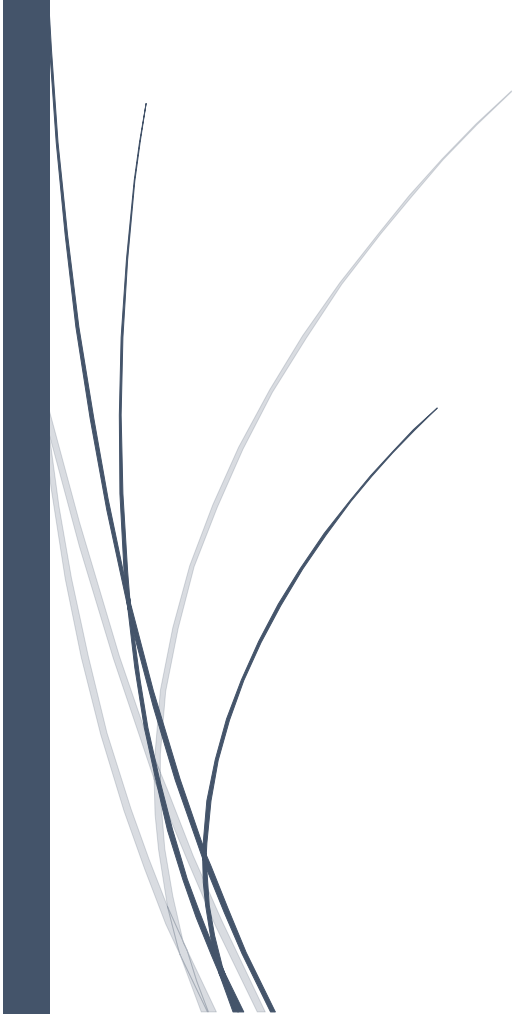
DESPLIEGUE DE APLICACIONES WEB

PROXY INVERSO EN NGINX



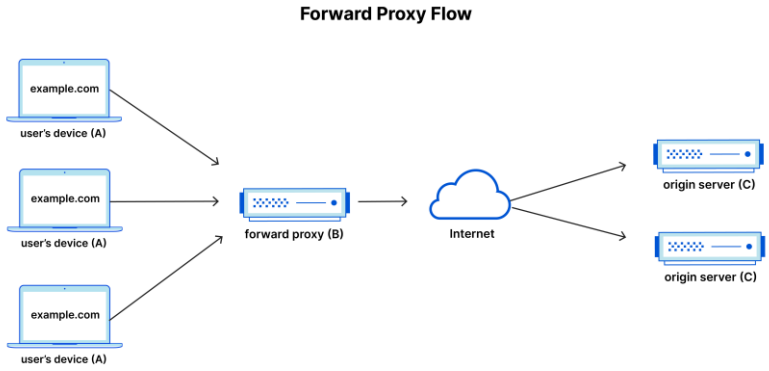
ÍNDICE

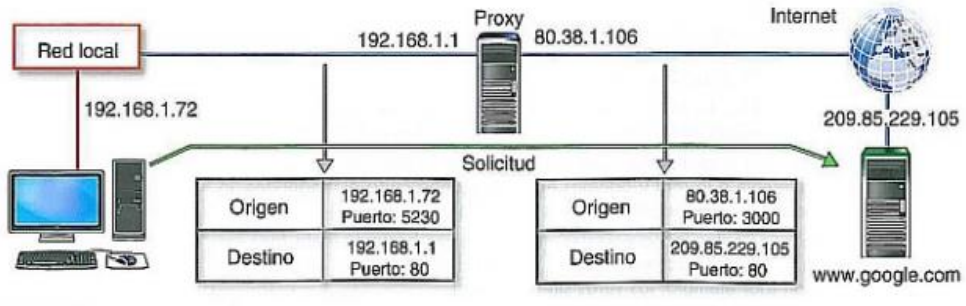
INTRODUCCIÓN

En esta práctica se va a aprender a realizar un proxy inverso sobre el servidor web Nginx.

A. ¿QUÉ ES UN PROXY? DIFERENCIA ENTRE PROXY DE REENVÍO Y PROXY INVERSO

Pero antes de empezar con la práctica vamos a explicar qué es un servidor proxy. Este tipo de servidor se encuentra entre un grupo de máquinas cliente como intermediario de estos e internet y se va a encargar de interceptar las solicitudes de estos y comunicarse con los servidores web, es decir, es algo que va a actuar como un intermediario en la comunicación cliente – servidor.





Cuando un usuario realiza una petición web a través de un cliente se construye un paquete en el que se contiene una dirección de origen formada por la IP del equipo y un puerto de salida; y una dirección de destino que estará formada por la dirección IP del proxy y su puerto de escucha.

Una vez que la solicitud llega al proxy este generará un nuevo paquete para "hacerse pasar" por el equipo del usuario y conectarse con el servidor de destino, es decir, se vuelve a crear un paquete con origen y destino. Por

tanto, cuando el paquete llegue a los servidores del servidor destino sólo habrá información del proxy y no del equipo origen.

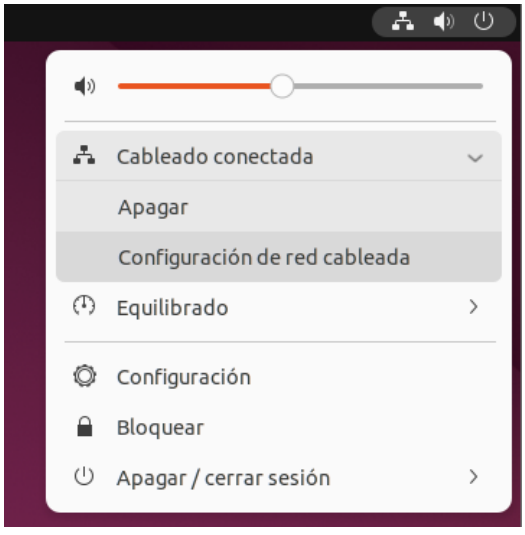
Ahora llega el momento de la respuesta del servidor y este construirá una respuesta con destino al servidor proxy. Cuando esta respuesta llegue al proxy, este se va a encargar de modificar las cabeceras de origen y destino poniéndose a sí mismo como origen y al equipo del usuario como destino.

