

Software RAID on SUSE Linux

WHAT?

Basic information about software RAIDs.

WHY?

You need information on RAID levels, or you want to configure or monitor a RAID.

EFFORT

15 minutes of reading time.

GOAL

You will be able to configure a software RAID using YaST.

Publication Date: 23 May 2023

Contents

- 1 Software RAID on SUSE Linux 2
- 2 RAID levels 2
- 3 Configuring software RAID using YaST 5
- 4 Managing software RAID 7

1 Software RAID on SUSE Linux

The purpose of RAID (redundant array of independent disks) is to combine several hard disk partitions into one large virtual hard disk to optimize performance, data security, or both. Most RAID controllers use the SCSI protocol, because it can address a larger number of hard disks in a more effective way than the IDE protocol and is more suitable for parallel processing of commands. There are some RAID controllers that support IDE or SATA hard disks. Software RAID provides the advantages of RAID systems without the additional cost of hardware RAID controllers. However, this requires some CPU time and has memory requirements that make it unsuitable for real high performance computers.

2 RAID levels

RAID implies several strategies for combining several hard disks in a RAID system, each with different goals, advantages, and characteristics. These variations are commonly known as RAID levels.

The RAID levels can be split into the following categories:

Standard levels

These levels and their associated data formats are standardized by the Storage Networking Industry Association (SNIA) in the Common RAID Disk Drive Format (DDF) standard. The standard RAID levels are RAID 0, RAID 1, RAID 2, RAID 3, RAID 4, RAID 5 and RAID 6. For details, refer to [Section 2.1, "Standard RAID levels"](#).

Nested levels

Combine already existing arrays into a new array. For example, RAID 0 + 1 or RAID 1 + 0.

Non-standard levels

Usually, these are proprietary RAID configurations designed to meet specific needs, for example, Linux MD RAID 10.

2.1 Standard RAID levels

Originally, there were only five standard levels of RAID, but other levels have evolved as described in the following sections.

2.1.1 RAID 0

RAID 0 improves the performance of your data operations by spreading out blocks of each file across multiple disks. This data distribution is called striping. The overall capacity is a sum of capacity of disk in the RAID. The benefit of RAID 0 is that the throughput of read and write operations to any file is multiplied by the number of drives because reads and writes are done concurrently.

The disadvantage of RAID 0 is that it does not provide any data backup, so if a disk fails, the whole RAID is destroyed and there is data loss.

2.1.2 RAID 1

RAID 1 provides adequate security for your data, because the data is copied to another hard disk 1:1. This is known as *hard disk mirroring*. This level does not provide striping, so it does not provide a higher read or write throughput. However, the array continues to operate as long as at least one drive is functioning.

RAID 1 requires at least two devices.

2.1.3 RAID 2

In RAID 2, the striping is performed on a bit level. This RAID level is currently not used in practice.

2.1.4 RAID 3

In RAID 3, the striping is performed on a byte level with a dedicated parity drive. RAID 3 is not commonly used in practice.

2.1.5 RAID 4

RAID 4 provides a block-level striping with a dedicated parity drive. If a data disk fails, the parity data is used to create a replacement disk. However, the parity disk might create a bottleneck for write access. This level requires at least three devices.

RAID 4 is not commonly used in practice.

2.1.6 RAID 5

RAID 5 is an optimized compromise between Level 0 and Level 1 in terms of performance and redundancy. The hard disk space equals the number of disks used minus one. The data is distributed over the hard disks as with RAID 0, including the parity data. Parity blocks are there for security reasons. They are linked to each other with XOR, enabling the contents to be reconstructed by the corresponding parity block in case of system failure.

With RAID 5, no more than one hard disk can fail at the same time. If one hard disk fails, it must be replaced when possible to avoid the risk of losing data.

RAID 5 requires at least three disks.

2.1.7 RAID 6

RAID 6 consists of block-level striping with double distributed parity. RAID 6 provides extremely high data fault tolerance by sustaining multiple simultaneous drive failures. Even if two of the hard disks fail, the system continues to be operational, with no data loss.

