



UNIVERSIDAD
DE GRANADA

PRÁCTICA 3: BALANCEO DE CARGA EN UN SITIO WEB

ANA BUENDÍA RUIZ-AZUAGA

Práctica 3: Balanceo de carga en un sitio web

Correo electrónico

anabuenrúa@correo.ugr.es

E.T.S. INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, a 23 de abril de 2022

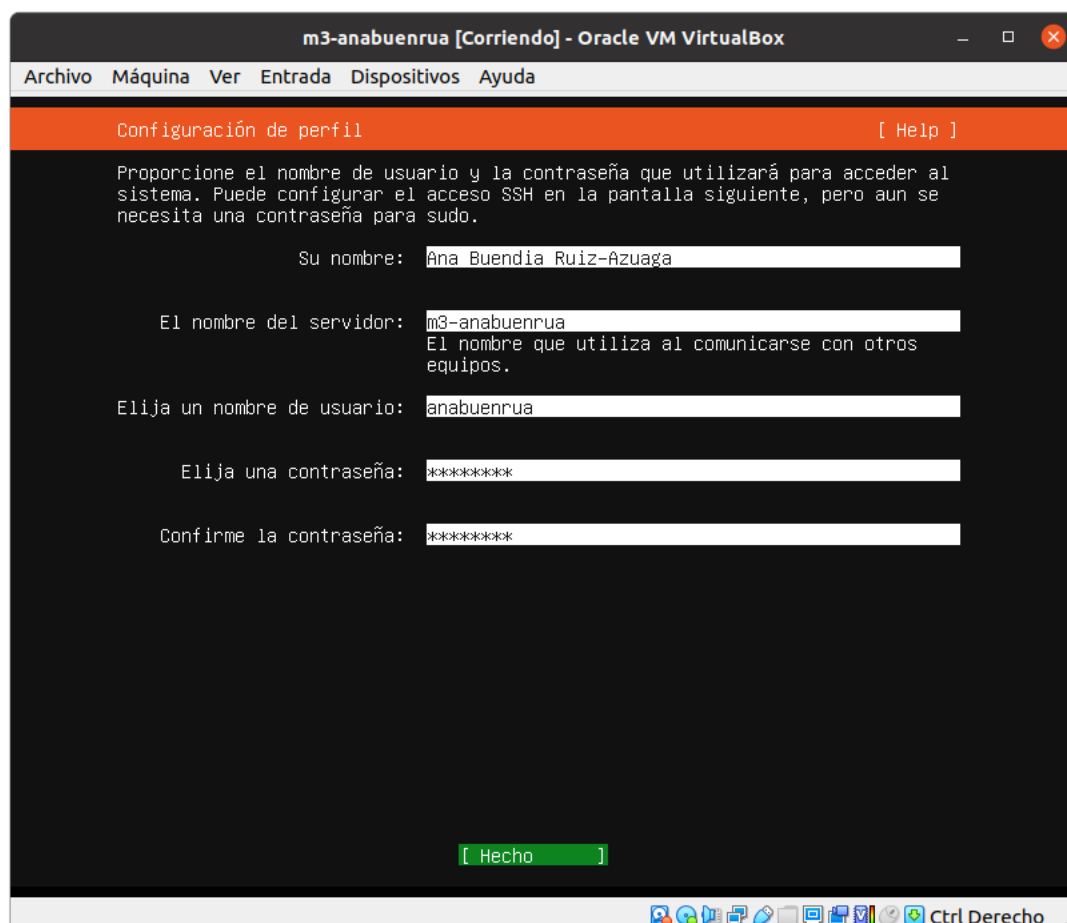
ÍNDICE GENERAL

1.	INSTALACIÓN	3
2.	NGINX	5
2.1.	Ponderación	6
2.2.	Otras opciones avanzadas	7
3.	HAPROXY	11
3.1.	Ponderación	11
3.2.	Opciones avanzadas	12
4.	ESTADÍSTICAS DE HAPROXY	15
4.1.	Opciones avanzadas	16
5.	GOBETWEEN	19
5.1.	Opciones avanzadas	19
6.	ZEVENET	24
6.1.	Opciones avanzadas	25
7.	POUND	32
7.1.	Opciones avanzadas	32
8.	SOMETER LA GRANJA WEB A UNA CARGA	37
9.	BIBLIOGRAFÍA	41

INSTALACIÓN

Comenzamos instalando la máquina virtual como las anteriores, rellenando los datos relativos a la m3, como se muestra en (1).

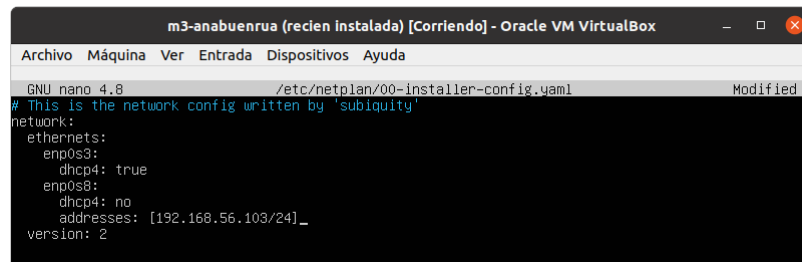
Figura 1: Instalación de m3



Añadimos el adaptador de red solo-anfitrión y configuramos la red editando el fichero `/etc/netplan/00-installer-config.yaml` asignando la ip 192.168.56.103, (2).

Finalmente ejecutamos `sudo netplan apply` para hacer efectivos los cambios.

Figura 2: Configuración de la red de m3, fichero /etc/netplan/00-installer-config.yaml.



```
m3-anabuenrwa (recien instalada) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/netplan/00-installer-config.yaml Modified
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: no
      addresses: [192.168.56.103/24]_
  version: 2
```

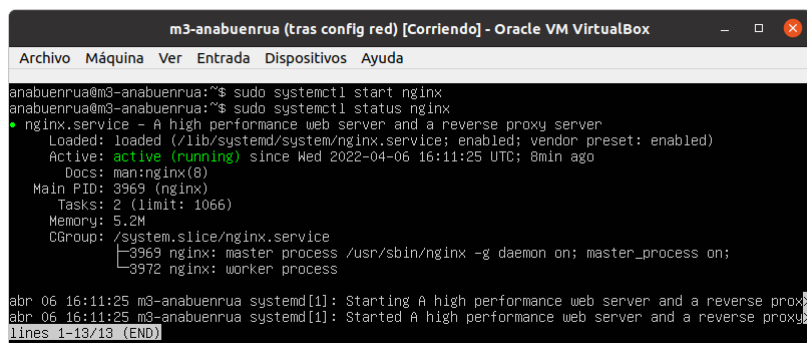
NGINX

Comenzamos instalando nginx, para ello ejecutamos:

```
sudo apt-get update && sudo apt-get dist-upgrade
&& sudo apt-get autoremove
sudo apt-get install nginx
```

Lanzamos nginx con `sudo systemctl start nginx`, y comprobamos que está activo, puede verse en (3).

Figura 3: Lanzamiento y comprobación del estado de nginx.



```
m3-anabuenrúa (tras config red) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
anabuenrúa@m3-anabuenrúa:~$ sudo systemctl start nginx
anabuenrúa@m3-anabuenrúa:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-04-06 16:11:25 UTC; 8min ago
     Docs: man:nginx(8)
   Main PID: 3969 (nginx)
    Tasks: 2 (limit: 1066)
   Memory: 5.2M
   CGroup: /system.slice/nginx.service
           └─3969 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─3972 nginx: worker process

abr 06 16:11:25 m3-anabuenrúa systemd[1]: Starting A high performance web server and a reverse proxy server: nginx.
abr 06 16:11:25 m3-anabuenrúa systemd[1]: Started A high performance web server and a reverse proxy server: nginx.
lines 1-13/13 (END)
```

Ahora deshabilitamos el servidor web para que nginx actúe como balanceador, para lo que vamos a editar el fichero `/etc/nginx/nginx.conf`, dejándolo como en (4):

Y ahora modificamos el archivo `/etc/nginx/conf.d/default.conf` como se ve en (5).

Y reiniciamos el servicio con `sudo systemctl restart nginx` para asegurarnos de que se aplica la configuración.

Comprobamos ahora accediendo a `http://192.168.56.103/swap.html` en el navegador o mediante `curl` (como en (6) que las máquinas se turnan para servir el sitio web.

Figura 4: Deshabilitamos el servidor web modificando el fichero de configuración /etc/nginx/nginx.conf.

```

GNU nano 4.8 /etc/nginx/nginx.conf Modified
# Gzip Settings
##

gzip on;

# gzip_vary on;
# gzip_proxied any;
# gzip_comp_level 6;
# gzip_buffers 16 8k;
# gzip_http_version 1.1;
# gzip_types text/plain text/css application/json application/javascript text/xml application/xml;

##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
#include /etc/nginx/sites-enabled/*;

}

#mail {
#
# See sample authentication script at:
# http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
# auth_http localhost/auth.php;
# pop3_capabilities "TOP" "USER";
# imap_capabilities "IMAP4rev1" "UIDPLUS";
#
# server {
#
#     listen          localhost:110;
#     protocol        pop3;
#     proxy           on;
#
# }

```

Figura 5: Fichero de configuración /etc/nginx/conf.d/default.conf .

```

GNU nano 4.8 /etc/nginx/conf.d/default.conf Modified
upstream balanceador_anabuenruea {
    server 192.168.56.101;
    server 192.168.56.102;
}

server {
    listen 80;
    server_name balanceador_anabuenruea;

    access_log /var/log/nginx/balanceador_anabuenruea.access.log;
    error_log /var/log/nginx/balanceador_anabuenruea.error.log;
    root /var/www;

    location / {
        proxy_pass http://balanceador_anabuenruea;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

2.1 PONDERACIÓN

Ahora vamos a repartir la carga entre las máquinas en función de pesos. Para ello, en /etc/nginx/conf.d/default.conf vamos a usar el parámetro weight.

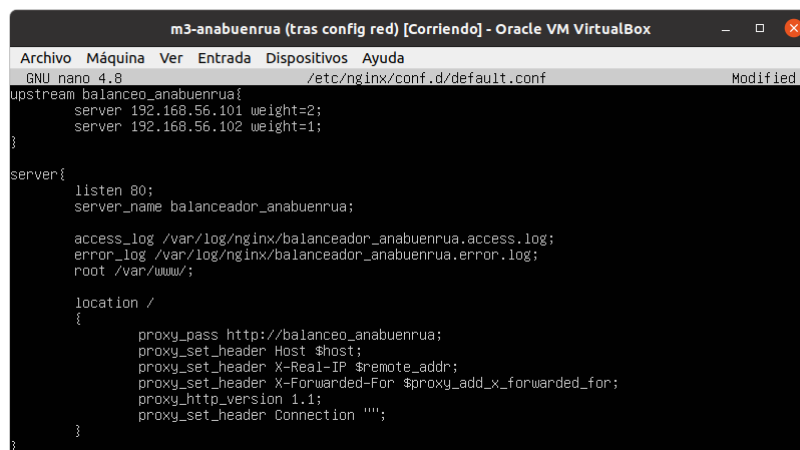
Figura 6: Comprobación del funcionamiento de la configuración de nginx con roundrobin.

```
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 1
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 2
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$
```

Como ejemplo, vamos a suponer que m1 tiene el doble de capacidad que m2.

Editamos el fichero `/etc/nginx/conf.d/default.conf` como en (7).

Figura 7: Fichero de configuración `/etc/nginx/conf.d/default.conf` para ponderación.



```
m3-anabuenrúa (tras config red) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/nginx/conf.d/default.conf Modified
upstream balanceo_anabuenrúa{
    server 192.168.56.101 weight=2;
    server 192.168.56.102 weight=1;
}

server{
    listen 80;
    server_name balanceador_anabuenrúa;

    access_log /var/log/nginx/balanceador_anabuenrúa.access.log;
    error_log /var/log/nginx/balanceador_anabuenrúa.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_anabuenrúa;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Relanzamos nginx con `sudo systemctl restart nginx` y comprobamos que funciona en (8).

2.2 OTRAS OPCIONES AVANZADAS

Ahora vamos a configurar que todas las peticiones que vengan de la misma IP se dirijan a la misma máquina servidora final. Para ello, usamos `ip_hash` en el fichero de configuración, como en (9).

De nuevo, reiniciamos el servicio para hacer efectiva la configuración y comprobamos que funciona en (10).

También podemos activar las conexiones `keepalive` por 3 segundos, ilustrado en (11).

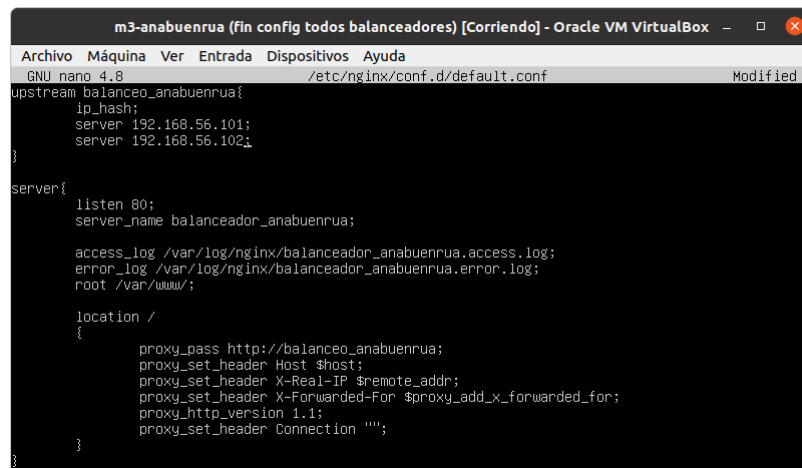
Figura 8: Comprobación del funcionamiento de la configuración de nginx con ponderación.

```

mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 1
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 1
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 2
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$

```

Figura 9: Fichero de configuración /etc/nginx/conf.d/default.conf con ip_hash.



```

m3-anabuenrúa (fin config todos balanceadores) [Corriendo] - Oracle VM VirtualBox
GNU nano 4.8 /etc/nginx/conf.d/default.conf Modified
upstream balanceo_anabuenrúa{
    ip_hash;
    server 192.168.56.101;
    server 192.168.56.102;
}

server{
    listen 80;
    server_name balanceador_anabuenrúa;

    access_log /var/log/nginx/balanceador_anabuenrúa.access.log;
    error_log /var/log/nginx/balanceador_anabuenrúa.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_anabuenrúa;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

Otra opción es usar `down`, que marca un servidor como permanentemente offline, y requiere de ser usado con `ip_hash`, como se ve es (12).

Relanzamos el servicio y comprobamos que funciona en (13).

También podemos usar `backup`, que reserva este servidor y solo le pasa tráfico si alguno de los otros servidores no configurados como `backup` está caído u ocupado, como se muestra en (14).

Y ahora siempre nos atiende `m1`.

Finalmente, para evitar errores al reiniciar la máquina virtual, se va a deshabilitar que se lance `nginx` al encender la máquina. Para ello se ejecuta `sudo systemctl disable nginx`.

Figura 10: Comprobación del funcionamiento de la configuración de nginx con ip_hash.

