

Linux software RAID using mdadm

This guide will outline the basics of software-based RAID using mdadm in Linux.

Mdadm is particularly useful if you need some resiliency against data loss due to disk failure and don't mind taking the performance hit on system resources that goes along with software RAID.

Please note: Software RAID is by no means perfect and is not a substitute to a hardware-based RAID solution. I will not accept any responsibility if you follow this guide and you lose your data!

Here is a quick breakdown of what is needed to create a software RAID array using mdadm:

- Create array

Linux software RAID using mdadm

Written by Admin

Thursday, 22 September 2011 18:50 - Last Updated Wednesday, 23 December 2015 16:36

- Verify array
- Create mdadm configuration file (to ensure that RAID array gets created every time the system is booted)
- Make mount point
- Add mount entry to fstab

Let's begin;

Create array

Syntax:

```
mdadm --create --chunk=X --level=Y --raid-devices=Z
```

If we were going to create a RAID 5 array with 3 devices (/dev/sdc1/ dev/sdd1 & /dev/sde1), a block size of 32Kb and virtual device name of /dev/md0 the command would be:

```
mdadm --create /dev/md0 --chunk=32k --level=5 --raid-devices=3 /dev/sdc1 /dev/sdd1 /dev/sde1
```

mdadm supports the following RAID levels:

- RAID 0 - Block level stripe
- RAID 1 - Mirror
- RAID 4 - RAID 0 with an additional device for parity
- RAID 5 - RAID 4 with parity distributed between all disks in the array
- RAID 6 - RAID 5 with 2 parity segments per stripe
- RAID 10 - RAID 1 mirrorsets striped like RAID 0

As a side note, the chunk size is an optional parameter. If this option is not specified, mdadm will default to 64k block size.

Verify array

Now that we have created the mdadm RAID array, we need to verify that the array has been correctly;

Syntax:

```
cat /proc/mdstat
```

Which should give us an output similar to the below:

```
Personalities : [raid5] read_ahead 1024 sectors md0 : active raid5 sde1[2] sdd1[1] sdc1[0]
4120448 blocks level 5, 32k chunk, algorithm 3 [3/3] [UUU]
unused devices:
```

The parts of the command output that we're really interested in are the **"md0 : active raid5 sde1[2] sdd1[1] sdc1[0]"** and **"unused devices:"**

" and "

unused devices:

" which confirms the following;

Linux software RAID using mdadm

Written by Admin

Thursday, 22 September 2011 18:50 - Last Updated Wednesday, 23 December 2015 16:36

- There is an active mdadm RAID array pointing at /dev/md0.
- The active mdadm RAID array is a RAID 5 array consisting of disks /dev/sdc1, /dev/sdd1 & sde1.
- There are no mdadm unused devices.

All of the above is our desired configuration, based upon the configuration we set in the "Create Array" section of this guide.

Create mdadm configuration file

Now that we have got a RAID array configured within mdadm, we need to ensure that our RAID array gets correctly created (pieced back together) after a reboot.

Linux software RAID using mdadm

Written by Admin

Thursday, 22 September 2011 18:50 - Last Updated Wednesday, 23 December 2015 16:36

This is accomplished by performing the following command.

For Ubuntu or Debian:

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

For other distros:

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

Create mount point

Making the mount point is the easiest part of this entire process. It's a simple matter of making an empty folder, that's all there is to it really.

Linux software RAID using mdadm

Written by Admin

Thursday, 22 September 2011 18:50 - Last Updated Wednesday, 23 December 2015 16:36

Assuming that we want to have our RAID array mounted into /mnt/RAID, all we would do is:

```
mkdir /mnt/RAID
```

It doesn't get easier than this, does it?

Add mount entry to fstab

This is the final part to our guide (well 2 parts) to our guide. I'll this section into two distinct operations; Formatting our new RAID array & adding the fstab entry.

Formatting the RAID array

Linux software RAID using mdadm

Written by Admin

Thursday, 22 September 2011 18:50 - Last Updated Wednesday, 23 December 2015 16:36

I'm going to be using gParted for the purpose of this guide as it's a nice and easy gui partition editor.

- Open up gParted (0.8.1)
- From the drop-down list on the right-hand side of the window, select our /dev/md0 device
- Click on 'new' on the left hand side of the window, to bring up the partition create dialogue
- From the "File system" dropdown box, select "ext3" or "ext4"
- Accept all other remaining defaults and click on "Add"

Add mount entry to fstab

Next we need to create the fstab entry to make the system always mount the device on boot.
Take a backup of the existing fstab:

```
cp -p /etc/fstab /etc/fstab_bak
```

Open the fstab in your favorite editor and add the following line to the end of it:

```
/dev/md0 /mnt/RAID ext3 defaults 0 0
```

This is basically telling our system

Linux software RAID using mdadm

Written by Admin

Thursday, 22 September 2011 18:50 - Last Updated Wednesday, 23 December 2015 16:36

- Where the source device is (/dev/md0)
- The folder that we wish to mount it in (/mnt/RAID)
- The file system type (ext3 or ext4 based upon your choices when formatting)
- The mount options (in this case 'defaults')
- The dump option (0 means ignore)
- The fsck order (0 means "do not check", you can (should) enter a value in here to tell the o/s what fsck priority you wish to designate for this mountpoint (typically 2).

Save these changes and reboot the system; you're done!