

haproxy – heartbeat – nginx

Servidores Web de Altas Prestaciones

2015

Leopoldo Castillo Maldonado
Juan Esteban Moreno López

INDICE

1. Introducción.....	3
2. Configuración interfaces.....	7
3. Heartbeat.....	13
4. HAProxy.....	16
5. Nginx.....	18
6. Funcionamiento.....	19
7. Wireshark.....	23
8. Referencias.....	25

1 Introducción

Vamos a configurar una granja web con alta disponibilidad con balanceo de carga en Ubuntu Server 12.04 utilizando HAProxy, HeartBeat y Nginx.

Para ello vamos a crear 5 máquinas virtuales:

haproxy1 – *Es el balanceador maestro (ejecuta Heartbeat y HAProxy) [Dirección IP: 192.168.1.101]*

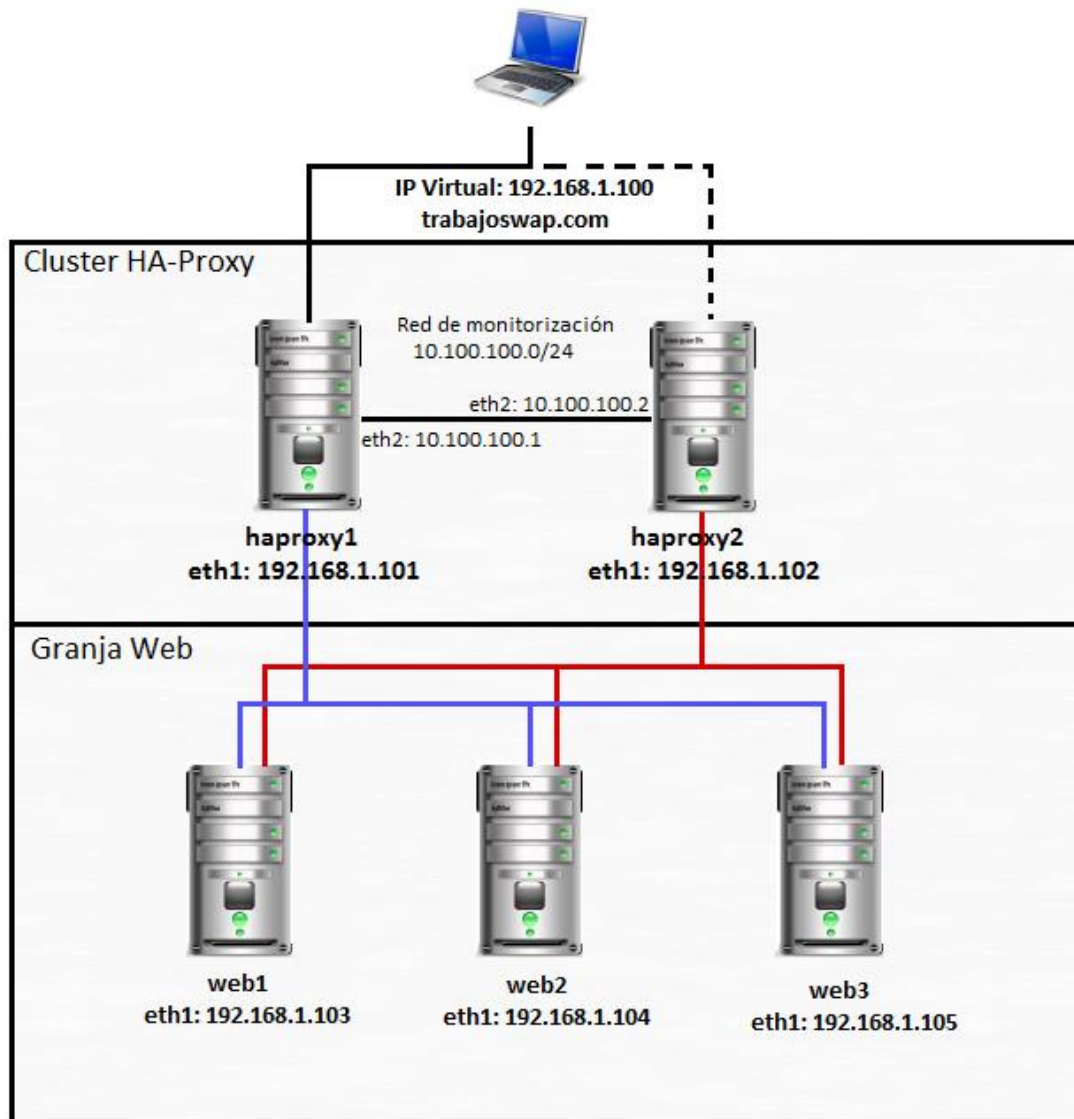
haproxy2 – *Es el balanceador secundario (failover) (ejecuta Heartbeat y HAProxy) [Dirección IP: 192.168.1.102]*

web1 – *Primer servidor web (ejecuta nginx) [IP address: 192.168.1.103]*

web2 – *Segundo servidor web (ejecuta nginx) [IP address: 192.168.1.104]*

web3 – *Tercer servidor web (ejecuta nginx) [IP address: 192.168.1.105]*

El siguiente esquema muestra cómo serán conectados los servidores:

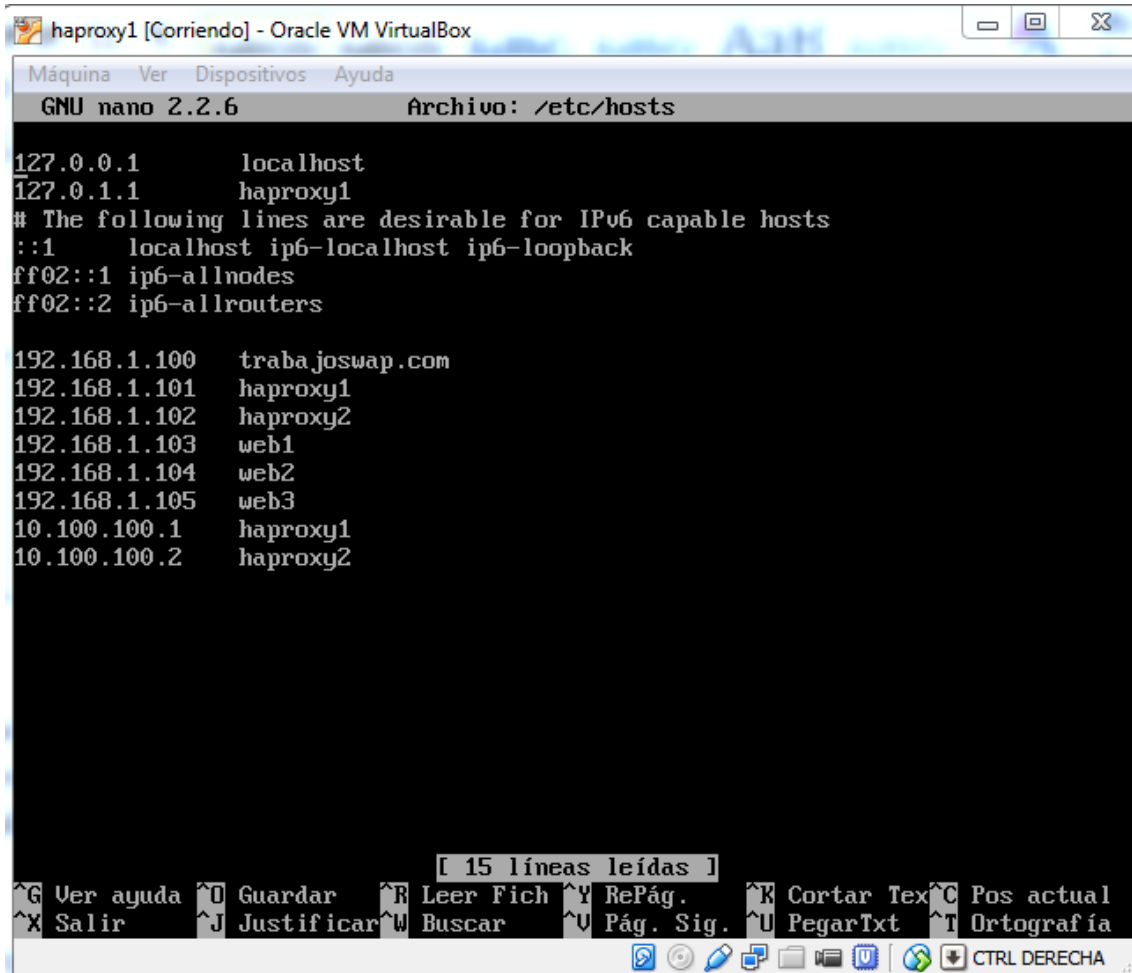


En primer lugar, añadimos lo siguiente en cada uno de los archivos “/etc/hosts” para asegurarnos de que todas las máquinas pueden resolver los nombres de host:

```
192.168.1.100      trabajoswap.com #Esta es nuestra IP flotante (IP virtual)
para nuestra granja web.
192.168.1.101      haproxy1
192.168.1.102      haproxy2
192.168.1.103      web1
192.168.1.104      web2
192.168.1.105      web3
10.100.100.1       haproxy1
```

10.100.100.2	haproxy2
--------------	----------

haproxy1:



The screenshot shows a terminal window titled 'haproxy1 [Corriendo] - Oracle VM VirtualBox'. The terminal is running the GNU nano 2.2.6 text editor, editing the file /etc/hosts. The content of the file is as follows:

```
127.0.0.1    localhost
127.0.1.1    haproxy1
# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters

192.168.1.100  trabajoswap.com
192.168.1.101  haproxy1
192.168.1.102  haproxy2
192.168.1.103  web1
192.168.1.104  web2
192.168.1.105  web3
10.100.100.1   haproxy1
10.100.100.2   haproxy2
```

The terminal window includes a status bar at the bottom with various keyboard shortcuts and a message indicating that 15 lines have been read.

haproxy2:

```

GNU nano 2.2.6 Archivo: /etc/hosts

127.0.0.1    localhost
127.0.1.1    haproxy2

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

192.168.1.100  trabajoswap.com
192.168.1.101  haproxy1
192.168.1.102  haproxy2
192.168.1.103  web1
192.168.1.104  web2
192.168.1.105  web3
10.100.100.1   haproxy1
10.100.100.2   haproxy2

[ 16 líneas leídas ]
Ver ayuda  Guardar  Leer Fich  RePág.  Cortar Tex  Pos actual
Salir  Justificar  Buscar  Pág. Sig.  PegarTxt  Ortografía
CTRL DERECHA

