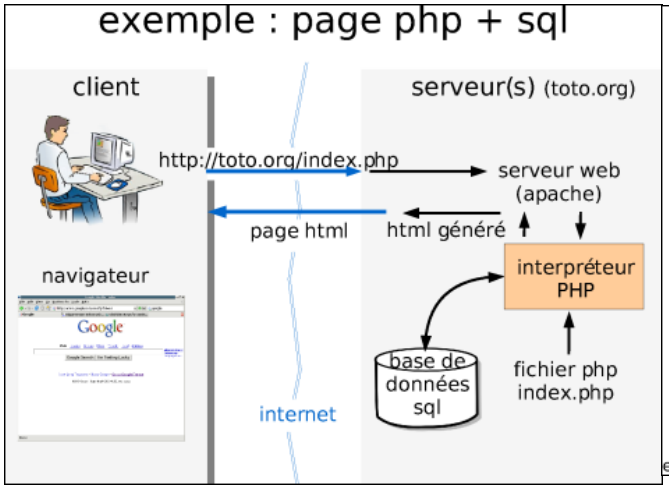
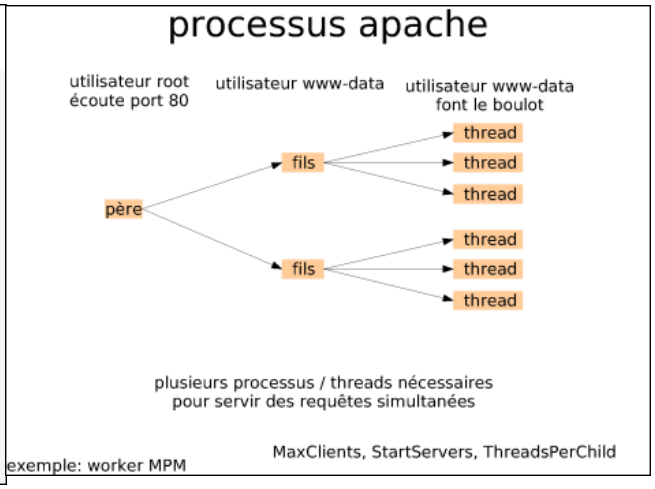
Mise en place du serveur Apache

Installation du paquet apache2 sous Linux

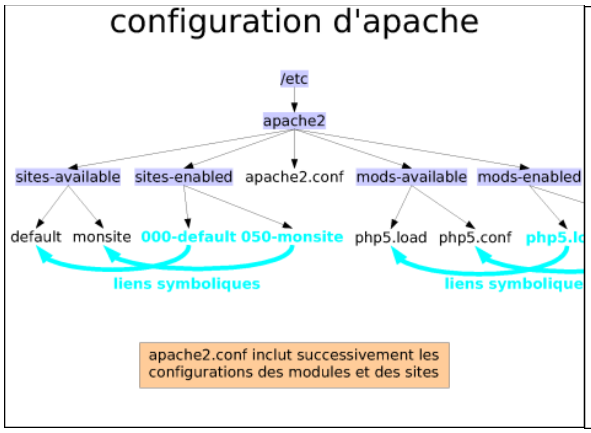
La première étape est l'installation du paquet apache2. Pour ce faire, ouvrez une console et tapez ceci :

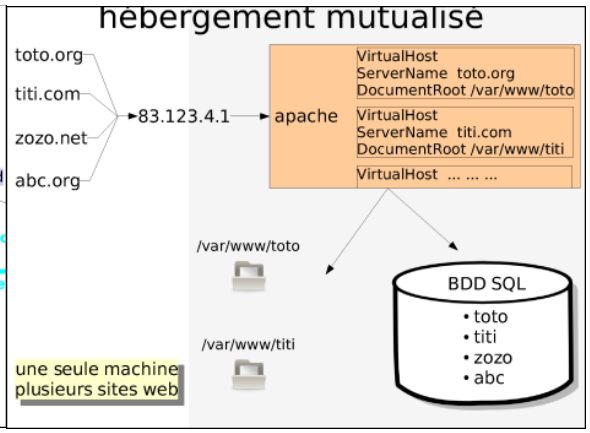
```
apt-get install apache2
```





Configuration d'apache





Fichier apache2.conf

Le fichier apache2.conf est le principal fichier de configuration du serveur web. C'est à partir de ce fichier que sont inclus les autres fichiers de configuration. Parcourez ce fichier et essayez de comprendre le fonctionnement

Les directives dans ce fichier s'appliquent par défaut à tous les hôtes virtuels gérés par le serveur web (à moins que les directives ne soient redéfinies dans la configuration d'un hôte virtuel particulier).

apache2.conf : mods

On peut ajouter au serveur apache des extensions appelés 'mods' ou 'modules'.
Les extensions sont définies par un fichier .load et par des directives de configuration dans un fichier .conf. Ces fichiers se trouvent dans un répertoire mods-available et pour qu'elle soient activés on y fait des liens symboliques dans le répertoire mods-enabled. La prise en compte de ces liens peut alors être faite tout simplement avec un Include dans le fichier apache2.conf

Visitez les répertoires en question pour voir en comprendre le fonctionnement.

