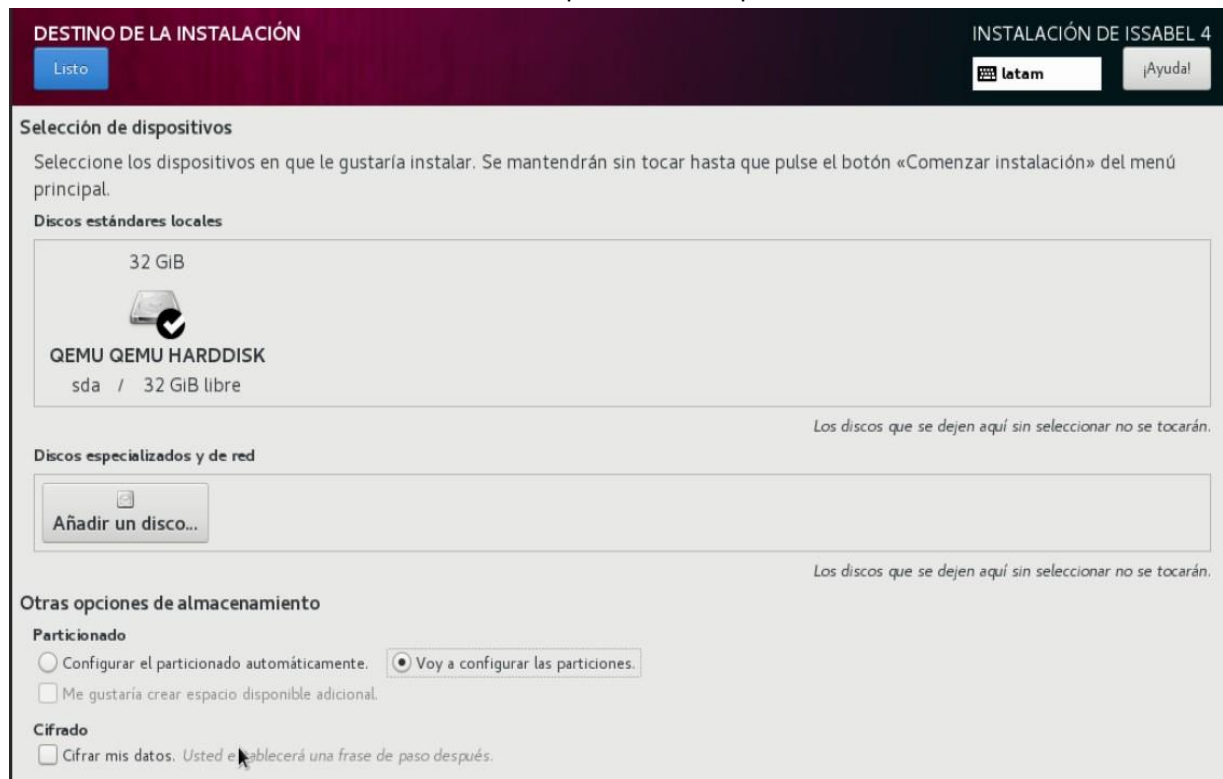


Configuración de Alta Disponibilidad (High Availability – HA) en Issabel

Entorno: Dos máquinas virtuales en Proxmox, cargados con el último ISO de Issabel y con dos interfaces de red cada una. Accedemos por la consola de las máquinas virtuales.

En este apartado no se tratará el tema de la configuración inicial de Issabel, como **Idioma del teclado**, **Configuración de red**, entre otros. La versión de Asterisk elegida es la versión 11.

En **Destino de la Instalación** se debe seleccionar el dispositivo en el que se va a instalar Issabel:



DESTINO DE LA INSTALACIÓN INSTALACIÓN DE ISSABEL 4


Listo latam ¡Ayuda!

Selección de dispositivos

Seleccione los dispositivos en que le gustaría instalar. Se mantendrán sin tocar hasta que pulse el botón «Comenzar instalación» del menú principal.


Discos estándares locales

32 GiB

 **QEMU QEMU HARDDISK**
sda / 32 GiB libre

Los discos que se dejen aquí sin seleccionar no se tocarán.

Discos especializados y de red

 **Añadir un disco...**

Los discos que se dejen aquí sin seleccionar no se tocarán.

Otras opciones de almacenamiento

Particionado


☐ Configurar el particionado automáticamente. ☒ Voy a configurar las particiones.

☐ Me gustaría crear espacio disponible adicional.

Cifrado

☐ Cifrar mis datos. Usted establecerá una frase de paso después.

