Title 5: RAID and Advanced Filesystem Features in Linux (mdadm, ext4, and XFS Tuning, Performance Test)

What is RAID?

RAID (Redundant Array of Independent Disks) is a technology that combines multiple physical disks into a single logical unit to improve performance, redundancy, or both.
The idea is simple:
Instead of relying on one disk (which can fail or be slow), we use multiple disks together for speed, reliability, or both.
RAID is handled either by:
Software RAID → e.g., mdadm on Linux
Hardware RAID → via dedicated RAID controller card

RAID Levels (Types)

There are many RAID levels, but these are the most common:

RAID Level	Description	Minimum Disks	Storage Efficiency	Fault Tolerance	Performance
RAID 0 (Striping)	Data split (striped) across disks for speed. No redundancy.	2	100%	None	🚀 High read/write
RAID 1 (Mirroring)	Data duplicated on each disk.	2	50%	1 disk	⚡ Good read, slower write
RAID 5 (Striping with Parity)	Data + parity distributed across disks.	3	(N-1)/N	1 disk	⚡ Good read, moderate write
RAID 6 (Double Parity)	Like RAID 5 but can survive 2 disk failures.	4	(N-2)/N	2 disks	⚡ Moderate
RAID 10 (1+0)	Combination of mirroring + striping.	4	50%	1 per mirror	🚀 Excellent overall

---

Purpose of RAID

RAID is mainly used for:
Data Redundancy → Prevent data loss if a disk fails.
Performance Improvement → Increase read/write speed.
Larger Logical Volumes → Combine disks into one large storage.
High Availability → Ensure continuous access even after disk failure.

Pros (Advantages)

Benefit	Explanation
Fault Tolerance	Some RAID levels (like RAID 1, 5, 6, 10) protect against disk failure.
Improved Speed	Striping improves read/write performance (especially RAID 0, 5, 10).
Scalability	Easy to expand storage by adding more disks.
Availability	Systems remain operational even if one disk fails (depending on RAID level).

Don’ts / Limitations

Caution	Why
Don’t use RAID as a backup	RAID protects against hardware failure, not accidental deletions or ransomware.
RAID rebuilds can be risky	Rebuilding after a disk failure can stress other disks, potentially leading to a second failure.
RAID 0 has no redundancy	One disk fails → total data loss.
Requires identical disks	Mismatched sizes reduce efficiency.
Software RAID consumes CPU	Minimal, but still a factor compared to hardware RAID.

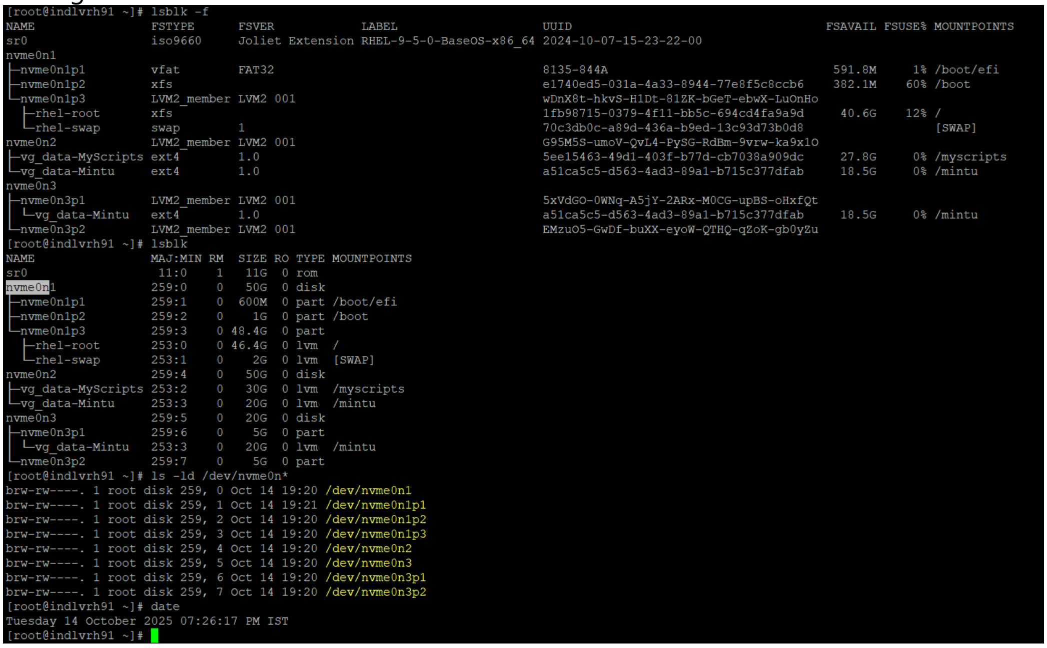
RAID 5 Example (the one you’re using)
Needs 3 or more disks.
Combines striping (speed) + parity (redundancy).
If one disk fails, you can rebuild from parity.
You get N-1 usable capacity.

Example:
If you have three 512 MB disks →
RAID 5 usable = 512 MB × (3−1) = **~1 GB usable space**

Disk Count	Disk Size (Each)	Total Virtual Storage	Notes
3 disks	500 MB each	~1 GB usable	Good for basic RAID 5 test
5 disks	1 GB each	~4 GB usable	Better for extended practice & benchmarks

Step 1: Creating 3 Disks of 500 MB Each

Before adding 3 disks, the output of ***lsblk, ls -ld /dev/nvme0n*.***it helps us to compare easily, after adding disks.

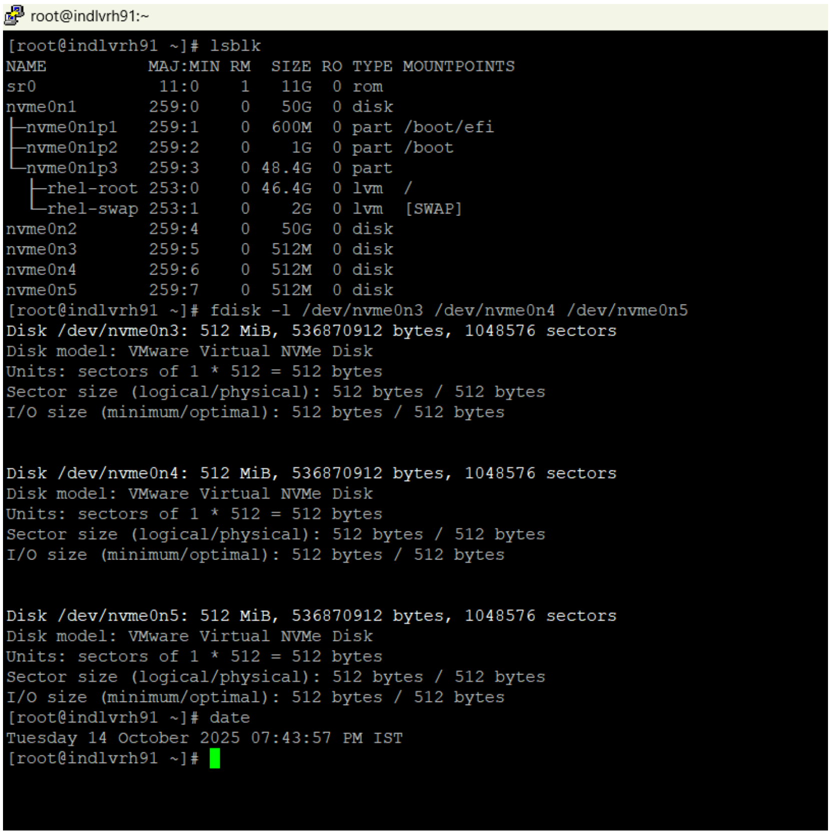


```
[root@indlvrh91 ~]# lsblk -f
NAME                FSTYPE      FSVER            LABEL                    UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
sr0                 iso9660     Joliet Extension RHEL-9-5-0-BaseOS-x86_64 2024-10-07-15-23-22-00
nvme0n1
├─nvme0n1p1         vfat        FAT32                                     8135-844A                               591.8M     1% /boot/efi
├─nvme0n1p2         xfs                                                   e1740ed5-031a-4a33-8944-77e8f5c8ccb6    382.1M    60% /boot
└─nvme0n1p3         LVM2_member LVM2 001                                  wDnX8t-hkvS-H1Dt-81ZK-bGeT-ebwX-LuOnHo
  ├─rhel-root       xfs                                                   1fb98715-0379-4f11-bb5c-694cd4fa9a9d     40.6G    12% /
  └─rhel-swap       swap        1                                         70c3db0c-a89d-436a-b9ed-13c93d73b0d8                  [SWAP]
nvme0n2             LVM2_member LVM2 001                                  G95M5S-umoV-QvL4-PySG-RdBm-9vrw-ka9x1O
├─vg_data-MyScripts ext4        1.0                                       5ee15463-49d1-403f-b77d-cb7038a909dc     27.8G     0% /myscripts
└─vg_data-Mintu     ext4        1.0                                       a51ca5c5-d563-4ad3-89a1-b715c377dfab     18.5G     0% /mintu
nvme0n3
├─nvme0n3p1         LVM2_member LVM2 001                                  5xVdGO-0WNq-A5jY-2ARx-M0CG-upBS-oHxfQt
│ └─vg_data-Mintu   ext4        1.0                                       a51ca5c5-d563-4ad3-89a1-b715c377dfab     18.5G     0% /mintu
└─nvme0n3p2         LVM2_member LVM2 001                                  EMzuO5-GwDf-buXX-eyoW-QTHQ-qZoK-gb0yZu
[root@indlvrh91 ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sr0                  11:0    1   11G  0 rom
nvme0n1             259:0    0   50G  0 disk
├─nvme0n1p1         259:1    0  600M  0 part /boot/efi
├─nvme0n1p2         259:2    0    1G  0 part /boot
└─nvme0n1p3         259:3    0 48.4G  0 part
  ├─rhel-root       253:0    0 46.4G  0 lvm  /
  └─rhel-swap       253:1    0    2G  0 lvm  [SWAP]
nvme0n2             259:4    0   50G  0 disk
├─vg_data-MyScripts 253:2    0   30G  0 lvm  /myscripts
└─vg_data-Mintu     253:3    0   20G  0 lvm  /mintu
nvme0n3             259:5    0   20G  0 disk
├─nvme0n3p1         259:6    0    5G  0 part
│ └─vg_data-Mintu   253:3    0   20G  0 lvm  /mintu
└─nvme0n3p2         259:7    0    5G  0 part
[root@indlvrh91 ~]# ls -ld /dev/nvme0n*
brw-rw----. 1 root disk 259, 0 Oct 14 19:20 /dev/nvme0n1
brw-rw----. 1 root disk 259, 1 Oct 14 19:21 /dev/nvme0n1p1
brw-rw----. 1 root disk 259, 2 Oct 14 19:20 /dev/nvme0n1p2
brw-rw----. 1 root disk 259, 3 Oct 14 19:20 /dev/nvme0n1p3
brw-rw----. 1 root disk 259, 4 Oct 14 19:20 /dev/nvme0n2
brw-rw----. 1 root disk 259, 5 Oct 14 19:20 /dev/nvme0n3
brw-rw----. 1 root disk 259, 6 Oct 14 19:20 /dev/nvme0n3p1
brw-rw----. 1 root disk 259, 7 Oct 14 19:20 /dev/nvme0n3p2
[root@indlvrh91 ~]# date
Tuesday 14 October 2025 07:26:17 PM IST
[root@indlvrh91 ~]#
```

