environnement lors de l'appel du script (la commande `env` retourne la liste des variables d'environnement et leur valeur). Ce programme est toujours placé dans le sous-répertoire `programmes`.

9 Les server-side scripts

Il est possible de mettre du code dans les différents documents publiés qui sera interprété par le serveur. Ce code doit être filtré par des interpréteurs avant que le serveur ne le retourne au client Il peut-être également filtré lorsqu'il est envoyé par le client (requête POST ou PUT) et reçu par le serveur. La directive `Files` permet de placer ces filtres-interpréteurs. Par exemple, la directive suivante :

```
<Files *.php>
SetOutputFilter PHP
SetInputFilter PHP
</Files>
```

permet d'indiquer que les fichiers portant l'extension `.php` doivent être filtrés par l'interpréteur PHP.

L'option `Includes` permet l'exécution du code côté serveur.

M23. Examinez le contenu du fichier de configuration `<serverRoot>/conf.d/php.conf`.

M24. Créez le sous-répertoire `<homeUserDir>/web/php`. Créez dans ce répertoire un fichier appelé `bonjour.php` qui permet d'afficher ***bonjour !*** en gras à l'aide d'au moins une instruction php. Testez ce programme en ouvrant l'url `http ://localhost/php/bonjour.php`.