



CLÚSTER DE ALTA DISPONIBILIDAD CON HEARTBEAT

TOMÁS SÁNCHEZ, DEILER MOSQUERA, RAFAEL VEJARANO



UTP 2021

© Clúster De Alta Disponibilidad Con Heartbeat, por Tomás Sánchez, Deiler Mosquera, Rafael Vejarano. Universidad Tecnológica de Panamá (UTP) Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-Compartir Igual 4.0 Internacional. Para ver esta licencia: <https://creativecommons.org/licenses/by-nc-sa/4.0>

Contenido

Introducción	ii
1. Requerimientos.....	1
2. Implementación	1
a. Instalación de Servidores	1
b. Configuración	3
c. Funcionamiento.....	11
Conclusiones	15
Referencias	15

Introducción

Un clúster de alta disponibilidad está conformado por dos máquinas de las mismas características, que están totalmente sincronizada y que permiten que el sistema siga funcionando con normalidad. La alta disponibilidad permite que un servicio funcione correctamente ante un fallo de software o hardware.

Esta capacidad de los clústeres de restablecer en pocos segundos un servicio, manteniendo la integridad de los datos, permite que en muchos casos los usuarios no tengan por qué notar que se ha producido un problema, cuando una avería de este tipo, en un sistema sin clúster, podría dejarles sin servicio durante horas.

En esta guía veremos cómo construir en un ambiente experimental, un clúster de alta disponibilidad mediante la implementación de máquinas virtuales y el servicio de clúster **Heartbeat**.

1. Requerimientos

Para el Desarrollo de este proyecto se van a requerir los siguientes componentes:

- Dos servidores virtuales con Ubuntu Server 18.04 LTS (<http://old-releases.ubuntu.com/releases/18.04.4/ubuntu-18.04-live-server-amd64.iso>).
- Tarjetas de red en modo puente.
- Putty.
- Heartbeat.

En este desarrollo se estará empleando Proxmox para construir los servidores virtuales. Proxmox, es un entorno de virtualización de servidores de código abierto basado en Linux Debian. En esta guía no se estará desarrollando la instalación de Proxmox por lo que se recomienda ir a la documentación oficial en el siguiente enlace: <https://www.proxmox.com/en/>. Sin embargo, puede emplearse VirtualBox o VMWare para la creación de las máquinas virtuales.

2. Implementación

a. Instalación de Servidores

Se inicia sesión en el servidor de virtualización **Proxmox** con las credenciales de acceso dadas durante el proceso de instalación (ver figura 1).

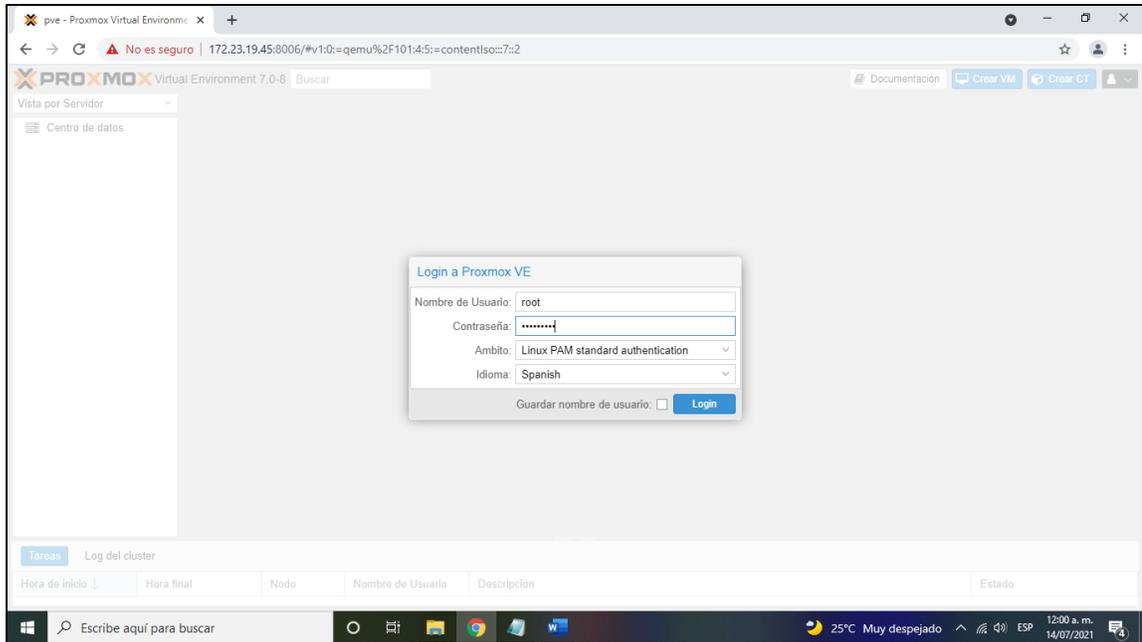


Figura. 1 . Interfaz de inicio de sesión en Proxmox.

Se han de crear dos máquinas virtuales con nombre los nombres CMaster y CSlave (ver figura 2).

