

ZFS

Siste ord innen filsystemer

Trond Endrestøl

Fagskolen Innlandet, IT-avdelingen

23. februar 2014

Foredragets filer I

- Filene til foredraget er tilgjengelig gjennom:
 - Subversion: `svn co svn://svn.ximalas.info/zfs-foredrag`
 - Web: svnweb.ximalas.info/zfs-foredrag
 - Begge metodene er tilgjengelig med både IPv4 og IPv6
- [zfs-foredrag.foredrag.pdf](#) vises på lerretet
- [zfs-foredrag.handout.pdf](#) er mye bedre for publikum å se på
- [zfs-foredrag.handout.2on1.pdf](#) og [zfs-foredrag.handout.4on1.pdf](#) er begge velegnet til utskrift
- *.169.pdf-filene er i 16:9-format
- *.1610.pdf-filene er i 16:10-format

Foredragets filer II

- Foredraget er mekket ved hjelp av [GNU Emacs](#), [AUCTeX](#), [pdfTeX](#) fra [MiKTeX](#), [L^AT_EX](#)-dokumentklassa [beamer](#), [Subversion](#), [TortoiseSVN](#) og [Adobe Reader](#)
- Hovedfila bærer denne identifikasjonen:
`$Ximalas: trunk/zfs-foredrag.tex 12 2014-02-23 12:56:54Z trond $`
- Driverfila for denne PDF-fila bærer denne identifikasjonen:
`$Ximalas: trunk/zfs-foredrag.handout.2on1.169.tex 3 2013-12-23 13:42:53Z trond $`
- Copyright © 2014 Trond Endrestøl
- Dette verket er lisensiert med: [Creative Commons](#), [Navngivelse-DeLPåSammeVilkår 3.0 Norge](#) (CC BY-SA 3.0)



Oversikt over hele foredraget

Del 1: Lagringssystemer

- 1 Hva kan et lagringssystem bestå av?
- 2 Hva kan gå galt i et lagringssystem?
- 3 Hvor kan det gå galt i et lagringssystem?

Oversikt over hele foredraget

Del 2: ZFS?

- 4 Hva er ZFS?
- 5 Et eksempel på en pool
- 6 Et eksempel på filsystemer i ZFS
- 7 Et annet eksempel på filsystemer i ZFS
- 8 Hva er grensene til ZFS?
- 9 Hvordan virker ZFS?
- 10 ZFS og RAID-kontrollere
- 11 Hvor kommer ZFS fra?
- 12 Versjonsnummer i ZFS
 - Pool-versjonsnummer
 - Filsystem-versjonsnummer
- 13 Fremtiden for ZFS?

Oversikt over hele foredraget

Del 3: ZFS!

- 14 Administrasjon av ZFS
 - `zpool`
 - `zfs`
- 15 Oppretting av pooler
 - Enkle pool-eksempler
 - Avanserte pool-eksempler
- 16 `zpool`-egenskaper
- 17 `zfs`-egenskaper

Oversikt over hele foredraget

Del 4: Oppstartsmiljøer

Del I

Lagringsystemer

Oversikt over del 1: Lagringssystemer

- 1 Hva kan et lagringssystem bestå av?
- 2 Hva kan gå galt i et lagringssystem?
- 3 Hvor kan det gå galt i et lagringssystem?

Hva kan et lagringssystem bestå av?

