

# ALTA DISPONIBILIDAD CON ISSABEL

La Receta para configuración de cluster de HA  
con Issabel PBX



**JUAN OLIVA**  
**SECURITY CONSULTANT**



**[PRIMERA EDICIÓN]**  
**Agosto de 2017**



Copyright (c) 2017

Esta obra está licenciada bajo la Licencia **Creative Commons**  
Atribución-NoComercial-CompartirIgual 3.0 Unported. Para ver una  
Copia de esta licencia, visite:  
<http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Si luego de leerla todavía tiene alguna duda acerca de esta licencia,  
envíe una carta a Creative Commons, 171 2nd Street, Suite 300, San  
Francisco, California, 94105, USA.

Modificaciones y sugerencias de la obra a [joliva@silcom.com.pe](mailto:joliva@silcom.com.pe)

**Primera Edición**

**Dedicado para Angélica y Sebastián**

## 1. Agradecimiento

El desarrollo de este documento no hubiera sido posible, sin el apoyo y confianza de:

Proyecto Issabel **www.issabel.org** por permitirme ser parte de esta grandiosa nueva comunidad.

A mi esposa y mi hijo, que gracias a su paciencia y cariño proporcionan la inspiración para seguir adelante.

## 2. Acerca del autor

**Juan Oliva Córdova**  
**Jefe de Proyectos e Ingeniería**  
**Silcom VoIP Security Assessment**

Acerca de SILCOM, empresa peruana dedicada a seguridad informática y Telefonía IP. La cual cuenta con más de 15 años de experiencia en el mercado, desarrollando proyectos para empresas nacionales y del extranjero.

Juan, es especialista en seguridad informática a nivel senior, ha realizado pruebas de penetración y Ethical Hacking para entidades del gobierno y del sector financiero, así como en entidades nacionales y del extranjero.

Se encuentra especializado en hacking de servicios de Voz sobre IP (VoIP) donde ha conseguido mostrar las graves vulnerabilidades que existen respecto a estos elementos. Ponente de eventos nacionales e internacionales desde hace varios años mantiene un blog personal <http://jroliva.wordpress.com> el cual se ha convertido un punto de consulta obligatorio, sobre temas seguridad informática, cuenta con certificaciones vigentes en Hacking y Linux.

### 3. Introducción

El objetivo del presente documento es guiar de manera detallada en la receta para la instalación y configuración de un cluster de alta disponibilidad en modo activo - pasivo sobre la plataforma de comunicaciones Issabel PBX.

#### Acerca de Issabel PBX

Es una plataforma de comunicaciones Software Libre y código abierto basada en Asterisk (Digium the Asterisk Company) integra funcionalidades de PBX, correo electrónico, tareas de colaboración, así como video llamadas.

Todas las instrucciones del presente material fueron realizadas con la versión que corresponde al archivo ISO `issabel4-USB-DVD-x86_64-20170621.iso`.

Juan Oliva  
SILCOM

## Índice de Contenidos

1.	Agradecimiento .....	5
2.	Acerca del autor.....	6
3.	Introducción.....	7
4.	Conceptos, escenario y esquema de funcionamiento. ....	1
5.	Instalar y configurar DRBD .....	2
6.	Crear el CLUSTER para automatizar alta disponibilidad .....	6
7.	Crear recurso para el uso de IP Flotante.....	7
8.	Crear recurso para el uso de DRBD .....	7
9.	Crear recurso para el uso de MariaDB .....	9
10.	Crear recurso para el uso de ASTERISK .....	11
11.	Crear recurso para el uso para APACHE.....	13
12.	Referencias y Bibliografía.....	16