```
<HTML>
<BODY>
MAQUINA 2
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 2
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 2
Web de ejemplo de anabuenrúa para SWAP
Email: anabuenrúa@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$
```

Figura 11: Fichero de configuración /etc/nginx/conf.d/default.conf con keepalive.

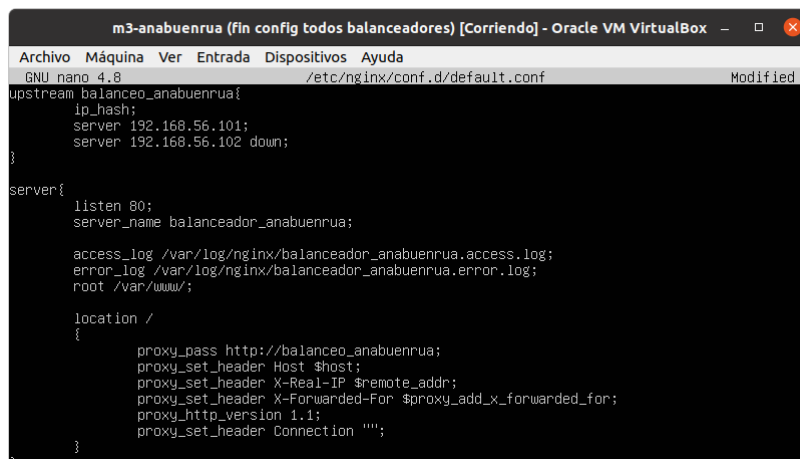
```
m3-anabuenrúa (fin config todos balanceadores) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 4.8 /etc/nginx/conf.d/default.conf Modified
upstream balanceo_anabuenrúa{
    server 192.168.56.101;
    server 192.168.56.102;
    keepalive 3;
}

server{
    listen 80;
    server_name balanceador_anabuenrúa;

    access_log /var/log/nginx/balanceador_anabuenrúa.access.log;
    error_log /var/log/nginx/balanceador_anabuenrúa.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_anabuenrúa;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Figura 12: Fichero de configuración /etc/nginx/conf.d/default.conf con down.



```

m3-anabuenruea (fin config todos balanceadores) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/nginx/conf.d/default.conf Modified
upstream balanceo_anabuenruea{
    ip_hash;
    server 192.168.56.101;
    server 192.168.56.102 down;
}

server{
    listen 80;
    server_name balanceador_anabuenruea;

    access_log /var/log/nginx/balanceador_anabuenruea.access.log;
    error_log /var/log/nginx/balanceador_anabuenruea.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_anabuenruea;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

Figura 13: Comprobación del funcionamiento de la configuración de nginx con down.

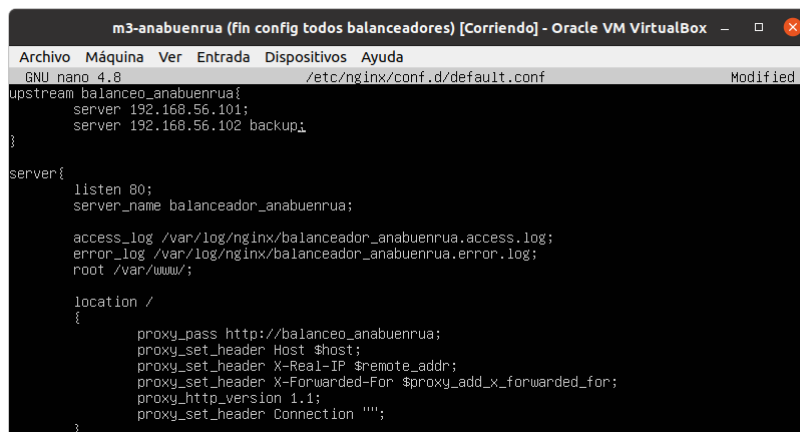


```

<HTML>
<BODY>
MAQUINA 1
Web de ejemplo de anabuenruea para SWAP
Email: anabuenruea@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 1
Web de ejemplo de anabuenruea para SWAP
Email: anabuenruea@correo.ugr.es
</BODY>
</HTML>
mapachana@mapachana-Aspire-E5-574G:~$ curl http://192.168.56.103/swap.html
<HTML>
<BODY>
MAQUINA 1
Web de ejemplo de anabuenruea para SWAP
Email: anabuenruea@correo.ugr.es
</BODY>
</HTML>

```

Figura 14: Fichero de configuración /etc/nginx/conf.d/default.conf con backup.



```

m3-anabuenruea (fin config todos balanceadores) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/nginx/conf.d/default.conf Modified
upstream balanceo_anabuenruea{
    server 192.168.56.101;
    server 192.168.56.102 backup;
}

server{
    listen 80;
    server_name balanceador_anabuenruea;

    access_log /var/log/nginx/balanceador_anabuenruea.access.log;
    error_log /var/log/nginx/balanceador_anabuenruea.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://balanceo_anabuenruea;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}

```

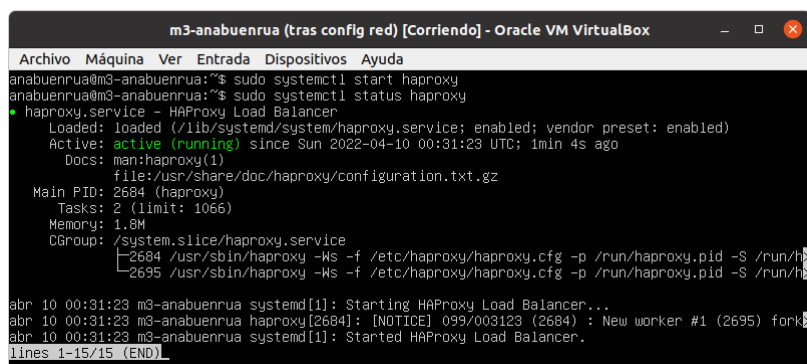
HAPROXY

Comenzamos parando nginx, de forma que no interfiera con HAProxy, ya que usan el mismo puerto. Para pararlo, ejecutamos `sudo systemctl stop nginx`.

Ahora instalamos HAProxy con `sudo apt-get install haproxy`.

Lanzamos HAProxy con `sudo systemctl start haproxy` y comprobamos su estado, como se muestra en (15).

Figura 15: Lanzamiento y comprobación del estado de haproxy.



```
m3-anabuenruea (tras config red) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
anabuenruea@m3-anabuenruea:~$ sudo systemctl start haproxy
anabuenruea@m3-anabuenruea:~$ sudo systemctl status haproxy
● haproxy.service - HAProxy Load Balancer
   Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-04-10 00:31:23 UTC; 1min 4s ago
     Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 2684 (haproxy)
      Tasks: 2 (limit: 1066)
     Memory: 1.8M
    CGroup: /system.slice/haproxy.service
            └─2684 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/hb
              2695 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/hb

abr 10 00:31:23 m3-anabuenruea systemd[1]: Starting HAProxy Load Balancer...
abr 10 00:31:23 m3-anabuenruea haproxy[2684]: [NOTICE] 099/003123 (2684) : New worker #1 (2695) fork
abr 10 00:31:23 m3-anabuenruea systemd[1]: Started HAProxy Load Balancer.
lines 1-15/15 (END)
```

Editamos ahora el archivo de configuración `/etc/haproxy/haproxy.cfg` como en (16).

Y relanzamos el servicio con `sudo systemctl restart haproxy`.

Comprobamos que funciona, va sirviendo por turnos cada máquina, al igual que lo hacía nginx.

3.1 PONDERACIÓN

Para configurar la ponderación, de nuevo editamos el fichero `/etc/haproxy/haproxy.cfg`, usando la opción `weight`. De nuevo se ha supuesto a m1 con el doble de capacidad de m2, como se ve en (17).

Y relanzamos el servicio y comprobamos que funciona, de nuevo m1 sirve 2 de cada 3 peticiones que realizamos.

3.2 OPCIONES AVANZADAS

Por analogía y para que sirva de comparativa con cómo se realizan algunas configuraciones en nginx, se van a usar configuraciones ya usadas antes.

Por ejemplo, para usar ip hash, editando de nuevo el archivo de configuración como en (18).

Y si relanzamos y comprobamos que funciona, pues siempre nos atiende m1.

Finalmente, también podemos poner de nuevo m2 en backup, como se muestra en (19).

Y relanzando comprobamos que funciona, ya que siempre nos atiende m1.

Finalmente, para evitar errores al reiniciar la máquina virtual, se va a deshabilitar que se lance HAProxy al encender la máquina. Para ello se ejecuta `sudo systemctl disable haproxy`.

Figura 16: Fichero de configuración /etc/haproxy/haproxy.cfg .

```

GNU nano 4.8 /etc/haproxy/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermedi
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-EC
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_P
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

    # Estadísticas
    stats socket /var/lib/haproxy/stats

listen stats
    bind *:9999
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth anabuenrúa:anabuenrúa
    stats refresh 30s
    stats admin if TRUE

defaults

```

```

    log      global
    mode     http
    option   httplog
    option   dontlognull
    timeout  connect 5000
    timeout  client  50000
    timeout  server  50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend http-in
    bind *:80
    default_backend balanceo_anabuenrúa

backend balanceo_anabuenrúa
    server m1 192.168.56.101:80 maxconn 32
    server m2 192.168.56.102:80 maxconn 32

```

Figura 17: Fichero de configuración /etc/haproxy/haproxy.cfg para ponderación.

```

frontend http-in
    bind *:80
    default_backend balanceo_anabuenruea

backend balanceo_anabuenruea
    server m1 192.168.56.101:80 maxconn 32 weight 2
    server m2 192.168.56.102:80 maxconn 32 weight 1

```

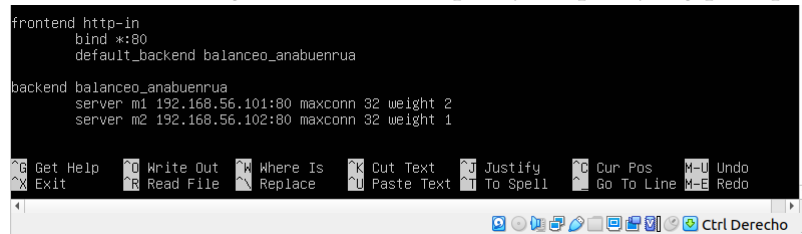


Figura 18: Fichero de configuración /etc/haproxy/haproxy.cfg para ip_hash.

```

frontend http-in
    bind *:80
    default_backend balanceo_anabuenruea

backend balanceo_anabuenruea
    balance source
    hash-type consistent
    server m1 192.168.56.101:80 maxconn 32
    server m2 192.168.56.102:80 maxconn 32

```

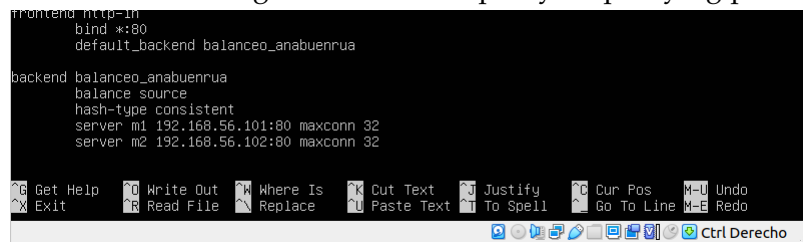


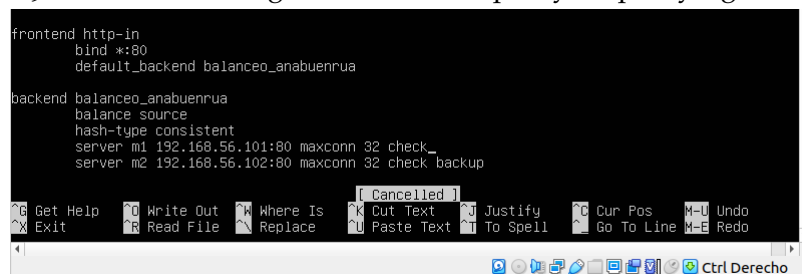
Figura 19: Fichero de configuración /etc/haproxy/haproxy.cfg con backup.

