

4.2. 偷懒的快照 snapshot

对于一个分布式文件系统来说，快照是一个比较重要的功能，而快照这个概念，在很多地方也能遇到，例如安装虚拟机用的 virtualbox 或者 vmware 这些软件，也可以给虚拟机做快照，但是具体的实现可能各不相同，那么 glusterfs 中的快照功能其实是有点“偷懒”的，下面就来了解一下。

4.2.1. lvm2 原理

首先 glusterfs 的快照需要 lvm2 功能的支持，也就是说，并不是一个普通的操作系统就可以进行创建快照的，下面来了解一下。

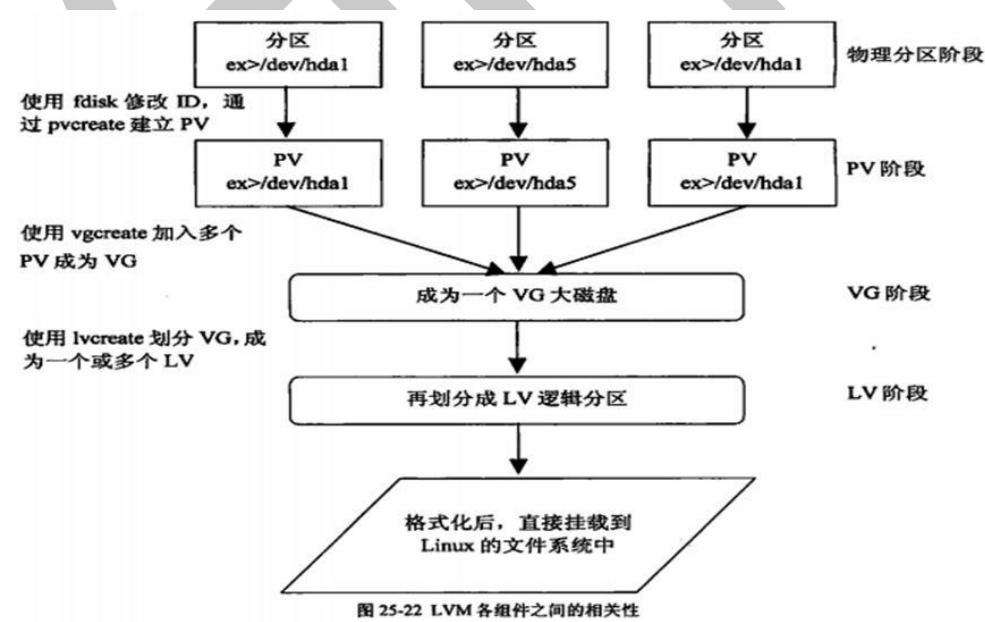
```
1. [root@gfs02 ~]# gluster volume info test-snapshot
2.
3. Volume Name: test-snapshot
4. Type: Replicate
5. Volume ID: 09c5d8b7-3c79-487e-8dc5-48b15db809ec
6. Status: Started
7. Snapshot Count: 0
8. Number of Bricks: 1 x 3 = 3
9. Transport-type: tcp
10. Bricks:
11. Brick1: 192.168.0.110:/glusterfs/test-snapshot
12. Brick2: 192.168.0.111:/glusterfs/test-snapshot
13. Brick3: 192.168.0.112:/glusterfs/test-snapshot
14. Options Reconfigured:
15. cluster.granular-entry-heal: on
16. storage.fips-mode-rchecksum: on
17. transport.address-family: inet
18. nfs.disable: on
19. performance.client-io-threads: off
20.
21.
22. [root@gfs02 ~]# gluster snapshot create test-snapshot-210
    616 test-snapshot no-timestamp
```

23. snapshot create: failed: Snapshot is supported only for thin provisioned LV. Ensure that all bricks of test-snapshot are thinly provisioned LV.
24. Snapshot command failed

这里提示快照是需要 lv 功能支持的,那么也就是 brick 的目录也是要 lvm2 格式的,而 lvm2 是 Linux 中一个功能,而 linux 中的 lvm2 主要是以下几点组成的。

