

- **Instalación y configuración de nginx**

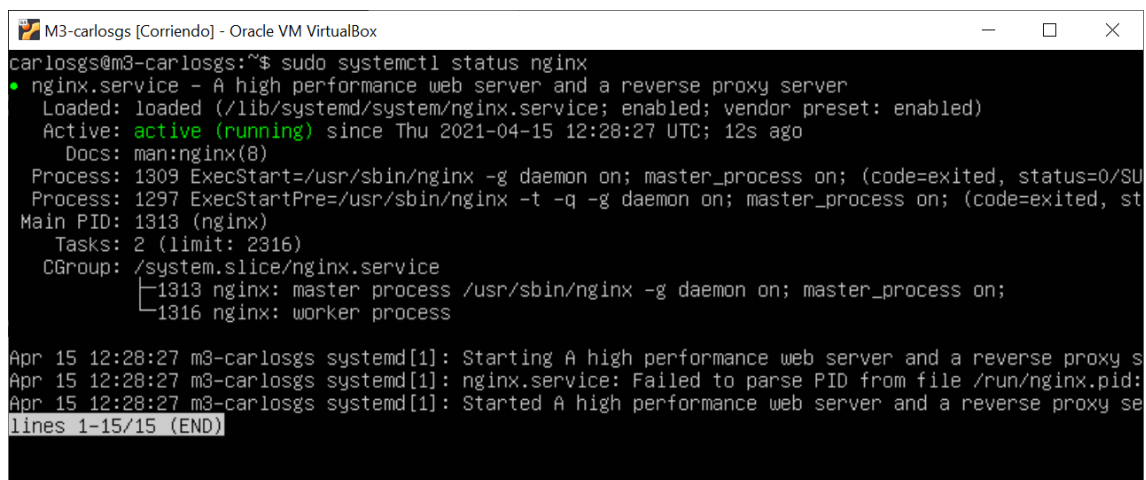
Lo primero es instalar nginx en la maquina encargada del balanceo (M3 en este caso)

**sudo apt-get update && sudo apt-get dist-upgrade && sudo apt-get autoremove**

**sudo apt-get install nginx**

**sudo systemctl start nginx**

Comprobamos con **sudo systemctl status nginx** que esta funcionando y ha sido instalado correctamente



```
M3-carlosgs [Corriendo] - Oracle VM VirtualBox
carlosgs@m3-carlosgs:~$ 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 Thu 2021-04-15 12:28:27 UTC; 12s ago
     Docs: man:nginx(8)
   Process: 1309 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 1297 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
  Main PID: 1313 (nginx)
    Tasks: 2 (limit: 2316)
   CGroup: /system.slice/nginx.service
           └─1313 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─1316 nginx: worker process

Apr 15 12:28:27 m3-carlosgs systemd[1]: Starting A high performance web server and a reverse proxy server: nginx.
Apr 15 12:28:27 m3-carlosgs systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid: Invalid PID '0'.
Apr 15 12:28:27 m3-carlosgs systemd[1]: Started A high performance web server and a reverse proxy server: nginx.
lines 1-15/15 (END)
```

Ahora nos metemos en el archivo de configuración en **/etc/nginx/conf.d/default.conf** y ponemos la configuración tal y como se indica en la práctica.

```
upstream balanceo_usuarioUGR
{
    server ip_maquinaM1;
    server ip_maquinaM2;
}

Server
{
    listen 80;
    server_name balanceador_usuarioUGR;

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

    location /
    {
        proxy_pass http://balanceo_usuarioUGR;
    }
}
```

```

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 "";
}
}

```

```

M3-carlosgs [Corriendo] - Oracle VM VirtualBox
GNU nano 2.9.3 /etc/nginx/conf.d/default.conf

upstream balanceo_carlosgs
{
    server 192.168.56.101;
    server 192.168.56.102;
}

server
{
    listen 80;
    server_name balanceador_carlosgs;

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

    location /
    {
        proxy_pass http://balanceo_carlosgs;
        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 "";
    }
}

carlosgs@m3-carlosgs:~$ sudo nano /etc/nginx/conf.d/default.conf

