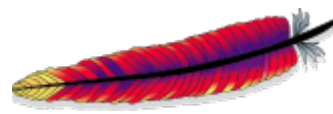


SERVEUR WEB LINUX LAMP



Le Serveur APACHE



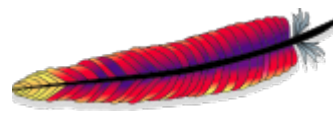
Définition d'un serveur web :

- Un serveur http ou démon http ou HTTPd (HTTP daemon) ou (moins précisément) serveur web, est un logiciel servant des requêtes respectant le protocole de communication Hypertext Transfert Protocol (HTTP), qui a été développée pour le World Wide Web.
- C'est un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur web,il est utilisé pour héberger un site web, il sert des pages web pour le clients en utilisant le protocole http.
- Le terme « serveur web » peut aussi désigner le serveur http (le logiciel) lui même. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles.

Les serveurs HTTP les plus utilisés sont :

- Apache HTTPD de la Apache Software Foudation, successeur du NCSA httpd ;
- Internet Information Services de Microsoft (IIS)
- Sun one DE Sun Microsystems (anciennement iPlanet de Netscape)
- Les serveur Web Zeus de Zeus Technology
- WAS (Webspher Application Server) de IBM

Le Serveur APACHE



HISTORIQUE :

Fin 1994 : le serveur web le plus utilisé dans le monde est celui du National Institute for Supercomputing Applications (NCSA httpd 1.3 - Université de l'Illinois). Le développement s'arrête suite au départ du développeur Rob McCool.

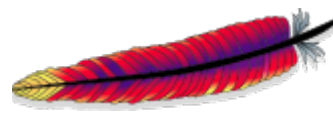
Début 1995 : naissance de la première équipe Apache (11 personnes) qui va coordonner la mise en commun de centaines de corrections et d'améliorations faite au serveur par des développeurs indépendants pour leurs propres besoins. Première version publique de Apache (0.6.2)

Début 1996 : suite à la sortie de la version 1.0 et moins d'un an après la création du groupe, Apache dépasse son "papa" en terme d'utilisation et devient le serveur Web le plus utilisé dans le monde

Aujourd'hui : Apache est toujours le serveur Web le plus utilisé. La "Apache Software Foundation" compte 132 membres à ce jour , des centaines de contributeurs et développe plusieurs projets liés à la diffusion d'information sur internet (serveur java, produits XML, langages de script...)

Le plus populaire est Apache qui sert environ **72% des sites Web en 2008**
puis 65 % en 2011.

