

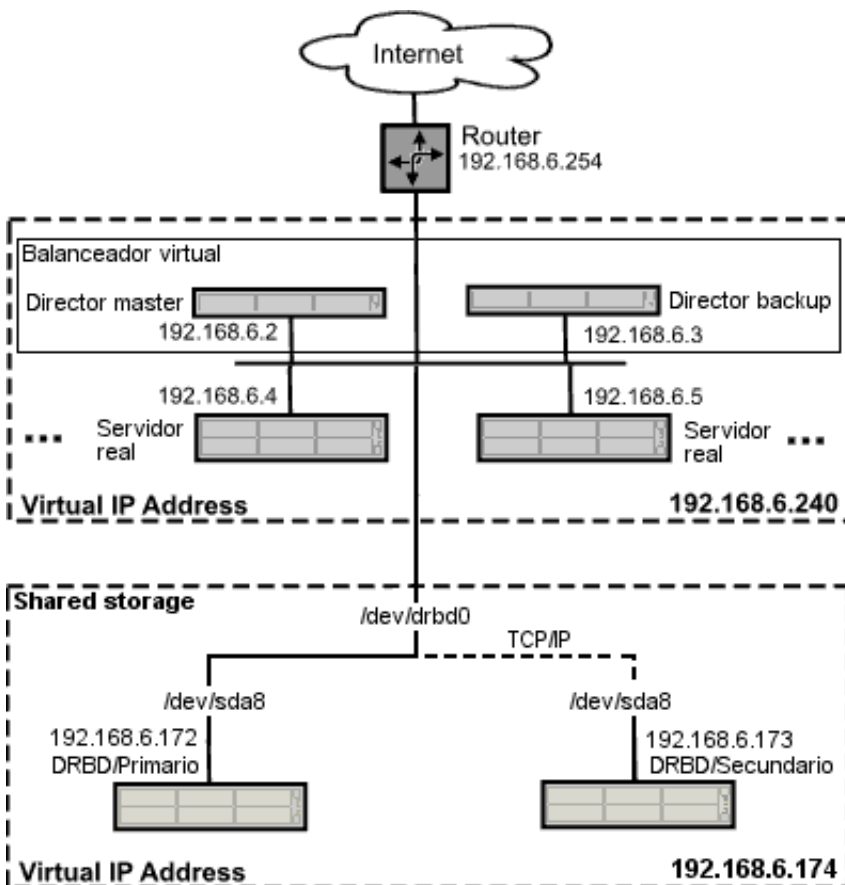
## Mirroring remoto con DRBD en Linux

[DRBD](#) (*Distributed Replicated Block Device*, [drbd.org](http://drbd.org)) permite mirroring remoto en tiempo real (equivalente a RAID-1 en red), algo muy difícil de conseguir con otros sistemas como [rsync](#) ya que éste no puede trabajar en tiempo real por su consumo de memoria y CPU.

DRBD crea un dispositivo de bloques *drbd0* accesible desde ambos servidores. El servidor primario es el que tiene acceso RW en el dispositivo *drbd0*: cada vez que escribe algo en *drbd0* lo escribe en la partición física y esos mismos datos se envían por TCP/IP al servidor secundario (que sólo tiene acceso RO) consiguiendo que ambas particiones físicas estén sincronizadas, exactamente igual que un RAID-1.

Vamos a ver un ejemplo con DRBD, HeartBeat y NFS para implementar [almacenamiento compartido](#) en un cluster LVS. Buscamos tres cosas:

1. **almacenamiento compartido**: para que todos los servidores web de un cluster LVS con Apache sirvan los mismos datos exportaremos el directorio */data/export* mediante NFS para que lo monten los servidores reales del cluster.
2. **tener una copia de los datos (mirroring)**: el contenido de *drbd0* (*/data*) estará sincronizado mediante DRBD entre *server1* y *server2*.
3. **alta disponibilidad**: usaremos HeartBeat en los servidores DRBD para activar el secundario si el primario falla, consiguiendo de ese modo un almacenamiento de alta disponibilidad.