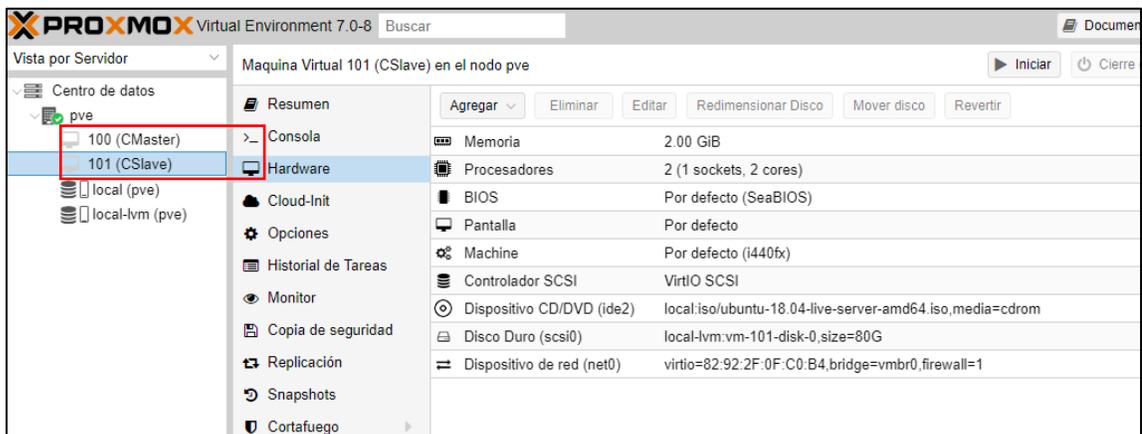


Figura. 2. Creación de máquinas virtuales

Se estará empleando la versión estable del servidor de Ubuntu 18.04.4, la cual podrá descargar desde la siguiente dirección:

[http://old-releases.ubuntu.com/releases/18.04.4/ubuntu-18.04-live-server-
amd64.iso](http://old-releases.ubuntu.com/releases/18.04.4/ubuntu-18.04-live-server-amd64.iso)

Ambos servidores se han construido con las siguientes especificaciones generales:

Tabla 1. Detalles de Hardware y Software de VM.

Terminal	Detalles
CMaster	RAM: 2GiB
CSlave	Procesadores: 2 (1 socket, 2 cores)
	Disco duro: 80 GiB
	Red: Puente
	Ubuntu 18.04.5 LTS

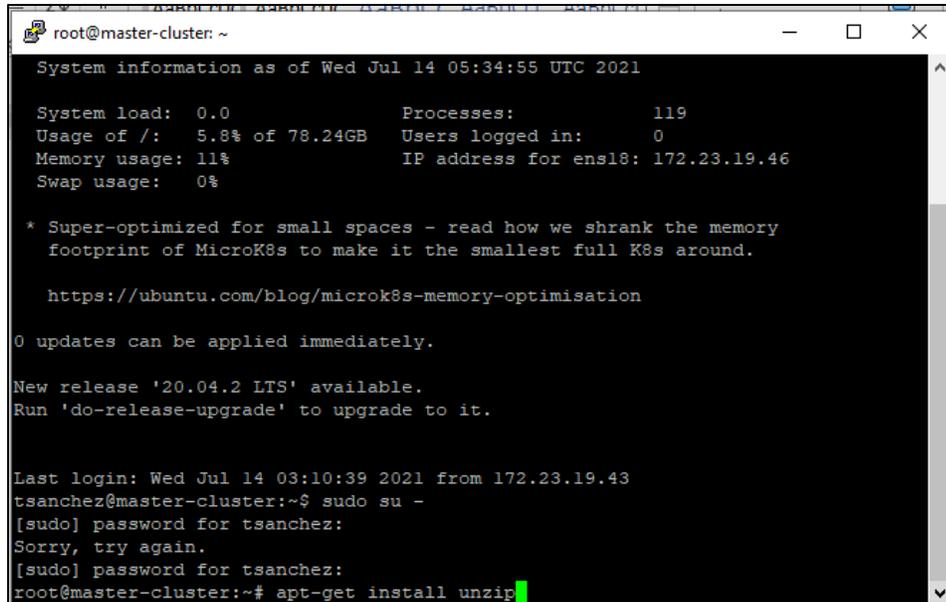
b. Configuración

Una vez se han instalado las máquinas virtuales con Ubuntu Server 18.04 LTS, se procede a actualizar los repositorios en cada una de las máquinas desde la línea de comandos mediante la instrucción:

```
sudo apt-get update
```

Seguidamente se instala UNZIP a través del comando (ver figura No.3):

```
apt-get install unzip.
```



```
root@master-cluster: ~  
System information as of Wed Jul 14 05:34:55 UTC 2021  
  
System load: 0.0          Processes:          119  
Usage of /: 5.8% of 78.24GB Users logged in: 0  
Memory usage: 11%       IP address for ens18: 172.23.19.46  
Swap usage: 0%  
  
* Super-optimized for small spaces - read how we shrank the memory  
  footprint of MicroK8s to make it the smallest full K8s around.  
  
  https://ubuntu.com/blog/microk8s-memory-optimisation  
  
0 updates can be applied immediately.  
  
New release '20.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Wed Jul 14 03:10:39 2021 from 172.23.19.43  
tsanchez@master-cluster:~$ sudo su -  
[sudo] password for tsanchez:  
Sorry, try again.  
[sudo] password for tsanchez:  
root@master-cluster:~# apt-get install unzip
```

Figura. 3. Instalación de UNZIP

Hay que recordar que debe iniciar sesión con el usuario root para que se permita los cambios. Pero antes debe iniciar Putty para acceder a estos servidores. Inicialmente los servidores poseerán direcciones DHCP proporcionadas por su router. Se recomienda utilizar direcciones estáticas para garantizar el acceso a los servidores. Para este caso se han configurado los equipos con las direcciones, con dirección IP 172.23.46, 172.23.19.47, puertos 4243 y 4242 respectivamente. El acceso para la configuración se hará mediante putty (ver figura No. 4).

