Despliegue de aplicaciones webBalance de Carga con Proxy Inverso en NGINX

Línea corta

Pablo Galván Tejada y Rubén Peraita

# Índice

1. Configuración previa del servidor web 1
2. Configuración previa del servidor web 2
3. Configuración del proxy inverso y balanceador de carga
4. Comprobación del funcionamiento del balanceador de carga
   1. Ambos servidores operativos
   2. Servidor web 1 no operativo
   3. Servidor web 1 no operativo
5. Conclusión

# 

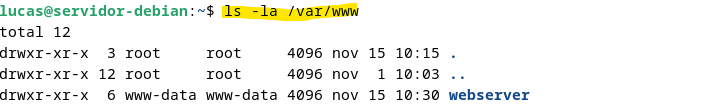
# 

# Configuración previa del servidor web 1

Para esta práctica, será necesario modificar completamente la configuración del servidor web. Eliminaremos cualquier referencia al sitio web previamente configurado, como los archivos en /var/www/, y estableceremos un nuevo directorio raíz en /var/www/ para iniciar desde cero.

Se procederá a eliminar el directorio /webserver/ y se implementará un servidor web básico que mostrará el identificador del servidor en ejecución. Dicho servidor estará alojado en la carpeta webserver1.

Dentro de este directorio, se creará un archivo denominado index.html, que servirá como página de inicio del sitio web:



Eliminaremos el directorio /webserver/



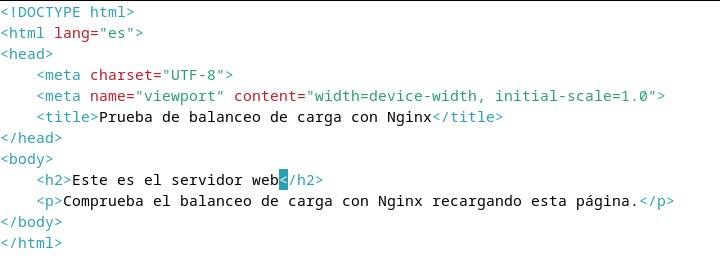
Se va a configurar un servidor web básico para mostrar el número de servidor actual. Este servidor se alojará en el directorio 'webserver1'.



Dentro de esa carpeta, crearemos un archivo llamado 'index.html'. Este archivo será la página principal de nuestro sitio web:

*sudo nano /var/www/webserver1/index.html*

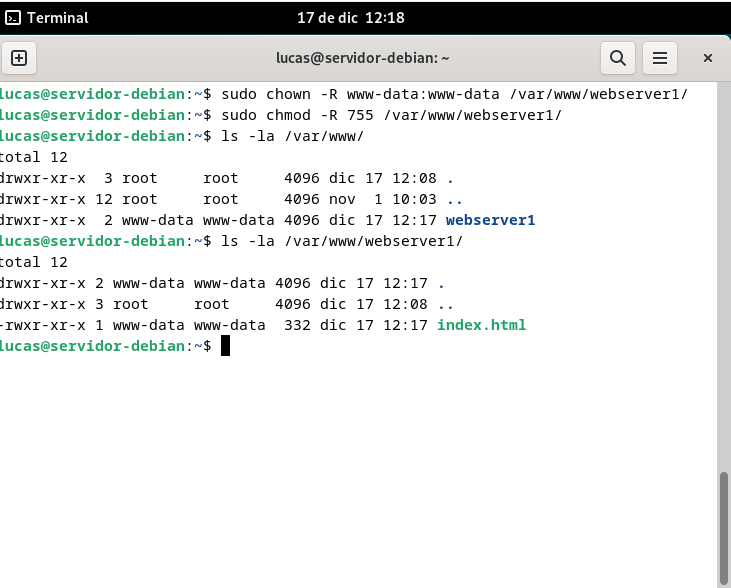
Dentro del .html crearemos una estructura HTML por ejemplo la siguiente:



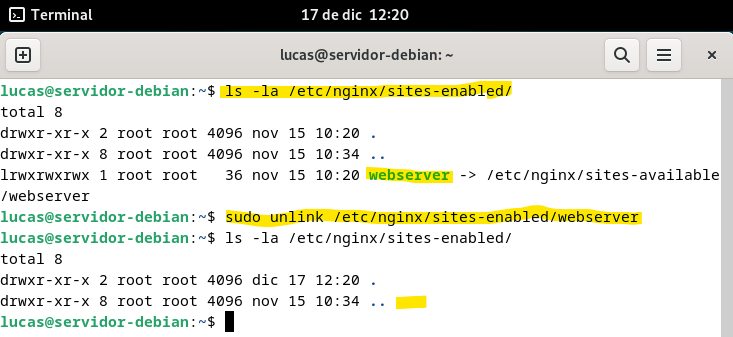
Una vez que hayamos guardado el archivo, necesitamos darle los permisos correctos para que el servidor web pueda leerlo y modificarlo. Esto es importante para que el sitio web funcione correctamente.

sudo chown -R www-data:www-data /var/www/webserver1/

sudo chmod -R 755 /var/www/webserver1/



Comenzaremos la configuración del sitio web eliminando el enlace simbólico 'webserver' que se encuentra en el directorio '/etc/nginx/sites-enabled/'. Este enlace ya no es necesario para nuestra nueva configuración.



Vamos a hacer un pequeño ajuste en la configuración de nuestro sitio web. Para eso, vamos a modificar el nombre de un archivo de configuración. Este archivo se encuentra en el directorio '/sites-available/' y en lugar de llamarse 'webserver', lo cambiaremos a 'webserver1'.