### 1. Configuración inicial de server1 y server2

- a. **Particiones**: particionaremos *server1* y *server2* de la siguiente manera:

*/dev/sda1*: /boot, primaria, ext3, bootable.

*/dev/sda5*: /, lógica, ext3.

*/dev/sda6*: lógica, swap.

*/dev/sda7*: 150 MB, *unmounted*, lógica, ext3 (contendrá los meta-datos de DRBD).

*/dev/sda8*: *unmounted*, lógica, ext3 (contendrá el directorio */data*).

## Atención

- `/dev/sda7` y `/dev/sda8` deben tener idéntico tamaño en ambos servidores.
- Los meta-datos de DRBD (logs de estado y de flujo, al estilo de un journal) necesitan 128 MB, por lo que `/dev/sda7` tendrá 150 MB.
- No hay que montar ni `/dev/sda7` ni `/dev/sda8`. El instalador nos preguntará:

*No ha asignado 'mount point' a la partición #7 de /dev/sda. ¿Desea regresar al menú de particionado?*

Responderemos *No, continuar con el particionado.*

- b. **/etc/fstab**: comprobaremos en ambos servidores que `/dev/sda7` y `/dev/sda8` no aparecen en `/etc/fstab`, que debe ser similar a:

```
# File_system  Mount_point  Type  Options  Dump Pass
/dev/sda5      /             ext3  defaults  0    1
/dev/sda1      /boot        ext3  defaults  0    2
/dev/sda6      none         swap  sw        0    0
proc           /proc        proc  defaults  0    0
/dev/hdc       /media/cdrom0 iso9660 ro,user,noauto 0    0
/dev/fd0       /media/floppy0 auto    rw,user,noauto 0    0
```

- c. **Instalar un cliente NTP** (paquete `ntpdate`): conectaremos ambos servidores al mismo servidor NTP para asegurarnos de que tienen la misma hora del sistema (*System Time*). Su sintaxis es:

```
# ntpdate <server>
```

Para comprobar la hora del sistema:

```
# date
```

## 2. DRBD

- a. **compilar el módulo DRBD** (paquetes `drbd0.7-module-source` `drbd0.7-utils`): en ambos servidores (necesitamos también las fuentes del kernel):

```
# apt-get install kernel-headers-2.6.8-2-386
# apt-get install drbd0.7-module-source drbd0.7-utils
# cd /usr/src/
# tar xvfz drbd0.7.tar.gz
# cd modules/drbd/drbd
# make
# make install
```

- b. **cargar el módulo DRBD**: en ambos servidores:

```
# modprobe drbd
```

- c. **/etc/drbd.conf**: editar el archivo de configuración de DRBD (idéntico en ambos servidores):

```
resource r0 {
    protocol C;
    incon-degr-cmd "halt -f";
    startup {
        degr-wfc-timeout 120;    # 2 minutes
    }
    disk {
        on-io-error    detach;
    }
    net {
    }
    syncer {
        rate 10M;
        group 1;
        al-extents 257;
    }
    on server1 {
```

```

        device    /dev/drbd0;
        disk      /dev/sda8;
        address   192.168.6.172:7788;
        meta-disk /dev/sda7[0];
    }
    on server2 {
        device    /dev/drbd0;
        disk      /dev/sda8;
        address   192.168.6.173:7788;
        meta-disk /dev/sda7[0];
    }
}

```

- d. **activar DRBD**: en ambos servidores:

```

# drbdadm up all
# cat /proc/drbd
version: 0.7.10 (api:77/proto:74)
SVN Revision: 1743 build by phil@mescal, 2005-01-31 12:22:07
0: cs:Connected st:Secondary/Secondary ld:Inconsistent
ns:0 nr:0 dw:0 dr:0 al:0 bm:1548 lo:0 pe:0 ua:0 ap:0
1: cs:Unconfigured

```

Vemos que ningún servidor ha sido configurado como primario (*Secondary/Secondary*) y que los datos son inconsistentes (*Inconsistent*), ya que todavía no hemos hecho la sincronización inicial.

