

Off-Site Backups for Personal and Small Businesses

- Introduction
- Solution
 - Hardware
- Synology DS212j Setup
 - Perform the following Update
 - Setup the Volumes
 - Install ipkg
 - Introduction
 - Steps
 - Install Packages
 - Common Errors Installing Packages
 - Setup Remote Backup User
 - Creating the Backup Destination
 - Scripting
 - Adding the Cronjob
 - Connecting with Clients
 - Mac OS X Auto Mount
 - Mac OS X Hidden Mount with GUI
 - Connect to Server
 - Mac OS X Mount Hidden Share with CLI
 - Mac OS X Unmount Hidden Share with GUI
 - Mac OS X Unmount Hidden Share with CLI
 - Linux Mount with CLI
 - Improving the Automatic Backup
 - CrashPlan GUI with Ubuntu Desktop
 - Transfer speed fix test
- References

Introduction

I have since come up with a better solution using [Crashplan](#) rather than hacking Synology.

This article is workable, but not up to par in quality to be considered complete.

The requirements outlined here are for,

- Off-site (from where the servers are located) backup for small businesses.
- Double as a centralized storage area for a small or home office
- Provide media functionality for a small home (which also doubles as an additional backup site)

Solution

Synology which has an easily hackable system and one of the best software platforms we have seen.

Hardware

- DS212j using the Marvel Kirkwood mv6281 ARM chipset with 16-bit@DDR2, 256MB of RAM
- Two 3 Terabyte Drives

Synology DS212j Setup

Perform the following Update

- update firmware
- sync the time server
- disabling cache management (when UPS N/A)

Setup the Volumes

- S.M.A.R.T. test
- enable the home directory for SSH

Install ipkg

Introduction

ipkg is the packaging system for the lightweight debian based linux system provided by Synology.

At a high level, to install ipkg, as **root**,

- Download and run a script which the community calls a bootstrap file specific to the NAS processor hardware
- Modify `.profile` to include ipkg in the path

Steps

The following procedure was successful with DSM 4.0-2233.

Determine the processor of your NAS. The DS212j uses the Marvel Kirkwood mv6281 ARM chipset with 16-bit@DDR2, 256MB of RAM.

Look at the [Synology wiki](#) to determine what bootstrap that matches the NAS processor hardware.

Not sure why by default the links point to the unstable directory. However, at least for the version used here the bootstrap in unstable and stable are identical.

The **special** bootstrap instructions on the wiki as of May 5, 2012 for DSM 4.0 do not seem complete nor correct. Using DSM 4.0-2233 did not result in errors so ignore the **special** bootstrap instructions,

NEW: If you have DSM 4.0 there is an additional step. In the file `/root/.profile` you need to comment out (put a `#` before) the lines `"PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/syno/sbin:/usr/syno/bin:/usr/local/sbin:/usr/local/bin"` and `"export PATH"`. To do this enter the command `"vi /root/.profile"` to open the file in `vi`. Now change `vi` to edit mode by pressing the `"i"` key on your keyboard. Use the down cursor key to move the cursor to the start of the line `"PATH=/sbin..."` and put a `"#"` in front of this line so it is now `"#PATH=/sbin..."`. Do the same for the line below so it is now `"#export PATH"`. Now press the escape key (to exit edit mode) and type `"ZZ"` (note they are capitals) to tell `vi` to save the file and exit. For background info on why this is necessary for DSM 4 refer to <http://forum.synology.com/enu/viewtopic.php?p=185512#p185512>

Instead following the **normal** bootstrap installation instructions, log in through ssh as **root** and download the bootstrap,

```
cd /volume1/@tmp
wget
http://ipkg.nslu2-linux.org/feeds/optware/cs08q1armel/cross/unstable/syno-
mvkw-bootstrap_1.2-7_arm.xsh
```

Make sure to download the bootstrap that matches the NAS processor hardware!

