## 高可用集群理论

### 1、高可用集群基本概念

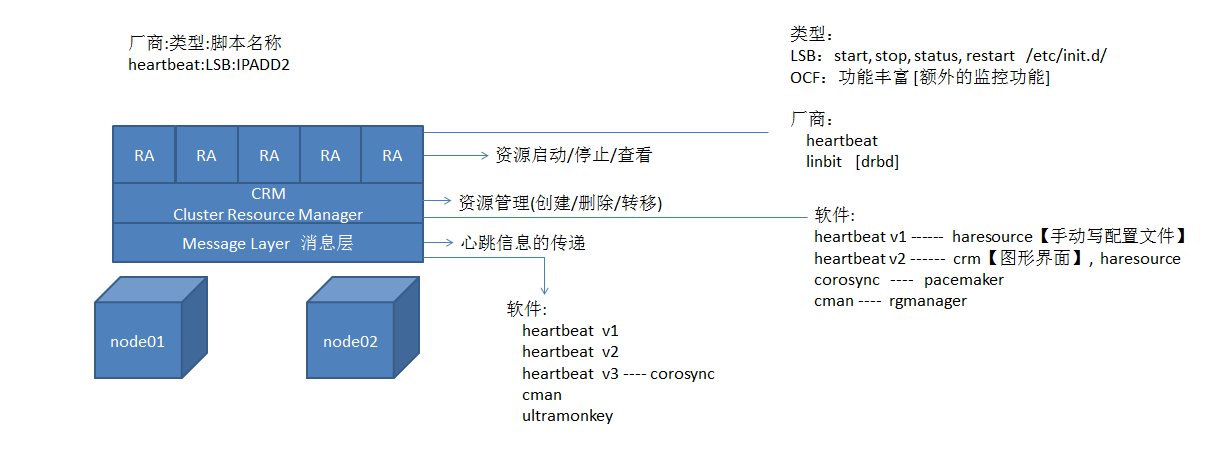
1) 服务 service

2) 资源 resource

组成服务的元素

3) 故障转移 fail over

### 2、高可用集群工作架构



### 3、资源类型

1) 主资源 primitive

同一时刻只能运行在一个节点上的资源

2) 主从资源 Master/Slave资源

同一时刻可以运行在两个节点上[drbd]

3) 克隆资源 Clone

同一时刻可以运行在多个节点上的资源 (stonith)

4) 资源组

保证多个资源同进同退

### 4、资源约束constraint关系

1) 顺序约束 order constraint

定义资源的启动顺序 【分数score】

2) 排列约束 collation constraint

保证多个资源可在同一个节点上运行或者多个资源可同时转移

INFINITY 无穷大

3) 位置约束 location constraint

定义资源对节点的倾向性

INFINITY 无穷大

-INFINITY

### 5、Fence设备

在集群脑裂(split-brain)时，避免集群节点争抢文件系统资源，造成文件系统损坏

隔离

节点隔离

资源隔离

Fence设备实现：

电源交换机

UPS

IBM Blade 刀片

HP iLO

DELL DRAC

测试设备(ssh) ssh 1.1.1.1 "poweroff"