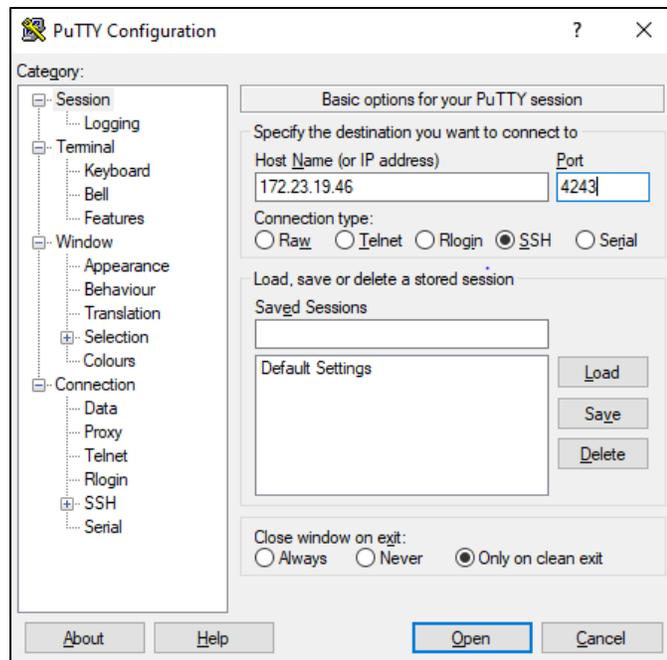


Figura. 4. Inicio de sesión remota al servidor CMaster.

Lo siguiente es descargar e instalar el paquete Heartbeat en cada uno de los servidores virtuales (CMaster y CSlave) a través del comando:

apt-get install heartbeat.

```
tsanchez@master-cluster: ~  
tsanchez@172.23.19.46's password:  
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-147-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Thu Jul 15 11:56:27 UTC 2021  
  
System load:  1.71                Processes:            121  
Usage of /:   5.8% of 78.24GB     Users logged in:     0  
Memory usage: 12%                IP address for ens18: 172.23.19.46  
Swap usage:   0%  
  
* Super-optimized for small spaces - read how we shrank the memory  
  footprint of MicroK8s to make it the smallest full K8s around.  
  
  https://ubuntu.com/blog/microk8s-memory-optimisation  
  
0 updates can be applied immediately.  
  
Last login: Wed Jul 14 05:34:58 2021 from 172.23.19.43  
tsanchez@master-cluster:~$ apt-get install heartbeat
```

Figura. 5. Ejecutando comando para instalar el paquete heartbeat en ambos servidores.

Una vez descargados en ambos servidores, Heartbeat, buscamos el paquete en la siguiente ruta:

usr/share/doc/heartbeat

En esta carpeta se encontrarán los archivos comprimidos: ha.cf.gz, haresources.gz y authkeys. Se procederá a descomprimir los archivos en cada uno de los servidores con el comando:

gzip -d ha.cf.gz

gzip -d haresources.gz

A continuación, se deben copiar los archivos ha.cf, haresources y authkeys a la carpeta /etc/ha.d.

cp ha.cf /etc/ha.d

cp haresources /etc/ha.d

cp authkeys /etc/ha.d

Se debe ubicar en el directorio `/etc/ha.d`. Ahora, se procederá a la edición de los archivos `ha.cf`, `haresources` y `authkeys`.

Para la edición de estos archivos se utilizará el editor de Linux nano.

nano ha.cf

En el archivo `ha.cf`, se deben “descomentar” las siguientes líneas:

- `keepalive 2`, sirve para verificar la comunicación entre los servidores.
- `deadtime 30` se debe cambiar a `deadtime 5`, para iniciar un servicio.
- `udpport 694`.
- `bcast eth0`, mediante el comando `ifconfig`, verifique el nombre de su tarjeta de red. Este nombre es distinto dependiendo del software de máquina virtual que esté utilizando. Para este caso es `ens18`.
- `mcast eth0 225.0.0.1 694 1 0`, cambiar al que corresponda en su equipo.
- `ucast eth0 192.168.1.2`, Si se trata del archivo de configuración del master, esta dirección IP debe apuntar al IP del esclavo, y si es la configuración del esclavo, esta debe dirigirse a la IP del maestro. En nuestro caso sería, **`ucast ens18 172.23.19.47`**. Es importante aclarar que luego de “descomentar” se debe adicionar abajo la línea **`upd ens18`**.
- `auto_failback on`, validar que este esté en “on”.
- `node ken3`, `ken3` sería el nombre del nodo maestro. En nuestro caso, `node master-cluster`.
- `node kathy`, cambiar al nombre del nodo esclavo. En nuestro caso, `node slave-cluster`.

Nota: Estos mismos pasos, en la configuración del servidor maestro se realizan en el servidor esclavo con la diferencia en que en el apartado o línea “`ucast ens18...`” se coloca la dirección IP del servidor maestro (ver figura No. 6).

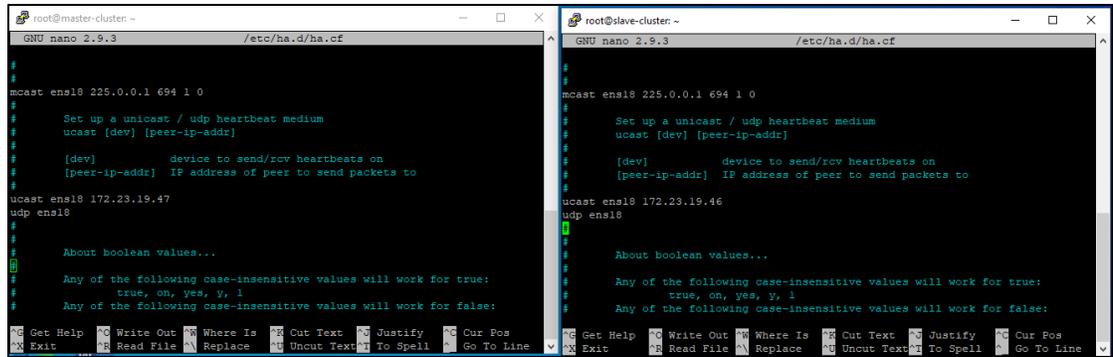


Figura. 6. Configuración de ucast de master y esclavo

El segundo archivo que se debe modificar es “haresources”. Al editarlo, se debe ir hasta el final de la página y agregar el nombre del equipo o nodo maestro, la dirección IP virtual que va a ser de referencia para uso del servicio por parte de los clientes y el servicio utilizado. Para cada equipo, nodo maestro y nodo esclavo se asignará la ip 172.23.19.48, la cual corresponderá a la IP virtual. La línea quedará escrita de la siguiente manera (ver figura No. 7):

