RAID - mdadm (Software RAID)

(Re)build RAID

Replace a failing or failed disk

1. Check the RAID was created

   ```
   cat /proc/mdstat
   ```

2. Check disk status

   ```
   smartctl -H /dev/sdX
   ```

3. Find serial number

   ```
   smartctl -i /dv/sdX
   ```

4. If the new disk contains partitions
   1. Stop any Raid partitions with

      ```
      mdadm --stop /dev/md1
      ```

   2. Remove the superblocks

      ```
      mdadm --zero-superblock /dev/sdX1
      ```

   3. Remove existing partitions with `gdisk /dev/sdX`

      ```
      Command (m for help): d
      ```

   4. Create new RAID partition (if asked remove the existing signature)

      ```
      Command (m for help): d
      Command (m for help): n
      Command (m for help): t,fd00
      ```

5. Add the new drive to the RAID

   ```
   mdadm /dev/md0 --add /dev/sdc1
   ```

6. If the system does not need to use the disks during resync you may want to (temporarily) increase the sync speed:

   ```
   echo 1000000 > /proc/sys/dev/raid/speed_limit_max
   ```

   Before doing this check the current sync speed:

   ```
   cat /proc/sys/dev/raid/speed_limit_max
   ```

7. If the RAID is incomplete, rebuilding (resyncing) of the RAID starts instantly. If the RAID is complete including the bad drive, and you just added a spare drive, you can proceed as follows (requires mdadm 3.3+ and a 3.2+ kernel)

   ```
   mdadm /dev/md0 --replace /dev/sdX1 --with /dev/sdc1
   ```

   sdX1 is the device you want to replace, sdc1 is the preferred device to do so and must be declared as a spare on your array. The –`with` option is optional, if not specified, any available spare will be used. After resyncing the RAID the replaced drive will be marked as failed.

8. Remove the replaced disk which is marked as failed after resyncing has completed

   ```
   mdadm --manage /dev/md0 --remove /dev/sdX1
   ```

9. Remove a disk from the array which was physically removed before

   ```
   mdadm --remove /dev/md0
   ```

10. Compare ouput of `mdadm -Es` to the contents of `/etc/mdadm/mdadm/conf`

    ```
    mdadm -Es >> /etc/mdadm/mdadm.conf
    ```

- How to safely replace a not-yet-failed disk in a Linux RAID5 array?
- mdadm: How to Remove a Drive From a RAID Array
- mdadm
- Recovering a failed software RAID
- Activating and deactivating volume groups
- RAID Recovery
- Linux Create Software RAID 1 (Mirror) Array
- Linux Rename a RAID Array From md0 to md2
- Mdadm missing superblock
- How do I rename an mdadm raid array?
- How can I quickly copy a GPT partition scheme from one hard drive to another?
- 7 Steps to Safely Replace a Drive in a Linux MD RAID Array
- Replacing and resizing a linux software raid
- Resizing an encrypted filesystem with LVM on Linux
- Extend a LUKS encrypted partition to fill disk
- Reducing Linux Software RAID Rebuild Speed?

Check wether all volumes get mounted during system boot

- to check wether root and swap are mounted, enter:

  ```
  mount
  free -m -t
  ```

- to check mismatching uuid's, enter:

  ```
  blkid
  ls -la /dev/disk/by-uuid
  ```

```
cat /etc/fstab
```

- to fix, replace the uuid's found in `/etc/fstab` with the ones found in `/dev/disk`. Make sure you copy the correct uuid (md0, md1) to the respective entry in fstab.

```
vim /etc/fstab
```

Resync

Most Debian and Debian-derived distributions create a cron job which issues an array check at 0106 hours each first Sunday of the month in /etc/cron.d/mdadm. This task appears as resync in /proc/mdstat and syslog. So if you suddenly see RAID-resyncing for no apparent reason, this might be a place to take a look.

