



Installation d'un serveur HTTP (Hypertext Transfer Protocol) sous Débian 6





Table des matières

Installation d'un serveur HTTP (Hypertext Transfer Protocol) sous Débian 6	1
• Serveur HTTP ou Serveur WEB	3
• Le modèle Client-Serveur	3
Le serveur Web Apache.....	9
L'arborescence du serveur Web Apache2	9
1 – Mise en Œuvre - Topologie et versions utilisées.....	11
2 – Installation et Configuration du serveur WEB Apache 2.....	12
Installation de plusieurs serveurs web (Virtual Hosts) sur une instance Apache2	12
3 – Vérification du fonctionnement du server HHTTP WEB Apache 2.....	18

● Serveur HTTP ou Serveur WEB

Le serveur Web s'inscrit dans l'architecture informatique dite client-serveur.

Son rôle est de " servir " des pages Web sur la demande d'un internaute utilisant un logiciel client Web dont la dénomination usuelle est navigateur Web. Pour cela, il met en œuvre le protocole correspondant, HTTP.

Un ordinateur faisant office de serveur Web ou exécutant, entre autres processus serveurs, un processus serveur Web, doit être relié par une connexion permanente à Internet, avec une adresse IP fixe, de sorte que les sites Web qu'il sert soient disponibles en permanence.

On peut également parler de serveur Web pour désigner un serveur de pages HTML ne fonctionnant que sur un réseau local, sans liaison directe à Internet.

Mais si l'on veut être rigoureux, il vaut mieux dans ce cas parler de serveur HTTP.

Les moyennes et grandes entreprises disposent en général de leur propre serveur Web, tandis que les travailleurs indépendants et les particuliers font héberger leurs pages Web chez leur FAI.

Une solution intermédiaire est également proposée par les Fournisseurs de Services Internet (FSI), notamment à l'intention des PME : l'hébergement d'un ordinateur serveur Web entièrement réservé à l'entreprise.

● Le modèle Client-Serveur

C'est la description du fonctionnement coopératif entre le serveur et le client. Les services internet sont conçus selon cette architecture. Ainsi, chaque application est composée de logiciel serveur et logiciel client. A un logiciel serveur, peut correspondre plusieurs logiciels clients développés dans différents environnements: Unix, Mac, PC...; la seule obligation est le respect du protocole entre les deux processus communicants. Ce protocole étant décrit dans un RFC (Request For Comment).

L'architecture client/serveur désigne un mode de communication entre des ordinateurs et des logiciels. Les mots « serveur » et « client » désignent les logiciels de type serveur et client dans cette architecture, logiciels fonctionnant sur les ordinateurs qu'on nomme par abus de langage serveur informatique et poste client.

- Le serveur est à l'écoute d'un réseau informatique, prêt à répondre aux requêtes envoyées par des clients.
- Les clients sont généralement pilotés par des utilisateurs. Ils prennent l'initiative d'envoyer des requêtes au serveur, puis attendent la réponse pour la donner, le cas échéant, à l'utilisateur.

- Un serveur est capable de servir plusieurs clients simultanément, jusqu'à plusieurs milliers.
- Le serveur et le client utilisent le même protocole de communication.

L'architecture client/serveur correspond à la généralisation des ordinateurs de type PC dans les entreprises (dans les agences, les points de vente, ...) et est une architecture à deux niveaux :

- les règles de gestion, les traitements et les accès aux données sont réalisés sur le serveur central,
- les contrôles de saisie, les enchaînements des dialogues sont effectués sur les postes client.

Ce modèle minimise les flux sur le réseau et tire parti de la puissance des machines locale et centrale.



Caractéristiques de l'architecture Client-Serveur :

On peut essayer de préciser ce que l'on entend par modèle client-serveur, bien qu'il n'y ait pas de consensus sur une définition, en donnant les caractéristiques suivantes :

Service :

Le serveur est fournisseur de services. Le client est consommateur de services.

Protocole :

C'est toujours le client qui déclenche la demande de service. Le serveur attend passivement les requêtes des clients.

Partage des ressources :

Un serveur traite plusieurs clients en même temps et contrôle leurs accès aux ressources.

Localisation :

Le logiciel client-serveur masque aux clients la localisation du serveur.

Hétérogénéité :

Le logiciel client-serveur est indépendant des plates-formes matérielles et logicielles.

Redimensionnement :

Il est possible d'ajouter et de retirer des stations clientes. Il est possible de faire évoluer les serveurs.

Intégrité :

Les données du serveur sont gérées sur le serveur de façon centralisée. Les clients restent indépendants.

Souplesse et adaptabilité :

On peut modifier le module serveur sans toucher au module client. La réciproque est vraie. Si une station est remplacée par un modèle plus récent, on modifie le module client (en améliorant l'interface, par exemple) sans modifier le module serveur

Présentation de l'architecture Client-Serveur :

Dans l'architecture client-serveur, une application est constituée de trois parties : l'interface utilisateur, la logique des traitements, et la gestion des données. Le client n'exécute que l'interface utilisateur (interfaces graphiques de type Windows) et la logique des traitements (formuler la requête), laissant au serveur de bases de données la gestion complète des manipulations de données. Le – qui apparaît dans client-serveur correspond à tout un ensemble complexe de logiciels appelé middleware qui se charge de toutes les communications entre les processus. Nous essayons maintenant de donner les principales évolutions en cours dans le domaine des systèmes d'information, basées sur le modèle client-serveur.

Les deux types de réseaux les plus fréquents sont :

- Le réseau local sans serveur, connexion de poste à poste
- les réseaux organisés autour d'un serveur (Client/Serveur)
- Ces deux types de réseau ont des capacités différentes.

