

## Balanceo de carga



## Índice

<b>Descripción.....</b>	<b>3</b>
Finalidad del servicio:.....	3
Máquinas en las que se aloja:.....	4
Paquetes usados y versión de los mismos:.....	4
<b>Configuración.....</b>	<b>5</b>
Configuración previa:.....	5
Configuración del servicio.....	8
<b>Verificación del funcionamiento.....</b>	<b>12</b>
Pruebas y comprobaciones.....	12
<b>Alternativas.....</b>	<b>13</b>
Fuentes.....	13

## **Descripción**

### **Finalidad del servicio:**

En esta práctica utilizaremos Nginx para balancear la carga entre dos servidores Apache2 upstream que están funcionando en el backend de la infraestructura. Gracias a este balanceo, mediante el algoritmo round-robin con peso, podemos decidir cada cuantas peticiones a la web cambiaremos de servidor, de manera que repartimos la carga equitativamente ya que el peso es "1" para los dos servidores, cuando un cliente se conecte a la web de balanceo.com (realmente es pro.ausiasmarch.es:52228 porque hay redireccionamiento de puertos para que funcione desde el "exterior") esta le redirigirá a "Server 1" y que el siguiente que entre a la web se redirija a "Server 2".

En las máquinas del backend, se alojan los Apache2 que, localmente, apuntan al puerto 80 por defecto, y utilizan las rutas de configuración por defecto, lo único que hemos cambiado es el index.html que se aloja en la ruta "/var/www/html" y añadido "Server1" y en el otro Apache2 "Server 2".

Cabe recalcar que las peticiones están llegando solo a una IP y que esta está solamente repartiendo trabajo hasta este punto, pero hay una regla más que hemos aplicado a este trabajo. El servidor con Nginx corriendo por detrás, está configurado para que también sirva una página web estática, que en este caso la hemos llamado ed2estatica.com (el nombre lo dice todo), y esta a diferencia de la anterior, está en otro puerto para que pueda haber redireccionamiento de puertos y no haya confusión a la hora de acceder a las webs. Para eso, he necesitado aplicar un cambio al puerto local por el que está funcionando, en vez del 80 ahora será el 9999, así en el pfsense se puede validar como "otro puerto" y reenviar a esa página estática bajo el nombre de "pro.ausiasmarch.es:52229".

En caso de que una regla estuviese apuntando al puerto 52229, por ejemplo, habría un conflicto de redirección, solo funcionaria balanceo.com. Pero, más adelante, en alternativas explico una forma un poco más simple que se podría haber adoptado para conectar con las urls correctas.

**Máquinas en las que se aloja:**

Resource Pool: Ed2SRI		
Summary	Add Remove	
Members	Type ↑	Description
Permissions	lxc	120 (NginxE2)
	lxc	121 (Backend1)
	lxc	122 (Backend2)

Para esta práctica necesitaremos 3 contenedores, aunque también se pueden utilizar VMs, pero para agilizar nosotros nos hemos decantado por los CTs. Tenemos “NginxE2” que será nuestro balanceador de carga, “Backend1” y “Backend2” (upstream) serán los contenedores que por detrás estén corriendo Apache2.

Además, hemos configurado los “Ids” de los CTs de manera que sean la dirección de host de la IP correspondiente, o sea, “NginxE2” será la IP “172.16.1.120”, “Backend1” será la IP “172.16.1.121” y “Backend2” será la IP “172.16.1.122”.

**Paquetes usados y versión de los mismos:**

```
root@NginxE2:/etc/nginx/conf.d# nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
```

Tenemos la última versión de Nginx instalado.

```
root@Backend1:~# apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built: 2023-10-26T13:44:44
```

Tenemos la última versión de Apache2 instalado.

```
root@NginxE2:/etc/nginx/conf.d# curl -V
curl 7.81.0 (x86_64-linux-gnu) libcurl/7.81.0
```

Tenemos la versión 7.81.0 de curl, que lo utilizamos mucho para comprobar el funcionamiento de las webs en local desde consola.

