RAID - mdadm (Software RAID)

Grow RAID1 to RAID5

1. Existing RAID1 array is md127, array to be removed is md126
2. Backup all data
3. Make sure the devices are unmounted

```
sudo umount /dev/md126
```

4. Check the details

```
sudo mdadm --detail /dev/md126
```

5. Stop the device

```
mdadm -S /dev/md126
```

6. Zero out the superblocks of physical devices that belong to the MD device

```
sudo mdadm --zero-superblock /dev/sde1
```

7. Change the array from RAID1 to RAID5 (still only 2 drives)

```
sudo mdadm /dev/md127 --grow --level=5
```

8. Add the new drive to the array

```
sudo mdadm /dev/md127 --add /dev/sde1
```

9. Start the sync for all 3 devices

```
sudo mdadm --grow /dev/md127 --raid-devices 3
```

10. Watch the progress with

```
watch -n 15 -d cat /proc/mdstat
```

- Converting RAID1 array to RAID5 using the mdadm grow command
- How to remove Linux software RAID MD device?
- How To Create RAID Arrays with mdadm on Ubuntu 16.04
- How To Manage RAID Arrays with mdadm on Ubuntu 16.04
- Software RAID How-to
- Disassemble a RAID 1 arragement without removing/reinstalling the system

Desired configuration of the disks

hdd	partition	raid1	lvm	mount point
sda	sda1	md0	-	/

hdd	partition	raid1	lvm	mount point
sdb	sdb1	md0		
sda	sda2	-	-	swap
sdb	sdb2	-		swap
sdc	sdc1	md1	vg1	/home, /srv,
sdd	sdd1	md1	vg1	/media
sde	sde1	md2	vg1	
sdf	sdf1	md2	vg1	
sdg	sdg1	md3	vg2	/backup
sdh	sdh1	md3	vg2	/backup

see also Hardware configuration

Replace a failed RAID disk

Replace the disk, then copy the partition information from the good disk, randomize the UUID, and re-read the partition information into the system. First, install gdisk from the Debian Universe repositories.

```
apt-get install gdisk
sgdisk -R=/dev/sdb /dev/sda
sgdisk -G /dev/sdb
partprobe
```

Taken from How can I quickly copy a GPT partition scheme from one hard drive to another?

Other resources on mdadm and How do I rename an mdadm raid array?:

- 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

Setup configuration:

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

Install Base

1. Install Ubuntu Hardy (8.04) Alternate Install: F6-F6-Expert Mode
2. Install Raid1 with MD package activated and assign mount points

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:

```
ls -la /dev/disk/by-uuid
cat /etc/fstab
```

to fix, do:

```
vim /etc/fstab
```

replace the uuid's found in fstab with the ones found in /dev/disk. Make sure you copy the correct uuid (md0, md1) to the respective entry in 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 = 200000
```

- Set for all Raids:

```
sudo sysctl -w dev.raid.speed_limit_min=50000
sudo sysctl -w dev.raid.speed_limit_max=1000000
```

read-ahead

- Set current read-ahead (in 512-byte sectors) per Raid device (default value is 256 or 512):

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

- Set read-ahead (32MB):

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

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 Raid:

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

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?

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