基于DRBD的KVM群集构建

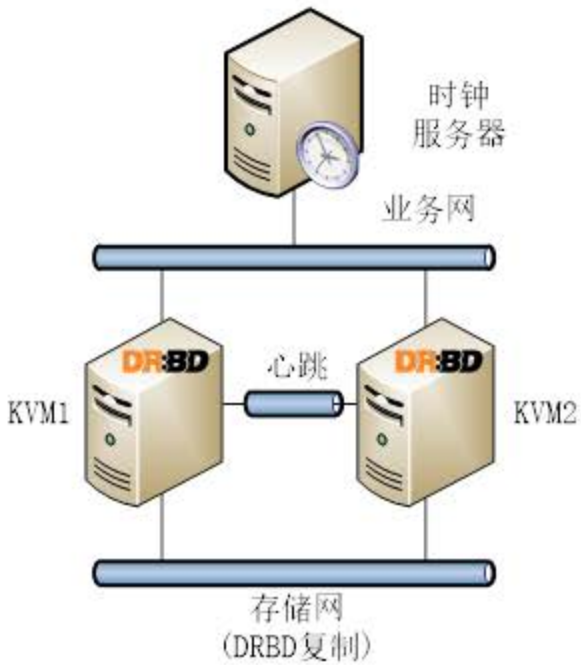
概述

- 规划设计
- 节点准备
  - 阶段1：操作系统安装
  - 阶段2：群集组件安装
  - 阶段3：群集节点准备
- 双主DRBD资源准备
- 配置STONITH（virtd)
- 配置DLM
- 配置CLVM
- 配置GFS2
- 向群集添加虚拟机资源
- 群集测试

群集资源约束：

DLM →
DRBD → CLVM → File System → Virtual Domain

规划设计



主机	LAN	Corosync	Storage(DRBD)
kvm1	192.168.1.231	172.16.1.231	10.0.1.231
kvm2	192.168.1.232	172.16.1.232	10.0.1.232

DLM →
DRBD → CLVM → File System → Virtual Domain

节点准备-阶段1：操作系统安装

- 操作系统安装
- 通过kickstart简化安装
- 操作系统升级

```
install
cdrom
text
keyboard --vckeymap=us --xlayouts='us'
lang en_US.UTF-8
network  --bootproto=dhcp --device=eth0 --noipv6
network  --hostname=localhost.localdomain
auth --enableshadow --passalgo=sha512
rootpw --plaintext 123456
kipx
timezone Asia/Shanghai --isUtc
ignoredisk --only-use=sda
bootloader --append=" crashkernel=auto" --
location=mbr --boot-drive=sda
autopart --type=lvm
clearpart --none --initlabel
reboot
firstboot --disable

%packages
@base
@core
@gnome-desktop
@virtualization-client
@virtualization-hypervisor
@virtualization-platform
@virtualization-tools
```

```
pacemaker
pcs
corosync
fence-agents-all

iscsi-initiator-utils

dlm
lvm2-cluster
gfs2-utils

kexec-tools
policycoreutils-python
psmisc

tigervnc-server

%addon com_redhat_kdump --enable --
reserve-mb='auto'
%end
```

节点准备-阶段2：群集组件安装

- 配置yum库
- 安装 Pacemaker 等群集组件

```
# yum -y install pacemaker corosync pcs \
psmisc policycoreutils-python fence-agents-all
```

节点准备-阶段3：群集节点准备

- 配置主机名及解析
- 配置SSH Key互信(可选）
- 配置时钟
- 配置防火墙
- 配置pcs守护程序
- 配置hacluster账户密码
- 集群配置文件

```
# hostnamectl set-hostname kvm1
# vi /etc/hosts

# ssh-keygen -t rsa -P ''
# ssh-copy-id -i ~/.ssh/id_rsa.pub root@kvm2

# /sbin/ntpdate time.windows.com
# crontab -e

# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
# firewall-cmd --reload

# systemctl start pcsd
# systemctl enable pcsd

# echo "linuxplus" | passwd --stdin hacluster
# pcs cluster auth kvm1 kvm2

# pcs cluster setup --name cluster1 kvm1 kvm2

# pcs cluster start --all
```

