Proxmox VE mit ZFS

Stand: 06.06.24

© 2024 Marco Gabriel, inett GmbH

ZFS Filesystem



- 128 Bit Filesystem
- Für sehr große Datenvolumina geeignet
- Kein Clusterfilesystem
- Permanente Integritätsprüfung und Selbstheilung
- Integriert Volume Manager, RAID und Filesystem
- Moderne Features wie Copy-On-Write, Snapshots, SSDs für Cache und Logs, Komprimierung, Deduplizierung, asynchroner Sync
- Benötigt keinen Hardware RAID Controller
- Hohe Performance
- Für Server / Datacenter ausgelegt

© 2024 Marco Gabriel, inett GmbH

ZFS in Proxmox VE



- Proxmox VE verwendet ZFS on Linux
- Im Proxmox Installer auswählbar
- Software-Raid Konfiguration für Proxmox VE
- Proxmox VE kann von ZFS booten
- VM und Container Images möglich
- Auf Filesystem oder als Volume (zvol Block)

© 2024 Marco Gabriel, inett GmbH

ZFS Administration

© 2024 Marco Gabriel, inett GmbH

Pools

- ZFS Pool erstellen

```
zpool create -f -o ashift=12 <pool-name> <device>
```

- Globale Komprimierung aktivieren

```
zfs set compression=lz4 <pool-name>
```

- ashift=12 sorgt für richtige „Ausrichtung“ (Alignment) bei 4k Festplatten und SSDs

© 2024 Marco Gabriel, inett GmbH

RAID 0/1/10 Pools

- RAID 0 (performant aber unsicher)
  - `zpool create -f -o ashift=12 <pool-name> <device1> <device2>`
- RAID 1 (2 Disks, redundant)
  - `zpool create -f -o ashift=12 <pool-name> mirror <device1> <device2>`
- RAID 10 (4 Disks, redundant und performant)
  - `zpool create -f -o ashift=12 <pool-name> mirror <device1> <device2> mirror <device3> <device4>`

© 2024 Marco Gabriel, inett GmbH

RAIDZ1/2/3 Pools

- RAIDZ1 (3 Disks, redundant, optimierte Kapazität)
  - `zpool create -f -o ashift=12 <pool-name> raidz1 <device1> <device2> <device3>`
- RAIDZ2 (4 Disks, 2-fach redundant)
  - `zpool create -f -o ashift=12 <pool-name> raidz2 <device1> <device2> <device3> <device4>`
- RAIDZ3 (5 Disks, 3-fach redundant)
  - `zpool create -f -o ashift=12 <pool-name> raidz3 <device1> <device2> <device3> <device4> <device5>`

© 2024 Marco Gabriel, inett GmbH

Pool vergrößern (RAID1/10)

- Partitionstabelle von existierendem Device auf neues Device kopieren und für jedes neue Device wiederholen:
  - `sgdisk <existing_device> -R <new_device>`
  - Beispiel: `sgdisk /dev/sdc -R /dev/sdi`
- GUID neu erstellen:
  - `sgdisk -G <new device>`
  - Beispiel: `sgdisk -G <new_device>`

© 2024 Marco Gabriel, inett GmbH

Pool vergrößern (RAID1/10)

- `zpool add <devices>` fügt vdevs zu einem pool hinzu

```
  pool: rpool
 state: ONLINE
  scan: scrub repaired 0B in 0h35m with 0 errors on Sun Mar 11 00:59:39 2018
config:

        NAME                                            STATE     READ WRITE CKSUM
        rpool                                           ONLINE       0     0     0
          mirror-0                                      ONLINE       0     0     0
            scsi-3600508b1001c1bc00dd554d42a70a9fa      ONLINE       0     0     0
            scsi-3600508b1001cf286081f2e92715cfea7      ONLINE       0     0     0
```