Las razones para utilizar un servidor proxy son muy variadas, por ejemplo:

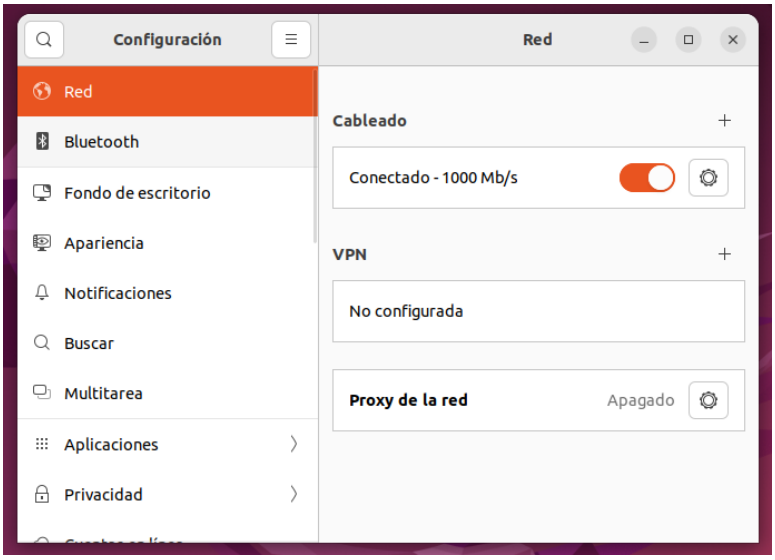
- **Evitar restricciones de navegación estatales o institucionales.** Algunos gobiernos o instituciones ofrecen a sus usuarios una versión limitada de internet. Esta restricción se puede evitar usando un proxy de reenvío ya que permite que el usuario se conecte al proxy en lugar de directamente a los sitios web que se encuentran bloqueados.
- **Bloquear el acceso a cierto contenido.** Es el caso contrario al anterior, ahora vamos a configurar el proxy para bloquear el acceso de un grupo de usuarios a ciertos sitios. Por ejemplo, una red escolar que se configura con un proxy que, si recibe respuestas de los servidores de redes sociales, las filtre y no las muestre en pantalla.
- **Proteger la identidad de los usuarios en línea.** Al utilizar un proxy de reenvío se puede hacer más difícil el rastreo de los usuarios en línea.

Si queremos configurar esto podríamos verlo en la máquina virtual Ubuntu de prácticas anteriores que hemos utilizado como cliente.

Accedemos a la "Configuración de red cableada" que se encuentra en el menú de GNOME → Cableado conectada → Configuración de red cabledada:



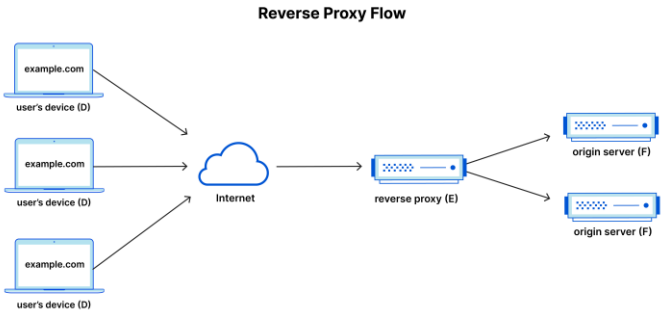
Esto nos abrirá una ventana de configuración de red:



Si accedemos a los ajustes del proxy de red y elegimos la opción "Manual" podremos ver la configuración del proxy:

En función del tipo de proxy de reenvío que queramos configurar rellenaremos más o menos campos, si queremos que todas las conexiones se filtren podríamos simplemente rellenar el último campo (servidor socks). En los campos lo que tendremos que poner es la IP del proxy y su puerto de escucha.

Pero esto que hemos explicado es un proxy normal también conocido como servidor proxy o proxy de reenvío, lo que vamos a hacer en esta práctica es un proxy inverso, es decir, un proxy que no está situado justo a continuación de los clientes sino justo antes de los servidores.



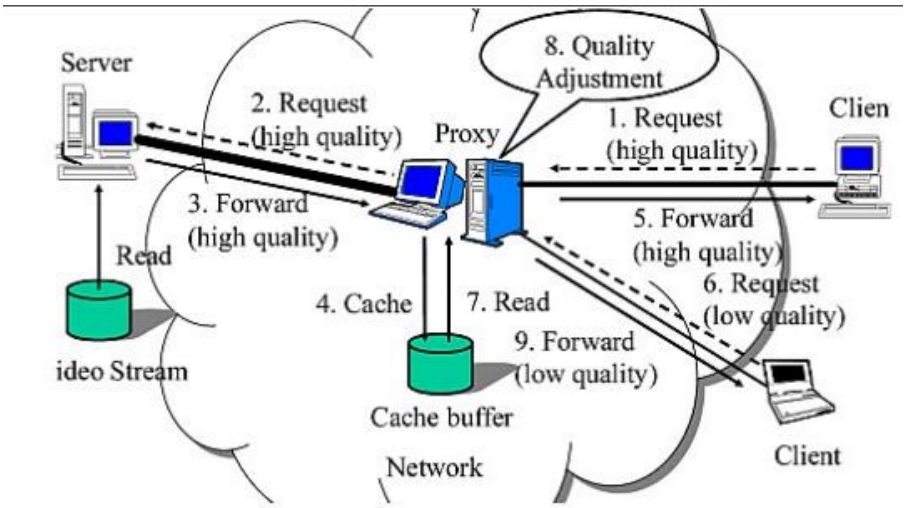
La diferencia entre uno y otro tipo de servidor proxy es que mientras el de reenvío se encuentra frente a un cliente y garantiza que ningún servidor de origen se comunique nunca directamente con un cliente en específico, el proxy inverso se encuentra frente al servidor de origen y garantiza que ningún cliente se comunica nunca directamente con él.