Verify Disks

First, make sure these disks are **not in use or mounted**.

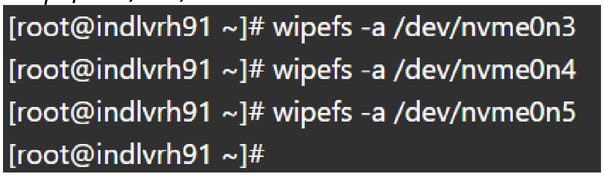
#lsblk; #blkid



```
[root@indlvrh91 ~]# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sr0            11:0    1   11G  0 rom
nvme0n1       259:0    0   50G  0 disk
├─nvme0n1p1   259:1    0  600M  0 part /boot/efi
├─nvme0n1p2   259:2    0    1G  0 part /boot
└─nvme0n1p3   259:3    0 48.4G  0 part
  ├─rhel-root 253:0    0 46.4G  0 lvm  /
  └─rhel-swap 253:1    0    2G  0 lvm  [SWAP]
nvme0n2       259:4    0   50G  0 disk
nvme0n3       259:5    0  512M  0 disk
nvme0n4       259:6    0  512M  0 disk
nvme0n5       259:7    0  512M  0 disk
[root@indlvrh91 ~]# fdisk -l /dev/nvme0n3 /dev/nvme0n4 /dev/nvme0n5
Disk /dev/nvme0n3: 512 MiB, 536870912 bytes, 1048576 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/nvme0n4: 512 MiB, 536870912 bytes, 1048576 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/nvme0n5: 512 MiB, 536870912 bytes, 1048576 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
[root@indlvrh91 ~]# date
Tuesday 14 October 2025 07:43:57 PM IST
[root@indlvrh91 ~]#
```

If any of those devices (nvme0n3, nvme0n4, nvme0n5) show partitions or filesystems, **wipe them sssclean**:

#wipefs -a /dev/nvme0n3
#wipefs -a /dev/nvme0n4
#wipefs -a /dev/nvme0n5



```
[root@indlvrh91 ~]# wipefs -a /dev/nvme0n3
[root@indlvrh91 ~]# wipefs -a /dev/nvme0n4
[root@indlvrh91 ~]# wipefs -a /dev/nvme0n5
[root@indlvrh91 ~]#
```

Now, three disks /dev/nvme0n3, /dev/nvme0n4, and /dev/nvme0n5 are **completely clean** and ready to be used for your RAID 5 array.

Step 2: Install Required Packages

Make sure mdadm (RAID tool) and xfsprogs / e2fsprogs are installed:
#dnf install -y mdadm xfsprogs e2fsprogs
Here, I have already installed all the supported packages at the time of patching.

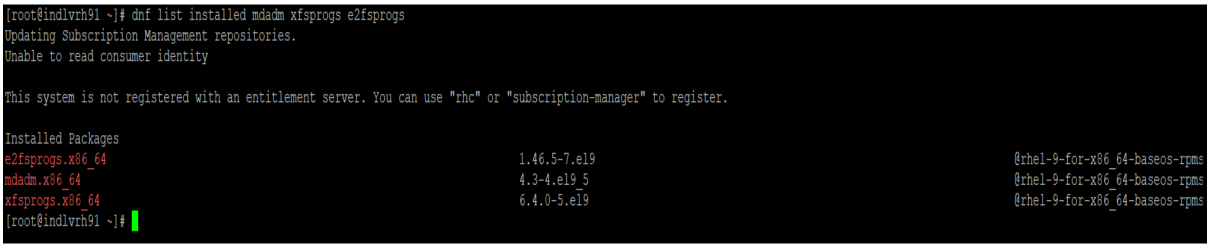
Using Yum/dnf list installed
# dnf list installed mdadm xfsprogs e2fsprogs

Expected output:
You should see something like this if installed:
Installed Packages
mdadm.x86_64   4.3-4.el9_5   @rhel-9-baseos
xfsprogs.x86_64   6.4.0-5.el9   @rhel-9-baseos
e2fsprogs.x86_64   1.46.5-7.el9 @rhel-9-baseos

Using rpm -q (checks individually)
#rpm -q mdadm
#rpm -q xfsprogs
#rpm -q e2fsprogs

Expected output if installed:

- e2fsprogs → version 1.46.5-7.el9
- mdadm → version 4.3-4.el9_5
- xfsprogs → version 6.4.0-5.el9

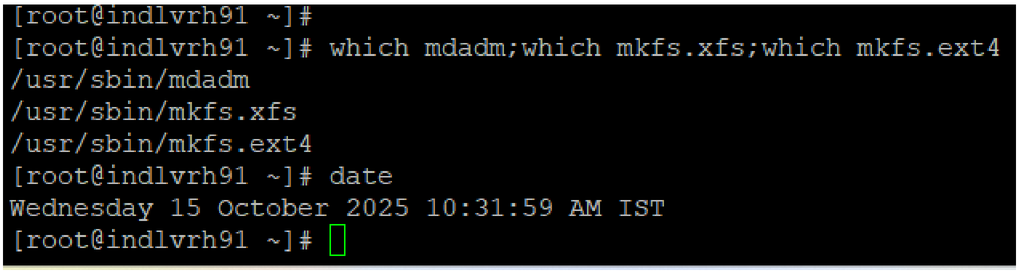


```
[root@indlvrh91 ~]# dnf list installed mdadm xfsprogs e2fsprogs
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Installed Packages
e2fsprogs.x86_64                                    1.46.5-7.el9                                    @rhel-9-for-x86_64-baseos-rpms
mdadm.x86_64                                        4.3-4.el9_5                                     @rhel-9-for-x86_64-baseos-rpms
xfsprogs.x86_64                                     6.4.0-5.el9                                     @rhel-9-for-x86_64-baseos-rpms
[root@indlvrh91 ~]#
```

Check if the binaries are available
#which mdadm; #which mkfs.xfs; #which mkfs.ext4

Expected output:

- /usr/sbin/mdadm
- /usr/sbin/mkfs.xfs
- /usr/sbin/mkfs.ext4



```
[root@indlvrh91 ~]#
[root@indlvrh91 ~]# which mdadm;which mkfs.xfs;which mkfs.ext4
/usr/sbin/mdadm
/usr/sbin/mkfs.xfs
/usr/sbin/mkfs.ext4
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 10:31:59 AM IST
[root@indlvrh91 ~]#
```

Step 3: Create RAID 5 Array

We'll use /dev/nvme0n3, /dev/nvme0n4, /dev/nvme0n5.