- Z.B. `zpool add rpool mirror /dev/sdf /dev/sdg`

```
  pool: rpool
 state: ONLINE
  scan: scrub repaired 0B in 0h35m with 0 errors on Sun Mar 11 00:59:39 2018
config:

        NAME                                            STATE     READ WRITE CKSUM
        rpool                                           ONLINE       0     0     0
          mirror-0                                      ONLINE       0     0     0
            scsi-3600508b1001c1bc00dd554d42a70a9fa      ONLINE       0     0     0
            scsi-3600508b1001cf286081f2e92715cfea7      ONLINE       0     0     0
          mirror-1                                      ONLINE       0     0     0
            scsi-3600508b1001c3a216505dd5c5507a4bf      ONLINE       0     0     0
            scsi-3600508b1001c4f71ff646acecdd2aec0      ONLINE       0     0     0
```

© 2024 Marco Gabriel, inett GmbH

SSD Cache

- SSD Cache hinzufügen
  - `zpool add -f <pool-name> cache <device>`
  - Lohnt nur bei großen Backend Pools, RAM ist effizienter
  - Beispiele:
    - 4 GB ARC (RAM): ca. 80% Hitrate
    - 100 GB L2ARC (SSD): ca. 2% Hitrate

- SSD SLOG hinzufügen
  - `zpool add -f <pool-name> log <device>`
  - 100% Schreibzugriffe → SSD/NVMe DWPD beachten!
  - Hilft nur bei synchronen Schreibvorgängen

© 2024 Marco Gabriel, inett GmbH

SSD Cache

- Cache und Log können auf einer SSD liegen
- Log sollte klein, Cache sollte groß sein
- Bei 500 GB SSD z.B. 1 bis 16 GB Log, 450+ GB Cache
- SSD Cache und Log hinzufügen
  - `zpool add -f <pool-name> cache <device1.part1> log <device1.part2>`

© 2024 Marco Gabriel, inett GmbH

ZFS Special Device

- Speichert Metadaten, Dedup Tabellen und optional kleine Dateien eines ZFS Pools
- Gleiche Redundanz wie Pool benötigt
- Größe = 2% der Poolgröße
- Kann nicht rückgängig gemacht werden
- Bei Ausfall fällt der gesamte ZFS Pool aus
- Einsatz im Proxmox Backup Server, um Zugriff auf Metadaten von HDDs zu beschleunigen

© 2024 Marco Gabriel, inett GmbH

ZFS Special Device

- ZFS Pool erzeugen mit Special Device:
  - `zpool create -f -o ashift=12 <pool> mirror <device1> <device2> special mirror <device3> <device4>`
- ZFS Pool erweitern um Special Device:
  - `zpool add <pool> special mirror <device1> <device2>`
- Files bis 4K direkt auf Special Device schreiben:
  - `zfs set special_small_blocks=4K <pool>`

© 2024 Marco Gabriel, inett GmbH

Pool Upgrade

```
root@pve:~# zpool status
  pool: rpool
 state: ONLINE
status: Some supported features are not enabled on the pool. The pool can
still be used, but some features are unavailable.
action: Enable all features using 'zpool upgrade'. Once this is done,
the pool may no longer be accessible by software that does not support
the features. See zpool-features(5) for details.
```

© 2024 Marco Gabriel, inett GmbH

Pool Upgrade

```
root@pve:~# zpool upgrade rpool
This system supports ZFS pool feature flags.

Enabled the following features on 'rpool':
  redaction_bookmarks
  redacted_datasets
  bookmark_written
  log_spacemap
  livelist
  device_rebuild
  zstd_compress
```

© 2024 Marco Gabriel, inett GmbH

Fehlerbehandung

- Ausfall der Cache SSD ist unkritisch → ZFS liest dann (langsam) von den normalen Festplatten
- Ausfall der Log SSD ist unkritisch, sofern nicht gleichzeitig RAM defekt ist
- Ausfall einer Festplatte ist unkritisch, wenn Redundanz konfiguriert wurde

- Fehlerhaftes Device austauschen:
  - `zpool replace -f <pool-name> <old device> <new-device>`