Cambiamos dentro de este archivo de configuración lo siguiente:

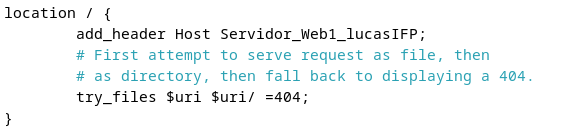
El directorio raíz de nuestra pagina



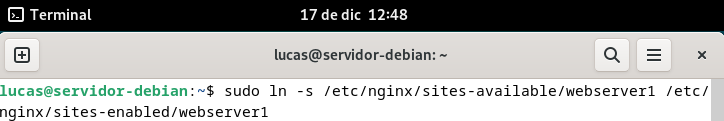
El nombre de nuestro servidor web:



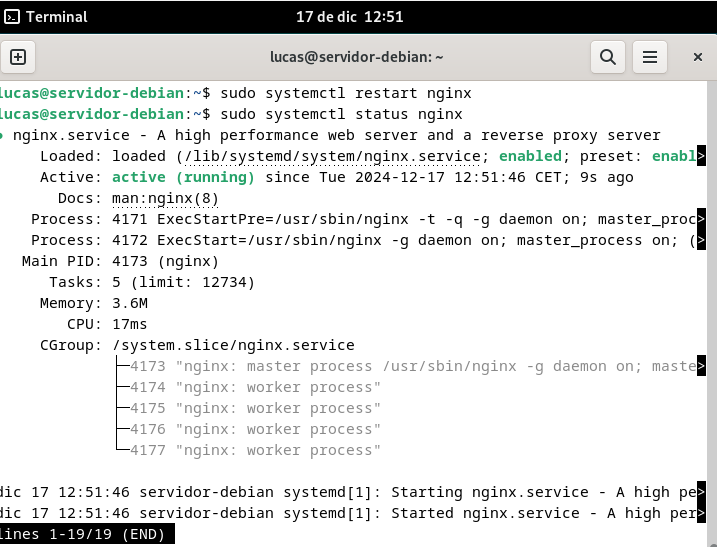
La cabecera personalizada a Servidor\_Web\_lucasIFP



Ahora debemos crear el enlace simbólico en /sites-enabled:



Una vez que hayamos terminado de hacer los cambios, reiniciaremos el servidor web Nginx. Luego, comprobaremos si está funcionando correctamente y si los cambios se han aplicado.



# 

# 

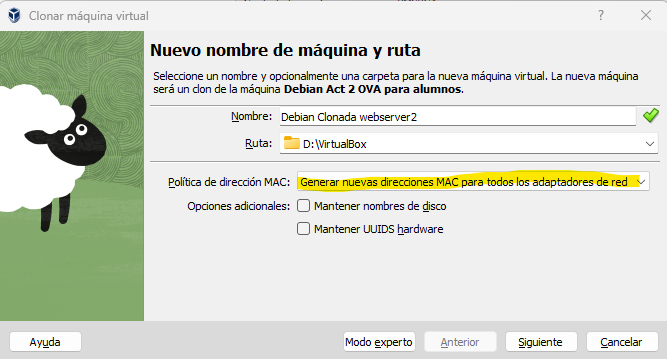
# Configuración previa del servidor web 2

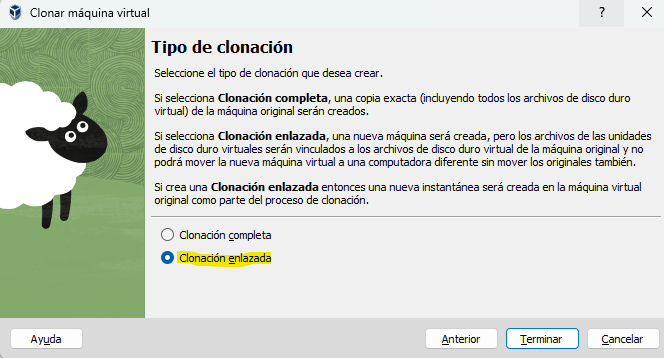
El segundo servidor será una réplica exacta del primero, pero con una dirección MAC diferente para asignarle su propia dirección IP en la red.

Se renombrará el directorio /var/www/webserver1 a webserver2 y se modificará el archivo index.html para reflejar que pertenece al segundo servidor.

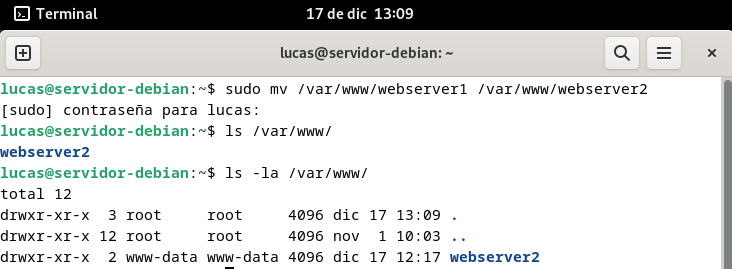
Será necesario eliminar el enlace simbólico que apuntaba al archivo de configuración del servidor 1 y renombrar este último a webserver2.

Después de replicar las modificaciones realizadas en webserver1, se creará el nuevo enlace simbólico en /sites-enabled/ y se reiniciará el servicio NGINX.