## Configuración

### Configuración previa:

#### NginxE2:

Container 120 (NginxE2) on node 'PROXMOX7nd-dl' No Tags

Summary Add Edit Remove Volume Action Revert

Console

Resources

Network

DNS

Options

Task History

Backup

Replication

Snapshots

Firewall

Permissions

Memory	512.00 MiB
Swap	512.00 MiB
Cores	1
Root Disk	local:120/vm-120-disk-0.raw,size=6G

Edit: Network Device (veth)

Name: eth0 IPv4: ☐ Static ☒ DHCP

MAC address: 56:AC:59:1D:5F:BE IPv4/CIDR: 172.16.1.120/24

Bridge: vmbr1 Gateway (IPv4): 172.16.1.254


VLAN Tag: no VLAN IPv6: ☐ Static ☒ DHCP ☐ SLAAC

Firewall: ☒ IPv6/CIDR: None

Gateway (IPv6):

Help Advanced ☐ OK Reset

## Backend1:

Container 121 (Backend1) on node 'PROXMOX7nd-dl' No Tags 

Summary

> Console

Resources

Network

DNS

Options

Task History


Backup

Replication

Snapshots


Firewall

Permissions

Add 


Edit

Remove


Volume Action 

Revert

Memory	512.00 MiB
Swap	512.00 MiB
Cores	1
Root Disk	local:121/vm-121-disk-0.raw,size=6G

Edit: Network Device (veth) 

Name:	<input type="text" value="eth0"/>	IPv4:	<input type="radio"/> Static <input checked="" type="radio"/> DHCP
MAC address:	<input type="text" value="8A:CB:99:E1:54:5C"/>	IPv4/CIDR:	<input type="text" value="172.16.1.121/24"/>
Bridge:	<input type="text" value="vmbr1"/>	Gateway (IPv4):	<input type="text" value="172.16.1.254"/>
VLAN Tag:	<input type="text" value="no VLAN"/>	IPv6:	<input type="radio"/> Static <input checked="" type="radio"/> DHCP <input type="radio"/> SLAAC
Firewall:	<input checked="" type="checkbox"/>	IPv6/CIDR:	<input type="text" value="None"/>
		Gateway (IPv6):	<input type="text"/>


 Help






















Advanced ☐


OK

Reset


## Backend2:

Container 122 (Backend2) on node 'PROXMOX7nd-df' No Tags 

 Summary	 Add  Edit  Remove  Volume Action  Revert
> Console	 Memory 512.00 MiB
 Resources	 Swap 512.00 MiB
 Network	 Cores 1
 DNS	 Root Disk local:122/vm-122-disk-0.raw,size=6G
 Options	
 Task History	
 Backup	
 Replication	
 Snapshots	
 Firewall 	
 Permissions	

Edit: Network Device (veth) 

Name:	<input type="text" value="eth0"/>	IPv4: <input type="radio"/> Static <input checked="" type="radio"/> DHCP
MAC address:	<input type="text" value="A6:4F:D2:C9:43:A9"/>	IPv4/CIDR: <input type="text" value="172.16.1.122/24"/>
Bridge:	<input type="text" value="vmbr1"/>	Gateway (IPv4): <input type="text" value="172.16.1.254"/>
VLAN Tag:	<input type="text" value="no VLAN"/>	IPv6: <input type="radio"/> Static <input checked="" type="radio"/> DHCP <input type="radio"/> SLAAC
Firewall:	<input checked="" type="checkbox"/>	IPv6/CIDR: <input type="text" value="None"/>
		Gateway (IPv6): <input type="text"/>

