
Servidor Web Apache

Índex

Fonaments i protocols	1
Funcionament bàsic d'una petició HTTP	1
Capçalera Host i Virtual Hosts	2
HTTPS i TLS	2
Documentació de referència	2
1. Què és Apache Web Server	2
Arquitectura de processament (MPM)	3
2. Instal·lació d'Apache a Ubuntu	3
2.1. Actualitza el sistema	3
2.2. Instal·la el paquet apache2	3
2.3. Verifica la instal·lació	3
2.4. Comprova que el servei està actiu	4
2.5. Obre el tallafocs (UFW)	4
2.6. Comprova l'accés	4
3. Estructura de directoris i fitxers	5
4. Gestió del servei	5
5. Gestió de mòduls	6
Mòduls més utilitzats	6
6. Configuració dels Virtual Hosts	7
6.1. Prepara el document root	7
6.2. Crea el fitxer de Virtual Host	7
6.3. Activa el lloc i desactiva el lloc per defecte	8
6.4. Resol el nom localment (proves sense DNS)	8
6.5. Discriminació del trànsit entrant	8
7. Opcions de configuració habituals	9
7.1. Directiva Options	9
7.2. Directiva AllowOverride	9
7.3. Control d'accés (Require)	9
7.4. Pàgines d'índex i tipus MIME	9
7.5. Canvia el port d'escolta	10
7.6. Compresió de contingut (gzip)	10
7.7. Capçaleres de seguretat (mod_headers)	10
7.8. Ajusta l'MPM event	10
8. Autenticació i control d'accés	11
8.1. Autenticació Basic	11
8.2. Autenticació Digest	12
8.3. Comparativa Basic vs. Digest	13
8.4. Control d'accés combinat (autenticació + IP)	13
8.5. Verifica l'accés dels usuaris	13
9. Execució de codi al servidor vs. al client	14

9.1. Prova l'execució de codi al servidor (PHP)	14
9.2. Provar l'execució de codi al client (JavaScript)	15
10. HTTPS i certificats digitals	15
10.1. Certificat gratuït amb Let's Encrypt (Certbot)	15
10.2. Certificat autosignat (entorns de pràctiques sense domini públic)	16
11. Monitoratge, registres i anàlisi	17
11.1. Monitoratge en temps real amb mod_status	17
11.2. Registres (logs)	17
11.3. Anàlisi de logs per a estadístiques i incidències	18
Problemes habituals	18
12. Resum d'ordres	19
Índex de figures	20

Fonaments i protocols

Un servidor web és un programa que escolta peticions sota el protocol **HTTP** (*HyperText Transfer Protocol*) o la seva variant xifrada **HTTPS** (HTTP sobre TLS/SSL), i hi respon enviant el recurs sol·licitat (una pàgina, una imatge, un JSON generat dinàmicament, etc.).

Funcionament bàsic d'una petició HTTP

1. El client (navegador) obre una connexió **TCP** contra el servidor, normalment al port **80** (HTTP) o **443** (HTTPS).
2. El client envia una petició amb un **mètode** (GET, POST, PUT, DELETE...), una **ruta** (/index.html) i unes **capçaleres** (Host, User-Agent, Accept...).
3. El servidor processa la petició: localitza el recurs, l'executa (si és codi dinàmic) o simplement el llegeix del disc.
4. El servidor respon amb un **codi d'estat** (200 OK, 301 Moved Permanently, 403 Forbidden, 404 Not Found, 500 Internal Server Error...), capçaleres de resposta i, opcionalment, un cos (el contingut).