```

Servidores web:

```

web1 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6 Archivo: /etc/hosts

127.0.0.1    localhost
127.0.1.1    web1

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

192.168.1.100  trabajoswap.com
192.168.1.101  haproxy1
192.168.1.102  haproxy2
192.168.1.103  web1
192.168.1.104  web2
192.168.1.105  web3
10.100.100.1   haproxy1
10.100.100.2   haproxy2

[ 17
usuario@web1:~$

```

```

web2 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6 Archivo: /etc/hosts

127.0.0.1    localhost
127.0.1.1    web2

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

192.168.1.100  trabajoswap.com
192.168.1.101  haproxy1
192.168.1.102  haproxy2
192.168.1.103  web1
192.168.1.104  web2
192.168.1.105  web3
10.100.100.1   haproxy1
10.100.100.2   haproxy2

[ 17
usuario@web2:~$

```

```

web3 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6 Archivo: /etc/hosts

127.0.0.1    localhost
127.0.1.1    web3

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

192.168.1.100  trabajoswap.com
192.168.1.101  haproxy1
192.168.1.102  haproxy2
192.168.1.103  web1
192.168.1.104  web2
192.168.1.105  web3
10.100.100.1   haproxy1
10.100.100.2   haproxy2

[ 16 líneas escr
usuario@web3:~$

```

2. Configuración interfaces

Para la configuración de las interfaces tendremos 3 tarjetas de red para los dos balanceadores y 2 tarjetas de red (eth0 y eth1) para los servidores web:

- eth0 -> NAT
- eth1 -> RED INTERNA.
- eth2 -> RED INTERNA.

eth0 la dejaremos por defecto y será la utilizada para tener acceso a internet.

eth1 será utilizada para la red interna a la que estarán conectados todos los equipos.

eth2 permitirá a Heartbeat enviar mensajes “Estoy vivo” constantemente entre sí, cuando el maestro deje de responder, el proxy de conmutación por error se hará cargo y se activará el balanceador esclavo.

La configuración de la red en **/etc/network/interfaces** debe quedar de la siguiente manera:

haproxy1 (balanceador maestro):

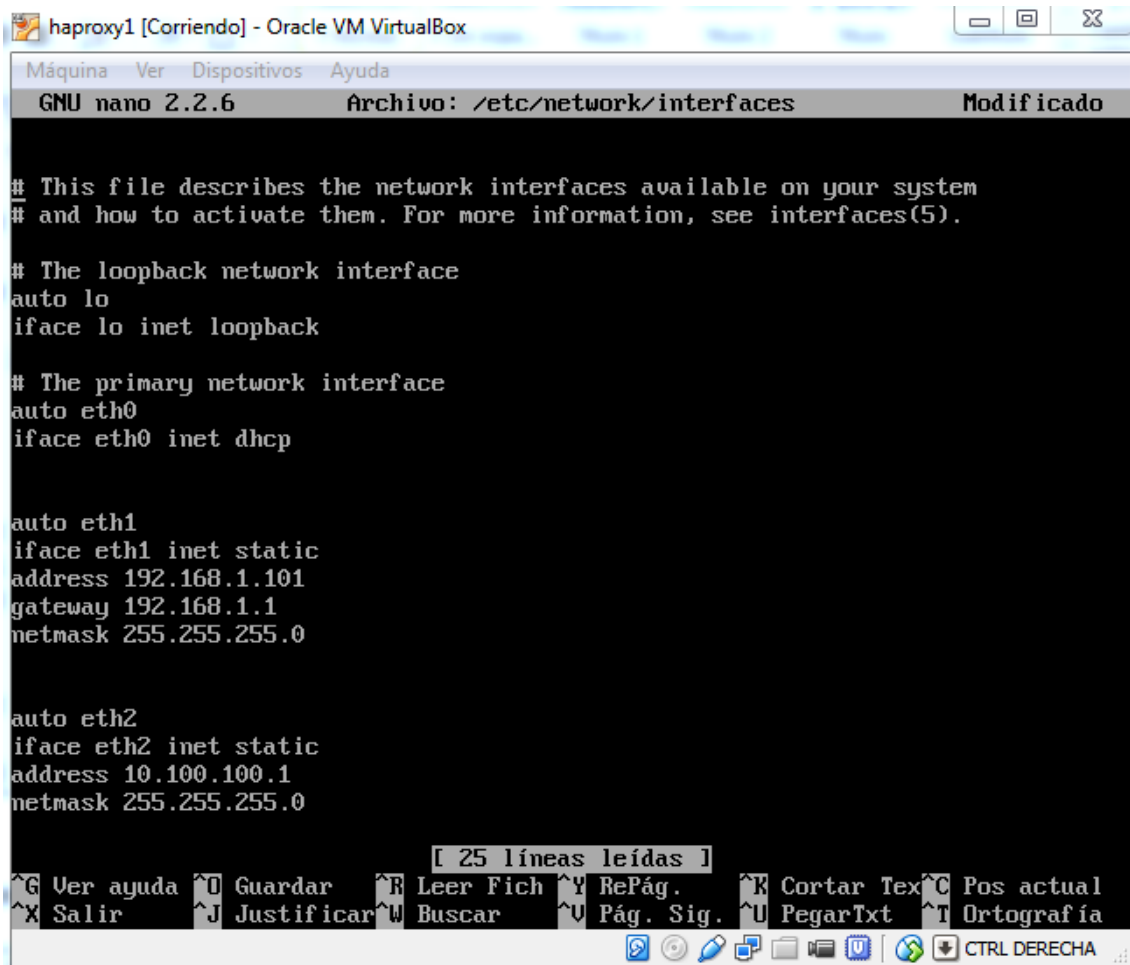
```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
    address 192.168.1.101
    gateway 192.168.1.1
    netmask 255.255.255.0

auto eth2
iface eth2 inet static
    address 10.100.100.1
    netmask 255.255.255.0
```



The screenshot shows a terminal window titled 'haproxy1 [Corriendo] - Oracle VM VirtualBox'. The window displays the GNU nano 2.2.6 editor editing the file /etc/network/interfaces. The content of the file is as follows:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.101
gateway 192.168.1.1
netmask 255.255.255.0

auto eth2
iface eth2 inet static
address 10.100.100.1
netmask 255.255.255.0
```

The terminal window also shows a status bar at the bottom with various keyboard shortcuts and a message '[25 líneas leídas]'.

haproxy2 (balanceador secundario):

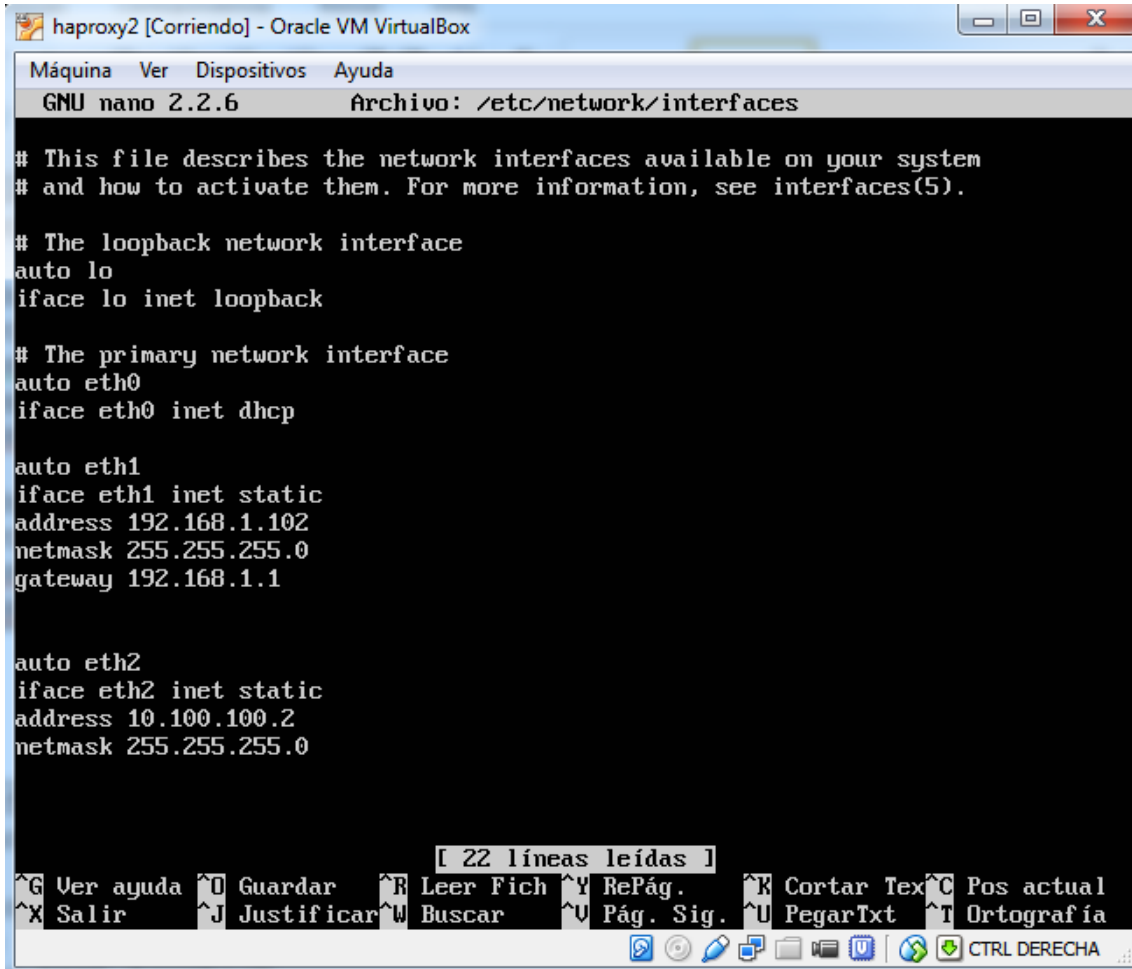
```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.102
gateway 192.168.1.1
netmask 255.255.255.0

auto eth2
iface eth2 inet static
address 10.100.100.2
netmask 255.255.255.0
```

The screenshot shows a terminal window titled 'haproxy2 [Corriendo] - Oracle VM VirtualBox'. Inside, the GNU nano 2.2.6 text editor is open, editing the file '/etc/network/interfaces'. The file contains configuration for four network interfaces: 'lo' (loopback), 'eth0' (primary, DHCP), 'eth1' (static, 192.168.1.102), and 'eth2' (static, 10.100.100.2). The nano editor's status bar at the bottom indicates '[22 líneas leídas]' and shows various keyboard shortcuts like '^G Ver ayuda', '^O Guardar', etc.