**[PAGINA DEJADA INTENCIONALMENTE EN BLANCO]**

# RECETA PARA LA INSTALACION Y CONFIGURACION DE CLUSTER DE ALTA DIPONIBILIDAD CON ISSABEL PBX

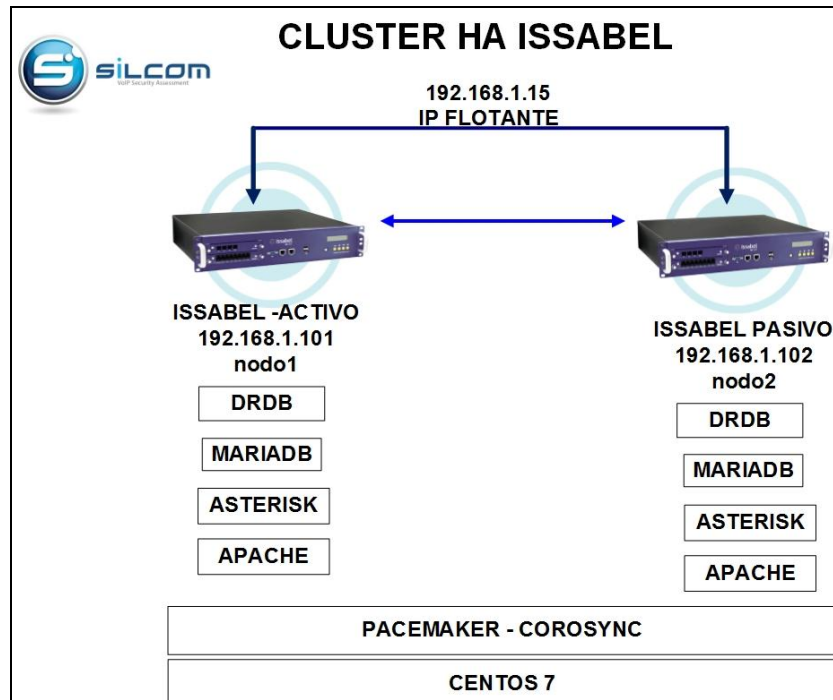
## 4. Conceptos, escenario y esquema de funcionamiento.

En escenarios donde la continuidad del negocio es un factor clave, contar alta disponibilidad para servicios que se consideran críticos dentro de una organización es sumamente importante. En este caso concreto, vamos a ver la implementar la alta disponibilidad para la plataforma de comunicaciones Issabel PBX.

Para ello, se instalará y configurará un cluster en modo activo pasivo, si bien es cierto la plataforma Issabel, es sumamente estable como para confiar toda la carga a un solo recurso, estas no son ajenas a fallas de otro tipo como factores, eléctricos, climáticos, de hardware, etc. Es por ello en caso de que uno de las plataformas fallase por los factores ya mencionados, dejaría sin comunicaciones a la organización, por lo tanto, es crítico tener el servicio en alta redundancia de forma que si un servidor falla, otro asume el servicio de manera transparente.

El software que hemos usado para tener alta disponibilidad se llama Pacemaker así como corosync, los cuales reemplazan al muy conocido heartbeat, el cual se encuentra descontinuado para Centos 7 que es el sistema operativo base de Issabel.

El escenario de funcionamiento a realizar seria el siguiente:



El esquema de funcionamiento a realizar sería el siguiente:

- Dos servidores Issabel PBX, virtuales o físicos.
- Cada servidor con 10 GB de disco duro
- Esquema de particionamiento , 1GB SWAP , 6GB Partición raíz , 4GB libres para DRDB
- Nombres para cada servidor, nodo1.local y nodo2.local

## 5. Instalar y configurar DRBD

Configurar las direcciones IP estáticas

192.168.1.101, Nodo1

192.168.1.102, Nodo2

Configurar los nombres de HOST en ambos nodos

The screenshot shows the 'Network Parameters' configuration page in a web interface. The 'Host (Ex. host.example.com):' field is set to 'nodo1.local'. The 'Default Gateway:' field is set to '192.168.1.1'. The 'Primary DNS:' field is set to '192.168.1.1'. The 'Secondary DNS:' field is empty. There are 'Save' and 'Cancel' buttons at the top left. A 'Required field' indicator is visible at the top right. The footer text reads 'Issabel is licensed under GPL. 2006 - 2017.'

The screenshot shows the 'Network Parameters' configuration page in a web interface. The 'Host (Ex. host.example.com):' field is set to 'nodo2.local'. The 'Default Gateway:' field is set to '192.168.1.1'. The 'Primary DNS:' field is set to '192.168.1.1'. The 'Secondary DNS:' field is empty. There are 'Save' and 'Cancel' buttons at the top left. A 'Required field' indicator is visible at the top right. The footer text reads 'Issabel is licensed under GPL. 2006 - 2017.'