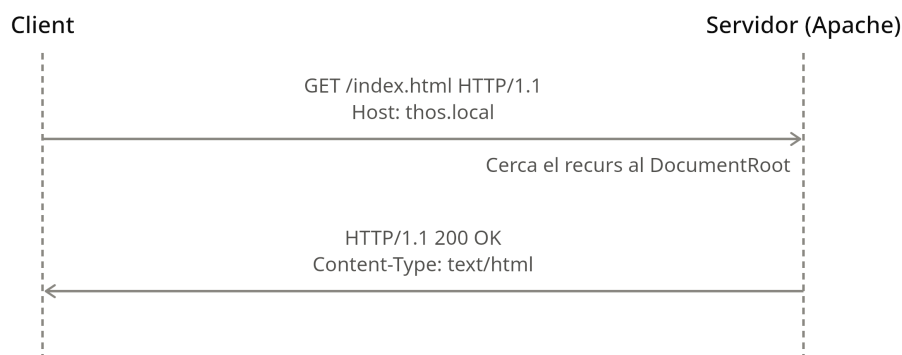


Figura 1: Seqüència HTTP

Capçalera Host i Virtual Hosts

Des d'HTTP/1.1, totes les peticions inclouen la capçalera **Host**, que indica a quin nom de domini es dirigeix la petició. Això és el que permet a un mateix servidor (amb una única IP) distingir entre diversos llocs web: Apache llegeix aquesta capçalera i la compara amb els `ServerName/ServerAlias` definits a cada `VirtualHost` per decidir quina configuració aplicar (vegeu la secció 6).

HTTPS i TLS

HTTPS afegeix una capa de xifratge **TLS** (*Transport Layer Security*, successor de SSL) entre TCP i HTTP. Garanteix:

- **Confidencialitat:** el contingut de la comunicació no es pot llegir si s'intercepta.
- **Integritat:** les dades no es poden modificar pel camí sense detectar-ho.
- **Autenticitat:** el certificat digital del servidor confirma que es parla amb el domini correcte (vegeu la secció 8).

Documentació de referència

- Documentació oficial d'Apache HTTP Server: <https://httpd.apache.org/docs/2.4/>
- Especificació HTTP/1.1 (RFC 9110-9112): <https://httpwg.org/specs/>
- Mozilla Developer Network (MDN) sobre HTTP: <https://developer.mozilla.org/ca/docs/Web/HTTP>

1. Què és Apache Web Server

Apache HTTP Server, conegut simplement com a **Apache**, és un servidor web de codi obert mantingut per l'Apache Software Foundation. És un dels servidors web més utilitzats del món, juntament amb Nginx, i s'encarrega de rebre peticions HTTP/HTTPS dels navegadors i clients, i respondre-hi servint pàgines web, fitxers o redirigint les peticions a aplicacions dinàmiques (PHP, Python, etc.).

Característiques principals:

- **Multiplataforma:** funciona a Linux, Windows, macOS i altres sistemes Unix.
- **Modular:** el seu comportament s'amplia carregant o descarregant mòduls (SSL, reescriptura d'URL, autenticació, compressió, etc.) sense recompilar el binari.
- **Multi-site (Virtual Hosts):** un únic servidor pot allotjar diversos llocs web independents, diferenciats per nom de domini, IP o port.
- **Suport per a llenguatges dinàmics:** mitjançant mòduls com `mod_php` o proxys cap a PHP-FPM, Python (WSGI) o aplicacions Node.js.
- **Configuració basada en fitxers de text** organitzats de manera molt clara als directoris `/etc/apache2/`.
- **Llicència Apache 2.0**, de codi obert.

Arquitectura de processament (MPM)

Apache pot gestionar les connexions concurrents de diferents maneres, segons el **MPM (Multi-Processing Module)** actiu:

MPM	Funcionament	Quan s'utilitza
prefork	Un procés per connexió, sense threads	Compatibilitat amb mòduls no thread-safe (p. ex. mod_php clàssic)
worker event	Processos amb múltiples threads Variant de worker, optimitzat per a connexions keep-alive inactives	Millor rendiment amb mòduls thread-safe Per defecte a Ubuntu 24.04 , millor rendiment en entorns d'alta concurrència

A Ubuntu, si s'instal·la libapache2-mod-php, el sistema canvia automàticament a prefork, ja que mod_php no és thread-safe.

Documentació oficial sobre els MPM: <https://httpd.apache.org/docs/2.4/mpm.html>

2. Instal·lació d'Apache a Ubuntu

2.1. Actualitza el sistema

Actualitza la llista de paquets:

```
sudo apt update
```

Actualitza els paquets instal·lats:

```
sudo apt upgrade
```

2.2. Instal·la el paquet apache2

```
sudo apt install apache2
```

2.3. Verifica la instal·lació

```
apache2 -v
```

Sortida esperada (la versió pot variar):

```
Server version: Apache/2.4.58 (Ubuntu)
Server built:   ...
```

2.4. Comprova que el servei està actiu

```
sudo systemctl status apache2
```

Apache s'inicia automàticament després de la instal·lació i queda habilitat per arrencar amb el sistema.