Pour la suite du TP, vous aurez besoin d'installer un serveur DNS. Préparez une machine virtuelle avec ce service activé. Créez une zone et intégrez y vos hôtes

Création d'une page HTML de test

Avant de passer à la configuration du serveur apache, nous allons créer une mini page web en HTML qui sera la page d'accueil de notre serveur Apache.

Créons tous d'abord le dossier qui contiendra cette page :

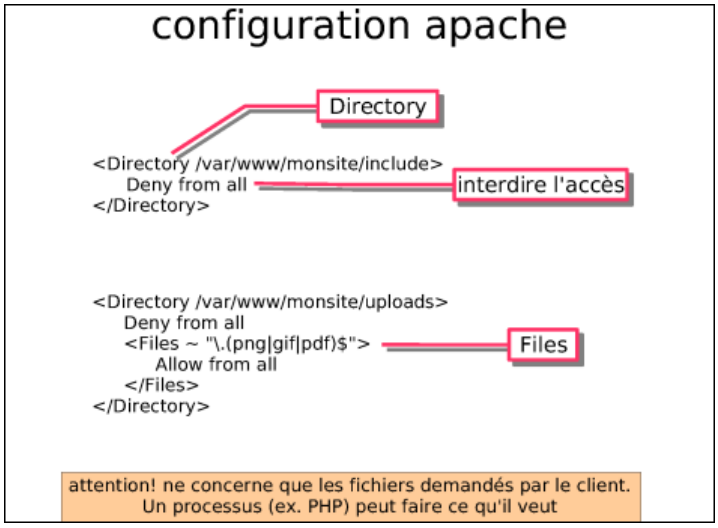
```
mkdir /var/www/site
cd /var/www/site
```

À l'intérieur de ce dossier, on crée une page HTML nommée index.html :

```
<html>
<head> <title> Notre serveur Apache </title> </head>
<body> <p> Voici la page d'accueil de votre serveur Apache ! </p> </body>
</html>
```

Enregistrez votre travail, et voilà, nous possédons maintenant une page web pour notre serveur apache. Passons désormais à la configuration du serveur.

Configuration du serveur Apache



Mise en place de sécurité

Nous allons aborder deux exemples de configuration du serveur.

1er exemple : Configuration du serveur apache sans restrictions d'accès c'est à dire sans mot de passe. Dans ce cas, tout le monde peut accéder à notre serveur.
2ème exemple : Configuration du serveur apache avec authentification basique des utilisateurs en respectant simultanément les règles suivantes : accès permis seulement à partir de la machine salon identifiée par son adresse IP ou son nom et accès par mot de passe pour les utilisateurs admin et master.

1er exemple :

Déplaçons-nous dans le dossier apache2 :

```
cd /etc/apache2/sites-available
```

Créons dans ce dossier le fichier **site**. Ce fichier contiendra la configuration du serveur apache pour le nom de domaine site.tssi.org.

```
nano site.conf
```

Et insérez à l'intérieur les quelques lignes suivantes :

```
   <VirtualHost *:80>
Indiquez également ici l'adresse IP de la machine serveur
   ServerName site.ingetis.lan
Cette ligne est optionnelle. Vous pouvez indiquez le nom de domaine de la machine si
elle en possède un
   DocumentRoot /var/www/site
Indiquez ici le chemin de l'accès au fichier index.html
   <Directory /var/www/site>
Indiquez ici le chemin du répertoire apache par défaut
   Order allow,deny
Cette ligne représente les différentes options pour un utilisateur.
Soit permettre, soit interdire l'accès
   Allow from all
Cette ligne permet l'accès au serveur à tous les utilisateurs
   </Directory>
</VirtualHost>
```

Maintenant que vous avez créé le fichier de configuration dans le dossier **sites-available**, il est nécessaire "d'activer" ce domaine en créant un lien symbolique dans le dossier **sites-enabled** (qui est le dossier contenant les sites dits "actifs") :

```
ln -s /etc/apache2/sites-available/site.conf /etc/apache2/sites-enabled/site.conf
```

