

第一周实战任务 2

实验目标

使用 `/dev/sdb` 创建一个 LVM 存储 `/data`:
`/data` 使用逻辑卷 `/dev/vgdata/lv_data` (10G)
扩容 LV 到 20G
验证文件系统扩容成功
再扩容 LV 回 15G (模拟释放空间)
验证整个扩容过程数据不损坏

EXT4 版本扩容缩容

一、扩容

1、确认新磁盘 `/dev/sdb`

```
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0   20G  0 disk
└─sda1     8:1    0    1G  0 part /boot
└─sda2     8:2    0   19G  0 part
  └─centos-root 253:0    0   17G  0 lvm  /
  └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb        8:16   0   30G  0 disk
sr0       11:0   1   4.4G 0 rom
```

2、创建 LVM 环境

创建 PV 把物理磁盘变成 LVM 可用的物理卷

```
[root@localhost ~]# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created.
```

创建 VG 创建存储池，将 PV 加入卷组。

```
[root@localhost ~]# vgcreate vgdata /dev/sdb
Volume group "vgdata" successfully created
```

创建 LV (初始 10G) 在 VG 中划出 10G, 用来挂载 `/data`

```
[root@localhost ~]# lvcreate -L 10G -n lv_data vgdata
Logical volume "lv_data" created.
```

3、创建 EXT4 文件系统并挂载到 `/data`

1、格式化为 ext4

```
[root@localhost ~]# mkfs.ext4 /dev/vgdata/lv_data
mke2fs 1.42.9 (28-Dec-2013)
文件系统标签=
OS type: Linux
块大小=4096 (log=2)
```

```
分块大小=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2621440 blocks
131072 blocks (5.00%) reserved for the super user
第一个数据块=0
Maximum filesystem blocks=2151677952
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
```

```
Allocating group tables: 完成
正在写入 inode 表: 完成
Creating journal (32768 blocks): 完成
Writing superblocks and filesystem accounting information: 完成
```

2、创建挂载点并挂载

```
[root@localhost ~]# mkdir /data
[root@localhost ~]# mount /dev/vgdata/lv_data /data
```

3、验证挂载

```
[root@localhost ~]# df -Th /data
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata-lv_data ext4  9.8G  37M  9.2G     1% /data

[root@localhost ~]# df -Th /data
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata-lv_data ext4  9.8G  37M  9.2G     1% /data
```

4、写入测试文件（用于验证缩容后数据是否损坏）

```
[root@localhost ~]# echo "lvm-resize-test" > /data/check.txt
[root@localhost ~]# md5sum /data/check.txt
aece90d101416b5e64911fdc882c3237  /data/check.txt
[root@localhost ~]# cat /data/check.txt
lvm-resize-test
```

```
[root@localhost ~]# echo "lvm-resize-test" > /data/check.txt
[root@localhost ~]# md5sum /data/check.txt
aece90d101416b5e64911fdc882c3237  /data/check.txt
```

4、扩容 LV 从 10G → 20G

1、扩容 LV

```
[root@localhost ~]# lvextend -L 20G /dev/vgdata/lv_data
      New size (5120 extents) matches existing size (5120 extents).
```

2、扩容文件系统（EXT4 自动填满整个 LV）

```
[root@localhost ~]# resize2fs /dev/vgdata/lv_data
resize2fs 1.42.9 (28-Dec-2013)
```

```
Filesystem at /dev/vgdata/lv_data is mounted on /data; on-line resizing required
old_desc_blocks = 2, new_desc_blocks = 3
The filesystem on /dev/vgdata/lv_data is now 5242880 blocks long.
```

3、验证

```
[root@localhost ~]# df -Th /data
文件系统           类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata-lv_data ext4  20G  44M  19G   1% /data
```

```
[root@localhost ~]# df -Th /data
文件系统           类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata-lv_data ext4  20G  44M  19G   1% /data
[root@localhost ~]#
```

二、缩容

1、缩容 LV 从 20G → 15G (必须严格按顺序)

顺序必须:

卸载 → fsck → 缩文件系统 → 缩 LV → 挂载

1.1 卸载

```
[root@localhost ~]# umount /data
```

1.2 检查文件系统

```
[root@localhost ~]# e2fsck -f /dev/vgdata/lv_data
e2fsck 1.42.9 (28-Dec-2013)
第一步: 检查 inode, 块, 和大小
第二步: 检查目录结构
第三步: 检查目录连接性
Pass 4: Checking reference counts
第五步: 检查簇概要信息
/dev/vgdata/lv_data: 12/1310720 files (0.0% non-contiguous), 126312/5242880 blocks
```

```
[root@localhost ~]# e2fsck -f /dev/vgdata/lv_data
e2fsck 1.42.9 (28-Dec-2013)
第一步: 检查 inode, 块, 和大小
第二步: 检查目录结构
第三步: 检查目录连接性
Pass 4: Checking reference counts
第五步: 检查簇概要信息
/dev/vgdata/lv_data: 12/1310720 files (0.0% non-contiguous), 126312/5242880 blocks
```

1.3 先缩文件系统到 15G

```
[root@localhost ~]# resize2fs /dev/vgdata/lv_data 15G
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vgdata/lv_data to 3932160 (4k) blocks.
The filesystem on /dev/vgdata/lv_data is now 3932160 blocks long.
```

解释: 先把“货架”缩小, 否则后面缩 LV 会直接截断数据。

这里将 LV 比喻成“房子”, 把系统文件比喻成“货架”

如果先把房子砍断的话，那么货架必定会损坏。反之，先将货架缩小，在将房子缩小，那么货架也不会受到损坏。

1.4 再缩 LV 到 15G

```
[root@localhost ~]# lvreduce -L 15G /dev/vgdata/lv_data
WARNING: Reducing active logical volume to 15.00 GiB.
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce vgdata/lv_data? [y/n]: y
Size of logical volume vgdata/lv_data changed from 20.00 GiB (5120 extents) to 15.00 GiB
(3840 extents).
Logical volume vgdata/lv_data successfully resized.
```

1.5 重新挂载，检测

```
[root@localhost ~]# mount /dev/vgdata/lv_data /data
[root@localhost ~]# df -Th /data
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata-lv_data ext4  15G   41M  14G   1% /data

[root@localhost ~]# mount /dev/vgdata/lv_data /data
[root@localhost ~]# df -Th /data
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata-lv_data ext4  15G   41M  14G   1% /data
```

2、最终测验

验证数据未损坏（MD5 校验）

```
[root@localhost ~]# md5sum /data/check.txt
aece90d101416b5e64911fdc882c3237  /data/check.txt
[root@localhost ~]# cat /data/check.txt
lvm-resize-test
```

```
[root@localhost ~]# md5sum /data/check.txt
aece90d101416b5e64911fdc882c3237  /data/check.txt
[root@localhost ~]# cat /data/check.txt
Lvm-resize-test
[root@localhost ~]#
```

可以看到得到与缩容前 完全相同的 MD5 值。

```
[root@localhost ~]# df -h
文件系统          容量 已用 可用 已用% 挂载点
devtmpfs           898M   0  898M   0% /dev
tmpfs              910M   0  910M   0% /dev/shm
tmpfs              910M  9.6M  901M   2% /run
tmpfs              910M   0  910M   0% /sys/fs/cgroup
/dev/mapper/centos-root    17G  2.4G  15G  14% /
/dev/sda1           1014M 151M  864M  15% /boot
tmpfs              182M   0  182M   0% /run/user/0
/dev/mapper/vgdata-lv_data 15G   41M  14G   1% /data
```

```
[root@localhost ~]# df -h
文件系统 容量 已用 可用 已用% 挂载点
devtmpfs 898M 0 898M 0% /dev
tmpfs 910M 0 910M 0% /dev/shm
tmpfs 910M 9.6M 901M 2% /run
tmpfs 910M 0 910M 0% /sys/fs/cgroup
/dev/mapper/centos-root 17G 2.4G 15G 14% /
/dev/sdal 1014M 151M 864M 15% /boot
tmpfs 182M 0 182M 0% /run/user/0
/dev/mapper/vgdata-lv_data 15G 41M 14G 1% /data
[root@localhost ~]#
```

