Diese Anleitung erklärt Schritt für Schritt wie man ein Software-RAID 0, 1, 5, 6 oder 10 unter Ubuntu/Debian/Linux mittels mdadm nachträglich einrichtet und wartet.

Die hier vorgestellte Lösung ist für alle Speichermedien wie Festplatten, SSDs, USB-Sticks, USB-Festplatten, externe Festplatten, Flash-Speicherkarten, usw. geeignet.
Ein gewisses Basiswissen insbesondere über die entsprechenden RAID-Level wird hierbei vorausgesetzt.

Nr.Software-RAID Handbuch:

1 Laufwerke & Anschlüsse kennzeichnen
2 mdadm Tools installieren
3 Laufwerke partitionieren
4 RAID anlegen
5 Buildstatus überwachen
6 RAID-Konfiguration speichern
7 RAID Alignment berechnen & Formatierung
8 RAID-Verbund einhängen/mounten
9 Hot-Spare-Laufwerk hinzufügen
10 E-Mail Benachrichtigung bei Ausfall
11 RAID-Verbund & Partitionen anzeigen
12 RAID-Verbund wieder aktivieren (Laufwerksdefekt)
13 RAID-Laufwerk ersetzen (bei defekt)
14 RAID wieder einbinden (nach Systemwechsel)
15 RAID-Verbund im Live-System bereitstellen
16 RAID 5/6 vergrößern (mit weiteren Laufwerken)
17 RAID 0 vergrößern (mit weiteren Laufwerken)
18 RAID Laufwerke durch größere ersetzen
19 RAID 5 in RAID 6 konvertieren
20 RAID Integritätstest (checkarray)
21 RAID-Verbund komplett auflösen
22 RAID-Benchmark
23 RAID Tuning Tipps
24 Logfile überprüfen & Fehlerbehebung
25 ata /dev/sd zuordnen
26 SMART-Werte eines Laufwerks überprüfen
27 Laufwerk austauschen (bei SMART-Fehler/Speicherplatz erhöhen)
28 Laufwerktest mit smartctl
29 Laufwerktest mit badblocks

1a. Laufwerke & Anschlüsse kennzeichnen:

Zunächst sollte man die Speichermedien bzw. die Anschlüsse kennzeichnen, so dass man diese bei einem Ausfall leichter wechseln kann.
Am einfachsten lässt man sich zu allen Laufwerken die Seriennummer anzeigen und notiert sich die Zuordnung:
Laufwerk Seriennummer Anzeigen:
hdparm -i /dev/sdX | grep Serial

Oder über die smartmontools - diese muss man zuerst installieren:
apt-get install smartmontools
Danach kann man sich die Laufwerksdaten wie folgt anzeigen lassen:
smartctl -i /dev/sdX

Beispielausgabe:
root@NAS:~# hdparm -i /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf | grep Serial
Model=WDC WD30EFRX-68EUZN0, FwRev=82.00A82, SerialNo=WD-WCC4N3USVJCY
Model=WDC WD30EFRX-68EUZN0, FwRev=82.00A82, SerialNo=WD-WCC4N6THYCA6
Model=WDC WD30EFRX-68EUZN0, FwRev=82.00A82, SerialNo=WD-WCC4N3USVH6P
Model=WDC WD30EFRX-68EUZN0, FwRev=82.00A82, SerialNo=WD-WCC4N3USVYLH
Model=WDC WD30EFRX-68EUZN0, FwRev=82.00A82, SerialNo=WD-WCC4N5HP8KEZ

1b. Die UUID mit blkid ermitteln und Laufwerke & Anschlüsse kennzeichnen:

Wichtig:
Man sollte beachten, dass sich die Reihenfolge der Laufwerke bei einem Ausfall, beim Vertauschen von Kabeln, beim Auswechseln vom Mainboards, bzw. beim Controller-Austausch usw. verändern kann.
Man sollte sich daher zusätzlich noch die Zuordnung der UUID (universally unique identifier) zu allen Laufwerken bzw. Dateisystemen anzeigen und notieren.
Beachtet dabei bitte in diesem Zusammenhang die UUIDs von blkid werden nur für bereits angelegte Partitionen ausgegeben.
Führt diesen Punkt daher erst aus nachdem ihr die Partitionen und den RAID-Verbund fertig angelegt habt, denn erst dann werden auch die RAID-spezifischen Merkmale mit ausgegeben.
Den Punkt könnt ihr daher beim ersten Mal überspringen.

Wichtig:
blkid benötig Rootrechte.
blkid versucht immer die aktuellen Daten abzufragen, die so ermittelten Daten werden in einem Cache abgespeichert.
Startet man blkid als Standarduser, so hält man ggf. nur alte Informationen aus dem Cache.

blkid -g # Einen vorhandenen UUID-Informations-Cache löschen (ggf. bei

Laufwerkswechsel empfohlen).
blkid # Gerätezuordnungen: Laufwerk & UUIDs anzeigen.
blkid -c /dev/null # Aktuelle Zuordnung der Laufwerke & UUIDs (universally unique identifier) anzeigen, ein ggf. vorhandener Cache wird ignoriert (empfohlene Variante).

Beispielausgabe:

- System- und Swap-Laufwerk ist hier [/dev/sdax].
- [/dev/sdb1] bis [/dev/sdf1] ergeben hier den RAID-Verbund [/dev/md0].
- [UUID] kennzeichnet bei den Laufwerken [/dev/sdb1] bis [/dev/sdf1] die ID des zugehörigen RAID Systems.
- [UUID-SUB] (UUID subystem) ist die eindeutige physische Laufwerks ID.
- [PARTUUID] kennzeichnet die ID der GPT-Partition (diese ändert sich ggf. beim Formatieren oder Dateisystemwechsel).

raiduser@NAS:~$ blkid
/dev/sda1: UUID="6da4eaa7-37ad-4501-8e02-efc56208c4f6" TYPE="ext4" PARTUUID="4c52e762-01"
/dev/sda5: UUID="4860814b-647b-4f13-a05d-e81dd500441e" TYPE="swap" PARTUUID="4c52e762-05"
/dev/sdb1: UUID="6842743a-3a0c-1b11-f11c-14df3a13f246" UUID_SUB="58630813-0a10-a31f-0ba7-c37e63e1153f" LABEL="NAS:0" TYPE="linux_raid_member" PARTLABEL="primary" PARTUUID="ffa7aa26-645b-4083-91de-ae0cd01eabaa"
/dev/sdc1: UUID="6842743a-3a0c-1b11-f11c-14df3a13f246" UUID_SUB="e91ff19b-5f82-7f41-93db-7bf4e047d26a" LABEL="NAS:0" TYPE="linux_raid_member" PARTLABEL="primary" PARTUUID="0f5663a9-b1a7-4624-9048-9d1adb01cd3b"
/dev/sdd1: UUID="6842743a-3a0c-1b11-f11c-14df3a13f246" UUID_SUB="ea0c21e3-78c1-6988-1545-743ef36b69ba" LABEL="NAS:0" TYPE="linux_raid_member" PARTLABEL="primary" PARTUUID="12c6b111-b064-4680-bf30-24ab999cada0"
/dev/sde1: UUID="6842743a-3a0c-1b11-f11c-14df3a13f246" UUID_SUB="f331a9ab-9e34-ea56-c53f-4165fb8ce2ec" LABEL="NAS:0" TYPE="linux_raid_member" PARTLABEL="primary" PARTUUID="0e1e5b7e-6072-4fb5-8a38-033a0c35d04f"
/dev/sdf1: UUID="6842743a-3a0c-1b11-f11c-14df3a13f246" UUID_SUB="76422459-339e-dabf-7bb6-f502f61557f3" LABEL="NAS:0" TYPE="linux_raid_member" PARTLABEL="primary" PARTUUID="adaeb8b5-7eb3-4c58-918a-3fa64cb57a2a"
/dev/md0: UUID="312f727f-5333-442e-99c9-85825caa53d6" TYPE="ext4"

Anschließend schaltet man das System aus und kennzeichnet die Speichermedien bzw. die Anschlüsse z. B. mit Aufklebern mit [/dev/sda] bzw. mit der [UUID].
Hat man sich die die Laufwerk-Geräte-Namen [/dev/sdX], die [Seriennummer] und die [UUID]s notiert, dann ist später ein Wechsel eines defekten Laufwerks relativ unproblematisch.

2. mdadm installieren:

Ein RAID-Verbund aus mehreren Laufwerken unterstützt der Linux Kernel automatisch.
Zur Einrichtung und Wartung wird das Paket (Programm) mdadm verwendet.

Hinweis:

Bei der Installation von mdadm wird gleichzeitig postfix (ein mail transfer agent) eingerichtet, sofern man postfix nicht bereits konfiguriert hat.
Postfix wird benötigt wenn man z. B. E-Mails über einen Laufwerksausfall erhalten möchte.
Empfehlenswert finde ich wenn man zuerst postfix einrichtet (hier geht's zur Anleitung) und erst dann das RAID-System aufsetzt.

Die notwendigen Pakete zum anlegen des RAID-Verbundes installieren:
Installation mit postfix:
apt-get install mdadm

Installation ohne postfix (mail transfer agent):
apt-get install --no-install-recommends mdadm

3. Laufwerke partitionieren:

Bevor man Laufwerke in einem RAID-System verwenden kann muss man diese partitionieren.
Wichtig: Die identische Partitionierung muss man ggf. auch für Ersatzlaufwerke, Erweiterungslaufwerke und Hot-Spare-Laufwerke vornehmen - man sollte sich das Partitionsschema also am besten extra notieren.

Eine neue leere Partitionstabelle anlegen:
parted /dev/sdX mklabel gpt

Eine Partition für den RAID-Verbund anlegen:
Für Tests empfiehlt es sich zuerst eine kleine Partition zu verwenden, 1% ist dabei völlig ausreichend, setzt man 100% ein so wird das ganze Laufwerk verwendet:
parted -a optimal -- /dev/sdX mkpart primary 2048s 1%

Es empfiehlt sich am Ende etwas Reserve zu lassen, um für Ausfälle gewappnet zu sein (8192 Sektoren).
Dies ermöglicht später ggf. die Verwendung eines alternativen Laufwerks mit weniger Sektoren.
parted -a optimal -- /dev/sdX mkpart primary 2048s -8192s
parted -a optimal -- /dev/sdX mkpart primary 2048s 98% # Empfohlen für SSDs (TRIM-Reserve).
Am Anfang des Laufwerks sollte man 2048 Sektoren auslassen, um das Alignment der ersten Partition exakt auf 1 MB zu setzen.
Dieses Alignment wird benötigt, wenn man die optimale Leistung erreichen möchte.
2048 Sektoren * 512 Bytes (ein Sektor) = 1048576 Bytes = 1024 KB = ergeben exakt 1 MB.

Den Typ der ersten Partition auf RAID ändern:
parted /dev/sdX set 1 raid on

4. RAID anlegen:

Beim Erstellen eines neuen RAID-Verbundes wird ein neues blockorientiertes Gerät mit einem neuen Namen z. B. [/dev/md0] erzeugt.
Gleichzeitig wird bei der Einrichtung auf allen betroffenen Partitionen im RAID-Verbund ein jeweiliger Superblock gespeichert.
Dieser Superblock enthält alle grundlegenden Informationen zum RAID-Verbund, unter anderem: die Aufgabe, die Position des Laufwerks im Verbund sowie den RAID-Level.
Mit Hilfe dieser Superblock Informationen kann man den RAID-Verbund jederzeit ansprechen - das klappt dann sogar von einem Live-System aus.
Wichtig: Den Superblock vom RAID-Verbund sollte man nicht mit den Superblöcken vom Dateisystem (z. B. ext4) verwechseln.

Zum Erstellen muss man nun das entsprechende Level auswählen und die entsprechenden Laufwerke mit Partition angeben:
Für **RAID 0** (striping) **keine Sicherheit**, mindestens 2 HDs:
mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1

Für **RAID 1** (mirroring/Spiegelung) **einfache Sicherheit**, mindestens 2 HDs:
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1

Für **RAID 5** (block-level striping mit Paritätsinformation) **einfache Sicherheit**, mindestens 3 HDs:
mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1

Für **RAID 6** (block-level striping mit doppelter verteilter Paritätsinformation) **hohe Sicherheit**, mindestens 4 HDs (unsere Empfehlung♥):
mdadm --create /dev/md0 --level=6 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
mdadm --create /dev/md0 --level=6 --raid-devices=5 /dev/sd[bcdef]1

Für **RAID 10** (kombiniertes stripping & mirroring/Spiegelung) **mittlere Sicherheit**, mindestens 4 HDs (Geschwindigkeitsempfehlung🚀):
mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1

Bei noch fehlenden Laufwerken (z. B. bei Umstellungen) **anfangs keine Sicherheit**:
mdadm --create /dev/md0 --level=X --raid-devices=X /dev/sdb1 missing
Bei [--raid-devices] gibt man die Anzahl der zu verwendenden Laufwerke (Daten + Parity) an, Hot-Spare-Laufwerke lässt man hier weg.

Die Frage [continue creating array?] mit y für yes bestätigen.

optionale Parameter:
[--chunk=64] = die Angabe einer alternativen Chunkgröße in KB (ist bei neuen Systemen nicht mehr nötig).
[--verbose] = ausführliche Anzeige.

Superblock Version festlegen:
[--metadata=0.9] = Superblock im alten Format am Ende speichern - maximal 28 Laufwerke - maximal nur 2 TB - nur für alte Systeme (nicht mehr empfohlen).

[--metadata=1.0] = Superblock am Ende speichern - dies erschwert das erweitern/vergrößern eines RAID-Verbundes beim Laufwerkstausch (daher nicht empfohlen).
[--metadata=1.1] = Superblock am direkt am Anfang speichern (nicht empfohlen).
[--metadata=1.2] = Superblock am Anfang nach 4K speichern (das ist Standard und sollte so belassen werden - den Parameter muss man daher nicht mit angeben).
Eine andere Version des Superblocks muss man also nur bei älteren Systemen angeben.

5. Buildstatus überwachen:

Den aktuellen Stand anzeigen:
cat /proc/mdstat

Oder die dauerhafte Überwachung des Status (Aktualisierung alle X Sekunden, Abbruch der Anzeige mit Strg + c):
watch -n1 cat /proc/mdstat

Während der rebuild läuft kann man das System jederzeit beenden und neu starten. Der rebuild wird bei einem Neustart des Systems automatisch fortgeführt, bis dieser abgeschlossen ist.