Ou de taper la commande : **a2ensite site**

Afin de prendre en compte votre configuration, redémarrez le serveur apache.

```
/etc/init.d/apache2 restart
```

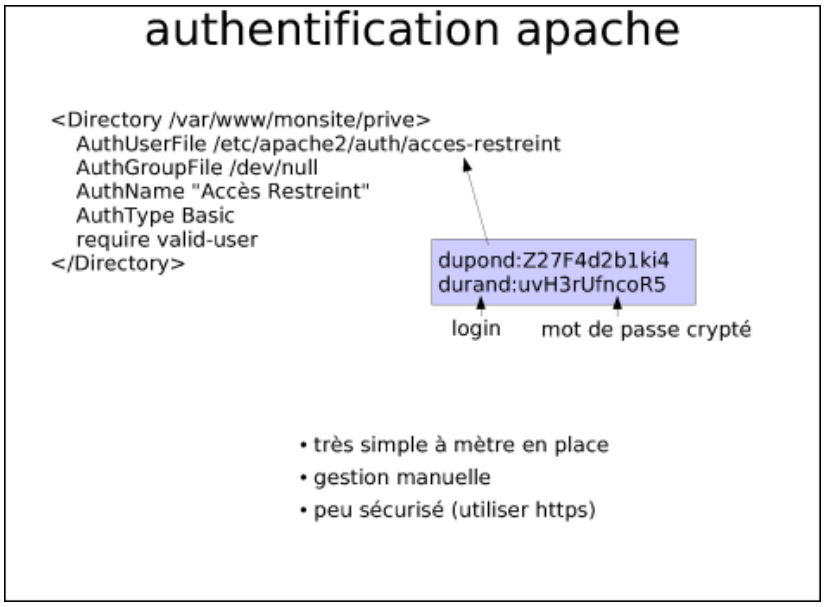
Vous pouvez désormais tester l'accès à votre serveur apache en tapant dans un navigateur web

http://192.168.0.100
ou (si vous avez configuré un serveur DNS)
http://site.ingetis.lan

Afin de suivre les connexions au serveur apache et avoir des renseignements sur les erreurs liées à son utilisation, on peut utiliser les commandes suivantes.

```
tail -f /var/log/apache2/access.log
tail -f /var/log/apache2/error.log
```

2ème exemple :



Intéressons-nous maintenant au deuxième exemple, un peu plus intéressant en configurant l'accès au serveur apache qu'à partir d'une seule machine (formation) et en protégeant l'accès par mot de passe pour les utilisateurs admin et superadmin.

Les modifications se font au niveau du même fichier :

```
nano /etc/apache2/sites-available/site.conf
```

Remplacez maintenant les lignes du fichier par celles-ci :

```
NameVirtualHost 192.168.0.100:80
<VirtualHost 192.168.0.100:80>
   ServerName site.ingetis.lan
   DocumentRoot /var/www/site
   <Directory /var/www/site>
     Order allow,deny
     Allow from 192.168.0.101
Cette ligne définit l'accès au serveur uniquement à partir de la machine
possedant l'adresse IP 192.168.0.101
     Authname "formation"
Indiquez sur cette ligne le nom de la machine qui pourra se connecter.
     Authtype basic
Cette ligne indique que le système utilisera le fichier .htpasswd pour
gérer les utilisateurs
     AuthUserFile /var/www/site/.htpasswd
Il s'agit du fichier qui sera utilisé pour gérer les connexions.
     Require valid-user
Cette ligne indique que pour que la connexion s'effectue, il est necessaireque l'util
isateur existe.
     Satisfy all
</Directory>
</VirtualHost>
```

Enregistrez et quittez le fichier. Passons maintenant à la configuration du fichier **.htaccess** dans le dossier à protéger :

```
Authname "Accès protégé"
Cette ligne permettra d'afficher Accès protégé lors de l'accès au serveur
AuthType Basic
AuthUserFile /var/www/site/.htpasswd
Cette ligne indique le chemin du fichier qui contiendra les mots de passe
de connexion des utilisateurs.
Require valid-user
```

Pour finir, créons l'accès pour les deux utilisateurs admin et superadmin. Tapez simplement dans une terminal les lignes suivantes :

```
htpasswd -c -m .htpasswd admin
htpasswd -m .htpasswd superadmin
```

Enfin, afin de prendre en compte votre configuration, redémarrez le serveur apache.

```
/etc/init.d/apache2 restart
```

Voilà, vous pouvez tester l'accès à votre serveur apache. Normalement, vous ne devriez pouvoir y accéder qu'à partir de la machine "formation" identifiée par l'adresse IP 192.168.0.101 et en utilisant un des deux utilisateurs admin ou superadmin.