Enkel server med innebygget lagring

- 1 Strømforsyning
- 2 Strømkabler
- 3 Harddisker
- 4 Firmware i harddisker
- 5 I/O-kabler
- 6 Harddiskkontroller integrert i/tilkoblet hovedkortet
- 7 Firmware i harddiskkontroller
- 8 Hovedkort og dets kobberbaner
- 9 South bridge
- 10 DMA-kontroller på hovedkortet
- 11 Arbeidsminne tilkoblet hovedkortet
- 12 CPU

Avansert server med ekstern lagring

- 1 Strømforsyning
- 2 Strømkabler
- 3 Harddisker montert i diskhyller
- 4 Firmware i harddisker
- 5 I/O-kabler
- 6 Harddiskkontroller
- 7 Firmware i harddiskkontrolleren
- 8 Grensesnitt i harddiskkontrolleren for ekstern I/O
- 9 Kabler for ekstern I/O (og kanskje en superrask switch)
- 10 Grensesnitt for ekstern I/O tilkoblet hovedkortet
- 11 Hovedkort og dets kobberbaner
- 12 South bridge
- 13 DMA-kontroller på hovedkortet
- 14 Arbeidsminne tilkoblet hovedkortet
- 15 CPU

Hva kan gå galt i et lagringssystem?

- Data som en gang ble lagret, avleses senere som noe helt annet

Hvor kan det gå galt i et lagringssystem? I

- Kort sagt: overalt

Hvor kan det gå galt i et lagringssystem? II

- ① Strømforsyningen
 - Underdimensjonert; lavere spenning/strøm til øvrige komponenter; uttørkede elektrolyttkondensatorer; utilstrekkelig avkjøling
- ② Strømkabel mellom strømforsyning og harddisk
 - Vakkell i kontaktene; (begynnende) brudd i lederne
- ③ Harddisk
 - Slitasje på indre deler; programmeringsfeil i firmware; vibrasjoner; lese fra/skrive til feil diskblokk
- ④ I/O-kabel mellom harddisk og harddiskkontroller
 - Vakkell i kontaktene; (begynnende) brudd i lederne
- ⑤ Harddiskkontroller
 - Programmeringsfeil i firmware
- ⑥ Grensesnittet mellom harddiskkontroller og hovedkort
 - Vakkell i PCIe-kontakt

Hvor kan det gå galt i et lagringssystem? III

- ⑦ Kobberbanene i hovedkortet
 - (Begynnende) brudd etter ESD-skade, utilstrekkelig avkjøling
 - ⑧ Grensesnitt mellom DMA-kontroller og hovedkort
 - Se kobberbanene i hovedkortet
 - ⑨ Grensesnitt mellom hovedkort og arbeidsminne
 - Vakkell i soklene
 - ⑩ Arbeidsminne
 - ESD-skade; kosmisk stråling; alfapartikler fra radioaktiv forurensning i IC-innkapslingen
- Hvor mange muligheter ble dette?
 - Hvor mange harddisker har du i dine systemer?
 - Har du kontrollen?

Del II

ZFS?

Oversikt over del 2: ZFS?

- 4 Hva er ZFS?
- 5 Et eksempel på en pool
- 6 Et eksempel på filsystemer i ZFS
- 7 Et annet eksempel på filsystemer i ZFS
- 8 Hva er grensene til ZFS?
- 9 Hvordan virker ZFS?
- 10 ZFS og RAID-kontrollere
- 11 Hvor kommer ZFS fra?
- 12 Versjonsnummer i ZFS
 - Pool-versjonsnummer
 - Filsystem-versjonsnummer
- 13 Fremtiden for ZFS?

Hva er ZFS?

- ZFS er
 - ① Logisk volumhåndterer («Logical Volume Manager», LVM)
 - ② Filsystem med bl.a. snapshots, kloner, kompresjon og deduplisering
 - ③ Tilbyr også «zvolumer» som lagringsenheter for andre filsystemer
- ZFS tar
 - Dataintegritet på alvor
 - Deretter brukervennlighet (for administratorer)
 - Hastighet kommer i senere rekker
- ZFS er langt enklere enn «Storage Spaces» i Microsoft Windows Server 2012

Hva er ZFS?

- Tradisjonelt oppsett av Storage Spaces
 - Velge ut harddisker og opprette en pool
 - Opprette et volum med ønsket størrelse og lagringsform
 - striping
 - speiling, eller
 - RAID 5
 - Opprette *ett* filsystem på volumet
 - NTFS
 - ReFS
 - Begynne å lagre data

Hva er ZFS?

