Expanding a software RAID on Debian by migrationg to new large hard disks

The setup: two physical drives of a RAID 1 (mirror) will be replaced with two more capacious drives, we will do a "hot" replacement directly from the running operating system, without the need to boot from an external boot media. This guide has been tested on Debian distributions 6, 7, 8 both 32 and 64 bits.

We assume that the two drives are partitioned with a root and swap partition and that these two partitions build up the two raid devices md0 and md1:

- **The sda1 and sdb1 partition that make up the volume md0 (root)**
- **The sda2 and sdb2 partitions that make up the volume md1 (swap)**

First, do a backup of all your data, even if the guide has been tested several times, bad luck is still there ... We provide (in my opinion, the best solution) for a replacement, we disconnect one of the two old drives with the machine off, then connect one new temporary drive, then resynchronize, so we will still have the RAID functioning and a single drive (with RAID in degraded mode) and you can still recreate a functioning RAID if something does not work. To do this just type these commands after replacing the full drive with the empty one:

```
sfdisk -d /dev/hd source | sfdisk —force /dev/hd empty
mdadm —zero-superblock /dev/hd empty
mdadm —add /dev/md(raid) /dev/hd(empty)
```

Repeat the last two lines for each RAID / partition on the system

(for example:

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

then

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

etc ..

In this way, we forced the partition table from the source drive to the empty target, then add this to the RAID.

Obviously you have to wait until the end of the reconstruction process, monitored by:

```
watch cat / pro / mdstat
```

(And of course CTRL + C to exit the process)

Shutdown the machine at this point, we add two new disks, which become respectively sdc and sdd, then start the server and run the command:

```
fdisk -l
```

The result should look similar to this:

```
Disk /dev/sda: ……MB , …….Bytes
… heads, …sectors/track , …... cylinders
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk: identifier : ……..

Device boot Start End Blocks Id   System

/dev/sda2       970          1044 602437+ fd   Linux raid autodetect

/dev/sda2           1 969 7783461 fd   Linux raid autodetect

……

/dev/sdb1     *       1 969 7783461 fd   Linux raid autodetect

/dev/sdb2           970       1044 602437+ fd   Linux raid autodetect

……

Disk /dev/sdc: …Mb, …….Bytes

……

Disk /dev/sdd: …Mb, …….Bytes

……

Disk /dev/md0: …Mb, …….Bytes

….…

Disk /dev/md1: …Mb, …….Bytes
```

Now, type:

```
fdisk /dev/sdc
```

we create a new partition with the **n** command,

we select **p** primary partition,

we select the partition number (**1**,2,3, etc ...),

we select the starting cylinder (**1**),

we select the size of the primary partition by typing the **+** symbol followed by the size in GB and **GB** from letters (for example: **+480GB**)

we make it bootable with the command **a**, selecting the same number of partition.

we change the partition ID in "**fd**" (Linux Raid Autodetect) with **t**, and writing as Hex code, just "**fd**"

Then we repeat ALL the commands for the second partition (the partition will be the swap).

We will save the changes with **w.**

Obviously, pay attention to the size of the partitions. The best solution is to subtract the current swap size of the total of the free space to create the working partition with the widest possible space.

Now we will perform exactly the same operations on the SDD drive, making sure to create the same partition with the same sizes as we previously used.

At this point we do an exchange between the first drive "full" and the first new drive, adding the latter as a spare to the RAID:

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

Then we should verify the raid disk situation, it should be like this:

```
md0: active raid 1 sdc1[S] sda1[2] sdb1[3]

……blocks super 1.2 [2/2] [UU]
```

Then we put in the original drive:

```
mdadm -f /dev/md0 /dev/sda1
```

and remove it from the RAID:

```
mdadm -r /dev/md0 /dev/sda1
```

Now we verify that the reconstruction started automatically and wait until it is completed by monitoring again with:

```
watch cat /proc/mdstat
```

(And of course CTRL + C to exit the process)

At the end we install the bootloader (grub) on the new drive (/ dev / sdc):

```
grub-install /dev/sdc
```

The same process just has to be repeated for each of the system partition:

```
mdadm -add /dev/md0 /dev/sdd1
```

```
mdadm -f /dev/md0 /dev/sdb1
```

```
mdadm -r /dev/md0 /dev/sdb1
```

And we wait again until the reconstruction is finished:

```
watch cat /proc/mdstat
```

(And of course CTRL + C to exit the process)

Next, do the same procedure for the first swap partition of two new drives:

```
mdadm -add /dev/md1 /dev/sdc2
```

```
mdadm -f /dev/md0 /dev/sda2
```

```
mdadm -r /dev/md0 /dev/sda2
```

Wait again until it has finished the reconstruction:

```
watch cat /proc/mdstat
```

(And of course CTRL + C to exit the process)

And the last drive:

```
mdadm -add /dev/md1 /dev/sdd2
```

```
mdadm -f /dev/md0 /dev/sdb2
```

```
mdadm -r /dev/md0 /dev/sdb2
```

Wait again until it has finished the reconstruction:

```
watch cat /proc/mdstat
```

(And of course CTRL + C to exit the process)

At this point reinstall the bootloader on both drives:

```
grub-install /dev/sdc
```

```
grub-install /dev/sdd
```

At this point we find ourselves with the old drive expelled from the RAID, and the new drives are fitted and operational. But with the usable space below the maximum allowed by the partition

Then we proceed with the enlargement of the partition and the verification thereof, as usual for both volumes:

```
mdadm –grow –raid-devices=2 /dev/md0
```

```
mdadm -A –scan
```

```
mdadm –grow /dev/md0 –size=max
```

```
mdadm –examine –scan
```

```
e2fsck -f /dev(md0
```

```
resize2fs /dev/md0
```

```
mdadm –grow –raid-devices=2 /dev/md1
```

```
mdadm -A –scan
```

```
mdadm –grow /dev/md1 –size=max
```

```
mdadm –examine –scan
```

```
e2fsck -f /dev/md1
```

```
resize2fs /dev/md1
```

When finished, turn off the PC, unlink the old drives, and restart the machine.

Even if the restart was working, we reconstruct the grub for security reasons since the launch of the new drive will have taken on the name of the old (sda and sdb):

```
mv /boot/grub/device.map/boot/grub/device.map.old
```

```
grub-mkdevicemap
```

```
update-grub2 && grub-install /dev/sda && grub-install /dev/sdb
```

If you want to be absolutely sure that the new raid setup is working, shut down the machine, disconnect one of the new drives and start again. It should start without issues. Afterwards, shut down again, connect the drive and start. You can use this command to check the raid state.

```
watch cat /proc/mdstat
```

(And of course CTRL + C to exit the process)

We finished the work, **we celebrate**.

Suggested articles

1 Comment(s)

Add comment

Name *

Email *

p

I'm not a robot reCAPTCHA Privacy - Terms

Submit comment

Comments

From: Jehu **at:** *2017-01-02 00:40:59* Reply

With you did the for Raid 6 or 10 with GPT partitions...thx