```

Una vez hecho esto usamos **sudo service nginx restart** y si no da ningún mensaje de error es que vamos bien.

```

M3-carlosgs [Corriendo] - Oracle VM VirtualBox
carlosgs@m3-carlosgs:~$ sudo service nginx restart
carlosgs@m3-carlosgs:~$ 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 Thu 2021-04-15 12:41:00 UTC; 33s ago
     Docs: man:nginx(8)
  Process: 1363 ExecStop=/sbin/start-stop-daemon --quiet --stop --retry QUIT/5 --pidfile /run/nginx.
  Process: 1378 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SU
  Process: 1367 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, st
 Main PID: 1380 (nginx)
    Tasks: 2 (limit: 2316)
   CGroup: /system.slice/nginx.service
           └─1380 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─1383 nginx: worker process

Apr 15 12:41:00 m3-carlosgs systemd[1]: Stopped A high performance web server and a reverse proxy se
Apr 15 12:41:00 m3-carlosgs systemd[1]: Starting A high performance web server and a reverse proxy s
Apr 15 12:41:00 m3-carlosgs systemd[1]: nginx.service: Failed to parse PID from file /run/nginx.pid:
Apr 15 12:41:00 m3-carlosgs systemd[1]: Started A high performance web server and a reverse proxy se
lines 1-17/17 (END)

```

Si no funciona correctamente tenemos que modificar el archivo `/etc/nginx/nginx.conf` comentando la línea **include /etc/nginx/sites-enabled/\***; Para que no configure nginx como servidor web.

Y hacer otra vez **sudo service nginx restart**

En el archivo de configuración anterior `“/etc/nginx/conf.d/default.conf”` modificando el upstream podemos modificar la carga que le asignamos a cada máquina con el modificador “weight” dándole un valor numérico que indica la carga. Por defecto en todas las máquinas es 1.

```
upstream balanceo_usuarioUGR
{
server ip_maquinaM1 weight=1;
server ip_maquinaM2 weight=2;
}
```

Cada 3 peticiones la máquina 2 se lleva 2 peticiones.

En nuestro caso vamos a poner la máquina uno con weight = 2 como dice la práctica. También tenemos la opción de que todo el tráfico que venga de una IP se sirva durante toda la sesión en la misma máquina usando un balanceo por IP añadiendo la directiva **ip\_hash** al upstream. Lo malo de esto es que todos los usuarios detrás de un proxy o de una NAT irán a la misma máquina por lo que el balanceo puede que no sea equilibrado.

```
upstream balanceo_usuarioUGR
{
ip_hash;
server ip_maquinaM1;
server ip_maquinaM2;
}
```

También tenemos la opción de usar keepalive que debería de suponer una mejora de rendimiento en nuestras máquinas debido a que permite mantener una conexión con una persistencia de múltiples peticiones HTTP en vez de crear una conexión nueva para cada petición. Para ello usamos la directiva **keepalive <tiempo de mantenimiento de la conexión>**

```
upstream balanceo_usuarioUGR
{
server ip_maquinaM1;
server ip_maquinaM2;
keepalive 3;
}
```

Ademas contamos con otras opciones:

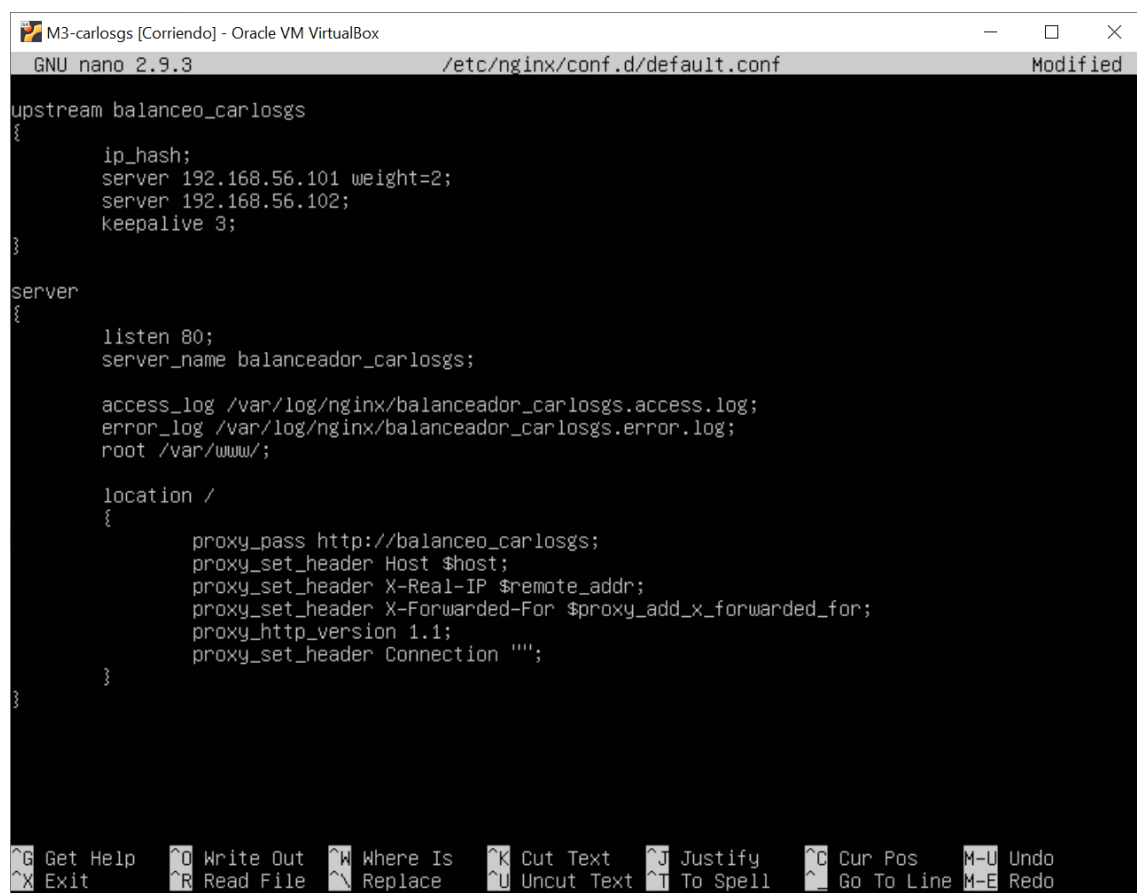
como el poder especificar el numero de intentos de comunicación erroneos con

- **max\_fails = NUMBER** sirve para especificar el numero de intentos de comunicación erroneos.
- **fail\_timeout = TIME** sirve para especificar el numero de segundos necesarios para considerar a servidor no operativo (es 1 por defecto)
- **down** sirve para marcar un servidor como permanentemente offline (para ser usado con ip\_hash)
- **max\_conns = NUMBER** sirve para limitar el numero de conexiones simultaneas activas al servidor

```
upstream balanceo_usuarioUGR
{
ip_hash
server ip_maquinaM1 max_fails=3 fail_timeout=30s;
server ip_maquinaM2 down;
server ip_maquinaM3;
}
```

Si la maquina 1 falla la petición pasa a la maquina 3

En nuestro caso solo vamos a activar el balanceo por IP y keepalive.



The screenshot shows a terminal window titled "M3-carlosgs [Corriendo] - Oracle VM VirtualBox". The terminal is running the GNU nano 2.9.3 text editor, editing the file /etc/nginx/conf.d/default.conf. The configuration file contains the following content:

```
upstream balanceo_carlosgs
{
    ip_hash;
    server 192.168.56.101 weight=2;
    server 192.168.56.102;
    keepalive 3;
}

server
{
    listen 80;
    server_name balanceador_carlosgs;

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

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

The terminal window also shows a status bar at the bottom with various keyboard shortcuts for editing and navigation.

Al cambiar el archivo tenemos que acordarnos de usar **sudo service nginx restart** para que se apliquen los cambios.

Si hacemos un curl a la maquinaM3 **curl 192.168.56.3** nos muestra la pagina de inicio de una de las maquinas lo que significa que esta repartiendo las peticiones entre las maquinas.