```

frontend http-in
    bind *:80
    default_backend balanceo_anabuenruea

backend balanceo_anabuenruea
    balance source
    hash-type consistent
    server m1 192.168.56.101:80 maxconn 32 check_
    server m2 192.168.56.102:80 maxconn 32 check backup

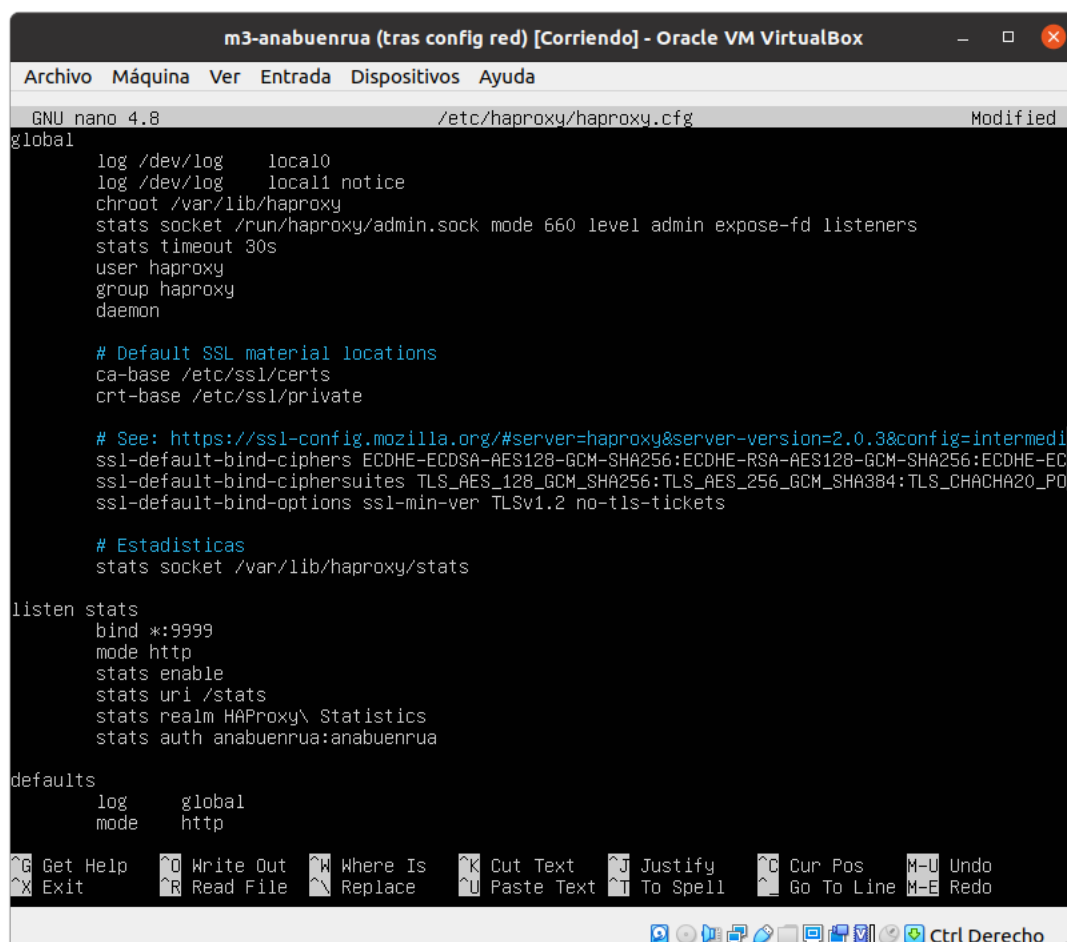
```



ESTADÍSTICAS DE HAPROXY

Para esta sección dejamos round-robin, como en la configuración inicial, y añadimos configuración necesaria para habilitar las estadísticas en `/etc/haproxy/haproxy.cfg`, como se ve en (20).

Figura 20: Fichero de configuración `/etc/haproxy/haproxy.cfg` para habilitar estadísticas.



```

m3-anabuenrúa (tras config red) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/haproxy/haproxy.cfg Modified
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermedi
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-EC
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_PQ
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

    # Estadísticas
    stats socket /var/lib/haproxy/stats

listen stats
    bind *:9999
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth anabuenrúa:anabuenrúa

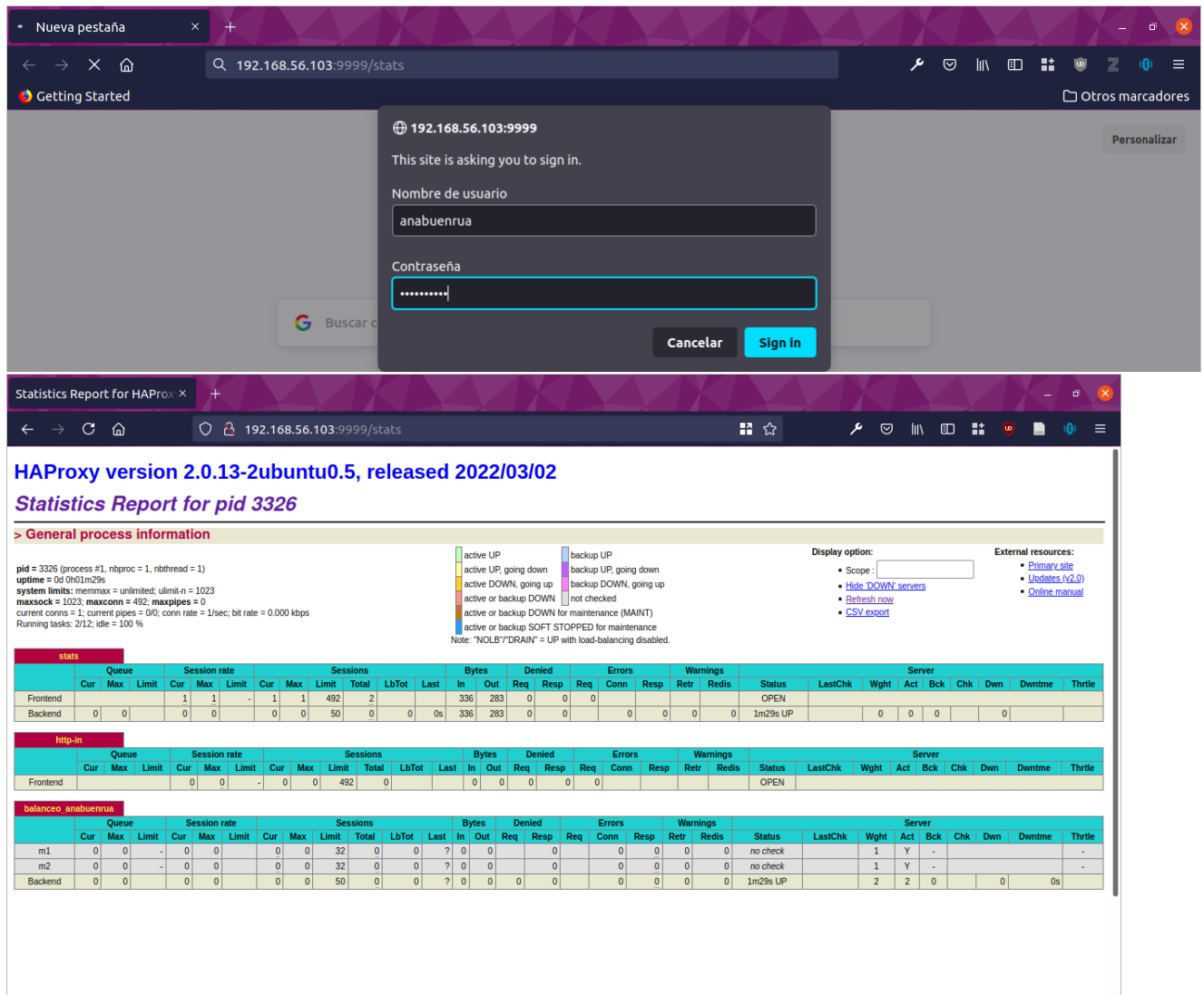
defaults
    log global
    mode http

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^N Replace   ^U Paste Text ^T To Spell   ^_ Go To Line  M-E Redo
Ctrl Derecho

```

Relanzamos haproxy y accedemos a las estadísticas, para lo que introducimos el usuario y contraseña antes de acceder, como puede verse en (21).

Figura 21: Accediendo a las estadísticas de HAProxy a través del navegador



192.168.56.103:9999

This site is asking you to sign in.

Nombre de usuario
anabuenruea

Contraseña
.....

Cancelar Sign In

Statistics Report for HAProxy

HAProxy version 2.0.13-2ubuntu0.5, released 2022/03/02

Statistics Report for pid 3326

> General process information

pid = 3326 (process #1, nbproc = 1, nbthread = 1)
uptime = 0d 0h 0m 29s
system limits: memmax = unlimited; ulimit-n = 1023
maxsock = 1023; maxconn = 492; maxpipes = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec; bit rate = 0.000 kbps
Running tasks: 2/12; idle = 100 %

active UP
active UP, going down
active DOWN, going up
active or backup DOWN
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance

backup UP
backup UP, going down
backup DOWN, going up
not checked
active or backup DOWN for maintenance (MAINT)
active or backup SOFT STOPPED for maintenance

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:
Scope:
[Hide DOWN servers](#)
[Refresh now](#)
[CSV export](#)

External resources:
[Primary site](#)
[Updates \(v2.0\)](#)
[Online manual](#)

stats		Queue		Session rate		Sessions		Bytes		Denied		Errors		Warnings		Server		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Status	
Frontend	0	0	0	0	0	0	0	0	0	492	0	0s	336	283	0	0	0	OPEN
Backend	0	0	0	0	0	0	0	0	0	50	0	0s	336	283	0	0	0	1m29s UP

http-in		Queue		Session rate		Sessions		Bytes		Denied		Errors		Warnings		Server		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Status	
Frontend	0	0	0	0	0	0	0	0	0	492	0	0s	0	0	0	0	0	OPEN

balanceo.anabuenruea		Queue		Session rate		Sessions		Bytes		Denied		Errors		Warnings		Server		
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Status	
m1	0	0	0	0	0	0	0	0	0	32	0	0s	0	0	0	0	0	no check
m2	0	0	0	0	0	0	0	0	0	32	0	0s	0	0	0	0	0	no check
Backend	0	0	0	0	0	0	0	0	0	50	0	0s	0	0	0	0	0	1m29s UP

4.1 OPCIONES AVANZADAS

Como opciones avanzadas se va a añadir el tiempo de refresco de la página y la posibilidad de marcar los servidores como down o maintenance.

Por ejemplo, vamos a usar un tiempo de refresco de 30 segundos.

Para ello, editamos de nuevo el fichero de configuración añadiendo las líneas:

```
stats refresh 30s
stats admin if TRUE
```

Luego el fichero quedaría como (22).

Figura 22: Fichero de configuración `/etc/haproxy/haproxy.cfg` con opciones avanzadas de las estadísticas.

```

m3-anabuenrúa (tras config red) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/haproxy/haproxy.cfg
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&server-version=2.0.3&config=intermedi
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-EC
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_PO
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets

    # Estadísticas
    stats socket /var/lib/haproxy/stats

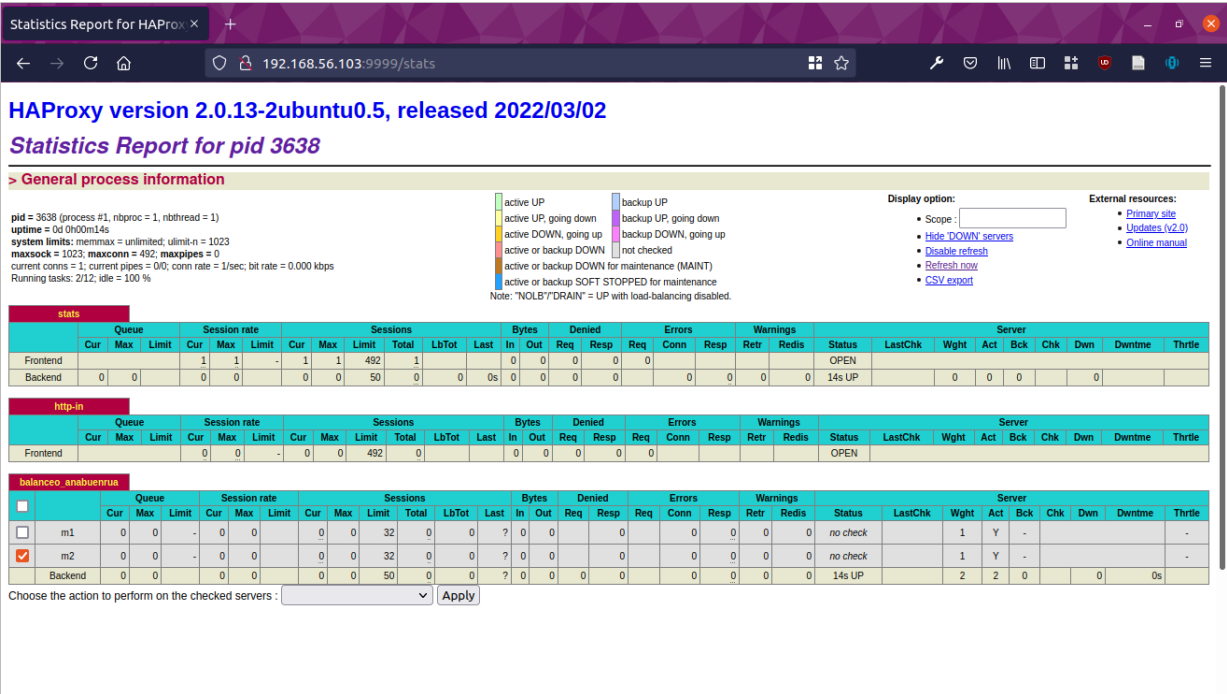
listen stats
    bind *:9999
    mode http
    stats enable
    stats uri /stats
    stats realm HAProxy\ Statistics
    stats auth anabuenrúa:anabuenrúa
    stats refresh 30s
    stats admin if TRUE_

defaults
    ^G Get Help    ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo
    ^X Exit        ^R Read File    ^N Replace      ^U Paste Text   ^T To Spell     ^G Go To Line   M-E Redo
  
```

Y relanzando de nuevo el servicio y accediendo a las estadísticas observamos los cambios en (23).

Si además se quisiera configurar el timeout, se usaría `stats timeout <n>s`.

Figura 23: Estadísticas de haproxy con opciones avanzadas.



GOBETWEEN

Comenzamos asegurandonos de que tanto nginx como haproxy no se estén ejecutando, y tras esto instalamos gobetween mediante snap ejecutando:

```
sudo snap install gobetween --edge
```

Ahora, vamos a configurar gobetween, para ello nos basamos en el archivo de configuración de `/snap/gobetween/current/config/gobetween.toml`, y creamos el fichero de configuración `/snap/gobetween/gobetween_config.toml` quedando como (24).

Hemos dejado los parámetros por defecto que venían (defaults) y las métricas activadas, ya que esta configuración ya venía realizada en la configuración de ejemplo. Las opciones de healthcheck se explicarán en el apartado de opciones avanzadas.

Y ahora lanzamos gobetween con la configuración establecida indicando el fichero. Como vemos en (25), no tenemos permiso de escucha para el puerto 80, y nos lanza un error.

Por tanto, ejecutamos gobetween con sudo, de forma que no da problemas, como puede verse en (26).

Y comprobamos que funciona, pues las máquinas m1 y m2 responden turnándose.

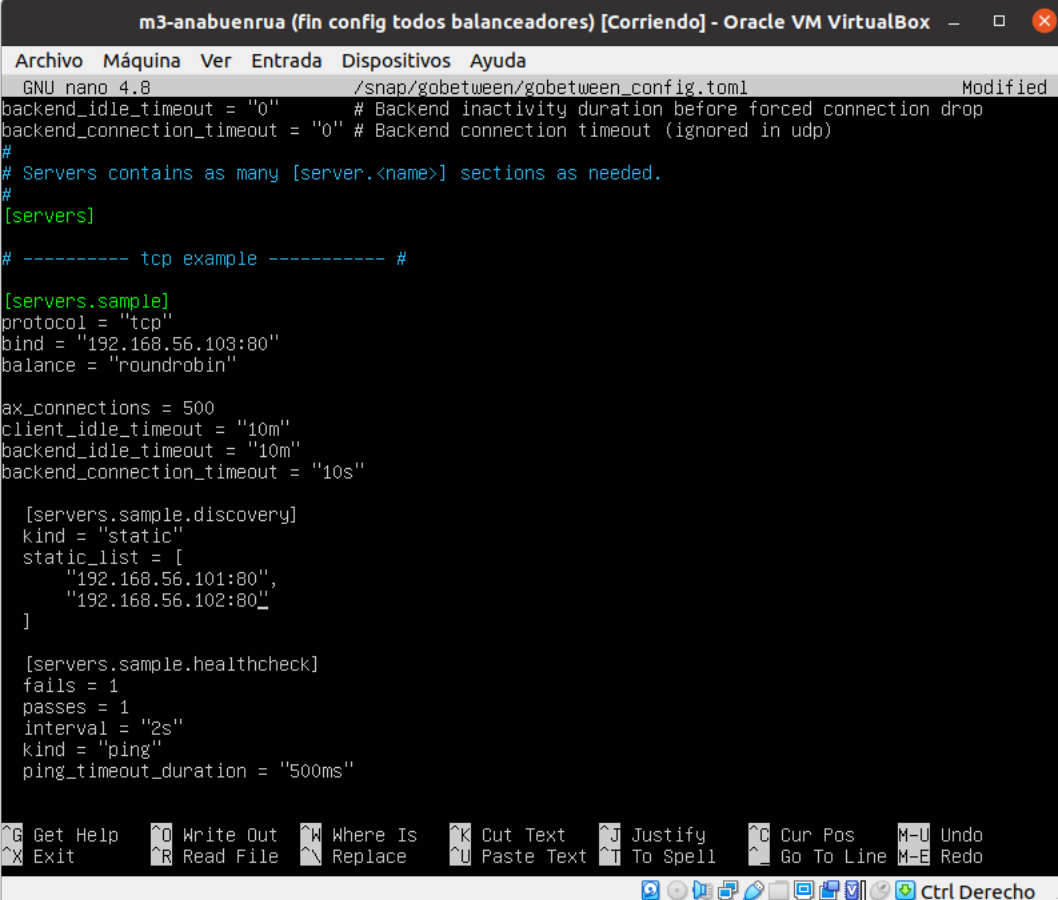
5.1 OPCIONES AVANZADAS

Además de round-robin, se puede usar ip_hash como se muestra en (27), también menor número de conexiones (28) y por ponderación (29).

Adicionalmente, se ha configurado el último bloque, healthcheck, que comprueba el estado de los servidores mediante ping (pues se ha configurado con este método) cada 2 segundos, y debe llegar una respuesta antes de 500ms. Si esta respuesta no llegara, como se ha perdido una respuesta (parámetro fails está a 1), el servidor se considera caído (down), y se marcará de nuevo como operativo cuando llegue una respuesta de un ping (parámetro passes está a 1).

En este caso, como gobetween no es un servicio, no hace falta desactivar que se lance automáticamente al encender la máquina virtual.

Figura 24: Fichero de configuración /snap/gobetween/gobetween_config.toml



```

GNU nano 4.8 /snap/gobetween/gobetween_config.toml Modified
backend_idle_timeout = "0" # Backend inactivity duration before forced connection drop
backend_connection_timeout = "0" # Backend connection timeout (ignored in udp)
#
# Servers contains as many [server.<name>] sections as needed.
#
[servers]
# ----- tcp example ----- #

[servers.sample]
protocol = "tcp"
bind = "192.168.56.103:80"
balance = "roundrobin"

max_connections = 500
client_idle_timeout = "10m"
backend_idle_timeout = "10m"
backend_connection_timeout = "10s"

[servers.sample.discovery]
kind = "static"
static_list = [
  "192.168.56.101:80",
  "192.168.56.102:80"
]

[servers.sample.healthcheck]
fails = 1
passes = 1
interval = "2s"
kind = "ping"
ping_timeout_duration = "500ms"

Get Help Write Out Where Is Cut Text Justify Cur Pos M-U Undo
Exit Read File Replace Paste Text To Spell Go To Line M-E Redo
Ctrl Derecho

```

Figura 25: Error al lanzar gobetween por no tener permisos de escucha del puerto 80.



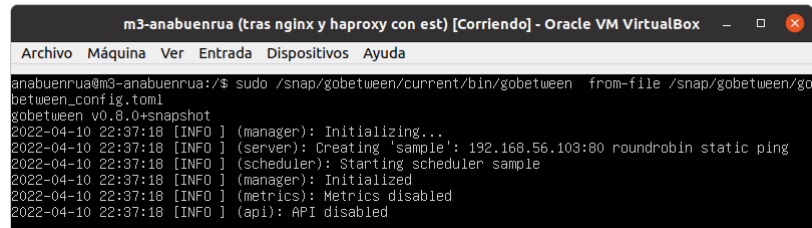
```

m3-anabuenrúa@anabuenrúa:/$ /snap/gobetween/current/bin/gobetween from-file /snap/gobetween/gobetween_config.toml
gobetween v0.8.0+snapshot
2022-04-10 22:36:05 [INFO ] (manager): Initializing...
2022-04-10 22:36:05 [INFO ] (server): Creating 'sample': 192.168.56.103:80 roundrobin static ping
2022-04-10 22:36:05 [INFO ] (scheduler): Starting scheduler sample
2022-04-10 22:36:05 [ERROR] (server.Listen): Error starting tcp server: listen tcp 192.168.56.103:80: bind: permission denied
2022-04-10 22:36:05 [INFO ] (server.Listen): Stopping sample
2022-04-10 22:36:05 [INFO ] (scheduler): Stopping scheduler sample
2022-04-10 22:36:05 [INFO ] (discovery): Stopping discovery {static keepalst 0 0 0xc000260e00 <nil> <nil> <nil> <nil> <nil> <nil>}
panic: runtime error: invalid memory address or nil pointer dereference
[signal SIGSEGV: segmentation violation code=0x1 addr=0x0 pc=0xd216cc]

goroutine 18 [running]:
github.com/yyjar/gobetween/metrics.RemoveServer(0xc0001e3464, 0x6, 0xc000278600)
    /build/gobetween/parts/gobetween/build/src/metrics/metrics.go:199 +0x7c
github.com/yyjar/gobetween/server/scheduler.(*Scheduler).Start.func1(0xc0001f4d30, 0xc000095900, 0xc0001f9ce0)
    /build/gobetween/parts/gobetween/build/src/server/scheduler/scheduler.go:159 +0x464
created by github.com/yyjar/gobetween/server/scheduler.(*Scheduler).Start
    /build/gobetween/parts/gobetween/build/src/server/scheduler/scheduler.go:113 +0x21f
anabuenrúa@m3-anabuenrúa:/$

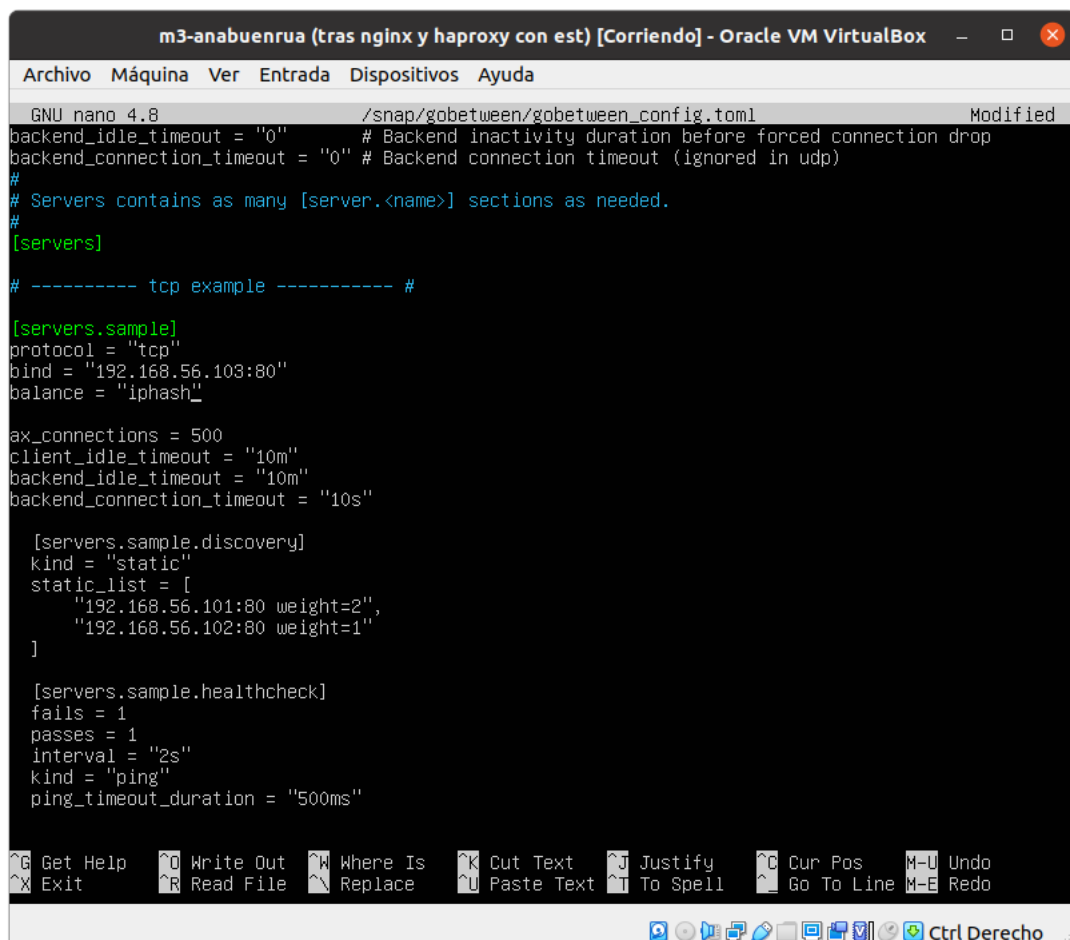
```

Figura 26: Lanzamiento de gobetween con sudo.



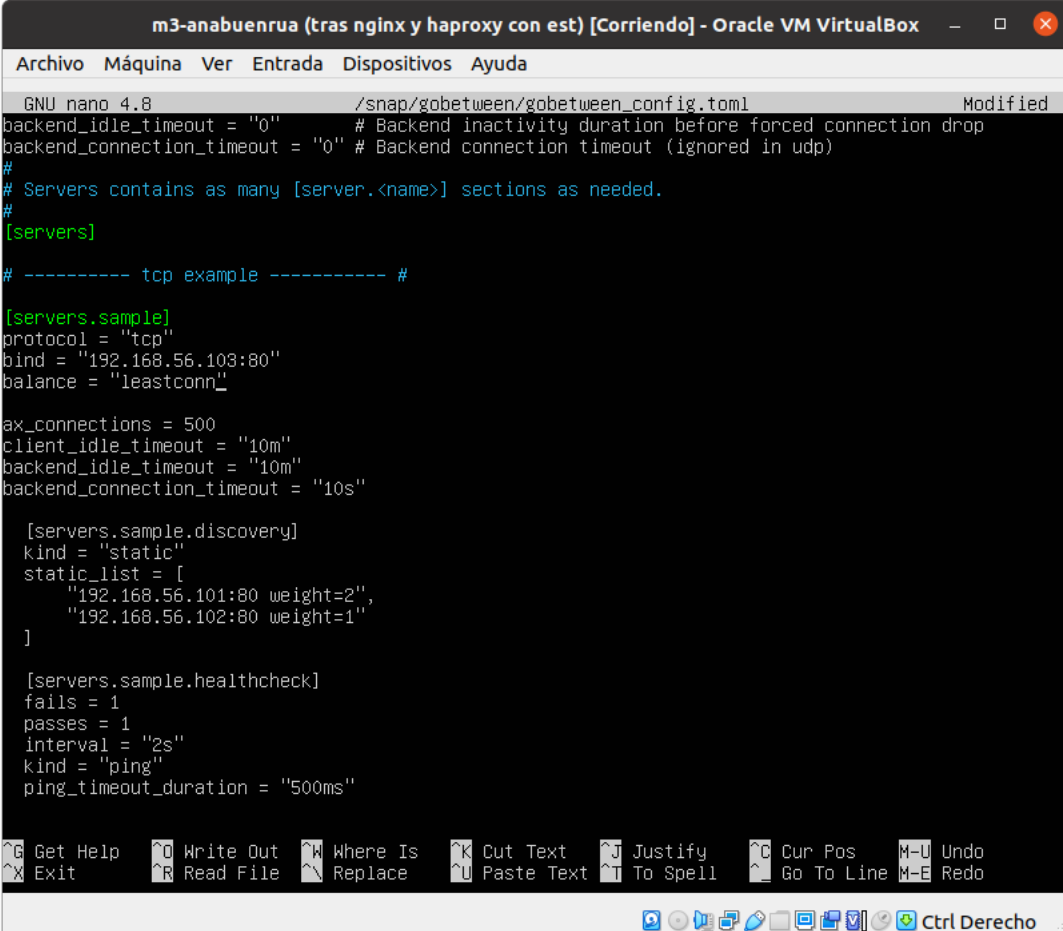
```
m3-anabuenruea (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
anabuenruea@m3-anabuenruea:/$ sudo /snap/gobetween/current/bin/gobetween from-file /snap/gobetween/go
between_config.toml
gobetween v0.8.0+snapshot
2022-04-10 22:37:18 [INFO ] (manager): Initializing...
2022-04-10 22:37:18 [INFO ] (server): Creating 'sample': 192.168.56.103:80 roundrobin static ping
2022-04-10 22:37:18 [INFO ] (scheduler): Starting scheduler sample
2022-04-10 22:37:18 [INFO ] (manager): Initialized
2022-04-10 22:37:18 [INFO ] (metrics): Metrics disabled
2022-04-10 22:37:18 [INFO ] (api): API disabled
```

Figura 27: Fichero de configuración /snap/gobetween/gobetween_config.toml con ip_hash