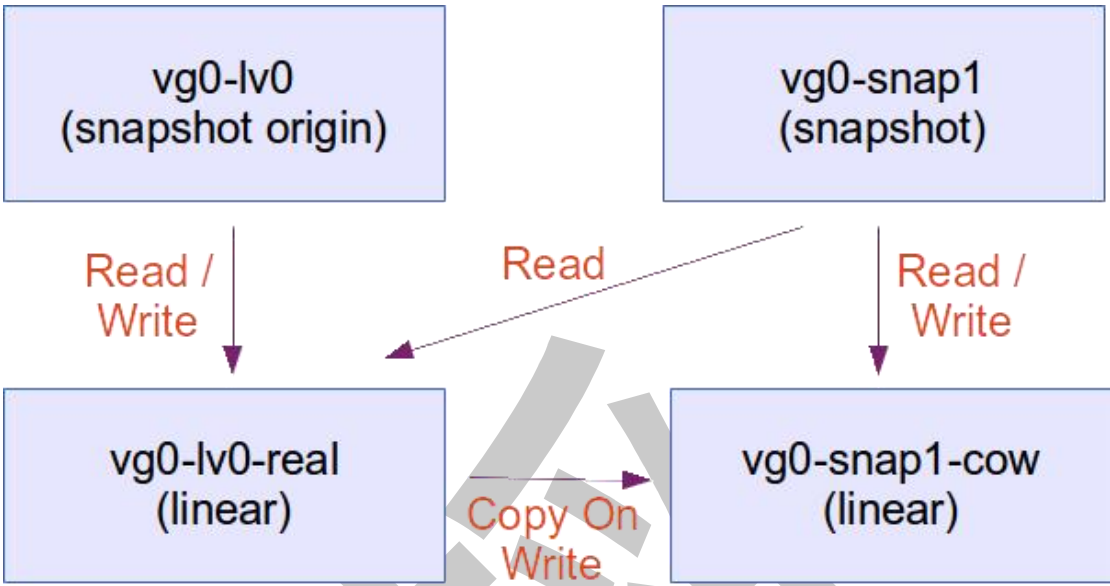
1. 将设备指定为物理卷
2. 用一个或者多个物理卷来创建一个卷组
3. 物理卷是用固定大小的物理区域 (Physical Extent, PE) 来定义的
4. 在物理卷上创建的逻辑卷是由物理区域 (PE) 组成
5. 可以在逻辑卷上创建文件系统

为了更好地理解一下 lvm 中的 pv,vg 和 lv 概念,可以用下面的图来理解。



那么这里 vg 和 lv 这些是如何组织的呢? 可以通过下图来理解一

下。



这里其实就是当创建一个快照之后，会创建一个新的目录位置，而新的数据则会指向到新的目录位置，但是读取旧数据的时候，还是从原来旧的目录进行读取。

4.2.2. lvm2 命令的使用

为了进一步理解这几个之间的关系，下面在客户端使用命令创建并且挂载 lvm2 目录。

```
1. [root@gfsclient01 ~]# lsblk
2. NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
3. sda                                  8:0    0  100G  0 disk
4. └─sda1                              8:1    0    1G  0 part /boot
5. └─sda2                              8:2    0   99G  0 part
```

```

6.    |centos-root 253:0    0   50G  0 lvm  /
7.    |centos-swap 253:1    0   3.9G  0 lvm  [SWAP]
8.    |centos-home 253:2    0  45.1G  0 lvm  /home
9.   sdb                8:16    0    5G  0 disk
10.  sr0                 11:0     1 1024M  0 rom
11.
12.

13. #创建 pv
14. [root@gfsclient01 ~]# pvcreate  --qq --metadatasize=512M
    --dataalignment=256K /dev/sdb
15.
16. [root@gfsclient01 ~]# pvcreate  --qq --metadatasize=512M
    --dataalignment=256K /dev/sdb
17. pvcreate: unrecognized option '--qq'
18.   Error during parsing of command line.
19.
20. [root@gfsclient01 ~]# pvcreate  -qq --metadatasize=512M -
    -dataalignment=256K /dev/sdb
21.
22. [root@gfsclient01 ~]# lsblk
23. NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
24. sda                                8:0      0  100G  0 disk
25. └─sda1                            8:1      0    1G  0 part /boot
26. └─sda2                            8:2      0   99G  0 part
27.    |centos-root 253:0    0   50G  0 lvm  /
28.    |centos-swap 253:1    0   3.9G  0 lvm  [SWAP]
29.    |centos-home 253:2    0  45.1G  0 lvm  /home
30. sdb                                8:16     0    5G  0 disk
31. sr0                               11:0     1 1024M  0 rom
32.
33. [root@gfsclient01 ~]# pvcreate  -qq --metadatasize=512M -
    -dataalignment=256K /dev/sdb
34.
35. [root@gfsclient01 ~]# pvs
36.  PV          VG      Fmt  Attr  PSize   PFree
37.  /dev/sda2   centos lvm2  a--   <99.00g 4.00m
38.  /dev/sdb    lvm2   ---    5.00g 5.00g
39.
40. #创建 vg 信息
41. [root@gfsclient01 ~]# vgcreate -qq --physicalextentsize=4
    M --autobackup=y lvm-sdb-vg /dev/sdb
42.