- e. **configurar el servidor primario**: ejecutaremos el siguiente comando en *server1* para que sea el servidor primario:

```
# drbdadm -- --do-what-I-say primary all
```

- f. **sincronización inicial** ejecutaremos el siguiente comando en *server1* para realizar la sincronización inicial entre *server1* y *server2*:

```
# drbdadm -- connect all
```

Durará un buen rato, por lo que tendremos paciencia. Para ver el progreso de la sincronización:

```

# cat /proc/drbd
version: 0.7.10 (api:77/proto:74)
SVN Revision: 1743 build by phil@mescal, 2005-01-31 12:22:07
0: cs:SyncSource st:Primary/Secondary ld:Consistent
ns:13441632 nr:0 dw:0 dr:13467108 al:0 bm:2369 lo:0 pe:23 ua:226 ap:0
[=====>.....] sync'ed: 53.1% (11606/24733)M
finish: 1:14:16 speed: 2,644 (2,204) K/sec
1: cs:Unconfigured

```

Cuando termine la salida será:

```

# cat /proc/drbd
SVN Revision: 1743 build by phil@mescal, 2005-01-31 12:22:07
0: cs:Connected st:Primary/Secondary ld:Consistent
ns:37139 nr:0 dw:0 dr:49035 al:0 bm:6 lo:0 pe:0 ua:0 ap:0
1: cs:Unconfigured

```

### 3. NFS

- a. **instalar el servidor NFS** (paquete *nfs-kernel-server*): en ambos servidores:

```
# apt-get install nfs-kernel-server
```

- b. **borrar los scripts de inicio de NFS**: NFS estará bajo el control de HeartBeat (que se encargará de lanzarlo) por lo que borraremos los scripts de inicio de NFS en ambos servidores:

```

# update-rc.d -f nfs-kernel-server remove
# update-rc.d -f nfs-common remove
# /etc/init.d/nfs-kernel-server stop

```

- c. **directorio a exportar**: en ambos servidores editaremos */etc/exports* y

añadiremos una entrada para exportar `/data/export` (en este punto la partición `/dev/sda8` todavía está vacía y sin montar):

```
/data/export/ 192.168.6.0/255.255.255.0(rw,no_root_squash, sync)
```

- d. **`/var/lib/nfs`**: NFS almacena información importante (locks, etc.) en `/var/lib/nfs`. Si `server1` falla `server2` le reemplazará pero la información que `server2` tiene en `/var/lib/nfs` será diferente de la que tenía `server1`. Para resolver este problema vamos a almacenar la información de `/var/lib/nfs` en la partición `/dev/sda8` que está sincronizada mediante DRBD entre `server1` y `server2`. Así, si `server1` falla `server2` dispondrá de toda su información. En `server1`:

```
# mkdir /data
# mount -t ext3 /dev/drbd0 /data
# mv /var/lib/nfs /data
# ln -s /data/nfs /var/lib/nfs
# mkdir /data/export
# umount /data
```

En `server2`:

```
# mkdir /data
# rm -fr /var/lib/nfs
# ln -s /data/nfs /var/lib/nfs
```

#### 4. HeartBeat

Instalaremos HeartBeat en ambos servidores. HeartBeat controla todo el asunto: lanza y detiene NFS en ambos servidores, los monitoriza y activa el servidor secundario si el primario falla y se encarga de que el servidor NFS responda en la IP virtual (VIP) `192.168.6.174`.

- a. **instalar HeartBeat** (paquete `heartbeat`): en ambos servidores:

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

- b. **`/etc/heartbeat/ha.cf`**: crearemos este archivo idéntico en ambos servidores:

```
logfacility      local0
bcast          eth0
keepalive      2
deadtime       10
node           server1
node           server2
```

- c. **`/etc/heartbeat/haresources`**: crearemos este archivo idéntico en ambos servidores:

```
server1 IPaddr::192.168.6.174/24/eth0 drbddisk::r0
        Filesystem::/dev/drbd0::/data::ext3 nfs-kernel-server
```

En este archivo se especifica el nombre del servidor primario (`server1`), la IP virtual (`192.168.6.174`), el *resource* DRBD definido en `/etc/drbd.conf` (`r0`), el dispositivo DRBD (`/dev/drbd0`, `/data`, `ext3`) y el servidor a monitorizar (`nfs-kernel-server`).

