corosync+pacemaker+crmsh实现高可用

2015-07-30      [0](javascript:void(0);)[个评论](http://www.2cto.com/net/201507/425844.html#SOHUCS)

[收藏](javascript:;)    [[http://www.2cto.com/statics/images/dl.jpg](http://www.2cto.com/index.php?m=member&c=content&a=publish&modid=1&siteid=1)我要投稿](http://www.2cto.com/index.php?m=member&c=content&a=publish&modid=1&siteid=1)

目录

1、引言及环境介绍

2、高可用环境部署

3、crmsh接口使用介绍

4、案例

5、总结

1、引言及环境介绍

在上一博文中介绍了一些关于高可用技术的理论基础知识，这一博文则是介绍corosync+pacemakcer这一高可用方案的安装部署，并会以实际的案例来演示高可用的实现，corosync提供集群的信息层（messaging layer）的功能，传递心跳信息和集群事务信息，pacemaker工作在资源分配层，提供资源管理器的功能，并以crmsh这个资源配置的命令接口来配置资源。在进入主题前先来介绍一下常见的开源高可用方案和这次环境搭建的系统环境。

常见的HA开源方案：

heartbeat v1 + haresources

heartbeat v 2 + crm

heartbeat v3 + cluster-glue + pacemaker

corosync + cluster-glue + pacemaker

cman + rgmanager

keepalived + script

此次测试的系统环境：

[root@nod1tomcat]#cat/etc/issue

CentOSrelease6.4(Final)

Kernel

onanm

[root@nod1tomcat]#uname-r

2.6.32-358.el6.x86\_64

两个节点都是采用相同的操作系统

2、高可用环境部署

[root@nod1~]#yum-yinstallpacemakcercorosync#pacemaker和corosync采用yum方式安装即可，前提是你要配置好yum源，注意：两个节点都要进行安装

/usr/sbin/corosync-keygen#为corosync生成authkey的命令，此命令是根据内核的熵池来生成认证文件的，如果熵池的随机性不足，则会运行此命令后一直卡着，此时用户只有不断的敲击键盘使产生足够的随机数后才能生成authkdy文件

生成集群节点间的认证文件：

[root@nod1~]#corosync-keygen#生成认证文件

CorosyncClusterEngineAuthenticationkeygenerator.

Gathering1024bitsforkeyfrom/dev/random.

Presskeysonyourkeyboardtogenerateentropy.

Presskeysonyourkeyboardtogenerateentropy(bits=80).

#熵池随机性不足时一直卡在这里，这里可以另开窗口进行其他的配置

提供corosync的配置文件，利用模板生成：

[root@nod1~]#cd/etc/corosync

[root@nod1corosync]#cpcorosync.conf.examplecorosync.conf

[root@nod1corosync]#ls

corosync.conf.exampleservice.dcorosync.confcorosync.conf.example.udpuuidgid.d

[root@nod1corosync]#vimcorosync.conf

#Pleasereadthecorosync.conf.5manualpage

compatibility:whitetank#表示兼容whitetank版本，其实是corosync0.8之前的版本

totem{#定义集群环境下各corosync间通讯机制

version:2

#secauth:Enablemutualnodeauthentication.Ifyouchooseto

#enablethis(on),thendoremembertocreateashared

#secretwithcorosync-keygen.

#secauth:off

secauth:on#表示基于authkey的方式来验证各节点

threads:0#启动的线程数，0表示不启动线程机制，默认即可

#interface:defineatleastoneinterfacetocommunicate

#over.Ifyoudefinemorethanoneinterfacestanza,youmust

#alsosetrrp\_mode.

interface{#定义哪个接口来传递心跳信息和集群事务信息

#Ringsmustbeconsecutivelynumbered,startingat0.

ringnumber:0#表示心跳信息发出后能够在网络中转几圈，保持默认值即可

#Thisisnormallythe\*network\*addressofthe

#interfacetobindto.Thisensuresthatyoucanuse

#identicalinstancesofthisconfigurationfile

#acrossallyourclusternodes,withouthavingto

#modifythisoption.

bindnetaddr:192.168.0.0#绑定的网络地址

#However,ifyouhavemultiplephysicalnetwork

#interfacesconfiguredforthesamesubnet,thenthe

#networkaddressaloneisnotsufficienttoidentify

#theinterfaceCorosyncshouldbindto.Inthatcase,

#configurethe\*host\*addressoftheinterface

#instead:

#bindnetaddr:192.168.1.1

#Whenselectingamulticastaddress,considerRFC

#2365(which,amongotherthings,specifiesthat

#239.255.x.xaddressesarelefttothediscretionof

#thenetworkadministrator).Donotreusemulticast

#addressesacrossmultipleCorosyncclusterssharing

#thesamenetwork.

mcastaddr:239.255.21.111#监听的多播地址，不要使用默认

#CorosyncusestheportyouspecifyhereforUDP

#messaging,andalsotheimmediatelypreceding

#port.Thusifyousetthisto5405,Corosyncsends

#messagesoverUDPports5405and5404.

mcastport:5405#corosync间传递信息使用的端口，默认即可

#Time-to-liveforclustercommunicationpackets.The

#numberofhops(routers)thatthisringwillallow

#itselftopass.Notethatmulticastroutingmustbe

#specificallyenabledonmostnetworkrouters.

ttl:1#包的生存周期，保持默认即可

}

}

logging{

#Logthesourcefileandlinewheremessagesarebeing

#generated.Whenindoubt,leaveoff.Potentiallyusefulfor

#debugging.

fileline:off

#Logtostandarderror.Whenindoubt,settono.Usefulwhen

#runningintheforeground(wheninvokingcorosync-f)

to\_stderr:no

#Logtoalogfile.Whensettono,thelogfileoption

#mustnotbeset.

to\_logfile:yes

logfile:/var/log/cluster/corosync.log

#Logtothesystemlogdaemon.Whenindoubt,settoyes.

to\_syslog:no#关闭日志发往syslog

#Logdebugmessages(veryverbose).Whenindoubt,leaveoff.

debug:off

#Logmessageswithtimestamps.Whenindoubt,settoon

#(unlessyouareonlyloggingtosyslog,wheredouble

#timestampscanbeannoying).

timestamp:on#打印日志时是否记录时间戳，会消耗较多的cpu资源

logger\_subsys{

subsys:AMF

debug:off

}

}

#新增加以下内容

service{

ver:0

name:pacemaker#表示以插件化方式启用pacemaker

}

aisexec{#运行openaix时所使用的用户及组，默认时也是采用root，可以不定义

user:root

group:root

}

当corosync-keygen命令顺利运行完成后，在/etc/corosync/目录下生成authkey认证文件：

[root@nod1corosync]#ls

authkeycorosync.conf.exampleservice.d

corosync.confcorosync.conf.example.udpuuidgid.d

[root@nod1corosync]#scpauthkeycorosync.confnod2.test.com:/etc/corosync/#把认证文件与配置文件拷贝到另一节点

[root@nod1corosync]#servicecorosyncstart#启动服务，不要忘记另一个节点也要把corosync服务启动

验证corosync服务是否正常启动，在集群环境应对每个服务器都要验证：

验证corosync是否启动成功：

[root@nod1corosync]#grep-eCorosyncClusterEngine/var/log/cluster/corosync.log#查看corosync集群引擎是否启动

Jul1921:45:48corosync[MAIN]CorosyncClusterEngine('1.4.7'):startedandreadytoprovideservice.

[root@nod1corosync]#grep-econfigurationfile/var/log/cluster/corosync.log#查看corosync的配置文件是否成功加载

Jul1921:45:48corosync[MAIN]Successfullyreadmainconfigurationfile'/etc/corosync/corosync.conf'.

查看定义的TOTEM接口是否启用：

[root@nod1corosync]#grepTOTEM/var/log/cluster/corosync.log

Jul1921:45:48corosync[TOTEM]Initializingtransport(UDP/IPMulticast).

Jul1921:45:48corosync[TOTEM]Initializingtransmit/receivesecurity:libtomcryptSOBER128/SHA1HMAC(mode0).

Jul1921:45:48corosync[TOTEM]Thenetworkinterface[192.168.0.201]isnowup.

Jul1921:45:48corosync[TOTEM]Aprocessorjoinedorleftthemembershipandanewmembershipwasformed.

验证启动时是否有错误：

[root@nod1corosync]#grepERROR/var/log/cluster/corosync.log

Jul1921:45:48corosync[pcmk]ERROR:process\_ais\_conf:YouhaveconfiguredaclusterusingthePacemakerpluginforCorosync.Thepluginisnotsupportedinthisenvironmentandwillberemovedverysoon.

Jul1921:45:48corosync[pcmk]ERROR:process\_ais\_conf:PleaseseeChapter8of'ClustersfromScratch'(http://www.clusterlabs.org/doc)fordetailsonusingPacemakerwithCMAN

#上边的错误信息可以忽略，这里报错的信息主要意思是说pacemaker是以插件的方式配置的，在以后的版本中将不再支持

验证pacemaker是否正常启动：

[root@nod1corosync]#greppcmk\_startup/var/log/cluster/corosync.log

Jul1921:45:48corosync[pcmk]info:pcmk\_startup:CRM:Initialized

Jul1921:45:48corosync[pcmk]Logging:Initializedpcmk\_startup

Jul1921:45:48corosync[pcmk]info:pcmk\_startup:Maximumcorefilesizeis:18446744073709551615

Jul1921:45:48corosync[pcmk]info:pcmk\_startup:Service:9

Jul1921:45:48corosync[pcmk]info:pcmk\_startup:Localhostname:nod1.test.com

3、crmsh接口使用介绍

pacemaker的配置接口有两种，一是crmsh，另一个是pcs，主里以crmsh的使用为例。

crmsh依赖pssh这个包，所以两个都需要分别在各个集群节点上进行安装，这两个包可以在这里进行[下载](http://www.2cto.com/soft)http://crmsh.github.io/

[root@nod1~]#ls

crmsh-2.1-1.6.x86\_64.rpmpssh-2.3.1-2.el6.x86\_64.rpm

[root@nod1~]#yuminstallcrmsh-2.1-1.6.x86\_64.rpmpssh-2.3.1-2.el6.x86\_64.rpm

crmsh的crm命令有两种模式：一种是命令模式，当执行一个命令，crmsh会把执行得到的结果输出到shell的标准输出；另一种是交互式模式；下边将有大量的例子来说明。

crm命令的使用：

[root@nod1~]#crm#直接使用crm进入交互式模式

crm(live)#

crm(live)#help#查看帮助信息获取crm支持哪些子命令

crmsh常用的子命令：

status：查看集群的状态信息

configure：配置集群的命令

node：管理节点状态

ra：配置资源代理

resource：管理资源的子命令，比如关闭一个资源，清除资源的当前状态（比如一些出错信息）

接下来先查看一下集群的状态信息：

[root@nod1~]#crm

crm(live)#status

Lastupdated:TueJul2121:21:352015

Lastchange:SunJul1923:01:342015

Stack:classicopenais(withplugin)#这里表示基于插件化的方式用openais中的corosync调用pacemaker来工作的

CurrentDC:nod1.test.com-partitionwithquorum#DesignatedCoordinate简称DC，表示指定的协调员，这里表示nod1.test.com就是集群中的事务协调员，“partitionwithquorum”就表示当前分区是拥有法定票数的

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes#表示配置了2个节点，预计的投票数为2票

0Resourcesconfigured#表示没有配置集群资源

Online:[nod1.test.comnod2.test.com]#这里显示两个节点都是在线的

查看集群默认的配置信息：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#show#使用show这个子命令就能查看当前集群的配置信息，使用“showxml”能以xml文件格式显示出当前的配置信息

nodenod1.test.com

nodenod2.test.com

propertycib-bootstrap-options:

dc-version=1.1.11-97629de

cluster-infrastructure=classicopenais(withplugin)

expected-quorum-votes=2

stonith-enabled=true

no-quorum-policy=stop

last-lrm-refresh=1436887216

crm(live)configure#verify#verify是检查配置文件是否有错误

error:unpack\_resources:Resourcestart-updisabledsincenoSTONITHresourceshavebeendefined

error:unpack\_resources:EitherconfiguresomeordisableSTONITHwiththestonith-enabledoption

error:unpack\_resources:NOTE:ClusterswithshareddataneedSTONITHtoensuredataintegrity

Errorsfoundduringcheck:confignotvalid

#这里报了一些错误，表示默认时没有定义STONITH设备，在corosync+pacemaker的集群是不允许的，当然可以定义忽略这个检查，下边有介绍

用property子命令定义集群的全局属性：

[root@nod1~]#crm

crm(live)configure#property#在crmsh接口中是支持tab键命令补全功能的，这里输入property后连续敲击两下tab键就可列出可配置的参数

batch-limit=maintenance-mode=remove-after-stop=

cluster-delay=migration-limit=shutdown-escalation=

cluster-recheck-interval=no-quorum-policy=start-failure-is-fatal=

crmd-transition-delay=node-action-limit=startup-fencing=

dc-deadtime=node-health-green=stonith-action=

default-action-timeout=node-health-red=stonith-enabled=

default-resource-stickiness=node-health-strategy=stonith-timeout=

election-timeout=node-health-yellow=stop-all-resources=

enable-acl=pe-error-series-max=stop-orphan-actions=

enable-startup-probes=pe-input-series-max=stop-orphan-resources=

is-managed-default=pe-warn-series-max=symmetric-cluster=

load-threshold=placement-strategy=

crm(live)configure#property stonith-enabled=false#把stonith设备的支持关闭，不然我们在想使用corosync的集群功能就需要定义stonith设备

crm(live)configure#show

nodenod1.test.com

nodenod2.test.com

propertycib-bootstrap-options:

dc-version=1.1.11-97629de

cluster-infrastructure=classicopenais(withplugin)

expected-quorum-votes=2

stonith-enabled=false#已是false状态

no-quorum-policy=stop

last-lrm-refresh=1436887216

crm(live)configure#verify#再校验配置就不会报错了

crm(live)configure#commit#提交配置

集群资源的配置

要想获取资源的详细信息就需要去ra（resource agent）中去查看，比如我们要定义一个虚拟ip资源：

[root@nod1~]#crm

crm(live)#ra

crm(live)ra#classes#查看集群资源有哪些类型

lsb

ocf/heartbeatpacemaker

service

stonith

crm(live)ra#listocf#列出ocf这个类型下有哪些资源代理，下边就有IPaddr这个关于设置ip的资源代理

CTDBClusterMonDelayDummyFilesystemHealthCPUHealthSMART

IPaddrIPaddr2IPsrcaddrLVMMailToRouteSendArp

SquidStatefulSysInfoSystemHealthVirtualDomainXinetdapache

conntrackdcontrolddb2dhcpdethmonitorexportfsiSCSILogicalUnit

mysqlnamednfsnotifynfsserverpgsqlpingpingd

postfixremotersyncdsymlinktomcat

crm(live)ra#metaocf:IPaddr#使用meta子命令能获取到一个资源的详细信息，即此资源的使用帮助信息

定义主资源用primitive命令：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#primitive webipocf:IPaddr params ip=192.168.0.100

crm(live)configure#verify

crm(live)configure#commit#一旦提交成功，此资源就开始生效了

crm(live)configure#cd..

crm(live)#status

Lastupdated:TueJul2122:14:432015

Lastchange:TueJul2122:12:442015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partition with quorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

1Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Started nod1.test.com#这里就是我们定义好的资源，在nod1.test.com节点启用了

[root@nod1~]#ipadd show

1:lo:mtu16436qdiscnoqueuestateUNKNOWN

link/loopback00:00:00:00:00:00brd00:00:00:00:00:00

inet127.0.0.1/8scopehostlo

inet6::1/128scopehost

valid\_lftforeverpreferred\_lftforever

2:eth0:mtu1500qdiscpfifo\_faststateUNKNOWNqlen1000

link/ether00:0c:29:07:89:febrdff:ff:ff:ff:ff:ff

inet192.168.0.201/24brd192.168.0.255scopeglobaleth0

inet192.168.0.100/24brd192.168.0.255scopeglobalsecondaryeth0

inet6fe80::20c:29ff:fe07:89fe/64scopelink

valid\_lftforeverpreferred\_lftforever

#我们定义的ip已生效

定义nginx的这个服务资源：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#primitiv enginx lsb:nginx#nginx这个服务是在lsb这个资源类别下的资源代理，primitive命令后的第一个nginx是定义集群资源的一个名称

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd..

crm(live)#status

Lastupdated:TueJul2122:25:002015

Lastchange:TueJul2122:24:582015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

nginx(lsb:nginx):Startednod2.test.com

#nginx这个资源在nod2.test.com节点启动起来了，这也验证了在高可用集群中集群会尽可能让资源分摊到各个节点的特性，而在实际环境中我们希望webip与nginx这两个资源是运行在同一个节点上的。

要想让多个资源同时运行在同一个节点上可以把多个资源定义在一个group中或定义排列约束（colocation）：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#group webservice webip nginx

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd..