En este caso será el disco duro previamente configurado y tendremos que configurar las particiones de manera manual:



PARTICIONADO MANUAL

Listo

Nueva Issabel 4 instalación

Usted ha creado todavía ningún punto de montaje para su instalación de Issabel 4. Puede:

- [Haga clic aquí para crearlos automáticamente.](#)
- Crear nuevos puntos de montaje pulsando el botón '+'.
Nuevos puntos de montaje utilizarán el siguiente esquema de particiones:

Partición estándar

+ - ↻

ESPACIO DISPONIBLE
32 GiB

ESPACIO TOTAL
32 GiB

La partición debe ser de la siguiente manera:

sda1→boot

sda2→swap

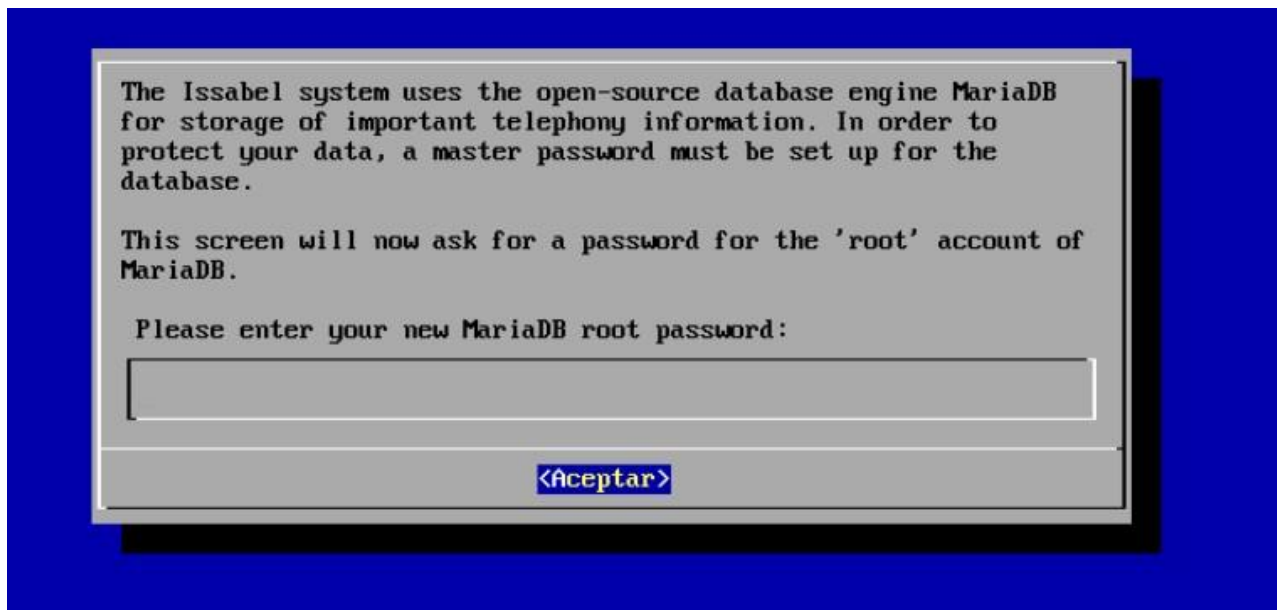
sda3→/

Nota: En servidores Lenovo, se debe realizar cuatro particiones: /boot/efi, /boot, swap, /.

Nota: El particionado tiene que ser manual porque debemos dejar espacio en disco para el punto de montaje, tema que veremos más adelante.

Luego de ello se procede a iniciar la instalación.

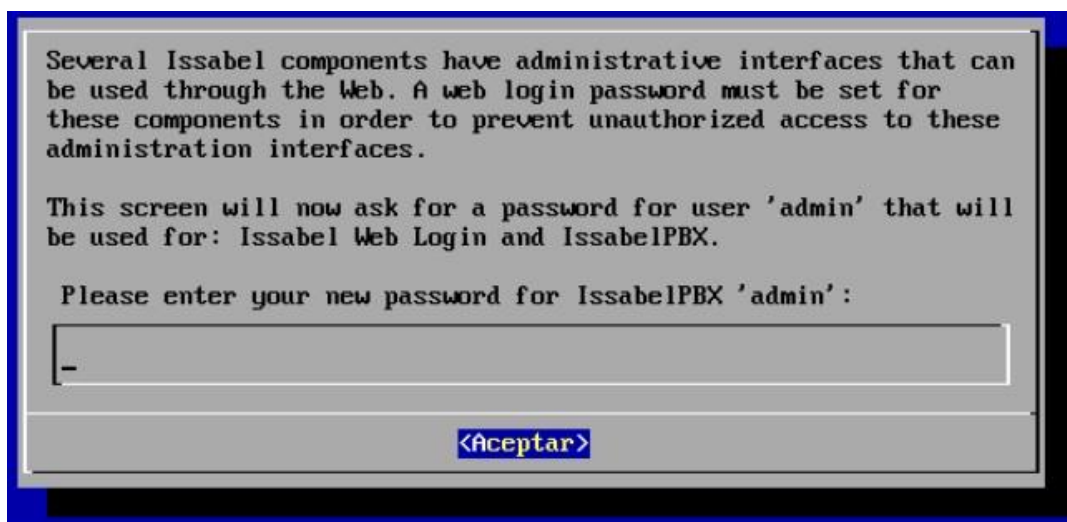
Culminada ello, aparecerán ventanas para colocar contraseñas. La primera es referente a la base de datos **MariaDB**:



Se coloca nuevamente la contraseña para reconfirmar:



Ahora, una ventana para la configuración de contraseña para el acceso web:



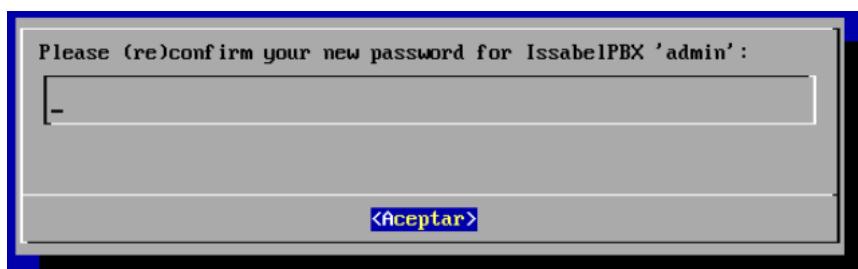
Several Issabel components have administrative interfaces that can be used through the Web. A web login password must be set for these components in order to prevent unauthorized access to these administration interfaces.

This screen will now ask for a password for user 'admin' that will be used for: Issabel Web Login and IssabelPBX.

Please enter your new password for IssabelPBX 'admin':

<Aceptar>

Colocamos nuevamente la contraseña para reconfirmar:



Please (re)confirm your new password for IssabelPBX 'admin':

<Aceptar>

Y aparecerá la siguiente información en la consola:

```
Issabel 4
Kernel 3.10.0-1062.el7.x86_64 on an x86_64

issabel login: root
Password:
Last login: Thu Feb  3 18:33:01 on

000  Issabel is a product meant to be configured through a web browser.
000  Any changes made from within the command line may corrupt the system
000  configuration and produce unexpected behavior; in addition, changes
0    made to system files through here may be lost when doing an update.

To access your Issabel System, using a separate workstation (PC/MAC/Linux)
Open the Internet Browser using the following URL:

Your opportunity to give back: http://www.patreon.com/issabel

System load:  0.00 (1min) 0.04 (5min) 0.05 (15min)      Uptime:   10 min
Asterisk:    Asterisk 11.25.3                        Active Calls: 0
Memory:      [====>-----] 13% 262/1998M
Usage on /:  [====>-----] 14% 2,5/20G
Swap usage:  0.0%
SSH logins:  1 open sessions
Processes:   121 total, 83 yours

[root@issabel ~]# _
```

Hasta aquí, los servidores no tienen IP. Recordar que hay más de una interfaz.

Ejecutamos **vim /etc/sysconfig/network-scripts/ifcfg-eth0** y configuramos una IP (como ejemplo configuramos 10.100.48.191). Verificamos que esté **UP** la interfaz y que tenga salida a Internet ejecutando un **ping 8.8.8.8**.

Hacemos lo mismo para la interfaz **eth1** y colocamos la IP 20.20.20.10 (esta interfaz será exclusivamente para la Alta Disponibilidad).

Apuntamos a la IP 10.100.48.191 en el browser y deberá aparecer lo siguiente:



Ingresamos con las credenciales previamente creadas, y vamos a **System**→**Network**→**Network Parameters**:

Vemos la configuración por defecto y que puede ser modificada.

Ahora, en la consola ejecutamos **vim /etc/hosts**:

```
127.0.0.1    localhost localhost.localdomain .localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1    localhost issabel.local
```

Vemos que el nombre de host por defecto es **localhost** y está asociado a la IP del loopback **127.0.0.1**. Lo que queremos hacer en este archivo es que el nuevo nombre de host se asocie a la IP de la interfaz para el HA.

Regresando a la web, damos click en **Edit Network Parameters** y cambiamos el nombre de host a **nodo1.flx** y colocamos el **Primary DNS 8.8.8.8**.

Le damos en **Save**. Retornamos a la consola para ver **/etc/hosts** y veremos que el nuevo nombre de host se asocia a la IP 127.0.0.1. Eso debe ser modificado. Desasociamos el nombre de host nodo1.flx al loopback y lo asociamos a la IP 20.20.20.10, que es para el HA:

```
127.0.0.1    localhost localhost.localdomain
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
127.0.0.1    localhost localhost.localdomain
20.20.20.10  nodo1.flx
```

Estos pasos previos tenemos que realizarlo en el otro servidor, el cual tendrá las IPs 10.100.48.192 y 20.20.20.11 para cada interfaz.

Respecto a los nombres de host, finalmente debe quedar de la siguiente manera:

Servidor 1 (10.100.48.191)

```
20.20.20.10 nodo1.flx
20.20.20.11 nodo2.flx
```

Servidor 2 (10.100.48.192)

```
20.20.20.11 nodo2.flx
20.20.20.10 nodo1.flx
```

Ahora, comenzaremos con la configuración del HA.

1. Instalación y Configuración de DRBD

Es importante validar que en el archivo `/etc/resolv.conf` esté configurado el DNS 8.8.8.8. De no ser así, hay que insertarlo en el script.

