

Presentatie

ZFS on Linux

Linux User Group Nijmegen

12-1-2016

# Wat is ZFS?

ZFS is een geavanceerd bestandssysteem met geïntegreerde functies zoals RAID, volume management en snapshots. Het heeft een standaard beveiliging tegen gegevenscorruptie en repareert gedetecteerde fouten automatisch.

# ZFS geschiedenis

- Zettabyte File System
- Ontwikkeld door Sun Microsystems
- Released in 2005 voor OpenSolaris
- Open Source onder Common Development and Distribution License (CDDL)
- Na overname Sun door Oracle in 2010 is ZFS weer closed source (proprietary code)

# Open ZFS

- Open Source ontwikkelingen voortgezet op Illumos (fork van OpenSolaris)
- Ports naar Linux (Native Kernel & Fuse), FreeBSD (inclusief FreeNAS), Mac OS X
- Open ZFS als paraplu-project in 2013 gelanceerd voor coördineren open source ontwikkeling ZFS <http://open-zfs.org/>
- Feature flags i.p.v. ZFS Pool versienummer en ZFS Filesystem versienummer
- Uitwisselbaar met andere platformen (zolang alle gebruikte features worden ondersteund)

# ZFS on Linux

- Huidige versie: 0.6.5.3, 13 oktober 2015
- <http://zfsonlinux.org/>
- Packages & documentatie beschikbaar voor
  - Arch Linux
  - Debian
  - Fedora
  - Funtoo
  - Gentoo
  - RHEL / CentOS / SL
  - Sabayon
  - SprezzOS
  - Ubuntu

# Installatie Ubuntu 14.04 LTS

## Adding this PPA to your system

You can update your system with unsupported packages from this untrusted PPA by adding **ppa:zfs-native/stable** to your system's Software Sources. ([Read about installing](#))

▼ [Technical details about this PPA](#)

This PPA can be added to your system manually by copying the lines below and adding them to your system's software sources.

Display sources.list entries for: Trusty (14.04) ▼

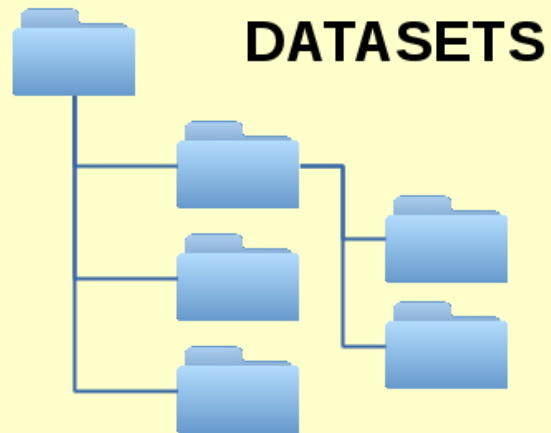
```
deb http://ppa.launchpad.net/zfs-native/stable/ubuntu trusty main
deb-src http://ppa.launchpad.net/zfs-native/stable/ubuntu trusty main
```

**Signing key:**  
[1024R/F6B0FC61](#) ([What is this?](#))

**Fingerprint:**  
E871F18B51E0147C77796AC81196BA81F6B0FC61

- *sudo add-apt-repository 'deb http://ppa.launchpad.net/zfs-native/stable/ubuntu trusty main'*
- *sudo apt-key add signingkey.txt* (Signing Key downloaden van [zfsonlinux.org](#))
- *sudo apt-get update*
- *sudo apt-get install ubuntu-zfs*

# Opbouw ZFS



- Directory structuur
- Beheren met het commando: *zfs*

## POOL



- Verzameling van harde schijven
- Beheren met het commando: *zpool*

# Zpool – VDEVs

## STRIPED



## MIRROR



## RAIDZ1



## RAIDZ2



- Pariteit disks (**paars**)
- Combinaties mogelijk (automatisch striped)
- Kleinste disk bepaald grootte (Mirror/Raidz)
- Pool alleen uitbreiden; niet inkorten!
- Disks kunnen worden vervangen
- Mirror/Raidz is géén backup!



