1. **OS相关配置**

**系统版本Kylin-Server-3.3-4-1904-111440-x86\_64**

* 1. **修改IP、主机名和地址解析**

以两台集群为例：

VIP：192.168.8.5 网关：192.168.8.1

192.168.8.2 pcmk-1

192.168.8.3 pcmk-2

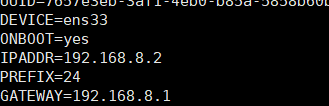
* 修改主机名

#vi /etc/hostname



* 修改IP地址和网关信息

# vi /etc/sysconfig/network-scripts/ifcfg-ens33



IP地址修改和网络重新加载可以用下面命令：

[root@pcmk-1 ~]# vi /etc/sysconfig/network-scripts/ifcfg-${device}

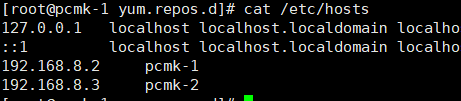
[root@pcmk-1 ~]# nmcli dev disconnect ${device}

[root@pcmk-1 ~]# nmcli con reload ${device}

[root@pcmk-1 ~]# nmcli con up ${device}

* 主机名地址解析

# vi /etc/hosts



* 1. **配置SSH**
* 创建和激活一个新的SSH KEY

[root@pcmk-1 ~]# ssh-keygen -t dsa -f ~/.ssh/id\_dsa -N ""

Generating public/private dsa key pair.

Your identification has been saved in /root/.ssh/id\_dsa.

Your public key has been saved in /root/.ssh/id\_dsa.pub.

The key fingerprint is:

91:09:5c:82:5a:6a:50:08:4e:b2:0c:62:de:cc:74:44 root@pcmk-1.clusterlabs.org

The key's randomart image is:

+--[ DSA 1024]----+

|==.ooEo.. |

|X O + .o o |

| \* A + |

| + . |

| . S |

| |

| |

| |

| |

+-----------------+

[root@pcmk-1 ~]# cp ~/.ssh/id\_dsa.pub ~/.ssh/authorized\_keys

* 在其它节点上安装KEY密钥

[root@pcmk-1 ~]# scp -r ~/.ssh pcmk-2:

The authenticity of host 'pcmk-2 (192.168.122.102)' can't be established.

ECDSA key fingerprint is SHA256:63xNPkPYq98rYznf3T9QYJAzlaGiAsSgFVNHOZjPWqc.

ECDSA key fingerprint is MD5:d9:bf:6e:32:88:be:47:3d:96:f1:96:27:65:05:0b:c3.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'pcmk-2,192.168.122.102' (ECDSA) to the list of known hosts.

root@pcmk-2's password:

id\_dsa

id\_dsa.pub

authorized\_keys

known\_hosts

* 测试SSH登录其它主机

[root@pcmk-1 ~]# ssh pcmk-2

pcmk-2

* 1. **关闭防火墙和SELINUX**

# systemctl status firewalld.service

# systemctl disable firewalld.service

# systemctl stop firewalld.service

# vim /etc/selinux/config

SELINUX=disabled

* 1. **时间同步**

1. node1作为服务器端；

# vim /etc/chrony.conf

注释掉server那4行；

allow 192.168.111/24

local stratum 9

# systemctl restart chronyd.service

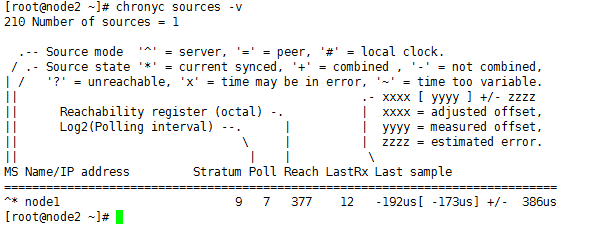
1. node2作为客户端；

# vim /etc/chrony.conf

注释掉server那4行；

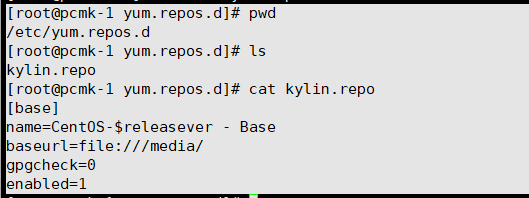
server 192.168.111.101 iburst

# systemctl restart chronyd.service



1. **集群初始设置**
   1. **安装集群软件**

* **配置yum源**



#yum clean all

#yum makecache

* 安装pacemaker 、 corosync 和pcs ，所有节点都安装

# yum install pacemaker pcs psmisc policycoreutils-python

* 1. **pcs初始配置**
* 启动pcsd服务和使它能开机自启

# systemctl start pcsd.service

# systemctl enable pcsd.service

* 安装pcs包后会创建一个hacluster的用户，需要为hacluster设置密码，各集群节点要设置相同的密码。

# passwd hacluster

Changing password for user hacluster.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

* 各节点做以上相同操作
  1. **corosync配置**
* 验证hacluster用户

[root@pcmk-1 ~]# pcs cluster auth pcmk-1 pcmk-2

Username: hacluster

Password:

pcmk-2: Authorized

pcmk-1: Authorized

* 生成和同步corosync配置

[root@pcmk-1 ~]# pcs cluster setup --name mycluster pcmk-1 pcmk-2

Destroying cluster on nodes: pcmk-1, pcmk-2...

pcmk-2: Stopping Cluster (pacemaker)...

pcmk-1: Stopping Cluster (pacemaker)...

pcmk-1: Successfully destroyed cluster

pcmk-2: Successfully destroyed cluster

Sending 'pacemaker\_remote authkey' to 'pcmk-1', 'pcmk-2'

pcmk-2: successful distribution of the file 'pacemaker\_remote authkey'

pcmk-1: successful distribution of the file 'pacemaker\_remote authkey'

Sending cluster config files to the nodes...

pcmk-1: Succeeded

pcmk-2: Succeeded

Synchronizing pcsd certificates on nodes pcmk-1, pcmk-2...

pcmk-2: Success

pcmk-1: Success

Restarting pcsd on the nodes in order to reload the certificates...

pcmk-2: Success

pcmk-1: Success

* 1. **启动集群服务和添加开机自启**
* 集群服务启动

# pcs cluster start --all

* 设置开机自启动

# pcs cluster enable --all

node1: Cluster Enabled

node2: Cluster Enabled

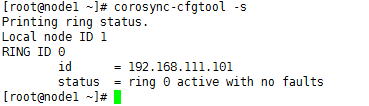
# systemctl list-unit-file可以看到corosync、pacemaker变为开机自启动；

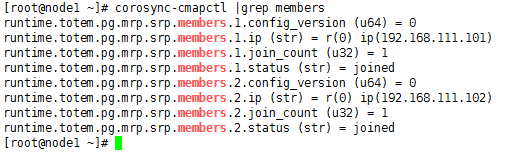
* 检查集群状态

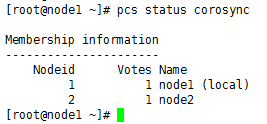
[root@pcmk-1 ~]# pcs cluster status

[root@pcmk-1 ~]# pcs status

* 1. **验证corosync安装和当前状态**







**3. 主备集群配置**

**3.1 添加VIP资源**

* 添加VIP资源，每30秒检查它一次是否运行。

#pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=192.168.8.5 cidr\_netmask=32 op monitor interval=30s

**注释:**

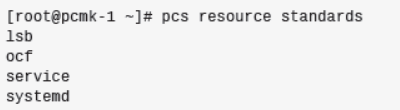
**ocf:heartbeat:IPaddr2**：

第一个字段(ocf)是资源脚本遵循的标准，以及在哪里可以找到它。

第二个字段(heartbeat)是标准的明确细化，对于ocf资源，它告诉集群资源脚本所在的ocf命名空间。

第三个字段(IPaddr2)是资源脚本的名称。

**可用的资源标准（第一字段）：**



**可用的资源提供者（第二字段）**

[root@pcmk-1 ~]# pcs resource providers

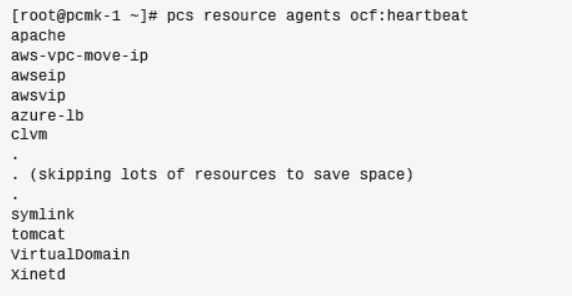
heartbeat

openstack

pacemaker

**可用的资源提供者（第三字段）**

Ocf资源提供者heartbeat提供的所有可用的资源代理



* 1. **添加mariadb资源**

pcs resource create mariadb ocf:pacemaker:mariadb op monitor interval=20s timeout=120s op start interval=0s timeout=120s op stop interval=0s timeout=120s

**3.3添加默认超时时间**

[root@pcmk-1 ~]# pcs resource op defaults timeout=240s

Warning: Defaults do not apply to resources which override them with their own defined values

[root@pcmk-1 ~]# pcs resource op defaults

timeout: 240s

**3.4没有fence设备禁用stonith**

# pcs property set stonith-enabled=false

**3.5 quorum仲裁机制**

# pcs property set no-quorum-policy=ignore

**3.6 添加Drbd存储**

* 各主机节点添加 5G硬盘sdb；
* 安装drbd包

CentOS 7 yum源安装方法：

下载更新elrepo yum源包

# rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org

# rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-3.el7.elrepo.noarch.rpm

安装Drbd kernel module 和utilities

# yum install -y kmod-drbd84 drbd84-utils

Kylin rpm包安装方法：

# rpm -ivh drbd90-utils-9.0.0-1.ky3.kb1.x86\_64.rpm

# rpm -ivh kmod-drbd90-9.0.9-1.ky3.kb1.x86\_64.rpm

* 配置 drbd

# vim /etc/drbd.d/mydata.res

resource mydata {

on pcmk-1 {

device /dev/drbd0;

disk /dev/sdb;

address 192.168.8.7:7789;

meta-disk internal;

}

on pcmk-2 {

device /dev/drbd0;

disk /dev/sdb;

address 192.168.8.8:7789;

meta-disk internal;

}

}

* Drbd初始化

创建metadata,所有节点上执行如下命令：

# drbdadm create-md mydata

* 确认DRBD kernel module是否加载，所有节点上执行如下命令：

[root@pcmk-1 drbd.d]# modprobe drbd

[root@pcmk-1 drbd.d]# lsmod | grep drbd

drbd 397041 0

libcrc32c 12644 4 xfs,drbd,nf\_nat,nf\_conntrack

* 启动drbd，所有节点上执行如下命令：

# systemctl start drbd 或者

#drbdadm up mydata

* 查看drbd 状态

# drbdadm status

如下命令也可查看状态；

# drbd-overview

* 在pcmk-1上执行如下命令，将它强制为主；

# drbdadm primary mydata --force

执行完毕，两台节点会进行数据同步，同步完成后如下所示：

[root@pcmk-1 ~]# cat /proc/drbd

version: 8.4.11-1 (api:1/proto:86-101)

GIT-hash: 66145a308421e9c124ec391a7848ac20203bb03c build by mockbuild@, 2018-04-26 12:10:42

1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----

ns:524236 nr:0 dw:0 dr:526364 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0

两边都显示UpToData;

* 在主节点上格式化创建文件系统

# mkfs.xfs /dev/drbd0

* 测试drbd，在主节点上执行：

//挂载drbd分区

[root@pcmk-1 drbd.d]# mount /dev/drbd0 /mnt/

//在分区中写入文件

[root@pcmk-1 mnt]# cp /var/www/html/index.html /mnt/

**3.7添加drbd资源**

* 在两个节点上停止drbd服务

# systemctl stop drbd

# umount /dev/drbd0

* 添加drbd设备资源

#pcs resource create DrbdDevice ocf:linbit:drbd drbd\_resource=mydata op monitor interval=60S

* 添加drbd设备clone资源；单主模式确保drbd服务在主机上

#pcs resource master DrbdDevice\_Clone DrbdDevice master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true

* 添加drbd文件系统挂载资源

# pcs resource create DrbdFS Filesystem device="/dev/drbd0" directory="/var/www/html" fstype="xfs"

Assumed agent name 'ocf:heartbeat:Filesystem' (deduced from 'Filesystem')

* 设置关联关系和限制条件

#pcs constraint colocation add DrbdFS with DrbdDevice\_Clone INFINITY with-rsc-role=Master

#pcs constraint order promote DrbdDevice\_Clone then start DrbdFS

with-rsc-role=Master: 约束的附加属性;这意味着此约束与主克隆相关联

**3.8添加资源组**

pcs resource group add my\_res VIP cups mysql httpd

pcs resource create rc-local systemd:rc-local --group my\_res

添加资源时直接将资源添加进资源组，在添加资源时直接按照顺序添加即可

**3.6添加资源粘性**

pcs constraint colocation add my\_res with DrbdFS INFINITY

**3.7添加资源启动顺序**

pcs constraint order DrbdFS then start my\_res

**附件A：pcs常用设置命令**

1. 设置资源在同一个节点上运行：

# pcs constraint colocation add WebSite with VIP INFINITY

1. 设置资源启动顺序：

# pcs constraint order VIP then WebSite

1. 没有fencing设备禁用stonith：

# pcs property set stonith-enabled=false

1. 两个节点忽略quorum，否则单节点不过半，无法提供服务：

# pcs property set no-quorum-policy=ignore

1. 设置资源粘性（防止在恢复后资源迁移）：

# pcs property set default-resource-stickiness="INFINITY"

1. 设置全局默认超时时间：

# pcs resource op defaults timeout=100s

1. 设置pingd

pcs resource create pingd ocf:pacemaker:ping params host\_list="192.168.189.2" multiplier="100" op monitor interval="10s" timeout=20s op start interval="0" timeout="90s" op stop interval="0" timeout="100s"

配置如下

Clone: PingCheck-clone

Resource: PingCheck (class=ocf provider=pacemaker type=ping)

Attributes: dampen=5s multiplier=100 host\_list="192.168.189.131 router"

Operations: start interval=0s timeout=60 (PingCheck-start-interval-0s)

stop interval=0s timeout=20 (PingCheck-stop-interval-0s)

monitor interval=30s timeout=10s (PingCheck-monitor-interval-30s)

1. 编辑某个资源配置：

# pcs resource update VIP op monitor interval=60s

1. 查看服务支持哪种类型：

pcs resource list | grep 服务名

1. 删除某个资源

pcs resource delete VIP

1. 删除约束资源或绑定资源(remove后面跟id，id使用pcs config或pcs constraint --full查看)：

pcs constraint remove colocation-cups-DrbdFS-INFINITY

pcs constraint remove order-DrbdFS-cups-mandatory

1. 添加资源组，my\_res为资源组名，后面为资源名

pcs resource group add my\_res VIP DrbdFS cups httpd mysql

移除资源组pcs resource group remove my\_res

1. 清除指定资源的状态与错误计数

pcs resource cleanup 资源名(httpd)

1. heartbeat资源配置中IPaddr和IPaddr2的区别

主要区别是IPaddr支持LVS，而IPaddr2和LVS有冲突