深度实践 KVM 之 DRBD 剖析

目 录

1. DRBD 介绍 3

1.1. DRBD 的定义及工作原理 3
1.2. DRBD 的复制模式（协议） 3

2. 测试环境搭建 4

2.1. 虚拟机的创建 4
2.2. 操作系统安装 5
2.3. 操作系统环境配置 5

3. DRBD 配置 5

3.1. 安装 ELREPO 源 5
3.2. 安装 DRBD 模块 7
3.3. 加载并查看 DRBD 模块 7
3.4. 修改配置文件 7
3.5. 重新启动 DRBD 服务 9
3.6. 创建资源 9
3.7. 其它命令 10

4. 安装和配置 COROSYNC 和 PACEMAKER 10

4.1. 安装 COROSYNC 和 PACEMAKER 10
4.2. 安装 CRMSH 12
4.3. 编辑 COROSYNC 配置文件 12
4.4. 创建 PACEMAKER 服务 13
4.5. 生成 AUTHKEY 文件 13
4.6. 查看集群运行状态 14
4.7. 其它命令 14

5. 配置集群服务 15

5.1. 配置 LVM 相关参数 15
5.2. 在 DRBD1 节点上面创建 PV、VG、LV 15
5.3. 安装 ISCSI TARGET 服务 16
5.4. 下载集群服务需要用到的 ISCSITARGET 脚本 17
5.5. 配置集群服务 17
5.6. 集群服务验证 20

6. 测试总结 25

近日抽了点时间拜读了力哥的宏篇大作《深度实践 KVM》一书中的部分章节，此书的内容相当丰富，确实给从事这个行业以及即将从事这个行业的人带来了不小的福利，因此我也认真的去研究了书中的部分操作，如分布式文件系统一篇中的 DRBD 部分，在测试过程中也发现了一些问题，现将我经过测试的步骤整理了一份分享给大家，供大家一起学习。

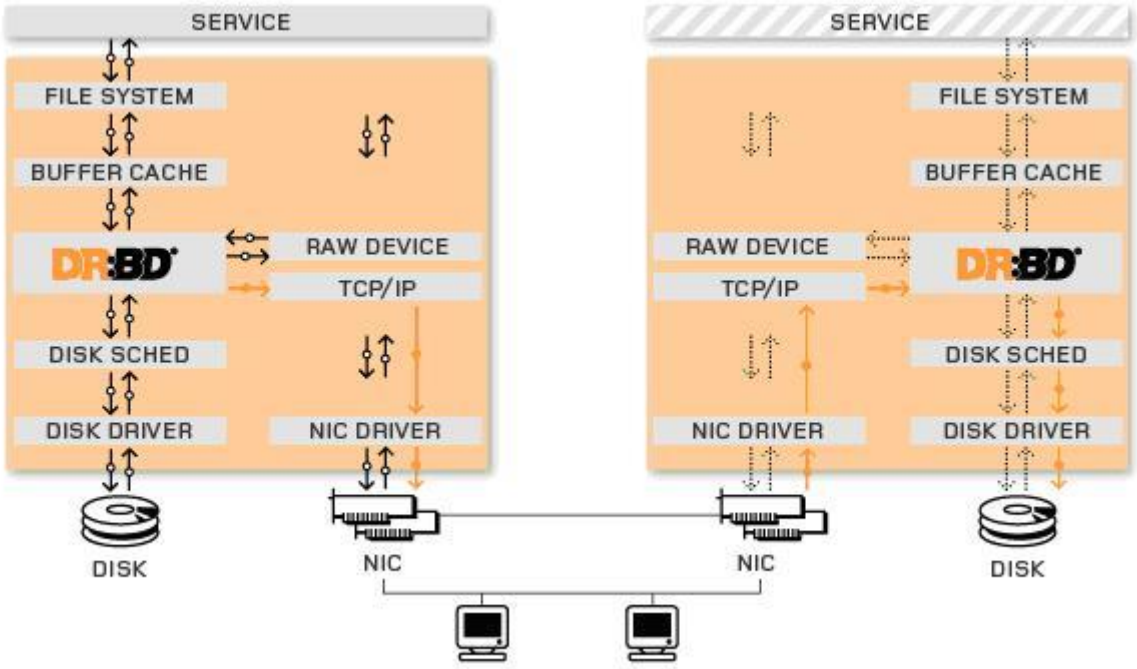
1. DRBD 介绍

1.1. DRBD 的定义及工作原理

所谓 DRBD(全称为 Distributed Replicated Block Device，简称 DRBD)，是分布式复制块设备的缩写，实际上是一种块设备的实现，主要被用于 Linux 平台下的高可用(HA)方案之中。他是有内核模块和相关程序而组成，通过网络通信来同步镜像整个设备，有点类似于一个网络 RAID-1 的功能。也就是说当你将数据写入本地的 DRBD 设备上的文件系统时，数据会同时被发送到网络中的另外一台主机之上，并以完全相同的形式记录在一个文件系统中（实际上文件系统的创建也是由 DRBD 的同步来实现的）。本地节点（主机）与远程节点（主机）的数据可以保证实时的同步，并保证 IO 的一致性。所以当本地节点的主机出现故障时，远程节点的主机上还会保留有一份完全相同的数据，可以继续使用，以达到高可用的目的。

在高可用(HA)解决方案中使用 DRBD 的功能，可以代替使用一个共享盘阵存储设备。因为数据同时存在于本地主机和远程主机上，在遇到需要切换的时候，远程主机只需要使用它上面的那份数据副本，就可以继续提供服务了。

实现原理如下图所示：



从上图我们可以清晰的看出 drbd 是以主从(Primary/Secondary)方式工作的。主节点上的 drbd 提升为 Primary 并负责接收写入数据，当数据到达 drbd 模块时，一份继续往下走写入到本地磁盘实现数据的持久化，同时并将接收到的要写入的数据发送一分到本地的 drbd 设备上通过 tcp 传到另外一台主机的 drbd 设备上（Secondary node），另一台主机上的对应的 drbd 设备再将接收到的数据存入到自己的磁盘当中。因此，drbd 对同一设备块每次只允许对主节点进行读、写操作，从节点不能写也不能读。

1.2. DRBD 的复制模式（协议）

A 协议：

异步复制协议。一旦本地磁盘写入已经完成，数据包已在发送队列中，则写被认为是完成的。在一个节点发生故障时，可能发生数据丢失，因为被写入到远程节点上的数据可能仍在发送队列。尽管，在故障转移节点上的数据是一致的，但没有及时更新。因此，这种模式效率最高，但是数据不安全，存在数据丢失。

B 协议：

内存同步（半同步）复制协议。一旦本地磁盘写入已完成且复制数据包达到了对等节点则认为写在主节点上被认为是完成的。数据丢失可能发生在参加的两个节点同时故障的情况下，因为在传输中的数据可能不会被提交到磁盘

