



Documentació Apache

17/11/2021

Alex Garrido Hernandez

SMX 2n G1 M7

2021-2022

Índex

APARTAT 1 - El protocol HTTP	2
1. Servidor Web	2
2. Protocol HTTP	2
3. Eines de Terminal	2
APARTAT 2 - Configuració del servidor WEB	3
1. Instal·lació del servidor Apache2	3
2. Anàlisi del servei	3
3. Comandes per interactuar amb el servei	4
SystemCtl	4
Apache2Ctl	4
4. Fitxers de configuració del servei	5
Fitxer Conf	5
Fitxer Html	5
Fitxer Ports	6
5. Estructura i funcionament dels directoris available/enabled.	6
Sites-available	6
Sites-enabled	6
A2ensite / A2dissite	6
6. VirtualHost	7
Tipus (IP / Port / Nom)	7
Configuració mínima	7
Directives bàsiques	8
Directives Extres	9
8. Restricció d'accés per IP	10
9. Restricció d'accés per autenticació BASIC	12
10. Restricció d'accés per autenticació DIGEST	13
VERIFICACIÓ DIGEST I BÀSIC	14
11. Seguretat amb TLS/SSL	15
12. Arxius de Log	16
13. Errors	17
14. Altres - Wordpress	19
APARTAT 3 - Activitat Global	21
Esquema	21

APARTAT 1 - El protocol HTTP

1. Servidor Web

Un servidor web o servidor HTTP processa una aplicació una informació per mostrar una pàgina web al client.

2. Protocol HTTP

Per obtenir una comunicació entre client i servidor, utilitzem normalment el protocol HTTP, aquest protocol permet enviar preguntes i respostes entre el client i el servidor.

L'última versió de HTTP és la versió nº3.

Què és el KeepAlive?

És una tècnica que permet realitzar múltiples peticions HTTP sense tancar la connexió TCP.

Què és el PipieLining?

És una tècnica que permet enviar múltiples peticions HTTP sense esperar les respostes de les anteriors.

3. Eines de Terminal

GET

El mètode GET sol·licita una representació d'un recurs específic. Les peticions que fan servir el mètode GET només han de recuperar dades.

HEAD

El mètode HEAD demana una resposta idèntica a una petició GET, però sense el cos de la resposta.

CURL

És una comanda que serveix per verificar la interacció de molts protocols

APARTAT 2 - Configuració del servidor WEB

1. Instal·lació del servidor Apache2

Per instal·lar el paquet Apache, farem servir la comanda **<sudo apt install apache2>**.

Abans de tot haurem de fer un **sudo apt update**.

Per saber quina versió tenim d'Apache, utilitzarem la comanda següent: **sudo apachectl -v**.

```
user@debian:~$ sudo apachectl -v
Server version: Apache/2.4.48 (Debian)
Server built:   2021-08-12T11:51:47
user@debian:~$
```

Com podem veure, la versió que fem servir és la 2.4.48.

2. Anàlisi del servei

Apache2 treballa pel port 80. Normalment, Apache2 crear per defecte un usuari anomenat **www-data**.

A continuació, per veure qui executa el servei, farem la següent comanda **ps aux | grep apache**.

Podem veure que **www-data** és un usuari que ha creat apache alhora que l'hem instal·lat.

```
user@debian:~$ ps aux | grep apache
root      47449  0.0  0.0  6484  4600 ?        Ss   09:59   0:00 /usr/sbin/apache2 -k start
www-data  47450  0.0  0.1 1997552 12024 ?        Sl   09:59   0:00 /usr/sbin/apache2 -k start
www-data  47451  0.0  0.1 1997552 12032 ?        Sl   09:59   0:00 /usr/sbin/apache2 -k start
user      47999  0.0  0.0   6180   712 pts/0    S+   10:46   0:00 grep apache
```

Si fem un **nmap** a la nostra IP, podem veure que el port que utilitza el protocol HTTP és **80**.

```
user@alexlive:~$ nmap 192.168.8.21
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-26 11:33 UTC
Nmap scan report for 192.168.8.21
Host is up (0.00013s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 0.06 seconds
```

3. Comandes per interactuar amb el servei

SystemCtl

Per iniciar un el servei d'Apache2, podem utilitzar la comanda **sudo systemctl start apache2**. Amb aquesta comanda iniciarem el servei, però si variem la paraula "start" a qualsevol de les altres (stop, restart, reload).

La diferència entre restart i reload, és que **"reload"** recarrega els arxius de nou a la memòria cau i **"restart"** fa un reinici del servei sencer.