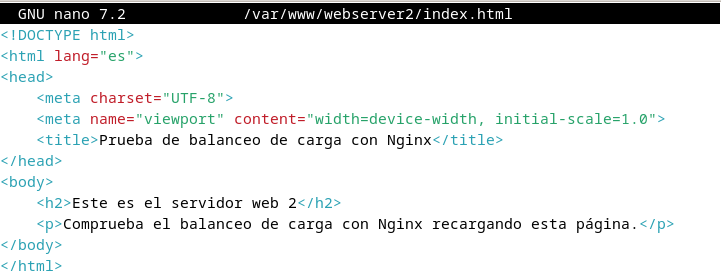




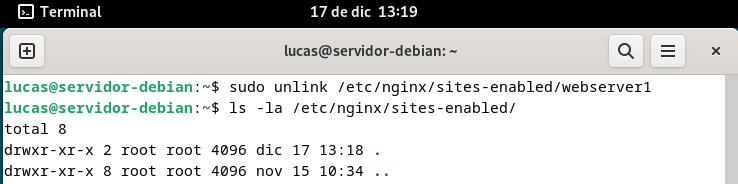
Ahora renombramos el directorio /var/www/webserer1 a webserver2:



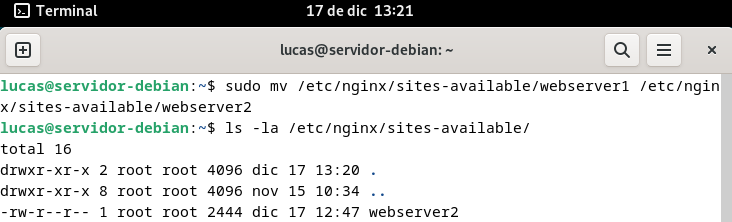
Vamos a cambiar un poquito el archivo donde está toda la información de este sitio web (index.html) para que diga que ahora estamos en el servidor número 2.



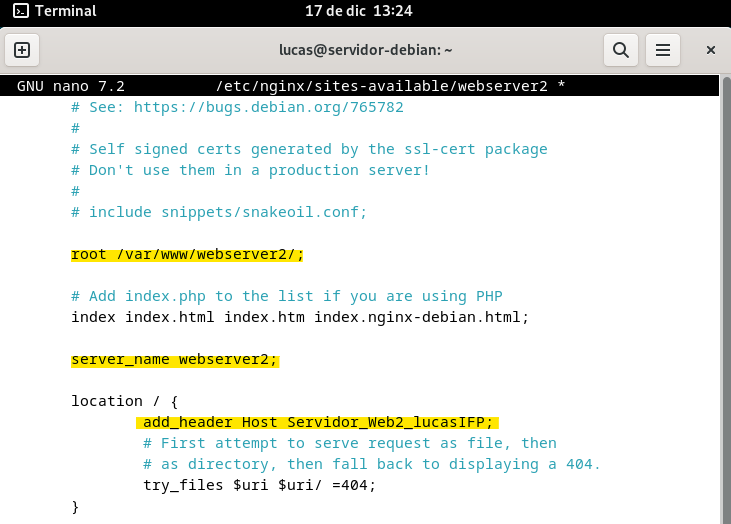
Hay que quitar el enlace simbólico de /etc/nginx/sites-enabled/ que señala al archivo /etc/nginx/sites-available/webserver1 para poder utilizar otra configuración.



Después de hacer esto, le pondremos un nombre nuevo a nuestro **archivo de configuración**. En vez de llamarse 'webserver1', se llamará 'webserver2'.



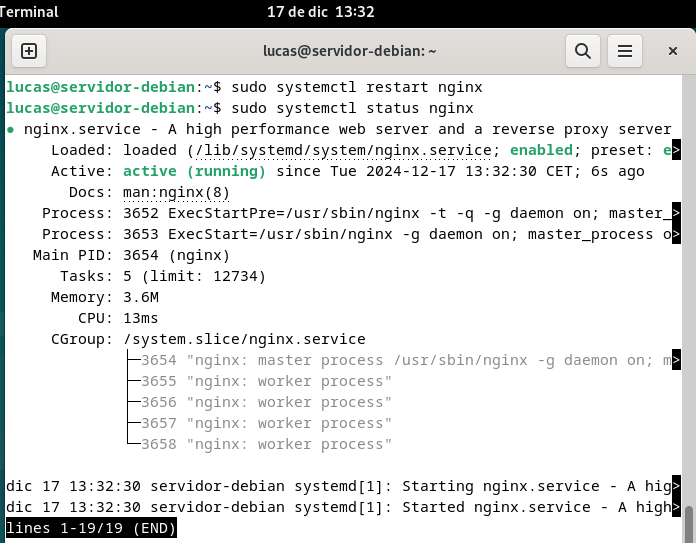
Dentro del archivo modificamos la mismas cosas que en nuestro archivo original



Después crearemos el enlace simbólico en /sites-enabled/



Una vez que hayamos terminado de hacer los cambios, reiniciaremos el servidor web Nginx. Luego, comprobaremos si está funcionando correctamente y si los cambios se han aplicado.



# Configuración del proxy inverso y balanceador de carga

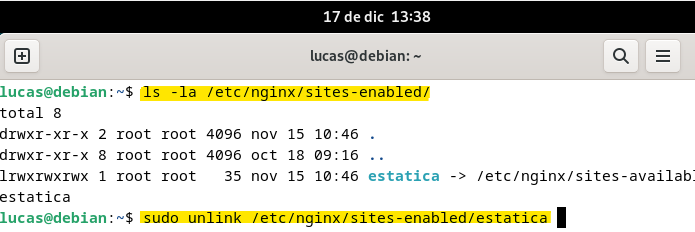
Se eliminará el enlace simbólico de la configuración anterior y se modificará el nombre del archivo estatica dentro de /etc/nginx/sites-available, cambiándolo por balanceo.

El código se actualizará para incluir la sección upstream denominada backend\_hosts, donde se definirán los servidores web (webserver1 y webserver2) en el puerto 8080. Se empleará el método de balanceo de carga aleatorio (random), lo que permitirá que cada solicitud sea enviada de forma equitativa a cualquiera de los dos servidores disponibles.

En la configuración del servidor proxy, se especificará el puerto de escucha, el dominio asociado y la directiva proxy\_pass, que enviará las solicitudes al grupo upstream.

