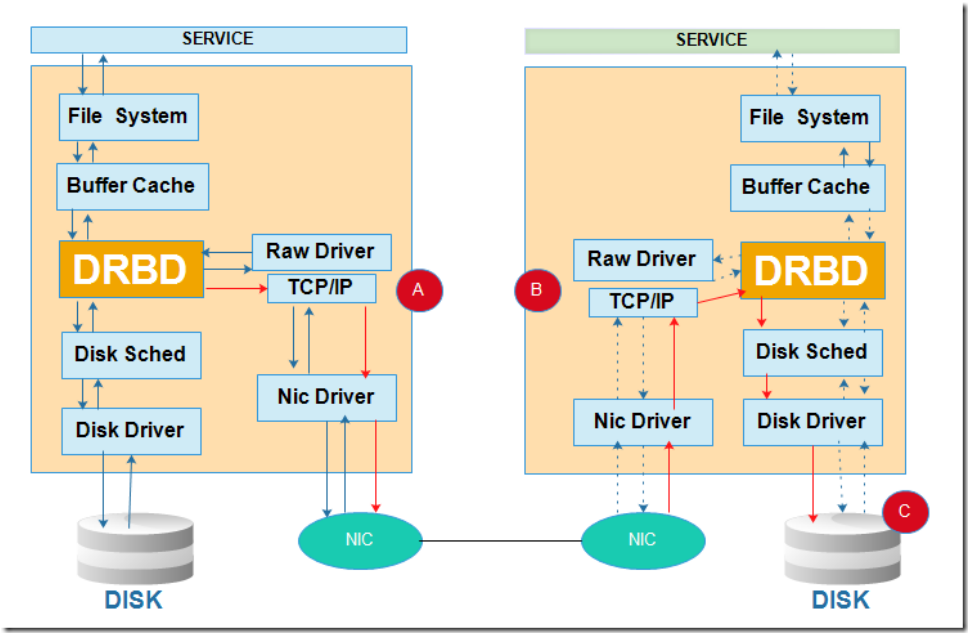
羊木狼

DRBD原理知识

DRBD(Distributed Relicated Block Device 分布式复制块设备)， 可以解决磁盘单点故障。一般情况下只支持2个节点。

大致工作原理如下图：



一般情况下文件写入磁盘的步骤是: 写操作 --> 文件系统 --> 内存缓存中 --> 磁盘调度器 --> 磁盘驱动器 --> 写入磁盘。而DRBD的工作机制如上图所示，数据经过buffer cache后有内核中的DRBD模块通过tcp/ip协议栈经过网卡和对方建立数据同步。

一、DRBD的工作模式

1、主从模型master/slave（primary/secondary）

这种机制，在某一时刻只允许有一个主节点。主节点的作用是可以挂在使用，写入数据等；从节点知识作为主节点的镜像，是主节点的备份。

这样的工作机制的好处是可以有效的避免磁盘出现单点故障，不会文件系统的错乱。

2、双主模型 dula primary(primary/primary)

所谓双主模型是2个节点都可以当做主节点来挂载使用。那么，思考这样一个问题？当第一个主节点对某一文件正在执行写操作，此时另一个节点也正在对同一文件也要执行写操作，结果会如何呢？？

一般这种情况会造成文件系统的错乱，导致数据不能正常使用。原因是：对文件的加速机制是由操作系统内核所管理的，一个节点对文件加速之后，另一个节点并不知道对方的锁信息。

公告

昵称：羊木狼
园龄：4年6个月
粉丝：0
关注：0
+加关注

< 2019年2月 >

日	一	二	三	四	五	六
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	1	2
3	4	5	6	7	8	9

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

随笔分类

集群

随笔档案

2014年9月 (3)