The performance for RAID 6 is slightly lower but comparable to RAID 5 in normal mode and single disk failure mode. It is very slow in dual disk failure mode. A RAID 6 configuration needs a considerable amount of CPU time and memory for write operations.

RAID 6 requires at least four disks. In general, it requires $N + 2$ disks, where N is the number of disks required to store data and 2 is for the dual parity.

2.2 Nested RAID

2.2.1 RAID 0+1

RAID 0 + 1, also called RAID 01, mirrors striped disks, so data are replicated and shared at the same time. The minimum number of disks is four.

2.2.2 RAID 1+0

RAID 1 + 0, also called RAID 10, is a combination of striping and mirroring. Data are distributed into several disks and each of these disks is mirrored to another disk.

3 Configuring software RAID using YaST

The YaST software RAID configuration can be reached from the YaST Expert Partitioner. This partitioning tool also enables you to edit and delete existing partitions and create new ones that should be used with software RAID. These instructions apply on setting up RAID levels 0, 1, 5, and 6.

1. Launch YaST.
2. In the *System* menu, select *Partitioner* on the right.
3. If necessary, create partitions that should be used with your RAID configuration. Do not format them and set the partition type to *0xFD Linux RAID*. When using existing partitions it is not necessary to change their partition type—YaST will automatically do so.

It is strongly recommended to use partitions stored on different hard disks to decrease the risk of losing data if one is defective (RAID 1 and 5) and to optimize the performance of RAID 0.

For RAID 0 at least two partitions are needed. RAID 1 requires exactly two partitions, while at least three partitions are required for RAID 5. A RAID 6 setup requires at least four partitions. It is recommended to use only partitions of the same size because each segment can contribute only the same amount of space as the smallest sized partition.

4. In the *Add* menu select *RAID*.
5. Select a *RAID Type* and *Add* an appropriate number of partitions from the *Available Devices* dialog.

You can optionally assign a *RAID Name* to your RAID. It will make it available as `/dev/md/NAME`. See [Section 4.1, “Naming software RAID”](#) for more information.

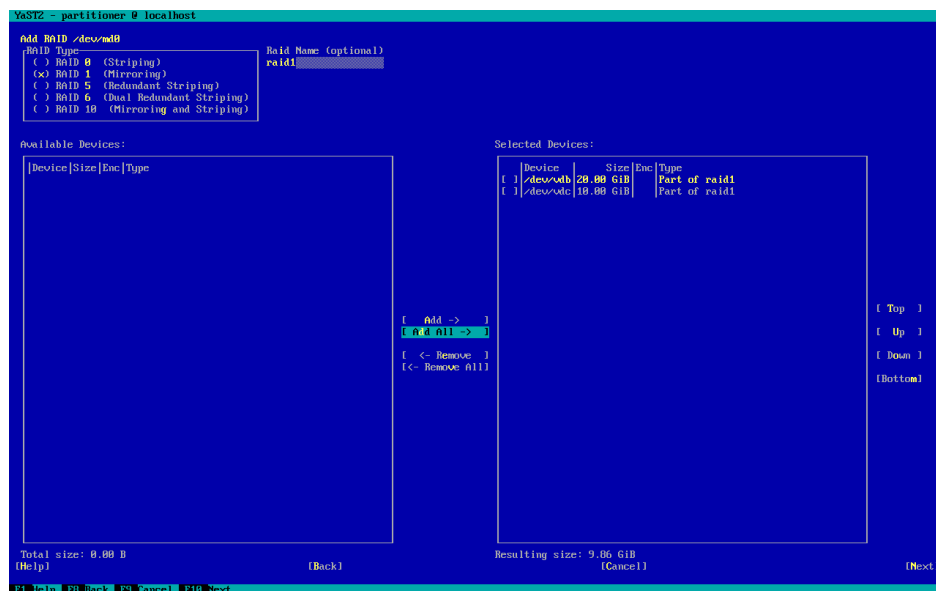


FIGURE 1: EXAMPLE RAID 1 CONFIGURATION

6. Proceed with *Next*.

7. Select the *Chunk Size* and, if applicable, the *Parity Algorithm*. The optimal chunk size depends on the type of data and the type of RAID. See https://raid.wiki.kernel.org/index.php/RAID_setup#Chunk_sizes for more information. More information on parity algorithms can be found with **man 8 mdadm** when searching for the `--layout` option. If unsure, stick with the defaults.