```

```

43. [root@gfsclient01 ~]# vgs
44.   VG          #PV #LV #SN Attr   VSize   VFree
45.   centos        1   3   0 wz--n- <99.00g  4.00m
46.   lvm-sdb-vg    1   0   0 wz--n- <4.50g <4.50g
47. [root@gfsclient01 ~]# pvs -o pv_name,pv_uuid,vg_name --re
    portformat=json /dev/sdb
48.   {
49.     "report": [
50.       {
51.         "pv": [
52.           {"pv_name":"/dev/sdb", "pv_uuid":"sl0tKG-SL4x-VgvT-S71I-VpmH-mk1D-LmMEBj", "vg_name":"lvm-sdb-vg"}
53.         ]
54.       }
55.     ]
56.   }
57.
58. [root@gfsclient01 ~]# udevadm info --query=symlink --name
    =/dev/sdb
59. disk/by-id/ata-VBOX_HARDDISK_VB4923bed4-cb43f4fd disk/by-
    id/lvm-pv-uuid-sl0tKG-SL4x-VgvT-S71I-VpmH-mk1D-LmMEBj disk
    /by-path                               /pci-000
    0:00:0d.0-ata-2.0
60.
61. [root@gfsclient01 ~]# vgdisplay -c lvm-sdb-vg
62.   lvm-sdb-vg:r/w:772:-1:0:0:0:-1:0:1:1:4714496:4096:1151:
    0:1151:t19Pps-GeL4-lzWv-msMM-MgFF-oqhD-iGgXqd
63.
64. #创建 lv 信息
65. [root@gfsclient01 ~]# lvcreate -qq --autobackup=y --pool
    metadatasize 8192K --chunksize 256K --size 1048576K --thin
    lvm-sdb                               -vg/lvm-
    sdb-tp --virtualsize 1048576K --name lvm-sdb-lvname
66.
67. [root@gfsclient01 ~]# lvs
68.   LV          VG          Attr      LSize   Pool
    Origin Data%  Meta%   Move Log Cpy%Sync Convert
69.   home          centos      -wi-ao---- <45.12g
70.   root          centos      -wi-ao---- 50.00g
71.   swap          centos      -wi-ao---- <3.88g
72.   lvm-sdb-lvname lvm-sdb-vg Vwi-a-tz-- 1.00g lvm-sdb-tp
    0.00

```

```

73.    lvm-sdb-tp      lvm-sdb-vg twi-aotz--    1.00g
        0.00    10.45
74.
75. [root@gfsclient01 ~]# ls -l /dev/mapper/
76. total 0
77. lrwxrwxrwx. 1 root root      7 Jun 15 12:22 centos-home
    -> ../dm-2
78. lrwxrwxrwx. 1 root root      7 Jun 15 12:22 centos-root
    -> ../dm-0
79. lrwxrwxrwx. 1 root root      7 Jun 15 12:22 centos-swap
    -> ../dm-1
80. crw----- . 1 root root 10, 236 Jun 15 12:22 control
81. lrwxrwxrwx. 1 root root      7 Jun 15 12:31 lvm--sdb--vg
    -lvm--sdb--lvname -> ../dm-7
82. lrwxrwxrwx. 1 root root      7 Jun 15 12:31 lvm--sdb--vg
    -lvm--sdb--tp -> ../dm-6
83. lrwxrwxrwx. 1 root root      7 Jun 15 12:31 lvm--sdb--vg
    -lvm--sdb--tp_tdata -> ../dm-4
84. lrwxrwxrwx. 1 root root      7 Jun 15 12:31 lvm--sdb--vg
    -lvm--sdb--tp_tmeta -> ../dm-3
85. lrwxrwxrwx. 1 root root      7 Jun 15 12:31 lvm--sdb--vg
    -lvm--sdb--tp-tpool -> ../dm-5
86.
87. #对 lv 进行格式化为 xfs 文件系统。
88. [root@gfsclient01 ~]# mkfs.xfs -i size=512 -n size=8192 /
    dev/mapper/lvm--sdb--vg-lvm--sdb--lvname
89. Discarding blocks...Done.
90. meta-data=/dev/mapper/lvm--sdb--vg-lvm--sdb--lvname isize
    =512    agcount=8, agsize=32768 blks
91.          =                sectsz=512    attr=2, pro
    jid32bit=1
92.          =                crc=1          finobt=0, s
    parse=0
93. data      =                bsize=4096    blocks=2621
    44, imaxpct=25
94.          =                sunit=64      swidth=64 b
    lks
95. naming    =version 2        bsize=8192    ascii-ci=0
    ftype=1
96. log       =internal log     bsize=4096    blocks=2560,
    version=2
97.          =                sectsz=512    sunit=64 bl
    ks, lazy-count=1

```

99.

