

Incremental backups with rsnapshot

A HOWTO guide for the QNAP TS-439 / TS-639, and an introduction to use of IPKG

Summary

rsnapshot is a file system backup utility based on rsync. Using rsnapshot is possible to take snapshots of your file system(s) at different points in time. Using hard links, rsnapshot creates the illusion of multiple full backups, while only taking up the space of one full backup plus changes. When coupled with SSH, it is possible to securely take snapshots of remote file systems as well¹.

This document describes how to set up rsnapshot on the QNAP TS-439 and TS-639 NAS boxes with a view to schedule a nested and rotating series of incremental backups (but I imagine that it would work on all x86 models, and perhaps others with minor modification). For example, you could schedule:

- Hourly incremental backups, for the past 24 hours.
- Daily incremental backups, for the past 7 days.
- Weekly incremental backups, for the past 4 weeks.
- Monthly incremental backups, for the past 3 months.

Why rsnapshot is so efficient: Hard links

Hard links are kind of like references to a file (go lookup inodes if you want a more accurate explanation). If you make additional hard links to a file you are just creating additional pointers to it. You can access, modify and copy the file through any hard link that points to it. However, deleting a hard link does not delete the file its points to *unless it is the last remaining link to it*. To put it another way, a file is not deleted from disk until the number of hard links pointing at it drops to zero.

So why should we care? Disk space! If we made a full physical copy of our file system each time we took a snapshot we would run out of disk space very quickly. It is more efficient to make a copy of the directory structure *using hard links instead*. Our snapshot will simply consist of *links* to the files already stored on the backup disk, plus full copies of any files that are new or have changed.

When files are deleted they will (of course) not appear in the next snapshot. However, they will still be available from *older snapshots* that contain hard links to them. If rsnapshot finds that some files have been modified, it will delink them and drop the new version into the next snapshot – but the *older versions* of the same files will still be available from older snapshots. So rsnapshot will give you a good time series of backups.

Requirements

You will need to install:

- **QPKG-IPKG**. This is a package manager that you can use to automatically download and install additional software packages onto your NAS, including rsnapshot.
- **rsnapshot**. This is a filesystem snapshot utility. It can take incremental snapshots of local and remote file systems. It makes use of rsync and hard links, so it is very space efficient.

Both of these programmes are excellent and easy to use. In the case of IPKG, you will doubtless be tempted to install some additional tools for your NAS after you have seen how easy it is.

Procedure

1. Install IPKG

IPKG is package manager that automates the download and installation of new software. We can use it to easily download and install rsnapshot plus all its dependencies in one go. The installation process is as follows:

- In the admin interface go to System tools => QPKG.
- Click on the 'Get QPKG' link and select Optware-IPKG.
- From the popup window, click on the appropriate link to download the right Optware-IPKG version for your QNAP model.
- Use the 'browse' button to select the package you downloaded and press 'upload' to install it on your system. Your QNAP needs a functioning internet connection for installation to work (otherwise it will hang at 60%).
- Click on the 'down arrow' next to the Optware-IPKG package to show the details – click the 'enable' button.

That's it! Like other package managers, IPKG can retrieve a list of available software packages from a 'feed' somewhere on the internet. To make IPKG retrieve the list, login to your NAS by SSH (Windows people can use PuTTY), and type:

```
ipkg update
```

To see the list of software packages available for installation type:

```
ipkg list
```

Or if you just want to see packages related to rsnapshot, you could type:

```
ipkg list | grep rsnapshot
```

2. Install rsnapshot

To download and install the rsnapshot package (and its dependencies), type:

```
ipkg install rsnapshot
```

Your NAS will retrieve and install everything for you. Too easy!!

2a. [Optional]: Install the nano editor

While you are at it, you might as well install a more user-friendly text editor like nano:

```
ipkg install nano
```

I'm going to stick to use of 'nano' in this post, but you can use the 'vi' editor instead if you prefer, which has the advantage of being installed on the QNAP by default.

3. Configure your backup jobs in rsnapshot.conf

You are now ready to set up your backup jobs. Basically, rsnapshot is controlled by a single configuration file, `rsnapshot.conf`, which you will find in the `/opt/etc` directory. This file governs:

- What directories are backed up (the source). You can backup directories from multiple remote machines if you want to. You can also have rsnapshot run scripts to backup databases etc.
- Where the snapshots are saved (the destination).
- How many snapshots (versions) of the source file systems you want to keep, and at what level (hourly, daily, weekly, monthly).

For instructions on how to configure this file (and set up your backup jobs), download the excellent rsnapshot HOWTO¹ guide from: <http://rsnapshot.org/howto/>. It is also worth downloading the rsnapshot manual (man page) from the same site, as it contains additional details about some of the options available.

It is **IMPORTANT** that you **READ** the [official] rsnapshot **HOWTO** guide **THOROUGHLY**, from section 4 (configuration) onwards, otherwise you are most certainly going to mess this up. The best way is to open `rsnapshot.conf` in your editor and carefully check each item as you read through the HOWTO one item at a time. To open the configuration file, type:

```
nano /opt/etc/rsnapshot.conf
```

Or if you insist on using the default editor, vi, you would type:

```
vi /opt/etc/rsnapshot.conf
```

Don't forget to save your changes before closing the file!

