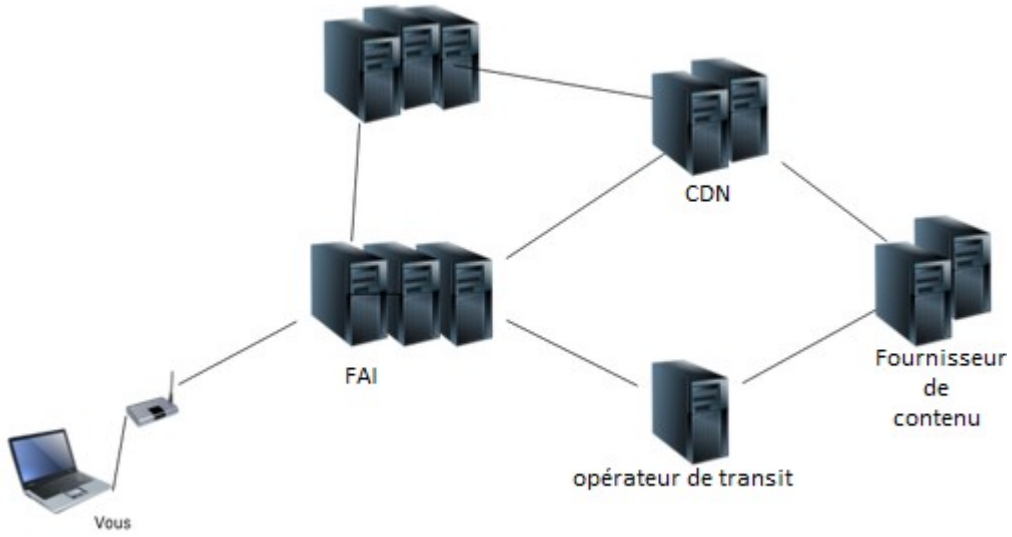
Serveur Apache

Sommaire

Fonctionnement global d'Internet

C'est un réseau mondial



FAI : Fournisseur d'Accès à Internet

CDN (Content Delivery Network) : copie et mise à disposition du contenu, permet d'accélérer les temps de chargement

Les opérateurs de transit font le lien entre les différents opérateurs

Les fournisseurs de contenu : Google, Youtube, Twitter, etc.

Il existe plusieurs chemins pour joindre une destination, cela permet à l'Internet d'être plus robuste

Les machines qui calculent les itinéraires s'appellent des routeurs

Les routeurs sont focalisés sur l'opération de routage et ne filtrent pas le contenu

Ils sont en capacité d'adapter le chemin emprunté par les données en fonction de la congestion du réseau

Certains itinéraires possèdent des machines qui bloquent une partie du contenu.

Ces machines, les proxys, sont des équipements de sécurité qui permettent de bloquer du contenu indésirable (filtrage d'URL)

Il peuvent également stocker temporairement une partie du contenu pour accélérer les échanges

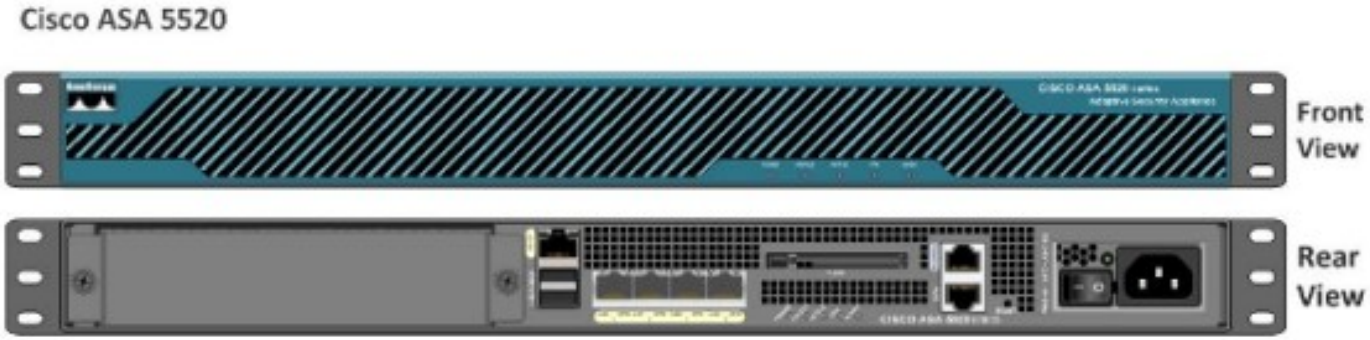
Souvent, le contenu statique est stocké au plus proche de la demande pour être resservi instantanément

