**CENTOS7 构建HA集群**

作者：linjiang2718

2014-11-10

CENTOS6 上支持的RHCS组件包主要有cman（心跳管理）、luci+ricci（配置工具）、rgmanager（资源管理），通过图形化配置相当简单，但是自CENTOS7开始，系统已经不再集成RHCS套件，并且在rpmfind上找不到支持centos7的RHCS组件包，在CENTOS7中默认采用corosync（心跳管理）、pacemaker（资源管理）、pcs（配置工具）同样可以构建HA集群，配置方法较之前有很大区别，但是原理相似。

## 一、系统环境配置

1.1、修改名称

hostnamectl set-hostname node01

1.2两台机器修改关闭防火墙、SELINUX、配置IP、修改HOSTS

禁止防火墙和selinux

# systemctl disable firewalld

# systemctl stop firewalld

修改/etc/sysconfig/selinux

确保SELINUX=disabled，然后执行setenforce 0 或者reboot服务器以生效

设置防火墙规则（如果关闭防火墙则不用配置）

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

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

配置IP地址及网关、DNS

vi /etc/sysconfig/network-scripts/ifcfg-eth0 #编辑配置文件,添加修改以下内容

BOOTPROTO=static #启用静态IP地址

ONBOOT=yes #开启自动启用网络连接

IPADDR=192.168.102.1 #设置IP地址

NETMASK=255.255.255.0 #设置子网掩码

GATEWAY=192.168.102.250 #设置网关

DNS1=8.8.8.8 #设置主DNS

DNS2=8.8.4.4 #设置备DNS

IPV6INIT=no #禁止IPV6

配置HOSTS，两边一样。

[root@node01 ~]# cat /etc/hosts

192.168.102.1 node01

192.168.102.2 node02

192.168.102.10 vip #集群IP

## 二、集群环境准备

两台机器启动PCSD

systemctl start pcsd.service

systemctl enable pcsd.service

两台机器passwd hacluster

输入同一个密码hacluster

配置节点之间的相互认证

pcs cluster auth node01 node02

[root@node01 ~]# pcs cluster auth node01 node02

Username: hacluster

Password:

node01: Authorized

node02: Authorized

出现以下显示表示认证成功。

[root@node02 ~]# pcs cluster auth node01 node02

node01: Already authorized

node02: Already authorized

各节点之间时间同步

在node1和node2分别进行时间同步，可以使用ntp实现。

[root@node1 ~]# ntpdate 172.16.0.1 //172.16.0.1 为时间服务器（若无时间服务器手动调整时间也可以）

## 三、集群创建

以nascluster为本集群的名称。

[root@node01 ~]# pcs cluster setup --name nascluster node01 node02

Shutting down pacemaker/corosync services...

Redirecting to /bin/systemctl stop pacemaker.service

Redirecting to /bin/systemctl stop corosync.service

Killing any remaining services...

Removing all cluster configuration files...

node01: Succeeded

node02: Succeeded

pcs在执行以上命令时会生产corosync.conf及修改cib.xml文件，corosync.conf问corosync的配置文件，cib.xml为pacemaker的配置文件。这两个配置文件是集群的核心配置，重装系统时建议做好这两个配置文件的备份。

开启集群

pcs cluster start --all 开启集群

pcs cluster enable --all 设置集群自动启动

pcs cluster status 查看集群状态

当有半数以上的节点在线时,这个集群就认为自己拥有法定人数了,是“合法”的,换而言之就是下面的公式: total\_nodes < 2 \* active\_nodes，也就是说假设集群中有4个节点，但是活动节点有3个，4<2\*3,即是合法的集群，如果活动节点有2个，4=2\*2不能满足total\_nodes < 2 \* active\_nodes了，这个集群就非法了。

因此在双节点的集群中只有当两者都在线时才是合法的的这个规则会让“双节点集群”毫无意义，但是我们可以控制Pacemaker发现集群达不到法定人数时候的行为。简单来说，我们告诉集群忽略它。

在两个节点的情况下设置以下值

pcs property set no-quorum-policy=ignore

集群故障时候服务迁移

pcs resource defaults migration-threshold=1

查看集群状态

[root@node01 ~]# pcs cluster status

Cluster Status:

Last updated: Tue Sep 16 06:35:31 2014

Last change: Tue Sep 16 06:35:24 2014 via crmd on node02

Stack: corosync

Current DC: node02 (2) - partition with quorum

Version: 1.1.10-29.el7-368c726

2 Nodes configured

0 Resources configured

PCSD Status:

node01: Online

node02: Online

查看集群成员

[root@node01 ~]# corosync-cmapctl |grep members

runtime.totem.pg.mrp.srp.members.1.config\_version (u64) = 0

runtime.totem.pg.mrp.srp.members.1.ip (str) = r(0) ip(192.168.102.1)

runtime.totem.pg.mrp.srp.members.1.join\_count (u32) = 1

runtime.totem.pg.mrp.srp.members.1.status (str) = joined

runtime.totem.pg.mrp.srp.members.2.config\_version (u64) = 0

runtime.totem.pg.mrp.srp.members.2.ip (str) = r(0) ip(192.168.102.2)

runtime.totem.pg.mrp.srp.members.2.join\_count (u32) = 1

runtime.totem.pg.mrp.srp.members.2.status (str) = joined

查看成员信息

[root@node01 ~]# pcs status corosync

Membership information

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

Nodeid Votes Name

1 1 node01 (local)

