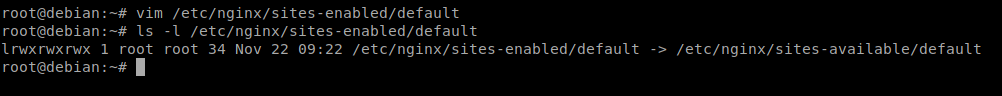
**CURIOSITATS:**

<http://info.cern.ch/>

**PANTALLAZOS / TEXT:**

**Fitxer de configuració, que és un soft link:**



Els dominis enables seran els dominis available.

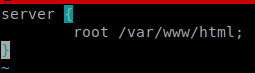
**Ens ‘carreguem’ l’arxiu:**

****

**Edició fitxer de configuració (/etc/nginx/sites-enabled/default):**

**root →** obligatori

**/var/www/html →** directori per defecte

****

**Reiniciem el servei:**

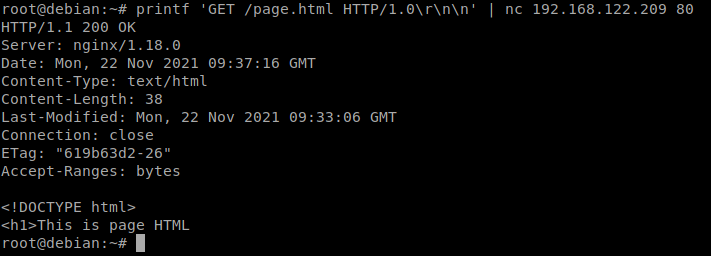
****

**Creem un fitxer dins del directori per defecte (/var/www/html):**

****

****

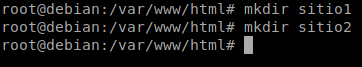
**Demanem el fitxer:**

****

**Demanem el fitxer v2 (f12 per + informació):**

****

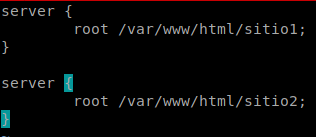
**Creem dos carpetes per tenir 2 llocs web diferenrt (dins de /var/www/html):**

****

**Creem un fitxer en cada directori:**

****

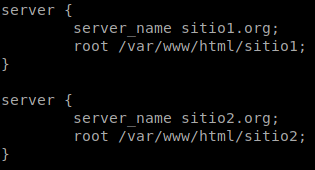
**Afegim els nous directoris web al fitxer /etc/nginx/sites-enabled/default:**

****

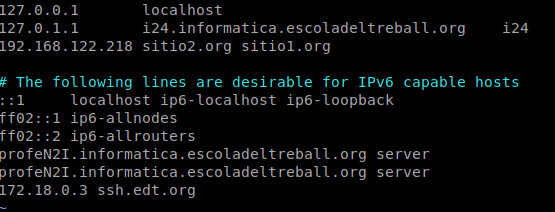
**Reiniciem el servei:**

****

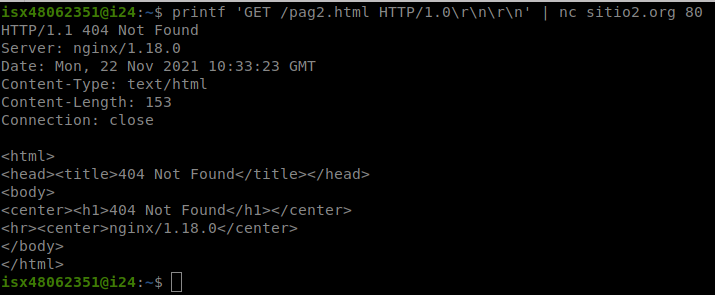
**Aclarim on ha de buscar la pàgina (/etc/nginx/sites-enabled/default):**

