

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Administration Réseaux

Pr. Abdelhak LAKHOUAJA

Département d'Informatique
Faculté des Sciences
Université Mohammed Premier
Oujda

SMI-S6

Année universitaire : 2017/2018

- Page web du cours :
<http://lakhouaja.oujda-nlp-team.net/teaching/bachelor-level/administration-reseaux/>
- Page web personnelle :
<http://lakhouaja.oujda-nlp-team.net/>

Chapitre 1

Introduction

- **ifconfig/ipconfig** : pour configurer une interface réseau ou afficher les informations concernant les interfaces réseaux (commandes déjà utilisées en S5).

- **ifconfig/ipconfig** : pour configurer une interface réseau ou afficher les informations concernant les interfaces réseaux (commandes déjà utilisées en S5).
- **route** : pour afficher/modifier la table de routage.

- **ifconfig/ipconfig** : pour configurer une interface réseau ou afficher les informations concernant les interfaces réseaux (commandes déjà utilisées en S5).
- **route** : pour afficher/modifier la table de routage.
- **ping** : pour tester la connexion entre deux machines.

- **ifconfig/ipconfig** : pour configurer une interface réseau ou afficher les informations concernant les interfaces réseaux (commandes déjà utilisées en S5).
- **route** : pour afficher/modifier la table de routage.
- **ping** : pour tester la connexion entre deux machines.
- **host** : c'est un utilitaire simple de conversion de noms DNS en adresse IP (et vice versa) :

Exemple :

```
alkhalil: $ host www.ump.ma  
www.ump.ma has address 196.200.156.5
```

- **nslookup** : disponible sous Linux et sous Windows, comme la commande **host**, cette commande permet d'afficher l'adresse IP d'un nom DNS.

Exemple :

```
alkhalil:~$ nslookup www.ump.ma
Server: 127.0.1.1
Address: 127.0.1.1#53
```

```
Non-authoritative answer:
Name: www.ump.ma
Address: 196.200.156.5
```


- **nslookup** : disponible sous Linux et sous Windows, comme la commande **host**, cette commande permet d'afficher l'adresse IP d'un nom DNS.

Exemple :

```
alkhalil:~$ nslookup www.ump.ma
Server: 127.0.1.1
Address: 127.0.1.1#53
```

```
Non-authoritative answer:
Name: www.ump.ma
Address: 196.200.156.5
```

- **netstat**
- **tcpdump/wireshark**

netstat -s | -a | -r | -n

Netstat fournit des statistiques sur les :

- paquets émis ou reçus
- erreurs
- collisions
- protocoles utilisés

- le nom et l'état des interfaces du système
`netstat -i`

- le nom et l'état des interfaces du système

```
netstat -i
```

- le contenu de la table de routage

```
netstat -r | n
```

- le nom et l'état des interfaces du système

```
netstat -i
```

- le contenu de la table de routage

```
netstat -r | n
```

- ainsi que l'état de tous les sockets

```
netstat -a
```

Wireshark ¹ est un outil d'analyse des réseaux qui permet de capturer et d'analyser les paquets qui circulent sur le réseau. Il peut être utilisé pour capturer les paquets qui circulent sur une interface ou pour visualiser le contenu d'un fichier qui des paquets capturés par un autre utilitaire tel que **tcpdump**. Il est multi-plateforme, il fonctionne sous Linux, Windows, MacOS, ...

1. Site officiel : <https://www.wireshark.org/>

interface de wireshark

exam_capture [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Enregistrer

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	6e:5f:98:37:0c:07	2e:18:cc:d4:8c:e7	ARP	42	Who has 192.168.1.2? Tell
2	0.000051	2e:18:cc:d4:8c:e7	6e:5f:98:37:0c:07	ARP	42	192.168.1.2 is at 2e:18:cc:
3	3.858131	192.168.1.2	192.168.1.1	TCP	74	54490 > http [SYN] Seq=0 Wi
4	3.858191	192.168.1.1	192.168.1.2	TCP	74	http > 54490 [SYN, ACK] Seq
5	3.858279	192.168.1.2	192.168.1.1	TCP	66	54490 > http [ACK] Seq=1 Ac
6	3.858509	192.168.1.2	192.168.1.1	HTTP	160	GET /xab HTTP/1.0 [Packet s

▶ Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)