2 1 node02

查看XML

[root@node01 ~]# pcs cluster cib

检查配置文件，发现提示没有STONITH设备。

[root@node01 ~]# crm\_verify -L -V

error: unpack\_resources: Resource start-up disabled since no STONITH resources have been defined

error: unpack\_resources: Either configure some or disable STONITH with the stonith-enabled option

error: unpack\_resources: NOTE: Clusters with shared data need STONITH to ensure data integrity

Errors found during check: config not valid

暂时关闭stonish，后面再详细讲讲stonish的配置。

pcs property set stonith-enabled=false

配置集群IP

pcs resource create vip ocf:heartbeat:IPaddr2 ip=192.168.102.10 cidr\_netmask=24 op monitor interval=30s

vip为自定义的一个集群IP的名称，监控时间为30S。

测试浮动IP地址在集群中能否正常切换：

停node01的集群服务pcs cluster stop node01

在node02上查看状态

[root@node02 ~]# pcs status

Cluster name: nascluster

Last updated: Tue Sep 16 07:28:33 2014

Last change: Tue Sep 16 07:26:41 2014 via cibadmin on node01

Stack: corosync

Current DC: node02 (2) - partition with quorum

Version: 1.1.10-29.el7-368c726

2 Nodes configured

1 Resources configured

Online: [ node02 ]

OFFLINE: [ node01 ]

Full list of resources:

vip (ocf::heartbeat:IPaddr2): Started node02 ##此条表示vip在node02上执行了

PCSD Status:

node01: Online

node02: Online

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/disabled

在node01恢复后，为防止node02资源迁回node01（迁来迁去对还是会对业务有一定影响）

pcs resource defaults resource-stickiness=100

pcs resource defaults

## 四、创建GFS2文件系统

mkfs.gfs2 -p lock\_dlm -j 2 -t nascluster:nas /dev/my\_vg/gfsdata

配置dlm锁监控时间

pcs cluster cib dlm\_cfg ##由于cib.xml非常重要，为避免出错不建议直接修改，故生产一个dlm\_cfg文件，将配置写入dlm\_cfg文件后，再将dlm\_cfg写入cib.xml中。

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 -f dlm\_cfg resource show

pcs cluster cib-push dlm\_cfg

普及一下dlm锁的知识：

Distributed Lock Manager，简称DLM，是一个分布式锁管理器，它是RHCS的一个底层基础构件，同时也为集群提供了一个公用的锁运行机制。DLM运行在每个节点上,GFS通过锁管理器的机制来同步访问文件系统的元数据。CLVM通过锁管理器来同步更新数据到LVM卷和卷组。  
 DLM不需要设定锁管理服务器，它采用对等的锁管理方式，大大提高了处理性能。同时，DLM避免了单个节点失败需要整体恢复的性能瓶颈。另外，DLM的请求是本地的，不需要网络请求，因此请求会立即生效。最后，DLM通过分层机制，可以实现多个锁空间的并行锁模式。

查看状态

[root@node01 nas]# pcs status

Cluster name: nascluster

Last updated: Tue Sep 16 07:48:09 2014

Last change: Tue Sep 16 07:44:08 2014 via cibadmin on node01

Stack: corosync

Current DC: node02 (2) - partition with quorum

Version: 1.1.10-29.el7-368c726

2 Nodes configured

3 Resources configured

Online: [ node01 node02 ]

Full list of resources:

vip (ocf::heartbeat:IPaddr2): Started node02

Clone Set: dlm-clone [dlm]

Started: [ node01 node02 ] ##DLM的状态

PCSD Status:

node01: Online

node02: Online

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

## 五、stonith配置

查看本系统支持的fence设备

pcs stonith list

查看即将要使用的fence设备相关信息

pcs stonith describe fence\_ilo4

由于服务器是HP DL380 GEN8的，支持ILO4，但是在实际配置中采用fence\_ilo4却发现怎么也不通,man fence\_ilo4发现fence\_ipmilan 也可以配置ilo4的FENCE设备，但是必须要加lanplus="true"参数。

pcs cluster cib stonith\_cfg ##生产初始配置文件stonith\_cfg

pcs -f stonith\_cfg stonith create ipmi-fence-node01 fence\_ipmilan parms lanplus="true" pcmk\_host\_list="node01" pcmk\_host\_check="static-list" action="reboot" ipaddr="192.168.103.1" login=USERID passwd=password op monitor interval=60s

pcs -f stonith\_cfg stonith create ipmi-fence-node02 fence\_ipmilan parms lanplus="true" pcmk\_host\_list="node02" pcmk\_host\_check="static-list" action="reboot" pcmk\_host\_list="node02" ipaddr="192.168.103.2" login=USERID passwd=password op monitor interval=60s