Los beneficios del uso de estos proxys inversos son los siguientes:

- **Balanceo de carga:** Puede ser que nuestro sitio web reciba una alta carga de solicitudes al ser un sitio web con cierta popularidad, obviamente no vamos a poder manejar todo ese tráfico con un único servidor y vamos a necesitar varios. En este caso el proxy inverso va a dividir ese tráfico entrante entre los diferentes servidores, de manera que cada usuario se conectara a un servidor en particular. En caso de que un servidor falle por completo, el resto de los servidores intensificarán su trabajo para suplirlo.
- **Protección contra ataques:** Otra de las ventajas de un proxy inverso es que un sitio web nunca va a necesitar revelar la IP de sus servidores. Esto hará más difícil el recibir ataques como por ejemplo un ataque de denegación de servicio o DDoS.
- **Almacenamiento en caché:** Un proxy inverso puede almacenar contenido en caché, de manera que su rendimiento será más rápido. Por ejemplo, un usuario de Madrid se conecta a un sitio web con proxy inverso con servidores web en Berlín, el usuario podrá conectar a un servidor proxy inverso de Madrid que luego se comunicará con el servidor de origen que se encuentra en Berlín. En este momento el servidor proxy almacenará en caché los datos de la respuesta que reciba de Berlín de manera que los siguientes usuarios que intenten entrar al sitio desde Madrid en lugar de tener que esperar a la respuesta desde Berlín, obtendrán la versión almacenada en la caché local de Madrid.
- **Cifrado SSL:** El cifrar y descifrar un SSL para cada cliente es computacionalmente caro para un servidor, con un proxy inverso vamos a poder descifrar todas las solicitudes entrantes y cifrar

todas las respuestas salientes, liberando de esa manera recursos en el servidor.

En referencia al uso como caché podríamos hablar también del proxy caché cuyo uso sería, por ejemplo, almacenar recursos que puedan ser solicitados de nuevo en un corto periodo de tiempo o que puedan ser repetitivamente utilizados (como las claves públicas y privada, vídeos de una web, etc.) con ello disminuiremos las solicitudes al servidor al ser el proxy el encargado de responder a estas peticiones realizadas por los clientes.



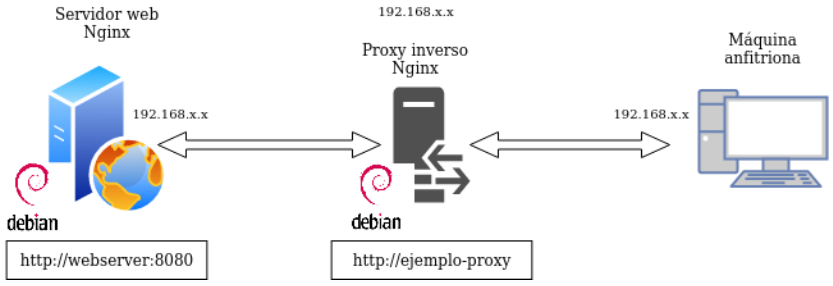
B. REQUISITOS PARA LA PRÁCTICA

Vamos a necesitar una máquina virtual que ejercerá de servidor web donde tengamos instalado el servidor web Nginx. En nuestro caso continuamos con la misma máquina virtual que en la práctica anterior, una máquina virtual servidora que tiene instalado el sistema operativo Debian 12.

Además de esta máquina necesitaremos una segunda máquina virtual Debian con servidor web Nginx que va a actuar de proxy inverso.

Para comprobar el funcionamiento primero utilizaremos la máquina cliente de Ubuntu hacer peticiones al servidor proxy que nos dirigirá al servidor web original.

Por último, usaremos nuestro equipo anfitrión como cliente y tendremos que actualizar el tipo de red de las máquinas.



CONFIGURACIONES EN LA MÁQUINA QUE EJERCERÁ DE SERVIDOR

Vamos a cambiar ciertas configuraciones en la máquina virtual que será el servidor web de nuestra práctica.

Primero vamos a cambiar el nombre de una de las webs, por ejemplo, la de www.estatica.es a webserver y vamos a eliminar el resto (lo cual nos servirá también como práctica para eliminar un sitio web de nuestro servidor web). Esto implica:

a) Actualizar el archivo /etc/hosts. Para ello usamos el comando:

sudo nano /etc/hosts

```
albertom-servidor@servidor-debian:~$ sudo nano /etc/hosts
```

En este fichero vamos a eliminar las referencias que teníamos a dominios web de prácticas anteriores, no necesitamos que esta máquina traduzca ningún dominio a una dirección IP ya que será la encargada de servir a las peticiones del proxy:

```
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       servidor-debian
```

b) Eliminar el directorio /var/www/fich_academia y modificar el nombre del directorio web_estatica a webserver:

```
albertom-servidor@servidor-debian:~$ ls -la /var/www
total 20
drwxr-xr-x  5 root     root     4096 oct 20 15:23 .
drwxr-xr-x 12 root     root     4096 oct  6 19:07 ..
drwxr-xr-x  7 www-data www-data 4096 mar 28  2019 fich_academia
drwxr-xr-x  2 root     root     4096 oct  6 19:07 html
drwxr-xr-x  6 www-data www-data 4096 oct  9 16:52 web_estatica
```