Il existe des machines qui sont capables de bloquer des échanges en fonction de la nature ou de la fréquence

Ces équipements de sécurité, les pares-feu, permettent aux serveurs fournissant le contenu de rester joignables

Ils sont responsables de la sécurité des échanges sur Internet grâce à l'application d'une politique de filtrage

Ils existent sous deux formes : matériel et logiciel



Les machines fournissant le contenu s'appellent des serveurs.

Ces machines sont généralement très puissantes et sont raccordées à des réseaux qui permettent d'exploiter pleinement cette puissance de calcul (data center)

Il existe deux types de serveurs : physique et virtuel.

Les serveurs physiques sont devenus tellement puissants que pour exploiter cette puissance, on est obligé de les « découper » en plusieurs serveurs virtuels

vmware®

PROXMOX

Principe d’une communication

Pour établir une communication avec un destinataire, il faut connaître son adresse

Cette adresse **doit être unique** pour éviter les confusions au niveau des routeurs

Il existe deux types d'adresse IP : publique et privée

Les adresses IP privées servent à la communication sur un réseau local (LAN)

Les adresses IP publiques servent à la communication des machines sur Internet

Pour assurer l'unicité, plusieurs organismes s'occupent de la distribution des adresses

Les adresses IP ne sont pas simples à retenir, c'est pourquoi on peut également utiliser des noms pour joindre une machine

Ces noms permettent de désigner des domaines et se louent pour une certaine durée

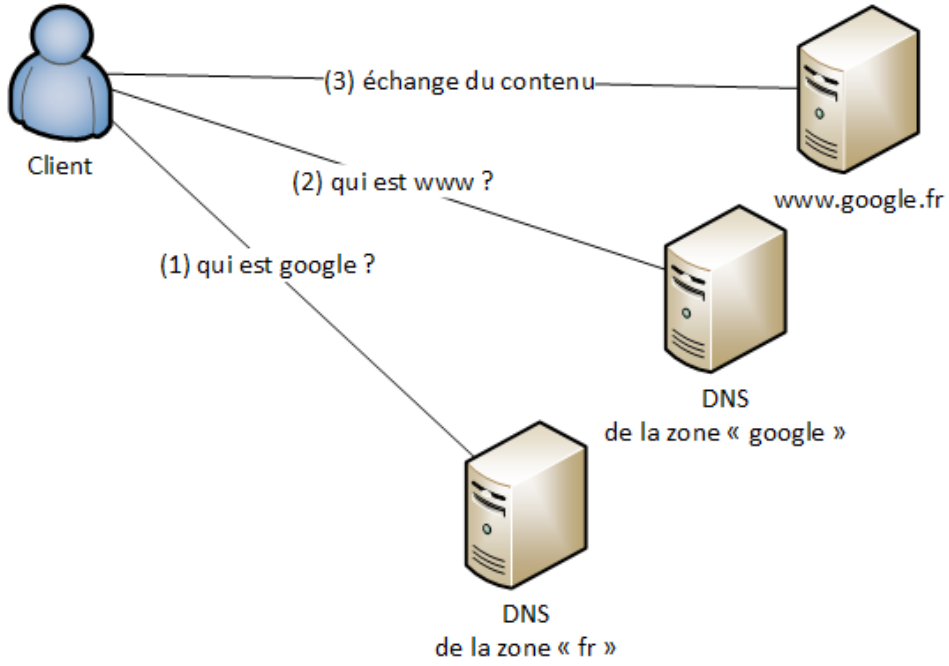
La translation entre nom de domaine et adresse IP est assurée par un certain type de serveur appelé DNS (Domain Name Server)

Les machines et programmes n'utilisent que les adresses IP pour communiquer, ce qui montre l'importance des DNS dans le fonctionnement de l'Internet

Pour assurer l'unicité des noms de domaines, plusieurs organismes s'occupent de la location des noms de domaine