▶ Ethernet II, Src: 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07), Dst: 2e:18:cc:d4:8c:e7 (2e:18:cc:d4:8c:e7)

▶ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)

▶ Transmission Control Protocol, Src Port: http (80), Dst Port: 54490 (54490), Seq: 1, Ack: 95, Len: 0

0000 2e 18 cc d4 8c e7 6e 5f 98 37 0c 07 08 00 45 00n_.7....E.
0010 00 34 e6 64 40 00 40 06 d1 0b c0 a8 01 01 c0 a8 .4.d@.@.
0020 01 02 00 50 d4 da 98 6a d0 d0 98 60 f0 4f 80 10 ...P...j ...`.0..
0030 0b 50 b3 06 00 00 01 01 08 02 00 08 36 b4 00 08 P 6

File: "/home/lakhouaja/SMI_S5/TP2/... Packets: 22 · Display... Profile: Default

Les colonnes se présentent comme suit :

No. : représente le numéro du paquet ;

Time : représente le temps de capture du paquet ;

Source : représente l'adresse IP ou MAC de la source ;

Destination : représente l'adresse IP ou MAC destination ;

Protocol : représente le type du protocole capturé ;

Length : représente la taille du paquet (en octets) ;

Info : représente une brève information concernant le paquet.

sous Linux, comme pour la commande **tcpdump**, **wireshark** ne peut pas être utilisé pour capturer des données en mode simple utilisateur. Pour capturer des données il faut passer en mode administrateur.

Description de l'interface

L'interface est découpé en trois zones :

1 supérieure : contient l'ensemble des paquets capturés

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	6e:5f:98:37:0c:07	2e:18:cc:d4:8c:e7	ARP	42	Who has 192.168.1.2? Tell
2	0.000051	2e:18:cc:d4:8c:e7	6e:5f:98:37:0c:07	ARP	42	192.168.1.2 is at 2e:18:cc:
3	3.858131	192.168.1.2	192.168.1.1	TCP	74	54490 > http [SYN] Seq=0 Wi
4	3.858191	192.168.1.1	192.168.1.2	TCP	74	http > 54490 [SYN, ACK] Seq
5	3.858279	192.168.1.2	192.168.1.1	TCP	66	54490 > http [ACK] Seq=1 Acl
6	3.858509	192.168.1.2	192.168.1.1	HTTP	160	GET /xab HTTP/1.0 [Packet s

Description de l'interface

L'interface est découpée en trois zones :

- 1 supérieure : contient l'ensemble des paquets capturés

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	6e:5f:98:37:0c:07	2e:18:cc:d4:8c:e7	ARP	42	Who has 192.168.1.2? Tell
2	0.000051	2e:18:cc:d4:8c:e7	6e:5f:98:37:0c:07	ARP	42	192.168.1.2 is at 2e:18:cc:
3	3.858131	192.168.1.2	192.168.1.1	TCP	74	54490 > http [SYN] Seq=0 Win
4	3.858191	192.168.1.1	192.168.1.2	TCP	74	http > 54490 [SYN, ACK] Seq=
5	3.858279	192.168.1.2	192.168.1.1	TCP	66	54490 > http [ACK] Seq=1 Ack
6	3.858509	192.168.1.2	192.168.1.1	HTTP	160	GET /xab HTTP/1.0 [Packet s

- 2 centrale : affiche les détails d'un paquet sélectionné sous forme de couches

```
▶ Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▶ Ethernet II, Src: 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07), Dst: 2e:18:cc:d4:8c:e7 (2e:18:cc:d4:8c:e7)
▶ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
▶ Transmission Control Protocol, Src Port: http (80), Dst Port: 54490 (54490), Seq: 1, Ack: 95, Len: 0
```

Description de l'interface

L'interface est découpée en trois zones :

- 1 supérieure : contient l'ensemble des paquets capturés

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	6e:5f:98:37:0c:07	2e:18:cc:d4:8c:e7	ARP	42	Who has 192.168.1.2? Tell
2	0.000051	2e:18:cc:d4:8c:e7	6e:5f:98:37:0c:07	ARP	42	192.168.1.2 is at 2e:18:cc:
3	3.858131	192.168.1.2	192.168.1.1	TCP	74	54490 > http [SYN] Seq=0 Win
4	3.858191	192.168.1.1	192.168.1.2	TCP	74	http > 54490 [SYN, ACK] Seq=
5	3.858279	192.168.1.2	192.168.1.1	TCP	66	54490 > http [ACK] Seq=1 Acc
6	3.858509	192.168.1.2	192.168.1.1	HTTP	160	GET /xab HTTP/1.0 [Packet s