****

**Dins de /etc/hosts/ (localment), especifiquem la IP de la màquina i els llocs web:**

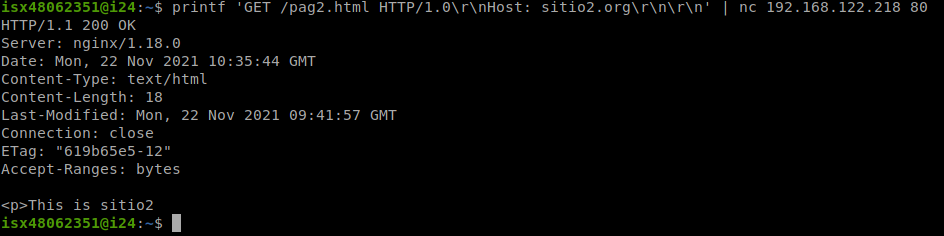
****

**‘nc’ adivina la IP encara que posem el hostname, però no trobarà el recurs:**

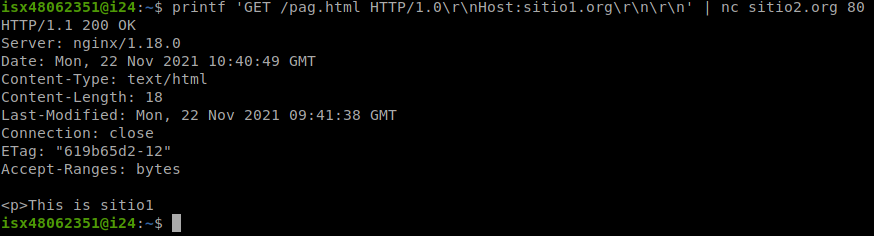
****

**Perquè ‘nc’ sapigue el servidor, hem d’especificar-li a la petició (Host: sitio2.org):**

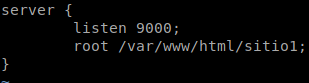
**Els últims ‘\r\n’ són perquè és OBLIGATORI una línea en blanc!:**

****

**Aquí ens retornarà la ‘pag.html’ del ‘sitio1’ perquè li estem especificant que el ‘Host’ és ‘sitio1’:**

****

**Deixem només un servidor (/etc/nginx/sites-enabled/default) i reiniciem:**

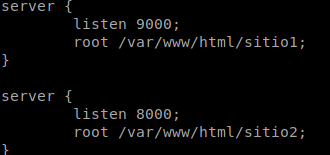
****

root@debian:/var/www/html# systemctl restart nginx.service

**Hem d’especificar el port perquè per defecte buscarà al port 80:**

****

**Especifiquem els ports pels que haurà d’escoltar als diferents ‘sitio1 i sitio2’:**

****

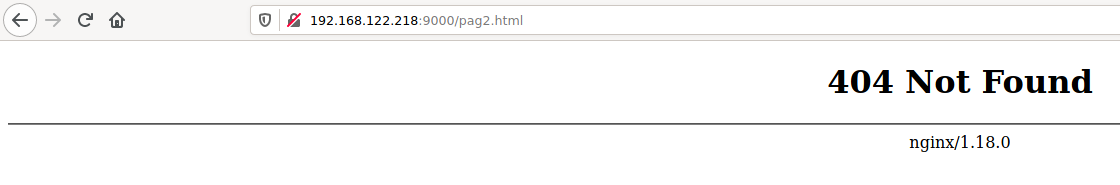
**Reiniciem:**

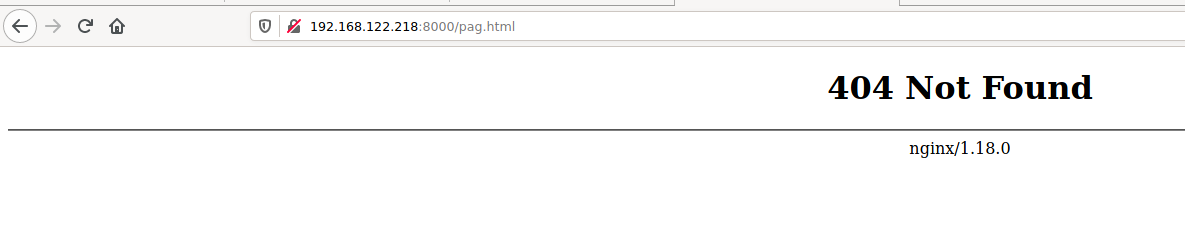
root@debian:/var/www/html# systemctl restart nginx.service

