

ITSI 1. Test Zusammenfassung (sogar auf Deutsch)

Backup und Recovery

Backup vs. Disaster Recovery:

Backup: Sicherung von Daten (RAID != BACKUP)

Disaster Recovery: Der ganze auszuführende Prozess im Falle eines Disasters (Backup zurückspielen, etc.)

RAID: (Redundant array of independent disks) (Ausfallsichere Sammlung von unabhängigen Festplatten) https://en.wikipedia.org/wiki/Standard_RAID_levels KEIN BACKUP!

Raid 0: Stripe (keine Ausfallsicherheit/Redundanz, schnell)

Daten werden auf mehreren Festplatten verteilt. Bsp.: 2x 2TB Platten = ein 4TB Volume

Raid 1: Mirror (Spiegelung) Gleiche Daten werden auf mehreren Festplatten doppelt gesichert.

Speed Bonus nur beim Lesen da von beiden Platten gleichzeitig gelesen werden kann.

Einfach wiederherzustellen da Daten ident. Bsp: 2x 2TB Platten = ein 2TB Volume

Raid 5: Block-level striping, Paritätsdaten auf mehreren Festplatten verteilt, daher schneller zum Lesen und langsamer zum Schreiben. Schwierig zu wiederherstellen da Paritätsdaten all over the place verteilt sind.

Bsp: 4x 2TB Platten = 5,2TB nutzbar, (nur [gut]) 1.8TB Parität

Raid 10: Kombination aus Raid1 und Raid0, mehrere Mirror (Raid1) mit Parität werden in ein Raid0 Array zusammengefasst. Ziemlich schnell und einfach. Teuer da nur der halbe Speicher verwendet werden kann (jede Festplatte hat genau eine andere mit den gleichen Daten) Bsp: 4x 8TB = 16TB nutzbar, 16TB Parität

Backup Strategien:

Zentrale Frage: Wer sichert was, wann, wie, womit, wohin und wie kann das Backup zurückgespielt werden.

Vollständig (selbsterklärend, einfach alles kopieren) (alles sichern = viel Speicher, dafür easy)

Differenziell (erst ein vollständiges Backup und danach nur die Änderungen relativ zu diesem vollständigen Backup, relativ viel Speicher aber immernoch einfach wiederherzustellen, Änderungen nach Full Backup mit jedem Differenziell erneut gesichert (doppelt und dreifach))

Inkrementell (ähnlich wie Differenziell aber nicht relativ zum letzten Full Backup sondern dem letzten vorangehenden Inkrementell Backup, daher keine doppelte Sicherung geänderter Daten, Nachteil: man braucht immer alle Backups um etwas wiederherstellen zu können und ziemlich kompliziert / langsam zum Wiederherstellen)

Beispielstrategie: Ein Fullbackup pro Monat, ein Differenzielles Backup jede Woche und ein Inkrementelles Backup jeden Tag.

Hot Backup vs. Cold Backup

HotBackup: Im laufenden Betrieb

ColdBackup: Runterfahren, Sichern, Hochfahren

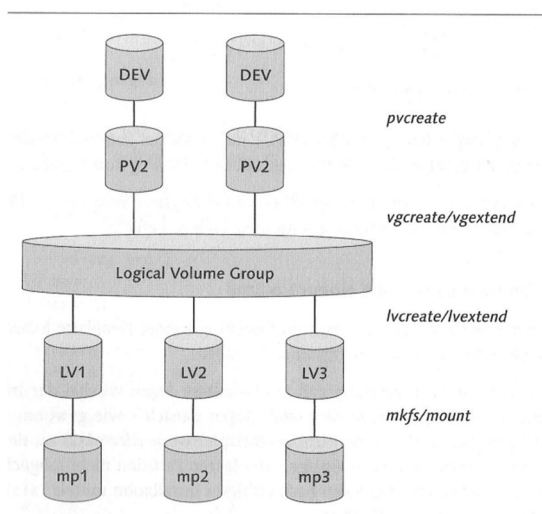
(LVM) Snapshot: Im laufenden Betrieb, unabhängig von anwendung, auch bei db möglich

LVM (Logival Volume Manager)

- Erweiterbarkeit im Betrieb
- Snapshots
- Für virtuelle Server. - Backup kompletter Maschinen im laufenden Betrieb.
- Auf fielen Systemen standard

Begriffe:

- PV → Physical Volume, representation einer echten Festplatte im LVM System
- VG → Eingensammlung aus PV in LVM (kann auch ein einziges sein)
- LV → Logical Volume, liegt auf einer VG, „Partition“, kann gemountet werden (solang fs vorhanden)
- PE → Physical Extent, kleinste verwaltungseinheit eines PV (einfach merken, in der praxis weniger relevant)
- LE → Zu PE passende logische Verwaltungseinheit



Befehle:

- `pvdisplay` → list all physical volumes
- `vgdisplay` → list all volume groups
- `lvdisplay` → list all logical volumes
- `pvcreate` → formatiert Festplatte zur Verwendung mit LVM
 - usage: `pvcreate /dev/<drivename>`
 - drivename: for example `sda` or `sdb` (don't use system volume like `sda1`!)
- `vgcreate` → erstellt eine volume group
 - usage: `vgcreate <vgname> <drivelist>`
 - `vgname`: sinnvoller name für `vg`
 - `drivelist`: lister der für `lvm` initialisierten geräte (eins oder mehrere)
„`/dev/sdb /dev/sdc`“ oder nur „`/dev/sdb`“
- `lvcreate` → erstellt logical volume
 - usage: `lvcreate -L <size> -n <lvname> <vgname>`
 - `size`: gewünschte gröÙe, suffix wie `K` für Kilobyte, `M` für Megabyte, `G` für Gigabyte, ...
 - `lvname`: sinnvoller name für das logical volume
 - `vgname`: name der volume group auf der das `lv` erzeugt werden soll
 - parameter `-snapshot` zum erstellen eines snapshot von einem anderen `lv`. Diese sind in `lvm` auch beschreibbar / mountbar. GRÖÙE DES SNAPSHOT MUS MINDESTENS DIE GRÖÙE DER DATEN AUF DEM `LV` HABEN (nicht des ganzen volumes), LÄUFT DER SNAPSHOT FOLL KANN ER NICHTMEHR VERWENDET WERDEN!
 - advanced: parameter wie „`-m1`“ für `mirror(raid1)` oder „`-i 3 -l 8`“ für 3 stripes je 8kb
- `lvs`, `vgs`, `pvs` → formatted output about `lv`'s `vg`'s and `pv`'s
- `lvremove` → logical volume löschen
 - usage: `lvremove <path_to_lv>` z.B: `lvremove /dev/myvg/mylv`

Zugriff auf logical volumes über `/dev/<vgname>/<lvname>` !!!

Zusätzliche Befehle:

- `mkfs` → „make filesystem“ → dateisystem erstellen
 - usage: `mkfs.<type> <device>` z.B: `mkfs.ext4 /dev/myvg/mylv`
 - `type`: welches dateisystem (`ext4`, `ext3`, `FAT32`, ...)
- `mount` → ein VORHANDENES dateisystem in den dateibaum einbinden
 - usage: `mount <device> <mountpath>`
 - `device`: pfad auf ein gerät / datei mit dateisystem drauf! z.B. `/dev/myvg/mylv`
 - `mountpath`: meist ein davor erstellter ordner im `/mnt` verzeichnis z.B. `/mnt/mylv`

`tar` (tape archiver): das linux tool um mehrere dateien in ein archiv zusammenzufassen (ähnlich zu `zip`, `rar`, `7z` etc unter windows aber standartmäßig keine kompression dafür scheisse schnell)

usage archiv erstellen: tar -xzf <archivname>.tar.gz -C <originalordner>

x → Kompressierung

c → create

f → Archiv (Blockungsfaktor) einfach hinnehmen pls

.tar: archiv, .gz: compressed (gzip)

usage archiv entpacken: tar -xzf <myarchive>.tar.gz

x → Komprimiertes Archiv

f → Archiv (Blockungsfaktor) einfach hinnehmen pls

v → verbose: immer optional, gibt eine liste der dateien im archiv in der console aus

z → read/extract gzipped archive

Wichtige Parameter:

--exclude=/tmp

--absolute-names

Laut Hauptmann:

rsync

- Spiegeln von Verzeichnissen
- Prüfsummenvergleich
- Diverse Übertragungswege
- Rechte, Zugriffszeiten, Links nur bei Linux System auf der Backup Seite mitgesichert.

Usage: rsync -a <pfadvonwo> <pfadwohin>

Sichern lokaler Daten | rsync -a /data /backup |

Sicherung über SSH | rsync -a ./daten root@backupserver.com:/backup

(ssh connections zählen als normale dateipfade!) ↗

rsync Parameter

-a, --archive, Kombination von -rlptgoD

-v, --verbose → alle dateien die synchronisiert wurden in die console schreiben

-r, --recursive → unterordner mitnehmen

-l, --links → links mitnehmen

Cronjobs:

You can edit the crontab config

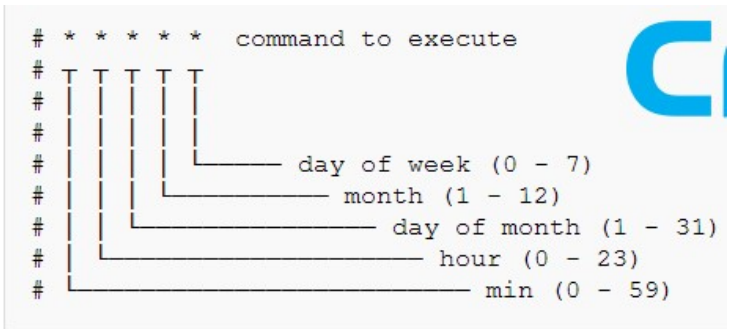
Unter /etc/crontab or typing "crontab -e"

Beispiel:

```
* /10 * 1 * 2 root bash /path/to/my/script.sh
```

Jeden ersten tag im monat wird wenn es ein Dienstag ist alle 10 minuten folgendes script als user root ausgeführt. Macht zwar kein sinn aber als beispiel hald.

