RAID Simulation with Loop Devices (mdadm)

Objective

Simulate and understand the behavior of the following RAID types using loop devices:

RAID 0 (Striping)
RAID 1 (Mirroring)
RAID 5 (Striping with Parity)
RAID 10 (Mirror of Stripes)

Preparation

1. Create working directory and virtual disk images

We need at least 8 disk images:
RAID 0: 2 devices
RAID 1: 2 devices
RAID 5: 3 devices
RAID 10: 4 devices
Total: 11 disks needed

To simplify reuse and testing, we will prepare **12 disk image files**.

```
mkdir -p ~/raid-lab && cd ~/raid-lab
```

```
for i in {1..12}; do
  dd if=/dev/zero of=disk${i}.img bs=1M count=100
done
```

Command: mkdir -p ~/raid-lab && cd ~/raid-lab

Component	Description
mkdir	**Make Directory** – used to create one or more directories.
-p	**Parent** – tells `mkdir` to create parent directories as needed (e.g., it won't error if the directory already exists).
~/raid-lab	**Home-relative path** – creates the `raid-lab` directory inside the current user's home directory.
&&	**Logical AND operator** – runs the second command only if the first (`mkdir`) is successful.
cd	**Change Directory** – changes the current working directory.
~/raid-lab	**Target path** – navigates into the `raid-lab` directory inside the user's home.

for i in {1..12}; do dd if=/dev/zero of=disk${i}.img bs=1M count=100 done

Component	Description
for i in {1..12}	**Loop statement** – initiates a `for` loop that iterates `i` from 1 to 12.
do	**Loop body start** – marks the beginning of the commands to run in each loop iteration.
dd	**Data Duplicator** – a low-level command-line utility to copy and convert raw data.
if=/dev/zero	**Input File** – uses `/dev/zero`, a special file that provides an endless stream of null (zero) bytes.
of=disk${i}.img	**Output File** – writes the output to a file named `disk1.img`, `disk2.img`, etc., depending on the current value of `i`.
bs=1M	**Block Size** – sets the read and write block size to 1 Megabyte.
count=100	**Count** – specifies to copy 100 blocks of 1M each, creating a 100MB file.
done	**Loop end** – signifies the end of the `for` loop.

```
tiago-paquete@Linux:~$ mkdir -p ~/raid-lab && cd ~/raid-lab
tiago-paquete@Linux:~/raid-lab$ for i in {1..12}; do
> dd if=/dev/zero of=disk${i}.img bs=100 count=100
> done
================================================================================
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000378404 s, 26.4 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000285611 s, 35.0 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.00022997 s, 43.5 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000240122 s, 41.6 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000216519 s, 46.2 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000225764 s, 44.3 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000203164 s, 49.2 MB/s
100+0 records in
100+0 records out
```

```
10000 bytes (10 kB, 9.8 KiB) copied, 0.000217891 s, 45.9 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000199675 s, 50.1 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000245585 s, 40.7 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000196784 s, 50.8 MB/s
100+0 records in
100+0 records out
10000 bytes (10 kB, 9.8 KiB) copied, 0.000242045 s, 41.3 MB/s
```

```
tiago-paquete@Linux:~/raid-lab$ ls -l
total 144
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk1.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk10.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk11.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk12.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk2.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk3.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk4.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk5.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk6.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk7.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk8.img
-rw-rw-r-- 1 tiago-paquete tiago-paquete 10000 May  4 09:15 disk9.img
```

2. Attach disk images to loop devices

We let the system assign the next available loop devices and capture their mappings.

```
for i in {1..12}; do
  sudo losetup -fP --show disk${i}.img
done
```

To confirm assignments:

```
losetup -a | grep raid-lab
```

Record the output. You will use the loop device names (e.g., /dev/loop21, /dev/loop22, ...) in the RAID creation steps.

Command: for i in {1..12}; do sudo losetup -fP --show disk${i}.img; done

Component	Description
for	Starts a loop in the shell. Used to iterate over a sequence of values.
i	Loop variable – will take on values from 1 to 12 in this case.
in {1..12}	Bash sequence expression – expands to the numbers 1 through 12.
do	Marks the beginning of the block of commands to execute in each iteration of the loop.
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
losetup	Tool to set up and control loop devices (virtual block devices backed by regular files).
-f	Finds the first unused loop device automatically.
-P	Forces kernel to scan for partitions on the loop device (used with partitioned image files).
--show	Displays the name of the loop device that was set up.
disk${i}.img	Filename of the disk image – uses shell variable expansion to insert the current value of `i` (e.g., `disk1.img`, `disk2.img`, ...).
done	Ends the `for` loop block.

Command: losetup -a | grep raid-lab

Component	Description
losetup	Tool to manage loop devices. Without options, it shows the current loop device setup.
-a	Displays all loop devices and the files they are associated with.
grep	Searches input for lines matching a pattern.
raid-lab	Search pattern – filters only the lines that contain the string `raid-lab`. Likely used to locate loop devices related to a RAID lab setup.

```
tiago-paquete@Linux:~/raid-lab$ for i in {1..12}; do
> sudo losetup -fP --show disk${i}.img
> done
==========================================================================
/dev/loop28
losetup: disk1.img: Warning: file does not fit into a 512-byte sector;
the end of the file will be ignored.
…
…
==========================================================================
```

⚠️ Explanation of the Warning

This message comes from losetup when:
The size of the file (disk*.img) is not a multiple of 512 bytes, which is the default sector size for block devices.

losetup aligns loop devices on 512-byte boundaries.

Since the files are 10,000 bytes, and 10,000 ÷ 512 = 19.53 (not whole), some trailing bytes are ignored by the kernel.