Serveurs : ordinateurs spécialisés dans la fourniture et le stockage des ressources partagées des utilisateurs du réseau

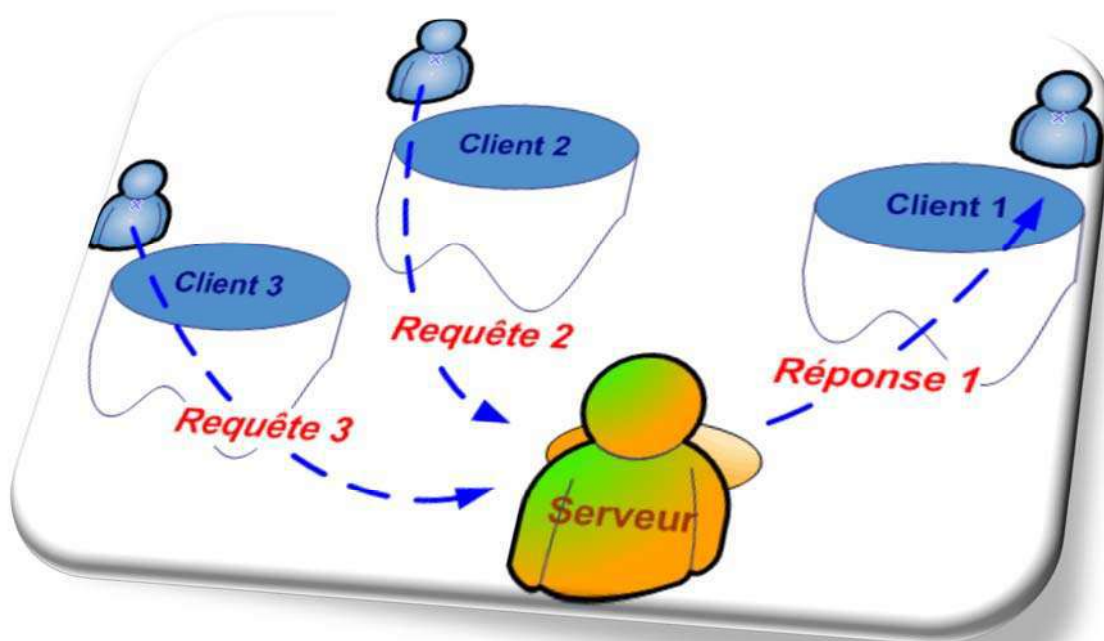
Clients : ordinateurs qui accèdent aux ressources partagées fournies par un serveur du réseau.

Le type de réseau à installer dépend des critères suivants :

- Taille de l'entreprise ou de l'organisation
- Niveau de sécurité nécessaire
- Type d'activité
- Volume du trafic sur le réseau
- Besoins des utilisateurs du réseau
- Budget (disposition financière)

Fonctionnement du Serveur HTTP :

Le principe de base est une connexion de type client/serveur : un client (le navigateur) se connecte sur un serveur, émet une requête et le serveur répond. La connexion est une simple ouverture de socket TCP/IP généralement sur port 80, mais on utilise aussi de temps en temps le port 8080 (pour les connexions sur proxy, par exemple).



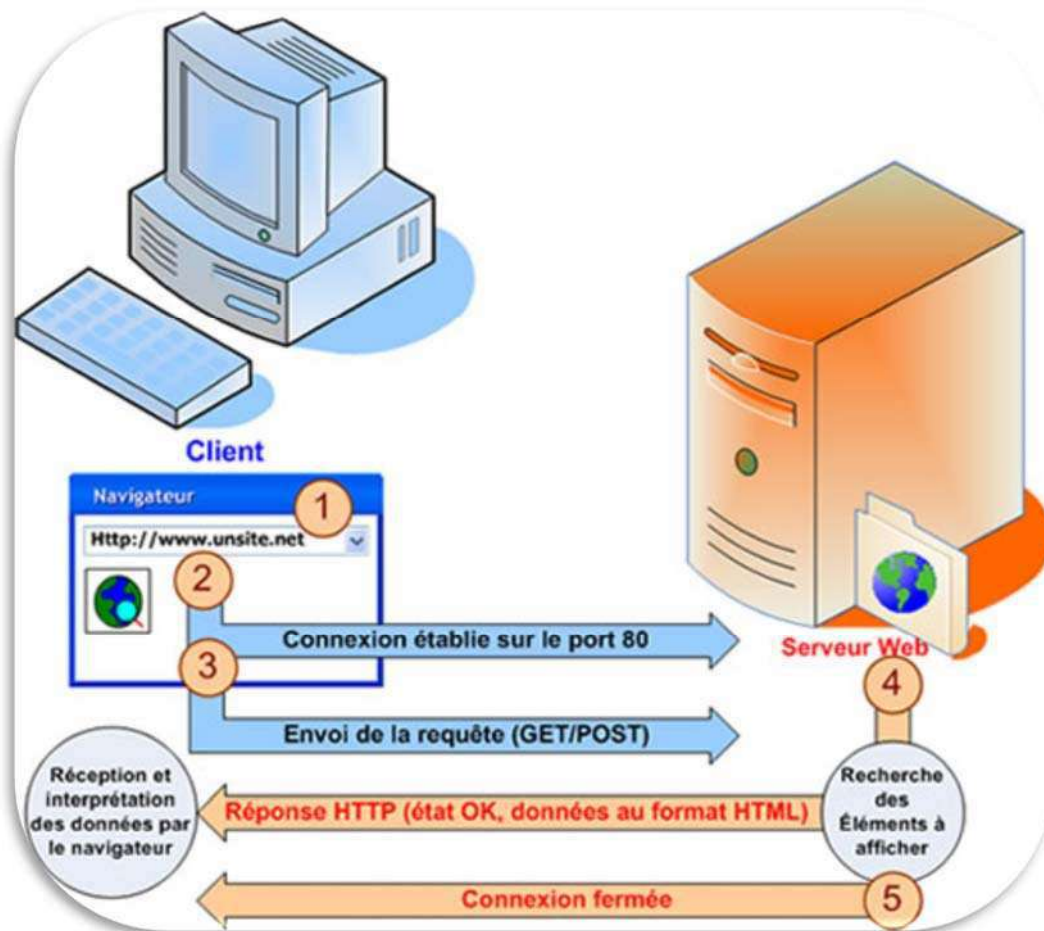
Le protocole le plus utilisé pour communiquer avec un serveur Web sur Internet est le protocole HTTP (Hyper Text Transfer Protocole). Il définit les règles de communication entre un client (navigateur) et un serveur Web.

