Using Network Attached Storage with Linux

by Andy Pepperdine

I acquired a WD My Cloud device to act as a demonstration, and decide whether to use it myself later. This paper is my experience of how to use it from a Linux system (Mint 17 Mate).

Introduction

Network Attached Storage (NAS) is the name for a device that you plug into your local network and can access from any other device on the network. In principle, you could also access it from the wider Internet, but for this demonstration, I will not consider that case.

The WD device is small, quiet and uses little power – ideal to place next to your router out of the way, and connect it directly to the router via one of the routers ethernet ports. It runs Linux itself, and WD will supply the source code on request, something to check with any other such device.

However, there is no encryption on it by default, so if the device gets stolen, the drive could be removed, inserted into another machine and read. Access to the various data can be controlled by passwords, however, so separation by user is possible. There is a public area that anyone can access. Other private areas are created when accounts for users are created on the device.

For this paper, I am not going to change the default settings for names in order to make it clear what is going on. I would recommend that you should change such things to better suit your environment and make them less likely to conflict with anyone else.

Other devices are available, and I suspect they have similar features. Read the manual.

Methods of access

Files on the NAS can be seen from Windows via Windows normal sharing features, and the manual contains full descriptions of how to do that.

They can be seen from an Apple device by a standard means, and again the manuals tell you what you need to know.

WD provide downloadable applications for each of Apple and Windows to give natural access to the remote device. Those cases are provided with good support and full documentation "out of the box", but I have not been able to test as expected.

For Linux, there is no explicit support. However, it is possible to see the Windows shares via Samba, the Linux interface to Windows networks. It is also possible, after some work, to set it up so that access can be made to NFS shares as the normal Unix method of sharing over a network, and that is what this paper is primarily about.

Warning!

If you use the Samba share method with Linux be very careful how you name your files to be placed on the NAS. Files with names containing a colon (:) may not be addressed correctly, and may create files that cannot be deleted, accessed, or renamed through the Samba route. This caused me some considerable problems initially, but fortunately I found it early enough so that there was only test data on the NAS and it was possible to do a machine reset, deleting all data, and not lose much.

Preparation

This is easy, just plug it in and it takes about three minutes to boot up first time.

The simple settings and manipulation are through what they call the dashboard, and that is seen via any internet browser by pointing it at the correct IP address on your local network.

To see where it has been placed, you could find out by looking in your router connection list, but that is probably a last resort. A more likely method, assuming a sane router, is to put the url *wdmycloud.local* into the url field of your browser. If you need the actual IP address later, then another method is to open a terminal window and use the command

```
ping wdmycloud.local
```

then look at the output produced.

A similar method would work for other devices – see your documentation.

For more complex settings, you will have to remotely login to the device via secure shell, *ssh*.

As a general point, the recommended way to obtain the IP address of a Linux machine on its local network is issue the command:

```
hostname -I
```

so, you can use this after you have logged in to the WD device to ensure you get the correct address. For NFS access to the shared data, you must know its IP address.

Setting up user and share

We will follow the instructions in [1], which are very clear and I will not repeat most of them.

But it is worth reiterating something. When logging on first with *ssh*, using the command in a terminal window:

```
ssh root@wdmycloud.local
```

it will ask you whether you want to continue as the authenticity cannot be verified. Answer Yes in order to continue. It will then prompt you for the *root* password (the default currently is welc0me). As soon as you have negotiated this, change the password to something more secure. Note that logging on to the *root* user will give you untrammelled access to everything. The *admin* user which you normally see on the dashboard will provide access to the essential parts of the file system operations, but not to the whole system.

So when logged in as *root*, you need to be exceptionally careful what you are doing. Mistakes could completely ruin the device.

To change the password, after logging in via *ssh* issue the command:

```
passwd
```

and follow the prompts.

Fortunately, by default the set up does not allow access from the Internet, only from the local network, so you have time to change the password before opening up the device to the whole world. If you do allow access to files from everywhere, make sure you choose a strong *root* password.

Access permissions

The way NFS is set up on the device is that permissions are handled by matching the UIDs of users on each of the machines. So if the UID on your laptop for your user is, say, 1001, then so must the UID be on the WD device.

To find your UID on any machine, run the command:

```
id -u
```

and note the resulting *userid*.