This warning does not block usage, but for RAID or partition work, it is better to avoid this warning.

2.1. ⚠️ Create images with a multiple of 512 bytes

Create images with a multiple of 512 bytes, such as:

- 512 × 20 = 10,240 bytes

So, update your dd command like this:

```
for i in {1..12}; do
  dd if=/dev/zero of=disk${i}.img bs=512 count=20
done
```

Then detach the old loop devices and reattach:

```
sudo losetup -D   # Detach all loop devices

for i in {1..12}; do
  sudo losetup -fP --show disk${i}.img
done
```

Command: for i in {1..12}; do dd if=/dev/zero of=disk${i}.img bs=512 count=20; done

Component	Description
for	Bash loop keyword to start a **for loop**.
i	Loop variable that takes on each value in the given range.
in {1..12}	Bash brace expansion to generate a sequence of numbers from 1 to 12.
do	Indicates the beginning of the loop body.
dd	Command-line utility to **convert and copy files**, often used for low-level operations on raw data.
if=/dev/zero	**Input file**: special file that produces null (zero) bytes endlessly.
of=disk${i}.img	**Output file**: dynamically named image file (e.g., disk1.img, disk2.img...). `${i}` is the current loop index.
bs=512	**Block size**: sets both read and write block size to 512 bytes.
count=20	Copies only 20 blocks (20 × 512 bytes = 10 KiB per file).
done	Ends the `for` loop.

Command: sudo losetup -D

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
losetup	Utility to manage **loop devices**, which allow files to be treated as block devices.
-D	Detach **all** loop devices currently configured on the system.

Command: for i in {1..12}; do sudo losetup -fP --show disk${i}.img; done

Component	Description
for	Bash loop keyword to start a **for loop**.
i	Loop variable that takes on each value in the given range.
in {1..12}	Bash brace expansion to generate a sequence of numbers from 1 to 12.
do	Indicates the beginning of the loop body.
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges.
losetup	Utility to manage **loop devices**.
-f	Finds the first available loop device.
-P	Forces the kernel to **scan for partitions** after setup (important for image files with partitions).
--show	Displays the **name** of the loop device (e.g., `/dev/loop0`) after setup.
disk${i}.img	Image file to be associated with a loop device (e.g., disk1.img, disk2.img...).
done	Ends the `for` loop.

```
tiago-paquete@Linux:~/raid-lab$ for i in {1..12}; do
> dd if=/dev/zero of=disk${i}.img bs=512 count=20
> done
================================================================================
20+0 records in
20+0 records out
10240 bytes (10 kB, 10 KiB) copied, 0.000192139 s, 53.3 MB/s
…
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo losetup -D


tiago-paquete@Linux:~/raid-lab$ for i in {1..12}; do
```

```
>  sudo losetup -fP --show disk${i}.img
> done
==========================================================================
/dev/loop16
/dev/loop17
/dev/loop18
/dev/loop19
/dev/loop20
/dev/loop21
/dev/loop22
/dev/loop23
/dev/loop24
/dev/loop25
/dev/loop26
/dev/loop27
==========================================================================
```

RAID Configuration Exercises

RAID 0 (Striping)

Introduction:
RAID 0 splits (stripes) data evenly across two or more drives with no redundancy. It focuses purely on performance.

Characteristics:
No fault tolerance – if one disk fails, all data is lost.
High performance for read/write.
Full combined capacity of all disks is usable.

Minimum Devices: 2
Use Cases:
Temporary data
High-speed cache
Gaming systems

Visual Representation (striping):

Disk 1: [A1][A3][A5][A7]
Disk 2: [A2][A4][A6][A8]

To configure:
sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/loop16 /dev/loop17

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --create --verbose /dev/md0
--level=0 --raid-devices=2 /dev/loop16 /dev/loop17
==========================================================================
[sudo] password for tiago-paquete:
device /dev/loop16 is too small (10K) for required metadata!
mdadm: ddf: Cannot create this array on device /dev/loop16 - a container
is required.
mdadm: Cannot create this array on device /dev/loop16
mdadm: device /dev/loop16 not suitable for any style of array
==========================================================================
```

⚠️ This output indicates that mdadm was **unable to create the RAID 0 array** because the loop devices /dev/loop16 and /dev/loop17 are **still too small** for RAID metadata, even though they're aligned to 512 bytes.

RAID 0 (Striping) ⚠️ Troubleshooting

```
rm -f disk*.img  # Clean up old ones

# Create 12 images of 5MB each (good for metadata + space)
for i in {1..12}; do
  dd if=/dev/zero of=disk${i}.img bs=1M count=100
done

# Detach previous loop devices
sudo losetup -D

# Reattach all images as loop devices
for i in {1..12}; do
  sudo losetup -fP --show disk${i}.img
done
```

```
tiago-paquete@Linux:~/raid-lab$ rm -f disk*.img

tiago-paquete@Linux:~/raid-lab$ for i in {1..12}; do
  dd if=/dev/zero of=disk${i}.img bs=1M count=100
> done
================================================================================
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.0546958 s, 1.9 GB/s
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.0492932 s, 2.1 GB/s
…
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo losetup -D

tiago-paquete@Linux:~/raid-lab$ for i in {1..12}; do
  sudo losetup -fP --show disk${i}.img
done
================================================================================
/dev/loop16
/dev/loop17
/dev/loop18
/dev/loop19
/dev/loop20
/dev/loop21
/dev/loop22
/dev/loop23
/dev/loop24
```