- 2 centrale : affiche les détails d'un paquet sélectionné sous forme de couches

```
▶ Frame 7: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▶ Ethernet II, Src: 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07), Dst: 2e:18:cc:d4:8c:e7 (2e:18:cc:d4:8c:e7)
▶ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.2 (192.168.1.2)
▶ Transmission Control Protocol, Src Port: http (80), Dst Port: 54490 (54490), Seq: 1, Ack: 95, Len: 0
```

- 3 Zone inférieure : présente le paquet sous format octale et ASCII

```
0000  2e 18 cc d4 8c e7 6e 5f 98 37 0c 07 08 00 45 00  .....n_ .7....E.
0010  00 34 e6 64 40 00 40 06 d1 0b c0 a8 01 01 c0 a8  .4.d@.@. ....
0020  01 02 00 50 d4 da 98 6a d0 d0 98 60 f0 4f 80 10  ...P...j ...`.0..
0030  0b 50 b3 06 00 00 01 01 08 03 00 08 36 b4 00 08  P                6
```

Il est possible de ne pas afficher tous les paquets en les filtrant. Par exemple, on peut afficher juste les paquets **http**, en tapant **http** dans la zone **Filter** :. Il est possible aussi d'utiliser des expressions.

Exemples de filtres

Filtre/expression	Signification
tcp	afficher seulement les paquets TCP

Exemples de filtres

Filtre/expression	Signification
tcp	afficher seulement les paquets TCP
ip.src==192.168.1.2	afficher seulement les paquets qui sortent de 192.168.1.2

Exemples de filtres

Filtre/expression	Signification
tcp	afficher seulement les paquets TCP
ip.src==192.168.1.2	afficher seulement les paquets qui sortent de 192.168.1.2
ip.dst==192.168.1.1 && http	afficher les paquets HTTP qui partent vers 192.168.1.1

Exemples de filtres

Filtre/expression	Signification
tcp	afficher seulement les paquets TCP
ip.src==192.168.1.2	afficher seulement les paquets qui sortent de 192.168.1.2
ip.dst==192.168.1.1 && http	afficher les paquets HTTP qui partent vers 192.168.1.1
ip && !udp	afficher les paquets IP mais n'afficher pas les paquets UDP

Chapitre 2

Serveurs web

Protocole HTTP : (HyperText Transfer Protocol)

- Le protocole HTTP est le protocole de transport de données le plus utilisé sur Internet depuis 1990.
- La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML).
- La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message.
- Le but du protocole HTTP est de permettre un transfert des fichiers entre un navigateur (le client) et un serveur Web.

Pourquoi utiliser HTTP ?

HTTP est devenu le protocole de communication de l'Internet. Il :

- est disponible sur toutes les plates-formes ;
- est simple. Ne requière que peu de support pour fonctionner correctement ;
- offre un niveau de sécurité simple et efficace ;
- est utilisable à travers des pare-feu.

HTTP fonctionne selon le schéma classique client/serveur :

- connexion du client vers le serveur ;
- demande d'une information via une **méthode** ;
- renvoi de l'**information** ou une **erreur** ;
- déconnexion.

1xx : Information

2xx : Succès (par exemple : 200 ok).

3xx : Redirection.

4xx : Erreurs (par exemple : 404 Not Found).

5xx : Erreurs venant du serveur HTTP (par exemple : 501 Not Implemented).

On verra en TP plus de codes de retours.

Les méthodes HTTP sont les suivantes :

- GET** : demande de la ressource située à l'URL spécifiée ;
- HEAD** : demande de l'en-tête de la ressource située à l'URL spécifiée ;
- POST** : envoi de données au programme situé à l'URL spécifiée ;
- PUT** : envoi de données à l'URL spécifiée ;
- DELETE** : suppression de la ressource située à l'URL spécifiée.

