

ClusterHA com DRBD(Dual-primary),Pacemaker/Corosync/Stonith-fence_xvm, DLM/CLVM e gfs2fs

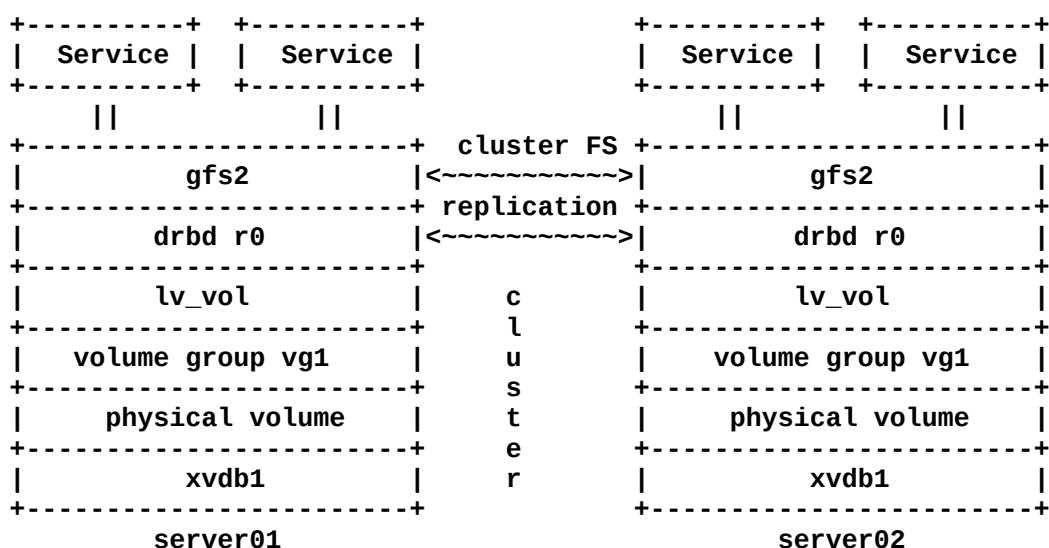
Este tutorial nada mais é do que um compilado com algumas modificações de diversos tutoriais pesquisados na internet, listarei abaixo alguns deles:

- http://www.voleg.info/Linux_RedHat6_cluster_drbd_GFS2.html
- <https://www.tecmint.com/setup-drbd-storage-replication-on-centos-7/>
- https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/_initialize_drbd.html
- <https://www.learnitguide.net/2016/07/how-to-install-and-configure-drbd-on-linux.html>
- <https://www.atlantic.net/cloud-hosting/how-to-drbd-replication-configuration/>
- https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/_initialize_drbd.html
- <https://www.golinuxcloud.com/ste-by-step-configure-high-availability-cluster-centos-7/>
- <https://www.justinsilver.com/technology/linux/dual-primary-drbd-centos-6-gfs2-pacemaker/>
- https://www.ibm.com/developerworks/community/blogs/mhhaque/entry/how_to_configure_red_hat_cluster_with_fencing_of_two_kvm_guests_running_on_two_ibm_powerkvm_hosts?lang=en

.. e muitos outros, conforme decorreremos citaremos mais links.

Aqui esta uma representação quase gráfica do projeto com base no link:

- <https://icimov.github.io/blog/high-availability/Clustering-with-Pacemaker-DRBD-and-GFS2-on-Bare-Metal-servers-in-SoftLayer/>



Configuração dos discos e das partições:

Nós utilizamos duas máquinas virtuais com dois discos rígidos idênticos de 50,3Gb e juntamente com isso duas partições de 10,7Gb.

IPC: Estas duas máquinas virtuais pertencem a mesma rede e é importante salientar que estas duas partições são idênticas!!

Abaixo, encontram-se as saídas do comando `lsblk` em ambas máquinas virtuais.

```
[root@n2drbd ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1 114,5M 0 rom
xvda         202:0    0   50G 0 disk
├─xvda1       202:1    0    1G 0 part /boot
├─xvda2       202:2    0   49G 0 part
│   └─centos-root 253:0    0  45,1G 0 lvm /
│       └─centos-swap 253:1    0   3,9G 0 lvm [SWAP]
└─xvdb        202:16   0   10G 0 disk
```

```
[root@n1drbd ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1 1024M 0 rom
xvda         202:0    0   50G 0 disk
├─xvda1       202:1    0    1G 0 part /boot
├─xvda2       202:2    0   49G 0 part
│   └─centos-root 253:0    0  45,1G 0 lvm /
│       └─centos-swap 253:1    0   3,9G 0 lvm [SWAP]
└─xvdb        202:16   0   10G 0 disk
```

Já quanto ao sistema operacional, usamos a versão mais atual do CentOS, que no momento que esta documentação está sendo escrita é a 7. Juntamente com os programas Pacemaker, Corosync, Stonith, Fence, DLM, CLVM, gfs2fs e etc, todos em suas versões mais atuais e estáveis lançadas até o momento.

Lembrando que toda esta configuração bem como seus resultados serão feitos via terminal, e os prints de cada configuração serão colocados neste arquivo. Antes de começarmos colocarei os prints de como meu disco rígido está organizado.

Iniciando a configuração, primeiramente precisamos deixar essas duas máquinas prontas para receber a configuração inicial, e o primeiro passo é definirmos os seus IP's como estáticos, para criarmos uma conexão ssh estável entre elas.

Acessaremos as máquinas via SSH para este tutorial. Caso você não saiba usar o SSH veja este tutorial: (<http://rberaldo.com.br/usando-o-ssh/>), usaremos somente SSH em nível básico. Os ips das máquinas são:

vm1 : 10.255.255.108 → ssh cam@10.255.255.108

```
cam@cam-IPMH61R3:~$ ssh cam@10.255.255.108
The authenticity of host '10.255.255.108 (10.255.255.108)' can't be established.
ECDSA key fingerprint is SHA256:pLURwLEKU//0ZjhspTmyf/oSTSSUDaU/sDgkFtRuaHg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.255.255.108' (ECDSA) to the list of known hosts.
Connection reset by 10.255.255.108 port 22
cam@cam-IPMH61R3:~$ ssh cam@10.255.255.108
cam@10.255.255.108's password:
Last login: Fri Jul 19 10:30:22 2019 from 10.255.255.143
[cam@localhost ~]$ sudo su -
[sudo] senha para cam:
Último login: Qui Ago 15 17:02:40 -03 2019em tty1
[root@localhost ~]#
```

vm2 : 10.255.255.114 → ssh cam@10.255.255.114

```
cam@cam-IPMH61R3:~$ ssh cam@10.255.255.114
The authenticity of host '10.255.255.114 (10.255.255.114)' can't be established.
ECDSA key fingerprint is SHA256:pLURwLEKU//0ZjhspTmyf/oSTSSUDaU/sDgkFtRuaHg.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.255.255.114' (ECDSA) to the list of known hosts.
Connection reset by 10.255.255.114 port 22
cam@cam-IPMH61R3:~$ ssh cam@10.255.255.114
cam@10.255.255.114's password:
Last login: Mon Jul 15 17:09:20 2019
[cam@localhost ~]$ sudo su -
[sudo] senha para cam:
Último login: Qui Ago 15 15:48:36 -03 2019em tty1
[root@localhost ~]#
```

Lembrete, mantenha sempre o seu sistema atualizado, um exemplo de comando que pode te ajudar é o: `yum update` .

Para mudarmos o IP destas maquinas precisaremos acessar o arquivo: `vi /etc/sysconfig/network-scripts/ifcfg-eth0` . E depois de acessar mudei sua configuração para o seguinte estado:

Neste caso, mudei o endereço IP da mv1 para: **10.255.255.213** e o modo para estático.

```
[root@n1drbd ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=eth0
UUID=cef70f81-e98c-4471-aa2e-cc51528969eb
DEVICE=eth0
ONBOOT=yes
IPADDR=10.255.255.213
NETMASK=255.255.255.0
NETWORK=10.255.255.0
GATEWAY=10.255.255.1
DNS1=10.255.255.1
DNS2=8.8.4.4
```

Já neste, mudei o endereço IP da mv2 para: **10.255.255.234** e o modo também para estático.

```
[root@n2drbd ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=static
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=eth0
UUID=cef70f81-e98c-4471-aa2e-cc51528969eb
DEVICE=eth0
ONBOOT=yes
IPADDR=10.255.255.234
NETMASK=255.255.255.0
NETWORK=10.255.255.0
GATEWAY=10.255.255.1
DNS1=10.255.255.1
DNS2=8.8.4.4
```

Agora, com as placas de rede já configuradas, precisa-se redefinir o nome das maquinas virtuais no arquivo: `/etc /hosts`. Fiz isso com o comando: `“hostname novonome”`. Por exemplo:

```
[root@localhost ~]# hostname
n2drbd.camcluster
```

IPC: Existem varias formas, por exemplo: `hostnamectl set-hostname "lalala.pc.uou"`. Em que "lalala.pc.uou" é o novo hostname completo da minha maquina. Em caso de duvida, recomendo o link: -<https://www.hostinger.com.br/tutoriais/como-mudar-hostname-ubuntu/>

Agora que a fase inicial da preparação terminou, vamos começar de vez a configuração. Primeiramente vou mostrar um layout de como o projeto foi desenvolvido por mim e a forma que achei mas comoda pra implementar a configuração.

Etapa 1: de inicio nos temos a preparação da maquina quanto a rede e nome, que acabei de mostrar. Juntamente com a instalação do DRBD e a sua configuração em modo (DUAL-PRIMARY).E depois a instalação e configuração dos gerenciadores de cluster, o COROSYNC e o PACEMAKER.

Etapa 2: prosseguindo para etapa 2 temos as configurações para o funcionamento do CLUSTER PCS bem como a configuração deste cluster em conjunto com o Fence e Stonith utilizando um de seus agentes de segurança o (fence_xvm) que é voltado a maquinas virtuais, se você não esta utilizando maquinas virtuais ou virtualizadas, procure por seu agente especifico. Deve-se ter muito cuidado com a configuração deste recurso, por ele sera um apoio aos futuros recursos em todos os nós do cluster.

Dando continuidade nos configuraremos os recursos de DLM, CLVM(LVM). O primeiro é uma parte obrigatória do cluster e tem a função de gerenciar os blocos assim como o nome sugere, porque, se um dos nós do cluster cair, é nosso dever manter o outro nó do cluster limpo.

E o (LVM/CLVM) nada mais são que gerenciadores de volume lógico. Se vários nós do cluster exigirem acesso simultâneo de leitura / gravação a volumes LVM em um sistema ativo / ativo, você deverá usar o CLVMD.

O CLVMD fornece um sistema para coordenar a ativação e as alterações nos volumes de LVM nos nós de um cluster simultaneamente.

O serviço de bloqueio em cluster da CLVMD fornece proteção aos metadados do LVM, pois vários nós do cluster interagem com os volumes e fazem alterações em seu layout.

Etapa 3: Nesta etapa utilizaremos todos os recursos e criaremos os ultimos recursos para a conclusão do projeto. Dentre eles temos o PV(Physical Volume), o VG(Volume Group) e o LV(logical volume). No qual será montado o DRBD, e para isso mudaremos a configuração inicial feita no arquivo de recurso que nele está. E agora por fim, como o ultimo passo da ultima etapa vou formatar o disco simulado pelo DRBD com um sistema de arquivos especial chamado de GFS2FS e montando uma partição nele. Vale a pena mencionar que esta ultima etapa só deve ser feita em um dos servers(por que a mudança será replicada automaticamente).

IPC: A montagem e a configuração do DRBD foi feita no inicio por preferência minha, ela pode ser feita no final quando for necessaria sua implementação no código.

Adicionar o repositório dos pacotes para o funcionamento do DRBD:(Nos dois servers)

`[root@n1drbd ~]# rpm -ivh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm`

```
[root@n1drbd ~]# rpm -ivh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
```

Instalar o DRBD, depois de importar o repositório no qual ele se encontra usando os seguintes comandos.(nos dois servers)

```
[root@n1drbd ~]# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
[root@n1drbd ~]# yum install drbd kmod-drbd
```

```
[root@n1drbd ~]# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
[root@n1drbd ~]# yum install drbd kmod-drbd
```

Instale também as dependencias abaixo com o comando: “yum install drbd90-utils.x86_64 drbd84-utils-sysvinit.x86_64 kmod-drbd84.x86_64”

```
[root@n1drbd ~]# yum install drbd90-utils.x86_64 drbd90-utils-sysvinit.x86_64 kmod-drbd90.x86_64
Plugins carregados: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.facom.ufes.br
 * elrepo: mirror.pit.teraswitch.com
 * extras: mirror.ufscar.br
 * updates: mirror.ufscar.br
0 pacote drbd90-utils-9.6.0-1.el7.elrepo.x86_64 já está instalado em sua última versão
Resolvendo dependências
--> Executando verificação da transação
--> 0 pacote drbd90-utils-sysvinit.x86_64 0:9.6.0-1.el7.elrepo será instalado
--> 0 pacote kmod-drbd90.x86_64 0:9.6.0-1.el7.elrepo será instalado
--> Resolução de dependências finalizada
Dependências resolvidas

=====
Package                               Arq.                Versão              Repo                Tam.
=====
Instalando:
drbd90-utils-sysvinit                 x86_64              9.6.0-1.el7.elrepo elrepo              7.2 k
kmod-drbd90                           x86_64              9.6.0-1.el7.elrepo elrepo              271 k
Resumo da transação
=====
Instalar 2 Packages

Tamanho total do download: 278 k
Tamanho depois de instalado: 972 k
Is this ok [y/d/N]: y
Downloading packages:
(1/2): drbd90-utils-sysvinit-9.6.0-1.el7.elrepo.x86_64.rpm | 7.2 kB 00:00:01
```

Este comando deverá, depois de uma pergunta que deve ser respondida com (y), exibir uma resposta do tipo “Concluído”.

Agora o DRBD já está instalado, prepare o disco e as partições para fazerem parte de meu arquivo de configuração do recurso (não precisa se preocupar com a identificação do modulo o mesmo será carregado no próximo boot e se os arquivos de configuração estiverem corretos, tudo funcionará). Faça isso nos dois servers usando o comando: “cfdisk /dev /xvdb”. Atente-se ao nome de seu disco pode ser diferente dependendo da distribuição linux que esta sendo usada.

```
[root@n1drbd ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1 1024M  0 rom
xvda         202:0    0   50G  0 disk
├─xvda1      202:1    0    1G  0 part /boot
├─xvda2      202:2    0   49G  0 part
│   └─centos-root 253:0    0 45,1G  0 lvm /
│       └─centos-swap 253:1    0  3,9G  0 lvm [SWAP]
└─xvdb       202:16   0   10G  0 disk
    └─xvdb1   202:17   0   10G  0 part
```

Depois de acionar esse comando, selecione a opção, NOVA (para criar uma nova partição), e depois a opção GRAVAR, para confirmar a criação dessa nova partição em disco. Mude a

saida do comando: “lsblk” para este estado.(Nos dois servers) pode se usar o comado: fdisk também.

Agora. configure a utilização dos arquivos instalados junto com o pacote drdb. Acessando o arquivo: “cat /etc/selinux/config” e mudando o status de SELINUX para (disabled), no arquivo.

Exatamente como esta aqui!

IPC:Para aplicar alteração, é necessário o reboot da maquina!

```
[root@n1drbd ~]# vi /etc/selinux/config
[root@n1drbd ~]# cat /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

[root@n1drbd ~]#
```

Ou mude o estado do SELinux para permissivo, com os comandos abaixo.

```
[root@n1drbd ~]# setenforce permissive
```

```
[root@n1drbd ~]# sestatus
```

SELinux status:	enabled
SELinuxfs mount:	/sys/fs/selinux
SELinux root directory:	/etc/selinux
Loaded policy name:	targeted
Current mode:	permissive
Mode from config file:	enforcing
Policy MLS status:	enabled
Policy deny_unknown status:	allowed
Max kernel policy version:	31

O desativo o SELinux permanentemente com a sequencia de comandos.

Disabling SELinux permanently

Edit the /etc/selinux/config file, run:

```
sudo vi /etc/selinux/config
```

*Set SELINUX to **disabled**:*

```
SELINUX=disabled
```

Save and close the file in vi/vim. Reboot the Linux system:

```
sudo reboot
```

Para saber mais acesse o link: <https://www.cyberciti.biz/faq/disable-selinux-on-centos-7-rhel-7-fedora-linux/>

Revisando toda configuração até aqui.

Configuração de Firewall

Consulte a documentação do seu firewall para saber como abrir / permitir portas. Você precisará das seguintes portas abertas para seu cluster funcionar corretamente.

Portas:

Component	Protocol	Port
DRBD	TCP	7788
Corosync	UDP	5404, 5405
GFS2	TCP	2224, 3121, 21064

- iptables -I INPUT -p tcp --dport 2224 -j ACCEPT --- iptables -nL | grep 2224
- iptables -I INPUT -p tcp --dport 3121 -j ACCEPT --- iptables -nL | grep 3121
- iptables -I INPUT -p tcp --dport 21064 -j ACCEPT --- iptables -nL | grep 21064
- iptables -I INPUT -p udp --dport 5404 -j ACCEPT --- iptables -nL | grep 5404
- iptables -I INPUT -p udp --dport 5405 -j ACCEPT --- iptables -nL | grep 5405

Também habilite a porta 7788 no firewall, de ambas as máquinas para não sofrer futuros erros de validação, faça os comandos abaixo em todos os nós do projeto.

- firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.255.255.231" port port="7788" protocol="tcp" accept'
- firewall-cmd reload

```
[root@n1drbd ~]# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.255.255.234" port port="7788" protocol="tcp" accept'
success
[root@n1drbd ~]# firewall-cmd --reload
usage: see firewall-cmd man page
firewall-cmd: error: unrecognized arguments: --reload
[root@n1drbd ~]# firewall-cmd --reload
success
[root@n1drbd ~]# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.255.255.213" port port="7788" protocol="tcp" accept'
success
```

Os links usados até aqui, e os que ainda serão usados.

- <https://www.osradar.com/installing-and-configuring-a-drbd-cluster-in-centos-7/>
- <http://www.tadeubernacchi.com.br/desabilitando-o-firewalld-centos-7/>
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/sect-Security-Enhanced_Linux-Working_with_SELinux-Changing_SELinux_Modes#sect-Security-Enhanced_Linux-Enabling_and_Disabling_SELinux-Permissive_Mode
- <https://www.learnitguide.net/2016/07/how-to-install-and-configure-drbd-on-linux.html>
- <https://www.atlantic.net/cloud-hosting/how-to-drbd-replication-configuration/>
- <https://www.tecmint.com/setup-drbd-storage-replication-on-centos-7/>
- <https://major.io/2011/02/13/dual-primary-drbd-with-ocfs2/>
- <https://www.golinuxcloud.com/configure-gfs2-setup-cluster-linux-rhel-centos-7/>
- https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/ch09.html
- <https://www.lisenet.com/2016/02/cb-cluster-with-dual-primary-drbd-and-ocfs2-on-oracle-linux-7/>
- <http://www.voleg.info/stretch-nfs-cluster-centos-drbd-gfs2.html>
- <http://jensd.be/186/linux/use-drbd-in-a-cluster-with-corosync-and-pacemaker-on-centos-7>

- <https://icicimov.github.io/blog/high-availability/Clustering-with-Pacemaker-DRBD-and-GFS2-on-Bare-Metal-servers-in-SoftLayer/>
- <https://www.justinsilver.com/technology/linux/dual-primary-drbd-centos-6-gfs2-pacemaker/>
- <http://www.tadeubernacchi.com.br/desabilitando-o-firewalld-centos-7/>
- http://tutoriaisgnulinux.com/2013/06/08/redhat-cluster-configurando-fence_virt/
- <https://www.ntppool.org/zone/br>
- https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/selinux_users_and_administrators_guide/sect-Security-Enhanced_Linux-Working_with_SELinux-Changing_SELinux_Modes

IPC: Por motivos de eficiência. Usarei somente a copia dos comandos de agora em diante. E reinstalaremos o DRBD.

Instale o DRBD:

```
[root@n1drbd ~]# rpm -ivh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm
```

```
[root@n1drbd ~]# lsmod | grep -i drbd
```

Verifique se todos os hosts estão com os nomes e ips devidamente configurados.

```
[root@n1drbd ~]# cat /etc/hostname
```

```
[root@n1drbd ~]# cat /etc/hosts
```

Edite o arquivo “/etc/drbd.d/global_common.conf” e modifique a opção “usage-count de yes para no” e salve o arquivo, em todos os nós(mvs) do DRBD. Exemplo:

```
global {
    usage-count no;
}
```

E

nos dois nós do cluster crie o arquivo, “r0.res” dentro do diretório, “/etc/drbd.d”.

```
resource r0 {
    protocol C;

    syncer {
        #rate 100M;
        verify-alg sha1;
    }
    startup {
        wfc-timeout 0;
        # non-zero wfc-timeout can be dangerous
        degr-wfc-timeout 120;
        outdated-wfc-timeout 120;
        become-primary-on both;
    }
    disk {
        resync-rate
```

Neste primeiro momento, vamos deixar algumas linhas comentadas afim de que não atrapalhem na criação de recursos futuros


```

        33M;
        c-max-rate 110M;
        c-min-rate 10M;
        c-fill-target 16M;
        #fencing resource-and-stonith;
        no-disk-barrier;
        no-disk-flushes;
    }
    net {
        cram-hmac-alg sha1;
        shared-secret "my-secret";
        use-rle yes;
        allow-two-primaries yes;
        after-sb-0pri discard-zero-changes;
        after-sb-1pri discard-secondary;
        after-sb-2pri disconnect;
    }
    handlers {
        fence-peer"/usr/lib/drbd/rhcs_fence";
    }
    on n1drbd.camcluster {
        device /dev/drbd0;
        disk /dev/fileserver/r0;
        address 10.255.255.213:7788;
        meta-disk internal;
    }
    on n2drbd.camcluster {
        device /dev/drbd0;
        disk /dev/fileserver/r0;
        address 10.255.255.234:7788;
        meta-disk internal;
    }
}

```

Já este arquivo deve ficar no diretório: “/etc/init.d/loop-for-drbd” Para manter o modo dual-primary após o reboot.

```

#!/bin/sh
#
# Startup script for drbd loop device setup
#
# chkconfig: 2345 50 50
# description: Startup script for drbd loop device setup
#
#### BEGIN INIT INFO
# Provides: drbdloop
# Required-Start:
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
# Short-Description: set up drbd loop devices
# Description: Startup script for drbd loop device setup
#### END INIT INFO

```

```

DRBD_FILEDATA_SRC="/drbd-loop.img"
DRBD_FILEDATA_DEVICE="/dev/loop7"
LOSETUP_CMD=/sbin/losetup

# Source function library
. /etc/rc.d/init.d/functions

start () {
    echo -n $"Setting up DRBD loop devices..."
    $LOSETUP_CMD $DRBD_FILEDATA_DEVICE $DRBD_FILEDATA_SRC
    echo
}

stop() {
    echo -n $"Tearing down DRBD loop devices..."
    $LOSETUP_CMD -d $DRBD_FILEDATA_DEVICE
    echo
}

restart() {
    stop
    start
}

case "$1" in
    start)
        start
        RETVAL=$?
        ;;
    stop)
        stop
        RETVAL=$?
        ;;
    restart)
        restart
        RETVAL=$?
        ;;
    *)
        echo $"Usage: $0 {start|stop}"
        exit 1
esac

exit $RETVAL

```

Com esses arquivos em seus respectivos lugares, inicia-se a configuração do DRBD para uso em modo (DUAL-PRIMARY).

Não ativaremos o DRBD nesta etapa, por isso atense-se a configuração para que nada dê errado no momento da ativação do banco.

Para para os próximos passos use também os links como apoio: <https://www.golinuxcloud.com/ste-by-step-configure-high-availability-cluster-centos-7/>, https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/_install_the_cluster_software.html, **as instruções devem ser implementadas em todas as mvs/nós. Embora, neste guia só seja mostrado rodando o comando em uma só máquina.**

Proceda a instalação do corosync e pacemaker. Verifique se os hosts estão corretamente identificados no arquivo: “/etc/hosts”

```
[root@n1drbd ~]# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
#novos nomes e ips
10.255.255.213 n1drbd.camcluster n1drbd
10.255.255.234 n2drbd.camcluster n2drbd
```

Pare e desative o Network Manager em todos os pcs envolvidos.

```
[root@n1drbd ~]# systemctl disable NetworkManager
Removed symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
Removed symlink /etc/systemd/system/dbus-org.freedesktop.NetworkManager.service.
Removed symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
Removed symlink /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service.
```

Configure o servidor NTP

```
[root@n2drbd ~]# systemctl enable ntpd
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to
/usr/lib/systemd/system/ntpd.service.
[root@n2drbd ~]# firewall-cmd --add-service=high-availability
Warning: ALREADY_ENABLED: 'high-availability' already in 'public'
success
[root@n2drbd ~]# firewall-cmd --reload
success
```

Instale os rpms(repositórios) necessários.

```
[root@n1drbd ~]# yum install epel-release -y
```

Instale o pacemaker e os agentes de fence.

```
[root@n1drbd ~]# yum install pcs fence-agent-all -y          ou
[root@n1drbd ~]# yum install -y pacemaker pcs psmisc policycoreutils-python
```

Adicione as novas regras e reinicie o firewall.

```
[root@n1drbd ~]# firewall-cmd --permanent --add-service=high-availability
```

Verifique o status do selinux.

```
[root@n1drbd ~]# sestatus
```

Verifique se as portas TCP/UDP que habilitamos durante a primeira etapa estão realmente habilitadas, conforme a configuração de firewall inicial.

```
[root@n1drbd ~]# iptables -nL | grep <porta>
```

Se sim, configure a senha para o cluster:

```
[root@n1drbd ~]# echo password | passwd --stdin hacluster
```

Mudando senha para o usuário hacluster.

passwd: todos os tokens de autenticações foram atualizados com sucesso.

Inicie o gerenciador de cluster do pacemaker.Em cada nó.

```
[root@n1drbd ~]# systemctl enable --now pcsd
```

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

Autentique o hacluster em qualquer um dos mv's. Us:hacluster Se:password

```
[root@n1drbd ~]# pcs cluster auth n1drbd.camcluster n2drbd.camcluster
```

Agora, no seu primeiro pc, vm, node, etc, digite e execute o comando abaixo para criar o cluster e starta-lo.

```
[root@n1drbd ~]# pcs cluster setup --start --name mycluster n1drbd.camcluster n2drbd.camcluster
```

Destroying cluster on nodes: n1drbd.camcluster, n2drbd.camcluster...

n2drbd.camcluster: Stopping Cluster (pacemaker)...

n1drbd.camcluster: Stopping Cluster (pacemaker)...

n2drbd.camcluster: Successfully destroyed cluster

n1drbd.camcluster: Successfully destroyed cluster

Sending 'pacemaker_remote authkey' to 'n1drbd.camcluster', 'n2drbd.camcluster'

n1drbd.camcluster: successful distribution of the file 'pacemaker_remote authkey'

n2drbd.camcluster: successful distribution of the file 'pacemaker_remote authkey'

Sending cluster config files to the nodes...

n1drbd.camcluster: Succeeded

n2drbd.camcluster: Succeeded

Starting cluster on nodes: n1drbd.camcluster, n2drbd.camcluster...

n1drbd.camcluster: Starting Cluster (corosync)...

n2drbd.camcluster: Starting Cluster (corosync)...

n1drbd.camcluster: Starting Cluster (pacemaker)...

n2drbd.camcluster: Starting Cluster (pacemaker)...

Synchronizing pcsd certificates on nodes n1drbd.camcluster, n2drbd.camcluster...

n1drbd.camcluster: Success

n2drbd.camcluster: Success

Restarting pcsd on the nodes in order to reload the certificates...

n1drbd.camcluster: Success

n2drbd.camcluster: Success

O recurso de cluster está ativo. Caso nunca tenha visto, aqui estão alguns comandos para ajudar no gerenciamento deste recurso.

```
[root@n1drbd ~]# pcs
```

```
[root@n1drbd ~]# pcs status help
```

```
[root@n1drbd ~]# pacemakerd -features
```

Habilite o recurso com o pacemaker e o corosync

```
[root@n1drbd ~]# pcs cluster enable --all
```

```
n1drbd.camcluster: Cluster Enabled
```

```
n2drbd.camcluster: Cluster Enabled
```

Faça a checagem do status do cluster

```
[root@n1drbd ~]# pcs cluster enable --all
```

```
n1drbd.camcluster: Cluster Enabled
```

```
n2drbd.camcluster: Cluster Enabled
```

```
[root@n1drbd ~]# pcs cluster status
```

```
Cluster Status:
```

```
Stack: corosync
```

```
Current DC: n2drbd.camcluster (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition WITHOUT quorum
```

```
Last updated: Wed Aug 28 16:12:08 2019
```

```
Last change: Wed Aug 28 15:27:19 2019 by hacluster via crmd on n2drbd.camcluster
```

```
2 nodes configured
```

```
0 resources configured
```

```
PCSD Status:
```

```
n1drbd.camcluster: Online
```

```
n2drbd.camcluster: Online
```

Verifique o quorum do cluster

```
[root@n1drbd ~]# corosync-cfgtool -s
```

Verifique o status em tempo real do cluster

```
[root@n1drbd ~]# crm_mon
```

```
[root@n1drbd ~]# corosync-cmapctl | grep members
```

IPC: Em caso de erro, revisar passos anteriores.

Aqui abaixo estão alguns links de apoio para o caso haja erros de configuração ou sistema.

- <https://bugs.launchpad.net/debian/+source/pcs/+bug/1640923>

- <https://shgonzalez.github.io/linux/ha/2017/10/02/How-to-solve-pacemaker-error.html>

- <https://oss.clusterlabs.org/pipermail/pacemaker/2014-September/022536.html>

- <https://github.com/ClusterLabs/pcs/issues/153>
- <http://fibrevillage.com/sysadmin/317-pacemaker-and-pcs-on-linux-example-cluster-creation-add-a-node-to-cluster-remove-a-node-from-a-cluster-destroy-a-cluster>

Comando importante!

```
[root@n1drbd ~]# sudo pcs cluster setup --force n1drbd n2drbd --name mycluster
```

Configurando o fence e o stonith_xvm nos nós do cluster;

```
[root@n1drbd ~]# pcs stonith show
```

Se todos os passos foram feitos corretamente não haverá dispositivos configurados com este recurso. A mensagem de retorno vai ser algo parecido com isso: “NO stonith devices configured”.

```
[root@n1drbd ~]# pcs property set no-quorum-policy=freeze
```

```
[root@n1drbd ~]# pcs property set stonith-enabled=true
```

```
[root@n1drbd ~]# pcs property show
```

Cluster Properties:

cluster-infrastructure: corosync

cluster-name: mycluster

dc-version: 1.1.19-8.el7_6.4-c3c624ea3d

have-watchdog: false

stonith-enabled: true

Verifique o pacemaker e corosync e veja seu status de cluster.

```
[root@n1drbd ~]# pcs cluster cib
```

```
[root@n1drbd ~]# ps axf
```

```
[root@n1drbd ~]# journalctl -b | grep -i error
```

```
[root@n1drbd ~]# systemctl status firewalld
```

```
[root@n1drbd ~]# pcs stonith show
```

```
[root@n1drbd ~]# yum install pcs fence-agents-all -y
```

```
[root@n2drbd ~]# pcs stonith list
```

```
[root@n1drbd ~]# systemctl status pcsd.service
```

Criando o recurso fence/stonith nos nós do cluster.

```
[root@n1drbd ~]# pcs stonith create fence_n1 fence_xvm pcmk_host_list="n1drbd"
port="n1drbd.camcluster"
```

```
[root@n2drbd ~]# pcs stonith create fence_n2 fence_xvm pcmk_host_list="n2drbd"
port="n2drbd.camcluster"
```

```
[root@n1drbd ~]# pcs stonith show
```

Instale os pacotes faltosos nas maquinas(Execute este comando nas duas maquinas)

```
[root@n1drbd ~]#yum -y install fence-agents-all fence-agents-virsh fence-virt pacemaker-
remotepcs fence-virt resource-agents fence-virt-libvirt fence-virt-multicast
```

Habilite o agente de fence para execução do recurso

```
[root@n1drbd ~]# systemctl start fence_virt
[root@n1drbd ~]# systemctl enable fence_virt
```

Libere a porta tcp correspondente para execução do recurso de fence

```
[root@n1drbd ~]# firewall-cmd --add-port=1229/tcp --permanent
```

De agora em diante siga as instruções do site:

https://www.ibm.com/developerworks/community/blogs/mhhaque/entry/how_to_configure_red_hat_cluster_with_fencing_of_two_kvm_guests_running_on_two_ibm_powerkvm_hosts?lang=en para criação da chave de fence.

Para criação da chave xvm, faça:

```
[root@n2drbd tmp]# dd if=/dev/urandom of=/etc/cluster/fence_xvm.key bs=4k count=1
```

Copie o arquivo para todos os nós(mvs) via scp

```
[root@n1drbd ~]# sudo scp /etc/cluster/fence_xvm.key cam@n2drbd.camcluster:/tmp/
```

No nó(mv) de destino copie a chave para o diretório desejado

```
[root@n2drbd tmp]# mv fence_xvm.key /etc/cluster/fence_xvm.key
```

Configure a chave para a utilização

```
[root@n1drbd ~]# fence_virt -c
```

Neste tutorial se fez a configuração padrão, somente aceitando sem modificações

```
[root@n1drbd ~]# pcs stonith create fence_n2drbd fence_xvm pcmk_host_list="n1drbd"
port="n1drbd.camcluster"
[root@n1drbd ~]# pcs stonith create fence_n1drbd fence_xvm pcmk_host_list="n2drbd"
port="n2drbd.camcluster"
```

<https://www.unixarena.com/2016/01/rhel7-configuring-gfs2-on-pacemakercorosync-cluster.html/>

Uma configuração alternativa, este link apresenta uma configuração mais alternativa e simples.

Agora vamos ativar o recurso do agente fence e stonith.

```
[root@n1drbd ~]# pcs cluster fence_xvm enable
```

```
[root@n1drbd ~]# pcs cluster fence_xvm enable
```

```
[root@n1drbd ~]# pcs property set no-quorum-pilocy=freeze
```

Em todos os nos(mvs) do cluster. Aqui estão somente os exemplos.

```
[root@n1drbd ~]# systemctl enable fence_virt.service
```

```
[root@n1drbd ~]# systemctl start fence_virt.service
```

```
[root@n1drbd ~]# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="10.255.255.213" port port="1229" protocol="tcp" accept'
```

```
[root@n1drbd ~]# firewall-cmd --reload
```

```
[root@n1drbd ~]# iptables -xnvL
```

```
[root@n1drbd ~]# pcs property set stonith-enabled=true
```

Depois desta série de comandos o recurso de fence_xvm, stonith deverá startar. Para confirmar:

```
[root@n1drbd ~]# pcs status
```

Cluster name: mycluster

Stack: corosync

Current DC: n2drbd.camcluster (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum

Last updated: Wed Sep 4 09:55:17 2019

Last change: Wed Sep 4 08:54:31 2019 by root via cibadmin on n1drbd.camcluster

2 nodes configured

2 resources configured

Online: [n1drbd.camcluster n2drbd.camcluster]

Full list of resources:

xvmfence_n1 (stonith:fence_xvm): Started n1drbd.camcluster

xvmfence_n2 (stonith:fence_xvm): Started n2drbd.camcluster

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

```
[root@n1drbd ~]# pcs stonith show
xvmfence_n1 (stonith:fence_xvm): Started n1drbd.camcluster
xvmfence_n2 (stonith:fence_xvm): Started n2drbd.camcluster
```

O resultado obtido deve ser algo próximo do descrito acima. Se não for, existe algo errado com a configuração feita, revise os passos até aqui, ou veja os links de apoio. De agora em diante, por mais que use outros sites e artigos para pesquisa, o tutorial segue principalmente o link: <https://www.golinuxcloud.com/configure-gfs2-setup-cluster-linux-rhel-centos-7/>

Agora instale os recursos de DLM e CLVM que são pré-requisitos para montagem do sistema de arquivos especial GFS2FS que também será instalado.(Nas duas máquinas)

```
[root@n1drbd ~]# yum install gfs2-utils lvm2-cluster dlm
```

Antes de prosseguir é necessário que se mude ou confira a mudança nas seguinte propriedade do cluster:

```
[root@n1drbd ~]# pcs property set no-quorum-policy=freeze
```

Verifique se o recurso dlm-clone[dlm] esta em modo 'Started' se não estiver verifique, pois deve haver algum erro na configuração.

Configurando o recurso de CLVM.

```
[root@node1 ~]# grep locking_type /etc/lvm/lvm.conf | egrep -v '#' locking_type = 3
```

..para alterar dinamicamente.

```
[root@n1drbd ~]# lvmconf --enable-cluster
```

```
[root@n1drbd ~]# systemctl disable lvm2-lvmetad --now
```

Warning: Stopping lvm2-lvmetad.service, but it can still be activated by: lvm2-lvmetad.socket

Crie o recurso clvm, e verifique se os recursos estarão ativos.

```
[root@n1drbd ~]# pcs resource create clvmd ocf:heartbeat:clvm op monitor interval=30s on-fail=fence clone interleave=true ordered=true
```

```
[root@n1drbd ~]# pcs status
```

Cluster name: mycluster

Stack: corosync

Current DC: n2drbd.camcluster (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum

Last updated: Wed Sep 4 15:01:53 2019

Last change: Wed Sep 4 15:00:13 2019 by root via cibadmin on n1drbd.camcluster

2 nodes configured

6 resources configured

Online: [n1drbd.camcluster n2drbd.camcluster]

Full list of resources:

```
xvmfence_n1 (stonith:fence_xvm): Started n1drbd.camcluster
xvmfence_n2 (stonith:fence_xvm): Started n2drbd.camcluster
Clone Set: dlm-clone [dlm]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
Clone Set: clvmd-clone [clvmd]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
```

Daemon Status:

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

A saída deverá ser de alguma forma parecida com a que está descrita acima. Caso não esteja, verifique novamente a configuração. Mude a ordem de boot dos recursos.

```
[root@n1drbd ~]# pcs constraint order start dlm-clone then clvmd-clone
Adding dlm-clone clvmd-clone (kind: Mandatory) (Options: first-action=start then-action=start)
```

```
[root@n1drbd ~]# pcs constraint colocation add clvmd-clone with dlm-clone
```

Com estes recursos criados, configurados e organizados, configure agora o armazenamento compartilhado entre as duas máquinas virtuais(vm).

Pare o cluster n2drbd.camcluster

```
[root@n2drbd ~]# pcs cluster stop
```

Construa os volumes lógicos na primeira máquina virtual(mv1)

```
[root@n1drbd ~]# pvcreate /dev/xvdb1
```

```
[root@n1drbd ~]# vgcreate fileserver /dev/xvdb1
```

```
[root@n1drbd ~]# lvcreate --name r0 --size 9,9G fileserver
```

Verifique o status destes volumes

```
[root@n1drbd ~]# pvs
```

```
[root@n1drbd ~]# vgs
```

```
[root@n1drbd ~]# lvs
```

Reinicie o Cluster na segunda máquina virtual(mv2), e pare o da primeira, e repita.

```
[root@n2drbd ~]# pcs cluster start
```

```
[root@n1drbd ~]# pcs cluster stop
```

```
root@n2drbd ~]# pvcreate /dev/xvdb1
```

```
[root@n2drbd ~]# vgcreate fileserver /dev/xvdb1
```

```
[root@n2drbd ~]# lvcreate --name r0 --size 9,9G fileserver
```

Verifique o status destes volumes

```
[root@n2drbd ~]# pvs
```

```
[root@n2drbd ~]# vgs
```

```
[root@n2drbd ~]# lvs
```

Agora vamos ativar o nosso DRBD e colocar em modo dual-primary. Nos dois nós(vms)

```
[root@n1drbd ~]# drbdadm create-md r0 - 1º     [root@n1drbd ~]# drbdadm adjust r0 - 4º
```

```
[root@n1drbd ~]# drbdadm up r0 - 2º            [root@n1drbd ~]# watch cat /proc/drbd - 5º
```

```
[root@n1drbd ~]# drbdadm primary r0 - 3º
```

Monte o sistema de arquivos gfs2fs nos volumes criados (EM TODOS OS NÓS(VMS))

```
[root@n1drbd ~]# mkfs.gfs2 -j3 -p lock_dlm -t mycluster:gfs2fs /dev/drbd0
./          .anaconda-ks.cfg .bash_logout  .bashrc    .pki/      .tcshrc
../         .bash_history  .bash_profile .cshrc     .ssh/
[root@n1drbd ~]# mkfs.gfs2 -j3 -p lock_dlm -t mycluster:gfs2fs /dev/fileserver/r0
/dev/fileserver/r0 is a symbolic link to /dev/dm-2
This will destroy any data on /dev/dm-2
Are you sure you want to proceed? [y/n] y
Discarding device contents (may take a while on large devices): Done
Adding journals: Done
Building resource groups: Done
Creating quota file: Done
Writing superblock and syncing: Done
Device:                /dev/fileserver/r0
Block size:            4096
Device size:           9,90 GB (2595840 blocks)
Filesystem size:       9,90 GB (2595836 blocks)
Journals:              3
Journal size:          32MB
Resource groups:       43
Locking protocol:      "lock_dlm"
Lock table:            "mycluster:gfs2fs"
UUID:                  5438d7f8-a5cc-4264-b9af-78ee8b98598b
```

Onde

-t clusternome: fsname: é usado para especificar o nome da tabela de bloqueio

-j nn: especifica quantos diários (nós) são usados

-J: permite especificar o tamanho de journal. se não especificado, o journal terá um tamanho padrão de 128 MB. O tamanho mínimo é 8 MB (NÃO recomendado)

No comando, clusternome deve ser o nome do cluster pacemaker, pois usei mycluster, que é o nome do meu cluster.

O retorno do comando deve ser algo similar ao descrito acima, se isto não ocorrer, é provável que existam erros de configuração.

Crie manualmente o ponto de montagem em todos os nós(mvs) do cluster.

```
[root@n1drbd ~]# mkdir /clusterfs
```

Antes de criar o recurso de GFS2 valide manualmente se o sistema de arquivos 'lvcluster' esta funcionando e de forma apropriada.

No nó(vm)1 faça:

```
[root@n1drbd ~]# mount /dev/drbd0 /clusterfs/
```

No nó(vm)2 faça:

```
[root@n2drbd ~]# mount | grep clusterfs
```

Se não retornar nenhum resultado, monte a partição também nos outros nós(vms) e crie este novo recurso.

```
root@n1drbd ~]# pcs resource create gfs2fs Filesystem device="/dev/drbd0" directory="/clusterfs"
fstype=gfs2 options=noatime op monitor interval=10s on-fail=fence clone interleave=true
Assumed agent name 'ocf:heartbeat:Filesystem' (deduced from 'Filesystem')
```

Verifique o status do serviço que foi criado e veja se ele esta ativo e rodando, veja se ele esta com o status esperado.

```
[root@n1drbd ~]# pcs status
```

Cluster name: mycluster

Stack: corosync

Current DC: n2drbd.camcluster (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum

Last updated: Thu Sep 5 11:17:00 2019

Last change: Thu Sep 5 11:16:49 2019 by root via cibadmin on n1drbd.camcluster

2 nodes configured

8 resources configured

Online: [n1drbd.camcluster n2drbd.camcluster]

Full list of resources:

xvmfence_n1 (stonith:fence_xvm): Started n2drbd.camcluster

```
xvmfence_n2 (stonith:fence_xvm): Started n1drbd.camcluster
Clone Set: dlm-clone [dlm]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
Clone Set: clvmd-clone [clvmd]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
Clone Set: gfs2fs-clone [gfs2fs]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
```

Daemon Status:

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

O resultado destes comandos deve ser algo similar a saída mostrada acima. Portanto, nosso serviço gfs2fs é iniciado automaticamente em todos os nossos nós(vms) do cluster.

Agora organize a ordem de inicialização do recurso para GFS2 e CLVMD, para que, após a reinicialização de um nó, os serviços sejam iniciados na ordem correta, caso contrário, eles falharão ao iniciar

```
[root@n1drbd ~]# pcs constraint order start clvmd-clone then gfs2fs-clone
Adding clvmd-clone gfs2fs-clone (kind: Mandatory) (Options: first-action=start then-action=start)
```

```
[root@n1drbd ~]# pcs constraint colocation add gfs2fs-clone with clvmd-clone
```

Agora, já que nosso recurso / serviço está sendo executado corretamente. Crie um arquivo.

```
[root@n1drbd ~]# cd /clusterfs
```

```
[root@n1drbd ~]# touch file
```

Agora verifique que os arquivos são replicados em tempo real para o segundo nó(vm).

```
[root@n2drbd ~]# cd /clusterfs
```

```
[root@n2drbd ~]# ls
```

c.q.d..

```
[root@n1drbd ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0          11:0    1 1024M  0 rom
xvda         202:0    0   50G  0 disk
├─xvda1       202:1    0    1G  0 part /boot
└─xvda2       202:2    0   49G  0 part
   ├─centos-root 253:0    0 45,1G  0 lvm /
   └─centos-swap 253:1    0   3,9G  0 lvm [SWAP]
xvdb         202:16   0    10G  0 disk
├─xvdb1       202:17   0    10G  0 part
└─fileserver-r0 253:2    0    10G  0 lvm
   └─drbd0     147:0    0    10G  0 disk /clusterfs
```

```
[root@n1drbd ~]# pcs status
Cluster name: mycluster
Stack: corosync
Current DC: n2drbd.camcluster (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum
Last updated: Fri Sep  6 14:54:26 2019
Last change: Thu Sep  5 16:03:20 2019 by root via cibadmin on n1drbd.camcluster

2 nodes configured
8 resources configured

Online: [ n1drbd.camcluster n2drbd.camcluster ]

Full list of resources:

xvmfence_n1 (stonith:fence_xvm): Started n2drbd.camcluster
xvmfence_n2 (stonith:fence_xvm): Started n1drbd.camcluster
Clone Set: dlm-clone [dlm]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
Clone Set: clvmd-clone [clvmd]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]
Clone Set: gfs2fs-clone [gfs2fs]
  Started: [ n1drbd.camcluster n2drbd.camcluster ]

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

```
[root@n1drbd ~]# cat /proc/drbd
version: 8.4.11-1 (api:1/proto:86-101)
GIT-hash: 66145a308421e9c124ec391a7848ac20203bb03c build by mockbuild@, 2018-11-03 01:26:55
0: cs:Connected ro:Primary/Primary ds:UpToDate/UpToDate C r-----
   ns:10581208 nr:103068 dw:207064 dr:10503912 al:29 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:d oos:0
```