阅读排行榜

1. DRBD原理知识(4202)
2. 自动化运维工具之ansible(361)
3. Linux HA集群(246)

解决办法是：使用集群文件系统。集群文件系统使用分布式文件锁管理器，当一个节点对文件加锁之后会通过某种机制来通知其他节点锁信息，从而实现文件锁共享。

二、DRBD的复制模型

当某一进程对某一文件执行了写操作时，写操作在上图执行到那个过程时就认为文件已经同步完成。

复制协议：

A协议：异步复制（asynchronous）如上图 文件写操作执行到A点是就认为写入磁盘成功。性能好，数据可靠性差。

B协议：半同步复制（semi sync）如上图 文件写操作执行到B点是就认为写入磁盘成功。性能好，数据可靠性介于A和C之间。

C协议：同步复制（ sync）如上图 文件写操作执行到C点是就认为写入磁盘成功。性能差，数据可靠性高。也是drbd默认使用的复制协议

三、drbd的配置（主从模式）

实验环境：

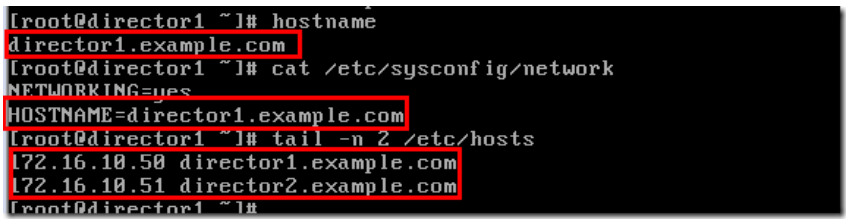
2个节点：

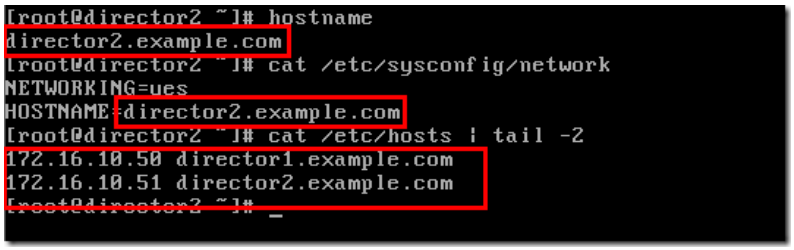
172.16.10.50 director1.example.com

172.16.10.51 director2.example.com

1、准备工作

```
# drbd 2个节点之间通信是基于主机名的
# 设置主机名和主机名解析文件
```






```
# 准备好大小相同的磁盘，这里使用大小相同的分区代替。只需划好分区就好，不需要格式化。
```

```
<a href="https://images0.cnblogs.com/blog/660139/201409/161935518317816.png"><img
style="display: inline; border-width: 0px;" title="5"
src="https://images0.cnblogs.com/blog/660139/201409/161935524711986.png" alt="5"
width="530" height="145" border="0"></a>
```

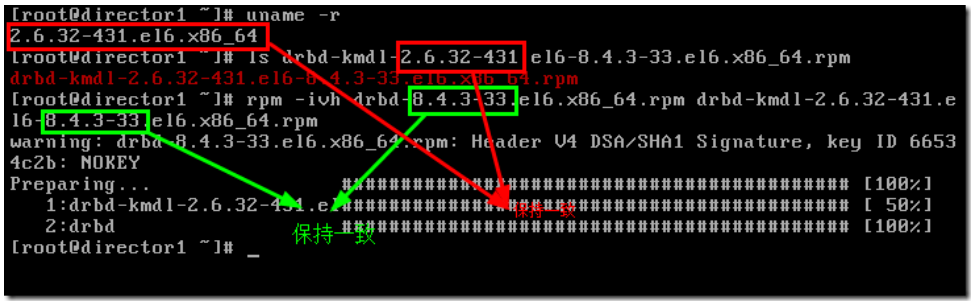
```
<a href="https://images0.cnblogs.com/blog/660139/201409/161935530966628.png"><img
style="display: inline; border-width: 0px;" title="6"
src="https://images0.cnblogs.com/blog/660139/201409/161935537687256.png" alt="6"
width="502" height="135" border="0"></a>
```

