

# LVM:

## Extending Partition Space:

### **When to Extend:**

- Running out of space on critical volumes.
- Allocating additional storage to applications or databases.

### **Steps:**

1. **Identify Available Space:** Use `fdisk -l` or `lsblk` or `lvs` to check for unallocated space.
2. **Resize Partition:** Utilize tools like `lvextend` to extend the partition.
3. **Resize Filesystem:** Apply `resize2fs` for ext4 or `xfs_growfs` for XFS to adjust the filesystem size.

**Note:** Always back up data before making changes.

## Extending partition workflow:

**Extending /data1(xfs) Partition to 1 GB and /data2(ext4) Partition to 500M**

### **Step 1: Verify Available Free Space in Volume Group:**

Before extending the logical volume, ensure that the volume group (`appvg`, `vg2`) has sufficient free space.

```

root@localhost ~]# lvs
  LV   VG     Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv01 appvg -wi-ao----  2.00g
  root  rhel   -wi-ao---- 16.41g
  swap  rhel   -wi-ao----  2.00g
  lv02 vg2    -wi-ao----  2.02g
[root@localhost ~]#
[root@localhost ~]# vgs
  VG #PV #LV #SN Attr  VSize  VFree
  appvg  2   1   0 wz--n-  4.99g  2.99g
  rhel   1   2   0 wz--n- 18.41g   0
  vg2    1   1   0 wz--n- <3.00g 996.00m
[root@localhost ~]#

```

Look for the **VFree** column to confirm that free space is available in **appvg** and **vg2**.

## Step 2: Extend the Logical Volume

If there is sufficient free space, extend the logical volume (**lv01**) to add 1 GB and (**lv02**) to 500M.

### Resize the Filesystem

After extending the logical volume, resize the filesystem to utilize the new space.

- For **ext4** filesystems:

```
sudo resize2fs /dev/vg2/lv02
```

```

root@localhost ~]# lvextend -L +500M /dev/vg2/lv02
      Size of logical volume vg2/lv02 changed from 2.02 GiB (518 extents) to 2.51 GiB (643 extents).
      Logical volume vg2/lv02 successfully resized.
[root@localhost ~]#
[root@localhost ~]# resize2fs /dev/vg2/lv02
resize2fs 1.46.5 (30-Dec-2021)
Filesystem at /dev/vg2/lv02 is mounted on /data2; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/vg2/lv02 is now 658432 (4k) blocks long.

[root@localhost ~]#

```

- For **XFS** filesystems:

```
sudo xfs_growfs /dev/appvg/lv01
```

```
[root@localhost ~]#
[root@localhost ~]# lvextend -L +1G /dev/appvg/lv01
  Size of logical volume appvg/lv01 changed from 2.00 GiB (512 extents) to 3.00 GiB (768 extents).
  Logical volume appvg/lv01 successfully resized.
[root@localhost ~]#
[root@localhost ~]# xfs_growfs /dev/appvg/lv01
meta-data=/dev/mapper/appvg-lv01 isize=512    agcount=4, agsize=65536 blks
          =                      sectsz=512  attr=2, projid32bit=1
          =                      crc=1    finobt=1, sparse=1, rmapbt=0
          =                      reflink=1   bigtime=1 inobtcount=1 nrext64=0
data     =                      bsize=4096   blocks=262144, imaxpct=25
          =                      sunit=0    swidth=0 blks
naming   =version 2           bsize=4096   ascii-ci=0, ftype=1
log      =internal log        bsize=4096   blocks=16384, version=2
          =                      sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                extsz=4096   blocks=0, rtextents=0
data blocks changed from 262144 to 786432
[root@localhost ~]#
```

### Step 3: Verify the Changes

Check the new size of the logical volume and the filesystem.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/appvg-lv01	3.0G	54M	2.9G	2%	/data1
/dev/mapper/vg2-lv02	2.5G	8.0K	2.3G	1%	/data2

**Note:** If the volume group does not have sufficient free space, you may need to add a new physical volume to the volume group before extending the logical volume.

## **Reducing Partition Space:**

### **When to Reduce:**

- Reclaiming space for other partitions.
- Preparing for disk migrations or reallocations.

### **Steps:**

1. **Unmount Partition:** Ensure the partition is unmounted using `umount`.
2. **Check Filesystem:** Run `e2fsck -f` to check for errors.
3. **Resize Filesystem:** Use `resize2fs` to shrink the filesystem.
4. **Adjust Partition Size:** Employ `lvreduce` to reduce the partition size.