```
/dev/loop25
/dev/loop26
/dev/loop27
================================================================================
```

RAID 0 (Striping) - After Troubleshooting Steps

To configure:
sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/loop16 /dev/loop17
sudo mkfs.ext4 /dev/md0
sudo mkdir -p /mnt/raid0
sudo mount /dev/md0 /mnt/raid0

To verify:
cat /proc/mdstat
lsblk | grep md0

Command: sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/loop16 /dev/loop17

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	Tool used to manage and create **MD (Multiple Device)** RAID arrays in Linux.
--create	Tells `mdadm` to create a new RAID device.
--verbose	Enables detailed output during the creation process.
/dev/md0	The name of the new RAID device to be created.
--level=0	Specifies the RAID level, in this case **RAID 0** (striping without redundancy).
--raid-devices=2	Indicates the number of devices to include in the array (2 devices here).
/dev/loop16	First loopback device to be used as a RAID member.
/dev/loop17	Second loopback device to be used as a RAID member.

Command: sudo mkfs.ext4 /dev/md0

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mkfs.ext4	Creates a new filesystem of type **ext4** on the specified device.
/dev/md0	Target device where the ext4 filesystem will be created (the RAID array previously created).

Command: sudo mkdir -p /mnt/raid0

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mkdir	Command to create directories.
-p	Ensures that parent directories are created as needed and suppresses error if the directory already exists.
/mnt/raid0	Path where the mount point (directory) will be created to access the RAID volume.

Command: sudo mount /dev/md0 /mnt/raid0

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mount	Mounts a filesystem onto a directory (mount point).
/dev/md0	The source device (RAID array) to mount.
/mnt/raid0	The target directory where the device will be mounted and accessed.

Command: cat /proc/mdstat

Component	Description
cat	**Concatenate** – used to read and display the contents of files to standard output.
/proc/mdstat	**Virtual file** – shows the status of Linux MD (multiple device) software RAID arrays. It's part of the `/proc` virtual filesystem, which provides kernel and system information.

Command: lsblk | grep md0

Component	Description
lsblk	**List Block Devices** – displays information about block devices (like disks, partitions, and RAID volumes) in a tree-like format.
grep	**Global Regular Expression Print** – searches input lines that match a given pattern.
md0	**Pattern** – the search term passed to `grep`; here, it's looking for lines containing `md0`, which usually refers to a specific RAID device (e.g., `/dev/md0`).

Command: sudo mount /dev/md0 /mnt/raid0

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mount	**Mount** – attaches a filesystem (e.g., a RAID device or partition) to the system's directory tree.
/dev/md0	**Device File** – the RAID device to be mounted. Represents a software RAID array managed by `mdadm`.
/mnt/raid0	**Mount Point** – the directory where the contents of the RAID device will be accessible after mounting.

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --create --verbose /dev/md0
--level=0 --raid-devices=2 /dev/loop16 /dev/loop17
================================================================================
mdadm: chunk size defaults to 512K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
================================================================================


tiago-paquete@Linux:~/raid-lab$ mkfs.ext4 /dev/md0
================================================================================
mke2fs 1.47.0 (5-Feb-2023)
mkfs.ext4: Permission denied while trying to determine filesystem size
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo !!
================================================================================
sudo mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 50176 4k blocks and 50176 inodes
Filesystem UUID: cbb8ed97-cf11-40a7-b6f4-6e3b98400a22
Superblock backups stored on blocks:
    32768

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
================================================================================
```

```
tiago-paquete@Linux:~/raid-lab$ sudo mkdir -p /mnt/raid0
tiago-paquete@Linux:~/raid-lab$ sudo mount /dev/md0 /mnt/raid0
tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
================================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
================================================================================

tiago-paquete@Linux:~/raid-lab$ lsblk | grep md0
================================================================================
└─md0         9:0    0   196M  0 raid0 /mnt/raid0
└─md0         9:0    0   196M  0 raid0 /mnt/raid0
================================================================================
```

RAID 1 (Mirroring)

Introduction:
RAID 1 duplicates (mirrors) data across two or more disks. If one disk fails, data is still accessible from the other.

Characteristics:
High fault tolerance
Lower write performance due to duplication
Usable space is only equal to one disk

Minimum Devices: 2

Use Cases:
Critical data
Operating system partitions
Small business storage

Visual Representation (mirroring):

Disk 1: [A1][A2][A3][A4]
Disk 2: [A1][A2][A3][A4]

To configure:
sudo mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/loop18 /dev/loop19
sudo mkfs.ext4 /dev/md1
sudo mkdir -p /mnt/raid1
sudo mount /dev/md1 /mnt/raid1

To verify:
cat /proc/mdstat
lsblk | grep md1

Command: sudo mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 /dev/loop18 /dev/loop19

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	Tool used to manage and create **MD (Multiple Device)** software RAID arrays.
--create	Instructs `mdadm` to create a new RAID array.

--verbose	Enables verbose mode; provides detailed output of the creation process.
/dev/md1	The target device name for the RAID array being created.
--level=1	Specifies the RAID level (RAID 1: mirroring).
--raid-devices=2	Defines the number of devices to be used in the RAID array.
/dev/loop18	First loop device used as a disk in the RAID array.
/dev/loop19	Second loop device used as a disk in the RAID array.

Command: sudo mkfs.ext4 /dev/md1

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mkfs.ext4	Creates a new **ext4 filesystem** on the specified device.
/dev/md1	The RAID device on which the ext4 filesystem will be created.

Command: sudo mkdir -p /mnt/raid1

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mkdir	Creates a new directory.
-p	Creates parent directories as needed (no error if existing).
/mnt/raid1	The path where the RAID array will be mounted.