Para finalizar esta configuración, se creará el enlace simbólico correspondiente dentro de /etc/nginx/sites-enabled/ y se actualizará el archivo /etc/hosts con las direcciones IP de webserver1 y webserver2.

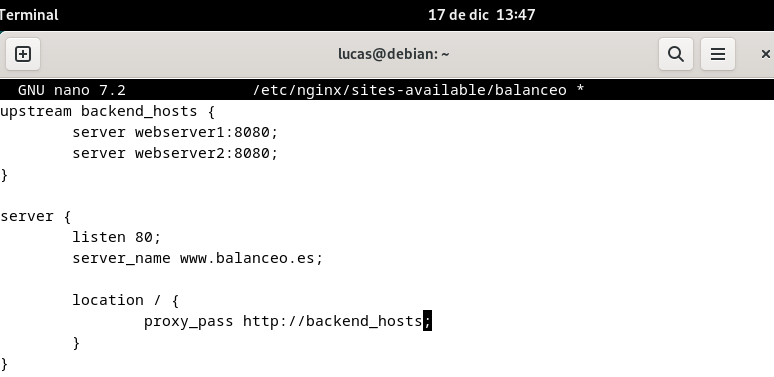
Antes de reiniciar NGINX, se recomienda encender ambas máquinas servidoras para evitar problemas de conectividad. Una vez reiniciado el servicio, se comprobará que todo funciona correctamente.



Una vez realizado este paso, procederemos a cambiar el nombre del archivo "estatica" ubicado en el directorio "/etc/nginx/sites-available" por "balanceo". Para ello, ejecutaremos el siguiente comando en la terminal:



Ahora cambiaremos el código para que quede así:

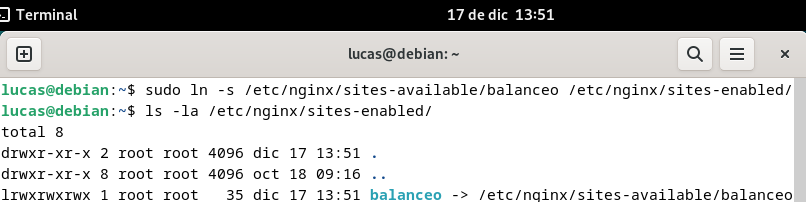


La sección upstream denominada backend\_hosts contiene la lista de servidores a los que se distribuirán las solicitudes entrantes al proxy. En nuestro caso, estos servidores son webserver1 y webserver2, ambos escuchando en el puerto 8080.

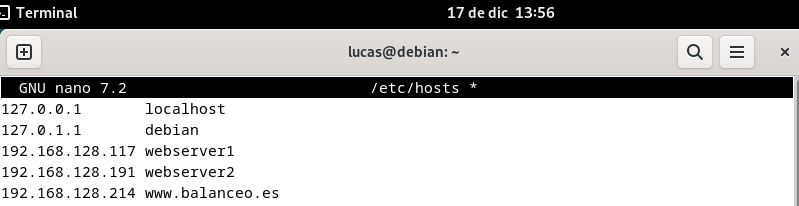
Esta sección utiliza el método de balanceo de carga random, que selecciona de forma aleatoria a cuál de los servidores enrutaremos cada solicitud. Es decir, cada vez que llega una petición, el proxy decide si la envía a webserver1:8080 o a webserver2:8080.

En la configuración del servidor, especificamos el puerto en el que escuchará el proxy, el nombre de dominio asociado y, en la ubicación raíz (/), utilizamos la directiva proxy\_pass para delegar las solicitudes al bloque upstream, permitiendo así la distribución aleatoria de la carga.

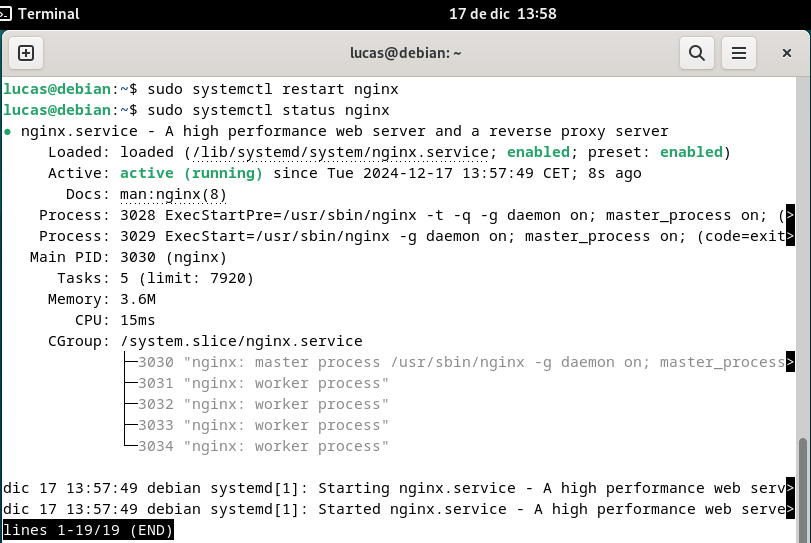
Ahora crearemos el enlace simbólico a este archivo en el directorio /etc/nginx/sites-enabled/



A continuación, debemos modificar el archivo de configuración de red /etc/hosts para añadir las IP de los servidores webserver1 y webserver2. Esto permitirá que el proxy los localice en la red.

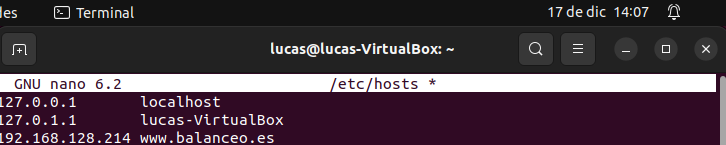


Ahora, con ambas máquinas servidor encendidas para evitar un error de host inalcanzable, reiniciamos el servidor web Nginx. Luego, comprobaremos si está funcionando correctamente y si los cambios se han aplicado.