**Caution:** Shrinking partitions can lead to data loss if not done correctly.

## **Reducing partition workflow:**

### **Reducing /data2(ext4) Partition to 2G**

#### **Step 1: Unmount the Partition**

```
[root@localhost ~]
```

```
[root@localhost ~]# umount /data2
[root@localhost ~]#
[root@localhost ~]# lsblk | grep vg2
└─vg2-lv02      253:3      0  2.5G  0 lvm
[root@localhost ~]#
[root@localhost ~]#
```

## Step 2: Check and Repair the Filesystem

- Run Filesystem Check:

```
[root@localhost ~]
```

```
[root@localhost ~]#
[root@localhost ~]# e2fsck -f /dev/vg2/lv02
e2fsck 1.46.5 (30-Dec-2021)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg2/lv02: 21/172032 files (0.0% non-contiguous), 28723/658432 blocks
[root@localhost ~]#
```

The **-f flag** forces a check even if the filesystem seems clean. This step is crucial before resizing

## Step 3: Resize the Filesystem

- Shrink the Filesystem:

```
[root@localhost ~]# resize2fs /dev/vg2/lv02 2G
resize2fs 1.46.5 (30-Dec-2021)
Resizing the filesystem on /dev/vg2/lv02 to 524288 (4k) blocks.
The filesystem on /dev/vg2/lv02 is now 524288 (4k) blocks long.

[root@localhost ~]#
```

This command **resizes the filesystem to 2 GB**. Ensure that the filesystem is smaller than the logical volume size before proceeding.

## Step 4: Reduce the Logical Volume Size

```
[root@localhost ~]# lvreduce -L -500M /dev/vg2/lv02
File system ext4 found on vg2/lv02.
File system size (2.00 GiB) is smaller than the requested size (2.02 GiB).
File system reduce is not needed, skipping.
Size of logical volume vg2/lv02 changed from 2.51 GiB (643 extents) to 2.02 GiB (518 extents).
Logical volume vg2/lv02 successfully resized.

[root@localhost ~]#
```

This command **reduces the logical volume size to 500 MB**. Be cautious, as this operation is irreversible and can lead to data loss if not done correctly

## Step 5: Remount the Partition

### Mount the Partition:

```
root@localhost:~#
[root@localhost ~]#
[root@localhost ~]# mount /data2
[root@localhost ~]#
[root@localhost ~]# lsblk | grep vg2
└─vg2-lv02      253:3    0     2G  0 lvm   /data2
[root@localhost ~]#
```

## Step 6: Verify the Changes

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# df -h /data2
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/vg2-lv02  2.0G   12K  1.8G   1% /data2
[root@localhost ~]#
[root@localhost ~]# lvs
  LV   VG     Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%S
  lv01 appvg -wi-ao----  3.00g
  root rhel  -wi-ao---- 16.41g
  swap rhel  -wi-ao----  2.00g
  lv02 vg2   -wi-ao----  2.02g
[root@localhost ~]#
[root@localhost ~]# vgs
  VG     #PV #LV #SN Attr   VSize  VFree
  appvg    2    1    0 wz--n-  4.99g  1.99g
  rhel     1    2    0 wz--n- 18.41g      0
  vg2      1    1    0 wz--n- <3.00g 996.00m
[root@localhost ~]#
```

*Ensure that the partition is mounted correctly and the size reflects the desired 2 GB.*

## Reducing `/data1(xfs)` Partition to 2G

The primary reason **XFS doesn't support shrinking** is its internal structure, which includes a central log and allocation groups. Shrinking would require moving data blocks, potentially disrupting the file system's integrity. Additionally, there has been minimal demand for shrinking capabilities, as storage needs typically grow rather than shrink.

## Workaround: Recreate the File System

To reduce the size of an XFS file system, you can follow these steps:

## Step 1: Backup Data:

Use **xfsdump** to create a **backup of the XFS filesystem**. This utility is designed for backing up and restoring XFS filesystems and supports incremental backups.

```
root@localhost ~]#
[root@localhost ~]# df -h /data1
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/appvg-lv01  3.0G   54M  2.9G  2% /data1
[root@localhost ~]#
[root@localhost ~]# lvs
  LV   VG     Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync
  lv01 appvg -wi-ao----  3.00g
    root rhel  -wi-ao---- 16.41g
    swap rhel  -wi-ao----  2.00g
  lv02 vg2   -wi-ao----  2.02g
[root@localhost ~]#
```

```
root@localhost /
[root@localhost /]#
[root@localhost /]# xfsdump -f /data1bkp.dump /data1
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.1.12 (dump format 3.0) - type ^C for status and control
=====
===== dump label dialog =====
please enter label for this dump session (timeout in 300 sec)
-> backup
session label entered: "backup"
-----
----- end dialog -----
xfsdump: level 0 dump of localhost.localdomain:/data1
```

## Step 2: Unmount the File System:

Unmount the filesystem to ensure data consistency during the resizing process.