6. RAID-Konfiguration speichern:

mdadm speichert bei einem neu aufgesetzten RAID-Verbund alle wichtigen Informationen zu dem RAID-Verbund direkt auf den jeweiligen Partitionen in einem so genannten Superblock. Die dort gespeicherten Informationen sind somit Systemunabhängig.
Trotzdem muss man beim erstellen/ändern/löschen eines RAID-Verbundes beachten, dass mdadm die lokale Konfigurations-Datei nicht selbstständig sichert bzw. aktualisiert.
Speichert man die Konfigurations-Datei nicht, so kann es unter anderem passieren das, der RAID-Verbund nach einem Neustart falsch als md127, md126, usw. ... eingebunden wird.
Man sollte die Konfiguration daher wie folgt sichern:
mdadm --examine --scan --verbose >> /etc/mdadm/mdadm.conf
Speichert die Konfiguration (die mdadm-Ausgabe wird am Ende angehängt).

update-initramfs -u -k all
Sorgt dafür das der Verbund nach dem Systemstart entsprechend der Angabe in der Konfiguration verfügbar ist.
Bzw. sorgt dafür das der Verbund wieder als md0 angesprochen wird, falls zuvor z. B. md127 vergeben war.
[-u] = (update) der Startumgebung.
[-k all] = (kernel all) RAID-Verbund für alle - auch ältere Kernelversionen bereitstellen.

nano /etc/mdadm/mdadm.conf
Ggf. die manuelle Bearbeitung der Konfiguration (bei Änderungen).

Wichtig:
Ändert man die Reihenfolge der Laufwerke, z. B. bei Austausch des Mainboards, Controllers

bzw. durch vertauschen der Anschlüsse, so kann es passieren das, der RAID-Verbund nicht mehr automatisch erkannt wird, obwohl nach wie vor in den Superblöcken der jeweiligen Partitionen die richtigen Informationen gespeichert sind. Das liegt daran, da die Konfiguration unter [/etc/mdadm/mdadm.conf] vorrangig vor den Information in den Superblöcken berücksichtigt wird.
Sollte dieses Problem auftreten, so muss man die Konfiguration entsprechend bearbeiten und neu speichern.

mdadm.conf Beispiel mit einem RAID-Verbund /dev/md0 (siehe die letzten 2 Zeilen):

```
root@NAS:~# cat /etc/mdadm/mdadm.conf
# mdadm.conf
#
# Please refer to mdadm.conf(5) for information about this file.
#

# by default (built-in), scan all partitions (/proc/partitions) and all
# containers for MD superblocks. alternatively, specify devices to scan, using
# wildcards if desired.
#DEVICE partitions containers

# auto-create devices with Debian standard permissions
CREATE owner=root group=disk mode=0660 auto=yes

# automatically tag new arrays as belonging to the local system
HOMEHOST <system>

# instruct the monitoring daemon where to send mail alerts
MAILADDR raid@ctaas.com

# definitions of existing MD arrays

# This file was auto-generated on Wed, 06 Jan 2016 10:50:22 +0100
# by mkconf $Id$
ARRAY /dev/md/0 level=raid6 metadata=1.2 num-devices=5
UUID=6842743a:3a0c1b11:f11c14df:3a13f246 name=NAS:0
  devices=/dev/sdf1,/dev/sde1,/dev/sdd1,/dev/sdc1,/dev/sdb1
```

7. RAID Alignment & Formatierung:

Alle Daten werden im RAID-Verbund in Blöcken - so genannten Chunks - gleichmäßig über alle Laufwerke verteilt gespeichert. Ausgenommen sind hier ggf. vorhandene Parity-Laufwerke und Hot-Spare-Laufwerke.
Um nun die optimale Leistung zu erreichen sollte man bei komplexen Verbünden wie RAID 0, 5, 6 oder 10 die Alignment-Werte entsprechend berechnen.

Hierzu kann man den folgenden Kalkulator verwenden: mkfs stride calculator im Web
Man kann das optimale Alignment aber auch leicht selbst berechnen, eine Erklärung hierzu folgt gleich.

Chunk Größe ermitteln:
Chunk Größe des RAID-Verbundes anzeigen:
mdadm --detail /dev/md0 | grep "Chunk Size"

Anzeige der Sektorgröße des Laufwerks:
smartctl -i /dev/sdX | grep "Sector Sizes"

Raid-Laufwerk Formatieren:
mit **ext4** Dateisystem:
mkfs.ext4 -v -m 1 -b 4096 -E stride=128,stripe-width=256 /dev/md0 # z. B. bei RAID 5 mit 3 Laufwerken (2 Datendisks + 1 Parity)
mkfs.ext4 -v -m .5 -b 4096 -E stride=128,stripe-width=384,lazy_itable_init=0,lazy_journal_init=0 -L "RAID-Daten" /dev/md0 # z. B. bei RAID 6 mit 5 Laufwerken (3 Datendisks + 2 Parity)
#
Parametererklärung und Alignment Berechnung:
[-v] = (verbose) ausführliche Anzeige.
[-m X] = wie viel % des Speicherplatzes werden für den super-user (root) reserviert? Der Standardwert ist 5 %.
Bei größeren Disks kann es daher sinnvoll sein den Wert zu reduzieren (z. B. auf .5 = 0.5%).
Hauptsächlich dient dies der Verhinderung der Fragmentierung, man sollte also nicht 0 wählen.
#
optimales Alignment Berechnen:
[-b 4096] = (block-size) ist die Größe der Dateisystemblöcke, eigentlich immer 4096 Bytes (4 KiB).
[stripe size] = ist die Chunk-Größe, diese beträgt bei neuen Systemen eigentlich immer 512 KiB - früher oft 64 KiB.
[stide=128] = ist ein berechneter Wert (stride 128 = stripe size 512 / block-size 4)
stride ist also die Chunk-Größe (512 KiB), geteilt durch die Dateisystemblöcke (4 KiB), so das dies eigentlich immer 128 KiB ergibt.
[stripe-width=xxx] = ist ein berechneter Wert (Datenlaufwerke * stride = stripe-width).
Die Anzahl der Datenlaufwerke (keine Parity- oder Hot-Spare-Laufwerke) * 128 KiB ergibt den optimalen stripe-width Wert.
Bsp.: Bei einem RAID 6 mit 5 Laufwerken, werden also nur die 3 Datenlaufwerke berücksichtigt (128 KiB stride * 3 discs = 384 kiB stripe-width).
#
[lazy_itable_init=0,lazy_journal_init=0] = alle Inode-Tabellen (inode table) beim erstellen des Dateisystems vollständig initialisieren (lazyinit).
[-L = (volume label) einen Datenträgernamen festlegen, maximal 16 Zeichen. Ohne dem Parameter wird kein Label (Datenträgername) vergeben.
Tipp: Das Label kann man auch zum mounten verwenden z. B. [mount LABEL="Label-Name"].
[/dev/md0] = ist die Angabe des entsprechenden RAID-Verbundes.

Alternatives Dateisystem:
mit **xfs** Dateisystem:
mkfs.xfs /dev/md0 # mit XFS-File-System formatieren.
xfs ermittelt automatisch die besten stride und stripe-width Werte.

Ergänzende Hinweise zum ext4-Dateisystem (Lazy Initialization/lazyinit):
Das ext4-Dateisystem verwendet zur Verwaltung Inode-Tabellen (inode table). Diese Inode-Tabellen müssen vor der ersten Verwendung initialisiert werden.
Standardmäßig werden die Inode-Tabelle vom mkfs.ext4/mke2fs Befehl nicht vollständig initialisiert um die Bereitstellung des Dateisystems zu beschleunigen.
Dadurch erfordert es aber, dass der Kernel die Initialisierung des Dateisystems im Hintergrund abschließt, wenn das Dateisystem zum ersten Mal gemountet wird.
Dieses Verhalten wurde ab der Kernel Version 2.6.37 eingeführt. Es betrifft daher praktisch alle nachfolgenden/aktuellen Linux Versionen.

Wichtig:
Insbesondere bei größeren Datenträgern (Festplatten bzw. RAID-Systemen) kann die endgültige Initialisierung der Inode-Tabellen
(die gedrosselt im Hintergrund abläuft) mehrere Stunden, Tage oder gar Wochen dauern (abhängig von der Datenträgergröße).
Das Anlegen der Inode-Tabellen erkennt man unter anderem an einer ungewöhnlichen Festplattenaktivität im Hintergrund.
Die Festplatten LED blinkt hier auch bei Nichtbelastung des Systems regelmäßig und ständig im Sekundentakt ununterbrochen immer weiter.
Die Zugriffe erzeugt dabei der im Hintergrund versteckt laufenden Prozess [ext4lazyinit].

Durch die im Hintergrund laufende Initialisierung steht zu Beginn daher nicht die volle Geschwindigkeit des Systems zur Verfügung.
Beachtet dies daher insbesondere bei Benchmarks auf neu angelegten Partitionen (die Ergebnisse wären sonst ggf. verfälscht).
Benchmarks sollte man daher erst durchführen nachdem die Initialisierung der Inode-Tabellen vollständig abgeschlossen wurde.

Wird das Anlegen der Inode-Tabellen unterbrochen z. B. durch ausschalten des Systems oder aushängen der Partition (umount),
dann wird die Initialisierung der Inode-Tabellen beim nächsten Neustart des Systems, bzw. beim nächsten einhängen der Partition (mount) entsprechend fortgesetzt.
Die vollständige Initialisierung der Inode-Tabellen für einen fehlerlosen Betrieb somit zwingend notwendig, diese kann man also nicht verhindern.

Das Anlegen der Inode-Tabellen kann man wie folgt beschleunigen:
Verwendet man beim [mkfs.ext4]-Befehl die zusätzlichen Parameter [-E lazy_itable_init=0,lazy_journal_init=0] (Hinweis: dieser optionale Parameter wurde oben entsprechend schon mit eingepflegt),
dann werden die Inode-Tabellen sofort vollständig initialisiert angelegt. Beachtet dabei, das erzeugen des Dateisystems dauert dadurch etwas länger.

Man hat also folgende Auswahlmöglichkeiten:

- Wenn man sofort direkten Zugriff auf das Dateisystem haben möchte und eine Hintergrundinitialisierung nicht stört, dann kann man die lazy-Parameter auch weglassen (dies ist auch problemlos möglich).
- Wenn man etwas warten kann und die Initialisierung des Dateisystems schnellstmöglich abschließen möchte, dann sollte man die Parameter mit angeben (eher empfohlen).

Beide Varianten sind also möglich, beide Varianten führen schlussendlich zum selben Ergebnis. Entscheiden muss man daher nach dem Anwendungszweck bzw. wie schnell das System zur Verfügung stehen muss.

8. RAID-Verbund einhängen/mounten:

Zunächst sollte man ein Verzeichnis zum einhängen/mounten anlegen:
mkdir /mnt/raid

Hinweis: Hängt man das Dateisystem unter [/media] statt unter [/mnt] ein, so erscheint meist ein Laufwerksicon auf dem Linux Desktop.

Danach kann man den RAID-Verbund so manuell einhängen:
mount /dev/md0 /mnt/raid

Soll der RAID-Verbund immer automatisch bei jedem Systemstart zur Verfügung stehen, so trägt man diesen hier wie folgt ein:
nano /etc/fstab

/dev/md0 /mnt/raid/ ext4 defaults,nosuid,noexec,nodev 1 2
Wichtig:
Der mountpunkt, sprich das Verzeichnis zum einhängen, muss bereits existieren.
RAID-Verbund-Nr., Pfad und Dateisystem muss man entsprechend anpassen.
Verwendet man mehrere arrays, so sollte man hier die eindeutige UUID verwenden.
#
Wichtig: Die folgenden Parameter sollte man für reine Daten/NAS-Laufwerke immer mit angeben:
[nosuid] = SUID- und SGID-Bits werden nicht interpretiert. Dies verhindert u. a. das ausführen von Programmen mit Datei-Besitzer-Rechten ('passwd' läuft z. B. immer als 'root').
[noexec] = Das lokale ausführen von Linux-Programmen und ggf. Skripten wird verhindert. In einer Netzwerkfreigabe abgelegte Windows .exe Dateien kann man immer noch starten.
[nodev] = Die Benutzung von Gerätedateien (direkter Block orientierter Direktzugriff) wird nicht erlaubt.
Diese Angaben erschweren das erlangen von höheren Rechte durch Systemlücken. Die Gefahr eines Hackerangriffs wird dadurch deutlich gesenkt.

Das automatische einhängen/mounten kann man auch ohne einen Neustart testen:
mount -a

Der Zugriff auf das Raidsystem ist ab sofort über den Pfad '/mnt/raid/' möglich.
Möchte man den RAID-Verbund auch im Netzwerk nutzen, dann gibt man diesen wie folgt frei: Link zur Dokumentation.
Danach kann man den Server natürlich auch mit Windows problemlos verwenden.

Optional: Möchte man eine Verknüpfung auf dem Desktop zum RAID-Verbund dann sollte man wie folgt einen Link anlegen:
ln -s /mnt/raid/ /root/Schreibtisch # Hier z. B. für den User root bei einer Xfce4 Umgebung.
ln -s /mnt/raid/ ~/Desktop # Hier z. B. für den aktuellen User bei einer Ubuntu Umgebung.

ln -s /mnt/raid/ /home/Arno/Desktop # Hier z. B. für den speziellen User mit dem Namen "Arno".
Die Pfade zum Desktop/Schreibtisch sind abhängig von der verwendeten Linux-Version teilweise verschieden - diese muss man daher entsprechend anpassen.