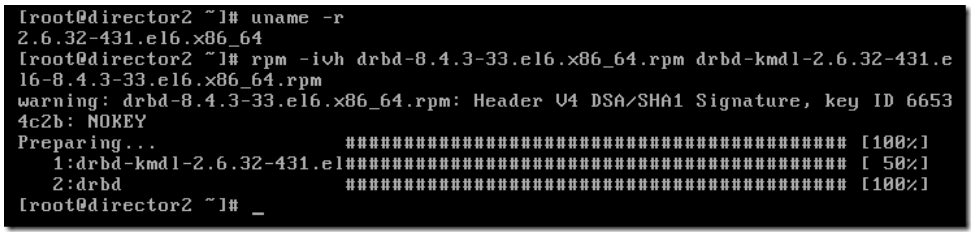
2、安装软件包

```
drbd共有两部分组成：内核模块和用户空间的管理工具。
其中drbd内核模块代码已经整合进Linux内核2.6.33以后的版本中，因此，如果内核版本高于此版本的话，
只需要安装管理工具即可；否则，您需要同时安装内核模块和管理工具两个软件包，并且此两者的版本号一定要保
持对应。
```

```
# 对应的内核模块的名字分别为 drbd-kmod
注意：
drbd和drbd-kmdl的版本要对应；另一个是drbd-kmdl的版本要与当前系统的内核版本(uname -r)相对应。
下载地址： <a href="http://www.rpmfind.net/linux/atrpms/el6-x86_64/atrpms/stable/">http://www.rpmfind.net/linux/atrpms/el6-x86_64/atrpms/stable/

</a>下载完成后，直接安装即可。
```





3、配置drbd

配置文件说明：

```
drbd的主配置文件为/etc/drbd.conf；为了管理的便捷性，目前通常会将些配置文件分成多个部分，且都保存至/etc/drbd.d/目录中，
主配置文件中仅使用"include"指令将这些配置文件片断整合起来。通常，/etc/drbd.d目录中的配置文件为global_common.conf和所有以.res结尾的文件。
其中global_common.conf中主要定义global段和common段，而每一个.res的文件用于定义一个资源。

在配置文件中，global段仅能出现一次，且如果所有的配置信息都保存至同一个配置文件中而不分开为多个文件的话，global段必须位于配置文件的最开始处。
目前global段中可以定义的参数仅有minor-count, dialog-refresh, disable-ip-verification和usage-count。

common段则用于定义被每一个资源默认继承的参数，可以在资源定义中使用的参数都可以在common段中定义。
实际应用中，common段并非必须，但建议将多个资源共享的参数定义为common段中的参数以降低配置文件的复杂度。

resource段则用于定义drbd资源，每个资源通常定义在一个单独的位于/etc/drbd.d目录中的以.res结尾的文件中。
资源在定义时必须为其命名，名字可以由非空白的ASCII字符组成。
每一个资源段的定义中至少要包含两个host子段，以定义此资源关联至的节点，其它参数均可以从common段或drbd的默认中进行继承而无须定义。
```

配置过程：

```
###############下面的操作在director1.example.com上完成。

1 配置/etc/drbd.d/global-common.conf
global {
        usage-count no; # 是否为drbd官方收集数据
        # minor-count dialog-refresh disable-ip-verification
}
# common是各个资源共用的选项
common {
        protocol C; # 复制协议

        handlers {
```

```
                pri-on-incon-degr "/usr/lib/drbd/notify-pri-on-incon-degr.sh;
/usr/lib/drbd/notify-emergency-reboot.sh; echo b > /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 > /proc/sysrq-trigger ; reboot -f";
                local-io-error "/usr/lib/drbd/notify-io-error.sh;
/usr/lib/drbd/notify-emergency-shutdown.sh; echo o > /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; # 发生i/o错误的处理方法，detach将镜像磁盘直接拔除
                #fencing resource-only;
        }

        net {
                cram-hmac-alg "sha1";
                shared-secret "mydrbdlab";
        }

        syncer {
                rate 1000M;
        }
}

2、定义一个资源/etc/drbd.d/test.res，内容如下：
resource test {
  on director1.example.com {
    device    /dev/drbd0;
    disk      /dev/sda3;
    address   172.16.10.50:7789;
    meta-disk internal;
  }
  on director2.example.com {
    device    /dev/drbd0;
    disk      /dev/sda3;
    address   172.16.10.51:7789;
    meta-disk internal;
  }
}
```

以上文件在两个节点上必须相同，因此，可以基于ssh将刚才配置的文件全部同步至另外一个节点。

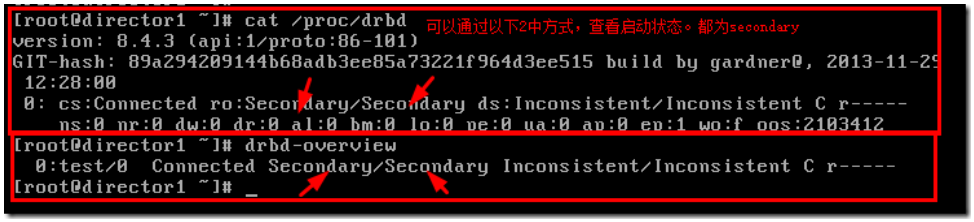
```
scp  /etc/drbd.d/*  director.example.com:/etc/drbd.d
```

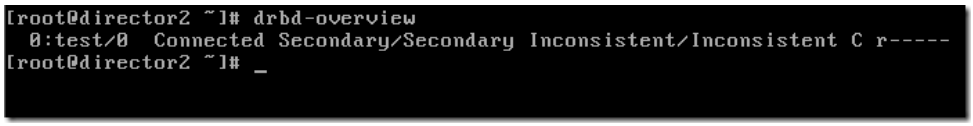
在两个节点上初始化已定义的资源并启动服务

```
1）初始化资源，在 director1 和 director2上分别执行：
drbdadm create-md test

2）启动服务，在 director1 和 director2 上分别执行：
/etc/init.d/drbd start
```

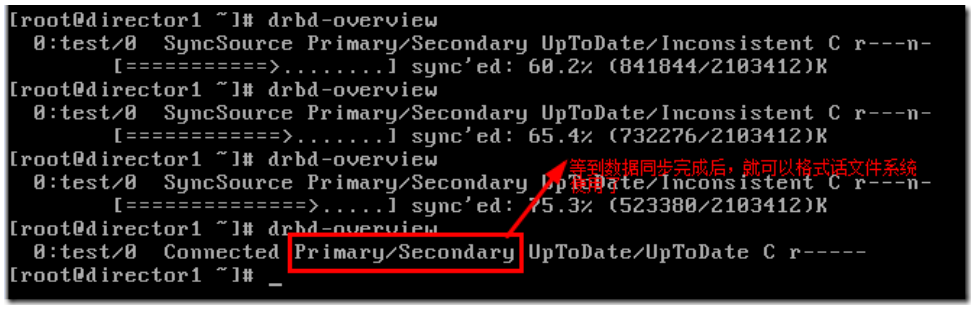
完成以上2步骤后，查看启动状态：





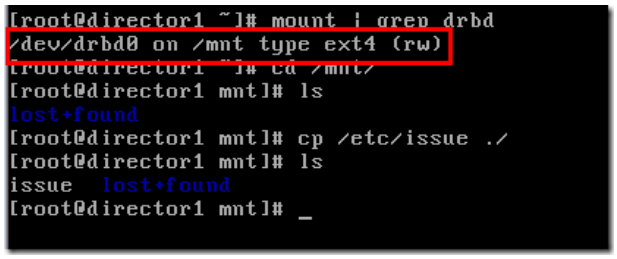
完成以上操作后，继续下面操作。同步metadata(元数据)

```
# 将director1.example.com 节点设置为Primary。在要设置为Primary的节点上执行如下命令：
drbdadm primary --force test
```