100.#创建目录进行挂载

```
101.[root@gfsclient01 ~]# mkdir -p /var/lib/gfs/lvm-sdb-vg/lv
    m-sdb-lvname
```

102.

```
103.[root@gfsclient01 ~]# mount -o rw,inode64,noatime,nouuid,
    noauto,x-systemd.automount /dev/mapper/lvm--sdb--vg-lvm--s
    db--lvna me /var/
    lib/gfs/lvm-sdb-vg/lvm-sdb-lvname
```

104.

```
105.[root@gfsclient01 ~]# df -h /var/lib/gfs/lvm-sdb-vg/lvm-s
    db-lvname
```

```
106.Filesystem      Size  Used  Avail
    106.1 /dev/sda1  100G   50G   50G
    106.2 /dev/sda2  100G   50G   50G
    106.3 /dev/sda3  100G   50G   50G
    106.4 /dev/sda4  100G   50G   50G
    106.5 /dev/sda5  100G   50G   50G
    106.6 /dev/sda6  100G   50G   50G
    106.7 /dev/sda7  100G   50G   50G
    106.8 /dev/sda8  100G   50G   50G
    106.9 /dev/sda9  100G   50G   50G
    106.10 /dev/sda10 100G   50G   50G
    106.11 /dev/sda11 100G   50G   50G
    106.12 /dev/sda12 100G   50G   50G
    106.13 /dev/sda13 100G   50G   50G
    106.14 /dev/sda14 100G   50G   50G
    106.15 /dev/sda15 100G   50G   50G
    106.16 /dev/sda16 100G   50G   50G
    106.17 /dev/sda17 100G   50G   50G
    106.18 /dev/sda18 100G   50G   50G
    106.19 /dev/sda19 100G   50G   50G
    106.20 /dev/sda20 100G   50G   50G
    106.21 /dev/sda21 100G   50G   50G
    106.22 /dev/sda22 100G   50G   50G
    106.23 /dev/sda23 100G   50G   50G
    106.24 /dev/sda24 100G   50G   50G
    106.25 /dev/sda25 100G   50G   50G
    106.26 /dev/sda26 100G   50G   50G
    106.27 /dev/sda27 100G   50G   50G
    106.28 /dev/sda28 100G   50G   50G
    106.29 /dev/sda29 100G   50G   50G
    106.30 /dev/sda30 100G   50G   50G
    106.31 /dev/sda31 100G   50G   50G
    106.32 /dev/sda32 100G   50G   50G
    106.33 /dev/sda33 100G   50G   50G
    106.34 /dev/sda34 100G   50G   50G
    106.35 /dev/sda35 100G   50G   50G
    106.36 /dev/sda36 100G   50G   50G
    106.37 /dev/sda37 100G   50G   50G
    106.38 /dev/sda38 100G   50G   50G
    106.39 /dev/sda39 100G   50G   50G
    106.40 /dev/sda40 100G   50G   50G
    106.41 /dev/sda41 100G   50G   50G
    106.42 /dev/sda42 100G   50G   50G
    106.43 /dev/sda43 100G   50G   50G
    106.44 /dev/sda44 100G   50G   50G
    106.45 /dev/sda45 100G   50G   50G
    106.46 /dev/sda46 100G   50G   50G
    106.47 /dev/sda47 100G   50G   50G
    106.48 /dev/sda48 100G   50G   50G
    106.49 /dev/sda49 100G   50G   50G
    106.50 /dev/sda50 100G   50G   50G
    106.51 /dev/sda51 100G   50G   50G
    106.52 /dev/sda52 100G   50G   50G
    106.53 /dev/sda53 100G   50G   50G
    106.54 /dev/sda54 100G   50G   50G
    106.55 /dev/sda55 100G   50G   50G
    106.56 /dev/sda56 100G   50G   50G
    106.57 /dev/sda57 100G   50G   50G
    106.58 /dev/sda58 100G   50G   50G
    106.59 /dev/sda59 100G   50G   50G
    106.60 /dev/sda60 100G   50G   50G
    106.61 /dev/sda61 100G   50G   50G
    106.62 /dev/sda62 100G   50G   50G
    106.63 /dev/sda63 100G   50G   50G
    106.64 /dev/sda64 100G   50G   50G
    106.65 /dev/sda65 100G   50G   50G
    106.66 /dev/sda66 100G   50G   50G
    106.67 /dev/sda67 100G   50G   50G
    106.68 /dev/sda68 100G   50G   50G
    106.69 /dev/sda69 100G   50G   50G
    106.70 /dev/sda70 100G   50G   50G
    106.71 /dev/sda71 100G   50G   50G
    106.72 /dev/sda72 100G   50G   50G
    106.73 /dev/sda73 100G   50G   50G
    106.74 /dev/sda74 100G   50G   50G
    106.75 /dev/sda75 100G   50G   50G
    106.76 /dev/sda76 100G   50G   50G
    106.77 /dev/sda77 100G   50G   50G
    106.78 /dev/sda78 100G   50G   50G
    106.79 /dev/sda79 100G   50G   50G
    106.80 /dev/sda80 100G   50G   50G
    106.81 /dev/sda81 100G   50G   50G
    106.82 /dev/sda82 100G   50G   50G
    106.83 /dev/sda83 100G   50G   50G
    106.84 /dev/sda84 100G   50G   50G
    106.85 /dev/sda85 100G   50G   50G
    106.86 /dev/sda86 100G   50G   50G
    106.87 /dev/sda87 100G   50G   50G
    106.88 /dev/sda88 100G   50G   50G
    106.89 /dev/sda89 100G   50G   50G
    106.90 /dev/sda90 100G   50G   50G
    106.91 /dev/sda91 100G   50G   50G
    106.92 /dev/sda92 100G   50G   50G
    106.93 /dev/sda93 100G   50G   50G
    106.94 /dev/sda94 100G   50G   50G
    106.95 /dev/sda95 100G   50G   50G
    106.96 /dev/sda96 100G   50G   50G
    106.97 /dev/sda97 100G   50G   50G
    106.98 /dev/sda98 100G   50G   50G
    106.99 /dev/sda99 100G   50G   50G
    106.100 /dev/sda100 100G   50G   50G
    106.101 /dev/sda101 100G   50G   50G
    106.102 /dev/sda102 100G   50G   50G
    106.103 /dev/sda103 100G   50G   50G
    106.104 /dev/sda104 100G   50G   50G
    106.105 /dev/sda105 100G   50G   50G
    106.106 /dev/sda106 100G   50G   50G
    106.107 /dev/sda107 100G   50G   50G
    106.108 /dev/sda108 100G   50G   50G
    106.109 /dev/sda109 100G   50G   50G
    106.110 /dev/sda110 100G   50G   50G
    106.111 /dev/sda111 100G   50G   50G
    106.112 /dev/sda112 100G   50G   50G
    106.113 /dev/sda113 100G   50G   50G
    106.114 /dev/sda114 100G   50G   50G
    106.115 /dev/sda115 100G   50G   50G
    106.116 /dev/sda116 100G   50G   50G
    106.117 /dev/sda117 100G   50G   50G
    106.118 /dev/sda118 100G   50G   50G
    106.119 /dev/sda119 100G   50G   50G
    106.120 /dev/sda120 100G   50G   50G
    106.121 /dev/sda121 100G   50G   50G
    106.122 /dev/sda122 100G   50G   50G
    106.123 /dev/sda123 100G   50G   50G
    106.124 /dev/sda124 100G   50G   50G
    106.125 /dev/sda125 100G   50G   50G
    106.126 /dev/sda126 100G   50G   50G
    106.127 /dev/sda127 100G   50G   50G
    106.128 /dev/sda128 100G   50G   50G
    106.129 /dev/sda129 100G   50G   50G
    106.130 /dev/sda130 100G   50G   50G
    106.131 /dev/sda131 100G   50G   50G
    106.132 /dev/sda132 100G   50G   50G
    106.133 /dev/sda133 100G   50G   50G
    106.134 /dev/sda134 100G   50G   50G
   
```

```
107./dev/mapper/lvm--sdb--vg-lvm--sdb--lvname 1014M    33M    98
    2M    4% /var/lib/gfs/lvm-sdb-vg/lvm-sdb-lvname
```