Normally the kernel will throttle the resync activity (c.f. nice) to avoid impacting the raid device performance.

However, it is a good idea to manage the resync parameters to get optimal performance.

Raid 1, 5, 6

Rebuild speed

- Get current system values:

```
sudo sysctl dev.raid.speed_limit_min
sudo sysctl dev.raid.speed_limit_max
```

- Default system values on Debian 10:

```
dev.raid.speed_limit_min = 1000
dev.raid.speed_limit_max = 200,000
```

- Reduce max limit to make server more responsive during resync (2021-12-05):

```
sudo sysctl -w dev.raid.speed_limit_min=10,000
sudo sysctl -w dev.raid.speed_limit_max=100,000
```

read-ahead

- Get current read-ahead (in 512-byte sectors) per Raid device (default value is 512 on Debian 10):

```
blockdev --getra /dev/mdX
```

- Set to 32 MB:

```
blockdev --setra 65536 /dev/mdX
```

- Set to 65536 on a server with 32GB memory, 32768 on a server with 8GB memory (2021-12-05)

Disable NCQ

- Get NCQ depth on each physical Drive in Raid (default value is 31):

```
cat /sys/block/sdX/device/queue_depth
```

- Disable NCQ:

```
echo 1 > /sys/block/sdX/device/queue_depth
```

Raid 5, 6 only

stripe_cache_size

It records the size (in pages per device) of the stripe cache which is used for synchronising all write operations to the array and all read operations if the array is degraded. The default is 256 which equals to 3MB memory consumption. Valid values are 17 to 32768. Make sure your system has enough memory available: memory_consumed = system_page_size * nr_disks * stripe_cache_size.

- Find system page size, on Debian 10 this is 4096:

```
getconf PAGESIZE
```

- Set to 384MB memory consumption on a 3 disk:

```
sudo echo 32768> /sys/block/md0/md/stripe_cache_size
```

- Set to 32768 on a server with 32 GB memory, set to 16384 on a server with 8 GB memory (2021-12-05)

Prepare RAID with single disk

Prepare new disk

1. If the new disk contains partitions
    1. Stop any Raid partitions with

    ```
    mdadm --stop /dev/md1
    mdadm --remove /dev/md1
    ```

    2. Remove the superblocks

    ```
    mdadm --zero-superblock /dev/sdX1
    ```

    3. Remove existing partitions with `fdisk /dev/sdX`

2. Create a new partition utilizing the full disk space. When asked, remove the existing signature. Change partition type to `Linux RAID`

    ```
    sudo fdisk /dev/sdX
    Command (m for help): d
    Command (m for help): n
    Command (m for help): t,29
    ```

3. Create the RAID

    ```
    mdadm --create /dev/mdX --level=raid1 --raid-devices=2 /dev/sdX1
    missing
    ```

4. Check the RAID was created

    ```
    cat /proc/mdstat
    ls /dev/md*
    ```

5. Add a second disk

    ```
    mdadm --manage /dev/mdX --add /dev/sdX1
    ```

Links

- Can I create a software RAID 1 with one device
- Convert a single drive system to RAID
- Removal of mdadm RAID Devices – How to do it quickly?
- Replacing A Failed Hard Drive In A Software RAID1 Array

Move RAID to a new machine

1. Scan for the old raid disks

    ```
    sudo mdadm --assemble --scan
    ```

2. Mount the raid manually to confirm

    ```
    blkid
    sudo mount /dev/md0 /mnt
    ```

3. Append info to mdadm.conf

    ```
    mdadm --detail --scan >> /etc/mdadm/mdadm.conf
    ```

4. Update initramfs

    ```
    update-initramfs -u
    ```

Troubleshooting

- Make sure the output of “mdadm –detail –scan” matches your /etc/mdadm/mdadm.conf
- Examine /etc/fstab

Links

- How do I move a Linux software RAID to a new machine?
- Using mdadm --examine to write mdadm.conf
- Can I transfer my mdadm Software raid to a new system in case of hardware failure?

Increase drive capacity in a RAID -> LVM -> CRYPT setup

Replace drives in RAID

1. Follow Replace a failing or failed disk at the top of this page
2. Resize the array to the maximum supported by the underlying partitions

   ```
   mdadm --grow /dev/md5 -z max
   ```

3. Follow the progress with

   ```
   watch -d cat /proc/mdstat
   ```

Increase LVM

1. Check size of physical volume with

   ```
   pvdisplay
   ```

2. Increase physical volume to utilize all available space

   ```
   pvresize /dev/mdX
   ```

3. Increase logical volume to utilize all available space

   ```
   lvextend -l +100%FREE /dev/<volume_group>/<logical_volume>
   ```

Increase LUKS

1. Inform LUKS to utilize all available space, you need the backup key to do this

   ```
   cryptsetup resize /dev/mapper/<volume_group>/<logical_volume>_crypt
   ```

Increase file system

1. On-line resize file system

   ```
   resize2fs -p /dev/mapper/<volume_group>/<logical_volume>_crypt
   ```

Links

- Resync
- 5 Tips To Speed Up Linux Software Raid Rebuilding And Re-syncing
- RAID resync – Best practices
- A guide to mdadm
- How To Resize RAID Partitions (Shrink & Grow) (Software RAID)
- Formatting a RAID volume with ext4
- How To Create RAID Arrays with mdadm on Ubuntu 16.04
- How To Resize RAID Partitions (Shrink & Grow) (Software RAID)
- Formatting a RAID volume with ext4
- How to wipe free disk space in Linux?
- Setting up swap space

Recovery Epione

google: add drive to degraded raid 1 linux

1. Identify the Array and New Drive Check the current status of your RAID array to identify the device name (e.g., /dev/md0) and confirm which drive is missing. Use the following command: bash cat /proc/mdstat Identify your new physical drive (e.g., /dev/sdb) using: bash lsblk 2. Copy Partition Table (Optional but Recommended) If your RAID is based on partitions rather than whole disks, you should copy the partition layout from the surviving drive (e.g., /dev/sda) to the new drive (e.g., /dev/sdb): bash # WARNING: This will overwrite the partition table on /dev/sdb sudo sfdisk -d /dev/sda | sudo sfdisk /dev/sdb 3. Add the New Drive to the Array Use the mdadm command to add the new drive (or its partition) to the degraded array: bash # If using the whole disk sudo mdadm /dev/md0 –add /dev/sdb

# If using a specific partition sudo mdadm /dev/md0 –add /dev/sdb1 4. Monitor the Rebuild Process Once added, the system will automatically begin synchronizing data to the new drive. You can monitor the progress (percentage and estimated time) with: bash watch -n 1 cat /proc/mdstat 5. Update Bootloader (If Necessary) If your RAID array contains the operating system, you must install the bootloader (GRUB) onto the new drive to ensure the system can boot if the other drive fails: bash sudo grub-install /dev/sdb Summary of Useful Commands Check details: sudo mdadm –detail /dev/md0 Remove a failed drive (if still listed): sudo mdadm /dev/md0 –remove /dev/sdb1 Zero out old metadata on a used disk: sudo mdadm –zero-superblock /dev/sdb1 mdadm raid1 irregular degraded without reason - Server Fault Jun 29, 2565 BE — One drive goes bad is not enough reason for I/O errors appearing on the MD RAID device. It should never display I/O...

Server Fault

How to re-add a drive when RAID is in degraded mode? Dec 20, 2559 BE — Sorted by: 26. Adding Back a RAID 1 Member. To add a RAID member back into the array, use: mdadm -a For example: $ ...

Server Fault

RAID1: How do I “Fail” a drive that's marked as “removed”? Dec 26, 2552 BE — There are just a few steps to get it back up and running. * Setup partitions on the replacement disk. These partiti...

Server Fault