Del lado del servidor maestro:

master-cluster 172.23.19.48 apache2

Del lado del servidor esclavo:

slave-cluster 172.23.19.48 apache2

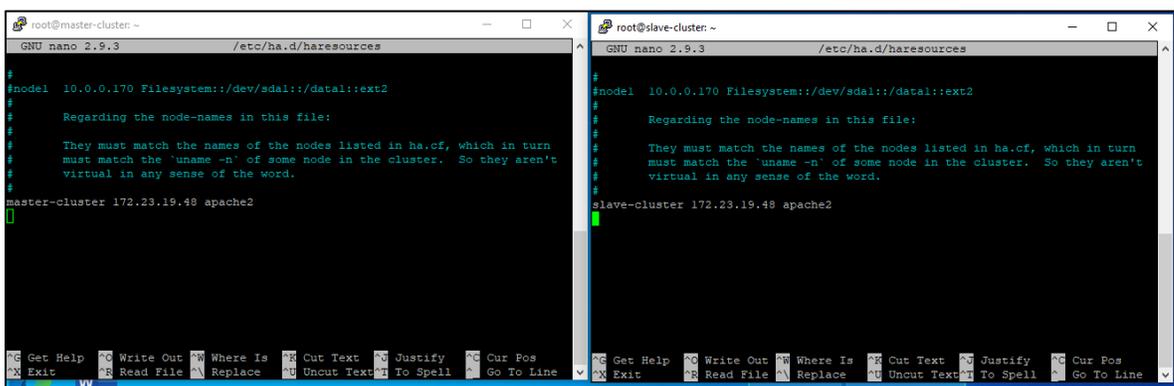


Figura. 7. Configuración de archivo haresources

El tercer y último archivo para modificar es “authkeys”, al cual se deberá agregar las líneas siguientes al final: “auth 1 y 1 crc”. Aplicarlo en cada servidor, como se muestra en la figura 14.

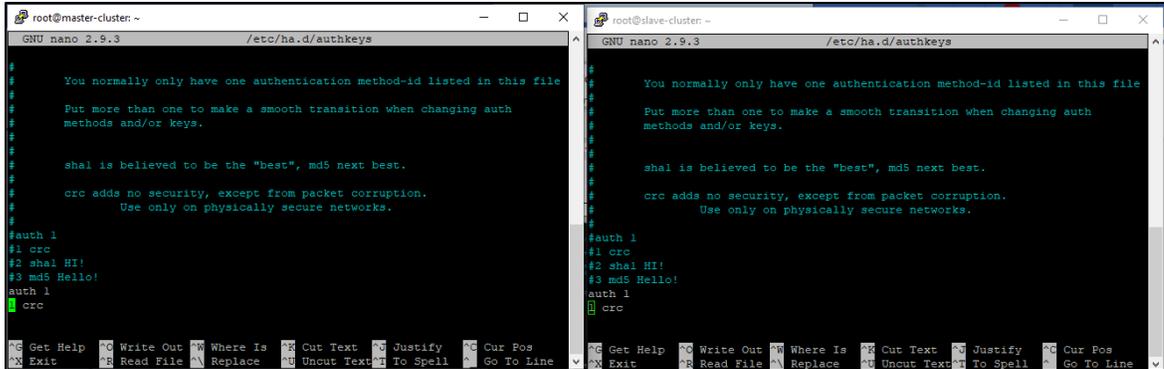


Figura. 8. Configuración de archivo authkeys

Nota: es importante colocarle los permisos de control total al archivo “authkeys” a sólo administradores ya que de forma predeterminada no viene establecidos. Este se valida a través del comando “ls -l”.

Para cambiarlo recuerde usar el comando (ver figura No. 9):

chmod 600 authkeys

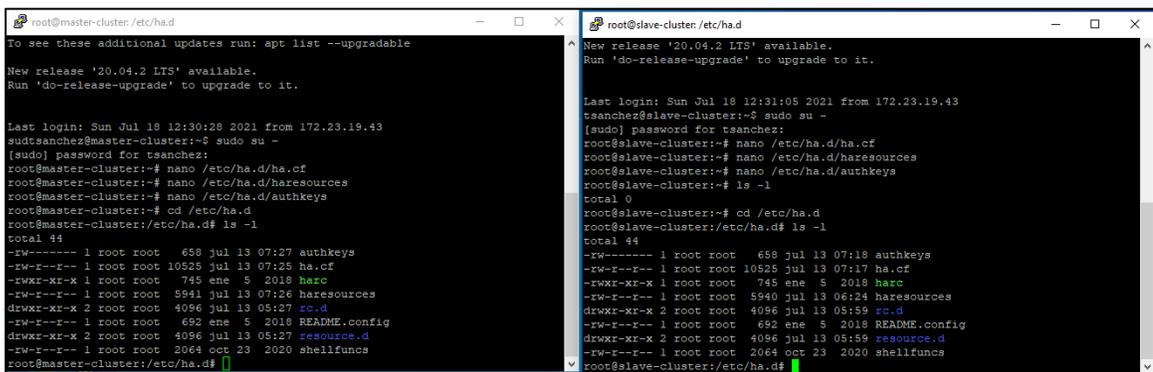


Figura. 9. Aplicar permisos de control total a sólo administrador.

