

Enxeñaría de Infraestruturas Informáticas

Grao en Enxeñaría Informática



UNIVERSIDADE DA CORUÑA

PRÁCTICAS SOBRE CLUSTERS DE ALTA DISPOÑIBILIDADE E BALANCEO DE CARGA

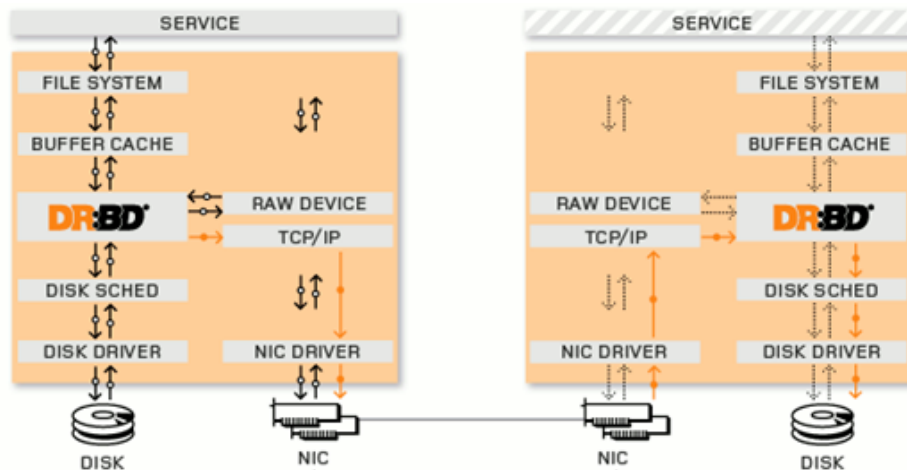
Xoán C. Pardo
xoan.pardo@udc.gal

PRACTICA 2

Cluster HA con almacenamiento replicado usando DRBD

Obxectivo

O obxectivo desta práctica é configurar un cluster HA que teña o almacenamento replicado. Para replicar o almacenamento vamos usar DRBD (<https://www.linbit.com/en/drbd-oss-distribution>). O núcleo da funcionalidade de DRBD está implementada nun *driver* do *kernel* que realiza as lecturas e escrituras en discos locais e, de maneira transparente, a replicación por rede en tempo real nunha ou máis localizacións de respaldo.



DRBD soporta case calquera dispositivo de almacenamento:

- unha partición de disco ou un disco completo.
- un dispositivo RAID software.
- un volume lóxico LVM (Linux).
- calquera dispositivo de bloques que o noso sistema recoñeza (p.e. un disco USB).

Isto proporciona moita flexibilidade, xa que poderíase crear un RAID 1 usando, por exemplo, unha partición no disco local dun computador e un disco USB externo conectado noutro computador remoto accesíbel pola rede. E todo en tempo real, os datos que se escriban no disco local, automaticamente son replicados usando a rede no disco USB remoto. Ademais poderíanse replicar tantos dispositivos como se quixera, sincronizándose todos simultaneamente.

Na práctica vamos configurar un RAID-1 usando o almacenamento dispoñible en 2 nós do cluster HA. Ademais aproveitarase a práctica para ver opcións máis avanzadas de Pacemaker, como o **emparellamento de recursos** e as **restricións de orde** á hora de parar/iniciar servizos.

Xustificación da práctica

Haberá que entregar unha memoria en formato PDF na que se xustifique a realización de cada parte desta práctica. Os detalles sobre o que hai que incluír na memoria danse ao final de cada parte. A entrega da memoria será a través dunha tarefa no Moodle. A data límite de entrega indicárase no seu momento.

PRIMEIRA PARTE: Configuración dun servidor web no cluster HA

Nesta primeira parte vamos instalar un servidor web e configuralo como recurso do cluster HA. Os pasos a seguir son os seguintes¹:

NOTA: os seguintes pasos asumen que partimos dunha configuración como a da primeira parte da práctica 1: un cluster con dous nós en roles activo/pasivo cunha IP virtual configurada como recurso. Para os valores de adherencia e preferencia podes usar os indicados na práctica ou non definilos.