crm(live)#status

Lastupdated:TueJul2122:30:192015

Lastchange:TueJul2122:30:172015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

nginx(lsb:nginx):Startednod1.test.com

#两个资源同时运行在nod1.test.com上了

接下验证资源是否能转移到其他节点上：

[root@nod1~]#crm node standby#把当前节点转换成standby状态

[root@nod1~]#crmstatus

Lastupdated:TueJul2122:37:142015

Lastchange:TueJul2122:37:092015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Nodenod1.test.com:standby

Online:[nod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

nginx(lsb:nginx):Startednod2.test.com

#webservice组中的资源已转换到了nod2.test.com节点上

再让nod1.test.com重新上线，观察资源是否能转移回来：

[root@nod1~]#crm node online#让当前节点重新上线

Youhavenewmailin/var/spool/mail/root

[root@nod1~]#crmstatus

Lastupdated:TueJul2122:38:372015

Lastchange:TueJul2122:38:332015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

nginx(lsb:nginx):Startednod2.test.com

#webservice组资源没有转换到nod1.test.com，是因为没有定义组对节点的倾向性

如果此时把nod2.test.com节点上的corosync服务停止，webservice这个组中的资源能够转换到nod1.test.com节点上吗？如下测试：

[root@nod2~]#servicecorosyncstop

SignalingCorosyncClusterEngine(corosync)toterminate:[确定]

Waitingforcorosyncservicestounload:.[确定]

Youhavenewmailin/var/spool/mail/root

在nod1.test.com节在上查看当前集群的状态：

[root@nod1~]#crmstatus

Lastupdated:TueJul2122:43:272015

Lastchange:TueJul2122:38:332015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionWITHOUTquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.com]

OFFLINE:[nod2.test.com]

从上边的输出信息可知资源并没有转移过来，为什么？仔细看上边的“Current DC: nod1.test.com - partition WITHOUT quorum”表示当前分区没有法定的票数，所以此节点不会正常工作，资源当然不会转移过来。那如何解决这个问题，方案不止一个，一是可以增加一个ping node节点，二是可以增加一个仲裁磁盘，三是让集群中的节点数成奇数个，四是直接忽略当集群没有法定票数时直接忽略，第四种是最简单的方式，操作如下：

[root@nod2~]#servicecorosyncstart#先把nod2.test.com的corosync服务启动

StartingCorosyncClusterEngine(corosync):[确定]

[root@nod1~]#crm

crm(live)#status

Lastupdated:TueJul2122:50:082015

Lastchange:TueJul2122:38:332015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

nginx(lsb:nginx):Startednod1.test.com

crm(live)configure#propertyno#敲击两下tab键后列出以no开头的可配置参数

no-quorum-policy=node-health-green=node-health-strategy=

node-action-limit=node-health-red=node-health-yellow=

crm(live)configure#propertyno-quorum-policy=#输入“no-quorum-policy=”再敲击两下tab键后列出一些帮助信息

no-quorum-policy(enum,[stop]):Whattodowhentheclusterdoesnothavequorum

WhattodowhentheclusterdoesnothavequorumAllowedvalues:stop,freeze,ignore,suicide

crm(live)configure#property no-quorum-policy=ignore#设置其值为ignore

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#show#显示当前的配置信息

nodenod1.test.com

attributesstandby=off

nodenod2.test.com

primitivenginxlsb:nginx

primitivewebipIPaddr

paramsip=192.168.0.100

groupwebservicewebipnginx

propertycib-bootstrap-options:

dc-version=1.1.11-97629de

cluster-infrastructure=classicopenais(withplugin)

expected-quorum-votes=2

stonith-enabled=false

no-quorum-policy=ignore

last-lrm-refresh=1436887216

crm(live)#status

Lastupdated:TueJul2122:54:002015

Lastchange:TueJul2122:51:102015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

nginx(lsb:nginx):Startednod1.test.com

#当前资源已运行在nod1.test.com上

在nod1.test.com上停止corosync服务，再观察资源是否能转移到nod2.test.com上：

[root@nod1~]#servicecorosyncstop

SignalingCorosyncClusterEngine(corosync)toterminate:[OK]

Waitingforcorosyncservicestounload:.[OK]

[root@nod2~]#crm#在nod2.test.com上进行crm管理接口

crm(live)#status

Lastupdated:TueJul2122:56:522015

Lastchange:TueJul2122:52:252015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionWITHOUTquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod2.test.com]

OFFLINE:[nod1.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

nginx(lsb:nginx):Startednod2.test.com

#资源已成功转移到nod2.test.com上，所以在两个节点的高可用的环境，要设置“no-quorum-policy=ignore”，忽略节点的得到的法定票数不大于一半时的情况

如果是我们把在nod2.test.com上的nginx进程杀掉，集群资源会被转移到nod1.test.com上吗？如下测试：

[root@nod1~]#servicecorosyncstart#先把nod1.test.com上的corosync服务启动

StartingCorosyncClusterEngine(corosync):[OK]

[root@nod1~]#crmstatus

Lastupdated:WedJul2222:22:562015

Lastchange:WedJul2222:19:552015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

nginx(lsb:nginx):Startednod2.test.com

再切换到nod2.test.com节点上杀掉nginx进程：

[root@nod2~]#pgrepnginx

1798

1799

[root@nod2~]#killallnginx#杀掉nginx进程

[root@nod2~]#pgrepnginx#检验nginx进程是否被杀掉，没有任何信息输出表示nginx进程已不存在

[root@nod2~]#crmstatus

Lastupdated:WedJul2222:26:092015

Lastchange:WedJul2222:19:552015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

nginx(lsb:nginx):Startednod2.test.com

上边查看集群状态时发现资源还是在nod2.test.com节点上，这在实际的生产环境中是不允许的，所以需要让集群能监控我们定义的资源，如果发现某资源不存在了，自己会尝试启动这一资源，如果尝试启动不成功，则会转移资源，下 边就来说说如何定义监控资源。

[root@nod2~]#servicenginxstart#先把上边杀掉的nginx启动起来

正在启动nginx：[确定]

要定义资源的监控时也是在用全局定义命令primitive定义资源时一同定义，接下来我们先把之前定义的资源删掉后重新定义：

[root@nod1~]#crm

crm(live)#resource

crm(live)resource#show

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Started

nginx(lsb:nginx):Started

#进入资源管理命令可查看当前集群配置资源的情况，上边表示两个资源都是处理started状态

crm(live)resource#stop webservice#停掉webservice这个组中的所有资源，要删除资源，必须让资源处理stoppped状态

crm(live)resource#show

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Stopped

nginx(lsb:nginx):Stopped

crm(live)resource#cd..

crm(live)#configure

crm(live)configure#edit#输入edit命令回车后会调用vi编辑器直接去编辑资源定义的配置文件，如下所示

nodenod1.test.com

attributes standby=on

nodenod2.test.com

primitivenginxlsb:nginx#这是定义的资源，需要删除

primitivewebipIPaddr#这是定义的资源，需要删除

paramsip=192.168.0.100

groupwebservicewebipnginx#这是定义的资源，需要删除

metatarget-role=Stopped

propertycib-bootstrap-options:

dc-version=1.1.11-97629de

cluster-infrastructure=classicopenais(withplugin)

expected-quorum-votes=2

stonith-enabled=false

no-quorum-policy=ignore

last-lrm-refresh=1436887216

#vim:setsyntax=pcmk

在上边打开的编辑窗口中删除我们自己定义的资源，再保存退出，最后保留的内容如下：

nodenod1.test.com

attributesstandby=on

nodenod2.test.com

propertycib-bootstrap-options:

dc-version=1.1.11-97629de

cluster-infrastructure=classicopenais(withplugin)

expected-quorum-votes=2

stonith-enabled=false

no-quorum-policy=ignore

last-lrm-refresh=1436887216

#vim:setsyntax=pcmk

crm(live)configure#verify#检查语法

crm(live)configure#commit#提交配置

crm(live)resource#cd#回到根目录

crm(live)#status#查看集群状态

Lastupdated:WedJul2221:33:072015

Lastchange:WedJul2221:31:452015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

0Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

从状态信息输出发现我们定义的资源已被删除了，现在开始重新定义带监控的资源：

crm(live)configure#primitive webip ocf:IPaddr params ip=192.168.0.100 op monitor timeout=20s interval=60s

crm(live)configure#primitive webserver lsb:nginx op monitor timeout=20s interval=60s

crm(live)configure#group webservice webip webserver

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd

crm(live)#status

Lastupdated:WedJul2222:29:592015

Lastchange:WedJul2222:28:012015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

webserver(lsb:nginx):Startednod1.test.com

这样带监控的资源就定义好了，上边在定义监控是的那些参数的意义可以在使用类似的命令查看“crm(live)ra# meta ocf:IPaddr”。现在我们再到nod1.test.com节点上把nginx杀掉，观察会发生什么现象：

[root@nod1~]#pgrepnginx

3056

3063

[root@nod1~]#killallnginx

[root@nod1~]#pgrepnginx

[root@nod1~]#pgrepnginx

[root@nod1~]#pgrepnginx#等了几十秒后，nginx又被重新启动了

3337

3338

再看一下集群的状态信息，如下：

[root@nod1~]#crmstatus

Lastupdated:WedJul2222:33:292015

Lastchange:WedJul2222:28:012015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

webserver(lsb:nginx):Startednod1.test.com

Failedactions:

webserver\_monitor\_60000onnod1.test.com'notrunning'(7):call=23,status=complete,last-rc-change='WedJul2222:32:022015',queued=0ms,exec=0ms#这里报告了webserver这个资源没有运行

如果我们kill掉nginx后，让nginx无法启动，又是怎样一个情况呢，我们这样来测试，把nginx杀掉后，立刻去修改nginx的配置文件，随便增加一些行，让nginx的配置文件无法通过语法检测，这样自然nginx就无法启动了，说做就做：

[root@nod1~]#killallnginx

[root@nod1~]#echotest>>/etc/nginx/nginx.conf

[root@nod1~]#nginx-t

nginx:[emerg]unexpectedendoffile,expecting;or}in/etc/nginx/nginx.conf:44

nginx:configurationfile/etc/nginx/nginx.conftestfailed

[root@nod1~]#crmstatus

Lastupdated:WedJul2222:37:422015

Lastchange:WedJul2222:28:012015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com#看这里资源被转移到nod2.test.com了

webserver(lsb:nginx):Startednod2.test.com

Failedactions:

webserver\_start\_0onnod1.test.com'unknownerror'(1):call=30,status=complete,last-rc-change='WedJul2222:37:022015',queued=0ms,exec=70ms#这里也报告一个未知的错误

上边的两个测试证明，集群对资源能实现监控，并在资源不可用时能测试重新启动资源，如果不成功则转移资源。测试完了不要忘记恢复nod1.test.com节点上的nginx配置。

4、资源约束

资源约束定义我们期望资源运行在某一个节点上，或期望某些资源会在一起，而不使用组的方式定义。

接着上边的实验，我们希望webip与webserver这两个资源始终是在一起的，而不用定义webservice这个group来实现，那做如下操作：

[root@nod1~]#crm

crm(live)#status

Lastupdated:WedJul2222:46:262015

Lastchange:WedJul2222:28:012015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

Failedactions:

webserver\_start\_0onnod1.test.com'unknownerror'(1):call=30,status=complete,last-rc-change='WedJul2222:37:022015',queued=0ms,exec=70ms

先把上边资源的报错信息清理掉：

[root@nod1~]#crm

crm(live)#resource

crm(live)resource#cleanup webserver#清理资源的一些状态信息

Cleaningupwebserveronnod1.test.com

Cleaningupwebserveronnod2.test.com

Waitingfor2repliesfromtheCRMd..OK

crm(live)resource#cd

crm(live)#status

Lastupdated:WedJul2222:47:532015

Lastchange:WedJul2222:47:472015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

接下来删除webservice这个组资源：

[root@nod1~]#crm

crm(live)#resource

crm(live)resource#status

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Started

webserver(lsb:nginx):Started

crm(live)configure#deletewebservice#删除组资源

crm(live)configure#verify

crm(live)configure#commit

crm(live)#status

Lastupdated:WedJul2223:00:132015

Lastchange:WedJul2223:00:092015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod1.test.com#组被删除后，两个资源被集群平均分布在各节点上

webserver(lsb:nginx):Startednod2.test.com#webserver运行在nod2.test.com上

4.1、定义排列约束(colocation)

排列约束是定义让两个资源是否在一起：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#helpcolocation#查看colocation帮助信息

crm(live)configure#colocation webserver\_with\_webip inf: webserver webip#这里表示webserver资源与webip在一起的可能是正无穷的，即两资源一定要在一起

crm(live)configure#showxml#查看我们定义的约束

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd..

crm(live)#status

Lastupdated:WedJul2223:09:112015

Lastchange:WedJul2223:09:082015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod1.test.com#现在两个资源又都在nod1.test.com上运行了

webserver(lsb:nginx):Startednod1.test.com

4.2、定义顺序约束(order)

顺序约束表示资源的启动按照一定的顺序进行，而关闭则是一个相反的过程：

[root@nod1~]#crm

crm(live)configure#helporder#查看帮助

crm(live)configure#order webip\_before\_webserver mandatory: webip webserver#表示webip先于webserver启动，详细请看帮助信息

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#showxml#查看定义的详情

4.3、定义位置约束(location)

位置约束表示资源更倾向运行在哪个节点上。

[root@nod1~]#crm

crm(live)#status

Lastupdated:WedJul2223:20:082015

Lastchange:WedJul2223:15:392015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod1.test.com#此时资源是运行在nod1.test.com上的

webserver(lsb:nginx):Startednod1.test.com

定义位置约束让资源更倾向运行在nod2.test.com上：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#helplocation#查看帮助信息

crm(live)configure#location webip\_on\_nod2 webip inf: nod2.test.com #表示webip在nod2.test.com上的倾向性是正无穷的

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd

crm(live)#status

Lastupdated:WedJul2223:23:212015

Lastchange:WedJul2223:22:502015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

上边webip与webserver资源都已转移到了nod2.test.com，但webserver资源我们并没有定义它的位置约束，为什么它也转移到了nod2.test.com上了呢？因为我们定义过webip与webserver的排序约束，这两个资源在一起的分数(score)是inf（正无穷）的，所以webip在哪里，webserver就在哪里。

location的定义还有另外一种格式，如下：

[root@nod1~]#crm

crm(live)configure#deletewebip\_on\_nod2#先删除上边定义的location

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#location webip\_on\_nod1 webip rule inf: #uname eq nod1.test.com#表示webip运行在名称为nod1.test.com主机上的倾向性是正无穷的

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd

crm(live)#status

Lastupdated:WedJul2223:33:382015

Lastchange:WedJul2223:33:182015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod1.test.com

webserver(lsb:nginx):Startednod1.test.com

#上边的两个资源又转移到了nod1.test.com节点上。

接着再来定义一个location：

crm(live)configure#location webserver\_not\_on\_nod1 webserver rule-inf: #unameeqnod1.test.com #这里表示webserver资源不在nod1上的分数是负无穷

crm(live)configure#verify

crm(live)configure#commit

crm(live)configure#cd

crm(live)#status

Lastupdated:WedJul2223:41:252015

Lastchange:WedJul2223:41:192015

Stack:classicopenais(withplugin)

CurrentDC:nod2.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

2Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

webip与webserver从nod1.test.com上转移到了nod2.test上，为什么呢？虽然定义了webserver资源不在nod1上的分数是负无穷，但我们不是定义了webip对nod1.test.com的倾向性是正无穷么，这个“inf+(-inf)”等于什么呢？答案是“-inf”，所以资源绝对不会在nod1.test.com上。

5、案例

一个高可用集群一般会包含三类资源，一是虚拟ip，二是服务，三是共享存储，下边我们再把共享存储加上来一起说说高可用的实现，因有新的资源加入，在资源的约束上又会有所不同，所以先把上边的定义的ip资源、服务资源删除，重新来说说有三种资源的高可用性，怎样删除集群中的资源这里就不再赘述了，可以看看前边的操作。

资源删除后就是一个干净的集群，如下所示：

[root@nod1~]#crm

crm(live)#status

Lastupdated:FriJul2420:58:492015

Lastchange:FriJul2420:58:322015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

0Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

接下来准备共享存储，这里以nod0.test.com这个节点提供NFS共享存储为例：

[root@nod0~]#yum-yinstallnfs-utils

[root@nod0~]#vim/etc/exports

/web/htdocs192.168.0.0/24(rw)

[root@nod0~]#mkdir-pv/web/htdocs

[root@nod0~]#vim/web/htdocs/index.[html](http://www.2cto.com/kf/qianduan/css/" \t "_blank)

[root@nod0~]#servicerpcbindstart

Startingrpcbind:[OK]

[root@nod0~]#servicenfsstart

StartingNFSservices:[OK]

StartingNFSmountd:[OK]

StartingNFSdaemon:[OK]

StartingRPCidmapd:[OK]

[root@nod0~]#vim/etc/exports

/web/htdocs192.168.0.0/24(rw,no\_root\_squash)

[root@nod0~]#mkdir-pv/web/htdocs/

[root@nod0~]#echoNFSnode>/web/htdocs/index.html#这是提供的测试页面

[root@nod2~]#mount-tnfs192.168.0.200:/web/htdocs/usr/share/nginxhtml/#nfs第一次挂载很慢，所以先手动挂载一次

再在nod2.test.com上启动nginx，测试一下能否访问nod2.test.com节点上的ip：192.168.0.202测试页面：

[root@nod2~]#servicenginxstart

正在启动nginx：[确定]



测试通过了要关闭nginx服务，卸载共享存储：

[root@nod2~]#umount/usr/share/nginx/html/

Youhavenewmailin/var/spool/mail/root

[root@nod2~]#servicenginxstop

停止nginx:[确定]

接下来就去定义高可用集群的资源了：

[root@nod1~]#crm

crm(live)#configure

crm(live)configure#primitive webip ocf:IPaddr paramsip=192.168.0.100 op monitor timeout=10s interval=30s

crm(live)configure#primitive webserver lsb:nginx op monitor timeout=10s interval=30s

crm(live)configure#primitive webstore ocf:Filesystem params device=192.168.0.200:/web/ht docs directory=/usr/share/nginx/html fstype=nfs op monitor timeout=30s interval=60s

crm(live)configure#verify

WARNING:webip:specifiedtimeout10sformonitorissmallerthantheadvised20s

WARNING:webserver:specifiedtimeout10sformonitorissmallerthantheadvised15

WARNING:webstore:defaulttimeout20sforstartissmallerthantheadvised60#表示nfs共享存储要定义start时的超时时间，默认是20s，但建议是60s

WARNING:webstore:defaulttimeout20sforstopissmallerthantheadvised60#表示nfs共享存储要定义stop时的超时时间，默认是20s，但建议是60s

WARNING:webstore:specifiedtimeout30sformonitorissmallerthantheadvised40

在校验时报了如下错误，大概是说在设置资源时监控的时间值不对，按照提示做修改就是

crm(live)configure#cd..

