

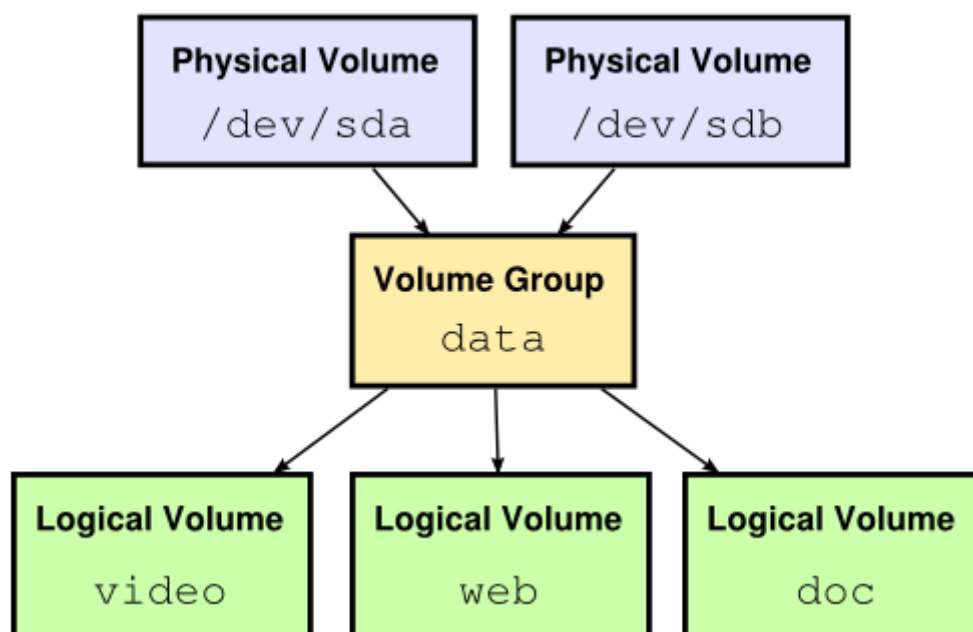
Linux逻辑卷LVM管理基础

逻辑卷管理器 **LVM(Logical Volume Manger)** 是一种存储管理系统，通俗的讲LVM就是将几个实体磁盘或者分区通过软件的方式组成起来，抽象到更高的层次。在更高的层次上，我们看到的是一个独立的虚拟大磁盘，在这个虚拟的大磁盘上，我们可以任意的创建分区和格式化，像使用传统分区一样使用通过LVM创建的分区。

LVM原理

LVM层次结构

LVM利用 Linux 内核 **device-mapper** 模块来实现存储的虚拟化，用户空间的工具集 **LVM2** 是 Linux 中新版本的管理工具，可通过yum安装lvm2包来安装LVM管理工具集。LVM通过将实体磁盘或者分区标记为LVM可识别的 **物理卷(PV,Physical volume)**，然后将物理卷组合封装成 **卷组(VG,volume group)**，卷组再分割成提供存储的块 **逻辑卷(LV,Logical volume)**。卷组以 **物理区段(PE,Phsical Extent)** 为块单位向逻辑卷分配容量。内核驱动模块md对设备进行映射，并在/dev/目录下创建相关的设备文件后，就可以进行格式化挂载使用。LVM层次结构如下：



相关概念

1. **物理卷(Physical volume)**: 物理卷是物理的磁盘或者分区。在使用物理卷, 必须将磁盘或分区初始化为物理卷, 初始化的过程就是给它打一个标签, 这个标签通常保存在设备的第512个字节处, 初始化的过程还会生成物理卷的唯一表示UUID以及元数据信息。
2. **卷组(Volume group)**: 将物理卷封装组合在一起成为卷组, 可以理解成卷组是一个存储池。卷组将存储空间划分成固定的块向上提供存储空间, 这个块称为物理区段(PE, Physical Extent), 物理区段PE的大小默认的4M。
3. **逻辑卷(LV, Logical Volume)**: 将物理卷加入卷组之后, 就可以创建任意大小的逻辑卷了, 创建好的逻辑卷可以格式化成任意文件系统并将它挂载使用。

