

SGF. IES Haría

UT2. Actividad 8

Alta disponibilidad de servicios con DRBD y Heartbet

Objetivo general: Configuración de servidores web de alta disponibilidad

Duración prevista: 3 horas

Software: VirtualBox, Debian versión estable, DRBD, Heartbeat y apache2

Mínimos que se persiguen en la actividad:

- Instalar y configurar un servicio en modo de alta disponibilidad

Recursos

- http://www.howtoforge.com/high_availability_nfs_drbd_heartbeat_p4
- <https://help.ubuntu.com/lts/serverguide/drbd.html>
- <http://docs.opsview.com/doku.php?id=opsview4.1:hamster-debian-howto>

Teoría:

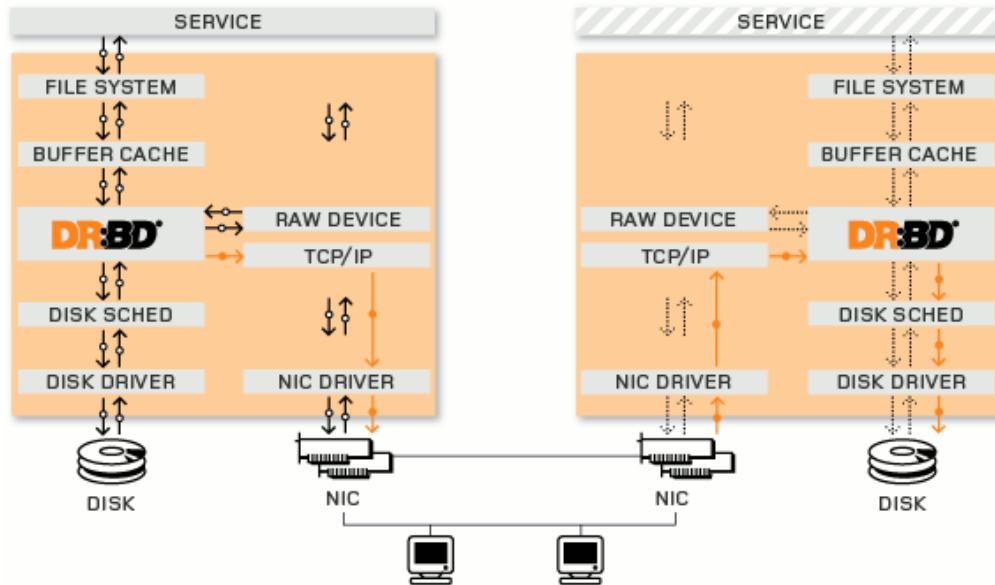
Como hemos visto la **redundancia** es un principio de diseño por el cual diversos sistemas de seguridad pueden hacer la misma función simultáneamente, garantizando en caso accidental de uno de ellos los otros sirvan de respaldo.

Vamos a montar un sistema en el que tenemos un servidor principal con un servicio (web) en ejecución. Los datos de la web se almacenan de forma redundante en un segundo servidor y además se configuraran los servicios de tal forma que si cae el nodo principal el secundario entrará en funcionamiento en poco tiempo y automáticamente reemplazando al principal.

DRBD

DRBD es un software que permite hacer **réplica de los datos de una partición entre varias máquinas**, por lo que nos permite tener la misma información en tiempo real en dos equipos. Para ello, este servicio se encarga de que los datos que escriben sobre uno de los discos, automáticamente serán replicados por red sobre el disco del segundo equipo. Es como un **RAID1**, pero entre diferentes equipos y a través de la red.

En una configuración típica de DRBD la partición replicada solamente se monta en una de las máquinas en la partición de la otra máquina los datos se replican, pero con la partición desmontada, por lo que solo una de las máquinas puede acceder a los datos (la que tiene la partición montada).