Unfortunately, the WD starts numbering at 1001, and Mint starts at 1000. It is easier to change the WD device to match than your working machine, since initially there will be no files on the WD device whose ownership ID has to be changed. So as soon as you have created a user on WD, change its UID by starting an *ssh* connection and do the following:

Get the current UID, and see if it needs changing (*username* is the name of the user whose UID needs to match):

```
id -u username
```

and note the value *oldid*. Make sure you do this so that you know what has to be changed elsewhere in the system.

Then change it to the desired value:

```
usermod -u userid username
```

Where *userid* is the value you want it to be.

If you have any files owned by this user, then you should change the ownership ID on all of the files:

```
cd /
find . -uid oldid -exec chown userid {} \;
```

where *oldid* is what it was, and *userid* is what you want it to be.

If you have only just created the user or share, there will be no files that need modifying.

Access the shared data from a client

Now that you have the device set up, we have to get to the data from another machine, and in Linux that might mean some changes to the configuration. Here are some things to look out for and which you may forget if you haven't recently changed it.

First, make sure that you allow the device to see you. If you have changed the file */etc/hosts.allow*, then make sure that your device is allowed, e.g. by adding the line:

```
ALL: 192.168.0.11
```

Put the actual IP address of your device in here.

You will also need to ensure that the */etc/default/nfs-common* file contains the line:

```
NEED_STATD=yes
```

Preparing the mount points

You will need dummy directories on your client machines to mount the shared data from the device on. I created directories under the /home area to match each of the nfs shares I wanted, like this:

```
sudo mkdir /home/nas /home/nas/Public
```

Because these are publicly available, I set the access on that directory open:

```
sudo chmod 777 /home/nas/Public
```

Mounting the shared area

This is done via a mount command:

```
sudo mount.nsf4 -o udp,vers=3,soft,intr,rsize=8192,wsize=8192 \
   192.168.0.11:/nfs/Public /home/nas/Public
```

Now you can test that you can see the public area through the directory */home/nas/Public* and create files and read them again.

Mounting on start up

It is not really convenient to have to manually mount every time you boot up, but there is a way of making the mount happen automatically by editing the */etc/fstab* file on the client by adding the line:

```
192.168.0.11:/nfs/Public /home/nas/Public nfs4 \
   _netdev,vers=3,user,rsize=8192,wsize=8192 0 0
```

The *_netdev* option will stop it trying to contact the device if there is no network available.

The *user* option will allow any user to mount the shared data instead of restricting it to the superuser.

Getting to your own private shared data

First create a mount point as for *Public*, for example at */home/nas/andy* by creating that directory.

By placing a similar line to the Public line in */etc/fstab* for the private share, then the user can mount his/her private share simply by issuing the command:

```
mount /home/nas/user
```

and the parameters will be picked up from the */etc/fstab* file. Such a command can then be inserted into the user's *.profile* file in order to give immediate access after logging in.

If you want the mount of a user's area to be performed only when the user logs in, and not at start up, then make the */etc/fstab* line look like this:

```
192.168.0.11:/nfs/andy     /home/nas/andy  nfs4 \
     _netdev,noauto,vers=3,user,rsize=8192,wsize=8192  0  0
```

The *noauto* option will suppress the automatic mounting on start up. Instead, you can include in the user's *.profile* file in the home directory, the lines:

```
if ! mountpoint -q /home/nas/andy; then
  mount /home/nas/andy
fi
```

which will mount the user's NAS area if it is not already mounted.

Non-local access

If you want to get to your device from outside your home, then there are a number of things to be aware of and to check. I have not yet tried any of this, so these are just points to remember.

First, ensure that your ISP will allow incoming signals to your home router. Several do not, since they do not want your line to be used by a server. In some cases, they restrict only certain types of signals, others will disallow all of them. So check.

Second, ensure that your router can pass through the relevant request signals by mapping a router port to your device, and make the necessary changes.

Third, follow the manual instructions to turn on access from the Internet. I would be interested in any efforts to do this and the results.

Encryption

The WD device comes with no encryption features, so all the data stored will be visible by any thief who takes the whole device and removes the disk to read it.

On Linux, it is possible to create an encrypted file system on top of the nfs shares, and so you can keep your secrets with your own locks.

Apple may have similar features, but I do not know about Windows.

References

[1] To set up NFS access, follow the instructions at the two urls below:

http://ubuntuforums.org/showthread.php?t=2247237

http://ubuntuforums.org/showthread.php?t=2247242