C 协议：

同步复制协议。只有在本地和远程节点的磁盘已经确认了写操作完成，写才被认为完成。没有数据丢失，所以这是一个群集节点的流行模式，但 I/O 吞吐量依赖于网络带宽。因此，这种模式数据相对安全，但是效率比较低。

2. 测试环境搭建

2.1. 虚拟机的创建

本次测试均在虚拟平台通过虚拟机的方式来搭建，虚拟机基于 VMware Workstation 10，大家可以根据自身配置的高低，设置合适的配置，本次测试所用虚拟机的配置如下。

上图中硬盘包含两块，10GB 容量的磁盘用于操作系统安装，20GB 容量的磁盘用于 DRBD 复制，网卡配置在测试环境中可以使用 1 块，生产环境建议用多块设置成 bond 模式，满足

链路的冗余和复制对带宽的要求。

2.2. 操作系统安装

本次测试所使用的操作系统版本为 CentOS6.6 原版，操作系统的安装过程没有特殊要求，组件选择部分可以只选择 Desktop 或 Basic 模式进行安装，具体安装过程略。

2.3. 操作系统环境配置

系统安装结束后，设置好两台虚拟机的主机名和 IP 地址，本次测试的系统配置如下表所示：

虚拟机 / 选项	虚拟机 1	虚拟机 2
主机名	drbd1	drbd2
本机 IP 地址	192.168.8.15/24	192.168.8.16/24
浮动 IP 地址	192.168.8.17	

添加如下信息到两台主机的/etc/hosts 文件

192.168.8.15 drbd1

192.168.8.16 drbd2

因为配置 drbd 不需要用到图形桌面，因此设置开机不启动图形界面

sed -i 's/id:5:initdefault:/id:3:initdefault:/g' /etc/inittab

禁用 selinux

sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config

setenforce 0

禁用相关服务

chkconfig iptables off

chkconfig ip6tables off

chkconfig NetworkManager off

service iptables stop

service ip6tables stop

service NetworkManager stop

安装系统软件包

yum install -y binutils compat-libstdc++-33 elfutils-libelf elfutils-libelf-devel elfutils-libelf-devel-static gcc gcc-c++ glibc glibc-common glibc-devel glibc-headers kernel-headers ksh libaio libaio-devel libgcc libgomp libstdc++ libstdc++-devel libXp libXp-devel make

3. DRBD 配置

3.1. 安装 elrepo 源

rpm -Uvh http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm

也可以手动配置 elrepo 源

vi /etc/yum.repos.d/elrepo.repo

```
### Name: ELRepo.org Community Enterprise Linux Repository for el6
### URL: http://elrepo.org/

[elrepo]
name=ELRepo.org Community Enterprise Linux Repository - el6
baseurl=http://elrepo.org/linux/elrepo/el6/$basearch/
        http://mirrors.coreix.net/elrepo/elrepo/el6/$basearch/
        http://jur-linux.org/download/elrepo/elrepo/el6/$basearch/
        http://repos.lax-noc.com/elrepo/elrepo/el6/$basearch/
        http://mirror.ventraip.net.au/elrepo/elrepo/el6/$basearch/
mirrorlist=http://mirrors.elrepo.org/mirrors-elrepo.el6
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
protect=0

[elrepo-testing]
name=ELRepo.org Community Enterprise Linux Testing Repository - el6
baseurl=http://elrepo.org/linux/testing/el6/$basearch/
        http://mirrors.coreix.net/elrepo/testing/el6/$basearch/
        http://jur-linux.org/download/elrepo/testing/el6/$basearch/
        http://repos.lax-noc.com/elrepo/testing/el6/$basearch/
        http://mirror.ventraip.net.au/elrepo/testing/el6/$basearch/
mirrorlist=http://mirrors.elrepo.org/mirrors-elrepo-testing.el6
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
protect=0

[elrepo-kernel]
name=ELRepo.org Community Enterprise Linux Kernel Repository - el6
baseurl=http://elrepo.org/linux/kernel/el6/$basearch/
        http://mirrors.coreix.net/elrepo/kernel/el6/$basearch/
        http://jur-linux.org/download/elrepo/kernel/el6/$basearch/
        http://repos.lax-noc.com/elrepo/kernel/el6/$basearch/
        http://mirror.ventraip.net.au/elrepo/kernel/el6/$basearch/
mirrorlist=http://mirrors.elrepo.org/mirrors-elrepo-kernel.el6
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
protect=0

[elrepo-extras]
name=ELRepo.org Community Enterprise Linux Extras Repository - el6
baseurl=http://elrepo.org/linux/extras/el6/$basearch/
```

http://mirrors.coreix.net/elrepo/extras/el6/$basearch/
http://jur-linux.org/download/elrepo/extras/el6/$basearch/
http://repos.lax-noc.com/elrepo/extras/el6/$basearch/
http://mirror.ventraip.net.au/elrepo/extras/el6/$basearch/
mirrorlist=http://mirrors.elrepo.org/mirrors-elrepo-extras.el6
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-elrepo.org
protect=0

```
[root@drbd1 ~]# rpm -Uvh http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noa
rch.rpm
Retrieving http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm
warning: /var/tmp/rpm-tmp.qs5fO9: Header V4 DSA/SHA1 Signature, key ID baadae52:
 NOKEY
Preparing...                ########################################### [100%]
   1:elrepo-release         ########################################### [100%]
```

```
[root@drbd2 ~]# rpm -Uvh http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noa
rch.rpm
Retrieving http://www.elrepo.org/elrepo-release-6-6.el6.elrepo.noarch.rpm
warning: /var/tmp/rpm-tmp.LnI8TP: Header V4 DSA/SHA1 Signature, key ID baadae52:
 NOKEY
Preparing...                ########################################### [100%]
   1:elrepo-release         ########################################### [100%]
```

3.2. 安装 drbd 模块

yum install -y drbd84-utils kmod-drbd84

```
  Installing : drbd84-utils-8.9.2-1.el6.elrepo.x86_64                       1/2
  Installing : kmod-drbd84-8.4.6-1.el6.elrepo.x86_64                        2/2
Working. This may take some time ...
Done.
  Verifying  : drbd84-utils-8.9.2-1.el6.elrepo.x86_64                       1/2
  Verifying  : kmod-drbd84-8.4.6-1.el6.elrepo.x86_64                        2/2

Installed:
  drbd84-utils.x86_64 0:8.9.2-1.el6.elrepo
  kmod-drbd84.x86_64 0:8.4.6-1.el6.elrepo

Complete!
```

3.3. 加载并查看 drbd 模块

modprobe drbd

lsmod | grep drbd

```
[root@drbd1 ~]# modprobe drbd
[root@drbd1 ~]# lsmod | grep drbd
drbd                  365931  0
libcrc32c               1246  1 drbd
```

```
[root@drbd2 ~]# modprobe drbd
[root@drbd2 ~]# lsmod | grep drbd
drbd                  365931  0
libcrc32c               1246  1 drbd
```