© 2024 Marco Gabriel, inett GmbH

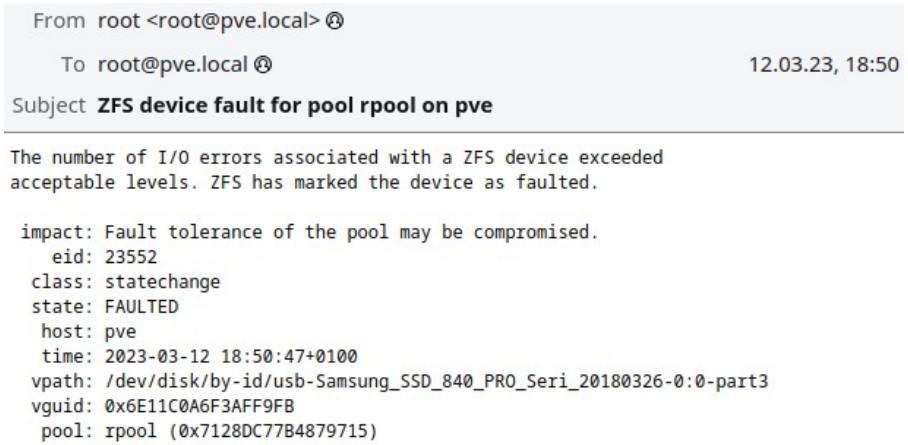
Best Practice

- Wichtig:
  - Redundanz verwenden, je nach Performance Anforderung den passenden RAID Level wählen
  - ECC RAM verwenden
  - RAM Limit konfigurieren
    - Proxmox VE <8.1 default: 50% RAM für Caching
    - Proxmox VE 8.1+ default: 10% / max. 16GB
- Weniger wichtig:
  - SSD Cache verwenden (Enterprise SSDs)
  - SSD Log verwenden (Enterprise SSDs / DWPD++)

© 2024 Marco Gabriel, inett GmbH

E-Mail Benachrichtigung

- Paket zfs-zed installieren:
  - `apt-get install zfs-zed`
- Konfiguration:
  - `nano /etc/zfs/zed.d/zed.rc`
    - `ZED_EMAIL_ADDR=„myemail@example.com"`


From root <root@pve.local>
To root@pve.local 12.03.23, 18:50
Subject **ZFS device fault for pool rpool on pve**

```
The number of I/O errors associated with a ZFS device exceeded
acceptable levels. ZFS has marked the device as faulted.

 impact: Fault tolerance of the pool may be compromised.
    eid: 23552
  class: statechange
  state: FAULTED
   host: pve
   time: 2023-03-12 18:50:47+0100
  vpath: /dev/disk/by-id/usb-Samsung_SSD_840_PRO_Seri_20180326-0:0-part3
  vguid: 0x6E11C0A6F3AFF9FB
   pool: rpool (0x7128DC77B4879715)
```

© 2024 Marco Gabriel, inett GmbH

RAM Limit konfigurieren

- PVE < 8.1: ZFS verwendet 50% des verfügbaren RAM als default → nicht gut auf einem Hypervisor

- `nano /etc/modprobe.d/zfs.conf`
  - 4 GB Limit:
    - `options zfs zfs_arc_max=4294967296`
  - 6 GB Limit:
    - `options zfs zfs_arc_max=6442450944`
- Wenn das root Filesystem ZFS ist:
  - `update-initramfs -u`
  - `pve-efiboot-tool refresh`

© 2024 Marco Gabriel, inett GmbH

Sonstige Empfehlungen

- Swappiness anpassen
  - `sysctl -w vm.swappiness=5`

© 2024 Marco Gabriel, inett GmbH

ZFS Auto Snapshot

- zfs-auto-snapshot
  - Erstellt automatisch Snapshots viertelstündlich, stündlich, täglich, wöchentlich und monatlich
  - löscht alte, abgelaufene Snapshots wieder
  - Zusätzlichen Speicherbedarf berücksichtigen und beobachten
  - `apt install zfs-auto-snapshot`

© 2024 Marco Gabriel, inett GmbH

ZFS Auto Snapshot

```
# cron job # /etc/cron[.d]/[hourly|daily|weekly|monthly]/zfs-auto-snapshot PATH="/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin„

*/15 * * * * root zfs-auto-snapshot --quiet --syslog --label=frequent --keep=4 //
```

- Erstellt alle 15 Minuten einen Snapshot aller Volumes („//“)
- behält die neuesten 4 davon, verwirft ältere
- Labelt sie „frequent“ im Namen des Snapshots