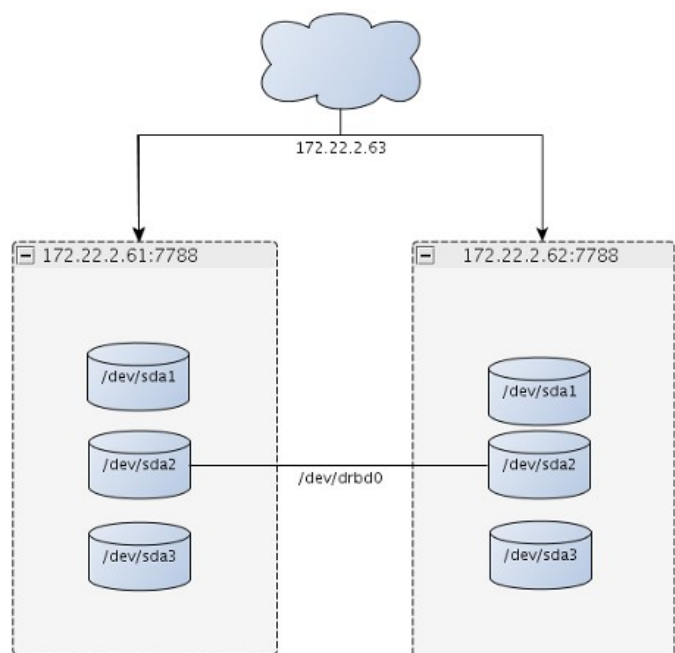
De esta forma, estaríamos montando un sistema de **cluster** en modo **activo/pasivo**: una de las máquinas accede a los datos y da servicio a la red y en caso de que esta falle se monta la partición replicada para poder seguir accediendo a los datos. Como vamos a estar replicando datos en tiempo real y continuamente, necesitamos una tarjeta de red dedicada en exclusiva para este trabajo de sincronización.

Después deberemos elegir los dispositivos que queremos utilizar. Deberán ser del mismo tamaño (para que no haya problemas a la hora de replicar datos de un dispositivo de mayor tamaño sobre otro menor).

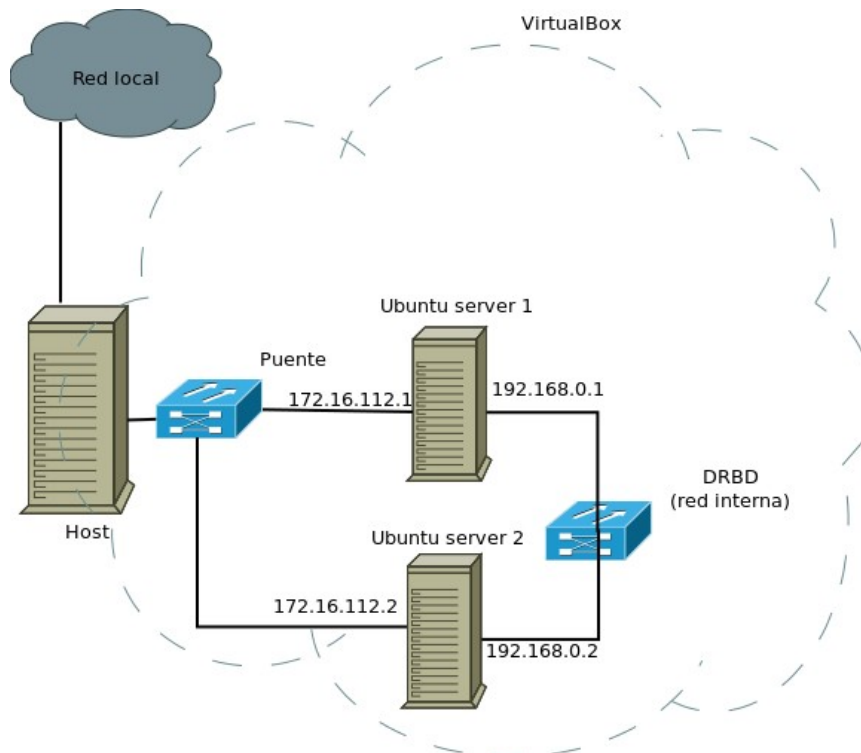
HEARTBEAT

El software heartbeat se encarga de detectar si alguno de los nodos del cluster de alta disponibilidad se ha venido abajo y en ese caso activa en el otro nodo los servicios configurados para ser replicados.

