

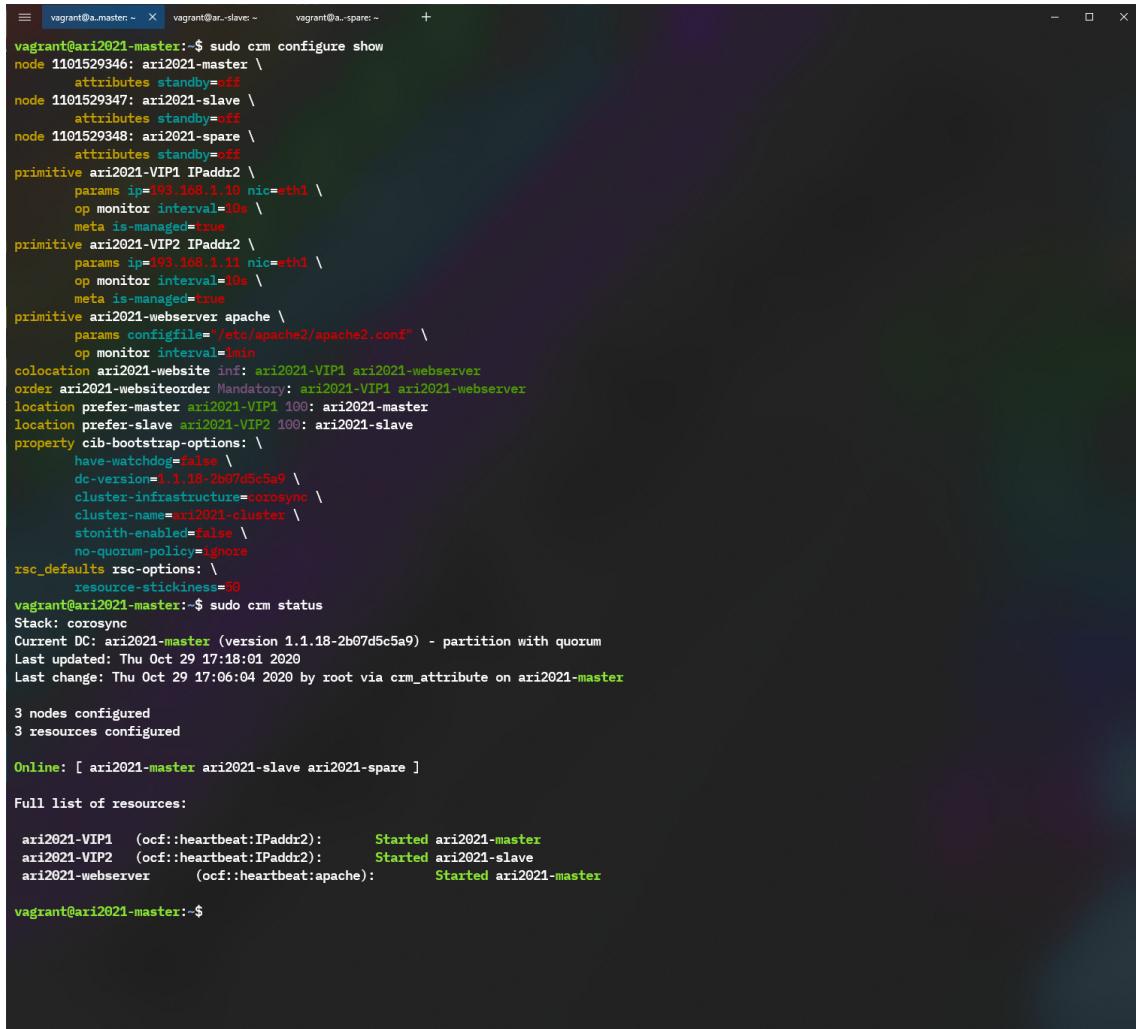
MEMORIA EII – PRÁCTICA II

Alonso Rodríguez Iglesias – EII 2020/21 – Práctica 2

JUSTIFICACIÓN DE LA PRIMERA PARTE

Se adjuntan capturas de la ejecución (en orden) de los siguientes comandos:

```
sudocrm configure show  
sudocrm status
```



```
vagrant@ari2021-master:~$ sudocrm configure show
node 1101529346: ari2021-master \
    attributes standby=off
node 1101529347: ari2021-slave \
    attributes standby=off
node 1101529348: ari2021-spare \
    attributes standby=off
primitive ari2021-VIP1 IPAddr2 \
    params ip=193.168.1.10 nic=eth1 \
    op monitor interval=10s \
    meta is-managed=true
primitive ari2021-VIP2 IPAddr2 \
    params ip=193.168.1.11 nic=eth1 \
    op monitor interval=10s \
    meta is-managed=true
primitive ari2021-webserver apache \
    params configfile="/etc/apache2/apache2.conf" \
    op monitor interval=1min
colocation ari2021-website inf: ari2021-VIP1 ari2021-webserver
order ari2021-websiteorder Mandatory: ari2021-VIP1 ari2021-webserver
location prefer-master ari2021-VIP1 100: ari2021-master
location prefer-slave ari2021-VIP2 100: ari2021-slave
property cib-bootstrap-options: \
    have-watchdog=false \
    dc-version=1.1.18-2b07d5c5a9 \
    cluster-infrastructure=corosync \
    cluster-name=ari2021-cluster \
    stonith-enabled=false \
    no-quorum-policy=ignore
rsc_defaults rsc-options: \
    resource-stickiness=50
vagrant@ari2021-master:~$ sudocrm status
Stack: corosync
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Thu Oct 29 17:18:01 2020
Last change: Thu Oct 29 17:06:04 2020 by root via cib_attribute on ari2021-master

3 nodes configured
3 resources configured

Online: [ ari2021-master ari2021-slave ari2021-spare ]

Full list of resources:

ari2021-VIP1 (ocf::heartbeat:IPAddr2):      Started ari2021-master
ari2021-VIP2 (ocf::heartbeat:IPAddr2):      Started ari2021-slave
ari2021-webserver (ocf::heartbeat:apache):     Started ari2021-master

vagrant@ari2021-master:~$
```

