

Synology as a FreeNAS backup

Apr 3, 2016

For a little while I'd been looking for a secondary NAS of some sort, preferably cheap, to act as a backup for my FreeNAS Server.

I didn't have anything particular in mind and was considering various off the shelf units.

While browsing ebay late one night I stumbled across a Synology DS212j going very cheap and due to the limited time remaining, did no research whatsoever and went for it.



On arrival it was a bit of a pain to get working until I figured out that the existing partition tables on the hard drives I was using were interfering with the installation process. After wiping the drives and using the Synology Assistant software to install the latest Synology DSM (6.0) I was finally able to get to the web interface.

From there I created the RAID, set the hostname and the various network settings and started reading about how best to get my data onto my new Synology NAS from FreeNAS.

Unfortunately, this is where I started to run into problems. While Synology DSM does support rsync it's not without quirks as you will see.

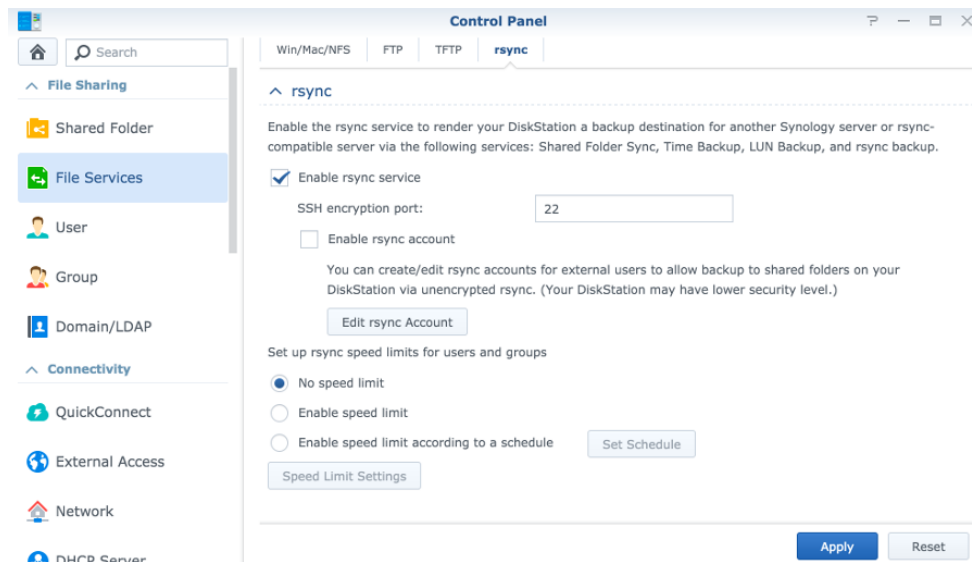
Users, Services, Permissions & Shares

Synology

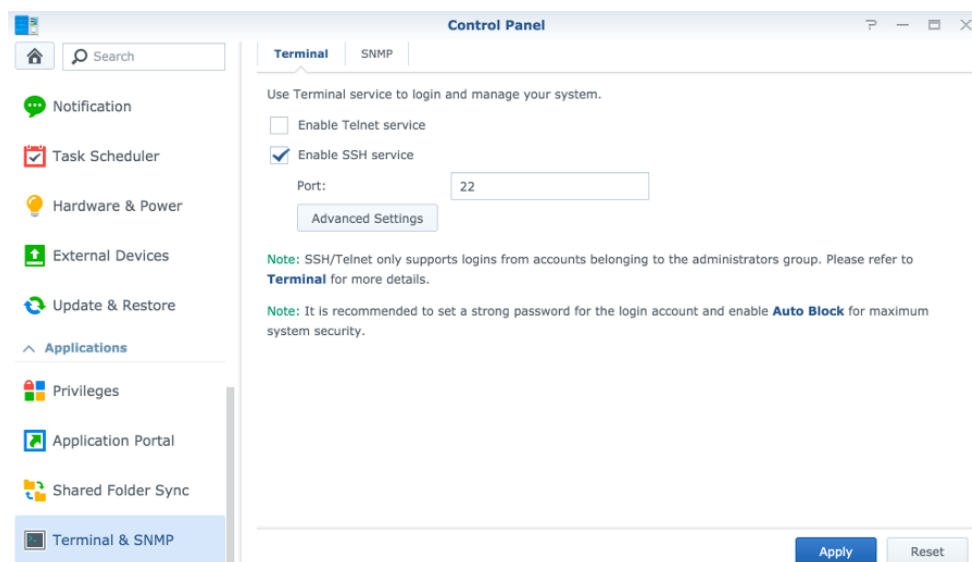
Personally I dislike using the built in Admin account for administration and so created a separate account for myself. I also created another separate service account for the single purpose of rsync backups. To use rsync over ssh rather than the rsync module, both accounts will need to be administrators.

Next I needed to enable the required services.

The rsync service can be enabled by navigating to the Control Panel > File Services > rsync, and selecting "Enable rsync service". Checking this option creates a new share called "NetBackup" which cannot be deleted or renamed without disabling the rsync service.



I then enabled the ssh service by navigating to the Control Panel > Terminal & SNMP > Terminal, and selecting "Enable SSH service".



To allow the new rsync service account to actually use rsync you have to assign the application permission which can be done by going to Control Panel > Privileges and editing the rsync permissions.

Info	User groups	Permissions	Quota	Applications	Speed Limit
Name	Preview	Group permissions	<input checked="" type="checkbox"/> Allow	<input type="checkbox"/> Deny	By IP
File Station	Allow	Allow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FTP	Allow	Allow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
rsync (Shared Fold..	Allow	Allow	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Rather than use the system default rsync share I decided to create a new one, giving my rsync user full permissions. Although I was unable to delete the NetBackup share I was able to set it to hidden and remove all permissions to it.

Name:	NetBackup
Description:	System default shared folder
Location:	Volume 1(Available: 823.02 GB) ▼
<input checked="" type="checkbox"/> Hide this shared folder in "My Network Places"	
<input type="checkbox"/> Hide sub-folders and files from users without permissions	
<input type="checkbox"/> Enable Recycle Bin	
<input type="checkbox"/> Restrict access to administrators only	
<input type="button" value="Empty Recycle Bin"/>	

Note: [How to set up a Recycle Bin emptying schedule](#)

At this point, I was able to manually rsync data to the NAS.

FreeNAS

In FreeNAS, rather than set up a task that runs under my own user account I decided to create a new account for the purpose. I required a home directory for the account and so I created a new dataset in my pool called "home" for this purpose.

I named the new account rsync-backup and used the following options:

User ID:	1002
Username:	rsync-backup
Primary Group:	wheel ▼
Home Directory:	/mnt/vol0/home/rsync-backu <input type="button" value="Browse"/>
Shell:	sh ▼
Full Name:	rsync-backup

The login shell option is temporary for configuration purposes and will be removed later.

SSH Key Authentication

For passwordless ssh key based authentication with rsync (necessary for an automated task unless using a password file), the user account on the target device must have an `~/.ssh/authorized_keys` file containing the public key of the sender.

Synology

To create this I ssh'd into the Synology as my rsync account to create the file, only to find that the user had no home directory. I found the correct location for the home directory as follows:

```
rsync@Synology:~$ echo $HOME
/var/services/homes/rsync
rsync@Synology:~$ cat /etc/passwd | grep rsync
rsync:x:1027:100::/var/services/homes/rsync:/bin/sh
```

I was then able to create the directory and file and set the permissions correctly as follows:

```
rsync@Synology:~$ sudo mkdir -p /var/services/homes/rsync/.ssh
rsync@Synology:~$ sudo chown -R rsync /var/services/homes/rsync
rsync@Synology:~$ touch /var/services/homes/rsync/.ssh/authorized_keys
```

I think that this method will be least likely to break due to an update.

FreeNAS

With the user account created I ssh'd into my FreeNAS box as root and used the following commands to switch to my new rsync account and create the ssh keys I would use for authentication.

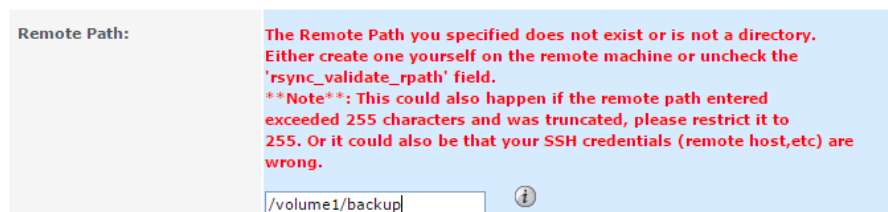
```
root@FreeNAS:~$ su rsync-backup
rsync-backup@FreeNAS:~$ ssh-keygen # Do NOT set a password for this key
rsync-backup@FreeNAS:~$ cat .ssh/id_rsa.pub
```

This output the newly created public key which I then copied and pasted it into the Public Key field for the rsync-backup user account in FreeNAS. I also pasted this key into the `authorized_keys` file I created earlier for the rsync user on the Synology NAS as the `ssh-copy-id` command does not appear to work in FreeNAS.

At this point it was possible to ssh from the rsync-backup user account on FreeNAS to the rsync account on the Synology without using a password:

```
rsync-backup@FreeNAS:~$ ssh 192.168.1.2 -l rsync # Substitute with the IP address of your
The authenticity of host '192.168.1.2 (192.168.1.2)' cant be established.
ECDSA key fingerprint is SHA256:e10151907f4ff16abe68224c2377c441403aad1051f75af482fbbc4c
No matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.2' (ECDSA) to the list of known hosts.
rsync@Synology:~$ exit
logout
Connection to 192.168.1.2 closed.
rsync-backup@FreeNAS:~$ ssh 192.168.1.2 -l rsync
rsync@Synology:~$
```

You will need to do this at least once to add your Synology to the list of known hosts on the FreeNAS server or rsync will fail from the GUI with the following error:



Rsync Task

At this point, everything was in place for passwordless rsync over ssh from my FreeNAS server to my Synology NAS.

I created the rsync task in FreeNAS using the documentation here as follows:

Tasks > Rsync Tasks > Add Rsync Task

Path:	<input type="text" value="/mnt/vol0/Media"/>	Browse
User:	<input type="text" value="rsync-backup"/>	
Remote Host:	<input type="text" value="rsync@Synology"/>	
Remote SSH Port:	<input type="text" value="22"/>	
Rsync mode:	<input type="text" value="Rsync over SSH"/>	
Remote Path:	<input type="text" value="/volume1/backup"/>	
Validate Remote Path:	<input checked="" type="checkbox"/>	
Direction:	<input type="text" value="Push"/>	
Short description:	<input type="text" value="Music -> Synology"/>	

I was then able to set the schedule and test the task by selecting it and clicking “run now”.

With everything working I went back and set the login shell for the FreeNAS rsync-backup user to “nologin”.

This appears to be working well and I have yet to have a failure.

NB: Hostnames, Usernames, IP addresses, paths and pubkey's/fingerprints were changed for the purposes of this post. Please substitute with your own.

I use the vi editor in both FreeNAS and Synology as it's available and I'm familiar with it. Feel free to use the editor or your choice