Command: sudo mount /dev/md1 /mnt/raid1

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mount	Attaches a filesystem to the system’s directory tree.
/dev/md1	The device (RAID array) to be mounted.
/mnt/raid1	The mount point where the filesystem will be accessible.

Command: cat /proc/mdstat

Component	Description
cat	Displays the content of a file.
/proc/mdstat	A virtual file showing the current status of MD (RAID) devices on the system.

Command: lsblk | grep md

Component	Description
lsblk	Lists information about block devices (disks and partitions).
grep md	Filters and displays only lines that contain the text `md`, typically referring to MD devices (RAID arrays).

⚠️

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --create --verbose /dev/md1
--level=1 --raid-devices=2 /dev/loop18 /dev/loop19
================================================================================
[sudo] password for tiago-paquete:
mdadm: Note: this array has metadata at the start and
    may not be suitable as a boot device.  If you plan to
    store '/boot' on this device please ensure that
    your boot-loader understands md/v1.x metadata, or use
    --metadata=0.90
mdadm: size set to 101376K
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
================================================================================
```

❗ Explanation of the Message

Part	Meaning
Note: this array has metadata at the start	By default, mdadm writes metadata (array configuration info) at the start of each device. This can interfere with bootloaders if the RAID is used for /boot.

may not be suitable as a boot device	It warns that this RAID array might not work for booting unless your bootloader supports mdadm's metadata format.
use --metadata=0.90	Suggests using the older metadata format 0.90 if you intend to use this array for booting (because it's stored at the end of the device and supported by more bootloaders).
size set to 101376K	The usable size of the array (based on the smallest member device).
Continue creating array?	mdadm is asking for confirmation before proceeding with creating the array.

```
tiago-paquete@Linux:~/raid-lab$ sudo mkfs.ext4 /dev/md1
================================================================================
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 25344 4k blocks and 25344 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo mkdir -p /mnt/raid1
tiago-paquete@Linux:~/raid-lab$ sudo mount /dev/md1 /mnt/raid1
tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
================================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
================================================================================

tiago-paquete@Linux:~/raid-lab$ lsblk | grep md
================================================================================
└─md0         9:0    0   196M  0 raid0 /mnt/raid0
└─md0         9:0    0   196M  0 raid0 /mnt/raid0
└─md1         9:1    0    99M  0 raid1 /mnt/raid1
└─md1         9:1    0    99M  0 raid1 /mnt/raid1
================================================================================
```

RAID 5 (Striping with Parity)

Introduction:
RAID 5 stripes data and parity information across all disks. It offers fault tolerance with good performance.

Characteristics:
Can survive 1 disk failure
Read speed similar to RAID 0, slower writes due to parity calculations
Loses one disk worth of usable space

Minimum Devices: 3

Use Cases:
File servers
Archival storage
General-purpose redundancy

Visual Representation (striping + parity):

Disk 1: [A1][A4][P3]
Disk 2: [A2][P2][A6]
Disk 3: [P1][A5][A3]

P = Parity block for redundancy

To configure:
sudo mdadm --create --verbose /dev/md2 --level=5 --raid-devices=3 /dev/loop20 /dev/loop21 /dev/loop22
sudo mkfs.ext4 /dev/md2
sudo mkdir -p /mnt/raid5
sudo mount /dev/md2 /mnt/raid5

To verify:
cat /proc/mdstat
lsblk | grep md2

Command: sudo mdadm --create --verbose /dev/md2 --level=5 --raid-devices=3 /dev/loop20 /dev/loop21 /dev/loop22

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	Tool used to manage MD (multiple device) software RAID arrays on Linux.
--create	Tells `mdadm` to create a new RAID array.
--verbose	Enables detailed output to show progress and debug information.
/dev/md2	The name of the new RAID device to be created.
--level=5	Specifies RAID level 5 (striping with distributed parity).
--raid-devices=3	Indicates that 3 devices will be used in the RAID array.
/dev/loop20	First loop device used as a disk in the array.
/dev/loop21	Second loop device used as a disk in the array.
/dev/loop22	Third loop device used as a disk in the array.

Command: sudo mkfs.ext4 /dev/md2

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mkfs.ext4	Creates an ext4 filesystem on the specified device. `mkfs` stands for “make filesystem.”
/dev/md2	Target device (the RAID 5 array) on which the ext4 filesystem will be created.

Command: sudo mkdir -p /mnt/raid5

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mkdir	Command to create a new directory.
-p	Ensures parent directories are created as needed; avoids errors if the directory already exists.
/mnt/raid5	Target directory where the RAID 5 volume will be mounted.

Command: sudo mount /dev/md2 /mnt/raid5

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mount	Command to mount a filesystem or device to a directory.
/dev/md2	Source device (RAID 5 array) to be mounted.
/mnt/raid5	Mount point where the device's filesystem will be attached in the directory tree.

Command: cat /proc/mdstat

Component	Description
cat	Outputs the contents of a file to the terminal.
/proc/mdstat	Virtual file showing the current status of MD (RAID) devices. Useful to monitor RAID creation and rebuild.

Command: lsblk | grep md2

Component	Description
lsblk	Lists information about block devices in a tree-like format.
grep	Searches input text for lines that match a pattern.
md2	The pattern `grep` searches for, showing only lines related to the RAID device `/dev/md2`.

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --create --verbose /dev/md2
--level=5 --raid-devices=3 /dev/loop20 /dev/loop21 /dev/loop22
======================================================================
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 100352K
mdadm: Fail to create md2 when using /sys/module/md_mod/parameters/
new_array, fallback to creation via node
mdadm: Fail to create md2 when using /sys/module/md_mod/parameters/
new_array, fallback to creation via node
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md2 started.
======================================================================
```

```
tiago-paquete@Linux:~/raid-lab$ sudo mkfs.ext4 /dev/md2
================================================================================
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 50176 4k blocks and 50176 inodes
Filesystem UUID: e4792ef0-f3ec-4d3e-b608-ad0f09a62699
Superblock backups stored on blocks:
    32768

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done2
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo mkdir -p /mnt/raid5
tiago-paquete@Linux:~/raid-lab$ sudo mount /dev/md2 /mnt/raid5
tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
================================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md2 : active raid5 loop22[3] loop21[1] loop20[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3]
[UUU]

md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
================================================================================

tiago-paquete@Linux:~/raid-lab$ lsblk | grep md
================================================================================
└─md0         9:0    0   196M  0 raid0 /mnt/raid0
└─md0         9:0    0   196M  0 raid0 /mnt/raid0
└─md1         9:1    0    99M  0 raid1 /mnt/raid1
└─md1         9:1    0    99M  0 raid1 /mnt/raid1
└─md2         9:2    0   196M  0 raid5 /mnt/raid5
└─md2         9:2    0   196M  0 raid5 /mnt/raid5
└─md2         9:2    0   196M  0 raid5 /mnt/raid5
================================================================================
```