```
M3-carlosgs [Corriendo] - Oracle VM VirtualBox
document root directory in <tt>/etc/apache2/apache2.conf</tt>.
</p>
<p>
    The default Ubuntu document root is <tt>/var/www/html</tt>. You
    can make your own virtual hosts under /var/www. This is different
    to previous releases which provides better security out of the box.
  </p>
</div>

<div class="section_header">
  <div id="bugs"></div>
    Reporting Problems
  </div>
<div class="content_section_text">
  <p>
    Please use the <tt>ubuntu-bug</tt> tool to report bugs in the
    Apache2 package with Ubuntu. However, check <a
    href="https://bugs.launchpad.net/ubuntu/+source/apache2"
    rel="nofollow">existing bug reports</a> before reporting a new bug.
  </p>
  <p>
    Please report bugs specific to modules (such as PHP and others)
    to respective packages, not to the web server itself.
  </p>
</div>

</div>
</div>
<div class="validator">
</div>
</body>
</html>

carlosgs@m3-carlosgs:~$
```

- **Instalación y configuración de haproxy**

Lo primero es instalarlo en la encargada del balanceo usando el comando

**sudo apt-get install haproxy**

Una vez instalado nos vamos al archivo de configuración **/etc/haproxy/haproxy.cfg** para indicarle cuando son nuestros servidores backend y que peticiones queremos que balancee.

El balanceador debe escuchar por el puerto 80 y redirigirlo a uno de los servidores.

Dentro del mismo archivo en **global** podemos cambiar parámetros como el número máximo de conexiones activas simultáneamente totales con **maxconn NUMBER** o hacer que se ejecute en segundo plano con **daemon**. En **defaults** podemos configurar el tiempo, ya sea en microsegundos (us), milisegundos(ms), segundos(s), minutos(m), horas(h) o días(d), necesario para considerar al servidor, al cliente o a la conexión fallida. Por último, en **defaults**, en **backend** o en **listen** se puede usar **balance <algorithm>** que sirve para seleccionar el algoritmo que se utilizará a la hora de seleccionar que servidor recibirá una petición. Para poner el algoritmo roundrobin usamos **balance roundrobin** si queremos que sea ponderado entonces solo tenemos que añadir **weight** tal y como se muestra a continuación.

```
global
    daemon
    maxconn 256

defaults
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

frontend http-in
    bind *:80
    default_backend balanceo_usuarioUGR

backend balanceo_usuarioUGR
    server m1 ip_maquinaM1:80 maxconn 32 weigh 2
    server m2 ip_maquinaM2:80 maxconn 32
```

```
M3-carlosgs [Corriendo] - Oracle VM VirtualBox
GNU nano 2.9.3 /etc/haproxy/haproxy.cfg

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_carlosgs

backend balanceo_carlosgs
    balance roundrobin
    server  m1 192.168.56.101:80 maxconn 32 weight 2
    server  m2 192.168.56.102:80 maxconn 32