Trabaja enviando latidos (ping), que verifican si el servidor principal está activo o no. Si al cabo de un cierto tiempo el servidor principal no responde dichos ping, heartbeat determina que se encuentra inactivo/caído, y automáticamente activa al servidor secundario para que asuma el control de la red.



Esquema de la práctica:



Configuración de la red

Debian server 1: server x 1

- Interfaz modo puente – eth0 – 172.16.112. x 1/16
- Interfaz red interna – eth1 – 192.168.0.1/24
- /etc/resolv.conf
 - Dominio – red.local
 - Nameserver – 172.16.0.1

Debian server 2: server x 2

- Interfaz modo puente – eth0 – 172.16.112. x 2/16
- Interfaz red interna – eth1 – 192.168.0.2/24
- /etc/resolv.conf
 - Dominio – red.local
 - Nameserver – 172.16.0.1

IP Virtual 172.16.112. x 5 será la IP virtual que represente al servidor web en modo cluster en la red local.

x – número de tu equipo en la red local

Particiones en los dos equipos

Se añadirá un segundo disco a la máquina virtual, crearemos en el una partición que

será en la que se repliquen los datos y otra para los metadatos de **drbd**.

- `/dev/sdb1 150M` ← se utilizará para los **metadatos** de DRBD
- `/dev/sdb2 2,85GB` ← será el disco que se replicará

Pasos de la Actividad:

Configurar el entorno virtual

1. Clona una de las máquinas virtuales de Ubuntu Server de las que dispones, dale el nombre **Debian Server DRBD 1**.
2. Asegúrate de que tiene dos tarjetas de red una en modo puente y la otra conectada a una red interna.
3. Añade un disco duro de 3GB a la máquina virtual.
4. Arranca la máquina virtual y utilizando Gparted crea dos particiones ext3 (sdb1 de 150M y sdb2 que ocupe el resto del disco) que ocupe todo el disco creado.
5. Apaga la máquina virtual y clónala con el nombre **Ubuntu Server DRBD 2**. Asegúrate de cambiar las MACs de la nueva máquina virtual.
6. Arranca las máquinas virtuales, inicia sesión como **root** y configura la red con los parámetros del esquema de la práctica:

Ubuntu server 1: **serverx1**

- Nombre (FQDN) del equipo: **serverx1** (`/etc/hostname` y `/etc/hosts`)
- Interfaz modo puente – `eth0` – `172.16.112.x1/16`
- Interfaz red interna – `eth1` – `192.168.0.1/24`
- Dominio – `red.local`

Ubuntu server 2: **serverx2**

- Nombre (FQDN) del equipo: **serverx2** (`/etc/hostname` y `/etc/hosts`)
- Interfaz modo puente – `eth0` – `172.16.112.x2/16`
- Interfaz red interna – `eth1` – `192.168.0.2/24`
- Dominio – `red.local`

7. Comprueba el correcto funcionamiento de la red:
 - **serverx1** y **serverx2** acceden a Internet
 - **serverx1** y **serverx2** son accesibles a través de la red interna.
8. Configura el cache de paquetes en ambos equipos

Sincronizando la hora

Para que el sistema funcione correctamente es importante que la fecha y la hora esté correctamente configurada

9. Instalamos el client de ntp en ambos equipos

```
# apt-get install ntp ntpdate
```

Comprueba ejecutando **date** que la fecha y la hora coinciden en ambos equipos

Instalando y configurando DRBD

10. Instalación de paquetes necesarios

```
# apt-get install drbd8-utils
```

11. Los dos equipos han de poder ser accesibles por el nombre desde el otro equipo. Para ello editamos `/etc/hosts` en ambos equipos y añadimos la IP interna y el nombre. Editamos `/etc/hosts` y añadimos

En server1:

```
192.168.0.2      serverx2.red.local  serverx2
```

En server2:

```
192.168.0.1      serverx1.red.local  serverx1
```

12. Configuramos drbd editando en ambos equipos `/etc/drbd.conf`. Borramos el contenido actual e insertamos:

```
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
    protocol C;
    startup {
        wfc-timeout 15;
        degr-wfc-timeout 60;
    }
    disk {
        on-io-error detach;
    }
    net {
        cram-hmac-alg sha1;
        shared-secret "secret";
    }
    on serverx1 {
        device /dev/drbd0;
        disk /dev/sdb2;
        address 192.168.0.1:7788;
        meta-disk /dev/sdb1[0];
    }
    on serverx2 {
        device /dev/drbd0;
        disk /dev/sdb2;
        address 192.168.0.2:7788;
        meta-disk /dev/sdb1[0];
    }
}
```

