

PRÁCTICA 4

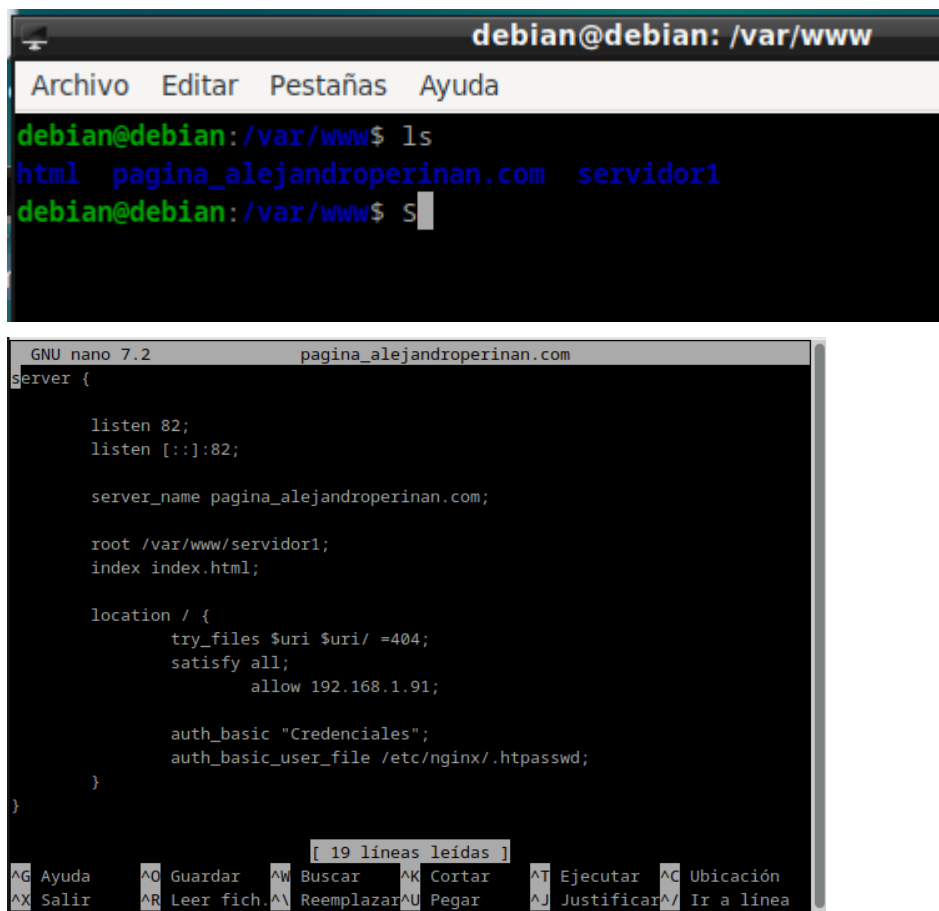
Configuración de Nginx como balanceador de carga

En esta práctica queremos configurar Nginx como Proxy Inverso, siguiendo el esquema dibujado en la pizarra. El objetivo es que el servidor que se encuentra al frente, sea capaz de repartir las solicitudes de los clientes entre los distintos servidores origen que podamos tener. Ya hemos comentado las ventajas que nos aporta realizar el balanceo de carga.

Habría que realizar otro clon de nuestra primera máquina virtual (la borraremos una vez terminadas las prácticas 4 y 5). Al clonarla, debemos indicarle que utilice una nueva MAC, para que así el proxy pueda obtener una dirección IP diferente al servidor.

En la máquina virtual original (la primera debian que instalamos), vamos a crear una nueva página, que se llame servidor 1. Debe servir una página donde se muestre claramente qué servidor es. En la máquina que hemos creado en el segundo párrafo, crearemos otra página servidor 2. Esta página deberá mostrar que está siendo servida por el servidor 2.