(Mostramos la configuración y el estado actual del cluster)

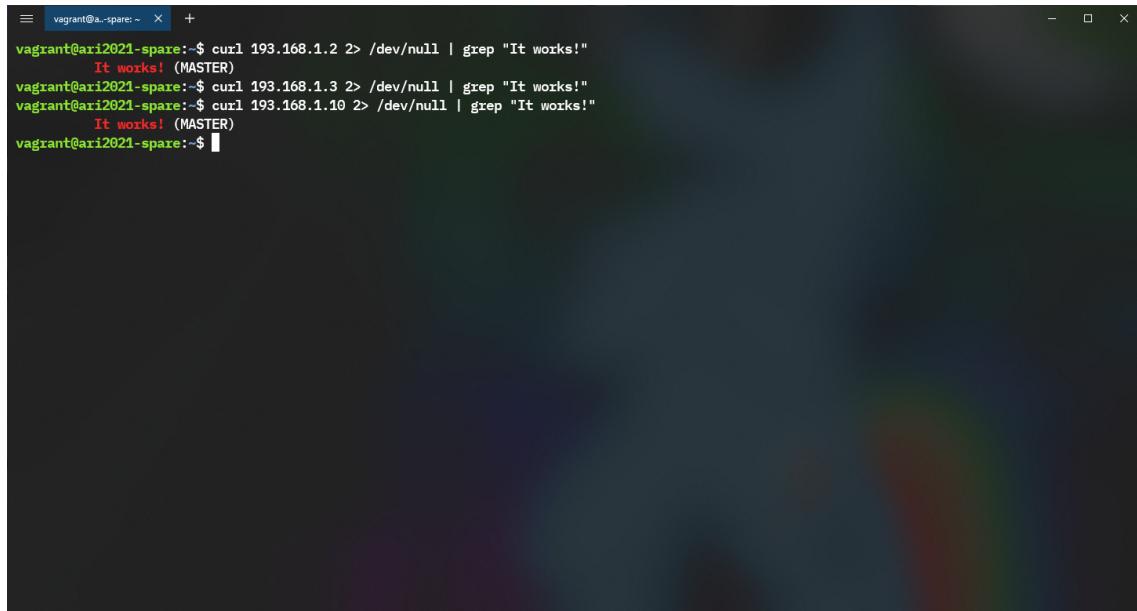
Una pequeña anotación sobre los siguientes comandos:

Inicialmente se hace un **curl** a cada IP, pero éste es un comando que muestra mucho output. Lo que he hecho han sido tres cosas:

- He modificado el **index.html** de apache, para mostrar al lado del **"It works!"**, entre paréntesis, si estamos accediendo al master o al slave
- Tras eso, he filtrado la salida de **curl** para que solo me muestre esa línea
- Como además me mostraba las stats, redirigí **stderr** a **/dev/null**.

Así, los siguientes tres comandos los he modificado como procede, y a partir de este primer ejemplo nos referiremos a ellos como **[3curl]**.

```
curl 193.168.1.2 2> /dev/null | grep "It works!"
curl 193.168.1.3 2> /dev/null | grep "It works!"
curl 193.168.1.10 2> /dev/null | grep "It works!"
```

A screenshot of a terminal window titled "vagrant@ari2021-spare:~". The window contains three lines of command-line text. Each line starts with "curl" followed by an IP address (193.168.1.2, 193.168.1.3, or 193.168.1.10). After the IP, there is a redirection operator (2>) followed by "/dev/null" and a pipe character (|). Finally, each line ends with "grep "It works!"" and a double quote. The output of the first two commands shows the text "It works! (MASTER)" on a new line, indicating they were run against the master node. The third command's output is not visible in the screenshot.

