How to replace a failed harddisk in Linux software RAID

This guide shows how to remove a failed hard drive from a Linux RAID1 array (software RAID), and how to add a new hard disk to the RAID1 array without losing data. I will use gdisk to copy the partition scheme, so it will work with large harddisks with GPT (GUID Partition Table) too.

On this page

1 Preliminary Note

In this example I have two hard drives, */dev/sda* and */dev/sdb*, with the partitions */dev/sda1* and */dev/sda2* as well as */dev/sdb1* and */dev/sdb2*.

`/dev/sda2` and `/dev/sdb2` make up the RAID1 array `/dev/md1`.

```
/dev/sda1 + /dev/sdb1 = /dev/md0
```

```
/dev/sda2 + /dev/sdb2 = /dev/md1
```

`/dev/sdb` has failed, and we want to replace it.

2 How Do I Tell If A Hard Disk Has Failed?

If a disk has failed, you will probably find a lot of error messages in the log files, e.g. `/var/log/messages` or `/var/log/syslog`.

You can also run

```
cat /proc/mdstat
```

and instead of the string `[UU]` you will see `[U_]` if you have a degraded RAID1 array.

3 Removing The Failed Disk

First we mark */dev/sdb1* as failed:

```
mdadm --manage /dev/md0 --fail /dev/sdb1
```

The output of

```
cat /proc/mdstat
```

should look like this:

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid6] [raid10]
md0 : active raid1 sda1[0] sdb1[2](F)
      24418688 blocks [2/1] [U_]

md1 : active raid1 sda2[0] sdb2[1]
      24418688 blocks [2/2] [UU]

unused devices: <none>
```

Then we remove */dev/sdb1* from */dev/md0*:

```
mdadm --manage /dev/md0 --remove /dev/sdb1
```

The output should be like this:

And

```
cat /proc/mdstat
```

should show this:

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid6] [raid10]
md0 : active raid1 sda1[0]
      24418688 blocks [2/1] [U_]

md1 : active raid1 sda2[0] sdb2[1]
      24418688 blocks [2/2] [UU]

unused devices: <none>
```

Now we do the same steps again for `/dev/sdb2` (which is part of `/dev/md1`):

```
mdadm --manage /dev/md1 --fail /dev/sdb2
```

```
cat /proc/mdstat
```

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid6] [raid10]
md0 : active raid1 sda1[0]
      24418688 blocks [2/1] [U_]
```

```
mdadm --manage /dev/md1 --remove /dev/sdb2
```

```
server1:~# mdadm --manage /dev/md1 --remove /dev/sdb2
mdadm: hot removed /dev/sdb2
```

```
cat /proc/mdstat
```

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid6] [raid10]
md0 : active raid1 sda1[0]
      24418688 blocks [2/1] [U_]

md1 : active raid1 sda2[0]
      24418688 blocks [2/1] [U_]

unused devices: <none>
```

Then power down the system:

```
shutdown -h now
```

and replace the old `/dev/sdb` hard drive with a new one (**it must have at least the same size as the old one - if it's only a few MB smaller than the old one then rebuilding the arrays will fail**).

After you have changed the hard disk `/dev/sdb`, boot the system.

The first thing we must do now is to create the exact same partitioning as on `/dev/sda`. We can do this with the command sgdisk from the gdisk package. If you havent installed gdisk yet, run this command to install it on Debian and Ubuntu:

```
apt-get install gdisk
```

For RedHat based Linux distributions like CentOS use:

```
yum install gdisk
```

and for OpenSuSE use:

```
yast install gdisk
```

The next step is optional but recomended. To ensure that you have a backup of the partition scheme, you can use sgdisk to write the partition schemes of both disks into a file. I will store the backup in the /root folder.

```
sgdisk --backup=/root/sda.partitiontable /dev/sda
sgdisk --backup=/root/sdb.partitiontable /dev/sdb
```

In case of a failure you can restore the partition tables with the --load-backup option of the sgdisk command.

```
sgdisk -R /dev/sdb /dev/sda
```

afterwards, you have to randomize the GUID on the new hard disk to ensure that they are unique

```
sgdisk -G /dev/sdb
```

You can run

```
sgdisk -p /dev/sda
sgdisk -p /dev/sdb
```

to check if both hard drives have the same partitioning now.