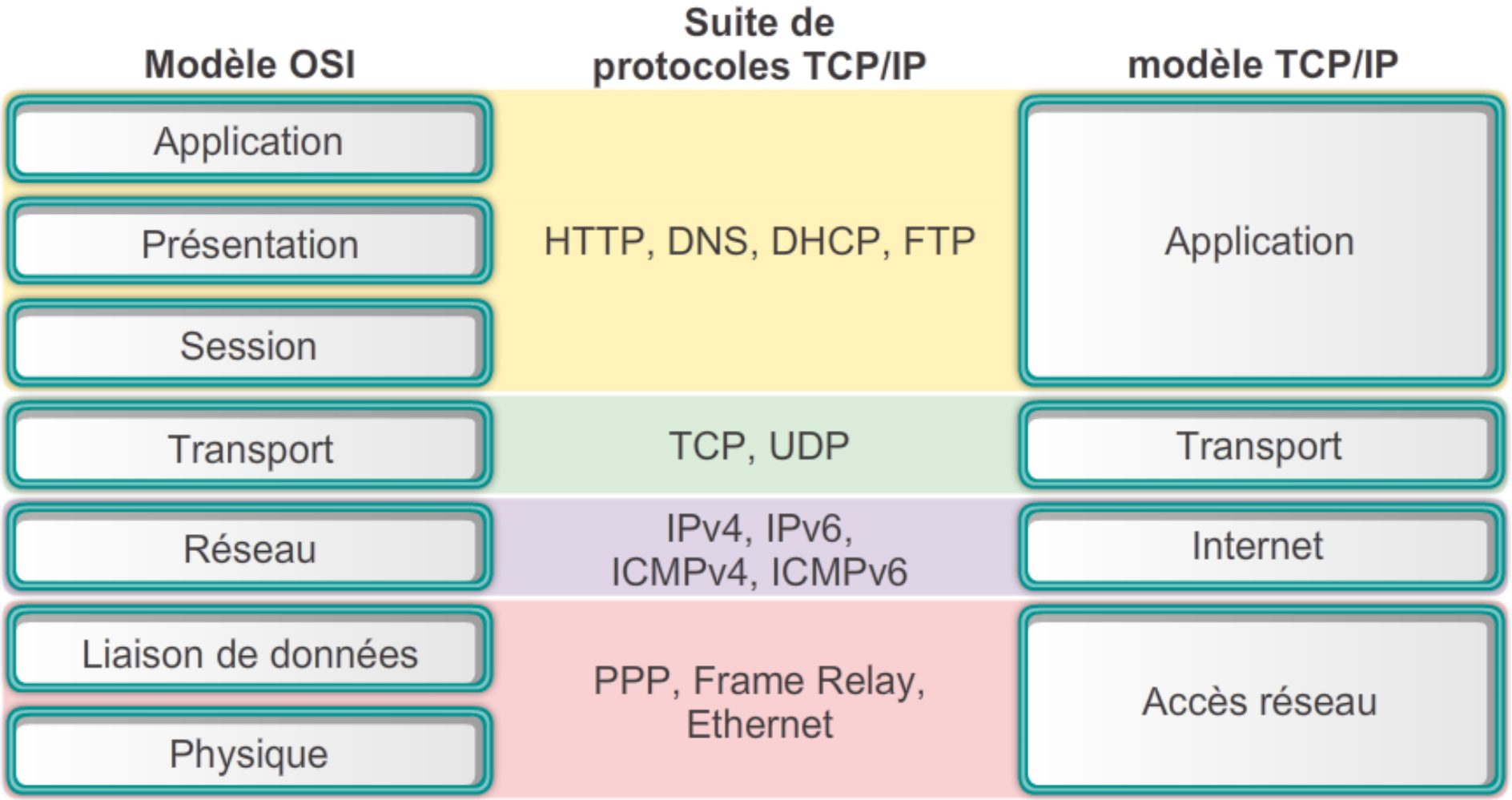
Comme pour le contenu, l'association entre IP et nom de domaine (résolution de nom) peut être mise dans un cache pour accélérer les échanges

En fonction de l'organisme qui gère le nom de domaine, ce cache peut mettre jusqu'à 24h pour se rafraîchir



En cas de lenteur, souvent les DNS sont responsables !