LVM相对于直接使用物理存储有以下优点:

1. 可以动态扩展或缩小逻辑卷的大小, 而不需要重新创建文件系统。通常缩小逻辑卷有一定的风险, 有可能会造成数据丢失。
2. 可以将几个容量相对较小的磁盘通过LVM虚拟成一个大容量的磁盘。
3. 支持对逻辑卷做快照。

LVM逻辑卷管理

创建逻辑卷先将使用 `pvcreate` 命令将磁盘或者物理卷初始为LVM可识别的物理卷, 如果是初始化分区为物理卷, 则需要修改分区的分区格式为 `Linux LVM`, 对应十六进制为8e。

```

1 Command (m for help): t
2 Selected partition 1
3 Hex code (type L to list all codes): 8e      # 将分区
   类型修改为8e
4 Changed type of partition 'Linux' to 'Linux LVM'
5 Command (m for help): p
6 Disk /dev/sdb: 21.5 GB, 21474836480 bytes, 41943040
   sectors
7 Units = sectors of 1 * 512 = 512 bytes
8 Sector size (logical/physical): 512 bytes / 512
   bytes
9 I/O size (minimum/optimal): 512 bytes / 512 bytes
10 Disk label type: dos
11 Disk identifier: 0x257e1377
12      Device Boot          Start            End          Blocks
   Id  System
13 /dev/sdb1                2048          4196351        2097152
   8e  Linux LVM      # 修改后的分区类型

```

将磁盘或者分区初始化为物理卷后，使用 `vgcreate` 命令将物理卷组合成卷组，最后使用 `lvcreate` 命令在卷组中创建逻辑卷，即可以格式化挂载使用逻辑卷，可以在逻辑卷上存储数据或者安装程序。

创建分区

```

1 [ root@node0 ~ ]# lsblk /dev/sdb
2 NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
3 sdb         8:16   0   20G  0 disk
4 └─sdb1      8:17   0    2G  0 part
5 └─sdb2      8:18   0    2G  0 part
6 └─sdb3      8:19   0    2G  0 part

```

创建物理卷

`pvccreate` 命令使用格式：`pvccreate [OPTION] /dev/DEVICE`

```
1 # 选项
2 -y # 自动回答为yes
3 -f # 强制创建，还可以使用-ff
4 -u # 指定uuid
```

```
1 # 将sdb1, sdb2初始化为物理卷
2 [ root@node0 ~ ]# pvcreate -y /dev/sdb{1,2}
3 Physical volume "/dev/sdb1" successfully created.
4 Physical volume "/dev/sdb2" successfully created.
```

物理卷的查看可以使用：`pvdisk`显示详细信息；`pvs`显示简要信息；`pvs`扫描并显示物理卷信息。

```
1 # 显示所有物理卷
2 [ root@node0 ~ ]# pvdisk
3 "/dev/sdb1" is a new physical volume of "2.00 GiB"
4 --- NEW Physical volume ---
5 PV Name          /dev/sdb1
6 VG Name
7 PV Size          2.00 GiB
8 Allocatable      NO
9 PE Size          0
10 Total PE         0
11 Free PE          0
12 Allocated PE     0
13 PV UUID          SOF8S8-xARP-CIZ1-F1qh-Xx0M-BSN5-CgFqEZ
14 "/dev/sdb2" is a new physical volume of "2.00 GiB"
15 --- NEW Physical volume ---
16 PV Name          /dev/sdb2
17 VG Name
18 PV Size          2.00 GiB
19 Allocatable      NO
20 PE Size          0
21 Total PE         0
22 Free PE          0
23 Allocated PE     0
```

```

24 PV UUID PVUMzY-iCwC-cjNK-N2r1-Dzzk-
    P9rH-9cnG8
25 # 查看指定物理卷
26 pvdisplay /dev/sdb1