2.5. Obre el tallafocs (UFW)

Ubuntu registra perfils d'UFW per a Apache durant la instal·lació:

```
sudo ufw app list
```

```
Available applications:
Apache
Apache Full
Apache Secure
OpenSSH
```

Perfil	Ports oberts
Apache	80/tcp (només HTTP)
Apache Full	80/tcp i 443/tcp (HTTP i HTTPS)
Apache Secure	443/tcp (només HTTPS)

```
sudo ufw allow 'Apache Full'
sudo ufw status
```

IMPORTANT

Si l'accés al servidor és remot, assegura't d'haver permès OpenSSH (`sudo ufw allow OpenSSH`) abans d'activar UFW amb `sudo ufw enable`.

2.6. Comprova l'accés

Obre un navegador i visita `http://IP_DEL_SERVIDOR/`. Hauria d'aparèixer la pàgina per defecte d'Apache ("Apache2 Ubuntu Default Page").

Per esbrinar la IP del servidor:

```
hostname -I
```

- Documentació oficial: <https://httpd.apache.org/docs/2.4/install.html>
- Guia d'Ubuntu: <https://ubuntu.com/server/docs/how-to/web-services/install-apache2/>

3. Estructura de directoris i fitxers

```
/etc/apache2/
├─ apache2.conf          # Fitxer de configuració principal
├─ ports.conf            # Directives Listen (ports d'escolta)
├─ envvars               # Variables d'entorn del servei
├─ magic                 # Dades per a la identificació de tipus MIME
├─ conf-available/       # Fragments de configuració disponibles
├─ conf-enabled/         # Fragments actius (enllaços simbòlics a
└─ ↪ conf-available/)
├─ mods-available/       # Mòduls disponibles
├─ mods-enabled/         # Mòduls actius (enllaços simbòlics a
└─ ↪ mods-available/)
├─ sites-available/      # Definicions de Virtual Hosts disponibles
├─ sites-enabled/        # Virtual Hosts actius (enllaços simbòlics
└─ ↪ a sites-available/)
```

El patró *-available/ vs. *-enabled/ permet tenir configuracions preparades sense activar-les. S'activen o desactiven mitjançant enllaços simbòlics gestionats amb eines pròpies (no cal crear-los manualment).

Altres ubicacions importants:

Ruta	Contingut
/var/www/html/	Document root per defecte (arrels dels llocs web)
/var/log/apache2/access.log	Registre de peticions
/var/log/apache2/error.log	Registre d'errors
/usr/sbin/apache2	Binari del servidor
/usr/sbin/apache2ctl	Eina de control del servei

4. Gestió del servei

```
# Inicia Apache
sudo systemctl start apache2

# Atura Apache
sudo systemctl stop apache2

# Reinicia (talla totes les connexions actives)
sudo systemctl restart apache2

# Recarrega la configuració sense tallar connexions
sudo systemctl reload apache2

# Habilita l'arrencada automàtica amb el sistema
sudo systemctl enable apache2

# Deshabilita l'arrencada automàtica
sudo systemctl disable apache2
```



```
# Comprova la sintaxi de la configuració abans d'aplicar canvis
sudo apache2ctl configtest
```

Sortida esperada de configtest si tot és correcte:

```
Syntax OK
```

CONSELL

És un bon costum executar sempre `apache2ctl configtest` abans de reload o restart per evitar deixar el servei caigut per un error de sintaxi.

5. Gestió de mòduls

Apache utilitza les eines `a2enmod` i `a2dismod` per activar i desactivar mòduls (crea/elimina els enllaços simbòlics entre `mods-available/` i `mods-enabled/`).

```
# Llista mòduls actius
apache2ctl -M

# Activa un mòdul
sudo a2enmod rewrite

# Desactiva un mòdul
sudo a2dismod status

# Recarrega després d'activar o desactivar mòduls
sudo systemctl restart apache2
```

