GNU / Linux — Apache

HTTP • configuration • Virtual Hosts • sécurité • HTTPS

Objectifs

Installer un serveur HTTP.
Comprendre les possibilités.
Les CGI et les modules de langages.
Cybersécurité.
Délégation de la configuration.

Code couleur du cours

Bleu : notions fondamentales et architecture.
Vert : mémos techniques et bonnes pratiques.
Orange : points d'attention ou éléments historiques.
Violet : corrections techniques intégrées au support.

Corrections techniques intégrées

Le contenu du support est conservé, avec correction des erreurs suivantes

HyperText Transfer Protocol, URI = Uniform Resource Identifier, TLD et orthographe générale.

HTTP/1.x est un protocole textuel : il n'est pas « basé sur Telnet » ; Telnet peut seulement servir à observer manuellement un échange TCP en clair.

URL et URN sont des catégories de URI : URI ≠ URL + URN.

Le nombre de méthodes HTTP n'est pas limité à neuf ; quatre méthodes principales sont étudiées ici.

« A patchy server » est un jeu de mots devenu légendaire, pas l'origine officielle du nom Apache.

MaxClients et MaxRequestsPerChild sont présentés avec leurs noms Apache 2.4 modernes : MaxRequestWorkers et MaxConnectionsPerChild.

Worker MPM n'est pas « non thread-safe » : ce sont les modules chargés qui doivent être compatibles avec les threads.

HTTP Basic n'est pas une 2FA à lui seul ; Digest ne chiffre pas le trafic et ne remplace pas HTTPS.

Les chemins Debian utilisent /etc/apache2/sites-available/.

Les certificats Let's Encrypt sont des certificats DV ; la notion de « classe 2 » est obsolète.

Les doublons strictement identiques de requêtes HTTP ont été retirés pour améliorer la lisibilité.

Plan

1. Le protocole HTTP

Définition et modèle de communication

- HyperText Transfer Protocol
- Récupérer des documents
  - Liés par un lien hypertexte <a href="lien">
- Protocole de la couche 7 OSI
  - Application
  - Process To Process
  - Des logiciels qui échangent de l'information sur un réseau
- Modèle Client-Serveur
  - Protocole de transport : TCP pour HTTP/1.1 et HTTP/2
  - Port 80 : HTTP
  - Port 443 : HTTPS

Mémo technique

HTTP décrit le sens des messages applicatifs.
TCP assure le transport fiable pour HTTP/1.1 et HTTP/2.
HTTPS = HTTP protégé par TLS.

Client, serveur, requête et réponse

- Client / Serveur
- Protocoles ?
  - Ensemble des règles qui permettent à deux entités de communiquer
- HTTP/1.x est un protocole textuel, comme plusieurs protocoles historiques tels que FTP, POP et SMTP.
- Système Requête / Réponse
  - Client (souvent le navigateur)
  - Serveur (le serveur web)
- Sans état

Client Navigateur →	Requête HTTP →	Serveur Apache →	Réponse HTTP →	Ressource HTML / image / JSON

« Sans état » ne veut pas dire « sans session »

HTTP ne mémorise pas automatiquement les requêtes précédentes.
Les applications ajoutent des cookies, des jetons ou des sessions côté serveur pour conserver un contexte.

Accès aux ressources : URI, URL et URN

- URI : Uniform Resource Identifier
  - Chaîne de caractères qui identifie une ressource.
  - schéma://hôte.nomdedomaine.tld/chemin/vers/ma/ressource
  - http://www.c2lr.fr/wp-content/uploads/2017/04/hard_drive_259x259.jpg
- URL = Uniform Resource Locator

- URN = Uniform Resource Name

Correction importante

Une URL et une URN sont des formes de URI.

On ne doit pas retenir la formule « URI = URL + URN » comme une addition.