Observación: El DNS puede variar, dependiendo del escenario en donde se desee implementar el HA. Lo importante es que los equipos físicos o máquinas virtuales tengan salida a Internet.

Actualizar el Kernel en ambos nodos

El DRBD necesita instalar unos módulos en el kernel y que funcione correctamente, lo que se debe hacer es actualizar el Kernel mediante los repos de Centos, simplemente edita el archivo:

```
/etc/yum.repos.d/CentOS-Base.repo
```

Busca las líneas "exclude=" y quita la palabra kernel* esto en todas las líneas de ese repo oficial, te debe de quedar así:

```
exclude=redhat-logos php53*
```

Guardas y ejecutas "yum -y update" espera que se actualice el sistema y el kernel, reinicias y luego tratas de instalar el DRBD

Instalar los paquetes FOP2 y el Asternic en ambos nodos

```
yum -y install fop2 asternic-ccstatspro
```

Esto es en caso se tenga adquirido tales productos. Caso contrario, ignorar este paso.

Nota: Las licencias del FOP2 y del Asternic no pueden estar dentro del punto de montaje.

Importar llave pública e Instalar repositorio EPEL en ambos nodos

```
rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org
```

```
rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
```

Instalar los paquetes de DRBD en ambos nodos

```
yum -y install drbd84-utils kmod-drbd84
```

Cargar el módulo DRBD en ambos nodos

Dentro del archivo `/etc/modules-load.d/drbd.conf` insertar:

```
drbd
```

Reiniciar ambos nodos

reboot

Verificamos el módulo DRBD cargado

```
lsmod | grep drbd
drbd                397041  0
libcrc32c           12644  1 drbd
```

Crear la partición en ambos nodos

```
[root@nodo1 ~]# fdisk /dev/sda
Welcome to fdisk (util-linux 2.23.2).
```

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Orden (m para obtener ayuda): **n**

Partition type:

p primary (3 primary, 0 extended, 1 free)

e extended

Select (default e): **p**

Selected partition 4

Primer sector (21602304-31457279, valor predeterminado 21602304): **[ENTER]**

Se está utilizando el valor predeterminado 21602304

Last sector, +sectors or +size{K,M,G} (21602304-31457279, valor predeterminado 31457279): **[ENTER]**

Se está utilizando el valor predeterminado 31457279

Partition 4 of type Linux and of size 4,7 GiB is set

Orden (m para obtener ayuda): **w**

¡Se ha modificado la tabla de particiones!

Llamando a ioctl() para volver a leer la tabla de particiones.

WARNING: Re-reading the partition table failed with error 16: Dispositivo o recurso ocupado.

The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)

Se están sincronizando los discos.

Reiniciar ambos nodos

reboot

Formatear la nueva partición en ambos nodos

```
[root@nodo1 ~]# mke2fs -j /dev/sda4 → Este valor numérico dependerá de cuantas particiones se hayan configurado
mke2fs 1.42.9 (28-Dec-2013)
```

Discarding device blocks: hecho

Etiqueta del sistema de ficheros=

OS type: Linux

Tamaño del bloque=4096 (bitácora=2)

Tamaño del fragmento=4096 (bitácora=2)

Stride=0 blocks, Stripe width=0 blocks

308256 inodes, 1231872 blocks

61593 blocks (5.00%) reserved for the super user

Primer bloque de datos=0

Número máximo de bloques del sistema de ficheros=1262485504

38 bloque de grupos

32768 bloques por grupo, 32768 fragmentos por grupo

8112 nodos-i por grupo

Respaldo del superbloque guardado en los bloques:

32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: hecho

Escribiendo las tablas de nodos-i: hecho

Creating journal (32768 blocks): hecho

Escribiendo superbloques y la información contable del sistema de ficheros: hecho

```
[root@nodo1 ~]# dd if=/dev/zero bs=1M count=500 of=/dev/sda4; sync
```

500+0 registros leídos

500+0 registros escritos

524288000 bytes (524 MB) copiados, 4,66537 s, 112 MB/s

Luego, en ambos nodos, agregar en el archivo **/etc/drbd.d/global_common.conf** lo siguiente:

```
net {
    after-sb-0pri discard-zero-changes;
    after-sb-1pri discard-secondary;
    after-sb-2pri disconnect;
}
```

Configurar DRBD en el nodo1

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

```
resource testdata1 {
    protocol C;
    startup { wfc-timeout 5; degr-wfc-timeout 3; }
    on nodo1.flx {
        device /dev/drbd0;
        disk /dev/sda4;
        address 20.20.20.10:7788;
        meta-disk internal;
    }
    on nodo2.flx {
        device /dev/drbd0;
        disk /dev/sda4;
        address 20.20.20.11:7788;
        meta-disk internal;
    }
}
```

Copiar el archivo testdata1.res al nodo2

```
[root@nodo1 ~]# scp /etc/drbd.d/testdata1.res root@10.100.48.192:/etc/drbd.d/
```

The authenticity of host 10.100.48.192 (10.100.48.192)' can't be established.