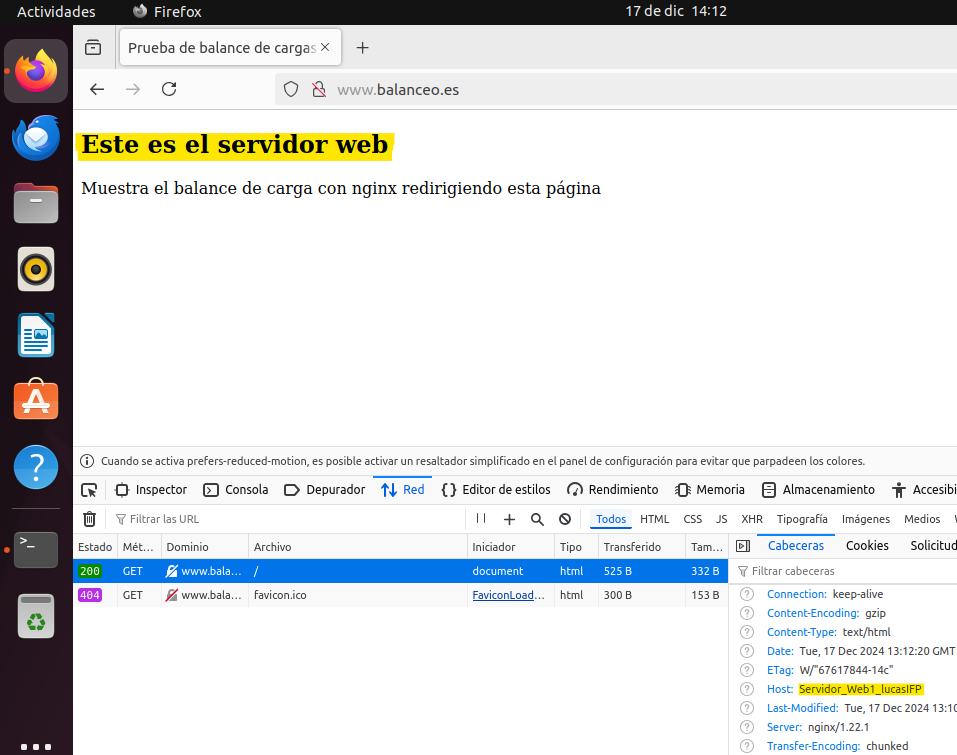
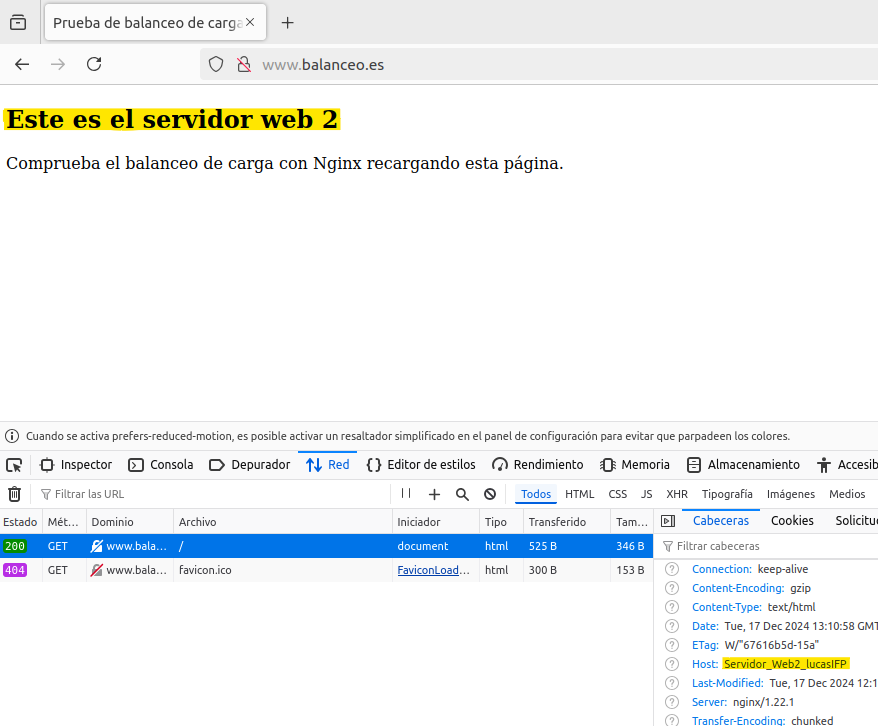
# Comprobación del funcionamiento del balanceo de carga

Procederemos a verificar el funcionamiento y la conexión a los dos servidores utilizando la máquina virtual cliente con Ubuntu instalado. Lo primero que debemos hacer es agregar la dirección IP del proxy al archivo /etc/hosts de la máquina cliente.



# AMBOS SERVIDORES OPERATIVOS

Primero verificaremos el funcionamiento del sitio web con ambos servidores en funcionamiento. Si accedemos al sitio web desde el navegador del equipo anfitrión, seremos dirigidos de forma aleatoria al servidor web 1 o al servidor web 2. Para una comprobación adicional, podemos utilizar las herramientas de desarrollador y revisar la cabecera personalizada que recibimos.

