

Práctica 3 SWAP

En esta práctica se trata de aprender a balancear la carga entre nuestros dos servidores m1 y m2 mediante el uso de una tercera máquina virtual m3 con varios balanceadores software.

1. NGINX

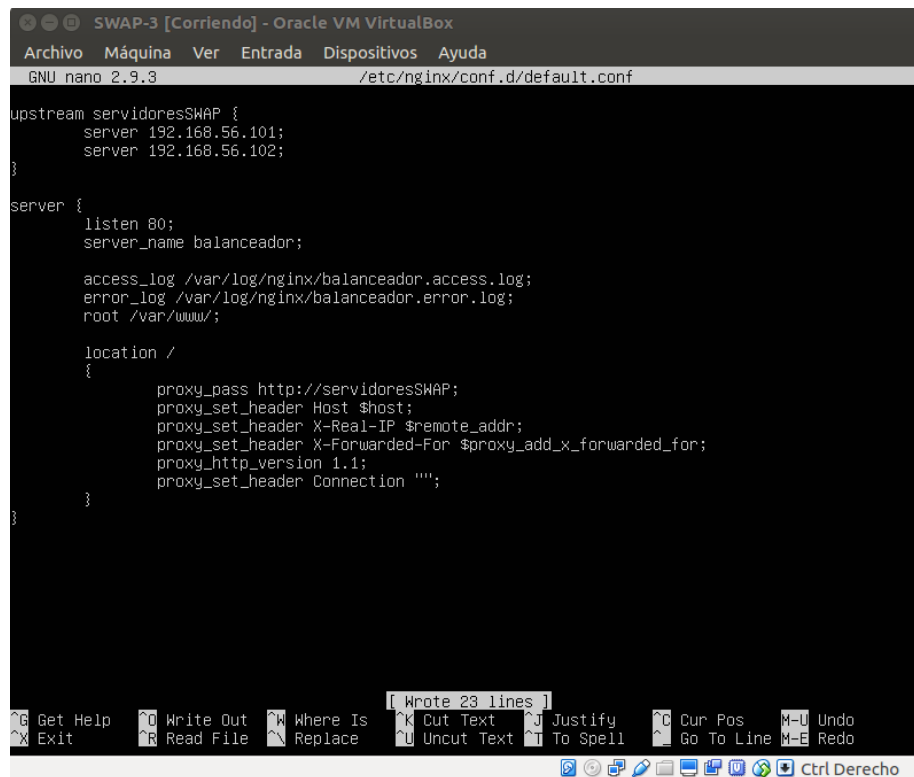
El primer balanceador que vamos a probar es NGINX. Este software es en realidad un servidor web ligero utilizado por varios sitios web de renombre. Además, tiene la funcionalidad de balanceador de carga, la cual es una de las funcionalidades más utilizadas para este software.

1.1. Configuración

Para configurar esta herramienta como balanceador de carga, debemos crear un archivo en `/etc/nginx/conf.d/default.conf`, dado que no existe en nuestro servidor tras su instalación.

En este archivo definimos un *upstream*, grupo al cual se va a redirigir el tráfico que reciba nuestro servidor web. En él se introducen las IP de nuestros servidores.