Configurar la resolución de nombres en ambos nodos

```
vim /etc/hosts
192.168.1.101  nodo1.local
192.168.1.102  nodo2.local
```

Instalar DRBD en ambos nodos

```
rpm -ivh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
yum --skip-broken -y install drbd*-utils kmod-drbd*
```

Configurar la carga del módulo en ambos nodos

```
vim /etc/modules-load.d/drbd.conf
drbd
```



```
reboot
```

Verificar la carga del módulo DRDB en ambos nodos

```
lsmod | grep drbd
```

Resultado:

```
drbd          405376  0
libcrc32c     12644  1 drbd
```

Crear la partición en ambos nodos

```
fdisk /dev/sda
```

Command (m for help): **n**

Command action

e extended

p primary partition (1-4)

**p**

Partition number (3,4, default 3): **[Enter]**

First sector (12584960-20971519, default 12584960): **[Enter]**

Last sector, +sectors or +size{K,M,G} (12584960-20971519, default 20971519): **[Enter]**

Using default value 20971519

Partition 3 of type Linux and of size 4 GiB is set

Command (m for help): **w**

Reiniciar

```
reboot
```

Formatear la nueva partición en ambos nodos

```
mke2fs -j /dev/sda3
```

```
dd if=/dev/zero bs=1M count=500 of=/dev/sda3; sync
```

Configurar DRDB en el nodo1

```
vi /etc/drbd.d/testdata1.res
```

```
resource testdata1 {
protocol C;
on nodo1.local {
    device /dev/drbd0;
    disk /dev/sda3;
    address 192.168.1.101:7788;
    meta-disk internal;
}
on nodo2.local {
    device /dev/drbd0;
    disk /dev/sda3;
    address 192.168.1.102:7788;
    meta-disk internal;
}
}
```

Copiar el archivo testdata1.res al nodo2

```
scp /etc/drbd.d/testdata1.res root@192.168.1.102:/etc/drbd.d/
```

Iniciar DRDB META DATA STORAGE en ambos nodos

```
[root@nodo1 ~] drbdadm create-md testdata1
[root@nodo2 ~] drbdadm create-md testdata1
```

Iniciar el servicio de DRDB en ambos nodos

```
systemctl start drbd
systemctl enable drbd
```

Configurar el nodo1 como DRDB primario

```
drbdadm primary testdata1
drbdadm primary testdata1 --force
```

Verificar

```
[root@nodo1 drbd.d]#
[root@nodo1 drbd.d]#
[root@nodo1 drbd.d]# cat /proc/drbd
version: 8.4.9-1 (api:1/proto:86-101)
GIT-hash: 9976da086367a2476503ef7f6b13d4567327a280 build by akemi@Build64R7, 2016-12-04 01:08:48
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:4193116 nr:0 dw:0 dr:4194028 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
[root@nodo1 drbd.d]#
[root@nodo1 drbd.d]#
```

Formatear la partición DRBD

```
[root@node1 ~]# mkfs.ext4 /dev/drbd0
```

Probar funcionamiento de la partición DRBD

Crear un archivo en la partición DRDB

```
[root@node1 ~]# mount /dev/drbd0 /mnt  
[root@node1 ~]# touch /mnt/prueba1.txt  
[root@node1 ~]# ls /mnt/  
[root@node1 ~]# umount /mnt
```

Hacer que el nodo1 sea el secundario

```
[root@node1 ~] drbdadm secondary testdata1
```

Hacer que el nodo2 sea el primario, luego montar la partición y verificar el archivo prueba2.txt

```
[root@node2 ~] drbdadm primary testdata1  
[root@node2 ~] mount /dev/drbd0 /mnt  
[root@node2 ~] ls /mnt  
[root@node2 ~] umount /mnt
```

Hacer que el nodo2 sea secundario nuevamente

```
[root@node2 ~] drbdadm secondary testdata1
```

Hacer que el nodo1 sea primario nuevamente

```
[root@node1 ~] drbdadm primary testdata1
```

Verificar DRDB

```
drbd-overview
```

```
[root@node1 drbd.d]#  
[root@node1 drbd.d]# drbd-overview  
0:testdata1/0 Connected Primary/Secondary UpToDate/UpToDate  
[root@node1 drbd.d]#  
[root@node1 drbd.d]#
```