meatware

### 6、法定票数策略 quorum policy

不满足法定票数的策略：

freeze 默认

处理完当前现有的请求，不再接受新的请求，停止服务运行

stop

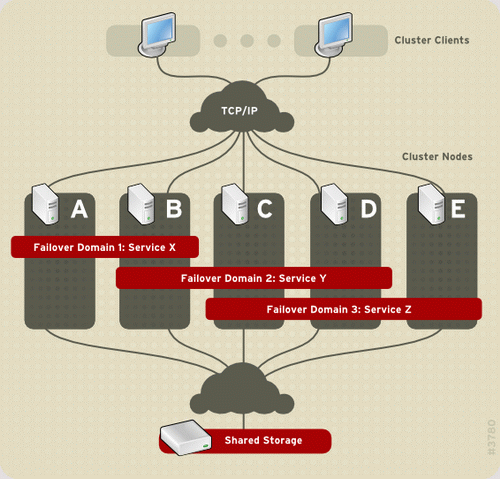
停止服务运行

ignore

忽略法定票数

## RHCS RedHat Cluster Suitable 红帽集群套件

Failover Domain 故障转移域



1、高可用

cman

rgmanger

2. GFS文件系统 Global File System 全局文件系统

dlm 机制 distributed lock manager 分布式锁管理器

借助于HA集群的message layers传递锁信息

配置工具

1、命令行 cman

2、Conga web图形界面

ricci 与web界面通信的代理进程

luci 提供web界面

RHCS集群创建大体步骤：

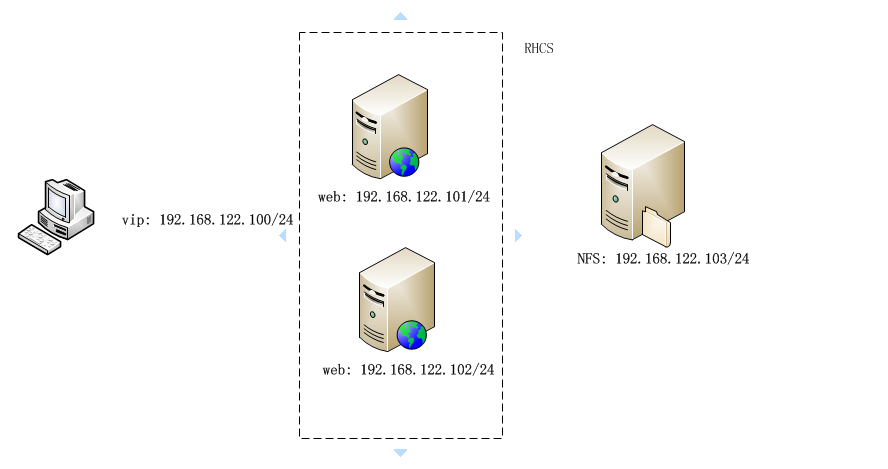
1、创建集群

2、创建故障转移域

3、创建资源

4、启动资源

## 示例01：RHCS + web + NFS



1、防火墙、SELinux, IP, 主机名, 时间同步, 添加所有主机名解析

[root@localhost ~]# hostname node01.linux.com

[root@localhost ~]# cat /etc/sysconfig/network

NETWORKING=yes

HOSTNAME=node01.linux.com

[root@node01 ~]# vim /etc/sysconfig/network-scripts/ifcfg-eth0

[root@node01 ~]# service network restart

[root@node01 ~]# service iptables stop

[root@node01 ~]# chkconfig iptables off

2、配置NFS存储

[root@storage ~]# yum install -y nfs-utils rpcbind

[root@storage ~]# cat /etc/exports

/webdata 192.168.122.101(ro) 192.168.122.102(ro)

[root@storage ~]# service rpcbind start

Starting rpcbind: [ OK ]

[root@storage ~]# service nfs start

Starting NFS services: [ OK ]

Starting NFS mountd: [ OK ]

Starting NFS daemon: [ OK ]

Starting RPC idmapd: [ OK ]

[root@storage ~]# chkconfig rpcbind on

[root@storage ~]# chkconfig nfs on

3、安装httpd，测试nfs的挂载

[root@node01 ~]# yum install -y httpd nfs-utils

测试nfs挂载

[root@node01 ~]# mount -t nfs 192.168.122.103:/webdata /var/www/html/

[root@node01 ~]# ls /var/www/html/

index.html

测试成功后卸载

[root@node01 ~]# umount /var/www/html/

确认httpd不是开机自启动状态

[root@node01 ~]# chkconfig --list httpd

httpd 0:关闭 1:关闭 2:关闭 3:关闭 4:关闭 5:关闭 6:关闭

4、配置RHCS集群

1) 在集群所有节点上安装ricci及RHCS集群的软件

[root@node01 ~]# yum install -y ricci

[root@node01 ~]# passwd ricci

[root@node01 ~]# service ricci start

[root@node01 ~]# chkconfig ricci on

[root@node01 ~]# netstat -antp | grep ricci

tcp 0 0 :::11111 :::\* LISTEN 1845/ricci

[root@node02 ~]# yum -y install cman rgmanager lvm2-cluster sg3\_utils gfs2-utils