Mòduls més utilitzats

Mòdul	Funció
<code>mod_rewrite</code>	Reescriptura d'URL (URL amigables, redireccions)
<code>mod_ssl</code>	Suport per a HTTPS/TLS
<code>mod_headers</code>	Manipulació de capçaleres HTTP
<code>mod_deflate</code>	Compressió de contingut (gzip)
<code>mod_proxy / mod_proxy_fcgi</code>	Proxy invers, útil per a PHP-FPM o backends d'aplicació
<code>mod_security2</code>	Tallafores d'aplicacions web (WAF)
<code>mod_status</code>	Pàgina d'estat del servidor en temps real
<code>mod_userdir</code>	Permet servir contingut des de <code>~usuari/public_html</code>

Índex complet de mòduls: <https://httpd.apache.org/docs/2.4/mod/>

6. Configuració dels Virtual Hosts

Els Virtual Hosts permeten que un mateix servidor allotgi múltiples llocs web, diferenciats per nom (ServerName), IP o port.

6.1. Prepara el document root

```
sudo mkdir -p /var/www/thos.local/public_html
sudo chown -R $USER:$USER /var/www/thos.local/public_html
sudo chmod -R 755 /var/www/thos.local
```

Crea una pàgina de prova:

```
nano /var/www/thos.local/public_html/index.html
```

```
<!DOCTYPE html>
<html>
  <head><title>Benvingut a thos.local</title></head>
  <body><h1>Funciona! Aquest és el lloc thos.local</h1></body>
</html>
```

6.2. Crea el fitxer de Virtual Host

```
sudo nano /etc/apache2/sites-available/thos.local.conf
```

```
<VirtualHost *:80>
  ServerAdmin webmaster@thos.local
  ServerName thos.local
  ServerAlias www.thos.local

  DocumentRoot /var/www/thos.local/public_html

  <Directory /var/www/thos.local/public_html>
    Options -Indexes +FollowSymLinks
    AllowOverride All
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/thos.local_error.log
  CustomLog ${APACHE_LOG_DIR}/thos.local_access.log combined
</VirtualHost>
```

6.3. Activa el lloc i desactiva el lloc per defecte

```
sudo a2ensite thos.local.conf
sudo a2dissite 000-default.conf
sudo apache2ctl configtest
sudo systemctl reload apache2
```

6.4. Resol el nom localment (proves sense DNS)

```
sudo nano /etc/hosts
```

```
127.0.0.1 thos.local www.thos.local
```

6.5. Discriminació del trànsit entrant

Apache pot identificar a quin lloc virtual ha d'enviar cada petició mitjançant tres mecanismes, que es poden combinar:

Mecanisme	Com funciona	Exemple
Per nom (<i>name-based</i>)	Apache llegeix la capçalera HTTP Host: de la petició i la compara amb els ServerName/ServerAlias de cada VirtualHost que escolta al mateix IP: port	Diversos dominis sobre la mateixa IP i port 80
Per IP (<i>IP-based</i>)	Cada lloc virtual queda associat a una adreça IP diferent del servidor (<VirtualHost 192.168.1.10:80>)	El servidor té diverses IP/interfícies de xarxa assignades
Per port	Cada lloc escolta a un port TCP diferent (<VirtualHost *:8080>)	Aplicacions internes que no han d'exposar-se al port 80/443

Per consultar quins llocs virtuals té carregats Apache i amb quina configuració de Host/IP/port resol cadascun:

```
sudo apache2ctl -S
```

Sortida d'exemple:

```
VirtualHost configuration:
*:80 thos.local
↪ (/etc/apache2/sites-enabled/thos.local.conf:1)
  alias www.thos.local
*:80 is a NameVirtualHost
  default server thos.local
↪ (/etc/apache2/sites-enabled/thos.local.conf:1)
```

Aquesta ordre és la manera més ràpida de verificar quin VirtualHost respondrà a una petició concreta abans de fer proves amb el navegador.