**Comprobem:**

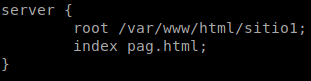
****

****

****

****

**Especifiquem el recurs per defecte:**

****

**Reiniciem:**

root@debian:/var/www/html# systemctl restart nginx.service

**Comprobem:**

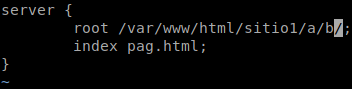
****

**Creem nous directoris i un nou fitxer:**

root@debian:/var/www/html# mkdir -p sitio1/a/b/c

root@debian:/var/www/html# vim a/b/c/tal.html

**Especifiquem al client que només haurà de buscar de la següent forma:**



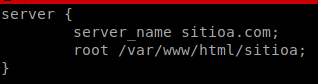
**Reiniciem:**

root@debian:/var/www/html# systemctl restart nginx.service

**Comprobem:**



**Canviar servername:**

****

**Escollim el que escolatarà firefox:** (curl mostra contingut web)

*Al server d'adalt té que haver un listen:7500 !!*****

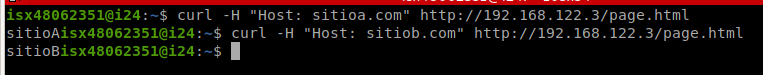
**-H –> Veure header (<h1>) de la pàgina: (ens donarà sempre “sitioA” perquè així ho tenim definit:**

****

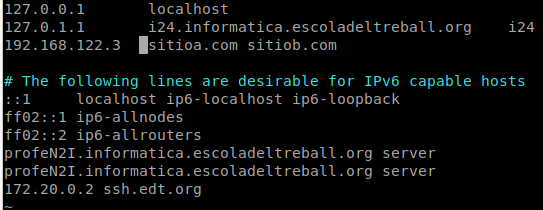
**Ara possem l’altre server:**

****

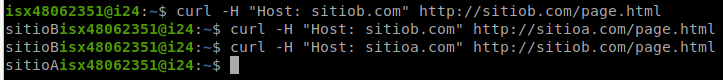
**Ara ens dona el “header” corresponent:**

****

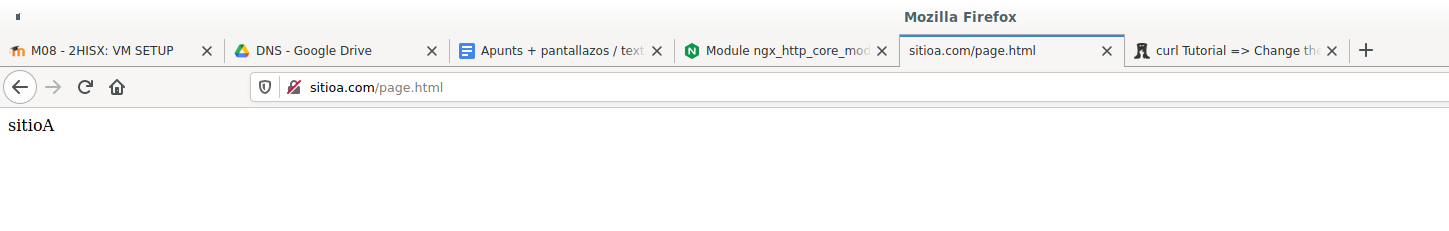
**/etc/hosts:**

****

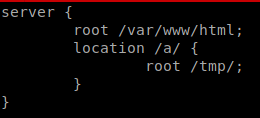
**nginx compara el “server\_name” amb el -H (host), llavors donarà el corresponent!**

****

**Amb “firefox” mostrarà el corresponent, SI O SI:**

****

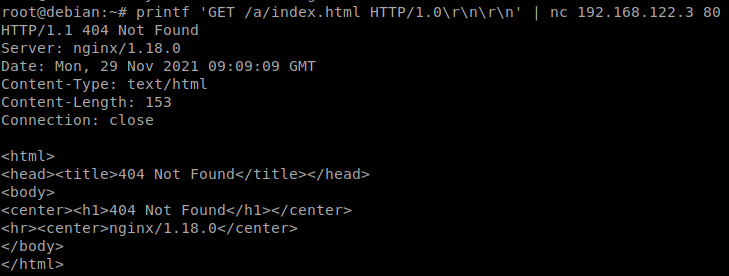
**Dins de “/etc/nginx/sites-enabled/default”, “location” busca la referència de la búsqueda amb la localització del fitxer:**