## 6. Crear el CLUSTER para automatizar alta disponibilidad

Instalar Corosync y Pacemaker en ambos servidores

```
yum -y install corosync pcs pacemaker
```

Cambiar la contraseña del usuario hacluster en ambos nodos

```
passwd hacluster  
Password:
```

Iniciar el servicio PCS en ambos nodos

```
systemctl start pcsd
```

Configurar el inicio de los servicios en ambos nodos

```
systemctl enable pcsd.service  
systemctl start pcsd.service  
systemctl enable corosync.service  
systemctl enable pacemaker.service
```

Configurar y autenticar los nodos, realizar solo en el nodo1

```
pcs cluster auth nodo1.local nodo2.local  
Username: hacluster  
Password:
```

Crear el cluster y configurar parámetros, realizar solo en el nodo1

```
pcs cluster setup --name cluster_voip nodo1.local nodo2.local  
pcs cluster start --all  
pcs cluster enable --all  
pcs property set stonith-enabled=false  
pcs property set no-quorum-policy=ignore
```

## 7. Crear recurso para el uso de IP Flotante

Crear recurso virtual\_ip para IP Flotante

```
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=192.168.1.15 cidr_netmask=32 op monitor interval=30s on-fail=restart
```

```
pcs cluster cib drbd_cfg  
pcs cluster cib-push drbd_cfg
```

## 8. Crear recurso para el uso de DRBD

Crear recurso DRBD y crear la replicación para los nodos

```
pcs -f drbd_cfg resource create DrbdData ocf:linbit:drbd drbd_resource=testdata1 op monitor interval=60s  
pcs -f drbd_cfg resource master DrbdDataClone DrbdData master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true
```

```
pcs status resources  
pcs cluster cib-push drbd_cfg  
pcs status resources
```

```
[root@nod01 ~]#  
[root@nod01 ~]# pcs status resources  
virtual_ip (ocf::heartbeat:IPaddr2): Started nod01.local  
Master/Slave Set: DrbdDataClone [DrbdData]  
Masters: [ nod01.local ]  
Slaves: [ nod02.local ]  
[root@nod01 ~]#  
[root@nod01 ~]#
```

Crear recurso FILESYSTEM para el punto de montaje automatizado

```
pcs cluster cib fs_cfg  
pcs -f fs_cfg resource create DrbdFS Filesystem device="/dev/drbd0" directory="/mnt/" fstype="ext4"  
pcs -f fs_cfg constraint colocation add DrbdFS with DrbdDataClone INFINITY with-rsc-role=Master  
pcs -f fs_cfg constraint order promote DrbdDataClone then start DrbdFS  
pcs -f fs_cfg constraint colocation add DrbdFS with virtual_ip INFINITY  
pcs -f fs_cfg constraint order virtual_ip then DrbdFS  
pcs cluster cib-push fs_cfg  
pcs status resources
```

```
[root@nod01 ~]#  
[root@nod01 ~]# pcs status resources  
virtual_ip (ocf::heartbeat:IPaddr2): Started nod01.local  
Master/Slave Set: DrbdDataClone [DrbdData]  
Masters: [ nod01.local ]  
Slaves: [ nod02.local ]  
DrbdFS (ocf::heartbeat:Filesystem): Started nod01.local  
[root@nod01 ~]#  
[root@nod01 ~]#  
[root@nod01 ~]#
```



Verificar que el punto de montaje está montado en /mnt/

```
df -h
```

Para probar el funcionamiento del CLUSTER apagar el nodo1 y el nodo2 deberá pasar la IP virtual y el punto de montaje /mnt/ al nodo2 Luego apagar el nodo2 y todo el control pasara al nodo1 volviendo el punto de montaje a este de manera automática.