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-11-12 11:17:35 UTC; 32min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 46324 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 46335 (apache2)
    Tasks: 55 (limit: 9120)
   Memory: 21.2M
      CPU: 97ms
   CGroup: /system.slice/apache2.service
           └─46335 /usr/sbin/apache2 -k start
             └─46336 /usr/sbin/apache2 -k start
               └─46337 /usr/sbin/apache2 -k start
```

Apache2Ctl

Per executar una prova de sintaxi del fitxer de configuració podrem fer servir la comanda **apache2ctl -t**. Aquesta comanda analitza els fitxers de configuració i si tot està correcte et diu "Syntax OK" o en el cas d'error, informació detallada sobre l'error de sintaxi particular.

```
user@debian:/etc/apache2$ sudo apache2ctl -t
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
Syntax OK
user@debian:/etc/apache2$
```

Per executar una comprovació que els nostres VirtualHosts funcionen correctament.

Aquesta comanda s'anomena **apache2ctl -S**.

```
master@alexcl:/etc/apache2/sites-available$ apache2ctl -S
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 192.168.
lly to suppress this message
VirtualHost configuration:
*:80               is a NameVirtualHost
    default server 192.168.51.11 (/etc/apache2/sites-enabled/000-default.conf:1)
    port 80 namevhost 192.168.51.11 (/etc/apache2/sites-enabled/000-default.conf:1)
    port 80 namevhost www.m7proves.cat (/etc/apache2/sites-enabled/m7.conf:1)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
```

Per saber quins mòduls tenim carregats al nostre servidor web; farem servir la comanda **apache2ctl -M**. Dita comanda ens indicarà quins mòduls tenim compartits o iniciats.

```
user@debian:/etc/apache2$ sudo apache2ctl -M
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
Loaded Modules:
core_module (static)
so_module (static)
watchdog_module (static)
http_module (static)
log_config_module (static)
logio_module (static)
version_module (static)
unixd_module (static)
access_compat_module (shared)
```

4. Fitxers de configuració del servei

Fitxer Conf

Aquest fitxer serveix per configurar el nostre virtualhost. Aquí podrem trobar opcions com canviar el **DocumentRoot**, el nom del nostre servidor de noms... Aquest arxiu el trobarem a **/etc/apache2/sites-available/ ...**

És molt important fer un copy del fitxer base.

```
<VirtualHost *:8080>
    #ServerName www.example.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/vermell

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Fitxer Html

Aquest fitxer serveix per configurar la nostra pàgina web. Aquí podrem posar tota aquella informació que nosaltres vulguem que surti a la nostra pàgina web. Aquest arxiu el podrem trobar a **/var/www/ ...**

Normalment, s'anomena index.html, tot i que es pot canviar...

```
<h1> PAGINA VERMELL </h1>

<style>

    body {
        background-color: red;
    }
</style>
```

Fitxer Ports

Aquest fitxer és bastant **important** perquè si volem que el nostre Virtualhost escolti per un altre port, haurem d'escriure aquell port que nosaltres vulguem a aquest fitxer.

Aquest fitxer el trobarem a la carpeta base de apache2: **/etc/apache2/ports.conf**

```
Listen 80
Listen 8080
<IfModule ssl_module>
    Listen 443
</IfModule>
```

5. Estructura i funcionament dels directoris available/enabled.

Sites-available

En aquest directori trobarem aquells arxius de configuració que hem anomenat abans, on podrem configurar tots els nostres virtuals hosts.

```
master@alexcl:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf  m7.conf
master@alexcl:/etc/apache2/sites-available$
```

Sites-enabled

En aquest directori trobarem aquells arxius de configuració que hem activat amb la comanda **a2ensite ""**.

```
master@alexcl:/etc/apache2/sites-enabled$ ls
000-default.conf  m7.conf
master@alexcl:/etc/apache2/sites-enabled$
```

A2ensite / A2dissite

Aquestes dues comandes serveixen per habilitar la nostra pàgina web. La seva sintaxi seria **a2ensite "nom de l'arxiu .conf"**

És molt important saber que des d'aquesta comanda haurem de tornar a carregar la memòria cau amb la comanda **systemctl reload apache2**.

6. VirtualHost

Tipus (IP / Port / Nom)

Com podem veure, a partir d'aquest arxiu .conf sabem que el nostre virtualhost viatja pel port predeterminat.