Es conveniente que configuremos las IPs de las dos máquinas para que sean estáticas y no tengamos que estar cambiando la configuración del balanceador.



```
debian@debian: /var/www
Archivo  Editar  Pestañas  Ayuda
debian@debian:/var/www$ ls
html  pagina_alejandroperinan.com  servidor1
debian@debian:/var/www$ s

GNU nano 7.2  pagina_alejandroperinan.com
server {

    listen 82;
    listen [::]:82;

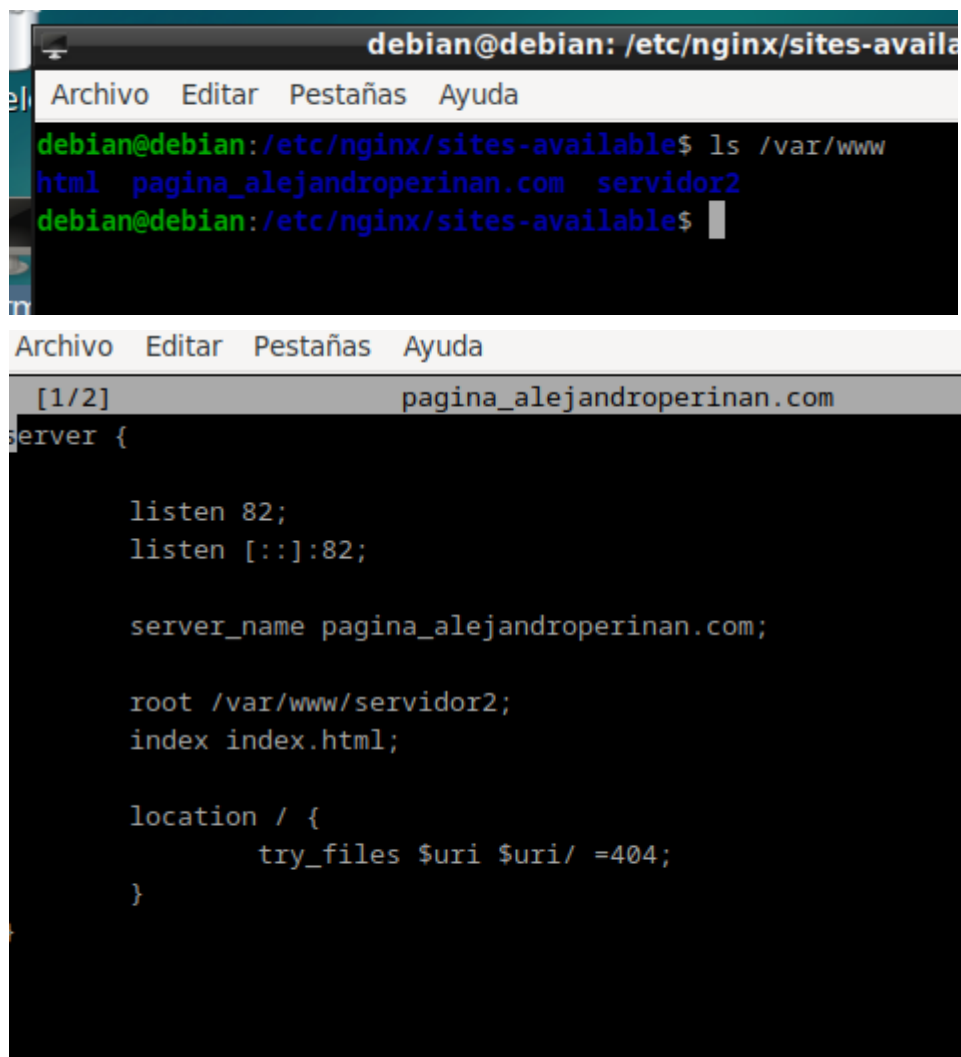
    server_name pagina_alejandroperinan.com;

    root /var/www/servidor1;
    index index.html;

    location / {
        try_files $uri/ =404;
        satisfy all;
        allow 192.168.1.91;

        auth_basic "Credenciales";
        auth_basic_user_file /etc/nginx/.htpasswd;
    }
}

[ 19 líneas leídas ]
^G Ayuda  ^O Guardar  ^W Buscar  ^K Cortar  ^T Ejecutar  ^C Ubicación
^X Salir  ^R Leer fich.  ^\ Reemplazar  ^U Pegar  ^J Justificar  ^_ Ir a línea
```



```
debian@debian: /etc/nginx/sites-available
Archivo  Editar  Pestañas  Ayuda
debian@debian:/etc/nginx/sites-available$ ls /var/www
html  pagina_alejandroperinan.com  servidor2
debian@debian:/etc/nginx/sites-available$

[1/2] pagina_alejandroperinan.com
server {

    listen 82;
    listen [::]:82;

    server_name pagina_alejandroperinan.com;

    root /var/www/servidor2;
    index index.html;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Con esto configuramos los 2 servidores.

Nos queda realizar la configuración del balanceador (será el proxy_inverso de la práctica anterior). Sigue el siguiente código:

```
upstream backend_hosts {
    random;
    server (IP y puerto del servidor 1);
    server (IP y puerto del servidor 2);
}

server {
```

```

listen 82;
server_name (IP del proxy);

location / {
    proxy_pass http://backend_hosts;
}
}

```

The screenshot shows a terminal window titled 'debian@debian: /etc/nginx/sites-available'. Inside, the GNU nano 7.2 editor is open, editing a file named 'pagina_alejandroperinan.com'. The configuration file contains the following content:

```

upstream backend_hosts {
    random;
    server 192.168.1.58:24;
    server 192.168.1.74:24;
}

server {

    listen 82;
    listen [::]:82;

    server_name pagina_alejandroperinan.com;

    location / {
        proxy_pass http://backend_hosts;
    }
}

```

At the bottom of the terminal, a status bar indicates '[16 lineas leidas]' and provides keyboard shortcuts for various editor functions like Ayuda, Guardar, Buscar, Cortar, Ejecutar, Ubicación, Salir, Leer fich, Reemplazar, Pegar, Justificar, and Ir a línea.

¿Qué otras opciones tenemos aparte de random en el bloque de los servidores del backend? ¿Para qué sirven las principales?

round-robin: Este es el método de equilibrio de carga predeterminado en Nginx y distribuye las solicitudes en un ciclo secuencial a través de los servidores en el grupo.

least_conn: Dirige las solicitudes al servidor con la menor cantidad de conexiones activas en ese momento.

ip_hash: Utiliza la dirección IP del cliente para determinar a qué servidor enviar la solicitud. Esto garantiza que las solicitudes del mismo cliente siempre se dirijan al mismo servidor, útil para sesiones persistentes.

least_time: Selecciona el servidor con el tiempo de respuesta más corto o la menor carga, dependiendo de los parámetros que especifiques.

Accede al navegador de la máquina cliente. Busca la dirección IP del balanceador y recarga la página. ¿Por qué cambia?

Por qué de forma aleatoria cada vez que recargamos la página el balanceador de carga nos redirige a un servidor de los 2 que tenemos, que cada uno tiene su IP