Para ver los logs de funcionamiento, visualizar con: tail -f /var/log/messages

## 9. Crear recurso para el uso de MariaDB

Desactivar MariaDB como servicio en ambos nodos

```
amportal stop
systemctl stop mariadb
systemctl disable mariadb
```

En el Nodo1

```
mkdir /mnt/mysql
mkdir /mnt/mysql/data
cd /mnt/mysql
cp -aR /var/lib/mysql/* /mnt/mysql/data
ls data/
mv /etc/my.cnf /mnt/mysql/
ln -s /mnt/mysql/my.cnf /etc/
```

Luego cambiar la directiva de DATADIR de my.cnf

```
vim /etc/my.cnf
datadir=/mnt/mysql/data
```

Reiniciar el nodo1 para que el nodo2 sea primario de drbd

```
reboot
```

Configurar solo en Servidor Secundario

```
pcs status resources
rm -rf /etc/my.cnf
ln -s /mnt/mysql/my.cnf /etc/
```

Verifica que el archivo my.cnf sea el mismo que editamos anteriormente

```
vim /etc/my.cnf
datadir=/mnt/mysql/data
```

Reiniciar el nodo2 para que el nodo1 sea primario de drbd

```
reboot
```

En el nodo1



```
pcs status resources
pcs resource create mysql ocf:heartbeat:mysql binary="/usr/bin/mysqld_safe" config="/etc/my.cnf" datadir="/mnt/mysql/data" \
pid="/var/lib/mysql/mysql.pid" socket="/var/lib/mysql/mysql.sock" \
additional_parameters="--bind-address=0.0.0.0" op start timeout=60s op stop timeout=60s op monitor interval=20s timeout=30s on-fail=standby
pcs cluster cib fs_cfg
pcs cluster cib-push fs_cfg
pcs -f fs_cfg constraint colocation add mysql with virtual_ip INFINITY
pcs -f fs_cfg constraint order DrbdFS then mysql
pcs cluster cib-push fs_cfg
pcs status resources
```

```
[root@nodo1 ~]#
[root@nodo1 ~]#
[root@nodo1 ~]#
[root@nodo1 ~]# pcs status resources
virtual ip (ocf::heartbeat:IPaddr2): Started nodo1.local
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nodo1.local ]
Slaves: [ nodo2.local ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo1.local
mysql (ocf::heartbeat:mysql): Started nodo1.local
[root@nodo1 ~]#
[root@nodo1 ~]#
[root@nodo1 ~]#
[root@nodo1 ~]#
```

Luego reiniciar el nodo y posteriormente el otro hasta que la sincronización este correcta.

reboot

```
[root@nodo2 ~]#
[root@nodo2 ~]# pcs status resources
virtual ip (ocf::heartbeat:IPaddr2): Started nodo2.local
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nodo2.local ]
Stopped: [ nodo1.local ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo2.local
mysql (ocf::heartbeat:mysql): Started nodo2.local
[root@nodo2 ~]#
[root@nodo2 ~]#
[root@nodo2 ~]#
[root@nodo2 ~]#
[root@nodo2 ~]#
```

Nota: Es posible que deba reiniciar varias veces ambos nodos para que se complete la sincronización automática del servicio.

## 10. Crear recurso para el uso de ASTERISK

Desactivar Asterisk como servicio en ambos nodos

```
service asterisk stop
systemctl disable asterisk
```

Descargar archivos de configuración del recurso en ambos nodos

```
cd /usr/lib/ocf/resource.d/heartbeat
wget https://raw.githubusercontent.com/ClusterLabs/resource-agents/master/heartbeat/asterisk
chmod 755 asterisk
```

Agregar recurso Asterisk al cluster, en el nodo1

```
pcs resource create asterisk ocf:heartbeat:asterisk params user="root" group="root" op monitor timeout="30"
pcs cluster cib fs_cfg
pcs cluster cib-push fs_cfg
pcs -f fs_cfg constraint colocation add asterisk with virtual_ip INFINITY
pcs -f fs_cfg constraint order mysql then asterisk
pcs cluster cib-push fs_cfg
pcs status resources
```

```
[root@nod01 mnt]#
[root@nod01 mnt]#
[root@nod01 mnt]# pcs status resources
virtual ip (ocf::heartbeat:IPAddr2): Started nod01.local
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nod01.local ]
Slaves: [ nod02.local ]
DrbdFS (ocf::heartbeat:Filesystem): Started nod01.local
mysql (ocf::heartbeat:mysql): Started nod01.local
asterisk (ocf::heartbeat:asterisk): Started nod01.local
[root@nod01 mnt]#
[root@nod01 mnt]#
[root@nod01 mnt]#
```