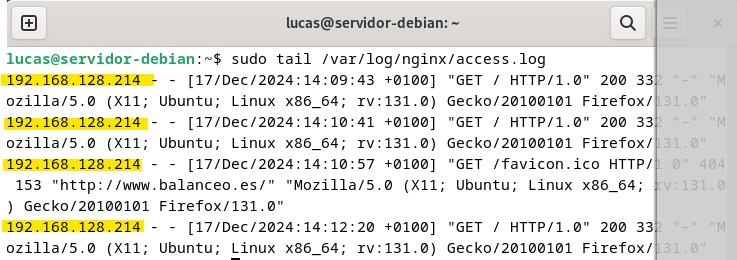
Servidor 1: Servidor 2:

Si revisamos los registros de acceso, podremos observar, como en la práctica anterior, que el cliente se conecta al proxy, y este, a su vez, se conecta con los servidores 1 y 2. Para acceder a los registros, podemos usar el siguiente comando:

sudo tail /var/log/nginx/access.log

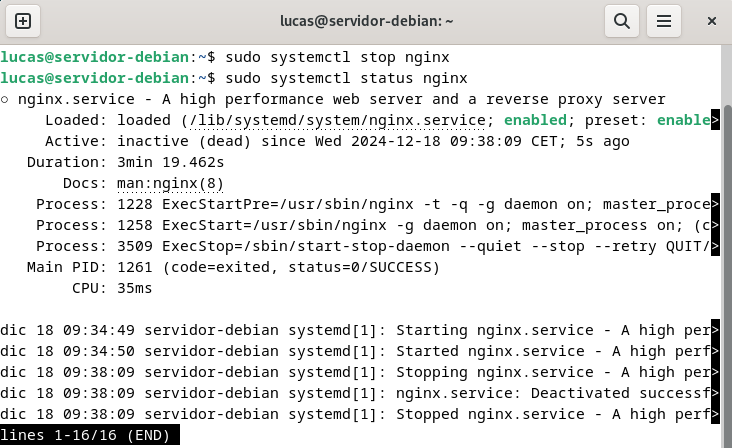


Ahora en el log de acceso del servidor 1 nos saldrán las peticiones desde nuestro proxy con ip 192.168.128.214:

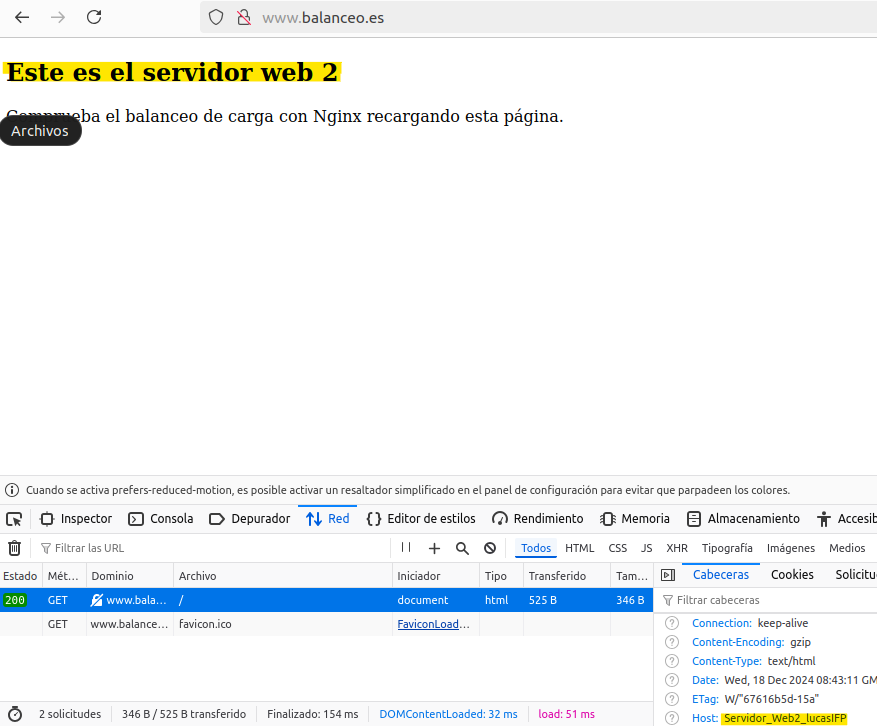
Y lo mismo para el servidor 2: 

# SERVIDOR WEB 1 NO OPERATIVO

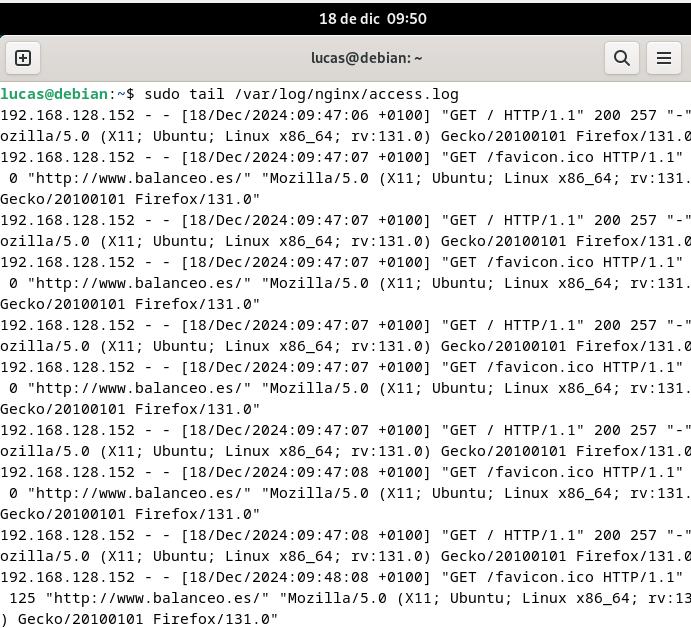
Simularemos una situación en la que el servidor web 1 ha caído para verificar que todas las solicitudes a nuestro proxy se redirijan al servidor web 2. Para ello, detendremos el servidor web 1 utilizando el siguiente comando de systemctl:  
 *sudo systemctl stop nginx.service*