3a. Testing your rsnapshot.conf configuration file

rsnapshot provides an easy way to check that your configuration file doesn't contain any syntax errors. Simply type:

```
rsnapshot configtest
```

If all is well, it will say 'syntax ok'. If there is a problem, it will spit errors at you. If you get errors, check that you have separated items in the file using tabs (spaces are not allowed in the configuration file).

4. Schedule (automate) your backup jobs in crontab

Section 5 of the [official] rsnapshot HOWTO provides instructions on how to set cron jobs to run your backups automatically. However, the crontab file gets overwritten every time your NAS reboots, deleting any changes you may have made unless you take steps to make them 'stick'. Open your crontab:

```
nano /etc/config/crontab
```

Append your desired cron jobs, as discussed in the rsnapshot HOWTO. Note that it is **important** to make sure your higher-level backups run before the lower ones. As an example, I'm using:

```
0 */4 * * * /opt/bin/rsnapshot hourly
30 23 * * * /opt/bin/rsnapshot daily
30 22 * * 0 /opt/bin/rsnapshot weekly
30 21 1 * * /opt/bin/rsnapshot monthly
```

Save the modified crontab, then relaunch it to make sure your changes 'stick' when your NAS is rebooted (otherwise they will be deleted). You do this by typing:

```
crontab /etc/config/crontab
```

5. Enabling key-based authentication

To automate rsnapshot, you also need to set up key-based authentication over SSH on the remote machines that you want to backup, so that they can be accessed without need for a password login. And you need to make sure this arrangement survives a reboot. Do not be discouraged! It's easy.

Basically, you need to copy the public SSH key from your backup NAS (the machine running rsnapshot, where all the backups will be pulled to) into the `authorized_keys` file on the target NAS (the machine(s) that you want to backup). To do this, login to SSH on your ***backup NAS***, and type the following²:

```
scp ~/.ssh/id_rsa.pub admin@hostname:~/.ssh/authorized_keys
```

Where `admin@hostname` is the ssh login for the ***target* NAS**. You can use the hostname or its IP address, whatever you prefer. You will be prompted to enter the admin password for the target NAS. Do it.

The last thing you need to do is set the correct access permissions on the configuration folder in the ***target* NAS**, otherwise the key-based authentication won't work² and you'll get an 'access denied (public key)' error or password challenge when your backup NAS tries to login. The problem is the folder permissions get reset when the NAS is rebooted (note that if the target machine is a normal computer you won't have to worry about this). However, there is a special file on your QNAP that can be used to run commands on startup³. We are going to use it to restore our changes to the folder permissions on start up (reboot). Login to your ***target* NAS** by SSH and open the `autorun.sh` file:

```
nano /tmp/config/autorun.sh
```

Add the following three lines to the file, then save it:

```
mount -t ext2 /dev/sdx6/tmp/config  
  
chown admin.administrators /mnt/HDA_ROOT/.config  
  
umount /tmp/config
```

This works for the x86 QNAP NAS (models TS-439, TS-509, TS-639, TS-809, TS-809U). If you are using a different model, then you need to change the first line (see the QNAP Wiki for correct command):

<http://wiki.qnap.com/wiki/Autorun.sh>.

6. Testing that it all works, and some troubleshooting

Type `rsnapshot hourly` to manually run an rsnapshot job. If all is well, your NAS will start copying the file structure from the target directories into the destination that you specified. It is best to just run this test on a ***small*** target directory first (rather than hundreds of gigs of data) so you don't have to wait ages to confirm the result! You can also run it several times and see how the `hourly.0`, `hourly.1`, `hourly.2` etc. snapshots build up in your destination directory. Also notice that subsequent runs of rsnapshot are very fast compared to the initial run, because only changes are being copied across.

Once you have that working, reboot your NAS (both the target and backup machines) and manually run an `rsnapshot hourly` job again. This is to make sure that all your changes can survive the reboot process. If

they don't, you may need to go over the key-based authentication and rsnapshot configuration steps again, depending on the error message you observe, to make sure you have them right.

Be warned that the first run of rsnapshot will take ***ages*** if you have a lot of data as it needs to physically copy all the files, although subsequent runs are very fast if you don't have too many files to update. The first run of rsnapshot also puts a heavy load (100%) on the processor, so don't do it while your system is busy with other things. You can observe load average by running the 'top' tool in another shell.

7. Congratulations, you're finished

That's it. Now just keep an eye on the rsnapshot logs (see below) and your backup directory over the next day / week / month to make sure it is working and that your backups are successfully being pulled from the remote target server(s).

8. Monitoring rsnapshot

Monitoring and testing is an essential part of any backup procedure. There's nothing worse than finding out that your backup hasn't been working for six months on the day that your system crashes. rsnapshot has a logfile that records warnings and error messages. You can find/open it here:

```
nano /opt/var/log/rsnapshot
```

Make ***sure*** you inspect it regularly to make sure your backup jobs are running smoothly. If there are problems, they will be recorded here. You can increase or decrease the level of logging detail in the configuration file if you need to, `rsnapshot.conf`. See the official HOWTO for more information.

Enjoy!

Crushdepth (<http://www.crushdepth.net>).

References

1. **Rosenquist, N. (2004).** rsnapshot HOWTO. Available online at: <http://rsnapshot.org/howto/>
2. **QNAP Wiki.** How to set up authorized keys. Available online at: http://wiki.qnap.com/wiki/How_To_Set_Up_Authorized_Keys
3. **QNAP Wiki.** Autorun. Available online at: <http://wiki.qnap.com/wiki/Autorun.sh>