

Zentrale Datensicherung mit Rsnapshot

Übersicht:

- RSNAPSHOT: Backup- bzw. Snapshot-Programm, das Kopien bzw. Sicherungen von lokalen und entfernten Dateien und Verzeichnisstrukturen anlegt und organisiert. Es kann interaktiv (eher: von Hand) oder vom System (cron) gestartet werden. Die Intervalle, Aufbewahrungsfristen und Versionsanzahlen sind fast beliebig flexibel. Das Backup erfolgt auf ein schreibbares Dateisystem (oft temporär gemountet oder Wechselfestplatte); für andere Backupmedien (Bänder, optische usw.) eignet es sich nicht bzw. sind andere Programme nötig.
- Es ist ein PERL-SKRIPT, das mit rsync arbeitet; durch Config-Datei können viele Optionen voreingestellt werden (können aber auch beim Aufruf mitgegeben werden).
- RSYNC ist ein Kopier/Synchronisierprogramm mit vielen Funktionen; am wichtigsten (von vielen) hierbei: Unterstützung von hard-links, effizienter Dateivergleich, effiziente Übertragung übers Netzwerk, Übertragung übers Netz per ssh oder rsync-server.

Vorraussetzungen:

- RSNAPSHOT selbst muss nur auf dem Host installiert sein, der die Backups ausführt bzw. vorhält (PERL logischerweise auch)
- auf allen beteiligten Hosts wird RSYNC zwingend benötigt
- SSH sehr zu empfehlen (Sicherheit und remote execution); darauf kann aber theoretisch verzichtet werden, wenn man einen RSYNC-SERVER auf dem zu sichernden Host einsetzt (möglich, aber eher unsicher und nicht empfohlen).
- PERL und SSH auf allen Unixderivaten installiert, RSYNC für alle verfügbar, die übrigen Befehle auch (GNU-Versionen bieten die umfangreichste Funktionalität). Auch für DOS-Derivate gibt es RSYNC- und SSH-Versionen; auf deren Konfiguration geht ein eigenes RSNAPSHOT-HOWTO ein (/usr/share/doc/rsnapshot-1.3.1/HOWTOs/rsnapshot-windows-howto).
- bei Verwendung von SSH muss auf den Netzwerkhosts eine nicht-interaktive Login-Möglichkeit existieren, mit der die zu sichernden Dateien gelesen werden können. Bei Beachtung der Sicherheitskonsequenzen bietet sich ein passwortloser SSH-Schlüssel für root vom Sicherungshost mit der SSHD-Option „permit root login without password“ an. (-->ssh-keygen)

Installation/Konfiguration

- mit der Softwareverwaltung installieren oder runterladen; dann ggf in System-Dir kopieren.
- conf-file kopieren/bearbeiten; wichtigere Kleinigkeiten daraus:
- linux-**gnu-befehle** sollten alle aktiv sein.
- die Kommentierung gibt ganz nützliche Infos (wie immer).
- Option „**no_create_root**“ verhindert, dass snapshot erstellt wird, wenn zb USB-Festplatte nicht gemountet ist – nützlich, damit das FS des Backup-Servers nicht vollläuft.
- die Intervalle bieten sich nach Zeit an; es ist zu beachten, dass immer

nur beim kleinsten ein wirklicher Snapshot gemacht wird; bei grösseren Intervallen wird der letzte vom nächst kleineren Intervall eins hoch rotiert, wenn dieser existiert.

- auf jeden Fall eine Überlegung lohnen die Rsync-Optionen, da mit diesen das gesamte Verhalten bzgl Archivierung usw beeinflusst wird (*man rsync*).
- SSH-Optionen: ungewöhnlicher Port, Username oä.
- WICHTIG: Includes/Excludes: hier entscheiden sich Ein- bzw Ausschlüsse (Frage der persönlichen Arbeitsweise und Praktikabilität); hängt natürlich auch von der Art der Daten und Aufgabe des Rechners ab. /home lässt sich leicht sichern; Teile von /usr/src oder gar /proc müssten aufwändiger formuliert werden.
Auf jeden Fall richtig testen (nicht nur mit *rsnapshot -t*) was rauskommt
- kleinere In-/Excludes lassen sich direkt in der Konfigurationsdatei eintragen; bei grösseren Listen ist eine eigene Datei dafür möglicherweise sinnvoller.
- bei der Eintragung der Backup-Hosts und -ziele auf das vorgegeben Dateiformat achten, sonst gibt es Fehler (*rsnapshot -tV*).
- Skripte können vor/nach Snapshot ausgeführt werden (*cmd_preexec/cmd_postexec*) : sehr praktisch; so kann zb das lokale Backupziel gemountet werden; remote können Vorgänge ausgeführt werden (/boot oä gemountet, DB gedumpte usw...). Bei Postexec genauso.
- bei *backup_script* wird das angegebene Skript ausgeführt und dessen Ergebnisse im angegebenen Zielverzeichnis gesichert. Praktisch für Datenbank-Dumps.

Vorteile:

- schnell
- relativ platzsparend: benötigt theoretisch nur Speicherplatz für ein Vollbackup und jeweils Platz für die Dateien, die sich geändert haben (*hard links*)
- flexibel
- lässt sich in andere Skripte einbauen
- zusätzliche Sicherheit und Vielfalt bei Backups durch verschiedene und andere Medien; zb täglicher Wechsel der snapshot_root oder zusätzliches Sichern des snapshot-Mediums auf Bänder
- Snapshot-root kann so konfiguriert und gemountet werden, dass User sich ihre Backups selbst zurückholen können

Nachteile:

- sichert immer komplette Dateien, keine Diffs
- für Netzwerksicherung SSH oder RSYNCD zwingend
- Dateisystem des Backupmediums muss hard links unterstützen
- auf manchen Wechselmedien (USB) Dateioperationen langsam
- rsync kann cpu-intensiv sein

Im Internet:

- rsnapshot.org
- rsync.samba.org