****

**Reiniciem:**

root@debian:~# systemctl restart nginx.service

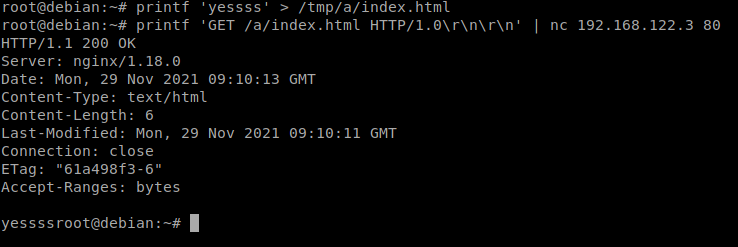
**Comprobem:**

****

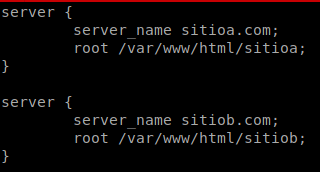
**Primer busca a ‘/var/www/html’, i després, busca a location, com no hi ha res a ‘/tmp/, hem de crear-ho:**

root@debian:~# mkdir /tmp/a

**Comprobem:**

****

**Especifiquem dos servers per les diferents rutes:**

****

**Reiniciem:**

root@debian:~# systemctl restart nginx.service

**No podem accedir com ‘sitioa.com / sitiob.com’ perquè ‘nginx’ no pot accedir a les carpetes, només als fitxers, llavors hem de fer el següent:**



**Creem ‘locations’ específiques per especificar-li on ha de buscar quan la carpeta sigui ‘/images/’:**

**Amb location especifiquem d'on ha de baixar els recursos de la web quan arriba una solicitud.**

****

**Reiniciem:**

root@debian:~# systemctl restart nginx.service

**Ara ja veiem la imatge:**

****

**Fem una especifiació amb ~ PER (quan el fitxer acabi en ‘jpg’, ho busques a ‘/var/www/html’:**

****

**Reiniciem:**

root@debian:~# systemctl restart nginx.service

**Comprobem (no hauria de funcionar):**

****

**Fem una especifiació amb PER (quan el fitxer acabi en ‘svg’ (icone), ho busques a ‘/var/www/html’:**

****

**Reiniciem:**

root@debian:~# systemctl restart nginx.service

**Comprobem (hauria de funcionar):**

****

**Ara tenim**

vim sitioa/page.html

<h1> This is sitioa</h1>

<img src='images/icon.svg'>

vim sitioa/page.html

<h1> This is sitiob</h1>

<img src='images/icon.svg'>

**Si fem que directament busqui a ‘/images’ els .svg, donarà error ja que busca a /var/www/html/images/images → ja que tenim al fitxer ‘.../sitioa/page.html’ ‘/images’**

****

**Ara, canviem el fitxer ‘/var/www/html/sitioa/page.html’**

****

**Reiniciem:**

root@debian:/var/www/html# systemctl restart nginx.service

**Comprobem:**

****

**Canviem el nom del directori ‘images’:**

root@debian:/var/www/html# mv images/ imgs

**Canviem el fitxer ‘/etc/nginx/sites-enabled/default’:**

****

**Reiniciem:**

root@debian:/var/www/html# systemctl restart nginx.service

**Comprobem:**

****

**Exemples amb PER:**

****

**SSL:**

**CLIENT: 443 (HTTPS) SERVIDOR**

**FIREFOX ------------------------------------------------------------------> SERVIDOR**

**clau**

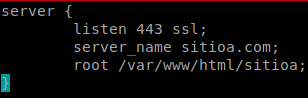
**<-----------------------------------------------------------------**

**contingut**

**<-----------------------------------------------------------------**

**Si volem que el tràfic sigui encriptat sobre el port que hem possat, hem d’afegir ‘ssl’:**

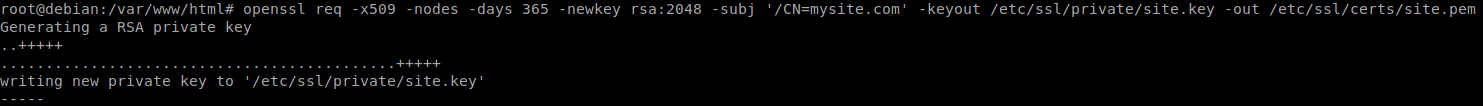
**443 → HTTPS per defecte.**

****

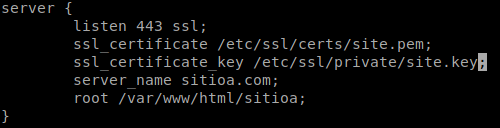
**Fem la següent ordre:**

root@debian:/var/www/html**#** openssl req -x509 -nodes -days 365 -newkey rsa:2048 -subj '/CN=mysite.com' -keyout /etc/ssl/private/site.key -out /etc/ssl/certs/site.pem

Ens genera una clau privada i una pública (la que li envia al client).