- ZFS organiserer lagringen i pooler som kan bestå av
 - 1 Enkeltharddisker/partisjoner
 - 2 Striping (RAID 0) mellom to eller flere harddisker/partisjoner
 - 3 Speiling (RAID 1) mellom to eller flere harddisker/partisjoner
 - 4 raidz1 (RAID 5, enkel paritet) over tre eller flere harddisker/partisjoner
 - 5 raidz2 (RAID 6, dobbel paritet) over fire eller flere harddisker/partisjoner
 - 6 raidz3 («RAID 7», trippel paritet) over fem eller flere harddisker/partisjoner
- Visse kombinasjoner av det overstående er også mulig
- Filsystemet blir opprettet samtidig med poolen
 - Nye filsystemer kan opprettes i et hierarki

Et eksempel på en pool

```
trond@enterprise:~>zpool status enterprise_zdata
pool: enterprise_zdata
state: ONLINE
scan: scrub repaired 0 in 2h15m with 0 errors on Wed Jan  1 07:18:51 2014
config:
```

NAME	STATE	READ	WRITE	CKSUM
enterprise_zdata	ONLINE	0	0	0
raidz1-0	ONLINE	0	0	0
ada2	ONLINE	0	0	0
ada3	ONLINE	0	0	0
ada4	ONLINE	0	0	0

errors: No known data errors

```
trond@enterprise:~>zfs get creation enterprise_zdata
NAME          PROPERTY  VALUE                SOURCE
enterprise_zdata  creation  Sun Jan  8 14:14 2012  -
```

- Kommando for å vise status
- Poolen heter enterprise_zdata

- Består av én «vdev» («virtual device»), raidz1, striping med enkel paritet
- Medlemmene er de tre harddiskene ada2, ada3 og ada4
- Poolen har det bra og er ONLINE
- Det samme gjelder for vdev-en og dens tre medlemmer
- «Null hull» i telleverkene
- Siste skrubbing avsluttet 1. januar 2014, kl. 07:18:51
- Ingen feil registrert siden 8. januar 2012, kl. 14:14 (har aldri kjørt zpool clear)

Et eksempel på filsystemer i ZFS

```
trond@enterprise:~>zfs list -r enterprise_zroot
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
enterprise_zroot	35,0G	406G	144K	legacy
enterprise_zroot/ROOT	3,47G	413G	144K	legacy
enterprise_zroot/ROOT/20131126-r258614	1,43G	413G	1,43G	legacy
enterprise_zroot/ROOT/20131207-r259060	1,02G	413G	1,02G	legacy
enterprise_zroot/ROOT/20140103-r260223	1,02G	413G	1,02G	legacy
enterprise_zroot/do-not-destroy	144K	407G	144K	legacy
enterprise_zroot/media	208K	406G	208K	/media
enterprise_zroot/nfs	152K	406G	152K	/nfs
enterprise_zroot/tmp	6,87M	9,99G	6,87M	/tmp
enterprise_zroot/usr	17,5G	406G	144K	/usr
enterprise_zroot/usr/compat	168K	406G	168K	/usr/compat
enterprise_zroot/usr/local	2,63G	406G	2,53G	/usr/local
enterprise_zroot/usr/local/certs	220K	406G	220K	/usr/local/certs
enterprise_zroot/usr/local/etc	105M	406G	101M	/usr/local/etc
enterprise_zroot/usr/local/etc/namedb	144K	406G	144K	/usr/local/etc/namedb
enterprise_zroot/usr/local/etc/shellkonfig3	3,63M	406G	320K	/usr/local/etc/shellkonfig3
enterprise_zroot/usr/obj	4,63G	406G	4,63G	/usr/obj
enterprise_zroot/usr/packages	472M	406G	472M	/usr/packages
enterprise_zroot/usr/ports	8,99G	406G	1,54G	/usr/ports
enterprise_zroot/usr/ports/distfiles	3,71G	406G	3,71G	/usr/ports/distfiles
enterprise_zroot/usr/ports/local	288K	406G	288K	/usr/ports/local
enterprise_zroot/usr/ports/packages	3,74G	406G	3,74G	/usr/ports/packages
enterprise_zroot/usr/ports/workdirs	336K	406G	336K	/usr/ports/workdirs
enterprise_zroot/usr/src	826M	406G	826M	/usr/src
enterprise_zroot/var	6,38G	406G	10,1M	/var
enterprise_zroot/var/crash	1,19G	406G	1,19G	/var/crash
enterprise_zroot/var/db	70,4M	406G	23,8M	/var/db
enterprise_zroot/var/db/darkstat	512K	406G	512K	/var/db/darkstat

T. Endrestøl (FSI/IT)

ZFS

23. februar 2014

21 / 58

enterprise_zroot/var/db/ports	8,18M	406G	8,18M	/var/db/ports
enterprise_zroot/var/db/sup	12,8M	406G	12,8M	/var/db/sup
enterprise_zroot/var/empty	144K	406G	144K	/var/empty
enterprise_zroot/var/log	733M	406G	733M	/var/log
enterprise_zroot/var/mail	3,68M	406G	328K	/var/mail
enterprise_zroot/var/named	392K	406G	392K	/var/named
enterprise_zroot/var/run	332K	406G	332K	/var/run
enterprise_zroot/var/spool	4,39G	406G	2,13M	/var/spool
enterprise_zroot/var/spool/cvsup	4,39G	406G	4,39G	/var/spool/cvsup
enterprise_zroot/var/tmp	1,49M	10,0G	1,49M	/var/tmp
enterprise_zroot/var/unbound	144K	406G	144K	/var/unbound

Et annet eksempel på filsystemer i ZFS

