

Balanceo de carga con proxy inverso en Nginx

Para poder acceder al proxy inverso con el FDQN necesitaremos poner en los archivos de los servidores el nombre de dominio por el cual queremos acceder:

Servidor Nginx:

```
a24gadilu@debian:~$ cat /etc/nginx/sites-available/a24gadilu
server {
    listen 8080;
    server_name www.a24gadilu.local;

    location / {
        proxy_pass http://192.168.201.10:80;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Servidor Apache:

```
a24gadilu@debian:/etc/apache2/sites-available$ cat a24gadilu.conf
<VirtualHost *:80>
    ServerName www.a24gadilu.local
    DocumentRoot /var/www/a24gadilu/html/static-website-example
    <Directory /var/www/a24gadilu/html/static-website-example>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Para realizar el balance de carga deberemos tener dos servidores de apache para ello clonaremos la máquina que tenemos de apache ya configurada previamente.

Deberemos tener instalado php en las dos máquinas que contienen apache:

Usaremos el siguiente comando:

“sudo apt install php”

```
a24gadilu@debian:~$ sudo apt install php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  nginx-common
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.2 libsodium23 php-common php8.2 php8.2-cli php8.2-common php8.2-opcache php8.2-
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php8.2 libsodium23 php php-common php8.2 php8.2-cli php8.2-common php8.2-opcache php
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 4.677 kB de archivos.
Se utilizarán 21,7 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://security.debian.org/debian-security bookworm-security/main amd64 php8.2-common amd64 8.2
kB]
Des:2 http://deb.debian.org/debian bookworm/main amd64 php-common all 2:93 [13,1 kB]
Des:3 http://deb.debian.org/debian bookworm/main amd64 libsodium23 amd64 1.0.18-1 [161 kB]
Des:4 http://deb.debian.org/debian bookworm/main amd64 php all 2:8.2+93 [3.628 B]
Des:5 http://security.debian.org/debian-security bookworm-security/main amd64 php8.2-opcache amd64 8.2
```

Yo ya tenía instalado php en la máquina 1 de apache, como ha sido la máquina clonada en la segunda máquina de apache ya tengo php instalado.

Ahora deberemos acceder al archivo de configuración de apache para asegurarnos que lo primero que se va a ejecutar sea el index.php y no el index.html. Accedemos a la ruta del archivo

“sudo nano /etc/apache2/mods-enabled/dir.conf”

```
GNU nano 7.2 /etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
```

Guardamos y reiniciamos apache.

Ahora deberemos crear dos ficheros index.php en cada uno de nuestros servidores de apache que contengan el siguiente contenido:

Deberemos crearlos en la ruta /var/www/a24gadilu/html

Servidor apache1:

```
GNU nano 7.2
<?php
echo "<h1>Tabla del 1</h1>";
for ($i = 1; $i <= 10; $i++) {
    echo "1 x $i = " . (1 * $i) . "<br>";
}
?>
```

Servidor apache 2:

```
GNU nano 7.2
<?php
echo "<h1>Tabla del 2</h1>";
for ($i = 1; $i <= 10; $i++) {
    echo "2 x $i = " . (2 * $i) . "<br>";
}
?>
```

Deberemos cambiar la ruta DocumentRoot de configuración de nuestros servidores apache, que es donde se encontraran nuestros archivos php:

```
GNU nano 7.2 /etc/apache2/sites-available/a24gadilu.conf *
<VirtualHost *:80>
    ServerName www.a24gadilu.local
    DocumentRoot /var/www/a24gadilu/html
    <Directory /var/www/a24gadilu/html>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Ahora configuraremos el servidor de nginx que actúa como proxy inverso para hacer el balanceo de carga.

Deberemos editar el archivo de configuración del servidor:

“sudo nano /etc/nginx/sites-available/a24gadilu”

Donde añadiremos las direcciones ip de los servidores apache y donde deberemos cambiar la ruta del proxy inverso y redirigirlo a nuestros servidores apache.