RAID 10 (Mirror of Stripes)

Introduction:
RAID 10 (also known as RAID 1+0) combines mirroring and striping. It mirrors data across pairs and stripes the data across those mirrors.

Characteristics:
High performance and high fault tolerance
Can survive multiple disk failures (as long as one disk per mirror survives)
Only 50% of total disk space is usable

Minimum Devices: 4

Use Cases:
Databases
High-availability systems
Virtual machine storage

Visual Representation (striped mirrors):

Disk 1: [A1][A3] <- Mirror Pair 1
Disk 2: [A1][A3]

Disk 3: [A2][A4] <- Mirror Pair 2
Disk 4: [A2][A4]

To configure:

```
sudo mdadm --create --verbose /dev/md3 --level=10 --raid-devices=4 /dev/loop23 /dev/loop24 /dev/loop25 /dev/loop26
sudo mkfs.ext4 /dev/md3
sudo mkdir -p /mnt/raid10
sudo mount /dev/md3 /mnt/raid10
```

To verify:

```
cat /proc/mdstat
lsblk | grep md3
```

Command: sudo mdadm --create --verbose /dev/md3 --level=10 --raid-devices=4 /dev/loop23 /dev/loop24 /dev/loop25 /dev/loop26

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	Tool used to create, manage, and monitor Linux software RAID arrays.
--create	Tells `mdadm` to create a new RAID array.
--verbose	Enables detailed output during the RAID creation process.
/dev/md3	The name of the RAID device to be created. It will appear as a block device once built.
--level=10	Specifies the RAID level. RAID 10 combines mirroring and striping for redundancy and performance.
--raid-devices=4	Indicates that 4 devices will be used to build the RAID array.
/dev/loop23 ... 26	The actual devices (loopback in this case) to include in the RAID array.

Command: sudo mkfs.ext4 /dev/md3

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges.
mkfs.ext4	Makes a new ext4 filesystem on the specified device. Used to format a storage device with ext4.
/dev/md3	The target device where the new ext4 filesystem will be created (the RAID device).

Command: sudo mkdir -p /mnt/raid10

Component	Description
sudo	**Superuser Do** – runs the command with administrative privileges.
mkdir	Command to create a directory.
-p	Ensures parent directories are created as needed and avoids errors if the directory already exists.
/mnt/raid10	The full path of the directory to be created for mounting the RAID device.

Command: sudo mount /dev/md3 /mnt/raid10

Component	Description
sudo	**Superuser Do** – runs the command with administrative privileges.
mount	Command to mount a filesystem to a specified directory.
/dev/md3	The RAID device to be mounted.
/mnt/raid10	The mount point where the filesystem will be accessible.

Command: cat /proc/mdstat

Component	Description
cat	Displays the contents of a file or output stream.
/proc/mdstat	A virtual file that shows the current status of all active RAID arrays on the system. Useful for monitoring.

Command: lsblk | grep md3

Component	Description
lsblk	Lists information about all available or the specified block devices.
grep md3	Filters and displays lines from the input that contain the string `md3`. Useful to locate specific devices.

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --create --verbose /dev/md3
--level=10 --raid-devices=4 /dev/loop23 /dev/loop24 /dev/loop25 /dev/
loop26
======================================================================
mdadm: layout defaults to n2
mdadm: layout defaults to n2
mdadm: chunk size defaults to 512K
mdadm: size set to 100352K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md3 started.
======================================================================

tiago-paquete@Linux:~/raid-lab$ sudo mkfs.ext4 /dev/md3
======================================================================
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 50176 4k blocks and 50176 inodes
Filesystem UUID: 2dfb55f0-78b6-4f8c-80f1-26b66be48a9a
Superblock backups stored on blocks:
    32768