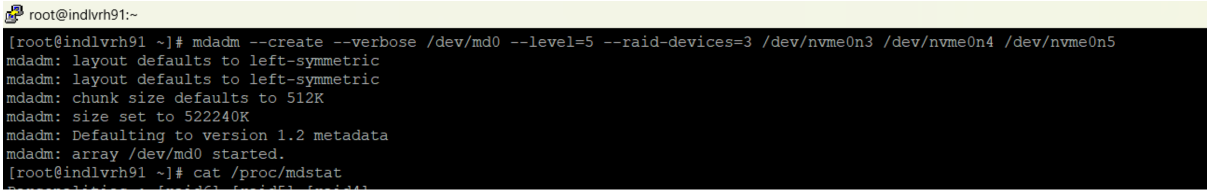
#mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/nvme0n3 /dev/nvme0n4 /dev/nvme0n5

Detail* explanation *of command

Part	Meaning / Function	Explanation
mdadm	RAID management tool	The Linux utility used to create, assemble, monitor, and manage software RAID arrays.
--create	Create new array	Tells mdadm to create a new RAID device. This erases existing data on listed disks.
--verbose	Detailed output	Enables verbose mode, showing what mdadm is doing — useful for learning or troubleshooting.
/dev/md0	Target RAID device	The logical RAID device that will be created. You can name it /dev/md0, /dev/md1, etc.
--level=5	RAID level	Specifies RAID 5 (striped with parity) — combines speed + redundancy, requires ≥3 disks.
--raid-devices=3	Number of disks	Defines how many physical drives participate in this RAID array. In this case: 3 disks.
/dev/nvme0n3 /dev/nvme0n4 /dev/nvme0n5	Physical disks	Lists the actual disk devices used to build the RAID. These must be wiped clean before creation.

Quick Calculation

Total Disks	RAID Level	Formula	Usable Capacity	Fault Tolerance
3 × 512 MB	RAID 5	(N - 1) × Disk Size	(3 - 1) × 512 MB = **~1 GB**	Can survive **1 disk failure**



```
[root@indlvrh91 ~]# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/nvme0n3 /dev/nvme0n4 /dev/nvme0n5
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 522240K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@indlvrh91 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
```

Monitor build progress:

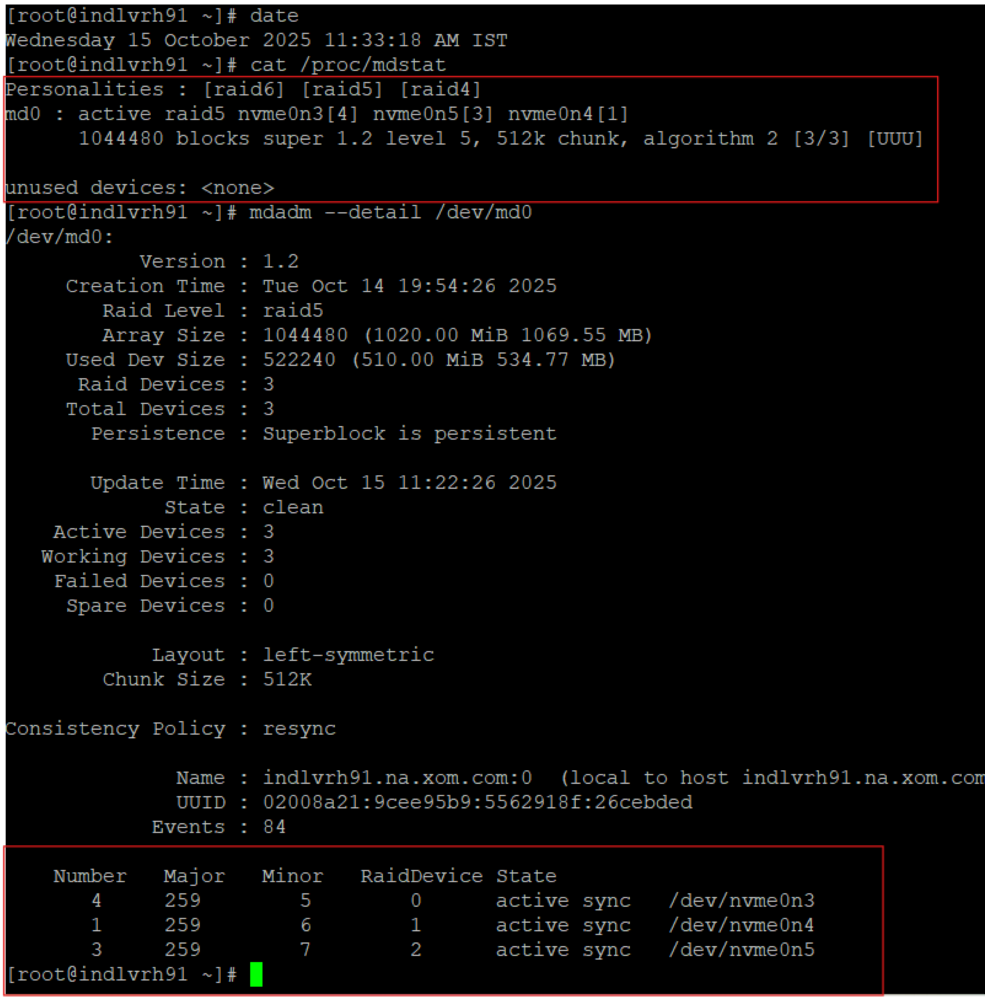
#watch cat /proc/mdstat

Optional Verification After Creation

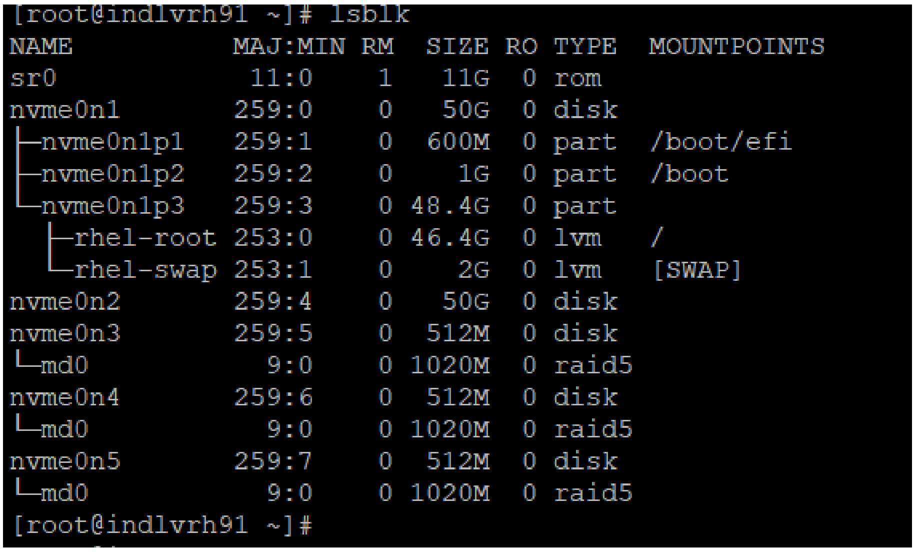
#cat /proc/mdstat   # Shows the RAID progress and sync status

#mdadm --detail /dev/md0   # Shows RAID level, devices, state, UUID

#lsblk   # Shows new md device with size



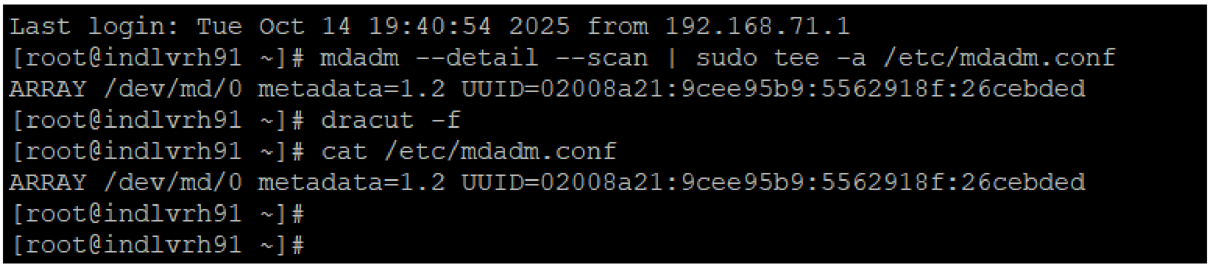
```
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 11:33:18 AM IST
[root@indlvrh91 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 nvme0n3[4] nvme0n5[3] nvme0n4[1]
      1044480 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
[root@indlvrh91 ~]# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Oct 14 19:54:26 2025
        Raid Level : raid5
        Array Size : 1044480 (1020.00 MiB 1069.55 MB)
     Used Dev Size : 522240 (510.00 MiB 534.77 MB)
      Raid Devices : 3
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Wed Oct 15 11:22:26 2025
             State : clean
    Active Devices : 3
   Working Devices : 3
    Failed Devices : 0
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : indlvrh91.na.xom.com:0  (local to host indlvrh91.na.xom.com
              UUID : 02008a21:9cee95b9:5562918f:26cebded
            Events : 84

    Number   Major   Minor   RaidDevice State
       4     259        5        0      active sync   /dev/nvme0n3
       1     259        6        1      active sync   /dev/nvme0n4
       3     259        7        2      active sync   /dev/nvme0n5
[root@indlvrh91 ~]# 
```



```
[root@indlvrh91 ~]# lsblk
NAME            MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sr0              11:0    1   11G  0 rom
nvme0n1         259:0    0   50G  0 disk
├─nvme0n1p1     259:1    0  600M  0 part  /boot/efi
├─nvme0n1p2     259:2    0    1G  0 part  /boot
└─nvme0n1p3     259:3    0 48.4G  0 part
  ├─rhel-root 253:0    0 46.4G  0 lvm   /
  └─rhel-swap 253:1    0    2G  0 lvm   [SWAP]
nvme0n2         259:4    0   50G  0 disk
nvme0n3         259:5    0  512M  0 disk
└─md0             9:0    0 1020M  0 raid5
nvme0n4         259:6    0  512M  0 disk
└─md0             9:0    0 1020M  0 raid5
nvme0n5         259:7    0  512M  0 disk
└─md0             9:0    0 1020M  0 raid5
[root@indlvrh91 ~]#
```