Le Serveur APACHE

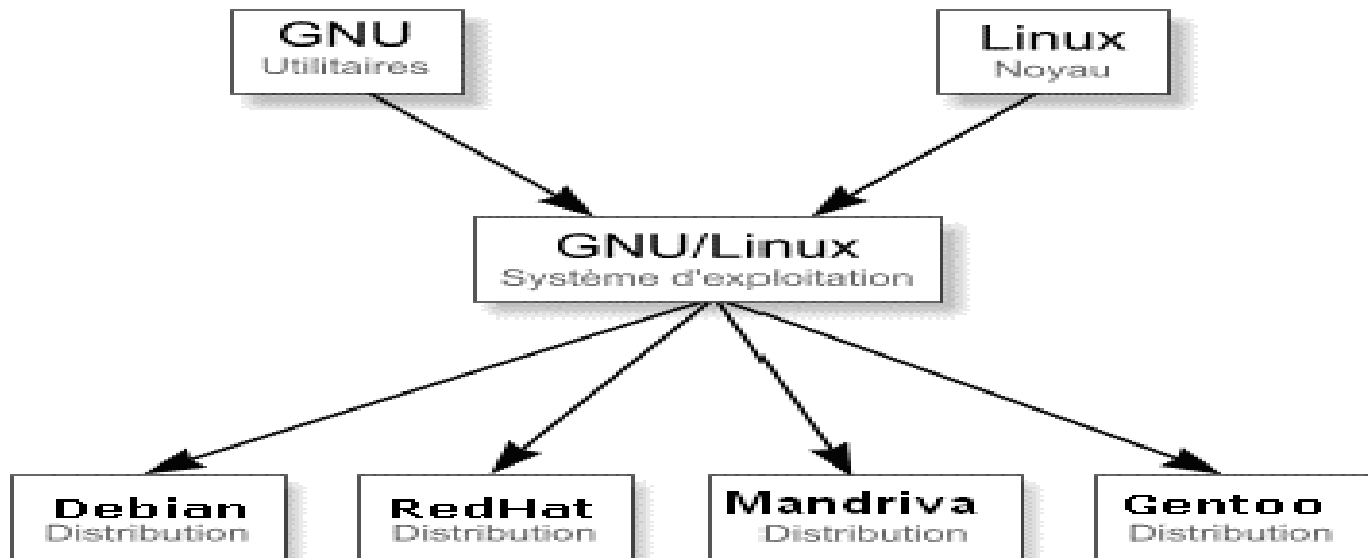


INSTALLATION:

Apache est quasi -disponible pour tous les plateformes de systèmes d'exploitation(Unix , Linux , Windows ...).

Cependant pour installer un serveur web , il faut choisir le meilleur OS adapté et optimisé pour le service. Linux étant l'OS natif d'apache, il est donc le mieux disposé en tant serveur web.

les 4 principaux distribution linux dans le monde optimisé pour le web:



Le Serveur APACHE



INSTALLATION:

L'installation qui va suivre a été effectuée sur Fedora 16 verne

Par défaut à l'installation de Fedora, les accès sont restreints en fonction des règles SELinux en vigueur sur la machine. Il est en mode dite « **Appliqué** » (Enforcing).

Afin d'éviter toute restriction au bon fonctionnement des services. On va mettre SeLinux en mode debug (**mode permissive**). En mode permissif, les règles SELinux seront interrogées, les erreurs d'accès logguées, mais l'accès ne sera pas bloqué.

(**Disabled**) : SELinux est désactivé. Rien ne sera restreint, rien ne sera loggué

Pour connaître l'état actuel de SeLinux, il faut faire la commande

```
# getenforce
Enforcing
```

Le Serveur APACHE



INSTALLATION:

Dans notre cas on doit mettre SeLinux, en mode Permissive , donc on doit passer la commande

```
# setenforce 0
```

Et inversement, en mode appliqué , donc on repassera la commande

```
# setenforce 1
```

Pour la suite on met le mode permissive c'est-à-dire setenforce 0

Le Serveur APACHE



INSTALLATION:

//installation du paquet apache

#yum install httpd

//suppression de la page de test

#rm -f /etc/httpd/conf.d/welcome.conf

//création d'un lien symbolique Perl pour MAJ Perl

#ln -s /usr/bin/perl /usr/local/bin/perl

Le Serveur APACHE



INSTALLATION ET CONFIGURATION:

Avant toutes chose, il faut faire un backup du fichier de configuration par default d'apache

```
# cat /etc/httpd/conf/httpd.conf > /etc/httpd/conf/httpd.conf.old
```

Ouvrez et modifier le fichier httpd.conf avec nano + option numero de ligne

```
# nano -c /etc/httpd/conf/httpd.conf
```

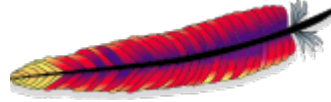
dans **etc/httpd/conf/httpd.conf** modifier les lignes comme suivantes :

**//Ligne 43: Pour cacher le type de server utilisé et l'OS, on met
ServerTokens **Prod****

**// Ligne 75: en test et dans notre cas LAN mettre ON
KeepAlive **On****

KeepAlive définit si votre serveur autorisera des connexions persistantes (c'est-à-dire plusieurs demandes par connexion). KeepAlive peut être utilisée pour empêcher tout client de consommer trop de ressources du serveur.

Le Serveur APACHE



INSTALLATION ET CONFIGURATION: #nano -c /etc/httpd/conf/httpd.conf

dans **etc/httpd/conf/httpd.conf** modifier les lignes comme suivantes :

//Ligne 135: mettez l'adresse ip-fixe de votre serveur apache

Listen 192.168.1.30: 80

// Ligne 261: email de l'admin si log ou change status du server
(peut necessiter de configurer un smtp server pour l'envoi des mails)

ServerAdmin root@dns-serveur.tld

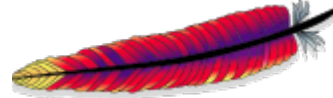
// Ligne 275: le dns du serveur auprès du registrar

ServerName www.dns-serveur.tld:80

Dans le cas où, on a enregistré un DNS auprès d'un registrar (en entreprise, les sites web):

Exemple : narutopiece.razaray.com.

