

## RAID 5:

### What is RAID 5 (Striping with Distributed Parity)?

- **Definition:** RAID 5 Splits(Stripes) data across three or more disks with **parity information** distributed among the disks. Parity allows data recovery if one disk fails.
- **Purpose:** Offers a **balance of performance, storage efficiency, and fault tolerance.**
- **Tradeoff:** Can only tolerate **one disk failure**. Rebuild time can be long, and write performance is slightly lower than RAID 0 due to parity calculation.

### Why RAID 5?

- **Efficient Redundancy:** Unlike RAID 1, you get usable capacity of  $(N - 1)$  disks (e.g., **3x1TB = 2TB usable**), while still protecting against a single disk failure.
- **Balanced Performance:**
  - **Reads:** Fast, like RAID 0 – data is striped.
  - **Writes:** Slightly slower than RAID 0 – parity must be calculated and written.
- **Fault Tolerance:**
  - If **one disk fails**, the array can still operate.
  - **Parity data** allows full recovery of the lost disk's data.
- **Cost-Effective:** More storage efficiency than mirroring, ideal for file servers and read-intensive systems.

To configure **RAID 5 (Striping with Distributed Parity)** on Linux:

1. LVM (Logical Volume Manager)
2. MDADM (Multiple Device Administration)

## **RAID 5 Striping with Parity using LVM – Full Workflow**

### **Step 1: Add New Disks & Initialize as PVs**

After attaching your disks (e.g., `/dev/sdb`, `/dev/sdc`, `/dev/sdd`), convert them into **LVM Physical Volumes**.

Command: `pvcreate /dev/sdb /dev/sdc /dev/sdd`

```
root@localhost ~]# lsblk | grep -E 'sdb|sdc|sdd'
sdb          8:16    0      3G  0 disk
sdc          8:32    0      3G  0 disk
sdd          8:48    0      3G  0 disk
[root@localhost ~]#
[root@localhost ~]# pvcreate /dev/sdb /dev/sdc /dev/sdd
  Physical volume "/dev/sdb" successfully created.
  Physical volume "/dev/sdc" successfully created.
  Physical volume "/dev/sdd" successfully created.
[root@localhost ~]#
```

#### **Explanation:**

This command initializes the raw disks and marks them as **LVM-compatible physical volumes (PVs)**. LVM cannot use disks unless they are first converted to PVs. This is the foundation for building a volume group.

Check with:

```
root@localhost:~ [root@localhost ~]# pvs
PV          VG   Fmt Attr PSize PFree
/dev/sda3   rhel lvm2 a-- 18.41g    0
/dev/sdb      lvm2 ---  3.00g 3.00g
/dev/sdc      lvm2 ---  3.00g 3.00g
/dev/sdd      lvm2 ---  3.00g 3.00g
[root@localhost ~]#
```

## Step 2: Create a Volume Group (VG)

Create a **Volume Group (VG)** that includes the three physical volumes

**Command:** `vgcreate raidvg /dev/sdb /dev/sdc /dev/sdd`

```
root@localhost:~ [root@localhost ~]# vgcreate raidvg /dev/sdb /dev/sdc /dev/sdd
  Volume group "raidvg" successfully created
[root@localhost ~]#
[root@localhost ~]# vgs raidvg
  VG      #PV #LV #SN Attr   VSize  VFree
  raidvg     3    0    0 wz--n- <8.99g <8.99g
[root@localhost ~]#
```

### Explanation:

This creates a Volume Group named `raidvg` using the three physical volumes

## Step 3: Create a RAID 5 Logical Volume

To stripe across the 3 PVs:

**Command:** `lvcreate -L 5G --type raid5 -i 2 -n raidlv raidvg`

```

root@localhost:~#
[root@localhost ~]# lvcreate -L 5G --type raid5 -i 2 -n raidlv raidvg
  Using default stripesize 64.00 KiB.
  Logical volume "raidlv" created.
[root@localhost ~]#
[root@localhost ~]# lvdisplay /dev/raidvg/raidlv
--- Logical volume ---
LV Path              /dev/raidvg/raidlv
LV Name             raidlv
VG Name             raidvg
LV UUID             eykvvR-jHGT-S8ef-7et6-8y19-z8Zb-EqKUQs
LV Write Access     read/write
LV Creation host, time localhost.localdomain, 2025-07-08 14:18:00 +0530
LV Status           available
# open              0
LV Size             5.00 GiB
Current LE          1280
Segments            1
Allocation          inherit
Read ahead sectors auto
 - currently set to 8192
Block device        253:8

[root@localhost ~]#

```

- **lvcreate**

This is the LVM command used to create a new logical volume.