```
Máquina Ver Dispositivos Ayuda
GNU nano 2.2.6 Archivo: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.102
netmask 255.255.255.0
gateway 192.168.1.1

auto eth2
iface eth2 inet static
address 10.100.100.2
netmask 255.255.255.0

[ 22 líneas leídas ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^U Pág. Sig. ^U PegarTxt ^T Ortografía
CTRL DERECHA
```

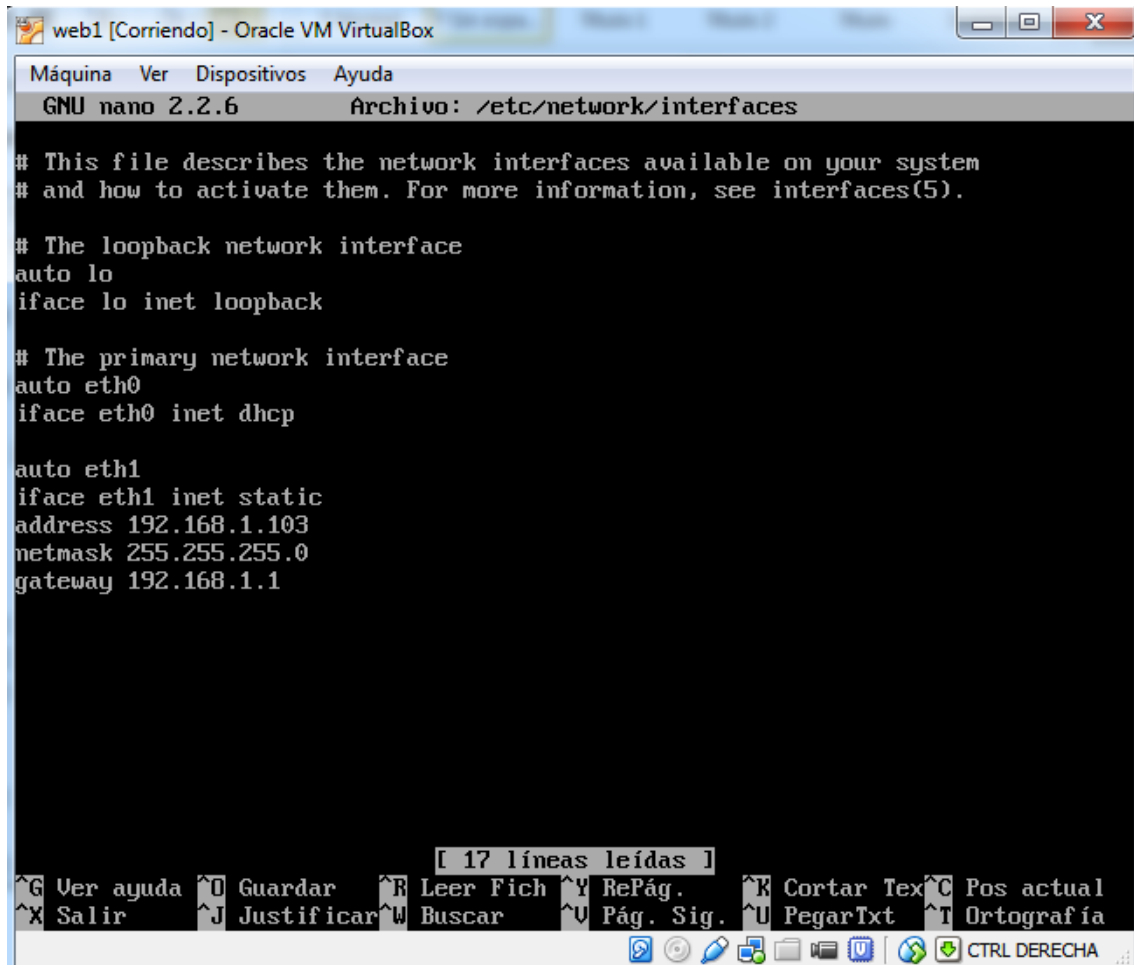
web1:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.103
netmask 255.255.255.0
gateway 192.168.1.1
```



```
web1 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6 Archivo: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.103
netmask 255.255.255.0
gateway 192.168.1.1

[ 17 líneas leídas ]
Ver ayuda  Guardar  Leer Fich  RePág.  Cortar Tex  Pos actual
Salir  Justificar  Buscar  Pág. Sig.  PegarTxt  Ortografía
CTRL DERECHA
```

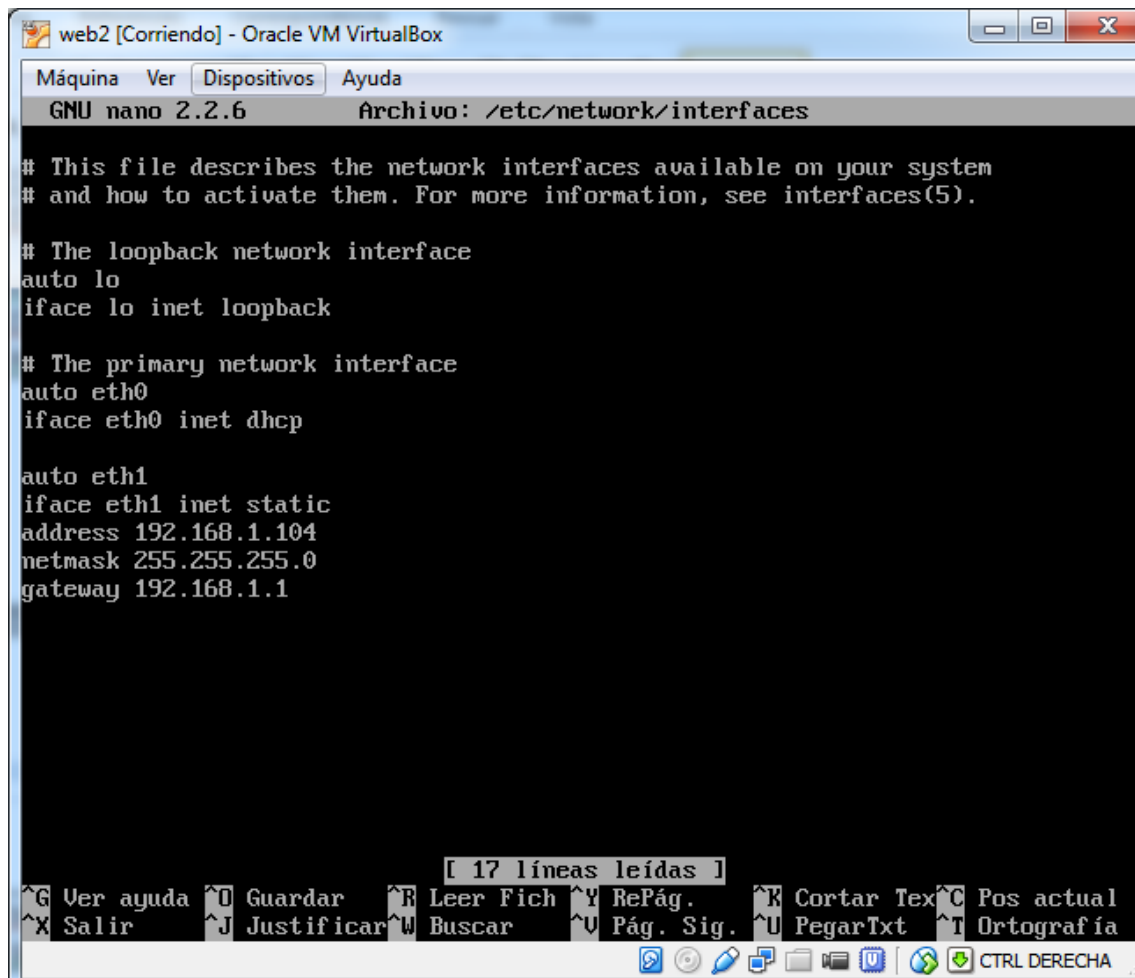
web2:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.104
netmask 255.255.255.0
gateway 192.168.1.1
```



```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.104
netmask 255.255.255.0
gateway 192.168.1.1
```

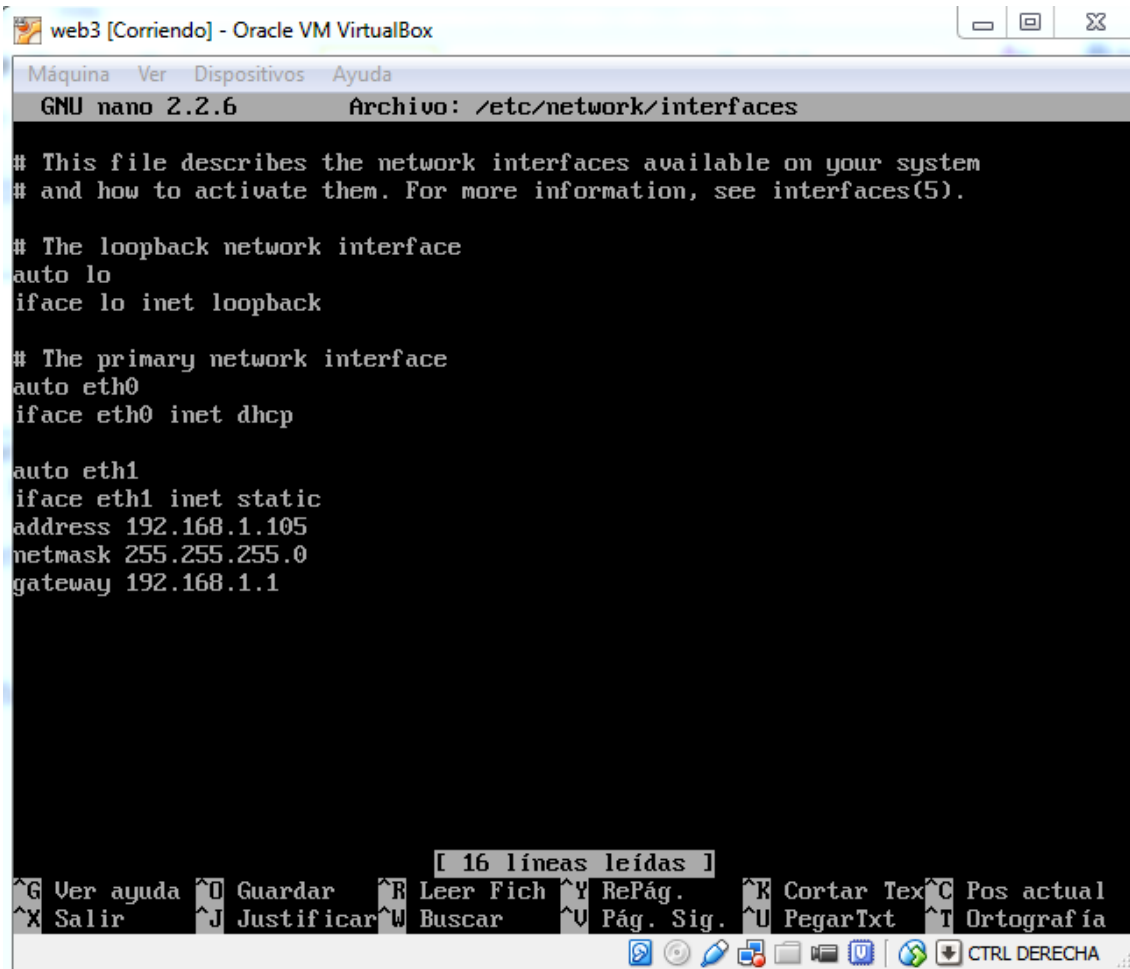
web3:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.105
netmask 255.255.255.0
gateway 192.168.1.1
```