Copiar carpetas y archivos la partición DRBD en el nodo1

```
cd /mnt/
tar -zcvf var-asterisk.tgz /var/log/asterisk
tar -zcvf var-lib-asterisk.tgz /var/lib/asterisk
tar -zcvf usr-lib64-asterisk.tgz /usr/lib64/asterisk
tar -zcvf var-spool-asterisk.tgz /var/spool/asterisk
tar -zcvf etc-asterisk.tgz /etc/asterisk
```

```
tar xvfz var-asterisk.tgz
tar xvfz var-lib-asterisk.tgz
tar xvfz usr-lib64-asterisk.tgz
tar xvfz var-spool-asterisk.tgz
tar xvfz etc-asterisk.tgz
```

```
rm -rf /var/log/asterisk
rm -rf /var/lib/asterisk
rm -rf /usr/lib64/asterisk/
rm -rf /var/spool/asterisk/
rm -rf /etc/asterisk
ln -s /mnt/var/log/asterisk /var/log/asterisk
ln -s /mnt/var/lib/asterisk /var/lib/asterisk
ln -s /mnt/usr/lib64/asterisk /usr/lib64/asterisk
ln -s /mnt/var/spool/asterisk /var/spool/asterisk
ln -s /mnt/etc/asterisk /etc/asterisk
```

Configurar enlaces simbólicos en el nodo2

```
rm -rf /var/log/asterisk
rm -rf /var/lib/asterisk
rm -rf /usr/lib64/asterisk/
rm -rf /var/spool/asterisk/
rm -rf /etc/asterisk
```

```
ln -s /mnt/var/log/asterisk /var/log/asterisk
ln -s /mnt/var/lib/asterisk /var/lib/asterisk
ln -s /mnt/usr/lib64/asterisk /usr/lib64/asterisk
ln -s /mnt/var/spool/asterisk /var/spool/asterisk
ln -s /mnt/etc/asterisk /etc/asterisk
```

Reiniciar el nodo1 y confirmar que el nodo2 carga el servicio de asterisk

```
[root@nodo2 ~]#
[root@nodo2 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo2.local
Master/Slave Set: DrbdDataClone [DrbdData]
  Masters: [ nodo2.local ]
  Stopped: [ nodol.local ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo2.local
mysql (ocf::heartbeat:mysql): Started nodo2.local
asterisk (ocf::heartbeat:asterisk): Started nodo2.local
[root@nodo2 ~]#
[root@nodo2 ~]#
[root@nodo2 ~]#
[root@nodo2 ~]#
```

## 11. Crear recurso para el uso para APACHE

Desactivar APACHE como servicio en ambos nodos

```
service httpd stop
systemctl disable httpd
```

Agregar esta página en el servidor Apache para comprobar el estado del servicio en ambos nodos.

```
vim /etc/httpd/conf.d/server-status.conf
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

Comentar la línea RewriteRule del archivo issabel.conf para que soporte HTTP

```
vim /etc/httpd/conf.d/issabel.conf
#RewriteRule (.*) https://% {HTTP_HOST}% {REQUEST_URI}
```

Agregar recurso APACHE al cluster, en el nodo1

```
pcs resource create Httpd apache configfile="/etc/httpd/conf/httpd.conf" statusurl="http://127.0.0.1/server-status"
pcs cluster cib fs_cfg
pcs cluster cib-push fs_cfg
pcs -f fs_cfg constraint colocation add Httpd with virtual_ip INFINITY
pcs -f fs_cfg constraint order asterisk then Httpd
pcs cluster cib-push fs_cfg
pcs status resources
```

```
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nod01.local
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nod01.local ]
Slaves: [ nod02.local ]
DrbdFS (ocf::heartbeat:Filesystem): Started nod01.local
mysql (ocf::heartbeat:mysql): Started nod01.local
asterisk (ocf::heartbeat:asterisk): Started nod01.local
Httpd (ocf::heartbeat:apache): Started nod01.local
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
```

Copiar carpetas y archivos la partición DRBD en el nodo1

```
cd /mnt/  
tar -zcvf var-www.tgz /var/www  
tar -zcvf var-log-httpd.tgz /var/log/httpd
```

```
tar xvfz var-www.tgz  
tar xvfz var-log-httpd.tgz
```

```
rm -rf /var/www/  
rm -rf /var/log/httpd  
ln -s /mnt/var/www /var/www  
ln -s /mnt/var/log/httpd /var/log/httpd
```

Configurar enlaces simbólicos en el nodo2