Requête HTTP

Une requête HTTP est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

- une ligne de requête précise la méthode qui doit être appliquée, et la version du protocole utilisée. La ligne comprend trois éléments séparés par un espace :
 - la méthode ;
 - l'URL ;
 - la version du protocole utilisé par le client (généralement HTTP/1.0) ;

Exemple :

GET / HTTP/1.0

Requête HTTP

Une requête HTTP est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

- une ligne de requête précise la méthode qui doit être appliquée, et la version du protocole utilisée. La ligne comprend trois éléments séparés par un espace :
 - la méthode ;
 - l'URL ;
 - la version du protocole utilisé par le client (généralement HTTP/1.0) ;

Exemple :

GET / HTTP/1.0

- les champs d'en-tête de la requête : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation, ...).

Réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

- Une ligne de statut : c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. La ligne comprend trois éléments devant être séparés par un espace :
 - La version du protocole utilisé
 - Le code de statut
 - La signification du code

Réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

- Une ligne de statut : c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. La ligne comprend trois éléments devant être séparés par un espace :
 - La version du protocole utilisé
 - Le code de statut
 - La signification du code
- Les champs d'en-tête de la réponse : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête

Réponse HTTP

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

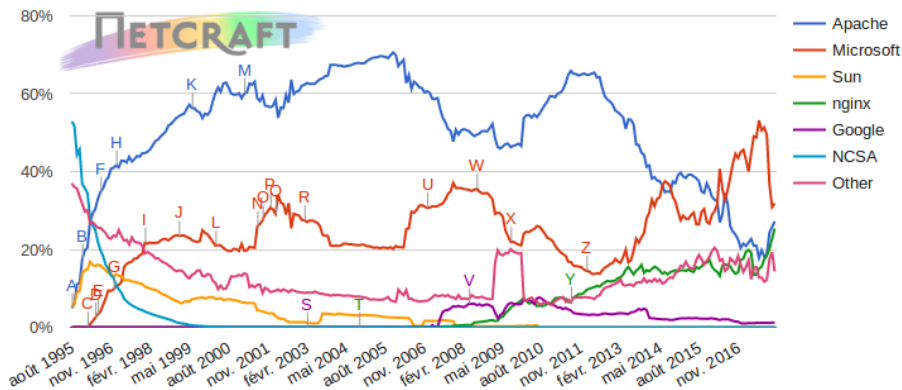
- Une ligne de statut : c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. La ligne comprend trois éléments devant être séparés par un espace :
 - La version du protocole utilisé
 - Le code de statut
 - La signification du code
- Les champs d'en-tête de la réponse : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom qualifiant le type d'en-tête, suivi de deux points (:) et de la valeur de l'en-tête
- Le corps de la réponse : il contient le document demandé

Principaux serveurs :

- Apache
- Microsoft : Internet Information Server (IIS)
- nginx
- gws (Google Web Server)

Serveurs Web en ligne 2018

D'après Netcraft², le serveur apache est le deuxième parmi les serveurs web les plus utilisés.



2. <http://survey.netcraft.com>

Serveurs Web en ligne 2018

Le tableau suivant, montre la part du marché des principaux fournisseurs des serveurs web (selon netcraft).

<i>Développement</i>	12/2017	<i>Pourcentage</i>	01/2018	<i>Pourcentage</i>	<i>Changement</i>
Microsoft	536 million	30,89%	575 million	31,85%	0.96
Apache	446 million	25,74%	491 million	27,21%	1.47
nginx	396 million	22,83%	458 million	25,39%	2.56
Google	21 million	1,23%	22 million	1,20%	-0.03

Serveur web apache

Comme on l'a vu, apache³ est parmi les serveurs web les plus populaires. Il est robuste et extensible. Il est distribué sous une licence "Open source" (Licence Apache).