Documentació oficial sobre Virtual Hosts: <https://httpd.apache.org/docs/2.4/vhosts/>

7. Opcions de configuració habituals

7.1. Directiva Options

Controla comportaments dins d'un <Directory>:

```
Options -Indexes          # Desactiva el llistat de directoris
Options +FollowSymLinks    # Permet seguir enllaços simbòlics
Options +Includes          # Permet Server Side Includes (SSI)
Options +ExecCGI           # Permet l'execució de scripts CGI
```

7.2. Directiva AllowOverride

Determina quines directives es poden sobreescriure mitjançant fitxers .htaccess:

```
AllowOverride All          # Permet totes les directives
AllowOverride None         # Ignora els fitxers .htaccess (millor
↳ rendiment)
AllowOverride FileInfo Options # Només permet aquests grups de
↳ directives
```

7.3. Control d'accés (Require)

```
Require all granted        # Accés obert a tothom
Require all denied         # Bloqueja l'accés a tothom
Require ip 192.168.1.0/24   # Només des d'aquesta subxarxa
Require not ip 10.0.0.5     # Bloqueja una IP concreta
```

7.4. Pàgines d'índex i tipus MIME

```
DirectoryIndex index.html index.php
AddType application/json .json
AddDefaultCharset UTF-8
```

7.5. Canvia el port d'escolta

A /etc/apache2/ports.conf:

```
Listen 8080
```

I al VirtualHost corresponent:

```
<VirtualHost *:8080>
    ...
</VirtualHost>
```

7.6. Compressió de contingut (gzip)

```
sudo a2enmod deflate
```

```
<IfModule mod_deflate.c>
    AddOutputFilterByType DEFLATE text/html text/css text/javascript
    ↪ application/javascript application/json
</IfModule>
```

7.7. Capçaleres de seguretat (mod_headers)

```
sudo a2enmod headers
```

```
Header always set X-Content-Type-Options "nosniff"
Header always set X-Frame-Options "SAMEORIGIN"
Header always set Strict-Transport-Security "max-age=63072000"
```

7.8. Ajusta l'MPM event

A /etc/apache2/mods-available/mpm_event.conf:

```
<IfModule mpm_event_module>
    StartServers      4
    MinSpareThreads   75
    MaxSpareThreads   250
    ThreadLimit        64
    ThreadsPerChild    25
    MaxRequestWorkers  400
    MaxConnectionsPerChild 10000
</IfModule>
```

- Llista completa de directives: <https://httpd.apache.org/docs/2.4/mod/core.html>
- mod_deflate: https://httpd.apache.org/docs/2.4/mod/mod_deflate.html
- mod_headers: https://httpd.apache.org/docs/2.4/mod/mod_headers.html

8. Autenticació i control d'accés

Apache permet restringir l'accés a determinats recursos exigint que l'usuari s'identifiqui. Per defecte, l'accés a un lloc web és **anònim** (no cal identificar-se); per fer-lo **autènticat**, cal protegir el <Directory>, <Location> o <Files> corresponent amb una de les directives AuthType.

Apache ofereix principalment dos mètodes d'autenticació HTTP natius: **Basic** i **Digest**. Tots dos es poden combinar amb diferents *providers* (fitxer de text, base de dades, LDAP...), però el més habitual als entorns docents és el proveïdor de fitxer (mod_authn_file).

8.1. Autenticació Basic

Amb Basic, el navegador envia l'usuari i la contrasenya codificats en Base64 (no xifrats) a cada petició. **No és segura si no es combina amb HTTPS**, ja que Base64 no és xifrat i les credencials viatgen pràcticament en text clar.

Mòduls necessaris (sol venir actiu per defecte a Ubuntu):

```
sudo a2enmod auth_basic authn_file authz_user
sudo systemctl reload apache2
```

Crea el fitxer de contrasenyes amb htpasswd (inclòs al paquet apache2-utils):

```
sudo apt install apache2-utils

# Primer usuari: crea el fitxer (-c)
sudo htpasswd -c /etc/apache2/.htpasswd alumne1

# Usuaris addicionals: sense -c, per no esborrar el fitxer
sudo htpasswd /etc/apache2/.htpasswd alumne2
```

Protegeix un directori al VirtualHost o en un bloc <Directory>:

```
<Directory /var/www/thos.local/public_html/privat>
    AuthType Basic
    AuthName "Àrea restringida"
    AuthUserFile /etc/apache2/.htpasswd
    Require valid-user
</Directory>
```

També es pot definir des d'un fitxer .htaccess (cal AllowOverride AuthConfig o AllowOverride All al <Directory> pare):

```
AuthType Basic
AuthName "Àrea restringida"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
```

```
sudo apache2ctl configtest
sudo systemctl reload apache2
```

8.2. Autenticació Digest

Digest evita enviar la contrasenya en clar: el navegador calcula un *hash* MD5 a partir de la contrasenya, el *realm*, el mètode HTTP i un valor aleatori (*nonce*) que proposa el servidor, i només envia aquest resum (*digest*).