La requête la plus simple du protocole HTTP est formée de GET suivi d'une URL qui pointe sur des données (fichier statiques, traitement dynamique...).

Elle est envoyée par un navigateur quand nous saisissons directement une URL dans le champ d'adresse du navigateur.

Le serveur HTTP répond en renvoyant les données demandées.

En détail :



En tapant l'URL d'un site, l'internaute envoie (via le navigateur) une requête au serveur. Une connexion s'établit entre le client et le serveur sur le port 80 (port par défaut d'un serveur Web).

Le navigateur envoie une requête demandant l'affichage d'un document. La requête contient entre autres la méthode (GET, POST, etc.) qui précise comment l'information est envoyée. Le serveur répond à la requête en envoyant une réponse HTTP composée de plusieurs parties, dont : l'état de la réponse, à savoir une ligne de texte qui décrit le résultat du serveur (code 200 pour un accord, 400 pour une erreur due au client, 500 pour une erreur due au serveur), les données à afficher.

Une fois la réponse reçue par le client, la connexion est fermée. Pour afficher une nouvelle page du site, une nouvelle connexion doit être établie.

Avant de commencer :

Un serveur HTTP ou daemon HTTP est un logiciel servant des requêtes respectant le protocole de communication client-serveur Hypertext Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

HTTP : est le protocole du web qui est en charge de transférer les pages web du serveur vers le navigateur. Il transmet également au serveur les informations saisies dans un formulaire.

URL : Uniform Resource Locator, indication d'un emplacement d'un document et le service permettant d'y accéder.

Syntaxe : **<protocole>://<adresse du serveur >: <port >/ <chemin du document>**

HTML : HyperText Markup Language, est un langage de présentation de document, à travers des balises ou tags qui sont mélangées avec le texte du document.

C'est le logiciel du navigateur qui interprète un code source HTML et qui affiche la page.

Possibilités de HTML :

1. Structurer un document : titres, sous-titres, paragraphes, listes.
2. Enrichir un document : gras, souligné, italique.
3. Inclure des liens hypertextes.
4. Inclure des références à des données multimédia : image, son, vidéo
5. Créer des formulaires.

```
<HTML>
< ! - - commentaire - - >
  <HEAD>
    <TITLE> Le titre </TITLE>
  </HEAD>
  <BODY>
    <H1> Très Gros Titre</H1>
    <p> Le corps de la page : ce qui apparaît dans la fenêtre du
navigateur <p>
  </BODY>
</HTML>
```

<HTML> </HTML> Délimite un document HTML

<HEAD> </HEAD> l'en-tête du document

<BODY> </BODY> le corps du document

Les spécifications du style du document : police de caractères, couleurs, emplacement ... sont exclues du document HTML et mises dans un document CSS (Cascading Style Sheets), qui prend en charge l'aspect du site (feuille de style).

Le serveur Web Apache

L'arborescence du serveur Web Apache2

Les fichiers de paramétrage se trouvent sous linux dans le répertoire **/etc/apache2**, les fichiers de données (les pages web) dans **/var/www** et les fichiers logs dans **/var/log/apache2**.

apache2.conf -> fichier de base du serveur, contient des « includes » (voir plus bas)

conf.d -> paramétrage du serveur d'une manière globale : par exemple pour la sécurité, le character-set ...

envvars -> les variables environnement

httpd.conf -> fichier présent uniquement pour la compatibilité avec certains logiciels qui ont besoin de voir ce fichier, même vide. Plus utilisé.

magic -> permet de définir le type de fichier d'après son extension, pour que le serveur adapte sa réponse au client (choix du type MIME renvoyé, la langue,..)

mods-available -> les modules supplémentaires disponibles

mods-enabled -> les modules supplémentaires activés

.htaccess -> fichier de configuration dynamique du serveur. Son emplacement spécifie le répertoire qu'il paramètre.

ports.conf -> pour définir et ajouter des ports d'écoute du serveur

sites-available -> les fichiers de configuration des sites disponibles, mais pas en ligne

sites-enabled -> les fichiers de configuration des sites en ligne

commenter default port * :80

Un des points forts d'Apache est son architecture modulaire.