2) 在nfs存储端安装luci

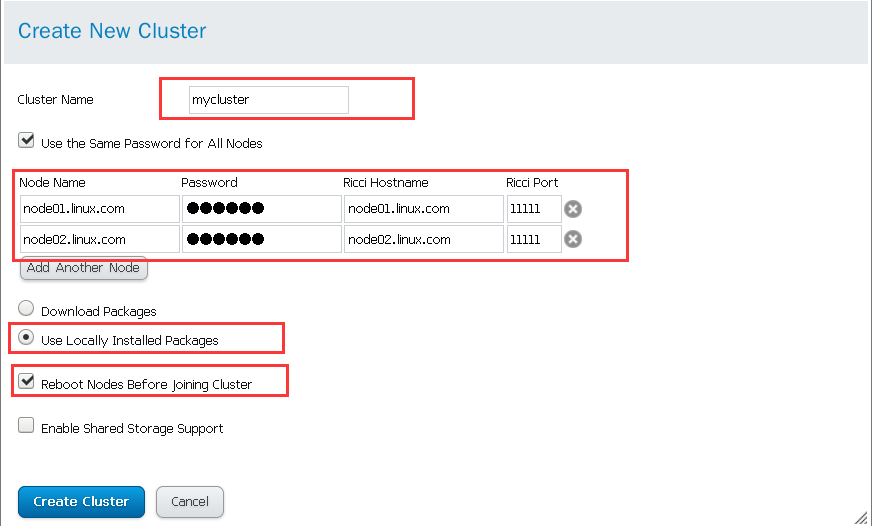
[root@storage ~]# service luci start

[root@storage ~]# chkconfig luci on

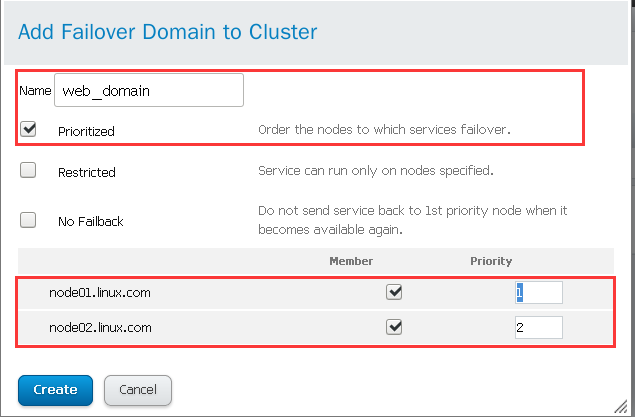
Point your web browser to https://storage.linux.com:8084 (or equivalent) to access luci