Mòdul necessari:

```
sudo a2enmod auth_digest
sudo systemctl reload apache2
```

Crea el fitxer de contrasenyes amb `htdigest`. A diferència de `htpasswd`, cal indicar el **realm** (ha de coincidir exactament amb `AuthName`):

```
# Format: htdigest -c fitxer "realm" usuari
sudo htdigest -c /etc/apache2/.htdigest "Àrea restringida" alumne1
sudo htdigest /etc/apache2/.htdigest "Àrea restringida" alumne2
```

Configuració del directori protegit:

```
<Directory /var/www/htos.local/public_html/privat>
    AuthType Digest
    AuthName "Àrea restringida"
    AuthDigestProvider file
    AuthDigestDomain /privat/
    AuthUserFile /etc/apache2/.htdigest
    Require valid-user
</Directory>
```

```
sudo apache2ctl configtest
sudo systemctl reload apache2
```

AVÍS

Tot i que Digest sembla més segur que Basic perquè no envia la contrasenya en clar, la documentació oficial d'Apache adverteix que ja no compleix aquell objectiu de disseny: un atacant *man-in-the-middle* pot forçar fàcilment el navegador a baixar a Basic, i l'algorisme MD5 és prou ràpid perquè un atacant passiu pugui obtenir la contrasenya per força bruta amb maquinari gràfic actual. A la pràctica, **es recomana sempre Basic combinat amb**

HTTPS en lloc de Digest; aquest darrer es manté sobretot per compatibilitat amb sistemes antics.

8.3. Comparativa Basic vs. Digest

	Basic	Digest
Credencials a la xarxa	Base64 (no xifrat)	Hash MD5 del <i>digest</i>
Eina per gestionar usuaris	htpasswd	htdigest
Necessita el realm exacte	No	Sí (AuthName = realm)
Recomanació actual	Usar sempre amb HTTPS	Evitar en sistemes nous; només per compatibilitat
Suport de navegadors	Universal	Universal, però alguns clients antics fallen

8.4. Control d'accés combinat (autenticació + IP)

Es pot exigir que es compleixin totes les condicions (AND) o només una (OR):

```
# Cal autenticar-se i venir de la xarxa del centre
<RequireAll>
    Require valid-user
    Require ip 192.168.0.0/24
</RequireAll>

# N'hi ha prou amb una de les dues condicions
<RequireAny>
    Require valid-user
    Require ip 127.0.0.1
</RequireAny>
```

8.5. Verifica l'accés dels usuaris

Per comprovar que l'autenticació funciona correctament:

```
# Petició sense credencials: ha de retornar 401 Unauthorized
curl -I http://thos.local/privat/

# Petició amb credencials Basic
curl -I -u alumne1:contrasenya http://thos.local/privat/

# Petició amb credencials Digest
curl -I --digest -u alumne1:contrasenya http://thos.local/privat/
```

Els intents fallits i correctes d'autenticació també queden registrats al log d'errors i d'accés (vegeu la secció 11), útils per detectar intents d'accés no autoritzats.

Documentació oficial:

- `mod_auth_basic` https://httpd.apache.org/docs/2.4/mod/mod_auth_basic.html
- `mod_auth_digest` https://httpd.apache.org/docs/2.4/mod/mod_auth_digest.html
- `htpasswd` <https://httpd.apache.org/docs/2.4/programs/htpasswd.html>
- `htdigest` <https://httpd.apache.org/docs/2.4/programs/htdigest.html>

9. Execució de codi al servidor vs. al client

És important diferenciar on s'executa el codi d'una pàgina web:

	Codi al servidor (<i>server-side</i>)	Codi al client (<i>client-side</i>)
On s'executa	Apache (o el llenguatge que invoca: PHP, Python...)	Al navegador de l'usuari
Llenguatges típics	PHP, Python (WSGI), Node.js darrere d'un proxy	JavaScript, HTML, CSS
Què veu el client	Només el resultat (HTML/JSON ja generat)	El codi font complet, visible des del navegador
Exemple	Generar una pàgina amb dades d'una base de dades	Validar un formulari abans d'enviar-lo, animacions