Allocating group tables: done
```

```
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo mkdir -p /mnt/raid10
tiago-paquete@Linux:~/raid-lab$ sudo mount /dev/md3 /mnt/raid10
tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
================================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md3 : active raid10 loop26[3] loop25[2] loop24[1] loop23[0]
      200704 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

md2 : active raid5 loop22[3] loop21[1] loop20[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3]
[UUU]

md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
================================================================================

tiago-paquete@Linux:~/raid-lab$ lsblk | grep md
================================================================================
└─md0         9:0    0   196M  0 raid0  /mnt/raid0
└─md0         9:0    0   196M  0 raid0  /mnt/raid0
└─md1         9:1    0    99M  0 raid1  /mnt/raid1
└─md1         9:1    0    99M  0 raid1  /mnt/raid1
└─md2         9:2    0   196M  0 raid5  /mnt/raid5
└─md2         9:2    0   196M  0 raid5  /mnt/raid5
└─md2         9:2    0   196M  0 raid5  /mnt/raid5
└─md3         9:3    0   196M  0 raid10 /mnt/raid10
└─md3         9:3    0   196M  0 raid10 /mnt/raid10
└─md3         9:3    0   196M  0 raid10 /mnt/raid10
└─md3         9:3    0   196M  0 raid10 /mnt/raid10
================================================================================
```

Hot Spare (Standby Disk)

Introduction:
A *Hot Spare* is a standby disk that is not actively used during normal RAID operations but is automatically integrated into the array when an active disk fails. It increases the fault tolerance of RAID arrays by enabling **automatic recovery** without manual intervention.

Characteristics:
Not used during normal operation
Automatically replaces a failed disk in supported RAID levels (e.g., RAID 1, 5, 6, 10)
Helps minimize downtime and risk of data loss
Can be shared among multiple RAID arrays

Minimum Devices:
At least **one extra** disk in addition to the minimum required RAID disks

Use Cases:
Enterprise storage arrays
Systems requiring high availability
Unattended servers and NAS appliances

Visual Representation (RAID 5 with Hot Spare):

Disk 1: [A1][A4][P3]
Disk 2: [A2][P2][A6]
Disk 3: [P1][A5][A3]
Disk 4: [Standby] <- Hot Spare

P = Parity block for redundancy
When any disk fails, Disk 4 (Hot Spare) automatically takes its place and a rebuild begins.

To Configure (example for RAID 5 + hot spare)
Since the RAID 5 is already created and running, all you need to do is add the hot spare:

```
sudo mdadm --add /dev/md2 /dev/loop27
```

This will add /dev/loop27 to /dev/md2 as a **standby disk**.
The system will not use it unless one of the primary RAID members fails.

To Verify
1. View RAID status and check for hot spare

cat /proc/mdstat

After adding the spare, output should still show [3/3] [UUU] and list /dev/loop27 as **(S)** or spare.

2. Detailed view with mdadm

sudo mdadm --detail /dev/md2

In the output:
You should see /dev/loop27 with "spare" or "spare rebuilding" status.
RAID Devices = 3, Total Devices = 4.

Command: sudo mdadm --add /dev/md2 /dev/loop27

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	Tool used to manage **MD (Multiple Device)** software RAID arrays.
--add	Adds a new device to an existing RAID array.
/dev/md2	Target **RAID device** (md2) – the RAID array to which a device will be added.
/dev/loop27	The **loopback device** being added to the RAID array (simulates a block device using a file).

Command: cat /proc/mdstat

Component	Description
cat	Displays the content of files by printing them to the terminal.
/proc/mdstat	Virtual file showing the status of all active MD (RAID) arrays in the system.

Command: sudo mdadm --detail /dev/md2

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	Tool used to manage **MD (Multiple Device)** software RAID arrays.
--detail	Displays detailed information about a specific RAID device.
/dev/md2	The **RAID device** (md2) for which detailed status and configuration are shown.

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --add /dev/md2 /dev/loop27
=======================================================================
[sudo] password for tiago-paquete:
mdadm: added /dev/loop27
=======================================================================

tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
=======================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md3 : active raid10 loop26[3] loop25[2] loop24[1] loop23[0]
      200704 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

md2 : active raid5 loop27[4](S) loop22[3] loop21[1] loop20[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3]
[UUU]

md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
=======================================================================

tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
=======================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md3 : active raid10 loop26[3] loop25[2] loop24[1] loop23[0]
      200704 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

md2 : active raid5 loop27[4](S) loop22[3] loop21[1] loop20[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3]
[UUU]

md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
```

```
      200704 blocks super 1.2 512k chunks

unused devices: <none>
================================================================================

tiago-paquete@Linux:~/raid-lab$ sudo mdadm --detail /dev/md2
================================================================================
/dev/md2:
           Version : 1.2
     Creation Time : Sun May  4 11:32:53 2025
        Raid Level : raid5
        Array Size : 200704 (196.00 MiB 205.52 MB)
     Used Dev Size : 100352 (98.00 MiB 102.76 MB)
      Raid Devices : 3
     Total Devices : 4
       Persistence : Superblock is persistent

       Update Time : Sun May  4 12:10:03 2025
             State : clean
    Active Devices : 3
   Working Devices : 4
    Failed Devices : 0
     Spare Devices : 1

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : Linux:2  (local to host Linux)
              UUID : b6exxxxd:7bxxxf37:bxxxx10a:fxxxx8c3
            Events : 19

    Number   Major   Minor   RaidDevice State
       0       7       20        0      active sync   /dev/loop20
       1       7       21        1      active sync   /dev/loop21
       3       7       22        2      active sync   /dev/loop22

       4       7       27        -      spare   /dev/loop27
================================================================================
```

Disk Failure Simulation

Simulate a Disk Failure to Test Spare
sudo mdadm --fail /dev/md2 /dev/loop21

Then monitor:
watch cat /proc/mdstat

You'll see the spare (/dev/loop27) being used for automatic rebuild.