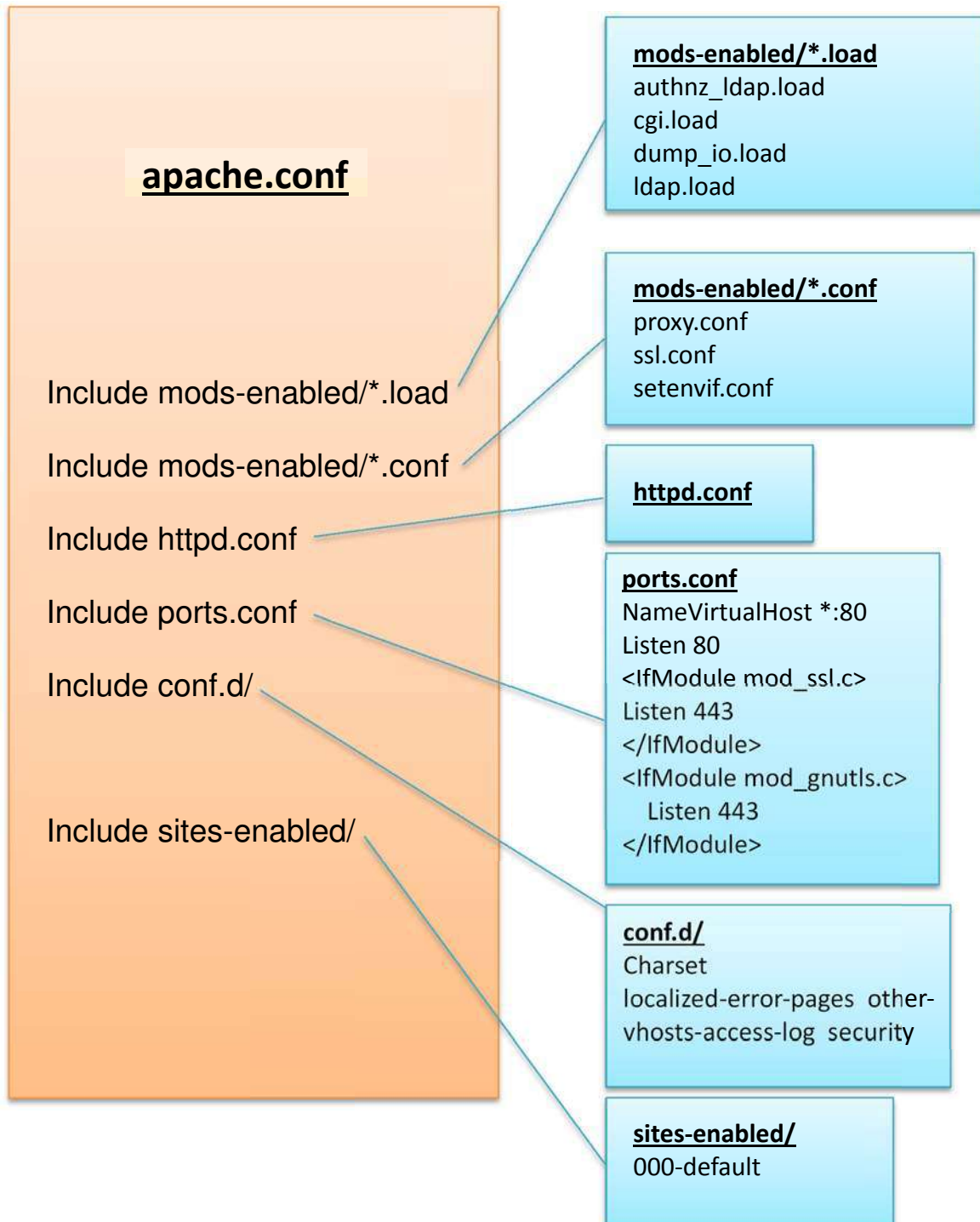
Il est facile d'ajouter ou de supprimer des modules selon les besoins : database, langage de programmation du style Python, Perl, PHP, Java,

Après chaque modification des fichiers de configuration, redémarrer le serveur avec la commande :