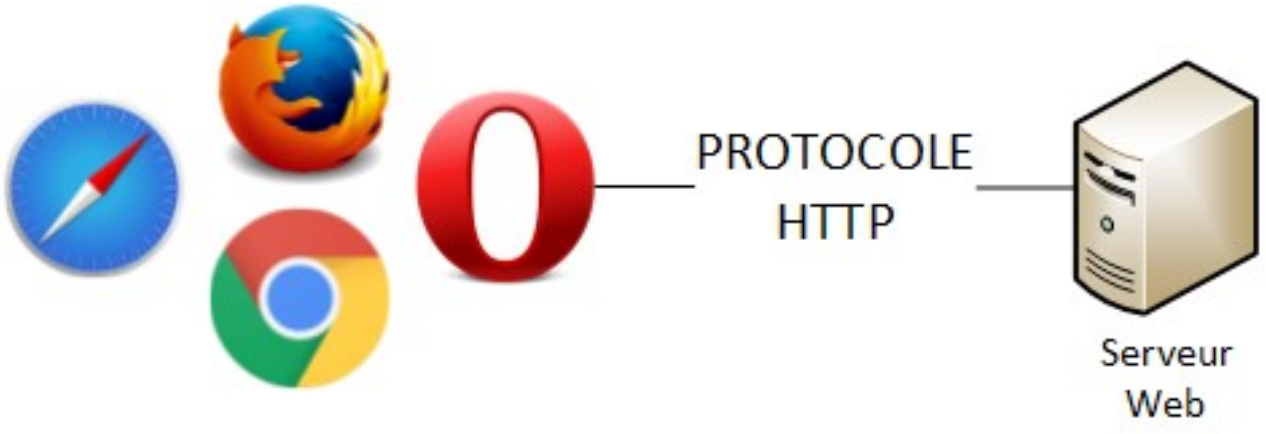
Pour que la communication fonctionne, les différentes machines d'Internet utilisent toutes un modèle : le modèle OSI



Lorsque l'on surfe sur le web, la communication entre les programmes est régie grâce à un protocole : HTTP(S)

En informatique, un protocole est la mise en application d'une norme ou RFC

La communication se fait entre un programme client (le navigateur ou butineur) et un programme serveur (Apache, Nginx, Tomcat, …)



Le protocole HTTP

Un message HTTP (HyperText Transfert Protocol) est décomposé en deux parties : en-tête et corps

Les en-têtes servent à décrire le contenu et permettent de spécifier la langue, l'encodage des caractères, la longueur du message, …

Le corps permet de véhiculer le contenu et peut être au format texte ou binaire (image, musique, …)

HTTP se place au dessus de TCP et fonctionne selon un principe de requête/réponse :

- le client transmet une requête comportant des informations sur le document demandé;
- le serveur renvoie le document si disponible ou, le cas échéant, un message d'erreur.

Le format général d'une URL HTTP est le suivant :

http://<hôte>:<port>/<chemin>?<requête>#<fragment>

<hôte> → nom d'hôte ou adresse IP;
<port> → généralement 80 ou 443;
<chemin> → permet de désigner le fichier désiré;
<requête> → suite de tuples (clés/valeurs) séparés par des '&';
<fragment> → permet d'indiquer une position dans la page.

Encodage d'URL

Les caractères ne pouvant être représentés dans une URL comme ';' '/' '?' '&', …) doivent être échappés.

Afin de rendre "escaped" un caractère (par exemple '&'), il faut remplacer ce caractère par son code ASCII codé en hexadécimal (26 pour '&') et le faire précéder par le caractère '%'.

Le caractère '&' devient donc %26 et un espace ' ' devient %20.

Exemples :
http://www.exemple.com
http://www.exemple.com:80/news/search?cle=valeur
http://www.exemple.com/texte%20avec%20espace/exemple.html

Voici les méthodes définies dans le protocole.