carlosgs@m3-carlosgs:~$ _
```

Una vez hemos terminado de configurar el servicio como siempre usamos el comando

**sudo service haproxy restart**

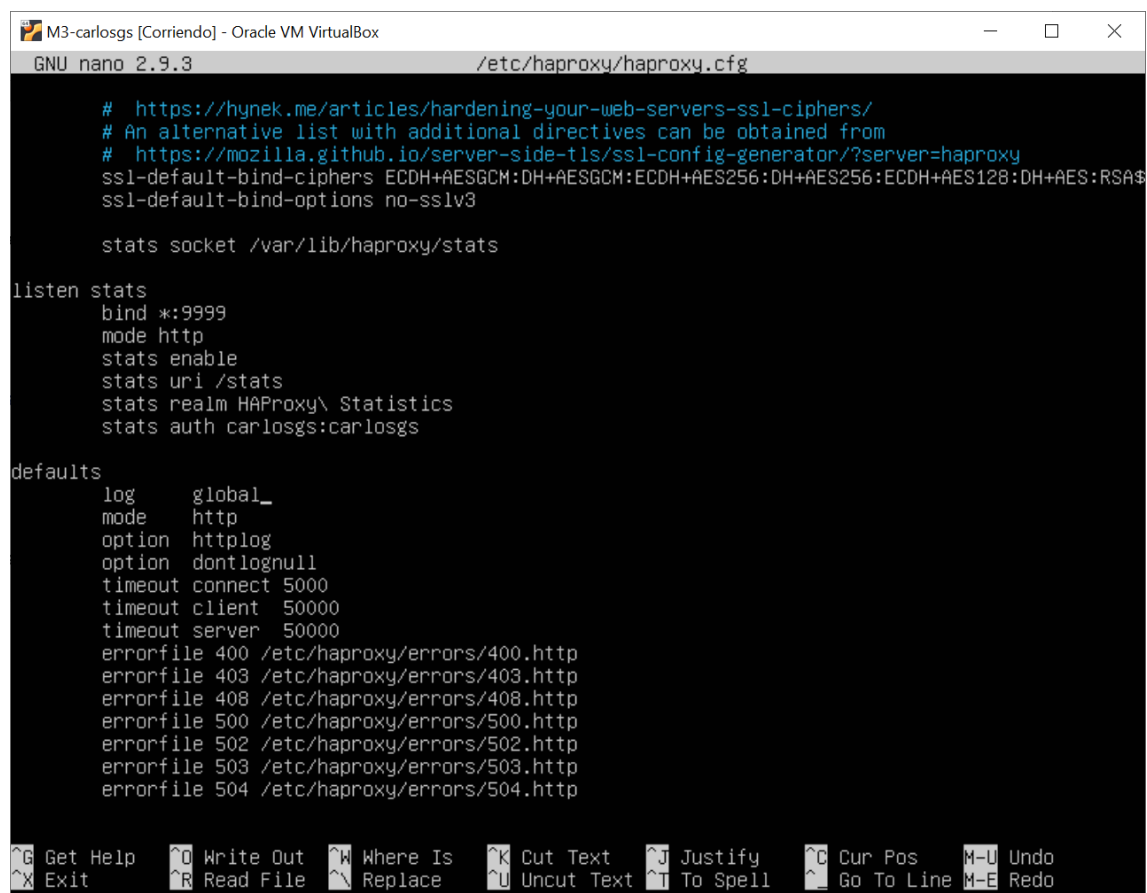
Si hacemos un **curl 192.168.56.103/ejemplo.html** vemos que nos devuelve la pagina web que creamos para una de las otras maquinas en las practicas anteriores.

```
M3-carlosgs [Corriendo] - Oracle VM VirtualBox
carlosgs@m3-carlosgs:~$ curl 192.168.56.103/ejemplo.html
<HTML>
<BODY>
Web de ejemplo de carlosgs para SWAP M1
Email: carlosgs@correo.ugr.es
</BODY>
</HTML>
carlosgs@m3-carlosgs:~$
```

Ahora vamos a habilitar las estadísticas del balanceador. Para ello volvemos al archivo de configuración `/etc/haproxy/haproxy.cfg` y añadimos

```
global
    stats socket /var/lib/haproxy/stats

listen stats
    bind 192.168.56.103:9999
    mode http
    stats enable
    stats uri / stats
    stats realm HAProxy\ Statistics
    stats auth carlosgs:carlosgs
```



The screenshot shows a terminal window titled "M3-carlosgs [Corriendo] - Oracle VM VirtualBox". The window displays the configuration file `/etc/haproxy/haproxy.cfg` being edited with the `GNU nano 2.9.3` editor. The configuration includes comments about SSL ciphers, a `stats` section for global statistics, a `listen stats` section for listening on `192.168.56.103:9999` in `http` mode with authentication, and a `defaults` section for general settings like `log`, `mode`, `option`, and `timeout`. The bottom of the window shows the nano editor's command palette with various shortcuts like `^G Get Help`, `^O Write Out`, etc.

```
# 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

stats socket /var/lib/haproxy/stats

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

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
```

Ahora si ponemos en nuestro navegador la dirección <http://192.168.56.103:9999/stats> nos pedirá autenticación que en nuestro caso es la que hemos puesto en el archivo de configuración y nos mostrará lo siguiente.