Command: sudo mdadm --fail /dev/md2 /dev/loop21

Component	Description
sudo	**Superuser Do** – runs the command that follows with elevated (root) privileges. Required when the command needs administrative access.
mdadm	The **MD (Multiple Device) Admin** tool – used for managing Linux MD (RAID) arrays, such as creating, assembling, or modifying RAID sets.
--fail	Tells `mdadm` to mark a device as **faulty** in a RAID array, simulating device failure. This is useful for testing RAID recovery or fault tolerance.
/dev/md2	The **RAID array device** – the MD device (e.g., RAID-1 or RAID-5) you are targeting.
/dev/loop21	The **member device** of the RAID array – in this case, a loopback device used to simulate a disk. It will be marked as failed in the array.

Command: watch cat /proc/mdstat

Component	Description
watch	A utility that **runs a command repeatedly**, displaying the output and updating it at regular intervals (default: every 2 seconds). Useful for monitoring.
cat	The **concatenate** command – used here to read and display the contents of a file.
/proc/mdstat	A **virtual file** that shows the current status of all MD (RAID) devices, including sync status, degraded arrays, and rebuilding progress.

```
tiago-paquete@Linux:~/raid-lab$ sudo mdadm --fail /dev/md2 /dev/loop21
================================================================================
mdadm: set /dev/loop21 faulty in /dev/md2
================================================================================

tiago-paquete@Linux:~/raid-lab$ watch cat /proc/mdstat
================================================================================
Every 2.0s: cat /proc/mdstat
Linux: Sun May  4 12:18:47 2025

Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md3 : active raid10 loop26[3] loop25[2] loop24[1] loop23[0]
      200704 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

md2 : active raid5 loop27[4] loop22[3] loop21[1](F) loop20[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3]
[UUU]

md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
================================================================================
```

Cleanup Procedure

Current lab state (recap)

Layer	What you created	Devices / paths involved	Status (per last `cat /proc/mdstat`)
RAID 0	array /dev/md0 mounted at /mnt/raid0	/dev/loop16, /dev/loop17	healthy (`[UU]`)
RAID 1	/dev/md1 → /mnt/raid1	/dev/loop18, /dev/loop19	healthy (`[UU]`)
RAID 5 + hot-spare	/dev/md2 → /mnt/raid5	Active: `/dev/loop20`, `/dev/loop22`, hot-spare: `/dev/loop27`, Faulty: `/dev/loop21`	degraded but clean (`[UUU]` + one **(F)**)
RAID 10	/dev/md3 → /mnt/raid10	/dev/loop23, /dev/loop24, /dev/loop25, /dev/loop26	healthy (`[UUUU]`)
Loop devices	/dev/loop16 … /dev/loop27	all mapped to ~/raid-lab/disk{1..12}.img	attached
Image files	12 × 100 MiB in `~/raid-lab`	disk1.img … disk12.img	present
Mount points	/mnt/raid0, /mnt/raid1, /mnt/raid5, /mnt/raid10	in use	mounted

1. Unmount RAID mount points

umount -l /mnt/raid0 2>/dev/null
umount -l /mnt/raid1 2>/dev/nulld
umount -l /mnt/raid5 2>/dev/null
umount -l /mnt/raid10 2>/dev/null

This unmounts the mounted RAID arrays from the filesystem.

The **-l** (lazy) option ensures that if a process is still using the mount point, it will be unmounted as soon as it's no longer in use.

2>/dev/null suppresses any error messages (e.g., if the mount point is already unmounted).

```
tiago-paquete@Linux:~/raid-lab$ sudo umount -l /mnt/raid0 2>/dev/null
==========================================================================
[sudo] password for tiago-paquete:
==========================================================================
tiago-paquete@Linux:~/raid-lab$ sudo umount -l /mnt/raid1 2>/dev/null
tiago-paquete@Linux:~/raid-lab$ sudo umount -l /mnt/raid5 2>/dev/null
tiago-paquete@Linux:~/raid-lab$ sudo umount -l /mnt/raid10 2>/dev/null
```

2. Stop and remove all active md (RAID) devices

```
for md in /dev/md{3,2,1,0}; do
  mdadm --stop "$md" 2>/dev/null
  mdadm --remove "$md" 2>/dev/null
done
```

This loop attempts to stop and remove all the created RAID arrays (/dev/md0 to /dev/md3).

--stop halts the array.
--remove detaches the array from the kernel so it can no longer be accessed.

Errors are suppressed silently with 2>/dev/null (e.g., if an array was already stopped).

```
tiago-paquete@Linux:~/raid-lab$ for md in /dev/md{3,2,1,0}; do
> mdadm --stop "$md" 2>/dev/null
> mdadm --remove "$md" 2>/dev/null
> done
```

3. Zero out RAID metadata from all loop devices

```
for dev in /dev/loop{16..27}; do
  mdadm --zero-superblock "$dev" 2>/dev/null
done
```

This wipes the RAID metadata stored in the superblock of each loop device.

Essential for ensuring that the disks can be reused in future RAID configurations.

Again, errors are suppressed (e.g., if a device was already wiped or not in use).

```
tiago-paquete@Linux:~/raid-lab$ for dev in /dev/loop{16..27}; do
> mdadm --zero-superblock "$dev" 2>/dev/null
> done
```

4. Detach all loop devices linked to raid-lab disk images

losetup -a | grep raid-lab | cut -d: -f1 | xargs -r -n1 losetup -d

losetup -a lists all loop devices and their backing files.

grep raid-lab filters only the ones associated with your lab disk images.

cut -d: -f1 extracts the device name (e.g., /dev/loop16).

xargs -r -n1 losetup -d detaches each one safely.

```
tiago-paquete@Linux:~/raid-lab$ sudo losetup -a | grep raid-lab | cut
-d: -f1 | xargs -r -n1 sudo losetup -d
```