- **-L 5G**

This specifies the size of the logical volume you want to create — in this case, 5 gigabytes

- **--type raid5**

- You're telling LVM to create the logical volume using a **RAID 5 layout**.
- RAID 5 uses **striping with distributed parity**, meaning it splits data across multiple disks and adds parity for fault tolerance.
- RAID 5 can tolerate **one disk failure**

- **-i 2**

- **-i** stands for number of data stripes.
- In RAID 5, the total number of disks used = **data stripes (-i) + 1 (for parity)**.
- So **-i 2** → 2 data devices + 1 parity device = **3 physical volumes**
- If you had 4 disks, you could use **-i 3** (3 data + 1 parity)

- **raidvg**

- This is the name of the **Volume Group (VG)** in which the logical volume will be created.
- This VG should be created from **3 physical volumes (disks)**, each at least 3 GB in size

**Check with:**

**Command:**

**lvs raidvg/raidlv**

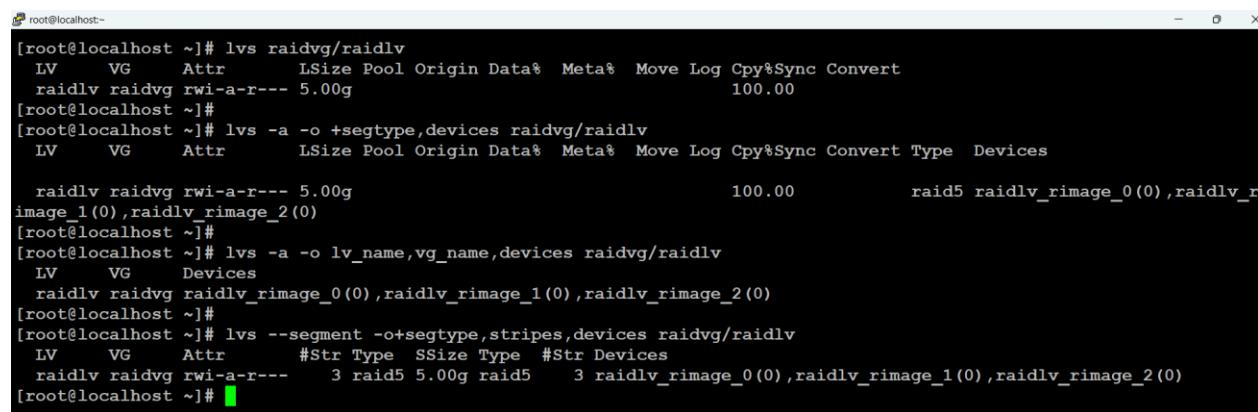
**lvs -a -o +devices**

**lvs -a -o +segtype,devices raidvg/raidlv**

**lvs -a -o lv\_name,vg\_name,devices raidvg/raidlv**

**lvs --segment -o+segtype,stripes,devices raidvg/raidlv**

**lvdisplay -m /dev/raidvg/raidlv**



```
root@localhost ~]# lvs raidvg/raidlv
  LV   VG     Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  raidlv raidvg rwi-a-r--- 5.00g
[root@localhost ~]#
[root@localhost ~]# lvs -a -o +segtype,devices raidvg/raidlv
  LV   VG     Attr       LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Type  Devices
  raidlv raidvg rwi-a-r--- 5.00g
                                         100.00          raid5 raidlv_rimage_0(0),raidlv_r
image_1(0),raidlv_rimage_2(0)
[root@localhost ~]#
[root@localhost ~]# lvs -a -o lv_name,vg_name,devices raidvg/raidlv
  LV   VG     Devices
  raidlv raidvg raidlv_rimage_0(0),raidlv_rimage_1(0),raidlv_rimage_2(0)
[root@localhost ~]#
[root@localhost ~]# lvs --segment -o+segtype,stripes,devices raidvg/raidlv
  LV   VG     #Str Type  SSize Type  #Str Devices
  raidlv raidvg rwi-a-r---    3 raid5 5.00g raid5    3 raidlv_rimage_0(0),raidlv_rimage_1(0),raidlv_rimage_2(0)
[root@localhost ~]#
```