```
root@localhost:/  
[root@localhost /]# umount /data1  
[root@localhost /]#  
[root@localhost /]# lsblk | grep appvg  
└─appvg-lv01 253:2      0      3G  0 lvm  
└─appvg-lv01 253:2      0      3G  0 lvm  
[root@localhost /]#
```

- If the filesystem is busy, identify and stop the processes using it.
- Use `lsof /data1` or `fuser /data1` to find and terminate these processes if necessary.

### Step 3: Delete the Logical Volume (LV)

Remove the existing logical volume to free up space for the new, smaller volume.

```
root@localhost:/  
[root@localhost /]#  
[root@localhost /]# lvremove /dev/appvg/lv01  
Do you really want to remove active logical volume appvg/lv01? [y/n]: y  
Logical volume "lv01" successfully removed.  
[root@localhost /]#
```

### Step 4: Create a New Logical Volume with Reduced Size

Create a new logical volume with the desired size.

```
[root@localhost ~]# lvcreate -L 2G -n lv01 appvg
WARNING: xfs signature detected on /dev/appvg/lv01 at offset 0. Wipe it? [y/n]: y
Wiping xfs signature on /dev/appvg/lv01.
Logical volume "lv01" created.
[root@localhost ~]#
[root@localhost ~]# lvs
  LV   VG Attr       LSize  Origin Data%  Meta%  Move Log Cpy%Sync Convert
  lv01 appvg -wi-a----  2.00g
  root  rhel -wi-ao--- 16.41g
  swap  rhel -wi-ao---  2.00g
  lv02  vg2  -wi-ao---  2.02g
[root@localhost ~]#
```

- This command creates a **2GB logical volume named `lv01`** in the volume group **appvg**.
- Adjust the size (**-L 2G**) as needed.

## Step 5: Format the New Logical Volume with XFS and Mount the New Filesystem

Create a new XFS filesystem on the newly created logical volume and Mount the newly created filesystem to make it accessible.

```
[root@localhost ~]#
[root@localhost ~]# mkfs.xfs /dev/appvg/lv01
meta-data=/dev/appvg/lv01      isize=512    agcount=4, agsize=131072 blks
                               =          sectsz=512  attr=2, projid32bit=1
                               =          crc=1    finobt=1, sparse=1, rmapbt=0
                               =          reflink=1 bigtime=1 inobtcount=1 nrext64=0
data     =          bsize=4096   blocks=524288, imaxpct=25
                               =          sunit=0    swidth=0 blks
naming   =version 2      bsize=4096   ascii-ci=0, ftype=1
log      =internal log   bsize=4096   blocks=16384, version=2
                               =          sectsz=512  sunit=0 blks, lazy-count=1
realtime =none           extsz=4096   blocks=0, rtextents=0
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# mount /dev/appvg/lv01 /data1
[root@localhost ~]#
[root@localhost ~]#
```

- This command formats `/dev/appvg/lv01` with the XFS filesystem.
- Alternatively, you can use `mount -a` if the entry exists in `/etc/fstab`.
- Ensure that the mount point `/data1` is empty before mounting.

## Step 7: Restore the Backup

Restore the data from the backup to the new filesystem.

```
[root@localhost /]# xfsrestore -f /data1bckp.dump /data1
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 3.1.12 (dump format 3.0) - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: localhost.localdomain
xfsrestore: mount point: /data1
xfsrestore: volume: /dev/mapper/appvg-lv01
xfsrestore: session time: Wed Jun 4 14:05:20 2025
```

- This command restores the contents of `/data1bckp.dump` to `/data1`.
- Check the data in `/data1`

## Important Notes:

- **Backup Data:** Always back up important data before performing operations that modify disk partitions.
- **Filesystem Compatibility:** Shrinking is not supported on XFS filesystems. Attempting to shrink an XFS filesystem will result in an error.
- **Data Integrity:** Improperly reducing the size of a filesystem or logical volume can result in data loss. Ensure that the filesystem is smaller than the logical volume before reducing its size.

- **Online vs. Offline Operations:**

- **Extending:** Can be done online (while the filesystem is mounted).
- **Reducing:** Requires the filesystem to be unmounted.