Para eliminar el directorio /fich_academia debemos usar el siguiente comando:

sudo rm -Rf /var/www/fich_academia/

Este comando va a borrar de forma recursiva dentro del directorio y va a hacer un borrado forzado (f) sin preguntar.

```
albertom-servidor@servidor-debian:~$ sudo rm -Rf /var/www/fich_academia/
albertom-servidor@servidor-debian:~$ ls -la /var/www/
total 16
drwxr-xr-x  4 root     root     4096 oct 27 16:05 .
drwxr-xr-x 12 root     root     4096 oct  6 19:07 ..
drwxr-xr-x  2 root     root     4096 oct  6 19:07 html
drwxr-xr-x  6 www-data www-data 4096 oct  9 16:52 web_estatica
```

También vamos a borrar el directorio html, dejando únicamente el de web_estática, el comando es el mismo que antes, pero cambiando el nombre del directorio a borrar.

```
albertom-servidor@servidor-debian:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 root     root     4096 oct 27 16:07 .
drwxr-xr-x 12 root     root     4096 oct  6 19:07 ..
drwxr-xr-x  6 www-data www-data 4096 oct  9 16:52 web_estatica
```

Por último, en este directorio sólo nos quedaría cambiar el nombre del directorio web_estatica a webserver, para ello hacemos uso del comando mv:

sudo mv /var/www/web_estatica /var/www/webserver

```
albertom-servidor@servidor-debian:~$ sudo mv /var/www/web_estatica/ /var/www/webserver
albertom-servidor@servidor-debian:~$ ls -la /var/www/
total 12
drwxr-xr-x  3 root     root     4096 oct 27 16:09 .
drwxr-xr-x 12 root     root     4096 oct  6 19:07 ..
drwxr-xr-x  6 www-data www-data 4096 oct  9 16:52 webserver
```

c) Eliminar los enlaces simbólicos del directorio /etc/nginx/sites-enabled/. Para conseguir esto utilizaremos el comando unlink seguido de la dirección del enlace.

Ahora mismo tenemos estos enlaces:

```
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 oct 20 15:38 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
lrwxrwxrwx 1 root root   34 oct  6 19:07 default -> /etc/nginx/sites-available/default
lrwxrwxrwx 1 root root   42 oct 20 15:38 www.academia.es -> /etc/nginx/sites-available/www.academia.es
lrwxrwxrwx 1 root root   42 oct  9 17:05 www.estatica.es -> /etc/nginx/sites-available/www.estatica.es
```

Por lo tanto, necesitaremos introducir los siguientes 3 comandos:

sudo unlink /etc/nginx/sites-enabled/default

sudo unlink /etc/nginx/sites-enabled/www.academia.es

sudo unlink /etc/nginx/sites-enabled/www.estatica.es

```
albertom-servidor@servidor-debian:~$ sudo unlink /etc/nginx/sites-enabled/default
albertom-servidor@servidor-debian:~$ sudo unlink /etc/nginx/sites-enabled/www.academia.es
albertom-servidor@servidor-debian:~$ sudo unlink /etc/nginx/sites-enabled/www.estatica.es
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 oct 27 16:16 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
```

d) Eliminar los archivos default y www.academia.es del directorio /etc/nginx/sites-available/ y modificar el de www.estatica.es.

```
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-available/
total 20
drwxr-xr-x 2 root root 4096 oct 23 18:12 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
-rw-r--r-- 1 root root 2412 mar 14  2023 default
-rw-r--r-- 1 root root 2722 oct 23 18:12 www.academia.es
-rw-r--r-- 1 root root 2405 oct  9 16:57 www.estatica.es
```

Para eliminar los archivos vamos a usar el comando rm:

sudo rm /etc/nginx/sites-available/default

sudo rm /etc/nginx/sites-available/www.academia.es

```
albertom-servidor@servidor-debian:~$ sudo rm /etc/nginx/sites-available/default
albertom-servidor@servidor-debian:~$ sudo rm /etc/nginx/sites-available/www.academia.es
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-available/
total 12
drwxr-xr-x 2 root root 4096 oct 27 16:20 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
-rw-r--r-- 1 root root 2405 oct  9 16:57 www.estatica.es
```

Ahora vamos a renombrar el archivo www.estatica.es a webserver, para ello de nuevo debemos hacer uso del comando mv:

sudo mv /etc/nginx/sites-available/www.estatica.es /etc/nginx/sites-available/webserver

```
albertom-servidor@servidor-debian:~$ sudo mv /etc/nginx/sites-available/www.estatica.es /etc/nginx/sites-available/webserver
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-available/
total 12
drwxr-xr-x 2 root root 4096 oct 27 16:22 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
-rw-r--r-- 1 root root 2405 oct  9 16:57 webserver
```

e) Modificar este archivo de configuración que acabamos de renombar. Utilizaremos el comando:

sudo nano /etc/nginx/sites-available/webserver

```
albertom-servidor@servidor-debian:~$ sudo nano /etc/nginx/sites-available/webserver
```

Y llevamos a cabo las siguientes modificaciones:

- Los puertos de escucha IPv4 e IPv6 pasan de ser el puerto 80 a ser el 8080.

```
listen 8080;
listen [::]:8080;
```

- El root del directorio ha sido modificado en el apartado b) así que lo tendremos que modificar aquí también pasando de /var/www/web_estatica/ a /var/www/webserver/.

```
root /var/www/webserver/;
```