1. Instala o Apache nos dous nós do cluster.

```
$ sudo apt-get install apache2
```

2. Borra o contido por defecto da páxina /var/www/html/index.html e crea unha nova páxina, en ambos nós. Usa contidos diferentes na páxina de cada nó para que nos permita diferenciar cal é o que está servindo a páxina.
3. Comproba que o Apache estea executándose nos dous nós.

```
$ systemctl status apache2
```

4. Inícialo se non fora así.

```
$ systemctl start apache2
```

5. Instala un navegador web modo texto (p.e. links) nun dos nós ou, se o prefires, usa curl, que xa non é necesario instalalo.
6. Usando o navegador que instalaches no paso anterior ou curl, proba a cargar as páxinas dos nós usando as súas IPs (http://193.168.1.2, http://193.168.1.3) e a IP virtual (http://193.168.1.10).

Fíxate no nó que está servindo a páxina na IP virtual (o nó activo). **Nos seguintes pasos vamos supoñer que ese nó é o *master*.**

7. Para o *master* e comproba que o *slave* asume os recursos e pasa a servir a páxina web (se as fixeches distintas no paso 4, ao recargar a páxina no navegador, debería cambiar e pasar a ser a do *slave*).
8. Volve arrincar o *master* e para o *slave*. Comproba que o *master* volve servir a páxina ao recargar no navegador a IP virtual.
9. Volve arrincar o *slave* e para o seu servidor Apache.

```
slave$ systemctl stop apache2
```

10. Comproba que se agora paras o *master*, o *slave* non pasa a servir a páxina cando recargamos no navegador a IP virtual. Isto débese a que o servidor web non está configurado como un recurso do cluster, só a IP virtual o está.
11. Volve arrincar o *master* e volve iniciar o Apache no *slave*.

¹ Cando sexa necesario distinguir en que nó se introducen os comandos usaremos nos exemplos os *prompts* *master\$* e *slave\$*.

Configuración do servidor web como recurso

Para configurar o servidor web como un recurso do cluster HA os pasos a seguir son os seguintes:

12. Configura o Apache como recurso do cluster usando o axente de recurso `ocf:heartbeat:apache`².

```
$ sudo crm configure primitive xcpm1920-webserver \  
> ocf:heartbeat:apache params configfile=/etc/apache2/apache2.conf \  
> op monitor interval=1min
```

13. Emparella ambos recursos.

```
$ sudo crm configure collocation xcpm1920-website \  
> INFINITY: xcpm1920-VIP xcpm1920-webserver
```

14. Configura a orde dos recursos en caso de parada/reinicio.

```
$ sudo crm configure order xcpm1920-websiteorder \  
> mandatory: xcpm1920-VIP xcpm1920-webserver
```

15. Proba de novo a cargar as páxinas usando as súas IPs (`http://193.168.1.2`, `http://193.168.1.3`) e a IP virtual (`http://193.168.1.10`).

Comprobación da configuración do servidor web

Vamos asumir de novo que sexa o *master* o que está a servir a páxina na IP virtual.

16. Para o servidor Apache no *slave*. Comproba que non serve a páxina web usando a súa IP (`http://193.168.1.3`).
17. Para agora o *master* e comproba que o *slave* asume os recursos: que serve a súa páxina web (`http://193.168.1.3`) e a IP virtual (`http://193.168.1.10`).
18. Volve arrincar o *master* e para o seu servidor Apache. Comproba que non serve a páxina web usando a súa IP (`http://193.168.1.2`).
19. Para agora o *slave* e comproba que o *master* asume os recursos: que serve a súa páxina web (`http://193.168.1.2`) e a IP virtual (`http://193.168.1.10`).
20. Volve arrincar o *slave*.

² https://www.mankier.com/7/ocf_heartbeat_apache

Xustificación da primeira parte

Inclúe na memoria de xustificación desta práctica as capturas de pantalla que mostren a execución (comando e resultado) da seguinte secuencia de comandos:

NOTA: executa a secuencia de comandos partindo dun estado do cluster no que o nó activo sexa o *master*.

```
#Comproba o estado de partida
$ sudo crm configure show
$ sudo crm status
$ curl 193.168.1.2
$ curl 193.168.1.3
$ curl 193.168.1.10

#Para o master e comproba que o slave asume os recursos
$ sudo crm node standby xcpm1920-master
$ curl 193.168.1.3
$ curl 193.168.1.10

#Reinicia o master
$ sudo crm node online xcpm1920-master
$ sudo crm status

#Para o slave e comproba que o master asume os recursos
$ sudo crm node standby xcpm1920-slave
$ curl 193.168.1.2
$ curl 193.168.1.10

#Reinicia o slave
$ sudo crm node online xcpm1920-slave

#Comproba o estado final
$ sudo crm status
$ curl 193.168.1.2
$ curl 193.168.1.3
$ curl 193.168.1.10
```

Explica BREVEMENTE o comportamento que observaches no cluster ao executar a secuencia de comandos.