Therearechangespending.Doyouwanttocommitthem(y/n)?n#这里不要提交，当然也可以用edit命令调用vi编辑器去编辑xml文件

crm(live)#configure#进入配置模式重新定义资源

crm(live)configure#primitive webip ocf:IPaddr paramsip=192.168.0.222 op monitor timeout=20s interval=30s

crm(live)configure#verify

crm(live)configure#primitive webserver lsb:nginx op monitor timeout=15s interval=30s

crm(live)configure#verify

crm(live)configure#primitive webstore ocf:Filesystem paramsdevice=192.168.0.200:/web/htdocs directory=/usr/share/nginx/html fstype=nfs op monitor timeout=30s interval=60s op start timeout=60s op stop timeout=60s

crm(live)configure#verify

WARNING:webstore:specifiedtimeout30sformonitorissmallerthantheadvised40#这里还有一个值设置不对

crm(live)configure#edit#直接进入编辑模式进行修改，修改后就是下边这样

nodenod1.test.com

attributesstandby=off

nodenod2.test.com

attributesstandby=off

primitivewebipIPaddr

paramsip=192.168.0.222

opmonitortimeout=20sinterval=30s

primitivewebserverlsb:nginx

opmonitortimeout=15sinterval=30s

primitivewebstoreFilesystem

paramsdevice=192.168.0.200:/web/htdocsdirectory=/usr/share/nginx/htmlfstype=nfs

opmonitortimeout=40sinterval=60s

opstarttimeout=60sinterval=0

opstoptimeout=60sinterval=0

propertycib-bootstrap-options:

dc-version=1.1.11-97629de

cluster-infrastructure=classicopenais(withplugin)

expected-quorum-votes=2

stonith-enabled=false

no-quorum-policy=ignore

last-lrm-refresh=1437576541

#vim:setsyntax=pcmk

#记得保存退出

crm(live)configure#verify#现在校验就没有错误了

crm(live)configure#commit#提交配置

接下来定义三个资源的一些约束，思考一下，有VIP、有服务、有共享存储的一个高可用集群需要怎样一些约束关系呢？第一：集群在正常工作时三个资源应该是运行在一个节点上的，而三个资源间又有一些小的约束关系，VIP要与服务(nginx)要在一起，服务(nginx)要与共享存储在一起，这些可以用排列约束(colocation)，也可以用组(group)的方式实现；第二：各个资源的启动次序，VIP应该是先于服务启动，共享存储得先挂载上才启动服务吧；接下来就去定义这些：

crm(live)configure#group webservice webip webstore webserver#定义一个组包含三个资源

crm(live)configure#order webip\_before\_webstore\_before\_webserver inf: webip webstore webserver#定义顺序约束，定义这三个资源的启动顺序一定(inf)是先启动webip，接着是webstore，最后是webserver，而关闭则是相反的过程

crm(live)configure#verify

crm(live)configure#showxml#查看配置的xml文件

如果这三个资源对集群节点没有倾向性那就直接可以commit了，特别是在当今虚拟化泛滥的年代，高可用一样部署xem、kvm、openstack这样的虚拟环境下，集群资源对虚拟资源的倾向性表现得不明显了。

crm(live)configure#commit

crm(live)configure#cd..

crm(live)#status

Lastupdated:FriJul2422:25:062015

Lastchange:FriJul2422:25:022015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

3Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

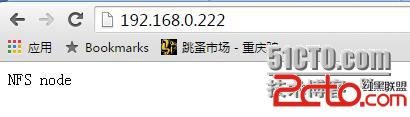
webip(ocf::heartbeat:IPaddr):Startednod1.test.com

webstore(ocf::heartbeat:Filesystem):Startednod1.test.com

webserver(lsb:nginx):Startednod1.test.com

#从上边的输出信息可知资源运行在了nod1.test.com这个节点上了

现在访问http服务测试一下，访问的是我们定义的VIP，如下：



现在测试一下集群资源是否能正常转移，把nod1.test.com节点置于standby状态，看资源是否能转移到nod2.test.com节点上：

[root@nod1~]#crmnodestandby

[root@nod1~]#crmstatus

Lastupdated:FriJul2422:28:032015

Lastchange:FriJul2422:27:532015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

3Resourcesconfigured

Nodenod1.test.com:standby

Online:[nod2.test.com]

ResourceGroup:webservice

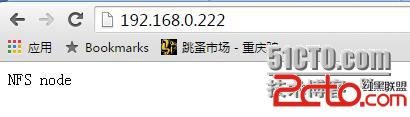
webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webstore(ocf::heartbeat:Filesystem):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

#上边输出信息中看到资源都转移到了nod2.test.com上了

再去刷新一下访问页面：如下依然是有效的，如下：



经测试资源能正常切换，接下来还要测试定义的资源监控是否生效，可以去尝试停止nginx服务或umount共享存储，等监控资源的时间到时集群就会尝试重新启动服务或挂载共享存储：

[root@nod2~]#servicenginxstop

停止nginx：[确定]

过一会后，集群就监控到异常了，如下：

[root@nod1~]#crmstatus

Lastupdated:FriJul2422:42:052015

Lastchange:FriJul2422:36:002015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

3Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webstore(ocf::heartbeat:Filesystem):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

Failedactions:

webserver\_monitor\_30000onnod2.test.com'notrunning'(7):call=41,status=complete,last-rc-change='FriJul2422:37:252015',queued=0ms,exec=0ms

再来测试一下共享存储是否能监控并恢复，如下：

[root@nod2~]#umount/usr/share/nginx/html/

现在去访问web，就是打开nginx的默认页面了，如下：

当检测时间一到，集群就会发现异常，并尝试恢复，如下：

[root@nod1~]#crmstatus

Lastupdated:FriJul2422:44:032015

Lastchange:FriJul2422:36:002015

Stack:classicopenais(withplugin)

CurrentDC:nod1.test.com-partitionwithquorum

Version:1.1.11-97629de

2Nodesconfigured,2expectedvotes

3Resourcesconfigured

Online:[nod1.test.comnod2.test.com]

ResourceGroup:webservice

webip(ocf::heartbeat:IPaddr):Startednod2.test.com

webstore(ocf::heartbeat:Filesystem):Startednod2.test.com

webserver(lsb:nginx):Startednod2.test.com

Failedactions:

webserver\_monitor\_30000onnod2.test.com'notrunning'(7):call=41,status=complete,last-rc-change='FriJul2422:37:252015',queued=0ms,exec=0ms

webstore\_monitor\_60000onnod2.test.com'notrunning'(7):call=39,status=complete,last-rc-change='FriJul2422:42:552015',queued=0ms,exec=0ms

现在访问web页面又恢复了，如下：



至此，corosync+pacemaker+crmsh的高可用的实现已演示完毕。

6、总结

作为一个[Linux](http://www.2cto.com/os/linux/)运维工程师，掌握高可用架构是必不可少的技能，刚学习高可用时感觉那些理论知识就不好理解，可现在把上边的实验做完后，感觉对高可用架构有了新的认识，并对上一博客中提到的理论知识也有了新的认识。

在利用corosync+pacemaker且是两个节点实现高可用时，需要注意的是要设置全局属性把stonith设备关闭，忽略法定票数不大于一半的机制，即：

crm(live)configure#propertyno-quorum-policy=ignore

crm(live)configure#propertystonith-enabled=false

一，crm有两种工作方式

1，批处理模式,就是在shell命令行中直接输入命令  
2，交互式模式（crm(live)#）进入到crmsh中交互执行

二，命令详解

1，一级子命令

[root@node1 corosync]# crm    
crm(live)# help    
This is crm shell, a Pacemaker command line interface.    
Available commands:    
    cib              manage shadow CIBs //cib沙盒    
    resource         resources management //所有的资源都在这个子命令后定义    
    configure        CRM cluster configuration //编辑集群配置信息    
    node             nodes management //集群节点管理子命令    
    options          user preferences //用户优先级    
    history          CRM cluster history //命令历史记录    
    site             Geo-cluster support    
    ra               resource agents information center //资源代理子命令（所有与资源代理相关的程都在此命令之下）    
    status           show cluster status //显示当前集群的状态信息    
    help,?           show help (help topics for list of topics)//查看当前区域可能的命令    
    end,cd,up        go back one level //返回第一级crm(live)    
    quit,bye,exit    exit the program //退出crm（live）交互模式

2,resource资源状态控制子命令

crm(live)resource# help    
vailable commands:    
        status           show status of resources //显示资源状态信息    
        start            start a resource //启动一个资源    
        stop             stop a resource //停止一个资源    
        restart          restart a resource //重启一个资源    
        promote          promote a master-slave resource //提升一个主从资源    
        demote           demote a master-slave resource //降级一个主从资源    
        manage           put a resource into managed mode  //将一个资源加入到管理模式下    
        unmanage         put a resource into unmanaged mode  //从管理模式下去除一个资源    
        migrate          migrate a resource to another node //将资源迁移到另一个节点上    
        unmigrate        unmigrate a resource to another node    
        param            manage a parameter of a resource //管理资源的参数    
        secret           manage sensitive parameters //管理敏感参数    
        meta             manage a meta attribute //管理源属性    
        utilization      manage a utilization attribute    
        failcount        manage failcounts //管理失效计数器    
        cleanup          cleanup resource status //清理资源状态    
        refresh          refresh CIB from the LRM status //从LRM（LRM本地资源管理）更新CIB（集群信息库），在    
        reprobe          probe for resources not started by the CRM //探测在CRM中没有启动的资源    
        trace            start RA tracing //启用资源代理（RA）追踪    
        untrace          stop RA tracing //禁用资源代理（RA）追踪    
        help             show help (help topics for list of topics) //显示帮助   
        end              go back one level //返回一级（crm(live)#）    
        quit             exit the program //退出交互式程序

3,configure资源定义子命令

crm(live)configure# help    
Available commands:    
        node             define a cluster node //定义一个集群节点    
        primitive        define a resource //定义资源    
        monitor          add monitor operation to a primitive //对一个资源添加监控选项（如超时时间，启动失败后的操作）    
        group            define a group //定义一个组类型（将多个资源整合在一起）    
        clone            define a clone //定义一个克隆类型（可以设置总的克隆数，每一个节点上可以运行几个克隆）    
        ms               define a master-slave resource //定义一个主从类型（集群内的节点只能有一个运行主资源，其它从的做备用）    
        rsc\_template     define a resource template //定义一个资源模板    
        location         a location preference //定义位置约束优先级（默认运行于那一个节点（如果位置约束的值相同，默认倾向性那一个高，就在那一个节点上运行））    
        colocation       colocate resources //排列约束资源（多个资源在一起的可能性）   
        order            order resources //资源的启动的先后顺序    
        rsc\_ticket       resources ticket dependency    
        property         set a cluster property //设置集群属性    
        rsc\_defaults     set resource defaults //设置资源默认属性（粘性）    
        fencing\_topology node fencing order //隔离节点顺序    
        role             define role access rights //定义角色的访问权限    
        user             define user access rights //定义用用户访问权限    
        op\_defaults      set resource operations defaults //设置资源默认选项    
        schema           set or display current CIB RNG schema    
        show             display CIB objects //显示集群信息库对    
        edit             edit CIB objects //编辑集群信息库对象（vim模式下编辑）    
        filter           filter CIB objects //过滤CIB对象    
        delete           delete CIB objects //删除CIB对象    
        default-timeouts set timeouts for operations to minimums from the meta-data    
        rename           rename a CIB object //重命名CIB对象    
        modgroup         modify group //改变资源组    
        refresh          refresh from CIB //重新读取CIB信息    
        erase            erase the CIB //清除CIB信息    
        ptest            show cluster actions if changes were committed    
        rsctest          test resources as currently configured    
        cib              CIB shadow management    
        cibstatus        CIB status management and editing    //cib状态管理和编辑    
        template         edit and import a configuration from a template   //编辑或导入配置模板    
        commit           commit the changes to the CIB //将更改后的信息提交写入CIB    
        verify           verify the CIB with crm\_verify //CIB语法验证    
        upgrade          upgrade the CIB to version 1.0 //升级CIB到1.0    
        save             save the CIB to a file //将当前CIB导出到一个文件中（导出的文件存于切换crm 之前的目录）    
        load             import the CIB from a file //从文件内容载入CIB    
        graph            generate a directed graph    
        xml              raw xml    
        help             show help (help topics for list of topics) //显示帮助信息    
        end              go back one level //回到第一级(crm(live)#)    
        quit             exit the program  //退出crm交互模式

4,node节点管理子命令

crm(live)# node    
crm(live)node# help    
Node management and status commands.    
Available commands:    
    status           show nodes status as XML //以xml格式显示节点状态信息    
    show             show node //命令行格式显示节点状态信息    
    standby          put node into standby //模拟指定节点离线（standby在后面必须的FQDN）    
    online           set node online // 节点重新上线    
    maintenance      put node into maintenance mode  //将一个节点状态改为maintenance    
    ready            put node into ready mode   //将一个节点状态改为ready    
    fence            fence node //隔离节点    
    clearstate       Clear node state //清理节点状态信息    
    delete           delete node //删除 一个节点    
    attribute        manage attributes    
    utilization      manage utilization attributes    
    status-attr      manage status attributes    
    help             show help (help topics for list of topics)    
    end              go back one level  //回到上一次    
    quit             exit the program    //退出

5,ra资源代理子命令

crm(live)# ra    
crm(live)ra# help    
Available commands:    
        classes          list classes and providers //为资源代理分类    
        list             list RA for a class (and provider)//显示一个类别中的提供的资源    
        meta             show meta data for a RA //显示一个资源代理序的可用参数（如meta ocf:heartbeat:IPaddr2）    
        providers        show providers for a RA and a class    
        help             show help (help topics for list of topics)    
        end              go back one level    
        quit             exit the program

三，实例  
1，系统资源查看

crm(live)ra# classes   //系统所使用的资源代理    
lsb    
ocf / heartbeat pacemaker    
service    
stonith

2，默认系统资源列表

crm(live)ra# list ocf   //列表默认系统资源    
CTDB               ClusterMon         Delay              Dummy              Filesystem         HealthCPU    
IPaddr             IPaddr2            IPsrcaddr          HealthCPU          HealthSMART        HealthSMART    
LVM                MailTo             Route              SendArp            Squid    
Stateful           SysInfo            SystemHealth       VirtualDomain      Xinetd    
apache             conntrackd         controld           db2                dhcpd    
ethmonitor         exportfs           iSCSILogicalUnit   mysql              named    
nfsnotify          nfsserver          pgsql              ping               pingd    
postfix            remote             rsyncd             symlink            [tomcat](http://www.111cn.net/tags.php/tomcat/)

3，查看外加的系统资源

crm(live)ra# list lsb  //列表外加的系统资源    
NetworkManager     abrt-ccpp          abrt-oo[ps](http://www.111cn.net/fw/photo.html)          abrtd              acpid    
atd                auditd             autofs             blk-availability   certmonger    
corosync           corosync-notifyd   cpuspeed           crond              cups    
dnsmasq            firstboot          haldaemon          halt               hsqldb    
ip6tables          iptables           irqbalance         jexec              kdump    
killall            lvm2-lvmetad       lvm2-monitor       mdmonitor          messagebus    
mysqld             netconsole         netfs              network            nfs    
nfslock            nginx              nmb                ntpd               ntpdate    
oddjobd            openfire           pacemaker          php-fpm            portreserve    
postfix            psacct             quota\_nld          rdisc              redis    
restorecond        rngd               rpcbind            rpcgssd            rpcidmapd    
rpcsvcgssd         rsyslog            sandbox            saslauthd          single    
slapd              smartd             smb                snmpd              snmptrapd    
spice-vdagentd     sshd               sssd               sysstat            udev-post    
vsftpd             wdaemon            winbind            wpa\_supplicant     ypbind

4，查看系统资源的参数

crm(live)ra# meta ocf:heartbeat:IPaddr   //查看系统资源IPaddr的参数    
Manages virtual IPv4 and IPv6 addresses (Linux specific version) (ocf:heartbeat:IPaddr)    
    
This Linux-specific resource manages IP alias IP addresses.    
It can add an IP alias, or remove one.    
In addition, it can implement Cluster Alias IP functionality    
if invoked as a clone resource.    
    