```
trond@enterprise:~>zfs list -r enterprise_zdata
```

NAME	USED	AVAIL	REFER	MOUNTPOINT
enterprise_zdata	326G	586G	224K	legacy
enterprise_zdata/do-not-destroy	192K	587G	192K	legacy
enterprise_zdata/home	315G	586G	442K	/home
enterprise_zdata/home/trond	291G	586G	269M	/home/trond
enterprise_zdata/home/trond/arbeid	15,8M	586G	15,8M	/home/trond/arbeid
enterprise_zdata/home/trond/bzrarbeid	343M	586G	343M	/home/trond/bzrarbeid
enterprise_zdata/home/trond/c	5,10M	586G	5,10M	/home/trond/c
enterprise_zdata/home/trond/cvsroot	655K	586G	655K	/home/trond/cvsroot
enterprise_zdata/home/trond/download	3,92G	586G	3,92G	/home/trond/download
enterprise_zdata/home/trond/fra-defiant	35,8G	586G	35,8G	/home/trond/fra-defiant
enterprise_zdata/home/trond/fra-mentor	281M	586G	281M	/home/trond/fra-mentor
enterprise_zdata/home/trond/fra-sovereign	104G	586G	104G	/home/trond/fra-sovereign
enterprise_zdata/home/trond/gitarbeid	4,46G	586G	4,46G	/home/trond/gitarbeid
enterprise_zdata/home/trond/hgarbeid	464M	586G	464M	/home/trond/hgarbeid
enterprise_zdata/home/trond/iptraf	4,37M	586G	4,37M	/home/trond/iptraf
enterprise_zdata/home/trond/knuth	27,0M	586G	27,0M	/home/trond/knuth
enterprise_zdata/home/trond/mail	284M	586G	224M	/home/trond/mail
enterprise_zdata/home/trond/public_html	1,00G	586G	1,00G	/home/trond/public_html
enterprise_zdata/home/trond/rfc	2,29G	586G	2,29G	/home/trond/rfc
enterprise_zdata/home/trond/steam	122G	586G	122G	/home/trond/steam
enterprise_zdata/home/trond/svnarbeid	12,4G	586G	12,4G	/home/trond/svnarbeid
enterprise_zdata/home/trond/svnroot	192K	586G	192K	/home/trond/svnroot
enterprise_zdata/home/trond/svnup	2,95G	586G	2,95G	/home/trond/svnup
enterprise_zdata/home/trond/tmp	212M	586G	212M	/home/trond/tmp

T. Endrestøl (FSI/IT)

ZFS

23. februar 2014

22 / 58

Hva er grensene til ZFS?

- ZFS er stort sett grenseløs
 - 128-bit diskadresser
 - Maks. 2^{48} poster i hver katalog
 - Maks. 2^{64} bytes (16 EiB, 16 exbibytes) for hver fil
 - Maks. 2^{64} bytes for hvert attributt
 - Maks. 2^{78} bytes (256 ZiB, 256 zebibytes) i hver pool
 - Maks. 2^{56} attributter for hver fil (egentlig begrenset til 2^{48} attributter)
 - Maks. 2^{64} enheter tilknyttet en gitt pool
 - Maks. 2^{64} pooler i et og samme system
 - Maks. 2^{64} filsystemer i samme pool
 - Ref.: <http://en.wikipedia.org/wiki/ZFS>
- Vis meg det systemet som klarer å sprengre noen av disse grensene!

Hvordan virker ZFS?

- ZFS unngår RAID 5-skrivehullet til eldre RAID-kontrollere som
 - 1 Skriver nye data til de samme datablokkene som tidligere
 - 2 Leser gamle, urørte data fra de samme datablokkene i samme stripe
 - 3 Regner ut ny paritet for datablokkene i samme stripe
 - 4 Skriver oppdatert paritet til de samme paritetsblokkene som tidligere
 - Hva skjer *nå* og *senere* hvis du får strøbrudd mellom punktene 1 og 4?
 - Har diskkontrolleren batteribeskyttet minne?

Hvordan virker ZFS?

- ZFS skriver komplette striper; data og paritet samtidig
- ZFS bruker «copy-on-write»; skriver nye data til ledige diskblokker
- Endringer som hører sammen, samles i transaksjonsgrupper («txg»)

Hvordan virker ZFS?

- Sjekksommer brukes for alt som blir lagret
 - ZFS kontrollerer at leste data er de samme som ble skrevet
 - Oppdages avvik, leter ZFS etter alternativer
 - Finnes alternativer, enten speilkopier eller paritet, så
 - 1 Leveres korrekte data til applikasjonen, og
 - 2 Avviket korrigeres automatisk på den syke disken («resilver»)
 - Finnes ingen alternativer, så må filene restaureres fra backup

- **Ikke** bruk ZFS sammen med RAID-kontrollere!
- RAID-kontrolleren kan i verste fall motarbeide ZFS
 - RAID-kontrolleren kan finne på å
 - Stokke om på skriverrekkefølgen
 - Utsette skriving av nye data
 - Harddiskene kan også oppføre seg som skissert over
 - Har du skifta batteriet i RAID-kontrolleren din?
- Sett RAID-kontrolleren i JBOD-modus, eller
- La hver harddisk være sitt enslige RAID 0-volum

Hvor kommer ZFS fra?

- Utviklet av Jeffrey Bonwick, Matthew Ahrens og flere kollegaer ved Sun Microsystems, Inc.
- Arbeidet begynte i 2001 og første prototyp ble ferdig 31. oktober 2001 (halloween)
- ZFS → Solaris, oktober 2005
- ZFS er lisensiert etter «Common Development and Distribution License» (CDDL)
- ZFS → OpenSolaris, november 2005
- ZFS → FreeBSD, april 2007
- Linux' GPL v2-lisens kompliserer import av ZFS
 - ZFS i Linux gjennom FUSE gjenstår som en (treg) mulighet
 - Brian Behlendorf ved Lawrence Livermore National Laboratory (LLNL) har laget «Native ZFS for/on Linux»
- ZFS var tilgjengelig i Mac OS X 10.5, bare read-only, men har vært tilbaketrasket siden oktober 2009
- Noen Mac OS X-entusiaster har laget sine egne ZFS-varianter
- Andre OS med ZFS-støtte: OpenIndiana, FreeNAS, PC-BSD, GNU/kFreeBSD og NetBSD

Versjonsnummer i ZFS

- Pool-versjonene 1–28 og filsystem-versjonene 1–5 er tilgjengelig gjennom OpenSolaris og illumos
- Pool-versjonene 29-34 og filsystem-versjon 6 er bare tilgjengelig i Solaris 11 (Express)
- OpenSolaris har gått videre til feature-flags og pool-versjon 1000
- illumos har gått videre til feature-flags og pool-versjon 5000
- De fleste OS-er utenom Solaris, samarbeider om videreutviklingen av illumos-varianten
- Listene på de neste slidene er kopiert fra <http://en.wikipedia.org/wiki/ZFS>

Versjonsnummer i ZFS I

Pool-versjonsnummer

- 1 First release
- 2 Ditto Blocks
- 3 Hot spares, double-parity RAID-Z (raidz2), improved RAID-Z accounting
- 4 zpool history
- 5 gzip compression for ZFS datasets
- 6 "bootfs" pool property
- 7 ZIL: adds the capability to specify a separate Intent Log device or devices
- 8 ability to delegate zfs(1M) administrative tasks to ordinary users
- 9 CIFS server support, dataset quotas
- 10 Devices can be added to a storage pool as "cache devices"
- 11 Improved zpool scrub/resilver performance

Versjonsnummer i ZFS II

Pool-versjonsnummer

- 12 Snapshot properties
- 13 Properties: `usedbysnapshots`, `usedbychildren`, `usedbyreservation`, and `usedbydataset`
- 14 `passthrough-x aclinherit` property support
- 15 Properties: `userquota`, `groupquota`, `userused` and `groupused`; also required FS v4
- 16 STMF property support
- 17 triple-parity RAID-Z
- 18 ZFS snapshot holds
- 19 ZFS log device removal
- 20 `zle` compression algorithm that is needed to support the ZFS deduplication properties in ZFS pool version 21, which were released concurrently
- 21 Deduplication

Versjonsnummer i ZFS III

Pool-versjonsnummer

- 22 `zfs receive` properties
- 23 slim ZIL
- 24 System attributes. Symlinks now their own object type. Also requires FS v5.
- 25 Improved pool scrubbing and resilvering statistics
- 26 Improved snapshot deletion performance
- 27 Improved snapshot creation performance (particularly recursive snapshots)
- 28 Multiple virtual device replacements
- 29 RAID-Z/mirror hybrid allocator
- 30 ZFS encryption
- 31 Improved `'zfs list'` performance