SEGUNDA PARTE: Configuración dun RAID-1 usando DRBD

Nesta segunda parte vamos instalar DRBD³, configurar un disco replicado entre os nós e preparar os contidos do disco para servir a páxina web. Os pasos a seguir son os seguintes:

Creación dos discos virtuais para o RAID-1

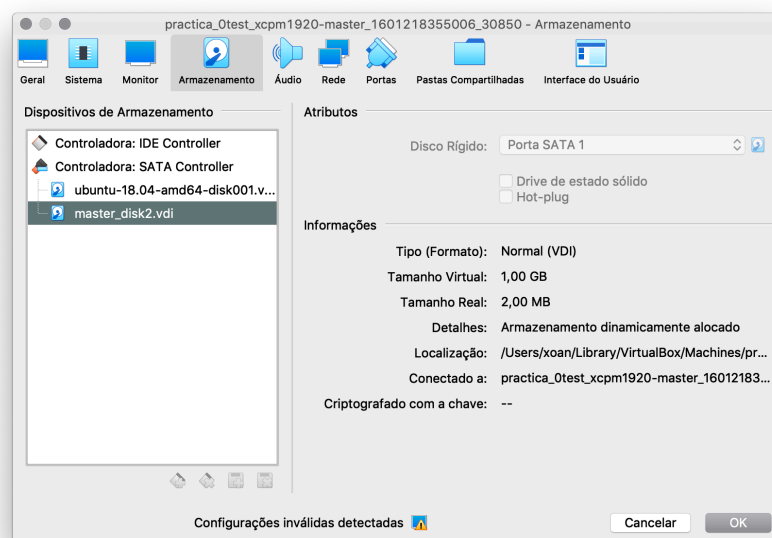
21. Se tes as VMs en execución páraas con: `vagrant halt`
22. Modifica o Vagrantfile do proxecto para engadir un novo disco de 1GB nos nós *master* e *slave*:
 - Activa o *flag* experimental ao comezo do Vagrantfile⁴.

```
ENV[ 'VAGRANT_EXPERIMENTAL' ] = 'disks'
```

- Engade o novo disco na configuración do *master* e do *slave*, p.e. para o *master* sería:

```
master.vm.disk :disk, size: "1GB", name: "master_disk2"
```

23. Levanta de novo as VMs.
24. Comproba que o novo disco está dispoñible en cada VM.
 - Exemplo en VirtualBox para *master*:



- Exemplo no terminal para *slave*:

```
vagrant@xcpm1920-slave:~$ lsblk /dev/sdb
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sdb   8:16   0  1G  0 disk
```

³ A documentación da versión 8.x de DRBD está en <https://docs.linbit.com/docs/users-guide-8.3>

⁴ Podes consultar a documentación de Vagrant aquí: <https://www.vagrantup.com/docs/disks/usage>

Configuración de DRBD

Para instalar e configurar DRBD segue os pasos seguintes **en ambos nós**.

25. Instala DRBD.

```
$ sudo apt-get install drbd8-utils
```

NOTA: se se che pregunta pola configuración de Postfix durante a instalación, escolle a opción "*No configuration*" e acepta.

26. Comproba se se cargou o módulo drbd no *kernel*.

```
$ lsmod | grep drbd
```

Se non aparece inicio e volve comprobar se se cargou.

```
$ sudo modprobe drbd
```

27. Crea o ficheiro `/etc/drbd.d/xcpm1920-drbd1.res`

```
resource xcpm1920-drbd1 {
    meta-disk internal;
    device /dev/drbd1;
    disk /dev/sdb;
    syncer {
        verify-alg sha1;
    }
    on xcpm1920-master {
        address 193.168.1.2:7789;
    }
    on xcpm1920-slave {
        address 193.168.1.3:7789;
    }
}
```

Esta configuración específica o seguinte:

- Crea un recurso que define un volume DRBD (`/dev/drbd1`) que utiliza os discos de 1GB (`/dev/sdb`) como dispositivos de almacenamento.
- O recurso forma parte dun cluster formado por 2 nós que se comunican usando o porto TCP 7789.
- Os metadatos de DRBD almacénanse nos mesmos discos que os datos (*meta-disk internal*).
- Úsase o algoritmo `sha1` para verificar que os bloques dos discos están sincronizados (*syncer verify-alg sha1*).