访问luci界面

1、创建集群

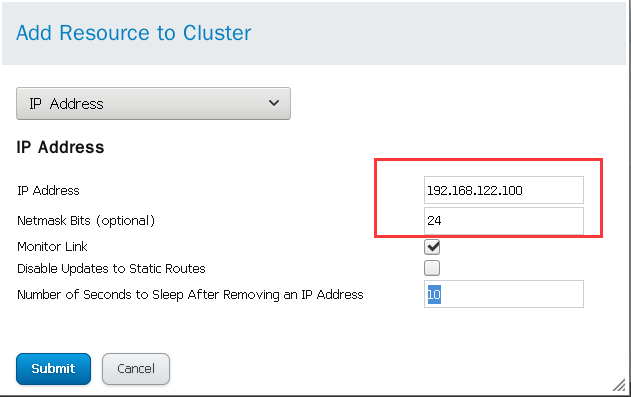


2、创建故障转移域

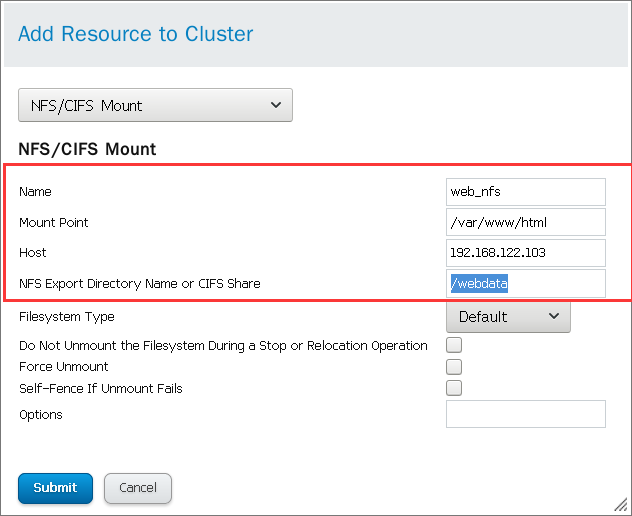


3、创建资源

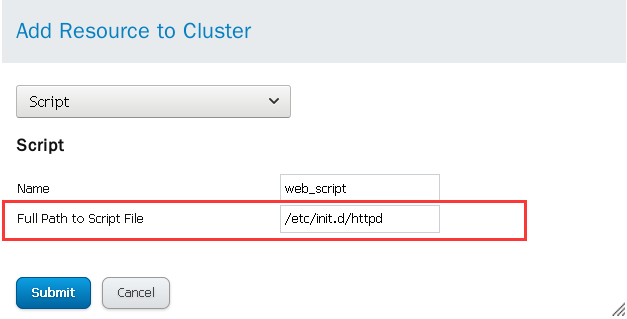
1) 创建IP地址资源



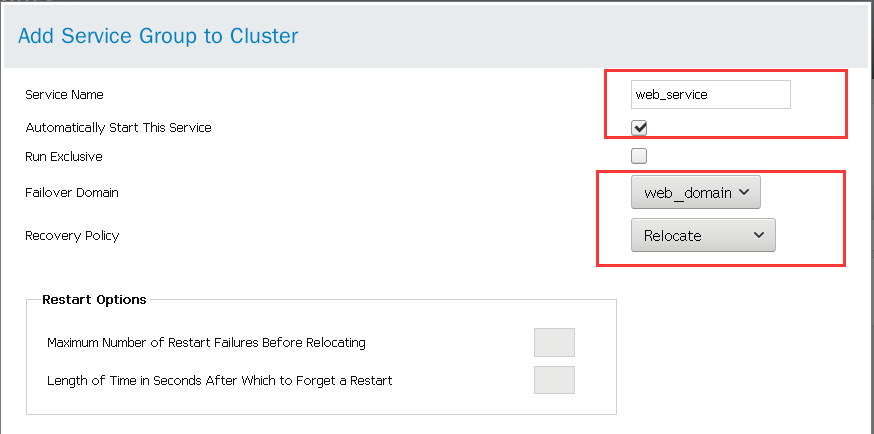
2) 创建文件系统资源

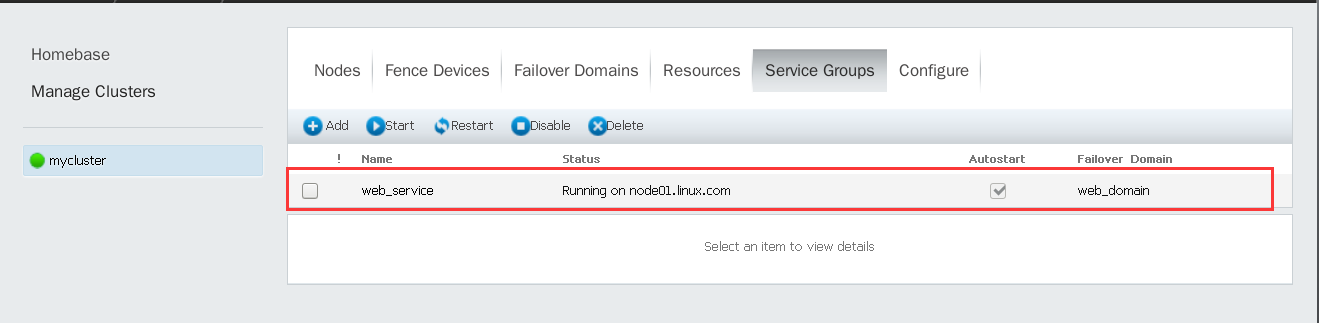


3) 创建启动httpd服务的脚本资源



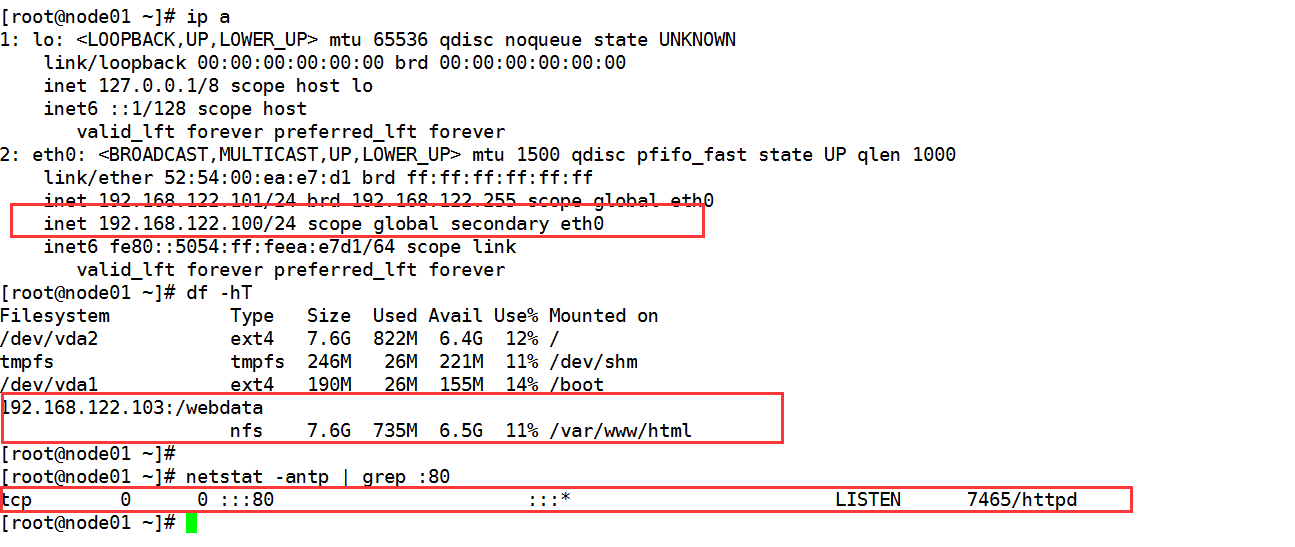
4、创建web服务





验证：

1、查看资源



2、通过命令行工具查看集群状态

[root@node01 ~]# clustat

Cluster Status for mycluster @ Thu Sep 27 14:34:30 2018

Member Status: Quorate

Member Name ID Status

------ ---- ---- ------

node01.linux.com 1 Online, Local, rgmanager

node02.linux.com 2 Online, rgmanager

Service Name Owner (Last) State

------- ---- ----- ------ -----

service:web\_service node01.linux.com started

3、手动转移资源

[root@node01 ~]# clusvcadm -r web\_service -m node02.linux.com

4、模拟故障转移

[root@node01 ~]# service httpd stop

[root@node01 cluster]# ls /etc/cluster/cluster.conf

/etc/cluster/cluster.conf

[root@node01 ~]# service cman status

cluster is running.

[root@node01 ~]# service rgmanager status

rgmanager (pid 1577) is running...