Histoire du protocole

Version	Repère historique	Apports essentiels
HTTP/0.9	1990 — Tim Berners-Lee — création du World Wide Web.	Très minimal : récupération de documents hypertexte.
HTTP/1.0	Standardisé en 1996.	Ajout d'en-têtes, de types de contenu, d'authentification et de mécanismes de cache.
HTTP/1.1	RFC initial en 1997, révision majeure en 1999.	Connexions persistantes, pipelining, négociation de contenu, en-tête Host obligatoire.
HTTP/2	Standardisé en 2015.	Multiplexage, compression d'en-têtes HPACK et mécanisme Server Push.

Exemple de requête HTTP

```
POST /en/html/index.html HTTP/1.1
Host: www.website.com
User-Agent: Mozilla/5.0 (Windows;en-GB; rv:1.8.0.11)
Accept: text/xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 39

name=MyName&male=yes
```

Élément	Rôle
POST /en/html/index.html HTTP/1.1	Méthode, cible de la requête et version HTTP.
Host	Nom du site demandé ; indispensable au Virtual Hosting par nom.
User-Agent	Identifie le logiciel client.
Accept*	Formats, langues, encodages et jeux de caractères acceptés.
Content-Type	Format du corps envoyé.
Content-Length	Taille du corps en octets.
name=MyName&male=yes	Corps de la requête POST.

Les méthodes HTTP

- HTTP définit plusieurs méthodes standard et extensibles.
- Quatre méthodes principales nous intéressent ici :

Méthode	Fonction
GET	Récupère la représentation d'une ressource.
POST	Soumet des données à la ressource identifiée.
PUT	Crée ou remplace la représentation d'une ressource à l'adresse visée.
DELETE	Demande la suppression de la ressource spécifiée.

Les réponses et les statuts

- Le protocole prévoit 5 catégories

Catégorie	Signification
1xx	Informations du protocole
2xx	Succès
3xx	Redirections
4xx	Erreurs côté client
5xx	Erreurs côté serveur

Statut	Lecture simple
200 OK	La requête a réussi.
201 Created	Une ressource a été créée.
301 Moved Permanently	La ressource a été déplacée définitivement.
403 Forbidden	Le serveur comprend la requête mais refuse l'accès.
404 Not Found	La ressource n'a pas été trouvée.
405 Method Not Allowed	La méthode n'est pas autorisée pour cette ressource.
500 Internal Server Error	Erreur interne du serveur.
501 Not Implemented	Fonctionnalité ou méthode non prise en charge par le serveur.

2. Apache HTTP Server

Origine du projet

- Apache est issu du serveur httpd de la NCSA.
- Une série de correctifs et d'améliorations a été appliquée au code existant.
- Le jeu de mots « a patchy server » est devenu une légende populaire.
- Le nom Apache a été choisi par respect et appréciation pour les peuples et tribus Apache.
- Naissance de la fondation Apache
  - L'une des plus actives dans l'Open Source
  - Une licence de logiciel libre
- Le code a ensuite connu une réécriture et une évolution complète.

Repère historique

Le support associait directement le nom à « A patchy HTTPd Server ».
Il faut retenir que cette expression est un jeu de mots postérieur devenu célèbre, pas l'étymologie officielle.

Audience en 2023

- Le serveur le plus utilisé pendant longtemps...
- Mais Nginx est également très présent.
- Les parts de marché évoluent selon les mesures et les périodes.

Attention

Cette partie est un repère historique du support (2023), pas un classement figé.
Les performances dépendent de la charge, des modules, du MPM, de la configuration et du matériel.

3. Gestion du multi-processus : les MPM

- MPM : Multi-Processing Modules
  - Intégrés dans le cœur du logiciel
  - Parallélisme dans le traitement des requêtes
- Trois MPM principaux sous Unix dans Apache 2.4 : Prefork, Worker et Event.
- ITK est un MPM tiers, utilisé dans certains contextes d'hébergement mutualisé.

Deux grandes familles

Famille	Principe
Non threadé	Prefork : plusieurs processus, un thread par processus.
Threadé / hybride	Worker et Event : plusieurs processus contenant plusieurs threads.