Méthodes	1.0	1.1	Description
Get			Permet de demander un document
Post			Permet de transmettre des données (d'un formulaire par exemple) à l'URL spécifiée dans la requête. L'URL désigne en général un script Perl, PHP…
Head			Permet de ne recevoir que les lignes d'en-tête de la réponse, sans le corps du document
Options			Permet au client de connaître les options du serveur utilisables pour obtenir une ressource
Put			Permet de transmettre au serveur un document à enregistrer à l'URL spécifée dans la requête
Delete			Permet d'effacer la ressource spécifiée
Trace			Permet de signaler au serveur qu'il doit renvoyer la requête telle qu'il la reçue
Connect			Permet de se connecter à un *proxy* ayant la possibilité d'effectuer du *tunneling*

La requête transmise par le client au serveur comprend :

- une ligne de requête (request-line) contenant la méthode utilisée, l'URL du service demandé, la version utilisée de HTTP;
- une ou plusieurs **lignes d'en-têtes**, chacune comportant un nom et une valeur.

La requête peut **optionnellement** contenir une entité (contenu).

L'entité (contenu) est transmise après les lignes d'en-têtes, séparée de la dernière en-tête par un double CRLF (carriage return et linefeed)

Voici un exemple d'en-tête:

- - - - -

GET /index.html HTTP/1.1
Host: www.example.com
Accept: */*
Accept-Language: fr
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:32.0)
Gecko/20100101 Firefox/32.0
Connection: Keep-Alive

- - - - -

Les pointillés ne sont pas transmis, ils permettent de visualiser le double CRLF

Dans cet exemple:

- le client demande le document à l'adresse http://www.example.com/index.html;
- il accepte tous les types de document en retour mais préfère les documents en français;
- utilise un navigateur compatible Mozilla 32.0 sur un système WindowsNT 6.1 (Windows 7)
- signale au serveur qu'il faut garder la connexion TCP ouverte à l'issue de la requête (car il a d'autres requêtes à transmettre).

La réponse transmise par le serveur au client comprend :

- une ligne de statut (status-line) contenant la version de HTTP utilisée et un code d'état;
- une ou plusieurs lignes d'en-têtes, chacune comportant un nom et une valeur;
- Le corps du document retourné (les données HTML ou binaires par exemple).

Une réponse ne contient pas obligatoirement un corps comme par exemple quand il s'agit d'une réponse à une requête HEAD.

Dans ce cas, seule la ligne de statut et les en-têtes sont retournés.

```
HTTP/1.1 200 OK
Date: Mon, 15 Dec 2003 23:48:34 GMT
Server: Apache/1.3.27 (Darwin) PHP/4.3.2 mod_perl/1.26
DAV/1.0.3
Cache-Control: max-age=60
Expires: Mon, 15 Dec 2003 23:49:34 GMT
Last-Modified: Fri, 04 May 2001 00:00:38 GMT
ETag: "26206-5b0-3af1f126"
Accept-Ranges: bytes
Content-Length: 1456
Content-Type: text/html

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
...
```

Dans cet exemple:

- le code 200 nous indique que le document demandé a été trouvé;
- pour faciliter la gestion du cache du client, le serveur transmet la date actuelle, la date de dernière modification du document et la date d'expiration.
- Content-Type nous apprend que le document retourné est de type HTML
- ContentLength indique que le corps du document a une longueur de 1456 octets.
- Server renseigne sur le logiciel serveur utilisé. L'envoi d'une telle information n'est pas recommandé d'un point de vue sécuritaire.

En-têtes génériques

Certains en-têtes peuvent se trouver aussi bien dans la requête que dans la réponse:

Champ	Description
Content-length	Longueur en octets des données suivant les en-têtes
Content-type	Type MIME des données qui suivent
Connection	Indique si la connexion TCP doit rester ouverte (*Keep-Alive*) ou être fermée (*close*)

En-têtes de la requête

Les en-têtes suivants n'existent que dans les requêtes HTTP.
Seul l'en-tête Host est obligatoire dans la version 1.1 de HTTP

Champ	Description
Accept	Types MIME que le client accepte
Accept-encoding	Méthodes de compression supportées par le client
Accept-language	Langues préférées par le client (pondérées)
Cookie	Données de *cookie* mémorisées par le client
Host	Hôte virtuel demandé
If-modified-since	Ne retourne le document que si modifié depuis la date indiquée
If-none-match	Ne retourne le document que sil a changé
Referer	URL de la page à partir de laquelle le document est demandé
User-agent	Nom et version du logiciel client