Run the installer,

```

DiskStation> chmod +x syno-mvkw-bootstrap_1.2-7_arm.xsh
DiskStation> ./syno-mvkw-bootstrap_1.2-7_arm.xsh
Optware Bootstrap for syno-mvkw.
Extracting archive... please wait
bootstrap/
bootstrap/bootstrap.sh
bootstrap/ipkg-opt.ipk
bootstrap/ipkg.sh
bootstrap/optware-bootstrap.ipk
bootstrap/wget.ipk
1232+1 records in
1232+1 records out
Creating temporary ipkg repository...
Installing optware-bootstrap package...
Unpacking optware-bootstrap.ipk...Done.
Configuring optware-bootstrap.ipk...Modifying /etc/rc.local
Done.
Installing ipkg...
Unpacking ipkg-opt.ipk...Done.
Configuring ipkg-opt.ipk...WARNING: can't open config file:
/usr/syno/ssl/openssl.cnf
Done.
Removing temporary ipkg repository...
Installing wget...
Installing wget (1.12-2) to root...
Configuring wget
Successfully terminated.
Creating /opt/etc/ipkg/cross-feed.conf...
Setup complete.
BusyBox v1.16.1 (2012-06-06 04:34:01 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
DiskStation> view /usr/syno/ssl/openssl.cnf
/bin/sh: view: not found

# Note it looks like the the openssl.cnf WARNING is normal.

```

Edit the root account's .profile file and ensure the **/opt/bin** is located at the **beginning** of the path,

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin:/usr/syno/sbin:/usr/syno/bin:/usr/local/sbin:/usr/local/bin
```

```
vi ~/.profile
```

You final bash should look like this,

```
umask 022

PATH=/opt/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/syno/sbin:/usr/syno/bin:/usr/local/sbin:/usr/local/bin
export PATH

#This fixes the backspace when telnetting in.
#if [ "$TERM" != "linux" ]; then
#    stty erase
#fi

HOME=/root
export HOME

TERM=${TERM:-cons25}
export TERM

PAGER=more
export PAGER

PS1="\`hostname`> "

alias dir="ls -al"
alias ll="ls -la"
```

Log out and the terminal and log back in as **root**,

Verify ipkg is working and at the same time update the package list,

```
ipkg update
Downloading
http://ipkg.nslu2-linux.org/feeds/optware/cs08q1armel/cross/unstable/Packages.gz
Inflating
http://ipkg.nslu2-linux.org/feeds/optware/cs08q1armel/cross/unstable/Packages.gz
Updated list of available packages in /opt/lib/ipkg/lists/cross
Successfully terminated
```

The ipkg update dialog will show the repository being used. In the above example, load a browser and go to <http://ipkg.nslu2-linux.org/feeds/optware/cs08q1armel/cross/unstable/> to see the list of software available for installation on their site.

Install Packages

Installing packages with ipkg is similar to using apt-get with Debian or Ubuntu. Synology keeps a [manual online](#) for reference. You should though be able to get by with the following common commands,

```
ipkg list # lists the available packages
ipkg install rsync # example of installing rsync
```

With this setup, we install the following,

```
ipkg install rsync # installs rsync for making backups
ipkg install htop # nice monitoring of system
ipkg install mlocate # easily find files
```


Common Errors Installing Packages

```
ipkg_conf_init: Failed to create temporary directory `(null)': Permission
denied
```

The reason for this error is that you are not logged in as root.

Setup Remote Backup User

Rather than using root to pull down data from other system we will use remotebackup.

The remotebackup user could not be created using the shell because it was not possible to add the user to groups, change the password or specify a UID upon creation of the user. The user was not recognized by the system.  So Roderick did you just use the UI? Then you can't define the uid.

Create the backup group in the command line and give it the GID of 34 to follow the ubuntu standard,

```
addgroup -g 34 backup # This will not work.
```

Instead, add the remotebackup user manually by editing the /etc/group file