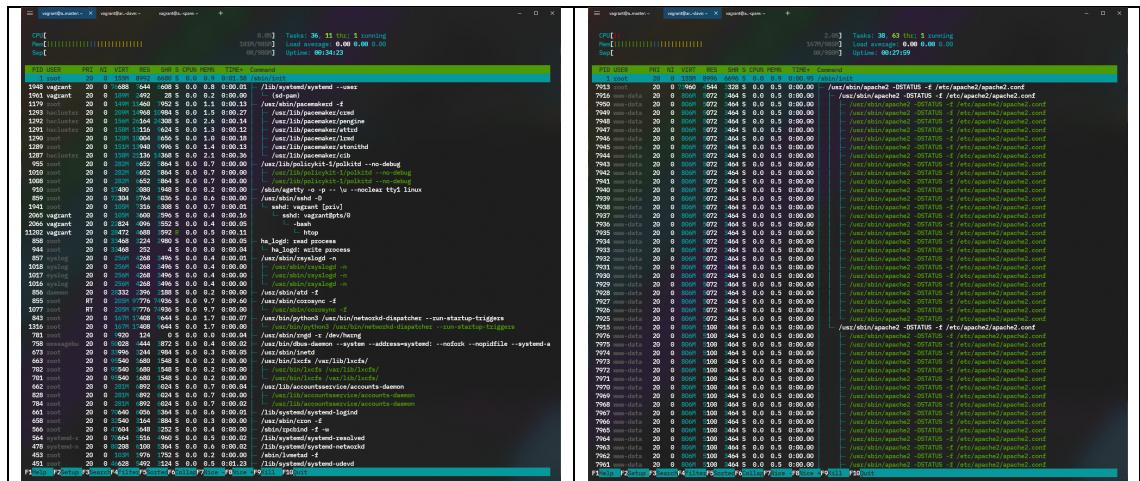
(Vemos que el servidor está alojado en master, y no está iniciado en el slave)

```
sudo crm node standby ari2021-master
[3curl]
```

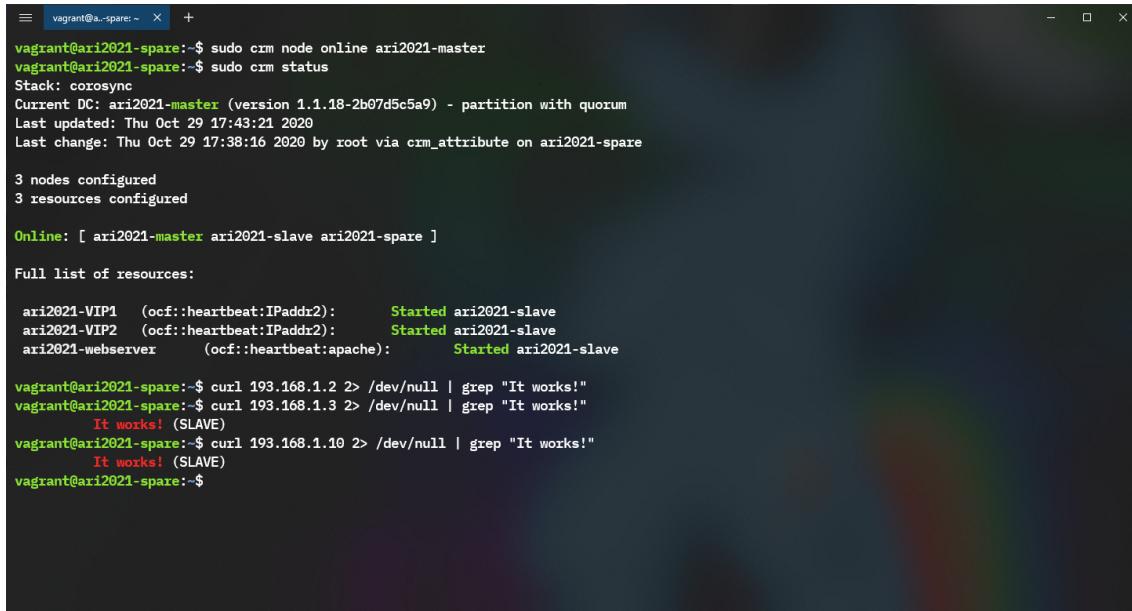
```
vagrant@ari2021-spares:~ % sudo crm node standby ari2021-master
vagrant@ari2021-spares:~ % curl 193.168.1.2 2> /dev/null | grep "It works!"
It works! (SLAVE)
vagrant@ari2021-spares:~ % curl 193.168.1.3 2> /dev/null | grep "It works!"
It works! (SLAVE)
vagrant@ari2021-spares:~ %
```

(El slave asume los recursos de master)

Además, con **htop** podemos comprobar que apache se ha parado en **master** y se ha iniciado en **slave**:



```
sudo crm node online ari2021-master
sudo crm status
[3curl]
```



```
vagrant@ari2021-spare:~$ sudo crm node online ari2021-master
vagrant@ari2021-spare:~$ sudo crm status
Stack: corosync
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Thu Oct 29 17:43:21 2020
Last change: Thu Oct 29 17:38:16 2020 by root via cmm_attribute on ari2021-spare

3 nodes configured
3 resources configured

Online: [ ari2021-master ari2021-slave ari2021-spare ]

Full list of resources:

ari2021-VIP1    (ocf::heartbeat:IPAddr2):     Started ari2021-slave
ari2021-VIP2    (ocf::heartbeat:IPAddr2):     Started ari2021-slave
ari2021-webserver (ocf::heartbeat:apache):   Started ari2021-slave

vagrant@ari2021-spare:~$ curl 193.168.1.2 2> /dev/null | grep "It works!"
vagrant@ari2021-spare:~$ curl 193.168.1.3 2> /dev/null | grep "It works!"
      It works! (SLAVE)
vagrant@ari2021-spare:~$ curl 193.168.1.10 2> /dev/null | grep "It works!"
      It works! (SLAVE)
vagrant@ari2021-spare:~$
```

(El servidor sigue corriendo en slave, ya que por la stickiness, no se mueve el recurso de vuelta al master cuando está online de nuevo)

```

sudo crm node standby ari2021-slave
sudo crm status
[3curl]