NOTA: ademais das opcións que se configuran no ficheiro `.res` de cada recurso, poden configurarse no ficheiro `/etc/drbd.d/global_common.conf` as opcións globais que se aplican a todos os recursos. Para ver a configuración completa pode usarse o comando:

```
$ sudo drbdadm dump
```

28. Inicia os metadatos do novo recurso DRBD.

```
$ sudo drbdadm create-md xcpm1920-drbd1
```

29. Inicia o recurso DRBD.

```
$ sudo drbdadm up xcpm1920-drbd1
```

- Comproba que se creou unha partición para DRBD no disco de 1GB. Exemplo no terminal para *slave*:

```
vagrant@xcpm1920-slave:~$ lsblk /dev/sdb
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sdb          8:16   0    1G  0 disk
└─drbd1 147:1   0 1024M  1 disk
```

30. Comproba o estado de DRBD.

```
$ cat /proc/drbd
```

Debería aparecer un recurso co disco en estado inconsistente (ds:Inconsistent/Inconsistent) e sen sincronizar (oos:1021)⁵.

Neste punto o recurso DRBD está configurado e iniciado en ambos nós, pero aínda non se configurou a sincronización. **Nesta práctica vaise facer a sincronización desde o *master* cara ao *slave*.**

31. Inicia a sincronización do recurso DRBD desde o *master*.

```
master$ sudo drbdadm -- --overwrite-data-of-peer primary xcpm1920-drbd1
```

32. Comproba de novo o estado de DRBD (compróbaos tanto no *master* como no *slave*).

```
$ cat /proc/drbd
```

Agora amosarase unha porcentaxe coa evolución da sincronización que rematará cando os discos alcancen o estado ds:UpToDate/UpToDate e estean sincronizados (oos:0).

NOTA: Durante a sincronización o estado da conexión indicará cs:SyncSource no *master* e cs:SyncTarget no *slave*.

Unha vez rematada a sincronización, o estado no *master* indicará o seguinte:

- cs:Connected, estado da conexión.
- ro:Primary/Secondary, roles dos nós (o *master* primeiro).
- ds:UpToDate/UpToDate, estados dos discos sincronizados (o disco local primeiro).
- ns:10212, KB enviados pola rede ao *slave*.
- dr:10364, KB lidos do disco local do *master*.
- oos:0, KB que faltan por sincronizar.

⁵ oos = *out-of-sync*

Pola súa banda, o estado no *slave* indicará o seguinte:

- *cs*:Connected, estado da conexión.
- *ro*:Secondary/Primary, roles dos nós (o *slave* primeiro).
- *ds*:UpToDate/UpToDate, estados dos discos sincronizados (o disco local primeiro).
- *nr*:10212, KB recibidos pola rede desde o *master*.
- *dw*:10212, KB escritos no disco local do *slave*.
- *oos*:0, KB que faltan por sincronizar.

NOTA: poden usarse tamén os seguintes comandos para consultar o estado de DRBD

```
$ sudo drbd-overview
$ sudo service drbd status
$ sudo systemctl status drbd
```

Preparación dos contidos do disco replicado

Vamos preparar unha páxina web que se almacenará no disco replicado. Os pasos son os seguintes:

33. Formata e monta o volume DRBD no *master*.

```
master$ sudo mkfs.ext4 /dev/drbd1
master$ sudo mount /dev/drbd1 /mnt
```

34. Comproba que se montou o volume.

```
master$ df -h /dev/drbd1
```

35. Copia a páxina web do *master* ao novo volume, modifícaa para distinguila da que xa tiñamos e desmonta o volume.

```
master$ sudo cp /var/www/html/index.html /mnt
master$ sudo nano /mnt/index.html
master$ sudo umount /mnt
```

Comprobación da configuración de DRBD

Vamos comprobar que a replicación do disco funciona correctamente. Os pasos son os seguintes:

36. Tenta montar o volume DRBD no *slave*.

```
slave$ sudo mount /dev/drbd1 /mnt
```

Ao ter o *slave* o rol de secundario, o volume non pode montarse e obtemos unha mensaxe de erro indicándonolo.