5. Delete all mount points and lab files

rm -rf /mnt/raid{0,1,5,10}
rm -rf ~/raid-lab

Removes the empty mount directories you created.
Deletes the ~/raid-lab directory and all disk image files used for the simulation.

```
tiago-paquete@Linux:~/raid-lab$ sudo rm -rf /mnt/raid{0,1,5,10}
tiago-paquete@Linux:~/raid-lab$ rm -rf ~/raid-lab
```

6. Final verification steps

echo -e "\n>>> losetup check (should be empty):"
losetup -a | grep raid-lab || echo " OK – no raid-lab loops"

Confirms that no loop devices remain attached to your RAID lab disk images.

echo -e "\n>>> md status (should list none):"
cat /proc/mdstat

Displays current RAID status — should show no active arrays.

```
tiago-paquete@Linux:~/raid-lab$ rm -rf /mnt/raid{0,1,5,10}
========================================================================
rm: cannot remove '/mnt/raid0': Permission denied
rm: cannot remove '/mnt/raid1': Permission denied
rm: cannot remove '/mnt/raid5': Permission denied
rm: cannot remove '/mnt/raid10': Permission denied
========================================================================

tiago-paquete@Linux:~/raid-lab$ sudo rm -rf /mnt/raid{0,1,5,10}
tiago-paquete@Linux:~/raid-lab$ rm -rf ~/raid-lab
tiago-paquete@Linux:~/raid-lab$ echo -e "\n>>> losetup check (should be
empty):"
========================================================================

>>> losetup check (should be empty):
========================================================================

tiago-paquete@Linux:~/raid-lab$ losetup -a | grep raid-lab || echo "  OK
– no raid-lab loops"
========================================================================
/dev/loop19: []: (/home/tiago-paquete/raid-lab/disk4.img (deleted))
/dev/loop27: []: (/home/tiago-paquete/raid-lab/disk12.img (deleted))
/dev/loop17: []: (/home/tiago-paquete/raid-lab/disk2.img (deleted))
/dev/loop25: []: (/home/tiago-paquete/raid-lab/disk10.img (deleted))
/dev/loop23: []: (/home/tiago-paquete/raid-lab/disk8.img (deleted))
/dev/loop21: []: (/home/tiago-paquete/raid-lab/disk6.img (deleted))
/dev/loop18: []: (/home/tiago-paquete/raid-lab/disk3.img (deleted))
/dev/loop26: []: (/home/tiago-paquete/raid-lab/disk11.img (deleted))
/dev/loop16: []: (/home/tiago-paquete/raid-lab/disk1.img (deleted))
/dev/loop24: []: (/home/tiago-paquete/raid-lab/disk9.img (deleted))
/dev/loop22: []: (/home/tiago-paquete/raid-lab/disk7.img (deleted))
/dev/loop20: []: (/home/tiago-paquete/raid-lab/disk5.img (deleted))
========================================================================
```

```
tiago-paquete@Linux:~/raid-lab$ echo -e "\n>>> md status (should list
none):"
==========================================================================
>>> md status (should list none):
==========================================================================

tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
==========================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md3 : active raid10 loop26[3] loop25[2] loop24[1] loop23[0]
      200704 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

md2 : active raid5 loop27[4] loop22[3] loop21[1](F) loop20[0]
      200704 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3]
[UUU]

md1 : active raid1 loop19[1] loop18[0]
      101376 blocks super 1.2 [2/2] [UU]

md0 : active raid0 loop17[1] loop16[0]
      200704 blocks super 1.2 512k chunks

unused devices: <none>
==========================================================================
```

⚠️

What's left:

- All four RAID arrays (`md0`, `md1`, `md2`, `md3`) are still running.
- All loop devices are still attached — even though the backing image files have been deleted.

Final troubleshooting steps

```
# Stop and remove all md arrays
for md in /dev/md{3,2,1,0}; do
  sudo mdadm --stop "$md" 2>/dev/null
  sudo mdadm --remove "$md" 2>/dev/null
done

# Zero superblocks from all loop devices
for dev in /dev/loop{16..27}; do
  sudo mdadm --zero-superblock "$dev" 2>/dev/null
done

# Detach remaining loop devices forcibly
sudo losetup -a | grep raid-lab | cut -d: -f1 | xargs -r -n1 sudo losetup -d

# Final checks
echo -e "\n>>> losetup check (should be empty):"
losetup -a | grep raid-lab || echo "  OK – no raid-lab loops"

echo -e "\n>>> md status (should list none):"
cat /proc/mdstat
```

```
tiago-paquete@Linux:~/raid-lab$ for md in /dev/md{3,2,1,0}; do
  sudo mdadm --stop "$md" 2>/dev/null
  sudo mdadm --remove "$md" 2>/dev/null
done

tiago-paquete@Linux:~/raid-lab$ for dev in /dev/loop{16..27}; do
  sudo mdadm --zero-superblock "$dev" 2>/dev/null
done

tiago-paquete@Linux:~/raid-lab$ sudo losetup -a | grep raid-lab | cut
-d: -f1 | xargs -r -n1 sudo losetup -d

tiago-paquete@Linux:~/raid-lab$ echo -e "\n>>> losetup check (should be
empty):"
================================================================================
>>> losetup check (should be empty):
================================================================================

tiago-paquete@Linux:~/raid-lab$ losetup -a | grep raid-lab || echo "  OK
– no raid-lab loops”
================================================================================
  OK – no raid-lab loops
================================================================================
```

```
tiago-paquete@Linux:~/raid-lab$ echo -e "\n>>> md status (should list
none):"
================================================================================
>>> md status (should list none):
================================================================================

tiago-paquete@Linux:~/raid-lab$ cat /proc/mdstat
================================================================================
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>
================================================================================
```