```

```

vagrant@ari2021-spares:~$ sudo crm node standby ari2021-slave
vagrant@ari2021-spares:~$ sudo crm status
Stack: corosync
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Thu Oct 29 17:45:10 2020
Last change: Thu Oct 29 17:45:07 2020 by root via cib_attribute on ari2021-spares

3 nodes configured
3 resources configured

Node ari2021-slave: standby
Online: [ ari2021-master ari2021-spares ]

Full list of resources:

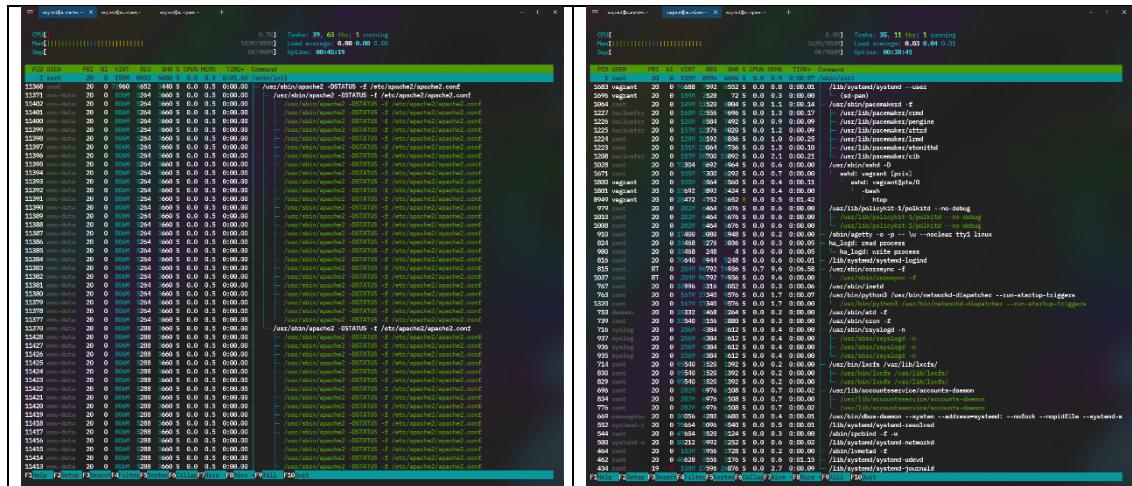
ari2021-VIP1 (ocf::heartbeat:IPaddr2): Started ari2021-master
ari2021-VIP2 (ocf::heartbeat:IPaddr2): Started ari2021-spares
ari2021-webserver (ocf::heartbeat:apache): Started ari2021-master

vagrant@ari2021-spares:~$ curl 193.168.1.2 2> /dev/null | grep "It works!"
It works! (MASTER)
vagrant@ari2021-spares:~$ curl 193.168.1.3 2> /dev/null | grep "It works!"
It works! (MASTER)
vagrant@ari2021-spares:~$ curl 193.168.1.10 2> /dev/null | grep "It works!"
It works! (MASTER)
vagrant@ari2021-spares:~$

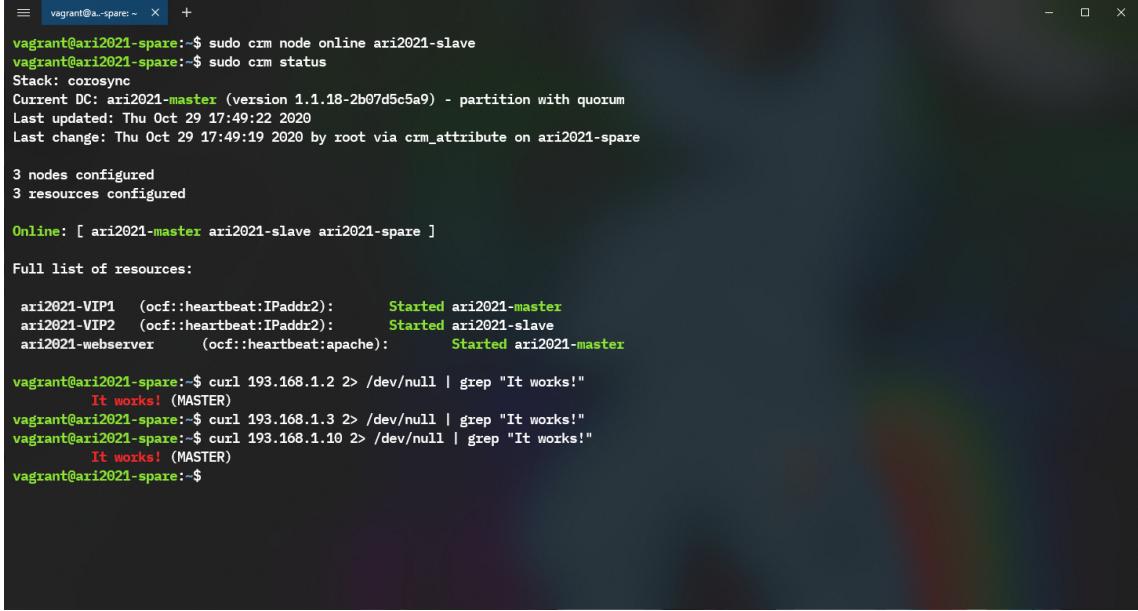
```