Von Windows aus kann man nun die Laufwerke mit einem Batch-Script verbinden:
notepad logon.cmd # unter Windows

Hier einfach das folgende schöne Script einfügen:

```
@echo off
title Netzwerk:
echo    _______
echo   ^|:'    __^| ____ ____ __ __ _____ ____
echo   ^|__     '^|' -__^|:   _^|: ^|: ^|: -__^|:   _^|
echo   ^|_______^|_____^|__^|  \___/^|_____^|__^|
echo.
color e0

echo ... verbinde NAS Datenlaufwerke ...
echo.

if not exist s:\ ( net use s: \\192.168.0.13\DATA   /user:BENUTZERNAME
PASSWORT /persistent:no ) else ( echo NAS s: ist bereits verbunden. )
if not exist r:\ ( net use r: \\192.168.0.13\BACKUP /user:BENUTZERNAME
PASSWORT /persistent:no ) else ( echo NAS r: ist bereits verbunden. )

if not exist s:\ (
        color c0
        title Netzwerk: Fehler
        echo.
        echo FEHLER: Server nicht erreichbar!
        pause
        exit 1) else (
        color a0
        title Netzwerk: Ok
        echo.
        echo Netzlaufwerke wurden verbunden. )

timeout /t 13
```

s: ist dabei gedacht als Serverlaufwerk für die Daten.
r: ist dabei für den Zugriff auf nur lesbare Backups (read-only) gedacht.
Hinweis: Die ASCII-Grafik wurde dabei mit '^' so maskiert, dass die '|' Pipe Symbole korrekt dargestellt werden. Die Grafik sieht beim Ausführen also besser aus.
Laufwerksbuchstaben, IP-Adresse, Freigabename, Benutzername und das Passwort muss man entsprechend anpassen.

9. Hot-Spare-Laufwerk hinzufügen:

Will man ein optionales Hot-Spare-Laufwerk (oft nur als Hotspare oder spare bezeichnet) hinzufügen,
so muss man dieses Laufwerk erst identisch wie die bereits vorhandenen Laufwerke im RAID-Verbund partitionieren.

Anschließend kann man das Laufwerk wie folgt hinzufügen:
mdadm /dev/md0 --add /dev/sdX1

Sollte nun ein Laufwerk aus dem RAID-Verbund ausfallen, so übernimmt das Hot-Spare-Laufwerk dann sofort die Funktion des ausfallenden Laufwerks.
Es startet praktisch sofort nach einem [fail] ein automatischer rebuild auf das Hot-Spare-Laufwerk.

Ob dies auch Funktioniert kann man einfach Testen in dem man ein Laufwerk manuell als [fail] kennzeichnet.
Hinweis: Diesen Test sollte man nicht im Produktionsbetrieb machen!

Wichtig:
Nach dem man das Hot-Spare-Laufwerk hinzugefügt hat, muss man die [mdadm.conf]-Konfiguration aktualisieren.
Ebenso sollte man nun die veränderte Laufwerkszuordnung und ggf. die UUID-Device-Zuordnung nochmals aktualisieren/dokumentieren.

10. E-Mail Benachrichtigung bei Ausfall:

mdadm kann über einen MTA (mail transfer agent) des Systems, den Anwender per E-Mail, über den Ausfall eines Laufwerks informieren.
Als MTA (mail transfer agent) richtet man am besten postfix als Satellitensystem ein.
Satellitensystem bedeutet hierbei, dass das System keine E-Mails empfängt, sondern diese nur bei Bedarf über einen Smarthost versendet.
A. Zuerst muss Postfix eingerichtet werden:
apt-get install postfix libsasl2-modules mailutils

Parametererklärung:
[postfix] = ist der eigentliche MTA (mail transfer agent).
[libsasl2-modules] = wird benötigt um eine Absenderauthentifizierung zu ermöglichen (wg. Spamschutz zwingend notwendig).
[mailutils] = installiert unter anderem das Konsolen-Programm [mail], welches man für Test E-Mails verwenden kann.

Bei der Installation muss man die Postfix-Basiskonfiguration durchführen (es kommen ein paar Fragen die man wie folgt beantwortet):
1. Bei [Allgemeine Art der Konfiguration:] wählt man den Eintrag [Satellitensystem] aus (Versand über Smarthost).

2. Bei [System-E-Mail-Name:] übernimmt man den [PC-Name/Hostname] einfach unverändert so wie er ist.

3. Bei [SMTP-Relay-Server (leere Eingabe: keiner):] ändert man den Wert: [smtp.localdomain] z. B. in [smtp.ionos.de:465] (bei ionos.de) oder in [w01xxxxx.kasserver.com:465] (bei all-inkl.com) ab.

Wichtige ergänzende Hinweise:

Die Angabe des SMTP-Servers muss hier also mit der Angabe des Ports meist :465 erfolgen.
Zur besseren Erklärung haben ich hier zwei Beispiele für ionos.de und all-inkl.com angegeben.
Hier muss man also nur einen Eintrag eigeben. Die SMTP-Server-Angaben müssen logischerweise zur verwendeten E-Mailadresse passen.
Wenn Ihr einen anderen E-Mail-Anbieter/Hoster verwendet müsst ihr diesen Eintrag entsprechend anpassen.

Beim Anbieter ionos.de (ehemals 1und1.de) wurde früher der SMTP-Server smtp.1und1 über den Port :587 verwendet.
Bei Neuinstallationen sollte man hier bei ionos.de (ehemals 1und1.de) zukünftig den neuen SMPT-Server smpt.ionos.de über den Port :465 verwenden (wie auch oben im Beispiel angegeben).

Typischerweise werden für dem E-Mailversand die folgenden Ports genutzt:
Port 25 = Standard-MTA (veraltet/ohne Authentifizierung).
Port 587 = Meist mit SMTP-Auth also mit Authentifizierung vor dem Senden (durchaus noch gängig).
Port 465 = Neuere sichere Authentifizierung über SSL/TLS (empfohlen).
Beachtet auch alle E-Mails werden trotz Authentifizierung z. B. über SSL/TLS im Klartext also unverschlüsselt übertragen.

Beim Anbieter all-inkl.com muss für jede Konfiguration individuell der Wert w01xxxxx mit der entsprechenden Login-Kennung von all-inkl.com ausgetauscht werden.
Aus dem oben angegebenen Beispiel w01xxxxx.kasserver.com:465 wird dann z. B. w0148930.kasserver.com:465 bzw. für jede Domain gibt es hier also ein individuelles Login.

Will man Änderungen an der Postfix-Konfiguration vornehmen, so startet man den Assistenten wie folgt neu:
dpkg-reconfigure postfix

Es kommen hier noch weitere Fragen wie z. B. zum Empfänger usw., diese überspringt man einfach in dem man die vorgegebenen Werte so belässt - hier wählt man also immer [<OK>] aus.
B. Postfix Konfiguration anpassen:
nano /etc/postfix/main.cf

In der Datei [main.cf] fügt man am Start folgendes hinzu:
main.cf Start-Änderung
#
# Absender: umschreiben
sender_canonical_maps = hash:/etc/postfix/sender_canonical

# Absender: Authentifizierung nötig = Ja
smtp_sasl_auth_enable = yes

# Absender: SMTP Server E-Mail und Passwort muss in der sasl_password stehen.
smtp_sasl_password_maps = hash:/etc/postfix/sasl_password

Absender: Verschlüsselungseinstellungen = Nicht anonym anmelden, also ein Benutzername und Passwort senden.

smtp_sasl_security_options = noanonymous

# Aufgrund gestiegener Sicherheitsanforderungen "**E-Mail made in Germany**" sind folgende Punkte ebenso notwendig:
smtp_tls_wrappermode = yes
smtp_tls_security_level = encrypt
# **Wichtig:** Beachtet hier dabei dass der Eintrag 'smtp_tls_security_level=may' in Konfigurationsdatei in neueren Versionen > Ubuntu 18.04 schon vorhanden ist.
# Diesen vorhandenen Eintrag muss man noch mit einem führenden Rautezeichen '#' auskommentieren oder man löscht die Zeile.

#
main.cf Ende-Änderung

C. Als nächstes muss man die [sasl_password]-Datei erstellen:
nano /etc/postfix/sasl_password

Hier muss man dann die Zugangsdaten wie folgt eingeben:
E-Mail-SMTP-Server EuereEmail@Adresse.de:EuerPasswort
Den ":" zwischen E-Mailadresse und Passwort nicht vergessen.
Die SMTP-Server Angabe erfolgt hier ohne den Port ":465" bzw. ":587".
So das der Eintrag z. B. so aussehen sollte:

Hier ein Beispiel für ionos.de:
smtp.ionos.de raid@ctaas.com**:**EuerPassword

Hier noch ein weiteres Beispiel für all-inkl.com (die Login-Kennung & E-Mail entsprechend anpassen):
w01xxxxx.kasserver.com raid@ctaas.com:EuerPassword

Hinweise:
Verwendet hier nur einen Eintrag. Also den passenden zur E-Mailadresse.
Die hier hinterlegte Adresse wird für das Login im E-Mailpostfach verwendet,
diese dient also der Authentifizierung beim Provider, um Spam-Versand zu verhindern.
Postfix funktioniert daher nicht mit reinen Weiterleitungsadressen.
E-Mailadressen, die beim Provider (in unserem Falle ionos) so konfiguriert sind,
dass diese E-Mails direkt an eine andere Adresse weiter leiten, funktionieren nicht.
Man muss sich also in das hier angegebene E-Mail-Postfach einloggen können.

D. Um einen Missbrauch der Zugangsdaten zu verhindern, sichert man anschließend die gespeicherten Zugangsdaten so ab, dass diese nur noch der User root lesen kann:
chmod 600 /etc/postfix/sasl_password # Berechtigungen setzen.
ls -l /etc/postfix/sasl_password # Berechtigungen prüfen.

E. Jetzt erzeugt man aus der [sasl_password]-Datei eine Datenbank für Postfix:
postmap /etc/postfix/sasl_password

Dies erzeugt automatisch die [/etc/postfix/sasl_password**.db**]-Datenbank.
Diesen Befehl muss man immer wiederholen, sofern man in der [sasl_password]-Datei Änderungen vorgenommen hat.
F. Korrekten Absender in der [sender_canonical]-Datei setzen:

nano /etc/postfix/sender_canonical

Hinweis: raid@ctaas.com ist hier nur Beispielhaft angegeben.
Hier muss man eine also die eigene E-Mailadresse eingeben:
EuerLinuxUsername raid@ctaas.com
www-data raid@ctaas.com
root raid@ctaas.com

 # Erklärungen:
Es erfolgt hier also die Zuweisung zwischen Benutzername und Absender-E-Mailadresse.
EuerLinuxUsername sendeadresse@eigenedomain.de = ist der lokale Benutzername.
www-data sendeadresse@eigenedomain.de = über www-data sendet ggf. ein apache2 Webserver E-Mails, wenn man keinen apache2 installiert hat kann man die Zeile auch weg lassen.
root sendeadresse@eigenedomain.de = ist der lokale root (supervisor/admin) User.

G. Daraus muss man ebenfalls eine Datenbank für postfix erzeugen:
postmap /etc/postfix/sender_canonical

Dies erzeugt automatisch die [/etc/postfix/sender_canonical**.db**]-Datenbank.
Diesen Befehl muss man immer wiederholen, sofern man in der [sender_canonical]-Datei Änderungen vorgenommen hat.
H. Um alle Änderungen zu übernehmen muss man postfix Neustarten:
service postfix restart # Bei aktuellen Systemen.
systemctl reload postfix # Bei neuen Systemen mit systemd.
/etc/init.d/postfix restart # Bei älteren Systemen.

I. Eine Testmail über das System versenden:
Bevor man die mdadm Konfiguration anpasst sollte man zuerst auf Systemebene die postfix Konfiguration überprüfen. Hierzu versendet man einfachsten eine Test-E-Mail über das System.
Um ggf. auftretende Fehler besser zu erkennen, ist es Hilfreich wenn man gleichzeitig eines der folgenden Logfiles [mail.log] oder [syslog] überwacht.
tail -F /var/log/mail.log

Nun öffnet man ein weiteres Terminal-Fenster bzw. eine neue Console und sendet wie folgt eine Test-E-Mail:
Den Inhalt, mit dem entsprechenden Subjekt, an die E-Mailadresse raid@ctaas.com senden:
echo "Inhalt" | mail -s "Subjekt" raid@ctaas.com

Alternativ kann man wie folgt den Inhalt einer beliebigen Datei versenden:
nano Mailtext.txt # Hier einen beliebigen Text eingeben und speichern.
mail -s "Betreff der E-Mail" raid@ctaas.com < Mailtext.txt # Den Inhalt der Datei Testmail.txt an raid@ctaas.com senden.

Weiterhin kann man die E-Mailwarteschlange (queue) prüfen - diese sollte sich in kürzester Zeit leeren:
mailq

Sollte das versenden nicht geklappt haben, so kann man wie folgt alle E-Mails aus der Warteschlange löschen:

postsuper -d ALL # (delete all)

Den Posteingang der E-Mail sollte man ebenso entsprechend prüfen.
Wenn eine Test-E-Mail angekommen ist, kann man die E-Mailadresse in der mdadm Konfiguration eintragen.
J. mdadm die E-Mailadresse bekannt machen:
Um bei Ausfällen eine Nachricht zu erhalten, muss man hier unter [MAILADDR] nun statt [root] die eigene E-Mailadresse eintragen:
nano /etc/mdadm/mdadm.conf

...
MAILADDR raid@ctaas.com
...

Alternativ kann man auch den Assistenten starten und hier die E-Mailadresse eingeben.
dpkg-reconfigure mdadm

K. mdadm E-Mailtest:
Hat man nun die E-Mail-Adresse in der mdadm Konfigurationsdatei hinterlegt, so kann man sich wie folgt eine Test-E-Mail zusenden:
mdadm --monitor --scan --test --oneshot

Man sollte nun eine E-Mail mit den Details des RAID-Arrays erhalten.

Hier mal eine Beispiel E-Mail eines produktiven Systems - 'NAS' ist hier der PC-Name, es handelt sich hierbei um ein RAID 6 (md0) mit 5 Laufwerken ohne Hot-Spare-Laufwerk:
E-Mail Betreff : *TestMessage event on /dev/md/0:NAS*
E-Mail Absender: *mdadm monitoring*

This is an automatically generated mail message from mdadm
running on NAS

A TestMessage event had been detected on md device /dev/md/0.

Faithfully yours, etc.

P.S. The /proc/mdstat file currently contains the following:

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md0 : active raid6 sdb1[3] sdd1[1] sdc1[2] sdf1[4] sde1[0]
8790389760 blocks super 1.2 level 6, 512k chunk, algorithm 2 [5/5] [UUUUU]
bitmap: 0/22 pages [0KB], 65536KB chunk

unused devices: <none>

Ein kleiner Hinweis hierzu:
In älteren Linux Umgebungen wurde ein RAID-Verbund immer unter '/dev/md/0' usw. abgelegt.
In neueren Linux Umgebungen befindet sich der RAID-Verbund in der Regel immer direkt unter '/dev/md0' also nicht nochmal in einem Unterordner.
Aus Kompatibilitätsgründen findet man in neueren Umgebungen trotzdem in dem

Unterordner '/dev/md/' dann meist einen Symbolischen Link zum eigentlichen RAID-Verbund also einen Softlink auf '/dev/md0'.