双主DRBD资源准备

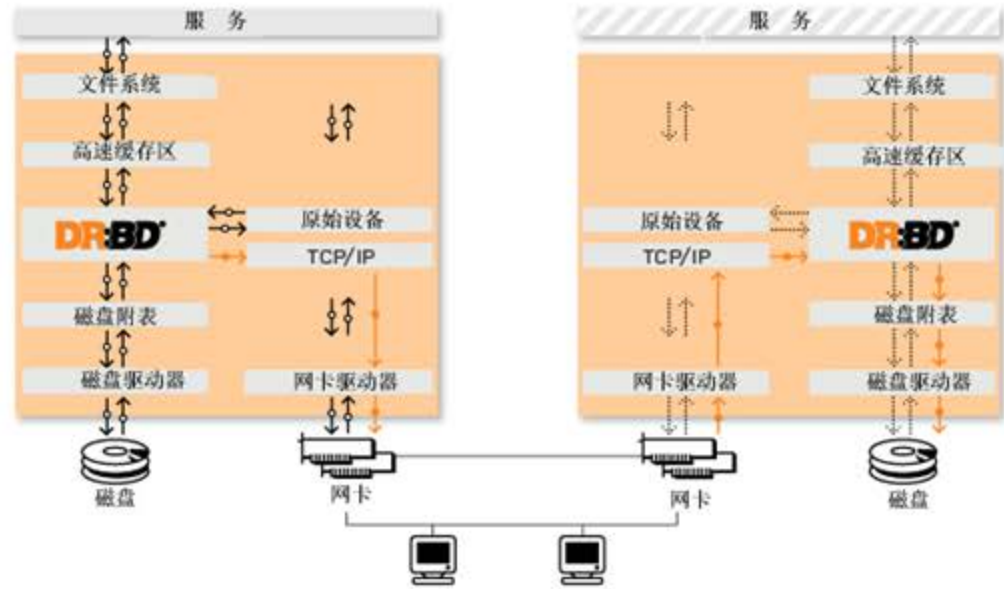
- DRBD概述
- DRBD软件安装
- 为DRBD配置防火墙和SELinux
- 准备DRBD的磁盘
- 配置DRBD参数
- 初始化及同步DRBD磁盘

DRBD概述

- Distributed Replicated Block Device(分布式复制块设备 DRBD)
- 是一种基于软件的，无共享，复制的存储解决方案，在服务器之间的对块设备（硬盘，分区，逻辑卷等）进行镜像
- 可以认为是基于网络的RAID1
- DRBD镜像数据
  - 实时性：当应用对磁盘的数据进行修改时，复制立即发生
  - 透明性：应用程序的数据存储在镜像设备上是独立和透明的，数据可存储在不同的服务器上
  - 同步镜像和异步镜像：
    - 同步镜像，当本地发申请进行写操作进行时，同步写到两台服务器上
    - 异步镜像，当本地写申请已经完成对本地的写操作时，开始对对应的服务器进行写操作

DRBD 体系结构

- 内核模块
  - DRBD技术的核心功能是通过一个Linux内核模块实现的。
- 用户空间管理工具
  - 为了能够管理和配置DRBD的资源，DRBD配备了一些管理工具与内核模块进行通信。
  - drbdadm
  - drbdsetup
  - drbdmeta



DRBD 核心特性

- 资源角色
  - 单主模式 Single-primary mode
  - 双主模式 Dual-primary mode
- 复制模式
  - 协议A：Asynchronous replication protocol
  - 协议B：Memory synchronous (semi-synchronous) replication protocol
  - 协议C：Synchronous replication protocol
    - 就目前而言应用最多和应用最广泛的为协议C

DRBD软件安装

- Linux内核2.6.33以后的版本中，只需要安装管理工具即可
- CentOS不包含这些工具，需要从第三方的可信的软件仓库来获得

```
# vi /etc/yum.conf 修改keepcache=1

# 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
# cat /etc/yum.repos.d/elrepo.repo

# yum -y install -y kmod-drbd84 drbd84-utils

# cd /var/cache/yum/x86_64/7/elrepo/packages/
# ls
drbd84-utils-8.9.5-1.el7.elrepo.x86_64.rpm
kmod-drbd84-8.4.7-1_1.el7.elrepo.x86_64.rpm
# scp *.rpm node2:/tmp

在节点2上进行安装
```