13. Creamos los **meta-datos** en la partición sdb1 de **ambos servidores**:

```
# drbdadm create-md r0
```

Normalmente sale un error similar a este:

```
Found ext3 filesystem
```

```
This would corrupt existing data.  
If you want me to do this, you need to zero out the first part  
of the device (destroy the content).  
You should be very sure that you mean it.  
Operation refused.
```

```
Command 'drbdmeta 1 v08 /dev/sdb1 0 create-md' terminated with exit code  
40  
drbdadm create-md r0: exited with code 40
```

Lo solucionamos **destruyendo** todos los datos presentes en la partición:

```
# dd if=/dev/zero of=/dev/sdb1 bs=1M count=150  
  
128+0 records in  
128+0 records out  
134217728 bytes (134 MB) copied, 0.333349 seconds, 403 MB/s
```

Volvemos a ejecutar el en **ambos equipos**:

```
# drbdadm create-md r0
```

14. Iniciamos el servicio. En ambos servidores ejecutamos

```
# /etc/init.d/drbd start
```

15. Configuramos **server1** como servidor principal. En **server1** ejecutamos:

```
# drbdadm -- --overwrite-data-of-peer primary all
```

Para poder comprobar en cada momento que máquina es la principal y cuál es la secundaria lo podemos hacer ejecutando:

```
# cat /proc/drbd  
  
version: 8.4.3 (api:1/proto:86-101)  
  
srcversion: 1A9F77B1CA5FF92235C2213  
  
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----  
  
    ns:113604 nr:8 dw:113612 dr:4938 al:30 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f  
    oos:0
```

El parámetro **ro:Primary/Secondary** nos indica que la máquina en la que hemos ejecutado el comando es la primaria y este conectada a otra que hace de secundario.

16. Damos formato al dispositivo replicado virtual y lo montamos. En el servidor principal **server1** ejecutamos.

```
# mkfs.ext3 /dev/drbd0  
  
# mount /dev/drbd0 /srv
```

17. Comprobando el funcionamiento

Copiamos algunos archivos al disco replicado. En el **servidor principal** ejecutamos:

```
# cp -r /etc/default /srv
```

Desmontamos srv. En el **servidor principal** ejecutamos:

```
# umount /srv
```

Cambiamos server1 al rol de servidor secundario. En server1 ejecutamos:

```
# drbdadm secondary r0
```

Convertimos server2 en principal. En server2 ejecutamos:

```
# drbdadm primary r0
```

Montamos la partición. En server2 ejecutamos:

```
# mount /dev/drbd0 /srv
```

Si miramos el contenido de /srv con ls veremos que tiene los archivos copiados.

Volvemos a dejar **server1** como servidor principal, ejecutamos:

```
server2 → sudo umount /srv
```

```
server2 → sudo drbdadm secondary r0
```

```
server1 → sudo drbdadm primary r0
```

```
server1 → sudo mount /dev/drbd0 /srv
```

Instalando y configurando Heartbeat

Heartbeat es la herramienta que controlará todo el sistema. Lo instalaremos en ambos servidores y se encargará de monitorizar al otro servidor. Así, si **serverx1** se viene abajo, heartbeat en el **serverx2** lo detectará y hará que **serverx2** tome el control.