(Ahora el servicio si que cambia de servidor, ya que el slave ha ido a standby)

Además, de nuevo podemos comprobar los procesos en cada servidor con **htop**:



```
sudo crm node online ari2021-slave  
sudo crm status  
[3curl]
```



```
vagrant@ari2021-spare:~$ sudo crm node online ari2021-slave  
vagrant@ari2021-spare:~$ sudo crm status  
Stack: corosync  
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum  
Last updated: Thu Oct 29 17:49:22 2020  
Last change: Thu Oct 29 17:49:19 2020 by root via cmm_attribute on ari2021-spare  
  
3 nodes configured  
3 resources configured  
  
Online: [ ari2021-master ari2021-slave ari2021-spare ]  
  
Full list of resources:  
  
ari2021-VIP1 (ocf::heartbeat:IPaddr2): Started ari2021-master  
ari2021-VIP2 (ocf::heartbeat:IPaddr2): Started ari2021-slave  
ari2021-webserver (ocf::heartbeat:apache): Started ari2021-master  
  
vagrant@ari2021-spare:~$ curl 193.168.1.2 2> /dev/null | grep "It works!"  
    It works! (MASTER)  
vagrant@ari2021-spare:~$ curl 193.168.1.3 2> /dev/null | grep "It works!"  
vagrant@ari2021-spare:~$ curl 193.168.1.10 2> /dev/null | grep "It works!"  
    It works! (MASTER)  
vagrant@ari2021-spare:~$
```

(de nuevo, aunque volvamos a activar el slave, el servicio ya se encuentra en el master, y por stickiness, no va a cambiar de vuelva al slave)

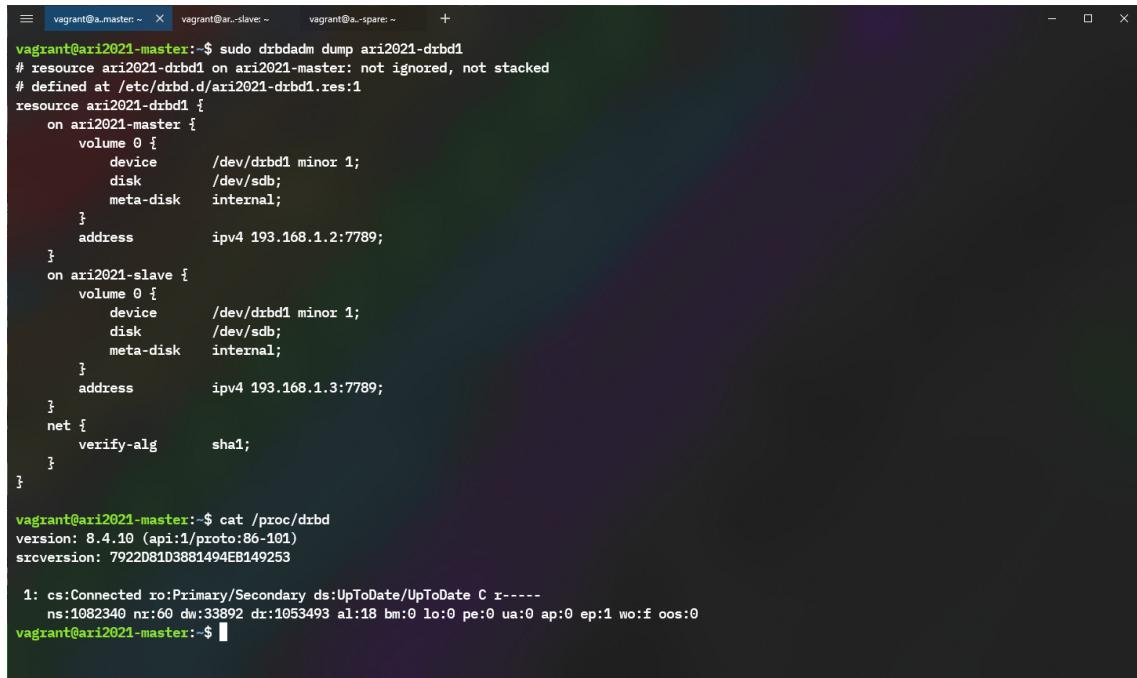
La principal conclusión de este apartado, es que el servidor se comporta como debería. Iniciando el servicio cuando es necesario, y deteniéndolo cuando no.

Además, la stickiness hace que no migremos el servicio mientras no sea estrictamente necesario.

JUSTIFICACIÓN DE LA SEGUNDA PARTE

Se adjuntan capturas de la ejecución (en orden) de los siguientes comandos:

```
sudo drbdadm dump ari2021-drbd1  
cat /proc/drbd
```



```
vagrant@ari2021-master:~$ sudo drbdadm dump ari2021-drbd1  
# resource ari2021-drbd1 on ari2021-master: not ignored, not stacked  
# defined at /etc/drbd.d/ari2021-drbd1.res:1  
resource ari2021-drbd1 {  
    on ari2021-master {  
        volume 0 {  
            device      /dev/drbd1 minor 1;  
            disk       /dev/sdb;  
            meta-disk  internal;  
        }  
        address     ipv4 193.168.1.2:7789;  
    }  
    on ari2021-slave {  
        volume 0 {  
            device      /dev/drbd1 minor 1;  
            disk       /dev/sdb;  
            meta-disk  internal;  
        }  
        address     ipv4 193.168.1.3:7789;  
    }  
    net {  
        verify-alg   sha1;  
    }  
}  
  
vagrant@ari2021-master:~$ cat /proc/drbd  
version: 8.4.10 (api:1/proto:86-101)  
srcversion: 7922D81D3881494EB149253  
  
1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----  
    ns:1082340 nr:60 dw:33892 dr:1053493 al:18 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0  
vagrant@ari2021-master:~$
```