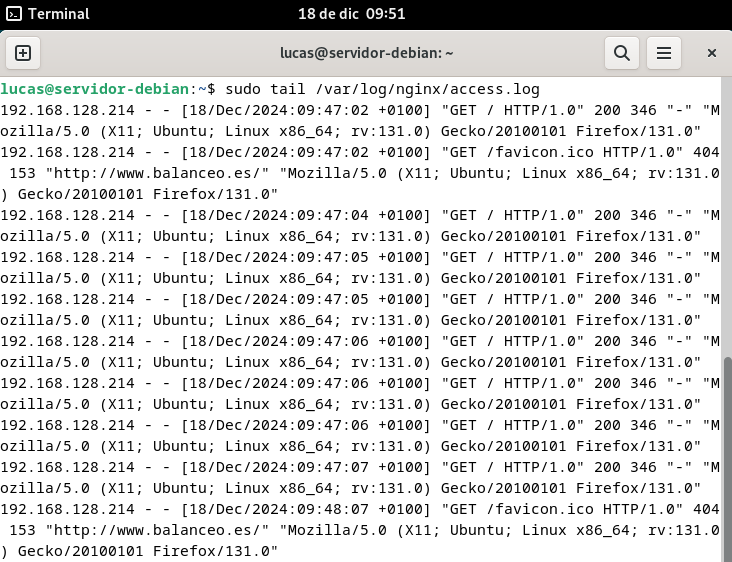
Ahora al buscar el proxy solo redireccionará al cliente al servidor web 2



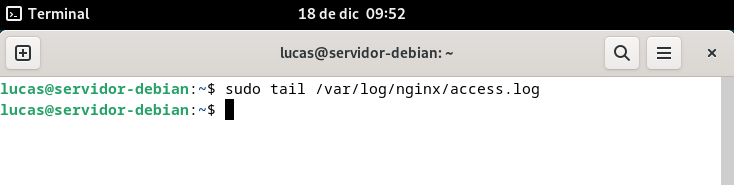
Ahora comprobamos mediante los logs que, en efecto, ninguna de las peticiones ha llegado al servidor 1:

Logs del proxy:  


del servidor web 2:



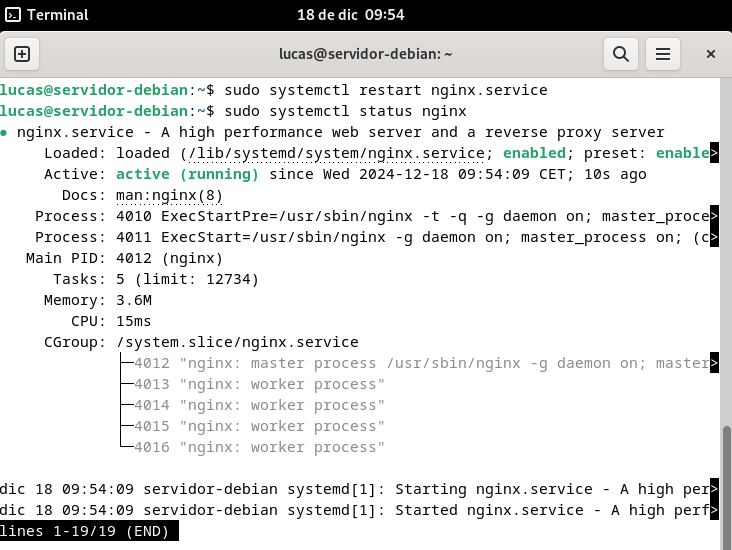
del servidor web 1:



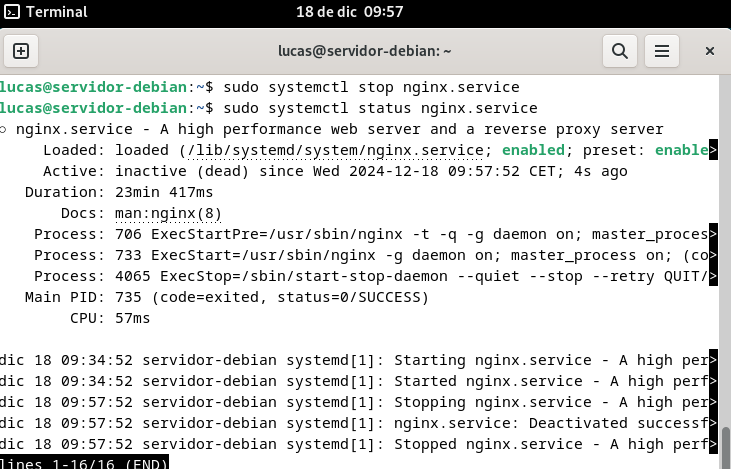
Como vemos en esta imagen, el servidor web 1 no ha recibido ninguna petición.

