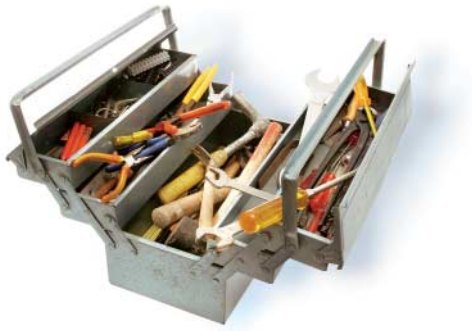
ZFS-Snapshots im Netz synchronisieren

Schnell gesichert



Michael Plura

Das Skript *zrep* hilft dem ZFS-Administrator dabei, Snapshots des Dateisystems im Netz synchron vorzuhalten.

Das Dateisystem ZFS bietet mächtige Funktionen wie Prüfsummen, Deduplizierung und Snapshots. Ein dreiteiliges Tutorial zum Einsatz von ZFS startete in *iX* 2/2017 [1]. Das frei verfügbare Skript *zrep* von Philip Brown erweitert die ZFS-Fähigkeiten und vereinfacht das Replizieren von Snapshots – vor allem im Netz.

zrep nutzt die KornShell (*ksh*) als Interpreter, bei Linux muss daher zunächst das Paket *ksh*, bei FreeBSD *ksh93* installiert werden. Anschließend lädt man das *zrep*-Skript herunter (mit *fetch* unter FreeBSD, mit *wget* unter Linux):

```
fetch http://www.bolthole.com/solaris/zrep/zrep
```

Da FreeBSD *ksh93* verwendet, *zrep* aber auf eine Solaris-Umgebung ausgelegt ist, muss man dort die erste Zeile des Skripts (Shebang-Zeile) ändern:

```
!#/usr/local/bin/ksh93 -p
```

Weil *zrep* einen automatischen *root*-Zugang zum Zielsystem benötigt, muss SSH auf dem Backup-Server für die Authentifizierung per Public-Key-Verfahren eingerichtet werden. Soll *zrep* auch im Failover-Modus arbeiten, muss sich der Backup-Server zusätzlich per *root*-SSH auf dem Quell-Server anmelden können.

Mit ZFS-Bordmitteln ist das Übertragen eines Snapshots auf einen Backup-Server schnell erledigt:

```
zfs snapshot pool01/daten@snap01
zfs send pool01/daten@snap01 | \
 ssh <server02> zfs recv -F pool02/backup
```

Das sichert einen ersten Snapshot von *pool01* mit dem Dataset *daten* auf „Server01“ und überträgt ihn in *pool02* ins Dataset *backup* auf „Server02“. Die Option *–F* stellt auf dem Zielserver den Zustand des vorigen Snapshots sicher, etwaige Änderungen verfallen. Alle weiteren Snapshots überträgt man per

```
zfs snapshot pool01/daten@snap02
zfs send -i pool01/daten@snap01
pool01/daten@snap02 | ssh <server02> ⤸
                      zfs recv pool02/backup
```

Die Option *–i* überträgt lediglich das Inkrement zwischen dem ersten und zweiten Snapshot. Da ZFS bis zu 2^{64} Snapshots ohne Performanceverlust vorhalten kann, lässt sich – genügend Festplattenplatz vorausgesetzt – per *cron*-Job jede Minute ein Snapshot anlegen und sichern. In der Praxis schreibt sich der ZFS-Systemverwalter dazu ein paar Skripte, die die Snapshots rotieren und alte löschen. Diese Aufgaben übernimmt *zrep* und ergänzt sie um überwachte, kontinuierliche Replikation sowie optional einen Failover-Mechanismus.

Vereinfachtes Handling

Ist *zrep* wie oben beschrieben eingerichtet, muss man es zunächst mit Informationen über die zu sichernden Datasets und das Backup-System initialisieren:

```
zrep init pool01/daten <server02> pool02/backup
```

Falls nicht vorhanden, legt *zrep* das Dataset *backup* automatisch an. Soll das Werkzeug weitere Datasets sichern, müssen diese ebenfalls initialisiert werden, damit spätere Backups sie berücksichtigen. Die eigentliche Replikation startet der Befehl

```
zrep sync pool01/daten
```

Das Ziel muss man nicht angeben, es ergibt sich aus der Initialisierung. Wird statt eines konkreten Quell-Datasets (hier *pool01/daten*) die Option *all* angegeben, sichert *zrep* alle zuvor initialisierten Replikationen nacheinander in einem Vorgang. Das ist bei einem überschaubaren Dataset- und Datenvolumen praktisch, für aufwendige Replikationen aber nicht empfehlenswert, da *zrep* als Skript nur single-threaded arbeitet. In diesem Fall sollte man besser mehrere *zrep*-Instanzen für die einzelnen Datasets parallel starten.

Wie beim obigen Beispiel mit *zfs send/recv* kann *zrep* über einen *cron*-Job im Minutentakt gestartet werden. Der *zrep*-Entwickler vertritt die Meinung, dass eine Endlosschleife eine hochfrequente Replikation ermöglicht. Besser ist wohl, eine einminütige Pause einzufügen (Zeile mit Kommentarzeichen):

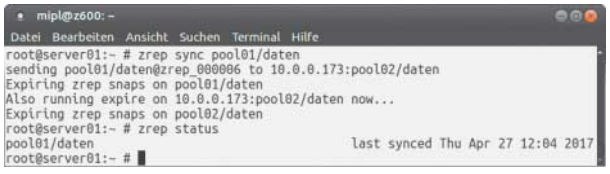
```
while true;
  do zrep sync all;
  # sleep 60
done
```

Mit *zrep status* und *zrep list* lässt sich die korrekte Arbeit von *zrep* überwachen (siehe Abbildung). Sollen die Rollen zwischen Quelle (Master) und Ziel (Stand-by-Host) bei einem Ausfall vertauscht werden, übernimmt dies das Schlüsselwort *failover* beim Master sowie *takeover* vom Stand-by-System aus (Details erläutert die *zrep*-Dokumentation).

Unter FreeBSD muss man für korrektes Funktionieren von *zrep* noch das von Linux bekannte */proc*-Dateisystem in der Datei */etc/fstab* nachtragen:

```
proc /proc procfs rw 0 0
```

Ein *mount /proc* oder ein Neustart macht das */proc*-Dateisystem verfügbar. (tiw)



***zrep* fasst alle Schritte zum Synchronisieren von ZFS-Snapshots übers Netz zusammen und überwacht den Status des Replikats.**

Michael Plura

lebt in Schweden und ist freier Autor mit den Schwerpunkten IT-Sicherheit, Virtualisierung und freie Betriebssysteme.

Literatur

[1] Michael Plura; Dateisysteme; Raumgreifend; ZFS, Teil 1: Konzepte und Grundlagen; *iX* 2/2017, S. 108

Alle Links: www.ix.de/ix1706135