存放日志的目录 ：/var/log/cluster

## 添加fence设备

1、物理机上创建kvm实现fence的key

[root@localhost ~]# yum -y install fence-virt fence-virtd fence-virtd-libvirt fence-virtd-multicast

[root@localhost ~]# mkdir -p /etc/cluster

[root@localhost ~]# dd if=/dev/urandom of=/etc/cluster/fence\_xvm.key bs=4k count=1

2、将物理机的fence\_xvm.key文件分别拷贝到集群节点的/etc/cluster目录

[root@localhost ~]# scp /etc/cluster/fence\_xvm.key 192.168.122.101:/etc/cluster/

[root@localhost ~]# scp /etc/cluster/fence\_xvm.key 192.168.122.102:/etc/cluster/

3、在物理机上创建fence\_virt配置文件

[root@localhost ~]# fence\_virtd -c //创建fence-virt配置文件，回车或输入y

/etc/fence\_virt.conf

4、物理机上启动fence\_virtd服务

[root@localhost ~]# systemctl start fence\_virtd

[root@localhost ~]# systemctl enable fence\_virtd

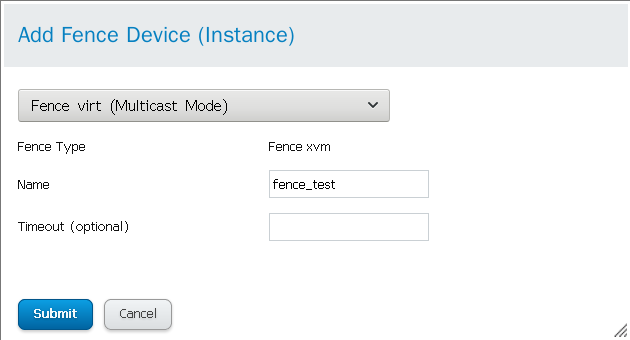
[root@localhost ~]# ps aux |grep fence //查看fence进程是否启动

root 15911 0.0 0.0 140272 3528 ? Ss 12:08 0:00 /usr/sbin/fence\_virtd -w

[root@localhost ~]# netstat -tunlp |grep :1229

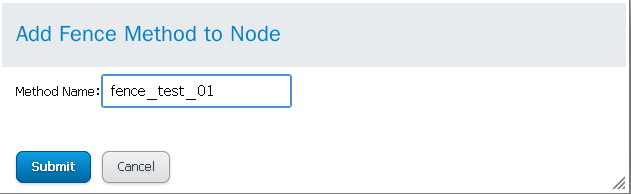
udp 0 0 0.0.0.0:1229 0.0.0.0:\* 14375/fence\_virtd

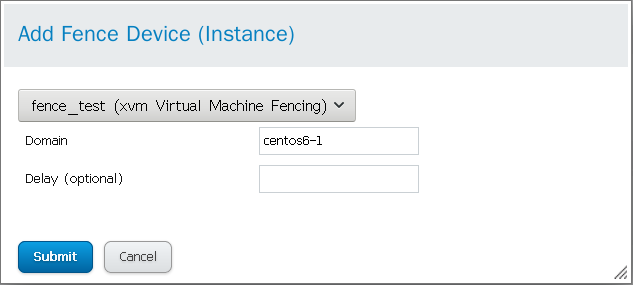
5、在集群中添加fence设备



6、为集群节点绑定fence设备

1) 添加fence方法





7、测试fence设备

[root@node01 ~]# fence\_xvm -H centos6-2

8、模拟心跳故障

[root@node01 ~]# ifconfig eth0 down

## 示例02：RHCS + MySQL + clvmd + iSCSI[集群逻辑卷 clvmd服务]

一、配置iSCSI存储共享块设备

[root@storage ~]# yum install -y scsi-target-utils

[root@storage ~]# vim /etc/tgt/targets.conf

38 <target iqn.2018-09.com.linux:jf1-jg1-emc-disk >

39 backing-store /dev/vdb

40 initiator-address 192.168.122.101

41 initiator-address 192.168.122.102

42 incominguser admin redhat

43 </target>

[root@storage ~]# service tgtd start

Starting SCSI target daemon: [ OK ]

[root@storage ~]# chkconfig tgtd on

[root@storage ~]# netstat -antp | grep tgtd

tcp 0 0 0.0.0.0:3260 0.0.0.0:\* LISTEN 2163/tgtd

tcp 0 0 :::3260 :::\* LISTEN 2163/tgtd

二、在集群节点上连接共享存储，并创建集群逻辑卷/dev/vg01/lv01

[root@node01 ~]# yum install -y iscsi-initiator-utils

添加认证的用户名、密码

[root@node01 ~]# vim /etc/iscsi/iscsid.conf

53 node.session.auth.authmethod = CHAP

57 node.session.auth.username = admin

58 node.session.auth.password = redhat

[root@node01 ~]# iscsiadm -m discovery -t st -p 192.168.122.103

Starting iscsid: [ OK ]

192.168.122.103:3260,1 iqn.2018-02.com.linux:disk01

[root@node01 ~]# iscsiadm -m node -T iqn.2018-02.com.linux:disk01 -p 192.168.122.103 -l

[root@node01 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 252:0 0 8G 0 disk