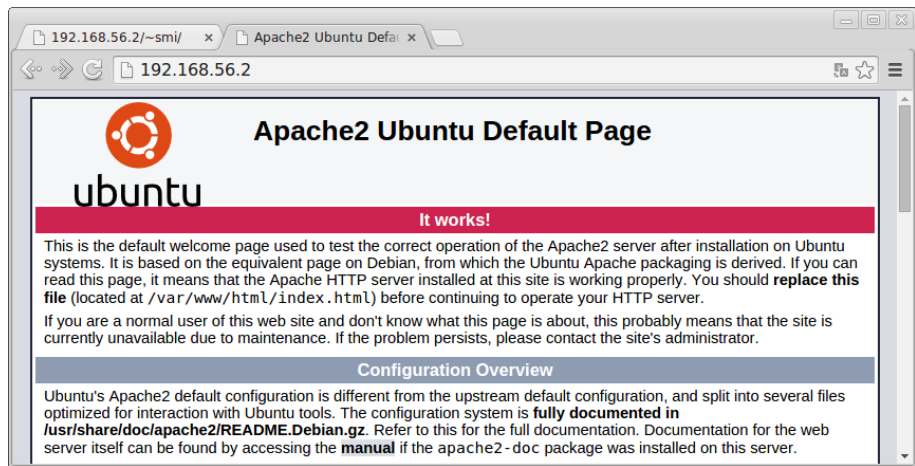
Il est disponible sur plusieurs plateformes (Linux, windows, ...)

Une installation minimale peut être faite en ligne de commande de la façon suivante :

```
#sudo apt-get install apache2
```

Vérification de l'installation


Pour vérifier l'installation, il suffit d'utiliser un navigateur web.



The screenshot shows a web browser window with the address bar set to `192.168.56.2`. The page content includes the Ubuntu logo, the title "Apache2 Ubuntu Default Page", a red banner with the text "It works!", and a paragraph of text explaining that the Apache HTTP server is working properly. Below this is a section titled "Configuration Overview" with further details about the default configuration.

192.168.56.2/~smi/ x Apache2 Ubuntu Defai x

192.168.56.2



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** `/usr/share/doc/apache2/README.Debian.gz`. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire **/etc/apache2** :

`apache2.conf` : fichier de configuration principale.

Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire **/etc/apache2** :

apache2.conf : fichier de configuration principale.

envvars : contient les variables d'environnement propres à apache.

Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire **/etc/apache2** :

apache2.conf : fichier de configuration principale.

envvars : contient les variables d'environnement propres à apache.

conf-available/ : contient des fichiers de configuration additionnels disponibles.

Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire **/etc/apache2** :

apache2.conf : fichier de configuration principale.

envvars : contient les variables d'environnement propres à apache.

conf-available/ : contient des fichiers de configuration additionnels disponibles.

conf-enabled/ : contient des fichiers de configuration activés. Ils sont utilisés dans **apache2.conf** par la ligne :

```
IncludeOptional conf-enabled/*.conf
```

Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire **/etc/apache2** :

apache2.conf : fichier de configuration principale.

envvars : contient les variables d'environnement propres à apache.

conf-available/ : contient des fichiers de configuration additionnels disponibles.

conf-enabled/ : contient des fichiers de configuration activés. Ils sont utilisés dans **apache2.conf** par la ligne :

```
IncludeOptional conf-enabled/*.conf
```

ports.conf : directives de configuration pour les ports et les adresses IP d'écoutes.

Fichiers et répertoires de configuration

`mods-available/` : contient une série de fichiers **.load** et **.conf**. Un fichier **.load** contient les paramètres de configuration nécessaires pour charger un module en question. Le fichier **.conf** correspondant, les paramètres de configuration nécessaires pour utiliser le module en question.

Fichiers et répertoires de configuration

mods-available/ : contient une série de fichiers **.load** et **.conf**. Un fichier **.load** contient les paramètres de configuration nécessaires pour charger un module en question. Le fichier **.conf** correspondant, les paramètres de configuration nécessaires pour utiliser le module en question.

mods-enabled/ : pour utiliser un module (activer), il faut mettre un lien symbolique vers le fichier **.load** (et **.conf**, s'il existe) du module associé dans le dossier **mods-available**.

Fichiers et répertoires de configuration

mods-available/ : contient une série de fichiers **.load** et **.conf**. Un fichier **.load** contient les paramètres de configuration nécessaires pour charger un module en question. Le fichier **.conf** correspondant, les paramètres de configuration nécessaires pour utiliser le module en question.

mods-enabled/ : pour utiliser un module (activer), il faut mettre un lien symbolique vers le fichier **.load** (et **.conf**, s'il existe) du module associé dans le dossier **mods-available**.

sites-available/ : même chose que **mods-available/**, mais cette fois pour les sites virtuels. Ce n'est pas obligé d'avoir le même nom pour le site et le fichier.

