Inhaltsverzeichnis

ZFS & ZPool (File System)	1
Was ist ZFS oder Zpool?	1
Einrichten	2
btrfs (B-tree File System)	4
Installation und Konfiguration	
BtrFs einrichten	
1. Disk einrichten	
2. xxx	5
3. xxx	
4. xxx	5
5. xxx	5
6. xxx	5



update:
0/11/17 22:45 start:linux:filesystem:btrfs https://jmz-elektronik.ch/dokuwiki/doku.php?id=start:linux:filesystem:btrfs&rev=1605649525

2025/10/30 17:47 1/6 ZFS & ZPool (File System)

ZFS & ZPool (File System)

Was ist ZFS und was ist ein ZPool? Folgende Zittate können als kurze Erklährung herangezogen werden.

Was ist ZFS oder Zpool?

Ein Zittag aus der FreeBSD Online-Dokumentation. ZFS ist ein fundamental anderes Dateisystem aufgrund der Tatsache, dass es mehr als ein Dateisystem ist. ZFS kombiniert die Rolle eines Dateisystems mit dem Volumemanager, was es ermöglicht, zusätzliche Speichermedien zu einem laufenden System hinzuzufügen und diesen neuen Speicher sofort auf allen auf dem Pool existierenden Dateisystemen zur Verfügung zu haben. Durch die Kombination von traditionell getrennten Rollen ist ZFS in der Lage, Einschränkungen, die zuvor RAID-Gruppen daran gehindert hatten, zu wachsen. Jedes Gerät auf höchster Ebene in einem Pool wird ein vdev genannt, was eine einfache Platte oder eine RAID-Transformation wie ein Spiegel oder RAID-Z-Verbund sein kann. ZFS-Dateisysteme (datasets genannt), haben jeweils Zugriff auf den gesamten freien Speicherplatz des gesamten Pools. Wenn Blöcke aus diesem Pool allokiert werden, verringert sich auch der freie Speicherplatz für jedes Dateisystem. Dieser Ansatz verhindert die allgegenwärtige Falle von umfangreichen Partitionen, bei denen freier Speicherplatz über alle Partitionen hinweg fragmentiert wird.

Ein Zittat aus der Wiki Online-Dokumentation. ZFS ist ein von Sun Microsystems entwickeltes transaktionales Dateisystem, das zahlreiche Erweiterungen für die Verwendung im Server- und Rechenzentrumsbereich enthält. Hierzu zählen die vergleichsweise große maximale Dateisystemgröße, eine einfache Verwaltung selbst komplexer Konfigurationen, die integrierten RAID-Funktionalitäten, das Volume-Management sowie der prüfsummenbasierte Schutz vor Datenübertragungsfehlern. Der Name ZFS stand ursprünglich für Zettabyte File System, ist aber inzwischen ein Pseudo-Akronym, wodurch die Langform nicht mehr gebräuchlich ist.

Ein Zittag aus dem Ubuntu Wiki Online-Dokumentation. ZFS filesystems are built on top of virtual storage pools called zpools. A zpool is constructed of virtual devices (vdevs), which are themselves constructed of block devices: files, hard drive partitions, or entire drives, with the last being the recommended usage.[6] Block devices within a vdev may be configured in different ways, depending on needs and space available: non-redundantly (similar to RAID 0), as a mirror (RAID 1) of two or more devices, as a RAID-Z group of three or more devices, or as a RAID-Z2 group of four or more devices.[7] Besides standard storage, devices can be designated as volatile read cache (ARC), nonvolatile write cache, or as a spare disk for use only in the case of a failure. Finally, when mirroring, block devices can be grouped according to physical chassis, so that the filesystem can continue in the face of the failure of an entire chassis.

Zusammenfassend hier die technischen Eckdaten des ZFS:

Bedeutung	Grösse oder Dimmension
Wortlänge	128 Bit
Volumen Manager	integriert
Ausfallsicherheit	RAID-1 RAID-Z1 (1 Parity-Bit analog RAID 5)
max Grösse des Dateisystems	16 EiB(=64 Bit Gross)

Bedeutung	Grösse oder Dimmension
max. Größe jedes Pools	128 Bit Gross
max. Anzahl an Dateien in einem Verzeichnis	48 Bit Gross
max. Anzahl an Geräten im Pool	64 Bit Gross
max. Anzahl an Dateisystemen im Pool	64 Bit Gross
max. Anzahl an Pools im System	64 Bit Gross

Einrichten

Ein zpool zeigt sich aus Sicht des Anwender wie ein Verzeichnis analog dem mount Verzeichnis bei Linux (/mnt/...). Um ein zpool laufwerk (Verzeichnis) einzurichten benötigt man nur wenige Schritte. Folgend einige Beispiele gefunden auf dieser Seite.

1. #ZPool Types 2. #ZPool Stripe group: 3. zpool create vol0 /dev/sda /dev/sdb /dev/sdc 4. #Der Pool heisst hier vol0 und man kann in als Ordner im 5. #Linux Dateisystem sehen. Darin befinden sich 3 Laufwerke 6. #die alle zu einem zusammengefasst werden. Die Grösse 7. #ist also sda + sdb + sdc = max Speicherkapazität. 8. #Achtung hier keine Datensichheit. Ist ein Laufwerk 9. #defekt können Sie Daten verlieren. 10. 11. #ZPool mirror group: 12. zpool create vol0 mirror /dev/sda /dev/sdb 13. #Alles wie oben erwähnt. Nur hier werden 2 Laufwerke 14. #gespiegelt. Damit sind auf beiden Laufwerken 15. #identischen Daten vorhanden. Wenn ein Laufwerk 16. #ausfällt so sind die Daten auf dem anderen 17. #noch enthalten. Die Speichergrösse ist allerdings 18. #nur so gross wie die des kleinsten Laufwerks. 19. 20. 21. #ZPool raidz group: Similar to RAID5. 22. zpool create vol0 raidz /dev/sda /dev/sdb /dev/sdc 23. 24. #ZPool raidz2 set: Similar to RAID5 with dual parity. 25. zpool create vol0 raidz2 /dev/sdb /dev/sdc1 /dev/sdd /dev/sde

Jetzt stellt sich die Frage wie man diese Pools (Disk Pools) auch wieder löschen kann.

1.

2025/10/30 17:47 3/6 ZFS & ZPool (File System)

Eine einmal eingesetzte Harddisk oder auch zwei (Mirror) kann ausfallen und muss unter Umständen ausgetauscht werden. Damit eine z.B. Harddisk defekt ist, muss man diese letztendlich austauschen. Mit folgende Befehlen lässte sich eine HD aus dem zpool entfernen.

```
1. # Erst muss festellen wie der zpool aufgebaut ist bzw. ein defekt
   vorliegt.
 2. zpool status
 3. # Ausgabe:
      pool: zfs_disk_pool
 4.
    state: ONLINE
 5.
      scan: scrub repaired OB in 1 days 02:03:00 with 0 errors on Mon Nov
 6.
   9 02:27:01 2020
 7. config:
 8.
 9.
            NAME
                           STATE
                                     READ WRITE CKSUM
10.
            zfs disk pool
                           ONLINE
                                         0
                                               0
                                                     0
11.
              mirror-0
                           ONLINE
                                         0
                                               0
                                                     0
12.
                sdb
                           ONLINE
                                        0
                                               0
                                                     0
13.
                                        0
                                               0
                sdc
                           ONLINE
                                                     0
14.
              mirror-1
                           ONLINE
                                        0
                                               0
                                                     0
15.
                sdd
                           ONLINE
                                         0
                                               0
                                                     0
16.
                sde
                           ONLINE
                                         0
                                               0
                                                     0
17.
18. # Nun muss noch festgestellt werden welche HD zu entfernen ist.
19. # Mit folgendem Befehl lässt sich eine detaillierte Ausgabe anzeigen.
20. lsblk
21.
22. # Hier ein Beispiel.
23. NAME
                                 MAJ:MIN RM
                                               SIZE RO TYPE MOUNTPOINT
24. sda
                                   8:0
                                           0 465.8G 0 disk
   ⊢sda1
25.
                                   8:1
                                              1007K 0 part
                                           0
    ⊢sda2
                                   8:2
26.
                                               512M 0 part
                                           0
27.
    ∟sda3
                                   8:3
                                           0 465.3G 0 part
28.
       -pve-swap
                                 253:0
                                           0
                                                 8G 0 lvm
                                                            [SWAP]
29.
       -pve-root
                                 253:1
                                                96G 0 lvm
                                          0
30.
       -pve-data tmeta
                                 253:2
                                          0
                                               31.
        ∟pve-data-tpool
                                          0 338.4G 0 lvm
                                 253:4
                                           0 338.4G 0 lvm
32.
          —pve-data
                                 253:5
33.
           -pve-vm--100--disk--0 253:6
                                                32G 0 lvm
          └pve-vm--800--disk--0 253:7
34.
                                                56G 0 lvm
                                          0
        pve-data tdata
35.
                                 253:3
                                          0 338.4G 0 lvm
36.
        ∟pve-data-tpool
                                 253:4
                                          0 338.4G 0 lvm
37.
           -pve-data
                                 253:5
                                           0 338.4G 0 lvm
38.
           -pve-vm--100--disk--0 253:6
                                                32G 0 lvm
                                           0
39.
           -pve-vm--800--disk--0 253:7
                                          0
                                                56G 0 lvm
40. sdb
                                   8:16
                                          0
                                             12.8T 0 disk
12.8T 0 part
                                   8:17
                                          0
42. ∟sdb9
                                   8:25
                                                 8M
                                                     0 part
                                           0
43. sdc
                                   8:32
                                          0
                                              12.8T
                                                     0 disk
8:33
                                              12.8T
                                           0
                                                     part
```

```
45. ∟sdc9
                                  8:41
                                         0
                                              M8
                                                  0 part
46. sdd
                                  8:48
                                            7.3T
                                        0
                                                  0 disk
47. ⊢sdd1
                                            7.3T
                                  8:49
                                         0
                                                  part
48. ∟sdd9
                                  8:57
                                        0
                                              8M
                                                  part
49. sde
                                  8:64
                                        0
                                            7.3T
                                                  0 disk
8:65
                                        0
                                            7.3T
                                                  0 part
8:73
                                        0
                                              M8
                                                  part
52.
53. # In diesem Beispiel möchte ich die beiden 8TB HD's austauschen. Also
   'sdd' und 'sde'.
54. # Wie bei 'zpool status' ersichtlich befinden sich beide in 'mirror-1'.
55. # Mit folgendem Befehl werden nun beide aus dem zpool herausgenommen.
```