També visualitzem que podem tenir vinculat un servidor de noms.

```
<VirtualHost *:80>
    ServerName www.smx2.cat
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

```
<VirtualHost *:8080>
    #ServerName www.example.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/vermell

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Configuració mínima

Per comprovar que el nostre arxiu .conf és correcte utilitzarem la comanda **apache2ctl -t**.

```
user@debian:/etc/apache2$ sudo apache2ctl -t
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the
'ServerName' directive globally to suppress this message
Syntax OK
user@debian:/etc/apache2$
```

Validarem que tot és correcte accedint a la pàgina des del navegador.



- Directives bàsiques

ServerName

Aquesta directiva serveix per vincular el nostre virtualhost amb un servidor de noms.

DocumentRoot

Aquesta directiva serveix per vincular aquell document o aquell directori on trobarem l'arxiu de la pàgina web (html)

ServerAdmin

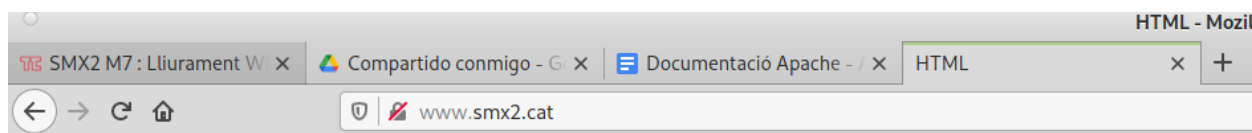
Aquesta directiva serveix per vincular un correu electrònic de l'administrador de la pàgina, tot i que mai el farem servir

ErrorLog

Aquesta directiva vincula un arxiu .log per guardar tots els errors que podrem tenir en el nostre virtualhost o la pàgina web.

ServerAlias

Aquesta directiva servirà per escurçar o donar un altre format de nom al nostre servidor web. Encara que sempre predominarà el nom principal.



ALEX GARRIDO HERNANDEZ

Esta página web es una página HTML válida.

Com podem comprovar, la directiva ServerName funciona correctament, igual que DocumentRoot, ja que ens carrega la pàgina que volem

- Directives Extres

ErrorDocument

Aquesta directiva serveix per vincular un document amb un error que nosaltres vulguem. Per exemple, si volem una pàgina personalitzada per l'error 404. Normalment, es vincula a la carpeta **/var/www**.

Redirect i Alias

Amb aquesta directiva podrem escurçar aquells directoris que nosaltres vulguem per tenir un àmbit més còmode amb la pàgina web.

Per exemple, si tenim un directori d'imatges anomenat `/var/www/M7/imatges`, amb la directiva el que podrem fer és que sigui `/imatges`.

Alias /imatges/ /var/www/M7/imatges

Si tenim una sol·licitud de la pàgina <https://www.M7.cat/imatges/pepito.jpg>, es tradueix automàticament a `/var/www/M7/imatges/pepito.jpg`

Options indexes

Aquesta directiva serveix per excloure l'índex automàtic de la pàgina.

Directory

Aquesta directiva fa referència a moltes més... Utilitzarem aquesta directiva per quan vulguem afegir Directives extremes al nostre arxiu `.conf` (**com per exemple una autenticació BASIC**). És molt important saber que haurem de vincular l'arxiu `.html` a aquesta directiva → **<Directory /var/www/ " " >**

DirectoryIndex

Aquesta directiva es fa servir per dir quin és l'arxiu web que volem fer servir per al nostre servidor.

8. Restricció d'accés per IP

Si nosaltres el que volem donar excepcions a una pàgina en concret, farem servir la directiva **Require**.

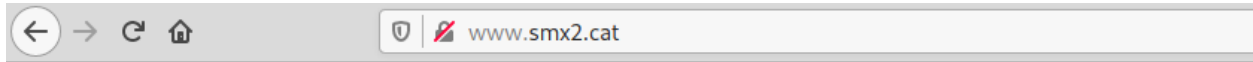
La funció d'aquesta directiva és administrar el nostre virtual, amb això podrem denegar o permetre l'entrada a diferents IP's.