```
GNU nano 7.2 /etc/nginx/sites-available/a24gadilu
upstream apache_servers {
    server 192.168.201.10;
    server 192.168.201.15;
}

server {
    listen 8080;
    server_name www.a24gadilu.local;

    location / {
        proxy_pass http://apache_servers;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Una vez terminado miraremos los logs de cada servidor con el siguiente comando “sudo tail -f /var/log/apache2/access.log /var/log/apache2/error.log”:

Servidor 1:

```
a24gadilu@debian:~$ sudo tail -f /var/log/apache2/access.log /var/log/apache2/error.log
==> /var/log/apache2/access.log <==
==> /var/log/apache2/error.log <==
[Sun Oct 20 19:53:20.988308 2024] [core:notice] [pid 1403:tid 1403] AH00094: Command line: '/usr/sbin/apache2'
[Sun Oct 20 19:57:33.667155 2024] [mpm_prefork:notice] [pid 1403:tid 1403] AH00169: caught SIGTERM, shutting down
[Sun Oct 20 19:57:34.732430 2024] [mpm_prefork:notice] [pid 1457:tid 1457] AH00163: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Sun Oct 20 19:57:34.732463 2024] [core:notice] [pid 1457:tid 1457] AH00094: Command line: '/usr/sbin/apache2'
[Sun Oct 20 20:17:30.285043 2024] [mpm_prefork:notice] [pid 1457:tid 1457] AH00169: caught SIGTERM, shutting down
[Sun Oct 20 20:17:31.347022 2024] [mpm_prefork:notice] [pid 1569:tid 1569] AH00163: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Sun Oct 20 20:17:31.347050 2024] [core:notice] [pid 1569:tid 1569] AH00094: Command line: '/usr/sbin/apache2'
[Sun Oct 20 20:50:07.430827 2024] [mpm_prefork:notice] [pid 1569:tid 1569] AH00169: caught SIGTERM, shutting down
[Sun Oct 20 20:50:08.500553 2024] [mpm_prefork:notice] [pid 1789:tid 1789] AH00163: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Sun Oct 20 20:50:08.500586 2024] [core:notice] [pid 1789:tid 1789] AH00094: Command line: '/usr/sbin/apache2'
```

Servidor 2:

```
a24gadilu@debian:/var/www/a24gadilu/html$ sudo tail -f /var/log/apache2/access.log /var/log/apache2/error.log
==> /var/log/apache2/access.log <==
==> /var/log/apache2/error.log <==
[Sun Oct 20 19:56:32.828669 2024] [core:notice] [pid 1792:tid 1792] AH00094: Command line: '/usr/sbin/apache2'
[Sun Oct 20 20:17:30.194975 2024] [mpm_prefork:notice] [pid 1792:tid 1792] AH00169: caught SIGTERM, shutting down
[Sun Oct 20 20:17:31.324395 2024] [mpm_prefork:notice] [pid 2075:tid 2075] AH00163: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Sun Oct 20 20:17:31.324474 2024] [core:notice] [pid 2075:tid 2075] AH00094: Command line: '/usr/sbin/apache2'
[Sun Oct 20 20:25:57.337885 2024] [mpm_prefork:notice] [pid 2075:tid 2075] AH00169: caught SIGTERM, shutting down
[Sun Oct 20 20:25:58.476198 2024] [mpm_prefork:notice] [pid 2198:tid 2198] AH00163: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Sun Oct 20 20:25:58.476248 2024] [core:notice] [pid 2198:tid 2198] AH00094: Command line: '/usr/sbin/apache2'
[Sun Oct 20 20:45:11.664511 2024] [mpm_prefork:notice] [pid 2198:tid 2198] AH00169: caught SIGTERM, shutting down
[Sun Oct 20 20:46:08.495321 2024] [mpm_prefork:notice] [pid 2406:tid 2406] AH00163: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Sun Oct 20 20:46:08.495379 2024] [core:notice] [pid 2406:tid 2406] AH00094: Command line: '/usr/sbin/apache2'
```

Ahora si accedemos al navegador y ponemos nuestra ruta www.a24gadilu.local:8080 accederemos a el primer servidor donde nos mostrara la tabla de multiplicar del 1 y si recargamos la página accederemos al segundo servidor donde nos mostrara la tabla de multiplicar del 2.

Servidor 1:

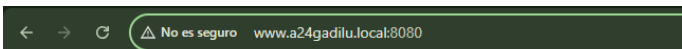


Tabla del 1

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
```

Servidor 2:

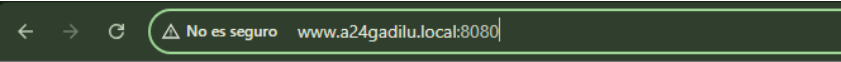


Tabla del 2

- 2 x 1 = 2
- 2 x 2 = 4
- 2 x 3 = 6
- 2 x 4 = 8
- 2 x 5 = 10
- 2 x 6 = 12
- 2 x 7 = 14
- 2 x 8 = 16
- 2 x 9 = 18
- 2 x 10 = 20