Step 4: Save RAID Configuration (Persistence)

Make sure RAID auto-assembles after reboot:
#mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
#dracut -f
#cat /etc/mdadm.conf



```
Last login: Tue Oct 14 19:40:54 2025 from 192.168.71.1
[root@indlvrh91 ~]# mdadm --detail --scan | sudo tee -a /etc/mdadm.conf
ARRAY /dev/md/0 metadata=1.2 UUID=02008a21:9cee95b9:5562918f:26cebded
[root@indlvrh91 ~]# dracut -f
[root@indlvrh91 ~]# cat /etc/mdadm.conf
ARRAY /dev/md/0 metadata=1.2 UUID=02008a21:9cee95b9:5562918f:26cebded
[root@indlvrh91 ~]#
[root@indlvrh91 ~]#
```

This saves the array info and rebuilds the initramfs image with RAID support.

Step 5: Create a Filesystem

Create a filesystem (ext4 or XFS) on /dev/md0:
Ext4:
#mkfs.ext4 /dev/md0
#mkdir -p /mnt/raid-ext4
#mount /dev/md0 /mnt/raid-ext4
#df -hT /mnt/raid-ext4

XFS (RHEL 9 default), I have chosen.
Format /dev/md0 with XFS
#mkfs.xfs /dev/md0
#mkfs.xfs → creates an XFS filesystem on the RAID device.
By default, it uses the full array size and the default allocation settings.

Create a mount point

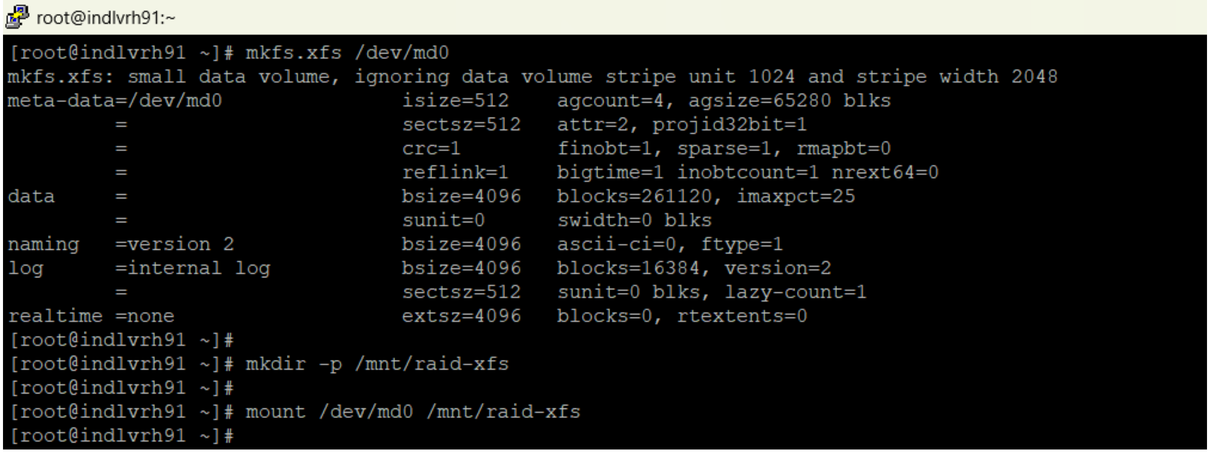
#mkdir -p /mnt/raid-xfs

This is where your RAID filesystem will be accessible.

Mount the XFS filesystem

#mount /dev/md0 /mnt/raid-xfs

The RAID array is now ready for use under /mnt/raid-xfs.



root@indlvrh91:~

```
[root@indlvrh91 ~]# mkfs.xfs /dev/md0
mkfs.xfs: small data volume, ignoring data volume stripe unit 1024 and stripe width 2048
meta-data=/dev/md0               isize=512    agcount=4, agsize=65280 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=261120, 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@indlvrh91 ~]#
[root@indlvrh91 ~]# mkdir -p /mnt/raid-xfs
[root@indlvrh91 ~]#
[root@indlvrh91 ~]# mount /dev/md0 /mnt/raid-xfs
[root@indlvrh91 ~]#
```

Verify the mount and filesystem

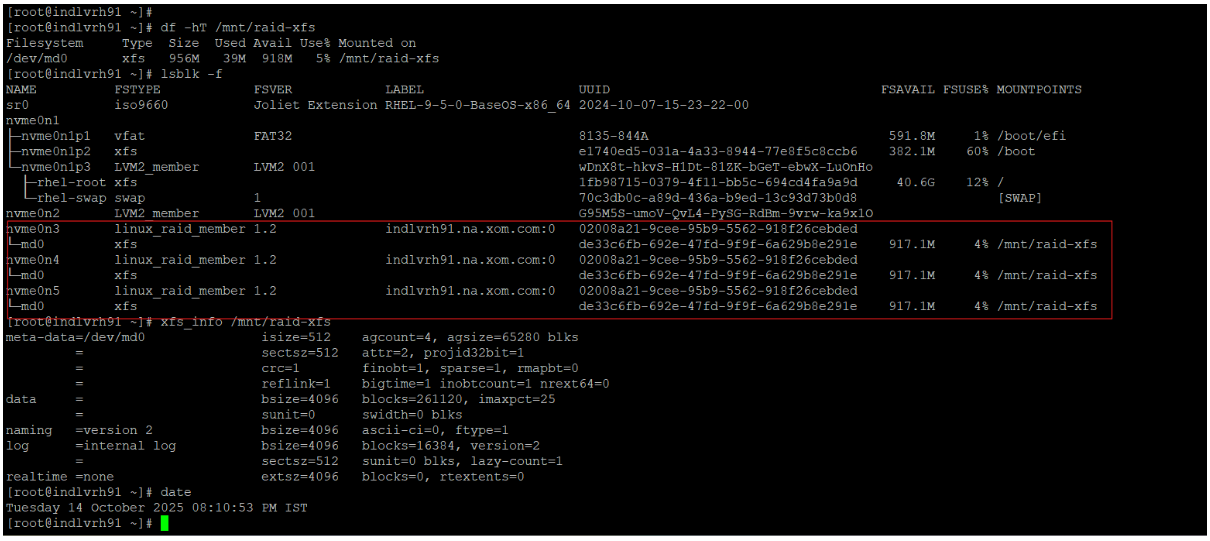
#df -hT /mnt/raid-xfs

#xfs_info /mnt/raid-xfs

#df -hT → shows size, used, and filesystem type.

#lsblk -f → shows RAID device and filesystem info.

#xfs_info → gives detailed info about XFS layout, block size, and metadata.



```
[root@indlvrh91 ~]#
[root@indlvrh91 ~]# df -hT /mnt/raid-xfs
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/md0       xfs   956M   39M  918M   5% /mnt/raid-xfs
[root@indlvrh91 ~]# lsblk -f
NAME          FSTYPE            FSVER            LABEL                    UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
sr0           iso9660           Joliet Extension RHEL-9-5-0-BaseOS-x86_64 2024-10-07-15-23-22-00
nvme0n1
├─nvme0n1p1   vfat              FAT32                                     8135-844A                               591.8M     1% /boot/efi
├─nvme0n1p2   xfs                                                         e1740ed5-031a-4a33-8944-77e8f5c8ccb6    382.1M    60% /boot
└─nvme0n1p3   LVM2_member       LVM2 001                                  wDnX8t-hkvS-H1Dt-81ZK-bGeT-ebwX-LuOnHo
  ├─rhel-root xfs                                                         1fb98715-0379-4f11-bb5c-694cd4fa9a9d     40.6G    12% /
  └─rhel-swap swap              1                                         70c3db0c-a89d-436a-b9ed-13c93d73b0d8                  [SWAP]
nvme0n2       LVM2_member       LVM2 001                                  G95M5S-umoV-QvL4-PySG-RdBm-9vrw-ka9x1O
nvme0n3       linux_raid_member 1.2              indlvrh91.na.xom.com:0   02008a21-9cee-95b9-5562-918f26cebded
└─md0         xfs                                                         de33c6fb-692e-47fd-9f9f-6a629b8e291e    917.1M     4% /mnt/raid-xfs
nvme0n4       linux_raid_member 1.2              indlvrh91.na.xom.com:0   02008a21-9cee-95b9-5562-918f26cebded
└─md0         xfs                                                         de33c6fb-692e-47fd-9f9f-6a629b8e291e    917.1M     4% /mnt/raid-xfs
nvme0n5       linux_raid_member 1.2              indlvrh91.na.xom.com:0   02008a21-9cee-95b9-5562-918f26cebded
└─md0         xfs                                                         de33c6fb-692e-47fd-9f9f-6a629b8e291e    917.1M     4% /mnt/raid-xfs
[root@indlvrh91 ~]# xfs_info /mnt/raid-xfs
meta-data=/dev/md0               isize=512    agcount=4, agsize=65280 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=261120, 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@indlvrh91 ~]# date
Tuesday 14 October 2025 08:10:53 PM IST
[root@indlvrh91 ~]#
```

Auto-mount on Boot

Add an entry to /etc/fstab:

#blkid /dev/md0

Copy the UUID of /dev/md0.