A continuación, definimos el código del servidor, es decir, nuestro balanceador. Debemos hacer que escuche el puerto 80, ponerle un nombre al servidor, definir los archivos de log, el root, el protocolo HTTP y la redirección de tráfico.



```
SWAP-3 [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 /etc/nginx/conf.d/default.conf

upstream servidoresSWAP {
    server 192.168.56.101;
    server 192.168.56.102;
}

server {
    listen 80;
    server_name balanceador;

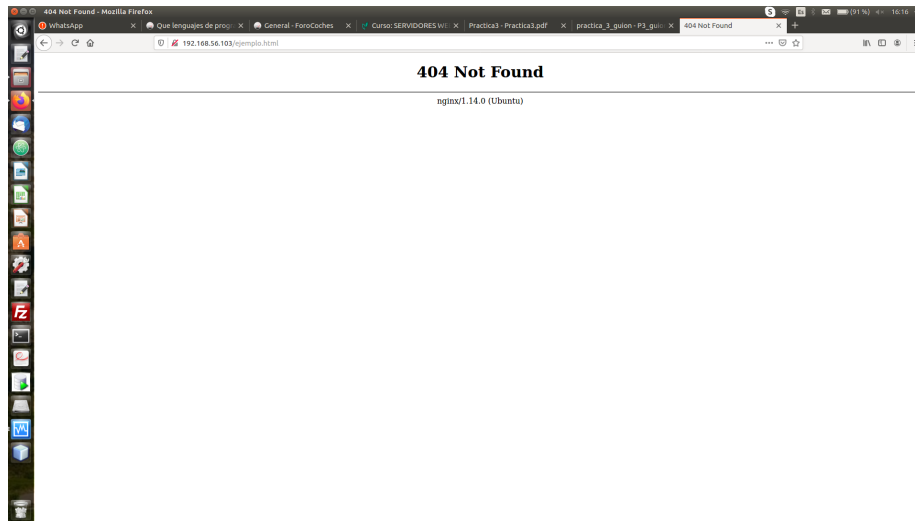
    access_log /var/log/nginx/balanceador.access.log;
    error_log /var/log/nginx/balanceador.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://servidoresSWAP;
        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 "";
    }
}
```

Si quisiéramos definir un reparto ponderado, debemos definir para cada servidor un parámetro *weight*, igual al peso que le queramos dar, con una línea del estilo de la siguiente:

```
server ip1 weight=2;
server ip2 weight=1;
```

Reiniciamos el servicio nginx y probamos desde nuestro navegador, dándonos el siguiente resultado:



Este error se debe a que en el archivo de configuración de nginx situado en `/etc/nginx/nginx.conf` el servidor web está configurado como tal, con lo cual no llega a redirigir el tráfico. Para ello comentamos la línea que provoca este hecho, relativa a los sitios habilitados.

Una vez hecho esto, probamos a ejecutar curl hacia m3, y vemos lo siguiente:

```
aizpoz@aizpoz-HP-Notebook: ~  
<p>Soy la máquina 1</p>  
</body>  
</html>  
aizpoz@aizpoz-HP-Notebook:~$ curl 192.168.56.103/ejemplo.html  
<html>  
  <head>  
    <meta charset="utf-8">  
  </head>  
  <body>  
    <h1>Web de ejemplo de adizqpoz para SWAP</h1>  
    <p>Soy la máquina 1</p>  
  </body>  
</html>  
aizpoz@aizpoz-HP-Notebook:~$ curl 192.168.56.103/ejemplo.html  
<html>  
  <head>  
    <meta charset="utf-8">  
  </head>  
  <body>  
    <h1>Web de ejemplo de adizqpoz para SWAP</h1>  
    <p>Soy la máquina 2</p>  
  </body>  
</html>  
aizpoz@aizpoz-HP-Notebook:~$
```