If used as a clone, you should explicitly set clone-node-max >= 2,    
and/or clone-max < number of nodes. In case of node failure,    
clone instances need to be re-allocated on surviving nodes.    
This would not be possible if there is already an instance on those nodes,    
and clone-node-max=1 (which is the default).    
    
Parameters (\*: required, []: default):    
    
ip\* (string): IPv4 or IPv6 address    
    The IPv4 (dotted quad notation) or IPv6 address (colon hexadecimal notation)    
    example IPv4 "192.168.1.1".    
    example IPv6 "2001:db8:DC28:0:0:FC57:D4C8:1FFF".    
。。。。。。。。。。。。。。。。。。。省略。。。。。。。。。。。。。。。。。。。。。。。。

5，属性修改

crm(live)configure# property stonith-enabled=false //禁用stonith-enable    
    
crm(live)configure# property no-quorum-policy=ignore //忽略投票规则

6，定义一资源所使用的命令

/\*\*   
 \* primitive:定义一资源所使用的命令   
 \* webip:为资源起一个名字   
 \* ocf:heartbeat;IPaddr：所使用资源代理的类别，由谁提供的那一个代理程序   
 \* op monitor 对webip做监控   
 \* interval:间隔时间   
 \* timeout：超时时间   
 \* on-fail：失败自起   
 \*/    
crm(live)configure# primitive webip ocf:heartbeat:IPaddr params ip=192.168.10.130 op monitor interval=30s timeout=20s on-fail=restart    
    
crm(live)configure# primitive nginx\_res lsb:nginx //那些在/etc/init.d/\*的脚本就是属于lsb的    
params后面的参数，可以通过meta命令来查找，不同的系统资源参数是不一样的。

7，定义排列约束

/\*\*   
 \* 定义排列约束   
 \* colocation：排列约束命令   
 \* nginx\_web : 约束名   
 \* inf:#(可能性，inf表示永久在一起，也可以是数值)   
 \* webip nginx\_res：#资源名称   
 \*/    
crm(live)configure# colocation nginx\_web inf: nginx\_res webip

8，定义资源启动顺序

/\*\*   
 \* 定义资源启动顺序   
 \* order : 顺序约束的命令   
 \* nginx\_after\_ip : 约束ID   
 \* mandatory: #指定级别（此处有三种级别：mandatory：强制， Optional：可选，Serialize：序列化）   
 \* webip nginx\_res：#资源名称,这里书写的先后顺序相当重要   
 \*/    
crm(live)configure# order nginx\_after\_ip mandatory: webip nginx\_res

9，定义节点权重

/\*\*   
 \* 定义节点权重   
 \* location：节点权重命令   
 \* webip\_and\_webnfs\_and\_webserver：约束名称   
 \* webip 500: node1：webip这个资源的node1节点的权重是500   
 \*/    
crm(live)configure# location webip\_and\_webnfs\_and\_webserver webip 500: node1

10，定义默认资源属性

/\*\*   
 \* 定义默认资源属性   
 \* rsc\_defaults：默认资源   
 \* 这样定义代表集群中每一个资源的默认权重，只有当资源服务不在当前节点时，权重才会生效，比如，这里我定义了三个资源webip、webnfs、webserver，对每一个资源的权重为100，那么加在一起就变成了300,之前己经定义node1的位置约束的值为500，当node1宕机后，重新上线，这样就切换到node1上了。   
 \*/    
crm(live)configure# rsc\_defaults resource-stickiness=100

11，删除资源或者节点

crm(live)configure# delete nginx\_res  //删除一个资源，约束，排序，组等    
crm(live)node# delete node1  //删除一个节

12，资源其他

crm(live)configure# verify //审核    
crm(live)configure# commit //提交    
crm(live)configure# show //显示

13，节点挂起和在线

crm(live)node# standby node1   //节点挂起    
crm(live)node# online node1    //节点上线

特别注意：

使用了crmsh，就不在需要安装heartbeat，之前的版本中都需要安装heartbeat以利用其crm进行资源管理。网上大量的教程中都是基于pacemaker1.1.8以前的版本的。

1.配置主机名

节点名称很关键，集群每个节的名称都得能互相解析。/etc/hosts中的主机名配置结果必须跟”uname -n”的解析的结果一致。

node1/node2都添加：

01.node1示例：

02.# vim /etc/hosts

03.172.16.1.11 node1.hoo.com node1

04.172.16.1.12 node2.hoo.com node2

05.# ifconfig eth0 172.16.1.11  up

06.# hostname node1.hoo.com

07.# uname -n

08.# vim /etc/sysconfig/network  //重启才生效，source也不生效

09.hostname=node1.hoo.com

2.时间同步

node1,node2都配置

1.# crontab -e

2.\*/5 \* \* \* \* /usr/sbin/ntpdate  172.16.0.1 &> /dev/null

3.配置node之间SSH互信

1.node1:

2.#ssh-keygen  -t rsa -f ~/.ssh/id\_rsa  -P ''

3.# ssh-copy-id -i .ssh/id\_rsa.pub root@node2.hoo.com

4.node1:

5.#ssh-keygen  -t rsa -f ~/.ssh/id\_rsa  -P ''

6.# ssh-copy-id -i .ssh/id\_rsa.pub root@node1.hoo.com

7.node1：测试

8.#  ssh node2.hoo.com ‘date’;date

4.安装httpd

node1,node2

1.# yum install httpd -y

2.# ssh node2 "yum install httpd -y "

3.# ss -tunl | grep 80

4.# ssh node2 "ss -tunl | grep 80"

node1,node2均关闭httpd的自启动，httpd由Pacemaker管理

1.# chkconfig httpd off

2.# ckconfig --list httpd  //检查

node1,node2提供测试页

1.[root@node1 ~]# echo "node1.hoo.com" >/var/www/html/index.html

2.[root@node2 ~]# echo "node2.hoo.com" >/var/www/html/index.html

5.安装corosync和pacemaker

所有节点均安装

1.# yum install corosync pacemaker -y

2.# rpm -ql corosync

3.//配置目录：/etc/corosync

4.//配置模板：/etc/corosync/corosync.conf.example

5.//服务器脚本：/etc/rc.d/init.d/corosync

6.安装资源管理器客户端命令接口工具crmsh

从pacemaker 1.1.8开始，crmsh发展成了一个独立项目，叫crmsh。pacemaker默认不提供命令接口工具，需要单独安装crmsh。

1.//crmsh依赖pssh

2.# yum -y install pssh-2.3.1-2.el6.x86\_64.rpm

3.# yum -y install crmsh-1.2.6-4.el6.x86\_64.rpm

7.配置主配置文件

01.[root@node1 ~]# cd /etc/corosync/

02.[root@node1 corosync]# cp corosync.conf.example corosync.conf

03.[root@node1 corosync]# vi corosync.conf

04.compatibility: whitetank   #表示是否兼容0.8之前的版本

05.totem {        #图腾，定义集群中各节点中通信机制以及参数

06.version: 2    #图腾的协议版本号，不修改

07.secauth: on   #安全认证功能是否启用，当使用aisexec时，会非常消耗CPU

08.threads: 0    #实现认证时的并行线程数，0表示默认配置

09.interface {   # 指定在哪个接口发心跳信息，子模块

10.ringnumber: 0   #冗余环号，节点有多个网卡是可定义，避免心跳信息成环。

11.bindnetaddr: 192.168.1.0   #绑定心跳网段

12.mcastaddr: 226.94.8.8      #心跳组播地址

13.mcastport: 5405            #心跳组播使用端口

14.ttl: 1                     #表示只向外播一次

15.}

16.}

17.logging {          # 跟日志相关

18.fileline: off       #指定要打印的行

19.to\_stderr: no       #是否发送到标准错误输出，即屏幕

20.to\_logfile: yes     #记录到日志文件

21.to\_syslog: no       #记录到系统日志syslog

22.logfile: /var/log/cluster/corosync.log  #日志文件路径

23.debug: off          #是否启动调试

24.timestamp: on       #是否打印时间戳

25.logger\_subsys {     #日志的子系统

26.subsys: AMF

27.debug: off

28.}

29.}

30.service {

31.ver:  0

32.name: pacemaker        #定义corosync启动时同时启动pacemaker

33.# use\_mgmtd: yes

34.}

35.

36.aisexec {     #表示启动ais的功能时，以哪个用户的身份运行

37.#这个块可不定义，corosync默认就是以root身份运行的

38.user: root

39.group: root

40.}

41.amf {     # 跟编程接口相关的

42.mode: disabled

43.}

8.生成认证key

用corosync-keygen生成key时，由于要使用/dev/random生成随机数，因此如果新装的系统操作不多，如果没有足够的熵，狂敲键盘即可，随意敲，敲够即可。（关于random使用键盘敲击产生随机数的原理可自行google）

实验演示没有足够的熵，这里将采用投机的方式，生产环境，切切不可。

1.//投机的认证方式

2.# mv /dev/random /dev/lala

3.# ln /dev/urandom /dev/random   //将随机数生成器链接至伪随机数生成器

4.# corosync-keygen    //生成密钥文件，用于节点通信，会生成authkey文件

5.# rm -rf /dev/random        //删除链接

6.# mv /dev/lala /dev/random     //还原随机数生成器

9.copy配置给node2

1.[root@node1 corosync]# scp authkey corosync.conf node2:/etc/corosync/

10.启动corosync

1.[root@node1 corosync]# service corosync start

2.[root@node1 corosync]# ssh node2  "service corosync start"

11.检查启动情况

1.//查看corosync引擎是否正常启动

2.[root@node1 corosync]# grep -e "Corosync Cluster Engine" -e "configuration file" /var/log/messages

3.//查看初始化节点通知是否正常发出

4.[root@node1 corosync]# grep  TOTEM  /var/log/messages

5.//检查启动过程中是否有错误产生

6.[root@node1 corosync]# grep ERROR: /var/log/messages | grep -v unpack\_resources

7.//查看pacemaker是否正常启动

8.[root@node1 corosync]# grep pcmk\_startup /var/log/messages

四、基于corosync实现web高可用资源管理

1.crmsh使用介绍

01.[root@node1 ~]# crm  //进入crmsh

02.crm(live)# help   #查看帮助

03.This is crm shell, a Pacemaker command line interface.

04.Available commands:

05.cib              manage shadow CIBs            #CIB管理模块

06.resource         resources management     #资源管理模块

07.configure        CRM cluster configuration  #CRM配置，包含资源粘性、资源类型、资源约束等

08.node             nodes management  #节点管理

09.options          user preferences  #用户偏好

10.history          CRM cluster history  #CRM 历史

11.site             Geo-cluster support  #地理集群支持

12.ra               resource agents information center #资源代理配置

13.status           show cluster status  #查看集群状态

14.help,?           show help (help topics for list of topics)  #查看帮助

15.end,cd,up        go back one level  #返回上一级

16.quit,bye,exit    exit the program  #退出

17.crm(live)# configure         #进入配置模式

18.crm(live)configure# show     #查看当前配置

19.crm(live)configure# verify   #检查当前配置语法

20.crm(live)configure# commit   #提交配置

21.crm(live)configure# cd

22.crm(live)# ra   //进入RA（资源代理配置）模式

23.crm(live)ra# help

24.This level contains commands which show various information about

25.the installed resource agents. It is available both at the top

26.level and at the `configure` level.

27.Available commands:

28.classes          list classes and providers    ##查看RA类型

29.list             list RA for a class (and provider)  ##查看指定类型（或提供商）的RA

30.meta,info        show meta data for a RA   ##查看RA详细信息

31.providers        show providers for a RA and a class  ##查看指定资源的提供商和类型

32.help,?           show help (help topics for list of topics)

33.end,cd,up        go back one level

34.quit,bye,exit    exit the program

35.crm(live)ra# cd

36.crm(live)# status   //查看集群状态

2.stonith参数的调整

禁用stonith功能，corosync默认是启用stonith功能的，没有stonith设备，若直接去配置资源的话，verif会报错，无法commit。

1.[root@node1 corosync]# crm configure    #进入crm命令行模式配置资源等

2.crm(live)configure# property    #切换到property目录下，可以用两次tab键进行补全和查看

3.usage: property [$id=&lt;set\_id&gt;] &lt;option&gt;=&lt;value&gt;    # property的用法和格式

4.crm(live)configure# property stonith-enabled=false   #禁用stonith-enabled

5.crm(live)configure# verify      #检查设置的属性是否正确

6.crm(live)configure# commit    #提交

7.crm(live)configure# show     #查看当前集群的所有配置信息

3.配置web集群

a).定义IP

1.crm(live)configure# primitive webip ocf:heartbeat:IPaddr params ip=172.16.1.100

2.crm(live)configure# verify

3.crm(live)configure# commit

4.crm(live)configure# cd

5.crm(live)# status

验证

1.[root@node1 ~]# ifconfig

2.发现：eth0:0有IP

转移资源

1.[root@node2 ~]# ssh node1 'service corosync stop'

2.[root@node2 ~]# ssh node1 'ifconfig'

3.[root@node2 ~]# ifconfig

4.发现：IP转移

b).配置httpd资源

1.crm(live)configure# primitive webserver lsb:httpd   //定义资源，资源类型为lsb

2.crm(live)configure# verify

3.crm(live)configure# show

4.crm(live)configure# commit

5.crm(live)# status

6.//资源运行于两个节点上，默认以均衡的方式进行工作,尽可能将不同的资源运行在不同的节点上;

4.资源约束定义

若想将多个资源运行在同一个节点上，则，做成组，或定义排列约束。

资源约束则用以指定在哪些群集节点上运行资源，以何种顺序装载资源，以及特定资源依赖于哪些其它资源。

pacemaker共给我们提供了三种资源约束方法：

1）Resource Location(资源位置）:定义资源可以、不可以或尽可能在哪些节点上运行； 2）Resource Collocation(资源排列）：排列约束定义集群资源可以、不可以在某个节点上同时运行； 3）Resource Order(资源顺序)：顺序约束定义集群资源在节点上启动的顺序；

a).定义组，将资源运行在同一个节点上

1.//定义资源组webservice

2.crm(live)configure# help group

3.crm(live)configure# group webservice webip sebserver

4.crm(live)configure# verify

5.crm(live)configure# commit

6.crm(live)configure# show

7.crm(live)configure#cd

8.crm(live)# status

[浏览器](http://www.it165.net/edu/ewl/)访问测试，成功！

测试：

01.//让node1离线，测试是否转移：

02.crm(live)# node

03.crm(live)node# help standby

04.crm(live)node# standby node1.hoo.com   //需要使用完整名称

05.crm(live)configure#cd

06.crm(live)# status

07.//转移到node2

08.让node1重新上线：

09.# crm node online node1.hoo.com

10.# crm status

b).定义排列约束

01.先删除组：

02.crm(live)configure#help delete

03.crm(live)configure# delete webservice

04.//无法删除，正在运行

05.crm(live)configure#cd

06.crm(live)# resource

07.crm(live)resource# help

08.crm(live)resource# help stop

09.crm(live)resource# stop sebservice

10.crm(live)resource# status webservice

11.crm(live)resource#cd

12.crm(live)configure#delete webservice

13.crm(live)configure#show

14.crm(live)configure#commit

15.crm(live)configure#cd

16.crm(live)#status

17.//负载均衡

18.//定义排列约束：

19.crm(live)configure#help colocation

20.//role(可省）:每一个资源都可以定义很多角色， 每一个资源代理在启动一个资源时，会经过很多阶段，刚准备启动叫premote,启动起来叫start，停止为stop。role在主从模型中用的多点，定义主的启动了，从的才能启动或其它role。