En-têtes de la réponse

Champ	Description
Allowed	Méthodes HTTP autorisées pour cette URI (comme POST)
Content-encoding	Méthode de compression des données qui suivent
Content-language	Langue dans laquelle le document retourné est écrit
Date	Date et heure UTC courante
Expires	Date à laquelle le document expire
Last-modified	Date de dernière modification du document
Location	Adresse du document lors d'une redirection
Etag	Numéro de version du document
Pragma	Données annexes pour le navigateur (par exemple, no.cache)
Server	Nom et version du logiciel serveur
Set-cookie	Permet au serveur d'écrire un *cookie* sur le disque du client

Lorsque le serveur renvoie un document, il lui associe un code de statut renseignant le client sur le résultat de la requête (requête invalide, document non trouvé…).

Les principales valeurs des codes de statut HTTP sont détaillées dans le tableau ci-après.

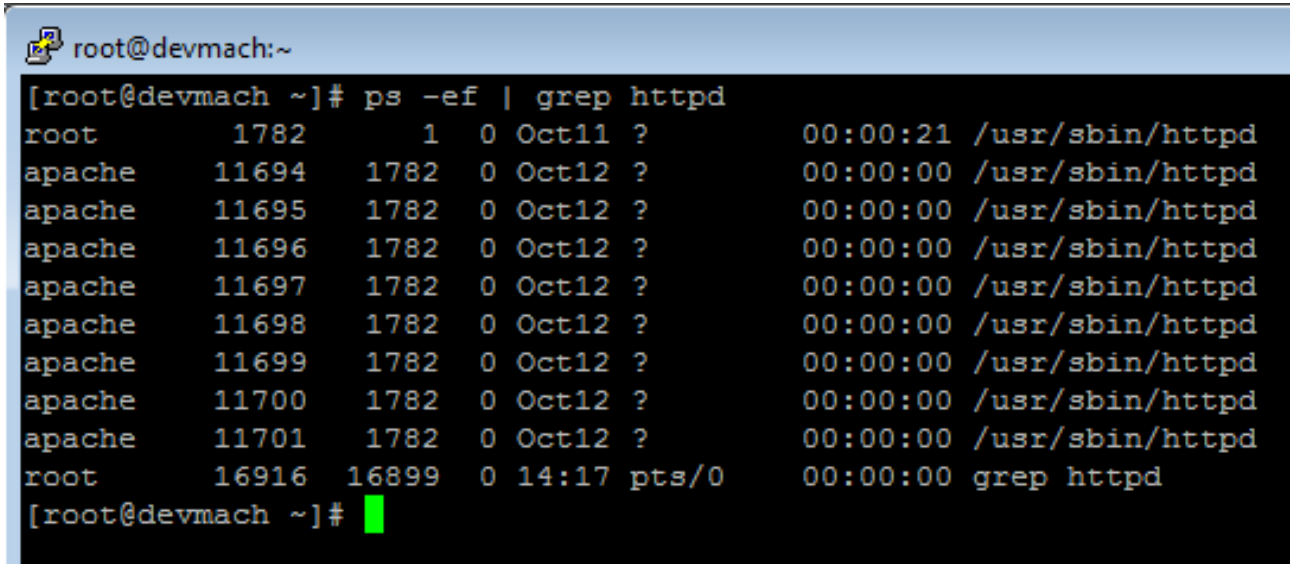
Code	Nom	Description
Information 1xx		
100	Continue	Utiliser dans le cas où la requête possède un corps.
101	Switching protocol	Réponse à une requête
Succès 2xx		
200	OK	Le document a été trouvé et son contenu suit
201	Created	Le document a été créé en réponse à un PUT
202	Accepted	Requête acceptée, mais traitement non terminé
204	No response	Le serveur n'a aucune information à renvoyer
206	Partial content	Une partie du document suit
Redirection 3xx		
301	Moved	Le document a changé d'adresse de façon permanente
302	Found	Le document a changé d'adresse temporairement
304	Not modified	Le document demandé n'a pas été modifié
Erreurs du client 4xx		
400	Bad request	La syntaxe de la requête est incorrecte
401	Unauthorized	Le client n'a pas les privilèges d'accès au document
403	Forbidden	L'accès au document est interdit
404	Not found	Le document demandé n'a pu être trouvé
405	Method not allowed	La méthode de la requête n'est pas autorisée
Erreurs du serveur 5xx		
500	Internal error	Une erreur inattendue est survenue au niveau du serveur
501	Not implemented	La méthode utilisée n'est pas implémentée
502	Bad gateway	Erreur de serveur distant lors d'une requête *proxy*