37. Cambia os roles primario e secundario do *master* e do *slave*.

```
master$ sudo drbdadm secondary xcpm1920-drbd1
master$ sudo drbd-overview

slave$ sudo drbdadm primary xcpm1920-drbd1
slave$ sudo drbd-overview
```

38. Monta de novo o volume DRBD no *slave*.

```
slave$ sudo mount /dev/drbd1 /mnt
slave$ ls /mnt
slave$ cat /mnt/index.html
```

Ao ter agora o *slave* o rol de primario, o volume móntase e podemos acceder ao seu contido.

39. Desmonta o volume DRBD do *slave* e recupera os roles orixinais do *master* e do *slave*.

Xustificación da segunda parte

Inclúe na memoria de xustificación desta práctica as capturas de pantalla que mostren a execución (comando e resultado) da seguinte secuencia de comandos:

NOTA: executa a secuencia de comandos partindo dun estado no que o *master* teña o rol primario e *slave* o secundario.

```
#Comproba o estado actual
master$ sudo drbdadm dump xcpm1920-drbd1
master$ cat /proc/drbd

#Cambia roles primario/secundario
master$ sudo drbdadm secondary xcpm1920-drbd1
slave$ sudo drbdadm primary xcpm1920-drbd1
slave$ cat /proc/drbd

#Monta o volume no slave
slave$ sudo mount /dev/drbd1 /mnt
slave$ cat /mnt/index.html

#Desmonta o volume e desfai o cambio de roles
slave$ sudo umount /mnt
slave$ sudo drbdadm secondary xcpm1920-drbd1
master$ sudo drbdadm primary xcpm1920-drbd1
master$ cat /proc/drbd
```

Explica BREVEMENTE o comportamento que observaches no cluster ao executar a secuencia de comandos.

TERCEIRA PARTE: Configuración de DRBD como recurso do cluster

Nesta terceira parte vamos engadir DRBD como recurso do cluster HA. Os pasos son os seguintes:

40. Desactiva o servizo DRBD no *master* e no *slave* e comproba que está parado. Isto faise para evitar que o servizo DRBD interfira co axente DRBD de Pacemaker. Así Pacemaker pasa a ter o control exclusivo para iniciar e parar o servizo DRBD.

```
$ systemctl stop drbd
```

- Exemplo para *master*:

```
vagrant@xcpm1920-master:~$ systemctl status drbd
• drbd.service - LSB: Control DRBD resources.
  Loaded: loaded (/etc/init.d/drbd; generated)
  Active: inactive (dead)
  Docs: man:systemd-sysv-generator(8)
vagrant@xcpm1920-master:~$ cat /proc/drbd
version: 8.4.10 (api:1/proto:86-101)
srcversion: 1FE4996664CF3ED2A7CD900
```

41. Configura o DRBD como recurso do cluster. No exemplo seguinte amósase como facelo usando o modo interactivo de *crm*.

- Crea unha copia da CIB para realizar as modificacións sobre ela.

```
$ sudo crm
crm(live)# cib new drbd
```

- Crea o recurso para o volume DRBD usando o axente de recurso *ocf:linbit:drbd*⁶.

```
crm(drbd)# configure primitive xcpm1920-webdata \
> ocf:linbit:drbd params drbd_resource=xcpm1920-drbd1 \
> op monitor timeout=20 interval=20 role=Slave \
> op monitor timeout=20 interval=10 role=Master
```

- Para permitir que se executen dúas copias do recurso DRBD, unha con rol primario e outra con rol secundario en nós diferentes do cluster, crea un recurso multiestado.

```
crm(drbd)# configure ms xcpm1920-webdataclone xcpm1920-webdata \
> meta master-max=1 master-node-max=1 \
> clone-max=2 clone-node-max=1 notify=true
```

Os recursos multiestado son un tipo especial de recurso que permite xestionar clons (instancias ou copias) dun recurso simple. Cada clon pode adoptar un de dous estados (roles ou modos) diferentes: *master* ou *slave*⁷. No noso caso o estado *master* equivale ao primario do recurso DRBD e o *slave* ao secundario. O significado das opcións utilizadas é o seguinte:

- *configure ms*, crear un recurso multiestado.
- *clone-max*, número máximo de clons do recurso no cluster.
- *clone-node-max*, número máximo de clons que poden iniciarse nun mesmo nó do cluster.