```
vi /etc/group
backup:x:34:remotebackup
```

Now we change the UID to 3001 [following bonsaiframework standard](#) and give shell access for remotebackup by editing the /etc/passwd file but do not forget to backup first

```
cp /etc/passwd ./passwd.yr-mo-dy.v0.0.username.bck
vi /etc/passwd
remotebackup:x:1000:100:Remote
Backup:/var/services/homes/remotebackup:/sbin/nologin
#change the number "1000" to 3001 and the "/sbin/nologin" to /bin/sh
remotebackup:x:3001:100:Remote
Backup:/var/services/homes/remotebackup:/bin/sh
```

To allow serveradmin to login, change the default shell from nologin to /bin/sh. The chsh command is not currently available on the package site so you must edit the passwd file manually,

```
cd ~
cp /etc/passwd ./passwd.2012.08.12.v0.0.tpham.bck # Make a backup first.
sudo vi /etc/passwd # Edit the file with vi or whatever is your favourite
editor.
```

We also need to change the environment home folder,

```
echo "HOME=$PWD" > .profile
```

Verify that serveradmin can log in,

```
su - remotebackup

BusyBox v1.16.1 (2012-06-06 04:34:01 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.
DiskStation> pwd # Check that your home directory is right.
/volume1/homes/remotebackup
DiskStation>
```

Add the [private keys](#) to remotebackup required to log into other systems to transfer backups.

Creating the Backup Destination

The backup destination will only have r/w access by remotebackup

...

Scripting

Note that the ash is the default shell. Synergy selected ash because it is a lightweight version of bash and **generally** compatible.

Creating the [rsync](#) script

```
#!/bin/bash
su - remotebackup && rsync -av --delete -e ssh remotebackup@ip:/source/
/destination/
#you must be remotebackup user to run this command in the script
```

To test run scripts the command is

```
sh script.sh
```

Adding the Cronjob

To add the script to cron edit the crontab located in /etc/crontab make sure you are root.

If you are unsure of how to schedule time view the [cron page](#)

```
vi /etc/crontab
0TAB5TAB*TAB*TAB*TABrootTAB/path/to/script.sh #The cronjob must be run by
the root user or the cronjob will not work and you must use TABS instead of
spaces.
```

When adding or removing commands to the crontab make sure to restart the crond service so the commands take effect.