为DRBD配置防火墙和SELinux

- 配置防火墙，将corosync、drbd的专用网段设置为全开放

```
[ALL]# firewall-cmd --permanent --zone=trusted --add-source=172.16.1.0/24
[ALL]# firewall-cmd --permanent --zone=trusted --add-source=10.0.1.0/24
[ALL]# firewall-cmd --reload

# firewall-cmd --zone=trusted --list-sources
10.0.1.0/24 172.16.1.0/24
# firewall-cmd --get-active-zones
public
  interfaces: eth0 eth1 eth2
            或 eno16777728 eno33554960 eno50332184
trusted
  sources: 10.0.1.0/24 172.16.1.0/24
```

- 配置SELinux

```
[ALL]# semanage permissive -a drbd_t
```

准备DRBD复制的LV

- 在每个节点上准备相同大小的LV

```
[ALL]# fdisk -l /dev/sdb
创建一个分区

[ALL]# pvcreate /dev/sdb1
[ALL]# vgcreate drbdvg0 /dev/sdb1
[ALL]# lvcreate --name lvdrbd0 --size 4G drbdvg0

[ALL]# lvscan
  ACTIVE            '/dev/drbdvg0/lvdrbd0' [4.00 GiB] inherit
  ACTIVE            '/dev/centos/swap' [2.00 GiB] inherit
  ACTIVE            '/dev/centos/root' [37.46 GiB] inherit
```

配置DRBD参数文件

- 全局参数文件
  - /etc/drbd.d/global_common.conf
  - 通常保持默认值。
  - 安装计数：usage-count yes;
- 创建配置文件


```
[ALL]# vi /etc/drbd.d/r0.res
resource r0 {
    protocol C;
    meta-disk internal;
    device /dev/drbd0;
    disk /dev/drbdvg0/lvdrbd0;
    syncer {
        verify-alg sha1;
    }
    on node1 {
        address  10.0.1.231:7789;
    }
    on node2 {
        address  10.0.1.232:7789;
    }
    net {
        allow-two-primaries;
        after-sb-0pri discard-zero-changes;
        after-sb-1pri discard-secondary;
        after-sb-2pri disconnect;
    }
    disk {
         fencing resource-and-stonith;
    }
    handlers {
        fence-peer "/usr/lib/drbd/crm-fence-peer.sh";
        after-resync-target "/usr/lib/drbd/crm-unfence-peer.sh";
    }
}
```

DRBD初始化及同步

- 初始化

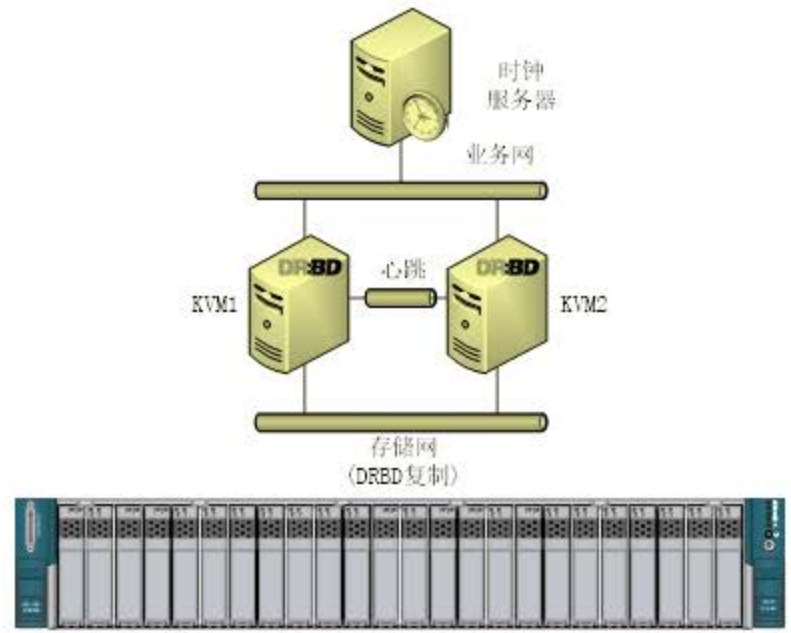
```
[ALL]# drbdadm create-md r0
[ALL]# modprobe drbd
[ALL]# drbdadm up r0
[ALL]# cat /proc/drbd
version: 8.4.7-1 (api:1/proto:86-101)
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by phil@Build64R7, 2016-01-12 14:29:40
 0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r-----
    ns:0 nr:0 dw:0 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:4194140
```