解释：创建一个名为ipmi-fence-node01的fence设备名称用于建node01的fence，pcmk\_host\_check="static-list"的功能是将node01与192.168.103.1对应，后面login=USERID passwd=password op monitor interval=60s不再解释。

pcs -f stonith\_cfg stonith

检查stonith\_cfg中stonith配置信息

pcs -f stonith\_cfg property set stonith-enabled=true 上文关闭了stonish，现在开启stonish

pcs -f stonith\_cfg property

检查stonith\_cfg中stonith是否已经开启

pcs cluster cib-push stonith\_cfg 将stonith\_cfg写入cib.xml

node02上测试FENCE是否成功stonith\_admin --reboot node01

node01上测试FENCE是否成功stonith\_admin --reboot node02

pcs cluster standby node01 将node01业务VIP迁移到node02上测试集群是否正常。

在集群所有节点重启后发现gfs的文件系统无法自动挂载，定义了fstab也不行，经过分析发现系统开机时候执行了pcs cluster start，而且很快就返回启动成功的结果，但是系统日志里面显示仍然还启动完成，考虑到系统进入系统之后集群有可能还没启动完成，所有自然无法挂载gfs的文件系统。

如果无法自动挂载，我自己编写了一个循环挂载的脚本。供参考。

编辑脚本

mountnas.sh

#!/bin/bash

i=1

while(($i<50))

do

mount |grep nas

if [ $? = 1 ];then mount /dev/my\_vg/gfsdata /nas

else exit

fi

sleep 3

done

chmod 777 /etc/rc.local centos7 还需要给rc.local加个权限，要不然开机不会执行rc.local

在/etc/rc.local

加入bash /mountnas.sh

## 六、配置多路心跳

在rhcs中，可以配置多路心跳，一路将fence作为心跳，一路将网络作为心跳，在corosync pacemaker的集群中，找了国内外很多技术文章，均未做相关描述的。主要是因为corosync pacemaker的集群相对来说已经很可靠，在未配置多路心跳之前将业务网卡DOWN后，发现 node02的集群马上失效，集群IP立即切换至node01上，未出现脑裂的情况。

作为自己研究，尝试了再配置了一个心跳。

在/etc/corosync.conf中，毕竟corosync管理心跳，所以只有在它里面想办法了，在centos7之前的corosync.conf配置中，会定义interface {}作为多个网络，但是在centos7中经过PCS统一管理后，经过测试发现pacemaker只认nodelist{}作为网络，难道这又是一个变动？

在totem {}

定义：

rrp\_mode: passive #默认为none，修改为passive才可以支持两个网段

nodelist {

node{

ring0\_addr:node01

ring0\_addr:test01 -test01为第二个心跳

}

node{

ring0\_addr:node02

ring0\_addr:test02 -test02为第二个心跳

}

}

记得修改host表。

重启集群即可生效，目前node01，node02对应192.168.102.0网段，test01 test02对应192.168.103.0网段,同时可以在192.168.103.0网段上再新建一个业务IP，

在配置集群IP之前需要对test01 test02配置认证。

[root@node01 ~]# pcs cluster auth test01 test02

Username: hacluster

Password:

test01: Authorized

test02: Authorized

出现以下显示表示认证成功。

[root@node02 ~]# pcs cluster auth test01 test02

test01: Already authorized

test02: Already authorized

配置集群IP

pcs resource create testip ocf:heartbeat:IPaddr2 ip=192.168.103.10 cidr\_netmask=24 op monitor interval=30s 注意testip与上文的vip已经名字不一样，两个集群IP的名字不能一致。

## 七、配置集群应用

以apache为例：

apache安装过程略。

pcs resource create Web ocf:heartbeat:apache configfile=/etc/httpd/conf/httpd.conf statusurl="http://localhost/server-status" op monitor interval=1min

pcs status查看apache状态，

pcs constraint colocation add Web vip INFINITY 将apache与vip绑定在同一个节点上。

pcs constraint order vip then Web，指定集群IP先启动，然后再启动apache

pcs constraint location Web prefers node01=200 指定node01优先启动apache

crm\_simulate -sL 查看资源黏性值，集群业务优先在资源黏性值高的节点上运行。

Current cluster status:

Online: [ node01 node02 ]

vip (ocf:heartbeat:IPaddr2): Started node01

Web (ocf:heartbeat:apache): Started node01

Allocation scores:

native\_color: vip allocation score on node01: 200

native\_color: vip allocation score on node02: 50

native\_color: Web allocation score on node01: 200

native\_color: Web allocation score on node02: 50

Transition Summary:

手动切换集群资源

以上配置由集群自动分配资源到node01、node02上，有时候为了维护，必须手动迁移资源到指定节点上。

# pcs constraint location Web prefers node01=INFINITY

# pcs constraint --full 查看资源状态

# pcs constraint remove location-Web-node01-INFINITY 将控制权还给集群