Ceci termine la partie concernant la configuration d'un serveur HTTP apache.

Passons maintenant aux différents types d'hébergements possibles pour un serveur apache.

Hébergement virtuel

Hébergement virtuel par nom de domaine

Ce type d'hébergement est utilisé pour accueillir plus d'un nom de domaine sur le même ordinateur sur la même adresse IP.

Afin de tester ce type d'hébergement virtuel, chaque machine doit être recensée sous plusieurs noms différents sur le DNS en utilisant les enregistrements de type A et CNAME.

Rajoutez dans votre DNS les lignes suivantes :

```
siteVH1 IN CNAME site
siteVH2 IN CNAME site
```

Créons les trois fichiers index.html contenant comme informations le nom de domaine du serveur virtuel et plaçons-les dans les répertoires correspondant :

```
mkdir /var/www/siteVH1
mkdir /var/www/siteVH2
```

À l'intérieur de chaque dossier, on crée une page HTML nommée index.html :

```
<html>
<head> <title> Notre serveur Apache </title> </head>
<body> <p> Voici la page d'accueil de votre serveur Apache siteVH1 ! </p>
</body>
</html>
```

```
<html>
<head> <title> Notre serveur Apache </title> </head>
<body> <p> Voici la page d'accueil de votre serveur Apache siteVH2 ! </p>
</body>
</html>
```

On crée deux fichiers de configuration pour nos deux nouveaux domaines dans le dossier /etc/apache2/sites-available afin de créer ces deux serveurs virtuels pointant sur deux répertoires différents.

```
nano /etc/apache2/sites-available/siteVH1.conf
```

On ajoute dans ce fichier les lignes suivantes :

```
<VirtualHost *:80>
            ServerName siteVH1.ingetis.lan
            ServerAlias siteVH1
            DocumentRoot /var/www/siteVH1
            <Directory /var/www/siteVH1>
                    Order allow,deny
                    Allow from all
            </Directory>
</VirtualHost>
```

De même pour l'autre domaine :

```
nano /etc/apache2/sites-available/siteVH2.conf
```

On ajoute dans ce fichier les lignes suivantes :

```
<VirtualHost *:80>
            ServerName siteVH2.ingetis.lan
            ServerAlias siteVH2
            DocumentRoot /var/www/siteVH2
            <Directory /var/www/siteVH2>
                    Order allow,deny
            Allow from all
            </Directory>
</VirtualHost>
```

C'est l'heure d'activer nos deux hébergements virtuels. Pour ce faire, il faut créer des liens symboliques dans le dossier /etc/apache2/sites-enabled (qui recensent les sites actifs) pour que la configuration soit lue lors du démarrage d'apache.

```
ln -s /etc/apache2/sites-available/siteVH1.conf /etc/apache2/sites-
enabled/siteVH1.conf
ln -s /etc/apache2/sites-available/siteVH2.conf /etc/apache2/sites-
enabled/siteVH2.conf
```

Redémarrez maintenant le serveur apache !

```
/etc/init.d/apache2 restart
```

Vous devriez pouvoir accéder à chaque page web correspondante à chaque nom de domaine.

- http://site.ingetis.lan
- http://siteVH1.ingetis.lan
- http://siteVH2.ingetis.lan

Tout en ne possédant qu'une seule adresse IP, il est possible de définir plusieurs sous-domaines pour gérer différentes pages web.

Hébergement virtuel par port

Ce type d'hébergement permet d'utiliser un serveur web en précisant le port qu'il utilise. Si le port n'est pas précisé, la connexion ne s'effectuera pas !

Dans ce cas, chaque serveur sera adressé par une requête HTTP à l'aide d'un numéro de port différent. Par exemple pour la machine "formation" :

http://site.ingetis.lan:80
http:// siteVH1.ingetis.lan:40001
http:// siteVH2.ingetis.lan:40002

Afin que apache écoute sur les ports correspondants, on les ajoute dans le fichier ports.conf :

```
/etc/apache2/ports.conf
Listen 80
Listen 40001
Listen 40002
```

Effectuons maintenant les modifications dans chaque fichier de configuration pour nos différents domaines :

```
nano /etc/apache2/sites-available/site.conf
```

```
<VirtualHost *:80>
        ServerName site.ingetis.lan
        ServerAlias site
        DocumentRoot /var/www/site
        <Directory /var/www/site>
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

Puis dans le fichier de configuration du domaine siteVH1 :

```
<VirtualHost *:40001>
        ServerName siteVH1.ingetis.lan
        ServerAlias siteVH1
        DocumentRoot /var/www/siteVH1
        <Directory /var/www/siteVH1>
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