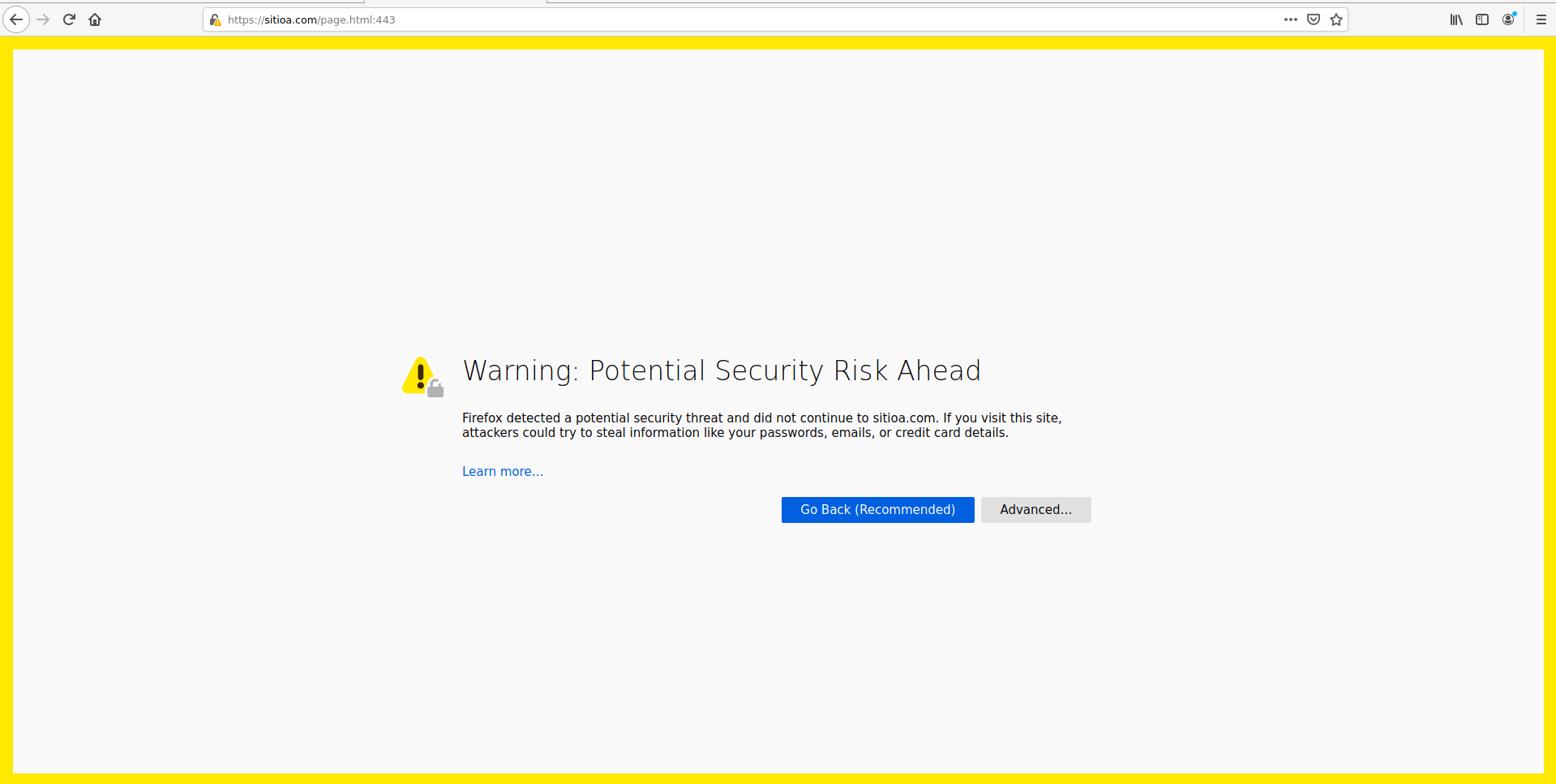
**A continuació, afegim les claus al fitxer ‘/etc/nginx/sites-enabled/default’:**

****

**Reiniciem:**

root@debian:/var/www/html# systemctl restart nginx.service

**Comprobem:**

****

****

**Capturem tràfic per HTTPS:**

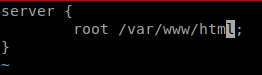
isx48062351@i24:~$ sudo tcpdump -i virbr0 -w https.pcap