- d. **`/etc/heartbeat/authkeys`**: crearemos este archivo idéntico en ambos servidores:

```
auth 3
3 md5 mi_password
```

Aquí definimos el mecanismo de autenticación (`md5`) y el password para que los dos demonios `heartbeat` de los servidores se autentifiquen uno contra el otro (`mi_password`). Sólo `root` debe tener permisos de lectura sobre `/etc/heartbeat/authkeys` por lo que haremos:

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

#### 5. Arrancar y comprobar

- a. **arrancar DRBD**: en ambos servidores:

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

- b. **arrancar HeartBeat**: en ambos servidores:

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

- c. **comprobar la IP virtual**: en *server1*:

```
# ifconfig
eth0 Link encap:Ethernet HWaddr 00:0C:29:A1:C5:9B
      inet addr:192.168.6.172 Bcast:192.168.6.255 Mask:255.255.255.0
      inet6 addr: fe80::20c:29ff:feal:c59b/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:18992 errors:0 dropped:0 overruns:0 frame:0
      TX packets:24816 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:2735887 (2.6 MiB) TX bytes:28119087 (26.8 MiB)
      Interrupt:177 Base address:0x1400

eth0:0 Link encap:Ethernet HWaddr 00:0C:29:A1:C5:9B
      inet addr:192.168.6.174 Bcast:192.168.6.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      Interrupt:177 Base address:0x1400
```

Si hacemos esto mismo en *server2* no veremos la IP virtual.

- d. **comprobar los dispositivos montados**: en *server1*:

```
# df -h
Filesystem              Size  Used Avail Use% Mounted on
/dev/sda5                4.6G  430M  4.0G  10% /
tmpfs                   126M    0  126M   0% /dev/shm
/dev/sda1                 89M   11M   74M  13% /boot
/dev/drbd0                24G   33M   23G   1% /data
```

Si hacemos esto mismo en *server2* no veremos */dev/drbd0*.

- e. **simular el fallo de server1**: creamos en *server1* el archivo */data/export/test1* y simulamos el fallo de *server1*:

```
# touch /data/export/test1
# /etc/init.d/heartbeat stop
```

Ahora, si todo funciona, si ejecutamos en *server2*:

```
# ifconfig: veremos la IP virtual 192.168.6.174
# df -h: veremos el dispositivo /dev/drbd0
# ls -l /data/export: veremos el archivo test1 (el mirror funciona).
```

- f. **simular que server1 vuelve a estar en línea**: vamos a comprobar que cuando *server1* vuelva a estar en línea es activado como primario y recupera la IP virtual y el dispositivo */dev/drbd0*. Creamos en *server2* el archivo */data/export/test2*:

```
# touch /data/export/test2
```

Levantamos *server1*:

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

Esperamos unos segundos y, si todo funciona, si ejecutamos en *server1*:

```
# ifconfig: veremos la IP virtual 192.168.6.174 otra vez.
# df -h: veremos el dispositivo /dev/drbd0 otra vez.
# ls -l /data/export: veremos el archivo test2 (el mirror funciona).
```

- g. **comprobar NFS desde los clientes**: para acceder desde los clientes a los datos exportados mediante NFS por *server1* y *server2* usaremos la IP virtual *192.168.6.174*:

```
# mount 192.168.6.174:/data/export /var/www
```

Para montar el dispositivo al arrancar editaremos */etc/fstab* y añadiremos una

entrada:

```
# File_system      Mount_point Type Options Dump Pass
192.168.6.174:/data/export /var/www  nfs  rw      0    0
```

---

Article printed from Estréllate y Arde: <http://www.estrelateyarde.org>

URL to article: <http://www.estrelateyarde.org/discover/drbd-en-linux>

Copyright © 2010 Estrelate y Arde