Le serveur Web

Le rôle du serveur web est de servir des ressources au travers du protocole HTTP

Lorsque l'on parle d'un serveur web, on désigne généralement le processus qui s'exécute sur une machine et qui répond aux requêtes des clients

Un tel processus s'appelle un démon

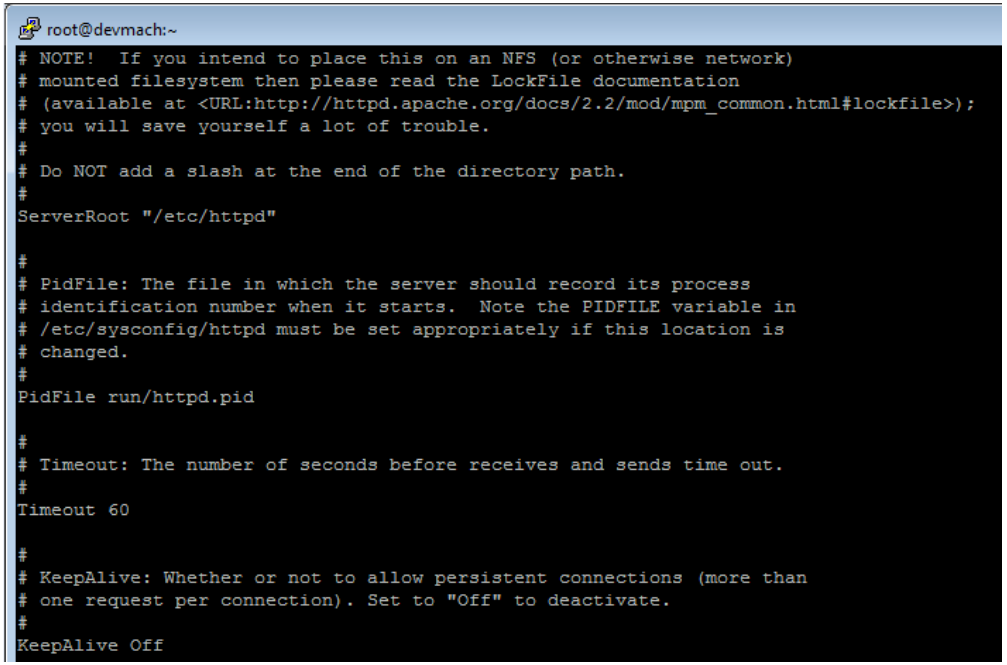


Comme la plupart des programmes, son comportement peut être modifié grâce à un fichier de configuration

Dans le cas d'Apache, ce fichier se nomme 'httpd.conf'

Il est possible de modifier beaucoup de paramètres, notamment :

- Le port d'écoute
- Le répertoire de travail
- Les fichiers de logs
- Les sites hébergés
- Les URLs de redirection
- ...



Le processus répond aux requêtes en servant des fichiers disponibles dans un répertoire particulier appelé répertoire de travail

Ce répertoire est généralement ‘www’

Il y a une correspondance qui est faite entre les URLs et l’arborescence sous le répertoire de travail

http://monserver/index.php → c:/wamp/www/index.php

http://monserver/img/wis.jpg → c:/wamp/www/img/wis.jpg

Le processus serveur note absolument toute son activité dans des fichiers

Cette pratique s'appelle la journalisation ou « logging »