Una vez finalizada las modificaciones en los archivos se debe reiniciar el servicio “heartbeat”. Cambiase al directorio:

cd /etc/init.d/

Ejecute el comando (ver figura No. 10):

./heartbeat restart

```
root@master-cluster:/etc/init.d
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Jul 18 12:30:28 2021 from 172.23.19.43
tsanchez@master-cluster:~$ sudo su -
[sudo] password for tsanchez:
root@master-cluster:~# nano /etc/ha.d/ha.cf
root@master-cluster:~# nano /etc/ha.d/haresources
root@master-cluster:~# nano /etc/ha.d/hakeys
root@master-cluster:~# cd /etc/ha.d
root@master-cluster:/etc/ha.d# ls -l
total 44
-rw-r--r-- 1 root root 658 jul 13 07:27 authkeys
-rw-r--r-- 1 root root 10525 jul 13 07:25 ha.cf
-rwxr-xr-x 1 root root 745 ene 5 2018 harc
-rw-r--r-- 1 root root 5941 jul 13 07:26 haresources
drwxr-xr-x 2 root root 4096 jul 13 05:27 resource.d
-rw-r--r-- 1 root root 692 ene 5 2018 README.config
drwxr-xr-x 2 root root 4096 jul 13 05:27 resource.d
-rw-r--r-- 1 root root 2064 oct 23 2020 shellfuncs
root@master-cluster:/etc/ha.d# cd /etc/init.d
root@master-cluster:/etc/init.d# ./heartbeat restart
[ OK ] Restarting heartbeat (via systemctl): heartbeat.service.
root@master-cluster:/etc/init.d#
```

```
root@slave-cluster:/etc/init.d
Last login: Sun Jul 18 12:31:05 2021 from 172.23.19.43
tsanchez@slave-cluster:~$ sudo su -
[sudo] password for tsanchez:
root@slave-cluster:~# nano /etc/ha.d/ha.cf
root@slave-cluster:~# nano /etc/ha.d/haresources
root@slave-cluster:~# nano /etc/ha.d/hakeys
root@slave-cluster:~# ls -l
total 0
root@slave-cluster:~# cd /etc/ha.d
root@slave-cluster:/etc/ha.d# ls -l
total 44
-rw-r--r-- 1 root root 658 jul 13 07:18 authkeys
-rw-r--r-- 1 root root 10525 jul 13 07:17 ha.cf
-rwxr-xr-x 1 root root 745 ene 5 2018 harc
-rw-r--r-- 1 root root 5940 jul 13 06:24 haresources
drwxr-xr-x 2 root root 4096 jul 13 05:59 resource.d
-rw-r--r-- 1 root root 692 ene 5 2018 README.config
drwxr-xr-x 2 root root 4096 jul 13 05:59 resource.d
-rw-r--r-- 1 root root 2064 oct 23 2020 shellfuncs
root@slave-cluster:/etc/ha.d# cd /etc/init.d
root@slave-cluster:/etc/init.d# ./heartbeat restart
[ OK ] Restarting heartbeat (via systemctl): heartbeat.service.
root@slave-cluster:/etc/init.d#
```

Figura. 10. Reiniciando el servicio heartbeat.

Para verificar que todo se encuentra funcionando, ejecute el comando:

./heartbeat status

Este comando le mostrará todo lo concerniente al servicio y detalles de errores si existiese (ver figura No. 11).

```
root@master-cluster:/etc/init.d
[ OK ] Restarting heartbeat (via systemctl): heartbeat.service.
root@master-cluster:/etc/init.d# ./heartbeat status
● heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-07-19 02:26:29 UTC; 3min 40s ago
     Process: 2436 ExecStop=/usr/lib/heartbeat/heartbeat -k (code=exited, status=0/SUCCESS)
   Main PID: 2558 (heartbeat)
     Tasks: 10 (limit: 2314)
   CGroup: /system.slice/heartbeat.service
           └─2558 heartbeat: master control process
             └─2589 heartbeat: FIFO reader
               └─2591 heartbeat: write: bcast ens18
                 └─2592 heartbeat: read: bcast ens18
                   └─2593 heartbeat: write: mcast ens18
                     └─2594 heartbeat: read: mcast ens18
                       └─2595 heartbeat: write: ucast ens18
                         └─2596 heartbeat: read: ucast ens18
                           └─2597 heartbeat: write: bcast ens18
                             └─2598 heartbeat: read: bcast ens18

jul 19 02:26:41 master-cluster heartbeat[2558]: [2558]: info: remote resource.ed
jul 19 02:26:41 master-cluster heartbeat[2558]: [2558]: info: master-cluster.gn
jul 19 02:26:41 master-cluster heartbeat[2558]: [2558]: info: standby: slave_...
jul 19 02:26:41 master-cluster heartbeat[3152]: [3152]: info: give up foreign_...
jul 19 02:26:41 master-cluster heartbeat[3152]: [3152]: info: foreign HA res_...
jul 19 02:26:41 master-cluster heartbeat[2558]: [2558]: info: Local standby_...
jul 19 02:26:43 master-cluster heartbeat[2558]: [2558]: WARN: 1 lost packet(...
jul 19 02:26:43 master-cluster heartbeat[2558]: [2558]: info: remote resource.ed
jul 19 02:26:43 master-cluster heartbeat[2558]: [2558]: info: No pkts missin...
jul 19 02:26:43 master-cluster heartbeat[2558]: [2558]: info: Other node com...
Hint: Some lines were ellipsized, use -l to show in full.
root@master-cluster:/etc/init.d#
```

```
root@slave-cluster:/etc/init.d
[ OK ] Restarting heartbeat (via systemctl): heartbeat.service.
root@slave-cluster:/etc/init.d# ./heartbeat status
● heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-07-19 02:26:38 UTC; 4min 3s ago
     Process: 2578 ExecStop=/usr/lib/heartbeat/heartbeat -k (code=exited, status=0/SUCCESS)
   Main PID: 2711 (heartbeat)
     Tasks: 10 (limit: 2314)
   CGroup: /system.slice/heartbeat.service
           └─2711 heartbeat: master control process
             └─2740 heartbeat: FIFO reader
               └─2742 heartbeat: write: bcast ens18
                 └─2743 heartbeat: read: bcast ens18
                   └─2745 heartbeat: write: mcast ens18
                     └─2746 heartbeat: read: mcast ens18
                       └─2747 heartbeat: write: ucast ens18
                         └─2748 heartbeat: read: ucast ens18
                           └─2749 heartbeat: write: bcast ens18
                             └─2750 heartbeat: read: bcast ens18

jul 19 02:26:41 slave-cluster heartbeat[2711]: [2711]: info: remote resource.ed
jul 19 02:26:41 slave-cluster heartbeat[2711]: [2711]: info: standby: slave_...
jul 19 02:26:41 slave-cluster heartbeat[2711]: [2711]: info: Local Resource_...
jul 19 02:26:41 slave-cluster heartbeat[2711]: [2711]: info: master-cluster_...
jul 19 02:26:42 slave-cluster heartbeat[2711]: [2711]: info: standby: acquir...
jul 19 02:26:42 slave-cluster heartbeat[2711]: [2711]: info: acquire local H...
jul 19 02:26:42 slave-cluster heartbeat[2711]: [2711]: info: local HA resour...
jul 19 02:26:42 slave-cluster heartbeat[2711]: [2711]: info: Standby resourc...
jul 19 02:26:42 slave-cluster heartbeat[2711]: [2711]: info: Initial resourc...
jul 19 02:26:43 slave-cluster heartbeat[2711]: [2711]: info: remote resource...
Hint: Some lines were ellipsized, use -l to show in full.
root@slave-cluster:/etc/init.d#
```