 Help Advanced ☐

Ed2 Balanceador de carga												
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP/UDP	*	*	WAN address	52230	172.16.1.120	22 (SSH)	Acceso SSH a NginxEd2	
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP/UDP	*	*	WAN address	52231	172.16.1.121	22 (SSH)	Acceso SSH a Backend1	
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP/UDP	*	*	WAN address	52232	172.16.1.122	22 (SSH)	Acceso SSH a Backend2	

Para poder empezar a trabajar con los contenedores, yo he creado unas reglas en el Pfsense para poder operar desde ssh...

```
root@NginxEd2:~# cat /etc/ssh/sshd_config | grep "PermitRootLogin" | head -1
PermitRootLogin yes
```

Y esta línea del archivo sshd\_config en la ruta “/etc/ssh/” hay que editarlo para que quede de la manera que se muestra en la imagen, porque de lo contrario no podremos conectarnos por ssh a nuestro CT puesto que accedemos como root y necesitamos este permiso especial.

## Configuración del servicio

### Backend1 y Backend2:

```
root@Backend1:~# systemctl status apache2
* apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2024-01-28 16:40:29 UTC; 7min ago
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 1236 (apache2)
    Tasks: 55 (limit: 19067)
  Memory: 7.5M
    CPU: 48ms
  CGroup: /system.slice/apache2.service
          |-1236 /usr/sbin/apache2 -k start
          |-1238 /usr/sbin/apache2 -k start
          ~-1239 /usr/sbin/apache2 -k start

Jan 28 16:40:29 Backend1 systemd[1]: Starting The Apache HTTP Server...
Jan 28 16:40:29 Backend1 systemd[1]: Started The Apache HTTP Server.
```

```
root@Backend2:~# systemctl status apache2
* apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2024-01-28 16:40:48 UTC; 6min ago
    Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 1231 (apache2)
    Tasks: 55 (limit: 19067)
  Memory: 5.5M
    CPU: 43ms
  CGroup: /system.slice/apache2.service
          |-1231 /usr/sbin/apache2 -k start
          |-1233 /usr/sbin/apache2 -k start
          ~-1234 /usr/sbin/apache2 -k start

Jan 28 16:40:48 Backend2 systemd[1]: Starting The Apache HTTP Server...
Jan 28 16:40:48 Backend2 systemd[1]: Started The Apache HTTP Server.
```

Nos aseguramos que instalamos Apache2 con “apt update && apt install apache2 -y” y el servicio está corriendo exitosamente (se comprueba con “systemctl status apache2”).



```
root@Backend1:/etc/apache2/sites-enabled# cat 000-default.conf | grep "DocumentRoot"
    DocumentRoot /var/www/html
root@Backend1:/etc/apache2/sites-enabled# cat /var/www/html/index.html
<h1>Bienvenido al Servidor 1 de Ed2</h1>
```

```
root@Backend2:/etc/apache2/sites-enabled# cat 000-default.conf | grep "DocumentRoot"
    DocumentRoot /var/www/html
root@Backend2:/etc/apache2/sites-enabled# cat /var/www/html/index.html
<h1>Bienvenido al Servidor 2 de Ed2</h1>
```

Nos aseguramos que las rutas por defecto de los índices que tienen los archivos de configuración de Apache2 en los servidores backend está establecida como `"/var/www/html"` ya que ahí está el índice que nosotros hemos modificado con un `"h1"` para que aparezca un mensaje de bienvenida de `"Server1"` y `"Server2"`.

Con el cat mostramos el contenido que hay en los “index.html”. Además, cabe mencionar que los archivos de configuración están en la ruta de “sitios habilitados” de Apache2 (“/etc/apache2/sites-enabled”), ya que él toma la configuración de esta ruta por defecto.

## NginxEd2

**hosts:**

```
root@NginxEd2:~# cat /etc/hosts
127.0.0.1      localhost
#Balanceo de carga grupal
172.16.1.120   balanceo.com ed2estatica.com
```

Para empezar la configuración del servicio de Nginx, editamos el archivo de hosts para que nuestra IP (172.16.1.120) resuelva a los dominios que indiquemos, en este caso, para diferenciar el balanceo de la estática, hemos nombrado a la página de balanceo como “balanceo.com” y a la estática como “ed2estatica.com”.