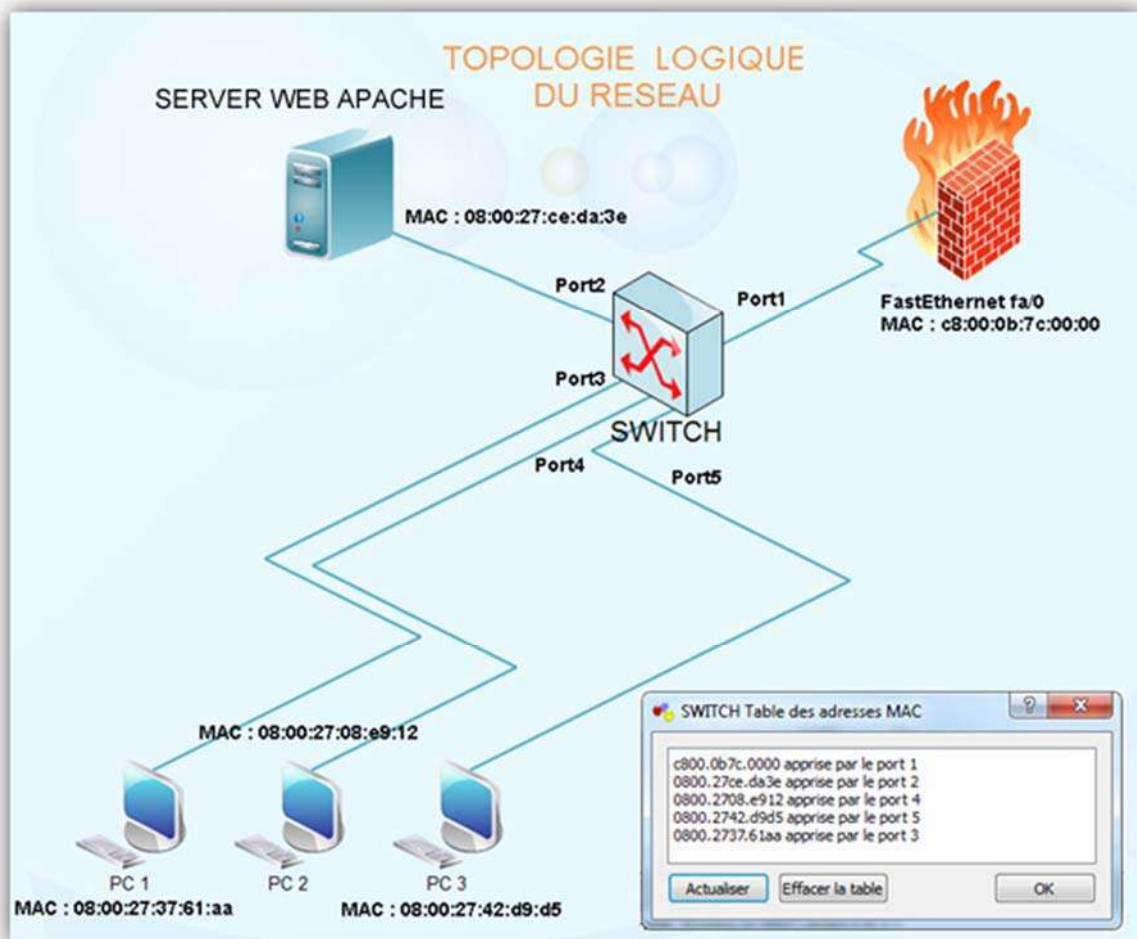
```
# service apache2 reload
```

Le chargement des fichiers se fait par « **include** » dans le fichier **apache2.conf**

```
Include mods-enabled/*.load -> le chargement du module dans le  
serveur  
Include mods-enabled/*.conf -> le fichier de paramétrage du module  
Include httpd.conf  
Include ports.conf  
Include conf.d/  
Include sites-enabled/
```



1 – Mise en Œuvre Topologie et versions utilisées



L'installation sera effectuée sous Linux Débian 2.6.32-5-amd64.

La version du serveur Apache sera la 2.2

2 - Installation et Configuration du serveur WEB Apache 2

```
# aptitude install apache2
```

En installant le service Apache2, un service web de base est déjà disponible en ligne dans le fichier `/etc/apache2/sites-enabled/000-default`.

Il suffit d'ajouter et de modifier les pages HTML dans le répertoire `/var/www` pour donner forme à votre site.

En tapant l'adresse de votre serveur web dans votre navigateur web vous devriez obtenir le message : **it work's !!**



Installation de plusieurs serveurs web (Virtual Hosts) sur une instance Apache2

Sources : <http://httpd.apache.org/docs/2.2/fr/vhosts/ip-based.html>

Un **Virtual Host** est un serveur web servi par Apache2.

Les Virtual Hosts sont gérés comme des modules.

Chaque site obtient un fichier de configuration qui lui est propre.

En ajoutant une adresse IP (alias) à une interface virtuelle, on peut créer plusieurs serveurs virtuels.

Penser à adapter les fichiers de configuration à votre propre infrastructure !!!

Création des interfaces virtuelles dans linux :

Dans le fichier `/etc/network/interfaces` ajouter (après l'interface `eth0`) 2 interfaces `eth0:0` et `eth0:1` en y adressant les paramètres IP :

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
address 172.25.205.244
netmask 255.255.255.0
gateway 172.25.205.254

auto eth0:0
iface eth0:0 inet static
address 172.25.205.245
netmask 255.255.255.0
gateway 172.25.205.254
broadcast 172.25.205.255
network 172.25.205.0
dns-nameservers 172.25.205.244

auto eth0:1
iface eth0:1 inet static
address 172.25.205.246
netmask 255.255.255.0
gateway 172.25.205.254
broadcast 172.25.205.255
network 172.25.205.0
dns-nameservers 172.25.205.244
```

Informations à ajouter manuellement :

- adresse IP de l'interface
- masque de sous réseau
- passerelle par défaut
- adresse de diffusion
- adresse IP du réseau
- adresse IP du serveur DNS principal

- Les deux interfaces virtuelles

Activer les deux interfaces virtuelles

```
# ifup eth0:0
# ifup eth0:1
```

Puis vérifier avec la commande `ifconfig` que les 3 interfaces sont bien disponibles.

Le répertoire `/etc/apache2/sites-available` vous permet de préparer les sites avant de les mettre en ligne.

Le plus simple est d'utiliser le fichier par défaut (**default**, le *Default Virtual Host*), de le copier et de le modifier selon vos besoins.

Nous prenons pour exemples les noms de **www2** et **www3** pour nos deux sites virtuels.

Créer un fichier `www2` dans le répertoire `/etc/apache2/sites-available` .

Les adresses IP à utiliser sont celles configurées précédemment dans le fichier `/etc/network/interfaces` : **eth0:0** , **eth0:1**

Le premier site web virtuel `www2` :

```
<VirtualHost 172.25.205.245:80 >
ServerAdmin webmaster_www2@grp5.info-msj.net
DocumentRoot /var/www/www2
ServerName www2.grp5.info-msj.net
ErrorLog /var/log/apache2/www2_error_log
TransferLog /var/log/apache2/www2_access_log
</VirtualHost>
```

Dans ce fichier vous retrouvez :

- l'adresse IP du 1^{er} serveur web virtuel **www2** accessible sur le port 80 (http)
- l'adresse email de l'administrateur système
- le répertoire de stockage du serveur web **www2**
- le nom par lequel le serveur sera accessible (répertoire.nom-de-votre-domaine)
- l'emplacement des fichiers de log

Le deuxième site web virtuel `www3` :