```
m3-anabuenruea (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 4.8 /snap/gobetween/gobetween_config.toml Modified
backend_idle_timeout = "0" # Backend inactivity duration before forced connection drop
backend_connection_timeout = "0" # Backend connection timeout (ignored in udp)
#
# Servers contains as many [server.<name>] sections as needed.
#
[servers]
# ----- tcp example ----- #
[servers.sample]
protocol = "tcp"
bind = "192.168.56.103:80"
balance = "iphash"
max_connections = 500
client_idle_timeout = "10m"
backend_idle_timeout = "10m"
backend_connection_timeout = "10s"
[servers.sample.discovery]
kind = "static"
static_list = [
  "192.168.56.101:80 weight=2",
  "192.168.56.102:80 weight=1"
]
[servers.sample.healthcheck]
fails = 1
passes = 1
interval = "2s"
kind = "ping"
ping_timeout_duration = "500ms"
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^N Replace ^U Paste Text ^T To Spell ^G Go To Line M-E Redo
Ctrl Derecho
```

Figura 28: Fichero de configuración /snap/gobetween/gobetween_config.toml con mínimo número de conexiones



```

GNU nano 4.8 /snap/gobetween/gobetween_config.toml Modified
backend_idle_timeout = "0" # Backend inactivity duration before forced connection drop
backend_connection_timeout = "0" # Backend connection timeout (ignored in udp)
#
# Servers contains as many [server.<name>] sections as needed.
#
[servers]
# ----- tcp example ----- #

[servers.sample]
protocol = "tcp"
bind = "192.168.56.103:80"
balance = "leastconn"

max_connections = 500
client_idle_timeout = "10m"
backend_idle_timeout = "10m"
backend_connection_timeout = "10s"

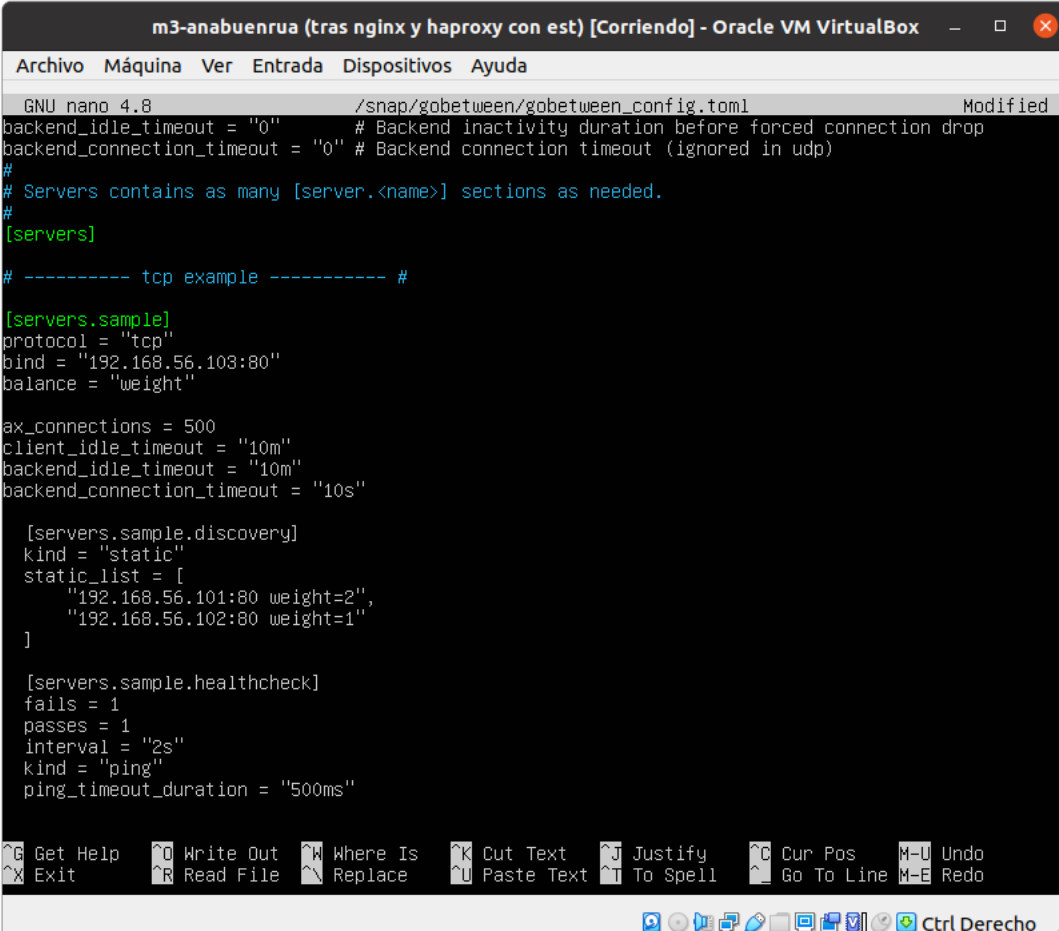
[servers.sample.discovery]
kind = "static"
static_list = [
  "192.168.56.101:80 weight=2",
  "192.168.56.102:80 weight=1"
]

[servers.sample.healthcheck]
fails = 1
passes = 1
interval = "2s"
kind = "ping"
ping_timeout_duration = "500ms"

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^G Go To Line M-E Redo

```

Figura 29: Fichero de configuración /snap/gobetween/gobetween_config.toml con ponderación