Hier dazu mal ein Beispielausgabe:
root@NAS1:/# ls -l /dev/md/0
insgesamt 0
drwxr-xr-x 1 root root 6 Feb 2 11:50 /dev/md/0 -> ../md0

Dieser Softlink hier ↑ wird angelegt,
damit auch ältere Software kompatibel mit dem RAID funktioniert.
Der RAID-Verbund '0' ist somit nicht vorhanden sondern nur ein Softlink auf den RAID-Verbund im darunter liegenden Verzeichnis eben auf '../md0'.

Man erkennt z. B. oben in der TestMessage das hier auch '/dev/md/0' verwendet wird. Tatsächlich lag der RAID-Verbund aber auch hier unter '/dev/md0'.
Wie gesagt, das ist nur ein Schönheitsfehler der durch die Kompatibilität hier fälschlicherweise '/dev/md/0' ausgibt, tatsächlich wäre hier '/dev/md0' richtig.

Will man diesen Statusbericht bei jedem Neustart erhalten, z. B. nach Stromausfall bzw. nach Systemupdates mit Neustart, so ergänzt man den folgenden Wert so:
nano /etc/default/mdadm

...
DAEMON_OPTIONS="--syslog --test"
...

Hinweis:
In aktuellen Versionen wird die TestMessage nicht mehr beim Start/booten des Systems versand.
Es gibt hier einen offenen Bug der noch nicht gefixt wurde (siehe auch Link).
Die Parameterergänzung --test kann man also weglassen oder wenn man den Parameter bereits ergänz hat kann man es aber auch so lassen (vielleicht behebt ja mal jemand den Bug).

Man muss sich nun aber keine Sorgen machen, denn die Benachrichtigung funktioniert trotzdem so wie hier beschrieben im Fehlerfall.

Hier nun ein paar Beispiel-Meldungen bei defekten vom Raid-Verbund im Fehlerfall:
Bei einem defekt im laufenden System erhält man eine E-Mail.
Hier mal eine Beispiel E-Mail eines Test-Systems - 'NAS' ist hier der PC-Name, es handelt sich hierbei um ein RAID 6 (md0) mit 4 Laufwerken ohne Hot-Spare-Laufwerk.
Hier ist Beispielsweise das Laufwerk /dev/sdb1 ausgefallen.
E-Mail Betreff : *Fail event on /dev/md/0:NAS*
E-Mail Absender: *mdadm monitoring*

This is an automatically generated mail message from mdadm
running on NAS

A Fail event had been detected on md device /dev/md/0.

It could be related to component device /dev/sdb1.

Faithfully yours, etc.

P.S. The /proc/mdstat file currently contains the following:

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]

md0 : active raid6 sdf1[5](S) sdd1[2] sdc1[1] sdb1[4](F) sde1[3]
14336 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/3] [_UUU]

unused devices:

Startet man das defekte System nun neu, dann erhält man nun bei jedem Neustart per E-Mail eine Meldung die auf eine Verschlechterung (degraded) des RAID-Verbund hinweist.
Hier erkennt man nur am fehlenden sdb1 das dieses Laufwerk wohl defekt ist.
E-Mail Betreff : *DegradedArray event on /dev/md/0:NAS*
E-Mail Absender: *mdadm monitoring*

This is an automatically generated mail message from mdadm
running on NAS

A DegradedArray event had been detected on md device /dev/md/0.

Faithfully yours, etc.

P.S. The /proc/mdstat file currently contains the following:

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md0 : active raid6 sde1[3] sdc1[1] sdd1[2]
2082816 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/3] [_UUU]

unused devices:

Ergänzend vielleicht noch ein wichtiger Hinweis:
Wenn man ein Hot-Spare-Laufwerk verbaut hat, dann springt dieses direkt ein (das ist auch so gewollt und Sinnvoll).
Leider erhält man dann bei einem defekt eines Laufwerks keine E-Mail wenn das Hot-Spare-Laufwerk die Funktion des ausgefallenen Laufwerks übernimmt.
Hier wäre meiner Meinung nach eine Info-E-Mail angebracht. Eventuell ist es ein Bug oder fehlendes Feature?
Man erhält dann also erst wieder eine Information, wenn dann ein neues Laufwerk ausfällt und eben kein Hot-Spare-Laufwerk mehr zur Verfügung steht.
Man sollte daher doch von Zeit zu Zeit manuell prüfen ob das Hot-Spare-Laufwerk bereits aktiv als Ausfalllaufwerk genutzt wird.

11. RAID-Verbund & Partitionen anzeigen:

Informationen zum RAID-Verbund anzeigen:
mdadm --detail /dev/md0
mdadm --examine --scan --verbose
cat /proc/mdstat

Details einer Partition anzeigen:
mdadm --examine /dev/sdX1
Eindeutige UUID der entsprechenden Partition ausgeben (Device UUID):
mdadm --examine /dev/sdX1 | grep UUID

Tipp:
Man sollte sich diese Ausgaben speichern, denn diese können im Fehlerfall sehr Hilfreich sein.

12. RAID-Verbund wieder aktivieren:

Falls der RAID-Verbund nicht mehr ansprechbar ist, weil man z. B. ein defektes Laufwerk entfernt hat und [mdadm --examine /dev/md0] die Meldung [State : inactive] meldet, so kann man den RAID-Verbund (wenn das Level >= 1 ist) wie folgt wieder aktivieren:
mdadm --run /dev/md0

Der RAID-Verbund sollte danach wieder ansprechbar sein - der Status wechselt dabei auf [State : clean, degraded].
Ausgefallene Laufwerke sollte man anschließend schnellstmöglich ersetzen.

Sollte [mdadm --run /dev/md0] noch nicht funktionieren, so kann man den RAID-Verbund mit allen noch verbliebenen Laufwerken wie folgt wieder in Betrieb nehmen:
mdadm --assemble --run /dev/md0 /dev/sdb1 /dev/sdc1 ...
mdadm --create --assume-clean --level=X --raid-devices=X /dev/md0 /dev/sdb1 /dev/sdc1 ... missing
Hierbei ist insbesondere die richtige Reihenfolge der Laufwerke wichtig.
Zuvor sollte man ein Backup der Partitionen/Laufwerke machen!

Wenn mehrere Laufwerke auf Grund eines Controllerdefektes ausfallen (also kein Laufwerkschaden vorliegt), so kann man den RAID-Verbund ggf. wie folgt reparieren.
1. Schritt: RAID-Verbund stoppen:
mdadm --stop /dev/md0

2. Schritt: RAID-Verbund soweit wie möglich zusammenfügen.
mdadm --assemble /dev/md0 /dev/sdb1 /dev/sdc1 --force
Hierbei muss man die zuletzt funktionierende Konfiguration verwenden.

3. Schritt: Ggf. weitere Partitionen dem Verbund hinzufügen:
mdadm --re-add /dev/md0 /dev/sdd1 /dev/sde1 /dev/sdf1

13. RAID-Laufwerk ersetzen:

Wichtig:
Bevor man ans Wechseln des defekten Laufwerks geht sollte man folgendes beachten:
Fällt ein Laufwerk z. B. [/dev/sdd] komplett aus, so rutschen möglicherweise alle anderen Laufwerke in der Reihe nach.
Das bedeutet z. B. das ein Laufwerk welches bisher als [/dev/sde] angesprochen wurde, nun als [/dev/sdd] angesprochen wird.
Der RAID-Verbund kann trotzdem noch angesprochen werden, da hier die eindeutige Zuordnung mittels UUID (universally unique identifier) verwendet wird.
Auf jeden Fall sollte man hier mit Ruhe und Sorgfalt vorgehen, um zu vermeiden das man versehentlich ein falsches Laufwerk wechselt - denn letzteres könnte alle übrigen Daten zerstören.

Sofern der RAID-Level >= 1 ist, so kann man den RAID-Verbund wie folgt reparieren.
1. Schritt: Feststellen welches Laufwerk man wechseln möchte (durch Logs, E-Mail bzw. den RAID-Verbund Status prüfen).
Zuordnung der Laufwerke & UUIDs (universally unique identifier) anzeigen, ein ggf. vorhandener Cache wird ignoriert:
blkid -c /dev/null
Eindeutige UUID des entsprechenden Laufwerks ausgeben (Device UUID).
mdadm --examine /dev/sdX1 | grep UUID
Laufwerk Daten wie die Seriennummer Anzeigen:
hdparm -i /dev/sdX
RAID-Verbund-Status anzeigen - defekte Laufwerke werden meist mit einem **(F)** gekennzeichnet:
mdadm --detail /dev/md0

2. Schritt: Die defekte Partition (Laufwerk) als Fehlerhaft (faulty) markieren (sofern dies nicht schon automatisch geschehen ist):
mdadm --fail /dev/md0 /dev/sdX1

3. Schritt: Die defekte Partition (Laufwerk) aus dem RAID-Verbund entfernen:
mdadm --remove /dev/md0 /dev/sdX1
Wichtig: Sollte man das defekte Laufwerk nicht aus dem System ausbauen, so muss man den Superblock vom defekten Laufwerk löschen.

4. Schritt: Die neue Partition (Laufwerk) hinzufügen:
Wichtig: Das neue Laufwerk muss man zuvor entsprechend partitionieren.
mdadm --add /dev/md0 /dev/sdX1

5. Schritt: Den Rebuildstatus überprüfen:
watch -n1 cat /proc/mdstat

6. Schritt: Die mdadm.conf aktualisieren.
Ebenso sollte man nun die veränderte Laufwerkszuordnung und ggf. die UUID-Device-Zuordnung nochmals aktualisieren/dokumentieren.

Tipps:
1. Vor dem Wechsel eines defekten Laufwerks sollte man wenn möglich die Daten extern sichern.
2. Ist man beim Ausbau unsicher ob dies wirklich "das defekte Laufwerk" ist, so ist es

einfacher und sicherer wenn man zuerst ein zusätzliches Hot-Spare-Laufwerk einbaut und wartet bis darauf der rebuild abgeschlossen ist. Erst dann entfernt man das defekte Laufwerk. Sollte kein interner Anschluss mehr frei sein, so kann man ggf. vorübergehend auch ein externes USB-Laufwerk verwenden.

14. RAID wieder einbinden (Systemwechsel):

Man kann einen bestehenden RAID-Verbund nach einer Neuinstallation bzw. nach Wechsel des Betriebssystems wie folgt einbinden:

1. Zuerst sollte man mdadm installieren und
2. ggf. zusätzlich postfix konfigurieren.
3. Nach einem Neustart des Systems sollten alle ehemaligen RAID-Verbünde automatisch verfügbar sein.
reboot

4. Optional: Wird ein RAID-Verbund nicht erkannt (z. B. alte RAID-Verbünde mit altem Superblock), so kann man diese wie folgt per Hand aktivieren:
Alle RAID-Verbünde suchen und bereitstellen:
mdadm --assemble --scan

nur den RAID-Verbund mit der entsprechenden UUID bereitstellen:
mdadm --assemble --scan --uuid=12345678:12345678:12345678:12345678

Den RAID-Verbund anhand folgender Partitionen zusammen setzen (entsprechend anpassen):
mdadm --assemble /dev/md0 /dev/sdb1 /dev/sdc1 /dev/sdd1 ...
Wichtig: Man darf hier nicht die Option [--create] verwenden,
denn diese Option würde ein neuen Verbund erstellen und alle vorhandenen Daten löschen!

5. Zuletzt muss man die Konfiguration in der mdadm.conf speichern.

15. RAID-Verbund im Live-System:

Will man einen RAID-Verbund über ein Live System ansprechen (Empfehlung: xubuntu release images), so sind folgende Schritte nötig:
1. Schritt: mdadm Installation ohne postfix (mail transfer agent):
apt-get install --no-install-recommends mdadm

2. Schritt: RAID-Verbund suchen und anzeigen:
mdadm --examine --scan --verbose
Unter anderem werden die betroffenen Laufwerke, die UUID sowie der Hostname des ursprünglichen Linux-Systems ausgegeben.

3. Schritt: RAID-Verbund suchen und zusammensetzen:
mdadm --assemble --scan

Falls das zusammensetzen erfolgreich war, erhält man Informationen über alle betroffenen Laufwerke.
#
Manchmal schlägt [mdadm --assemble --scan] fehl - wenn man auf defekte bzw. fehlende Laufwerke zugreifen muss.
Hier muss man alle die zum RAID-Verbund dazugehörigen Laufwerke wie folgt übergeben:
mdadm --assemble --run /dev/md0 /dev/sdX1 ...

4. Schritt: RAID-Status prüfen:
cat /proc/mdstat

5. Schritt: mount-Punkt anlegen:
mkdir /mnt/raid

6. Schritt: RAID-Verbund entsprechend einhängen:
mount -v /dev/md0 /mnt/raid

16. RAID 5/6 vergrößern:

Reicht der Speicherplatz bei einem RAID-Verbund nicht mehr aus, so kann man diesen durch zusätzliche Laufwerke erweitern.
Einen RAID-Verbund mit Level 5 oder Level 6 erweitert man mit folgenden Schritten:
1. Neue Laufwerke muss man zunächst partitionieren und dem bestehenden RAID-Verbund als Hot-Spare-Laufwerk hinzufügen.

2. Danach sollte man zuerst die Konfiguration aktualisieren.

3. Danach kann man den RAID-Verbund wie folgt erweitern:
mdadm --grow --raid-devices=X /dev/md0 --backup-file=/root/md0.bak
#
Parametererklärung:
[--grow] = wachsen/erweitern des Verbundes.
[--raid-devices=X] = hier gibt man die Anzahl der zu verwendenden Laufwerke (Daten + Parity - ohne Hot-Spare) an,
aus denen der RAID-Verbund nach der Erweiterung bestehen soll.
[/dev/md0] = gibt an welcher RAID-Verbund erweitert werden soll.
[--backup-file=/root/md0.bak] = legt im Home-Verzeichnis von root eine Sicherung kritischer Bereiche ab.
Wichtig: Diese Sicherung darf nicht auf dem zu erweiternden RAID-Verbund liegen.
Neuere mdadm-Versionen speichern übrigens trotz einer Ausgabe nicht immer eine Sicherung.
Ohne den [--backup-file=]-Parameter wird der RAID-Verbund überhaupt nicht vergrößert und es kommt zu einem Abbruch - den Parameter muss man also zwingend angeben.

watch -n1 cat /proc/mdstat
Den Status überwachen - der Status wechselt dabei auf [State: clean, reshaping].