Amb aquesta directiva el que estem fent és donar permís d'entrada a la nostra pàgina web a **tothom**, però el PC amb IP **192.168.8.20 no tindrà accés a la nostra pàgina**

```
user@alexglive: /var/log
File Edit View Search Terminal Help
GNU nano 5.4 m7.conf *
<VirtualHost *:80>
    #ServerName www.example.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/blau
    <Directory /var/www/blau>
        <RequireAny>
            Require ALL granted
            Require not ip 192.168.8.20
        </RequireAny>
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Com podem comprovar, el nostre company **Joel Martínez** no té accés a la nostra pàgina web.



Forbidden

You don't have permission to access this resource.

Apache/2.4.48 (Debian) Server at www.smx2.cat Port 80

```

user@debian: ~
File Edit View Search Terminal Help
  inet 192.168.8.20/16 brd 192.168.255.255 scope global dynamic noprefixroute
enp4s0
    valid_lft 4875sec preferred_lft 4875sec
  inet6 fe80::5c4c:23fd:8bca:1ae9/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
3: enp5s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state D
OWN group default qlen 1000
    link/ether 64:70:02:11:ad:ea brd ff:ff:ff:ff:ff:ff
4: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN grou
p default qlen 1000
    link/ether 5a:24:f3:ab:d5:37 brd ff:ff:ff:ff:ff:ff permaddr 50:46:5d:ae:3a:3
7
root@joellive:/home/user#

```

Recordem que ell ha canviat el seu arxiu `resolv.conf` i apunta al nostre servidor bind.

9. Restricció d'accés per autenticació BASIC

Per començar, aquest tipus d'autenticació no viatja de manera xifrada si ho diferenciem amb el tipus DIGEST. Utilitzem aquesta directiva quan nosaltres volem tenir un tipus d'identificació a l'hora d'entrar a la nostra pàgina web o a un subdirectori.

```

master@alexc2: /etc/apache2/sites-available
File Edit View Search Terminal Help
GNU nano 2.9.3 m7.conf

<VirtualHost *:80>
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/m7
    <Directory /var/www/m7>
        ErrorDocument 404 /error.html
        DirectoryIndex m7.html
        AuthType Basic
        AuthUserFile "/etc/apache2/fitxerdeusuaris
        Require valid-user
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

Com podem veure, tenim diferents paràmetres:

- **AuthType:** Serà el mètode d'identificació per accedir a la pàgina web, en aquest cas és BASIC.
- **AuthUserFile:** En la ubicació que anomenen serà l'arxiu on guardem els usuaris i les contrasenyes per accedir.
- **Require valid-user:** Aquesta directiva serveix per donar accés a aquelles persones que siguin vàlides dintre de l'arxiu d'identificació.

Si no posem aquesta directiva, tot i que sortí la pestanya d'identificació, si posem un usuari no vàlid, entrarà igualment.

MOLT IMPORTANT

Si utilitzem el mètode d'autenticació, haurem d'activar el mòdul

(a2enmod auth basic)

10. Restricció d'accés per autenticació DIGEST

En aquest altre tipus d'autenticació, una diferència clau que tenim és que la informació viatja encriptada.

És molt important saber que tots dos mètodes són fiables, però s'utilitza molt més el mètode DIGEST, ja que no tindrem problema amb la informació encriptada.

L'autenticació DIGEST porta les mateixes directives que el BASIC, això no obstant, el seu **AuthType** serà **DIGEST**.

MOLT IMPORTANT

Tant per a un mètode o altre, haurem de crear l'arxiu on volem que es guardi les contrasenyes i els usuaris.

Per això farem servir la comanda "htpasswd" (si volem fer DIGEST haurem d'afegir l'opció -C perquè la informació sigui encriptada)

L'exemple per la comanda seria la següent:

htpasswd -c /etc/apache2/contraseña pepito

Amb aquesta comanda ens sortirà que donem una contrasenya a aquest usuari, i ja tindrem l'usuari creat.

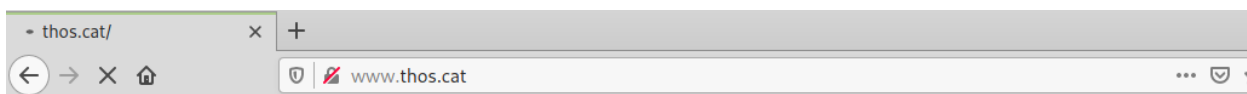
```
master@alexc2:/etc/apache2/sites-available$ sudo htpasswd -c /etc/apache2/contraseña pepito
New password:
Re-type new password:
Adding password for user pepito
master@alexc2:/etc/apache2/sites-available$ █
```

Podrem veure la nostra contrasenya encriptada, en aquest cas l'usuari es diu **"root"**.

```
root:hola:f152e74b77f46d66165d8bed4af73c8f
```