The screenshot shows a terminal window titled 'web3 [Corriendo] - Oracle VM VirtualBox'. Inside, the GNU nano 2.2.6 text editor is open, editing the file /etc/network/interfaces. The file content is as follows:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 192.168.1.105
netmask 255.255.255.0
gateway 192.168.1.1
```

The nano editor's status bar at the bottom shows '[16 líneas leídas]' and a menu of keyboard shortcuts. The VM window's title bar includes standard minimize, maximize, and close buttons.

Una vez configurada la red reiniciamos el servicio de red con el siguiente comando:

```
$ sudo service networking restart
```

3. Heartbeat

Heartbeat es un demonio que proporciona infraestructura de servicios de clúster a sus clientes. Esto permite a los clientes saber sobre la presencia (o desaparición) de los procesos en otras máquinas e intercambiar fácilmente mensajes con ellos.

Instalación y Configuración

Para instalar heartbeat en Ubuntu ejecutamos el siguiente comando en ambos balanceadores:

```
$ sudo apt-get install heartbeat
```

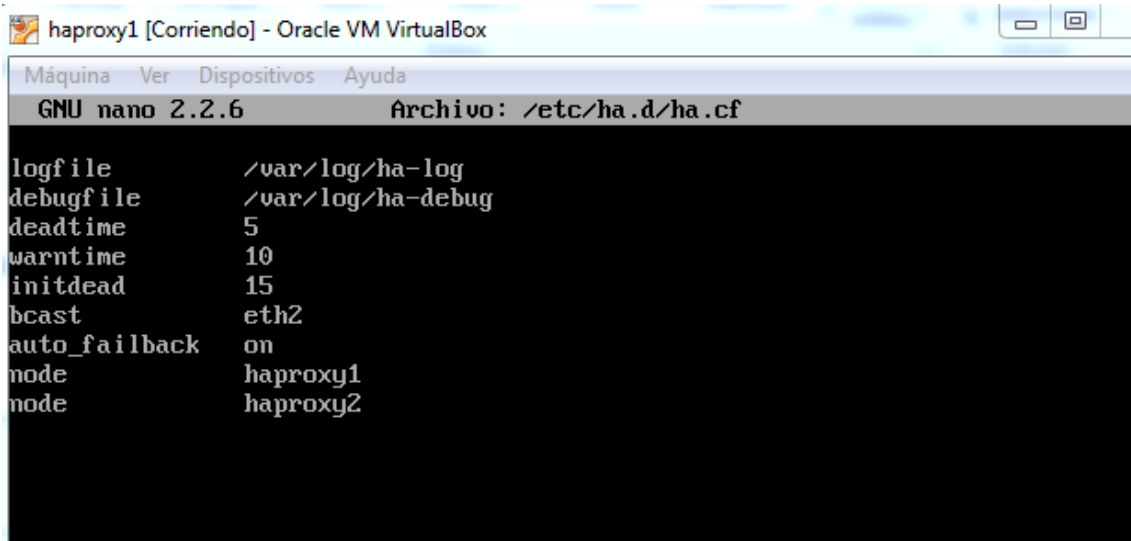
Esto creará un directorio en “/etc” llamado “ha.d”, donde vamos a trabajar.

Estos son los tres ficheros de configuración:

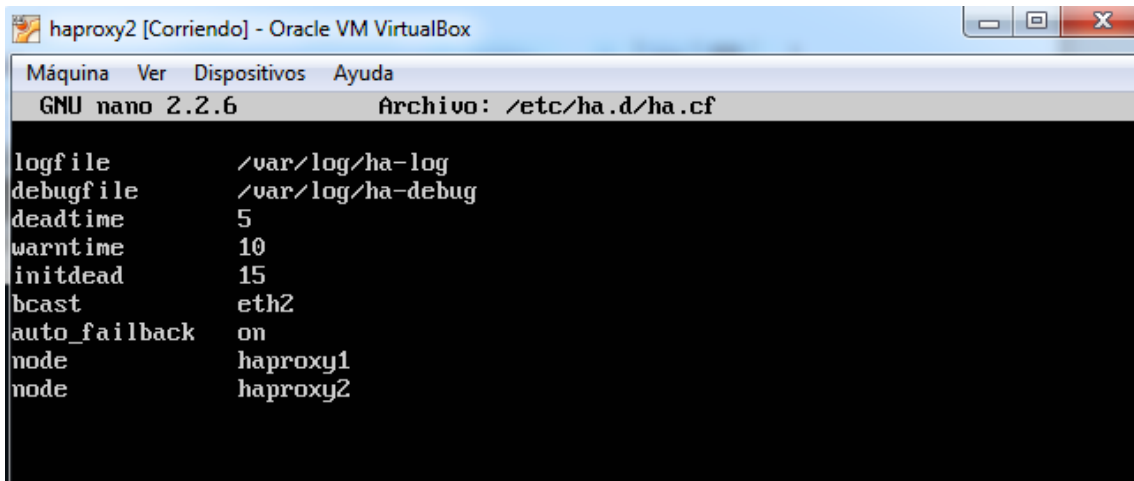
```
ha.cf
authkeys
haresources
```

Primero creamos el fichero **/etc/ha.d/ha.cf** en ambos balanceadores con el siguiente contenido:

```
logfile          /var/log/ha-log # Fichero que almacenará los logs
debugfile        /var/log/ha-debug # Fichero debug
deadtime         5 # Tras 5 segundos sin recibir respuesta del
otro nodo, se considera como muerto.
warntime         10 # Si durante 10 segundos no responde el otro
nodo, este nodo está en alerta
initdead         15 # Establece el intervalo inicial de detección
de tiempo muerto.
bcast            eth1 # Indica la interfaz de red donde se
levantará heartbeat
auto_failback    on # Permite que un nodo vuelva a ser maestro
automáticamente.
node             haproxy1 # Añade el nodo haproxy1 al cluster
node             haproxy2 # Añade el nodo haproxy2 al cluster
```

A screenshot of a terminal window titled 'haproxy1 [Corriendo] - Oracle VM VirtualBox'. The window shows the GNU nano 2.2.6 text editor editing the file /etc/ha.d/ha.cf. The configuration content is as follows:

```
logfile      /var/log/ha-log
debugfile    /var/log/ha-debug
deadtime     5
warntime     10
initdead     15
bcast        eth2
auto_failback on
node         haproxy1
node         haproxy2
```

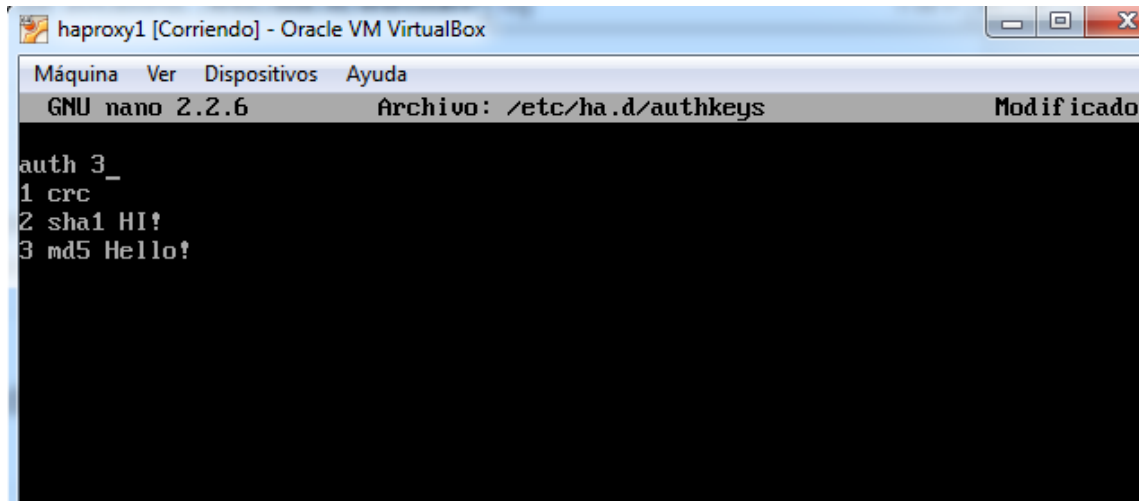
A screenshot of a terminal window titled 'haproxy2 [Corriendo] - Oracle VM VirtualBox'. The window shows the GNU nano 2.2.6 text editor editing the file /etc/ha.d/ha.cf. The configuration content is identical to the one in the haproxy1 window:

```
logfile      /var/log/ha-log
debugfile    /var/log/ha-debug
deadtime     5
warntime     10
initdead     15
bcast        eth2
auto_failback on
node         haproxy1
node         haproxy2
```

El siguiente fichero a configurar es **/etc/ha.d/authkeys**

Añadimos lo siguiente en ambos balanceadores:

```
auth 3
1 crc
2 sha1 yourpasswordhere
3 md5 yourpasswordhere
```



```
haproxy1 [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
GNU nano 2.2.6 Archivo: /etc/ha.d/authkeys Modificado

auth 3_
1 crc
2 sha1 HI!
3 md5 Hello!
```

Y le cambiamos los permisos del fichero a 600:

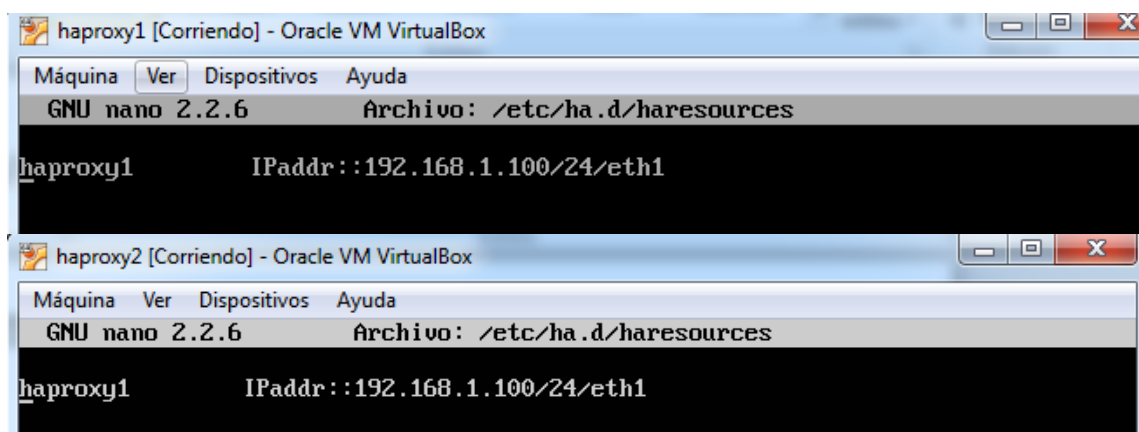
```
chmod 600 /etc/ha.d/authkeys
```

El último fichero a configurar es **/etc/ha.d/haresources**

Añadimos la siguiente línea en ambos balanceadores:

```
haproxy1 IPaddr::192.168.1.100/24/eth1
```

Donde se define el nombre del nodo maestro, dirección IP que utilizará Heartbeat para servir los recursos, máscara de subred en formato de bits y el nombre de interfaz de red donde se creará la interfaz virtual.



```
haproxy1 [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
GNU nano 2.2.6 Archivo: /etc/ha.d/haresources

haproxy1 IPaddr::192.168.1.100/24/eth1

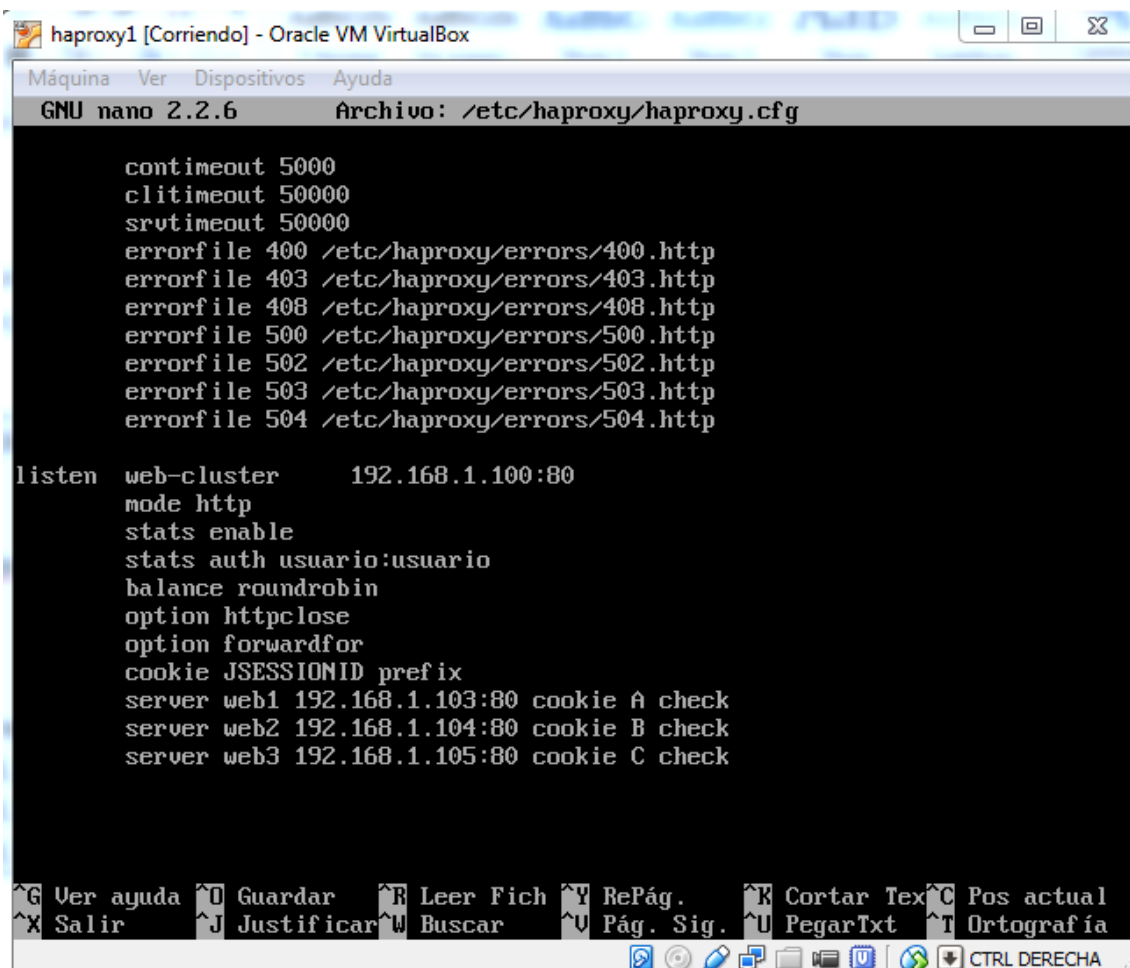
haproxy2 [Corriendo] - Oracle VM VirtualBox
Máquina Ver Dispositivos Ayuda
GNU nano 2.2.6 Archivo: /etc/ha.d/haresources

haproxy1 IPaddr::192.168.1.100/24/eth1
```

4. HAProxy

La configuración en ambos servidores es idéntica de nuevo por lo que añadimos las siguientes líneas en el fichero “/etc/haproxy/haproxy.cfg” de ambos nodos.

```
listen web-cluster 192.168.1.100:80
    mode http
    stats enable
    stats auth admin:password # Aquí ponemos nuestro usuario y contraseña
    balance roundrobin
    option httpclose
    option forwardfor
    cookie JSESSIONID prefix
    server web1 192.168.1.103:80 cookie A check
    server web2 192.168.1.104:80 cookie B check
    server web3 192.168.1.105:80 cookie C check
```



The screenshot shows a terminal window titled 'haproxy1 [Corriendo] - Oracle VM VirtualBox'. The window displays the configuration file '/etc/haproxy/haproxy.cfg' using the GNU nano 2.2.6 editor. The configuration includes global settings like timeouts and error files, followed by a 'listen' block for 'web-cluster' on IP 192.168.1.100:80. This block sets mode to http, enables stats with 'admin:usuario' authentication, uses roundrobin balancing, and defines three servers: web1, web2, and web3, each with a unique cookie and a 'check' option. The bottom of the window shows the nano editor's command palette with various shortcuts like 'Ver ayuda', 'Guardar', 'Leer Fich', etc.

```
haproxy1 [Corriendo] - Oracle VM VirtualBox
GNU nano 2.2.6 Archivo: /etc/haproxy/haproxy.cfg

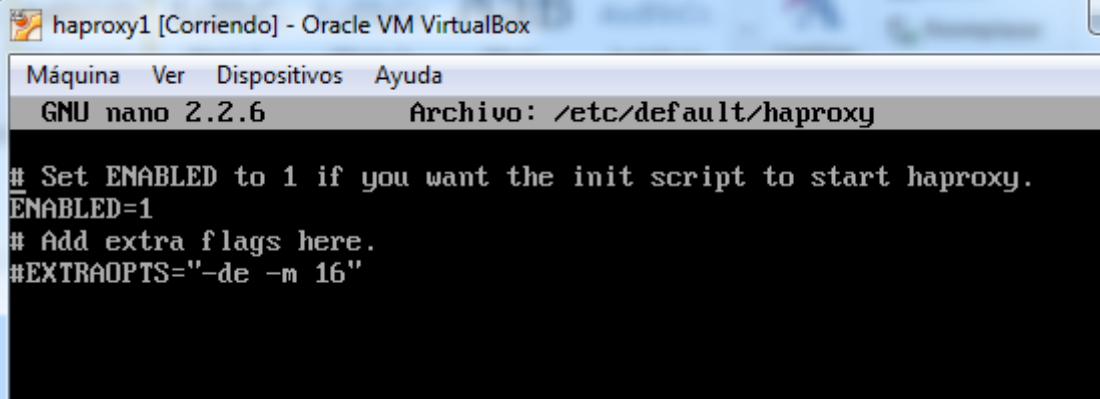
    timeout 5000
    clitimeout 50000
    srvtimerout 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

listen web-cluster 192.168.1.100:80
    mode http
    stats enable
    stats auth usuario:usuario
    balance roundrobin
    option httpclose
    option forwardfor
    cookie JSESSIONID prefix
    server web1 192.168.1.103:80 cookie A check
    server web2 192.168.1.104:80 cookie B check
    server web3 192.168.1.105:80 cookie C check
```


El último paso, antes de iniciar el servicio HAProxy es habilitar el demonio, de nuevo, este paso es el mismo en ambos servidores:

```
$ sudo nano /etc/default/haproxy
```

Y cambiamos: `ENABLED=0` a `ENABLED=1`



The screenshot shows a terminal window titled 'haproxy1 [Corriendo] - Oracle VM VirtualBox'. Inside the terminal, the GNU nano 2.2.6 editor is open, editing the file /etc/default/haproxy. The visible content of the file is:

```
# Set ENABLED to 1 if you want the init script to start haproxy.  
ENABLED=1  
# Add extra flags here.  
#EXTRAOPTS="-de -m 16"
```

Ahora iniciamos haproxy con el siguiente comando:

```
$ sudo /etc/init.d/haproxy start
```

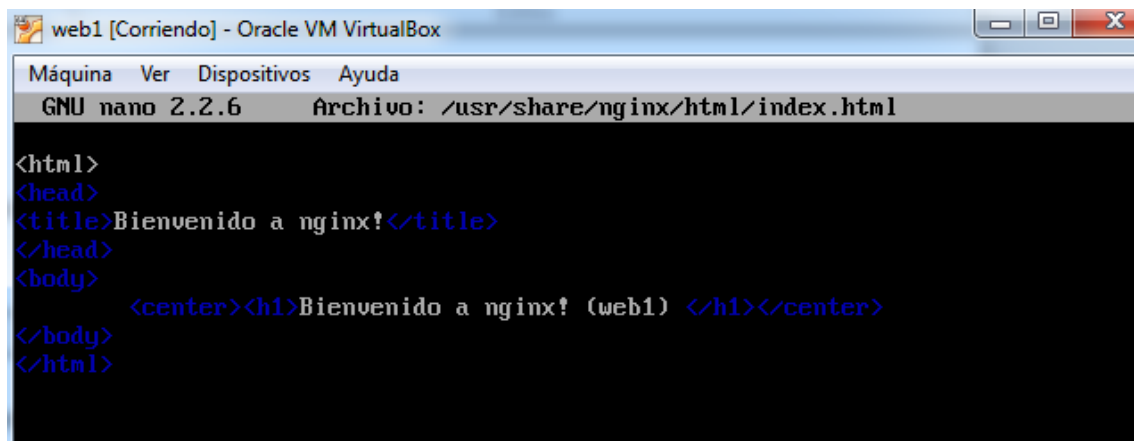
5. Nginx

Ahora vamos a instalar nginx en cada uno de los servidores web, para ello ejecutamos el siguiente comando:

```
$ sudo apt-get install nginx
```

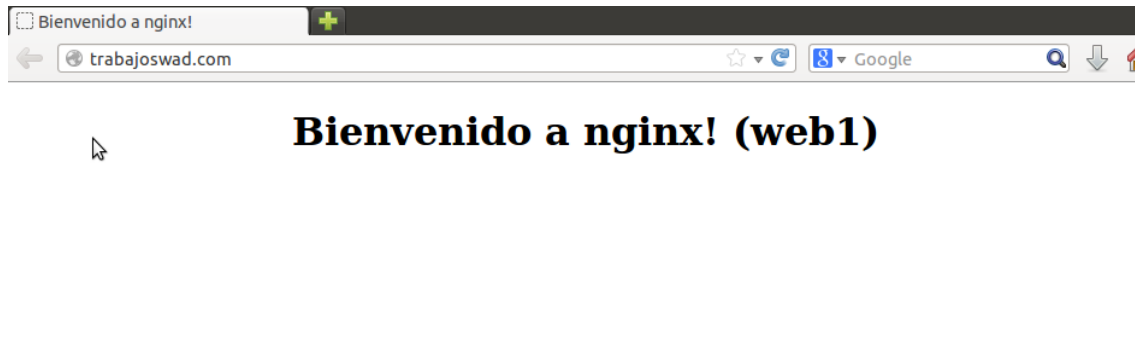
Para probar que el balanceador funciona correctamente editamos las páginas de cada servidor web. Para ello abrimos el archivo “/usr/share/nginx/html/www/index.html” y cambiamos el mensaje de bienvenida para cada servidor:

```
<html>
<head>
<title>Bienvenido a nginx!</title>
</head>
<body>
<center><h1>Bienvenido a nginx! (web1) </h1></center>
</body>
</html>
```



6. Funcionamiento

Ahora, vamos a comprobar el funcionamiento accediendo desde el navegador web a la dirección: <http://trabajoswap.com>.

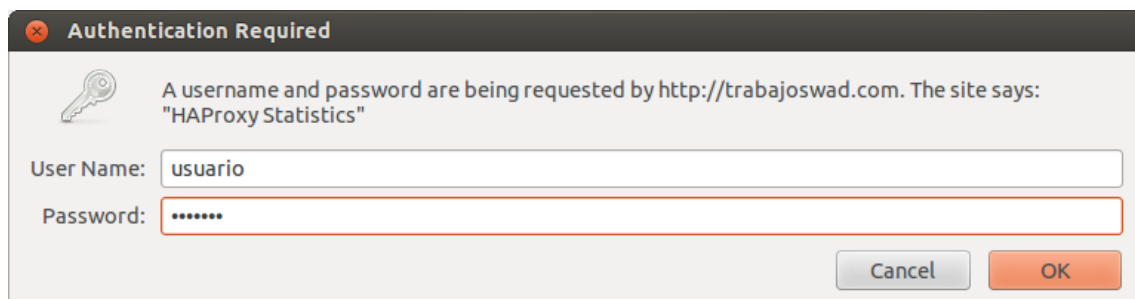


Si recargamos la página nos mostrará la página del servidor “web2” y después el “web3” y volvería a “web1” etc... Esto es debido a que en la configuración de haproxy pusimos “balance roundrobin”.

Podemos ver las estadísticas del clúster a través de la siguiente dirección:

<http://trabajoswap.com/haproxy?stats>

Nos pedirá usuario y contraseña:



Y una vez introducido usuario y contraseña vemos la página con las estadísticas:

Bienvenido a nginx! Statistics Report for HAProxy

trabajoswad.com/haproxy?stats

HAProxy version 1.4.24, released 2013/06/17

Statistics Report for pid 1004

> General process information

pid = 1004 (process #1, nbproc = 1)
 uptime = 0d 0h16m24s
 system limits: memmax = unlimited; ulimit-n = 4014
 maxsock = 4014; maxconn = 2000; maxpipes = 0
 current conns = 1; current pipes = 0/0
 Running tasks: 1/4

active UP
 active UP, going down
 active DOWN, going up
 active or backup DOWN
 active or backup DOWN for maintenance (MAINT)
 Note: UP with load-balancing disabled is reported as "NOLB".

backup UP
 backup UP, going down
 backup DOWN, going up
 not checked

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

External resources:
[Primary site](#)
[Updates \(v1.4\)](#)
[Online manual](#)

web-cluster														Server																	
	Queue			Session rate			Sessions				Bytes		Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrt							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In													Out	Req	Resp	Req	Conn	Resp	Retr
Frontend																OPEN															
web1	0	0	-	0	2	-	0	1	-	4	4	1 335	1 393	0	0	0	0	0	0	0	0	0	16m4s UP	L4OK in 1ms	1	Y	-	0	1	18s	-
web2	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	1m20s UP	L4OK in 1ms	1	Y	-	0	1	15m1s	-	
web3	0	0	-	0	1	-	0	1	-	3	3	1 012	1 604	0	0	0	0	0	0	0	0	15m53s UP	L4OK in 1ms	1	Y	-	0	1	28s	-	
Backend	0	0	-	0	3	-	0	1	-	2 000	7	7	3 002	12 823	0	0	0	0	0	0	0	0	16m4s UP		3	3	0		1	17s	-

Por ejemplo si apagamos el servidor web2, en esta página podemos ver que está apagado:

Bienvenido a nginx! Statistics Report for HAProxy

trabajoswad.com/haproxy?stats

HAProxy version 1.4.24, released 2013/06/17

Statistics Report for pid 1004

> General process information

pid = 1004 (process #1, nbproc = 1)
 uptime = 0d 0h17m14s
 system limits: memmax = unlimited; ulimit-n = 4014
 maxsock = 4014; maxconn = 2000; maxpipes = 0
 current conns = 1; current pipes = 0/0
 Running tasks: 1/4

active UP
 active UP, going down
 active DOWN, going up
 active or backup DOWN
 active or backup DOWN for maintenance (MAINT)
 Note: UP with load-balancing disabled is reported as "NOLB".

backup UP
 backup UP, going down
 backup DOWN, going up
 not checked

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

External resources:
[Primary site](#)
[Updates \(v1.4\)](#)
[Online manual](#)

web-cluster														Server																	
	Queue			Session rate			Sessions				Bytes		Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrt							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In													Out	Req	Resp	Req	Conn	Resp	Retr
Frontend																OPEN															
web1	0	0	-	0	2	-	0	1	-	4	4	1 335	1 393	0	0	0	0	0	0	0	0	16m54s UP	L4OK in 1ms	1	Y	-	0	1	18s	-	
web2	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	2m10s UP 1/3	L4TOUT in 2002ms	1	Y	-	2	1	15m1s	-							
web3	0	0	-	0	1	-	0	1	-	3	3	1 012	1 604	0	0	0	0	0	0	0	0	16m43s UP	L4OK in 1ms	1	Y	-	0	1	28s	-	
Backend	0	0	-	0	3	-	0	1	-	2 000	7	7	3 377	22 385	0	0	0	0	0	0	0	0	16m54s UP		3	3	0		1	17s	-

Bienvenido a nginx! Statistics Report for HAProxy

trabajoswad.com/haproxy?stats

HAProxy version 1.4.24, released 2013/06/17

Statistics Report for pid 1004

> General process information

pid = 1004 (process #1, nbproc = 1)
 uptime = 0d 0h17m35s
 system limits: memmax = unlimited; ulimit-n = 4014
 maxsock = 4014; maxconn = 2000; maxpipes = 0
 current conns = 1; current pipes = 0/0
 Running tasks: 1/4

active UP
 active UP, going down
 active DOWN, going up
 active or backup DOWN
 active or backup DOWN for maintenance (MAINT)
 Note: UP with load-balancing disabled is reported as "NOLB".

backup UP
 backup UP, going down
 backup DOWN, going up
 not checked

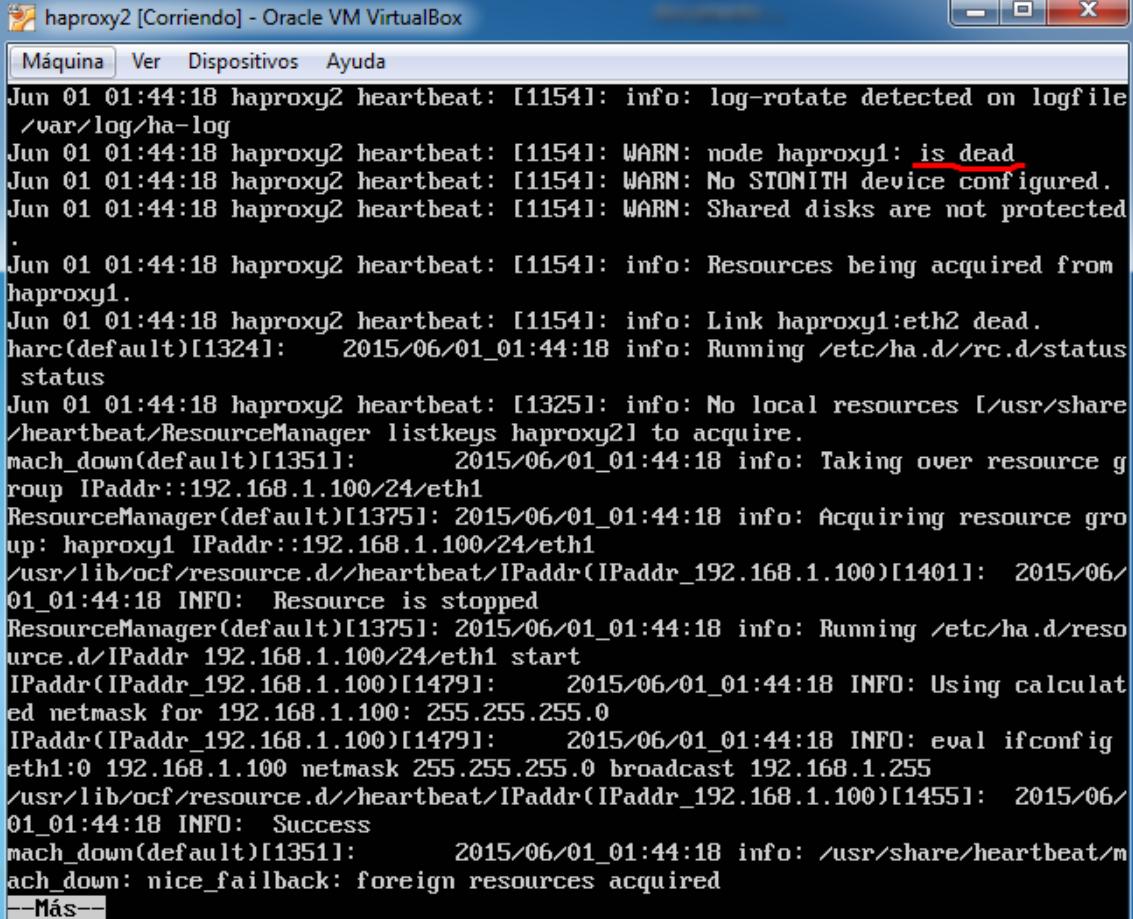
Display option:
[Hide 'DOWN' servers](#)
[Refresh now](#)
[CSV export](#)

External resources:
[Primary site](#)
[Updates \(v1.4\)](#)
[Online manual](#)

web-cluster														Server																	
	Queue			Session rate			Sessions				Bytes		Denied	Errors	Warnings	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrt							
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	In													Out	Req	Resp	Req	Conn	Resp	Retr
Frontend																OPEN															
web1	0	0	-	0	2	-	0	1	-	4	4	1 335	1 393	0	0	0	0	0	0	0	0	17m15s UP	L4OK in 0ms	1	Y	-	0	1	18s	-	
web2	0	0	-	0	0	-	0	0	-	0	0	0	0	0	0	17s DOWN	L4TOUT in 2002ms	1	Y	-	2	2	15m18s	-							
web3	0	0	-	0	1	-	0	1	-	3	3	1 012	1 604	0	0	0	0	0	0	0	0	17m4s UP	L4OK in 0ms	1	Y	-	0	1	28s	-	
Backend	0	0	-	0	3	-	0	1	-	2 000	7	7	3 752	31 960	0	0	0	0	0	0	0	0	17m15s UP		2	2	0		1	17s	-

Veamos ahora el funcionamiento de heartbeat. Para ello vamos a simular un fallo en balanceador maestro (haproxy1) apagando el equipo.

Podemos comprobar que el balanceador esclavo pasa a ser maestro y todo sigue funcionando correctamente. Vamos a comprobarlo mirando el contenido del fichero log:



```
Jun 01 01:44:18 haproxy2 heartbeat: [1154]: info: log-rotate detected on logfile
/var/log/ha-log
Jun 01 01:44:18 haproxy2 heartbeat: [1154]: WARN: node haproxy1: is dead
Jun 01 01:44:18 haproxy2 heartbeat: [1154]: WARN: No STONITH device configured.
Jun 01 01:44:18 haproxy2 heartbeat: [1154]: WARN: Shared disks are not protected
.
Jun 01 01:44:18 haproxy2 heartbeat: [1154]: info: Resources being acquired from
haproxy1.
Jun 01 01:44:18 haproxy2 heartbeat: [1154]: info: Link haproxy1:eth2 dead.
harc(default)[1324]: 2015/06/01_01:44:18 info: Running /etc/ha.d/rc.d/status
status
Jun 01 01:44:18 haproxy2 heartbeat: [1325]: info: No local resources [/usr/share
/heartbeat/ResourceManager listkeys haproxy2] to acquire.
mach_down(default)[1351]: 2015/06/01_01:44:18 info: Taking over resource g
roup IPAddr::192.168.1.100/24/eth1
ResourceManager(default)[1375]: 2015/06/01_01:44:18 info: Acquiring resource gro
up: haproxy1 IPAddr::192.168.1.100/24/eth1
/usr/lib/ocf/resource.d/heartbeat/IPAddr(IPAddr_192.168.1.100)[1401]: 2015/06/
01_01:44:18 INFO: Resource is stopped
ResourceManager(default)[1375]: 2015/06/01_01:44:18 info: Running /etc/ha.d/reso
urce.d/IPAddr 192.168.1.100/24/eth1 start
IPAddr(IPAddr_192.168.1.100)[1479]: 2015/06/01_01:44:18 INFO: Using calculat
ed netmask for 192.168.1.100: 255.255.255.0
IPAddr(IPAddr_192.168.1.100)[1479]: 2015/06/01_01:44:18 INFO: eval ifconfig
eth1:0 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255
/usr/lib/ocf/resource.d/heartbeat/IPAddr(IPAddr_192.168.1.100)[1455]: 2015/06/
01_01:44:18 INFO: Success
mach_down(default)[1351]: 2015/06/01_01:44:18 info: /usr/share/heartbeat/m
ach_down: nice_failback: foreign resources acquired
--Más--
```

Podemos ver todo el proceso desde que detecta que haproxy1 ha caído, hasta que levanta la interface y asigna la ip virtual 192.168.1.100 al balanceador secundario (haproxy2).

Ahora volvemos a encender el balanceador maestro (haproxy1) y vemos de nuevo el fichero log:

```

Máquina Ver Dispositivos Ayuda
Jun 01 01:51:40 haproxy2 heartbeat: [1154]: info: Heartbeat restart on node haproxy1
Jun 01 01:51:40 haproxy2 heartbeat: [1154]: info: Link haproxy1:eth2 up.
Jun 01 01:51:40 haproxy2 heartbeat: [1154]: info: Status update for node haproxy1: status init
Jun 01 01:51:40 haproxy2 heartbeat: [1154]: info: Status update for node haproxy1: status up
harc(default)[1579]: 2015/06/01_01:51:40 info: Running /etc/ha.d/rc.d/status status
harc(default)[1593]: 2015/06/01_01:51:40 info: Running /etc/ha.d/rc.d/status status
Jun 01 01:51:41 haproxy2 heartbeat: [1154]: info: all clients are now paused
Jun 01 01:51:42 haproxy2 heartbeat: [1154]: info: Status update for node haproxy1: status active
harc(default)[1607]: 2015/06/01_01:51:42 info: Running /etc/ha.d/rc.d/status status
Jun 01 01:51:42 haproxy2 heartbeat: [1154]: info: remote resource transition completed.
Jun 01 01:51:42 haproxy2 heartbeat: [1154]: info: haproxy2 wants to go standby [foreign]
Jun 01 01:51:43 haproxy2 heartbeat: [1154]: info: all clients are now resumed
Jun 01 01:51:43 haproxy2 heartbeat: [1154]: info: standby: haproxy1 can take our foreign resources
Jun 01 01:51:43 haproxy2 heartbeat: [1621]: info: give up foreign HA resources (standby).
ResourceManager(default)[1635]: 2015/06/01_01:51:43 info: Releasing resource group: haproxy1 IPAddr::192.168.1.100/24/eth1
ResourceManager(default)[1635]: 2015/06/01_01:51:43 info: Running /etc/ha.d/resource.d/IPAddr 192.168.1.100/24/eth1 stop
--Más--

```

```

Jun 01 01:51:43 haproxy2 heartbeat: [1621]: info: give up foreign HA resources (standby).
ResourceManager(default)[1635]: 2015/06/01_01:51:43 info: Releasing resource group: haproxy1 IPAddr::192.168.1.100/24/eth1
ResourceManager(default)[1635]: 2015/06/01_01:51:43 info: Running /etc/ha.d/resource.d/IPAddr 192.168.1.100/24/eth1 stop
IPAddr(IPAddr_192.168.1.100)[1693]: 2015/06/01_01:51:43 INFO: ifconfig eth1: 0 down
/usr/lib/ocf/resource.d/heartbeat/IPAddr(IPAddr_192.168.1.100)[1669]: 2015/06/01_01:51:43 INFO: Success
Jun 01 01:51:43 haproxy2 heartbeat: [1621]: info: foreign HA resource release completed (standby).
Jun 01 01:51:43 haproxy2 heartbeat: [1154]: info: Local standby process completed [foreign].
Jun 01 01:51:44 haproxy2 heartbeat: [1154]: WARN: 1 lost packet(s) for [haproxy1] [12:14]
Jun 01 01:51:44 haproxy2 heartbeat: [1154]: info: remote resource transition completed.
Jun 01 01:51:44 haproxy2 heartbeat: [1154]: info: No pkts missing from haproxy1!
Jun 01 01:51:44 haproxy2 heartbeat: [1154]: info: Other node completed standby takeover of foreign resources.
root@haproxy2:/var/log#