btrfs (B-tree File System)

Für das neue Dateisystem btrfs (B-tree File System) gibt es an dieser Stelle eine englische Anleitung. Damit lassen sich viele verschieden Dateisysteme und über verschiedene Laufwerke erstellen. Ein kurzes Youtube Video erklärt was btrfs ist und wie einige Befehlen funktionieren. Dieser Bericht über btrfs erklärt die Optimierungsmöglichenkeit von btrfs. Diese Hilfe ist spezielle für Ubuntu Systeme gedacht.

Beim Einsatz von mehreren Haddisk oder SSD empfield sich ein btrfs Raid einzusetzen.

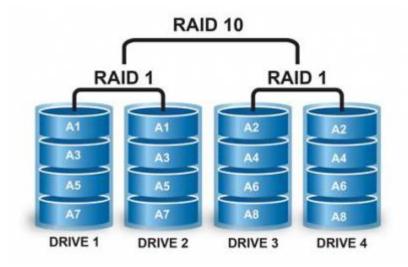
Installation und Konfiguration

Das Dateisystem **btrfs** ist ein System dem sich auf einfachste weise Speichermedien hinzuzufügen und entfernen lassen ohne dabei die Daten zu verlieren. Letzteres sollte man natürlich nur mit vorsicht vornehmen will man nicht Daten verlieren. Es lassen sich auch Relationen wie RaidO, Raid1, Raid10 erstellen und diese lassen sich auch im nachhinein einrichten. Diese kurze wiki Beschreibung hilft beim ersten Einrichten eines btrfs Dateisystems.

BtrFs einrichten

Als Beispiel wie btrfs einzurichten ist, soll folgende Situation herhalten. Stellen Sie sich vor Sie wollen zukünftig ein **Raid10** besitzen, haben aber heute noch nicht das Geld gleich vier Harddisk zu kaufen. Infolge dessen wird man erst eine Platte kaufen und diese einrichten. Das Gute dabei ist, dass man sich zu diesem Zeitpunkt noch nicht entscheiden muss welches Raid man letztendlich anwenden will. Nachfolgende Grafik zeigt der Aufbau eines Raid10 Systems auf.

2025/10/30 17:47 5/6 ZFS & ZPool (File System)



٠.

1. Disk einrichten

mkfs.btrfs -d single /dev/sdb //Hier Ihre Laufwerksname

2. xxx

Tool → btrfs-prog ...

3. xxx

mkfs.btrfs

4. xxx

mount → over uuid

5. xxx

nano /etc/fstab

6. xxx

...folgt..

From:

https://jmz-elektronik.ch/dokuwiki/ - Bücher & Dokumente

Permanent link:

https://jmz-elektronik.ch/dokuwiki/doku.php?id=start:linux:filesystem:btrfs&rev=1605649525

Last update: 2020/11/17 22:45