108.

```
109.[root@gfscclient01 ~]# lsblk
```

[illegible]

```
111.sda      8:0    0 100G  0 d
   isk
```

```
112. |sda1      8:1    0    1G  0 p
    art /boot
```

113. L _{sda2}	8:2	0	99G	0 p
art				

```
114. |centos-root          253:0    0    50G  0 1
      vm /
```

```
115. |centos-swap          253:1    0  3.9G  0 1
      vm [SWAP]
```

```
116.  └─centos-home          253:2    0 45.1G  0 1
      vm /home
```

```
117.sdb      8:16  0    5G  0 d
isk
```

```
118.|-lvm--sdb--vg-lvm--sdb--tp_tmeta    253:3      0      8M  0 1
      vm
```

```
119. |  └─lvm--sdb--vg-lvm--sdb--tp-tpool 253:5      0      1G  0 1
      vm
```

```
120. | |lvm--sdb--vg-lvm--sdb--tp      253:6      0      1G  1  1
      vm
```

```

121.|    └─lvm--sdb--vg-lvm--sdb--lvname 253:7      0      1G  0 1
    vm  /var/lib/gfs/lvm-sdb-vg/lvm-sdb-lvname
122.└─lvm--sdb--vg-lvm--sdb--tp_tdata  253:4      0      1G  0 1
    vm
123.    └─lvm--sdb--vg-lvm--sdb--tp-tpool 253:5      0      1G  0 1
    vm
124.    └─lvm--sdb--vg-lvm--sdb--tp      253:6      0      1G  1 1
    vm
125.    └─lvm--sdb--vg-lvm--sdb--lvname 253:7      0      1G  0 1
    vm  /var/lib/gfs/lvm-sdb-vg/lvm-sdb-lvname
126.sr0