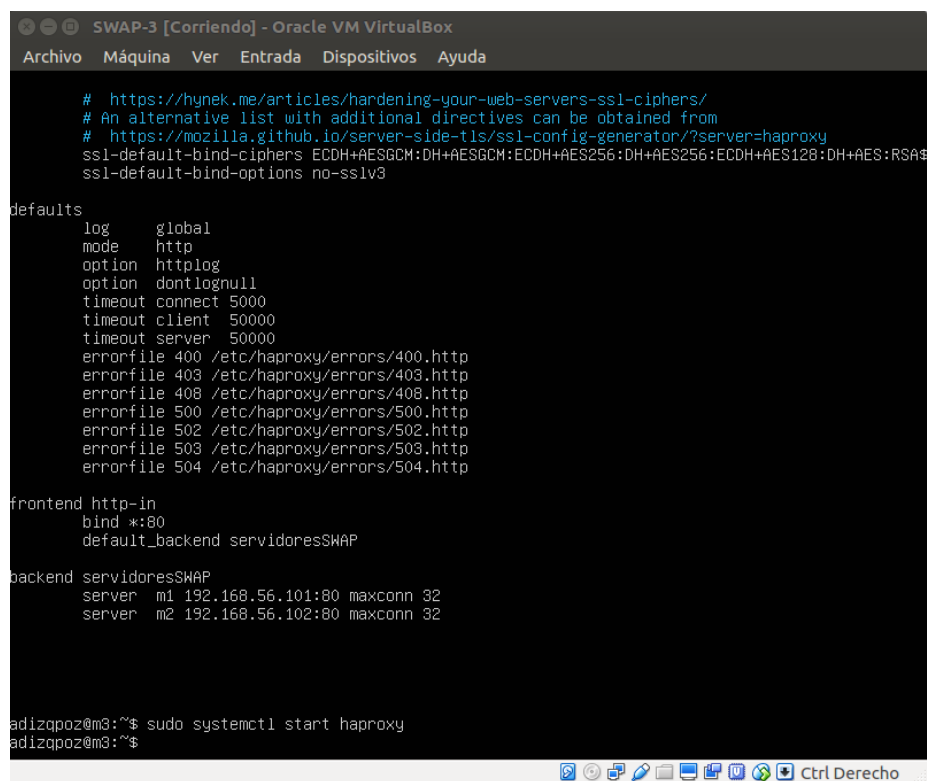
Con esto, ya hemos creado un balanceador de carga sencillo con NGINX.

2. HAProxy

HAProxy es un balanceador de carga que asume las funcionalidades de proxy, con lo cual es capaz de redirigir cualquier tipo de tráfico.

Se trata de un balanceador de altas prestaciones, con lo cual su uso está muy indicado en caso de usar un balanceador software.

Tras instalarlo, para realizarle la configuración hay que definir el frontend, es decir, la recepción de peticiones desde el exterior, y el backend, en el que incluimos nuestros servidores, y definimos el método de reparto.

A screenshot of a terminal window titled "SWAP-3 [Corriendo] - Oracle VM VirtualBox". The window shows the configuration of HAProxy. At the top, there are comments about SSL configurations with links to hynek.me and mozilla.github.io. The configuration includes a 'defaults' block with settings for log, mode, option, timeout, and errorfiles. Below that is a 'frontend http-in' block with 'bind *:80' and 'default_backend servidoresSWAP'. Then there is a 'backend servidoresSWAP' block with two servers: 'm1 192.168.56.101:80 maxconn 32' and 'm2 192.168.56.102:80 maxconn 32'. At the bottom, the user runs 'sudo systemctl start haproxy' and the prompt returns.

```
# https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/  
# An alternative list with additional directives can be obtained from  
# https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy  
ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA  
ssl-default-bind-options no-sslv3  
  
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 servidoresSWAP  
  
backend servidoresSWAP  
    server  m1 192.168.56.101:80 maxconn 32  
    server  m2 192.168.56.102:80 maxconn 32  
  
adizqpoz@m3:~$ sudo systemctl start haproxy  
adizqpoz@m3:~$
```

Si quisiéramos definir un reparto ponderado, debemos definir para cada servidor un parámetro *weight*, igual al peso que le queramos dar, con una línea del estilo de la siguiente:

```
server m1 ip1:80 maxconn 32 weight 2  
server m2 ip2:80 maxconn 32 weight 1
```

Posteriormente, iniciamos el servicio haproxy, asegurándonos de que nginx está inhabilitado, y probamos el balanceo:

```
aizpoz@aizpoz-HP-Notebook:~$ curl 192.168.56.103/ejemplo.html
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Web de ejemplo de adizqpoz para SWAP</h1>
    <p>Soy la máquina 1</p>
  </body>
</html>
aizpoz@aizpoz-HP-Notebook:~$ curl 192.168.56.103/ejemplo.html
<html>
  <head>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Web de ejemplo de adizqpoz para SWAP</h1>
    <p>Soy la máquina 2</p>
  </body>
</html>
aizpoz@aizpoz-HP-Notebook:~$
```

Con esto, ya hemos creado un balanceador de carga sencillo con HAProxy, con a simple vista una configuración más intuitiva que la de NGINX.