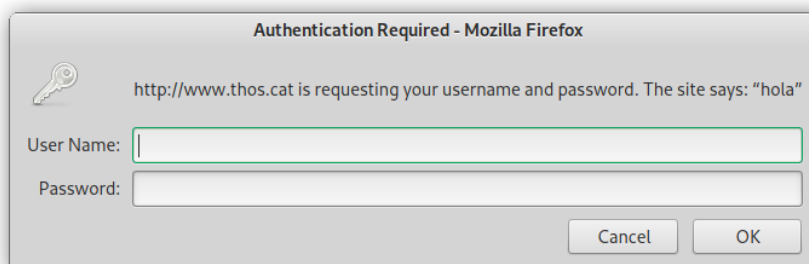
VERIFICACIÓ DIGEST I BÀSIC

Com podem veure, quan entrem a la pàgina no ens surt res. Però quan volem entrar a un directori diferent de la pàgina principal (**i tenim configurat l'accés digest o bàsic**) ens sortirà una petita finestra d'inici de sessió on allà posarem el nostre usuari i contrasenya creats anteriorment.



HOLA ESTA ES MI PAGINA WEB

[Visita la meva pagina d'Inventari](#)
[Visita la meva pagina de Wordpress](#)



11. Seguretat amb TLS/SSL

Per començar, la seguretat d'una pàgina web és molt important, per això existeix el port 443 per on viatja el protocol HTTPS.

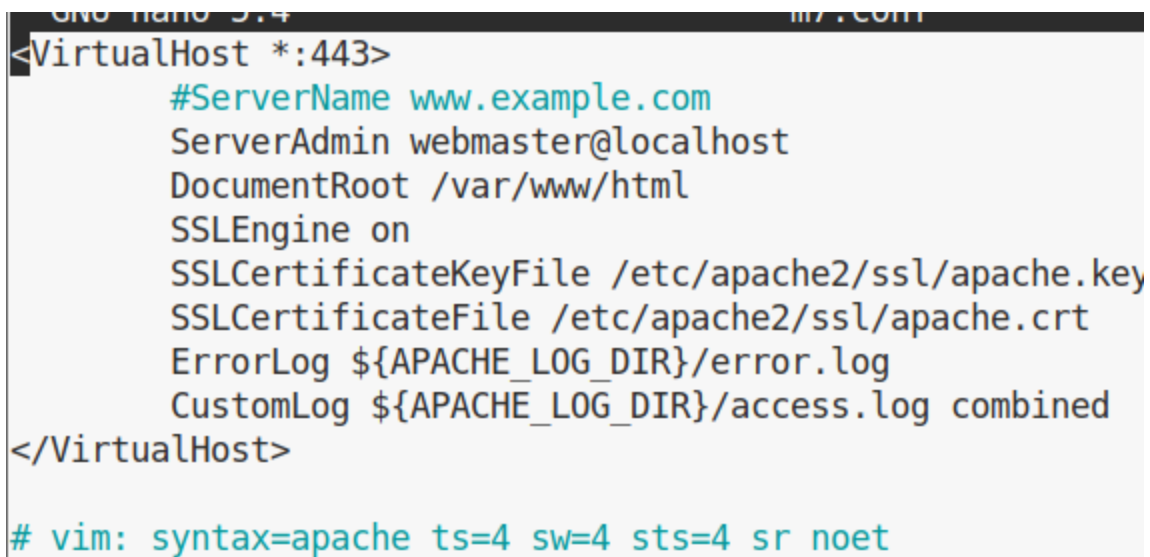
Per configurar aquest protocol farem el següent, primer de tot haurem d'activar el mod de HTTPS, la comanda és la següent: **"a2enmod ssl"**

A continuació, en el nostre fitxer `.conf` ficarem les següents directives:

SSLEngine on

SSLCertificateKeyFile /etc/apache2/ssl/apache.key

SSLCertificateFile /etc/apache2/ssl/apache.crt



```
<VirtualHost *:443>
    #ServerName www.example.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateKeyFile /etc/apache2/ssl/apache.key
    SSLCertificateFile /etc/apache2/ssl/apache.crt
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Apache.key i Apache.crt són els arxius que haurem de crear dins d'una carpeta (en aquest cas "ssl") per tenir el certificat de seguretat a la nostra web.

Per crear aquests dos arxius haurem de fer la següent comanda:

**openssl req -x509 -days 30 -nodes -newkey RSA:2048 -out ./ssl/apache.crt -keyout
./ssl/apache.key**

Després d'aquesta comanda ens demanarà una sèrie de dades