Busquem al navegador https://sitioa.com/page.html

isx48062351@i24:~$: sudo wireshark

Obrim el fitxer ‘https.pcap’

**Borramos la configuración del otro día (/etc/nginx/sites-enabled/default):**

****

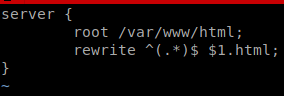
**Probamos:**

****

****

**Quan arribi una petició ‘rewrite’ canvia el que hi hagi i posa ‘.html’**

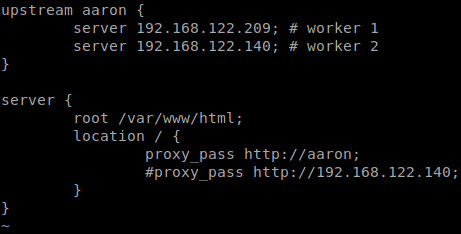
EX: 192.168.122.3/documents → afegirà ell sol el ‘.html’:

****

**Comprobem:**

****

**upstream (anar cap a munt):**

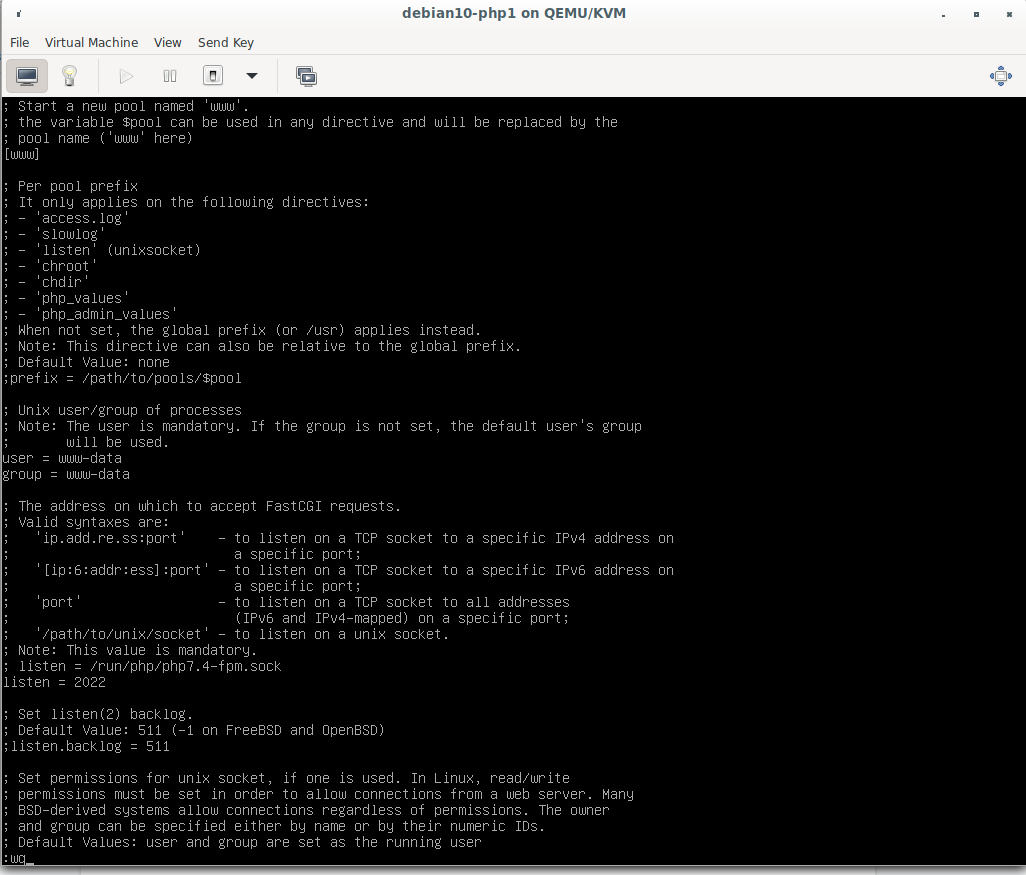
****

**proxy\_pass → passar-ho a un servidor nginx**

**apt install php-fpm → configuració PHP**

/etc/php/7.4/fpm/pool.d/www.conf **→ Configuració ‘php’**

**Afegim la línea ‘listen = 2022’**

****

**Reiniciem:**

systemctl restart php7.4-fpm.service

**Afegim un nou fitxer dins de la carpeta ‘/var/www/html’ anomenat index.php (si la carpeta no existeix, hem de crear-la):**