- El server_name fue modificado en el apartado a) así que igualmente deberá ser modificado aquí pasando de www.estatica.es a webserver.

```
server_name webserver;
```

f) Por último, debemos volver a crear el enlace simbólico. Para conseguirlo utilizamos el siguiente comando:

sudo ln -s /etc/nginx/sites-available/webserver /etc/nginx/sites-enabled/

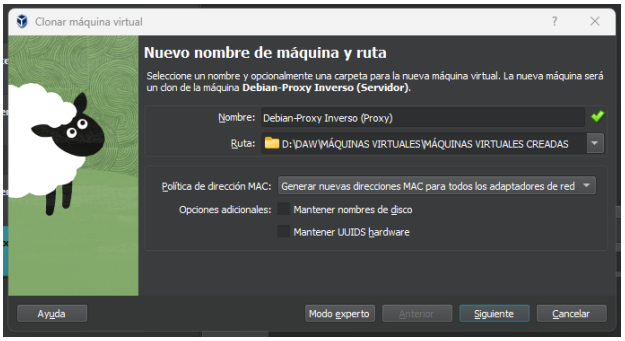
```
albertom-servidor@servidor-debian:~$ sudo ln -s /etc/nginx/sites-available/webserver /etc/nginx/sites-enabled/
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 oct 27 16:30 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
lrwxrwxrwx 1 root root   36 oct 27 16:30 webserver -> /etc/nginx/sites-available/webserver
```

Al haber estado tocando los archivos de configuración de Nginx debemos reiniciar el servicio para que estos cambios tengan efecto.

```
albertom-servidor@servidor-debian:~$ sudo systemctl restart nginx.service
albertom-servidor@servidor-debian:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Fri 2023-10-27 16:32:16 CEST; 4s ago
       Docs: man:nginx(8)
    Process: 2131 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 2132 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 2133 (nginx)
      Tasks: 3 (limit: 2284)
     Memory: 2.3M
        CPU: 14ms
     CGroup: /system.slice/nginx.service
             ├─2133 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─2134 "nginx: worker process"
             └─2135 "nginx: worker process"

oct 27 16:32:16 servidor-debian systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
oct 27 16:32:16 servidor-debian systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

CONFIGURACIONES EN LA MÁQUINA VIRTUAL QUE EJERCERÁ DE PROXY INVERSO

Para conseguir la máquina proxy podemos hacer una clonación enlazada de la máquina servidora. Es importante que en la opción de "política de dirección MAC" elijamos la opción de generar una nueva MAC para la máquina, de lo contrario no tendremos una IP en esta nueva máquina.



Lo que queremos conseguir ahora es que cuando introduzcamos en el navegador la dirección www.estatica.es el servidor proxy nos redirija al webserver:8080 que hemos creado en la sección anterior.

Lo primero que debemos hacer es eliminar toda referencia a webserver en /var/www por lo que eliminaremos el directorio:

sudo rm -Rf /var/www/webserver

```
albertom-servidor@servidor-debian:~$ sudo rm -Rf /var/www/webserver
albertom-servidor@servidor-debian:~$ ls -la /var/www
total 8
drwxr-xr-x  2 root root 4096 oct 28 22:46 .
drwxr-xr-x 12 root root 4096 oct  6 19:07 ..
```

También vamos a eliminar el enlace simbólico de /etc/nginx/sites-enabled:

sudo unlink /etc/nginx/sites-enabled/webserver

```
albertom-servidor@servidor-debian:~$ sudo unlink /etc/nginx/sites-enabled/webserver
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 oct 28 22:49 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
```

Por último, borramos el archivo de configuración de configuración de webserver de /sites-available:

sudo rm /etc/nginx/sites-available/webserver

```
albertom-servidor@servidor-debian:~$ sudo rm /etc/nginx/sites-available/webserver
albertom-servidor@servidor-debian:~$ ls -la /etc/nginx/sites-available
total 8
drwxr-xr-x 2 root root 4096 oct 28 22:51 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
```

Ahora vamos crear un archivo de configuración en el directorio en este directorio, en nuestro caso lo llamaremos estatica. Para ello usaremos el comando:

sudo nano /etc/nginx/sites-available/estatica

```
albertom-servidor@servidor-debian:~$ sudo nano /etc/nginx/sites-available/estatica
```

Este archivo será más simple que los que hemos creado en anteriores prácticas:

```
server {
        listen 80;
        listen [::]:80;
        server_name www.estatica.es;

        location / {
                proxy_pass http://webserver:8080;
        }
}
```

- En listen debemos poner el puerto donde está escuchando el proxy inverso.
- En server_name irá nuestro dominio.
- En el proxy_pass se indicará a dónde se van a redirigir las peticiones, es decir, al servidor web, a la dirección webserver por el puerto 8080.

Una vez hecho esto debemos hacer el enlace simbólico correspondiente:

```
albertom-servidor@servidor-debian: $ ls -la /etc/nginx/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 oct 28 11:03 .
drwxr-xr-x 8 root root 4096 oct 23 17:23 ..
lrwxrwxrwx 1 root root   35 oct 27 17:07 estatica -> /etc/nginx/sites-available/estatica
```

Como último paso debemos configurar el archivo /etc/hosts para traducir la dirección "webserver" por la IP de nuestra máquina que funciona como servidor web:

```
  GNU nano 7.2
127.0.0.1       localhost
127.0.1.1       servidor-debian
10.0.2.25       webserver
10.0.2.27       www.estatica.es
```

Reiniciamos el servicio Nginx para finalizar.

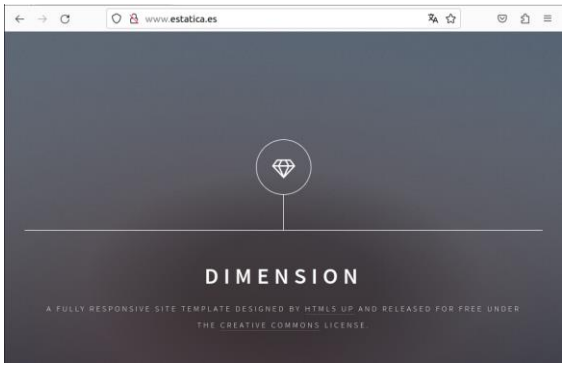
```
albertom-servidor@servidor-debian: $ sudo systemctl restart nginx.service
albertom-servidor@servidor-debian: $ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Fri 2023-10-27 17:23:20 CEST; 9s ago
       Docs: man:nginx(8)
    Process: 1909 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 1910 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 1911 (nginx)
      Tasks: 3 (limit: 2284)
     Memory: 2.3M
        CPU: 19ms
     CGroup: /system.slice/nginx.service
             ├─1911 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─1912 "nginx: worker process"
             └─1913 "nginx: worker process"
```

COMPROBACIONES DE QUE EL PROXY Y SERVIDOR ESTÁN CONECTADOS

Ahora vamos a trabajar en la máquina de Ubuntu que actúa como cliente, vamos a modificar el archivo /etc/hosts/ para asignar el dominio www.estatica.es a la IP que corresponde a nuestro proxy inverso:

```
  GNU nano 6.2
127.0.0.1       localhost
127.0.1.1       albertomcliente-VirtualBox
10.0.2.27       www.estatica.es
```

Ahora si accedemos desde un navegador web a www.estatica.es accederemos al proxy (que utiliza el puerto 80 para escuchar peticiones) que a su vez se comunicará con la máquina servidora a través de la dirección DNS webserver (nuestra máquina servidora utiliza el puerto 8080 para la escucha) y así acceder a los recursos del sitio web:



Si accedemos a los archivos de log de ambos servidores podemos ver los accesos que se están produciendo, vamos a empezar con el log del proxy:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/access.log
10.0.2.22 - - [27/Oct/2023:17:49:00 +0200] "GET /images/pic02.jpg HTTP/1.1" 200 8904 "http://www.estatica.es/" "Mozil
la/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - - [27/Oct/2023:17:49:00 +0200] "GET /images/pic03.jpg HTTP/1.1" 200 9697 "http://www.estatica.es/" "Mozil
la/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - - [27/Oct/2023:17:49:00 +0200] "GET /images/pic01.jpg HTTP/1.1" 200 10064 "http://www.estatica.es/" "Mozi
lla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - - [27/Oct/2023:17:49:00 +0200] "GET /assets/js/util.js HTTP/1.1" 200 12433 "http://www.estatica.es/" "Moz
illa/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - - [27/Oct/2023:17:49:00 +0200] "GET /assets/js/main.js HTTP/1.1" 200 8801 "http://www.estatica.es/" "Mozi
lla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.22 - - [27/Oct/2023:17:49:00 +0200] "GET /assets/css/font-awesome.min.css HTTP/1.1" 200 29063 "http://www.esta
tica.es/assets/css/main.css" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
```

Desde la dirección 10.0.2.22 se están produciendo varias solicitudes a nuestro servidor proxy y, como podemos ver en la siguiente captura, esa IP se corresponde con nuestra máquina virtual cliente:

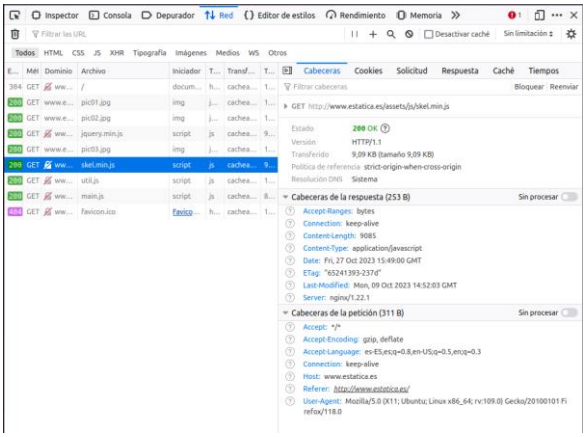
```
albertom-cliente@albertoMartinezPerez:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:a5:f1:e4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.22/24 brd 10.0.2.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::6a7:b4ec:6bbc:f53c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Ahora vamos a comprobar el log de acceso de nuestro servidor web:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/access.log
10.0.2.27 - - [27/Oct/2023:17:49:00 +0200] "GET /assets/js/jquery.min.js HTTP/1.0" 200 95957 "http://www.estatica.es/
" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.27 - - [27/Oct/2023:17:49:00 +0200] "GET /images/pic02.jpg HTTP/1.0" 200 8904 "http://www.estatica.es/" "Mozil
la/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.27 - - [27/Oct/2023:17:49:00 +0200] "GET /images/pic03.jpg HTTP/1.0" 200 9697 "http://www.estatica.es/" "Mozil
la/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.27 - - [27/Oct/2023:17:49:00 +0200] "GET /images/pic01.jpg HTTP/1.0" 200 10064 "http://www.estatica.es/" "Mozi
lla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.27 - - [27/Oct/2023:17:49:00 +0200] "GET /assets/js/util.js HTTP/1.0" 200 12433 "http://www.estatica.es/" "Moz
illa/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
10.0.2.27 - - [27/Oct/2023:17:49:00 +0200] "GET /assets/js/main.js HTTP/1.0" 200 8801 "http://www.estatica.es/" "Mozi
lla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/118.0"
```

En este caso las peticiones se están realizando desde la dirección 10.0.2.27 que es la dirección IP de nuestro proxy inverso.

Otra forma de ver estas solicitudes es a través de las herramientas de desarrollador del navegador accesibles desde F12.



En esta imagen vemos como el archivo skel.min.js fue solicitado a través de una solicitud GET con un estado 200 OK (es decir, de recepción correcta). Además de esto vemos las cabeceras HTTP tanto de la petición como de la respuesta.