⁶ https://www.mankier.com/7/ocf_linbit_drbd

⁷ Aínda que os nomes coincidan, non confundilo cos nós do cluster.

- `master-max`, número máximo de clons que poden ter simultaneamente o rol de *master* no cluster.
- `master-node-max`, número máximo de clons que poden ter simultaneamente o rol de *master* nun mesmo nó do cluster.
- `notify`, notifica ao axente do recurso dos cambios no estado dos clons.
- Comproba que se creou o recurso multiestado e grava os cambios.

```
crm(drbd)# resource show
crm(drbd)# cib commit drbd
crm(drbd)# quit
```

- "Limpa" o estado do recurso para que se reinicie.

```
$ sudo crm resource cleanup xcpm1920-webdataclone
```

- Comproba a nova configuración.

```
$ sudo crm configure show
$ sudo crm status
$ cat /proc/drbd
```

- Exemplo para *master* (fixate en que agora DRBD está funcionando de novo, pero faino baixo o control de Pacemaker):

```
vagrant@xcpm1920-master:~$ sudo crm status
Stack: corosync
Current DC: xcpm1920-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Sat Oct 17 12:44:06 2020
Last change: Sat Oct 17 12:44:01 2020 by root via crm_resource on xcpm1920-master

3 nodes configured
4 resources configured

Online: [ xcpm1920-master xcpm1920-slave ]
Offline: [ xcpm1920-spare ]

Full list of resources:

xcpm1920-VIP (ocf::heartbeat:IPaddr2): Started xcpm1920-master
xcpm1920-webserver (ocf::heartbeat:apache): Started xcpm1920-master
Master/Slave Set: xcpm1920-webdataclone [xcpm1920-webdata]
Masters: [ xcpm1920-master ]
Slaves: [ xcpm1920-slave ]

vagrant@xcpm1920-master:~$ cat /proc/drbd
version: 8.4.10 (api:1/proto:86-101)
srcversion: 1FE4996664CF3ED2A7CD900

1: cs:StandAlone ro:Primary/Unknown ds:UpToDate/DUnknown r-----
   ns:0 nr:0 dw:0 dr:640 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

42. Para poder utilizar o volume DRBD temos tamén que configurar o sistema de ficheiros que se usará con el. No exemplo seguinte amósase como configurar un sistema de ficheiros como recurso do cluster usando de novo o modo interactivo de `crm`.

- Crea unha copia da CIB para realizar as modificacións sobre ela.

```
$ sudo crm
crm(live)# cib new fs
```

- Crea o recurso para o sistema de ficheiros usando o axente de recurso `ocf:heartbeat:Filesystem`⁸. Usaremos como punto de montaxe o directorio usado por defecto polo servidor Apache para servir a páxina web que creamos anteriormente no disco DRBD.

```
crm(fs)# configure primitive xcpm1920-webfs \
> ocf:heartbeat:Filesystem params device=/dev/drbd1 \
> directory=/var/www/html fstype=ext4
```

- Configura o emparellamento do sistema de ficheiros co clon DRBD que teña o rol de *master*. Configura ademais a orde de parada/reinicio: o recurso DRBD ten que iniciarse antes que o sistema de ficheiros.

```
crm(fs)# configure collocation xcpm1920-fs-on-drbd \
> inf: xcpm1920-webfs xcpm1920-webdataclone:Master
crm(fs)# configure order xcpm1920-fs-after-drbd \
> inf: xcpm1920-webdataclone:promote xcpm1920-webfs:start
```

- Configura tamén o emparellamento do sistema de ficheiros co servidor Apache e a orde de parada/reinicio: o sistema de ficheiros ten que iniciarse antes que o servidor Apache.

```
crm(fs)# configure collocation xcpm1920-web-with-fs \
> inf: xcpm1920-webserver xcpm1920-webfs
crm(fs)# configure order xcpm1920-web-after-fs \
> inf: xcpm1920-webfs xcpm1920-webserver
```

- Comproba a nova configuración e grava os cambios.