Crontab lädt neue config sofort nach speichern der config datei und sag dir bescheid wenn was nicht passt.



dd (Disk Dump): <https://linux.die.net/man/1/dd>

Exakte kopie eines Laufwers (bzw Datei, da das ding functioniert auch mit dateil weil guess what ne datei kann auch n eigenes dateisystem representieren und so)

Parameter:

if → Input File / Input Device

of → Output File / Output Device

bs → Byte Size (aka wie viele Bytes)

count → Wie oft (count * bs) (wenn leer dann alles)

seek → skip n output blocks

skip → skip n input blocks

Beispiel sichern der ersten 512 bytes eines datenträgers: dd if=/dev/sda of=/my/file bs=512 count=1

In die anrere richtung einfach if und of umdrehen.

Beispiel kopieren ganzer festplatte (nicht nur datein also alles! Mbr, partitionen, dateisysteme, alles!)

```
dd if=/dev/sda of=/my/drive/copy bs=1024
```

Komprimierung mit gzip selbserklärend, einfach was wohin „gzip <myfile> <mycompressedfile>.

„Entkomprimieren“ mit „gunzip -c <mycompressedfile> <wohin>“

Probleme dd

- Partitionen dürfen bei Sicherung nicht gemountet sein.
- Größenproblem bei 1:1 Kopie.
- Eher für kleine Systeme bis max. 10GB.
- Nicht für Fileserver geeignet.

Hauptmann sagt:

MBR (Master boot record) Die ersten 512 bytes einer festplatte die mit mbr formatiert ist (also ziemlich alles, MBR = BIOS, GPT (Guid partition table, bissl neuer hald) nur mit UEFI)

Sinn: wichtig für den bootprozess (starten des os durch bios), und man muss hald wissen können wo die partitionen auf der platte actually sind weil raten ned so ideal.

Aufbau einer Festplatte MBR

Adresse	Funktion / Inhalt		Größe (Bytes)	
hex	dez			
0x0000	0	Bootloader	440	
0x01B8	440	Datenträgersignatur	4	
		(seit Windows 2000)		
0x01BC	444	Null	2	
		(0x0000)		
0x01BE	446	Partitionstabelle	64	
0x01FE	510	55hex	Bootsektor-Signatur	2
			(wird vom BIOS für den ersten Bootloader geprüft)	
0x01FF	511	AAhex		
Gesamt:				512

REAR hamma ned gmacht, selber noch nie verwendet den kas

Sinn dahinter: basicly n ganzes system das das wiederherstellen fon backups leichter machen soll und hald einfach das ganze system backupt, inklusive boot, kernel, system, treiber, etc

Den ganzen magnetband kas habts da auch nochmal da vollständigkeit halber, danke

SSH Schick ich euch einfach mein NWTK Protokoll dazu, das müsstest aber mittlerweile eh können, bei fragen schreibts ma einfach, schönes wochenende

Backup Hardware

Tape Library

Groß:

- Bis zu 10000 Slots
- Bis zu 64 Laufwerke

Klein:

- 5-20 Slots
- 1-4 Laufwerke

LTO, Linear Tape Open

Tape Library Komponenten (Bandroboter)



- Roboterarm
- Etikettenleser
- Laufwerke
- Bandarchiv

LTO Längsspurverfahren

- Magnetband mit mehreren Metern in der Sekunde an den Schreib-/Leseköpfen vorbeigeführt wird.
- Datenspurten liegen parallel zum Magnetband.
- Fester Schreib-/Lesekopf mit mehreren 100 parallelen Datenspurten.
- Oberhalb und unterhalb eines Datenbandes sind vordefinierte Servospurten zur Synchronisation, Positionierung und Kalibrierung angeordnet.

LTO

- 1/2-Zoll-Magnetband
- Aufzeichnung im Längsspurverfahren
- Kontinuierliche Entwicklung
- WORM Bänder
- Reinigungsband
- SCSI od. SAS Interface
- Seit Generation 5 Linear Tape Filesystem.

Evolution LTO

- Ultrium 1 100 GB
- Ultrium 2 200 GB
- Ultrium 3 400 GB
- Ultrium 4 800 GB
- Ultrium 5 1.500 GB
- Ultrium 6 2.500 GB Ab 2000 Euro
- Ultrium 7 6.400 GB (Stand Ende 2015)
- Ultrium 8 12.800 GB