Wichtig: Vor der Erweiterung sollte man eine Datensicherung durchführen.

Das System sollte man während der Erweiterung nicht neu starten, da eine abgebrochene Erweiterung nicht automatisch fortgesetzt wird.

Sollte die Erweiterung fehlschlagen:
so muss man die Konfiguration neu bearbeiten wenn man zwischenzeitlich das System neu gestartet hat,
da aus dem bisherigen Hot-Spare-Laufwerk ja nun ein Datenlaufwerk geworden ist.
Anschließend kann man die Erweiterung wie folgt fortsetzen (nur wenn zuvor eine Sicherungsdatei angelegt wurde):
mdadm /dev/md0 --grow --continue --backup-file=/root/md0.bak
Falls keine Sicherung angelegt wurde, so muss man den RAID-Verbund manchmal wieder aktivieren und ggf. manuell Vergrößern siehe Punkt 18.2.

4. Wenn die Erweiterung abgeschlossen ist muss man noch die mdadm.conf-Konfigurationsdatei aktualisieren.
Ebenso sollte man nun die veränderte Laufwerkszuordnung und ggf. die UUID-Device-Zuordnung nochmals aktualisieren/dokumentieren.

5. Zuletzt muss man noch das Dateisystem erweitern, um den neu entstandenen Speicherplatz nutzen zu können.

Erweiterung eines ext4-Dateisystems:
1. Schritt: Dateisystem aushängen (zuvor alle Dienste [service xyz stop] die auf dem Laufwerk laufen beenden):
umount /dev/md0

2. Schritt: Vor der Erweiterung muss man das Dateisystem auf Fehler überprüfen:
fsck.ext4 -f /dev/md0

3. Schritt: Dateisystem auf die neue Maximalgröße erweitern.
resize2fs /dev/md0

4. Schritt: Dateisystem wieder einhängen (mountpfad entsprechend anpassen):
mount /dev/md0 /mnt/raid

Erweiterung eines XFS-Dateisystems:
xfs_growfs /dev/md0

17. RAID 0 vergrößern:

Reicht der Speicherplatz bei einem RAID-Verbund nicht mehr aus, so kann man diesen durch zusätzliche Laufwerke erweitern.
Einen RAID 0 Verbund kann man über folgende Schritte erweitern:
1. Neue Laufwerke muss man zunächst partitionieren.

2. Da RAID 0 keine Hot-Spare-Laufwerke unterstützt muss man die Erweiterung und das hinzufügen des neuen Laufwerks in einem Schritt durchführen:

mdadm --grow /dev/md0 --raid-devices=X --add /dev/sdX1
#
Parametererklärung:
[--grow] = wachsen/erweitern des Verbundes.
#
[/dev/md0] = gibt an welcher RAID-Verbund erweitert werden soll.
#
[--raid-devices=X] = hier gibt man die Anzahl der zu verwendenden Laufwerke an,
aus denen der RAID-Verbund nach der Erweiterung bestehen soll.
#
Mit [--add /dev/sdX1] wird das neu hinzu zu fügende Lauferk angegeben.

Wichtig: Vor der Erweiterung sollte man möglichst eine Datensicherung durchführen.
Bei der Erweiterung wird der RAID-Verbund zunächst in ein RAID-Level 4 konvertiert,
der Laufwerksstatus steht währenddessen auf [State : clean, FAILED, reshaping].
Man sollte dann abwarten bis der reshape abgeschlossen wurde.
Also warten bis der Laufwerksstatus auf [State : clean, degraded] oder auf [State : clean] steht (der Status ist dabei abhängig von der Laufwerksanzahl).
Der Status wechselt nur auf 'clean, degraded', wenn bei dem umgestellten RAID-Verbund der ja jetzt vorübergehend ein RAID-Level 4 hat ein Laufwerk fehlt.
Das 'Problem' löst sich dann mit der Zurückkonvertierung in ein RAID-Level 0 wieder von ganz allein.
Die zwischenzeitigen 'clean, FAILED ...' oder 'clean, degrade' Statusmeldungen sind also völlig normal, diese kann man während der Konvertierung demnach ignorieren.
watch -n1 cat /proc/mdstat
mdadm --detail /dev/md0

Wichtig: Das System sollte man während der Erweiterung möglichst nicht neu starten, da eine abgebrochene Erweiterung nicht automatisch fortgesetzt wird.
Wurde das System während der Erweiterung trotzdem neu gestartet, so kann man die Erweiterung wie folgt fortsetzen:
mdadm /dev/md0 --grow --continue

3. Im Anschluss wird dann der RAID-Verbund wieder in ein RAID-Level 0 konvertiert:
mdadm --grow --level=0 /dev/md0

Das RAID-Level ändert sich wieder auf 0 und der Laufwerkstatus wechselt wieder auf [State : clean] (alles ok).
mdadm --detail /dev/md0

4. Wenn die Erweiterung abgeschlossen ist muss man noch die mdadm.conf-Konfigurationsdatei aktualisieren.
Ebenso sollte man nun die veränderte Laufwerkszuordnung und ggf. die UUID-Device-Zuordnung nochmals aktualisieren/dokumentieren.

5. Zuletzt muss man noch das Dateisystem erweitern, um den neu entstandenen Speicherplatz nutzen zu können.

18. RAID Laufwerke durch größere ersetzen:

Reicht der Speicherplatz in einem RAID-Verbund nicht mehr aus, so kann man diesen meist durch schrittweise austauschen einzelner Laufwerke vergrößern.
Dies kann durchaus im laufenden Betrieb geschehen (minimale downtime), sofern Hot-Swap unterstützt wird. Anderenfalls beendet man den Server während des Laufwerkstausches immer kurzzeitig.
Der Superblock-Version des zu vergrößernden RAID-Verbundes muss dabei am Anfang der jeweiligen Partitionen liegen (also in der Version 1.1 oder 1.2 vorliegen), denn nur dann kann man den RAID-Verbund und das Dateisystem entsprechend vergrößern.
Ist die Superblock-Version kleiner (also nur 0.9/1.0), so würde ich von einer Vergrößerung abraten, da hier der Superblock am Ende der jeweiligen Partition liegt.

Ablauf:
Man tauscht hier nach und nach einzelne Laufwerke aus und wartet immer ab bis der rebuild entsprechend abgeschlossen ist.
Diesen Vorgang wiederholt man so lange, bis man alle Laufwerke des RAID-Verbunds gewechselt hat.
Je nach Größe der Laufwerke bzw. des RAID-Verbundes kann demnach so eine Umstellung durchaus mehrere Tage in Anspruch nehmen.
Während der Umstellung ist der RAID-Verbund einer deutlich höherer Belastung ausgesetzt.
Um einen Datenverlust vorzubeugen sollte man, daher zuvor alle Daten sichern.

Für den Austausch gibt es zwei Möglichkeiten:
1a. Hat man keinen weiteren Laufwerkanschluss mehr frei, so kann man ein Laufwerk ersetzen in dem man dieses als fehlerhaft kennzeichnet und entsprechend ersetzt.
Während des rebuilds ist der Status des RAID-Verbundes [State: clean, degraded].
Weitere Laufwerksausfälle während des rebuilds können Datenverluste nach sich ziehen.

1b. Hat man jedoch einen freien Laufwerksanschluss zur Verfügung, so wechselt man die Laufwerke besser über replace (empfohlen).
Während des replace bleibt hier der Status des RAID-Verbundes [State: clean, resyncing].
Weitere Laufwerksausfälle während des rebuilds sind hier weniger kritisch.

2. Hat man alle Laufwerke erfolgreich ausgetauscht, so kann man nun den RAID-Verbund auf die volle Größe erweitern:
mdadm --grow /dev/md0 -z max

Der Status des RAID-Verbundes wechselt dabei auf [State: clean, recyncing].

3. Am Ende muss man noch die mdadm.conf-Konfigurationsdatei aktualisieren.
Ebenso sollte man nun die veränderte Laufwerkszuordnung und ggf. die UUID-Device-Zuordnung nochmals aktualisieren/dokumentieren.

4. Um den neu entstandenen Speicherplatz nutzen zu können, muss man noch das Dateisystem erweitern.

19. RAID 5 in RAID 6 konvertieren:

Will man ein RAID 5 in ein RAID 6 konvertieren, so muss man zuerst ein Hot-Spare-Laufwerk hinzufügen.
Danach sollte man zuerst die Konfiguration aktualisieren.
Anschließend kann man den RAID-Verbund wie folgt konvertieren:
mdadm --grow /dev/md0 --level=6 --backup-file=/root/md0.bak
Wichtig: Die Sicherung darf nicht auf dem RAID-Verbund gespeichert werden.
Der Status wechselt dabei auf [State: clean, degraded, reshaping].

Sollte das konvertieren Fehlschlagen, so sollte man zuerst die Konfiguration prüfen.
#
Wurde das konvertieren unterbrochen, so muss man diese wie folgt wiederholen\fortsetzen:
mdadm --run /dev/mdX
mdadm --grow --continue --backup-file=/root/md0.bak /dev/md0
Dann sollte der RAID-Verbund wieder zur Verfügung stehen.

Wichtig: Das System sollte man während der Erweiterung nicht neu starten, da eine abgebrochene Konvertierung nicht automatisch fortgesetzt wird. Vor dem konvertieren sollte man eine Datensicherung durchführen.
Weiterhin muss man nachdem die Konvertierung abgeschlossen ist, nochmals die Konfiguration aktualisieren - da das Hot-Spare-Laufwerk ja nun ein Daten- bzw. Parity-Laufwerk ist.

Es gibt noch verschiede andere Konvertierungsmöglichkeiten - es ist jedoch nicht alles möglich.
Bei Interesse sollte man die Dokumentation (manpages) lesen:
man mdadm
man mdadm.conf

20. RAID Integritätstest (checkarray):

Vorabinformationen:
Alle folgenden Zitate sind den manpages entnommen [man md].
Requesting a scrub will cause md to read every block on every device in the array, and check that the data is consistent.
For RAID1 and RAID10, this means checking that the copies are identical.
For RAID4, RAID5, RAID6 this means checking that the parity block is (or blocks are) correct.

Man kann demnach die Datenintegrität eines RAID-Verbundes testen, in dem man alle RAID-Verbund-Blöcke einliest und diese jeweils mit den dazu gehörigen Daten-Blöcken bzw. Parity-Blöcken vergleicht. Diesen Vorgang nennt man scrubbing.
Bei einem RAID 1 oder 10 Verbund wird hierbei verglichen, ob die jeweiligen zusammengehörigen Blöcke identisch sind.
Bei einem RAID 4, 5 oder 6 Verbund werden die jeweiligen Blöcke mit den dazugehörigen Daten- und Parity-Blöcken verglichen.

If a read error is detected during this process, the normal read-error handling causes correct data to be found from other devices
and to be written back to the faulty device. In many case this will effectively fix the bad block.

Tritt ein normaler Lesefehler auf (bad block), so wird das die Standard-Fehlerbehandlung durchgeführt, dabei werden die korrekten Daten anhand der anderen Laufwerke ermittelt und auf das fehlerhafte Laufwerk zurückgeschrieben. In manchen Fällen kann dies den Fehler beheben.

If all blocks read successfully but are found to not be consistent, then this is regarded as a mismatch.

Wenn alle Blöcke erfolgreich lesbar waren, der Inhalt jedoch inkonsistent (also unterschiedlich) ist, so werden diese als nichtübereinstimmend (mismatch) angesehen.

Für den Integritätstest wird das [checkarray]-Script verwendet.
Weitere Dokumentationen findet man über:
man md
md = multiple device (Linux Software Raid).
Insbesondere der Teil '**SCRUBBING AND MISMATCHES**' trifft hier zu.

bzw. über:
/usr/share/mdadm/checkarray --help
Hilfe zum checkarray Script anzeigen.

Beispielausgabe:
```
root@NAS:~# /usr/share//mdadm/checkarray --help
checkarray -- MD array (RAID) redundancy checker tool
Copyright © martin f. krafft <madduck@debian.org>
Released under the terms of the Artistic Licence 2.0

Usage: checkarray [options] [arrays]

Valid options are:
 -a|--all       check all assembled arrays (ignores arrays in command
line).
 -s|--status    print redundancy check status of devices.
 -x|--cancel    queue a request to cancel a running redundancy check.
 -i|--idle      perform check in a lowest scheduling class (idle)
 -l|--slow      perform check in a lower-than-standard scheduling class
 -f|--fast      perform check in higher-than-standard scheduling class
 --realtime     perform check in real-time scheduling class (DANGEROUS!)
 -c|--cron      honour AUTOCHECK setting in /etc/default/mdadm.
 -q|--quiet     suppress informational messages
                (use twice to suppress error messages too).
 -h|--help      show this output.
 -V|--version   show version information.

Examples:
 checkarray --all --idle
 checkarray --quiet /dev/md[123]
 checkarray -sa
 checkarray -x --all

Devices can be specified in almost any format. The following are
equivalent:
 /dev/md0, md0, /dev/md/0, /sys/block/md0
```

```
You can also control the status of a check with /proc/mdstat file.
```

Im Verzeichnis [/usr/share/doc/mdadm/] findet man noch eine Vielzahl anderer Dokumentationen.
Einige Dokumentationen liegen dort nur als Archiv vor, diese muss man zuvor entpacken (z. B. [gzip -d /usr/share/doc/mdadm/FAQ.gz]).
Insbesondere interessant ist folgende:
more /usr/share/doc/mdadm/README.checkarray

Beispielausgabe:
root@NAS:~# cat /usr/share/doc/mdadm/README.checkarray
checkarray notes
==========

checkarray will run parity checks across all your redundant arrays. By
default, it is configured to run on the first Sunday of each month, at 01:06
in the morning. This is realised by asking cron to wake up every Sunday with
/etc/cron.d/mdadm, but then only running the script when the day of the month
is less than or equal to 7. See #380425.

