

# Balanceador Galera Haproxy

Ahora procederemos a configurar un balanceador en alta disponibilidad usando HAProxy y Keepalived

En primer lugar instalaremos los paquetes necesarios

```
apt install haproxy net-tools keepalived
```

Procederemos a configurar HAProxy, para ello deberemos de crear un usuario en nuestro Cluster de Galera, para las peticiones de conectividad.

```
CREATE USER 'haproxy' '@' %' ;  
GRANT PROCESS ON *.* TO 'haproxy' '@' %' ;
```

Ahora configuraremos HAProxy Editamos el archivo de configuración

```
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=intermediate  
    ssl-default-bind-ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-  
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20  
    ssl-default-bind-options ssl-min-ver TLSv1.2 no-tls-tickets  
  
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

listen galera_cluster
    bind 0.0.0.0:3306
    mode tcp
    #option tcplog
    option tcpka
    option mysql-check user haproxy
    balance source
    server db1 192.168.15.221:3306 check
    server db2 192.168.15.222:3306 check
    server db3 192.168.15.223:3306 check

listen stats
    bind 0.0.0.0:8080
    mode http
    option httplog
    stats enable
    stats uri /
    stats realm Strictly\ Private
    stats auth admin:MIPASSWORD

```

En las opciones de configuración pondremos en el balance source, existe la opción de roundrobin, o leastconn, pero teniendo en cuenta que es un sistema de tolerancia y no de reparto de carga, la mejor opción es la de source.

Una vez configurado, procederemos a probar.

En primer lugar arrancaremos el HAProxy

```
service haproxy start
```

Suponiendo que la IP de nuestro HAProxy sea la 192.168.15.222, desde cualquier máquina de la red, ejecutaremos:

```
mysql -uroot -pMYPASSWORD -h192.168.15.222 -e "show variables like 'wsrep_node_name' ;"
```

Nos devolverá algo así

```

+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_node_name | Nodo1 |
+-----+-----+

```

Si tenemos algún fallo del tipo "Layer7 wrong status: Host '193.168.15.224' is blocked because of many connection errors; unblock with 'mariadb-admin flush-hosts'" En la interfaz gráfica

ejecutaremos el comando:

```
mariadb-admin flush-hosts
```

Interfaz gráfica (en el puerto 8080)

A continuación dejaremos el servicio configurado para que arranque con nuestra máquina

```
systemctl enable haproxy
```

Ahora que esto funciona, vamos con el keepalived.

Para ello la IP que hemos configurado (La 192.168.15.224) será nuestra IP flotante, es decir se mantendrá como IP de acceso independientemente de a que servidor HAProxy atacemos.

Por esto, vamos a repetir la configuración en otra máquina a la cual le asignaremos la IP 192.168.15.226, y la que acabamos de configurar será la 192.168.15.225. Esto puede resultar un poco lioso, pero vamos a ver porqué. Cada equipo tiene una IP la 225 en el que acabamos de configurar, y la 226 en otro equipo nuevo.

Ambos tienen IP diferentes, pero comparten una IP global a través del Keealived.

```
nano /etc/keepalived/keepalived.conf
```

Y escribiremos esto:

```
global_defs {
    notification_email {
        sistemas@ateinco.com.com
    }
    notification_email_from sistemas@ateinco.com
    smtp_server 127.0.0.1
    smtp_connect_timeout 30
    router_id LVS_DEVEL
}
vrrp_script chk_haproxy {
    script "killall -0 haproxy"
    interval 1
    weight -2
}
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    mcast_src_ip 192.168.15.225
```

```

priority 100
advert_int 1
authentication {
    auth_type PASS
    auth_pass 876543
}
virtual_ipaddress {
    192.168.15.224/24 dev eth0 label eth0:1
}
track_interface {
    eth0
}
track_script {
    chk_haproxy
}
}

```

Como vemos el equipo en la configuración de Keepalived tiene dos IP (la 225 y la 224) y un valor importante el priority que hemos puesto en 100

Ahora modificaremos el fichero `/etc/sysctl.conf`

Y agregaremos este valor

```
net.ipv4.ip_nonlocal_bind = 1
```

Además comprobaremos que la IP de nuestra tarjeta de red y el nombre de la interfaz están bien configurados en nuestro ejemplo de keepalived hemos puesto eth0, pero puede ser ens18, ens19, etc. Ahora arrancamos keepalived

```
systemctl start keepalived
```

Y comprobamos que todo funciona, volvemos a ejecutar:

```
mysql -uroot -pMYPASSWORD -h192.168.15.222 -e "show variables like 'wsrep_node_name' ;"
```

Si todo está correcto habilitamos keepalived al inicio

```
systemctl enable keepalived
```

Ahora ya tenemos un nodo configurado. Vamos al otro.

Básicamente la configuración es gemela a la de este nodo, salvo por la configuración de keepalived

Tendremos que cambiar la IP que será la 192.168.15.226 y la priority que la pondremos a 99

```

global_defs {
    notification_email {
        sistemas@ateinco.com.com
    }
}

```

```
notification_email_from sistemas@ateinco.com
smtp_server 127.0.0.1
smtp_connect_timeout 30
router_id LVS_DEVEL
}
vrrp_script chk_haproxy {
    script "killall -0 haproxy"
    interval 1
    weight -2
}
vrrp_instance VI_1 {
    state MASTER
    interface eth0
    virtual_router_id 51
    mcast_src_ip 192.168.15.225
    priority 99
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 876543
    }
    virtual_ipaddress {
        192.168.15.224/24 dev eth0 label eth0:1
    }
    track_interface {
        eth0
    }
    track_script {
        chk_haproxy
    }
}
```

---

Revision #1

Created 17 May 2022 14:28:22 by Admin

Updated 17 May 2022 14:28:49 by Admin