Building A Pi Based NAS

Table of Contents

1 Colophon

2 Introduction

This is a guide to building a NAS[1] using a Raspberry Pi SBC.

It is assumed that the reader is familiar with the Linux command line and at least one text editor. Desktop users will need to open a terminal to execute many of the commands in this guide.

While aimed at Raspberry Pi OS, much will apply to any Linux distribution.

2.1 What's Not Covered

- OS installation.
- Networking configuration.
- Active Directory and other professional features and environments such as centralised user management.
- Clients other than Linux and Windows.
- iSCSI, ATAoE, NBD, and other SAN[2] protocols.
- Use of the NAS as a network boot and/or NFS root server.
- Use of the NAS as a network print server.
- Windows versions other than Windows 10 and 11.

2.2 Requirements:

- An internet connection for software update and installation.
- Raspberry Pi (any model) with a configured and working network connection.
- One or more directories on the Pi that you wish to share.
- An understanding of how Linux file/directory owner, group, and permissions work.
- For a headless[3] server, VNC or ssh enabled.

[1] **N**etwork **A**ttached **S**torage

[2] **S**torage **A**rea **N**etwork

[3] I.E. no monitor, mouse, or keyboard connected.

2.3 Conventions

```
Text like this indicates input to or output from the command line.
```

`Text like this` also refers to full or partial commands but is not generally intended to be entered into the command line as is.

“SD card” refers equally to full size and micro SD cards. IF should also be taken to refer to any boot storage medium in use.

Where “pi” occurs as a username, replace as appropriate.

“server” refers to the NAS, “client” to any device that accesses resources provided by the NAS.

“l-space” is a sample hostname. Replace with the actual hostname of your server.

All non system paths used in this guide follow the Filesystem Hierarchy Standard[4] though this is not mandatory.

2.4 Notes

- The file system in use on the server does not matter to the client. The client is not even aware of what it is.
- Versions of Windows 10 and 11 except Home and Pro have guest access to remote shares disabled by default. See https://docs.microsoft.com/en-us/troubleshoot/windows-server/networking/guest-access-in-smb2-is-disabled-by-default
- Windows 10 and 11 have SMB v1 disabled by default. Network browsing will not see the NAS without this enabled or WSDD installed on the server.
- The version of Samba available for Raspberry Pi OS Bullseye has SMB v1 disabled.
- NAS software (Samba, NFS, etc.) cannot[5] grant permissions to clients that the server’s file system does not grant to it.
- The SMB/CIFS[6] protocol does not support Linux owner, group, and permissions. Windows clients don’t use them, Linux clients set them in the driver at mount time.

[4] https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

[5] Not strictly true. Clients will think they have the permissions but attempts to exercise them will fail.

[6] As used by Samba.

2.5 References

https://www.raspberrypi.com/documentation/

https://www.samba.org/

https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html

https://github.com/thagrol/Guides/blob/main/fstab.pdf

`man` pages for `samba`, `smb.conf`, `nfs`, `exports`, `mount`, `fstab`, and `fdisk`.

3 A Minimal NAS

This is about as minimal as it can get on the NAS.

1. Enable ssh:

   ```
   sudo touch /boot/ssh
   sudo reboot
   ```

Clients will need to use an appropriate SFTP program. The server cannot be mapped to a drive letter or browsed to on windows clients. Linux clients may also mount the NAS using sshfs.

Client users will be able to access the server's entire file system subject to permissions granted to their user.

Additional users can be created per section 6.2.1

Note: The default user name and password for Raspberry Pi OS are well know. It is recommended to at least change the password when enabling ssh especially if opening it up to the internet via port forwarding on your router.

3.1 Using sshfs On Linux Clients

Install sshfs:

```
sudo apt update
sudo apt install -y sshfs
```

To mount a server:

```
sshfs user@host:directory mount point
```

Enter password when prompted.

For example:

```
sshfs pi@raspberrypi:/home/pi $HOME/remote
```

To unmount:

```
fusermount -u mount point
```

For example:

```
fusermount -u $HOME/remote
```

Sudo is not required and should not be used.[7]

Key based authentication can be used and is configured in the normal way for ssh.

`user@` is optional. If not provided the name of the current user will be used.

[7] Because: security. It will also likely not work as on Raspberry Pi OS root has no password.

4 Install Server Software

This will install but not configure the required server side software. Samba (SMB/CIFS) is more widely supported than NFS but has significant limitation for Linux clients[8] due to its origin and age[9].

While better for Linux clients, NFS is not supported on Home editions of MS Windows.

1. Ensure your OS installation is up to date:

   ```
   sudo apt update && sudo apt upgrade -y
   ```

2. Install samba

   ```
   sudo apt install -y samba samba-common-bin smbclient cifs-utils
   ```

3. Install the NFS server:

   ```
   sudo apt install -y nfs-kernel-server
   ```

[8] More on this later.