XFS 版本扩容缩容

一、扩容

这里的步骤和上面的基本一致，所以就简单过一遍

```
[root@localhost ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 20G 0 disk
└─sda1 8:1 0 1G 0 part /boot
└─sda2 8:2 0 19G 0 part
  ├─centos-root 253:0 0 17G 0 lvm /
  └─centos-swap 253:1 0 2G 0 lvm [SWAP]
sdb 8:16 0 30G 0 disk
└─vgdata-lv_data 253:2 0 15G 0 lvm
sdc 8:32 0 30G 0 disk
sr0 11:0 1 4.4G 0 rom
```

```
[root@localhost ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 20G 0 disk
└─sda1 8:1 0 1G 0 part /boot
└─sda2 8:2 0 19G 0 part
  ├─centos-root 253:0 0 17G 0 lvm /
  └─centos-swap 253:1 0 2G 0 lvm [SWAP]
sdb 8:16 0 30G 0 disk
└─vgdata-lv_data 253:2 0 15G 0 lvm
sdc 8:32 0 30G 0 disk
sr0 11:0 1 4.4G 0 rom
[root@localhost ~]#
```

1、创建 LVM 环境

```
[root@localhost ~]# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created.
[root@localhost ~]# vgcreate vgdata2 /dev/sdc
/dev/vgdata: already exists in filesystem
Run `vgcreate --help` for more information.
[root@localhost ~]# lvcreate -L 10G -n lv_xdata2 vgdata2
Logical volume "lv_xdata2" created.
[root@localhost ~]# df -Th /xdata02
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/vgdata2-lv_xdata2	xfs	10G	33M	10G	1%	/xdata02

2、创建 XFS 文件系统并挂载

```
[root@localhost ~]# mount /dev/vgdata2/lv_xdata2 /xdata02
[root@localhost ~]# df -Th /xdata02
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata2-lv_xdata2 xfs   10G  33M  10G   1% /xdata02
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0  20G  0 disk
└─sda1     8:1    0   1G  0 part /boot
└─sda2     8:2    0  19G  0 part
  └─centos-root 253:0  0  17G  0 lvm  /
  └─centos-swap 253:1  0   2G  0 lvm  [SWAP]
sdb        8:16   0  30G  0 disk
└─vgdata-lv_data 253:2  0  15G  0 lvm
└─vgdata-lv_xdata 253:3  0  10G  0 lvm  /xdata
sdc        8:32   0  30G  0 disk
└─vgdata2-lv_xdata2 253:4  0  10G  0 lvm  /xdata02
sr0       11:0   1  4.4G 0 rom
```

```
[root@localhost ~]# df -Th /xdata02
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata2-lv_xdata2 xfs  10G  33M  10G   1% /xdata02
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0  20G  0 disk
└─sda1     8:1    0   1G  0 part /boot
└─sda2     8:2    0  19G  0 part
  └─centos-root 253:0  0  17G  0 lvm  /
  └─centos-swap 253:1  0   2G  0 lvm  [SWAP]
sdb        8:16   0  30G  0 disk
└─vgdata-lv_data 253:2  0  15G  0 lvm
└─vgdata-lv_xdata 253:3  0  10G  0 lvm  /xdata
sdc        8:32   0  30G  0 disk
└─vgdata2-lv_xdata2 253:4  0  10G  0 lvm  /xdata02
sr0       11:0   1  4.4G 0 rom
```

3、扩容 LV（逻辑卷）以及 XFS 文件系统 将 10G 扩容到 20G

```
[root@localhost ~]# lvextend -L 20G /dev/vgdata2/lv_xdata2
  Size of logical volume vgdata2/lv_xdata2 changed from 10.00 GiB (2560 extents) to 20.00 GiB
  (5120 extents).

Logical volume vgdata2/lv_xdata2 successfully resized.

