

LVM

2018年3月27日 23:41

1. 创建LVM

新建三块硬盘 (100GB * 3)

将物理设备 (可以是物理磁盘/dev/sdb、也可以是物理分区/dev/sdb1) 格式化为PV (物理卷)

```
[root@test ~]# fdisk -l
```

首先需要使用 pvcreate 命令将/dev/sdb1和/dev/sdc格式化成PV

```
[root@test ~]# pvcreate /dev/sdb /dev/sdd /dev/sdc
```

这里可以看到创建成功了, 其中pvcreate是创建命令, 后面参数是需要初始化的物理设备, 多个设备之间使用空格分隔。可以使用 pvdisplay 命令或者 pvs 命令查看已经存在的PV信息:

```
[root@test ~]# pvremove /dev/sdd # 如有必要可以使用pvremove删除物理卷
```

```
[root@test ~]# pvs
```

```
[root@test ~]# pvdisplay
```

```
[root@test ~]# pvs
PV          VG          Fmt  Attr  PSize   PFree
  /dev/sda2 centos_test lvm2 a--  <19.00g      0
  /dev/sdb          lvm2 ---  100.00g 100.00g
  /dev/sdc          lvm2 ---  100.00g 100.00g
```

可以看到有三个设备已经被初始化成了PV, 这里不需要关注/dev/sda2分区, 这是在安装系统时自动初始化的PV, 这里/dev/sdb磁盘和/dev/sdc磁盘是初始化的PV。既然创建了PV那么就需要VG (PV组) 了, 下面我们来使用 vgcreate 命令来创建VG (卷组)

```
[root@test ~]# vgcreate myvg /dev/sdb
```

其中vgcreate是创建命令myvg是这个VG组的名称, /dev/sdb1是指将这个已经初始化成PV的设备添加套这个卷组中, 如果需要添加多个设备使用空格分隔。我们可以使用 vgdisplay 或者 vgs 命令查看卷组信息:

```
[root@test ~]# vgdisplay
```

```
[root@test ~]# vgs
```

```
[root@test ~]# vgs
VG          #PV #LV #SN Attr   VSize   VFree
centos_test    1  2  0 wz--n- <19.00g      0
myvg           1  0  0 wz--n- <100.00g <100.00g
```

我们可以看到我们创建的名称为myvg的卷组, 大小为小于100G, 也就是说最大没有100G可用, PE大小是4M, 有25599个PE。

```
[root@test ~]# vgremove myvg # 如有需要可以使用 vgremove 命令删除卷组
```

创建逻辑卷了, 这里使用 lvcreate -n 逻辑卷名称 -l 逻辑卷PE数 卷组名 命令来创建逻辑卷:

```
[root@test ~]# lvcreate -n mylv -l 25599 myvg
```

在这里可以看到名称为mylv的逻辑卷创建成功, 其中 -l 选项可以换成 -L, 用来只用以磁盘大小为单位的数值, 比如说K、M、G、T等。但是这里的VG可用大小是5119PE和小于20.0G, 使用G做单位不知道具体PE数小于多少, 不好给定一个具体的值, 所以这里我们使用PE作为单位。接下来我们可以使用 lvdisplay 或者 lvs 命令查看逻辑卷信息

```
[root@test ~]# lvs
```

```
[root@test ~]# lvs
LV VG          Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sy
nc Convert
  root centos_test -wi-ao--- <17.00g
  swap centos_test -wi-ao---  2.00g
  mylv myvg      -wi-a----- <100.00g
```

```
[root@test ~]# lvdisplay
```

可以看到具体的逻辑卷信息了，创建的这个逻辑卷在操作系统中映射的文件的绝对路径为“/dev/myvg/mylv”，但是一般逻辑卷会在“/dev/mapper”目录下面创建一个软连接“/dev/mapper/myvg-mylv”，软连接名称为卷组名称再加上逻辑卷名称。如果有必要我们可以使用 `lvremove /dev/myvg/mylv` 命令删除逻辑卷。这个逻辑卷和物理分区一样，需要先格式化成合适的文件系统，然后挂载到某一个目录上就可以了，格式化分区和挂载分区与上面 `fdisk` 分区中的方法一样

```
[root@test ~]# mkfs.xfs /dev/myvg/mylv
```

2. 增加LVM大小

[root@test mapper]# `lvresize -L +10G /dev/myvg/mylv` # 添加逻辑卷，在初始化的时候只给了100g，现在再给100G（多次执行上述命令）

```
[root@test xl]# xfs_growfs /dev/myvg/mylv
```

```
[root@test mapper]# lvresize -L +10G /dev/myvg/mylv
Insufficient free space: 2560 extents needed, but only 2559 available
# 如果由于逻辑卷空间满了需要扩充，我们可以使用 vgs 命令查看还有没有可供逻辑卷扩充的空间
```

```
[root@test mapper]# vgs
VG          #PV #LV #SN Attr   VSize   VFree
centos_test 1   2   0 wz--n-   <19.00g    0
myvg        2   1   0 wz--n-  199.99g <10.00g
```

此时需要扩展vg了，先使用 `vgcreate` 初始化硬盘或者分区

```
[root@test mapper]# pvcreate /dev/sdd
```

```
[root@test mapper]# vgextend myvg /dev/sdd # 扩展vg卷组
```

```
[root@test mapper]# vgs
VG          #PV #LV #SN Attr   VSize   VFree
centos_test 1   2   0 wz--n-   <19.00g    0
myvg        3   1   0 wz--n- <299.99g 109.99g
```

```
[root@test mapper]# lvextend -l +5119 /dev/myvg/mylv #扩展lv
```

```
[root@test mapper]# lvs
LV VG          Attr      LSize   Pool Origin Data%  Meta%  Move Log Cpy%Syn
c Convert
root centos_test -wi-ao---- <17.00g
swap centos_test -wi-ao----    2.00g
mylv myvg        -wi-ao---- 249.98g
```

```
[root@test mapper]# df -hT
文件系统 类型 容量 已用 可用 已用% 挂载点
/dev/mapper/centos_test-root xfs 17G 4.4G 13G 26% /
devtmpfs devtmpfs 894M 0 894M 0% /dev
tmpfs tmpfs 910M 0 910M 0% /dev/shm
tmpfs tmpfs 910M 11M 900M 2% /run
tmpfs tmpfs 910M 0 910M 0% /sys/fs/cgroup
/dev/sdal xfs 1014M 179M 836M 18% /boot
tmpfs tmpfs 182M 28K 182M 1% /run/user/0
/dev/mapper/myvg-mylv xfs 230G 33M 230G 1% /mnt
```

扩充逻辑卷成功，但是使用 `df -hT` 命令查看，就会发现逻辑卷大小并没有发生变化，其实这是因为逻辑卷大小虽然扩充了，但是逻辑卷上面的文件系统并没有更新，所以需要先更新文件系统才能真正使用到扩充后的空间。注意这里使用 `xfs_growfs /dev/myvg/mylv` 命令更新一下文件系统，不能重新格式化整个分区的文件系统

```
[root@test mapper]# xfs_growfs /dev/myvg/mylv
```

```
[root@test mapper]# df -hT
文件系统 类型 容量 已用 可用 已用% 挂载点
/dev/mapper/centos_test-root xfs 17G 4.4G 13G 26% /
devtmpfs devtmpfs 894M 0 894M 0% /dev
tmpfs tmpfs 910M 0 910M 0% /dev/shm
tmpfs tmpfs 910M 11M 900M 2% /run
tmpfs tmpfs 910M 0 910M 0% /sys/fs/cgroup
/dev/sdal xfs 1014M 179M 836M 18% /boot
tmpfs tmpfs 182M 28K 182M 1% /run/user/0
/dev/mapper/myvg-mylv xfs 250G 33M 250G 1% /mnt
```

可以看到，说明扩充分区成功了。这里需要注意，由于用的是xfs的文件系统，所以使用 `xfs_growfs` 命令来更新文件系统，如果是ext2/ext3/ext4等文件系统则需使用 `resize2fs` 命令来更新文件系统了。

3. 减小LVM大小

在平时我们不只是需要扩充逻辑卷，还有可能需要收缩（减小）或者卸载逻辑卷，注意xfs文件系统只支持增大分区空间的情况，不支持减小的情况，硬要减小的话，只能在减小后将逻辑分区重新通过mkfs.xfs命令重新格式化才能挂载上，这样的话这个逻辑分区上原来的数据就丢失了。

但是ext文件系统可以支持减小减小逻辑卷操作，接下来我们做ext收缩逻辑卷操作。对逻辑卷进行收缩操作之前，如果逻辑卷已经挂载到了目录上必须先卸载逻辑卷的挂载，然后缩小文件系统，最后才是缩小逻辑卷，而且收缩的大小也不能超过剩余空间大小。

卸载了逻辑卷的挂载之后，需要先收缩文件系统，这一步一定是要在收缩逻辑卷之前操作，在这之前我已经将逻辑卷格式化成了ext4，所以这里我们使用 `resize2fs` 来执行收缩操作：

```
[root@test ~]# umount /dev/myvg/mylv
[root@test ~]# resize2fs /dev/myvg/mylv 30G
[root@test ~]# e2fsck -f /dev/vg0/lv0 对磁盘检测

[root@test ~]# lvreduce -L 30G /dev/myvg/mylv
[root@test ~]# mount /dev/myvg/mylv /mnt/
[root@test ~]# df -hT
```