```
synoservice --restart crond
```

To view the logs to see if your cronjob was run is located in /var/log/messages

Connecting with Clients

Mac OS X Auto Mount

Open Finder and click DiskStation on the left tab.

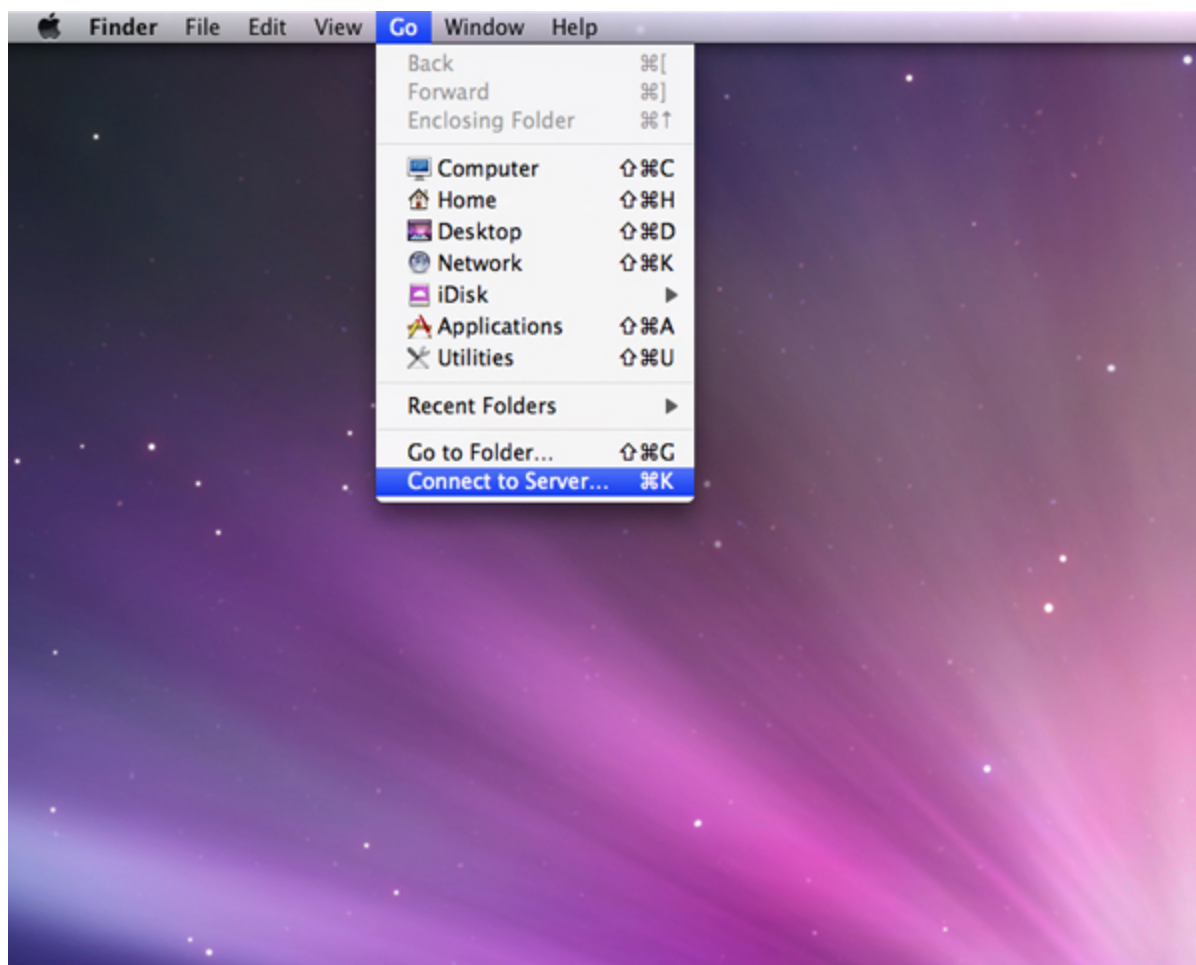
Look to top right and click button "Connect As..."

Mac OS X Hidden Mount with GUI

To mount a hidden share as a specific user perform the following steps. It is assumed that in DiskStation Mac file service has been enabled.

Connect to Server

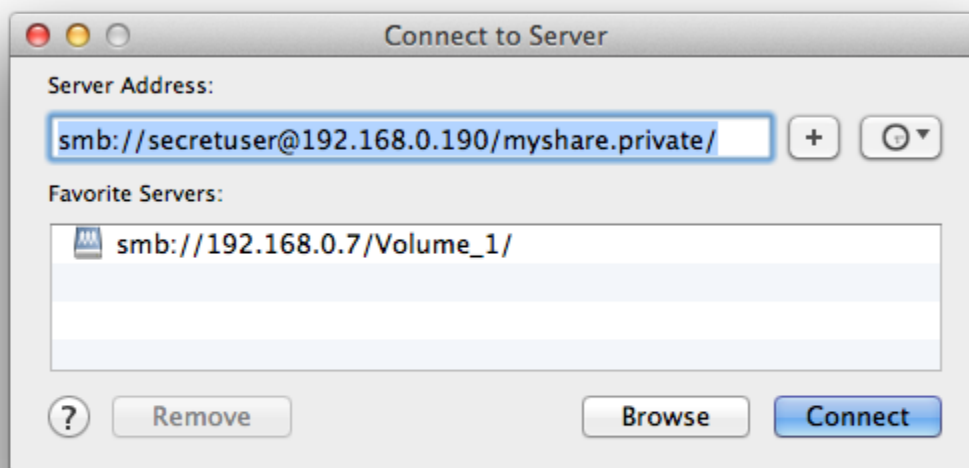
Use the key combination **command-k** or choose **Go > Connect to Server** from the **menu** bar.



Type the following,

smb://secretuser@192.168.0.190/myshare.hidden/

Type Synology DiskStation's IP address or server name proceeded by smb:// or afp://, the id of the user to log in with, the share path and click **Connect**,



Put here which protocol (smb or afp) is better to use.

According to the [Synology website](#), it is better performance, it is recommended that you connect to the shared folders via SMB.

Enter the user credentials with authentication to access the shared folder. And then click **Connect** to connect to the shared folder.

Now this network share will not show up in the SHARED listing in the file manager. Instead look for the share in the **/Volumes** folder. In this example, **/Volumes/myshare.private/**.

Mac OS X Mount Hidden Share with CLI

The advantage of the CLI (Command Line Interface) is that it is not obvious to another casual user that you had mounted the hidden share and (if you got to research this) you can delete the history entries quickly and remove all traces of the private share.