```
<VirtualHost 172.25.205.246:80 >
ServerAdmin webmaster_www3@grp5.info-msj.net
DocumentRoot /var/www/www3
ServerName www3.grp5.info-msj.net
ErrorLog /var/log/apache2/www3_error_log
TransferLog /var/log/apache2/www3_access_log
</VirtualHost>
```


Pour activer les 2 sites taper la commande suivante :

```
# a2ensite www2
```

```
# a2ensite www3
```

Dans le fichier `/etc/hosts` ajouter les 2 adresses :

```
127.0.0.1      localhost.localdomain  localhost
127.0.1.1      francois.GRP5.info-msj.net  francois
172.25.205.244 francois.GRP5.info-msj.net  francois
172.25.205.245 www2.GRP5.info-msj.net  www2
172.25.205.246 www3.GRP5.info-msj.net  www3

# The following lines are desirable for IPv6 capable hosts
::1          ip6-localhost ip6-loopback
fe00::0      ip6-localnet
ff00::0      ip6-mcastprefix
ff02::1      ip6-allnodes
ff02::2      ip6-allrouters
```

adresses IP des 2 serveurs web

Respecter la syntaxe IP <TAB>ADRESSE-DU-SERVEUR<TAB>Nom du serveur web

Créer les répertoires `/var/www/www2` et `/var/www/www3`

Copier le fichier `index.html` qui se trouve dans `/var/www` dans chacun de ces répertoires.

```
root@francois:/# cd var
root@francois:/var# cd www
root@francois:/var/www# ls
index.html  ldap  webmail  www2  www3
root@francois:/var/www# tree
.
├── index.html
├── ldap -> /usr/share/phpldapadmin/
├── webmail -> /usr/share/squirrelmail/
├── www2
│   └── index.html
└── www3
    └── index.html

4 directories, 3 files
root@francois:/var/www# _
```


Puis modifier le fichier **index.html** comme ceci par exemple pour le **www2** :

```
cd /var/www/www2

# cat index.html

<html><body><h1>Bienvenue sur ce site www2 de grp5.info-msj.net</h1>
<p>Page par défaut pour site www2.grp5.info-msj.net</p>
</body></html>
```

Modifier ensuite le fichier **/etc/apache2/ports.conf** :

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default
# This is also true if you have upgraded from before 2.2.9-3 (i.e. from
# Debian etch). See /usr/share/doc/apache2.2-common/NEWS.Debian.gz and
# README.Debian.gz

NameVirtualHost *:80
NameVirtualHost www2.grp5.info-msj.net:80
NameVirtualHost www3.grp5.info-msj.net:80
Listen 80

<IfModule mod_ssl.c>
# If you add NameVirtualHost *:443 here, you will also have to change
# the VirtualHost statement in /etc/apache2/sites-available/default-ssl
# to <VirtualHost *:443>
# Server Name Indication for SSL named virtual hosts is currently not
# supported by MSIE on Windows XP.
Listen 443
</IfModule>

<IfModule mod_gnutls.c>
Listen 443
</IfModule>
```

- l'adresse IP des 2 serveurs web virtuels **www2** et **www3** accessibles sur le port 80 (http)

Ajouter maintenant les enregistrements A et PTR dans le serveur DNS (dans les 2 fichiers de votre zone) des 2 serveurs web :

```
$ORIGIN grp5.info-msj.net.
$TTL 1600
@ IN SOA francois.grp5.info-msj.net. root.grp5.info-msj.net. (
    2013022809 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

@ IN NS francois.grp5.info-msj.net.
@ IN MX 10 francois.grp5.info-msj.net.
@ IN MX 10 ldap.grp5.info-msj.net.
@ IN MX 10 test.grp5.info-msj.net.
phpmyadmin IN A 172.25.205.123
owncloud IN A 172.25.205.113
francois IN A 172.25.205.244
ldap IN A 172.25.205.101
test IN A 172.25.205.133
www2.grp5.info-msj.net. IN A 172.25.205.245
www3.grp5.info-msj.net. IN A 172.25.205.246
```

```
$ORIGIN 205.25.172.in-addr.arpa.
$TTL 1600
@ IN SOA francois.grp5.info-msj.net. root.grp5.info-msj.net. (
    2013022809 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

@ IN NS francois.grp5.info-msj.net.
123 IN PTR owncloud.grp5.info-msj.net.
133 IN PTR test.grp5.info-msj.net.
113 IN PTR phpmyadmin.grp5.info-msj.net.
244 IN PTR francois.grp5.info-msj.net.
245 IN PTR www2.grp5.info-msj.net.
246 IN PTR www3.grp5.info-msj.net.
101 IN PTR ldap.grp5.info-msj.net.
```

Lancer enfin le service apache :

```
# service apache2 restart
```

3 – Vérification du fonctionnement du serveur HHTP WEB Apache 2

La commande **dig** permet de vérifier la bonne résolution des 2 serveurs web par le serveur DNS

```
root@francois:/etc/apache2/sites-available# dig www3.grp5.info-msj.net
; <<>> DiG 9.7.3 <<>> www3.grp5.info-msj.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38710
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;www3.grp5.info-msj.net.          IN      A
;; ANSWER SECTION:
www3.grp5.info-msj.net. 1600    IN      A      172.25.205.246
;; AUTHORITY SECTION:
grp5.info-msj.net.      1600    IN      NS     francois.grp5.info-msj.net.
;; ADDITIONAL SECTION:
francois.grp5.info-msj.net. 1600 IN      A      172.25.205.244
;; Query time: 1 msec
;; SERVER: 172.25.205.244#53(172.25.205.244)
;; WHEN: Tue Mar 12 15:40:43 2013
;; MSG SIZE  rcvd: 95
```

```
root@francois:/etc/apache2/sites-available# dig www2.grp5.info-msj.net
; <<>> DiG 9.7.3 <<>> www2.grp5.info-msj.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4052
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;www2.grp5.info-msj.net.          IN      A
;; ANSWER SECTION:
www2.grp5.info-msj.net. 1600    IN      A      172.25.205.245
;; AUTHORITY SECTION:
grp5.info-msj.net.      1600    IN      NS     francois.grp5.info-msj.net.
;; ADDITIONAL SECTION:
francois.grp5.info-msj.net. 1600 IN      A      172.25.205.244
;; Query time: 0 msec
;; SERVER: 172.25.205.244#53(172.25.205.244)
;; WHEN: Tue Mar 12 15:39:49 2013
;; MSG SIZE  rcvd: 95
```

