

Lors de ce TP vous allez, entre autre, manipuler le serveur web [Apache](#) dont le but est de rendre un service de réponse à des requêtes adressées en [HTTP](#) sur la machine sur laquelle il est exécuté.

### **Exercice 1 : Préliminaires**

Dans la suite du TP vous allez travailler avec une machine préparée par nos soins et disponibles dans le répertoire idoine : `/usr/local/virtual_machine/info-n3p1-X/m3102-tp3` (avec X correspondant à votre groupe exprimé en minuscule).

Cette machine est standard :

- elle contient une installation de Debian en version *stable*, aka *jessie*;
- il n'y a qu'un seul utilisateur sur le système (**root** dont le mot de passe est **root**);
- le seul service intéressant activé sur la machine est **ssh**.

**Q 1.** Pour ce TP vous aurez besoin de 2 machines virtuelles. Avant de commencer il vous faut donc la *cloner* en deux exemplaires en copiant ce répertoire sous les noms **serveur** et **client** dans `/usr/local/virtual_machine/info-n3p1-X/<votre login>` (avec X correspondant à votre groupe exprimé en minuscule).

**Q 2.** Une fois les deux machines créées, démarrez les et installez-vous, sur chacune d'elle, quelques outils pratiques (comme **less**, **vim** ou **tree** par exemple).

**Q 3.** Modifiez ensuite le fichier `/etc/hostname` de façon à nommer l'une d'elle **serveur** et l'autre **client**.

**Q 4.** Fixez statiquement les adresses IP de vos machines de façon à ce que le client ait une adresse de la forme **x.y.z.10** et le serveur par **x.y.z.200**.

**Q 5.** Redémarrez vos machines et vérifiez votre configuration.

**Q 6.** Après avoir lu la page **hosts** (5), configurez vos machines de façon à ce que, depuis chacune d'elle, vous soyez capable de faire référence (*i.e.* accéder via **ping** par exemple) :

- au serveur par les noms
  - **serveur**,
  - **serveur.tp**,
  - **serveur.iut.fr**,
  - **www.iut.fr**,
  - **www.pere.noel**;
- au client par les noms
  - **client**,
  - **client.tp**,
  - **client.iut.fr**.

Pendant tout le TP vous utiliserez les outils installés sur votre machine **client** pour vérifier le fonctionnement de votre service HTTP installé, lui, sur votre machine **serveur**.

**Q 7.** Installez les navigateurs web **w3m**, **firefox-esr** et **chromium** sur votre client.

**Q 8.** Sur le serveur créez autant d'utilisateurs que d'étudiants dans votre groupe.

### **Exercice 2 : Installation d'Apache**

**Q 1.** Installez les paquets **apache2** et **apache2-doc** sur votre serveur.

**Q 2.** Sur le serveur, créez les répertoires `/var/www/html/www.pere.noel`, `/var/www/html/iut` et `/var/www/html/default` puis créez dans chacun de ces répertoires un fichier, au format HTML, nommé **index.html** et au contenu de votre choix mais différent. Par exemple :

- votre liste de demande de cadeaux au père Noël dans `/var/www/html/www.pere.noel/index.html`,
- une définition du mot IUT dans `/var/www/html/iut/index.html`,
- un salut au monde dans `/var/www/html/default/index.html`.

**Q 3.** Démarrez un navigateur web sur votre machine hôte et visitez le serveur web installé sur votre serveur.

**Q 4.** Démarrez un navigateur web graphique de votre choix sur le client en le faisant afficher sur votre machine hôte. C'est ce navigateur qui vous servira pendant le reste du TP.

**Q 5.** Visitez les pages suivantes pour comprendre le fonctionnement d'Apache :

- `http://serveur`
- `http://serveur/manual`
- `http://serveur.tp`
- `http://serveur.iut.fr`
- `http://www.iut.fr`
- `http://www.pere.noel`
- `http://www.pere.noel/www.pere.noel`
- `http://www.pere.noel/iut`

**Q 6.** Sur le serveur, renommez le fichier `/var/www/html/index.html` en `/var/www/html/index.html.old` puis visitez les mêmes adresses qu'à la question précédente.

### **Exercice 3 : Configuration de base d'Apache**

L'intégralité de la configuration du serveur Apache est stockée dans le répertoire `/etc/apache2`.

Sa documentation est accessible localement dans le répertoire `/usr/share/doc/apache2-doc/manual`. Sur le serveur vous pouvez donc par exemple la consulter avec la commande

```
w3m /usr/share/doc/apache2-doc/manual/index.html
```

Elle est également servie par défaut par le serveur Apache installé, via, par exemple, l'URL `http://serveur/manual`. Vous pouvez donc la consulter avec le navigateur démarré sur votre client.

À tout moment dans ce TP, vous devez vous aider de **cette** documentation d'Apache (Google n'est pas votre ami dans ce TP).

**Q 1.** Lisez les sections « *Adresse IP et port d'écoute* », « *Fichiers de configuration* », « *Sections de configuration* » et « *Mise en correspondance des URLs avec le système de fichiers* » du *Guide de l'utilisateur*.

**Q 2.** Lisez le fichier `/usr/share/doc/apache2/README.Debian.gz`.

Sous Debian, pour manipuler un service (démarrer, stopper, rechercher, consulter l'état, etc.), par exemple `apache2` il faut utiliser la commande `service(1)` avec comme paramètre le nom du service à manipuler puis l'action à exécuter. Pour redémarrer Apache il faut par exemple appeler la commande

```
service apache2 restart
```

**Q 3.** Après avoir lu la page `a2enconf(8)` et étudié en détail le fichier `/etc/apache2/conf-available/apache2-doc.conf`, ajoutez et activez un fichier `/etc/apache2/conf-available/local-tp3.conf` permettant d'accéder à la documentation d'apache via `http://serveur/doc` en plus de `http://serveur/manual`.

**Q 4.** Après avoir étudié le fichier `/etc/apache2/sites-enabled/000-default.conf`, modifiez une directive de la configuration du serveur de façon à ce que les pages servies par défaut soient celles contenues dans le répertoire `/var/www/html/default`.

Sous Debian, toutes les fonctionnalités offertes par Apache ne sont pas activées, ni même installées par défaut. Beaucoup sont offertes par des *modules*, dont certains sont installés par défaut et d'autres non. La liste des modules disponibles est disponible dans la section *Modules* du *Manuel de référence*.

**Q 5.** Étudiez la documentation du module `mod_userdir`.

**Q 6.** Activez et configurez le module `mod_user_dir` de façon à ce que le répertoire `$HOME/web` de chaque utilisateur soit utilisé comme contenu des pages web spécifiques aux utilisateurs puis vérifiez votre configuration. La lecture de `a2enmod(8)` et du tutoriel « *Répertoires Web des utilisateurs (public\_html)* » est évidemment nécessaire.

**Q 7.** Après avoir étudié la documentation du module `mod_autoindex`, désactivez la génération automatique d'index dans le serveur par défaut configuré par `/etc/apache2/sites-enabled/000-default.conf`.

**Q 8.** Après avoir lu le tutoriel sur les « *fichiers .htaccess* » :

1. réactivez la génération automatique des index uniquement dans le répertoire web de **votre** utilisateur ;
2. protégez l'accès aux pages web de votre utilisateur par une authentification basique (avec comme login et mot de passe `toto`). Lire la page `htpasswd(1)` est sans doute une bonne idée.

**Q 9.** Après avoir lu la section « *Guide de réécriture des URLs* » dans la documentation (au minimum les parties « *Introduction aux expressions rationnelles et à mod\_rewrite* » et « *Utilisation de mod\_rewrite pour la redirection et la remise en correspondance avec le système de fichiers des URLs* »), faites en sorte que toutes les requêtes vers une URL de votre utilisateur de la forme `http://serveur/~login/<chemin>` renvoie le contenu de la page `http://serveur/~login/<chemin>.html`.

#### **Exercice 4 : Hôtes virtuels**

Le protocole HTTP permet de préciser le nom du serveur interrogé dans chacune des requêtes grâce aux entêtes et notamment le champ `Host` : , obligatoire depuis la version 1.1 du protocole.

Une machine peut-être atteinte via plusieurs noms. C'est le cas dans ce TP du serveur qui est accessible depuis le client par au moins les noms `www.pere.noel` et `www.iut.fr`.

En combinant ces deux aspects la plupart des serveurs web, comme Apache, permettent de gérer des serveurs (ou *hôtes*) virtuels (*virtual host* en anglais) : un même logiciel répondant aux requêtes adressées à plusieurs noms différents.

La section « *Hôtes virtuels* » de la documentation d'Apache décrit le fonctionnement et la configuration nécessaire. Une relecture rapide du fichier `/usr/share/doc/apache2/README.Debian.gz` n'est sans doute pas superflue.

**Q 1.** Après avoir lu la documentation idoine, créez les fichiers de configuration suivants dans `/etc/apache2/sites-available` qui permettent de définir deux hôtes virtuels :

`www.pere.noel` pour les appels à `www.pere.noel` qui renvoie les pages stockées dans `/var/www/html/www.pere.noel`  
`www.iut.fr` pour les appels à `www.iut.fr` qui renvoie les pages stockées dans `/var/www/html/iut`.

**Q 2.** Activez vos deux hôtes virtuels et vérifiez le fonctionnement de votre serveur.

La configuration des deux hôtes virtuels est sensiblement identique. Créer de nombreux hôtes virtuels peut vite devenir fastidieux si la même configuration doit être répétée des milliers de fois. Il existe un module, `mod_macro`, qui permet de définir des *macros*. Une macro est juste un modèle de configuration paramétrique. Utiliser une configuration répétitive revient juste à spécifier le nom de la macro et les valeurs effectives des paramètres. Ce module n'est pas installé par défaut dans votre version de Debian.

**Q 3.** Installez, activez et utilisez ce module pour simplifier la configuration de vos hôtes virtuels via une macro nommée `HoteVirtuel` que vous définirez dans le fichier `/etc/apache2/conf-available/local-macros.conf`.

**Q 4.** Modifiez la configuration de l'hôte virtuel `www.iut.fr` de façon à ce que toutes les requêtes sur cet hôte retourne le résultat d'une requête identique sur le site `www.iut-info.univ-lille1.fr`. Il s'agit de transformer cet hôte en simple *Reverse Proxy*.

#### **Exercice 5 : Bonus**

**Q 1.** Après avoir supprimé et purgé l'installation d'Apache sur votre serveur, refaites les questions du TP en utilisant le serveur web [nginx](#).