3. Aplicación de benchmark a nuestra granja web.

Una vez configurados ambos balanceadores debemos poner nuestra granja web a prueba. En este caso lo haremos con la herramienta *apache benchmark*.

Para ello, vamos a ejecutar la siguiente orden para ambos casos:

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

Para el balanceador NGINX obtenemos lo siguiente:

6

```

aizpoz@aizpoz-HP-Notebook:~$ ab -n 10000 -c 10 http://192.168.56.103/ejemplo.html
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.103 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests


Server Software:      nginx/1.14.0
Server Hostname:      192.168.56.103
Server Port:          80

Document Path:        /ejemplo.html
Document Length:      149 bytes

Concurrency Level:    10
Time taken for tests:  6.204 seconds
Complete requests:    10000
Failed requests:       0
Total transferred:    4180000 bytes
HTML transferred:     1490000 bytes
Requests per second:  1611.84 [#/sec] (mean)
Time per request:     6.204 [ms] (mean)
Time per request:     0.620 [ms] (mean, across all concurrent requests)
Transfer rate:        657.96 [Kbytes/sec] received


Connection Times (ms)
              min      mean[+/-sd] median   max
Connect:        0        0   0.6      0      6
Processing:      2        6   1.4      6     24
Waiting:         2        6   1.4      6     24
Total:          2        6   1.5      6     26


Percentage of the requests served within a certain time (ms)
 50%    6
 66%    6
 75%    7
 80%    7
 90%    8
 95%    9
 98%   10
 99%   11
100%   26 (longest request)
aizpoz@aizpoz-HP-Notebook:~$

```

Y para el balanceador HAProxy obtenemos los siguientes datos:




```

aizpoz@aizpoz-HP-Notebook:~$ ab -n 10000 -c 10 http://192.168.56.103/ejemplo.html
This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.56.103 (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
Completed 5000 requests
Completed 6000 requests
Completed 7000 requests
Completed 8000 requests
Completed 9000 requests
Completed 10000 requests
Finished 10000 requests


Server Software:      Apache/2.4.29
Server Hostname:      192.168.56.103
Server Port:          80

Document Path:        /ejemplo.html
Document Length:      149 bytes

Concurrency Level:    10
Time taken for tests:  5.355 seconds
Complete requests:    10000
Failed requests:       0
Total transferred:    4190000 bytes
HTML transferred:     1490000 bytes
Requests per second:  1867.54 [#/sec] (mean)
Time per request:     5.355 [ms] (mean)
Time per request:     0.535 [ms] (mean, across all concurrent requests)
Transfer rate:        764.16 [Kbytes/sec] received

Connection Times (ms)
              min    mean[+/-sd] median    max
Connect:        0      0    0.5      0      6
Processing:      1      5    1.4      5     17
Waiting:         1      5    1.4      5     17
Total:          2      5    1.4      5     18

Percentage of the requests served within a certain time (ms)
 50%      5
 66%      5
 75%      6
 80%      6
 90%      7
 95%      8
 98%      9
 99%     11
100%     18 (longest request)

```

Como vemos, al parecer HAProxy es un balanceador de carga más eficiente que NGINX, con lo cual, en principio, preferiremos el uso del primero.

Autor: Adrián Izquierdo Pozo

Si desea ver el archivo Markdown puede verlo en mi repositorio de Github