Prefork MPM

- Chaque requête est traitée par un processus enfant.
- Un seul thread par processus.
- Consomme davantage de mémoire car un processus est plus lourd qu'un thread.
- Privilégie la simplicité et la compatibilité avec des modules non thread-safe.
- Le MPM actif dépend de la distribution et de la configuration ; Prefork n'est pas universellement le MPM par défaut.

Directive Apache 2.4	Description
MaxRequestWorkers (ancien MaxClients)	Nombre maximal de requêtes ou connexions traitées simultanément.
MaxConnectionsPerChild (ancien MaxRequestsPerChild)	Nombre maximal de connexions qu'un processus enfant traite avant d'être remplacé.
ServerLimit	Limite supérieure du nombre de processus serveur.
StartServers	Nombre de processus lancés au démarrage.
MinSpareServers	Nombre minimal de processus inactifs conservés.
MaxSpareServers	Nombre maximal de processus inactifs conservés.

Worker MPM

- Modèle hybride multi-processus et multi-thread.
- Chaque processus enfant peut contenir plusieurs threads qui traitent les requêtes.
- Permet de traiter davantage de requêtes avec moins de ressources que Prefork.
- Les modules chargés doivent être compatibles avec un environnement threadé.

Directive Apache 2.4	Description
MaxRequestWorkers (ancien MaxClients)	Nombre maximal total de threads pouvant traiter des requêtes.
ThreadLimit	Limite supérieure de ThreadsPerChild.
ThreadsPerChild	Nombre de threads créés par processus enfant.
StartServers	Nombre de processus enfants lancés au démarrage.

MinSpareThreads	Nombre minimal de threads inactifs disponibles.
MaxSpareThreads	Nombre maximal de threads inactifs disponibles.

Event MPM

- Modèle similaire à Worker MPM.
- Gestion événementielle des connexions, notamment des connexions Keep-Alive.
- Optimisé pour un grand nombre de connexions simultanées.
- Options proches de Worker MPM.

Mémo MPM

Prefork : processus uniquement, compatibilité maximale, davantage de mémoire.

Worker : processus + threads, meilleure densité de requêtes.

Event : Worker amélioré pour la gestion événementielle et les connexions persistantes.

ITK MPM

- Isolation des privilèges dans des processus.
- Chaque Virtual Host peut utiliser son propre UID/GID.
- Peut répondre à certains besoins d'hébergement mutualisé.
- Plus exigeant en performances et nécessitant une installation/configuration spécifiques.

Correction de sécurité

ITK peut apporter une séparation de privilèges par Virtual Host.

Il ne doit pas être présenté comme le choix automatique « cybersec » : le contexte, la maintenance et les alternatives modernes doivent être évalués.

4. Les fichiers de configuration sous Debian

- Configuration principale, puis inclusion d'autres fichiers et répertoires.
- L'organisation varie selon les distributions : httpd.conf, apache2.conf, vhost.d...
- Ici, nous parlons de Debian, dans /etc/apache2/.
- Tous les fichiers inclus sont traités comme une configuration globale cohérente.

/etc/apache2/ apache2.conf →	ports.conf →	mods-enabled/ →	conf-enabled/ →	sites-enabled/

Emplacement Debian	Rôle
apache2.conf	Configuration principale et directives Include.
ports.conf	Ports et adresses d'écoute.
mods-available / mods-enabled	Modules disponibles et activés.
conf-available / conf-enabled	Fragments de configuration généraux.
sites-available / sites-enabled	Virtual Hosts disponibles et activés.

Directive <Directory> et Options

- La directive <Directory> applique des règles à un répertoire du système de fichiers.
- Options définit une liste d'options applicables.
- Dans les sous-répertoires, + ou - permet d'ajouter ou retirer des options par rapport au parent.