21.crm(live)configure#colocation webserver\_with\_webip inf: webserver webip

22.//让webserver与webip在一起

23.crm(live)configure#verify

24.crm(live)configure#commit

25.crm(live)configure#cd

26.crm(live)#status

c).定义顺序约束

1.义顺序约束：

2.crm(live)configure#help order

3.//kind类型：Mandatory | Optional | Serialize

4.强制，可选，顺序

5.crm(live)configure# order webip\_before\_webserver mandatory: webip webserver

6.//按启动顺序写，从左至右

7.crm(live)configure#commit

8.crm(live)configure#show

d).定义位置约束

1.crm(live)configure#location webip\_on\_node2 webip 200: node2.hoo.com

2.//定义为webip运行在node2的倾向性为200

3.crm(live)configure#verify

4.crm(live)configure#cd

5.crm(live)#status

6.//已转移到node2,node1默认为0;

观察，停掉node2

01.//停掉node2

02.# service corosync stop

03.#crm status

04.//资源没有转移到node1上，资源也没有显示了;

05.//显示：Current DC: node1.magedu.com - partition WITHOUT QUORUM,表明，node2挂了，node1不具备法定票数，不具备法定票数的默认操作为suicide或stop;suicide全部资源全挂，不显示任何资源

06.//两节点集群是一个特殊的集群，当node2挂了，希望资源转移到node1上，此时需要定义全局属性，一旦不具备法定票数时，应该忽略，而不是停止资源：

07.//上线node2:

08.#service corosync start

09.#crm status

10.//显示在node2上启动了;

11.#crm configure

12.crm(live)configure#property  no-quorum-policy=ignore

13.crm(live)configure#verify

14.crm(live)configure#show

15.crm(live)configure#commit

16.//观察，停掉node2:

17.# service corosync stop

18.#crm status

19.资源转移到node1，

20.仍显示：Current DC: node1.magedu.com - partition WITHOUT QUORUM

21.//启动node2:

22.#service corosync start

23.#crm status

24.//资源转移到node2,node2倾向性大

node1添加倾向值，观察

01.crm(live)configure#location webserver\_on\_node1 webserver 200: node1.hoo.com

02.//定义为webserver运行在node1的倾向性为200

03.crm(live)configure#verify

04.crm(live)configure#commit

05.crm(live)configure#show

06.#crm status

07.//此时仍在node2;

08.//停止node2:

09.# service corosync stop

10.#crm status

11.//转移到node1;

12.//上线node2:

13.#service corosync start

14.#crm status

15.//显示仍在node1;因为node1与node2倾向性一样，最终的倾向性取决于所有倾向性之和;

e).定义资源默认粘性

资源粘性生效于当前运行节点。资源运行在哪里，即在哪里生效。

01.crm(live)configure#help

02.crm(live)configure#edit

03.//删除webserver定义的对node1的200的倾向性;

04.#crm status

05.//此时所有服务都运行在node2

06.//配置粘性：

07.crm(live)configure#rsc-defaults resource-stickness=100

08.crm(live)configure#verify

09.crm(live)configure#show

10.crm(live)configure#commit

11.crm(live)configure#cd

12.crm(live)# status

测试：

01.//故障node2:

02.# service corosync stop

03.#crm status

04.//资源转移到node1;

05.//上线node2:

06.#crm status

07.//资源没有转移;

08.//对当前node的粘性和为200,与对node2的倾向性相同，故留在当前节点

09.//粘性定义，无关任一node，只生效当前所运行节点;

f).定义资源监控

1.//强行kill掉当前节点httpd

2.#killall httpd

3.#ss -tnl

4.#crm status

5.//显示服务照常运行

6.#service httpd status

7.//显示：httpd dead but subsys locked,subsys没有删除

8.//kill掉服务，不会自动重启;因为，节点没有故障，所有资源不会转移，默认情况下，pacemaker不会对任何资源进行监控,所以，即便是资源关掉了，只要节点没有故障，资源依然不会转移;要想达到资源转移的目的，得定义监控(monitoer);

01.//定义监控：

02.#crm configure

03.crm(live)configure#cd

04.crm(live)#status

05.//仍然在节点上运行着

06.crm(live)#resource

07.crm(live)resource# status

08.//仍然显示：start,没有定义监控，意识不到httpd停掉

09.crm(live)resource# stop webserver

10.crm(live)resource# stop webip

11.crm(live)resource# status

12.crm(live)resource# cleanup webserver

13.crm(live)resource# cleanup webip

14.//如果服务被非法关闭，需要清理，以免后续配置的一些麻烦

15.crm(live)resource# cd

16.crm(live)resource# configure

17.crm(live)configure# help monitor

18.crm(live)configure# monitor webserver 20s:15s

19.//监控时间：超时时间

20.crm(live)configure# verify

21.crm(live)configure#commit

22.crm(live)configure#cd

23.crm(live)# resource

24.crm(live)resource# start webip

25.crm(live)resource# start webserver

26.crm(live)resource#cd

27.crm(live)# status

28.//资源在node2上启动;

测试：

01.# service httpd stop

02.# ss -tnl

03.//查看日志：

04.#tail /var/log/cluster/corosync.log

05.//查看node2的httpd是否自启：

06.#ss -tnl

07.//启动成功;

08.//监控的意义：一旦发现服务没启动，就会尝试在当前主机上重启;

09.//配置ngnix争用端口

10.#service httpd stop  &&　service ngnix start

11.#ss -tnl

12.crm(live)#status

13.//重启无法成功，资源转移到node1

补充：

1.//定义资源的同时可以定义监控：

2.crm(live)configure# primitive vip ocf:heartbeat:Ipaddr  params ip=172.16.100.41 op monitor interval=30s timeout=15s op start timeout=20s op stop timeout=20s                            crm(live)resource#show

3.crm(live)configure#delet vip

4.//直接删除上面的定义

补充：CRM中的几个基本概念

1.资源类型

primitive（native）：基本资源，原始资源

group：资源组

clone：克隆资源（可同时运行在多个节点上），要先定义为primitive后才能进行clone。主要包含STONITH和集群文件系统（cluster filesystem）

master/slave：主从资源，如drdb（下文详细讲解）

2.RA类型

Lsb：linux表中库，一般位于/etc/rc.d/init.d/目录下的支持start|stop|status等参数的服务脚本都是lsb

ocf：Open cluster Framework，开放集群架构

heartbeat：heartbaet V1版本

stonith：专为配置stonith设备而用

3.资源粘性

资源粘性表示资源是否倾向于留在当前节点，如果为正整数，表示倾向，负数则会离开，-inf表示正无穷，inf表示正无穷。

4.资源约束

资源约束则用以指定在哪些群集节点上运行资源，以何种顺序装载资源，以及特定资源依赖于哪些其它资源。

**Resource Location（资源位置）：定义资源可以、不可以或尽可能在哪些节点上运行；**

**Resource Collocation（资源排列）：排列约束用以定义集群资源可以或不可以在某个节点上同时运行；**

**Resource Order（资源顺序）：顺序约束定义集群资源在节点上启动的顺序；**

**定义约束时，还需要指定值。资源安按值管理是集群工作方式的重要组成部分。从迁移资源到决定在已降级集群中停止哪些资源的整个过程是通过以某种方式改变资源值来实现的。值按每个资源来计算，资源值为负的任何节点都无法运行该资源。在计算出资源值后，集群选择值最高的节点。**

**有两个特殊值：inf(正无穷,表示只要有可能就要）、-inf(负无穷,表示只要有可能就不要)**

**定义资源约束时，也可以指定每个约束的值。值较高的约束先应用，值较低的约束后应用。通过使用不同的值为既定资源创建更多位置约束，可指定资源故障转移至的目标节点的顺序。**

法定票数问题：

在双节点集群中，由于票数是偶数，当心跳出现问题（脑裂）时，两个节点都将达不到法定票数，默认quorum策略会关闭集群服务，为了避免这种情况，可以增加票数为奇数（如前文的增加ping节点），或者调整默认quorum策略为【ignore】。

二、在CentOS 6.x上配置基于corosync的Web高可用

2.1 基本前提配置：

两个节点要实现时间同步、ssh互信、hosts名称解析、安装httpd，这里不做详细介绍。

记得一定要关闭httpd的自启动：

5:关闭    6:关闭

2.2 安装corosync：

注意，以下所有安装都是需要在所有节点上执行的。

2.2.1 安装依赖：

2.2.2 安装集群组件：

RHEL6.x以后的版本中，直接集成了corosync和pacemaker，因此直接使用yum安装即可。

2.2.3 安装crmsh实现资源管理：

从pacemaker 1.1.8开始，crm发展成了一个独立项目，叫crmsh。也就是说，我们安装了pacemaker后，并没有crm这个命令，我们要实现对集群资源管理，还需要独立安装crmsh。crmsh的rpm安装可从如下地址下载：

<http://download.opensuse.org/repositories/network:/ha-clustering:/Stable/CentOS_CentOS-6/x86_64/>

crmsh依赖于pssh，因此也需要通过上面地址下载pssh.rpm

wget <http://download.opensuse.org/repositories/network:/ha-clustering:/Stable/CentOS_CentOS-6/x86_64/pssh-2.3.1-3.2.x86_64.rpm>3.1-3.2.x86\_64.rpm crmsh-1.2.6-5.1.x86\_64.rpm

特别注意：

使用了crmsh，就不在需要安装heartbeat，之前的版本中都需要安装heartbeat以利用其crm进行资源管理。网上大量的教程中都是基于pacemaker1.1.8以前的版本的。

说明：

如果不用yum安装，则所有的rpm包均可在<http://clusterlabs.org/rpm/epel-5/x86_64/>下载，具体如下：

2.3 配置corosync：

2.3.1 主配置文件：

2.3.2 生成认证key：

使用corosync-keygen生成key时，由于要使用/dev/random生成随机数，因此如果新装的系统操作不多，如果没有足够的熵（关于random使用键盘敲击产生随机数的原理可自行google），可能会出现如下提示：

此时只需要在本地登录后狂敲键盘即可！

2.3.3  拷贝配置至节点2：

2.3.4 启动corosync

[root@node1 corosync]# s

2.3.5 检查启动情况：

查看corosync引擎是否正常启动：

查看初始化成员节点通知是否正常发出：

检查启动过程中是否有错误产生： ker with CMAN

查看pacemaker是否正常启动：

**可能存在的问题：**iptables没有配置相关策略，导致两个节点无法通信。可关闭iptables或配置节点间的通信策略。

3、集群资源管理

3.1 crmsh基本介绍

ng.com ]

3.2 法定票数问题：

在双节点集群中，由于票数是偶数，当心跳出现问题（脑裂）时，两个节点都将达不到法定票数，默认quorum策略会关闭集群服务，为了避免这种情况，可以增加票数为奇数（如前文的增加ping节点），或者调整默认quorum策略为【ignore】。

3.3 防​止​资​源​在​节​点​恢​复​后​移​动​

故障发生时，资源会迁移到正常节点上，但当故障节点恢复后，资源可能再次回到原来节点，这在有些情况下并非是最好的策略，因为资源的迁移是有停机时间的，特别是一些复杂的应用，如oracle数据库，这个时间会更长。为了避免这种情况，可以根据需要，使用本文**1.3.1**介绍的资源粘性策略。资源粘性为100

3.4 配置一个web集群

3.4.1 定义IP： ]

WebIP    (ocf::heartbeat:IPaddr):    Started node1.toxingwang.com

注意上述最后一行，定义的资源已经的node1上启动。使用ifconfig命令也可以看到该IP已生效：

此时，我们在node2上停止node1的corosync服务，就会发现该IP自动转移到node2：

3.4.2 配置httpd资源：

从上面的信息中可以看出WebIP和WebSite有可能会分别运行于两个节点上，这对于通过此IP提供Web服务的应用来说是不成立的，即此两者资源必须同时运行在某节点上。

3.4.3 资源约束

由此可见，即便集群拥有所有必需资源，但它可能还无法进行正确处理。资源约束则用以指定在哪些群集节点上运行资源，以何种顺序装载资源，以及特定资源依赖于哪些其它资源。pacemaker共给我们提供了三种资源约束方法：   
**1）Resource Location（资源位置）：**定义资源可以、不可以或尽可能在哪些节点上运行；   
**2）Resource Collocation（资源排列）：**排列约束用以定义集群资源可以或不可以在某个节点上同时运行；   
**3）Resource Order（资源顺序）：**顺序约束定义集群资源在节点上启动的顺序；

定义约束时，还需要指定分数。各种分数是集群工作方式的重要组成部分。其实，从迁移资源到决定在已降级集群中停止哪些资源的整个过程是通过以某种方式修改分数来实现的。分数按每个资源来计算，资源分数为负的任何节点都无法运行该资源。在计算出资源分数后，集群选择分数最高的节点。INFINITY（无穷大）目前定义为 1,000,000。加减无穷大遵循以下3个基本规则：   
1）任何值 + 无穷大 = 无穷大   
2）任何值 - 无穷大 = -无穷大   
3）无穷大 - 无穷大 = -无穷大

定义资源约束时，也可以指定每个约束的分数。分数表示指派给此资源约束的值。分数较高的约束先应用，分数较低的约束后应用。通过使用不同的分数为既定资源创建更多位置约束，可以指定资源要故障转移至的目标节点的顺序。

因此，对于前述的WebIP和WebSite可能会运行于不同节点的问题，可以通过以下命令来解决：

提交后查看状态：

接着，我们还得确保WebSite在某节点启动之前得先启动WebIP，这可以使用如下命令实现：

crm(live)configure# order httpd-after-ip mandatory: WebIP WebSite

查看效果：

[](http://www.toxingwang.com/wp-content/uploads/2013/10/image.png)

此外，由于HA集群本身并不强制每个节点的性能相同或相近，所以，某些时候我们可能希望在正常时服务总能在某个性能较强的节点上运行，这可以通过位置约束来实现：

3.4.4 测试：

[](http://www.toxingwang.com/wp-content/uploads/2013/10/image1.png)

**至此，一个基本的HA集群就搭建完成。**

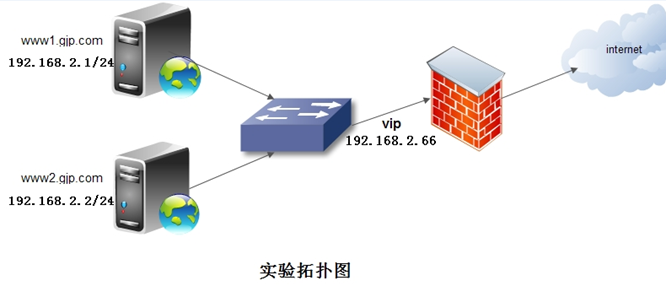
http://blog.51cto.com/image/skin/artType01.jpg Linux下高可用群集之corosync+openais+pacemaker+web+drbd

2012-10-25 21:28:00

标签：[corosync](http://blog.51cto.com/tag-corosync.html) [openais](http://blog.51cto.com/tag-openais.html) [pacemaker](http://blog.51cto.com/tag-pacemaker.html) [drbd](http://blog.51cto.com/tag-drbd.html) [web](http://blog.51cto.com/tag-web.html)

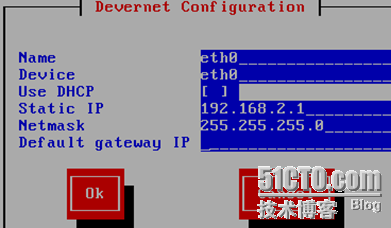
原创作品，允许转载，转载时请务必以超链接形式标明文章 [原始出处](http://guojiping.blog.51cto.com/5635432/1036951) 、作者信息和本声明。否则将追究法律责任。<http://guojiping.blog.51cto.com/5635432/1036951>

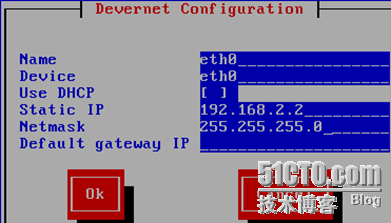
项目拓扑图：

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171583g42n.png)

corosync 具体配置：

1.配置IP   setup

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171588C1uW.png)

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171589Mv8M.png)

2.保证名称你能够相互解析：uname –r 必须相同

[root@www1 ~]# uname -rn   
www1.gjp.com 2.6.18-164.el5

www1.gjp.com上的配置：

[root@gjp99 ~]# cat /etc/sysconfig/network   
NETWORKING=yes   
NETWORKING\_IPV6=yes   
HOSTNAME=www1.gjp.com   
[root@gjp99 ~]# hostname www1.gjp.com   
[root@gjp99 ~]# hostname   
www1.gjp.com

logout登出重新登陆即可！

3.保证系统时钟一致

[root@www1 ~]# hwclock -s   
[root@www1 ~]# clock   
Tue 23 Oct 2012 05:20:36 PM CST  -0.017990 seconds

4.修改hosts（代替dns）

[root@www1 ~]# cat /etc/hosts   
# Do not remove the following line, or various programs   
# that require network functionality will fail.   
127.0.0.1   localhost.localdomain  localhost   
::1        localhost6.localdomain6 localhost6   
192.168.2.1     www1.gjp.com    www1   
192.168.2.2     www2.gjp.com    www2

[root@www1 ~]# ping www2.gjp.com   
PING www2.gjp.com (192.168.2.2) 56(84) bytes of data.   
64 bytes from www2.gjp.com (192.168.2.2): icmp\_seq=1 ttl=64 time=3.45 ms   
64 bytes from www2.gjp.com (192.168.2.2): icmp\_seq=2 ttl=64 time=0.658 ms