Figura. 11. Validar estatus de servicios heartbeat.

Nuestro clúster de alta disponibilidad es un servicio web, por lo que se procederá a instalar el servidor web en los equipos máster y esclavo. Para instalar **apache2** en los equipos, se ejecutará la siguiente instrucción:

sudo apt-get install apache2

c. Funcionamiento

Para validar el funcionamiento del clúster se editará la página del “**index.html**” de apache2 para ver como el clúster alta disponibilidad funciona. Para eso nos colocamos en la ruta “**/var/www/html**” y modificamos la cabecera que aparece en la figura No. 12 (MASTER-CLUSTER ON y SLAVE-CLUSTER ON”), de forma tal que se pueda identificar que equipo esta brindando el servicio por medio de la IP virtual que introducirá el cliente en un navegador.

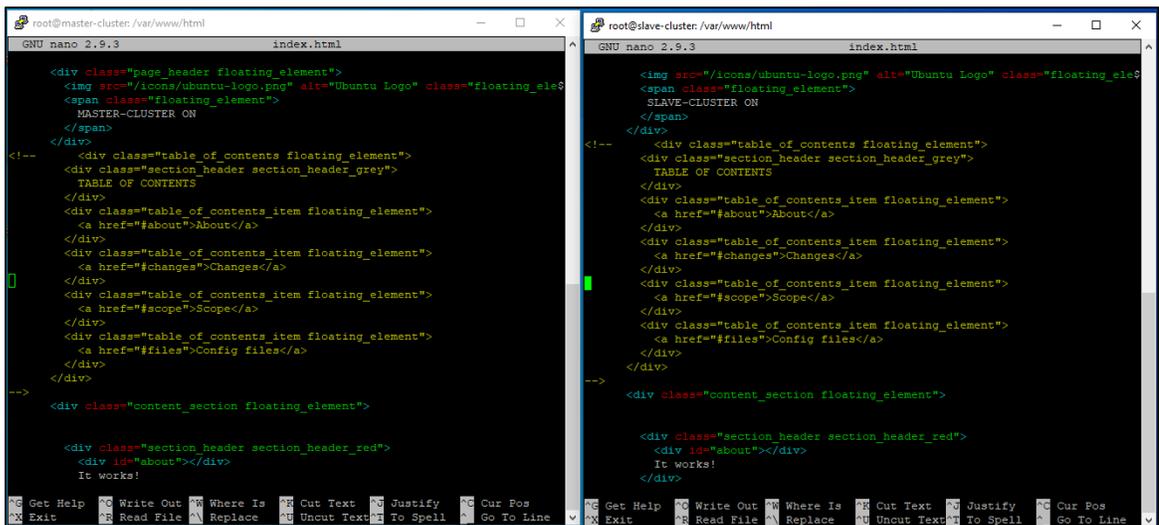


Figura. 12. Colocando identificador a maestro y esclavo en index.html.

Una vez realizado los cambios en ambos archivos de apache2, se debe reiniciar el servicio con el comando **apache2 restart** (ver figura No. 13).

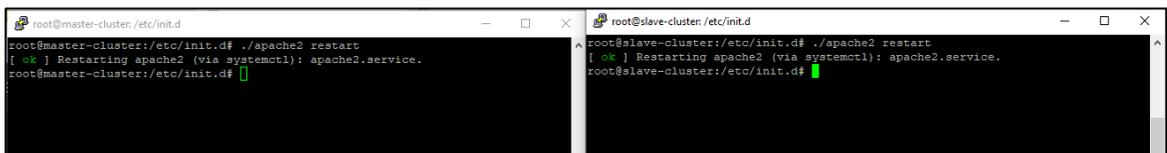


Figura. 13. Iniciando servicio de apache2.

Desde una maquina conectada a la misma red de los servidores, se abre el navegador y en la barra de direcciones se colocará la dirección de cada servidor para comprobar su disponibilidad en forma independiente (ver figuras 14 y 15).

Las direcciones IP se han encerrado en un recuadro en rojo para identificar el equipo que está brindando servicio.



Figura. 14. Consulta al servicio apache2 del nodo maestro.