- 同步

```
[root@node1 ~]# drbdadm primary --force r0
[root@node1 ~]# cat /proc/drbd
version: 8.4.7-1 (api:1/proto:86-101)
GIT-hash: 3a6a769340ef93b1ba2792c6461250790795db49 build by phil@Build64R7, 2016-01-12 14:29:40
 0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----
    ns:136496 nr:0 dw:0 dr:137408 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:4057644
        [>....................] sync'ed:  3.4% (4057644/4194140)K
        finish: 0:08:53 speed: 7,580 (7,580) K/sec

[root@node1 ~]# drbd-overview
 0:r0/0  SyncSource Primary/Secondary UpToDate/Inconsistent
       [====>...............] sync'ed: 26.1% (3102392/4194140)K
```

配置STONTH（virtd)

- Host服务器配置
- Guest机配置
- 为群集配置STONITH



Host服务器配置

- 软件包安装

```
# yum install -y fence-virt fence-virtd fence-virtd-libvirt fence-virtd-multicast
```

- 创建认证文件

```
# dd if=/dev/urandom of=/etc/cluster/fence_xvm.key bs=4096 count=1
```

- 生成配置文件

```
# fence_virtd -c
```

- 启动服务

```
# systemctl enable fence_virtd.service; systemctl start fence_virtd.service
```

- 验证配置

```
# fence_xvm -o list
# fence_xvm -o reboot -H vm1
```

Guest 配置

- 软件包安装

```
[ALL]# yum install -y fence-virt
```

- 同步配置文件

```
[ALL]# mkdir /etc/cluster
[ALL]# scp zzkvm1:/etc/cluster/fence_xvm.key /etc/cluster/
```

- 配置防火墙

```
# firewall-cmd --add-port=1229/tcp --permanent; firewall-cmd --reload
```

- 测试

```
# fence_xvm -o list
# fence_xvm -o reboot -H vm1
```

为群集配置STONTH

```
# pcs stonith create kvm-shooter fence_xvm pcmk_host_list="kvm1 kvm2"

# pcs status
......
kvm-shooter    (stonith:fence_xvm):    Started kvm1-cr
......

# pcs stonith show --full
 Resource: kvm-shooter (class=stonith type=fence_xvm)
  Attributes: pcmk_host_list="kvm1 kvm2"
  Operations: monitor interval=60s (kvm-shooter-monitor-interval-60s)

# pcs property --all |grep stonith-action
 stonith-action: reboot

# stonith_admin --reboot kvm2
  节点2将重新启动
```

安装群集文件系统软件

- OCFS2和GFS2是群集文件系统

```
[all]# yum -y install gfs2-utils dlm
……
Installed:
dlm.x86_64 0:4.0.2-6.el7
gfs2-utils.x86_64 0:3.1.8-6.el7

Dependency Installed:
dlm-lib.x86_64 0:4.0.2-6.el7
```

配置DLM

- 方法1

```
# pcs cluster cib dlm_cfg
# pcs -f dlm_cfg resource create dlm ocf:pacemaker:controld op monitor
interval=60s
# pcs -f dlm_cfg resource clone dlm clone-max=2 clone-node-max=1

# pcs cluster cib-push dlm_cfg
```

- 方法2

```
# pcs resource create dlm ocf:pacemaker:controld \
 op monitor interval=30s on-fail=fence \
 clone interleave=true ordered=true
```