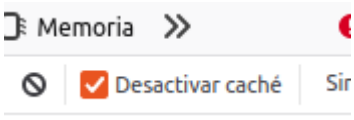
AÑADIR CABECERAS HTTP PERSONALIZADAS

Ahora vamos a añadir nuevas cabeceras HTTP personalizadas a nuestro sitio web, vamos a crear una para corroborar la conexión con el servidor proxy y otra para hacer lo mismo, pero con el servidor web. Primero lo vamos a realizar en nuestra máquina que ejerce de proxy inverso.

Para hacer esto debemos añadir una nueva directiva en el bloque location /, en concreto la directiva add_header seguida de la palabra Host y el nombre que queremos darle a nuestro host, en nuestro caso vamos a utilizar Proxy_inverso_albertoM

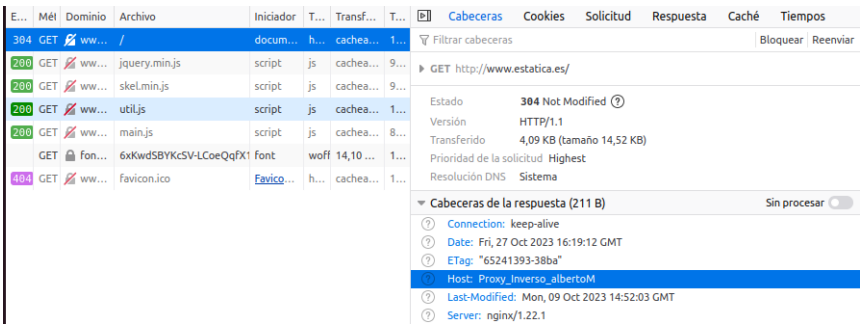
```
server {
        listen 80;
        listen [::]:80;
        server_name www.estatica.es;

        location / {
                proxy_pass http://webserver:8080;
                add_header Host Proxy_Inverso_albertoM;
        }
}
```

Reiniciamos el servidor Nginx y volvemos a cargar nuestro sitio web en el cliente, pero antes activamos la opción de desactivar caché:



Ya que, si no podríamos recibir la caché del navegador y, por tanto, resultados errores.

Ahora en la localización / nos debe aparecer la nueva cabecera que hemos creado en el proxy inverso:

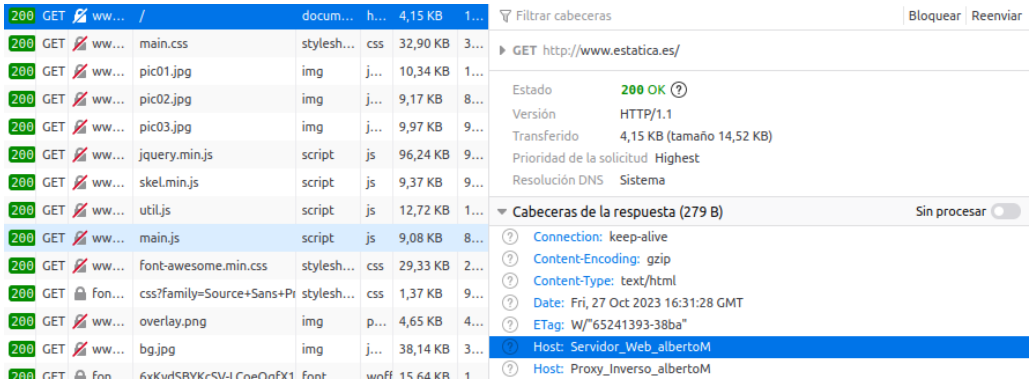


Vamos a crear ahora una cabecera específica para el servidor web, el modo de hacerlo es el mismo, debemos modificar el archivo de configuración del sitio en

el bloque location /, la directiva será la misma, pero en este caso lo que añadiremos será Host Servidor_Web_albertoM.

```
location / {
        add_header Host Servidor_Web_albertoM;
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
}
```

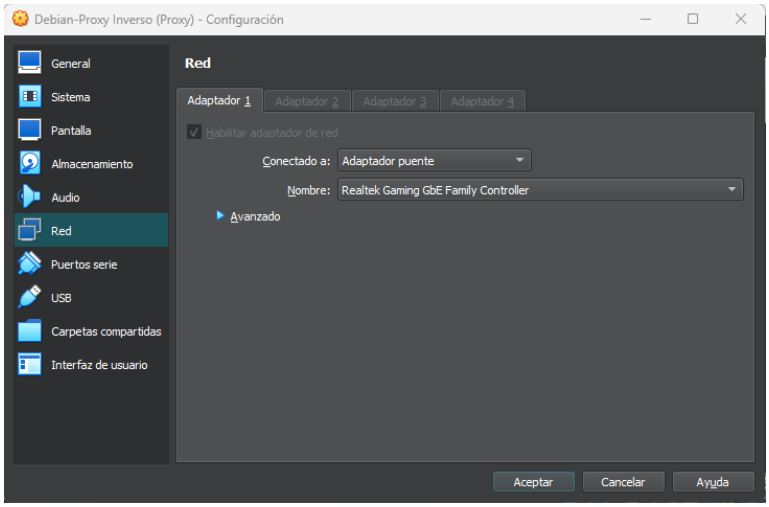
Tras reiniciar el servidor, hacemos un refresco del sitio web en nuestra máquina virtual cliente y nos debería aparecer la cabecera:



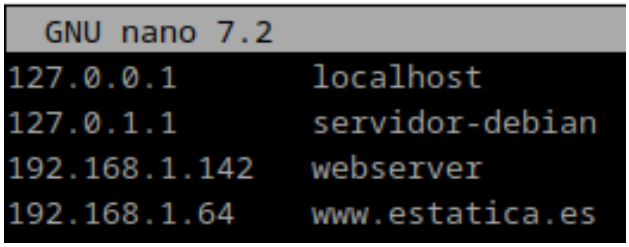
COMPROBACIÓN DEL FUNCIONAMIENTO CON EL ANFITRIÓN

Vamos ahora a utilizar nuestro equipo anfitrión como el equipo cliente en la conexión cliente-servidor.