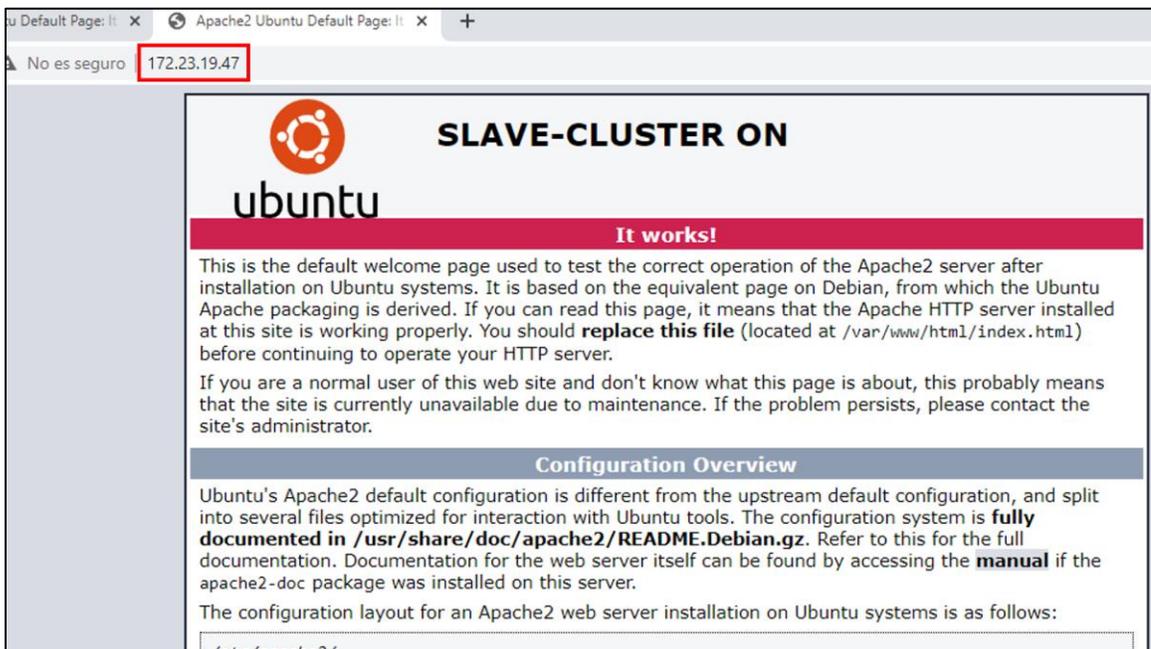


Figura. 15. Consulta al servicio apache2 del nodo esclavo.

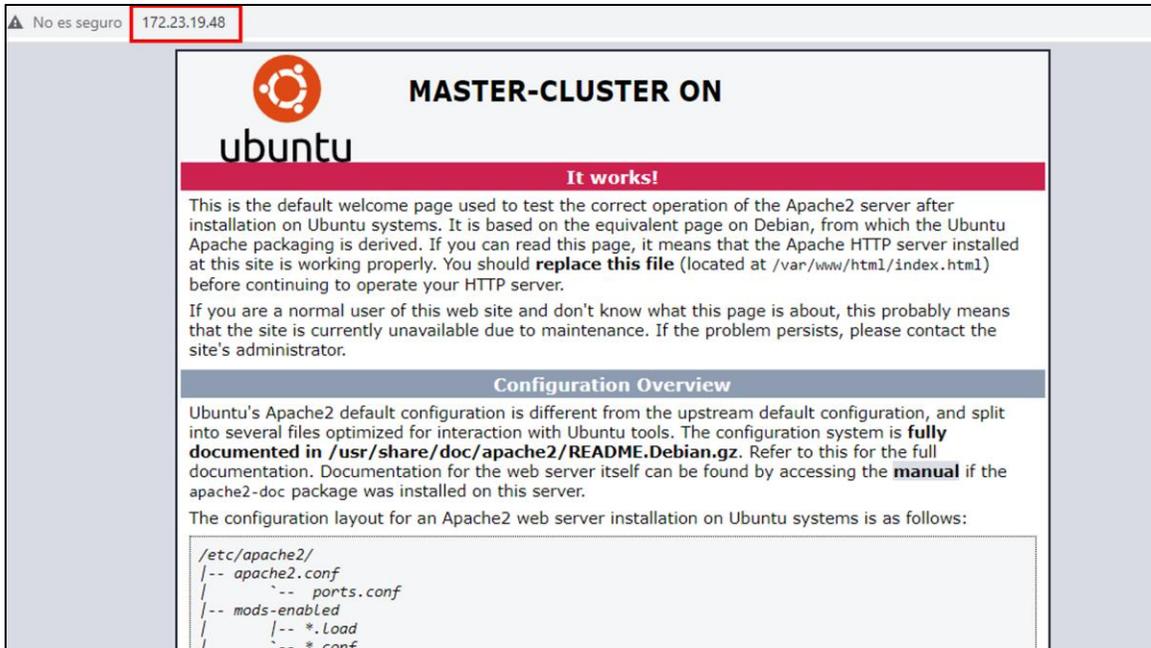


Figura. 16. Consulta al servicio apache2 desde el IP virtual dada al cliente.

Para verificar el funcionamiento total del clúster, se ha de suspender el servicio de apache2 en el servidor maestro (ver figura No. 17). Desde el navegador se introducirá la IP virtual, la cual deberá redirigir el servicio hacia el servidor esclavo automáticamente. Como se observa en la figura No. 18, el servidor maestro no responde directamente la solicitud de apache2 ya que se ha detenido el servicio.