Option	Description
All	Active toutes les options sauf MultiViews.
None	Désactive toutes les options.
ExecCGI	Permet l'exécution de scripts CGI dans le répertoire.
FollowSymLinks	Autorise Apache à suivre les liens symboliques.
Indexes	Liste le contenu du répertoire lorsqu'aucun fichier d'index n'existe.
MultiViews	Effectue une négociation de contenu à partir d'un nom sans extension précise.

```
<Directory /var/www/site>
   Options -Indexes +FollowSymLinks
</Directory>
```

Point sécurité

Options Indexes peut révéler la liste des fichiers d'un répertoire.

On le désactive généralement avec Options -Indexes lorsque ce comportement n'est pas voulu.

Directive Require

- Fonctionne comme un contrôle d'accès en couche application.
- Vérifie si un client ou un utilisateur peut accéder à une ressource.

- Peut utiliser ip, host, forward-dns, local et des expressions.

```
<Directory /var/www/intranet>
   Require ip 192.168.10.0/24
</Directory>
```

Mémo

Le pare-feu réseau filtre des flux avant l'application.

Require décide de l'accès à une ressource HTTP dans Apache.

Directive AllowOverride et .htaccess

- Autorise la délégation de certaines directives aux fichiers .htaccess.
- Les webmasters peuvent adapter un répertoire sans modifier directement la configuration globale root.

```
<Directory /var/www/www.site.com>
   AllowOverride All
</Directory>

<Directory /var/www/www.site.com>
   AllowOverride Options=Indexes,FollowSymLinks
</Directory>
```

Correction de syntaxe

FollowSymLinks n'est pas une catégorie AllowOverride autonome.

Pour limiter précisément les options autorisées dans .htaccess, Apache 2.4 accepte la syntaxe Options=Indexes,FollowSymLinks.

Performance et sécurité

AllowOverride None évite la recherche de fichiers .htaccess dans chaque niveau de répertoire.

La configuration centrale est généralement plus performante et plus facile à auditer.

N'autoriser .htaccess que si la délégation est réellement nécessaire.

5. Modules, CGI et gestionnaires de contenu

Activation d'un module sous Debian

- a2enmod crée des liens symboliques depuis mods-available vers mods-enabled.
- Le fichier .load contient la directive LoadModule ; un fichier .conf peut compléter la configuration.
- Il faut ensuite vérifier et recharger ou redémarrer Apache.

```
a2enmod <nom_du_module>
apache2ctl configtest
systemctl reload apache2
```

Exemple PHP lié à une version installée

```
root@vm001:~# a2enmod php8.2

LoadModule php_module /usr/lib/apache2/modules/libphp8.2.so

<IfModule php_module>
</IfModule>
```

Mise à jour du support

L'ancien exemple mélangeait php7.3 et mod_php5.

Le nom exact du module et le chemin dépendent de la version PHP installée.

CGI et gestionnaires de contenu

- Par défaut, Apache envoie un fichier statique dans la réponse HTTP.
- En Web dynamique :
  - Un script est exécuté.
  - Le script produit une sortie.
  - Cette sortie est placée dans la réponse HTTP.
- Les modules associent un gestionnaire à un type de contenu ou à une extension.

Requête .php / .cgi →	Apache →	Handler / CGI / PHP-FPM →	Exécution →	Réponse dynamique

Association par extension : AddHandler

Paramètre	Description
handler-name	Nom du gestionnaire fourni par le module.
file-ext	Extension confiée au gestionnaire au lieu d'être envoyée comme fichier statique.

```
AddHandler handler-name file-ext

AddHandler cgi-script .cgi
```

SetHandler, PHP-FPM et proxy_fcgi

- PHP-FPM et proxy_fcgi permettent d'utiliser des versions ou pools PHP différents.

- Dans des Virtual Hosts différents ou des répertoires différents d'un même site.

```
SetHandler handler-name

# Exemple historique : adapter le socket à la version réellement installée
<FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php/php7.2-fpm.sock|fcgi://localhost"
</FilesMatch>

<FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php/php7.0-fpm.sock|fcgi://localhost"
</FilesMatch>
```

Attention versions PHP

PHP 7.0 et 7.2 sont obsolètes : ces blocs sont conservés comme exemples historiques de syntaxe.

Sur Debian 12, le socket courant est généralement lié à PHP 8.2 si ce paquet est installé.

6. Authentification HTTP

- Le client doit s'authentifier pour accéder à une ressource.
- Peut protéger temporairement une ressource ou ajouter une barrière supplémentaire.
- S'applique à un bloc <Directory> ou <Location>.

Correction importante

Une authentification HTTP supplémentaire n'est pas automatiquement une 2FA.
Une vraie 2FA combine deux facteurs de nature différente : connaissance, possession ou biométrie.

Basic et Digest

Type	À retenir
Basic	Largement supporté. Les identifiants sont encodés en Base64, pas chiffrés par Basic lui-même. Utiliser HTTPS.
Digest	Mécanisme challenge-réponse qui évite d'envoyer le mot de passe en clair, mais repose historiquement sur MD5 et ne chiffre pas le contenu HTTP.

Règle de sécurité

Basic sur HTTPS est généralement préférable à Basic sur HTTP.
Digest ne remplace pas TLS : les données de la page restent lisibles sans HTTPS.

HTTP Authentification — Basic

```
<Directory "/var/www/www.c2lr.fr/planning">
    # Le type d'authentification
    AuthType Basic
    # Le titre de la fenêtre pour se connecter
    AuthName "Protected files"
    # Le fichier qui contient les comptes et les mots de passe hachés
    AuthUserFile /etc/apache2/auth/c2lr
    # Seuls les utilisateurs indiqués ont accès
    Require user c2lr admin
    # Sinon, accepter tout utilisateur présent dans le fichier
    # Require valid-user
</Directory>
```

Création des utilisateurs avec htpasswd

Option / argument	Description
-c	Crée le fichier. À utiliser uniquement la première fois, sinon le fichier existant est écrasé.
filename	Fichier contenant la base des utilisateurs.
username	Identifiant de l'utilisateur.

```
htpasswd [-c] filename username

htpasswd -c /etc/apache2/auth/c2lr c2lr
htpasswd /etc/apache2/auth/c2lr admin
```

Piège classique

La deuxième création ne doit pas utiliser -c.

Sinon le fichier est recréé et les comptes précédents sont perdus.

7. Virtual Hosting

- Plusieurs sites sur le même serveur.
- Deux techniques :
  - Un site par adresse IP, avec plusieurs adresses sur les interfaces réseau.
  - Virtual Hosting par nom, basé sur l'en-tête Host de la requête HTTP.

Même IP 192.168.190.130 →	Host: blog.c2lr.fr →	Apache choisit le VHost →	DocumentRoot /var/www/blog.c2lr.fr

```
POST /en/html/index.html HTTP/1.1
Host: www.website.com
...

name=MyName&male=yes
```

Virtual Hosting par nom de domaine

```
# /etc/apache2/sites-available/blog.c2lr.fr.conf
<VirtualHost *:80>
   ServerName blog.c2lr.fr
   DocumentRoot /var/www/blog.c2lr.fr
</VirtualHost>

# /etc/apache2/sites-available/www.c2lr.fr.conf
<VirtualHost *:80>
   ServerName www.c2lr.fr
   DocumentRoot /var/www/www.c2lr.fr
</VirtualHost>

a2ensite www.c2lr.fr.conf
apache2ctl configtest
systemctl reload apache2
```

Mémo Debian

sites-available : fichiers de Virtual Hosts disponibles.

sites-enabled : liens symboliques vers les sites activés.

a2ensite active un site ; a2dissite le désactive.

8. HTTPS, TLS et certificats

Module ssl

- Active le support HTTPS.
- Supporte des certificats issus d'une PKI ou auto-signés.

```
# Doit figurer dans la configuration d'écoute (ports.conf)
Listen 443

# Doit figurer dans le fichier de Virtual Host
SSLEngine on
# Le certificat contenant notamment la clé publique
SSLCertificateFile /etc/apache2/ssl/site1/server.crt
# La clé privée associée
SSLCertificateKeyFile /etc/apache2/ssl/site1/server.key
```

Ne pas confondre

SSLCertificateFile contient le certificat, qui inclut la clé publique et l'identité signée.

SSLCertificateKeyFile contient la clé privée : elle doit rester secrète et protégée.

Let's Encrypt

- Autorité de certification à but non lucratif fournissant gratuitement des certificats TLS automatisés.
- Permet d'obtenir un certificat signé par une autorité reconnue.
- Bénéficie des principes d'une PKI : infrastructure à clé publique.
- Certificats de validation de domaine (DV) : validation du contrôle du nom de domaine.

Correction du support

La notion « certificat de classe 2 » n'est plus la bonne manière de décrire les certificats Let's Encrypt.

Il faut retenir : certificat DV, sans validation étendue EV.

Let's Encrypt : fonctionnement avec ACME HTTP-01

- Installation de Certbot et de son plugin Apache.
- Exécution manuelle ou automatique de Certbot.
- Le chemin /.well-known/acme-challenge/ doit être servi publiquement par le site sur le port 80.
- Si une authentification HTTP protège tout le site, ce chemin doit rester accessible publiquement.
- Certbot ou le plugin choisi écrit un jeton de validation dans le répertoire configuré.
- Let's Encrypt tente de récupérer le jeton depuis Internet.
- Si la validation réussit, les certificats sont émis et la configuration peut être finalisée.

```
apt update
apt install certbot python3-certbot-apache

certbot --apache
```

Certbot crée un jeton →	/.well-known/ acme-challenge/ →	Let's Encrypt interroge le site →	Validation →	Certificat émis

Précision technique

Le jeton doit être accessible depuis l'extérieur sur le port 80 pour HTTP-01.

Le répertoire doit être inscriptible par l'outil ACME utilisé et lisible/servi par Apache ; il n'est pas obligatoirement écrit directement par www-data.

9. TODO du support

- HttpAuthDigest
- Exemple Config CGI avec Python
- Plusieurs versions de PHP
- Configuration des CORS et des CSP
- mod_user_dir, substitute, rewrite
- VirtualHost IP Based

10. Mémo technique final

Notion	À retenir
HTTP	Protocole applicatif requête/réponse, sans état par défaut.
Apache httpd	Serveur HTTP modulaire et extensible.
MPM	Modèle de traitement des connexions : Prefork, Worker ou Event.
<Directory>	Applique des directives à un répertoire du système de fichiers.
Options	Active ou désactive des comportements dans un répertoire.
Require	Autorisation d'accès applicative.
AllowOverride	Délégation de directives via .htaccess.
a2enmod	Active un module sous Debian.
a2ensite	Active un Virtual Host sous Debian.
AuthType Basic	Authentification HTTP ; à protéger par HTTPS.
htpasswd	Crée et gère un fichier de comptes HTTP Basic.
TLS / HTTPS	Chiffre et authentifie le canal HTTP.
Let's Encrypt	Autorité de certification automatisée fournissant des certificats DV.

Ordre de diagnostic Apache

1. apache2ctl configtest — vérifier la syntaxe.
2. systemctl status apache2 — vérifier le service.
3. apache2ctl -S — vérifier les Virtual Hosts.
4. curl -I http://nom-du-site — tester la réponse HTTP, puis consulter /var/log/apache2/error.log et access.log.

11. Sources officielles de vérification

- RFC 3986 — Uniform Resource Identifier (URI): Generic Syntax.

- RFC 9110 — HTTP Semantics et classes de statuts.
- Documentation Apache HTTP Server 2.4 — MPM, core, authentification, Virtual Hosts et modules.
- Apache Software Foundation — histoire du projet et origine du nom Apache.
- Documentation Let's Encrypt — ACME et challenge HTTP-01.

Les ajouts pédagogiques et corrections sont clairement distingués du contenu initial du support.