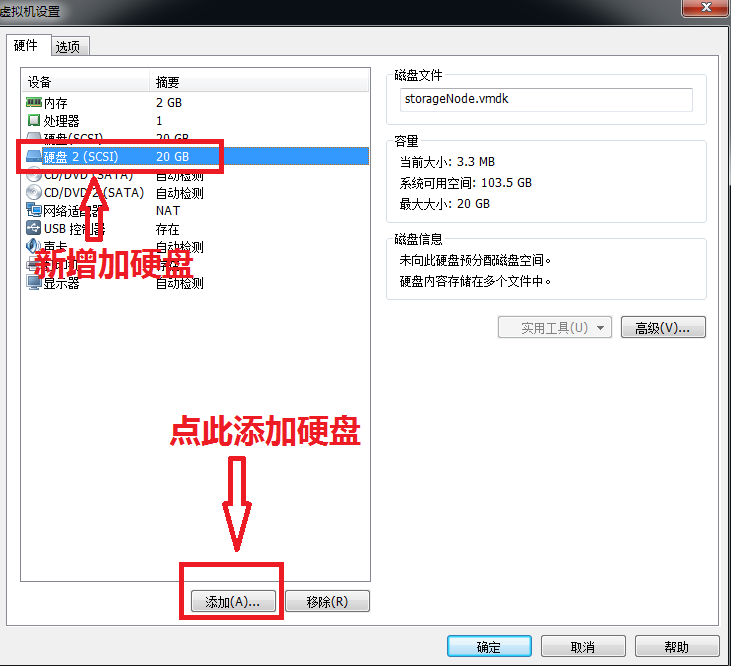
**阅读目录**

* [简单介绍](https://www.cnblogs.com/ssslinppp/p/5853312.html#_introduction)
* [章节1：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter1)创建系统分区
* [章节2：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter2)创建PV
* [章节3：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter3)创建VG
* [章节4：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter4)创建LV-->格式化LV-->挂载LV
* [章节5：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter5)LV扩容
* [章节6：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter6)VG扩容
* [章节7：](https://www.cnblogs.com/ssslinppp/p/5853312.html#_chapter7)参考链接

对于上面的 /dev/sdb1是如何来的那。这里由于使用的是虚拟机，是通过虚拟机增加的第二块硬盘：如下图：

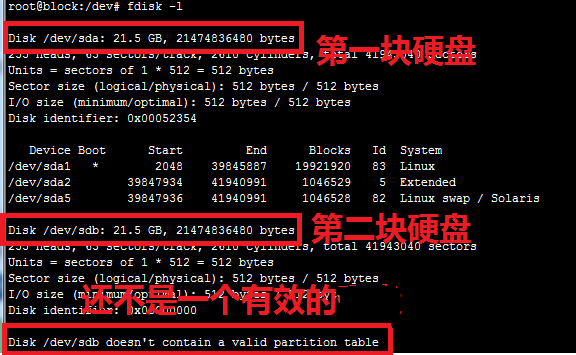


增加硬盘之后，我们通过

1. fdisk -l

*复制代码*

命令，可以查看到



所以我们通过

下面命令，即可完成：

1. fdisk /dev/sdb1
2. #Type in the followings:
3. n
4. p
5. 1
6. ENTER
7. ENTER
8. t
9. 8e
10. w