ECDSA key fingerprint is SHA256:u4hZ1le14TH064JVIX+RzyHgsS4dIDgqY/qa9MTgVcE.

ECDSA key fingerprint is MD5:88:1e:00:bf:41:fa:cc:e7:3c:b6:53:9d:be:7f:af:e6.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.100.48.192' (ECDSA) to the list of known hosts.

root@10.100.48.192's password:

```
testdata1.res 100% 320 879.8KB/s 00:00
```

Iniciar DRBD META DATA STORAGE en ambos nodos

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

--== Thank you for participating in the global usage survey ==--

The server's response is:

you are the 2059th user to install this version

initializing activity log

initializing bitmap (152 KB) to all zero
Writing meta data...
New drbd meta data block successfully created.
success

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

--== Thank you for participating in the global usage survey ==--
The server's response is:

you are the 2060th user to install this version
initializing activity log
initializing bitmap (152 KB) to all zero
Writing meta data...
New drbd meta data block successfully created.
Success

Iniciar el servicio DRBD en ambos nodos

```
[root@nodo1 ~]# systemctl start drbd  
[root@nodo1 ~]# systemctl enable drbd  
Created symlink from /etc/systemd/system/multi-user.target.wants/drbd.service to  
/usr/lib/systemd/system/drbd.service.
```

Configurar el nodo1 como DRBD primario

```
[root@nodo1 ~]# drbdadm primary testdata1  
0: State change failed: (-2) Need access to UpToDate data  
Command 'drbdsetup-84 primary 0' terminated with exit code 17  
[root@nodo1 ~]# drbdadm primary testdata1 --force  
[root@nodo1 ~]#
```

Verificar el proceso de sincronización de los nodos

```
[root@nodo1 ~]# cat /proc/drbd  
version: 8.4.11 (api:1/proto:86-101)  
srcversion: FC3433D849E3B88C1E7B55C  
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----  
ns:4927300 nr:0 dw:0 dr:4929404 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

Nota: Este proceso de sincronización puede demorar algunos minutos. Lo mostrado es el proceso ya finalizado.

Formatear la partición DRBD en el nodo1

```
[root@nodo1 ~]# mkfs.ext4 /dev/drbd0  
mke2fs 1.42.9 (28-Dec-2013)  
Discarding device blocks: hecho  
Etiqueta del sistema de ficheros=  
OS type: Linux  
Tamaño del bloque=4096 (bitácora=2)  
Tamaño del fragmento=4096 (bitácora=2)  
Stride=0 blocks, Stripe width=0 blocks  
308256 inodes, 1231825 blocks  
61591 blocks (5.00%) reserved for the super user  
Primer bloque de datos=0  
Número máximo de bloques del sistema de ficheros=1262485504  
38 bloque de grupos  
32768 bloques por grupo, 32768 fragmentos por grupo  
8112 nodos-i por grupo  
Respaldo del superbloque guardado en los bloques:  
32768, 98304, 163840, 229376, 294912, 819200, 884736
```


Allocating group tables: hecho
Escribiendo las tablas de nodos-i: hecho
Creating journal (32768 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Probar funcionamiento de la partición DRBD

Crear un archivo en la partición DRBD

```
[root@nodo1 ~]# mount /dev/drbd0 /mnt  
[root@nodo1 ~]# touch /mnt/prueba1.txt  
[root@nodo1 ~]# ls /mnt/  
[root@nodo1 ~]# umount /mnt
```

Hacer que el nodo1 sea el secundario

```
[root@nodo1 ~]# drbdadm secondary testdata1
```

Hacer que el nodo2 sea el primario, luego montar la partición y verificar el archivo *prueba1.txt*

```
[root@nodo2~]# drbdadm primary testdata1  
[root@nodo2~]# mount /dev/drbd0 /mnt  
[root@nodo2~]# ls /mnt  
[root@nodo2~]# umount /mnt
```

Al ejecutar "ls /mnt" en el nodo2, debe aparecer el archivo prueba1.txt

Hacer que el nodo2 sea secundario nuevamente

```
[root@nodo2~]# drbdadm secondary testdata1
```

Hacer que el nodo1 sea primario nuevamente

```
[root@nodo1 ~]# drbdadm primary testdata1
```

Verificar DRBD en ambos nodos

```
[root@nodo1 /]# drbdadm status  
testdata1 role:Primary  
disk:UpToDate  
peer role:Secondary  
replication:Established peer-disk:UpToDate
```

```
[root@nodo2 ~]# drbdadm status  
testdata1 role:Secondary  
disk:UpToDate  
peer role:Primary  
replication:Established peer-disk:UpToDate
```

Se muestra el estado correcto de los nodos. Y así se culmina la configuración del DRBD.

2. Crear Cluster para alta disponibilidad

Instalar Corosync y Pacemaker en ambos servidores

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

Cambiar la contraseña del usuario hacluster en ambos nodos

```
[root@nodo1 ~]# passwd hacluster
```

Cambiando la contraseña del usuario hacluster.

Nueva contraseña:

Vuelva a escribir la nueva contraseña:

passwd: todos los símbolos de autenticación se actualizaron con éxito.

Importante: Misma contraseña para ambos nodos.

Iniciar el servicio PCS en ambos nodos

```
systemctl start pcsd
```

Configurar el inicio de los servicios en ambos nodos

```
[root@nodo1 ~]# systemctl enable pcsd.service
```

Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to /usr/lib/systemd/system/pcsd.service.