```
rm -rf /var/www/  
rm -rf /var/log/httpd  
ln -s /mnt/var/www /var/www  
ln -s /mnt/var/log/httpd /var/log/httpd
```

Reiniciar el nodo1 y confirmar que el nodo2 carga el servicio de APACHE

```
[root@nodo2 ~]#  
[root@nodo2 ~]#  
[root@nodo2 ~]#  
[root@nodo2 ~]# pcs status resources  
virtual ip      (ocf::heartbeat:IPaddr2):      Started nodo2.local  
Master/Slave Set: DrbdDataClone [DrbdData]  
  Masters: [ nodo2.local ]  
  Stopped: [ nodo1.local ]  
DrbdFS (ocf::heartbeat:Filesystem):      Started nodo2.local  
mysql (ocf::heartbeat:mysql): Started nodo2.local  
asterisk      (ocf::heartbeat:asterisk):      Started nodo2.local  
Httpd (ocf::heartbeat:apache):      Started nodo2.local  
[root@nodo2 ~]#  
[root@nodo2 ~]#  
[root@nodo2 ~]#  
[root@nodo2 ~]#
```

## Validar recursos creados

```
[root@nod01 ~]#
[root@nod01 ~]# pcs resource show DrbdFS
Resource: DrbdFS (class=ocf provider=heartbeat type=Filesystem)
Attributes: device=/dev/drbd0 directory=/mnt/ fstype=ext4
Operations: start interval=0s timeout=60 (DrbdFS-start-interval-0s)
            stop interval=0s timeout=60 (DrbdFS-stop-interval-0s)
            monitor interval=20 timeout=40 (DrbdFS-monitor-interval-20)
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]# pcs resource show mysql
Resource: mysql (class=ocf provider=heartbeat type=mysql)
Attributes: binary=/usr/bin/mysqld safe config=/etc/my.cnf datadir=/mnt/mysql/data pid=/var/lib/mysql/mysql.pid socket=/var/lib/mysql/mysql.sock additional_parameters=-bind-address=0.0.0.0
Operations: promote interval=0s timeout=120 (mysql-promote-interval-0s)
            demote interval=0s timeout=120 (mysql-demote-interval-0s)
            start interval=0s timeout=60s (mysql-start-interval-0s)
            stop interval=0s timeout=60s (mysql-stop-interval-0s)
            monitor interval=20s on-fail=standby timeout=30s (mysql-monitor-interval-20s)
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]# pcs resource show asterisk
Resource: asterisk (class=ocf provider=heartbeat type=asterisk)
Attributes: user=root group=root
Operations: start interval=0s timeout=20 (asterisk-start-interval-0s)
            stop interval=0s timeout=20 (asterisk-stop-interval-0s)
            monitor interval=60s timeout=30 (asterisk-monitor-interval-60s)
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]# pcs resource show Httpd
Resource: Httpd (class=ocf provider=heartbeat type=apache)
Attributes: configfile=/etc/httpd/conf/httpd.conf statusurl=http://127.0.0.1/server-status
Operations: start interval=0s timeout=40s (Httpd-start-interval-0s)
            stop interval=0s timeout=60s (Httpd-stop-interval-0s)
            monitor interval=10 timeout=20s (Httpd-monitor-interval-10)
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
[root@nod01 ~]#
```



## 12. Referencias y Bibliografía

### **Proyecto Issabel**

<http://www.issabel.org>

### **Asterisk**

<http://www.asterisk.org>

### **Juan Oliva Blog**

<http://jroliva.net>

### **Pacemaker y corosync**

[http://clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Pacemaker\\_Explained/s-resource-ordering.html](http://clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Pacemaker_Explained/s-resource-ordering.html)

<http://www.alexlinux.com/mysql-active-active-cluster-with-pacemaker/>

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Configuring\\_the\\_Red\\_Hat\\_High\\_Availability\\_Add-On\\_with\\_Pacemaker/s1-multistateresource-HAAR.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Configuring_the_Red_Hat_High_Availability_Add-On_with_Pacemaker/s1-multistateresource-HAAR.html)

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-apache-active-passive-cluster-using-pacemaker-on-centos-7>