****

**Cambiem la informació del bal (debian) dins de (etc/nginx/sites-enabled/default):**

****

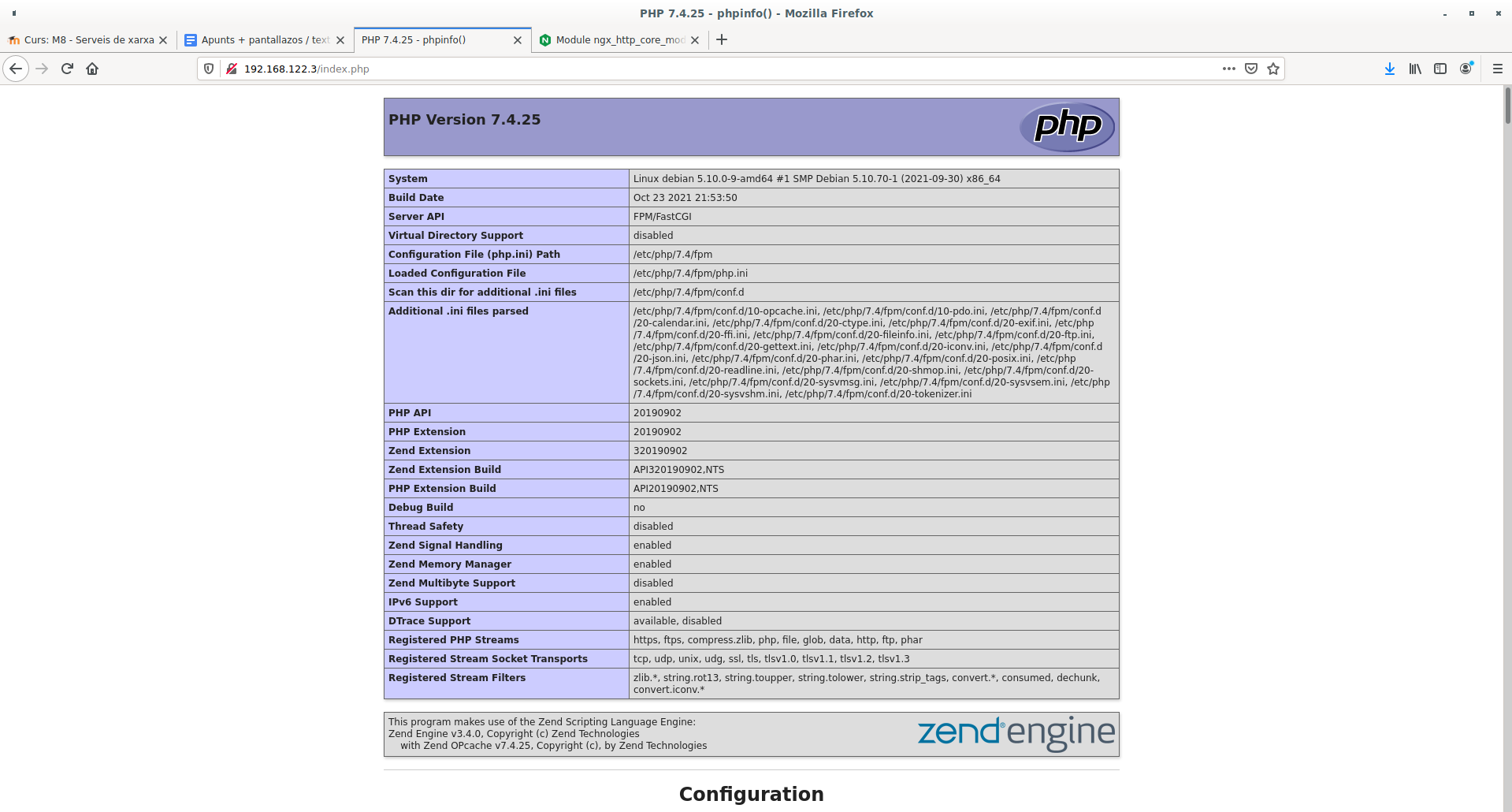
**fastcgi\_pass → passar-ho a un servidor php**

**fastcgi.conf → carpeta on ha de buscar el servidor PHP.**

**Reiniciem:**

systemctl restart nginx.service

**Comprobem:**

****