8. Choose a *Role* for the volume. Your choice here only affects the default values for the upcoming dialog. They can be changed in the next step. If in doubt, choose *Raw Volume (Unformatted)*.

9. Under *Formatting Options*, select *Format device*, then select the *File system*. The content of the *Options* menu depends on the file system. Usually there is no need to change the defaults. Under *Mounting Options*, select *Mount device*, then select the mount point. Select *Fstab Options* to add special mounting options for the volume.

10. Proceed with *Next*.

! Important: RAID on disks

While the partitioner makes it possible to create a RAID on top of disks instead of partitions, we do not recommend this approach for a number of reasons. Installing a boot-loader on such RAID is not supported, so you need to use a separate device for booting. Tools like `fdisk` and `parted` do not work properly with such RAIDs, which may lead to incorrect diagnosis and actions by a person who is unaware of the RAID's particular setup.

4 Managing software RAID

After you set up a RAID, you can perform additional administration tasks. For example:

- Change the default RAID names as described in [Section 4.1, “Naming software RAID”](#).
- Monitor the RAID as described in [Section 4.3, “Monitoring software RAIDs”](#).
- Configure the stripe size on the AArch64 architecture as described in [Section 4.2, “Configuring stripe size on RAID 5 on AArch64”](#).

4.1 Naming software RAID

4.1.1 Default names

By default, software RAID devices have names following the pattern `mdN`, where `N` is a number. For example, they can be accessed as `/dev/md127` and are listed as `md127` in `/proc/mdstat` and `/proc/partitions`.

4.1.2 Providing non-default names

As working with the default names might be clumsy, there are two ways how to work around this:

Providing a named link to the device

You can optionally specify a name for the RAID device when creating it with YaST or on the command line with `mdadm --create '/dev/md/ NAME'`. The device name will still be `mdN`, but the link `/dev/md/NAME` will be created:

```
> ls -og /dev/md
total 0
lrwxrwxrwx 1 8 Dec  9 15:11 myRAID -> ../md127
```

The device will still be listed as `md127` under `/proc`.

Providing a named device

If a named link to the device is not sufficient for your setup, add the line `CREATE names=yes` to `/etc/mdadm.conf` by running the following command:

```
> echo "CREATE names=yes" | sudo tee -a /etc/mdadm.conf
```

This will cause names like `myRAID` to be used as a “real” device name. The device will not only be accessible at `/dev/myRAID`, but also listed as `myRAID` under `/proc`. Note that this will only apply to RAIDs configured after the change to the configuration file. Active RAIDs will continue to use the `mdN` names until they get stopped and reassembled.



Warning: Incompatible tools

Not all tools may support named RAID devices. If a tool expects a RAID device to be named `mdN`, it will fail to identify the devices.

4.2 Configuring stripe size on RAID 5 on AArch64

By default, the stripe size is set to 4 kB. If you need to change the default stripe size, for example, to match the typical page size of 64 kB on AArch64, you can configure the stripe size manually using CLI:

```
> sudo echo 16384 > /sys/block/md1/md/stripe_size
```


The above command sets the stripe size to 16 kB. You can set other values such as 4096 or 8192 but the value must be a power of 2.

4.3 Monitoring software RAIDs

You can run **mdadm** as a daemon in the **monitor** mode to monitor your software RAID. In the **monitor** mode, **mdadm** performs regular checks on the array for disk failures. If there is a failure, **mdadm** sends an e-mail to the administrator. To define the time interval of the checks, run the following command:

```
mdadm --monitor --mail=root@localhost --delay=1800 /dev/md2
```

The command above turns on monitoring of the `/dev/md2` array in intervals of 1800 seconds. In the event of a failure, an e-mail will be sent to `root@localhost`.



Note: RAID checks are enabled by default

The RAID checks are enabled by default. It may happen that the interval between each check is not long enough and you may encounter warnings. Thus, you can increase the interval by setting a higher value with the **delay** option.