```
m3-anabuenrua (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /snap/gobetween/gobetween_config.toml Modified
backend_idle_timeout = "0" # Backend inactivity duration before forced connection drop
backend_connection_timeout = "0" # Backend connection timeout (ignored in udp)
#
# Servers contains as many [server.<name>] sections as needed.
#
[servers]
# ----- tcp example ----- #
[servers.sample]
protocol = "tcp"
bind = "192.168.56.103:80"
balance = "weight"
max_connections = 500
client_idle_timeout = "10m"
backend_idle_timeout = "10m"
backend_connection_timeout = "10s"
[servers.sample.discovery]
kind = "static"
static_list = [
    "192.168.56.101:80 weight=2",
    "192.168.56.102:80 weight=1"
]
[servers.sample.healthcheck]
fails = 1
passes = 1
interval = "2s"
kind = "ping"
ping_timeout_duration = "500ms"
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos   M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line  M-E Redo
Ctrl Derecho
```

ZEVENET

Para configurar Zevenet, vamos a instalar una iso, que descargamos de

<https://es.zevenet.com/productos/comunidad/>.

Comenzamos creando una nueva máquina virtual, de nuevo con 1GB de RAM, 10GB de disco duro dinámico y añadimos el adaptador de red solo-anfitrión antes de iniciar la máquina, de modo que detecte tanto el adaptador NAT como el solo-anfitrión automáticamente.

Al instalarlo, lo hacemos en español de España seleccionando el teclado español, luego seleccionamos el segundo adaptador de red (30) e introducimos la IP de la máquina (31), en este caso se ha usado 192.168.56.104.

Dejamos la netmask y gateway por defecto, al igual que el nameserver. Como hostname introduzco mi usuario de la ugr y como contraseña establezco Swap1234.

Seleccionamos la instalación guiada usando todo el disco, dejamos las particiones por defecto y tras instalar el grub completamos la instalación.

Finalmente, iniciamos la máquina, como se ve en (32).

Accedemos ahora a la dirección 192.168.56.104 por el puerto 444, donde nos identificamos con el usuario y contraseña establecida, como se ve en (33) y llegamos la dashboard (34).

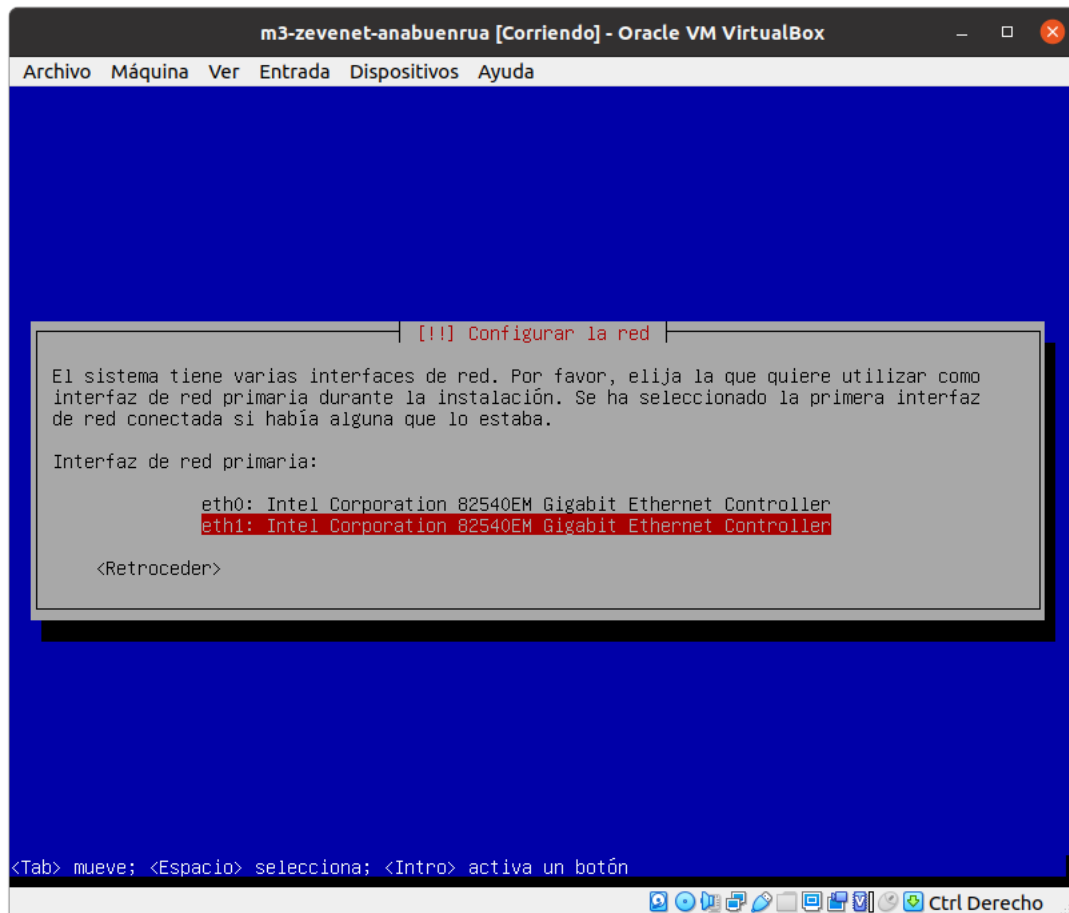
Ahora, en Network->NIC comprobamos y configuramos las redes para que quede así como en (35).

Y en LSLB->Farms creamos una nueva granja de red como se ve en (36).

Ahora, dando a editar podemos ver las opciones avanzadas, puede verse en (37).

Y podemos crear un servicio swap con m1 y m2 como backends como se muestra en (38).

Figura 30: Selección del adaptador de red durante la instalación de Zevenet.



6.1 OPCIONES AVANZADAS

Como opciones avanzadas podemos configurar el timeout de los backend de forma independiente, como ya hemos visto, así como asignar pesos, como se ve en (39).

Al hacer los cambios le damos al botón de restart de la granja y comprobamos que ya funciona.

También se puede configurar la persistencia por sesiones durante un tiempo máximo y comprobaciones del estado de los backend, como se muestra en (40).

Figura 31: Introducimos la dirección IP durante la instalación de Zevenet

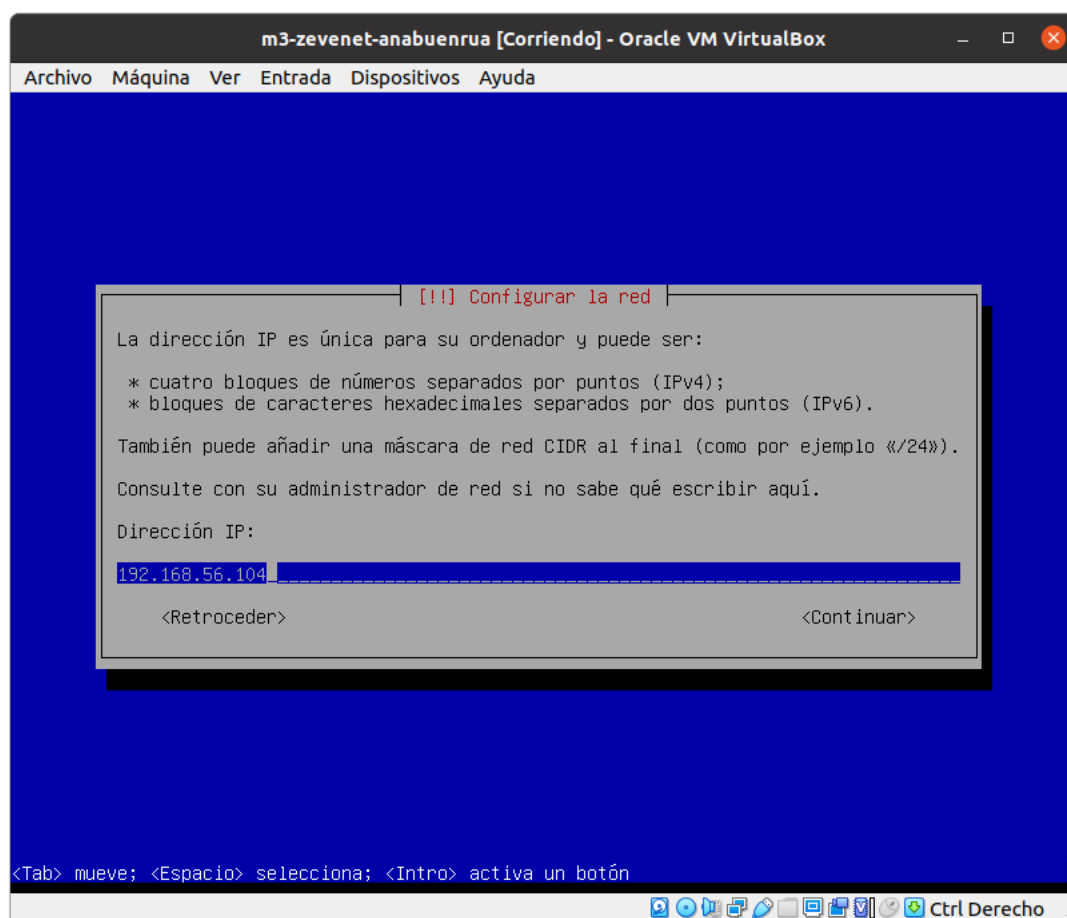


Figura 32: Zevenet tras la instalación.

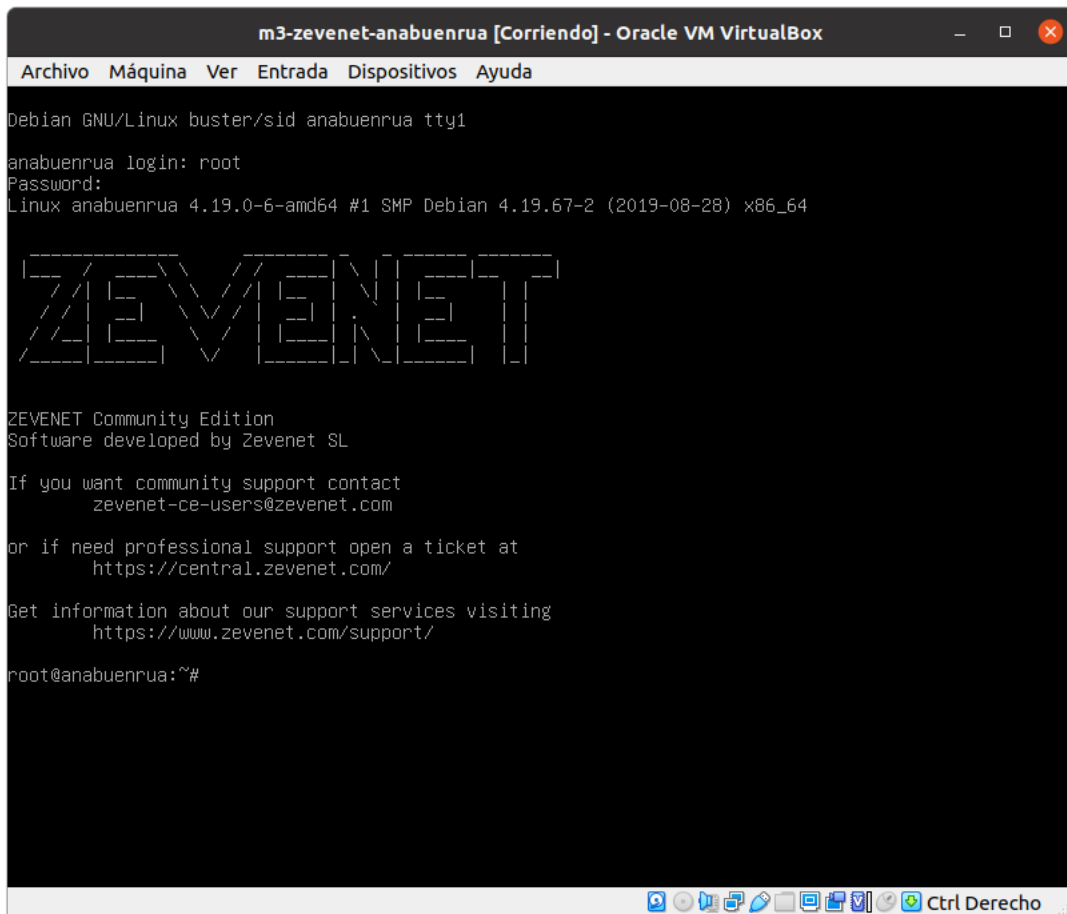


Figura 33: Autenticación para acceder al dashboard de Zevenet

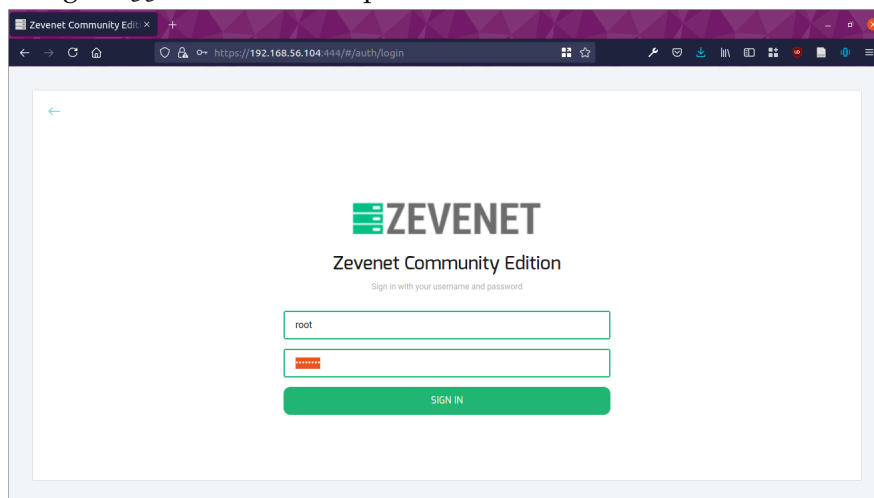


Figura 34: Dashboard de Zevenet

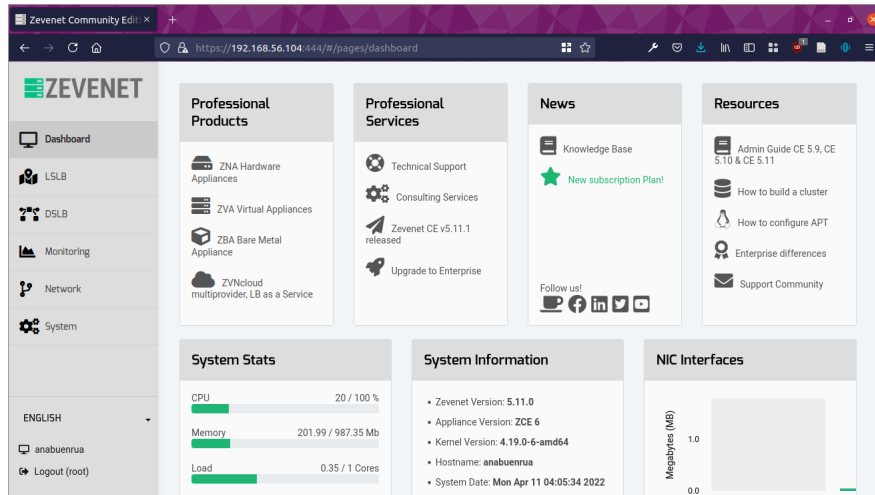


Figura 35: Configuración de las redes de Zevenet

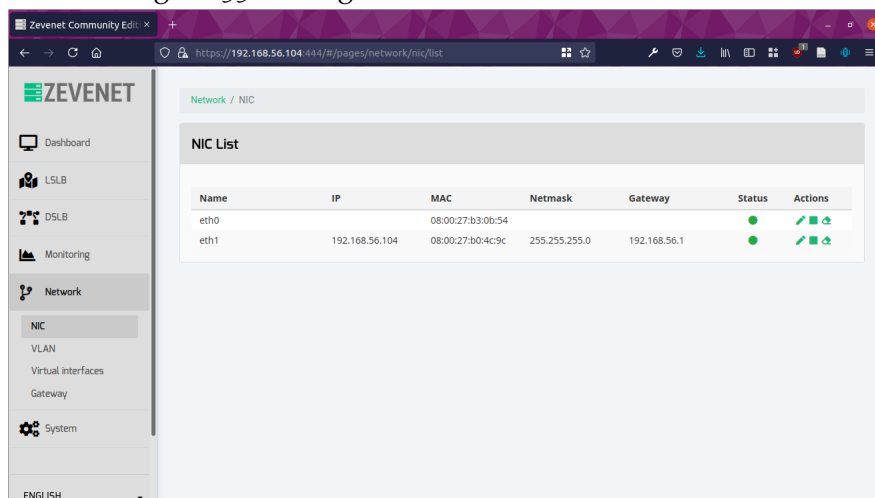


Figura 36: Creación de una granja en Zevenet

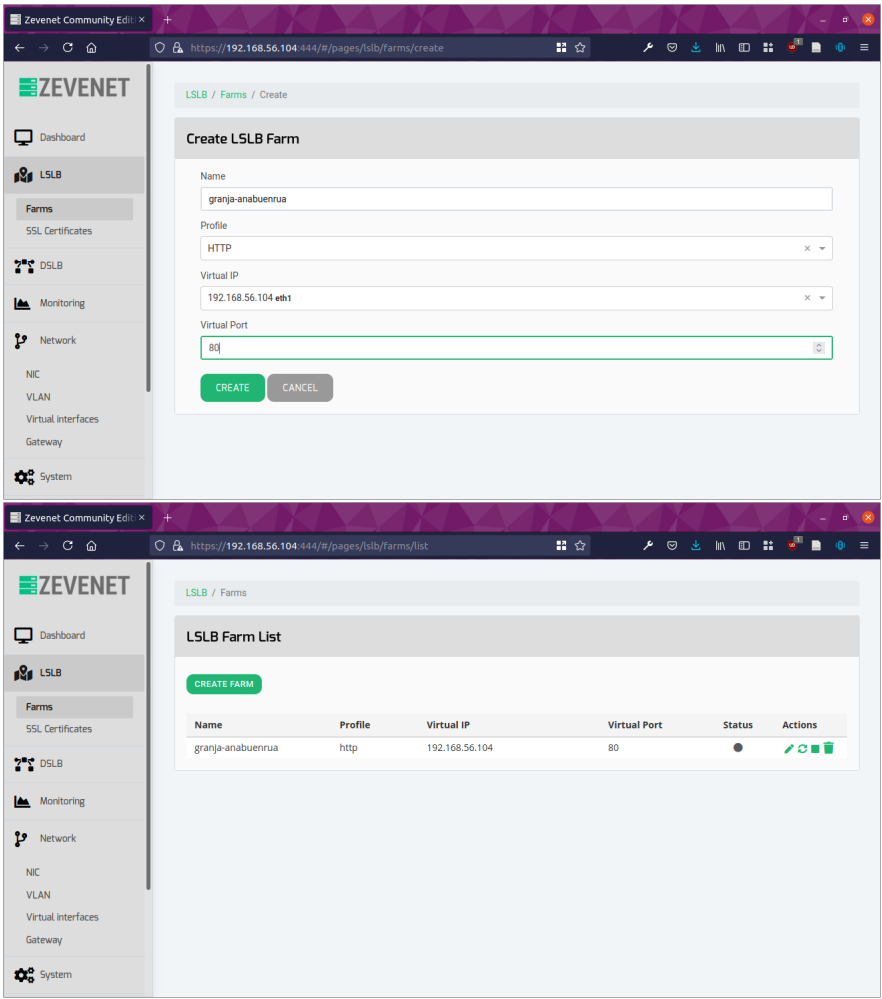


Figura 37: Opciones avanzadas de una granja en Zevenet

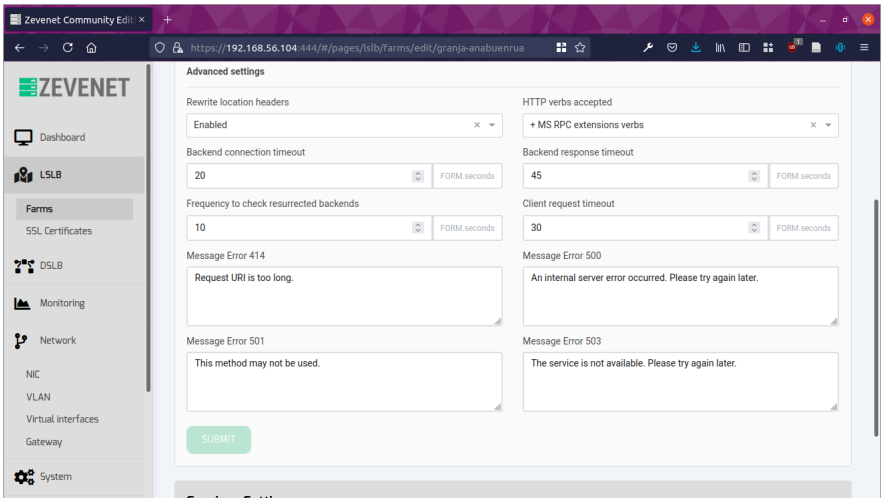


Figura 38: Creación del servicio swap en la granja

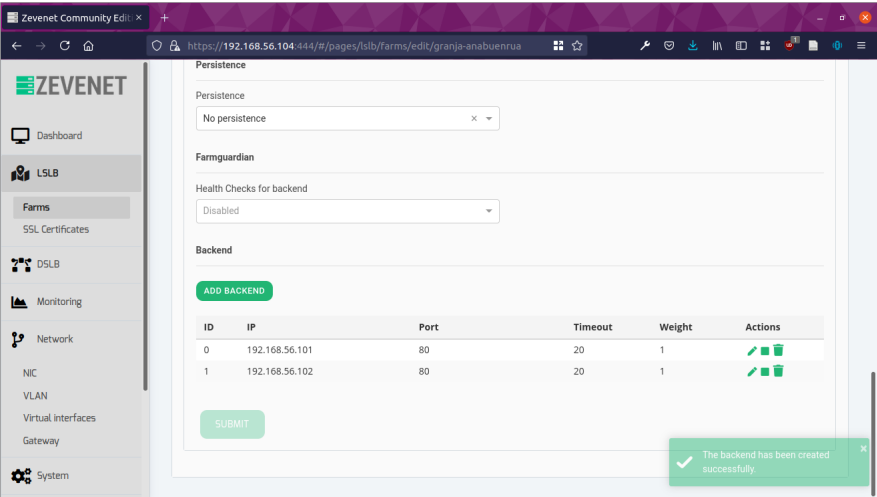


Figura 39: Configuración del timeout y asignación de pesos a m1 y m2 en Zevenet.

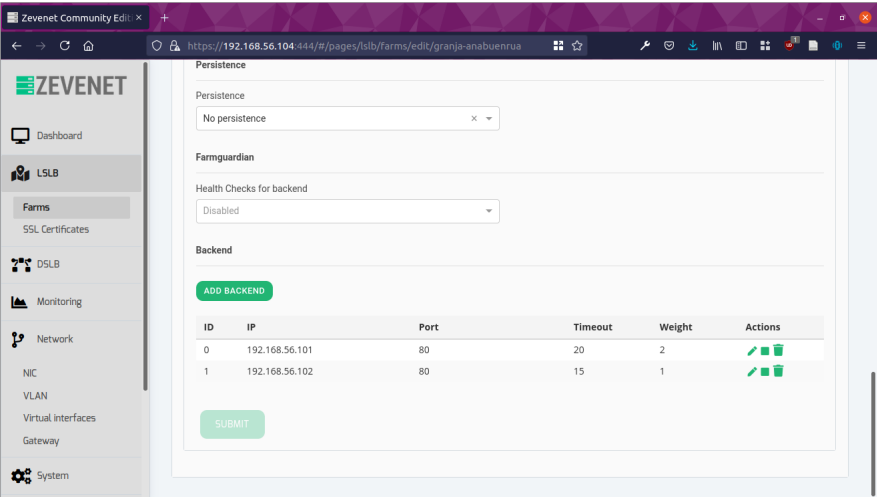
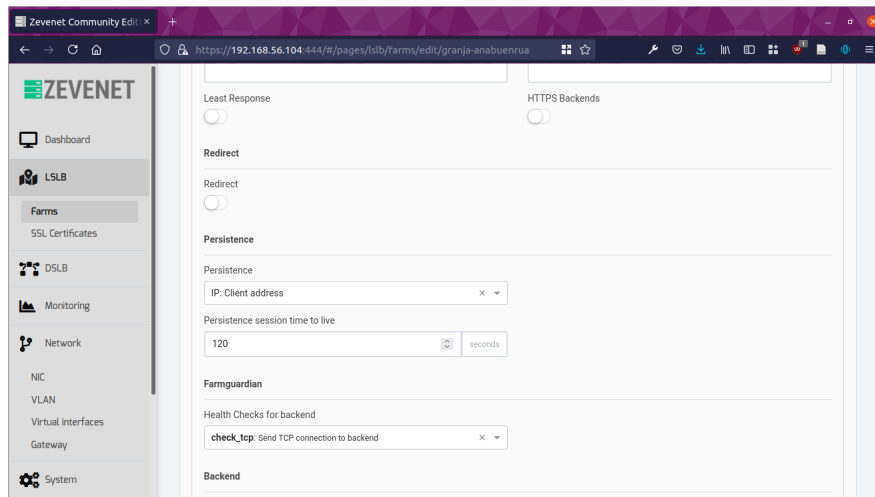


Figura 40: Configuración persistencia de sesiones y comprobaciones del estado de los backend en Zevenet.



POUND

Tras asegurarnos de que ninguno de los otros softwares usados está en marcha, descargamos pound, para ello vamos a ejecutar:

```
sudo apt-get update
sudo apt-get install pound
```

Y comprobamos que está funcionando con `sudo systemctl status pound`.

Vamos a editar el fichero de configuración `/etc/pound/pound.cfg` para que quede como en (41).

Y relanzamos el servicio con `sudo systemctl restart pound` y probamos a lanzarlo, pero no funciona, comprobamos el estado con `sudo systemctl status pound` y nos sale un mensaje diciendo que configuremos `startup=1` en `/etc/default/pound`, como se ve en (42).

Editamos `/etc/default/pound` dejándolo como en (43).

Tras esto relanzamos el servicio y comprobamos que ahora sí funciona, y al haber establecido las prioridades con `m1` el doble de `m2`, tenemos que `m1` recibe el doble de peticiones que `m2`.

7.1 OPCIONES AVANZADAS

Podemos añadir, por ejemplo, a las directivas globales `TimeOut`, que es el tiempo que se espera una respuesta del backend.

`xHTTP` define los verbos HTTP que se aceptan.

Además, si usamos `Emergency`, el servidor solo se usará cuando el resto de servidores fallen:

Editamos el fichero de configuración con estas opciones, como se ve en (44).

Relanzamos el servicio y vemos que ahora solo nos atiende `m1`.

Figura 41: Fichero de configuración /etc/pound/pound.cfg .

```

GNU nano 4.8 /etc/pound/pound.cfg Modified

## use hardware-acceleration card supported by openssl(1):
#SSLEngine      "<hw>"

# poundctl control socket
Control "/var/run/pound/poundctl.socket"

#####
## listen, redirect and ... to:

## redirect all requests on port 8080 ("ListenHTTP") to the local webserver (see "Service" below):
ListenHTTP
    Address 192.168.56.103
    Port    80

    ## allow PUT and DELETE also (by default only GET, POST and HEAD)??:
    xHTTP   1

    Service
        Backend
            Address 192.168.56.101
            Port    80
            Priority 2
        End
        Backend
            Address 192.168.56.102
            Port    80
            Priority 1
        End_
    End
End

```

Figura 42: Resultado de comprobar el estado de pound con systemctl status

```

anabuenrui@m3-anabuenrui:/$ sudo systemctl restart pound
anabuenrui@m3-anabuenrui:/$ sudo systemctl status pound
● pound.service - LSB: reverse proxy and load balancer
   Loaded: loaded (/etc/init.d/pound; generated)
   Active: active (exited) since Sun 2022-04-10 23:24:41 UTC; 17s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 2877 ExecStart=/etc/init.d/pound start (code=exited, status=0/SUCCESS)

abr 10 23:24:41 m3-anabuenrui systemd[1]: pound.service: Succeeded.
abr 10 23:24:41 m3-anabuenrui systemd[1]: Stopped LSB: reverse proxy and load balancer.
abr 10 23:24:41 m3-anabuenrui systemd[1]: Starting LSB: reverse proxy and load balancer...
abr 10 23:24:41 m3-anabuenrui pound[2877]: * pound will not start unconfigured.
abr 10 23:24:41 m3-anabuenrui pound[2877]: * Please configure; afterwards, set startup=1 in /etc/p
abr 10 23:24:41 m3-anabuenrui systemd[1]: Started LSB: reverse proxy and load balancer.
lines 1-12/12 (END)

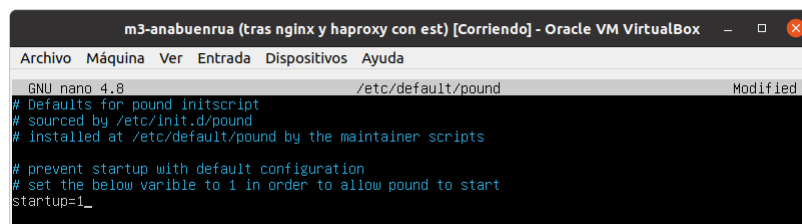
```

Además, al igual que en los otros sistemas hay algo similar a la ip hash, para que las peticiones de la misma ip las atienda la misma máquina en un margen de tiempo. Esto puede verse en (45).

Y relanzando comprobamos que siempre nos atiende la misma máquina.

Finalmente, y como con nginx y haproxy, se desactiva que se lance al inicio para evitar problemas por usar todos los servicios el mismo puerto con `sudo systemctl disable pound`.

Figura 43: Fichero /etc/default/pound .



The image shows a terminal window titled "m3-anabuenrwa (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox". The terminal is running GNU nano 4.8 and displays the contents of the file /etc/default/pound. The file contains several lines of configuration comments and a variable setting.

```
m3-anabuenrwa (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/default/pound Modified
# Defaults for pound initscript
# sourced by /etc/init.d/pound
# installed at /etc/default/pound by the maintainer scripts

# prevent startup with default configuration
# set the below variable to 1 in order to allow pound to start
startup=1_
```

Figura 44: Fichero /etc/pound/pound.cfg con opciones avanzadas (emergency).

```

m3-anabuenrúa (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/pound/pound.cfg Modified
## Minimal sample pound.cfg
##
## see pound(8) for details

#####
## global options:

User      "www-data"
Group     "www-data"

TimeOut 10
#RootJail  "/chroot/pound"

## Logging: (goes to syslog by default)
##      0      no logging
##      1      normal
##      2      extended
##      3      Apache-style (common log format)
LogLevel 1

## check backend every X secs:
Alive     30

## use hardware-acceleration card supported by openssl(1):
#SSLEngine  "<hw>"

# poundctl control socket
Control "/var/run/pound/poundctl.socket"

#####
## listen, redirect and ... to:

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo
^X Exit      ^R Read File  ^N Replace   ^U Paste Text ^I To Spell ^G Go To Line M-E Redo

```

```

m3-anabuenrúa (tras nginx y haproxy con est) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 4.8 /etc/pound/pound.cfg Modified
#SSLEngine  "<hw>"

# poundctl control socket
Control "/var/run/pound/poundctl.socket"

#####
## listen, redirect and ... to:

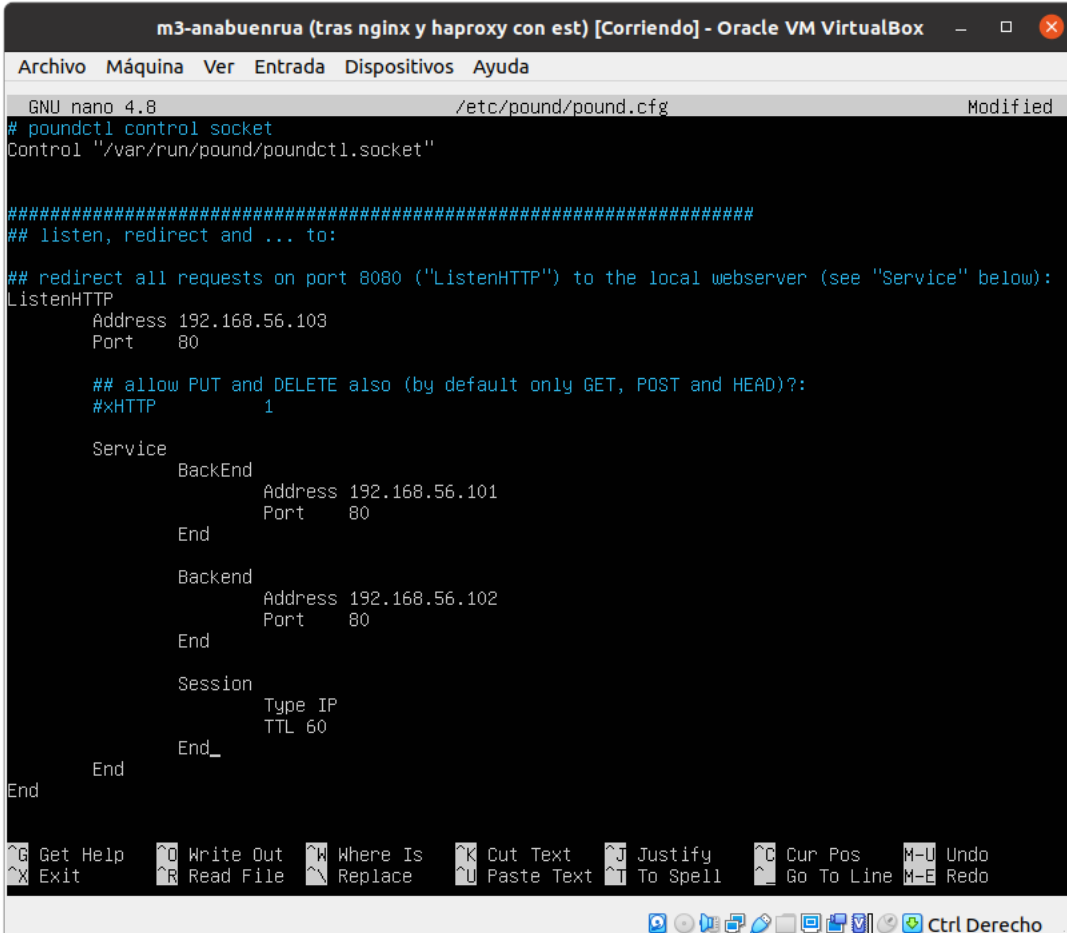
## redirect all requests on port 8080 ("ListenHTTP") to the local webserver (see "Service" below):
ListenHTTP
    Address 192.168.56.103
    Port    80

    ## allow PUT and DELETE also (by default only GET, POST and HEAD)?:
    #xHTTP 1

    Service
        Backend
            Address 192.168.56.101
            Port    80
        End
        Emergency
            Address 192.168.56.102
            Port    80
        End
    End
End

```

Figura 45: Fichero /etc/pound/pound.cfg con opciones avanzadas (session).