Il existe, de base, deux fichiers de logs : access_log et error_log

access_log permet de journaliser toutes les requêtes

```
[root@devmach ~]# cat /var/log/httpd/access_log
192.168.186.1 - - [13/Oct/2018:14:50:18 +0200] "GET / HTTP/1.1" 403 4961 "-" "Mozilla/5.0 (Windows NT 6.1; Win64; x
64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /icons/apache_pb.gif HTTP/1.1" 200 2326 "http://192.168.186.201
/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /icons/poweredby.png HTTP/1.1" 200 3956 "http://192.168.186.201
/" "Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /favicon.ico HTTP/1.1" 404 290 "-" "Mozilla/5.0 (Windows NT 6.1
; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
192.168.186.1 - - [13/Oct/2018:14:50:19 +0200] "GET /favicon.ico HTTP/1.1" 404 290 "-" "Mozilla/5.0 (Windows NT 6.1
; Win64; x64; rv:62.0) Gecko/20100101 Firefox/62.0"
[root@devmach ~]#
```

error_log permet de journaliser toutes les erreurs ou événements serveur

```
[root@devmach ~]# cat /var/log/httpd/error_log
[Fri Oct 12 17:20:02 2018] [notice] Digest: generating secret for digest authentication ...
[Fri Oct 12 17:20:02 2018] [notice] Digest: done
[Fri Oct 12 17:20:02 2018] [notice] Apache/2.2.15 (Unix) DAV/2 PHP/7.0.25 configured -- resuming normal operations
[Sat Oct 13 14:50:18 2018] [error] [client 192.168.186.1] Directory index forbidden by Options directive: /var/www/
html/
[Sat Oct 13 14:50:19 2018] [error] [client 192.168.186.1] File does not exist: /var/www/html/favicon.ico
[Sat Oct 13 14:50:19 2018] [error] [client 192.168.186.1] File does not exist: /var/www/html/favicon.ico
```

En fonction du langage de programmation utilisé, un autre fichier peut être créé comme par exemple php_error.log

```
[27-Jun-2018 21:54:01 UTC] PHP Notice:  Undefined variable: json in C:\wamp64\www\MusicSender\src\classes\shell\Youtube.class.php on line 162
[27-Jun-2018 22:04:10 UTC] PHP Notice:  Undefined offset: 1 in C:\wamp64\www\MusicSender\src\classes\shell\Youtube.class.php on line 179
[27-Jun-2018 22:06:28 UTC] PHP Notice:  Undefined offset: 1 in C:\wamp64\www\MusicSender\src\classes\shell\Youtube.class.php on line 175
[20-Jul-2018 11:35:53 Europe/Paris] PHP Fatal error:  Uncaught Error: Call to undefined method Logger::info() in C:\wamp64\www\DomoTech\src\index.php:10
Stack trace:
#0 {main}
  thrown in C:\wamp64\www\DomoTech\src\index.php on line 10

[20-Jul-2018 11:49:22 Europe/Paris] PHP Notice:  Use of undefined constant MSG_EOF - assumed 'MSG_EOF' in C:\wamp64\www\DomoTech\src\classes\controller\Modul

[20-Jul-2018 11:49:22 Europe/Paris] PHP Stack trace:

[20-Jul-2018 11:49:22 Europe/Paris] PHP   1. {main}() C:\wamp64\www\DomoTech\src\index.php:0

[20-Jul-2018 11:49:22 Europe/Paris] PHP   2. ModuleManager::retrieveNeighbours() C:\wamp64\www\DomoTech\src\index.php:11

[20-Jul-2018 11:49:22 Europe/Paris] PHP Warning:  socket_sendto() expects parameter 4 to be integer, string given in C:\wamp64\www\DomoTech\src\classes\contr

[20-Jul-2018 11:49:22 Europe/Paris] PHP Stack trace:
```

Le serveur peut étendre ses capacités (fonctionnalités) grâce à l'ajout de modules

Ces modules sont paramétrables grâce à des fichiers de configuration

Pour le module « mod_php », le fichier de configuration associé est « php.conf »

```
[root@devmach ~]# cat /etc/httpd/conf.d/php.conf
#
# PHP is an HTML-embedded scripting language which attempts to make it
# easy for developers to write dynamically generated webpages.
#
<IfModule prefork.c>
  LoadModule php7_module modules/libphp7.so
</IfModule>

<IfModule !prefork.c>
  LoadModule php7_module modules/libphp7-zts.so
</IfModule>

#
# Cause the PHP interpreter to handle files with a .php extension.
#
AddHandler php7-script .php
AddType text/html .php

#
# Add index.php to the list of files that will be served as directory
# indexes.
#
DirectoryIndex index.php
```

On peut terminer en regardant les tendances en matière de serveur web !