在群集中添加DRBD资源

- 首先，要保证两个状态为均为Secondary，状态为UpToDate

```
# cat /proc/drbd
version: 8.4.8-1 (api:1/proto:86-101)
GIT-hash: 22b4c802192646e433d3f7399d578ec7fecc6272 build by mockbuild@, 2016-10-13 19:58:26
 0: cs:Connected ro:Secondary/Secondary ds:UpToDate/UpToDate C r-----
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

- 添加资源

```
# pcs resource create VMData ocf:linbit:drbd
  drbd_resource=r0 op monitor interval=60s

# pcs resource master VMDataClone VMData \
  master-max=2 master-node-max=1 clone-max=2 clone-node-max=1 notify=true

# pcs status
……
 Master/Slave Set: VMDataClone [VMData]
     Masters: [ kvm1-cr kvm2-cr ]两个均是Master
……
```

考察双主Dual-Primary模式

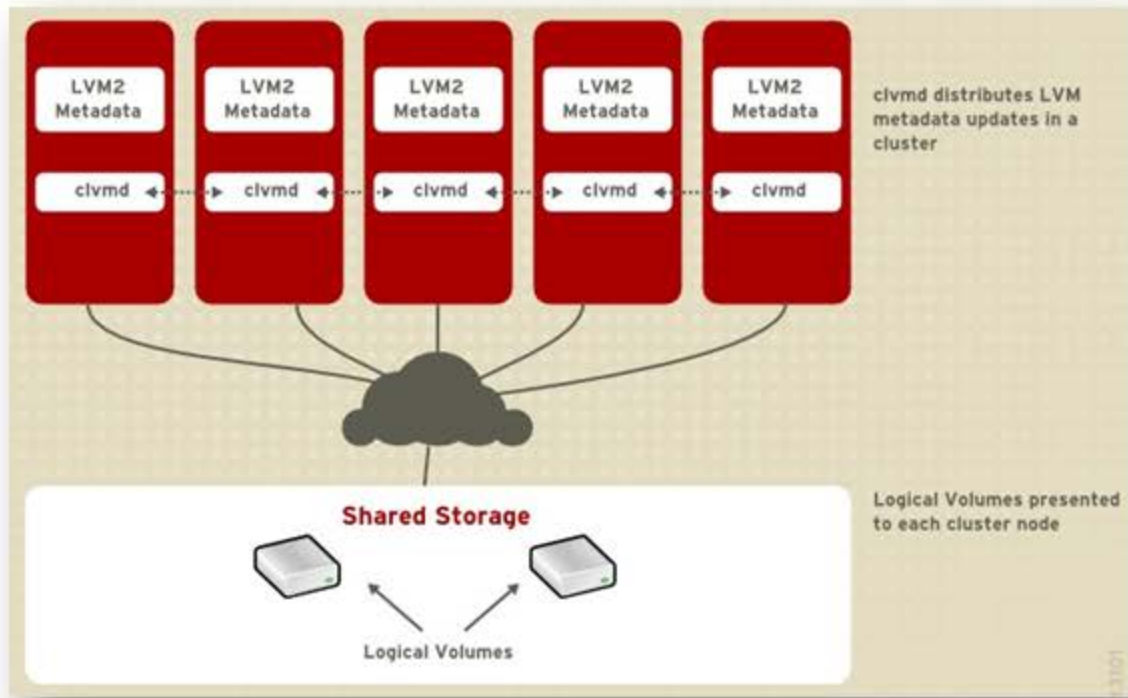
```
# cat /proc/drbd
version: 8.4.8-1 (api:1/proto:86-101)
GIT-hash: 22b4c802192646e433d3f7399d578ec7fecc6272 build by mockbuild@, 2016-10-13 19:58:26
 0: cs:Connected ro:Primary/Primary ds:UpToDate/UpToDate C r-----
    ns:0 nr:0 dw:0 dr:912 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0

# drbd-overview
 0:r0/0  Connected Primary/Primary UpToDate/UpToDate
```

配置CLVM

- 群集化LVM(CLVM)概述
- 安装并启用CLVM
- 向群集中添加CLVM资源
- 创建LV

群集化LVM(CLVM)概述

- CLVM(Clustered LVM)是 LVM 的一个集群方面的扩展。
- 允许一个集群的计算机通过 LVM 管理共享存储。
- clvmd 是 CLVM 的核心，作为pacemaker一个子进程来运行。



安装并启用CLVM

- 安装CLVM软件包

```
[ALL]# yum -y install lvm2-cluster
```

- 配置LVM并重新启动

```
[ALL]# lvmconf --enable-cluster
[ALL]# reboot

# grep locking_type /etc/lvm/lvm.conf
    locking_type = 3
```

- locking_type的值：

```
1 LVM uses local file-based locking, the standard mode.

3 LVM uses built-in clustered locking with clvmd. This is incompatible with
lvmetad. If use_lvmetad is enabled, LVM prints a warning and disables lvmetad
use.
```

向群集中添加CLVM资源

- 添加克隆的资源，即在每个节点上均运行clvmd

```
# pcs resource create clvmd ocf:heartbeat:clvm op monitor interval=30s \
  on-fail=fence clone interleave=true ordered=true

# pcs status
......
Full list of resources:

 ipmi-fencing   (stonith:fence_ipmilan):        Started kvm1-cr
 Clone Set: dlm-clone [dlm]
     Started: [ kvm1-cr kvm2-cr ]
 Clone Set: clvmd-clone [clvmd]
     Started: [ kvm1-cr kvm2-cr ]
......
```

配置约束

DLM →
DRBD → CLVM → File System → Virtual Domain

```
# pcs constraint order start dlm-clone then clvmd-clone
# pcs constraint colocation add clvmd-clone with dlm-clone

# pcs constraint order promote VMDataClone then start clvmd-clone
# pcs constraint colocation add clvmd-clone with VMDataClone
```

创建LV

- 修改lvm.conf的过滤虚属性，避免LVM会看重复的数据

```
[ALL]# vi /etc/lvm/lvm.conf
  将filter修改为：filter = [ "a|/dev/vd.*|","a|/dev/drbd*|", "r/.*/" ]

[ALL]# vgscan -v
```

- 创建LV

```
# pvcreate /dev/drbd0
# vgcreate vgvm0 /dev/drbd0
# lvcreate -n lvvm0 -l 100%FREE vgvm0

# lvscan
  ACTIVE            '/dev/vgdrbd0/lvdrbd0' [5.00 GiB] inherit
  ACTIVE            '/dev/centos/swap' [3.88 GiB] inherit
  ACTIVE            '/dev/centos/home' [25.57 GiB] inherit
  ACTIVE            '/dev/centos/root' [50.00 GiB] inherit
  ACTIVE            '/dev/vgvm0/lvvm0' [5.00 GiB] inherit
# vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  centos    1   3   0 wz--n- 79.51g 64.00m
  vgdrbd0   1   1   0 wz--n- 80.00g 75.00g
  vgvm0     1   1   0 wz--nc  5.00g     0
```

概述

- ~~规划设计~~
- ~~节点准备~~
  - ~~阶段1：操作系统安装~~
  - ~~阶段2：群集组件安装~~
  - ~~阶段3：群集节点准备~~
- ~~双主DRBD资源准备~~
- ~~配置STONITH (virtd)~~
- ~~配置DLM~~
- ~~配置CLVM~~
- 配置GFS2
- 向群集添加虚拟机资源
- 群集测试

群集资源约束：

DLM →
DRBD → CLVM → File System → Virtual Domain

配置GFS2

- 创建GFS2文件系统
- 向群集添加GFS2文件系统
- 配置SELinux

创建GFS2文件系统

```
# lvscan
  ACTIVE            '/dev/vgdrbd0/lvdrbd0' [5.00 GiB] inherit
  ACTIVE            '/dev/centos/swap' [3.88 GiB] inherit
  ACTIVE            '/dev/centos/home' [25.57 GiB] inherit
  ACTIVE            '/dev/centos/root' [50.00 GiB] inherit
  ACTIVE            '/dev/vgvm0/lvvm0' [5.00 GiB] inherit