# SERVIDOR WEB 2 NO OPERATIVO

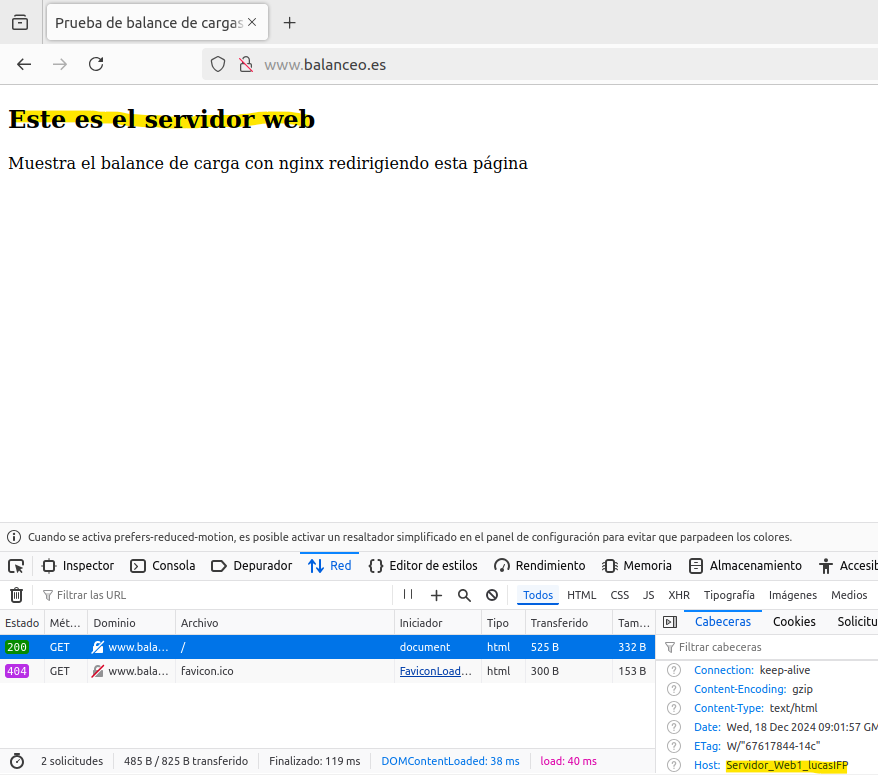
Lo primero que haremos será reiniciar nginx del servidor web 1:



Luego paralizaremos el servidor web 2:

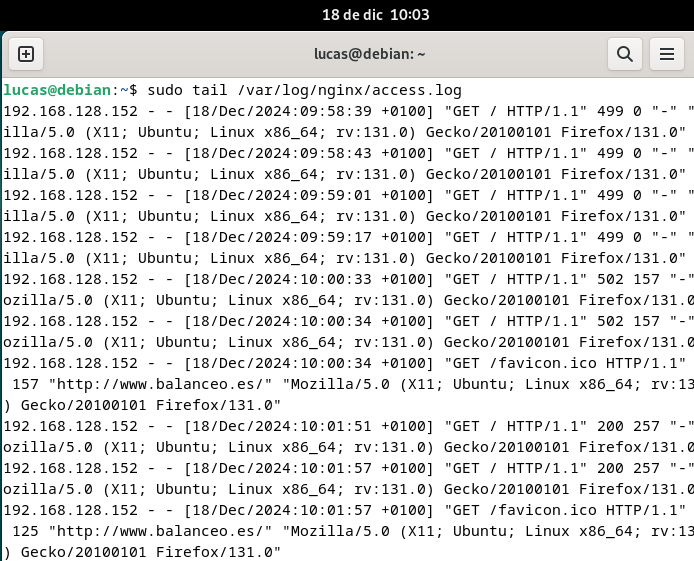


Ahora el proxy redireccionará al cliente al servidor web 1:

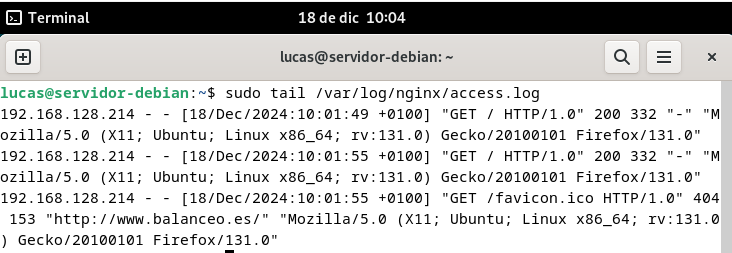


Ahora comprobamos mediante los logs que, en efecto, ninguna de las peticiones ha llegado al servidor 2:

los del proxy:

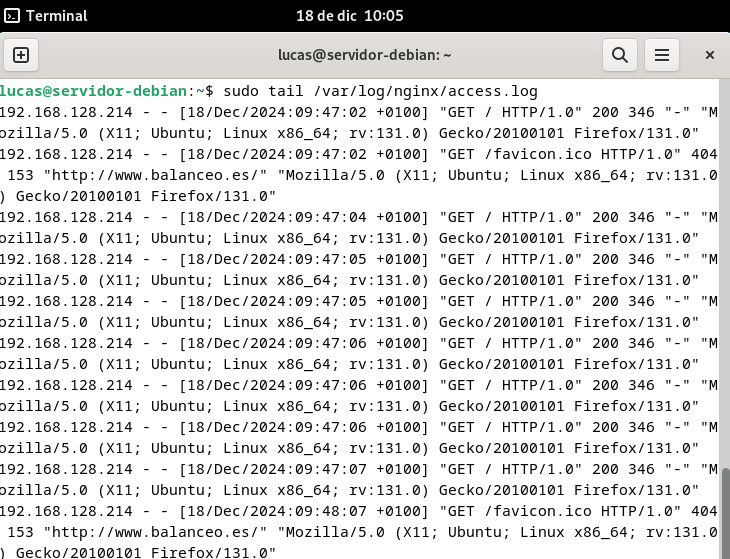


los del servidor 1 que ahora SÍ recibe peticiones:



y los del servidor web 2, el cual no ha recibido ninguna peticion nueva desde el

anterior punto:



### **Conclusión**

El uso de un proxy inverso en NGINX como balanceador de carga proporciona una solución eficaz para distribuir tráfico entre múltiples servidores. Las pruebas realizadas han demostrado que, en caso de fallo de uno de los servidores, el sistema redirige correctamente el tráfico al servidor en funcionamiento, garantizando la continuidad del servicio.