```
mkdir /Volumes/myshare.hidden # This can actually be any folder, but kept
here by convention and generally matches the share folder.
mount -t smbfs //secretuser@192.168.0.190/myshare.hidden/
/Volumes/myshare.hidden
```

The command **mount_smbfs** is a wrapper for "mount -t smbfs" so the following command sequence will also work, though according to the man page for mount_smbfs we should use mount -t,

```
mkdir /Volumes/myshare.hidden/ # This can actually be any folder, but kept
here by convention and generally matches the share folder.
mount_smbfs //secretuser@192.168.0.190/myshare.hidden/
/Volumes/myshare.hidden/
```

...

Isn't there a way to not have to manually make the directory before mounting?

...

Mac OS X Unmount Hidden Share with GUI

Do not know how to do this yet. Please share if you do.

Mac OS X Unmount Hidden Share with CLI

To be extra secure, unmount your hidden share when you have finished using it. Go to the command line and use the **umount** command. In this example it would be,

```
umount myshare.private # This command can be executed from any path.
# Look for command to remove unmount and mount entries from history.
```

How to clear history of last command - <http://thoughtsbyclayg.blogspot.ca/2008/02/how-to-delete-last-command-from-bash.html>

Linux Mount with CLI

Specifically tried with Lubuntu,

```
sudo apt-get install nfs-common
showmount -e 192.168.0.190 # List available shares
sudo mkdir /mnt/myshare.hidden
sudo mount 192.168.0.190:/volum1/myshare.hidden /mnt/myshare.hidden/
# Success as the actual user, not sure what happens if I don't create the
actual user or how to use a different name yet.
```

Improving the Automatic Backup

- Progress log
- Start and stop process times
- Time span
- File integrity - CRC checks
- Emergency Alerts
- Security restricting terminal access and permissions to remotebakup
- Scalability - backup files that get too large

It turns out we [can not create our own users with specific UIDs under 1024...](#) so that makes backing up and restoring with proper UIDs a bit more challenging. Maybe storing and then restoring UIDs during and after backup.

CrashPlan GUI with Ubuntu Desktop

If you want to use the GUI with Ubuntu Desktop,

```
/usr/local/crashplan/bin/desktop.sh
```

Transfer speed fix test

<http://forum.synology.com/enu/viewtopic.php?f=14&t=44749&start=30>

References

VNC Autostart - <http://blog.johngoulah.com/2013/01/ditching-vino-for-x11vnc/>

Auto Mounting - <https://help.ubuntu.com/community/Autofs>

File Transfer Speed Test - <http://askubuntu.com/questions/17275/progress-and-speed-with-cp>