© 2024 Marco Gabriel, inett GmbH

ZFS Auto Snapshot

```
zfs set com.sun:auto-snapshot=false rpool/swap
```

- Exkludiert rpool/swap beim Erstellen der automatischen Snapshots

```
zfs inherit com.sun:auto-snapshot rpool/swap
```

- Macht die Einstellung wieder rückgängig

© 2024 Marco Gabriel, inett GmbH

Tools & Tipps

- Rollback eines Snapshots
  - `root@pve:~# zfs rollback rpool/data/vm-106-disk-1@zfs-auto-snap_daily-2018-03-19-0525 [-r]`
  - -r wird benötigt, wenn der Snapshot nicht der letzte Snapshot ist (die zwischenzeitlich erstellten werden dann gelöscht)

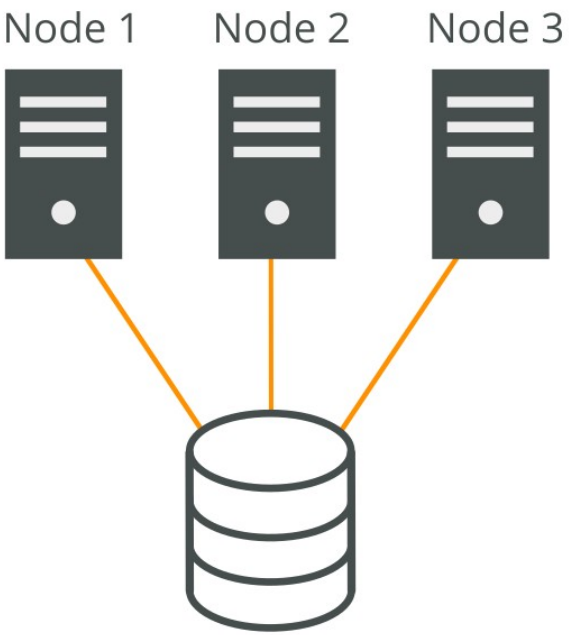
© 2024 Marco Gabriel, inett GmbH

PVE Storage Replication

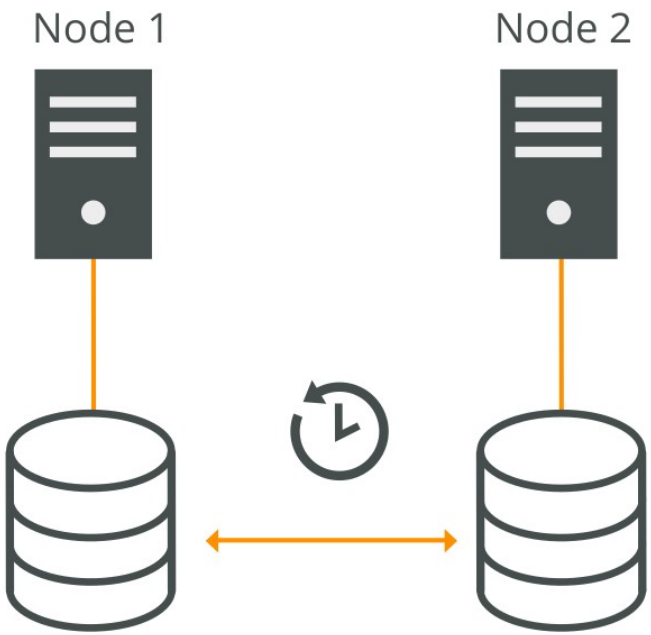
- Gut geeignet für kleine Setups aus zwei Servern
- Es liegt immer eine Kopie einer VM auf dem zweiten Server und kann dort nach Verlust des ersten Servers gestartet werden
- Geht in beide Richtungen
- ZFS muss auf beiden Hosts verwendet werden
- ZFS Storage/Pool muss gleichen Namen haben
- Sendet nach dem initialen Transfer nur noch die Differenz

© 2024 Marco Gabriel, inett GmbH

HA – Shared vs Replicated



All nodes see same shared storage



Nodes see local storage with potentially older data

© 2023 Marco Gabriel, inett GmbH