# Zpool Create

- Syntax : *zpool create poolname [vdev] disk1 disk2 disk3*
- Mirror:
  - *zpool create opslag mirror sdb sdc*
- Raidz
  - *zpool create opslag raidz1 sdb sdc sdd sde*
  - *zpool create opslag raidz2 sdb sdc sdd sde sdf*
- Hybride
  - *zpool create opslag mirror sdb sdc raidz1 sdd sde sdf raidz2 sdg sdh sdi sdj sdk*
- Status opvragen
  - *zpool status*

# Zpool properties

- Lezen
  - *zpool get all opslag*
- Schrijven
  - *zpool set autoexpand=on opslag*

# Zpool opmerkingen

- Mirror toevoegen/verwijderen disks mogelijk (behalve laatste)
  - *zpool attach/detach*
- *Raidz kan niet worden uitgebreid; wel vervangen door grotere disks*
  - *zfs replace & property autoexpand=on*
- *Voor testdoeleinden kunnen in plaats van disks ook bestanden worden gebruikt (minimaal 64KB)*
  - *zpool create poolname /tmp/file1 /tmp/file2*
- Raidz3 bestaat ook
- commando *zdb* voor debuggen

# ZFS Create Dataset

- Dataset aanmaken
  - *zfs create opslag/gegevens*
- Nesten datasets
  - *zfs create opslag/gegevens/intern*
  - Erft properties van parent (standaard instelling)
- *Status opvragen*
  - *zfs list*

# Dataset properties

- Lezen
  - *zfs get all opslag/gegevens*
- Schrijven
  - *zfs set quota=10G opslag/gegevens*
  - Sommige properties alleen lezen (streepje – in SOURCE kolom)
- Terug naar default
  - *zfs inherit quota opslag/gegevens*

# Mogelijkheden ZFS

- 128-bits bestandssysteem (16 exabytes)
- Gegevens integriteit by design
- Raid en LVM geïntegreerd
- Copy-on-write transactiemodel : Altijd consistent
- Snapshots & Cloning
- Deduplicatie
- Compressie
- Caching

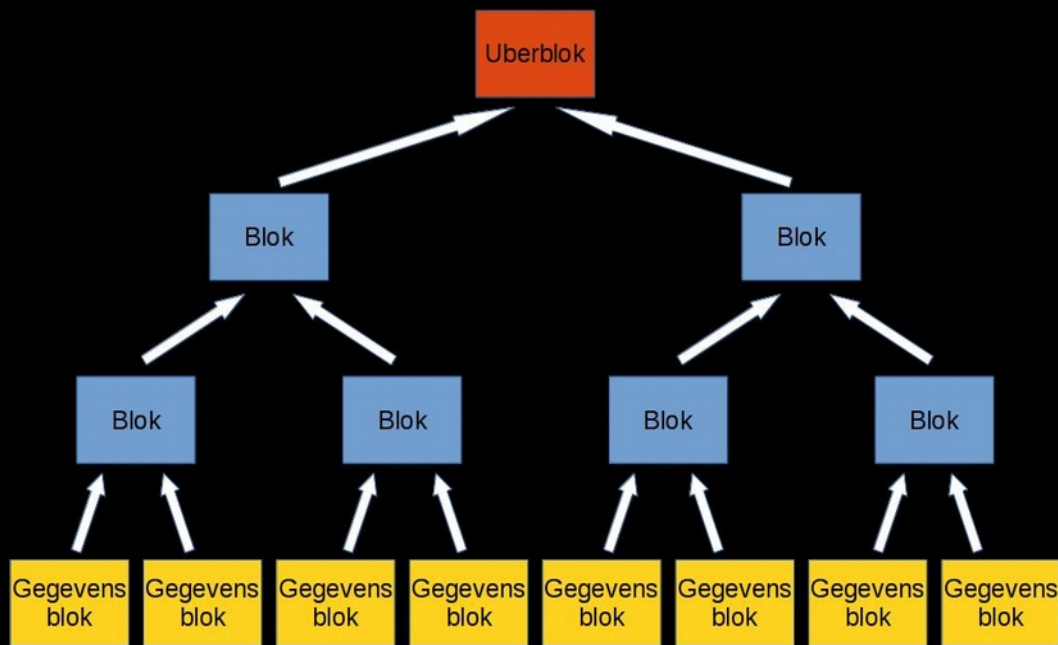
# Techniek – RAIDZ Pariteit

| RAID-5 |    |    |    |
|--------|----|----|----|
| A1     | A2 | A3 | Ap |
| B1     | B2 | Bp | B3 |
| C1     | Cp | C2 | C3 |
| Dp     | D1 | D2 | D3 |
| E1     | E2 | E3 | Ep |
| F1     | F2 | Fp | F3 |
| G1     | Gp | G2 | G3 |
| HP     | H1 | H2 | H3 |

| RAID-Z1 |    |     |     |
|---------|----|-----|-----|
| A1      | A2 | A3  | Ap  |
| A4      | A5 | A6  | Ap' |
| B1      | B2 | Bp  | C1  |
| C2      | C3 | Cp  | D1  |
| Dp      | E1 | E2  | E3  |
| Ep      | E4 | Ep' | F1  |
| Fp      | G1 | G2  | G3  |
| Gp      | H1 | H2  | Hp  |