Etape 1 : au niveau de l'administration du dns sur le serveur dns du registrar faut modifier la table de destination des adresses IP WAN du registrar vers votre IPWAN fixe de votre routeur



INSTALLATION ET CONFIGURATION: paramétrage DNS registrar

Paramètres DNS

Modifiez vos paramètres DNS.

Données du domaine

Nom de domaine : narutopiece.razaray.com

Paramètres DNS : **modifié** Contrôle DNS

Paramètres du serveur de noms

Modifier

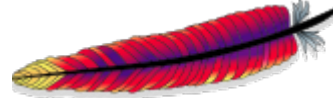
Serveur de noms : Serveur de noms [REDACTED]

Adresse IP (A-record) : **Adresse IP modifiée : 82.83.69.128**

Serveur email (MX-record) : Serveur email [REDACTED]

ip wan apache fixe

Le Serveur APACHE



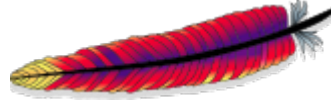
INSTALLATION ET CONFIGURATION: paramétrage table de routage NAT (routeur FAI)

Etape 2 : maintenant depuis votre routeur. Aller dans la page d'administration de votre routeur et rediriger les requetes clients sur le port 80 et/ou 8080 vers votre server apache (192.168.1.30) en précisant bien sur que c'est TCP puisque c'est http.

Redirections de ports:

Port	Protocole	Destination	Port	
80	tcp ▼	192.168.1.30	80	X
8080	tcp ▼	192.168.1.30	8080	X
	udp ▼	192.168.1.30		Ajouter

Le Serveur APACHE



INSTALLATION ET CONFIGURATION: #nano -c /etc/httpd/conf/httpd.conf

dans **etc/httpd/conf/httpd.conf** modifier les lignes comme suivantes :

//Ligne 330: permettre au serveur de suivre les liens symbolique
(exemple pour les redirections de Url simple et en cgi)

Options Indexes FollowSymLinks ExecCGI

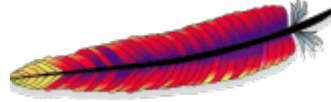
(ceci permet aussi d'activer les script ou bibliothèque CGI)

// Ligne 337: donner la priorité aux fichiers .htaccess

AllowOverride All

.htaccses: protège l'accès à un répertoire ou un fichier par un mot de passe, définir les pages d'erreurs personnalisées, gérer la réécriture d'url, les restriction d'accès à un répertoire ou un fichier.

Le Serveur APACHE



INSTALLATION ET CONFIGURATION: #nano -c /etc/httpd/conf/httpd.conf

dans **etc/httpd/conf/httpd.conf** modifier les lignes comme suivantes :

//Ligne 401: le(s) page(s) par défauts afficher par le server lors on accède à sa racine

DirectoryIndex index.html index.cgi index.php

// Ligne 535: Pour plus de sécurité, masquez l'identification du serveur sur les pages d'erreur (message Apache Server + numéro de version) .

AVEC La ligne 43 précédemment « ServerTokens Prod »

ServerSignature Off

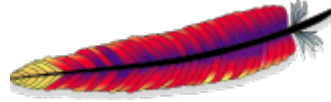
// Ligne 758: apache ne nous dictera pas le norme de nos pages. C'est déjà défini selon avec la norme W3 dans l'entête de nos pages

#AddDefaultCharset UTF-8

Exemple: pour dans l'entête de nos pages html et/ou php. Ici la balise meta content-type le définit déjà

`<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />`

Le Serveur APACHE



INSTALLATION ET CONFIGURATION: `#nano -c /etc/httpd/conf/httpd.conf`

dans `etc/httpd/conf/httpd.conf` modifier les lignes comme suivantes :

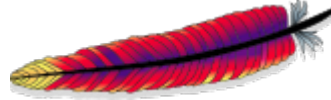
//Ligne 795: La directive ci-dessus indique à Apache qu'il doit permettre l'exécution des fichiers CGI. Vous devez aussi indiquer au serveur quels fichiers sont des fichiers CGI. La directive AddHandler suivante indique au serveur qu'il doit traiter tous les fichiers possédant une extension cgi ou pl en tant que programmes CGI :