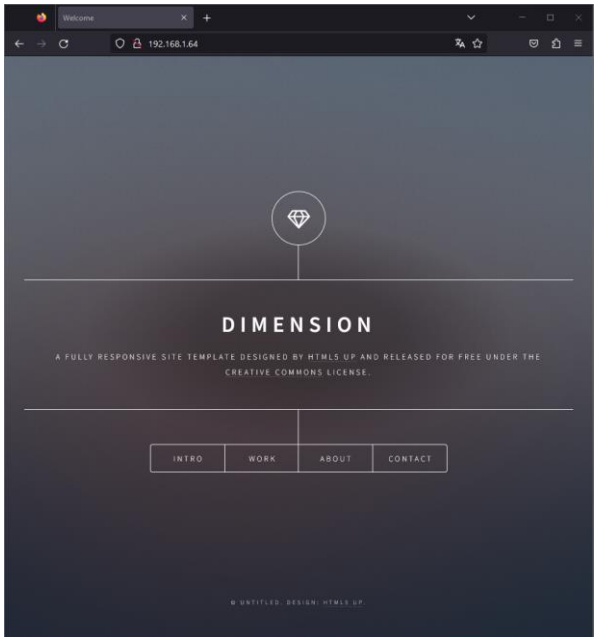
Para ello lo primero que debemos hacer es cambiar el tipo de conexión tanto en nuestra máquina virtual proxy como en nuestra máquina virtual servidor pasando de Red NAT a adaptador puente.



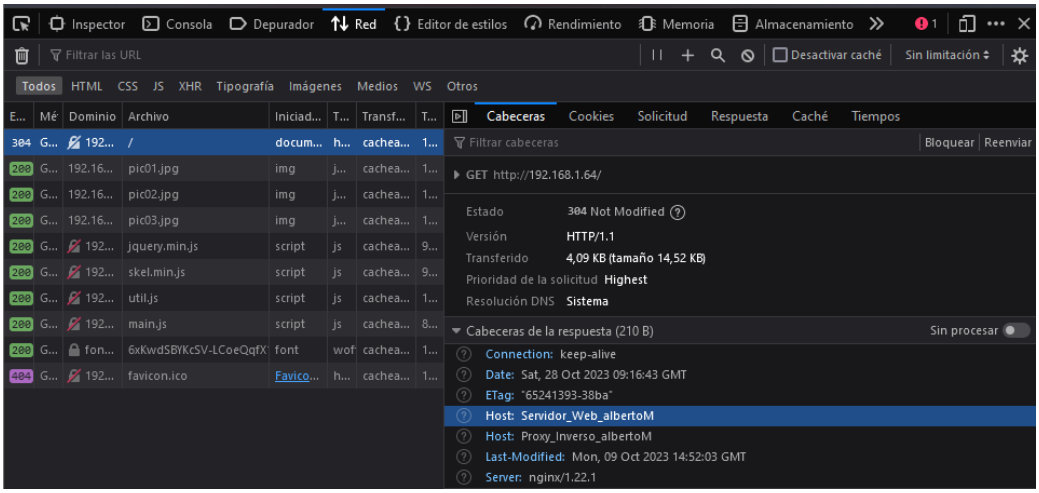
Una vez completado esto tendremos que cambiar el archivo /etc/hosts tanto en la máquina proxy con la nueva IPs que se haya asignado a la máquina servidora:



Para no tocar el archivo\etc\hosts de nuestra máquina anfitriona podemos entrar en www.estatica.es colocando la dirección IP del proxy en la barra de búsqueda del navegador. En este caso la dirección IP es 192.168.1.64.



Como se puede ver en la siguiente imagen, obtenemos ambas cabeceras personalizadas:



Y si accedemos a los logs de acceso del proxy y el servidor web obtenemos los siguientes resultados:

- Access.log del proxy:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/access.log
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /assets/js/main.js HTT
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /images/pic02.jpg HTTP
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /images/pic01.jpg HTTP
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /images/pic03.jpg HTTP
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /assets/css/font-awesc
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /images/overlay.png HT
192.168.1.10 - - [28/Oct/2023:11:14:30 +0200] "GET /assets/fonts/fontawes
01 Firefox/118.0"
```

Siendo la 192.168.1.10 la IP del equipo anfitrión:

```
PS C:\Windows\system32> ipconfig

Configuración IP de Windows


Adaptador de Ethernet Ethernet:

   Sufijo DNS específico para la conexión. . : home
   Vínculo: dirección IPv6 local. . . : fe80::3745:b13e:4895:366b%11
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.1.10
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

- Access.log del servidor:

```
albertom-servidor@servidor-debian:~$ sudo tail /var/log/nginx/access.log
192.168.1.64 - - [28/Oct/2023:11:14:30 +0200] "GET /assets/js/main.js HTT
192.168.1.64 - - [28/Oct/2023:11:14:30 +0200] "GET /images/pic02.jpg HTTP
192.168.1.64 - - [28/Oct/2023:11:14:30 +0200] "GET /images/pic01.jpg HTTP
192.168.1.64 - - [28/Oct/2023:11:14:30 +0200] "GET /images/pic03.jpg HTTP
192.168.1.64 - - [28/Oct/2023:11:14:30 +0200] "GET /assets/css/font-aweso
192.168.1.64 - - [28/Oct/2023:11:14:30 +0200] "GET /images/overlay.png HT
```