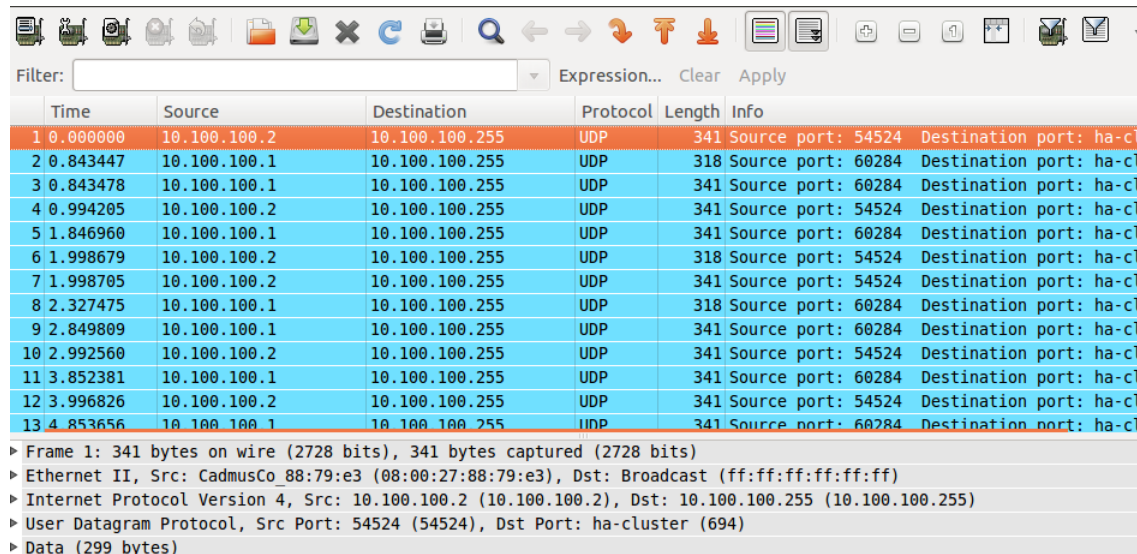
```

Y vemos como haproxy1 vuelve a tomar el control y haproxy2 pasa a standby.

7. Wireshark

Para comprobar si el servidor está activo se envían paquetes UDP broadcast a través de la interface configurada con la directiva bcast en el archivo “/etc/ha.d/ha.cf” manteniendo un registro de la cantidad de tiempo transcurrido desde que cada ordenador conocido envió su último mensaje.

Veamos una captura de wireshark con los dos balanceadores en funcionamiento:



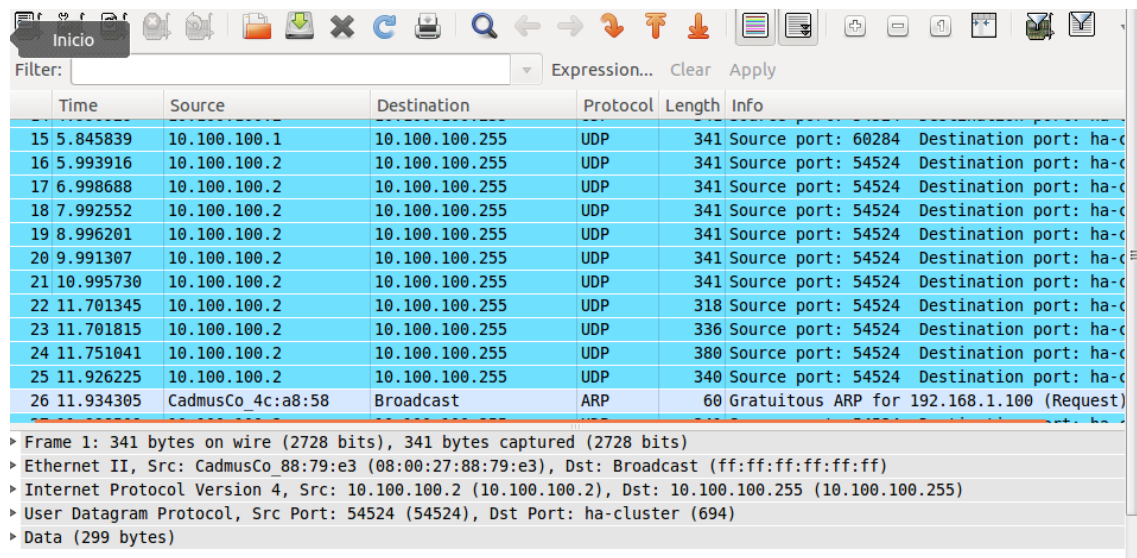
Filter: Expression... Clear Apply

Time	Source	Destination	Protocol	Length	Info
1 0.000000	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
2 0.843447	10.100.100.1	10.100.100.255	UDP	318	Source port: 60284 Destination port: ha-cl
3 0.843478	10.100.100.1	10.100.100.255	UDP	341	Source port: 60284 Destination port: ha-cl
4 0.994205	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
5 1.846960	10.100.100.1	10.100.100.255	UDP	341	Source port: 60284 Destination port: ha-cl
6 1.998679	10.100.100.2	10.100.100.255	UDP	318	Source port: 54524 Destination port: ha-cl
7 1.998705	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
8 2.327475	10.100.100.1	10.100.100.255	UDP	318	Source port: 60284 Destination port: ha-cl
9 2.849809	10.100.100.1	10.100.100.255	UDP	341	Source port: 60284 Destination port: ha-cl
10 2.992560	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
11 3.852381	10.100.100.1	10.100.100.255	UDP	341	Source port: 60284 Destination port: ha-cl
12 3.996826	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
13 4.853656	10.100.100.1	10.100.100.255	UDP	341	Source port: 60284 Destination port: ha-cl

▶ Frame 1: 341 bytes on wire (2728 bits), 341 bytes captured (2728 bits)
 ▶ Ethernet II, Src: CadmusCo 88:79:e3 (08:00:27:88:79:e3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ▶ Internet Protocol Version 4, Src: 10.100.100.2 (10.100.100.2), Dst: 10.100.100.255 (10.100.100.255)
 ▶ User Datagram Protocol, Src Port: 54524 (54524), Dst Port: ha-cluster (694)
 ▶ Data (299 bytes)

Podemos observar como ambos equipos están enviando paquetes UDP broadcast.

Ahora vamos a apagar el balanceador maestro (10.100.100.1) y vemos cómo deja de enviar paquetes UDP.

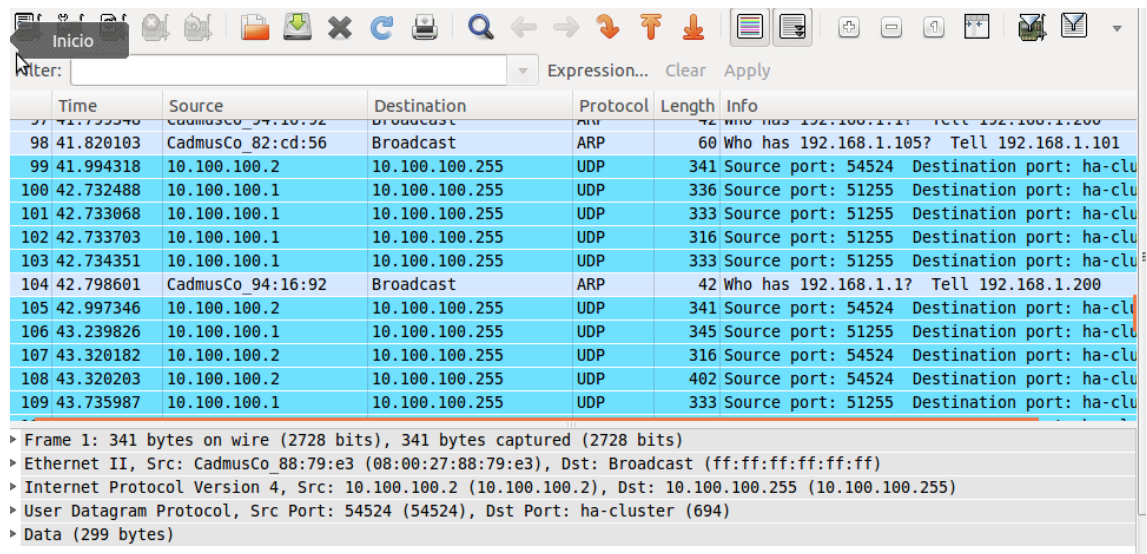


Filter: Expression... Clear Apply

Time	Source	Destination	Protocol	Length	Info
15 5.845839	10.100.100.1	10.100.100.255	UDP	341	Source port: 60284 Destination port: ha-cl
16 5.993916	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
17 6.998688	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
18 7.992552	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
19 8.996201	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
20 9.991307	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
21 10.995730	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-cl
22 11.701345	10.100.100.2	10.100.100.255	UDP	318	Source port: 54524 Destination port: ha-cl
23 11.701815	10.100.100.2	10.100.100.255	UDP	336	Source port: 54524 Destination port: ha-cl
24 11.751041	10.100.100.2	10.100.100.255	UDP	380	Source port: 54524 Destination port: ha-cl
25 11.926225	10.100.100.2	10.100.100.255	UDP	340	Source port: 54524 Destination port: ha-cl
26 11.934305	CadmusCo_4c:a8:58	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.100 (Request)

▶ Frame 1: 341 bytes on wire (2728 bits), 341 bytes captured (2728 bits)
 ▶ Ethernet II, Src: CadmusCo 88:79:e3 (08:00:27:88:79:e3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 ▶ Internet Protocol Version 4, Src: 10.100.100.2 (10.100.100.2), Dst: 10.100.100.255 (10.100.100.255)
 ▶ User Datagram Protocol, Src Port: 54524 (54524), Dst Port: ha-cluster (694)
 ▶ Data (299 bytes)

Lo volvemos a encender y vemos que vuelve a enviar paquetes:



Time	Source	Destination	Protocol	Length	Info
98	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-clu
99	10.100.100.1	10.100.100.255	UDP	336	Source port: 51255 Destination port: ha-clu
100	10.100.100.1	10.100.100.255	UDP	333	Source port: 51255 Destination port: ha-clu
101	10.100.100.1	10.100.100.255	UDP	316	Source port: 51255 Destination port: ha-clu
102	10.100.100.1	10.100.100.255	UDP	333	Source port: 51255 Destination port: ha-clu
103	10.100.100.1	10.100.100.255	UDP	333	Source port: 51255 Destination port: ha-clu
104	10.100.100.2	10.100.100.255	UDP	341	Source port: 54524 Destination port: ha-clu
105	10.100.100.1	10.100.100.255	UDP	345	Source port: 51255 Destination port: ha-clu
106	10.100.100.1	10.100.100.255	UDP	316	Source port: 54524 Destination port: ha-clu
107	10.100.100.2	10.100.100.255	UDP	402	Source port: 54524 Destination port: ha-clu
108	10.100.100.2	10.100.100.255	UDP	333	Source port: 51255 Destination port: ha-clu
109	10.100.100.1	10.100.100.255	UDP	333	Source port: 51255 Destination port: ha-clu

► Frame 1: 341 bytes on wire (2728 bits), 341 bytes captured (2728 bits)

► Ethernet II, Src: CadmusCo 88:79:e3 (08:00:27:88:79:e3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

► Internet Protocol Version 4, Src: 10.100.100.2 (10.100.100.2), Dst: 10.100.100.255 (10.100.100.255)

► User Datagram Protocol, Src Port: 54524 (54524), Dst Port: ha-cluster (694)

► Data (299 bytes)

8. Referencias

- [1]<http://blog.bobbyallen.me/2013/01/21/installing-a-high-availability-web-server-cluster-on-ubuntu-12-04-lts-using-haproxy-heartbeat-and-nginx/>
- [2]http://linux-ha.org/wiki/Ha.cf#node_directive
- [3]<https://www.howtoforge.com/high-availability-load-balancer-haproxy-heartbeat-debian-etch>
- [4] <http://www.htforos.com/webmasters/balanceador-carga-alta-disponibilidad-con-haproxy-t3744.html>
- [5]<http://wiki.nginx.org/Main>
- [6]<http://www.haproxy.org/>
- [7]<https://www.wireshark.org/>