├─vda1 252:1 0 200M 0 part /boot

└─vda2 252:2 0 7.8G 0 part /

sda 8:0 0 15G 0 disk

创建逻辑卷

[root@node01 ~]# pvcreate /dev/sda

[root@node01 ~]# vgcreate vg01 /dev/sda

[root@node01 ~]# lvcreate -L 7G -n lv01 vg01

[root@node01 ~]# lvscan

ACTIVE '/dev/vg01/lv01' [7.00 GiB] inherit

为了确保集群中所有节点可正常识别逻辑卷，需要重启clvmd服务

[root@node01 ~]# service clvmd restart

确保所有节点看到的逻辑卷状态为ACTIVE时, 可在任意节点上格式化

[root@node01 ~]# mkfs.ext4 /dev/vg01/lv01

三、在node01上安装mysql, 将逻辑卷挂载到mysql的数据目录， 初始化数据库服务

[root@node01 ~]# yum install -y mysql-server

[root@node01 ~]# mount /dev/vg01/lv01 /var/lib/mysql/

[root@node01 ~]# rm -rf /var/lib/mysql/\*

[root@node01 ~]# chown -R mysql.mysql /var/lib/mysql/

[root@node01 ~]# service mysqld start

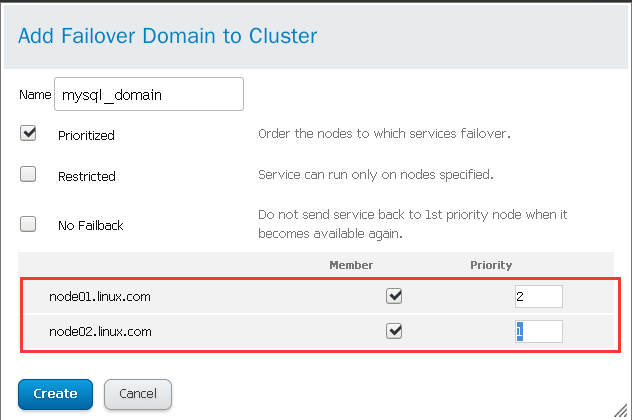
[root@node01 ~]# service mysqld stop

[root@node01 ~]# chkconfig mysqld off

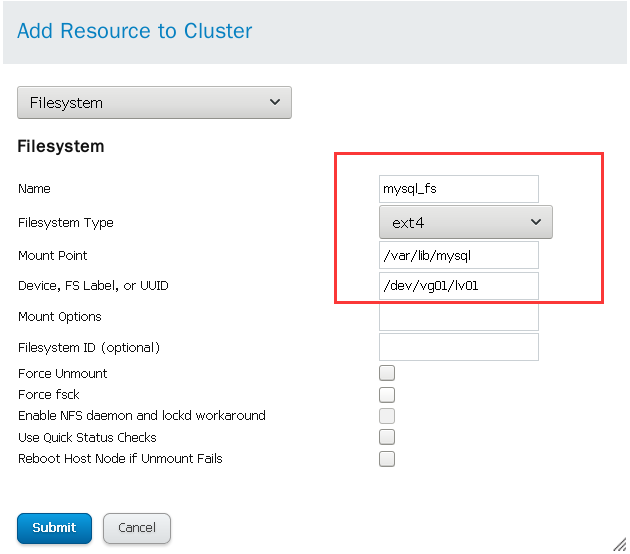
[root@node01 ~]# umount /dev/vg01/lv01

四、在集群中创建mysql服务

1) 创建故障转移域

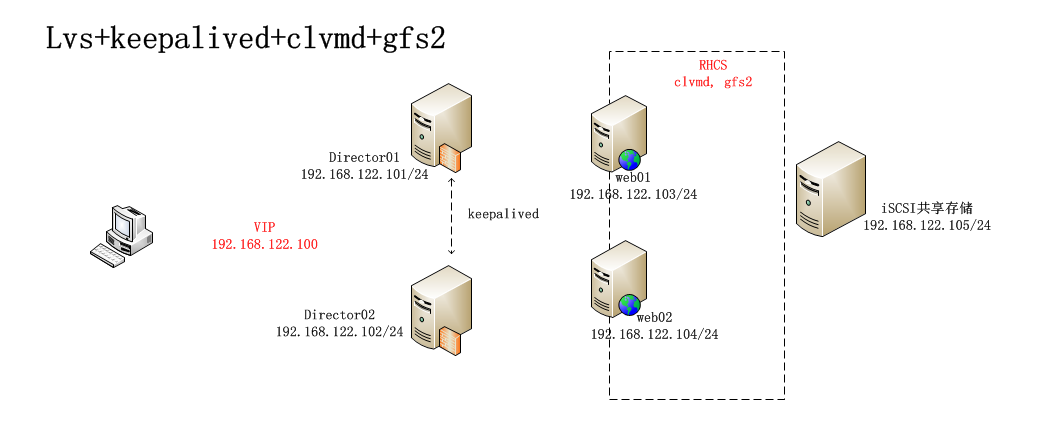


2) 创建mysql资源



启动mysql服务

## 示例03：lvs + keepalived + clvmd + gfs【全局文件系统】

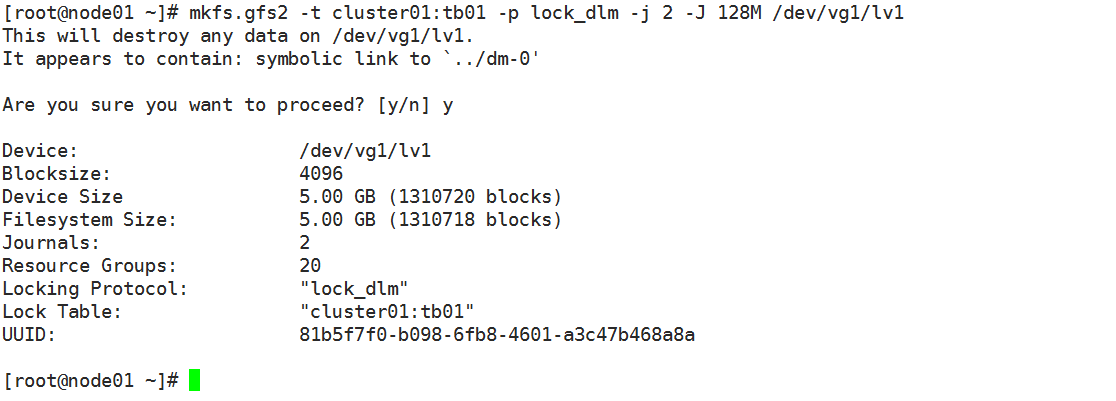


GFS：Global File System 全局文件系统

通过dlm distributed lock manager 分布式锁管理机制, 借助HA集群的Messaging Layer

多个节点产生并发写操作

支持节点数量不能超过16个节点



-t clustername:locktablename

-p lock\_dlm

-j 指定日志空间个数，此个数一定要大于等于使用gfs2文件系统存储的主机数

-J 指定一个日志空间的大小

扩展全局文件系统：

[root@web01 ~]# gfs2\_jadd -j 2 /dev/vg01/lv01

Filesystem: /var/www/html

Old Journals 2

New Journals 4

一、准备工作

1、配置IP、主机名

2、在所有机器中添加所有主机名称解析

3、各节点时间同步

4、关闭SELinux, 防火墙

二、配置存储

1、配置iscsi服务端

[root@iSCSI ~]# yum install -y scsi-target-utils

[root@iSCSI ~]# vim /etc/tgt/targets.conf

<target iqn.2017-09.com.linux:test01>

backing-store /dev/vdb

initiator-address 192.168.122.103

initiator-address 192.168.122.104

</target>

[root@iSCSI ~]# service tgtd start