- ZFS heeft kennis van bestanden op disk
- Variabele Stripe Width
- Schrijven data + pariteit in één transactie (atomisch)
- Zelf herstellende Raid
- Resilvering snel

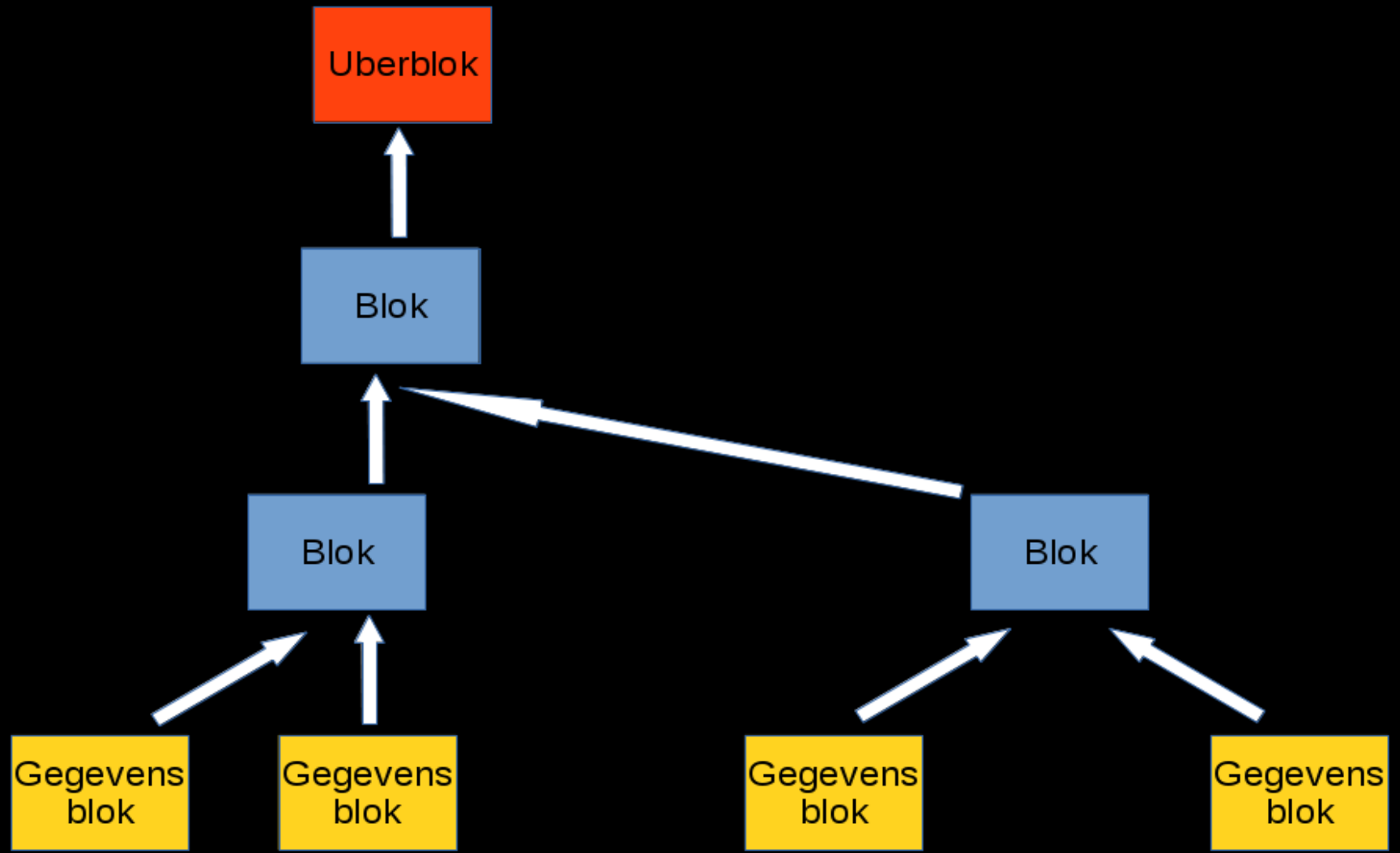
# Techniek – Blokstructuur



- Merkle/Hash tree
- Metadata gescheiden van daadwerkelijke data
- Gegevensblok (**geel**): daadwerkelijke data
- Blok (**blauw**): SHA-256 Hashwaarden
- Uberblock (**rood**): Topblok (redundant uitgevoerd)
- Controle gegevensintegriteit
  - *zpool scrub opslag*