Puis dans le fichier de configuration du domaine siteVH2 :

```
<VirtualHost *:40002>
        ServerName siteVH2.ingetis.lan
        ServerAlias siteVH2
        DocumentRoot /var/www/siteVH2
        <Directory /var/www/siteVH2>
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

C'est fini ! Il ne vous reste plus qu'à tester :

- http://site.ingetis.lan:80
- http://siteVH1.ingetis.lan:40001
- http://siteVH2.ingetis.lan:40002

Hébergement virtuel par adresse IP

Ce type d'hébergement permet, tout en associant deux adresses IP à une interface, de se connecter aux pages associées à chaque adresse IP.

Pour démarrer, on doit créer un alias eth0:1 sur l'interface eth0 afin d'associer 2 adresses ip à une seule interface. On ajoute pour cela les lignes suivantes dans le fichier /etc/network/interfaces :

```
auto ens33:1
iface ens33:1 inet static
address 192.168.0.102
netmask 255.255.255.0
broadcast 192.168.0.255
```

On modifie notre fichier Pour le domaine tssi :

```
nano /etc/apache2/sites-available/site.conf
```

```
<VirtualHost 192.168.0.102:80> //On indique simplement que pour accéder au domaine si
te.tssi.org, il faudra utiliser l'adresse IP 192.168.0.102
        ServerName site.ingetis.lan
        ServerAlias site
        DocumentRoot /var/www/site
        <Directory /var/www/site>
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

Vous pouvez faire la même chose pour les domaines siteVH1 et siteVH2 si vous le souhaitez. Puis tester :

- http://192.168.0.100:80
- http://192.168.0.102:80

Installer PHP puis MySQL

On installera PHP puis MySQL. On spécifiera un mot de passe administrateur pour la base de données (root). Il ne faudra surtout pas perdre ou oublier ce mot de passe.

Afin de laisser la possibilité à chacun des utilisateurs du système de poster ses pages web dans son dossier personnel, mettez un accès en place pour chaque user.

Par exemple l'utilisateur toto pourra poster ses pages web dans le dossier /home/toto/public_html

Elles seront accessibles à l'adresse http://site.ingetis.lan/~toto

Créer 2 utilisateurs linux : user1 et user2

Sur l'URL http://site.ingetis.lan/~user1 on installera GLPI

Sur l'URL http://site.ingetis.lan/~user2 on installera un Wordpress

Sécurisation :

Pour plus de sécurité, nous allons passer notre serveur web en https et activer le mode SSL

La 1ère étape est d'installer la prise en charge d'openssl (par apt-get)

Nous allons créer notre clé de chiffement :

openssl genrsa -out server.key 2048

Il faut ensuite générer le certificat :

```
openssl req -outform PEM -new -key server.key -x509 -days 365 -out server.crt
```

Rappelez le rôle d'un certificat :

Quel est le type de cryptage qui utilise l'algorithme RSA ?

Ce certificat va donc utiliser RSA avec une clé de 2048 bits, et créer les certificats server.crt et server.key dans le dossier apache2 d'une durée de validité de 365 jours.

Faites en sorte que le fichier .crt soit en lecture seule pour le propriétaire et pour les membres de son groupe uniquement.

Le certificat et la clé générés par défaut sont stockés dans « **/etc/ssl/certs** » et « **/etc/ssl/private** » mais si vous souhaitez stocker vos fichiers ailleurs, il faudra l'indiquer à Apache. Pour cela, éditez le fichier « **default-ssl** » contenant la configuration du site SSL. Il se trouve ici :

```
/etc/apache2/sites-available/default-ssl
```

Modifiez ces deux options si nécessaire afin d'indiquer le chemin vers vos fichiers :

```
SSLCertificateFile /chemin/server.crt
```

```
SSLCertificateKeyFile /chemin/server.key
```

Enregistrez puis quittez le fichier de configuration du site SSL. Ensuite, activez le module SSL et le site SSL :

```
a2enmod ssl
a2ensite default-ssl
service apache2 reload
```

Accédez à votre site en utilisant le préfixe HTTPS dans l'URL, cela devrait fonctionner

Si vous souhaitez qu'on accède à votre site web uniquement via le protocole HTTPS, il est intéressant de désactiver le site accessible sur le port 80 c'est-à-dire le site « **default** ». Pour cela on utilise la commande « **a2dissite** » qui permet de désactiver des sites dans Apache 2.

```
a2dissite default
```

Vous pouvez ensuite essayer d'accéder à votre site en HTTP et vous verrez qu'il n'est plus accessible