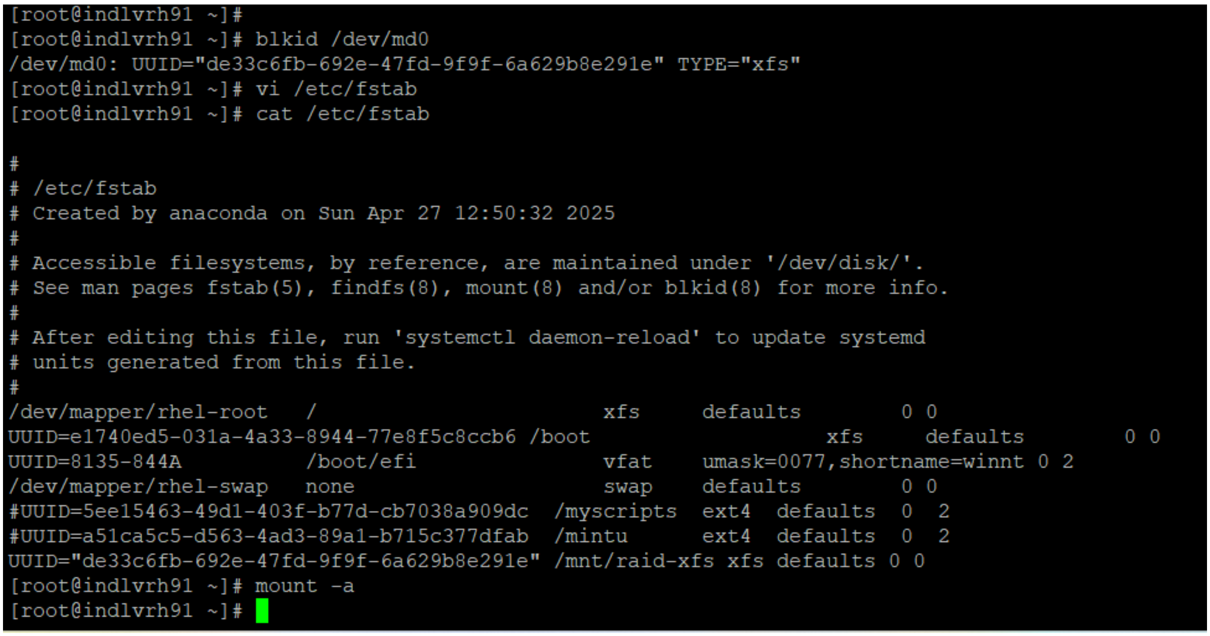
Then edit /etc/fstab:

#vi /etc/fstab

Add:

UUID="de33c6fb-692e-47fd-9f9f-6a629b8e291e" /mnt/raid-xfs xfs defaults 0 0

This ensures your RAID XFS mounts automatically after a reboot.



```
[root@indlvrh91 ~]#
[root@indlvrh91 ~]# blkid /dev/md0
/dev/md0: UUID="de33c6fb-692e-47fd-9f9f-6a629b8e291e" TYPE="xfs"
[root@indlvrh91 ~]# vi /etc/fstab
[root@indlvrh91 ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Sun Apr 27 12:50:32 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root   /                       xfs     defaults        0 0
UUID=e1740ed5-031a-4a33-8944-77e8f5c8ccb6 /boot                   xfs     defaults        0 0
UUID=8135-844A          /boot/efi               vfat    umask=0077,shortname=winnt 0 2
/dev/mapper/rhel-swap   none                    swap    defaults        0 0
#UUID=5ee15463-49d1-403f-b77d-cb7038a909dc  /myscripts  ext4  defaults  0  2
#UUID=a51ca5c5-d563-4ad3-89a1-b715c377dfab  /mintu      ext4  defaults  0  2
UUID="de33c6fb-692e-47fd-9f9f-6a629b8e291e" /mnt/raid-xfs xfs defaults 0 0
[root@indlvrh91 ~]# mount -a
[root@indlvrh91 ~]# 
```

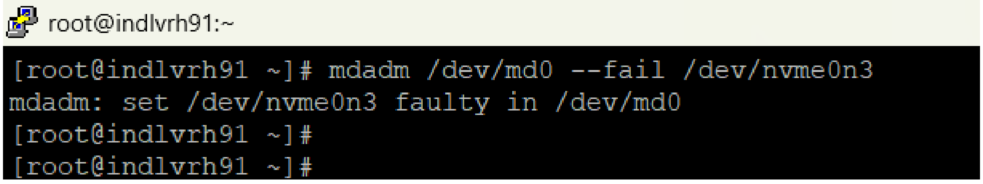
Step 6: Simulate Disk Failure and Rebuild (RAID 5)

Mark one disk as failed:

#mdadm /dev/md0 --fail /dev/nvme0n3

This tells RAID:

“Pretend /dev/nvme0n3 has failed.”



```
[root@indlvrh91 ~]# mdadm /dev/md0 --fail /dev/nvme0n3
mdadm: set /dev/nvme0n3 faulty in /dev/md0
[root@indlvrh91 ~]#
[root@indlvrh91 ~]#
```

Verify the failure status:

#mdadm --detail /dev/md0

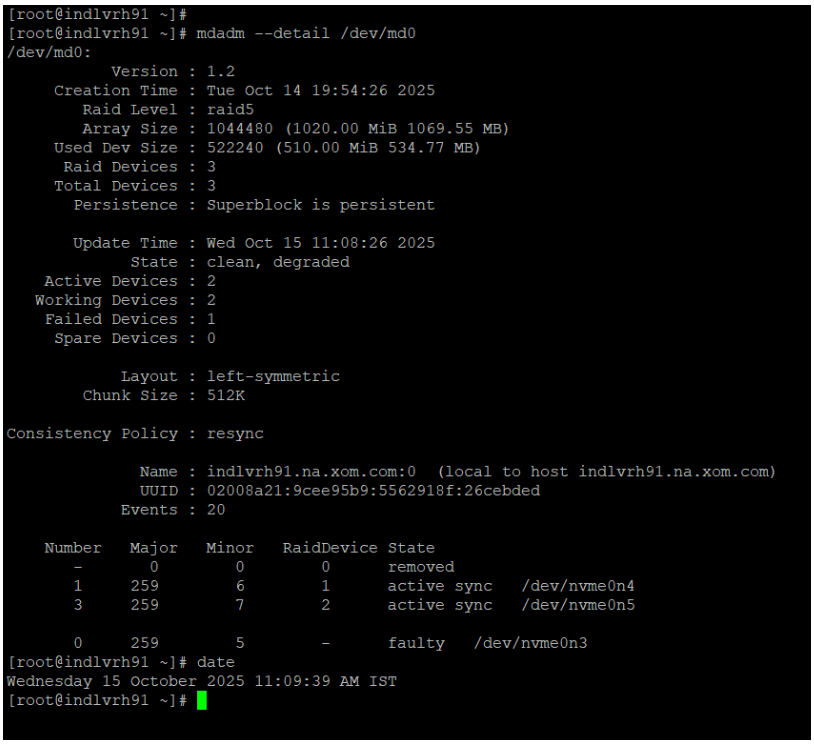
You’ll see something like:

/dev/nvme0n3 -> faulty

/dev/nvme0n4 -> active

/dev/nvme0n5 -> active

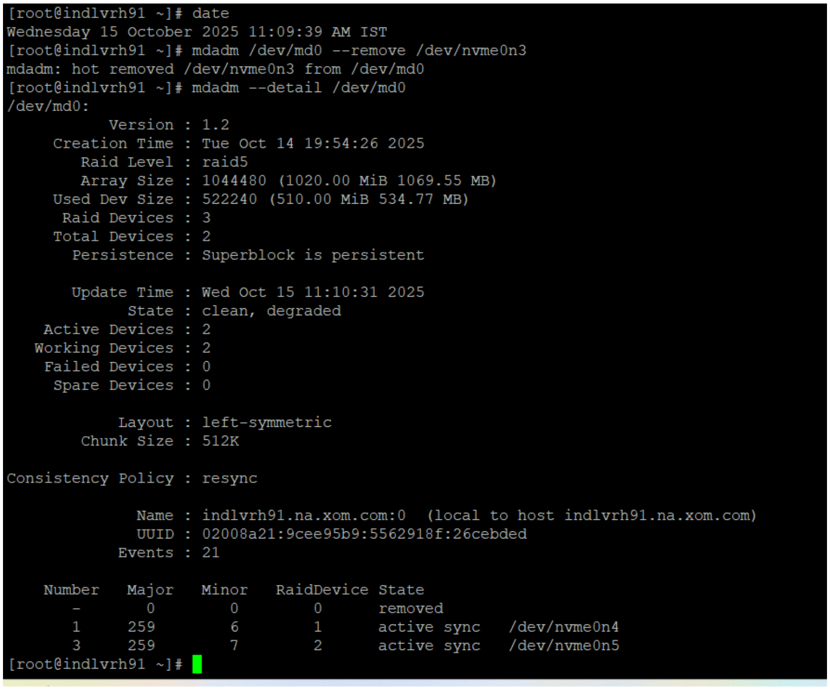
and RAID will be in **degraded mode**.



```
[root@indlvrh91 ~]#
[root@indlvrh91 ~]# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Oct 14 19:54:26 2025
        Raid Level : raid5
        Array Size : 1044480 (1020.00 MiB 1069.55 MB)
     Used Dev Size : 522240 (510.00 MiB 534.77 MB)
      Raid Devices : 3
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Wed Oct 15 11:08:26 2025
             State : clean, degraded
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 1
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : indlvrh91.na.xom.com:0  (local to host indlvrh91.na.xom.com)
              UUID : 02008a21:9cee95b9:5562918f:26cebded
            Events : 20

    Number   Major   Minor   RaidDevice State
       -       0        0        0      removed
       1     259        6        1      active sync   /dev/nvme0n4
       3     259        7        2      active sync   /dev/nvme0n5

       0     259        5        -      faulty   /dev/nvme0n3
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 11:09:39 AM IST
[root@indlvrh91 ~]#
```

Remove the failed device:

#mdadm /dev/md0 --remove /dev/nvme0n3

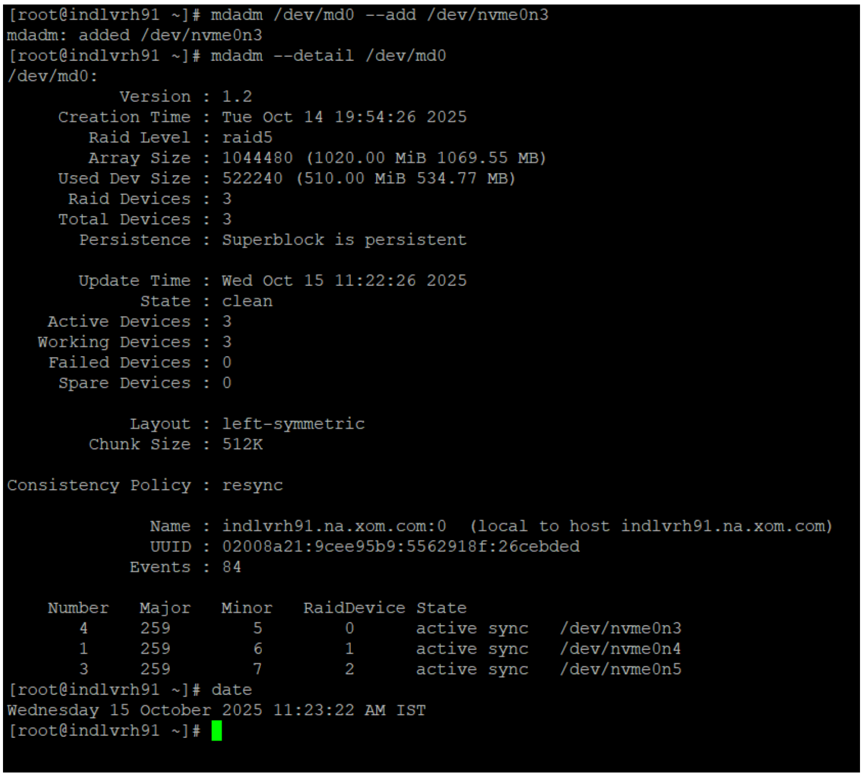


```
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 11:09:39 AM IST
[root@indlvrh91 ~]# mdadm /dev/md0 --remove /dev/nvme0n3
mdadm: hot removed /dev/nvme0n3 from /dev/md0
[root@indlvrh91 ~]# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Oct 14 19:54:26 2025
        Raid Level : raid5
        Array Size : 1044480 (1020.00 MiB 1069.55 MB)
     Used Dev Size : 522240 (510.00 MiB 534.77 MB)
      Raid Devices : 3
     Total Devices : 2
       Persistence : Superblock is persistent

       Update Time : Wed Oct 15 11:10:31 2025
             State : clean, degraded
    Active Devices : 2
   Working Devices : 2
    Failed Devices : 0
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : indlvrh91.na.xom.com:0  (local to host indlvrh91.na.xom.com)
              UUID : 02008a21:9cee95b9:5562918f:26cebded
            Events : 21

    Number   Major   Minor   RaidDevice State
       -       0        0        0      removed
       1     259        6        1      active sync   /dev/nvme0n4
       3     259        7        2      active sync   /dev/nvme0n5
[root@indlvrh91 ~]#
```

(Optional) Re-add the same device to simulate disk replacement:

#mdadm /dev/md0 --add /dev/nvme0n3

This triggers a **rebuild** process.

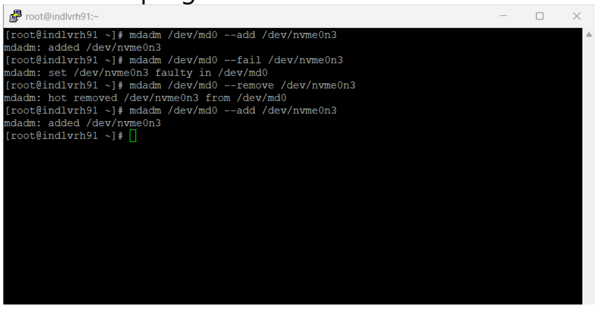


```
[root@indlvrh91 ~]# mdadm /dev/md0 --add /dev/nvme0n3
mdadm: added /dev/nvme0n3
[root@indlvrh91 ~]# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Oct 14 19:54:26 2025
        Raid Level : raid5
        Array Size : 1044480 (1020.00 MiB 1069.55 MB)
     Used Dev Size : 522240 (510.00 MiB 534.77 MB)
      Raid Devices : 3
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Wed Oct 15 11:22:26 2025
             State : clean
    Active Devices : 3
   Working Devices : 3
    Failed Devices : 0
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : indlvrh91.na.xom.com:0  (local to host indlvrh91.na.xom.com)
              UUID : 02008a21:9cee95b9:5562918f:26cebded
            Events : 84

    Number   Major   Minor   RaidDevice State
       4     259        5        0      active sync   /dev/nvme0n3
       1     259        6        1      active sync   /dev/nvme0n4
       3     259        7        2      active sync   /dev/nvme0n5
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 11:23:22 AM IST
[root@indlvrh91 ~]#
```

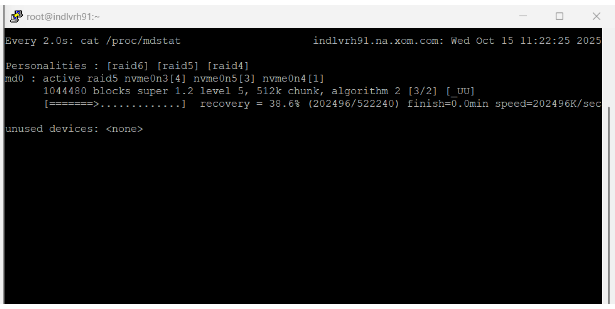
Watch rebuild progress:

#watch cat /proc/mdstat

You’ll see a progress bar as data is rebuilt.



```
[root@indlvrh91 ~]# mdadm /dev/md0 --add /dev/nvme0n3
mdadm: added /dev/nvme0n3
[root@indlvrh91 ~]# mdadm /dev/md0 --fail /dev/nvme0n3
mdadm: set /dev/nvme0n3 faulty in /dev/md0
[root@indlvrh91 ~]# mdadm /dev/md0 --remove /dev/nvme0n3
mdadm: hot removed /dev/nvme0n3 from /dev/md0
[root@indlvrh91 ~]# mdadm /dev/md0 --add /dev/nvme0n3
mdadm: added /dev/nvme0n3
[root@indlvrh91 ~]#
```



```
Every 2.0s: cat /proc/mdstat                    indlvrh91.na.xom.com: Wed Oct 15 11:22:25 2025

Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 nvme0n3[4] nvme0n5[3] nvme0n4[1]
      1044480 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
      [=======>.............]  recovery = 38.6% (202496/522240) finish=0.0min speed=202496K/sec

unused devices: <none>
```

Verify everything is healthy again:

#mdadm --detail /dev/md0

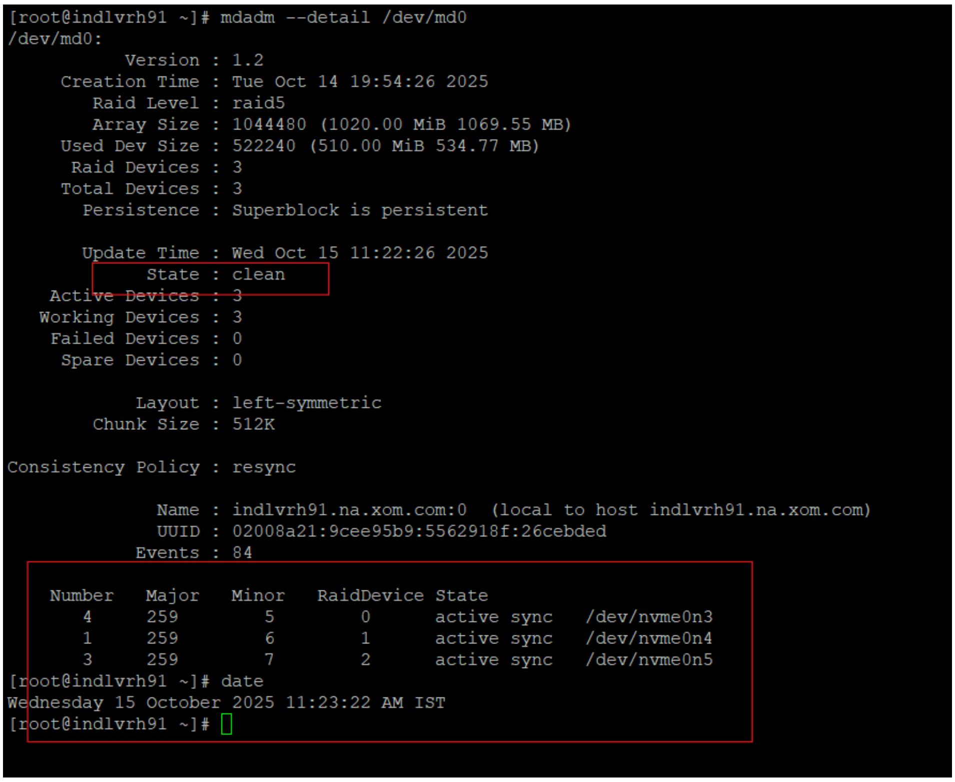
Expected:

State : clean

Active Devices : 3

Working Devices : 3

Failed Devices : 0



```
[root@indlvrh91 ~]# mdadm --detail /dev/md0
/dev/md0:
           Version : 1.2
     Creation Time : Tue Oct 14 19:54:26 2025
        Raid Level : raid5
        Array Size : 1044480 (1020.00 MiB 1069.55 MB)
     Used Dev Size : 522240 (510.00 MiB 534.77 MB)
      Raid Devices : 3
     Total Devices : 3
       Persistence : Superblock is persistent

       Update Time : Wed Oct 15 11:22:26 2025
             State : clean
    Active Devices : 3
   Working Devices : 3
    Failed Devices : 0
     Spare Devices : 0

            Layout : left-symmetric
        Chunk Size : 512K

Consistency Policy : resync

              Name : indlvrh91.na.xom.com:0  (local to host indlvrh91.na.xom.com)
              UUID : 02008a21:9cee95b9:5562918f:26cebded
            Events : 84

    Number   Major   Minor   RaidDevice State
       4     259        5        0      active sync   /dev/nvme0n3
       1     259        6        1      active sync   /dev/nvme0n4
       3     259        7        2      active sync   /dev/nvme0n5
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 11:23:22 AM IST
[root@indlvrh91 ~]#
```

Important Note

Rule	Why
Always fail/remove only **one disk** in RAID 5	RAID 5 can survive **only one disk failure**
Don't reboot during rebuild	Rebuilding is intensive and must complete
Use cat /proc/mdstat frequently	To monitor rebuild and ensure success

Step 7: Performance Testing

1. Write Test (Sequential Write Speed)

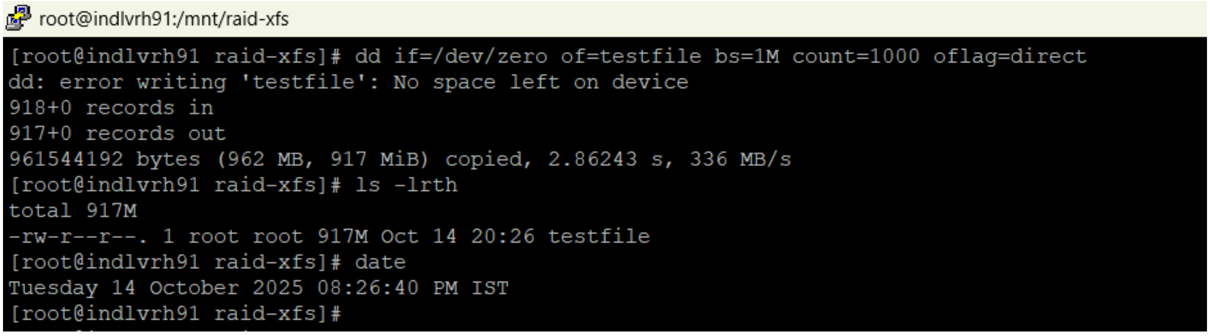
#cd /raid-xfs

#dd if=/dev/zero of=testfile bs=1M count=1000 oflag=direct

if=/dev/zero → reads zeroed bytes (simulates data write).

- of=testfile → file to write on RAID.
- bs=1M → block size = 1 MB.
- count=1000 → writes 1000 MB (1 GB).
- oflag=direct → bypass cache to measure actual disk speed.

After completion, it will show **MB/s write speed**.



root@indlvrh91:/mnt/raid-xfs

```
[root@indlvrh91 raid-xfs]# dd if=/dev/zero of=testfile bs=1M count=1000 oflag=direct
dd: error writing 'testfile': No space left on device
918+0 records in
917+0 records out
961544192 bytes (962 MB, 917 MiB) copied, 2.86243 s, 336 MB/s
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# date
Tuesday 14 October 2025 08:26:40 PM IST
[root@indlvrh91 raid-xfs]#
```

Analysis of Output

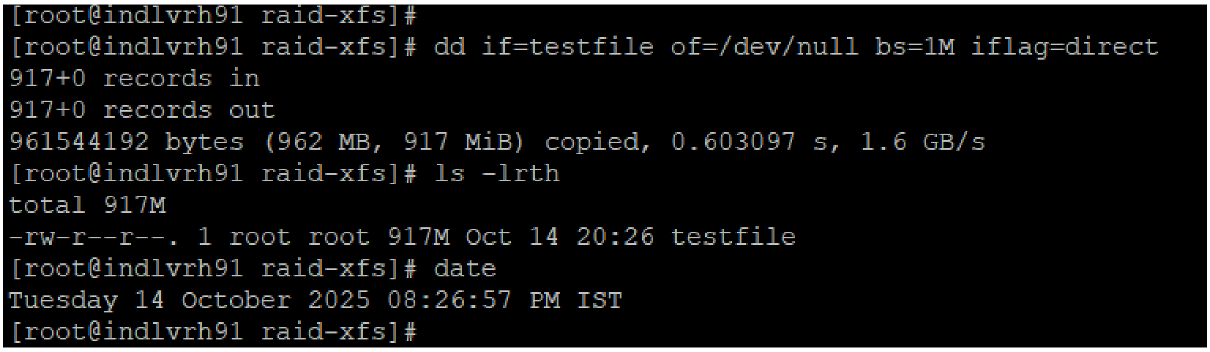
```
dd: error writing 'testfile': No space left on device
918+0 records in
917+0 records out
961544192 bytes (962 MB, 917 MiB) copied, 2.93044 s, 328 MB/s
```

No space left on device → Expected, because your RAID 5 array is only ~1020 MB in total and you tried to write 1000 MB.
Records in/out → 917 full 1 MB blocks written
Bytes copied → 962 MB actually written.
Speed → ~328 MB/s sequential write speed.
This shows your RAID is functioning and the XFS filesystem is working properly.

2. Read Test (Sequential Read Speed)

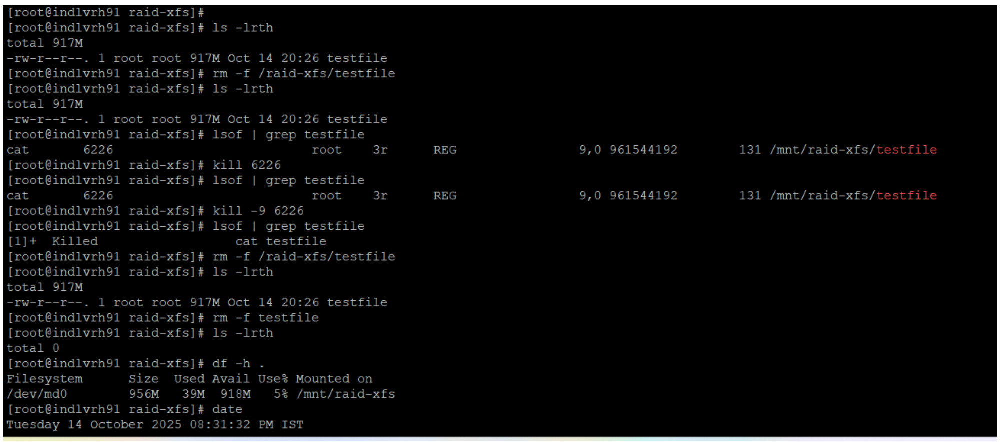
#dd if=testfile of=/dev/null bs=1M iflag=direct
Reads the file from disk and discards it.