# mkfs.gfs2 -p lock_dlm -j 2 -t cluster1:kvm1 /dev/vmvg0/lvvm0
/dev/vgvm0/lvvm0 is a symbolic link to /dev/dm-4
This will destroy any data on /dev/dm-4
Are you sure you want to proceed? [y/n]y

Device:                    /dev/vgvm0/lvvm0
Block size:                4096
Device size:               5.00 GB (1309696 blocks)
Filesystem size:           5.00 GB (1309695 blocks)
Journals:                  2
Resource groups:           21
Locking protocol:          "lock_dlm"
Lock table:                "cluster1:kvm1"
UUID:                      e0f7a40c-8c28-fa62-ee8c-d334e3ebe5a2
```

向群集中添加GFS2文件系统

- 添加克隆的资源，即在每个节点上均挂载文件系统

```
# pcs resource create VMFS Filesystem \
  device="/dev/vmvg0/lvvm0" directory="/vm" fstype="gfs2" clone

# pcs status
……
Clone Set: VMFS-clone [VMFS]
     Started: [ kvm1-cr kvm2-cr ]
……
```

- 配置约束：GFS2必须在clvmd 启动后启动，而且必须在同一个节点上

```
# pcs -f fs_cfg constraint order clvmd-clone then VMFS-clone

# pcs -f fs_cfg constraint colocation add VMFS-clone with clvmd-clone
```

配置SELinux

- 配置SELinux设定，不然虚拟机无法访问存储文件。

```
[ALL]# semanage fcontext -a -t virt_image_t "/vm(/.*)?"
[ALL]# restorecon -R -v /vm
```

- 如果没有semanage，那么安装policycoreutils-python

```
[ALL]# yum install policycoreutils-python
```

向群集添加虚拟机资源

- 准备测试用的虚拟机
- 测试机的动态迁移
- 创建虚拟机资源

准备测试用的虚拟机

- Window 2003 Server

```
virt-install --name=win2k3a \
 --disk device=disk,bus=virtio,path='/vm/win2k3a.qcow2' \
 --vcpus=1 --ram=512 \
 --network network=default,model=virtio \
 --graphics vnc \
 --boot hd
```

- CentOS 7.2

```
# virt-install --name=centos7a  \
 --disk device=disk,bus=virtio,path='/vm/centos7-1511-disk0.qcow2'  \
 --vcpus=1 --ram=512  \
 --network network=default,model=virtio  \
 --graphics vnc  --boot hd
```

测试机的动态迁移

- 配置源及目标宿主机的防火墙

```
[ALL]# firewall-cmd --add-port=16509/tcp --permanent
[ALL]# firewall-cmd --add-port=49152-49215/tcp --permanent
[ALL]# firewall-cmd -reload
```

- 使用virt-manager及virsh均可

```
# virsh migrate --domain centos7a \
  qemu+ssh://kvm1-cr/system --live
```

创建虚拟机资源

- 所有节点可以访问虚拟机配置文件和磁盘镜像文件
- 虚拟机由群集软件控制而不是由libvirt来控制

```
# virsh shutdown centos7a
# mkdir /vm/qemu_config
# virsh dumpxml centos7a > /vm/qemu_config/centos7a.xml
# pcs resource create centos7a_res VirtualDomain \
 hypervisor="qemu:///system" \
 config="/vm/qemu_config/centos7a.xml" \
 migration_transport=ssh \
 meta allow-migrate="true"
```

- 配置约束

```
# pcs constraint order start VMFS-clone then centos7a_res
```

迁移测试

- 移动资源

```
# pcs resource move win2k3a_res
# pcs resource move win2k3a_res kvm1-cr
```

资源属性：meta allow-migrate="true"决定了迁移模式

- 节点待机

```
# pcs cluster standby/unstandby kvm2-cr
```

- 节点停机

```
# pcs cluster stop
Stopping Cluster (pacemaker)...
Stopping Cluster (corosync)...
```

总结

- 规划设计
- 节点准备
  - 阶段1：操作系统安装
  - 阶段2：群集组件安装
  - 阶段3：群集节点准备
- 双主DRBD资源准备
- 配置STONITH（virtd)
- 配置DLM
- 配置CLVM
- 配置GFS2
- 向群集添加虚拟机资源
- 群集测试