*复制代码*

**简单介绍**

LVM相关概念介绍请自行Google，这里仅给出简单的介绍：

**基本的逻辑卷管理概念：**

* PV（Physical Volume）- 物理卷

物理卷在逻辑卷管理中处于最底层，它可以是实际物理硬盘上的分区，也可以是整个物理硬盘，也可以是raid设备。

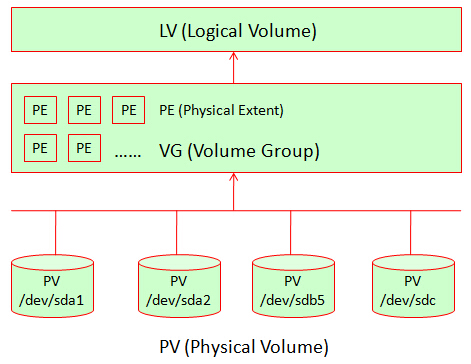
* VG（Volumne Group）- 卷组

卷组建立在物理卷之上，一个卷组中至少要包括一个物理卷，在卷组建立之后可动态添加物理卷到卷组中。一个逻辑卷管理系统工程中可以只有一个卷组，也可以拥有多个卷组。

* LV（Logical Volume）- 逻辑卷

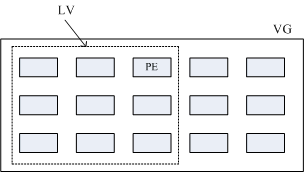
逻辑卷建立在卷组之上，卷组中的未分配空间可以用于建立新的逻辑卷，逻辑卷建立后可以动态地扩展和缩小空间。系统中的多个逻辑卷可以属于同一个卷组，也可以属于不同的多个卷组。

关系图如下：



* PE（Physical Extent）- 物理块

LVM 默认使用4MB的PE区块，而LVM的LV最多仅能含有65534个PE (lvm1 的格式)，因此默认的LVM的LV最大容量为4M\*65534/(1024M/G)=256G。PE是整个LVM 最小的储存区块，也就是说，其实我们的资料都是由写入PE 来处理的。简单的说，这个PE 就有点像文件系统里面的block 大小。所以调整PE 会影响到LVM 的最大容量！不过，在 CentOS 6.x 以后，由于直接使用 lvm2 的各项格式功能，因此这个限制已经不存在了。



***创建LVM系统分区方式的主要步骤如下：***

1. 1. 添加新的物理磁盘
2. 2. 创建新的分区，最多可分配【3个主分区，1个扩展分区】 或 【4个主分区】，扩展分区又可以有多个分区；
3. window常见的分配方式：
4. 方式1：[1个主分区（C盘）+1个扩展分区(包括3个分区，D，E，F盘)]
5. 方式2：[4个主分区，分别作为：C、D、E、F盘]
6. 若是使用LVM方式分区，分区时最好选择LVM分区方式；
7. 3. 创建PV;
8. 4. 创建VG；
9. 5. 创建LV；
10. 6. 格式化LV，ext3，ext4等格式；
11. 7. 挂载LV到指定目录；
12. 8. 将挂载信息写入：/etc/fstab;
13. 9. LVM扩容操作
14. 9.1 LV的扩容；
15. (1) 使用VG的Free空间，对LV扩容；
16. (2) 完成后需要在对文件系统扩容；
17. 9.2 VG的扩容
18. 方式1： 通过增加PV来实现
19. 方式2： 通过扩展raid（磁盘阵列）设备的容量来间接对VG扩容
20. 10. LVM缩减操作
21. 10.1 LV的缩减
22. 10.2 VG的缩减
23. 11. 删除LVM