- iflag=direct → bypass cache to measure actual read speed.



```
[root@indlvrh91 raid-xfs]#
[root@indlvrh91 raid-xfs]# dd if=testfile of=/dev/null bs=1M iflag=direct
917+0 records in
917+0 records out
961544192 bytes (962 MB, 917 MiB) copied, 0.603097 s, 1.6 GB/s
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# date
Tuesday 14 October 2025 08:26:57 PM IST
[root@indlvrh91 raid-xfs]#
```

3. Clean up the test file (Clearly explained in step 8)
#rm -f /raid-xfs/testfile
Frees space after testing.

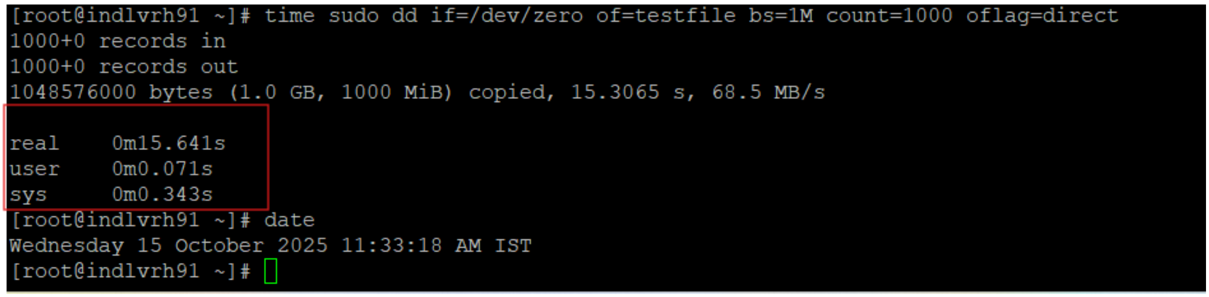


```
[root@indlvrh91 raid-xfs]#
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# rm -f /raid-xfs/testfile
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# lsof | grep testfile
cat       6226                    root    3r      REG                9,0 961544192        131 /mnt/raid-xfs/testfile
[root@indlvrh91 raid-xfs]# kill 6226
[root@indlvrh91 raid-xfs]# lsof | grep testfile
cat       6226                    root    3r      REG                9,0 961544192        131 /mnt/raid-xfs/testfile
[root@indlvrh91 raid-xfs]# kill -9 6226
[root@indlvrh91 raid-xfs]# lsof | grep testfile
[1]+  Killed                  cat testfile
[root@indlvrh91 raid-xfs]# rm -f /raid-xfs/testfile
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# rm -f testfile
[root@indlvrh91 raid-xfs]# ls -lrth
total 0
[root@indlvrh91 raid-xfs]# df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        956M   39M  918M   5% /mnt/raid-xfs
[root@indlvrh91 raid-xfs]# date
Tuesday 14 October 2025 08:31:32 PM IST
```

4. Use time for benchmarking

time sudo dd if=/dev/zero of=testfile bs=1M count=1000 oflag=direct

Shows the total time it took to write, giving a clear throughput number.



```
[root@indlvrh91 ~]# time sudo dd if=/dev/zero of=testfile bs=1M count=1000 oflag=direct
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 15.3065 s, 68.5 MB/s

real    0m15.641s
user    0m0.071s
sys     0m0.343s
[root@indlvrh91 ~]# date
Wednesday 15 October 2025 11:33:18 AM IST
[root@indlvrh91 ~]# 
```

Step 9: File Deletion & Process Lock Behavior

Scenario

Testing what happens when trying to delete a file that is still open by another process on an XFS filesystem mounted on RAID 5 (/mnt/raid-xfs).

Step-by-Step Analysis

1. List directory contents

ls -lrth

Shows one file named testfile, size **917 MB**.

Owner: root, Permissions: rw-r--r--

Confirms that the file exists in /mnt/raid-xfs/.

2. Try to remove the file

#rm -f /raid-xfs/testfile

Note: you used /raid-xfs (missing /mnt/).

Since the mount point is /mnt/raid-xfs, this path may not actually exist, so the file remains untouched.

3. Check open file handles

#lsof | grep testfile

Output:

cat 6226 root 3r REG 9,0 961544192 131 /mnt/raid-xfs/testfile

Meaning:

- **Process ID (PID):** 6226
- **Command:** cat
- **User:** root
- **File descriptor:** 3r (read-only)
- **Size:** ~917 MB

File is **currently open** by the cat process — this is why deletion doesn't work properly yet.

4. Kill the process holding the file

#kill 6226

Terminates the cat process.
However, sometimes the process might linger briefly in memory, so you verify again.

5. Recheck

#lsof | grep testfile

Still shows the same entry — meaning process is not yet fully gone.

6. Force kill the process

#kill -9 6226

Sends a SIGKILL signal — **immediate termination**.
Process [1]+ Killed cat testfile confirms success.

7. Reattempt file removal

#rm -f /mnt/raid-xfs/testfile

File successfully deleted this time.
Confirmed by ls -lrth → shows **total 0** (empty directory).

8. Check filesystem usage

#df -h .

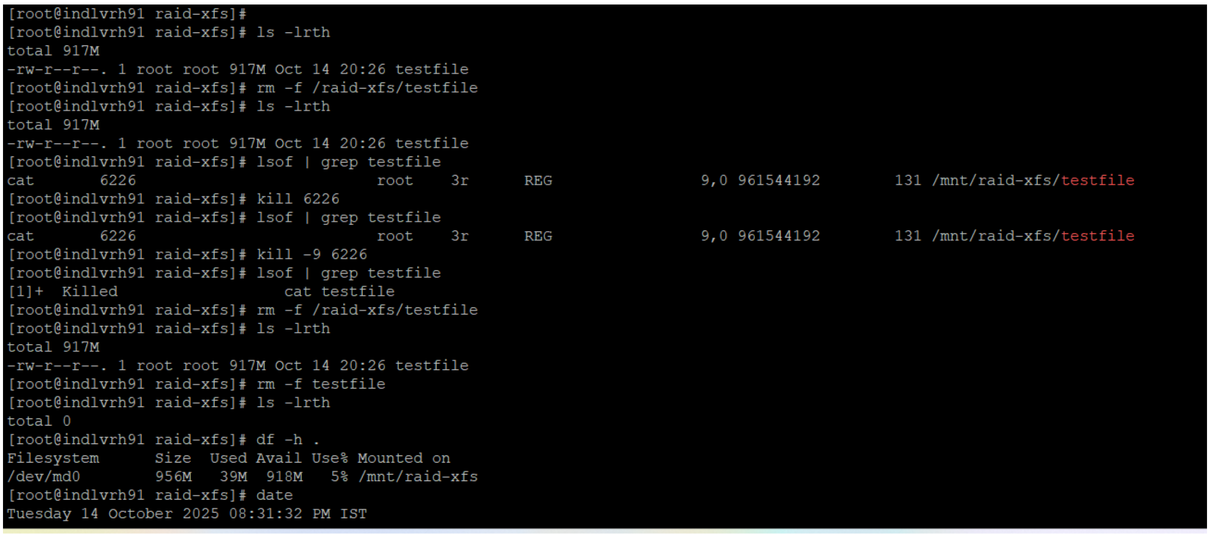
Output:

```
Filesystem  Size  Used  Avail  Use%  Mounted on
/dev/md0    956M  9.5M  947M   1%   /mnt/raid-xfs
```

Confirms that space used by the deleted file is now reclaimed.
The filesystem is healthy and mounted on **/mnt/raid-xfs.**

#date

Shows when this verification was done:
Tuesday 14 October 2025 08:31:32 PM IST



```
[root@indlvrh91 raid-xfs]#
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# rm -f /raid-xfs/testfile
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# lsof | grep testfile
cat       6226                          root    3r      REG                9,0 961544192        131 /mnt/raid-xfs/testfile
[root@indlvrh91 raid-xfs]# kill 6226
[root@indlvrh91 raid-xfs]# lsof | grep testfile
cat       6226                          root    3r      REG                9,0 961544192        131 /mnt/raid-xfs/testfile
[root@indlvrh91 raid-xfs]# kill -9 6226
[root@indlvrh91 raid-xfs]# lsof | grep testfile
[1]+  Killed                  cat testfile
[root@indlvrh91 raid-xfs]# rm -f /raid-xfs/testfile
[root@indlvrh91 raid-xfs]# ls -lrth
total 917M
-rw-r--r--. 1 root root 917M Oct 14 20:26 testfile
[root@indlvrh91 raid-xfs]# rm -f testfile
[root@indlvrh91 raid-xfs]# ls -lrth
total 0
[root@indlvrh91 raid-xfs]# df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        956M   39M  918M   5% /mnt/raid-xfs
[root@indlvrh91 raid-xfs]# date
Tuesday 14 October 2025 08:31:32 PM IST
```

Conclusion

We have set up a **RAID 5 + XFS setup on RHEL 9** is fully functional and stable.

- The **RAID array** has been successfully created, verified, and mounted.
- **XFS filesystem** is operational, formatted, and tested for read/write performance.
- You've successfully demonstrated the **entire lifecycle** of a RAID setup:
    1. Disk verification and cleanup
    2. RAID creation with mdadm
    3. Filesystem creation (XFS)
    4. Mounting and verification
    5. Performance benchmarking
    6. Process/file lock handling
    7. Safe cleanup and teardown