```

GNU nano 4.8 /etc/pound/pound.cfg Modified
# poundctl control socket
Control "/var/run/pound/poundctl.socket"

#####
## listen, redirect and ... to:

## redirect all requests on port 8080 ("ListenHTTP") to the local webserver (see "Service" below):
ListenHTTP
    Address 192.168.56.103
    Port    80

    ## allow PUT and DELETE also (by default only GET, POST and HEAD)?:
    #xHTTP 1

    Service
        Backend
            Address 192.168.56.101
            Port    80
        End
        Backend
            Address 192.168.56.102
            Port    80
        End
        Session
            Type IP
            TTL 60
        End_
    End
End

```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
 ^X Exit ^R Read File ^N Replace ^U Paste Text ^I To Spell ^G Go To Line M-E Redo

Ctrl Derecho

SOMETER LA GRANJA WEB A UNA CARGA

Comenzamos instalando apache benchmark en la máquina anfitriona con el comando:

```
sudo apt-get install -y apache2-utils
```

Ahora, lanzamos nginx y lo configuramos con roundrobin, como antes.

Lanzamos el benchmark, con 10000 peticiones con concurrencia 10:

```
ab -n 10000 -c 10 http://192.168.56.103/index.html
```

Obtenemos la siguiente información, que puede verse en (46).

Donde lo mas relevante son las peticiones por segundo, que es el criterio principal que vamos a usar para comparar.

También llama la atención cuál ha sido el tiempo más largo para atender a una petición.

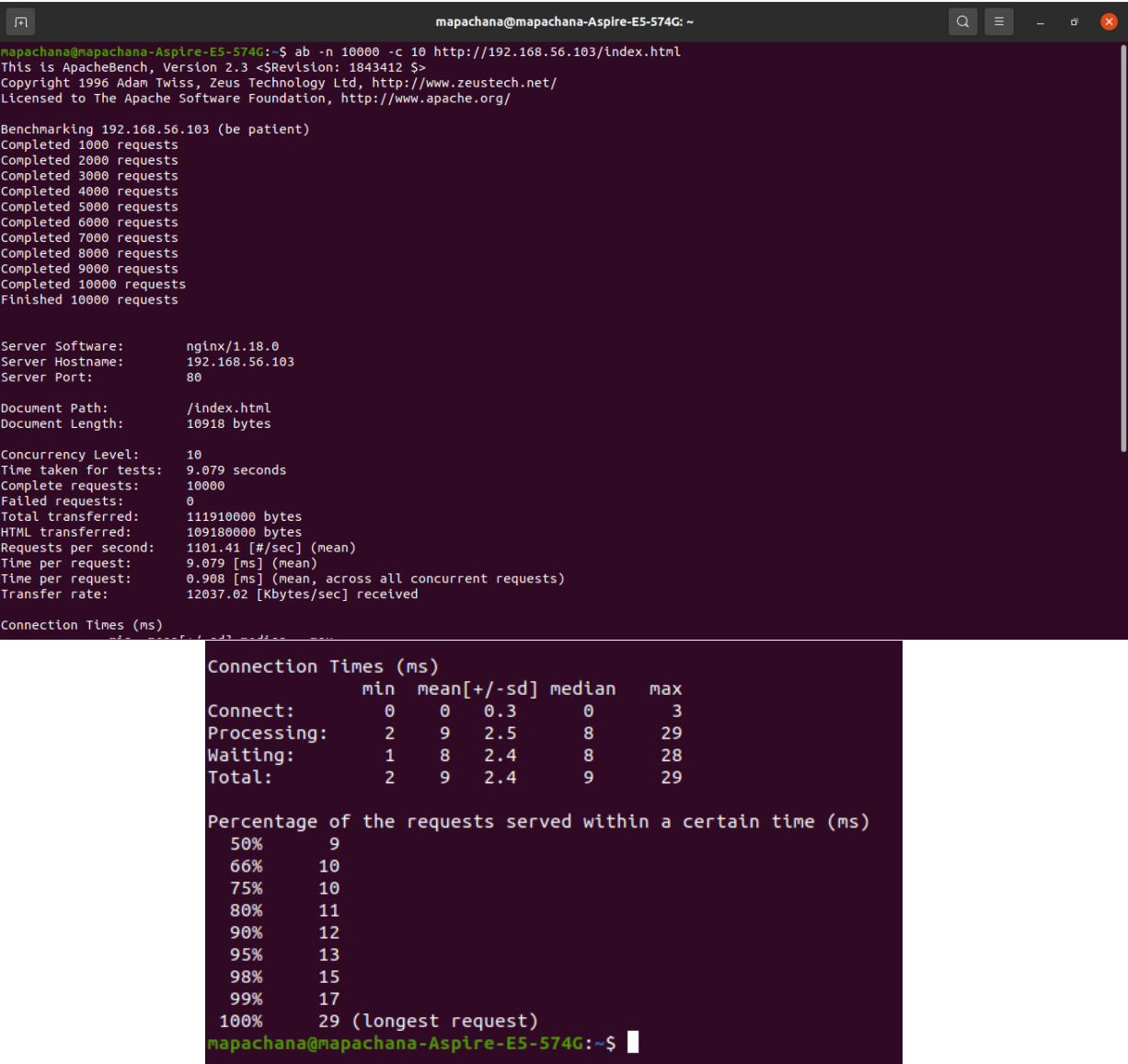
Otras opciones aparte de -n y -c para indicar el número de peticiones y la concurrencia son:

- -q: no muestra los mensajes de progreso.
- -v: modo verboso.
- -e file.csv: escribe el tiempo que se tardó en atender cierto porcentaje de las peticiones en el fichero indicado.
- -t: indica el tiempo máximo que va a ejecutarse el benchmark.
- -p y -u: indican si la petición es POST o PUT respectivamente, deben usarse con -T, que indica la cabecera del content-type.

Lanzamos este mismo benchmark a todos los servidores, con round-robin y ponderación, considerando que m1 tiene el doble de capacidad que m2.

Ahora, se someterá a la misma carga de 10000 peticiones con 10 de concurrencia a todos los balanceadores configurados, cada uno con round robin y ponderación, con m1 atendiendo el doble de peticiones que m2. Los resultados se muestran en la tabla (1).

Figura 46: Resultado de someter a carga a una granja web con nginx con 10000 peticiones y 10 de concurrencia.



Para hacer su comprensión más fácil e intuitiva, se han representado estos datos en una gráfica para las peticiones/s (47) y otra para las peticiones que más tiempo han llevado (48).

Se aprecia claramente como nginx consigue mejores prestaciones en roundrobin y ponderación que el resto de balanceadores, siendo así la mejor opción en principio.

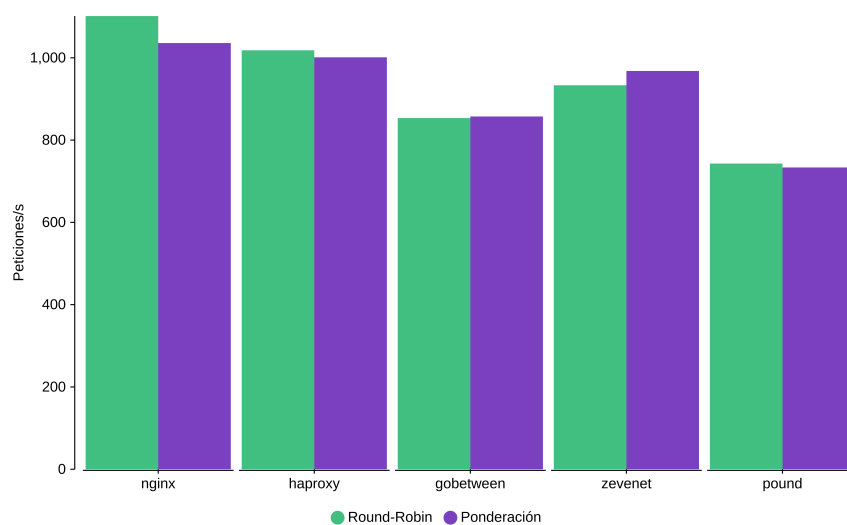
En segunda posición, se encontraría haproxy, seguido de zevenet y después gobetween.

Asimismo, destaca como pound consigue las peores en ambas configuraciones, convirtiéndose en la opción menos recomendable a escoger.

Cuadro 1: Resultados de someter a carga a las granjas con distintos balanceadores y configuraciones de round robin y ponderación

Balanceador	Modo	Peticiones/s	Longest request (ms)
nginx	round-robin	1101.41	29
nginx	ponderación	1035.80	46
haproxy	round-robin	1018.19	31
haproxy	ponderación	1001.13	42
gobetween	round-robin	853.46	39
gobetween	ponderación	857.36	44
zevenet	round-robin	933.23	42
zevenet	ponderación	967.92	51
pound	round-robin	742.90	132
pound	ponderación	733.41	121

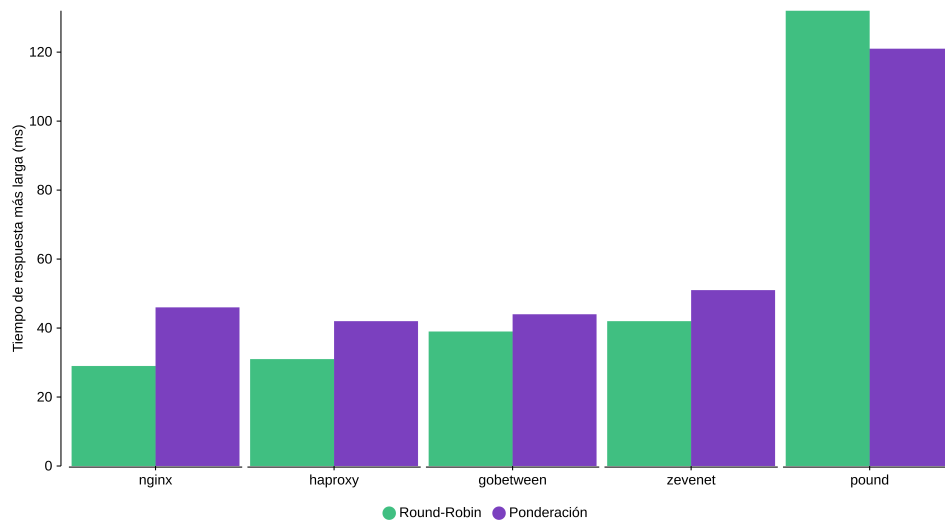
Figura 47: Comparación de peticiones/s respecto a configuraciones de round robin y ponderación en cada balanceador



Observando los tiempos de respuesta de las peticiones que más han tardado en atenderse, observamos una tendencia similar a la ya mencionada según las peticiones/s. El mejor sería nginx, seguido de haproxy, luego gobetween, zevenet y finalmente, por mucha distancia con el resto, pound.

Llama la atención como en la configuración por ponderación los tiempos de respuesta son, en general, significativamente mayores a los de round robin, esto puede deberse a que una máquina (m1) recibe el doble de peticiones que la otra, cuando ambas tienen la misma potencia, y por tanto las peticiones que más tardan pueden tardar casi el doble en ser atendidas que en round robin.

Figura 48: Comparación de mayor tiempo de respuesta respecto a configuraciones de round-robin y ponderación en cada balanceador



Pound es el único balanceador para el que los tiempos de respuestas con ponderación son menores a los tiempos de respuesta con roundrobin, lo que puede deberse a una casualidad (habría que lanzar el benchmark más veces y hacer la media) o simplemente ser así el funcionamiento de pound.

Este análisis no es concluyente sobre qué balanceador puede ser mejor para una granja web, pues influyen muchos más factores de los aquí mencionados, pero a juzgar por los datos, el mejor a priori sería nginx y el peor pound.

BIBLIOGRAFÍA

- Diapositivas y gui3n de la pr3ctica.
- <https://linux.die.net/man/8/pound>
- <https://es.zevenet.com>
- <https://snapcraft.io/install/gobetween/debian>
- https://ubuntu.pkgs.org/20.04/ubuntu-universe-amd64/pound_2.8-2_amd64.deb.html
- <https://www.zevenet.com/knowledge-base/community-edition/community-edition-v5-o-administration-guide/ce-v5-o-installation-guide/>
- <https://www.haproxy.com/documentation/hapee/latest/onepage/>
- <http://nginx.org/en/docs/>
- <https://gobetween.io/documentation.html>
- <https://www.zevenet.com/blog/tag/documentation/>