Cron will try to run the check at "idle I/O priority" (see ionice(1)), so that
the check does not overload the system too much. Note that this will only
work if all the component devices of the array employ the (default) "cfq" I/O
scheduler. See the kernel documentation[0] for information on how to verify
and modify the scheduler. checkarray does not verify this for you.

  0. http://www.kernel.org/doc/Documentation/block/switching-sched.txt

If you manually invoke checkarray, it runs with default I/O priority. Should
you need to run a check at a higher (or lower) I/O priority, then have a look
at the --idle, --slow, --fast, and --realtime options.

'check' is a read-only operation, even though the kernel logs may suggest
otherwise (e.g. /proc/mdstat and several kernel messages will mention
"resync"). Please also see question 21 of the FAQ.

If, however, while reading, a read error occurs, the check will trigger the
normal response to read errors which is to generate the 'correct' data and try
to write that out - so it is possible that a 'check' will trigger a write.
However in the absence of read errors it is read-only.

You can cancel a running array check with the -x option to checkarray.

 -- martin f. krafft <madduck@debian.org> Thu, 02 Sep 2010 10:27:29 +0200

Hier kann man verschiedenes herauslesen:

- Das jeden ersten Sonntag im Monat, gegen 01:06 Uhr, automatisch ein checkarray-Test, mit niedriger Priorität initiiert wird.

- Das in Punkt 21 der FAQ steht, das diese Überprüfung ggf. fälschlicherweise als [resync] bzw. [RebuildStarted event detected on md ...] erscheint, obwohl es nur ein check ist.
- Das normalerweise nur lesend zugegriffen wird.
- Und nur wenn ein Lesefehler auftreten sollte finden auch Schreibzugriffe statt, denn dann wird dieser gemäß der Standard-Fehlerbehandlung korrigiert (soweit möglich).

Dieser regelmäßige Test wird über folgenden cron-Job ausgeführt (die Uhrzeit weicht etwas ab):
cat /etc/cron.d/mdadm

Beispielausgabe:
root@NAS:~# cat /etc/cron.d/mdadm
#
cron.d/mdadm -- schedules periodic redundancy checks of MD devices
#
Copyright © martin f. krafft <madduck@madduck.net>
distributed under the terms of the Artistic Licence 2.0
#

By default, run at 00:57 on every Sunday, but do nothing unless the day of
the month is less than or equal to 7. Thus, only run on the first Sunday of
each month. crontab(5) sucks, unfortunately, in this regard; therefore this
hack (see #380425).
57 0 * * 0 root if [-x /usr/share/mdadm/checkarray] && [$(date +\%d) -le 7]; then
/usr/share/mdadm/checkarray --cron --all --idle --quiet; fi

Der Integritätstest wird demnach jeden ersten Sonntag im Monat, um 0:57 Uhr, mit niedriger Priorität, gestartet.

Diese Überprüfung kann man auch manuell starten, um z. B. neu installierte Systeme zu testen.

manuellen Test starten:
Der folgende Befehl initialisiert eine rein lesende Überprüfung (eine ggf. zuvor abgebrochene Überprüfung wird fortgesetzt):
echo check > /sys/block/md0/md/sync_action

If check was used, then no action is taken to handle the mismatch, it is simply recorded.

Werden dabei nicht übereinstimmende Blöcke gefunden, so werden diese nicht korrigiert sondern nur Protokolliert. Dabei wird der [mismatch_cnt]-Wert hoch gezählt (siehe unten). Weiterhin werden diese Fehler im syslog als [RebuildXX event detected on md device ...] protokolliert (XX kennzeichnet dabei, bei wie viel Prozent des Checks der Fehler auftrat).

Test beobachten:
Abhängig vom System bzw. der RAID-Größe kann eine Überprüfung - ähnlich wie ein rebuild - mehrere Stunden oder Tage dauern.
Den Test kann man wie folgt Überwachen:
watch -n2 cat /proc/mdstat

Beispielausgabe:

```
root@NAS:~# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0]
[raid1] [raid10]
md0 : active raid6 sdb1[3] sdd1[1] sdc1[2] sdf1[4] sde1[0]
      8790389760 blocks super 1.2 level 6, 512k chunk, algorithm 2 [5/5]
[UUUUU]
      [>....................]  check =  1.6% (48576396/2930129920)
finish=478.8min speed=100287K/sec
      bitmap: 0/22 pages [0KB], 65536KB chunk

unused devices: <none>
```

[check] kennzeichnet hierbei eindeutig die Überprüfung (also kein rebuild).

Test pausieren:
Die Überprüfung kann man vorzeitig pausieren (die Überprüfung kann später fortgesetzt werden), in dem man den ursprünglichen Wert [idle] zurück schreibt:
echo idle > /sys/block/md0/md/sync_action

Test auswerten:
Nachdem die Überprüfung abgeschlossen ist, kann man über diesen Befehl ausgeben viele Blöcke nicht mehr übereinstimmen:
cat /sys/block/md0/md/mismatch_cnt

Beispielausgabe:
root@NAS:~# cat /sys/block/md0/md/mismatch_cnt
0

0 = alles OK, alle Blöcke bzw. Parity-Prüfsummen passen korrekt zusammen.

Was ist nun zu tun bei einem Defekt?
A count of mismatches is recorded in the sysfs file md/mismatch_cnt. This is set to zero when a scrub starts and is incre-
mented whenever a sector is found that is a mismatch. md normally works in units much larger than a single sector and when
it finds a mismatch, it does not determine exactly how many actual sectors were affected but simply adds the number of sec-
tors in the IO unit that was used. So a value of 128 could simply mean that a single 64KB check found an error (128 x
512bytes = 64KB).
...
On a truly clean RAID5 or RAID6 array, any mismatches should indicate a hardware problem at some level - software issues
should never cause such a mismatch.

Beim Start der Überprüfung wird der [mismatch_cnt]-Wert zurückgesetzt und dann immer bei einer Nichtübereinstimmung von Blöcken erhöht. md arbeitet dabei normalerweise in Blöcken die größer als die physischen Laufwerkssektoren sind (hier wird denke ich die chunk-Größe herangezogen).
Tritt also ein Defekt auf, so kann man nicht eindeutig sagen wie viele Sektoren eines Laufwerks effektiv betroffen sind.

Wichtig ist aber:
Sollten hier also Fehler (Nichtübereinstimmungen) auftreten, so deutet dies meist auf verschiedene Hardwareprobleme (z. B. Laufwerksdefekte, Controllerfehler usw.) hin. Softwaredefekte sollte man in der Regel so gut wie ausschließen (siehe auch [man md]). Je nach Umgebung kann es Sinn machen den [mismatch_cnt]-Wert mit zu überwachen.

Reparaturmöglichkeiten bei Nichtübereinstimmung von Blöcken (mismatch):
If repair was used, then a mismatch will be repaired in the same way that resync repairs arrays.
For RAID5/RAID6 new parity blocks are written.
For RAID1/RAID10, all but one block are overwritten with the content of that one block.

Verwendet man repair, so werden Nichtübereinstimmungen (mismatch) genau wie bei einem resync des RAID-Verbundes (array) repariert.
Bei einem RAID 5 oder 6 Verbund werden hier neue Parity-Blöcke geschrieben.
Bei einem RAID 1 oder 10 Verbund wird ein Block mit den Daten des anderen Block's überschrieben.

Wichtig: Abhängig vom RAID Level kann dies jedoch gefährlich sein.
Bei einem RAID 1 (alle Daten liegen doppelt vor) kann man z. B. nicht zu 100% sagen welcher Sektor nun korrekte bzw. defekte Daten liefert.

In einem RAID 6 Verbund kann man normalerweise dank zweifacher Parity, einen einfachen Parity Fehler exakt lokalisieren und beheben. Dies wird jedoch derzeit von Linux Software RAID noch nicht direkt unterstützt (wo sind Entwickler die das beheben?). Stattdessen wird einfach davon ausgegangen, dass die Parity-Prüfsummen inkorrekt sind und es werden einfach ohne weitere Tests nur neue Parity-Blöcke geschrieben.

repair ist also sehr eingeschränkt.
Die Daten können danach durchaus wieder lesbar sein, allerdings können einige Blöcke durchaus inkonsistente/falsche Daten enthalten. Man sollte daher vor einem Reparaturversuch unbedingt eine Datensicherung durchführen.

Die Reparatur starten:
echo repair > /sys/block/md0/md/sync_action

raid6check:
Sofern man ein RAID 6 Verbund verwendet kann man ggf. auch raid6check verwenden. raid6check ist ein optionales ein Paket (Programm) (GitHub SourceCode) mit dem man exakt herausfinden kann welches Laufwerk den Parity-Fehler verursacht. raid6check ist allerdings ebenso nicht perfekt. Zum einen wird raid6check nicht Standardmäßig mit dem mdadm Paket installiert (unter Ubuntu gibt es kein fertiges Paket). Weiterhin kann raid6check gefundene Fehler nicht korrigieren. So das der Nutzen von raid6check relativ gering ist. Allerdings kann man hiermit das fehlerhafte Laufwerk ermitteln und austauschen (dies scheint mir sinnvoller als ein repair).
raid6check /dev/md0 0 0 | grep -i error > md0_err.log
Testet den angegebenen Verbund komplett.
Ein Logfile wird nur bei einem Fehler erzeugt.

21. RAID-Verbund komplett auflösen:

Einen RAID-Verbund bzw. den Superblock kann man nur auflösen bzw. löschen, wenn vom System her keine Schreibzugriffe mehr stattfinden.
Das bedeutet man muss zuvor alle Dienste und Prozesse die Zugriffe auf den RAID-Verbund ermöglichen beenden (z. B. [service samba stop] bzw. [kill -9 ProzessID]).
Des Weiteren muss man sicherstellen, das der RAID-Verbund nicht mehr eingehängt/gemountet ist.
RAID-Verbund aushängen:
umount /dev/md0

Anzeigen: Welcher Prozess bzw. welcher Service greift noch auf den RAID-Verbund zu? Nur nötig falls umount [target is busy] meldet:
lsof /dev/md0

Den RAID-Verbund stoppen:
mdadm --stop /dev/md0

Hinweis: Alle für den RAID-Verbund verwendeten Partitionen (Laufwerke) enthalten weiterhin alle Daten und RAID-Informationen.
Will man auf diesen Partitionen (Laufwerken) später einen neuen RAID-Verbund erstellen, so wird der alte RAID-Verbund erkannt und es erscheint eine Sicherheitsrückfrage ob man die Partitionen (Laufwerke) wirklich verwenden will.

Will man diese RAID-Informationen endgültig löschen, so muss man auf allen betroffenen Partitionen (Laufwerken) den Superblock löschen:
mdadm --zero-superblock /dev/sdX1

Wichtig:
Das löschen von Superblöcken von nicht mehr verwendeten Partitionen (Laufwerken) sollte man unbedingt vornehmen!
Denn es bestehen folgende Gefahren:
1. Verwendet man diese Laufwerke später im selben System wieder, so wird dieses Laufwerk (mit alten Daten) möglicherweise dem RAID-Verbund wieder zugewiesen, das wiederum kann dann ggf. alle Daten zerstören.
2. Verwendet man die Laufwerke in einem anderen System, so können RAID-Verbund Reste als [degraded] RAID-Verbund erscheinen.

Zuletzt sollte man die mdadm.conf-Konfiguration bearbeiten und die RAID-Verbund Einträge entfernen (einfach die zwei [ARRAY /dev/md/0 ...] Zeilen am Ende löschen oder auskommentieren).
Weiterhin muss man ggf. noch den mount Eintrag in der Datei [/etc/fstab] entfernen.

22. RAID-Benchmark:

Den RAID-Verbund sollte man zuerst mounten - danach wechselt man in den Pfad wo der

RAID-Verbund eingehängt ist.

Tipp: Beachtet bei Benchmark-Messungen mit einem ext4-Dateisystem, dass die lazy initialization (lazy-init) zuvor abgeschlossen sein muss (wenn man korrekte Messwerte benötigt).

Für erste Tests kann man den Befehl dd verwenden:
In die Testdatei schreiben:
dd if=/dev/zero of=Testdatei bs=1M count=1000
dd if=/dev/zero of=Testdatei bs=1k count=1000

Von der Testdatei lesen:
dd if=Testdatei of=/dev/zero bs=1k count=1000

dd Benchmark - Beispielausgabe von einem RAID 6 Verbund:
Die ermittelten Werte sind Idealwerte, da alles nur in eine einzige Zieldatei geschrieben wird, bzw. von dort gelesen wird.
In der Praxis, insbesondere über ein Netzwerk, fallen die Werte entsprechend anders (niedriger) aus.
Für einen Vergleich zwischen verschiedenen RAID-Varianten (auf ein und demselben System) kann man diese jedoch gut verwenden.

In die Testdatei schreiben:
root@NAS:/mnt/raid# dd if=/dev/zero of=Testdatei bs=1k count=1000
1000+0 Datensätze ein
1000+0 Datensätze aus
1024000 Bytes (1,0 MB) kopiert, 0,0547824 s, **18,7 MB/s**

root@NAS:/mnt/raid# dd if=/dev/zero of=Testdatei bs=4k count=1000
1000+0 Datensätze ein
1000+0 Datensätze aus
4096000 Bytes (4,1 MB) kopiert, 0,0693092 s, **59,1 MB/s**

root@NAS:/mnt/raid# dd if=/dev/zero of=Testdatei bs=1M count=1000
1000+0 Datensätze ein
1000+0 Datensätze aus
1048576000 Bytes (1,0 GB) kopiert, 6,29682 s, **167 MB/s**

root@NAS:/mnt/raid# dd if=/dev/zero of=Testdatei bs=2M count=10000
10000+0 Datensätze ein
10000+0 Datensätze aus
20971520000 Bytes (21 GB) kopiert, 98,4852 s, **213 MB/s**

Von der Testdatei (1,0 GB groß) lesen, der Cache wirkt hier mit:
root@NAS:/mnt/raid# dd if=Testdatei of=/dev/zero bs=1k count=1000
1000+0 Datensätze ein
1000+0 Datensätze aus
1024000 Bytes (1,0 MB) kopiert, 0,0064112 s, **160 MB/s**

root@NAS:/mnt/raid# dd if=Testdatei of=/dev/zero bs=2M count=100
100+0 Datensätze ein

100+0 Datensätze aus
209715200 Bytes (210 MB) kopiert, 0,12679 s, **1,7 GB/s**

hdparm Benchmark - von einem RAID 6 Verbund:
hdparm -tT /dev/sdX1 # ungepufferter und gepufferter Benchmark auf einem einzelnen Laufwerk:

Beispielausgabe von einem RAID 6 Verbund:
root@NAS:~# hdparm -tT /dev/md0

/dev/md0:
Timing cached reads: 2928 MB in 2.00 seconds = **1464.09 MB/sec**
Timing buffered disk reads: 1230 MB in 3.00 seconds = **409.53 MB/sec**

23. RAID Tuning Tipps:

Es gibt im Netz viele verschiede Tipps. Je nach verwendetem System kann der ein oder andere Tipp eine Steigerung der Geschwindigkeit bewirken.
Ein Allgemeingültiges Rezept gibt es hier leider nicht.
Auf unserem RAID Server mit Intel Pentium N3700, 4x 1.60GHz, 6W TDP mit extra PCI-E-SATA-Controllerkarte konnte ich trotz intensiver Tests bei einem RAID 6 Verbund keinerlei nennenswerte Geschwindigkeitssteigerungen feststellen, so dass ich keine dieser Funktionen verwendete.
Wer es trotzdem versuchen möchte kann mit den unten angegebenen Werten verschiede Einstellungen testen.