3.4. 修改配置文件

编辑/etc/drbd.conf，修改内容如下

vi /etc/drbd.conf

```
# You can find an example in    /usr/share/doc/drbd.../drbd.conf.example

#include "drbd.d/global_common.conf";     #禁用此行
include "drbd.d/*.res";
```

```
[root@drbd1 ~]# cat /etc/drbd.conf
# You can find an example in  /usr/share/doc/drbd.../drbd.conf.example

#include "drbd.d/global_common.conf";
include "drbd.d/*.res";
```

```
[root@drbd2 ~]# cat /etc/drbd.conf
# You can find an example in  /usr/share/doc/drbd.../drbd.conf.example

#include "drbd.d/global_common.conf";
include "drbd.d/*.res";
```

新建 drbd 资源配置文件，其中部分参数的设置需要根据实际环境来设置，可参考官方网站

vi /etc/drbd.d/clustervol.res

内容如下：

```
global {
        usage-count yes;
}
common {
        protocol C;
        disk {
                on-io-error detach;
                fencing resource-only;
        }
        net {
                cram-hmac-alg sha1;
                shared-secret "a6a0680c40bca2439dbe48343ddddcf4";
        }
        syncer {
                rate 100M;
        }
        handlers {
                fence-peer "/usr/lib/drbd/crm-fence-peer.sh";
                after-resync-target "/usr/lib/drbd/crm-unfence-peer.sh";
                pri-on-incon-degr "echo b > /proc/sysrq-trigger";
        }
}
resource clustervol {
        device /dev/drbd1;
        disk /dev/sdb1;
        meta-disk internal;
        net {
                max-buffers 8192;
                max-epoch-size 8192;
```

```
                sndbuf-size 2048k;
                unplug-watermark 127;
        }
        disk {
                disk-barrier no;
                disk-flushes no;
                resync-rate 100M;
                c-plan-ahead 200;
                c-max-rate 100M;
                c-min-rate 10M;
                c-fill-target 100M;
        }
        on drbd1 {
                address ipv4 192.168.8.15:7898;
        }
        on drbd2 {
                address ipv4 192.168.8.16:7898;
        }
}
```

3.5. 重新启动 drbd 服务

重启服务之前，我们需要对/dev/sdb 进行分区，创建/dev/sdb1 供 drbd 使用。

service drbd status

```
[root@drbd2 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015
-04-09 14:35:00
m:res  cs  ro  ds  p  mounted  fstype
```

service drbd restart

service drbd status

```
[root@drbd2 ~]# service drbd status
drbd driver loaded OK; device status:
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015
-04-09 14:35:00
m:res        cs         ro                 ds                 p  mounted  fst
ype
1:clustervol  Connected  Secondary/Secondary  Diskless/Diskless  C
```

重新 drbd 服务后，就可以查看到我们所配置的资源了

3.6. 创建资源

drbdadm create-md clustervol

drbdadm adjust clustervol

创建完资源后两台节点中/dev/sdb1 的数据为不一致状态，使用如下将其中一个节点提升为主节点后开始初始化同步数据，同步过程中可以输入下面的命令查看 drbd 的状态，可以看到当前正在同步数据

初次创建资源完成需要用如下命令提升当前节点为主

drbdsetup /dev/drbd1 primary --o

或

drbdadm -- --overwrite-data-of-peer primary all

```
[root@drbd1 ~]# drbdadm -- --overwrite-data-of-peer primary all
[root@drbd1 ~]# cat /proc/drbd
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00

 1: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----
    ns:44032 nr:0 dw:0 dr:44088 al:0 bm:0 lo:19 pe:21 ua:16 ap:1 ep:1 wo:d oos:20931092
        [>....................] sync'ed:  0.2% (20440/20472)M
        finish: 0:10:30 speed: 33,024 (33,024) K/sec
```

完成后的状态如下

cat /proc/drbd

```
[root@drbd1 ~]# cat /proc/drbd
version: 8.4.6 (api:1/proto:86-101)
GIT-hash: 833d830e0152d1e457fa7856e71e11248ccf3f70 build by phil@Build64R6, 2015-04-09 14:35:00

 1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
    ns:20967620 nr:0 dw:0 dr:20985448 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:d oos:0
```

3.7. 其它命令

设置节点对象为主

drbdsetup /dev/drbd1 primary

或

drbdadm primary clustervol

设置节点对象为从

drbdadm secondary all

分离 drbd 磁盘

drbdadm detach clustervol

查看资源

drbd-overview

```
[root@drbd2 ~]# drbd-overview
 1:clustervol/0  Connected Secondary/Primary UpToDate/UpToDate
```

4. 安装和配置 corosync 和 pacemaker

4.1. 安装 corosync 和 pacemaker

yum install -y libibverbs librdmacm lm_sensors libtool-ltdl openhpi-libs openhpi perl-TimeDate

```
  Installing : openhpi-libs-2.14.1-6.el6.x86_64                              1/9
  Installing : OpenIPMI-libs-2.0.16-14.el6.x86_64                            2/9
  Installing : libsysfs-2.1.0-7.el6.x86_64                                   3/9
  Installing : rdma-6.7_3.15-5.el6.noarch                                    4/9
  Installing : libibverbs-1.1.8-4.el6.x86_64                                 5/9
  Installing : librdmacm-1.0.19.1-1.el6.x86_64                               6/9
  Installing : openhpi-2.14.1-6.el6.x86_64                                   7/9
  Installing : lm_sensors-3.1.1-17.el6.x86_64                                8/9
  Installing : 1:perl-TimeDate-1.16-13.el6.noarch                            9/9
  Verifying  : rdma-6.7_3.15-5.el6.noarch                                    1/9
  Verifying  : 1:perl-TimeDate-1.16-13.el6.noarch                            2/9
  Verifying  : libibverbs-1.1.8-4.el6.x86_64                                 3/9
  Verifying  : libsysfs-2.1.0-7.el6.x86_64                                   4/9
  Verifying  : librdmacm-1.0.19.1-1.el6.x86_64                               5/9
  Verifying  : openhpi-2.14.1-6.el6.x86_64                                   6/9
  Verifying  : OpenIPMI-libs-2.0.16-14.el6.x86_64                            7/9
  Verifying  : lm_sensors-3.1.1-17.el6.x86_64                                8/9
  Verifying  : openhpi-libs-2.14.1-6.el6.x86_64                              9/9

Installed:
  libibverbs.x86_64 0:1.1.8-4.el6        librdmacm.x86_64 0:1.0.19.1-1.el6
  lm_sensors.x86_64 0:3.1.1-17.el6       openhpi.x86_64 0:2.14.1-6.el6
  openhpi-libs.x86_64 0:2.14.1-6.el6     perl-TimeDate.noarch 1:1.16-13.el6

Dependency Installed:
  OpenIPMI-libs.x86_64 0:2.0.16-14.el6       libsysfs.x86_64 0:2.1.0-7.el6
  rdma.noarch 0:6.7_3.15-5.el6

Complete!
```