```

这里通过 pv,vg 和 lv 操作之后,再对 mapper 进行格式化为常用的文件系统,然后就可以挂载使用了。另外注意,这里建议取的名字使用下划线,因为 lvm2 的 lvcreate 这些命令中使用横线的话,识别出现会和命名的有点区别,而使用下划线可以避免命令的统一。

同时这里还可以留意在对块设备进行操作前 lsblk 的输出,对比最后挂载目录之后,这里可以发现对应的 lv,vg 和 pv,还有和块设备的关联关系了。

这里也可以留意到,一个块设备进行格式化之后,这里需要一定空间进行保留元数据信息的,这里是有要求的,那么这个大小一般要求多少呢?通常保留不超过 16G 的大小。

下面给出一个脚本来进行创建。

```

1. [root@gfsclient01 ~]# cat lvm2.sh
2.
3. size="50G"
4.
5.
6. sizenum=`echo $size | tr -cd "[0-9]"`
7.
8. vsize=`awk -v x=$sizenum -v y=0.9 'BEGIN{printf "%.2f\n",
    x*y}'`"G"
9.
10. mountpath="/glusterfs/mounts"

```



```

11.
12. devices="/dev/sdb"
13. vgName="lv_sdb_vg_210616"
14.
15. sudo pvcreate -qq --metadatasize=512M --dataalignment=256
    K $devices
16.
17. sudo vgcreate -qq --physicalextentsize=4M --autobackup=y
    $vgName $devices
18.
19. #pvs show info
20. sudo pvs -o pv_name,pv_uuid,vg_name --reportformat=json $
    devices
21.
22. #udevadm
23. sudo udevadm info --query=symlink --name=$devices
24.
25. #vgdisplay -c
26. sudo vgdisplay -c $vgName
27.
28. #这里--thinpool 后面第一个参数是 thin 的名称,第二个是 vg 的名
    称

29. #-Zn 是为了 consider disabling zeroing,避免警告.
30. lvcreate -L $vsize --thinpool thinpool_$vgName $vgName -
    Zn
31.
32. #这里--thin -n后面的参数,第一个是thin volume名称,后面斜杠前
    是vg名称,斜杠后是thinpool名称

33. lvcreate -V $vsize --thin -n lvuser_$vgName $vgName/thinp
    ool_$vgName
34.
35. #format xfs
36. sudo mkfs.xfs -i size=512 -n size=4096 -s size=4096 -K /
    dev/mapper/$vgName-lvuser_$vgName
37.
38. #write into fstab
39. sudo mkdir -p $mountpath/$vgName/lvuser_$vgName/brick
40.

```

```

41. sudo mount -o rw,inode64,noatime,nouuid,noauto,x-systemd.
    automount /dev/mapper/$vgName-lvuser_$vgName $mountpath/
    $vgName/lvuser_$vgName/brick
42.
43. sudo awk "BEGIN {print \"/dev/mapper/$vgName-lvuser_$vgName
    $mountpath/$vgName/lvuser_$vgName/brick xfs rw,inode6
    4,noatime,nouuid,noauto,x-systemd.automount 1 2 \" >> \"/
    etc/fstab\"}"
44.
45. sudo df -h $mountpath/$vgName/lvuser_$vgName/brick

```

这里和前面的区别是 lvcreate 创建的时候，考虑了块设备的容量大小进行设置，并且改用 lvcreate -V 参数，这样可以把除了保留的空间以外都使用上，最后还会把这个挂载信息写入到/etc/fstab 文件中。

下面是其中一个节点的执行结果。

```

1. [root@gfs03 ~]# bash lvm2.sh
2. {
3.     "report": [
4.         {
5.             "pv": [
6.                 {"pv_name":"/dev/sdb", "pv_uuid":"EbzPs
    F-Qqkl-FMy7-11EZ-clld-dAyl-uolZbi", "vg_name":"lv_sdb_vg_2
    10616"}
7.             ]
8.         }
9.     ]
10. }
11. disk/by-id/ata-VBOX_HARDDISK_VBf2804901-94951830 disk/by-
    id/lvm-pv-uuid-EbzPsF-Qqkl-FMy7-11EZ-clld-dAyl-uolZbi disk
    /by-path/pci-0000:00:0d.0-ata-2.0
12.   lv_sdb_vg_210616:r/w:772:-1:0:0:0:-1:0:1:1:51900416:409
    6:12671:0:12671:D32P11-Ns7P-dfV4-2Nj0-Nwuh-04SE-fXMtYs
13.   Thin pool volume with chunk size 64.00 KiB can address
    at most 15.81 TiB of data.
14.   Logical volume "thinpool_lv_sdb_vg_210616" created.
15.   Logical volume "lvuser_lv_sdb_vg_210616" created.
16. meta-data=/dev/mapper/lv_sdb_vg_210616-lvuser_lv_sdb_vg_2
    10616 isize=512    agcount=16, agsize=737280 blks

