

# Configure c't-Debian-Server For Software-RAID-1

Andreas Schamanek

as2005@fam.tuwien.ac.at

## Revision History

Revision 1.00 2005-09-03 Revised by: (as)

The following article describes step by step how to set up a c't-Debian-Server (<http://www.heise.de/ct/ftp/projekte/srv/>) running on and booting from a Software-RAID-1 (<http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>) root filesystem using 2 mirrored disks.

HTML and PDF versions are available at <http://www.fam.tuwien.ac.at/~schamane/sysadmin/ctsrvraid/>

## 1. Introduction And Motivation

Please, read the Section 16 first (see end of this document). If you have already modified your system settings, or if you are storing valuable data, you definitely should not proceed without a functioning backup. You've been warned.

If you are not familiar with Linux, RAID and system administration, you are heading for trouble if you just follow this guide step by step. You've been warned again.

Our intention is *not* to build a fail-safe system nor a HA-Linux (<http://www.linux-ha.org/>). We are just using RAID-1 mirrored disks to make sure that the system *can* be restored easily if 1 disk crashes.

The following has been tested only twice (hey, we've got other things to do). It did work for us, it might work for you. It might also make your rats go berserk or just blow your mind.

## 2. Assumptions

The system is equipped with 2 similar hard disks (they do not have to be the same but it helps if they are at least similar). The first disk is attached to the primary IDE as master, it will be referenced as

`/dev/hda`. The second disk is attached to the secondary IDE also as master, it will be referenced as `/dev/hdc`.

We have installed a c't-Debian-Server (<http://www.heise.de/ct/ftp/projekte/srv/>) using the **auto** installation (we never managed to use a system installed by any of the other installation methods; see also c't-Debian-Server — soweit bekannte Probleme (<http://www.heise.de/ct/ftp/projekte/srv/buglist.shtml>)). The system is installed on `/dev/hda`. We had chosen "Gesamtes Laufwerk loeschen: IDE1 Master (hda)".

You do know what a Software-RAID (<http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>) is.

Lines of code starting with `$` are commands to be entered as user **root** (without the `$`). Lines starting with `#` are comments.

Note: Some commands could be written much shorter. Sometimes, we have chosen long versions for readability and safety.

### 3. The Plan

The **auto** installation has created 5 partitions on disk `/dev/hda`. On a reasonable big disk the partitions are e.g.

partition	size	type	mounted on
<code>/dev/hda1</code>	6.5 GB	83	<code>/</code>
<code>/dev/hda5</code>	400 MB	82	Linux swap
<code>/dev/hda6</code>	8 GB	83	<code>/home</code>
<code>/dev/hda7</code>	8 GB	83	<code>/var</code>
<code>/dev/hda8</code>	remaining	83	<code>/srv</code>

We will copy this layout to `/dev/hdc`. We will then create 4 RAID-1 arrays according to

RAID array	size	type	mounted on	built of
<code>/dev/md1</code>	6.5 GB	fd	<code>/</code>	<code>hda1 + hdc1</code>
<code>/dev/md6</code>	8 GB	fd	<code>/home</code>	<code>hda6 + hdc6</code>
<code>/dev/md7</code>	8 GB	fd	<code>/var</code>	<code>hda7 + hdc7</code>
<code>/dev/md8</code>	remaining	fd	<code>/srv</code>	<code>hda8 + hdc8</code>

Furthermore, we will continue to use `/dev/hda5` as Linux swap (un-RAID-ed). `/dev/hdc5` will be mounted to `/tmp`. It's OK if you do not want to do this but you will waste 400 MB disk space on `/dev/hdc`.

## 4. Preliminaries

### 4.1. Install Needed Software

```
$ apt-get install hdparm mdadm
```

You may need to confirm some information. Answer *Ja* for *Moechten Sie die RAID Laufwerke automatisch starten?* and for *Moechten Sie den RAID-Ueberwachungsdaemon starten?*.

### 4.2. Check That DMA Is Enabled For Your Disks

```
$ hdparm /dev/hda /dev/hdc
```

For each disk this should at least show

```
using_dma    = 1 (on)
```

The system will run fine if it does not. But, disk operations will be considerably slower.