Vorerst wäre anzumerken, dass all die unten aufgeführten Einstellungen nach einem Neustart immer zurückgesetzt werden.
Man sollte diese daher in ein Script speichern und dieses bei einem Neustart entsprechend mit starten.

nano turbo.sh

#!/bin/sh
 # rebuild/resync Geschwindigkeitslimit erhöhen:
mdadm setzt selbstständig Begrenzungen um das System nicht zu überlasten.
Aktuelle Werte anzeigen:
cat /proc/sys/dev/raid/speed_limit_max # 200000 ist oft Standard.
cat /proc/sys/dev/raid/speed_limit_min # 1000 ist oft Standard.
Diese Werte kann man individuell erhöhen:
echo 250000 > /proc/sys/dev/raid/speed_limit_max # Zu hohe max-Werte können das System blockieren.
echo 50000 > /proc/sys/dev/raid/speed_limit_min

read ahead erhöhen:
Es wird hierbei die Anzahl der im Voraus zu lesenden Blöcke angegeben.
32768 * 512 Bytes je Sektor = 16 MiB
65536 * 512 Bytes je Sektor = 32 MiB (max.)

Aktuelle Einstellungen anzeigen:
blockdev --report /dev/sdX # 256 ist oft Standard.
blockdev --setra 4096 /dev/sdX # **<- für alle RAID-Verbund Laufwerke 2 MiB**
blockdev --setra 32768 /dev/md0

stripe cache erhöhen:
Aktuellen Wert anzeigen:
cat /sys/block/md0/md/stripe_cache_size
256 ist meist Standard bis 32768 max.
echo 32768 > /sys/block/md0/md/stripe_cache_size

Die Scrip-Datei muss man einmalig ausführbar machen, dann kann man diese wie folgt starten:
chmod +x turbo.sh
./turbo.sh

24. Logfile überprüfen & Fehlerbehebung:

Auch wenn oft alles scheinbar ohne Fehler läuft, so kann es nie schaden mal einen Blick in die System-Logdateien ([/var/log/syslog] oder [/var/log/kern.log]) zu wagen.

Diese Logfiles sollte man nach typischen Fehlermeldungen durchsuchen z. B.:
cat /var/log/syslog | grep failed
cat /var/log/syslog | grep Error
cat /var/log/syslog | grep sector

Hinweis: Nicht jede Ausgabe deutet gleich auf einen Fehler hin.

Ein [... ataX.00: error: { UNC } ...] (uncorrectable error) weist jedoch meist auf defekte Sektoren hin.

Weitere Beispiele für einen Fehler:
...
Jan 19 10:20:10 NAS systemd[1]: Started Disk Manager.
Jan 19 10:20:10 NAS kernel: [459.052555] ata6.00: configured for UDMA/133
Jan 19 10:20:10 NAS kernel: [459.052565] ata6: EH complete
Jan 19 10:20:10 NAS udisksd[1331]: Acquired the name org.freedesktop.UDisks2 on the system message bus
Jan 19 10:21:44 NAS kernel: [553.141165] EXT4-fs (md0): mounted filesystem with ordered data mode. Opts: (null)
Jan 19 10:27:31 NAS kernel: [899.845973] ata3.00: exception Emask 0x0 SAct 0x7fffffff SErr 0x400000 action 0x6 frozen
Jan 19 10:27:31 NAS kernel: [899.846000] ata3: SError: { Handshk }
Jan 19 10:27:31 NAS kernel: [899.846018] ata3.00: failed command: WRITE FPDMA QUEUED
Jan 19 10:27:31 NAS kernel: [899.846049] ata3.00: cmd 61/40:00:10:7e:cc/05:00:01:00:00/40 tag 0 ncq 688128 out
Jan 19 10:27:31 NAS kernel: [899.846049] res 40/00:00:00:00:00/00:00:00:00:00/00 Emask

0x4 (timeout)
Jan 19 10:27:31 NAS kernel: [899.846067] ata3.00: status: { DRDY }
...
Jan 19 10:27:31 NAS kernel: [900.341908] ata3: EH complete
Jan 19 10:27:59 NAS systemd[1]: Starting Cleanup of Temporary Directories...
Jan 19 10:27:59 NAS systemd-tmpfiles[1529]: [/usr/lib/tmpfiles.d/var.conf:14] Duplicate line for path "/var/log", ignoring.
Jan 19 10:27:59 NAS systemd[1]: Started Cleanup of Temporary Directories.
Jan 19 10:29:15 NAS kernel: [1003.840959] ata3.00: exception Emask 0x0 SAct 0x0 SErr 0x400000 action 0x6 frozen
Jan 19 10:29:15 NAS kernel: [1003.840970] ata3: SError: { Handshk }
Jan 19 10:29:15 NAS kernel: [1003.840976] ata3.00: failed command: WRITE DMA EXT
Jan 19 10:29:15 NAS kernel: [1003.840986] ata3.00: cmd 35/00:60:a0:5c:c5/00:03:01:00:00/e0 tag 3 dma 442368 out
Jan 19 10:29:15 NAS kernel: [1003.840986] res 40/00:00:00:4f:c2/00:00:00:00:00/00 Emask 0x4 (timeout)
Jan 19 10:29:15 NAS kernel: [1003.840991] ata3.00: status: { DRDY }
Jan 19 10:29:15 NAS kernel: [1003.840998] ata3: hard resetting link
Jan 19 10:29:15 NAS kernel: [1004.332819] ata3: SATA link up 6.0 Gbps (SStatus 133 SControl 300)
Jan 19 10:29:15 NAS kernel: [1004.335440] ata3.00: configured for UDMA/133
Jan 19 10:29:15 NAS kernel: [1004.335456] ata3.00: device reported invalid CHS sector 0
Jan 19 10:29:15 NAS kernel: [1004.335475] ata3: EH complete
...

Diese Fehler hier ließen z. B. den SMART-Wert [199 UDMA_CRC_Error_Count] eines Laufwerks ständig Ansteigen.
Ursache war hier kein Festplattendefekt, sondern ein Übertragungsfehler zwischen der Festplatte und dem Controller.
Es muss also nicht immer gleich ein Laufwerksdefekt vorliegen. Ein Übertragungsfehler kann durch verschiedene Störungen hervorgerufen werden.

Man sollte insbesondere folgende Punkte prüfen:

- Wurden ungeeignete, geknickte, gequetschte oder anderweitig beschädigte SATA-Kabel/Datenkabel verwendet?
- Besteht die Möglichkeit des Übersprechens? Liegen SATA-Datenkabel mit anderen störenden Kabeln, Lüftern usw. nahe beieinander?
- Stecken alle SATA-Stromversorgungsstecker wirklich korrekt? Sind die Kontakte sauber?
- Steckt die Controllerkarte im PCIE-Slot wirklich komplett drin, oder wird diese möglicherweise beim verschrauben des Slotblech leicht herausgezogen?
- Hat man eine Controllerkarte verbaut bei der man Anschlüsse Jumpern kann (z. B. externes SATA auf internes SATA), so sollte man hier auf sehr guten Kontakt der Jumper achten.
- Natürlich kann letztendlich auch ein Laufwerkdefekt vorliegen. Bei stärker defekten Festplatten hört man hier jedoch meist ein regelmäßiges klackern und mdadm würde diesen defekt melden.

Hintergrundinformationen:
Serial ATA 6.0 Gbit/s, SATA Revision 3.x oder auch einfach nur SATA 6 Gbps genannt

überträgt bis zu 600 Mbytes/s. Bei dieser enormen Menge an Daten können geringste Übertragungsstörungen durch ungeeignete bzw. schlechte Verkabelung schnell zu einem Fehler führen. Eine fehlerhafte Übertragung wird in der Regel durch das System erkannt protokolliert und so lange wiederholt bis die Datenübertragung sauber abgeschlossen ist. Die Übertragungsstörung auf dem SATA-Datenkabelweg führt also nicht zu einem Datenfehler, daher meldet mdadm hier auch keinen Fehler. Allerdings kann die Datenübertragungsrate darunter leiden. Weiterhin wird das syslog und kernel.log aufgebläht, da derartige Fehler alle protokolliert werden.

Bei dem Fehler im Beispiel brachte ein anfänglicher SATA-Datenkabel tausch keine Abhilfe. Erst nach dem ich alle Anschlüsse nochmals überprüfte und die Jumperanschlüsse auf dem Controller reinigte und neu steckte trat der Fehler nicht mehr auf. Ein Anschließender Intensivtest der Festplatte meldete danach keinen Fehler mehr.

SATA-Geschwindigkeit begrenzen:

Sollte all dies nicht helfen, so kann man alternativ auch die SATA-Geschwindigkeit für einzelne Laufwerke herabsetzen. Man sollte sich hier nach dem langsamsten verbauten Laufwerk richten.

Man öffnet hierzu folgende Datei mit einem Editor:
nano /etc/default/grub

Hier ändert den folgenden Eintrag so ab:
...
GRUB_CMDLINE_LINUX="vga=0x0314 bootdegraded=true
libata.force=2:1.5G,3:1.5G,4:1.5G,5:1.5G,6:1.5G"
...
Die Zahl vor dem Doppelpunkt kennzeichnet das entsprechende Laufwerk (beim Zählen wird mit 1 begonnen).
Nach dem Doppelpunkt wird dem entsprechenden Laufwerk die jeweilige Geschwindigkeit (z. B. 1.5G, 3.0G, 6.0G) zugewiesen.
Weitere Laufwerke werden einfach nacheinander mit Komma getrennt benannt.
#
Optional: [vga=0x0314] legt die Konsolenauflösung auf 800x600 fest.

Die neuen Einstellungen muss man dem System zunächst bekannt machen:
update-grub

Die Einstellungen werden erst nach einem Neustart übernommen:
reboot

Anschließend sollte man überprüfen, ob die Einstellungen übernommen wurden:
dmesg | grep -i sata | grep 'link up'

Beispielausgabe:
root@NAS:~# dmesg | grep -i sata | grep 'link up'
[2.211776] ata1: SATA link up 1.5 Gbps (SStatus 113 SControl 300)

[2.219774] ata2: SATA link up 1.5 Gbps (SStatus 113 SControl 310)
[2.391836] ata3: SATA link up 1.5 Gbps (SStatus 113 SControl 310)
[2.395445] ata5: SATA link up 1.5 Gbps (SStatus 113 SControl 310)
[2.399365] ata4: SATA link up 1.5 Gbps (SStatus 113 SControl 310)
[2.399399] ata6: SATA link up 1.5 Gbps (SStatus 113 SControl 310)

oder einzelne Laufwerke abfragen:
smartctl -i /dev/sdX | grep SATA

Beispielausgabe:
root@NAS:~# smartctl -i /dev/sdb | grep SATA
SATA Version is: SATA 3.0, 6.0 Gb/s (current: 1.5 Gb/s)

Hier erkennt man beispielsweise das, dass eigentlich 6.0 Gb/s schnelle Laufwerk auf 1.5 Gb/s gedrosselt wird.

25. ata /dev/sd zuordnen:

In System-Logfiles werden Fehler oft nur mit einer ATA-Kennung z. B. [ataX.00...] aufgelistet.
Diesen ATA-Anschluss muss man nun dem entsprechenden Gerät [/dev/sdX] zuordnen.

Auflistung der Geräte (ata device correlate corresponds):
ls -l /sys/block/sd*

Beispielausgabe:
root@NAS:~# ls -l /sys/block/sd*
lrwxrwxrwx 1 root root 0 Feb 11 11:01 /sys/block/sda ->
../devices/pci0000:00/0000:00:13.0/ata1/host0/target0:0:0/0:0:0:0/block/sda
lrwxrwxrwx 1 root root 0 Feb 11 11:01 /sys/block/sdb ->
../devices/pci0000:00/0000:00:13.0/ata2/host1/target1:0:0/1:0:0:0/block/sdb
lrwxrwxrwx 1 root root 0 Feb 11 11:01 /sys/block/sdc ->
../devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata3/host2/target2:0:0/2:0:0:0/block/sdc
lrwxrwxrwx 1 root root 0 Feb 11 11:01 /sys/block/sdd ->
../devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata4/host3/target3:0:0/3:0:0:0/block/sdd
lrwxrwxrwx 1 root root 0 Feb 11 11:01 /sys/block/sde ->
../devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata5/host4/target4:0:0/4:0:0:0/block/sde
lrwxrwxrwx 1 root root 0 Feb 11 11:01 /sys/block/sdf ->
../devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata6/host5/target5:0:0/5:0:0:0/block/sdf

Zur Auflistung kann man alternativ auch das Programm lsscsi verwenden:
apt-get install lsscsi
lsscsi -s
lsscsi -sv

Beispielausgabe:
```
root@NAS:~# lsscsi -sv
[0:0:0:0]    disk    ATA      WDC WD360GD-00FL 8F31  /dev/sda   37.0GB
```

```
  dir: /sys/bus/scsi/devices/0:0:0:0
[/sys/devices/pci0000:00/0000:00:13.0/ata1/host0/target0:0:0/0:0:0:0]
[1:0:0:0]    disk    ATA      WDC WD30EFRX-68E 0A82  /dev/sdb   3.00TB
  dir: /sys/bus/scsi/devices/1:0:0:0
[/sys/devices/pci0000:00/0000:00:13.0/ata2/host1/target1:0:0/1:0:0:0]
[2:0:0:0]    disk    ATA      WDC WD30EFRX-68E 0A82  /dev/sdc   3.00TB
  dir: /sys/bus/scsi/devices/2:0:0:0
[/sys/devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata3/host2/target2:0:0/2
:0:0:0]
[3:0:0:0]    disk    ATA      WDC WD30EFRX-68E 0A82  /dev/sdd   3.00TB
  dir: /sys/bus/scsi/devices/3:0:0:0
[/sys/devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata4/host3/target3:0:0/3
:0:0:0]
[4:0:0:0]    disk    ATA      WDC WD30EFRX-68E 0A82  /dev/sde   3.00TB
  dir: /sys/bus/scsi/devices/4:0:0:0
[/sys/devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata5/host4/target4:0:0/4
:0:0:0]
[5:0:0:0]    disk    ATA      WDC WD30EFRX-68E 0A82  /dev/sdf   3.00TB
  dir: /sys/bus/scsi/devices/5:0:0:0
[/sys/devices/pci0000:00/0000:00:1c.0/0000:01:00.0/ata6/host5/target5:0:0/5
:0:0:0]
```

Die Hervorhebungen in dem Beispiel bedeuten folgendes:
[00] ist der onboard-Controller (hier sind 2 Laufwerke angeschlossen).
[01] ist ein extra PCIE-Controller (hier sind weitere 4 Laufwerke angeschlossen).
Daher ist hier die Darstellung ab dem 3. Laufwerk (sdc) immer leicht nach rechts verschoben.