其他说明：

凡是以“#”开头的行，都表示linux指令，如：

1. # fdisk -l

表示的是：

1. [root@localhost ~]# fdisk -l

如下是最初的系统基本信息：

操作系统：Centos7；

1. [root@localhost ~]# df -h
2. Filesystem               Size  Used Avail Use% Mounted on
3. /dev/mapper/centos-root   50G  1.3G   46G   3% /
4. devtmpfs                 911M     0  911M   0% /dev
5. tmpfs                    921M     0  921M   0% /dev/shm
6. tmpfs                    921M  8.5M  912M   1% /run
7. tmpfs                    921M     0  921M   0% /sys/fs/cgroup
8. /dev/sda1                477M  142M  310M  32% /boot
9. /dev/mapper/centos-home   47G  251M   44G   1% /home
10. tmpfs                    185M     0  185M   0% /run/user/0
11. [root@localhost ~]# fdisk -l
12. Disk /dev/sda: 107.4 GB, 107374182400 bytes, 209715200 sectors
14. Device Boot      Start         End      Blocks   Id  System
15. /dev/sda1   \*        2048     1026047      512000   83  Linux
16. /dev/sda2         1026048   209592319   104283136   8e  Linux LVM
17. --------LVM分区测试时，主要使用了sd{b,c,d,e,f,g}等----
18. Disk /dev/sdb: 10.7 GB, 10737418240 bytes, 20971520 sectors
19. Disk /dev/sdc: 10.7 GB, 10737418240 bytes, 20971520 sectors
20. Disk /dev/sdd: 16.1 GB, 16106127360 bytes, 31457280 sectors
21. Disk /dev/sde: 16.1 GB, 16106127360 bytes, 31457280 sectors
22. Disk /dev/sdf: 16.1 GB, 16106127360 bytes, 31457280 sectors
23. Disk /dev/sdg: 10.7 GB, 10737418240 bytes, 20971520 sectors
24. -----下面是系统安装后存在的，为方便说明，后面将其显示省略-----
25. Disk /dev/mapper/centos-root: 53.7 GB, 53687091200 bytes, 104857600 sectors
26. Disk /dev/mapper/centos-swap: 2147 MB, 2147483648 bytes, 4194304 sectors
27. Disk /dev/mapper/centos-home: 50.9 GB, 50944016384 bytes, 99500032 sectors

说明：

上面这些是安装完系统时就存在的，为方便后续的描述，在进行“df -h”时，省略掉了这些，其他指令也有部分省略。

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**一、创建系统分区**

选择/dev/sdb创建分区；

将整个sdb磁盘划分为一个sdb1分区，格式为Linux LVM；

由于显示信息过长，这里省略了部分不重要的显示信息。

1. # fdisk /dev/sdb
2. Command (m for help): n (创建新分区)
3. Partition type:
4. p   primary (0 primary, 0 extended, 4 free)
5. e   extended
6. Select (default p): p （选择创建主分区）
7. Partition number (1-4, default 1): 1 （分区号选择1）
8. First sector (2048-31457279, default 2048):
9. Using default value 2048
10. Last sector, +sectors or +size{K,M,G} (2048-31457279, default 31457279):
11. Using default value 31457279
12. Partition 1 of type Linux and of size 15 GiB is set
14. Command (m for help): p （查看已经创建的分区）
15. Disk /dev/sdb: 16.1 GB, 16106127360 bytes, 31457280 sectors
16. Device Boot      Start         End      Blocks   Id  System
17. /dev/sdb1            2048    31457279    15727616   83  Linux （系统默认为83，即：Linux）
19. Command (m for help): t （修改分区格式）
20. Selected partition 1
21. Hex code (type L to list all codes): 8e （8e为Linux LVM格式）
22. Changed type of partition 'Linux' to 'Linux LVM'
24. Command (m for help): p
25. Disk /dev/sdb: 16.1 GB, 16106127360 bytes, 31457280 sectors
26. Device Boot      Start         End      Blocks   Id  System
27. /dev/sdb1            2048    31457279    15727616   8e  Linux LVM
29. Command (m for help): w

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**二、创建PV**

首先说明一些简单指令，常见的查看指令包括如下三种：

1. pv相关：pvdisplay、pvs、pvscan，pvcreate

2. vg相关：vgdisplay、vgs、vgscan，vgcreate

3. lv相关：lvdisplay、lvs、lvscan，lvcreate

1. # pvcreate /dev/sdb1 （创建PV）
2. # pvdisplay
3. "/dev/sdb1" is a new physical volume of "10.00 GiB"
4. --- NEW Physical volume ---
5. PV Name               /dev/sdb1
6. VG Name
7. PV Size               10.00 GiB
8. Allocatable           NO
9. PE Size               0
10. Total PE              0
11. Free PE               0
12. Allocated PE          0
13. PV UUID               dZRM2W-X2Lz-I3Jz-i4S4-E8fN-BF4K-oZqhd3
14. # pvs
15. PV         VG     Fmt  Attr PSize  PFree
16. /dev/sdb1         lvm2 ---  10.00g 10.00g
17. # pvscan
18. PV /dev/sdb1               lvm2 [10.00 GiB]