[root@localhost ~]# xfs_growfs /xdata02
meta-data=/dev/mapper/vgdata2-lv_xdata2 isize=512    agcount=4, agsize=655360 blks
              =                      sectsz=512  attr=2, projid32bit=1
              =                      crc=1        finobt=0 spinodes=0
data        =                      bsize=4096   blocks=2621440, imaxpct=25
              =                      sunit=0     swidth=0 blks
naming      =version 2           bsize=4096   ascii-ci=0 ftype=1
```

```

log      =internal          bsize=4096   blocks=2560, version=2
        =
        sectsz=512    sunit=0 blks, lazy-count=1
realtime =none           extsz=4096   blocks=0, rtextents=0
data blocks changed from 2621440 to 5242880

```

4、验证扩容结果

```

[root@localhost ~]# df -Th /xdata02
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata2-lv_xdata2 xfs   20G  33M  20G   1% /xdata02
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0    0   20G  0 disk
└─sda1     8:1    0    1G  0 part /boot
└─sda2     8:2    0   19G  0 part
  └─centos-root 253:0    0   17G  0 lvm  /
  └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb       8:16   0   30G  0 disk
└─vgdata-lv_data 253:2    0   15G  0 lvm
└─vgdata-lv_xdata 253:3    0   10G  0 lvm  /xdata
sdc       8:32   0   30G  0 disk
└─vgdata2-lv_xdata2 253:4    0   20G  0 lvm  /xdata02
sr0      11:0   1   4.4G 0 rom

```

```

[root@localhost ~]# df -Th /xdata02
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata2-lv_xdata2 xfs  20G  33M  20G   1% /xdata02
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda       8:0    0   20G  0 disk
└─sda1     8:1    0    1G  0 part /boot
└─sda2     8:2    0   19G  0 part
  └─centos-root 253:0    0   17G  0 lvm  /
  └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb       8:16   0   30G  0 disk
└─vgdata-lv_data 253:2    0   15G  0 lvm
└─vgdata-lv_xdata 253:3    0   10G  0 lvm  /xdata
sdc       8:32   0   30G  0 disk
└─vgdata2-lv_xdata2 253:4    0   20G  0 lvm  /xdata02
sr0      11:0   1   4.4G 0 rom
[root@localhost ~]#

```

二、缩容

XFS 缩容 (20G → 9G)

注意：

XFS 不支持 shrink

只能通过“迁移法”完成缩容。

我们只对 **vgdata2 → lv_xdata2** 做缩容，
整个过程我们自己造一个新 LV 来接数据。

1、创建一个新的 LV (9G) 用于迁移

(没剩余空间了，只能 9G 了)

```
[root@localhost ~]# lvcreate -L 9G -n lv_xdata_new vgdata2  
Logical volume "lv_xdata_new" created.
```

创建效验文件（哈希值，后面用于测试文件有没有损坏）

```
[root@localhost ~]# echo "xfs-shrink-test" > /xdata02/check.txt  
[root@localhost ~]# md5sum /xdata02/check.txt  
2ef47a4b306b1ad555994df349271a8f /xdata02/check.txt  
total size is 0 speedup is 0.00  
[root@localhost ~]# echo "xfs-shrink-test" > /xdata02/check.txt  
[root@localhost ~]# md5sum /xdata02/check.txt  
2ef47a4b306b1ad555994df349271a8f /xdata02/check.txt
```

2、格式化

```
[root@localhost ~]# mkfs.xfs /dev/vgdata2/lv_xdata_new  
meta-data=/dev/vgdata2/lv_xdata_new isize=512    agcount=4, agsize=589824 blks  
          =                      sectsz=512   attr=2, projid32bit=1  
          =                      crc=1     finobt=0, sparse=0  
data     =                      bsize=4096   blocks=2359296, imaxpct=25  
          =                      sunit=0     swidth=0 blks  
naming   =version 2           bsize=4096   ascii-ci=0 ftype=1  
log      =internal log       bsize=4096   blocks=2560, version=2  
          =                      sectsz=512   sunit=0 blks, lazy-count=1  
realtime =none                extsz=4096   blocks=0, rtextents=0
```

3、挂载新 LV 到临时目录

```
[root@localhost ~]# mkdir /xdata_new  
[root@localhost ~]# mount /dev/vgdata2/lv_xdata_new /xdata_new
```

4、数据迁移（核心步骤）

```
[root@localhost ~]# yum -y install rsync  
[root@localhost ~]# rsync -avx /xdata02/ /xdata_new/  
sending incremental file list  
. /  
check.txt  
  
sent 130 bytes  received 38 bytes  336.00 bytes/sec  
total size is 16  speedup is 0.10
```

5、校验一致性（非常重要）

```
[root@localhost ~]# diff -r /xdata02 /xdata_new  
[root@localhost ~]#
```

无任何输出 = 完全一致。

6、卸载旧的 20G LV

卸载挂载点

```
[root@localhost ~]# umount /xdata02  
[root@localhost ~]# umount /xdata_new
```

删除旧的 20G LV

```
[root@localhost ~]# lvremove /dev/vgdata2/lv_xdata2  
Do you really want to remove active logical volume vgdata2/lv_xdata2? [y/n]: y  
Logical volume "lv_xdata2" successfully removed
```

```
[root@localhost ~]# lvremove /dev/vgdata2/lv_xdata2  
Do you really want to remove active logical volume vgdata2/lv_xdata2? [y/n]: y  
Logical volume "lv_xdata2" successfully removed
```

重命名新的 9G LV 为原名称

```
[root@localhost ~]# lvrename vgdata2 lv_xdata_new lv_xdata2  
Renamed "lv_xdata_new" to "lv_xdata2" in volume group "vgdata2"
```

挂载回原来的目录 /xdata02

```
[root@localhost ~]# mount /dev/vgdata2/lv_xdata2 /xdata02
```

7、最终验证扩容完成

```
[root@localhost ~]# df -Th /xdata02  
文件系统          类型 容量 已用 可用 已用% 挂载点  
/dev/mapper/vgdata2-lv_xdata2 xfs   9.0G  33M  9.0G    1% /xdata02  
[root@localhost ~]# lsblk  
NAME           MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sda            8:0    0   20G  0 disk  
├─sda1          8:1    0    1G  0 part /boot  
└─sda2          8:2    0   19G  0 part  
  ├─centos-root 253:0    0   17G  0 lvm   /  
  └─centos-swap 253:1    0    2G  0 lvm   [SWAP]  
sdb            8:16   0   30G  0 disk  
└─vgdata-lv_data 253:2    0   15G  0 lvm  
└─vgdata-lv_xdata 253:3    0   10G  0 lvm   /xdata  
sdc            8:32   0   30G  0 disk  
└─vgdata2-lv_xdata2 253:5    0    9G  0 lvm   /xdata02  
sr0           11:0   1  4.4G  0 rom
```

```
[root@localhost ~]# mount /dev/vgdata2/lv_xdata2 /xdata02
[root@localhost ~]# df -Th /xdata02
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vgdata2-lv_xdata2 xfs  9.0G  33M  9.0G   1% /xdata02
[root@localhost ~]# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0  20G  0 disk 
└─sda1     8:1    0   1G  0 part /boot
└─sda2     8:2    0  19G  0 part 
  ├─centos-root 253:0    0  17G  0 lvm  /
  └─centos-swap 253:1    0   2G  0 lvm  [SWAP]
sdb        8:16   0  30G  0 disk 
└─vgdata-lv_data 253:2    0  15G  0 lvm 
└─vgdata-lv_xdata 253:3    0  10G  0 lvm  /xdata
sdc        8:32   0  30G  0 disk 
└─vgdata2-lv_xdata2 253:5    0   9G  0 lvm  /xdata02
sr0       11:0   1  4.4G  0 rom 
[root@localhost ~]#
```

缩容后再次计算 MDS

```
[root@localhost ~]# md5sum /xdata02/check.txt
2ef47a4b306b1ad555994df349271a8f  /xdata02/check.txt
[root@localhost ~]# cat /xdata02/check.txt
xfs-shrink-test
```

```
[root@localhost ~]# md5sum /xdata02/check.txt
2ef47a4b306b1ad555994df349271a8f  /xdata02/check.txt
[root@localhost ~]# cat /xdata02/check.txt
xfs-shrink-test
[root@localhost ~]#
```

可以看到和缩容前一模一样

缩容练习完美收官