- 32 One MB block support
- 33 Improved share support
- 34 Sharing with inheritance

Versjonsnummer i ZFS I

Filsystem-versjonsnummer

- ① First release
- ② Enhanced directory entries. In particular, directory entries now store the object type. For example, file, directory, named pipe, and so on, in addition to the object number.
- ③ Support for sharing ZFS file systems over SMB. Case insensitivity support. System attribute support. Integrated anti-virus support.
- ④ Properties: userquota, groupquota, userused and groupused
- ⑤ System attributes; symlinks now their own object type
- ⑥ Multilevel file system support

Fremtiden for ZFS?

- Oracle kjøpte opp Sun Microsystems, Inc., 27. januar 2010
- Oracle gjorde OpenSolaris om til «ClosedSolaris» i mai 2010
- Hele ZFS-teamet hos Oracle sa opp på dagen, omtrent 90 dager etter denne avgjørelsen ifølge Bryan Cantrill
- ZFS lever videre hos
 - Oracle Solaris
 - illumos/OpenZFS
 - OpenIndiana
 - FreeBSD
 - Delphix
 - iXsystems
 - Joyent
 - NetBSD
 - Nexenta
 - Linux

Del III

ZFS!

Oversikt over del 3: ZFS!

- 14 Administrasjon av ZFS
 - zpool
 - zfs
- 15 Oppretting av pooler
 - Enkle pool-eksempler
 - Avanserte pool-eksempler
- 16 zpool-egenskaper
- 17 zfs-egenskaper

Administrasjon av ZFS

- To kommandoer (med underkommandoer)
 - 1 `zpool`
 - Administrasjon av lagringspoolene
 - 2 `zfs`
 - Administrasjon av filsystemer, zvolumer, snapshots, kloner, m.m.
- Det finnes en tredje kommando: `zdb`
 - Brukes for å avlese de indre detaljene til ZFS
 - Bør bare brukes av eksperter ...
 - ... eller av de nysgjerrige ...

Administrasjon av ZFS I

`zpool`-kommandoer

- `zpool add`
 - Brukes for å innføre en helt ny vdev-gruppe med harddisker/partisjoner
- `zpool attach`
 - Brukes for å tilføye en harddisk/partisjon til en eksisterende vdev-gruppe
- `zpool clear`
 - Brukes for å nullstille tellerne for lese-, skrive- og sjekksumfeil
- `zpool create`
 - Brukes for å opprette pooler
- `zpool destroy`
 - Brukes for å ødelegge pooler
- `zpool detach`
 - Brukes for å fjerne en harddisk/partisjon fra en vdev-gruppe

Administrasjon av ZFS II

zpool-kommandoer

- `zpool export`
 - Brukes for å eksportere en pool, for senere import i samme eller et annet system
- `zpool get`
 - Brukes for å vise verdien til alle eller utvalgte `zpool`-egenskaper
- `zpool history`
 - Brukes for å vise historikken til poolen
- `zpool import`
 - Brukes for å importere en pool eller å vise en liste over pooler som kan importeres
- `zpool iostat`
 - Brukes for å vise I/O-statistikk i sann tid
- `zpool labelclear`
 - Brukes for å fjerne alle spor av ZFS' disklabels
- `zpool list`

Administrasjon av ZFS III

zpool-kommandoer

- Brukes for å liste opp importerte pooler
- `zpool offline`
 - Brukes for å deaktivere en harddisk/partisjon
- `zpool online`
 - Brukes for (re)aktivere en harddisk/partisjon
- `zpool reguid`
 - Brukes for å tildele en ny, tilfeldig GUID til en bestemt pool
- `zpool remove`
 - Brukes for å fjerne en harddisk/partisjon
- `zpool reopen`
 - Brukes for ...
- `zpool replace`
 - Brukes for å fortelle ZFS at en harddisk/partisjon har blitt skiftet ut

Administrasjon av ZFS IV

zpool-kommandoer

- `zpool scrub`
 - Brukes for å lese gjennom alt aktivt innhold, og sjekke samsvaret mellom lagret data og lagrete sjekksummer
- `zpool set`
 - Brukes for å endre zpool-egenskapene
- `zpool split`
 - Brukes for å skille et speilmedlem fra resten av gruppa