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**三、创建VG**

创建VG时，可以使用如下几种方式来创建

1. raid：磁盘阵列；
2. 分区，如：/dev/sdb1;
3. 整个磁盘，如：/dev/sdc;

由于我们事先没有准备好磁盘阵列RAID，所以这里仅仅演示使用 ***分区***和***整个磁盘***来创建VG。

下面将使用“分区/dev/sdb1” 和 “磁盘/dev/sdc” 创建VG，并设置***PE大小为64M***。

1. # vgcreate vg0 /dev/sdb1 /dev/sdc -s 64M
2. Physical volume "/dev/sdc" successfully created
3. Volume group "vg0" successfully created
4. *使用分区/dev/sdb1 和 磁盘/dev/sdc 来创建vg，并设置PE大小为64M*

# pvs

  PV         VG     Fmt  Attr PSize  PFree

  /dev/sdb1  vg0    lvm2 a--  10.00g 10.00g

  /dev/sdc   vg0    lvm2 a--  10.00g 10.00g

1. # vgs
2. VG     #PV #LV #SN Attr   VSize  VFree
3. vg0      2   0   0 wz--n- 19.99g 19.99g
4. *上面知道：/dev/sdb1和/dev/sdb都是10G，所以vg0的大小为19.99G；*
6. # vgdisplay vg0
7. --- Volume group ---
8. VG Name               vg0
9. System ID
10. Format                lvm2
11. Metadata Areas        2
12. Metadata Sequence No  1
13. VG Access             read/write
14. VG Status             resizable
15. MAX LV                0
16. Cur LV                0
17. Open LV               0
18. Max PV                0
19. Cur PV                2
20. Act PV                2
21. VG Size               19.99 GiB
22. PE Size               64.00 MiB
23. Total PE              320
24. Alloc PE / Size       0 / 0
25. Free  PE / Size       5118 / 19.99 GiB
26. VG UUID               m58K2q-eIsC-1Z3T-DySZ-cgKb-Sd3L-UbrW2z

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**四、创建LV-->格式化LV-->挂载LV**

关于LV的操作主要有3个步骤，分别为：

1. 创建LV；
2. 格式化LV，如：ext3，ext4等；
3. 挂载LV到指定目录下；

**使用vg0创建lv1，初始大小设置为8G**

1. # lvcreate -L 8G -n lv1 vg0
2. Logical volume "lv1" created.
3. *从vg0中分出8G的空间给lv1使用*
5. # vgs
6. VG     #PV #LV #SN Attr   VSize  VFree
7. vg0      2   1   0 wz--n- 19.99g 11.99g
8. *由于分出了8G给lv1，所以VFree之剩下：11.99G*
10. # lvs
11. LV   VG     Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
12. lv1  vg0    -wi-a-----  8.00g
13. # lvdisplay
14. --- Logical volume ---
15. LV Path                /dev/vg0/lv1
16. LV Name                lv1
17. VG Name                vg0
18. LV UUID                wevRbf-Iy3b-wPfa-mK4a-DFoq-A24w-8q5axT
19. LV Write Access        read/write
20. LV Creation host, time localhost.localdomain, 2016-09-08 09:58:34 +0800
21. LV Status              available
22. # open                 0
23. LV Size                8.00 GiB
24. Current LE             2048
25. Segments               1
26. Allocation             inherit
27. Read ahead sectors     auto
28. - currently set to     8192
29. Block device           253:3

**使用ext3文件系统对LV进行格式化**

1. # mkfs.ext3 /dev/vg0/lv1
2. mke2fs 1.42.9 (28-Dec-2013)
3. Filesystem label=
4. OS type: Linux
5. Block size=4096 (log=2)
6. Fragment size=4096 (log=2)
7. Stride=0 blocks, Stripe width=0 blocks
8. 524288 inodes, 2097152 blocks
9. 104857 blocks (5.00%) reserved for the super user
10. First data block=0
11. Maximum filesystem blocks=2147483648
12. 64 block groups
13. 32768 blocks per group, 32768 fragments per group
14. 8192 inodes per group
15. Superblock backups stored on blocks:
16. 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
18. Allocating group tables: done
19. Writing inode tables: done
20. Creating journal (32768 blocks): done
21. Writing superblocks and filesystem accounting information: done