- **lvs**: Shows info about your Logical Volumes (LVs).
- **-a**: Includes all parts—like hidden mirror segments—not just the main logical volume.
- **-o lv\_name,vg\_name,devices**: Only shows these three pieces of info:

- **lv\_name**: The volume's name (e.g., **raidlv**)
- **vg\_name**: The group it belongs to (e.g., **raidvg**)
- **devices**: The actual disk parts storing the LV (like the mirrored copies)
- created a RAID 5 logical volume (**raidlv**) with **-i 2** (2 data stripes), which means 3 devices total: 2 data, 1 parity
- LVM creates **internal sub-volumes**:
  - **rimage\_0** → data stripe 1
  - **rimage\_1** → data stripe 2
  - **rimage\_2** → parity stripe
- The **(0)** indicates each is ACTIVE and online.

## Step 4: Format & Mount the LV

**Commands:**

**mkfs.xfs /dev/raidvg/raidlv**

**mkdir /data3**

**mount /dev/raidvg/raidlv /data2**

**df -h /data3**

```
root@localhost ~]# mkfs.xfs /dev/raidvg/raidlv
meta-data=/dev/raidvg/raidlv      isize=512    agcount=8, agsize=163824 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=1310592, imaxpct=25
         =                      sunit=16     swidth=32 blks
naming   =version 2            bsize=4096   ascii-ci=0, ftype=1
log      =internal log        bsize=4096   blocks=16384, version=2
         =                      sectsz=512   sunit=16 blks, lazy-count=1
realtime =none                 extsz=4096   blocks=0, rtextents=0
[root@localhost ~]#
[root@localhost ~]# blkid /dev/raidvg/raidlv
/dev/raidvg/raidlv: UUID="87527f08-ae5b-48b8-bf3f-8d25884ca38d" TYPE="xfs"
[root@localhost ~]#
```

```

root@localhost:~]# mkdir /data3
[root@localhost ~]# mount /dev/raidvg/raidlv /data3
[root@localhost ~]# df -h /data3
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/raidvg-raidlv  5.0G   68M  4.9G  2% /data3
[root@localhost ~]#

```

## Step 5: Persist the Mount in /etc/fstab

Ensure it auto-mounts after reboot:

**Command:**

```

vi /etc/fstab
cat /etc/fstab | grep -i /data3
mount | grep /data3

```

```

root@localhost:~]# vi /etc/fstab
[root@localhost ~]# cat /etc/fstab | grep -i /data3
/dev/raidvg/raidlv      /data3  xfs    defaults        0 0
[root@localhost ~]# mount | grep /data3
/dev/mapper/raidvg-raidlv on /data3 type xfs (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=64k,sunit=128,swidth=256,noquota)
[root@localhost ~]#

```

```

[root@localhost ~]# lsblk /dev/sdb /dev/sdc /dev/sdd
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sdb            8:16   0   3G  0 disk
└─raidvg-raidlv_rmeta_0 253:2   0   4M  0 lvm
  └─raidvg-raidlv      253:8   0   5G  0 lvm  /data3
  └─raidvg-raidlv_rimage_0 253:3   0   2.5G  0 lvm
    └─raidvg-raidlv      253:8   0   5G  0 lvm  /data3
sdc            8:32   0   3G  0 disk
└─raidvg-raidlv_rmeta_1 253:4   0   4M  0 lvm
  └─raidvg-raidlv      253:8   0   5G  0 lvm  /data3
  └─raidvg-raidlv_rimage_1 253:5   0   2.5G  0 lvm
    └─raidvg-raidlv      253:8   0   5G  0 lvm  /data3
sdd            8:48   0   3G  0 disk
└─raidvg-raidlv_rmeta_2 253:6   0   4M  0 lvm
  └─raidvg-raidlv      253:8   0   5G  0 lvm  /data3
  └─raidvg-raidlv_rimage_2 253:7   0   2.5G  0 lvm
    └─raidvg-raidlv      253:8   0   5G  0 lvm  /data3

```

## **Final Notes:**

- RAID 5 with LVM provides data redundancy and performance without sacrificing too much usable space.
- It can tolerate one disk failure and still function.
- Write speed is slightly reduced due to parity calculation, but read speed is good.
- It's a flexible and modern approach, combining LVM's manageability with RAID's reliability.