```
Country Name (2 letter code) [AU]:Es
State or Province Name (full name) [Some-State]:Barcelona
Locality Name (eg, city) []:Mataro
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Thosi
Organizational Unit Name (eg, section) []:Hola
Common Name (e.g. server FQDN or YOUR name) []:www.hola.cat
Email Address []:dasdasd@dasdas.com
```

Cada paraula de la comanda té un sentit:

- *x509* → És la versió del certificat.
- *RSA:2048* → És la longitud de la clau pública i privada. Contra més bits, té més seguretat.
- *-out* → Aquí indicarem quin és l'arxiu de sortida pel certificat → *./ssl/apache.crt*
- *-keyout* → Aquí indicarem quin és l'arxiu de clau pel certificat → *./ssl/apache.key*
- *-days* → serà la duració que té el certificat.

12. Arxius de Log

Apache2 té 2 arxius diferents de log, aquests dos arxius tenen diferents funcions.

El primer arxiu de log és **access.log**. Permet veure qui entra a la nostra pàgina web, des de quin sistema operatiu entra a la pàgina web, fins i tot amb quin navegador entra a la pàgina web.

El segon arxiu de log és **error.log**. Permet veure qualsevol error que tenim en el nostre servidor d'Apache2, tant si tenim un error de connexió, com si tenim un error d'arxius, o de sintaxi.

Aquests dos arxius els podem trobar a la carpeta */var/log/apache2*

```
File Edit View Search Terminal Help
root@alexglive:/var/log/apache2# ls
access.log  error.log  other_vhosts_access.log
root@alexglive:/var/log/apache2#
```

13. Errors

Error 1

Un error molt comú que podem tenir és no reiniciar la memòria cau d'Apache a l'hora d'activar el site o un mòdul

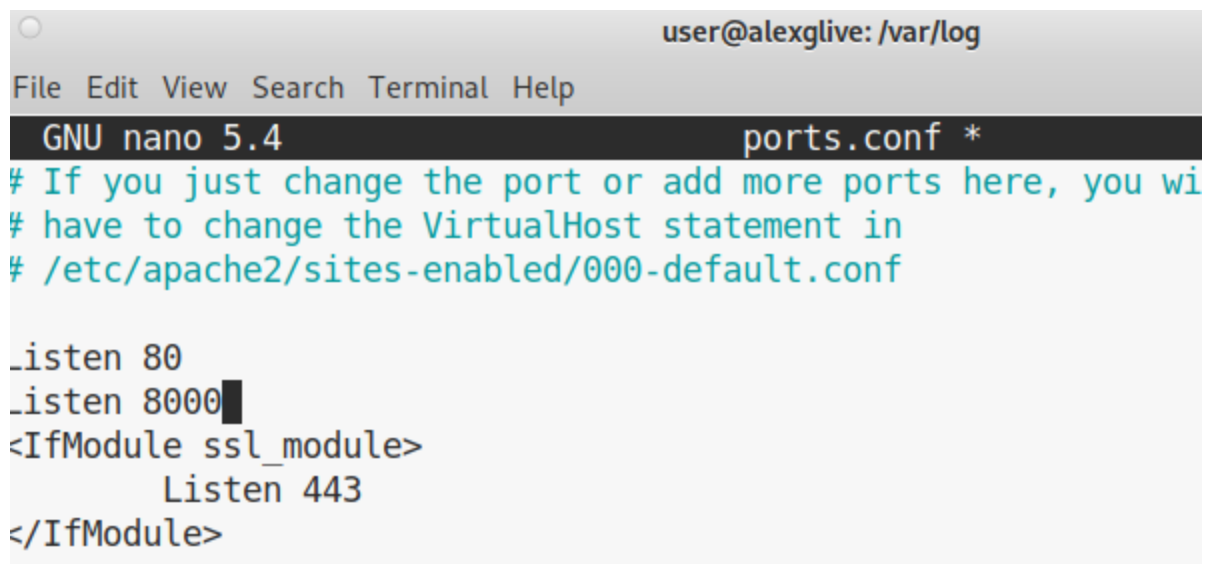
Per això existeix la comanda `a2ensite` o `a2enmod`, que serveix per activar aquestes pàgines web o mòduls que volem afegir.

Però existeix un altre variant que és per desactivar, anomenada `a2dissite` o `a2desmod`.

```
root@alexglive:/etc/apache2/sites-available# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@alexglive:/etc/apache2/sites-available# systemctl reload apache2
root@alexglive:/etc/apache2/sites-available# █
```

Error 2

El segon error que podem tenir és obrir els ports per on "escolta" el nostre servidor web. Això vol dir per on viatjarà la nostra informació de la pàgina web. Serà molt important anar al fitxer **ports.conf** per canviar el tipus de port que nosaltres vulguem, per defecte tindrem el port 80 obert.



```
user@alexglive: /var/log
File Edit View Search Terminal Help
GNU nano 5.4 ports.conf *
# If you just change the port or add more ports here, you wi
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 8000█
<IfModule ssl_module>
    Listen 443
</IfModule>
```

Si prestem atenció, podem veure que si activem el mòdul de SSL, el nostre servidor web escoltarà pel port 443 (HTTPS)

Error 3

Un altre error que podem tenir és el següent:

```
root@alexglive:/etc/apache2/sites-available# systemctl reload apache2
Job for apache2.service failed.
See "systemctl status apache2.service" and "journalctl -xe" for details.
root@alexglive:/etc/apache2/sites-available# █
```

Aquest error ens vol dir que no pot fer un “reload” al servei apache, ja que tenim un error de sintaxi o lògic en el nostre arxiu .conf

Com podem observar l’error que tenim és que la paraula “ALL” està en majúscules.

```
user@alexglive: ~
al Help
smx2.conf

smx2.cat
master@localhost
ar/www/smx2
r/www/smx2>
eAll>
  Require ALL granted
  Require not ip 192.168.8.20
.reAll>
```

Si canviem la paraula a minúscules, l’error no sortirà.

```
root@alexglive:/etc/apache2/sites-available# systemctl reload apache2
root@alexglive:/etc/apache2/sites-available# █
```

14. Altres - Wordpress

Si volem instal·lar un servei de WordPress, haurem de tenir en compte que necessitarem la comanda següent:

```
apt install php php-mysql php-curl php-gd php-mbstring php-xml php-xmlrpc  
php-soap php-intl php-zip -y
```

A continuació haurem de configurar un servidor de continguts. Per instal·lar maria-db farem el següent: **apt install mariadb-server**

Una vegada instal·lat haurem d'iniciar el servei i crear una database per configurar-la amb WordPress.

Per posar en marxa mariadb farem el següent:

```
systemctl start mariadb.service
```

```
● mariadb.service - MariaDB 10.5.12 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor pres
   Active: active (running) since Thu 2021-12-16 10:34:31 UTC; 27s ago
     Docs: man:mariabdb(8)
           https://mariadb.com/kb/en/library/systemd/
   Process: 48506 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /v
   Process: 48507 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_
   Process: 48509 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] &
   Process: 48572 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_
   Process: 48574 ExecStartPost=/etc/mysql/debian-start (code=exited, status=
 Main PID: 48556 (mariabdb)
    Status: "Taking your SQL requests now..."
     Tasks: 16 (limit: 9120)
    Memory: 93.9M
       CPU: 519ms
    CGroup: /system.slice/mariadb.service
            └─48556 /usr/sbin/mariabdb

Dec 16 10:34:31 alexglive /etc/mysql/debian-start[48579]: Processing databases
Dec 16 10:34:31 alexglive /etc/mysql/debian-start[48579]: information_schema
Dec 16 10:34:31 alexglive /etc/mysql/debian-start[48579]: mysql
Dec 16 10:34:31 alexglive /etc/mysql/debian-start[48579]: performance_schema
Dec 16 10:34:31 alexglive /etc/mysql/debian-start[48579]: Phase 6/7: Checking
lines 1-23
```

Després crearem la base de dades...

```
mariadb -u root
```

```
ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';
```

```
CREATE DATABASE wordpressdatabase;
```

```
MariaDB [(none)]> ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.000 sec)
```

```
MariaDB [(none)]> CREATE DATABASE wordpressdatabase;
Query OK, 1 row affected (0.000 sec)
```

A continuació haurem de descarregar WordPress:

wget https://es.wordpress.org/latest-es_ES.zip

unzip -q latest-es ES.zip -d /var/www/....

RECOMANACIÓ

Recomano fer la instal·lació de Wordpress a la carpeta /var/www/... per a després no moure la mateixa carpeta. Per això fem servir l'opció -d.

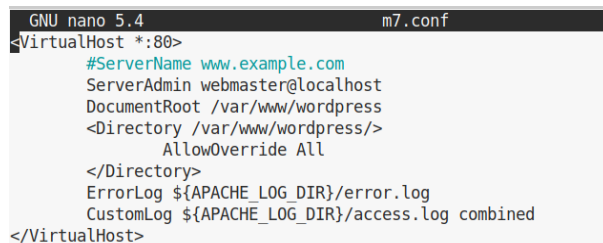
MOLT IMPORTANT

Al nostre arxiu .conf haurem de posar una directiva per donar permís a Wordpress i que pugui sobreescriure. La directiva és la següent

```
<Directory /var/www/wordpress>
```

```
    AllowOverride All
```

```
</Directory>
```



```
GNU nano 5.4                                m7.conf
<VirtualHost *:80>
    #ServerName www.example.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/wordpress
    <Directory /var/www/wordpress/>
        AllowOverride All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

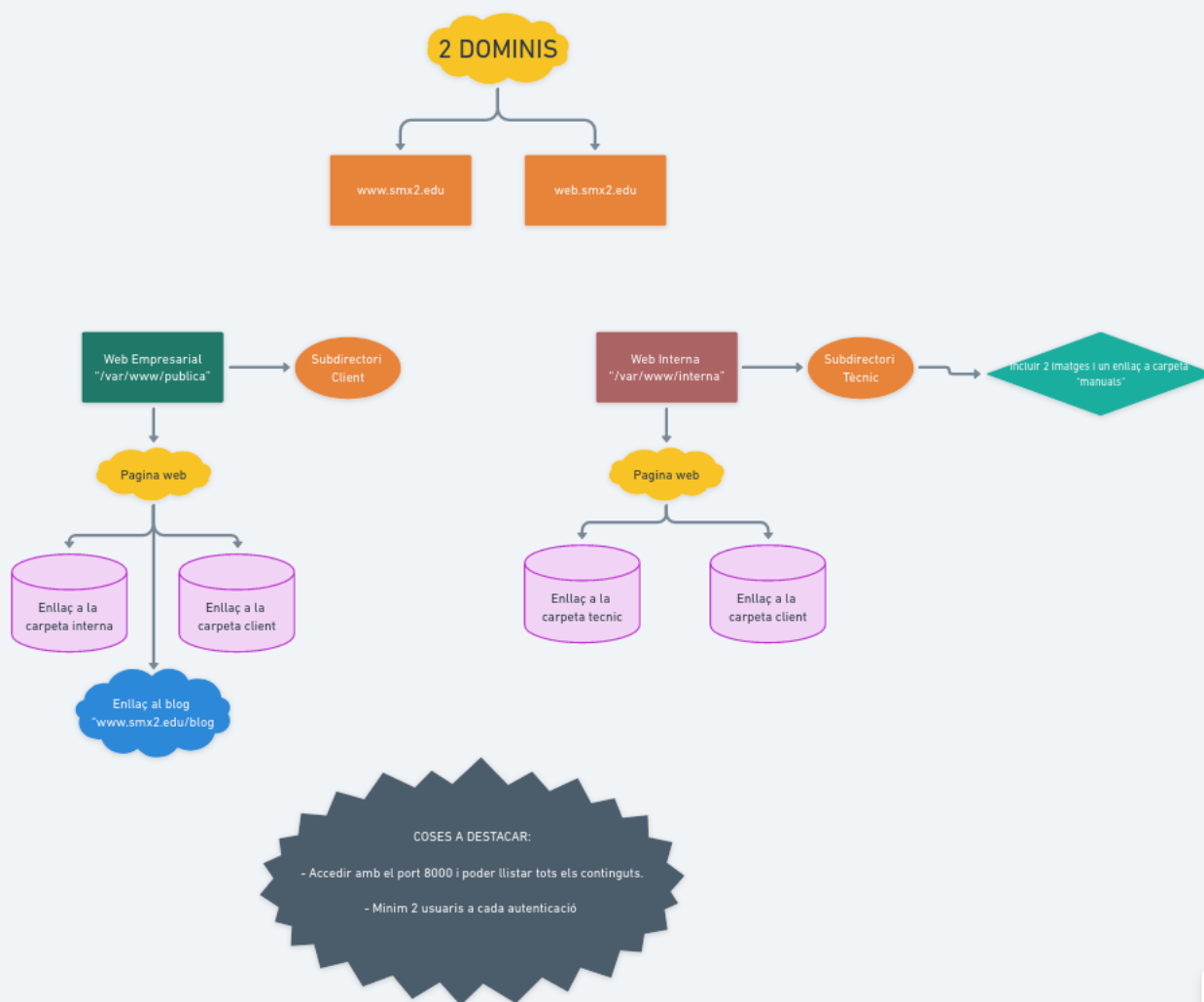
Per acabar, haurem de donar permís al mod de sobreescriptura i activarem el site.

a2enmod rewrite

a2ensite "arxiu .conf de la pagina"

APARTAT 3 - Activitat Global

1. Esquema



[LINK](#)