GETTING STARTED

WITH

RAID DEVICES

➢ MANAGING RAID:

- **Redundant Array of Independent Disks (RAID)** to store data across multiple drives. It can help to avoid data loss if a drive has failed.
- In a RAID, multiple devices, such as HDD, SSD, or NVMe are combined into an array to **accomplish performance** or **redundancy** goals not achievable with one large and expensive drive.
- RAID technology is beneficial for those who manage large amounts of data.
- The following are the primary reasons to deploy RAID:
  - It enhances speed
  - It increases storage capacity using a single virtual disk
  - It minimizes data loss from disk failure
  - The RAID layout and level online conversion

RAID TYPES:

FIRMWARE RAID:

- Firmware RAID, also known as ATARAID, is a type of software RAID where the RAID sets can be configured using a firmware-based menu.
- The firmware used by this type of RAID also hooks into the BIOS, allowing you to boot from its RAID sets.

HARDWARE RAID:

- Hardware RAID devices might be internal or external to the system.
- **Internal devices** commonly consist of a specialized controller card that handles the RAID tasks transparently to the operating system.
- **External devices** commonly connect to the system via SCSI, Fibre Channel, iSCSI, InfiniBand, or other high speed network interconnect and present volumes such as logical units to the system.

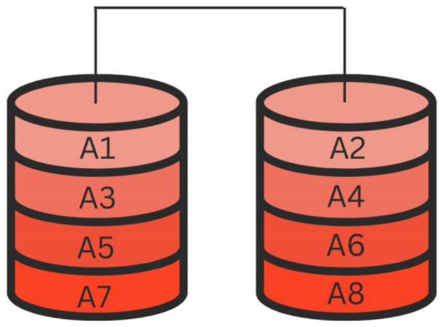
SOFTWARE RAID:

- A software RAID implements the various RAID levels in the kernel block device code. It offers the cheapest possible solution because expensive disk controller cards or hot-swap chassis are not required.
- Software RAID also works with any block storage, which are supported by the Linux kernel, such as SATA, SCSI, and NVMe.

RAID LEVELS and LINEAR SUPPORT:

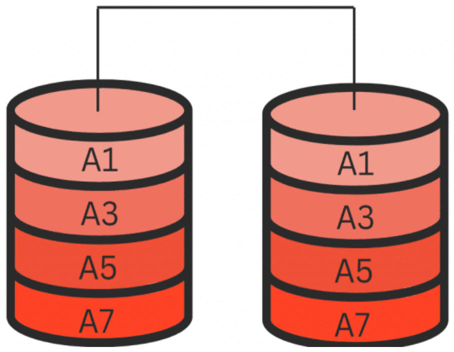
LEVEL 0 (STRIPING):

- RAID level 0, is a performance-oriented striped data mapping technique. This means the data being written to the array is broken down into stripes and written across the member disks of the array, allowing high I/O performance at low inherent cost but provides **no redundancy.**



LEVEL 1 (MIRRORING):

- RAID level 1, provides **redundancy** by writing identical data to each member disk of the array, leaving a mirrored copy on each disk.
- It operates with two or more disks, and provides very good data reliability and improves performance for read-intensive applications but **high costs.**

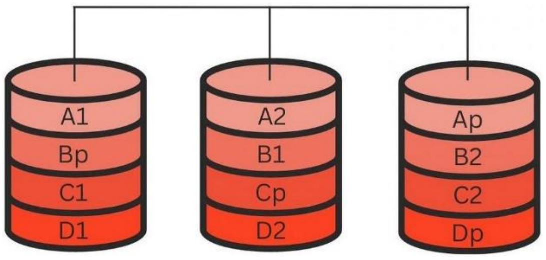


LEVEL 4 (PARITY):

- Level 4 uses parity concentrated on a single disk drive to protect data.
- Parity information is calculated based on the content of the rest of the member disks in the array.
- This information can then be used to reconstruct data when one disk in the array fails.

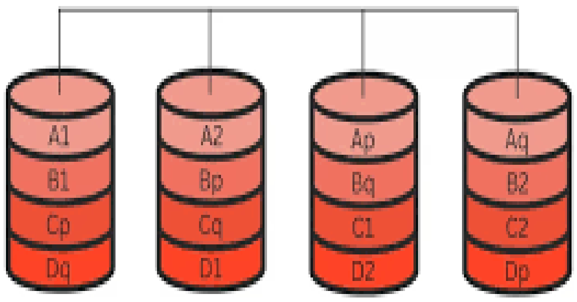
LEVEL 5 (DISTRUBUTED PARITY):

- This is the most common type of RAID. By distributing parity across all the member disk drives of an array, level 5 eliminates the write bottleneck inherent in level 4. The only performance bottleneck is the parity calculation process itself.
- It has asymmetrical performance, and reads substantially outperforming writes. The storage capacity of level 5 is calculated the same way as with level 4.