```
crm(fs)# configure show
crm(fs)# cib commit fs
crm(fsr)# quit
```

- Reinicia os servizos do cluster para partir dun estado "limpo" de erros.

```
$ sudo crm cluster stop
$ sudo crm cluster start
```

- Comproba o estado do cluster.

```
$ sudo crm status
```

⁸ https://www.mankier.com/7/ocf_heartbeat_Fileystem

- Exemplo para *master* (fixate en que o *master* ten o disco DRBD montado na ruta usada por defecto polo servidor Apache):

```
[vagrant@xcpm1920-master:~]$ sudo crm status
Stack: corosync
Current DC: xcpm1920-slave (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Sat Oct 17 14:56:06 2020
Last change: Sat Oct 17 14:56:04 2020 by root via crm_attribute on xcpm1920-master

2 nodes configured
5 resources configured

Online: [ xcpm1920-master xcpm1920-slave ]

Full list of resources:

xcpm1920-VIP (ocf::heartbeat:IPaddr2):          Started xcpm1920-master
xcpm1920-webserver (ocf::heartbeat:apache):      Started xcpm1920-master
Master/Slave Set: xcpm1920-webdataclone [xcpm1920-webdata]
    Masters: [ xcpm1920-master ]
    Slaves: [ xcpm1920-slave ]
xcpm1920-webfs (ocf::heartbeat:Filesystem):      Started xcpm1920-master

[vagrant@xcpm1920-master:~]$ df -h /dev/drbd1
Filesystem      Size  Used Avail Use% Mounted on
/dev/drbd1      992M  2,6M  923M   1% /var/www/html
```

Comprobación do funcionamento

Vamos comprobar que o cluster funciona consonte ao configurado.

43. Proba a cargar as páxinas dos nós usando as súas IPs (<http://193.168.1.2>, <http://193.168.1.3>) e a IP virtual (<http://193.168.1.10>).

Como estamos montando o volume DRBD en `/var/www/html`, o nó que teña o rol *master* (primario) servirá a páxina que creamos anteriormente nese volume, tanto se usamos a IP do nó como a IP virtual. O outro nó faiará ao non ter o servidor Apache levantado (lembra que o emparellamos co sistema de ficheiros, polo que Pacemaker só o manterá levantado no nó primario).

Vamos asumir que sexa o *master* o que está a servir a páxina desde a IP virtual.

44. Comproba que o *slave* non serve a páxina web usando a súa IP (<http://193.168.1.3>).
45. Para agora o *master* e comproba que o *slave* asume os recursos: que monta o volume DRBD (compróbaos co comando: `df -h`) e que serve a páxina web que almacenamos no volume DRBD usando a IP virtual (<http://193.168.1.10>) e a súa propia (<http://193.168.1.3>).
46. Volve arrincar o *master*. Comproba que non serve a páxina web usando a súa IP (<http://193.168.1.2>).
47. Para agora o *slave* e comproba que o *master* asume os recursos: que monta o volume DRBD e que volve servir a páxina web do volume DRBD na IP virtual (<http://193.168.1.10>).

Xustificación da terceira parte

Inclúe na memoria de xustificación desta práctica as capturas de pantalla que mostren a execución (comando e resultado) da seguinte secuencia de comandos:

NOTA: executa a secuencia de comandos partindo dun estado no que sexa o *master* o que serve a páxina desde a IP virtual.

```
#Comproba o estado de partida
$ sudo crm configure show
$ sudo crm status
$ curl 193.168.1.2
$ curl 193.168.1.3
$ curl 193.168.1.10

#Para e reinicia o master
$ sudo crm node standby xcpm1920-master
$ sudo crm node online xcpm1920-master

#Comproba o estado
$ sudo crm status
$ curl 193.168.1.2
$ curl 193.168.1.3
$ curl 193.168.1.10

#Para e reinicia o slave
$ sudo crm node standby xcpm1920-slave
$ sudo crm node online xcpm1920-slave

#Comproba o estado
$ sudo crm status
$ curl 193.168.1.2
$ curl 193.168.1.3
$ curl 193.168.1.10
```

Explica BREVEMENTE o comportamento que observaches no cluster ao executar a secuencia de comandos.