Fichiers et répertoires de configuration

mods-available/ : contient une série de fichiers **.load** et **.conf**. Un fichier **.load** contient les paramètres de configuration nécessaires pour charger un module en question. Le fichier **.conf** correspondant, les paramètres de configuration nécessaires pour utiliser le module en question.

mods-enabled/ : pour utiliser un module (activer), il faut mettre un lien symbolique vers le fichier **.load** (et **.conf**, s'il existe) du module associé dans le dossier **mods-available**.

sites-available/ : même chose que **mods-available/**, mais cette fois pour les sites virtuels. Ce n'est pas obligé d'avoir le même nom pour le site et le fichier.

sites-enabled/ : même chose que **mods-enabled/**.

magic : instructions pour déterminer le type **MIME** d'un fichier (**M**ultipurpose **I**nternet **M**ail **E**xtensions - Extensions Multi-usages de la Messagerie par Internet). Par exemple text/html et image/gif.

Remarque :

Par défaut, un seul serveur est disponible (le serveur par défaut). Il est disponible dans **apache2.conf** par la ligne :

```
IncludeOptional sites-enabled/*.conf
```

Activer/désactiver un module

Les commandes **a2enmod** et **a2dismod** sont disponibles pour activer ou désactiver un module.

Exemple : pages web personnelles

Pour permettre aux utilisateurs d'avoir leurs propres pages web disponibles via un lien de type :

```
http://NomSite/~utilisateur
```

```
http://localhost/~smi
```

On tape la commande

```
#a2enmod userdir
```

Il faut ensuite redémarrer apache en tapant la commande :

```
#service apache2 restart
```


Exemple d'une page personnelle :

Dans le répertoire personnelle de l'utilisateur **smi**, il faut créer le répertoire **public_html** avec les droits **-rwxr-xr-x** et mettre dedans le fichier **index.html**, avec les droits **-rwxr--r--**.



Les commandes **a2ensite** et **a2dissite** sont disponibles pour activer ou désactiver un site. On verra leurs utilisation dans les sections suivantes.

Configuration de base

Avant de commencer la configuration, il faut faire une sauvegarde des fichiers que vous voulez modifier. Par exemple :

```
#cp apache2.conf apache2.conf.save
```

- Port à écouter (ports.conf) :

```
Listen 80
```

- Emplacement par défaut des pages html : **/var/www/html**
mettre les fichiers concernant le site web dans ce répertoire.

- Pages par défaut (mods-enabled/dir.conf) :

```
DirectoryIndex index.html index.cgi index.pl  
index.php index.xhtml index.htm
```

Sites virtuelles

Apache permet de gérer plusieurs sites web. Chaque site est appelé serveur virtuel et possède sa propre configuration.

Il y a deux types de serveurs virtuels :

Apache permet de gérer plusieurs sites web. Chaque site est appelé serveur virtuel et possède sa propre configuration.

Il y a deux types de serveurs virtuels :

- 1 Serveurs par nom ; les mêmes sites utilisent la même adresse IP.
Par exemple :

IP	Nom
192.168.56.2	smi.ump.ma
192.168.56.2	sma.ump.ma

Apache permet de gérer plusieurs sites web. Chaque site est appelé serveur virtuel et possède sa propre configuration.

Il y a deux types de serveurs virtuels :

- 1 Serveurs par nom ; les mêmes sites utilisent la même adresse IP.
Par exemple :

IP	Nom
192.168.56.2	smi.ump.ma
192.168.56.2	sma.ump.ma

- 2 Serveurs par adresse IP ; chaque site utilise sa propre adresse IP.
Par exemple :

IP	Nom
192.168.56.2	smi.ump.ma
192.168.56.10	sma.ump.ma

Configuration de deux sites virtuels par nom

Dans cette exemple, nous allons configurer deux sites virtuels, le premier **smi.ump.ma** et le deuxième **sma.ump.ma**. Les deux sites utilisent la même adresse IP **192.168.56.2**.

Il faut déclarer les deux noms dans le fichier **/etc/hosts** :

192.168.56.2	smi.ump.ma	www.smi.ump.ma
192.168.56.2	sma.ump.ma	www.sma.ump.ma

On pourra utiliser un serveur DNS pour déclarer les noms (voir chapitre concernant le DNS).