Next we add `/dev/sdb1` to `/dev/md0` and `/dev/sdb2` to `/dev/md1`:

```
mdadm --manage /dev/md0 --add /dev/sdb1
```

```
server1:~# mdadm --manage /dev/md0 --add /dev/sdb1
mdadm: re-added /dev/sdb1
```

```
mdadm --manage /dev/md1 --add /dev/sdb2
```

Now both arays (`/dev/md0` and `/dev/md1`) will be synchronized. Run

```
cat /proc/mdstat
```

to see when it's finished.

During the synchronization the output will look like this:

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid6] [raid10]
md0 : active raid1 sda1[0] sdb1[1]
      24418688 blocks [2/1] [U_]
      [=>...................]  recovery =  9.9% (2423168/24418688) finish=2.8min speed=127535
K/sec

md1 : active raid1 sda2[0] sdb2[1]
      24418688 blocks [2/1] [U_]
      [=>...................]  recovery =  6.4% (1572096/24418688) finish=1.9min speed=196512
K/sec

unused devices: <none>
```

When the synchronization is finished, the output will look like this:

```
server1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid5] [raid4] [raid6] [raid10]
md0 : active raid1 sda1[0] sdb1[1]
      24418688 blocks [2/2] [UU]

md1 : active raid1 sda2[0] sdb2[1]
```

That's it, you have successfully replaced */dev/sdb*!

About Till Brehm

Over 20 years experience as Software Developer and Linux System Administrator. Till Brehm is the founder and lead developer of the ISPConfig Hosting Control Panel software (since 2000) and he founded HowtoForge in 2005 as a place to share Linux knowledge with other Linux enthusiasts.

view as pdf | print

Share this page:

Recommend Tweet Follow G+

Suggested articles

5 Comment(s)

Add comment

p

□ I'm not a robot

reCAPTCHA
Privacy - Terms

Submit comment

Comments

By: Alexander Reply

Thanks for the tutorial. There's a typo somewhere ('sdgisk' or something). But otherwise, great ;)
(BTW: It's pretty sad that such a complicated process is necessary to replace a faulty disk. That's why Linux will never catch up as a desktop operating system ;))

By: Lance Reply

Awesome! Worked perfectly for me. Thanks!

By: Mike Reply

That optional step to save a copy of the backup table saved my behind after making a syntax error. Thank you so much for the tutorial. For what its worth, I was able to restore the partition table by running this series of commands (which I'm sure can be shortened):

```
sgdisk /dev/sdc
```

Press W to write partition table to disk and exit

By: Chris Hoogendyk Reply

What about the boot sector and installing grub? You could end up with a system that won't boot.

By: Craig Conner Reply

If the drive you're looking to replace is physically removed before you're prepped it as described above, it may show up in /proc/mdstat even after it's physically removed. mdadm can help you clean up that sort of entry, too:mdadm /dev/md4 -r detached

Home › **How to replace a failed harddisk in Linux software RAID**

Sign up now!

Tutorial Info

Author:	Till Brehm
Tags:	centos, debian, fedora, linux, suse, ubuntu

Share This Page

Recommend Tweet Follow Y

40.2k Followers

Popular Tutorials

How to use grep to search for strings in files on the shell

How to create Docker Images with a Dockerfile

How to use the Linux ftp command to up- and download files on the shell

Vim Editor Basics

How to search files from the Terminal on Linux

Setting, Changing and Resetting MySQL Root Passwords

How to Setup ZSH and Oh-my-zsh on Linux

The Perfect Server - Ubuntu 18.04 (Bionic Beaver) with Apache, PHP, MySQL, PureFTPD, BIND, Postfix, Dovecot and ISPConfig 3.1