```
[root@nodo1 ~]# systemctl start pcsd.service
```

```
[root@nodo1 ~]# systemctl enable corosync.service
```

Created symlink from /etc/systemd/system/multi-user.target.wants/corosync.service to /usr/lib/systemd/system/corosync.service.

```
[root@nodo1 ~]# systemctl enable pacemaker.service
```

Created symlink from /etc/systemd/system/multi-user.target.wants/pacemaker.service to /usr/lib/systemd/system/pacemaker.service.

Configurar y autenticar los nodos. Realizar solo en el nodo1

```
[root@nodo1 ~]# pcs cluster auth nodo1.flx nodo2.flx
```

Username: hacluster

Password:

nodo2.flx: Authorized

nodo1.flx: Authorized

Crear el cluster y configurar parámetros. Realizar solo en el nodo1

```
[root@nodo1 ~]# pcs cluster setup --name cluster_voip nodo1.flx nodo2.flx
```

Destroying cluster on nodes: nodo1.flx, nodo2.flx...

nodo1.flx: Stopping Cluster (pacemaker)...

nodo2.flx: Stopping Cluster (pacemaker)...

nodo1.flx: Successfully destroyed cluster

nodo2.flx: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to 'nodo1.flx', 'nodo2.flx'
nodo1.flx: successful distribution of the file 'pacemaker_remote authkey'
nodo2.flx: successful distribution of the file 'pacemaker_remote authkey'
Sending cluster config files to the nodes...
nodo1.flx: Succeeded
nodo2.flx: Succeeded

Synchronizing pcsd certificates on nodes nodo1.flx, nodo2.flx...
nodo2.flx: Success
nodo1.flx: Success
Restarting pcsd on the nodes in order to reload the certificates...
nodo2.flx: Success
nodo1.flx: Success

[root@nodo1 ~]# pcs cluster start --all
nodo1.flx: Starting Cluster (corosync)...
nodo2.flx: Starting Cluster (corosync)...
nodo2.flx: Starting Cluster (pacemaker)...
nodo1.flx: Starting Cluster (pacemaker)...

[root@nodo1 ~]# pcs cluster enable --all
nodo1.flx: Cluster Enabled
nodo2.flx: Cluster Enabled

Establecer propiedades para el cluster. Solo en el nodo1

[root@nodo1 ~]# pcs property set stonith-enabled=false
[root@nodo1 ~]# pcs property set no-quorum-policy=ignore
[root@nodo1 ~]#

Se debe mostrar así:

[root@nodo1 ~]# pcs status
Cluster name: cluster_voip
Stack: corosync
Current DC: nodo1.flx (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Wed Nov 3 09:30:53 2021
Last change: Tue Nov 2 21:52:43 2021 by root via cibadmin on nodo1.flx

2 nodes configured
0 resource instances configured

Online: [nodo1.flx nodo2.flx]

Importante: Si no se muestra los dos nodos en línea en el estado del pcs (en ambos nodos), el cluster no funcionará correctamente.

Ahora se va a iniciar la creación de los recursos para el cluster. Todo esto se realiza solamente en el nodo1.

3. Crear recurso para IP flotante

```
pcs resource create virtual_ip ocf::heartbeat:IPaddr2 ip=10.100.48.193 cidr_netmask=23 nic=eth0:1 op monitor interval=30s on-fail=restart
```

```
[root@nodo1 ~]# pcs cluster cib drbd_cfg
[root@nodo1 ~]# pcs cluster cib-push drbd_cfg
CIB updated
```

4. Crear recurso para el uso de DRBD

Crear recurso DRBD y crear la replicación para los nodos.

```
[root@nodo1 ~]# pcs -f drbd_cfg resource create DrbdData ocf::linbit:drbd drbd_resource=testdata1 op monitor interval=60s
[root@nodo1 ~]# pcs -f drbd_cfg resource master DrbdDataClone DrbdData master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true
[root@nodo1 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1.flx
[root@nodo1 ~]# pcs cluster cib-push drbd_cfg
CIB updated
[root@nodo1 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1.flx
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nodo1.flx ]
Stopped: [ nodo2.flx ]
```

5. Crear recurso FILESYSTEM para el punto de montaje automatizado

```
[root@nodo1 ~]# pcs cluster cib fs_cfg
[root@nodo1 ~]# pcs -f fs_cfg resource create DrbdFS Filesystem device="/dev/drbd0" directory="/mnt/" fstype="ext4"
Assumed agent name 'ocf::heartbeat:Filesystem' (deduced from 'Filesystem')
[root@nodo1 ~]# pcs -f fs_cfg constraint colocation add DrbdFS with DrbdDataClone INFINITY with-rsc-role=Master
[root@nodo1 ~]# pcs -f fs_cfg constraint order promote DrbdDataClone then start DrbdFS
Adding DrbdDataClone DrbdFS (kind: Mandatory) (Options: first-action=promote then-action=start)
[root@nodo1 ~]# pcs -f fs_cfg constraint colocation add DrbdFS with virtual_ip INFINITY
[root@nodo1 ~]# pcs -f fs_cfg constraint order virtual_ip then DrbdFS
Adding virtual_ip DrbdFS (kind: Mandatory) (Options: first-action=start then-action=start)
[root@nodo1 ~]# pcs cluster cib-push fs_cfg
CIB updated
[root@nodo1 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1.flx
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nodo1.flx ]
Stopped: [ nodo2.flx ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo1.flx
```

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

```
[root@nodo1 ~]# df -h
S.ficheros    Tamaño Usados Disp Uso% Montado en
devtmpfs      984M    0 984M  0% /dev
tmpfs         996M  53M 943M  6% /dev/shm
tmpfs         996M  8,7M 987M  1% /run
tmpfs         996M    0 996M  0% /sys/fs/cgroup
/dev/sda3      8,8G  2,9G  5,4G  35% /
/dev/sda1      267M 146M 102M  59% /boot
tmpfs         200M    0 200M  0% /run/user/0
/dev/drbd0     4,6G  19M  4,3G  1% /mnt
```

Importante: Siempre validar que al reiniciar el nodo1, el nodo2 asuma el papel principal y los recursos agregados inicien en este nodo. Y que el punto de montaje /mnt/ también aparezca.

El estado del pcs en ambos nodos es el siguiente:

```
[root@nodo1 ~]# pcs status
Cluster name: cluster_voip
Stack: corosync
Current DC: nodo1.flx (version 1.1.23-1.el7_9.1-9acf116022) - partition with quorum
Last updated: Wed Nov  3 09:30:53 2021
Last change: Tue Nov  2 21:52:43 2021 by root via cibadmin on nodo1.flx
```

2 nodes configured
4 resource instances configured

Online: [nodo1.flx nodo2.flx]

Full list of resources:

```
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1.flx
Master/Slave Set: DrbdDataClone [DrbdData]
  Masters: [ nodo1.flx ]
  Slaves: [ nodo2.flx ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo1.flx
```

Daemon Status:

```
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Se concluye que la configuración de los primeros cuatro recursos fue exitosa.

6. Crear recurso para el uso de MariaDB

Desactivar el servicio MariaDB en ambos nodos

```
[root@nodo2 ~]# amportal stop
```

Please wait...

STOPPING ASTERISK

Waiting for Asterisk to Stop Gracefully...

Asterisk Stopped

```
[root@nodo2 ~]# systemctl stop mariadb
```

```
[root@nodo2 ~]# systemctl disable mariadb
```

```
Removed symlink /etc/systemd/system/multi-user.target.wants/mariadb.service.
```

Realizar en el nodo 1 la creación de la carpeta dentro del punto de montaje donde se almacenará la base de datos

```
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
```

Creando el recurso para la base de datos

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

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

reboot

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

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

Verificamos el servicio mariadb y vemos que está caído. Ni intentando reiniciar se levanta el servicio:

```
[root@nodo1 ~]# service mariadb status
```

Redirecting to /bin/systemctl status mariadb.service

● mariadb.service - MariaDB database server

Loaded: loaded (/usr/lib/systemd/system/mariadb.service; disabled; vendor preset: disabled)

Active: failed (Result: exit-code) since mié 2021-11-03 17:13:23 -05; 3h 48min ago

Process: 4295 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=1/FAILURE)

```
nov 03 17:13:23 nodo1.flx systemd[1]: Starting MariaDB database server...
```

```
nov 03 17:13:23 nodo1.flx systemd[1]: mariadb.service: control process exited, code=exited status=1
```

```
nov 03 17:13:23 nodo1.flx systemd[1]: Failed to start MariaDB database server.
```

```
nov 03 17:13:23 nodo1.flx systemd[1]: Unit mariadb.service entered failed state.
```

```
nov 03 17:13:23 nodo1.flx systemd[1]: mariadb.service failed.
```

Observación: Lo de MariaDB, no es necesario que esté activo, solo hay que validar el acceso a MySQL y que todos los registros del CDR aparezcan, y se conmuten al nodo2 cuando el nodo1 deje de funcionar.

7. Crear recurso para el uso de Asterisk

Desactivar Asterisk como servicio en ambos nodos

```
[root@nodo2 ~]# service asterisk stop
Redirecting to /bin/systemctl stop asterisk.service
[root@nodo2 ~]# systemctl disable asterisk
Removed symlink /etc/systemd/system/multi-user.target.wants/asterisk.service.
```

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

[root@nodo2 ~]# cd /usr/lib/ocf/resource.d/heartbeat
[root@nodo2 heartbeat]# wget https://raw.githubusercontent.com/ClusterLabs/resource-
agents/master/heartbeat/asterisk
--2021-11-02 19:14:20-- https://raw.githubusercontent.com/ClusterLabs/resource-
agents/master/heartbeat/asterisk
Resolviendo raw.githubusercontent.com (raw.githubusercontent.com). .. 185.199.108.133, 185.199.109.133,
185.199.110.133, ...
Conectando con raw.githubusercontent.com (raw.githubusercontent.com)[185.199.108.133]:443....conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 14844 (14K) [text/plain]
Grabando a: "asterisk"
```

```
100%[=====
=====>] 14.844  --.-K/s en 0,001s
```

2021-11-02 19:14:20 (12,5 MB/s) - "asterisk" guardado [14844/14844]

```
[root@nodo2 heartbeat]# chmod 755 asterisk
```

Agregar recurso asterisk al cluster

```
pcs resource create asterisk ocf:heartbeat:asterisk user="root" group="asterisk" 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@nodo1 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1.flx
Master/Slave Set: DrbdDataClone [DrbdData]
Masters: [ nodo1.flx ]
Slaves: [ nodo2.flx ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo1.flx
mysql (ocf::heartbeat:mysql): Started nodo1.flx
asterisk (ocf::heartbeat:asterisk): Started nodo1.flx
```


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
```

Observación: Las grabaciones se guardan en el /var/spool/monitor y ocupan mucho espacio, y en este caso, no será considerado para el HA.

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
```

Tanto en el nodo1 como en el nodo2, el estado de los recursos es el siguiente:

```
[root@nodo1 ~]# pcs status resources  
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo1.flx  
Master/Slave Set: DrbdDataClone [DrbdData]  
Masters: [ nodo1.flx ]  
Slaves: [ nodo2.flx ]  
DrbdFS (ocf::heartbeat:Filesystem): Started nodo1.flx  
mysql (ocf::heartbeat:mysql): Started nodo1.flx  
asterisk (ocf::heartbeat:asterisk): FAILED nodo1.flx (blocked)
```

Se procede a reiniciar el nodo1, y aparece un aviso referente a un proceso del cluster y que tomará un tiempo aproximado de 30 minutos. Pero, ya cerca o pasado los 20 minutos, comienza a iniciar el nodo1.

Dentro de ese tiempo, se verificó el estado de los recursos del pcs en el nodo2:

```
[root@nodo2 ~]# pcs status resources
virtual_ip (ocf::heartbeat:IPaddr2): Started nodo2.flx
Master/Slave Set: DrbdDataClone [DrbdData]
  DrbdData (ocf::linbit:drbd): FAILED nodo1.flx
  Masters: [ nodo2.flx ]
DrbdFS (ocf::heartbeat:Filesystem): Started nodo2.flx
mysql (ocf::heartbeat:mysql): Started nodo2.flx
asterisk (ocf::heartbeat:asterisk): Started nodo2.flx
```

Importante: No reiniciar el nodo 2 mientras el nodo1 está en el proceso indicado anteriormente.

El nodo1 inicia por completo. Su estado de recursos es el siguiente:

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

Es correcto porque al reiniciar el nodo1, el nodo2 debe tomar el rol primario. Verificamos el status del DRBD en ambos nodos:

```
[root@nodo1 ~]# drbdadm status
testdata1 role:Secondary
disk:UpToDate
peer role:Primary
  replication:Established peer-disk:UpToDate
[root@nodo2 ~]# drbdadm status
testdata1 role:Primary
disk:UpToDate
peer role:Secondary
  replication:Established peer-disk:UpToDate
```

Reiniciamos el nodo2, para que el nodo1 asuma el rol primario nuevamente.

Los recursos, incluido ASTERISK, vuelven a iniciarse en el nodo1, pero se verifica por ejemplo que el servicio Asterisk está inactivo. Levanto el servicio Asterisk ejecutando lo siguiente:

```
[root@nodo1 ~]# service asterisk start
[root@nodo1 ~]# service asterisk status
Redirecting to /bin/systemctl status asterisk.service
● asterisk.service - LSB: Asterisk PBX
   Loaded: loaded (/usr/lib/systemd/system/asterisk.service; disabled; vendor preset: disabled)
   Active: active (running) since mié 2021-11-03 15:32:56 -05; 4s ago
 Main PID: 26171 (asterisk)
   CGroup: /system.slice/asterisk.service
           └─26171 /usr/sbin/asterisk -U asterisk -G asterisk -mqf -C /etc/asterisk/asterisk.conf
```

Procedemos a configurar anexos desde la web (podemos ingresar con la IP del nodo1 o la IP flotante). Se crean exitosamente pero no se reflejan en la consola al ejecutar "sip show peers". Se ve un error de SQL. Para corregir eso se debe:

- Ejecutar "amportal stop" y luego un "amportal start".
- Ingresamos a la consola de Asterisk y validamos que ya no aparezca tal error.

Llegado a este punto, la configuración del HA está concluido.