## 5. Setup Kernel And Disk /dev/hdc

```
# add needed modules to /etc/modules
$ echo md >>/etc/modules
$ echo raid1 >>/etc/modules

# load RAID modules into kernel (_md_ might already exist)
# (use _modprobe_ instead of _insmod_ if _insmod_ does not work)
$ insmod md
$ insmod raid1
```

After the modules have been loaded, there must be a file `/proc/mdstat`. List its contents with

```
$ cat /proc/mdstat
```

This should look similar if not equal to

```
Personalities : [raid1]
read_ahead not set
unused devices: <none>
```

If `/proc/mdstat` does not exist, I am afraid you are reading the wrong document :-/ Otherwise, we are ready for action:

```
# copy the disk layout from /dev/hda to /dev/hdc
# this will permanently erase all data on /dev/hdc
$ sfdisk -d /dev/hda | sfdisk /dev/hdc
```

## Caution

If the two disks are not of the same type `sfdisk` might complain that it "*does not like the partitions*". If this is the case, you can create them yourself with

```
$ cfdisk /dev/hdc
```

(have a look first on the partitions on `/dev/hda`) or you can force `sfdisk` to do it anyway with

```
$ sfdisk -d /dev/hda | sfdisk /dev/hdc --force

# change partition types (IDs) on /dev/hdc
# fd ... Linux raid autodetect, 83 ... Linux
$ for v in 1 6 7 8 ; do sfdisk --change-id /dev/hdc $v fd ; done
$ sfdisk --change-id /dev/hdc 5 83
```

If you have forced `sfdisk` to write the partitions to `/dev/hdc` you are likely to see a few warnings such as

```
Warning: extended partition does not start at a cylinder boundary.
```

## 6. Create RAID-1 Arrays

```
# permanently erase data on /dev/hdc1
# you can skip this if you know that /dev/hdc1 is empty
$ dd if=/dev/zero of=/dev/hdc1 bs=1024 count=1024
$ mdadm --zero-superblock /dev/hdc1

# create RAID-1 arrays
$ mdadm --create /dev/md1 --level=1 --raid-disks=2 missing /dev/hdc1
$ mdadm --create /dev/md6 --level=1 --raid-disks=2 missing /dev/hdc6
$ mdadm --create /dev/md7 --level=1 --raid-disks=2 missing /dev/hdc7
$ mdadm --create /dev/md8 --level=1 --raid-disks=2 missing /dev/hdc8
```

Every `mdadm --create` should at least finish with a line like

```
mdadm: array /dev/md1 started.
```

```
# create filesystems on RAID arrays and on /dev/hdc5
$ for v in 1 6 7 8 ; do mkfs.ext3 /dev/md$v ; done
$ mkfs.ext3 /dev/hdc5

# add information about arrays to configuration file
```

```
$ cd /etc/mdadm
$ cp mdadm.conf mdadm.conf.o
$ mdadm --detail --scan >> mdadm.conf
```

## 7. Edit /etc/fstab

We have to change /etc/fstab to make it mount the RAID arrays. You can do the editing with `sed`:

```
# change to directory /etc
$ cd /etc

# create a backup of fstab
$ cp fstab fstab.o

# create a new file with exchanged devices
$ sed -e 's|/hda\([1678]\)|/md\1|' < fstab.o > fstab

# add a line to mount /dev/hdc5 as /tmp
$ echo "/dev/hdc5 /tmp ext3 defaults 0 0" >> fstab
```

## 8. Create A New initrd.img

The kernel of a c't-Debian-Server fortunately already uses a ramdisk image (`initrd.img`). The following steps will create a new `initrd.img`.

```
# change to the directory with all settings for _mkinitrd_
$ cd /etc/mkinitrd

# add modules for RAID to /etc/mkinitrd/modules
$ echo md >>modules
$ echo raid1 >>modules

# make a backup of /etc/mkinitrd/mkinitrd.conf and create a new one
$ cp mkinitrd.conf mkinitrd.conf.o
$ sed -e 's|^ROOT=.*|ROOT=/dev/md1|' < mkinitrd.conf.o > mkinitrd.conf
```

Instead of using `sed` you could just edit file `mkinitrd.conf` e.g. with

```
$ nano -w mkinitrd.conf
```

and change the line reading `ROOT=probe` to

```
ROOT=/dev/md1
```

After changing `mkinitrd.conf` we can create a new ramdisk image with

```
$ mkinitrd -o /boot/initrd.img-2.4.27-ct-1-raid
$ cd /
$ rm initrd.img
$ ln -s boot/initrd.img-2.4.27-ct-1-raid initrd.img
```

On a computer with a Celeron 300 CPU `mkinitrd` did take some time. So, just wait patiently and get yourself a fresh cup of espresso.

## 9. Setting Up The Boot-Manager (Step 1)

We have to edit `/boot/grub/menu.lst`

```
$ nano -w /boot/grub/menu.lst
```

Find the following paragraph near the end of the file

```
title          Debian GNU/Linux, kernel 2.4.27-ct-1
root           (hd0,0)
kernel         /boot/vmlinuz-2.4.27-ct-1 root=/dev/hda1 ro
initrd         /boot/initrd.img-2.4.27-ct-1
savedefault
boot
```

Insert a copy of this paragraph just above it and change it to/make it

```
title          Debian GNU/Linux, kernel 2.4.27-ct-1 RAID
root           (hd0,0)
kernel         /boot/vmlinuz-2.4.27-ct-1 root=/dev/md1 ro
initrd         /boot/initrd.img-2.4.27-ct-1-raid
savedefault
boot
```

To create a copy of a paragraph with the `nano` editor you can move e.g. the cursor to `title`, press `^K 7` times to cut 7 lines (6 lines of text + 1 empty line) into the buffer, then press 2 times `^U` to insert the buffer 2 times. Move to the first copy of the paragraphs and change the first one to the text shown above. `^X` quits and saves `nano`.

Alternatively, you can download the whole file from our website (<http://wox.at/as> is a short URL for <http://www.fam.tuwien.ac.at/~schamane>). This, of course, will overwrite any previous settings with some basic defaults.

```
$ cd /boot/grub
$ mv menu.lst menu.lst.o
$ wget http://wox.at/as/sysadmin/ctsrvraid/menu.lst
```

## 10. Copy Data From /dev/hda To The RAID Arrays

```
# change to single user mode (wait a few seconds ;)
$ init 1

# copy data of /dev/hda1 (i.e. / "root") to /dev/md1
$ mount /dev/md1 /mnt
# be especially careful with the following line
$ tar clf - --exclude '/tmp/*' / | ( cd /mnt ; tar xvpf - --same-owner)

# recreate /tmp on /dev/md1 and set permissions for /dev/hdc5
$ cd /mnt
$ test -d /tmp || mkdir /tmp
$ mount /dev/hdc5 tmp
$ chmod 1777 tmp
$ cd /
$ umount /mnt/tmp
$ umount /mnt

# copy data of /dev/hda6 (/home) to /dev/md6
$ mount /dev/md6 /mnt
$ cd /home
$ tar clf - . | ( cd /mnt ; tar xvpf - --same-owner)
$ umount /mnt

# copy data of /dev/hda7 (/var) to /dev/md7
$ mount /dev/md7 /mnt
$ cd /var
$ tar clf - . | ( cd /mnt ; tar xvpf - --same-owner)
$ umount /mnt
```

/srv should be empty anyway. Otherwise, follow the same pattern as above.

## 11. Setting Up The Boot-Manager (Step 2)

After editing or downloading /boot/grub/menu.lst we set up **grub**. (Note that starting **grub** takes some time.)

```
$ grub
```

When **grub** has started it will wait for commands with

```
grub>
```

Enter

```
device (hd0) /dev/hdc
root (hd0,0)
setup (hd0)
```

```
quit
```

See also <http://www.tldp.org/HOWTO/Software-RAID-HOWTO-7.html#ss7.3>

## 12. Reboot And Verify That RAID Devices Are Operating

At this point the RAID arrays consist only of partitions of disk `/dev/hdc`. Yet, if we reboot the system it shall mount the RAID arrays. This is also needed to free access to `/dev/hda`.

```
$ reboot
```

After the machine is up again login and look at `/proc/mdstat`.

```
$ cat /proc/mdstat
```

It should look like

```
Personalities : [raid1]
read_ahead 1024 sectors
md1 : active raid1 ide/host0/bus1/target0/lun0/part1[1]
      8000256 blocks [2/1] [_U]
...
md6 : active raid1 ide/host0/bus1/target0/lun0/part6[1]
      8000256 blocks [2/1] [_U]
...
```

The list of mounted filesystems can be shown with

```
$ mount
```

This should give

```
/dev/md1 on / type ext3 (rw,errors=remount-ro)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/md8 on /srv type ext3 (rw)
/dev/md7 on /var type ext3 (rw)
/dev/md6 on /home type ext3 (rw)
/dev/hdc5 on /tmp type ext3 (rw)
usbfs on /proc/bus/usb type usbfs (rw)
```

If your machine did not boot up normally, I am afraid, **you've got a problem** :- ( You *did* read the Section 16, didn't you? Start over and have a look at Further Reading.

## 13. Finish The RAID Arrays And Wait For The Sync

We are now going to add partitions hda1 to hda8 to the RAID arrays. This will immediately and permanently erase all data on these partitions.

```
# change to single user mode (gotta wait a bit again)
$ init 1

# change partition types (IDs) on /dev/hda
$ for v in 1 6 7 8 ; do sfdisk --change-id /dev/hda $v fd ; done

# add partitions to arrays
$ mdadm --add /dev/md1 /dev/hda1
$ mdadm --add /dev/md6 /dev/hda6
$ mdadm --add /dev/md7 /dev/hda7
$ mdadm --add /dev/md8 /dev/hda8
```

Every `mdadm --add` should finish with a line like

```
mdadm: hot added /dev/hda6
```

After hot-adding the partitions, Linux starts to sync data from `/dev/hdc` to `/dev/hda`. You can monitor this process with

```
$ watch -n 10 cat /proc/mdstat
```

Syncing may take from 30 minutes to a few hours depending on size of disks and speed of system. You should wait until the sync is done. Plenty of time for more espressos :-)

You can quit `watch` by pressing `^C`. Actually, you will have to quit and restart it in between because the screen will get mixed up whenever one of the arrays finishes the sync.

During the syncs your screen should show lines such as for `/dev/md1`

```
Personalities : [raid1]
read_ahead 1024 sectors
md1 : active raid1 ide/host0/bus0/target0/lun0/part1[2]
      ide/host0/bus1/target0/lun0/part1[1]
      8000256 blocks [2/1] [_U]
      [====>.....] recovery = 28.7% (2302528/8000256)
finish=8.4min speed=11181K/sec
...
```

Finally, when the sync has finished for all arrays, we write a new configuration file:

```
$ cd /etc/mdadm
$ cp mdadm.conf mdadm.conf.O
$ echo "DEVICE partitions" > mdadm.conf
```

```
$ mdadm --detail --scan >> mdadm.conf
```

And for some reason we have to re-make the `initrd.img`:

```
$ mkinitrd -o /boot/initrd.img-2.4.27-ct-1-raid
```

Then you might want to reboot your system to verify things.

```
$ reboot
```

When the system has booted

```
$ mount
```

should give the same results as above.

```
$ cat /proc/mdstat
```

should look like

```
Personalities : [raid1]
read_ahead 1024 sectors
md8 : active raid1 ide/host0/bus0/target0/lun0/part8[0]
      ide/host0/bus1/target0/lun0/part8[1]
      139909952 blocks [2/2] [UU]
...
md1 : active raid1 ide/host0/bus0/target0/lun0/part1[0]
      ide/host0/bus1/target0/lun0/part1[1]
      8000256 blocks [2/2] [UU]
...
unused devices: <none>
```

That's it.

## 14. Documenting Your RAID Arrays

The following command will print a detailed list of information about your arrays.

```
$ mdadm -D /dev/md1 /dev/md6 /dev/md7 /dev/md8
```

Any feedback (<mailto:as2005@fam.tuwien.ac.at>) is welcome.

## **15. Further Reading**

c't-Debian-Server <http://www.heise.de/ct/ftp/projekte/srv/>

The Software-RAID HOWTO <http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html>

Debian Software Root Raid Documentation <http://alioth.debian.org/projects/rootraidoc>

ctserver.org: Das Forum zum c't Debian-Server <http://ctserver.org/>

## **16. Disclaimer**

The information here is provided "as is", without warranty of any kind expressed or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall the copyright holders be liable for any claim, damages or other liability, whether in action of contract, tort or otherwise, arising from, out of or in connection with the information or the use or other dealings in the information here.