Effectuer un **ping** sur les 2 adresses IP des serveurs web pour tester leur bonne accessibilité :

```
root@francois:/etc/apache2/sites-available# ping 172.25.205.245
PING 172.25.205.245 (172.25.205.245) 56(84) bytes of data.
64 bytes from 172.25.205.245: icmp_req=1 ttl=64 time=0.191 ms
64 bytes from 172.25.205.245: icmp_req=2 ttl=64 time=0.097 ms
64 bytes from 172.25.205.245: icmp_req=3 ttl=64 time=0.060 ms
^C
--- 172.25.205.245 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.060/0.116/0.191/0.055 ms
root@francois:/etc/apache2/sites-available# ping 172.25.205.246
PING 172.25.205.246 (172.25.205.246) 56(84) bytes of data.
64 bytes from 172.25.205.246: icmp_req=1 ttl=64 time=0.139 ms
64 bytes from 172.25.205.246: icmp_req=2 ttl=64 time=0.053 ms
64 bytes from 172.25.205.246: icmp_req=3 ttl=64 time=0.051 ms
^C
--- 172.25.205.246 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.051/0.081/0.139/0.041 ms
root@francois:/etc/apache2/sites-available#
```

Tester enfin sur le navigateur web l'accès au site www2 et www3 :

Avec l'adresse `www2.xxxxxxxx.xxx`



Avec l'adresse IP du serveur :



Administration Réseau Linux – HTTP Server APACHE

Enfin analyse des requêtes **http** de l'accès au site web :

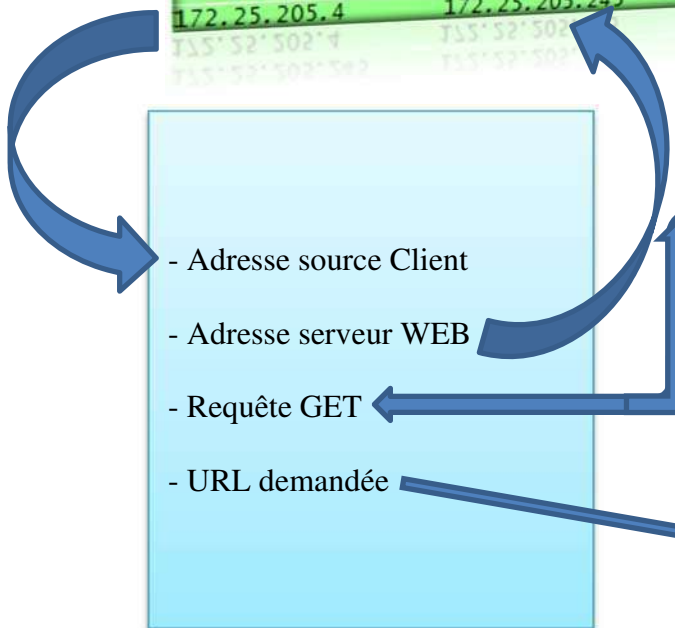
```
96 14.0886860 172.25.205.4 172.25.205.245 TCP 66 52010 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
97 14.0888810 172.25.205.245 172.25.205.4 TCP 66 http > 52010 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=64
98 14.0889850 172.25.205.4 172.25.205.245 TCP 54 52010 > http [ACK] Seq=1 Ack=1 win=65700 Len=0
99 14.0891880 172.25.205.4 172.25.205.245 HTTP 488 GET / HTTP/1.1
100 14.0895390 172.25.205.245 172.25.205.4 TCP 54 http > 52010 [ACK] Seq=1 Ack=435 win=6912 Len=0
101 14.0912760 172.25.205.245 172.25.205.4 HTTP 264 HTTP/1.1 304 Not Modified
102 14.2872280 172.25.205.4 172.25.205.245 TCP 54 52010 > http [ACK] Seq=435 Ack=211 win=65488 Len=0
103 14.4292750 37.252.230.18 172.25.205.4 SSL 102 Continuation Data

[+] Frame 99: 488 bytes on wire (3904 bits), 488 bytes captured (3904 bits) on interface 0
[+] Ethernet II, Src: AsustekC_21:d6:b1 (f4:6d:04:21:d6:b1), Dst: Vmware_4a:0a:77 (00:0c:29:4a:0a:77)
[+] Internet Protocol Version 4, Src: 172.25.205.4 (172.25.205.4), Dst: 172.25.205.245 (172.25.205.245)
[+] Transmission Control Protocol, Src Port: 52010 (52010), Dst Port: http (80), Seq: 1, Ack: 1, Len: 434
  Source port: 52010 (52010)
  Destination port: http (80)
  [Stream index: 1]
  Sequence number: 1 (relative sequence number)
  [Next sequence number: 435 (relative sequence number)]
  Acknowledgment number: 1 (relative ack number)
  Header length: 20 bytes
  [+] Flags: 0x018 (PSH, ACK)
  window size value: 16425
  [Calculated window size: 65700]
  [Window size scaling factor: 4]
  [checksum: 0x8482 [validation disabled]]
  [SEQ/ACK analysis]
[+] Hypertext Transfer Protocol
[+] GET / HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
  [Message: GET / HTTP/1.1\r\n]
  [Severity level: Chat]
  [Group: Sequence]
  Request Method: GET
  Request URI: /
  Request Version: HTTP/1.1
  Host: ww2.grp5.info-msj.net\r\n
  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:19.0) Gecko/20100101 Firefox/19.0\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
  Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
  If-Modified-Since: Tue, 12 Mar 2013 14:32:36 GMT\r\n
  If-None-Match: "30283-7a-4d7bb29252900"\r\n
  Cache-Control: max-age=0\r\n
  \r\n
  [Full request URI: http://ww2.grp5.info-msj.net/]