[root@iSCSI ~]# chkconfig tgtd on

[root@iSCSI ~]# netstat -antp | grep tgtd

tcp 0 0 0.0.0.0:3260 0.0.0.0:\* LISTEN 1429/tgtd

tcp 0 0 :::3260 :::\* LISTEN 1429/tgtd

[root@iSCSI ~]#

2、分别配置web01, web02连接后端存储

[root@web01 ~]# yum install -y iscsi-initiator-utils

[root@web01 ~]# iscsiadm -m discovery -t st -p 192.168.122.105

192.168.122.105:3260,1 iqn.2017-09.com.linux:test01

[root@web01 ~]# iscsiadm -m node -T iqn.2017-09.com.linux:test01 -p 192.168.122.105 -l

Logging in to [iface: default, target: iqn.2017-09.com.linux:test01, portal: 192.168.122.105,3260] (multiple)

Login to [iface: default, target: iqn.2017-09.com.linux:test01, portal: 192.168.122.105,3260] successful.

[root@web01 ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 252:0 0 8G 0 disk

├─vda1 252:1 0 200M 0 part /boot

└─vda2 252:2 0 7.8G 0 part /

sda 8:0 0 20G 0 disk

[root@web01 ~]#

web02采用相同操作连接后端存储

3、在web01/web02上创建RHCS集群，创建逻辑卷、gfs2文件系统

1) 集群创建结果如下

[root@web01 ~]# clustat

Cluster Status for mycluster @ Thu Sep 21 12:05:16 2017

Member Status: Quorate

Member Name ID Status

------ ---- ---- ------

web01 1 Online, Local

web02 2 Online

2) 在web01上创建集群逻辑卷

[root@web01 ~]# pvcreate /dev/sda

[root@web01 ~]# vgcreate vg1 /dev/sda

[root@web01 ~]# lvcreate -L 10G -n lv1 vg1

[root@web01 ~]# lvscan

ACTIVE '/dev/vg1/lv1' [10.00 GiB] inherit

分别启动web01, web02上的clvmd服务，确保两边的逻辑卷状态为ACTIVE

[root@web01 ~]# service clvmd start

[root@web01 ~]# chkconfig clvmd on

[root@web01 ~]# lvscan

ACTIVE '/dev/vg1/lv1' [10.00 GiB] inherit

[root@web02 ~]# service clvmd start

[root@web02 ~]# chkconfig clvmd on

[root@web02 ~]# lvscan

ACTIVE '/dev/vg1/lv1' [10.00 GiB] inherit

3) 创建gfs2文件系统

[root@web01 ~]# mkfs.gfs2 -t mycluster:tb01 -p lock\_dlm -j 2 -J 128M /dev/vg1/lv1

This will destroy any data on /dev/vg1/lv1.