9.1. Prova l'execució de codi al servidor (PHP)

```
sudo apt install php libapache2-mod-php
sudo systemctl restart apache2
```

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

Si es visita `http://IP_DEL_SERVIDOR/info.php` i es mostra la pàgina d'informació de PHP (no el codi font), el servidor està executant correctament codi PHP. Si, en canvi, el navegador mostra el text `<?php phpinfo(); ?>` literal, vol dir que Apache no ha interpretat el fitxer com a codi PHP (mòdul no actiu o extensió no reconeguda).

IMPORTANT

Recorda eliminar `info.php` un cop fetes les proves, ja que exposa informació detallada del servidor.

9.2. Provar l'execució de codi al client (JavaScript)

```
<!DOCTYPE html>
<html>
<head><title>Prova JS</title></head>
<body>
  <script>
    document.write("Aquest text l'ha generat JavaScript al navegador,
    ↩ no Apache.");
  </script>
</body>
</html>
```

En aquest cas, Apache només serveix el fitxer `.html` tal qual; és el navegador qui interpreta i executa l'etiqueta `<script>`. Es pot comprovar amb “Veure codi font” (Ctrl+U): el codi JavaScript hi apareixerà sencer, a diferència del PHP, que mai s'envia al client.

10. HTTPS i certificats digitals

Per assegurar les comunicacions entre client i servidor, cal xifrar el trànsit amb TLS. Apache pot fer-ho amb `mod_ssl` i un certificat digital, emès per una **Autoritat de Certificació (CA)** o autosignat.

10.1. Certificat gratuït amb Let's Encrypt (Certbot)

```
sudo apt install certbot python3-certbot-apache
sudo certbot --apache -d thos.local -d www.thos.local
```

Certbot detecta automàticament el `VirtualHost` corresponent, obté el certificat, configura `mod_ssl` i crea la redirecció de HTTP a HTTPS si es confirma durant l'assistent.

Renovació automàtica (Certbot instal·la un temporitzador `systemd` per defecte):

```
sudo certbot renew --dry-run
systemctl list-timers | grep certbot
```

10.2. Certificat autosignat (entorns de pràctiques sense domini públic)

Útil per a pràctiques en xarxa local, on no es disposa d'un domini real ni accés des d'Internet:

```
sudo a2enmod ssl
sudo mkdir -p /etc/apache2/ssl

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/apache2/ssl/thos.key \
  -out /etc/apache2/ssl/thos.crt
```

```
<VirtualHost *:443>
  ServerName thos.local
  DocumentRoot /var/www/thos.local/public_html

  SSLEngine on
  SSLCertificateFile /etc/apache2/ssl/thos.crt
  SSLCertificateKeyFile /etc/apache2/ssl/thos.key
</VirtualHost>
```

```
sudo a2ensite thos.local-ssl.conf
sudo apache2ctl configtest
sudo systemctl reload apache2
```

NOTA

Amb un certificat autosignat, els navegadors mostraran un avís de seguretat perquè no hi ha cap CA de confiança que l'avalí; és normal i acceptable en entorns de pràctiques.

mod_ssl: https://httpd.apache.org/docs/2.4/mod/mod_ssl.html

Certbot: <https://certbot.eff.org/>

Generador de configuracions SSL recomanades: <https://ssl-config.mozilla.org/>

11. Monitoratge, registres i anàlisi

11.1. Monitoratge en temps real amb mod_status

mod_status mostra una pàgina amb l'estat intern d'Apache: peticions per segon, treballadors actius/inactius, ús de CPU, etc. És útil per a proves de monitoratge i diagnosi de càrrega.

```
sudo a2enmod status
```

```
# /etc/apache2/mods-available/status.conf (ja ve preconfigurat;  
↪ restringir l'accés)  
<Location "/server-status">  
    SetHandler server-status  
    Require ip 127.0.0.1  
    Require ip 192.168.0.0/24  
</Location>
```

```
sudo systemctl reload apache2  
curl http://localhost/server-status
```

Per a una versió que es refresca automàticament: <http://localhost/server-status?refresh=5>.