### Comprobación de funcionamiento:

[illegible]

Para instalar Nginx “apt update && apt install nginx -y” de esta manera no nos debería dar problemas de instalación ya que solo instalamos los paquetes necesarios. Y para comprobar el estado del proceso “systemctl status nginx”, en nuestro caso ya está listo.

**Configuración de balanceo.com.conf:**

```
root@NginxEd2:/etc/nginx/conf.d# cat balanceo.com.conf
upstream backend {
    server 172.16.1.121:80 weight=1;
    server 172.16.1.122:80 weight=1;
}
server {
    listen 80;
    server_name balanceo.com;

    location / {
        proxy_redirect off;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarder-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_pass http://backend;
    }
}
```

Esta es la configuración de nuestra página de “**balanceo.com**”, que en nuestro caso la hemos creado en la ruta “**/etc/nginx/conf.d**” ya que pensamos que es una ruta más apropiada para el alojamiento de configuraciones, y además solo se necesita un directorio para operar, cosa que en “**sites-enabled**” necesitas dos: “**sites-available**” y “**sites-enabled**”. Pero hay un requisito, que el archivo de configuración acabe con la extensión “.conf”.

En esta configuración hay varios aspectos que comentar, lo primero es el bloque “**upstream backend**” en el cuál está la configuración del upstream que hemos nombrado como “**backend**”. Las dos líneas que hay describen que los servidores “172.16.1.121” y “172.16.1.122” escuchan por el puerto local 80 y el algoritmo es **round-robin con peso**, que como comentamos anteriormente, repartirá el tráfico de manera equitativa con una petición a cada servidor... La primera petición será para el primer servidor, la segunda para el segundo servidor, la tercera para el primer servidor, la cuarta para el segundo servidor.

Lo siguiente a comentar es el bloque de “**server**”, que esta escuchando por el puerto local 80, el nombre del servidor es “**balanceo.com**”, la directiva “**location /**” define cómo Nginx debe manejar las solicitudes dirigidas a la raíz del sitio web.

**proxy\_redirect off:** Evita que Nginx modifique las ubicaciones en las cabeceras de redirección recibidas del servidor backend.

**proxy\_set\_header X-Real-IP \$remote\_addr:** Preserva la dirección IP real del cliente a medida que la solicitud pasa a través de Nginx y llega al servidor backend.

**proxy\_set\_header X-Forwarded-For \$proxy\_add\_x\_forwarded\_for:** “X-Forwarded-For” este encabezado se utiliza para rastrear la IP del cliente a través de los proxies. “\$proxy\_add\_x\_forwarded\_for” agrega la dirección IP del cliente al final de la cadena.

**proxy\_set\_header Host \$http\_host:** El servidor backend pueda determinar qué dominio o subdominio se está solicitando (Campo “Host” en la petición HTTP).

**proxy\_pass http://backend:** Especifica la URL del servidor backend al que Nginx debe enviar las solicitudes.

## Configuración de página estática:

## default.conf

```
root@NginxEd2:/etc/nginx/conf.d# cat default.conf
server {
    listen 9999;
    root /var/www/ed2estatica.com/html;
    index index.html;

    server_name ed2estatica.com;

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

Como se puede ver, hemos definido un bloque “server” que escucha por el puerto local **9999**, hemos usado este puerto porque es un puerto fácil de recordar a la hora de generar redireccionamiento de puertos más adelante en el Pfsense, es necesario hacer esto porque antes ya habíamos configurado el puerto 80, que para una red local está genial, pero para poder entrar a la web desde un cliente externo necesitamos especificar otro puerto y reenviarlo a uno de los puertos que tenemos disponibles en nuestro rango de Pfsense, de lo contrario habría conflicto de “Host”.

Lo siguiente es definir cuál será el directorio raíz del index “/var/www/ed2estatica.com/html”, el index será **index.html**, el nombre del dominio es “ed2estatica.com”, y en la directiva “location /” definimos que si no lo encuentra de un código de estado “**404 not found**”.