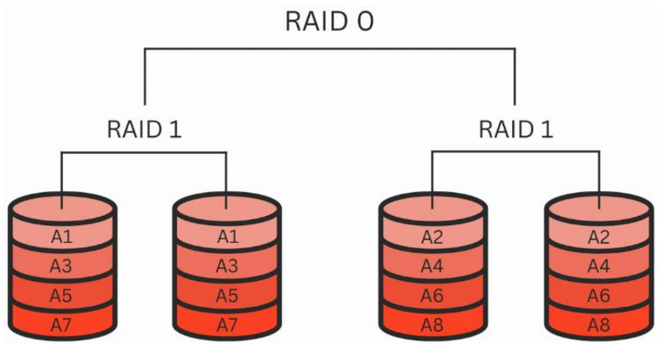
LEVEL 6 (DATA REDUNDANCY AND PRESERVATION):

- This is a common level of RAID when data redundancy and preservation, and not performance, are the paramount concerns, but where the space inefficiency of level 1 is not acceptable. Level 6 uses a complex parity scheme to be able to recover from the loss of any two drives in the array. This complex parity scheme creates a significantly higher CPU burden on software RAID devices and also imposes an increased burden during write transactions. As such, level 6 is considerably more asymmetrical in performance than levels 4 and 5.
- The total capacity of a RAID level 6 array is calculated similarly to RAID level 5 and 4, except that you must subtract two devices instead of one from the device count for the extra parity storage space.



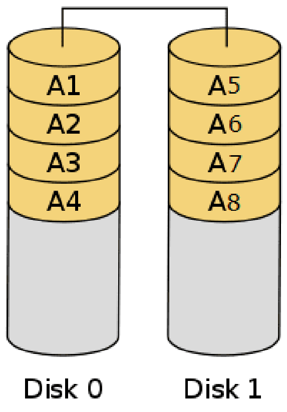
LEVEL 10 (MIRRORING & STRIPING):

- This RAID level attempts to combine the performance advantages of level 0 with the redundancy of level 1.
- It also reduces some of the space wasted in level 1 arrays with more than two devices.



LINEAR RAID:

- Linear RAID is a grouping of drives to create a larger virtual drive.
- In linear RAID, the chunks are allocated sequentially from one member drive, going to the next drive only when the first is completely filled. This grouping provides no performance benefit, as it is unlikely that any I/O operations split between member drives.
- Linear RAID also offers **no redundancy** and **decreases reliability.** If any one-member drive fails, the entire array cannot be used and data can be lost.



IMPLEMENTATING SOFTWARE RAID LEVEL 5:

- Create a software Redundant Array of Independent Disks (RAID) on an existing system using the "**mdadm**" utility.

→ **Create four partitions / take block devices:**

```
#fdisk /dev/nvme0n1
```

→ **Change Partition Type ID values:**

```
#partprobe /dev/nvme0n1
```

```
#lsblk
```

→ **Create a RAID array with three partitions :/dev/nvmeon1p{7,8,9}**

```
#mdadm -Cv /dev/md0 -n3 /dev/nvme0n1p{7,8,9} -l5
```

→ **Check the status of the RAID:**

```
#mdadm -D /dev/md0
```

→ **Detailed information about each device in the RAID:**

```
#mdadm --examine /dev/nvme0n1p7
```

→ **Create a file system on the RAID drive:**

```
#mkfs.xfs -f /dev/md0
```

→ **Create a mount point for RAID drive and mount it:**

```
#mkdir /raid-data
```

```
#mount /dev/md0 /raid-data
```

→ **Mounts the md0 RAID device automatically when the system boots, add an entry for your device to the /etc/fstab file:**

```
/dev/md0   /raid-data   xfs   defaults   0 0
```

```
#systemctl daemon-reload
```

```
#mount /raid-data
```

```
#df -h
```

WHAT TO DO WHEN A DISK FAILS:

- Suppose that a disk in the array failed. In that case, you need to remove that disk from the array and replace it with a working one.

→ **Manually fail a disk in the array:**

```
#mdadm -f /dev/md0 /dev/nvme0n1p9
```

→ **Verify that that disk in the array has failed:**

```
#mdadm -D /dev/md0
```

→ **To remove a disk from the array:**

```
#mdadm -r /dev/md0 /dev/nvme0n1p9
#mdadm -D /dev/md0
```

→ **Now to add a new device in the array:**

```
#mdadm -a /dev/md0 /dev/nvme0n1p10
#mdadm -D /dev/md0
```

Verify that it has been added properly

→ **You want to take RAID array in offline:**

```
#mdadm -S /dev/md0
#mdadm -D /dev/md0
```

→ **To start Again RAID array to online:**

```
#mdadm -A /dev/md0 /dev/nvme0n1p{7,8,10}
#mdadm -D /dev/md0
```

DELETING RAID ARRAY:

→ **Before delete first unmount it and remove entries from /etc/fstab:**

```
#umount -f /raid-data
#mdadm -Sv /dev/md0
```

→ **Now remove RAID array device:**

```
#mdadm -r /dev/md0
```