Heartbeat se encarga de iniciar y detener el servidor web apache en ambos servidores y de gestionar quien es el nodo principal y el secundario de drbd. Además provee una IP virtual (172.16.112.x5 en nuestro caso), de esa forma los clientes de la red ven un único servidor aunque el servicio esté replicado en dos equipos

18. Instalamos heartbeat y el servidor web apache. En los **dos servidores** ejecutamos:

```
# apt-get install heartbeat apache2
```

19. Detenemos el servicio apache y lo configuramos para que no se inicie en el arranque puesto que heartbeat será el encargado de gestionarlo. En ambos servidores ejecutamos:

```
# /etc/init.d/apache2 stop  
# update-rc.d -f apache2 remove
```

20. Cambiamos el directorio de inicio de apache a una carpeta en el disco replicado. En **ambos servidores** editamos el fichero `/etc/apache2/sites-available/000-default.conf` y sustituimos:

```
DocumentRoot /var/www
```

Por

```
DocumentRoot /srv/www
```

Y añadimos dentro de la configuración del host virtual las siguientes líneas :

```
<Directory /srv/www/>  
    Options +Indexes +FollowSymLinks +MultiViews  
    Require all granted  
</Directory>
```

21. Creamos una página web y la ubicamos en el inicio del servidor web. En **servidor 1** ejecutamos:

```
# mkdir /srv/www
```

Creamos el fichero `/srv/www/index.html` e insertamos:

```
<h1>heartbeat y drbd molan mazo</h1>
```

Para configurar heartbeat tenemos que crear varios ficheros.

22. En **ambos equipos** creamos el fichero `/etc/heartbeat/ha.cf` y le agregamos el siguiente contenido:

```
logfacility local0  
debugfile /var/log/ha-debug  
logfile /var/log/ha-debug  
  
keepalive 2  
warntime 10  
deadtime 15  
initdead 30  
  
bcast eth1  
  
node serverx1  
node serverx2
```



```
auto_failback off
```

```
respawn hacluster /usr/lib/heartbeat/ipfail  
apiauth ipfail gid=haclient uid=hacluster
```

23. En **ambos equipos** creamos el fichero `/etc/heartbeat/haresources` y le agregamos (en una sola línea)

```
serverx1 172.16.112.x5 drbdisk::r0 Filesystem::/dev/drbd0::/srv::ext3::noatime apache2
```

Recuerda que **x** es el número de tu equipo.

24. Creamos en ambos equipos el fichero con las claves que utilizarán ambos servidores para validarse. Editamos en `serverx1` y `serverx2` `/etc/heartbeat/authkeys` e insertamos:

```
auth 2  
2 sha1 cluster-4auth
```

25. Protegemos adecuadamente el fichero. En **ambos equipos** ejecutamos:

```
# chmod 600 /etc/heartbeat/authkeys
```

26. Iniciamos heartbeat. En ambos servidores ejecutamos:

```
# /etc/init.d/heartbeat start
```

Comprobaciones

Si todo ha ido bien en el servidor **principal** debería haber un dispositivo `eth0:0` con la IP común configurada:

```
$ ifconfig  
eth0      Link encap:Ethernet  direcciónHW 08:00:27:da:ba:00  
          Direc. inet:172.16.112.x  Difus.:172.16.255.255 Másc:255.255.0.0  
          Dirección inet6: fe80::a00:27ff:feba:ba00/64 Alcance:Enlace  
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1  
          Paquetes RX:1623 errores:0 perdidos:0 overruns:0 frame:0  
          Paquetes TX:1114 errores:0 perdidos:0 overruns:0 carrier:0  
          colisiones:0 long.colaTX:1000  
          Bytes RX:189736 (189.7 KB)  TX bytes:126284 (126.2 KB)  
  
eth0:0    Link encap:Ethernet  direcciónHW 08:00:27:da:ba:00  
          Direc. inet:172.16.112.x5 Difus.:192.168.1.255 Másc:255.255.255.0  
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1  
...  
...
```

Si accedes desde tu equipo a la URL
deberías ver la web creada anteriormente.

Si apagas el servidor principal, en el servidor secundario se debería crear `eth0:0` y la web ser **accesible** por la misma IP.

27. **Intercambiando roles.** Si queremos que los servidores intercambien sus papeles sin tener que apagar los equipos, esto es, que el principal se convierta en secundario y viceversa, lo podemos hacer reiniciando HeartBeat en el servidor principal.

```
# /etc/init.d/heartbeat start
```

Si has llegado a este punto enhorabuena. Eran bastantes pasos. Espero que lo hayas entendido.

Avisa al profesor para que te revise la práctica