```
root@NginxEd2:/var/www# mkdir -p ed2estatica.com/html
root@NginxEd2:/var/www# cd !$
cd ed2estatica.com/html
root@NginxEd2:/var/www/ed2estatica.com/html# nano index.html
```

A continuación, con las reglas establecidas anteriormente, nos dirigimos a la ruta “/var/www” y creamos la ruta “ed2estatica.com/html”, en la que estará el “index.html”.

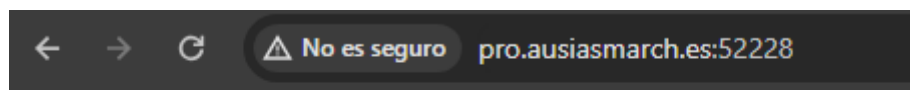
```
root@NginxEd2:/var/www/ed2estatica.com/html# cat index.html
<h1>Bienvenido a la pagina estatica de Ed2 >:v)</h1>
```

El contenido del “index.html”.

**Reglas HTTP Pfsense:**

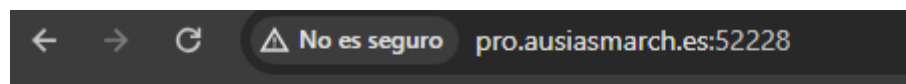
Ed2 Balanceador de carga												
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP/UDP	*	*	WAN address	52228	172.16.1.120	80 (HTTP)	Acceso HTTP a NginxEd2	  
<input type="checkbox"/>	<input checked="" type="checkbox"/>		WAN	TCP/UDP	*	*	WAN address	52229	172.16.1.120	9999	Acceso HTTP a NginxEd2	  

Aquí una imagen de las reglas establecidas en el Pfsense. La primera apunta al puerto destino **80**, que es en el que está alojado **“balanceo.com”** y la segunda apunta al puerto destino **9999**, en el que está alojado **“ed2estatica.com”**.

**Verificación del funcionamiento****Pruebas y comprobaciones**

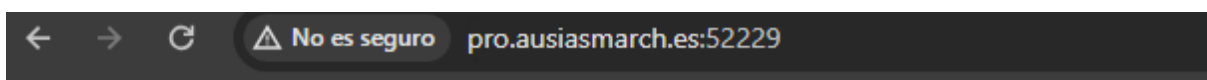
**Bienvenido al Servidor 1 de Ed2**

Entramos desde el navegador de nuestro host anfitrión a la url “pro.ausiasmarch.es:52228” (balanceo.com) y... Funciona!!! El redireccionamiento se configuró bien :).



**Bienvenido al Servidor 2 de Ed2**

Y al actualizar la página... Funciona el balanceo de carga!!



**Bienvenido a la pagina estatica de Ed2 >:v)**

Y cambiando el puerto al “52229”... También funciona la página estática!!

## **Alternativas**

Una de las medidas que se podía haber adoptado en vez de el doble redireccionamiento de puertos, uno para cada página web, se podría haber usado el comando curl para definir mediante la cabecera el dominio al que quiero ir, por ejemplo: “curl -H “Host: balanceo.com” pro.ausiasmarch.es:52228” o “curl -H “Host: ed2estatica.com” pro.ausiasmarch.es:52228”. Claro que suponiendo que el único puerto sería 52228, especificado en el Pfsense.

## **Fuentes**

**Github** de David Lopez Saorín —>

[https://github.com/david140104/Nginx-balanceo-de-carga/blob/main/balanceo\\_de\\_carga.pdf](https://github.com/david140104/Nginx-balanceo-de-carga/blob/main/balanceo_de_carga.pdf)