```

```

17.          =                      sectsz=4096  attr=2, pro
    jid32bit=1
18.          =                      crc=1        finobt=0, s
    parse=0
19. data      =                      bsize=4096    blocks=1179
    6480, imaxpct=25
20.          =                      sunit=16      swidth=16 b
    lks
21. naming    =version 2            bsize=4096    ascii-ci=0
    ftype=1
22. log       =internal log         bsize=4096    blocks=5760,
    version=2
23.          =                      sectsz=4096    sunit=1 blk
    s, lazy-count=1
24. realtime  =none                 extsz=4096    blocks=0, r
    textents=0
25. Filesystem                               Siz
    e Used Avail Use% Mounted on
26. /dev/mapper/lv_sdb_vg_210616-lvuser_lv_sdb_vg_210616 45
    G  33M  45G   1% /glusterfs/mounts/lv_sdb_vg_210616/lvus
    er_lv_sdb_vg_210616/brick

```

那么这里还可以检查一下 vgs 和 lvs 的信息，看看和之前的操作的区别。

```

1. [root@gfs03 ~]# lvs
2.   LV                               VG                               Attr      L
   Size  Pool                               Origin Data%  Meta%  Move
   Log Cpy%Sync Convert
3.   home                               centos                               -wi-ao---- <
   45.12g
4.   root                               centos                               -wi-ao----
   50.00g
5.   swap                               centos                               -wi-ao----
   <3.88g
6.   lvuser_lv_sdb_vg_210616  lv_sdb_vg_210616 Vwi-aot---
   45.00g thinpool_lv_sdb_vg_210616 0.05
7.   thinpool_lv_sdb_vg_210616 lv_sdb_vg_210616 twi-aot---
   45.00g 0.05 10.46
8.
9. [root@gfs03 ~]# vgs
10.  VG                               #PV #LV #SN Attr   VSize   VFree
11.  centos                           1   3   0 wz--n- <99.00g 4.00m
12.  lv_sdb_vg_210616                 1   2   0 wz--n- <49.50g 4.40g

```

可以看到这里保留了 10%的空间作为元数据保存空间的。

4.2.3. snapshot 创建

同理，下面对其他几个节点进行相同操作，并且创建一个 volume 然后创建快照，结果如下所示。

```
1. [root@gfs01 ~]# gluster volume create test-snapshot-lvm2
   replica 3 192.168.0.{110,111,112}:/glusterfs/mounts/lv_sdb
   _vg_210616/lvuser_lv_sdb_vg_210616/brick/vol
2. volume create: test-snapshot-lvm2: success: please start
   the volume to access data
3.
4. [root@gfs01 ~]# gluster volume start test-snapshot-lvm2
5. volume start: test-snapshot-lvm2: success
6.
7. [root@gfs01 ~]# gluster snapshot create test-snapshot-lvm
   2-210616 test-snapshot-lvm2 no-timestamp
8. snapshot create: success: Snap test-snapshot-lvm2-210616
   created successfully
9.
10. [root@gfs01 ~]# gluster snapshot list
11. test-snapshot-lvm2-210616
12.
13. [root@gfs01 ~]# gluster snapshot info
14. Snapshot                               : test-snapshot-lvm2-210616
15. Snap UUID                             : 7e7c2dbe-62bd-4253-b1e6-0f326
    600d117
16. Created                               : 2021-06-15 17:51:41 +0000
17. Snap Volumes:
18.
19.      Snap Volume Name                   : 1de228ed2d248e9bee60
    a5985fdf821
20.      Origin Volume name                 : test-snapshot-lvm2
21.      Snaps taken for test-snapshot-lvm2 : 1
22.      Snaps available for test-snapshot-lvm2 : 255
23.      Status                             : Stopped
```

这里创建了一个名为 test-snapshot-lvm2-210616 的快照，参数 no-timestamp 的作用是不添加随机字符串，否则默认快照名称是添加后缀的，这样不方便管理。

另外这里可以看到快照的默认状态是 stopped 的，但是并不代表快照没有生效，只是该快照的进程并没有激活，因为都是依赖 linux lvm2 的快照，因此这里没有办法看到快照的一些数据信息。

在快照中有两个比较重要的参数，分别是 snap-max-hard-limit 和 snap-max-soft-limit，这两个就是决定了该 volume 的最大快照数量，还有达到多少比例之后，会把之前的快照删掉，因为快照的空间也不是无尽的，因为保留过多的快照数量会非常占用空间的，而实际参数的使用效果，大家可以自行测试一下。