Configuration de deux sites virtuels par nom

Il faut créer les répertoires **smi** et **sma** associés dans **/var/www/html** :

```
# mkdir /var/www/html/smi  
# mkdir /var/www/html/sma
```


Configuration de deux sites virtuels par nom

Il faut créer les répertoires **smi** et **sma** associés dans **/var/www/html** :

```
# mkdir /var/www/html/smi
# mkdir /var/www/html/sma
```

Dans **/etc/apache2/sites-available/**, il faut créer deux fichiers :
smi.conf et **sma.conf**

Contenu du fichier smi.conf

```
<VirtualHost *:80>
    DocumentRoot /var/www/html/smi
    ServerName smi.ump.ma
    ServerAlias www.smi.ump.ma
</VirtualHost>
```

Avec :

DocumentRoot : emplacement par défaut des pages html ;

ServerName : nom du serveur virtuel ;

ServerAlias : autre nom (alias) du serveur virtuel.

Contenu du fichier sma.conf

```
<VirtualHost *:80>  
    DocumentRoot /var/www/html/sma  
    ServerName sma.ump.ma  
    ServerAlias www.sma.ump.ma  
</VirtualHost>
```

Activation des deux sites

Il faut activer les deux sites en tapant les commandes :

```
# a2ensite smi  
# a2ensite sma
```

Activation des deux sites

Il faut activer les deux sites en tapant les commandes :

```
# a2ensite smi  
# a2ensite sma
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Activation des deux sites

Il faut activer les deux sites en tapant les commandes :

```
# a2ensite smi  
# a2ensite sma
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Les deux sites seront accessibles via les liens :

<http://smi.ump.ma> ou <http://www.smi.ump.ma>

<http://sma.ump.ma> ou <http://www.sma.ump.ma>

Dans cet exemple, nous allons configurer un nouveau site virtuel **smp.ump.ma**, qui utilise une adresse IP différente.

Dans cet exemple, la machine doit être munie, soit de plusieurs interfaces réseaux soit de plusieurs adresses IP associées à la même interface réseau (on parle d'IP aliasing).

Pour affecter une seconde adresse IP à une interface réseau, il faut exécuter la commande :

```
# ifconfig eth0:0 192.168.56.10 up
```

Remplacez eth0 par une autre interface (par exemple eth1).

L'interface dispose, maintenant, de deux adresses distinctes :

- Adresse IP :192.168.56.2
- Alias IP : 192.168.56.10

A vérifier avec la commande :

```
# ifconfig
```


Pour rendre la configuration permanente, il faut ajouter les lignes suivantes au fichier **/etc/network/interfaces** :

```
auto eth0:0
iface eth0:0 inet static
address 192.168.56.10
netmask 255.255.255.0
```

Remarque : on peut ajouter autant d'interfaces qu'on veut (eth0:1, eth0:2 ...).

Configuration du site virtuel

Il faut ajouter au fichier **/etc/hosts**, les lignes suivantes :

```
192.168.56.10 smp.ump.ma  
192.168.56.10 www.smp.ump.ma
```

Il faut créer le répertoire **smp** associé dans **/var/www/html** :

```
# mkdir /var/www/html/smp
```

Dans **/etc/apache2/sites-available/**, il faut créer le fichier : **smp.conf**

Contenu du fichier smp.conf

```
<VirtualHost 192.168.56.10:80>  
    DocumentRoot /var/www/html/smp  
    ServerName smp.ump.ma  
    ServerAlias www.smp.ump.ma  
</VirtualHost>
```

Activation du nouveau site

Il faut activer le site en tapant la commande :

```
# a2ensite smp
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Le nouveau site sera accessible via les liens :

<http://smp.ump.ma> OU <http://www.smp.ump.ma>

Sécuriser apache

Apache est très modulaire. Dans le chapitre suivant, on verra un module concernant **php**. Dans cette section, on va utiliser un module important dans l'aspect sécurité. Le module **mod_ssl** ajoute la possibilité de crypter les communications entre le client et le serveur.

Le mode **mod_ssl** se trouve dans le package **apache2-common**. Pour l'activer, il faut taper la commande :

```
sudo a2enmod ssl
```

suivie de la commande

```
sudo service apache2 restart
```

Après l'activation, il faut utiliser le préfixe **https://** devant l'adresse du serveur dans la barre du navigateur (par exemple : <https://192.168.56.2/>).

On a vu un exemple d'utilisation avec **wireshark** (<https://www.wireshark.org>).