11.2. Registres (logs)

```
# Peticions ateses correctament  
sudo tail -f /var/log/apache2/access.log  
  
# Errors i avisos  
sudo tail -f /var/log/apache2/error.log  
  
# Logs específics d'un Virtual Host  
sudo tail -f /var/log/apache2/this.local_error.log
```

Format de log personalitzat:

```
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\"  
↪ \"%{User-Agent}i\" %D" detallat  
CustomLog ${APACHE_LOG_DIR}/access.log detallat
```

11.3. Anàlisi de logs per a estadístiques i incidències

Amb eines de línia d'ordres:

```
# Top 10 IPs amb més peticions
awk '{print $1}' /var/log/apache2/access.log | sort | uniq -c | sort
↪ -rn | head -10

# Codis d'estat HTTP més freqüents
awk '{print $9}' /var/log/apache2/access.log | sort | uniq -c | sort
↪ -rn

# Pàgines amb més errors 404
awk '$9 == 404 {print $7}' /var/log/apache2/access.log | sort | uniq
↪ -c | sort -rn

# Peticions per hora
awk '{print $4}' /var/log/apache2/access.log | cut -d: -f2 | sort |
↪ uniq -c
```

Per a anàlisis més completes i informes visuals (gràfics, navegadors, geolocalització), s'utilitzen eines especialitzades com **GoAccess** (en temps real, terminal o HTML) o **AWStats**:

```
sudo apt install goaccess
sudo goaccess /var/log/apache2/access.log -o
↪ /var/www/html/report.html --log-format=COMBINED
```

mod_status: https://httpd.apache.org/docs/2.4/mod/mod_status.html

mod_log_config (format dels logs): https://httpd.apache.org/docs/2.4/mod/mod_log_config.html

GoAccess: <https://goaccess.io/>

Problemes habituals

Síntoma	Possible causa	Solució
AH00558: apache2: Could not reliably determine the server's fully qualified domain name	Falta ServerName global	Afegir ServerName localhost a /etc/apache2/apache2.conf
Error 403 Forbidden	Permisos del directori o Require massa restrictiu	Revisar chmod/chown i la directiva Require
Error 404 sobre rutes amigables	mod_rewrite no actiu o .htaccess ignorat	sudo a2enmod rewrite i AllowOverride All
El lloc nou no es mostra	Virtual host no activat o sites-enabled desactualitzat	sudo a2ensite + systemctl reload apache2
Syntax OK, però el lloc no respon	Conflicte de ports o Listen no definit	Revisar ports.conf i el VirtualHost

12. Resum d'ordres

Ordre	Descripció
<code>sudo apt install apache2</code>	Instal·la Apache
<code>sudo systemctl status/start/stop/restart/reload apache2</code>	Gestiona el servei
<code>sudo apache2ctl configtest</code>	Verifica la sintaxi de la configuració
<code>sudo apache2ctl -S</code>	Mostra com es resolen els Virtual Hosts
<code>apache2ctl -M</code>	Llista els mòduls actius
<code>sudo a2enmod <mòdul> / sudo a2dismod <mòdul></code>	Activa/desactiva un mòdul
<code>sudo a2ensite <lloc.conf> / sudo a2dissite <lloc.conf></code>	Activa/desactiva un Virtual Host
<code>sudo ufw allow 'Apache Full'</code>	Obre els ports HTTP i HTTPS al tallafocs
<code>sudo htpasswd -c fitxer usuari</code>	Crea/gestiona usuaris per a autenticació Basic
<code>sudo htdigest -c fitxer "realm" usuari</code>	Crea/gestiona usuaris per a autenticació Digest
<code>sudo certbot --apache -d domini</code>	Obté i configura un certificat TLS
<code>curl -I -u usuari:pass http://servidor/</code>	Prova l'accés autenticat d'un usuari
<code>curl http://localhost/server-status</code>	Consulta l'estat intern del servidor
<code>sudo tail -f /var/log/apache2/error.log</code>	Monitorea errors en temps real

Índex de figures

1	Seqüència HTTP	1
---	--------------------------	---

Versions d'aquest document

- [HTML](#)
- [PDF](#)
- [ODT](#)
- [MD](#)

[Domini Públic \(CC0\)](#)