26. SMART-Werte eines Laufwerks überprüfen:

Die reinen SMART-Werte (S.M.A.R.T.) kann man sich wie folgt anzeigen lassen:
Nur Basic Informationen anzeigen wie z. B. die Seriennummern:
smartctl -i /dev/sdX
Alle Informationen anzeigen, einschließlich der SMART-Werte:
smartctl -a /dev/sdX

Beispielausgabe:

```
smartctl 6.4 2015-06-04 r4109 [x86_64-linux-4.4.0-2-generic] (local build)
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF INFORMATION SECTION ===
Model Family:     Western Digital Red
Device Model:     WDC WD30EFRX-68EUZN0
Serial Number:    WD-WCC4N3USVJCY
LU WWN Device Id: 5 0014ee 20cdff81d
Firmware Version: 82.00A82
User Capacity:    3.000.592.982.016 bytes [3,00 TB]
Sector Sizes:     512 bytes logical, 4096 bytes physical
Rotation Rate:    5400 rpm
Device is:        In smartctl database [for details use: -P show]
ATA Version is:   ACS-2 (minor revision not indicated)
SATA Version is:  SATA 3.0, 6.0 Gb/s (current: 6.0 Gb/s)
Local Time is:    Thu Feb  4 16:45:34 2016 CET
SMART support is: Available - device has SMART capability.
SMART support is: Enabled
```

```
=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED

General SMART Values:
Offline data collection status:  (0x00)       Offline data collection
activity
                                        was never started.
                                        Auto Offline Data Collection:
Disabled.
Self-test execution status:      (   0)       The previous self-test
routine completed
                                        without error or no self-test has
ever
                                        been run.
Total time to complete Offline
data collection:                (41640) seconds.
Offline data collection
capabilities:                    (0x7b) SMART execute Offline immediate.
                                        Auto Offline data collection on/off
support.
                                        Suspend Offline collection upon new
                                        command.
                                        Offline surface scan supported.
                                        Self-test supported.
                                        Conveyance Self-test supported.
                                        Selective Self-test supported.
SMART capabilities:            (0x0003)       Saves SMART data before
entering
                                        power-saving mode.
                                        Supports SMART auto save timer.
Error logging capability:        (0x01)       Error logging supported.
                                        General Purpose Logging supported.
Short self-test routine
recommended polling time:        (   2) minutes.
Extended self-test routine
recommended polling time:        ( 418) minutes.
Conveyance self-test routine
recommended polling time:        (   5) minutes.
SCT capabilities:              (0x703d) SCT Status supported.
                                        SCT Error Recovery Control supported.
                                        SCT Feature Control supported.
                                        SCT Data Table supported.

SMART Attributes Data Structure revision number: 16
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED
WHEN_FAILED RAW_VALUE
  1 Raw_Read_Error_Rate     0x002f   200   200   051    Pre-fail  Always
-       0
  3 Spin_Up_Time            0x0027   189   182   021    Pre-fail  Always
-       5525
  4 Start_Stop_Count        0x0032   100   100   000    Old_age   Always
-       51
  5 Reallocated_Sector_Ct   0x0033   200   200   140    Pre-fail  Always
-       0
  7 Seek_Error_Rate         0x002e   200   200   000    Old_age   Always
-       0
  9 Power_On_Hours          0x0032   100   100   000    Old_age   Always
-       428
 10 Spin_Retry_Count        0x0032   100   253   000    Old_age   Always
-       0
 11 Calibration_Retry_Count 0x0032   100   253   000    Old_age   Always
-       0
```

```
 12 Power_Cycle_Count       0x0032   100   100   000    Old_age   Always
-       50
192 Power-Off_Retract_Count 0x0032   200   200   000    Old_age   Always
-       15
193 Load_Cycle_Count        0x0032   200   200   000    Old_age   Always
-       410
194 Temperature_Celsius     0x0022   114   107   000    Old_age   Always
-       36
196 Reallocated_Event_Count 0x0032   200   200   000    Old_age   Always
-       0
197 Current_Pending_Sector  0x0032   200   200   000    Old_age   Always
-       0
198 Offline_Uncorrectable   0x0030   100   253   000    Old_age   Offline
-       0
199 UDMA_CRC_Error_Count    0x0032   200   200   000    Old_age   Always
-       0
200 Multi_Zone_Error_Rate   0x0008   100   253   000    Old_age   Offline
-       0

SMART Error Log Version: 1
No Errors Logged

SMART Self-test log structure revision number 1
Num  Test_Description    Status                  Remaining  LifeTime(hours)
LBA_of_first_error
# 1  Short offline       Completed without error       00%        15
-

SMART Selective self-test log data structure revision number 1
 SPAN  MIN_LBA  MAX_LBA  CURRENT_TEST_STATUS
    1        0        0  Not_testing
    2        0        0  Not_testing
    3        0        0  Not_testing
    4        0        0  Not_testing
    5        0        0  Not_testing
Selective self-test flags (0x0):
  After scanning selected spans, do NOT read-scan remainder of disk.
If Selective self-test is pending on power-up, resume after 0 minute delay.
```

Parametererklärungen:
[VALUE] = aktueller Wert.
[WORST] = bisher schlechtester Wert.
[THRESH] = Grenzwert
[TYPE] = bei Grenzwertunterschreitung, droht bei [Pre-fail] ein baldiger Ausfall / [Old-age] kennzeichnet manchmal auch nur verschiedene Werte wie z. B. die Temperatur.

Kritisch sind also nur Einträge bei denen die Werte [VALUE] bzw. [WORST] kleiner sind als der Grenzwert [THRESH].
Ist der dazugehörige Eintrag weiterhin noch mit [Pre-fail] gekennzeichnet, so droht ein Ausfall in kürzester Zeit.
Man sollte dann das entsprechende Laufwerk schnellstmöglich austauschen.
Hier im Beispiel ist das Laufwerk also in Ordnung, da alle Werte deutlich über dem Grenzwert [THRESH] liegen.

27. Laufwerk austauschen (SMART-Fehler):

Sollte ein Laufwerk sehr schlechte SMART-Werte haben (ein baldiger Ausfall droht), so sollte man dieses Laufwerk ersetzen.
Wenn der Zugriff auf das Laufwerk noch normal und ohne Fehler möglich ist, so kann man dieses Laufwerk wie folgt sicher austauschen.

Tipp:
replace kann man auch verwenden, wenn man relativ sicher nach und nach alle Laufwerke eines RAID-Verbundes wechseln möchte, um z. B. den Speicherplatz zu erhöhen.

1. Zuerst muss man dem RAID-Verbund ein Hot-Spare-Laufwerk hinzufügen.

2. Danach initialisiert man den Austausch (replace) wie folgt:
mdadm --replace /dev/md0 /dev/sdX1
watch -n1 cat /proc/mdstat

Dies setzt auf dem angegebenen Laufwerk eine Kennung das dieses Laufwerk ersetzt werden soll.
Sobald nun ein Hot-Spare-Laufwerk zur Verfügung steht, startet ein automatischer rebuild des angegebenen Laufwerks auf das Hot-Spare-Laufwerk.
Dies ähnelt dem [faulty]-Status jedoch mit dem Unterschied, dass das angegebene Laufwerk während des Rebuild-Prozesses weiterhin im RAID-Verbund verbleibt.
Erst nachdem der rebuild erfolgreich abgeschlossen wurde ändert sich der Status des angegebenen Laufwerks auf [faulty].
Der Vorteil der replace Lösung ist also:
Während des replace, ändert sich der Status vom RAID-Verbund nur auf [clean, resyncing].
Weitere Laufwerksausfälle während des rebuilds sind hier somit weniger kritisch.

3. Und erst dann kann man das fehlerhafte Laufwerk aus dem RAID-Verbund endgültig entfernen.
mdadm --remove /dev/md0 /dev/sdX1

4. Wichtig: Sollte man das Laufwerk nicht aus dem System ausbauen, so muss man den Superblock vom ausgetauschten Laufwerk löschen.

5. Zuletzt sollte man nicht vergessen die mdadm.conf zu aktualisieren.

28. Laufwerktest mit smartctl:

Ein Hinweis vorab:
Alle Testergebnisse werden dauerhaft auf dem Laufwerk gespeichert.
Man sollte diese Tests also möglichst nur in großem zeitlichen Abstand oder wenn man einen Defekt vermutet durchführen.

Will man mit smartctl einen Test der Laufwerke durchführen, so geht man wie folgt vor:
Schritt 1: Den Test initialisieren:
smartctl -t short /dev/sdX
Hinweis: Bevor man das Ergebnis aufrufen kann muss man dann erst ca. 2 Minuten warten!

smartctl -t long /dev/sdX
Hinweis: Dieser Test dauert je nach System- und Laufwerksgeschwindigkeit mind. 20 Minuten.

Schritt 2: Das Testergebnis anzeigen.
smartctl -l selftest /dev/sdX

Beispielausgabe:

```
root@NAS:~# smartctl -t short /dev/sdb
smartctl 6.4 2015-06-04 r4109 [x86_64-linux-4.4.0-2-generic] (local build)
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Execute SMART Short self-test routine immediately in off-
line mode".
Drive command "Execute SMART Short self-test routine immediately in off-
line mode" successful.
Testing has begun.
Please wait 2 minutes for test to complete.
Test will complete after Thu Feb  4 17:00:04 2016

Use smartctl -X to abort test.
root@NAS:~# smartctl -l selftest /dev/sdb
smartctl 6.4 2015-06-04 r4109 [x86_64-linux-4.4.0-2-generic] (local build)
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART Self-test log structure revision number 1
Num  Test_Description    Status                  Remaining  LifeTime(hours)
LBA_of_first_error
# 1  Short offline       Completed without error       00%        15
-
```

29. Laufwerktest mit badblocks:

Ab der Superblock Version 1.x werden Informationen über defekte Sektoren im Superblock abgelegt.
Anzeigen kann man diese wie folgt:
mdadm --examine-badblocks /dev/sdX1

Intensiver kann man die Laufweke mit badblocks testen.
badblocks überprüft auf Speichermedien wie z. B. Festplatten alle Sektoren auf defekte. Je nach Laufwerksgröße kann der Test schnell mehrere Tage dauern.
Hat man ein defektes Laufwerk ermittelt, so kann man dieses Laufwerk bzw. die Partition wie folgt überprüfen:
Das Laufwerk (**n**on-destructive **s**how **v**erbose) überprüfen.
Es wird hierbei der original Sektor gesichert, ein schreibest durchgeführt und am ende der zuvor gesicherte Sektor wieder zurückgeschrieben.
Es wird hier nur ein einfacher (single-pass) Test durchgeführt und die Daten des Laufwerks bleiben erhalten:

badblocks -nsv /dev/sdX

Hier wird jeder Sektor des Laufwerks wird mit verschiedenen Mustern beschrieben.
Standardmäßig: 0xaa (10101010), 0x55 (01010101), 0xff (11111111) und 0x00 (00000000).
Vorsicht: Dabei gehen vorhandenen Daten auf dem Laufwerk verloren (**w**rite **s**how **v**erbose):
badblocks -wsv /dev/sdX
badblocks -wsv -o defekte-Sektoren.txt /dev/sdX
Der Test einer WD Red 3 TB Festplatte dauerte über 4 Tage.

❑ Abschließende Hinweise:

Diese Dokumentation behandelt keine RAID-Systeme von denen man direkt booten kann, es geht hier hauptsächlich um einen extra RAID-Verbund für Daten. Des Weiteren werden nur neue GPT (GUID Partition Table) Systeme behandelt - alte MBR (Master-Boot-Record) Systeme werden hier zum großen Teil ausgelassen.
Natürlich lassen sich trotzdem eine Vielzahl der gezeigten Lösungen 1:1 für ältere und andere Systeme übernehmen.
Ein Großteil der Programmaufrufe setzt aufgrund der systemnahen Zugriffe zwingend einen direkten root Zugang voraus. Sie können alternativ aber auch alle Befehle mit einem führendem [sudo Programm] versehen.
IP-Adressen, E-Mailadressen, Namen u. ä. wurden für die Dokumentation geändert, hacken ist also zwecklos.
Die Nutzung der Anleitung erfolgt auf eigene Gefahr, für jegliche Schäden wird keine Garantie/Haftung übernommen.
Die Dokumentation entstand aus verschiedenen Tests unter Ubuntu 14.04 bis 22.04, Debian 7 bis 11, sowie produktiven Installationen. Diese Anleitung stellt somit eine Zusammenfassung wichtiger und empfohlener Schritte dar.
Bevor sie eventuell Fragen stellen bitte ich sie die Dokumentation komplett zu lesen.
Hinweise auf Fehler, Anregungen, Danksagungen oder ähnliches sind immer willkommen.

Design: