

Boolean:  Case 

---

# Installing Debian with SATA based RAID

**There is now a version of this document for 2.6 kernels [here!](#)**

I've read that ( <http://platform.progeny.com/anaconda/>) there will soon be an installer that will do raid installs and perhaps even support SATA, but today it is manual. My install on a Intel D865PERL mother board got 'interesting'. The last Debian (beta 4) testing installer does support SATA as does a version of Debian/Libranet, but going on to RAID is a manual task.

The basic idea was to Install all on one drive, then partition the second drive with just the same sizes. Install mdadm (the new and improved replacement for raidtools). I'm assuming we are using clean drives. If one re-uses previously used disks the superblocks must be zeroed (--zero-superblock option to mdadm) before adding the partitions to the array.

The whole mess is harder than it should be and I hope this page becomes the basis for someone's automating script (hint hint).

## Overview of steps

- Install Debian on first drive (/dev/hdg)
- Creating a degraded RAID1 array on disk 2 (/dev/hde)
- Update initrd
- Copying the Debian installation over from disk 1 (hdg to hde)
- Fix fstab on /Dev/md2
- Add disk 1 to the degraded array to create final raid array
- Update initrd again
- Produce /etc/mdadm/mdadm.conf
- Setup the monitoring daemon

In this example Disk1 and disk2 are /dev/hdg and /dev/hde respectively.

**The partitions I used are:**

```
/boot 200m
/swap 1Gig raid0
/ rest of drive
```

### A note on raid systems and swap:

It is possible for a system to crash if the swap area drive fails. If this is not a concern you could also set it up as simple swap areas or as raid0 to improve performance. (Raid1 also provides faster reads as a side benefit, but there is the cost slower writes - but swap tends to be read more times than written so you might still come out ahead.)).

## Install Debian on first drive

I used the following partitions

Device	Size	Id	Eventual mount point
/dev/hdg1	200M	83	/boot/
/dev/hdg2	1G	82	swap
/dev/hdg3	100G	83	/

I'm assuming you know how to do the above by running fdisk from the install of your choice.

After installing:

```
#apt-get install wajig (you don't want to be without wajig)
#wajig install mdadm
#wajig install rsync
#wajig install e3
```

## Partition Drive2 with fdisk

Device	Size	Id	Eventual mount point
/dev/hde1	200M	fd	/boot/
/dev/hde2	1G	fd	swap
/dev/hde3	100G	fd	/

## Create the raid devices

```
# mdadm --create /dev/md0 --level 1 --raid-devices=2 missing
```

*The above line is the long version of the next line just for reference. .*

```
#mdadm -Cv /dev/md0 -l1 -n2 missing /dev/hde1
#mdadm -Cv /dev/md1 -l1 -n2 missing /dev/hde2
#mdadm -Cv /dev/md2 -l1 -n2 missing /dev/hde3
```

This creates 3 degraded RAID1 devices (for /boot, swap, and /) consisting of a dummy drive "missing" (why was this so hard to figure out!)

A cat of /proc/mdstat will show your degraded raid devices are up and running

```
#cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 ide/host2/bus1/target0/lun0/part1[0]
979840 blocks [2/1] [U_]
md2 : active raid1 ide/host2/bus1/target0/lun0/part3[0]
116045440 blocks [2/1] [U_]
md1 : active raid1 ide/host2/bus1/target0/lun0/part2[0]
192640 blocks [2/1] [U_]
unused devices: <none>
```

Note how one drive in all three cases is missing ("\_") as opposed to 'Up and running' ("U").

## Creating the file systems and Mount them

```
#mkfs.ext3 /dev/md0
#mkfs.ext3 /dev/md2
#mkdir /mntroot
#mkdir/mntboot
#mount /dev/md2 /mntroot
#mount /dev/md0 /mntboot
```

## Fix up initrd

If you are using a SATA drive you pay attention!

edit /etc/mkinitrd/mkinitrd.conf and change:

```
root=probe
```

to

```
ROOT="/dev/md2 ext3"
```

This tells init to use what it takes to boot off of a raid device not the /dev/hdg device currently used.

Update kernel to kernel-img-2.4.26 (need this to get raid1.o module for mkinitrd)

Now run

```
#mkinitrd -o /boot/initrd.img-2.4.26 2.4.26
```

## Edit /boot/grub/menu.lst

Copy the old listing below and paste it below:

```
title Debian GNU/Linux, kernel 2.4.26
root (hd0,0)
kernel /vmlinuz-2.4.26 root=/dev/hde3 ro
initrd /initrd.img-2.4.26
savedefault
boot
```

Now edit the new stanzas noting the changes in bold:

```

title Debian GNU/Linux, kernel 2.4.26
Debian GNU/Linux, kernel 2.4.26 RAID
root (hd0,0)
kernel /vmlinuz-2.4.26 root=/dev/md2 ro
initrd /initrd.img-2.4.26
savedefault
boot

```

## Copy over everything to the degraded raid

```

#rsync -auHx --exclude=/proc/ --exclude=/sys/ --exclude=,
#mkdir /mntroot/proc
#chmod 555 /mntroot/proc
#rsync -auHx /boot/ /mntboot/

```

## Make Changes in /mntroot/etc/fstab

Changes in bold

```

# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
/dev/md1 none swap sw 0 0
/dev/md0 /boot ext3 defaults 0 2
/dev/md2 / ext3 defaults,errors=remount-ro 0 1
/dev/hda /media/cdrom iso9660 ro,user,noauto 0 0
/dev/fd0 /media/floppy auto rw,user,noauto 0 0
/dev/hda /cdrom iso9660 ro,user,noauto 0 0

```

## Reboot

If it won't work you should be able to boot off of the old grub listing

At this point the box should be running off the degraded RAID1 devices.

## Attach the original drive's partitions to the existing (degraded) RAID arrays:

```

# mdadm /dev/md0 -a /dev/hdg1
# mdadm /dev/md1 -a /dev/hdg2
# mdadm /dev/md2 -a /dev/hdg3

```

```

#cat /proc/mdstat

```

```

Personalities : [raid1]
read_ahead 1024 sectors
md1 : active raid1 ide/host2/bus1/target0/lun0/part2[0:979840 blocks [2/2] [UU]
md0 : active raid1 ide/host2/bus0/target0/lun0/part1[0:192640 blocks [2/2] [UU]
md2 : active raid1 ide/host2/bus1/target0/lun0/part3[0:116045440 blocks [2/1] [_U]
[====>.....] recovery = 23.9% (18112/72192)

```

```
unused devices: <none>
```

After a while all devices are in sync:

```
# cat /proc/mdstat

Personalities : [raid1]
read_ahead 1024 sectors
md1 : active raid1 ide/host2/bus1/target0/lun0/part2[0]
      979840 blocks [2/2] [UU]
md0 : active raid1 ide/host2/bus0/target0/lun0/part1[0]
      192640 blocks [2/2] [UU]
md2 : active raid1 ide/host2/bus1/target0/lun0/part3[0]
      116045440 blocks [2/1] [UU]
unused devices: <none>
```

## Regenerate initrd

Yes, this needs to happen again after the raid is complete!

```
#mkinitrd -o /boot/initrd.img-2.4.26 2.4.26
```

## Configure grub to boot from both drives

Install the MBR on the second disk

```
#grub --device-map=/boot/grub/device.map

>> root (hd0,0)
>> setup (hd0)
>> root (hd1,0)
>> setup (hd1)
>> quit
```

Now copy the grub stanzas in /boot/grub/menu.lst that we booted off of and edit the new stanzas noting the changes in bold:

```
title Debian GNU/Linux, kernel 2.4.26 RAID boot disk2
root (hd1,0)
kernel /vmlinuz-2.4.26 root=/dev/md2 ro
initrd /initrd.img-2.4.26
savedefault
boot
```

## Producing /etc/mdadm/mdadm.conf

I think this is mostly useful in recovery ?? Email me if you know.

```
edit /etc/mdadm/mdadm.conf
```

The top line in this example should be

```
DEVICE /dev/hdg* /dev/hde*
```

Then run:

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

This should produce a proper file -- but check:

edit the generated /mnt/etc/mdadm/mdadm.conf:

- put the devices= lines to the top of the file
- change "devices" to "DEVICE"
- replace "=" with <space> on the DEVICE lines

## Setup the monitoring daemon

Just run:

```
dpkg-reconfigure mdadm
```

## Test Test and Test

Test boot from both drives

Kill a drive and see if you get a email about the event.

Write up a step by step procedure to restore from a drive outage. (send a copy this way for this page!)

You should be all finished!

Please send notes of any typos/corrections to the email address below.

Special thanks to Onni Koskinen of Finland, whose gentle yet expert emails removed several glaring errors on this page and resulted in a vastly improved document.


## Q & As

**Re: section on configuring GRUB: will GRUB automatically boot from the good drive in the event of a disk failure?**

Yes, IF you install Grub on both drives and your BIOS will roll over to the first bootable drive.

How do you see what is in a initrd file?

```
mount -o loop /tmp/myinitrd /mnt/myinitrd
```

	3209 W.9th street Lawrence, KS 66049 USA	Ph FAX Email WEB	(785) 841 3089 (785) 841 0434 <a href="mailto:inform@xtronics.com">inform@xtronics.com</a> <a href="http://xtronics.com">http://xtronics.com</a>	<b>Bookmark this page</b>
-------------------------------------------------------------------------------------	------------------------------------------------------	---------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------

Boolean:  Case 

<a href="#">Transtronics Home Page</a> See our line of industrial control electronics	<a href="#">PLC's Index</a>	<a href="#">PC test equipment and EPROM programmer</a>	<a href="#">Process Control Panel meters</a>	<a href="#">BVUPS</a>	<a href="#">Resource library</a> handbooks, primers and spec sheets
------------------------------------------------------------------------------------------	-----------------------------	--------------------------------------------------------	----------------------------------------------	-----------------------	------------------------------------------------------------------------

[Corporate information and privacy statement](#)

(C) Copyright 1994-2005, Transtronics, Inc. All rights reserved  
Transtronics® is a registered trademark of Transtronics, Inc.