AddHandler cgi-script .cgi .pl

SAUVER /etc/httpd/conf/httpd.conf et redemarrer apache
service httpd start ; service httpd status

ATTENTION NE PAS ACTIVER APACHE AU DEMARRAGE AVEC:
systemctl enable httpd.service ni de #chkconfig --add httpd

Le Serveur APACHE



ADMINISTRATION D'APACHE:

Afin de gerer apache et créer facilement et aisement les serveur web on va installer le gestionnaire graphique d'apache:

```
#yum install system-config-httpd
```

//on le lance

```
# system-config-httpd
```

// laisser le shell en cours en fonctionnement . Sinon cet outil quittera.

The screenshot shows the 'HTTP Server Configuration' window with the 'Main' tab selected. The 'Basic Setup' section contains two text fields: 'Server Name' (empty) and 'Webmaster email address' (containing 'root@localhost'). The 'Available Addresses' section features a large empty list box and three buttons: '+ Add...', 'Edit...', and 'Delete'. At the bottom, there is a note 'Set addresses for Apache to listen for requests on.' and three buttons: 'Help', 'Cancel', and 'OK'.



INSTALLATION DE PHP : #nano -c /etc/httpd/conf/httpd.conf

On installe php sur le serveur afin de pouvoir executer les scripts php :

```
# yum install php php-mbstring php-pear
```

Php-mbstring fourni le support de traduction mais aussi Les jeux de caractères multi-octets (ex : les types *char*, *varchar* dans *Mysql*)

Pear : pour le framework php (*biblio de dev php*) l'authentification, les services web, le chiffrement, le cache et le XML

```
//on le redemarre apache
```

```
# service httpd stop
```

```
# service httpd start
```



SECURISE L'ACCES AU SERVEUR : .htaccess

Fonctionnement:

L'option **AccessFileName .htaccess** fixe globalement le nom des fichiers de parametres locaux. Un fichier **.htaccess**, présent dans un répertoire, peut contrôler complètement les accès à ce répertoire, pourvu que la permission soit accordée par l'option **AllowOverride AuthConfig** ou **AllowOverride All**

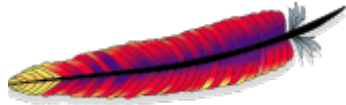
Alors, les directives contenues dans ce fichier seront systématiquement respectées avant toute autorisation. Voici les directives usuelles et leur signification :

AuthType basic, type d'authentification communément adopté, fait hélas circuler les mots de passe en clair ;

AuthName texte, affichera le texte comme invite dans la boite de dialogue ;

AuthUserFile chemin/fichier, précise le fichier qui contient les comptes et mots de passe des utilisateurs ayant droit d'accès ;

Require valid-user | liste-noms tous, ou seulement les comptes énumérés dans la liste, auront accès au répertoire.



SECURISE L'ACCES AU SERVEUR : **.htaccess exemple**

1) On va Créer le répertoire à sécuriser /var/www/html/prive, y placer quelques pages HTML. Tester leur accessibilité pour tous. Sinon penser à modifier les permissions Linux sur ces fichiers.

2) Créer dans ce répertoire à protéger le fichier .htaccess. En voici une écriture standard :

```
AuthUserFile /etc/httpd/users
AuthGroupFile /dev/null
AuthName "Accès privé"
AuthType Basic
# autres clauses
# AuthGroupFile /etc/httpd/conf/groups

<limit GET>
# ATTENTION : GET en majuscules !
require valid-user
# require user toto dupond
# require group profs
</limit>
```



SECURISE L'ACCES AU SERVEUR : **.htaccess exemple**

3. Dans ces conditions où se trouvera le fichier d'authentification ?
4. Créer un premier compte Apache avec la commande htpasswd .

```
# cd /etc/httpd/
```

```
# htpasswd -c users admin
```

---> mot de passe demandé (admin), puis confirmer.

5. Examiner le fichier **/etc/httpd/users**

L'utilitaire **htpasswd** a créé (**option -c**) le fichier **users** dans le **répertoire courant**, ici **/etc/httpd**, et y a enregistré **admin** avec son mot de passe **crypté**.

6. Ajouter un second compte, **toto**, puis d'autres htpasswd users toto
---> mot de passe demandé(toto), puis confirmer.