# Techniek – Copy On Write



# Snapshots

- Momentopname
- Alleen lezen
- Kan gebruikt worden voor het terughalen van individuele bestanden of gehele dataset en voor het back-uppen
- Razendsnel; kost nagenoeg geen performance. Wel diskruimte nadien bij wijzigingen.
- Snapshot maken
  - *zfs snapshot -r opslag/gegevens@versie1*
  - *zfs list -t snapshot*
- Verborgen directory `.zfs` (niet zichtbaar met `ls -a`)
- Verwijderen snapshot
  - *zfs destroy opslag/gegevens@versie1*

# STDOUT/STDIN

- Versturen/ontvangen snapshot over STDOUT/STDIN
  - *zfs send opslag/gegevens@versie1 | zfs recv backup/gegevens*
  - *zfs send opslag/gegevens@versie1 > backup.img*
- *Incremental backup*
  - *zfs send -i opslag/gegevens@versie1 opslag/gegevens@versie2 | zfs recv backup/gegevens*
- *Over SSH*
  - *zfs send opslag/gegevens@versie1 | ssh user@server.example.com "zfs recv backup/gegevens"*

# Clones

- Wordt gemaakt van snapshot
- Lezen/schrijven
- Maken clone
  - *zfs clone opslag/gegevens@versie opslag/kloon*
- Blijft gekoppeld aan snapshot
- Promoveren naar dataset; koppeling wordt verwijderd
  - *zfs promote opslag/kloon*
- Verwijderen kloon
  - *zfs destroy opslag/kloon*

# ZFS Compressie/Deduplicatie

- Compressie
  - lzjb, gzip, zle, lz4
  - *zfs set compression=lzjb opslag/gegevens*
- Deduplicatie
  - *zfs set dedup=on opslag/gegevens*
  - Gebruikt veel RAM
  - Mogelijk over de gehele pool

# VDEV Spare

- Reserve disk
  - *zpool create opslag raidz1 sdb sdc sdd spare sde*
  - *zpool add opslag spare sde*
- Automatisch vervangen beschadigde disk
  - *zpool set autoreplace=on opslag*
  - Uitgevoerd door ZED (ZFS Event Daemon)
- Handmatig vervangen
  - *zpool replace opslag sdc sde*

# VDEV Log (SLOG/ZIL)

- SSD, NVRAM
- Buffer voor schrijftransacties
- Altijd Mirror gebruiken
- Log toevoegen
  - *zpool add opslag log mirror sdk sdl*

# VEDEV Cache

- Adjustable Replacement Cache (ARC)
- ARC in RAM
- Gebaseerd op laatst gebruikte & meest gebruikte
- Level 2 ARC op SSD (L2ARC)
- Read only dus geen mirror nodig
- L2ARC toevoegen
  - *zpool add opslag cache sdm*



# ZVOL

- Block device
- Voordelen van zfs zoals raid, mirror, copy-on-write, snapshots etc.
- Formatteren als swap, ext4
- Aanmaken
  - *zfs create -V 10G opslag/disk1*

# Wat nog meer?

- zpool import/export
- zpool online/offline
- zpool history
- Integratie met NFS & Samba

# Best Practice

- 64 bits Linux systeem; 32 bits werkt niet
- Gebruik /dev/disk/by-id/ i.p.v. sdb sdc sdd
- Gehele harde schijf gebruiken
- Verdeel harde schijven over disk controllers
- Error-correcting code (ECC) geheugen
- Minimaal 2GB geheugen. Aanbevolen 8GB of 16GB geheugen
- Bij gebruik compression en deduplication extra geheugen benodigd
- Booten & Swap op ZFS mogelijk, maar vereist extra acties
- Uitlijnen 4096 bytes sectoren Advanced Format harddisks (ashift = bit shift)
  - Zpool property ashift=12
  - Optimale configuraties:
    - RAID-Z1: 3, 5 en 9 disks
    - RAID-Z2: 4, 6 en 10 disks
    - RAID-Z3: 5, 7 en 11 disks

FIN