```

以上信息显示将大小都为2G的sdb1和sdb2初始为物理卷，目前为止物理卷还不属于任何卷组，接下来就可以将物理卷组合为卷组。

创建卷组

使用 `vgcreate` 命令创建卷组。命令使用格式：`vgcreate -s PE_Size N[bBsSkKmMgGtTpPeE] VG_Name PV_Path`，选项-s指定PE的大小。

```

1 # 创建名称为VG-0的卷组
2 [ root@node0 ~ ]# vgcreate -s 8M VG-0 /dev/sdb{1,2}
3 volume group "VG-0" successfully created

```

使用 `vgdisplay` 命令查看卷组VG-0的信息

```

1 [ root@node0 ~ ]# vgdisplay -v VG-0      # -v或者-vv
    显示更加详细的信息
2 --- volume group ---
3 VG Name VG-0
4 System ID
5 Format lvm2
6 Metadata Areas 2
7 Metadata Sequence No 1
8 VG Access read/write
9 VG Status resizable
10 MAX LV 0
11 Cur LV 0
12 Open LV 0
13 Max PV 0
14 Cur PV 2
15 Act PV 2
16 VG Size 3.98 GiB
17 PE Size 8.00 MiB

```

```

18 Total PE 510
19 Alloc PE / Size 0 / 0
20 Free PE / Size 510 / 3.98 GiB
21 VG UUID EJY6ZQ-5hHu-erc0-DokI-aIlJ-
ETmn-I7YE4A
22 --- Physical volumes ---
23 PV Name /dev/sdb1
24 PV UUID SOf8S8-xARP-CIz1-F1qh-Xx0M-
BSN5-CgFqEZ
25 PV Status allocatable
26 Total PE / Free PE 255 / 255
27 PV Name /dev/sdb2
28 PV UUID PVUMzY-iCwC-cjNK-N2r1-Dzzk-
P9rH-9cnG80
29 PV Status allocatable
30 Total PE / Free PE 255 / 255

```

创建逻辑卷

创建逻辑卷的过程类似于从磁盘分区的过程。是用 `lvcreate` 命令创建逻辑卷。

命令使用格式: `lvcreate -[L|l|n] LV_NAME VG_NAME`

```

1 # 选项含义
2 -L|--size
   LogicalVolumeSize[b|B|s|S|k|K|m|M|g|G|t|T|p|P|e|E]
   # 指定大小
3 -l          # 以PE个数进行划分大小
4 -l 50%vg    # 按剩余容量的百分比进行划分逻辑卷大小
5 -n|--name LogicalVolume # 指定逻辑卷名称

```

在VG-0卷组中分别创建大小为1G，名称为LV-0的逻辑卷和大小为150个PE，名称为LV-1的逻辑卷。

```

1 [ root@node0 ~ ]# lvcreate -L 1G -n LV-0 VG-0
2   Logical volume "LV-0" created.
3 [ root@node0 ~ ]# lvcreate -l 150 -n LV-1 VG-0
4   Logical volume "LV-1" created.

```

使用 `lvdisplay` 对逻辑卷查看

```

1 [ root@node0 ~ ]# lvdisplay /dev/VG-0/LV-1
2   --- Logical volume ---
3   LV Path                /dev/VG-0/LV-1
4   LV Name                 LV-1
5   VG Name                 VG-0
6   LV UUID                 3mds0J-S2Dk-GZeP-jeaC-25cd-
hvqj-H2Rjkw
7   LV Write Access         read/write
8   LV Creation host, time node0.xuejinwei.com, 2017-
08-14 06:04:08 +0800
9   LV Status                available
10  # open                   0
11  LV Size                  1.17 GiB   # 逻辑卷中大小
12  Current LE               150
13  Segments                 1
14  Allocation                inherit
15  Read ahead sectors        auto
16  - currently set to       8192
17  Block device              253:1

```

查看生成的设备文件

在创建VG和LV的时，指定名称后，内核模块可以将这些名称映射成设备文件。

```

1 # 生成的设备文件
2 /dev/mapper/VG--0-LV--0 # 它有一个链接文件是/dev/VG-
0/LV-0
3 /dev/mapper/VG--0-LV--1 # 它的链接文件是/dev/VG-0/LV-1

```

薛晋玮 2017年3月4日