基于iSCSI的KVM群集构建

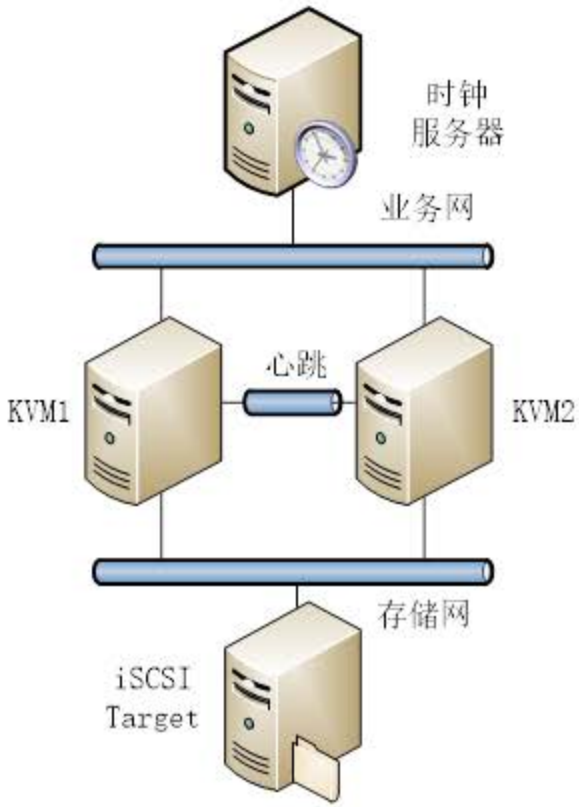
概述

- 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
- 配置iSCSI Target
- 配置STONITH（Disk)
- 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 → CLVM → File System → Virtual Domain

规划设计



主机	LAN	Corosync	Storage
labkvm1	192.168.1.231	172.16.1.231	10.0.1.231
labkvm2	192.168.1.232	172.16.1.232	10.0.1.232
stor1	192.168.1.235		10.0.1.235

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 labkvm1
# vi /etc/hosts

# ssh-keygen -t rsa -P ''
# ssh-copy-id -i ~/.ssh/id_rsa.pub root@labkvm2

# /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 labkvm1 labkvm2

# pcs cluster setup --name cluster1 labkvm1-cr labkvm2-cr

# pcs cluster start --all
```

配置iSCSI Target

- 使用Linux-IO构建iSCSI Target
- iSCSI Target提供
  - 存储资源
  - STONITH资源

使用Linux-IO构建iSCSI Target

- Linux上主要的开源Target项目
- Linux-IO Target 概述

- 实验：创建Linux-IO的iSCSI Target
  - 软件安装
  - 配置防火墙
  - 为Target准备后端存储
  - 配置Target及ACL
  - 配置主机对Target的访问

配置iSCSI Target

- targetcli安装
- 配置防火墙
- 为Target准备后端存储
- 配置Target及ACL
- 配置主机对Target的访问

```
/> ls
o- / ......................................................................... [...]
  o- backstores .............................................................. [...]
  | o- block .................................... [Storage Objects: 0]
  | o- fileio ................................... [Storage Objects: 2]
  | | o- disk01 ..... [/labstor1/disk01.img (1.0GiB) write-back activated]
  | | o- disk02 .... [/labstor1/disk02.img (20.0GiB) write-back activated]
  | o- pscsi .................................... [Storage Objects: 0]
  | o- ramdisk .................................. [Storage Objects: 0]
  o- iscsi ............................................................ [Targets: 1]
  | o- iqn.2016-10.linuxplus.srv:storage.target00.............. [TPGs: 1]
  |   o- tpg1 ................................... [no-gen-acls, no-auth]
  |     o- acls ........................................................ [ACLs: 2]
  |     | o- iqn.1994-05.com.redhat:labkvm1 ............. [Mapped LUNs: 2]
  |     | | o- mapped_lun0 .....................[lun0 fileio/disk01 (rw)]
  |     | | o- mapped_lun1 .....................[lun1 fileio/disk02 (rw)]
  |     | o- iqn.1994-05.com.redhat:labkvm2 ............. [Mapped LUNs: 2]
  |     |   o- mapped_lun0 .....................[lun0 fileio/disk01 (rw)]
  |     |   o- mapped_lun1 .....................[lun1 fileio/disk02 (rw)]
  |     o- luns ........................................................ [LUNs: 2]
  |     | o- lun0 ..................[fileio/disk01 (/labstor1/disk01.img)]
  |     | o- lun1 ..................[fileio/disk02 (/labstor1/disk02.img)]
  |     o- portals .................................................. [Portals: 1]
  |       o- 0.0.0.0:3260 ................................................... [OK]
  o- loopback ......................................................... [Targets: 0]
```

配置STONTH（Disk)

- 使用存储上的1GB的磁盘/dev/sda
- 使用磁盘的ID

```
# pcs stonith describe fence_scsi
# cat /proc/partitions

# ll /dev/disk/by-id/ | grep sda
lrwxrwxrwx. 1 root root  9 Oct 19 15:06 scsi-360014058f98fd66a2f64f93b4ecd812b -> ../../sda
lrwxrwxrwx. 1 root root  9 Oct 19 15:06 wwn-0x60014058f98fd66a2f64f93b4ecd812b -> ../../sda

# pcs stonith create scsi-shooter fence_scsi \
pcmk_host_list="labkvm1-cr labkvm2-cr"  \
devices="/dev/disk/by-id/wwn-0x60014058f98fd66a2f64f93b4ecd812b" \
meta provides=unfencing

# pcs stonith
# pcs stonith show scsi-shooter
```

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

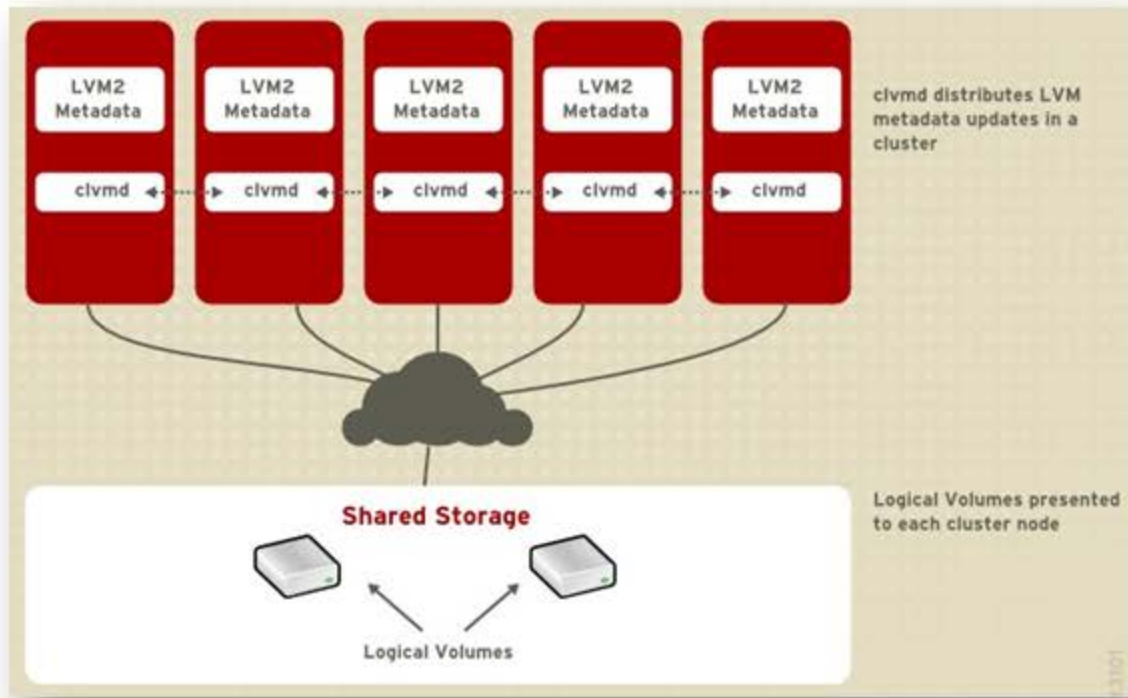
```
# pcs resource create dlm ocf:pacemaker:controld \
 op monitor interval=30s on-fail=fence \
 clone interleave=true ordered=true
```

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 labkvm1-cr
 Clone Set: dlm-clone [dlm]
     Started: [ labkvm1-cr labkvm2-cr ]
 Clone Set: clvmd-clone [clvmd]
     Started: [ labkvm1-cr labkvm2-cr ]
......
```

- 配置约束：clmvd必须在dlm启动后启动，而且必须在同一个节点上

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

创建LV

- 在群集中创建LV与在单节点上创建LV是一样的，命令没有区别
- 必须要保证群集基础架构运行正常，并有quorate法定人数

```
# fdisk /dev/sdb
创建一个分区，设置类型为8e即LVM
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            8192   167772159    83881984   8e  Linux LVM

# partprobe ; multipath -r

# pvcreate /dev/sdb1

# vgcreate vmvg0 /dev/sdb1
  Clustered volume group "vmvg0" successfully created
# vgs
  VG     #PV #LV #SN Attr   VSize  VFree
  centos   1   2   0 wz--n- 19.51g 40.00m
  vmvg0    1   0   0 wz--nc 79.99g 79.99g

# lvcreate -n lvvm0 -l 100%FREE vmvg0
  Logical volume "lvvm0" created.
```

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/vmvg0/lvvm0' [40.00 GiB] inherit
  ACTIVE            '/dev/centos/swap' [2.00 GiB] inherit
  ACTIVE            '/dev/centos/root' [17.47 GiB] inherit

# mkfs.gfs2 -p lock_dlm -j 2 -t cluster1:labkvm1 /dev/vmvg0/lvvm0
/dev/vmvg0/lvvm0 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

Device:                    /dev/vmvg0/lvvm0
Block size:                4096
Device size:               40.00 GB (10485760 blocks)
Filesystem size:           40.00 GB (10485758 blocks)
Journals:                  2
Resource groups:           161
Locking protocol:          "lock_dlm"
Lock table:                "cluster1:labkvm1"
UUID:                      20e100e0-22b6-735a-389f-dbd205c8f947
```

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 -f fs_cfg resource
 Clone Set: dlm-clone [dlm]
     Started: [ labkvm1-cr labkvm2-cr ]
 Clone Set: clvmd-clone [clvmd]
     Started: [ labkvm1-cr labkvm2-cr ]
 Clone Set: VMFS-clone [VMFS]
     Stopped: [ labkvm1-cr labkvm2-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://labkvm1-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 labkvm1-cr
资源属性：meta allow-migrate="true"决定了迁移模式
```

- 节点待机

```
# pcs cluster standby/unstandby labkvm2-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
- 配置iSCSI Target
- 配置STONTH（Disk)
- 配置DLM
- 配置CLVM
- 配置GFS2
- 向群集添加虚拟机资源
- 群集测试