**将lv1挂载到/mnt/lv1目录下**

1. # mkdir /mnt/lv1
2. # mount /dev/vg0/lv1 /mnt/lv1
3. # df -Th
4. Filesystem              Type      Size  Used Avail Use% Mounted on
5. /dev/mapper/vg0-lv1     ext3      7.8G   19M  7.4G   1% /mnt/lv1

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**五、LV扩容**

LV的扩容主要分2个步骤：

1. 使用VG的Free空间，对LV扩容；
2. 完成后需要在对文件系统扩容；

**查看vg的容量，考虑从vg中再分配一些空间给LV**

1. # vgs
2. VG     #PV #LV #SN Attr   VSize  VFree
3. vg0      2   1   0 wz--n- 19.99g 11.99g
4. *现在还有11.99G可用*
6. # lvextend -L +3G /dev/vg0/lv1
7. Size of logical volume vg0/lv1 changed from 8.00 GiB (2048 extents) to 11.00 GiB (2816 extents).
8. Logical volume lv1 successfully resized.
9. *将lv1添加3G容量*
11. # vgs
12. VG     #PV #LV #SN Attr   VSize  VFree
13. centos   1   3   0 wz--n- 99.45g 4.00m
14. vg0      2   1   0 wz--n- 19.99g 8.99g
15. *由原先的11.99G可用变为8.99G可用*
17. # lvs
18. LV   VG     Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
19. lv1  vg0    -wi-ao---- 11.00g
20. *由原先的8G增加到了11G*

**对文件系统进行扩容**

1. # df -Th
2. Filesystem              Type      Size  Used Avail Use% Mounted on
3. /dev/mapper/vg0-lv1     ext3      7.8G   19M  7.4G   1% /mnt/lv1
4. *可是发现实际容量并没有变化，还是7.8G*
5. *这时需要对文件系统进行扩容*

**下面是扩容指令：**

1. # resize2fs /dev/vg0/lv1
2. resize2fs 1.42.9 (28-Dec-2013)
3. Filesystem at /dev/vg0/lv1 is mounted on /mnt/lv1; on-line resizing required
4. old\_desc\_blocks = 1, new\_desc\_blocks = 1
5. The filesystem on /dev/vg0/lv1 is now 2883584 blocks long.
6. 对文件系统扩容
8. # df -Th
9. Filesystem              Type      Size  Used Avail Use% Mounted on
10. /dev/mapper/vg0-lv1     ext3       11G   21M   11G   1% /mnt/lv1
11. *发现此时lv1的容量变为了11G*

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**六、VG扩容**

主要有两种扩容方式：

* 方式1： 通过增加PV来实现
* 方式2： 通过扩展RAID（磁盘阵列）设备的容量来间接对VG扩容（这里不进行展示）

**<方式1>： 通过增加PV来实现**

使用/dev/sdd来创建一个PV

1. # pvcreate /dev/sdd
2. Physical volume "/dev/sdd" successfully created
3. *创建PV*
5. # pvs
6. PV         VG     Fmt  Attr PSize  PFree
7. /dev/sdb1  vg0    lvm2 a--  10.00g     0
8. /dev/sdc   vg0    lvm2 a--  10.00g  8.99g
9. /dev/sdd          lvm2 ---  15.00g 15.00g
10. *新增加的PV有15G*
12. # vgextend vg0 /dev/sdd
13. Volume group "vg0" successfully extended
14. *使用PV对VG扩容*
16. # vgs
17. VG     #PV #LV #SN Attr   VSize  VFree
18. vg0      3   1   0 wz--n- 34.99g 23.99g
19. *扩容后的vg0由19.99G变为了34.99G*

<方式2>： 通过扩展raid（磁盘阵列）设备的容量来间接对VG扩容

这里不进行阐述

[回到顶部](https://www.cnblogs.com/ssslinppp/p/5853312.html#_labelTop)

**七、参考链接**