The Sysadmin's Daily Grind: Rsnapshot

TARGET PRACTICE

The principle behind Rsnapshot is nothing new: use Rsync and SSH to back up files to another computer. What sets Rsnapshot apart is its simple setup and its rich collection of features.

BY CHARLY KÜHNAST

Rsnapshot [1] gives admins a space-saving backup system. The Rsnapshot tool leaves long, configurable intervals between full backups and creates daily incremental backups on the target system; and this saves a lot of space compared to a full backup. The rotational logic that underlies the system makes heavy use of hardlinks.

Under the hood, Rsnapshot uses SSH and Rsync. I decided to install the 89KB tarball on one of my lab systems, which happens to run Slackware 10, but DEB and RPM packages are available as downloads from [1] if you prefer. The following commands were all it took to install the Perl-based system:

```
tar xvzpf ➔
rsnapshot-1.1.6.tar.gz
cd rsnapshot-1.1.6
./configure --sysconfdir=/etc
su
make install
```

The next step was to copy the sample configuration file

```
cp /etc/rsnapshot.conf.default ➔
/etc/rsnapshot.conf
```

and remove a comment sign

```
#cmd_cp    /bin/cp
```

This option is commented out to be compatible with BSD systems. Removing the comment sign enables GNU *cp*, and that gives you more functionality. The next step is to configure the intervals at which Rsnapshot will run. Let's tell Rsnapshot to create daily backups and keep them for a week:

```
interval  daily   7
```

Four Weeks are not a Day

I decided that keeping the backups for seven days was not enough. On the other hand, I did not want to keep even more daily snapshots. My solution was to configure weekly backups, one per week for example, and to tell Rsnapshot to keep the backup for four weeks:

```
interval  weekly  4
```

Incidentally, you need to use tabs as separating characters in the configuration file. As I want the daily backup to run at 11 pm, and the weekly backup at 3 am every Sunday, I need to add the following to the crontab:

```
0 23 * * * ➔
/usr/local/bin/rsnapshot
0 3 * * 0 ➔
/usr/local/bin/rsnapshot
```

Of course, Rsnapshot still doesn't know which files to back up. This simple line tells Rsnapshot to back up my home directory on the same filesystem:

```
backup  /home/  localhost/backups/
```

But what use is a safe copy on the same disk? Experience tells us that hard disks, or at least the ones I have owned, are prone to dying even if you treat them well. I need to save the backup on a remote machine for more peace of mind. As my file server has the IP 10.0.0.140, the following line will do the trick:

```
backup   /home/  ➔
charly@10.0.0.140:/backups/
```

Of course, this assumes you have working SSH access to the file server.

I have covered the basic functionality of Rsnapshot, but the program has a lot more bells and whistles. The excellent manpage and the HOWTO on the website have a wealth of interesting information, such as includes, excludes, and tips on how to back up open databases. Happy target practice with your backup snapshots! ■

INFO

[1] Rsnapshot: *http://www.rsnapshot.org*

THE AUTHOR

Charly Kühnast is a Unix System Manager at the data-center in Moers, near Germany's famous River Rhine. His tasks include ensuring firewall security and availability and taking care of the DMZ (demilitarized zone).