(Mostramos la configuración actual de DRBD así como su estado)

```
sudo drbdadm secondary ari2021-drbd1  
sudo drbdadm primary ari2021-drbd1  
cat /proc/drbd
```



```
vagrant@ari2021-master:~$ sudo drbdadm secondary ari2021-drbd1  
vagrant@ari2021-master:~$  
  
vagrant@ari2021-slave:~$ sudo drbdadm primary ari2021-drbd1  
vagrant@ari2021-slave:~$ cat /proc/drbd  
version: 8.4.10 (api:1/proto:86-101)  
srcversion: 7922D81D3881494EB149253  
  
1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----  
    ns:60 nr:1082340 dw:1082400 dr:2389 al:10 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0  
vagrant@ari2021-slave:~$
```

(Vemos como se han cambiado los roles satisfactoriamente, ya que en el slave, aparece Primary/Secondary, refiriéndose a slave/master)

```
sudo mount /dev/drbd1 /mnt
ls -la /mnt
cat /mnt/index.html | grep "\[DRBD\]"
```

```
vagrant@a.mster: ~          vagrant@ar.-slave: ~          vagrant@a.-spare: ~      +
vagrant@ari2021-slave:~$ sudo mount /dev/drbd1 /mnt
vagrant@ari2021-slave:~$ ls -la /mnt
total 36
drwxr-xr-x  3 root root  4096 nov  5 13:21 .
drwxr-xr-x 24 root root  4096 oct 22 13:48 ..
-rw-r--r--  1 root root 10925 nov  5 13:21 index.html
drwx-----  2 root root 16384 nov  5 13:19 lost+found
vagrant@ari2021-slave:~$ cat /mnt/index.html | grep "\[DRBD\]"
It works! [DRBD]
vagrant@ari2021-slave:~$
```

(Montamos la primera y única partición del disco que maneja DRBD en /mnt, mostramos su contenido, y vemos que efectivamente, el archivo index.html se ha sincronizado satisfactoriamente desde master)

```
sudo umount /mnt
sudo drbdadm secondary ari2021-drbd1
sudo drbdadm primary ari2021-drbd1
cat /proc/drbd
```

```
vagrant@a.mster: ~          vagrant@ar.-slave: ~          vagrant@a.-spare: ~      +
vagrant@ari2021-slave:~$ sudo umount /mnt
vagrant@ari2021-slave:~$ sudo drbdadm secondary ari2021-drbd1
vagrant@ari2021-slave:~$
```



```
vagrant@a.mster: ~          vagrant@ar.-slave: ~          vagrant@a.-spare: ~      +
vagrant@ari2021-master:~$ sudo drbdadm primary ari2021-drbd1
vagrant@ari2021-master:~$ cat /proc/drbd
version: 8.4.10 (api:1/proto:86-101)
srcversion: 7922D81D3881494EB149253

1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
  ns:1082340 nr:132 dw:33964 dr:1054133 al:18 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
vagrant@ari2021-master:~$
```

(Desmontamos la partición de /mnt y hacemos el proceso contrario, es decir, poner al master como primario, y como vemos, marca ro:Primary/Secondary en master, así que todo ha salido como esperábamos)

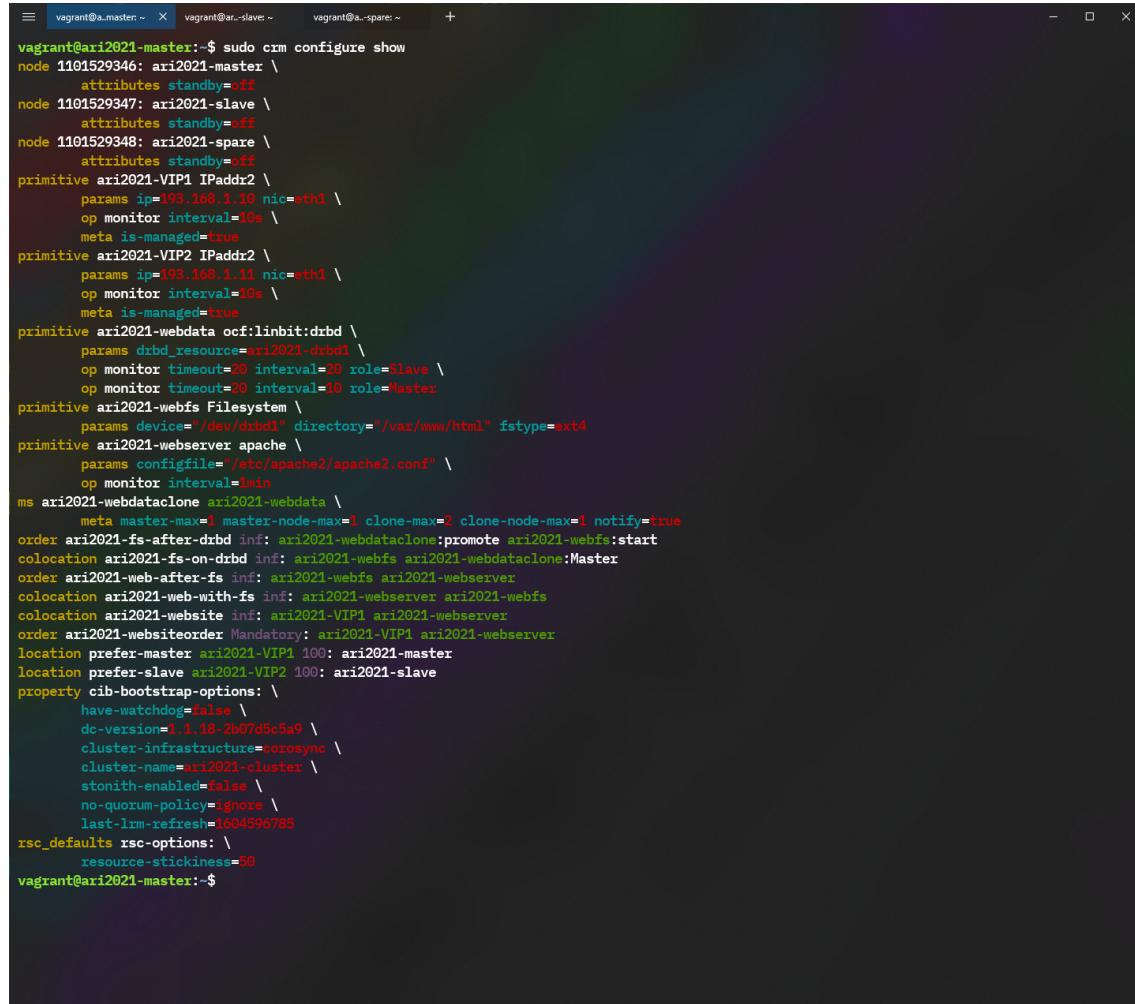
JUSTIFICACIÓN DE LA TERCERA PARTE

A parte de lo que se indica en la práctica, he realizado una acción preventiva, que ha sido mover el **index.html** original ubicado en **/var/www/html** a **~**, tanto en el master como en el slave, por si al montar la partición de DRBD nos diese algún error.

```
master$ sudo mv /var/www/html/index.html ~/index.html.master.bak  
slave$ sudo mv /var/www/html/index.html ~/index.html.slave.bak
```

Se adjuntan capturas de la ejecución (en orden) de los siguientes comandos:

sudo crm configure



```
vagrant@a2021-master:~$ sudo crm configure show  
node 1101529346: ari2021-master \  
    attributes standby=off  
node 1101529347: ari2021-slave \  
    attributes standby=off  
node 1101529348: ari2021-spare \  
    attributes standby=off  
primitive ari2021-VIP1 IPaddr2 \  
    params ip=193.168.1.10 nic=eth1 \  
    op monitor interval=10s \  
    meta is-managed=true  
primitive ari2021-VIP2 IPaddr2 \  
    params ip=193.168.1.11 nic=eth1 \  
    op monitor interval=10s \  
    meta is-managed=true  
primitive ari2021-webdata ocf:linbit:drbd \  
    params drbd_resource=ari2021-drbd1 \  
    op monitor timeout=20 interval=20 role=slave \  
    op monitor timeout=20 interval=10 role=Master  
primitive ari2021-webfs Filesystem \  
    params device="/dev/drbd1" directory="/var/www/html" fstype=ext4  
primitive ari2021-webserver apache \  
    params configfile="/etc/apache2/apache2.conf" \  
    op monitor interval=1min  
ms ari2021-webdataclone ari2021-webdata \  
    meta master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true  
order ari2021-fs-after-drbd inf: ari2021-webdataclone:promote ari2021-webfs:start  
colocation ari2021-fs-on-drbd inf: ari2021-webfs ari2021-webdataclone:Master  
order ari2021-web-after-fs inf: ari2021-webfs ari2021-webserver  
colocation ari2021-web-with-fs inf: ari2021-webserver ari2021-webfs  
colocation ari2021-website inf: ari2021-VIP1 ari2021-webserver  
order ari2021-websiteorder Mandatory: ari2021-VIP1 ari2021-webserver  
location prefer-master ari2021-VIP1 100: ari2021-master  
location prefer-slave ari2021-VIP2 100: ari2021-slave  
property cib-bootstrap-options: \  
    have-watchdog=false \  
    dc-version=1.1.18-2b07d5c5a0 \  
    cluster-infrastructure=corosync \  
    cluster-name=ari2021-cluster \  
    stonith-enabled=false \  
    no-quorum-policy=ignore \  
    last-lim-refresh=100045%785  
rsc_defaults rsc-options: \  
    resource-stickiness=50  
vagrant@a2021-master:$
```

(Miramos la configuración, podemos notar cómo ha crecido desde que hicimos la primera práctica :D)

```
sudo crm status
```

[3curl]

```
vagrant@ari2021-master:~ $ sudo crm status
Stack: corosync
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Thu Nov  5 17:54:18 2020
Last change: Thu Nov  5 17:45:21 2020 by root via cmm_attribute on ari2021-master

3 nodes configured
6 resources configured

Online: [ ari2021-master ari2021-slave ]
OFFLINE: [ ari2021-spare ]

Full list of resources:

ari2021-VIP1 (ocf::heartbeat:IPaddr2): Started ari2021-master
ari2021-VIP2 (ocf::heartbeat:IPaddr2): Started ari2021-slave
ari2021-webserver (ocf::heartbeat:apache): Started ari2021-master
Master/Slave Set: ari2021-webdataclone [ari2021-webdata]
  Masters: [ ari2021-master ]
  Slaves: [ ari2021-slave ]
ari2021-websfs (ocf::heartbeat:Filesystem): Started ari2021-master

vagrant@ari2021-master:$ curl 193.168.1.2 2> /dev/null | grep "It works!"
  It works! [DRBD]
vagrant@ari2021-master:$ curl 193.168.1.3 2> /dev/null | grep "It works!"
vagrant@ari2021-master:$ curl 193.168.1.10 2> /dev/null | grep "It works!"
  It works! [DRBD]
vagrant@ari2021-master:$ 
```

140 x 30

(Los recursos que utilizamos se encuentran originalmente en master, y así lo muestra tanto status, como el output de **curl**, tal como se pide en la justificación)

```
sudo crm node standby ari2021-master
sudo crm node online ari2021-master
sudo crm status
[3curl]
```

```
vagrant@ari2021-master:~ $ sudo crm node standby ari2021-master
vagrant@ari2021-master:~ $ sudo crm node online ari2021-master
vagrant@ari2021-master:~ $ sudo crm status
Stack: corosync
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Thu Nov  5 17:55:36 2020
Last change: Thu Nov  5 17:55:34 2020 by root via cmm_attribute on ari2021-master

3 nodes configured
6 resources configured

Online: [ ari2021-master ari2021-slave ]
OFFLINE: [ ari2021-spare ]

Full list of resources:

ari2021-VIP1 (ocf::heartbeat:IPaddr2): Started ari2021-slave
ari2021-VIP2 (ocf::heartbeat:IPaddr2): Started ari2021-slave
ari2021-webserver (ocf::heartbeat:apache): Started ari2021-slave
Master/Slave Set: ari2021-webdataclone [ari2021-webdata]
  Masters: [ ari2021-slave ]
  Slaves: [ ari2021-master ]
ari2021-websfs (ocf::heartbeat:Filesystem): Started ari2021-slave

vagrant@ari2021-master:$ curl 193.168.1.2 2> /dev/null | grep "It works!"
vagrant@ari2021-master:$ curl 193.168.1.3 2> /dev/null | grep "It works!"
  It works! [DRBD]
vagrant@ari2021-master:$ curl 193.168.1.10 2> /dev/null | grep "It works!"
  It works! [DRBD]
vagrant@ari2021-master:$ 
```

(Al apagar y volver a encender el master, se va al slave cuando se apaga, y se queda en él aún cuando se enciende, debido a la stickiness)

```

sudo crm node standby ari2021-slave
sudo crm node online ari2021-slave
sudo crm status
[3curl]

```

```

vagrant@ari2021-master:~$ sudo crm node standby ari2021-slave
vagrant@ari2021-master:~$ sudo crm node online ari2021-slave
vagrant@ari2021-master:~$ sudo crm status
Stack: corosync
Current DC: ari2021-master (version 1.1.18-2b07d5c5a9) - partition with quorum
Last updated: Thu Nov 5 17:56:41 2020
Last change: Thu Nov 5 17:56:36 2020 by root via crrm_attribute on ari2021-master

3 nodes configured
6 resources configured

Online: [ ari2021-master ari2021-slave ]
OFFLINE: [ ari2021-spares ]

Full list of resources:

ari2021-VIP1 (ocf::heartbeat:IPaddr2): Started ari2021-master
ari2021-VIP2 (ocf::heartbeat:IPaddr2): Started ari2021-slave
ari2021-webserver (ocf::heartbeat:apache): Started ari2021-master
Master/Slave Set: ari2021-webdataclone [ari2021-webdata]
  Masters: [ ari2021-master ]
  Slaves: [ ari2021-slave ]
ari2021-webfs (ocf::heartbeat:Filesystem): Started ari2021-master

vagrant@ari2021-master:~$ curl 193.168.1.2 2> /dev/null | grep "It works!"
  It works! [DRBD]
vagrant@ari2021-master:~$ curl 193.168.1.3 2> /dev/null | grep "It works!"
vagrant@ari2021-master:~$ curl 193.168.1.10 2> /dev/null | grep "It works!"
  It works! [DRBD]
vagrant@ari2021-master:~$
```

(Al apagar el slave, obtenemos el mismo resultado que antes, pero a la inversa: es decir, los recursos se migran de vuelta al master, ya que es el único online; y se quedan en él aunque encendamos el slave, debido, de nuevo, a la stickiness)

Realmente no tengo tanto que explicar acerca de esto, ya que queda bastante claro el qué hace y por qué, así pues, podemos concluir que el cluster se comporta como deseamos.

En este caso, que estamos tratando con sistemas ya un poco más pesados que el de la práctica anterior (tampoco tan pesados, pero en comparación...), ya que levantar apache es bastante más costoso que trasladar una IP virtual, que doy por hecho que se hará con el envío de un par de paquetes ARP, es decir, que virtualmente no tiene coste alguno.

Alonso Rodriguez Iglesias. 5-Noviembre-2020