It appears to contain: symbolic link to `../dm-0'

Are you sure you want to proceed? [y/n] y

Device: /dev/vg1/lv1

Blocksize: 4096

Device Size 10.00 GB (2621440 blocks)

Filesystem Size: 10.00 GB (2621438 blocks)

Journals: 2

Resource Groups: 40

Locking Protocol: "lock\_dlm"

Lock Table: "mycluster:tb01"

UUID: 8250ce7f-9a71-1459-5223-dcebecb508e6

在web01, web02上分别安装httpd，准备测试页面

[root@web01 ~]# yum install -y httpd

[root@web01 ~]# mount -t gfs2 /dev/vg1/lv1 /var/www/html/

[root@web01 ~]# echo "lvs + keepalived + rhcs + clvmd + gfs2 + iscsi" > /var/www/html/index.html

[root@web01 ~]# service httpd start

[root@web01 ~]# chkconfig httpd on

web02同上

[root@web02 ~]# rpm -q httpd

httpd-2.2.15-39.el6.centos.x86\_64

[root@web02 ~]# mount -t gfs2 /dev/vg1/lv1 /var/www/html/

[root@web02 ~]# ls /var/www/html/

index.html

三、配置LVS集群

1、在所有real server配置vip, 修改arp的内核参数

[root@web01 ~]# ip addr add dev lo 192.168.122.100/32

[root@web01 ~]# echo 1 > /proc/sys/net/ipv4/conf/all/arp\_ignore

[root@web01 ~]# echo 2 > /proc/sys/net/ipv4/conf/all/arp\_announce

[root@web02 ~]# ip addr add dev lo 192.168.122.100/32

[root@web02 ~]# echo 1 > /proc/sys/net/ipv4/conf/all/arp\_ignore

[root@web02 ~]# echo 2 > /proc/sys/net/ipv4/conf/all/arp\_announce

2、配置调度器

[root@director01 ~]# yum install -y ipvsadm keepalived

[root@director02 ~]# yum install -y ipvsadm keepalived

director01的配置文件

[root@director01 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id director01

}

vrrp\_instance VI\_1 {

state MASTER

interface eth0

virtual\_router\_id 51

priority 100

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

virtual\_ipaddress {

192.168.122.100

}

}

virtual\_server 192.168.122.100 80 {

delay\_loop 6

lb\_algo rr

lb\_kind DR

nat\_mask 255.255.255.0

persistence\_timeout 50

protocol TCP

real\_server 192.168.122.103 80 {

weight 1

HTTP\_GET {

url {

path /check.html

digest 5e9b13ce8f6c99f3f510756be58d15fe

}

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.122.104 80 {

weight 1

HTTP\_GET {

url {

path /check.html

digest 5e9b13ce8f6c99f3f510756be58d15fe

}

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

==================

使用检测页面md5值的方式作健康状态检测

[root@director02 ~]# genhash -s 192.168.122.103 -p 80 -u /check.html

MD5SUM = 5e9b13ce8f6c99f3f510756be58d15fe

==================

directory02的配置文件

[root@director02 ~]# cat /etc/keepalived/keepalived.conf

! Configuration File for keepalived

global\_defs {

router\_id director02

}

vrrp\_instance VI\_1 {

state BACKUP

interface eth0

virtual\_router\_id 51

priority 50

advert\_int 1

authentication {

auth\_type PASS

auth\_pass redhat

}

virtual\_ipaddress {

192.168.122.100

}

}

virtual\_server 192.168.122.100 80 {

delay\_loop 6

lb\_algo rr

lb\_kind DR

nat\_mask 255.255.255.0

persistence\_timeout 50

protocol TCP

real\_server 192.168.122.103 80 {

weight 1

HTTP\_GET {

url {

path /check.html

digest 5e9b13ce8f6c99f3f510756be58d15fe

}

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

real\_server 192.168.122.104 80 {

weight 1

HTTP\_GET {

url {

path /check.html

digest 5e9b13ce8f6c99f3f510756be58d15fe

}

connect\_timeout 3

nb\_get\_retry 3

delay\_before\_retry 3

}

}

}

[root@director02 ~]#

3、启动keepalived服务

[root@director01 ~]# service keepalived start

[root@director01 ~]# chkconfig keepalived on

**扩展LB集群：**

1) 连接后端存储

2) 加入HA集群

1) 安装RHCS集群的软件

2) 拷贝集群配置文件

3) web界面

4) clvmd服务

4) 扩展gfs文件系统的日志个数

5) 安装httpd, VIP, ar内核参数

6) 修改keepalived配置文件，添加real server