- `zpool status`
 - Brukes for å vise status til poolen, dens medlemmer og deres status, og telleverkene for lese-, skrive og sjekksumfeil
- `zpool upgrade`
 - Brukes for å oppgradere poolene til nye formater, vise hvilke pooler som er utdaterte, og hvilke versjoner som er tilgjengelig i systemet

Administrasjon av ZFS I

zfs-kommandoer

- `zfs allow`
 -
- `zfs bookmark`
 -
- `zfs clone`
 -
- `zfs create`
 -
- `zfs destroy`
 -
- `zfs diff`
 -

Administrasjon av ZFS II

zfs-kommandoer

- zfs get
 -
- zfs groupspace
 -
- zfs holds
 -
- zfs hold
 -
- zfs inherit
 -
- zfs jail
 -
- zfs list

Administrasjon av ZFS III

zfs-kommandoer

-
- zfs mount
 -
- zfs promote
 -
- zfs receive
 -
- zfs release
 -
- zfs rename
 -
- zfs rollback
 -

Administrasjon av ZFS IV

zfs-kommandoer

- zfs send
 -
- zfs set
 -
- zfs share
 -
- zfs snapshot
 -
- zfs unallow
 -
- zfs unjail
 -
- zfs unmount

Administrasjon av ZFS V

zfs-kommandoer

-
- zfs unshare
 -
- zfs upgrade
 -
- zfs userspace
 -

Oppretting av pooler

- `zpool create [opsjoner] navn-på-pool [organiseringstype] ingredienser [organiseringstype ingredienser] ...`
- Unngå å plassere mer enn 9 enheter i hver vdev
- I stedet for å stripe en pool over 20 harddisker, vurder å speile to og to harddisker i 10 grupper

Oppretting av pooler

Enkle pool-eksempler

- Singledisk:
 - `zpool create rpool da0`
- RAID 0 over to disker:
 - `zpool create rpool da0 da1`
- RAID 1 over to disker:
 - `zpool create rpool mirror da0 da1`
- RAID 5 over tre disker:
 - `zpool create rpool raidz1 da0 da1 da2`
- RAID 6 over fire disker:
 - `zpool create rpool raidz2 da0 da1 da2 da3`
- «RAID 7» over fem disker:
 - `zpool create rpool raidz3 da0 da1 da2 da3 da4`

Oppretting av pooler

Avanserte pool-eksempler

- RAID 1+0 (3 vdevs á 2 disker):
 - `zpool create rpool mirror da0 da1 mirror da2 da3 mirror da4 da5`
- RAID 5+0 (2 vdevs á 3 disker):
 - `zpool create rpool raidz1 da0 da1 da2 raidz1 da3 da4 da5`
- RAID 6+0 (2 vdevs á 4 disker):
 - `zpool create rpool raidz2 da0 da1 da2 da3 raidz2 da4 da5 da6 da7`
- RAID 1+5+0 (2 vdevs, 2 og 3 disker):
 - `zpool create rpool mirror da0 da1 raidz1 da2 da3 da4`

zpool-egenskaper |

- size
- capacity
- altroot
- health
- guid
- version
- bootfs
- delegation
- autoreplace
- cachefile
- failmode

zpool-egenskaper II

- listsnapshots
- autoexpand
- dedupditto
- dedupratio
- free
- allocated
- readonly
- comment
- expandsize
- freeing
- feature@async_destroy
- feature@empty_bpobj

zpool-egenskaper III

- feature@lz4_compress
- feature@multi_vdev_crash_dump
- feature@spacemap_histogram
- feature@enabled_txg
- feature@hole_birth
- feature@extensible_dataset
- feature@bookmarks

zfs-egenskaper I

- type
- creation
- used
- available
- referenced
- compressratio
- mounted
- quota
- reservation
- recordsize
- mountpoint

zfs-egenskaper II

- sharenfs
- checksum
- compression
- atime
- devices
- exec
- setuid
- readonly
- jailed
- snapdir
- aclmode
- aclinherit

zfs-egenskaper III

- canmount
- xattr
- copies
- version
- utf8only
- normalization
- casesensitivity
- vscan
- nbmand
- sharesmb
- refquota
- refreservation

zfs-egenskaper IV

- primarycache
- secondarycache
- usedbysnapshots
- usedbydataset
- usedbychildren
- usedbyrefreservation
- logbias
- dedup
- mlslabel
- sync
- refcompressratio
- written
- logicalused
- logicalreferenced

Del IV

Oppstartsmiljøer

Oversikt over del 4: Oppstartsmiljøer