yum install -y pacemaker pacemaker-libs corosync corosynclib

```
  Installing : libqb-0.17.1-1.el6.x86_64                                     1/9
  Installing : pacemaker-libs-1.1.12-8.el6.x86_64                            2/9
  Installing : corosynclib-1.4.7-2.el6.x86_64                                3/9
  Installing : corosync-1.4.7-2.el6.x86_64                                   4/9
  Installing : clusterlib-3.0.12.1-73.el6.1.x86_64                           5/9
  Installing : pacemaker-cli-1.1.12-8.el6.x86_64                             6/9
  Installing : pacemaker-cluster-libs-1.1.12-8.el6.x86_64                    7/9
  Installing : resource-agents-3.9.5-24.el6.x86_64                           8/9
  Installing : pacemaker-1.1.12-8.el6.x86_64                                 9/9
  Verifying  : pacemaker-libs-1.1.12-8.el6.x86_64                            1/9
  Verifying  : resource-agents-3.9.5-24.el6.x86_64                           2/9
  Verifying  : pacemaker-cli-1.1.12-8.el6.x86_64                             3/9
  Verifying  : pacemaker-1.1.12-8.el6.x86_64                                 4/9
  Verifying  : pacemaker-cluster-libs-1.1.12-8.el6.x86_64                    5/9
  Verifying  : corosync-1.4.7-2.el6.x86_64                                   6/9
  Verifying  : clusterlib-3.0.12.1-73.el6.1.x86_64                           7/9
  Verifying  : corosynclib-1.4.7-2.el6.x86_64                                8/9
  Verifying  : libqb-0.17.1-1.el6.x86_64                                     9/9

Installed:
  corosync.x86_64 0:1.4.7-2.el6          corosynclib.x86_64 0:1.4.7-2.el6
  pacemaker.x86_64 0:1.1.12-8.el6        pacemaker-libs.x86_64 0:1.1.12-8.el6

Dependency Installed:
  clusterlib.x86_64 0:3.0.12.1-73.el6.1
  libqb.x86_64 0:0.17.1-1.el6
  pacemaker-cli.x86_64 0:1.1.12-8.el6
  pacemaker-cluster-libs.x86_64 0:1.1.12-8.el6
  resource-agents.x86_64 0:3.9.5-24.el6

Complete!
```

4.2. 安装 crmsh

下载 repo 源文件

wget -P /etc/yum.repos.d/ http://download.opensuse.org/repositories/network:/ha-clustering:/Stable/CentOS_CentOS-6/network:ha-clustering:Stable.repo

```
[root@drbd1 ~]# wget -P /etc/yum.repos.d/ http://download.opensuse.org/repositor
ies/network:/ha-clustering:/Stable/CentOS_CentOS-6/network:ha-clustering:Stable.
repo
--2015-10-21 19:12:43--  http://download.opensuse.org/repositories/network:/ha-c
lustering:/Stable/CentOS_CentOS-6/network:ha-clustering:Stable.repo
Resolving download.opensuse.org... 195.135.221.134, 2001:67c:2178:8::13
Connecting to download.opensuse.org|195.135.221.134|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 350 [text/plain]
Saving to: "/etc/yum.repos.d/network:ha-clustering:Stable.repo"

100%[======================================>] 350         --.-K/s   in 0s

2015-10-21 19:12:45 (16.4 MB/s) - "/etc/yum.repos.d/network:ha-clustering:Stable
.repo" saved [350/350]
```

安装 crmsh

yum install -y crmsh

```
  Installing : python-pssh-2.3.1-4.2.x86_64                                 1/5
  Installing : pssh-2.3.1-4.2.x86_64                                        2/5
  Installing : redhat-rpm-config-9.0.3-44.el6.centos.noarch                 3/5
  Installing : python-dateutil-1.4.1-6.el6.noarch                           4/5
  Installing : crmsh-2.1-1.6.x86_64                                         5/5
  Verifying  : crmsh-2.1-1.6.x86_64                                         1/5
  Verifying  : python-dateutil-1.4.1-6.el6.noarch                           2/5
  Verifying  : pssh-2.3.1-4.2.x86_64                                        3/5
  Verifying  : redhat-rpm-config-9.0.3-44.el6.centos.noarch                 4/5
  Verifying  : python-pssh-2.3.1-4.2.x86_64                                 5/5

Installed:
  crmsh.x86_64 0:2.1-1.6

Dependency Installed:
  pssh.x86_64 0:2.3.1-4.2        python-dateutil.noarch 0:1.4.1-6.el6
  python-pssh.x86_64 0:2.3.1-4.2 redhat-rpm-config.noarch 0:9.0.3-44.el6.centos

Complete!
```

重命名刚才下载的 repo 源文件，我们以后还是使用原来的源进行 yum 安装

mv /etc/yum.repos.d/network:ha-clustering:Stable.repo /etc/yum.repos.d/network:ha-clustering:Stable.repo.bak

4.3. 编辑 corosync 配置文件

vi /etc/corosync/corosync.conf

```
aisexec {
        user: root
        group: root
}
totem {
        version: 2
```

```
        secauth: off
        threads: 0
        interface {
                ringnumber: 0
                bindnetaddr: 192.168.8.0      #业务网络
                mcastaddr: 239.255.1.1
                mcastport: 5405
                ttl: 1
        }
}
logging {
        fileline: off
        to_stderr: no
        to_logfile: yes
        logfile: /var/log/cluster/corosync.log
        to_syslog: no
        debug: off
        timestamp: on
        logger_subsys {
                subsys: AMF
                debug: off
        }
}
amf {
        mode: disabled
}
```

4.4. 创建 pacemaker 服务

vi /etc/corosync/service.d/pacemaker

```
service {
        name: pacemaker
        ver: 1
        #user_mgmtd: yes
}
```

4.5. 生成 authkey 文件

在其中一个节点执行下面的命令

corosync-keygen

为了加快生成速度，需要另开一个会话窗口，执行如下命令

tar cvj / | md5sum > /dev/null

```
[root@drbd1 ~]# corosync-keygen
Corosync Cluster Engine Authentication key generator.
Gathering 1024 bits for key from /dev/random.
Press keys on your keyboard to generate entropy.
Press keys on your keyboard to generate entropy (bits = 272).
Press keys on your keyboard to generate entropy (bits = 336).
Press keys on your keyboard to generate entropy (bits = 400).
Press keys on your keyboard to generate entropy (bits = 464).
Press keys on your keyboard to generate entropy (bits = 528).
Press keys on your keyboard to generate entropy (bits = 592).
Press keys on your keyboard to generate entropy (bits = 656).
Press keys on your keyboard to generate entropy (bits = 720).
Press keys on your keyboard to generate entropy (bits = 784).
Press keys on your keyboard to generate entropy (bits = 848).
Press keys on your keyboard to generate entropy (bits = 912).
Press keys on your keyboard to generate entropy (bits = 976).
Writing corosync key to /etc/corosync/authkey.
```

复制 authkey 文件到另一台主机

scp /etc/corosync/authkey drbd2:/etc/corosync/

修改集群日志属性

chown -R hacluster. /var/log/cluster/

启动集群服务

service corosync start

service pacemaker start

```
[root@drbd1 ~]# service corosync start
Starting Corosync Cluster Engine (corosync):               [  OK  ]
[root@drbd1 ~]# service pacemaker start
Starting Pacemaker Cluster Manager                         [  OK  ]
```

```
[root@drbd2 ~]# service corosync start
Starting Corosync Cluster Engine (corosync):               [  OK  ]
[root@drbd2 ~]# service pacemaker start
Starting Pacemaker Cluster Manager                         [  OK  ]
```

设置开机自启动

chkconfig corosync on

chkconfig pacemaker on

4.6. 查看集群运行状态

crm_mon

```
Last updated: Wed Oct 21 19:34:36 2015
Last change: Wed Oct 21 19:33:45 2015
Stack: classic openais (with plugin)
Current DC: drbd1 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
0 Resources configured


Online: [ drbd1 drbd2 ]
```

4.7. 其它命令

查看 corosync 是否运行正确

corosync-objctl | grep members | grep ip

查看集群常用日志命令

grep -e "Corosync Cluster Engine" -e "configuration file" /var/log/cluster/corosync.log
grep TOTEM /var/log/cluster/corosync.log
grep pcmk_startup /var/log/cluster/corosync.log
grep ERROR: /var/log/cluster/corosync.log | grep -v unpack_resources
tail -F /var/log/cluster/corosync.log

5. 配置集群服务

5.1. 配置 lvm 相关参数

为了不出现 LVM on LVM 的情况，需要设置磁盘过滤，并禁用宿主机的 LVM 写缓存修改 lvm 配置文件内容如下，使系统不扫描 drbd 所使用的/dev/sdb1 上面的 lvm 信息

vi /etc/lvm/lvm.conf

filter = ["r|/dev/sdb1.*|"]
write_cache_state = 0

清空已有的 lvm 缓存

rm -rf /etc/lvm/cache/*

5.2. 在 drbd1 节点上面创建 pv、vg、lv

将 drbd 生成的/dev/drbd1 同步盘创建成 pv 卷

pvcreate /dev/drbd1

```
[root@drbd1 ~]# pvcreate /dev/drbd1
  Physical volume "/dev/drbd1" successfully created
```

使用/dev/drbd1 创建成 VG: vg_iscsi

vgcreate vg_iscsi /dev/drbd1

```
[root@drbd1 ~]# vgcreate vg_iscsi /dev/drbd1
  Volume group "vg_iscsi" successfully created
```

在 VG: vg_iscsi 上创建 LV: iscsilun1，分配 10G 空间大小

lvcreate -L 10G -n iscsilun1 vg_iscsi

```
[root@drbd1 ~]# lvcreate -L 10G -n iscsilun1 vg_iscsi
  Logical volume "iscsilun1" created
```

查看两个节点的 lvm

```
[root@drbd1 ~]# pvscan
  PV /dev/sda2    VG vg_drbd1   lvm2 [9.51 GiB / 0    free]
  PV /dev/drbd1   VG vg_iscsi   lvm2 [19.99 GiB / 9.99 GiB free]
  Total: 2 [29.50 GiB] / in use: 2 [29.50 GiB] / in no VG: 0 [0   ]
[root@drbd1 ~]# vgscan
  Reading all physical volumes.  This may take a while...
  Found volume group "vg_drbd1" using metadata type lvm2
  Found volume group "vg_iscsi" using metadata type lvm2
[root@drbd1 ~]# lvscan
  ACTIVE            '/dev/vg_drbd1/lv_root' [8.51 GiB] inherit
  ACTIVE            '/dev/vg_drbd1/lv_swap' [1.00 GiB] inherit
  ACTIVE            '/dev/vg_iscsi/iscsilun1' [10.00 GiB] inherit
```

```
[root@drbd2 ~]# pvscan
  PV /dev/sdb1   VG vg_iscsi   lvm2 [19.99 GiB / 9.99 GiB free]
  PV /dev/sda2   VG vg_drbd2   lvm2 [9.51 GiB / 0    free]
  Total: 2 [29.50 GiB] / in use: 2 [29.50 GiB] / in no VG: 0 [0   ]
[root@drbd2 ~]# vgscan
  Reading all physical volumes.  This may take a while...
  Found volume group "vg_iscsi" using metadata type lvm2
  Found volume group "vg_drbd2" using metadata type lvm2
[root@drbd2 ~]# lvscan
  inactive          '/dev/vg_iscsi/iscsilun1' [10.00 GiB] inherit
  ACTIVE            '/dev/vg_drbd2/lv_root' [8.51 GiB] inherit
  ACTIVE            '/dev/vg_drbd2/lv_swap' [1.00 GiB] inherit
```

我们可以看到 drbd1 节点上面的 lvm 信息自动同步到 drdb2 节点，但是 drbd2 节点上面的 vg 和 lv 处于非激活状态，可以通过 lvm 相关命令先取消 drbd1 上面的 vg 激活状态，然后再 drbd2 节点上面激活 vg，因为 pv 基于 drbd 生成，因此需要配合 drbd 命令一起使用。

在 drbd1 节点取消激活并设置 drbd 为从

vgchange -an vg_iscsi

drbdsetup /dev/drbd1 secondary

在 drbd2 节点设置 drbd 为主，并激活 vg

drbdsetup /dev/drbd1 primary

vgchange -ay vg_iscsi

5.3. 安装 iSCSI Target 服务

yum install -y scsi-target-utils.x86_64

```
  Updating   : sg3_utils-libs-1.28-8.el6.x86_64                          1/5
  Installing : sg3_utils-1.28-8.el6.x86_64                               2/5
  Installing : perl-Config-General-2.52-1.el6.noarch                     3/5
  Installing : scsi-target-utils-1.0.24-16.el6.x86_64                    4/5
  Cleanup    : sg3_utils-libs-1.28-6.el6.x86_64                          5/5
  Verifying  : sg3_utils-1.28-8.el6.x86_64                               1/5
  Verifying  : perl-Config-General-2.52-1.el6.noarch                     2/5
  Verifying  : sg3_utils-libs-1.28-8.el6.x86_64                          3/5
  Verifying  : scsi-target-utils-1.0.24-16.el6.x86_64                    4/5
  Verifying  : sg3_utils-libs-1.28-6.el6.x86_64                          5/5

Installed:
  scsi-target-utils.x86_64 0:1.0.24-16.el6

Dependency Installed:
  perl-Config-General.noarch 0:2.52-1.el6     sg3_utils.x86_64 0:1.28-8.el6

Dependency Updated:
  sg3_utils-libs.x86_64 0:1.28-8.el6

Complete!
```

启动 iSCSI target 服务并设置为开机自启动

service tgtd start

chkconfig tgtd on

```
[root@drbd1 ~]# service tgtd start
Starting SCSI target daemon:                               [  OK  ]
[root@drbd1 ~]# chkconfig tgtd on
```

```
[root@drbd2 ~]# service tgtd start
Starting SCSI target daemon:                               [  OK  ]
[root@drbd2 ~]# chkconfig tgtd on
```

5.4. 下载集群服务需要用到的 iSCSITarget 脚本

安装 git 程序

yum install -y git

```
  Installing : 1:perl-Error-0.17015-4.el6.noarch                 1/3
  Installing : perl-Git-1.7.1-3.el6_4.1.noarch                   2/3
  Installing : git-1.7.1-3.el6_4.1.x86_64                        3/3
  Verifying  : git-1.7.1-3.el6_4.1.x86_64                        1/3
  Verifying  : perl-Git-1.7.1-3.el6_4.1.noarch                   2/3
  Verifying  : 1:perl-Error-0.17015-4.el6.noarch                 3/3

Installed:
  git.x86_64 0:1.7.1-3.el6_4.1

Dependency Installed:
  perl-Error.noarch 1:0.17015-4.el6      perl-Git.noarch 0:1.7.1-3.el6_4.1

Complete!
```

下载脚本

git clone https://github.com/ClusterLabs/resource-agents

复制下载的 iSCSITarget 脚本到本地和 drbd2 节点的 heartbeat 目录下面

cp resource-agents/heartbeat/iSCSITarget /usr/lib/ocf/resource.d/heartbeat/

scp resource-agents/heartbeat/iSCSITarget drbd2:/usr/lib/ocf/resource.d/heartbeat/

```
[root@drbd1 ~]# cp resource-agents/heartbeat/iSCSITarget /usr/lib/ocf/resource.d
/heartbeat/
[root@drbd1 ~]# scp resource-agents/heartbeat/iSCSITarget drbd2:/usr/lib/ocf/res
ource.d/heartbeat/
The authenticity of host 'drbd2 (192.168.8.16)' can't be established.
RSA key fingerprint is fe:e5:2f:6a:54:3b:cb:13:28:b1:4f:4f:9e:64:59:00.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'drbd2,192.168.8.16' (RSA) to the list of known hosts
.
root@drbd2's password:
iSCSITarget                                   100%   21KB  21.0KB/s   00:00
```

5.5. 配置集群服务

配置集群相关属性

 # crm

crm(live)# **configure**

crm(live)configure# **property stonith-enabled="false"**

crm(live)configure# **property no-quorum-policy="ignore"**

crm(live)configure# **property default-resource-stickiness="200"**

crm(live)configure# **commit**

```
[root@drbd2 ~]# crm
crm(live)# configure
crm(live)configure# property stonith-enabled="false"
crm(live)configure# property no-quorum-policy="ignore"
crm(live)configure# property default-resource-stickiness="200"
crm(live)configure# commit
```

配置 drbd 资源 p_drbd_clustervol

crm(live)configure# **primitive p_drbd_clustervol **
**ocf:linbit:drbd **
**params drbd_resource="clustervol" **
**op monitor interval="29" role="Master" **
op monitor interval="31" role="Slave"

```
crm(live)configure# primitive p_drbd_clustervol \
   > ocf:linbit:drbd \
   > params drbd_resource="clustervol" \
   > op monitor interval="29" role="Master" \
   > op monitor interval="31" role="Slave"
```

配置基于 p_drbd_clustervol 的主从资源 ms_drbd_clustervol

crm(live)configure# **ms ms_drbd_clustervol p_drbd_clustervol **
**meta master-max="1" master-node-max="1" clone-max="2" **
clone-node-max="1" notify="true"

```
crm(live)configure# ms ms_drbd_clustervol p_drbd_clustervol \
   > meta master-max="1" master-node-max="1" clone-max="2" \
   > clone-node-max="1" notify="true"
```

配置 IP 资源 p_ip_clustervolip

crm(live)configure# **primitive p_ip_clustervolip **
**ocf:heartbeat:IPaddr2 **
**params ip="192.168.8.17" cidr_netmask="24" **
op monitor interval="10s"

```
crm(live)configure# primitive p_ip_clustervolip \
   > ocf:heartbeat:IPaddr2 \
   > params ip="192.168.8.17" cidr_netmask="24" \
   > op monitor interval="10s"
```

配置 LVM 卷组资源 p_lvm_vg_iscsi

crm(live)configure# **primitive p_lvm_vg_iscsi **
**ocf:heartbeat:LVM **
**params volgrpname="vg_iscsi" **
op monitor interval="30s" timeout="60s" depth="0"

```
crm(live)configure# primitive p_lvm_vg_iscsi \
   > ocf:heartbeat:LVM \
   > params volgrpname="vg_iscsi" \
   > op monitor interval="30s" timeout="60s" depth="0"
```

配置 iscsi target 资源 p_target_clustervol

crm(live)configure# **primitive p_target_clustervol **

**ocf:heartbeat:iSCSITarget **
**params iqn="iqn.1994-05.com.redhat:clustervol.iscsilun1" **
**tid="1" **
op monitor interval="10s" timeout="30s"

```
crm(live)configure# primitive p_target_clustervol \
   > ocf:heartbeat:iSCSITarget \
   > params iqn="iqn.1994-05.com.redhat:clustervol.iscsilun1" \
   > tid="1" \
   > op monitor interval="10s" timeout="30s"
```

添加 Logical Unit 资源 p_lu_iscsilun1
crm(live)configure# **primitive p_lu_iscsilun1 ocf:heartbeat:iSCSILogicalUnit **
**params target_iqn="iqn.1994-05.com.redhat:clustervol.iscsilun1" lun="1" path="/dev/vg_iscsi/iscsilun1" implementation="tgt" **
op monitor interval="10s"

```
crm(live)configure# primitive p_lu_iscsilun1 ocf:heartbeat:iSCSILogicalUnit \
   > params target_iqn="iqn.1994-05.com.redhat:clustervol.iscsilun1" lun="1" pat
h="/dev/vg_iscsi/iscsilun1" implementation="tgt" \
   > op monitor interval="10s"
```

创建资源组
crm(live)configure# **group rg_clustervol **
**p_lvm_vg_iscsi **
**p_target_clustervol p_lu_iscsilun1 **
p_ip_clustervolip

```
crm(live)configure# group rg_clustervol \
   > p_lvm_vg_iscsi \
   > p_target_clustervol p_lu_iscsilun1 \
   > p_ip_clustervolip
```

指定资源组默认在 drbd 的主节点启动
crm(live)configure# **colocation c_clustervol_on_drbd **
inf: rg_clustervol ms_drbd_clustervol:Master

```
crm(live)configure# colocation c_clustervol_on_drbd \
   > inf: rg_clustervol ms_drbd_clustervol:Master
```

crm(live)configure# **order o_drbd_before_clustervol **
inf: ms_drbd_clustervol:promote rg_clustervol:start

```
crm(live)configure# order o_drbd_before_clustervol \
   > inf: ms_drbd_clustervol:promote rg_clustervol:start
```

提交配置
crm(live)configure# **commit**

提交完成可以看到集群中所有服务都已经在 drbd1 节点上运行

```
Last updated: Wed Oct 21 21:12:47 2015
Last change: Wed Oct 21 21:12:44 2015
Stack: classic openais (with plugin)
Current DC: drbd1 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
6 Resources configured


Online: [ drbd1 drbd2 ]

 Master/Slave Set: ms_drbd_clustervol [p_drbd_clustervol]
     Masters: [ drbd1 ]
     Slaves: [ drbd2 ]
 Resource Group: rg_clustervol
     p_lvm_vg_iscsi     (ocf::heartbeat:LVM):   Started drbd1
     p_target_clustervol        (ocf::heartbeat:iSCSITarget):   Started drbd1
     p_lu_iscsilun1     (ocf::heartbeat:iSCSILogicalUnit):      Started drbd1
     p_ip_clustervolip  (ocf::heartbeat:IPaddr2):       Started drbd1
```

5.6. 集群服务验证

通过 **ip a** 命令可以查看虚拟 IP 已经在 drbd1 上启动

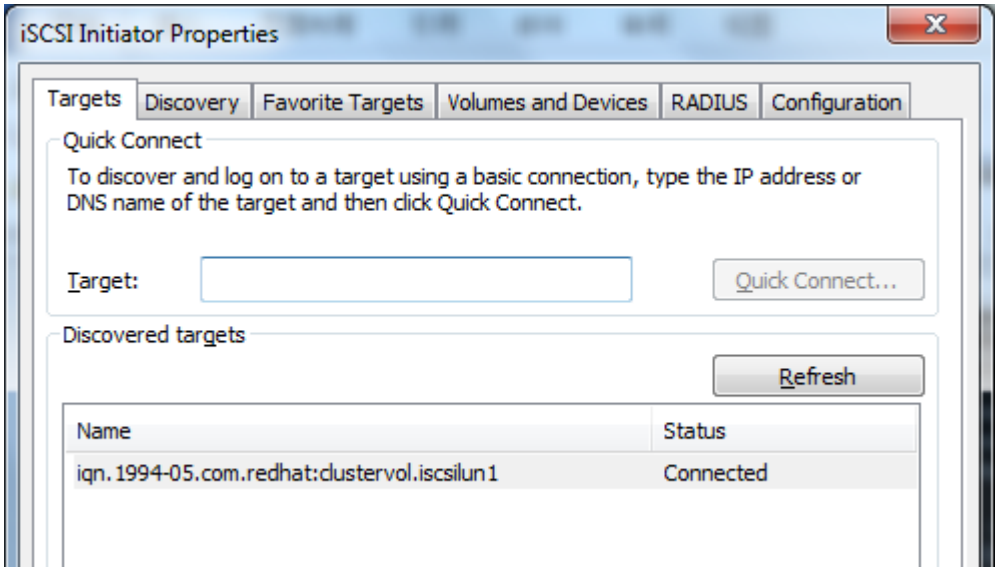
```
[root@drbd1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 00:0c:29:0a:92:fd brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.15/24 brd 192.168.8.255 scope global eth0
    inet 192.168.8.17/24 brd 192.168.8.255 scope global secondary eth0
    inet6 fe80::20c:29ff:fe0a:92fd/64 scope link
       valid_lft forever preferred_lft forever
```

通过 tgtadm 命令可以查看到 iSCSI 已经提供卷的映射

tgtadm --lld iscsi --mode target --op show

```
[root@drbd1 ~]# tgtadm --lld iscsi --mode target --op show
Target 1: iqn.1994-05.com.redhat:clustervol.iscsilun1
    System information:
        Driver: iscsi
        State: ready
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: IET     00010000
            SCSI SN: beaf10
            Size: 0 MB, Block size: 1
            Online: Yes
            Removable media: No
            Prevent removal: No
            Readonly: No
            Backing store type: null
            Backing store path: None
            Backing store flags:
        LUN: 1
            Type: disk
            SCSI ID: p_lu_iscsilun1
            SCSI SN: 9b544a8e
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Prevent removal: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /dev/vg_iscsi/iscsilun1
            Backing store flags:
    Account information:
    ACL information:
        ALL
```

通过 windows 的 iscsi 启动器已经可以成功连接 iscsi 卷



关闭 drbd1 节点主机，集群资源自动切换到 drbd2 节点上面，并且显示 drbd1 节点处于离线状态

```
Last updated: Wed Oct 21 21:22:20 2015
Last change: Wed Oct 21 21:22:12 2015
Stack: classic openais (with plugin)
Current DC: drbd2 - partition WITHOUT quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
6 Resources configured


Online: [ drbd2 ]
OFFLINE: [ drbd1 ]

 Master/Slave Set: ms_drbd_clustervol [p_drbd_clustervol]
     Masters: [ drbd2 ]
     Stopped: [ drbd1 ]
 Resource Group: rg_clustervol
     p_lvm_vg_iscsi     (ocf::heartbeat:LVM):   Started drbd2
     p_target_clustervol        (ocf::heartbeat:iSCSITarget):   Started drbd2
     p_lu_iscsilun1     (ocf::heartbeat:iSCSILogicalUnit):      Started drbd2
     p_ip_clustervolip  (ocf::heartbeat:IPaddr2):       Started drbd2
```

在 drbd2 上面可以正常查看到虚拟 IP

```
[root@drbd2 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP ql
en 1000
    link/ether 00:0c:29:65:24:73 brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.16/24 brd 192.168.8.255 scope global eth0
    inet 192.168.8.17/24 brd 192.168.8.255 scope global secondary eth0
    inet6 fe80::20c:29ff:fe65:2473/64 scope link
       valid_lft forever preferred_lft forever
```

在 drbd2 上面可以正常查看到卷的映射信息

```
[root@drbd2 ~]# tgtadm --lld iscsi --mode target --op show
Target 1: iqn.1994-05.com.redhat:clustervol.iscsilun1
    System information:
        Driver: iscsi
        State: ready
    I_T nexus information:
    LUN information:
        LUN: 0
            Type: controller
            SCSI ID: IET     00010000
            SCSI SN: beaf10
            Size: 0 MB, Block size: 1
            Online: Yes
            Removable media: No
            Prevent removal: No
            Readonly: No
            Backing store type: null
            Backing store path: None
            Backing store flags:
        LUN: 1
            Type: disk
            SCSI ID: p_lu_iscsilun1
            SCSI SN: 9b544a8e
            Size: 10737 MB, Block size: 512
            Online: Yes
            Removable media: No
            Prevent removal: No
            Readonly: No
            Backing store type: rdwr
            Backing store path: /dev/vg_iscsi/iscsilun1
            Backing store flags:
    Account information:
    ACL information:
        ALL
```

重新开启 drbd1 节点主机电源，可以看集群 drbd1 自动切换成在线状态

```
Last updated: Wed Oct 21 21:36:32 2015
Last change: Wed Oct 21 21:22:12 2015
Stack: classic openais (with plugin)
Current DC: drbd2 - partition with quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
6 Resources configured


Online: [ drbd1 drbd2 ]

 Master/Slave Set: ms_drbd_clustervol [p_drbd_clustervol]
     Masters: [ drbd2 ]
     Slaves: [ drbd1 ]
 Resource Group: rg_clustervol
     p_lvm_vg_iscsi     (ocf::heartbeat:LVM):   Started drbd2
     p_target_clustervol        (ocf::heartbeat:iSCSITarget):   Started drbd2
     p_lu_iscsilun1     (ocf::heartbeat:iSCSILogicalUnit):      Started drbd2
     p_ip_clustervolip  (ocf::heartbeat:IPaddr2):       Started drbd2
```

关闭 drbd2 节点主机，在 drbd1 节点主机上查看集群状态，但会发现一直处于如下显示状态，资源不会自动在 drbd1 上启动

```
Last updated: Wed Oct 21 21:39:07 2015
Last change: Wed Oct 21 21:22:12 2015
Stack: classic openais (with plugin)
Current DC: drbd1 - partition WITHOUT quorum
Version: 1.1.11-97629de
2 Nodes configured, 2 expected votes
6 Resources configured


Online: [ drbd1 ]
OFFLINE: [ drbd2 ]

 Master/Slave Set: ms_drbd_clustervol [p_drbd_clustervol]
     Slaves: [ drbd1 ]
     Stopped: [ drbd2 ]
```

出现此状态是因为集群在正常切换之后会在集群配置信息中添加 drbd fence 信息，预防因脑裂发生造成 drbd 同步磁盘中数据的损坏，要想正常启动集群，我们只需要修改或清除集群配置信息中的 drbd fence 信息即可

查看 crm 配置信息

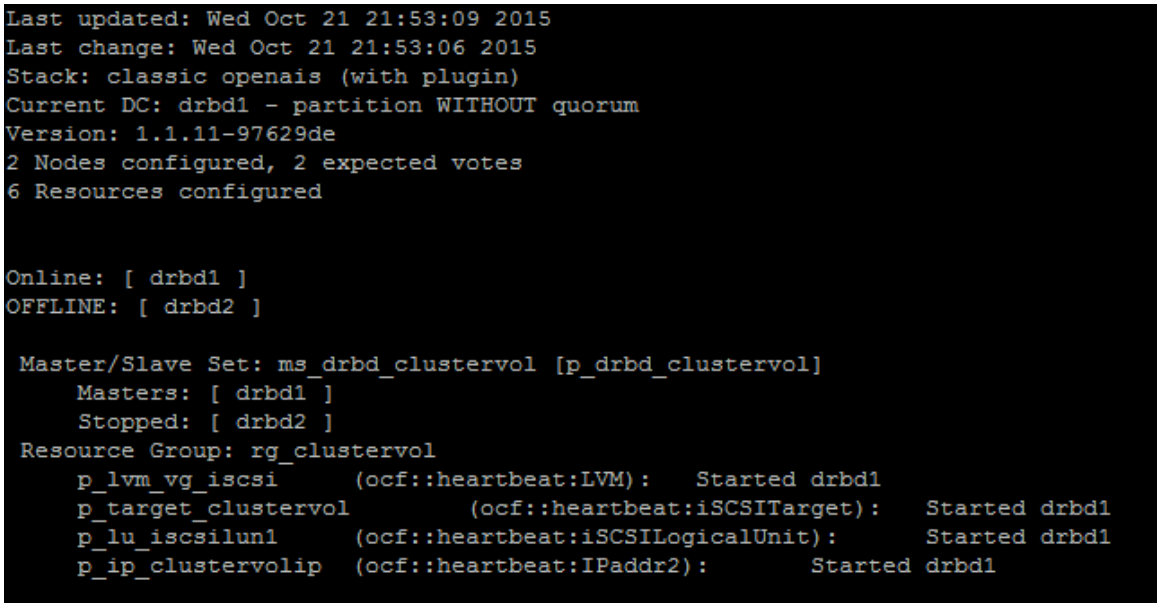
```
[root@drbd1 ~]# crm configure show
node drbd1
node drbd2
primitive p_drbd_clustervol ocf:linbit:drbd \
        params drbd_resource=clustervol \
        op monitor interval=29 role=Master \
        op monitor interval=31 role=Slave
primitive p_ip_clustervolip IPaddr2 \
        params ip=192.168.8.17 cidr_netmask=24 \
        op monitor interval=10s
primitive p_lu_iscsilun1 iSCSILogicalUnit \
        params target_iqn="iqn.1994-05.com.redhat:clustervol.iscsilun1" lun=1 pa
th="/dev/vg_iscsi/iscsilun1" implementation=tgt \
        op monitor interval=10s
primitive p_lvm_vg_iscsi LVM \
        params volgrpname=vg_iscsi \
        op monitor interval=30s timeout=60s depth=0
primitive p_target_clustervol iSCSITarget \
        params iqn="iqn.1994-05.com.redhat:clustervol.iscsilun1" tid=1 \
        op monitor interval=10s timeout=30s
group rg_clustervol p_lvm_vg_iscsi p_target_clustervol p_lu_iscsilun1 p_ip_clust
ervolip
ms ms_drbd_clustervol p_drbd_clustervol \
        meta master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=
true
location drbd-fence-by-handler-clustervol-ms_drbd_clustervol ms_drbd_clustervol
\
        rule $role=Master -inf: #uname ne drbd2
colocation c_clustervol_on_drbd inf: rg_clustervol ms_drbd_clustervol:Master
order o_drbd_before_clustervol inf: ms_drbd_clustervol:promote rg_clustervol:sta
rt
property cib-bootstrap-options: \
        dc-version=1.1.11-97629de \
        cluster-infrastructure="classic openais (with plugin)" \
        expected-quorum-votes=2 \
        stonith-enabled=false \
        no-quorum-policy=ignore \
        default-resource-stickiness=200
```

将上图红框中最后一句#uname ne drbd2 修改成#uname ne drbd1 即可正常启动相关服务

crm configure edit

location drbd-fence-by-handler-clustervol-ms_drbd_clustervol ms_drbd_clustervol \

rule $role=Master -inf: #uname ne drbd1



想要自动切换服务，可以在两个节点正常的时候，先删除上图红框中的信息，然后当主节点宕机或关机后，服务便可以自动切换，不需要手动修改配置文件

crm configure delete drbd-fence-by-handler-clustervol-ms_drbd_clustervol

6. 测试总结

如果发现集群服务都正常启动，但集群节点不能通讯，请检查防火墙是否打开，关闭即可解决

如果把关闭机器时候信息 corosync 服务需要很长的时间，可以手动 kill 掉 corosync 的服务: **ps -ef | grep corosync | awk '$8=="corosync" {print $2}' | xargs kill -9**