下面可以进行测试一下快照的回滚功能。

4.2.4. 快照回滚

下面进行快照的回滚测试，操作如下所示。

1. //这里先创建一个文件
2. [root@gfs03 ~]# ls /mnt/test-snapshot-lvm2/
3. 1.txt
- 4.
- 5.
6. [root@gfs03 ~]# gluster snapshot restore test-snapshot-lvm2-210616
7. Restore operation will replace the original volume with the snapshotted volume. Do you still want to **continue**? (y/n)
y

```

8. snapshot restore: failed: Volume (test-snapshot-lvm2) has
   been started. Volume needs to be stopped before restoring
   a snapshot.
9. Snapshot command failed
10.
11. //激活快照
12. [root@gfs03 ~]# gluster snapshot activate test-snapshot-
   lvm2-210616
13. Snapshot activate: test-snapshot-lvm2-210616: Snap activa
   ted successfully
14.
15. [root@gfs03 ~]# gluster snapshot status test-snapshot-l
   vm2-210616
16.
17. Snap Name : test-snapshot-lvm2-210616
18. Snap UUID : 7e7c2dbe-62bd-4253-b1e6-0f326600d117
19.
20.      Brick Path      : 192.168.0.110:/run/gluster/
   snaps/1de2288ed2d248e9bee60a5985fdf821/brick1/vol
21.      Volume Group    : lv_sdb_vg_210616
22.      Brick Running   : Yes
23.      Brick PID       : 10189
24.      Data Percentage : 0.05
25.      LV Size         : 45.00g
26.
27.
28.      Brick Path      : 192.168.0.111:/run/gluster/
   snaps/1de2288ed2d248e9bee60a5985fdf821/brick2/vol
29.      Volume Group    : lv_sdb_vg_210616
30.      Brick Running   : Yes
31.      Brick PID       : 9775
32.      Data Percentage : 0.05
33.      LV Size         : 45.00g
34.
35.
36.      Brick Path      : 192.168.0.112:/run/gluster/
   snaps/1de2288ed2d248e9bee60a5985fdf821/brick3/vol
37.      Volume Group    : lv_sdb_vg_210616
38.      Brick Running   : Yes
39.      Brick PID       : 10012
40.      Data Percentage : 0.05
41.      LV Size         : 45.00g

```

这里首先挂载该 volume，然后创建一个文件，因为无法直接 restore 回滚

快照的，需要先激活该快照，接着可以使用 `snapshot status` 看到快照的一些信息，这里包括一家使用的数据比例，brick 的 pid 和路径等信息，这里的路径和 volume brick 是不一样的，要留意一下下。

```
1. [root@gfs03 ~]# gluster volume stop test-snapshot-lvm2
2. Stopping volume will make its data inaccessible. Do you want to continue? (y/n) y
3. volume stop: test-snapshot-lvm2: success
4.
5. [root@gfs03 ~]# gluster snapshot restore test-snapshot-lvm2-210616
6. Restore operation will replace the original volume with the snapshotted volume. Do you still want to continue? (y/n) y
7. Snapshot restore: test-snapshot-lvm2-210616: Snap restored successfully
8.
9. [root@gfs03 ~]# umount /mnt/test-snapshot-lvm2
10.
11. [root@gfs03 ~]# gluster volume start test-snapshot-lvm2
12. volume start: test-snapshot-lvm2: success
13.
14. [root@gfs03 ~]# mount -t glusterfs 192.168.0.110:test-snapshot-lvm2 /mnt/test-snapshot-lvm2
15.
16. [root@gfs03 ~]# ls /mnt/test-snapshot-lvm2/
```

接着这里想要回滚快照的话，需要先停止该 volume，接着进行快照 restore，然后再次挂载的时候，就发现文件已经不存在了。这里还要注意一点，一旦快照被回滚了，那么这里就无法再次看到该快照了。

4.2.5. 删除快照

这里删除快照的方式比较简单，直接使用命令 `delete` 即可。

1. [root@gfs01 ~]# gluster snapshot **delete** test-snapshot-lvm2-21061701
2. Deleting snap will erase all the information about the snap. Do you still want to **continue**? (y/n) y
3. snapshot **delete**: test-snapshot-lvm2-21061701: snap removed successfully

但是如果还存在快照的时候,是无法直接删除 volume 的,会提示以下报错。

1. [root@gfs01 ~]# gluster volume **delete** test-snapshot-lvm2
2. Deleting volume will erase all information about the volume. Do you want to **continue**? (y/n) y
3. volume **delete**: test-snapshot-lvm2: failed: Cannot **delete** Volume test-snapshot-lvm2 ,as it has 1 snapshots. To **delete** the volume, first **delete** all the snapshots under it.

那么这里在使用快照的时候,就要留意一个问题了,当快照无法删除时,就会影响到 volume 的删除了,而无法删除 volume,又会影响到剔除节点等功能。那这里什么情况下会导致快照无法删除呢?就是当本地的 lvm2 出现问题的时候,例如 brick 所在的块设备,因为磁盘容量爆了,导致出现 io error 的时候,这个在生产环境中遇到过,导致快照异常了。