```

```
172.25.205.4 172.25.205.245 TCP 66 52010 > http [SYN]
172.25.205.245 172.25.205.4 TCP 66 http > 52010 [SYN,
172.25.205.4 172.25.205.245 TCP 54 52010 > http [ACK]
172.25.205.4 172.25.205.245 HTTP 488 GET / HTTP/1.1
172.25.205.245 172.25.205.4 TCP 54 http > 52010 [ACK]
172.25.205.245 172.25.205.4 HTTP 264 HTTP/1.1 304 Not M
172.25.205.4 172.25.205.245 TCP 54 52010 > http [ACK]
172.25.205.4 172.25.205.245 TCP 54 52010 > http [ACK]

```



```
[+] Hypertext Transfer Protocol
[+] GET / HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
  [Message: GET / HTTP/1.1\r\n]
  [Severity level: Chat]
  [Group: Sequence]
  Request Method: GET
  Request URI: /
  Request Version: HTTP/1.1
  Host: ww2.grp5.info-msj.net\r\n
  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:
  Accept: text/html,application/xhtml+xml,application
  Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.
  Accept-Encoding: gzip, deflate\r\n
  Connection: keep-alive\r\n
  If-Modified-Since: Tue, 12 Mar 2013 14:32:36 GMT\r\
  If-None-Match: "30283-7a-4d7bb29252900"\r\n
  Cache-Control: max-age=0\r\n
  \r\n
  [Full request URI: http://ww2.grp5.info-msj.net/]

```

```

No.    Time    Source                Destination            Protocol Length  Info
 9 14.6587400 172.25.205.4          172.25.205.245        TCP      66      52017 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
10 14.6588130 vmware_4a:0a:77      AsustekC_21:d6:b1     ARP      42      172.25.205.244 is at 00:0c:29:4a:0a:77
11 14.6588210 172.25.205.4          172.25.205.244        DNS      82      Standard query 0xa3a3 A ww2.grp5.info-msj.net
12 14.6589000 172.25.205.245       172.25.205.4          TCP      66      http > 52017 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=64
13 14.6589540 172.25.205.4          172.25.205.245        TCP      54      52017 > http [ACK] Seq=1 Ack=1 Win=65700 Len=0
14 14.6590180 172.25.205.4          172.25.205.245        HTTP     371     GET / HTTP/1.1
15 14.6591210 172.25.205.245       172.25.205.4          TCP      54      http > 52017 [ACK] Seq=1 Ack=318 Win=6912 Len=0
16 14.6595300 172.25.205.244       172.25.205.4          DNS     137     Standard query response 0xa3a3 A 172.25.205.245
17 14.6600560 172.25.205.245       172.25.205.4          HTTP     524     HTTP/1.1 200 OK (text/html)
18 14.8663830 172.25.205.4          172.25.205.245        TCP      54      52017 > http [ACK] Seq=318 Ack=471 Win=65228 Len=0
19 17.4831450 fe80::d04e:c2e2:718ff02::1:2 DINKPv6 162     solicit xid: 0x9713dd cid: 000100011850c9b10013f71aaadc

[Frame 17: 524 bytes on wire (4192 bits), 524 bytes captured (4192 bits) on interface 0]
[Ethernet II, Src: vmware_4a:0a:77 (00:0c:29:4a:0a:77), Dst: AsustekC_21:d6:b1 (f4:6d:04:21:d6:b1)]
[Internet Protocol Version 4, Src: 172.25.205.245 (172.25.205.245), Dst: 172.25.205.4 (172.25.205.4)]
[Transmission Control Protocol, Src Port: http (80), Dst Port: http (80), Seq: 1, Ack: 318, Len: 470]
[Hypertext Transfer Protocol]
  [HTTP/1.1 200 OK\r\n
    Date: Tue, 12 Mar 2013 14:55:56 GMT\r\n
    Server: Apache/2.2.16 (Debian)\r\n
    Last-Modified: Tue, 12 Mar 2013 14:32:36 GMT\r\n
    ETag: "30283-7a-4d7bb29252900"\r\n
    Accept-Ranges: bytes\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Content-Length: 132\r\n
    [Content length: 132]
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
    Content-encoded entity body (gzip): 132 bytes -> 122 bytes
  ]
[Line-based text data: text/html]
<html><body><h1>Bienvenue sur ce site ww2 de grp5.info-msj.net<h1>\n
<meta HTTP-EQUIV="REFRESH" CONTENT="3;\n
</body></html>\n
  
```

Réponse du serveur **HTTP/1.1 200 OK**

172.25.205.245 172.25.205.4 HTTP 524 HTTP/1.1 200 OK (text/html)

Contenu de la page **index.html**

```

[Line-based text data: text/html]
<html><body><h1>Bienvenue sur ce site ww2 de grp5.info-msj.net<h1>\n
<meta HTTP-EQUIV="REFRESH" CONTENT="3;\n
</body></html>\n
  
```