名称已经能够相互解析！

5. 挂载光盘并安装corosync所需安装包

[root@www1 ~]# mkdir /mnt/cdrom   
[root@www1 ~]# mount /dev/cdrom /mnt/cdrom   
mount: block device /dev/cdrom is write-protected, mounting read-only

[root@www2 ~]# scp \*.rpm www2:/root

在www2上拷贝上传的rpm包到www1的root目录下：   
[root@www1 ~]# yum localinstall -y \*.rpm –nogpgcheck

6.编辑corosync的配置文档

[root@www1 ~]# cd /etc/corosync/   
[root@www1 corosync]# ll   
total 20   
-rw-r--r-- 1 root root 5384 Jul 28  2010 amf.conf.example   
-rw-r--r-- 1 root root  436 Jul 28  2010 corosync.conf.example   
drwxr-xr-x 2 root root 4096 Jul 28  2010 service.d   
drwxr-xr-x 2 root root 4096 Jul 28  2010 uidgid.d

[root@www1 corosync]# cp corosync.conf.example corosync.conf   
[root@www1 corosync]# vim corosync.conf

compatibility: whitetank  (表示兼容corosync 0.86的版本，向后兼容，兼容老的版本，一些   
                           新的功能可能无法实用）

（图腾的意思  ，多个节点传递心跳时的相关协议的信息）   
totem {   
        version: 2  版本号   
        secauth: off  是否代开安全认证   
        threads: 0   多少个现成认证  0 无限制   
        interface {   
                ringnumber: 0     
                bindnetaddr: 192 168.2.0  通过哪个网络地址进行通讯，可以给个主机地址（给成192.168.2.0   
                mcastaddr: 226.94.1.1   
                mcastport: 5405   
        }     
}

logging {   
        fileline: off   
        to\_stderr: no  是否发送标准出错   
        to\_logfile: yes  日志   
        to\_syslog: yes   系统日志  （建议关掉一个），会降低性能   
        logfile: /var/log/cluster/corosync.log  （手动创建目录）   
        debug: off  排除时可以起来   
        timestamp: on 日志中是否记录时间

      一下是openais的东西，可以不用代开   
        logger\_subsys {   
                subsys: AMF   
                debug: off   
        }     
}

amf {   
        mode: disabled   
}   
补充一些东西，前面只是底层的东西，因为要用pacemaker

service {   
        ver: 0   
        name: pacemaker   
}   
虽然用不到openais ，但是会用到一些子选项

aisexec {   
        user: root   
        group: root   
}

7.为了便面其他主机加入该集群，需要认证，生成一个authkey

[root@www1 corosync]# corosync-keygen

[root@www1 corosync]# ll   
total 28   
-rw-r--r-- 1 root root 5384 Jul 28  2010 amf.conf.example   
-r-------- 1 root root  128 Oct 24 13:59 authkey   
-rw-r--r-- 1 root root  538 Oct 24 13:56 corosync.conf   
-rw-r--r-- 1 root root  436 Jul 28  2010 corosync.conf.example   
drwxr-xr-x 2 root root 4096 Jul 28  2010 service.d   
drwxr-xr-x 2 root root 4096 Jul 28  2010 uidgid.d

[root@www1 corosync]# scp -p authkey corosync.conf www2:/etc/corosync/

8.该目录必须提前创建

[root@www1 ~]# mkdir /var/log/cluster

[root@www1 corosync]# ssh www2 'mkdir  /var/log/cluster

9.启动corosync服务

[root@www1 corosync]# service corosync start   
Starting Corosync Cluster Engine (corosync):               [  OK  ]   
[root@www1 corosync]# ssh www2 'service corosync start'   
root@www2's password:   
Starting Corosync Cluster Engine (corosync):   
[  OK  ]

10.检测corosync是否无误

验证corosync引擎是否正常启动了

[root@www1 corosync]# grep -i  -e "corosync cluster engine" -e "configuration file" /var/log/messages  
Oct 24 11:09:04 www1 smartd[3260]: Opened configuration file /etc/smartd.conf   
Oct 24 11:09:04 www1 smartd[3260]: Configuration file /etc/smartd.conf was parsed, found DEVICESCAN, scanning devices   
Oct 24 17:08:33 www1 corosync[26362]:   [MAIN  ] Corosync Cluster Engine ('1.2.7'): started and ready to provide service.   
Oct 24 17:08:33 www1 corosync[26362]:   [MAIN  ] Successfully read main configuration file '/etc/corosync/corosync.conf'.

查看初始化成员节点通知是否发出

[root@www1 corosync]# grep -i totem /var/log/messages   
Oct 24 17:08:33 www1 corosync[26362]:   [TOTEM ] Initializing transport (UDP/IP).   
Oct 24 17:08:33 www1 corosync[26362]:   [TOTEM ] Initializing transmit/receive security: libtomcrypt SOBER128/SHA1HMAC (mode 0).   
Oct 24 17:08:33 www1 corosync[26362]:   [TOTEM ] The network interface is down.   
Oct 24 17:08:34 www1 corosync[26362]:   [TOTEM ] A processor joined or left the membership and a new membership was formed.

[root@www2 ~]# grep -i totem /var/log/messages   
Oct 24 17:09:07 www2 corosync[28610]:   [TOTEM ] Initializing transport (UDP/IP).   
Oct 24 17:09:07 www2 corosync[28610]:   [TOTEM ] Initializing transmit/receive security: libtomcrypt SOBER128/SHA1HMAC (mode 0).   
Oct 24 17:09:07 www2 corosync[28610]:   [TOTEM ] The network interface is down.   
Oct 24 17:09:08 www2 corosync[28610]:   [TOTEM ] A processor joined or left the membership and a new membership was formed.

检查过程中是否有错误产生

[root@www1 corosync]# grep -i error:  /var/log/messages  |grep -v unpack\_resources

[root@www2 ~]# grep -i error:  /var/log/messages  |grep -v unpack\_resources

不显示任何信息证明正确无误！

检查pacemaker时候已经启动了

[root@www1 corosync]# grep -i totem /var/log/messages   
Oct 24 17:08:33 www1 corosync[26362]:   [TOTEM ] Initializing transport (UDP/IP).   
Oct 24 17:08:33 www1 corosync[26362]:   [TOTEM ] Initializing transmit/receive sec   
Oct 24 17:08:33 www1 corosync[26362]:   [TOTEM ] The network interface is down.   
Oct 24 17:08:34 www1 corosync[26362]:   [TOTEM ] A processor joined or left the me   
[root@www1 corosync]# grep -i error:  /var/log/messages  |grep -v unpack\_resources   
[root@www1 corosync]# grep -i pcmk\_startup /var/log/messages   
Oct 24 17:08:34 www1 corosync[26362]:   [pcmk  ] info: pcmk\_startup: CRM: Initialized   
Oct 24 17:08:34 www1 corosync[26362]:   [pcmk  ] Logging: Initialized pcmk\_startup   
Oct 24 17:08:34 www1 corosync[26362]:   [pcmk  ] info: pcmk\_startup: Maximum core file size is: 4294967295   
Oct 24 17:08:34 www1 corosync[26362]:   [pcmk  ] info: pcmk\_startup: Service: 9   
Oct 24 17:08:34 www1 corosync[26362]:   [pcmk  ] info: pcmk\_startup: Local hostname: www1.gjp.com

[root@www2 ~]# grep -i pcmk\_startup /var/log/messages

前集群的节点上启动另外一个节点

[root@www1 ~]# /etc/init.d/corosync start   
Starting Corosync Cluster Engine (corosync):               [  OK  ]   
[root@www1 ~]# ssh www2  '/etc/init.d/corosync start'   
root@www2's password:   
Starting Corosync Cluster Engine (corosync): [  OK  ]

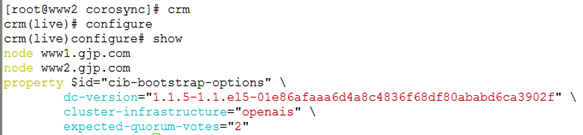
[root@www2 corosync]# crm status   
============   
Last updated: Wed Oct 24 20:11:19 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
0 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

提示：集群的节点之间的时间应该是同步的，

提供高可用服务   
在corosync中，定义服务可以用两种借口

1.图形接口  （使用hb—gui）   
2.crm  （pacemaker 提供,是一个shell）

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171591Vh9x.png)

用于查看cib的相关信息

如何验证该文件的语法错误

[root@www1 corosync]# crm\_verify  -L   
crm\_verify[4329]: 2012/10/25\_14:59:35 ERROR: unpack\_resources: Resource start-up disabled since no STONITH resources have been defined   
crm\_verify[4329]: 2012/10/25\_14:59:35 ERROR: unpack\_resources: Either configure some or disable STONITH with the stonith-enabled option   
crm\_verify[4329]: 2012/10/25\_14:59:35 ERROR: unpack\_resources: NOTE: Clusters with shared data need STONITH to ensure data integrity   
Errors found during check: config not valid   
  -V may provide more details

可以看到有stonith错误，在高可用的环境里面，会禁止实用任何支援   
可以禁用stonith

[root@www1 corosync]# crm   
crm(live)# configure   
crm(live)configure#  property stonith-enabled=false   
crm(live)configure# commit   
crm(live)configure# show   
node www1.gjp.com   
node www2.gjp.com   
property $id="cib-bootstrap-options" \   
    dc-version="1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f" \   
    cluster-infrastructure="openais" \   
    expected-quorum-votes="2" \   
    stonith-enabled="false"

再次进行检查

[root@www1 corosync]# crm\_verify  -L

    没有错误了   
    系统上有专门的stonith命令

stonith   -L   显示stonith所指示的类型   
crm可以使用交互式模式    
可以执行help   
保存在cib里面，以xml的格式

11.资源的配置

集群的资源类型有4种   
primitive   本地主资源 （只能运行在一个节点上）   
group     把多个资源轨道一个组里面，便于管理   
clone    需要在多个节点上同时启用的  （如ocfs2  ，stonith ，没有主次之分）   
master    有主次之分，如drbd

现在用的资源

ip地址  http服务  共享存储   
用资源代理进行配置   
ocf  lsb的   
使用list可以查看

[root@www1 corosync]# crm   
crm(live)# help

This is the CRM command line interface program.

Available commands:

    cib              manage shadow CIBs   
    resource         resources management   
    configure        CRM cluster configuration   
    node             nodes management   
    options          user preferences   
    ra               resource agents information center   
    status           show cluster status   
    quit,bye,exit    exit the program   
    help             show help   
    end,cd,up        go back one level

crm(live)# ra   
[](http://img1.51cto.com/attachment/201210/25/5635432_1351171597znQk.png)

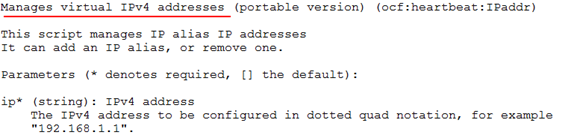
（是/etc/init.d目录下的）

crm(live)ra# list ocf heartbeat

实用info或者meta 用于显示一个资源的详细信息

  meta ocf:heartbeat:IPaddr  各个子项用：分开

crm(live)ra# meta ocf:heartbeat:IPaddr

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171602jREw.png)

配置一个资源，可以在configuration 下面进行配置

1.先资源名字

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171607RVWZ.png)

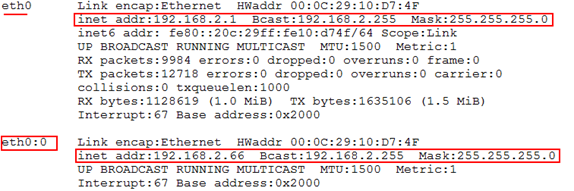
crm(live)configure# commit   
crm(live)configure# end   
crm(live)# status   
============   
Last updated: Thu Oct 25 15:18:54 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
1 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

webip    (ocf::heartbeat:IPaddr):    Started www1.gjp.com

可以看出该资源在node1上启动

[root@www1 corosync]# ifconfig |less

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171614nlTT.png)

[root@www1 corosync]# mount /dev/cdrom /mnt/cdrom   
mount: block device /dev/cdrom is write-protected, mounting read-only   
[root@www1 corosync]# yum install httpd -y

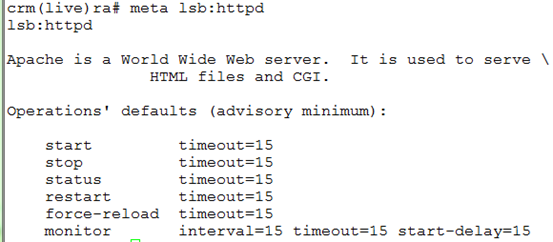
[root@www1 corosync]# service httpd status   
httpd is stopped   
[root@www1 corosync]# chkconfig --list |grep httpd   
httpd              0:off    1:off    2:off    3:off    4:off    5:off    6:off   
[root@www1 corosync]# crm   
crm(live)# ra   
crm(live)ra# classes   
heartbeat   
lsb   
ocf / heartbeat pacemaker   
stonith

定义web服务资源   
  在两个节点上都要进行安装   
  安装完毕后，可以查看httpd的lsb脚本

[root@www1 corosync]# crm ra list lsb

[root@www1 corosync]# crm   
或者

crm(live)# ra   
crm(live)ra# list lsb

[](http://img1.51cto.com/attachment/201210/25/5635432_13511716187Sm1.png)

crm(live)ra# end   
crm(live)# configure   
crm(live)configure# primitive webserver lsb:httpd

定义httpd的资源   
crm(live)configure# show   
node www1.gjp.com   
node www2.gjp.com   
primitive webip ocf:heartbeat:IPaddr \   
    params ip="192.168.2.66"   
primitive webserver lsb:httpd   
property $id="cib-bootstrap-options" \   
    dc-version="1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f" \   
    cluster-infrastructure="openais" \   
    expected-quorum-votes="2" \   
    stonith-enabled="false"   
crm(live)configure# commit   
crm(live)configure# end   
crm(live)# status   
============   
Last updated: Thu Oct 25 16:06:46 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
2 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

webip    (ocf::heartbeat:IPaddr):    Started www1.gjp.com   
webserver    (lsb:httpd):    Started www1.gjp.com

Failed actions:   
    webserver\_monitor\_0 (node=www2.gjp.com, call=3, rc=5, status=complete): not installed

如果www2.gjp.com上面已安装http服务，则会出现，ip在www1上，服务在www2上运行！

[root@www1 ~]# service httpd status   
httpd (pid  4897) is running...

[root@www1 ~]# echo "www1.gjp.com">/var/www/html/index.html   
[root@www1 ~]# crm   
crm(live)# configure   
crm(live)configure# help group

The `group` command creates a group of resources.

Usage:   
...............   
        group <name> <rsc> [<rsc>...]   
          [meta attr\_list]   
          [params attr\_list]

        attr\_list :: [$id=<id>] <attr>=<val> [<attr>=<val>...] | $id-ref=<id>   
...............   
Example:   
...............   
        group internal\_www disk0 fs0 internal\_ip apache \   
          meta target\_role=stopped   
...............

crm(live)configure# group web webip webserver   
crm(live)configure# commit   
[](http://img1.51cto.com/attachment/201210/25/5635432_13511716220qhd.png)

客户机测试：

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171626XptO.png)

[root@www1 ~]# crm status   
============   
Last updated: Thu Oct 25 16:34:28 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
1 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

Resource Group: web   
     webip    (ocf::heartbeat:IPaddr):    Started www1.gjp.com   
     webserver    (lsb:httpd):    Started www1.gjp.com

Failed actions:   
    webserver\_monitor\_0 (node=www2.gjp.com, call=3, rc=5, status=complete): not installed

模拟www1已经死掉：   
[root@www1 ~]# service corosync stop   
Signaling Corosync Cluster Engine (corosync) to terminate: [  OK  ]   
Waiting for corosync services to unload:.......            [  OK  ]

[root@www1 ~]# service httpd status   
httpd is stopped

[root@www2 Server]# crm status   
============   
Last updated: Thu Oct 25 16:43:01 2012   
Stack: openais   
Current DC: www2.gjp.com - partition WITHOUT quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
1 Resources configured.   
============

Online: [ www2.gjp.com ]   
OFFLINE: [ www1.gjp.com ]

Failed actions:   
    webserver\_monitor\_0 (node=www2.gjp.com, call=3, rc=5, status=complete): not installed

发现总有这个错误提示：

安装提示在www2上安装http服务，必须重启服务，否则，不能识别到，错误仍在！

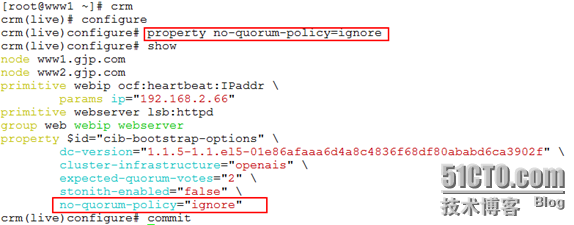
[root@www2 Server]# service corosync stop   
Signaling Corosync Cluster Engine (corosync) to terminate: [  OK  ]   
Waiting for corosync services to unload:^[[A.^H.....       [  OK  ]   
[root@www2 Server]# service corosync start   
Starting Corosync Cluster Engine (corosync):               [  OK  ]   
[root@www2 Server]# crm status   
============   
Last updated: Thu Oct 25 16:47:18 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
1 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

Resource Group: web   
     webip    (ocf::heartbeat:IPaddr):    Started www1.gjp.com   
     webserver    (lsb:httpd):    Started www1.gjp.com

解决www2接管不了服务的问题：

[root@www1 ~]# service corosync start   
Starting Corosync Cluster Engine (corosync):               [  OK  ]   
[root@www1 ~]# service httpd status   
httpd (pid  5233) is running...

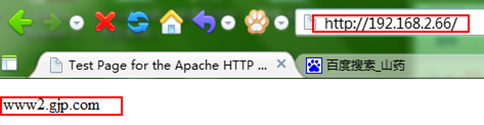
[](http://img1.51cto.com/attachment/201210/25/5635432_1351171634lYnu.png)

在www2上创建网站：

[root@www2 Server]# echo "www2.gjp.com " >/var/www/html/index.html

一旦www1死掉：

[root@www1 ~]# service corosync stop   
Signaling Corosync Cluster Engine (corosync) to terminate: [  OK  ]   
Waiting for corosync services to unload:.......            [  OK  ]

[](http://img1.51cto.com/attachment/201210/25/5635432_13511716378Yov.png)

能够正常访问：

[root@www2 Server]# service httpd status   
httpd (pid  4656) is running...

[root@www2 Server]# crm status   
============   
Last updated: Thu Oct 25 17:12:16 2012   
Stack: openais   
Current DC: www2.gjp.com - partition WITHOUT quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
1 Resources configured.   
============

Online: [ www2.gjp.com ]   
OFFLINE: [ www1.gjp.com ]

Resource Group: web   
     webip    (ocf::heartbeat:IPaddr):    Started www2.gjp.com   
     webserver    (lsb:httpd):    Started www2.gjp.com

如果www1恢复了，不能把权利争夺过来！观看如下配置：

[root@www1 ~]# service corosync start   
Starting Corosync Cluster Engine (corosync):               [  OK  ]   
[root@www1 ~]# crm status   
============   
Last updated: Thu Oct 25 17:20:44 2012   
Stack: openais   
Current DC: www2.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
1 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

Resource Group: web   
     webip    (ocf::heartbeat:IPaddr):    Started www2.gjp.com   
     webserver    (lsb:httpd):    Started www2.gjp.com

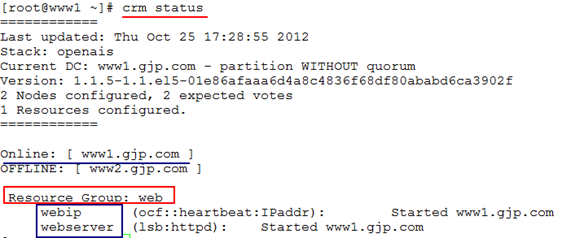
[](http://img1.51cto.com/attachment/201210/25/5635432_1351171643mIPo.png)

随便刷新都定位到www2上！

除非www2上的corosync服务死掉

[root@www2 Server]# service corosync stop   
Signaling Corosync Cluster Engine (corosync) to terminate: [  OK  ]   
Waiting for corosync services to unload:.......            [  OK  ]

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171650Zmt7.png)

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171657Ru7v.png)

www2.gjp.com上的配置：

[root@gjp99 ~]# cat /etc/sysconfig/network   
NETWORKING=yes   
NETWORKING\_IPV6=yes   
HOSTNAME=www2.gjp.com   
[root@gjp99 ~]# hostname www2.gjp.com

[root@www2 ~]# hwclock -s   
[root@www2 ~]# clock   
Tue 23 Oct 2012 05:20:32 PM CST  -0.018132 seconds

[root@www2 .ssh]# cat /etc/hosts   
# Do not remove the following line, or various programs   
# that require network functionality will fail.   
127.0.0.1   localhost.localdomain  localhost   
::1        localhost6.localdomain6 localhost6   
192.168.2.1     www1.gjp.com      www1   
192.168.2.2     www2.gjp.com      www2

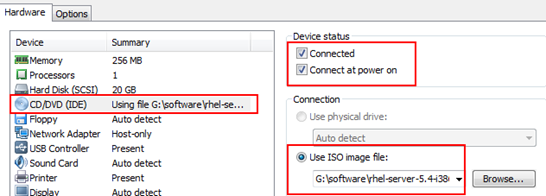
[root@www2 ~]# ping www1.gjp.com   
PING www1.gjp.com (192.168.2.1) 56(84) bytes of data.   
64 bytes from www1.gjp.com (192.168.2.1): icmp\_seq=1 ttl=64 time=1.11 ms   
64 bytes from www1.gjp.com (192.168.2.1): icmp\_seq=2 ttl=64 time=0.506 ms

名称已经能够相互解析！

[root@www2 ~]# cat /etc/yum.repos.d/rhel-debuginfo.repo   
[rhel-server]   
name=Red Hat Enterprise Linux Server   
baseurl=file:///mnt/cdrom/Server   
enabled=1   
gpgcheck=1   
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-redhat-release

[rhel-cluster]   
name=Red Hat Enterprise Linux Cluster   
baseurl=file:///mnt/cdrom/Cluster   
enabled=1   
gpgcheck=1   
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-redhat-release

要保证光驱已连接:

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171661ecBe.png)

[root@www2 ~]# mkdir /mnt/cdrom   
[root@www2 ~]# mount /dev/cdrom /mnt/cdrom

[root@www2 ~]# yum grouplist all   
Loaded plugins: rhnplugin, security   
This system is not registered with RHN.   
RHN support will be disabled.   
Setting up Group Process   
rhel-cluster                                                                | 1.3 kB     00:00       
rhel-cluster/primary                                                        | 6.5 kB     00:00       
rhel-server                                                                 | 1.3 kB     00:00       
rhel-server/primary                                                         | 732 kB     00:00       
rhel-cluster/group                                                          | 101 kB     00:00       
rhel-server/group                                                           | 1.0 MB     00:00       
Done

实现同一网段内的无障碍通讯！

[root@www2 ~]# ssh-keygen -t rsa   
Generating public/private rsa key pair.   
Enter file in which to save the key (/root/.ssh/id\_rsa):   
/root/.ssh/id\_rsa already exists.   
Overwrite (y/n)?   
[root@www2 ~]# cd .ssh/   
[root@www2 .ssh]# ls   
id\_rsa  id\_rsa.pub   
[root@www2 .ssh]# ssh-copy-id -i id\_rsa.pub www1   
10   
The authenticity of host 'www1 (192.168.2.1)' can't be established.   
RSA key fingerprint is 87:be:8b:a4:bd:11:11:10:c2:ec:2d:ef:02:68:f6:0e.   
Are you sure you want to continue connecting (yes/no)? yes   
Warning: Permanently added 'www1,192.168.2.1' (RSA) to the list of known hosts.   
root@www1's password:   
Now try logging into the machine, with "ssh 'www1'", and check in:

  .ssh/authorized\_keys

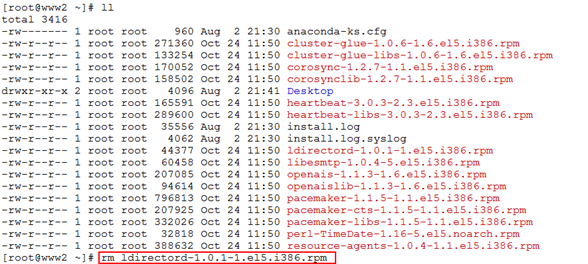
to make sure we haven't added extra keys that you weren't expecting.

[root@www2 .ssh]# scp /etc/yum.repos.d/rhel-debuginfo.repo  www1:/etc/yum.repos.d/   
rhel-debuginfo.repo                                              100%  318     0.3KB/s   00:00      
[root@www2 .ssh]# date   
Wed Oct 24 11:30:30 CST 2012   
[root@www2 .ssh]# ssh www1 'date'   
Wed Oct 24 11:30:40 CST 2012

[root@www1 ~]# ssh-keygen -t rsa

[root@www1 .ssh]# ssh-copy-id -i id\_rsa.pub www2

上传所需软件包：

[](http://img1.51cto.com/attachment/201210/25/5635432_1351171669Ytce.png)

[root@www2 ~]# mount /dev/cdrom /mnt/cdrom   
mount: block device /dev/cdrom is write-protected, mounting read-only   
[root@www2 ~]# yum localinstall -y \*.rpm –nogpgcheck

验证corosync引擎是否正常启动了

[root@www2 ~]#  grep -i  -e "corosync cluster engine" -e "configuration file" /var/log/messages   
Oct 24 11:09:03 www2 smartd[3259]: Opened configuration file /etc/smartd.conf   
Oct 24 11:09:03 www2 smartd[3259]: Configuration file /etc/smartd.conf was parsed, found DEVICESCAN, scanning devices   
Oct 24 17:09:07 www2 corosync[28610]:   [MAIN  ] Corosync Cluster Engine ('1.2.7'): started and ready to provide service.   
Oct 24 17:09:07 www2 corosync[28610]:   [MAIN  ] Successfully read main configuration file '/etc/corosync/corosync.conf'.

[root@www2 Server]# yum install httpd -y

DRBD的配置：

www1 的配置：

[root@www1 ~]# fdisk /dev/sda

The number of cylinders for this disk is set to 2610.   
There is nothing wrong with that, but this is larger than 1024,   
and could in certain setups cause problems with:   
1) software that runs at boot time (e.g., old versions of LILO)   
2) booting and partitioning software from other OSs   
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): n   
Command action   
   e   extended   
   p   primary partition (1-4)   
e   
Selected partition 4   
First cylinder (1354-2610, default 1354):   
Using default value 1354   
Last cylinder or +size or +sizeM or +sizeK (1354-2610, default 2610):   
Using default value 2610

Command (m for help): p

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1353      522112+  82  Linux swap / Solaris   
/dev/sda4            1354        2610    10096852+   5  Extended

Command (m for help): n   
First cylinder (1354-2610, default 1354): p   
First cylinder (1354-2610, default 1354):   
Using default value 1354   
Last cylinder or +size or +sizeM or +sizeK (1354-2610, default 2610): +2g

Command (m for help): p

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1353      522112+  82  Linux swap / Solaris   
/dev/sda4            1354        2610    10096852+   5  Extended   
/dev/sda5            1354        1597     1959898+  83  Linux

Command (m for help): w   
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.   
The kernel still uses the old table.   
The new table will be used at the next reboot.   
Syncing disks.

[root@www1 ~]# partprobe /dev/sda   
[root@www1 ~]# cat /proc/partitions   
major minor  #blocks  name

   8     0   20971520 sda   
   8     1     104391 sda1   
   8     2   10241437 sda2   
   8     3     522112 sda3   
   8     4          0 sda4   
   8     5    1959898 sda5

在节点2上做同样配置

安装drbd，用来构建分布式存储。

这里要选用适合自己系统的版本进行安装，我用到的是

drbd83-8.3.8-1.el5.centos.i386.rpm

kmod-drbd83-8.3.8-1.el5.centos.i686.rpm

[image](http://img1.51cto.com/attachment/201210/25/5635432_13511716764Axe.png)

[root@www1 ~]# yum localinstall -y drbd83-8.3.8-1.el5.centos.i386.rpm –nogpgcheck

[root@www1 ~]# yum localinstall -y kmod-drbd83-8.3.8-1.el5.centos.i686.rpm --nogpgcheck

在节点2上做同样操作

[root@www1 ~]# cp /usr/share/doc/drbd83-8.3.8/drbd.conf  /etc   
cp: overwrite `/etc/drbd.conf'? y  必须选择覆盖！     
[root@www1 ~]# scp /etc/drbd.conf  www2:/etc/

[root@www1 ~]# vim /etc/drbd.d/global\_common.conf   
[root@www1 ~]# cat /etc/drbd.d/global\_common.conf

global {   
        usage-count yes;   
        # minor-count dialog-refresh disable-ip-verification   
}

common {   
        protocol C;

        startup {   
                wfc-timeout  120;   
                degr-wfc-timeout 120;   
         }   
        disk {   
                  on-io-error detach;   
                  fencing resource-only;

          }   
        net {   
                cram-hmac-alg "sha1";   
                shared-secret  "mydrbdlab";   
         }   
        syncer {   
                  rate  100M;   
         }

}

[root@www1 ~]# vim /etc/drbd.d/web.res   
[root@www1 ~]# cat /etc/drbd.d/web.res   
resource  web {   
        on www1.gjp.com {   
        device   /dev/drbd0;   
        disk    /dev/sda5;   
        address  192.168.2.1:7789;   
        meta-disk       internal;   
        }

        on www2.gjp.com {   
        device   /dev/drbd0;   
        disk    /dev/sda5;   
        address  192.168.2.2:7789;   
        meta-disk       internal;   
        }     
}

开始初始化

双方借点上都要执行

drbdadm   create-md web

在双方的节点上启动服务

service drbd start

查看状态

[root@www1 ~]# drbdadm create-md web

[root@www1 ~]# service drbd start

启动时必须双方一块启动！

www2的配置：

[root@www2 Server]# fdisk /dev/sda

The number of cylinders for this disk is set to 2610.   
There is nothing wrong with that, but this is larger than 1024,   
and could in certain setups cause problems with:   
1) software that runs at boot time (e.g., old versions of LILO)   
2) booting and partitioning software from other OSs   
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1353      522112+  82  Linux swap / Solaris

Command (m for help): n   
Command action   
   e   extended   
   p   primary partition (1-4)   
e   
Selected partition 4   
First cylinder (1354-2610, default 1354):   
Using default value 1354   
Last cylinder or +size or +sizeM or +sizeK (1354-2610, default 2610):   
Using default value 2610

Command (m for help): p

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1353      522112+  82  Linux swap / Solaris   
/dev/sda4            1354        2610    10096852+   5  Extended

Command (m for help): n   
First cylinder (1354-2610, default 1354):   
Using default value 1354   
Last cylinder or +size or +sizeM or +sizeK (1354-2610, default 2610): +2g

Command (m for help): p

Disk /dev/sda: 21.4 GB, 21474836480 bytes   
255 heads, 63 sectors/track, 2610 cylinders   
Units = cylinders of 16065 \* 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System   
/dev/sda1   \*           1          13      104391   83  Linux   
/dev/sda2              14        1288    10241437+  83  Linux   
/dev/sda3            1289        1353      522112+  82  Linux swap / Solaris   
/dev/sda4            1354        2610    10096852+   5  Extended   
/dev/sda5            1354        1597     1959898+  83  Linux

Command (m for help): w   
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.   
The kernel still uses the old table.   
The new table will be used at the next reboot.   
Syncing disks.

[root@www2 Server]# partprobe /dev/sda   
[root@www2 Server]# cat /proc/partitions   
major minor  #blocks  name

   8     0   20971520 sda   
   8     1     104391 sda1   
   8     2   10241437 sda2   
   8     3     522112 sda3   
   8     4          0 sda4   
   8     5    1959898 sda5

[root@www1 ~]# scp drbd83-8.3.8-1.el5.centos.i386.rpm kmod-drbd83-8.3.8-1.el5.centos.i686.rpm www2:/root   
root@www2's password:   
drbd83-8.3.8-1.el5.centos.i386.rpm              100%  217KB 216.7KB/s   00:00      
kmod-drbd83-8.3.8-1.el5.centos.i686.rpm         100%  123KB 123.0KB/s   00:00

[root@www2 ~]# rpm -ivh drbd83-8.3.8-1.el5.centos.i386.rpm   
warning: drbd83-8.3.8-1.el5.centos.i386.rpm: Header V3 DSA signature: NOKEY, key ID e8562897   
Preparing...                ########################################### [100%]   
   1:drbd83                 warning: /etc/drbd.conf created as /etc/drbd.conf.rpmnew  
########################################### [100%]   
[root@www2 ~]# rpm -ivh kmod-drbd83-8.3.8-1.el5.centos.i686.rpm   
warning: kmod-drbd83-8.3.8-1.el5.centos.i686.rpm: Header V3 DSA signature: NOKEY, key ID e8562897   
Preparing...                ########################################### [100%]   
   1:kmod-drbd83            ########################################### [100%]   
[root@www2 ~]# cp /usr/share/doc/drbd83-8.3.8/drbd.conf   /etc/   
cp: overwrite `/etc/drbd.conf'? y

[root@www2 ~]# scp www1:/etc/drbd.d/global\_common.conf  /etc/drbd.d/global\_common.conf   
global\_common.conf                              100%  505     0.5KB/s   00:00

[root@www2 ~]# scp www1:/etc/drbd.d/web.res  /etc/drbd.d/web.res   
web.res                                         100%  348     0.3KB/s   00:00

[root@www2 ~]# drbdadm   create-md web

[root@www2 ~]# service drbd start   
Starting DRBD resources: [   
web   
Found valid meta data in the expected location, 2006929408 bytes into /dev/sda5.   
d(web) s(web) n(web) ].

[root@www1 ~]# service drbd start   
Starting DRBD resources: [ ].   
[root@www1 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent 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:b oos:195980

都为second 状态，没有同步

也可以

drbd-overview   
[root@www1 ~]# drbdadm   -- --overwrite-data-of-peer primary web

[root@www1 ~]# vim /etc/drbd.d/global\_common.conf

可调整同步速率：rate

[root@www1 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r----   
    ns:259716 nr:0 dw:0 dr:267904 al:0 bm:15 lo:1 pe:31 ua:256 ap:0 ep:1 wo:b oos:1701048   
    [=>..................] sync'ed: 13.4% (1701048/1959800)K delay\_probe: 25   
    finish: 0:00:37 speed: 45,120 (23,520) K/sec

[root@www1 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----   
    ns:1959800 nr:0 dw:0 dr:1959800 al:0 bm:120 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0

[root@www1 ~]# drbd-overview   
  0:web  Connected Primary/Secondary UpToDate/UpToDate C r----

[root@www2 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent 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:b oos:1959800   
[root@www2 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r----   
    ns:0 nr:1959800 dw:1959800 dr:0 al:0 bm:120 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:

创建文件系统（在主节点上实现）

mkfs -t ext3  -L drbdweb  /dev/drbd0

[root@www1 ~]# mkfs -t ext3  -L drbdweb  /dev/drbd0   
mke2fs 1.39 (29-May-2006)   
Filesystem label=drbdweb   
OS type: Linux   
Block size=4096 (log=2)   
Fragment size=4096 (log=2)   
245280 inodes, 489950 blocks   
24497 blocks (5.00%) reserved for the super user   
First data block=0   
Maximum filesystem blocks=503316480   
15 block groups   
32768 blocks per group, 32768 fragments per group   
16352 inodes per group   
Superblock backups stored on blocks:   
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done                              
Creating journal (8192 blocks): done   
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 37 mounts or   
180 days, whichever comes first.  Use tune2fs -c or -i to override.

[root@www1 ~]# mkdir /web   
[root@www1 ~]# mount /dev/drbd0 /web/   
[root@www1 ~]# mount   
/dev/sda2 on / type ext3 (rw)   
proc on /proc type proc (rw)   
sysfs on /sys type sysfs (rw)   
devpts on /dev/pts type devpts (rw,gid=5,mode=620)   
/dev/sda1 on /boot type ext3 (rw)   
tmpfs on /dev/shm type tmpfs (rw)   
none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)   
sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)   
/dev/hdc on /mnt/cdrom type iso9660 (ro)   
/dev/drbd0 on /web type ext3 (rw)

[root@www1 ~]# cd /web   
[root@www1 web]# echo "web1 " >index.html   
[root@www1 web]# ll   
total 20   
-rw-r--r-- 1 root root     6 Oct 25 21:11 index.html   
drwx------ 2 root root 16384 Oct 25 20:57 lost+found

[root@www2 ~]# mkdir /web2   
[root@www2 ~]# mount /dev/drbd0 /web2   
mount: block device /dev/drbd0 is write-protected, mounting read-only   
mount: Wrong medium type

从设备没有任何权限！

[root@www1 ~]# umount /web   
[root@www1 ~]# drbdadm secondary web   
[root@www1 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Secondary/Secondary ds:UpToDate/UpToDate C r----   
    ns:2024140 nr:0 dw:64340 dr:1959937 al:24 bm:135 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0

[root@www2 ~]# drbdadm primary web   
[root@www2 ~]# mount /dev/drbd0 /web2   
[root@www2 ~]# ll /web2   
total 20   
-rw-r--r-- 1 root root     6 Oct 25 21:11 index.html   
drwx------ 2 root root 16384 Oct 25 20:57 lost+found

[root@www2 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----   
    ns:40 nr:2024140 dw:2024180 dr:221 al:1 bm:120 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0   
[root@www2 ~]# cd /web2   
[root@www2 web2]# touch gjp.txt   
[root@www2 web2]# ll   
total 20   
-rw-r--r-- 1 root root     0 Oct 25 21:16 gjp.txt   
-rw-r--r-- 1 root root     6 Oct 25 21:11 index.html   
drwx------ 2 root root 16384 Oct 25 20:57 lost+found

注意：还原为www1为主，www2为辅，则必须把www2上的挂载点卸掉！然后再设置主备！

[root@www1 ~]# cat /proc/drbd   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r----   
    ns:2024140 nr:96 dw:64436 dr:1959937 al:24 bm:135 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:0

[root@www1 ~]# cd /var/www/html   
[root@www1 html]# ll   
total 4   
-rw-r--r-- 1 root root  0 Oct 25 16:14 gjp1   
-rw-r--r-- 1 root root 13 Oct 25 16:23 index.html   
[root@www1 html]# mv index.html /web/

mv: overwrite `/web/index.html'? y

必须覆盖，原来的index.html是随便写的，不是网站

[root@www1 html]# cd /web/   
[root@www1 web]# ll   
total 20   
-rw-r--r-- 1 root root     0 Oct 25 21:16 gjp.txt   
-rw-r--r-- 1 root root    13 Oct 25 16:23 index.html   
drwx------ 2 root root 16384 Oct 25 20:57 lost+found

[root@www1 web]# vim /etc/httpd/conf/httpd.conf

[image](http://img1.51cto.com/attachment/201210/28/5635432_1351416633VZHh.png)

将其修改为:

[image](http://img1.51cto.com/attachment/201210/28/5635432_1351416638Di68.png)

在www2上改变为

[image](http://img1.51cto.com/attachment/201210/28/5635432_1351416639OBwz.png)

下面实现COROSYNC自动调用brdb

brdb自动挂载挂载点， 由于访问的网站已经放到挂载点 /web下，所以全都能够自动实现!

修改如下：

由于两台corosync使用的是同一个配置文件，所以两台设备上的挂载点必须相同

即在www2上建立挂载点/web   并修改httpd.conf的网站默认目录/web

corosync 如何与drbd绑定？

把drbd添加到corosync服务上

代码添加

crm configure primitive drbd\_web\_FS ocf:heartbeat:Filesystem params device="/dev/drbd0" directory="/web" fstype="ext3"

crm configure primitive httpd\_drbd\_web ocf:heartbeat:drbd params drbd\_resource="web" op monitor interval="60s" role="Master" timeout="40s" op monitor interval="70s" role="Slave" timeout="40s"

crm configure master MS\_Webdrbd httpd\_drbd\_web meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"

crm configure colocation drbd\_web\_FS\_on\_MS\_Webdrbd inf: drbd\_web\_FS MS\_Webdrbd:Master

crm configure order drbd\_web\_FS\_after\_MS\_Webdrbd inf: MS\_Webdrbd:promote drbd\_web\_FS:start

crm configure property no-quorum-policy="ignore"

配置查看：

[](http://img1.51cto.com/attachment/201210/28/5635432_13514166447lRi.png)

[root@www1 ~]# cd /etc/drbd.d/   
[root@www1 drbd.d]# vim global\_common.conf

global {   
        usage-count no;   //注意该为no    
        # minor-count dialog-refresh disable-ip-verification   
}

common {   
        protocol C;

handlers {   
                pri-on-incon-degr "/usr/lib/drbd/notify-pri-on-incon-degr.sh; /usr/lib/drbd/notify-emergency-reboot.sh; echo b &gt; /proc/sysrq-trigger ; reboot -f";   
                pri-lost-after-sb "/usr/lib/drbd/notify-pri-lost-after-sb.sh; /usr/lib/drbd/notify-emergency-reboot.sh; echo b &gt; /proc/sysrq-trigger ; reboot -f";   
                local-io-error "/usr/lib/drbd/notify-io-error.sh; /usr/lib/drbd/notify-emergency-shutdown.sh; echo o &gt; /proc/sysrq-trigger ; halt -f";    
                fence-peer "/usr/lib/drbd/crm-fence-peer.sh";    
                split-brain "/usr/lib/drbd/notify-split-brain.sh root";    
                out-of-sync "/usr/lib/drbd/notify-out-of-sync.sh root";    
                before-resync-target "/usr/lib/drbd/snapshot-resync-target-lvm.sh -p 15 -- -c 16k";    
                after-resync-target /usr/lib/drbd/unsnapshot-resync-target-lvm.sh;   
}   
        startup {   
                wfc-timeout  120;   
                degr-wfc-timeout 120;   
         }     
        disk {   
                  on-io-error detach;

                 fencing resource-only;

          }   
        net {   
                cram-hmac-alg "sha1";   
                shared-secret  "mydrbdlab";   
         }   
        syncer {   
                  rate  100M;   
         }

}

由于我们在/etc/drbd.d/global\_common.conf配置文件中开启了资源隔离和脑列处理机制，所以在crm的配置文件cib中将会自动出现一个位置约束配置，当主节点宕机之后，禁止从节点变为主节点，以免当主节点恢复的时候产生脑裂，进行资源争用，但是我们此时只是为了验证资源能够流转，所以将这个位置约束删除：

[root@www1 drbd.d]# crm configure edit

[image](http://img1.51cto.com/attachment/201210/28/5635432_1351416651zGAU.png)

两台都要删除这两行！

node www1.gjp.com \   
        attributes standby="on"     
node www2.gjp.com \   
        attributes standby="off"   
primitive drbd\_web\_FS ocf:heartbeat:Filesystem \   
        params device="/dev/drbd0" directory="/web" fstype="ext3"   
primitive httpd\_drbd\_web ocf:heartbeat:drbd \   
        params drbd\_resource="web" \   
        op monitor interval="60s" role="Master" timeout="40s" \   
        op monitor interval="70s" role="Slave" timeout="40s"   
primitive webip ocf:heartbeat:IPaddr \   
        params ip="192.168.2.66"   
primitive webserver lsb:httpd   
group web webip webserver   
ms MS\_Webdrbd httpd\_drbd\_web \   
        meta master-max="1" master-node-max="1" clone-max="2" clone-node-max="1" notify="true"   
colocation drbd\_web\_FS\_on\_MS\_Webdrbd inf: drbd\_web\_FS MS\_Webdrbd:Master   
order drbd\_web\_FS\_after\_MS\_Webdrbd inf: MS\_Webdrbd:promote drbd\_web\_FS:start   
property $id="cib-bootstrap-options" \   
        dc-version="1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f" \   
        cluster-infrastructure="openais" \   
        expected-quorum-votes="2" \   
        stonith-enabled="false" \   
        no-quorum-policy="ignore"

[root@www1 drbd.d]# crm status   
============   
Last updated: Sun Oct 28 15:55:51 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
3 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

Resource Group: web   
     webip    (ocf::heartbeat:IPaddr):    Started www1.gjp.com   
     webserver    (lsb:httpd):    Started www1.gjp.com   
drbd\_web\_FS    (ocf::heartbeat:Filesystem):    Started www1.gjp.com   
Master/Slave Set: MS\_Webdrbd [httpd\_drbd\_web]   
     Masters: [ www1.gjp.com ]   
     Stopped: [ httpd\_drbd\_web:1 ]

[root@www1 drbd.d]# service drbd status

drbd driver loaded OK; device status:   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
m:res  cs            ro               ds                         p                 mounted  fstype   
0:web  WFConnection  Primary/Unknown  UpToDate/Outdated  C  /web     ext3

[root@www1 drbd.d]# crm status   
============   
Last updated: Sun Oct 28 16:08:38 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
3 Resources configured.   
============

Online: [ www1.gjp.com www2.gjp.com ]

Resource Group: web   
    webip    (ocf::heartbeat:IPaddr):    Started www1.gjp.com   
     webserver    (lsb:httpd):    Started www1.gjp.com   
drbd\_web\_FS    (ocf::heartbeat:Filesystem):    Started www2.gjp.com   
Master/Slave Set: MS\_Webdrbd [httpd\_drbd\_web]   
     Masters: [ www2.gjp.com ]   
     Slaves: [ www1.gjp.com ]

发现出现脑裂现象，解决如下：

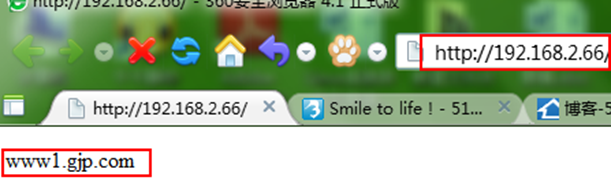
[image](http://img1.51cto.com/attachment/201210/28/5635432_13514166539qdU.png)

[root@www1 drbd.d]# watch -n 1 'crm status'

[](http://img1.51cto.com/attachment/201210/28/5635432_1351416658jU6o.png)

已可以同步：查看挂载点

[root@www1 drbd.d]# mount   
/dev/sda2 on / type ext3 (rw)   
proc on /proc type proc (rw)   
sysfs on /sys type sysfs (rw)   
devpts on /dev/pts type devpts (rw,gid=5,mode=620)   
/dev/sda1 on /boot type ext3 (rw)   
tmpfs on /dev/shm type tmpfs (rw)   
none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)   
sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw)   
/dev/drbd0 on /web type ext3 (rw)

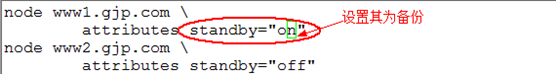
[](http://img1.51cto.com/attachment/201210/28/5635432_1351416670Mj87.png)

观看www2上的状态：

[root@www2 drbd.d]# mount   
/dev/sda2 on / type ext3 (rw)   
proc on /proc type proc (rw)   
sysfs on /sys type sysfs (rw)   
devpts on /dev/pts type devpts (rw,gid=5,mode=620)   
/dev/sda1 on /boot type ext3 (rw)   
tmpfs on /dev/shm type tmpfs (rw)   
none on /proc/sys/fs/binfmt\_misc type binfmt\_misc (rw)   
sunrpc on /var/lib/nfs/rpc\_pipefs type rpc\_pipefs (rw）

[root@www2 drbd.d]# service httpd status   
httpd is stopped   
[root@www2 drbd.d]# service drbd status   
drbd driver loaded OK; device status:   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.cntos.org, 2010-06-04 08:04:16   
m:res  cs          ro                 ds                 p      mounted  fstype   
0:web  StandAlone  Secondary/Unknown  UpToDate/Outdated  r----

模拟www1死掉了！

[](http://img1.51cto.com/attachment/201210/28/5635432_1351416678LPlT.png)

[root@www2 drbd.d]# crm status   
============   
Last updated: Sun Oct 28 17:25:27 2012   
Stack: openais   
Current DC: www1.gjp.com - partition with quorum   
Version: 1.1.5-1.1.el5-01e86afaaa6d4a8c4836f68df80ababd6ca3902f   
2 Nodes configured, 2 expected votes   
3 Resources configured.   
============

Node www1.gjp.com: standby   
Online: [ www2.gjp.com ]

Resource Group: web   
     webip    (ocf::heartbeat:IPaddr):    Started www2.gjp.com   
     webserver    (lsb:httpd):    Started www2.gjp.com   
Master/Slave Set: MS\_Webdrbd [httpd\_drbd\_web]   
     Masters: [ www2.gjp.com ]   
     Stopped: [ httpd\_drbd\_web:0 ]   
drbd\_web\_FS    (ocf::heartbeat:Filesystem):    Started www2.gjp.com   
[root@www2 drbd.d]# service httpd status   
httpd (pid  8509) is running...

[](http://img1.51cto.com/attachment/201210/28/5635432_1351416682gckP.png)

能够正常访问！

eth0      Link encap:Ethernet  HWaddr 00:0C:29:99:12:74    
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0   
          inet6 addr: fe80::20c:29ff:fe99:1274/64 Scope:Link   
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1   
          RX packets:192191 errors:0 dropped:0 overruns:0 frame:0   
          TX packets:103068 errors:0 dropped:0 overruns:0 carrier:0   
          collisions:0 txqueuelen:1000   
          RX bytes:121841514 (116.1 MiB)  TX bytes:13390418 (12.7 MiB)   
          Interrupt:67 Base address:0x2000

eth0:0    Link encap:Ethernet  HWaddr 00:0C:29:99:12:74    
          inet addr:192.168.2.66  Bcast:192.168.2.255  Mask:255.255.255.0   
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1   
          Interrupt:67 Base address:0x2000

lo        Link encap:Local Loopback    
          inet addr:127.0.0.1  Mask:255.0.0.0   
          inet6 addr: ::1/128 Scope:Host

[root@www2 drbd.d]# service drbd status   
drbd driver loaded OK; device status:   
version: 8.3.8 (api:88/proto:86-94)   
GIT-hash: d78846e52224fd00562f7c225bcc25b2d422321d build by mockbuild@builder10.centos.org, 2010-06-04 08:04:16   
m:res  cs          ro               ds                 p      mounted  fstype   
0:web  StandAlone  Primary/Unknown  UpToDate/Outdated  r----  ext3