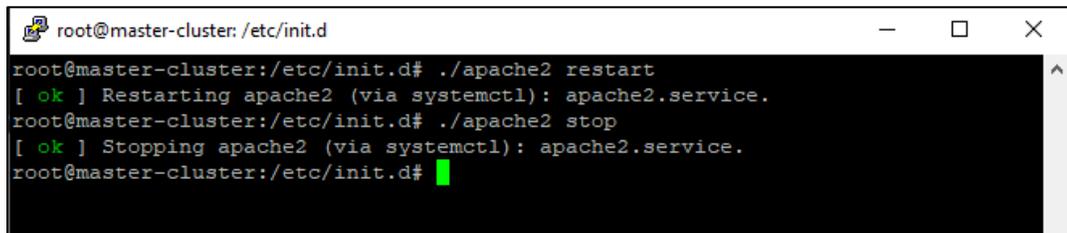


Figura. 17. Deteniendo el servicio de apache2.

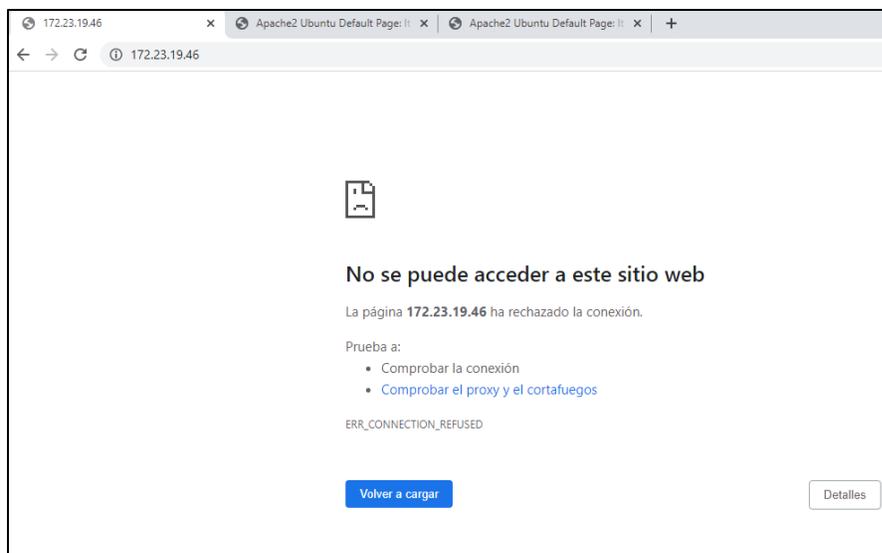


Figura. 18. Página de servidor maestro en desconexión.

Al refrescar la página que posee la IP virtual, esta inmediatamente dirige la solicitud para que sea atendida por el servidor esclavo, como se observa en la figura No. 19.

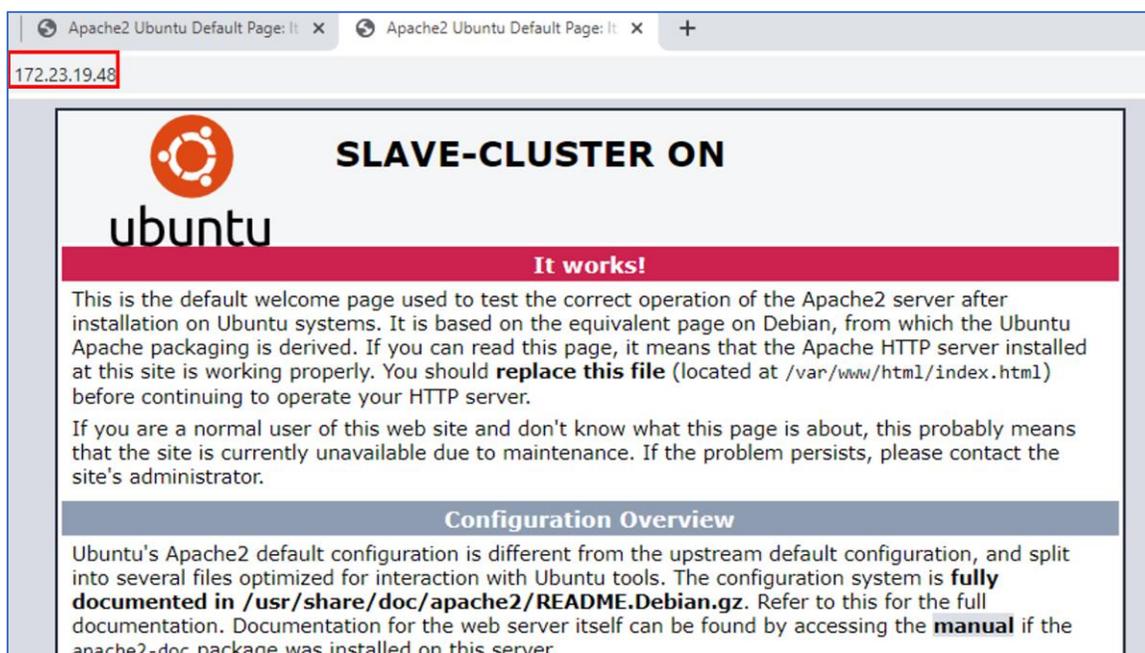


Figura. 19. Servidor esclavo toma el control de las peticiones en el IP de clientes.

Finalmente restauramos el servicio de apache2 en el servidor maestro y también reiniciamos el servicio de Heartbeat en el servidor maestro para devolver el control a este último.

Conclusiones

- Los sistemas distribuidos son susceptibles a fallas.
- La redundancia y replicación son las técnicas para atacar el problema.
- Un clúster de alta disponibilidad asegura un cierto grado absoluto de continuidad operacional durante un periodo de medición alto.
- Si se debe dar mantenimiento al nodo maestro, cualquier otro nodo esclavo puede tomar la responsabilidad de continuar brindando respuesta a las peticiones de clientes.

Referencias

Electrónica, E. d. (2011). Sistemas Distribuidos. *Departamento de Sistemas e Informáticas*, 8,21-25.

Sentinela. (15 de Noviembre de 2018). *YouTube*. Obtenido de Cluster de Alta Disponibilidad - ¿Cómo crear un clúster de alta disponibilidad?: <https://www.youtube.com/watch?v=iPOOISeEeP8>