7. Tester l'accès au répertoire **http://serveur/prive**. Pourquoi la protection ne semble-t-elle pas fonctionner ?

(remarque : service httpd reload permet de prendre en compte les changements de configuration)

8. Rechercher dans le fichier de configuration la section **<Directory /var/www/html>** qui fixe des directives par défaut pour le site principal. Par sécurité mettre si nécessaire L'option **AllowOverride None**

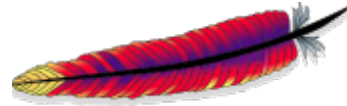


SECURISE L'ACCES AU SERVEUR : **.htaccess exemple**

9. Ajouter une option concernant le répertoire privé

```
<Directory /var/www/html/prive>  
AllowOverride ...  
Options -Indexes  
.....  
</Directory>
```

10. Retester normalement avec succès ! N'oubliez pas de relancer le navigateur quand on change de compte.



INSTALLATION DE MYSQL :

Mysql est un SGBD (Système de gestion de base de données). Il sert à enregistrer, stocker, trier, modifier, rechercher les données (le contenu d'un formulaire d'identité par exemple).

Pour l'installer, lancer la commande:

```
# yum install mysql-server
```

```
# service mysqld start ; service mysqld status
```

ATTENTION NE PAS ACTIVER MYSQLD AU DEMARRAGE AVEC:

```
# systemctl enable mysqld.service ni de #chkconfig --add mysqld
```



CONFIGURATION DE MYSQL :

Par défaut l'utilisateur mysql est root et il n'a pas de mots de passe. On va donc définir un mots de passe pour root , en se connectant sur le server mysql:

```
[root@www ~]# mysql -u root           # connect to MySQL
```

```
# show user info
```

```
mysql> select user,host,password from mysql.user;
```

user	host	password
root	localhost	
root	www.server.world	
root	127.0.0.1	
root	::1	
	localhost	
	www.server.world	

```
6 rows in set (0.00 sec)
```

```
# set root password
```

```
mysql> set password for root@localhost=password('password');
```

```
Query OK, 0 rows affected (0.00 sec)
```



CONFIGURATION DE MYSQL :

```
# set root password
```

```
mysql> set password for root@'127.0.0.1'=password('password');
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
# set root password
```

```
mysql> set password for root@'www.server.world'=password('password');
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select user,host,password from mysql.user;
```

user	host	password
root	localhost	XX
root	www.server.world	XX
root	127.0.0.1	XX
root	::1	
	localhost	
	www.server.world	

```
6 rows in set (0.00 sec)
```

```
# delete IPV6 user
```

```
mysql> delete from mysql.user where user='root' and host='::1';
```

```
Query OK, 1 rows affected (0.00 sec)
```



CONFIGURATION DE MYSQL :

```
# delete anonymous user
```

```
mysql> delete from mysql.user where user="";
```

```
Query OK, 2 rows affected (0.00 sec)
```

```
mysql> select user,host,password from mysql.user;
```

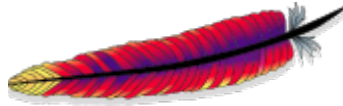
+-----+-----+-----+-----+			
user	host	password	
+-----+-----+-----+-----+			
root	localhost		XX
root	www.server.world		XX
root	127.0.0.1		XX
+-----+-----+-----+-----+			

```
3 rows in set (0.00 sec)
```

```
mysql> exit
```

```
# quit
```

```
Bye
```



CONFIGURATION DE MYSQL :

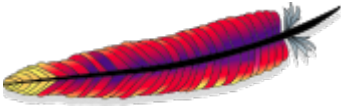
```
[root@www ~]# mysql -u root -p      # connect with root
Enter password:                     # MySQL root password
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.16 MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
```



INSTALLATION DE PHP MYADMIN : Interfaces web d'administration de mysql

// installation

```
# yum install phpMyAdmin php-mysql php-mcrypt
```

// Configurer phpmyadmin pour que vous ayez accès vous seul à phpmyadmin

```
# nano -c /etc/httpd/conf.d/phpMyAdmin.conf
```

// ligne 14: ajouter votre adresse IP localhost seulement + votre masque sous-réseau

Allow from 127.0.0.1 10.0.0.0/24

// redemarrer apache

```
# service httpd start ; service httpd status
```

Lancez ensuite **phpmyadmin** dans votre navigateur web comme ci-dessous:

