数据安全工具DRBD,extundelete

# 数据镜像软件DRBD

## 数据镜像软件DRDB介绍

分布式块设备复制(Distributed Replicated Block Device)，是一种基于软件的，基于网络的块复制存储解决方案，主要用于磁盘，分区进行数据镜像。

### DRBD基本功能

DRBD的核心功能是数据镜像，通过网络来镜像整个磁盘设备或者磁盘分区。对于DRBD数据镜像有如下特点：实时性，透明性，同步镜像，异步镜像。

### DRBD的构成

DRBD具体来说分为两个部分。一部分是内核模板，主要用于虚拟一个块设备；一部分是用户空间管理程序，主要用于和ＤＲＢＤ内核模块进行通讯，管理DRDB资源。一个DRBD设备有两个以上节点构成，分为主用节点和备用节点两个角色，在主用节点上，DRBD设备有读写操作的权限，可以初始化，创建和挂载文件系统。在备用节点上，DRBD设备无法挂在，只能用来接收主用节点发送过来的数据，也就是备用节点不能用于读写，只是用来保证数据缓冲区的一致性。

### DRBD的主要特性

单主模式

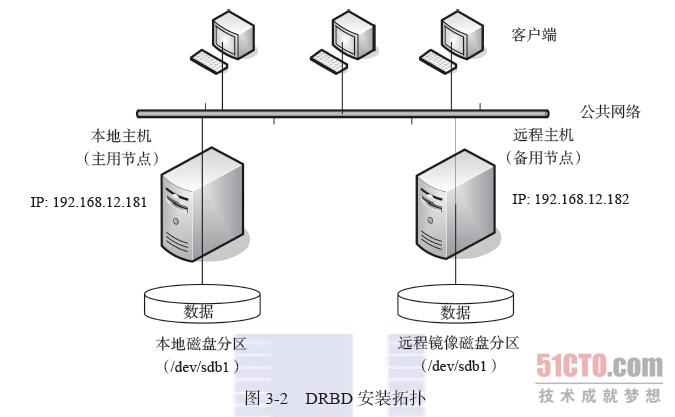
双主模式

复制模式(A,B,C协议)

传输完整性校验

脑裂通知和自动修复

## DRBD的安装和配置



### 安装DRBD

部署前说明：从DRBD 8.4.5开始的版本发生了变化，不再和以前一样只安装DRBD软件包就可以了，可能是从liunx 内核 2.6.33 开始DRBD模块被集成到内核中去了，所以从DRBD 8.4.5开始需要编译安装的软件为drbd和drbd-utils。drbd用于编译安装模块，drbd-utils用于管理drbd模块，并提供配置文件。

|  |
| --- |
| #准备安装环境  yum install -y make automake kernel kernel-devel kernel-headers gcc flex libxslt  #编译安装drbd  wget http://oss.linbit.com/drbd/8.4/drbd-8.4.6.tar.gz -P /usr/local/src/  cd /usr/local/src  tar -zxvf drbd-8.4.6.tar.gz  cd drbd-8.4.6  make KDIR=/usr/src/kernels/$(uname -r)/  make install  #编译成功模块安装位置  modprobe -l | grep -i drbd |
| #加载模块  modprobe drbd  lsmod | grep drbd |
| # 编译安装drbd-utils  #安装成功后drbd相关的工具(drbdadm,drbdsetup)被安装到/usr/local/drbd-utils-8.9.3/etc/sbin目录下  wget <http://oss.linbit.com/drbd/drbd-utils-8.9.3.tar.gz> -P /usr/local/src/  cd /usr/local/src  tar -zxvf drbd-utils-8.9.3.tar.gz  cd drbd-utils-8.9.3  ./configure --prefix=/usr/local/drbd --without-83support  make && make install  cp /usr/local/drbd/etc/rc.d/init.d/drbd /etc/rc.d/init.d/  chkconfig --add drbd  chkconfig --level 2345 drbd on  #链接drbd的命令到系统命令路径  ln -s   /usr/local/drbd/sbin/\*  /usr/bin/ |
| #centos7中安装方法，yum安装默认路径是/etc/drbd.d/下  rpm --import http://elrepo.org/RPM-GPG-KEY-elrepo.org  rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm  yum -y install drbd84-utils kmod-drbd84 |

### 配置DRBD

vi /usr/local/drbd/etc/drbd.conf

|  |
| --- |
| # drbd.conf  global {  usage-count no; #是否参加DRBD使用者统计，默认参加  }  common {  syncer { rate 200M; } #设置主用节点和备用节点同步时的网络速率最大值，单位是字节    }  resource r0 { #资源名字为r0  protocol C; #使用DRBD的第三种同步协议，表示收到远程主机的写入确认后认为写入完成  handlers {  pri-on-incon-degr "echo o > /proc/sysrq-trigger ; halt -f";  pri-lost-after-sb "echo o > /proc/sysrq-trigger ; halt -f";  local-io-error "echo o > /proc/sysrq-trigger ; halt -f";  fence-peer "/usr/lib64/heartbeat/drbd-peer-outdater -t 5";  pri-lost "echo pri-lost. Have a look at the log files. | mail -s 'DRBD Alert' root";  split-brain "/usr/lib/drbd/notify-split-brain.sh root";  out-of-sync "/usr/lib/drbd/notify-out-of-sync.sh root";  }  net {  cram-hmac-alg "sha1"; # DRBD同步时使用的验证方式和密码信息  shared-secret "MySQL-HA";  }  disk { # 使用dpod功能（drbd outdate-peer daemon）保证在数  # 据不同步时不进行切换  on-io-error detach;  fencing resource-only;  }  startup {  wfc-timeout 120;  degr-wfc-timeout 120;  }  device /dev/drbd0;  on master-drbd { # 每个主机的说明以on开头，后面是hostname（uname -n），  # 再后面的{}中为这个主机的配置  disk /dev/sdb1; # /dev/drbd0使用的磁盘分区是/dev/sdb1  address 192.168.12.181:7788; #设置DRBD的监听端口，用于与另一台主机通信  meta-disk internal;  }  on slave-drbd {  disk /dev/sdb1; # /dev/drbd0使用的磁盘分区是/dev/sdb1  address 192.168.12.182:7788; #设置DRBD的监听端口，用于与另一台主机通信  meta-disk internal; # DRBD的元数据存放方式  }  } |
| #CentOS7中使用vi /etc/drbd.conf  global {  usage-count no;    }  common {  syncer { rate 200M; }    }  resource r0 {  protocol C;    net {  cram-hmac-alg "sha1";  shared-secret "p@$$w0rd";  }  disk {  on-io-error detach;  fencing resource-only;  }    startup {  wfc-timeout 120;  degr-wfc-timeout 120;  }    device /dev/drbd0;  on image\_master {  disk /dev/sdb;  address 172.16.11.249:7788;  meta-disk internal;  }  on image\_slave {  disk /dev/sdb;  address 172.16.11.250:7788;  meta-disk internal;  }  } |

## DRBD的管理和维护

### 启动DRBD

--在两个节点执行

drbdadm create-md r0



/etc/init.d/drbd start 或者systemctl start drbd

--查看节点状态

|  |
| --- |
| [root@master-drbd ~]# cat /proc/drbd  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:2007644   * ro表示角色信息，第一次启动drbd时，两个drbd节点默认都处于Secondary状态。 * ds表示磁盘状态信息，“Inconsistent/Inconsistent”即“不一致/不一致”状态，表示两个节点的磁盘数据处于不一致状态。 * ns表示网络发送的数据包信息。 * dw表示磁盘写信息。 * dr表示磁盘读信息。 |

--设置主节点

rbdadm -- --overwrite-data-of-peer primary all

centos7：drbdadm primary --force r0

第一次执行完此命令后，如果需要设置哪个是主用节点，就可以使用另外一个命令：/sbin/drbdadm primary r0或者/sbin/drbdadm primary all

--挂载DRBD

mkfs.ext4 /dev/drbd0

mount /dev/drbd0 /mnt

#完成挂载后，就可以在/mnt目录下写数据了，此目录下的数据会自动同步到#备用节点上。

### RDBD主备节点切换

|  |
| --- |
| 方式一：停止服务切换  # 停止DRBD服务(主节点)  /etc/init.d/drbd stop  #查看备用节点的DRBD状态，并将备用节点升级为主用节点  cat /proc/drbd  drbdsetup /dev/drbd0 primary –o  #新的主用节点上挂载DRBD设备，  mount /dev/drbd0 /mnt  方式二：正常切换  #主用节点卸载磁盘分区，并切换成备份节点  umount /mnt  drbdadm secondary all  #备用节点切换成主用节点  drbdadm primary all  mount /dev/drbd0 /mnt |

# 数据恢复软件extundelete

## extundelete的介绍和原理

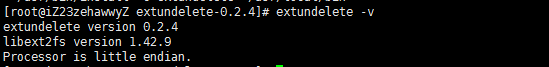
在Linux下，基于开源的数据恢复工具有很多，常见的有debugfs、R-Linux、ext3grep、extundelete等，比较常用的有ext3grep和extundelete，这两个工具的恢复原理基本一样，只是extundelete功能更加强大。

利用extundelete恢复文件时并不依赖特定文件格式，首先extundelete会通过文件系统的inode信息（根目录的inode一般为2）来获得当前文件系统下所有文件的信息，包括存在的和已经删除的文件，这些信息包括文件名和inode。然后利用inode信息结合日志去查询该inode所在的block位置，包括直接块，间接块等信息。最后利用dd命令将这些信息备份出来，从而恢复数据文件。



## 安装extundelete

|  |
| --- |
| yum install e2fsprogs\* -y  #yum install gcc gcc-c++ gcc-g77 –y  tar jxvf extundelete-0.2.4.tar.bz2  cd extundelete-0.2.4  ./configure  make && make install |



## extundelete用法解析

用法如下：

extundelete --help

命令格式: extundelete [options] [action] device-file

其中参数（options）有：

--version, -[vV]，显示软件版本号。

--help，显示软件帮助信息。

--superblock，显示超级块信息。

--journal，显示日志信息。

--after dtime，时间参数，表示在某段时间之后被删的文件或目录。

--before dtime，时间参数，表示在某段时间之前被删的文件或目录。

动作（action）有：

--inode ino，显示节点“ino”的信息。

--block blk，显示数据块“blk”的信息。

--restore-inode ino[,ino,...]，恢复命令参数，表示恢复节点“ino”的文件，恢复的文件会自动放在当前目录下的RESTORED\_FILES文件夹中，使用节点编号作为扩展名。

--restore-file 'path'，恢复命令参数，表示将恢复指定路径的文件，并把恢复的文件放在当前目录下的RECOVERED\_FILES目录中。

--restore-files 'path'，恢复命令参数，表示将恢复在路径中已列出的所有文件。

--restore-all，恢复命令参数，表示将尝试恢复所有目录和文件。

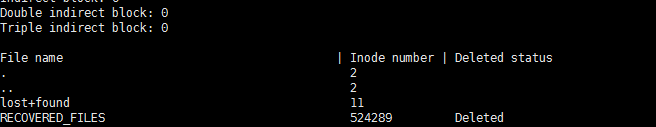
-j journal，表示从已经命名的文件中读取扩展日志。

-b blocknumber，表示使用之前备份的超级块来打开文件系统，一般用于查看现有超级块是不是当前所要的文件。

-B blocksize，表示使用数据块大小来打开文件系统，一般用于查看已经知道大小的文件。

## extundelete应用

###查询可恢复的数据信息

extundelete /dev/sdc1 --inode 2  


### extundelete恢复单个文件

extundelete /dev/sdc1 --restore-file passwd

### extundelete恢复单个目录

extundelete /dev/sdc1 --restore-directory /ganglia-3.4.0

### extundelete恢复所有误删除数据

extundelete /dev/sdc1 --restore-all

### extundelete恢复某个时间段的数据

extundelete --after 1379146740 --restore-all /dev/sdc1

在文件恢复成功后，extundelete命令默认会在执行命令的当前目录下创建一个RECOVERED\_FILES目录，此目录用于存放恢复出来的文件，所以执行extundelete命令的当前目录必须是可写的。