接下来创建文件系统，挂载使用

```
mke2fs -t ext4 -L DRBD /dev/drbd0
mount /dev/drbd0 /mnt/
```

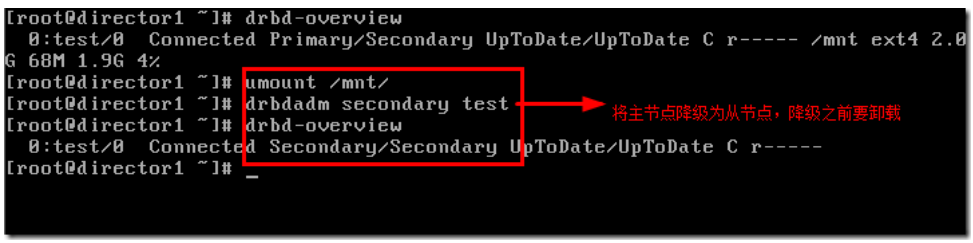


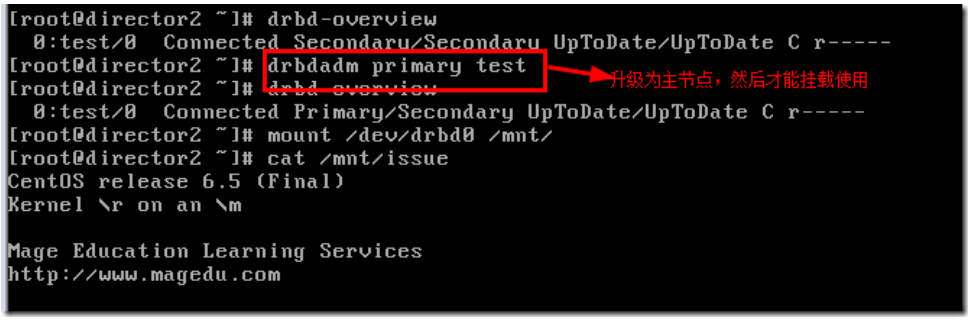
配置完成。

三、主从节点切换

drbd主从模型只有主节点才能挂载使用。所以就会有升级降级的操作。对主Primary/Secondary模型的drbd服务来讲，在某个时刻只能有一个节点为Primary，因此，要切换两个节点的角色，只能在先将原有的Primary节点设置为Secondary后，才能原来的Secondary节点设置为Primary。

具体使用如下：





这样的切换需手动升级，降级。通常drbd会于HA一起使用来达到自动切换的效果，此时drbd是HA的一种clone资源。

drbd的双主模型，需借助于集群文件系统，在以后会详细介绍。

好文要顶 关注我 收藏该文

羊木狼
关注 - 0
粉丝 - 0
+加关注

0 0

« 上一篇：Linux HA集群
» 下一篇：自动化运维工具之ansible

posted @ 2014-09-16 19:36 羊木狼 阅读(4202) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!
【推荐】专业便捷的企业级代码托管服务 - Gitee 码云

相关博文:
· DRBD试用
· IsPostBack原理
· DRBD试用
· 十七 DRBD配置
· drbd中文应用指南

最新新闻：
· 世界卫生组织公布预防听力损伤新标准，对智能手机提出新要求
· 字节跳动的支付业务终上正轨，但“逐梦金融圈”谈何容易
· 为什么说你应该停更“双微一抖”
· “墨子号”科研团队获美国2018年度克利夫兰奖
· 苹果失去“美国人最亲密品牌”称号 迪斯尼取而代之
» 更多新闻...

Copyright ©2019 羊木狼