[9] c.1992 with Windows 3.11 (Windows for Workgroups) though first designed in 1983 at IBM.

5 A Quick, Dirty, and Insecure Samba Server

Quick because it’s a minimal configuration. Dirty because it’s a minimal configuration and because samba is not ideal for Linux clients. Insecure because it gives all permissions to all users and does not require login.

That said, this may be good enough in some circumstances – where security is not required and all remote users and devices are trusted.

If you have not already done so, install Samba per section 4.

5.1 Configure Samba

1. Stop the samba server:

   ```
   sudo systemctl stop smbd nmbd
   ```

2. If it does not exist, create a directory to share:

   ```
   sudo mkdir -p /srv/shared
   ```

3. Open permissions on it:

   ```
   sudo chmod 777 /srv/shared
   ```

4. Backup your current smb.conf:

   ```
   sudo cp /etc/samba/smb.conf /etc/samba/smb.bak
   ```

5. Open /etc/samba/smb.conf in your preferred text editor. You will need to be root or use sudo.

6. Replace its contents with the following:

   ```
   [global]
     log file = /var/log/samba/log.%m
     server role = standalone
     map to guest = bad user

   [shared]
     path = /srv/shared
     read only = no
     guest ok = yes
     force create mode = 666
     force directory mode = 777
   ```

   See section 5.4 for a brief explanation of the above smb.conf.

7. Save and close.

8. Verify smb.conf:

   ```
   testparm
   ```

9. Fix any errors and reverify.

10. Start the samba server:

    ```
    sudo systemctl start smbd nmbd
    ```

5.2 Client Access

5.2.1 Linux Command Line

Remote shares can be made available via the `mount` command this normally requires root privileges. For example:

```
sudo mount -t cifs -o guest,rw,file_mode=666,dir_mode=777 //l-space/shared /mnt
```

`l-space` is the name or IP address of the NFS server. `/shared` is the share name as defined in the server's smb.conf.

Remote shares may also be added to /etc/fstab. For example[10]:

```
//l-space/shared /mnt cifs
defaults,nofail,_netdev,guest,rw,file_mode=666,dir_mode=777 0 0
```

As with the share definition, above, these mount options will allow full access by all users on the client though files and directories will appear to be owned by root.

The meaning of the mount options are:

`guest`	Use guest access. Do not prompt for password.
`rw`	Mount read/write.
`file_mode=666`	Override default file permissions.
`dir_mode=777`	Override default directory permissions.
`defaults`	Use default mount options
`nofail`	Allow boot to continue if the mount fails. Only useful; in fstab with auto (and defaults) mounts.
`_netdev`	Flag to the system that the mount requires the network to be up.

For further mount options refer to the documentation for the `mount` and `mount.cifs` commands.

For further information on /etc/fstab please refer to its documentation or to my guide at https://github.com/thagrol/Guides.

5.2.2 Linux Desktop

Mounts made via the command line or /etc/fstab will be available via the chosen mount point.

Most file managers can access shares by entering `smb://server/share` e.g. `smb://ankh/shared` in the address field.

Network browsing may be functional but only if both the server and client have the SMB v1 protocol enabled.

Please refer to the documentation for your chosen file manager for more details.

[10] An fstab entry must be on a single line. That may not be apparent here due to document formatting.

5.2.3 Microsoft Windows

Shares can be accessed by entering `\\server\\share` e.g. `\\l-space\shared` into the address filed of File Explorer or This PC. Shares may be assigned a drive letter via “Map network drive”.

Network browsing may be functional but only if both the server and client have the SMB v1 protocol enabled or the server has wsdd installed (see 14.2.2). SMB v1 is disabled by default on Windows 10.

Please refer to the documentation for your version of Windows and to section 2.4.

5.3 Adding a User’s Home Directory

It may be desirable to allow access to a user’s home directory. As with the rest of this section, the method below is insecure as any remote user can access the share with full permissions and without login.

1. Backup your smb.conf. You may wish to rename any existing backup first so it is not lost.
2. Open `/etc/samba/smb.conf` in your preferred text editor. You will need to be root or use sudo.
3. Add the following to the end of the file.

   ```

   [pi]
     path = /home/pi
     read only = no
     guest ok = yes
     force user = pi
     force group = pi
   ```

4. Save and close
5. Verify your new smb.conf:

   ```
   testparm
   ```

6. Fix any errors and reverify.
7. Restart the samba server:

   ```
   sudo systemctl restart smbd nmbd
   ```

Unlike the shared directory, above, this share definition does not force a set of permissions for new files and directories. Instead it forces all file system access to use the Linux user and group specified by `force user` and `force group`.

The Linux user must exist and is cases sensitive. An equivalent samba user is **not** required.

Permissions will be the default for that Linux user.

This approach will ensure all files in the user’s home directory that are created or edited by clients are owned by the correct server side Linux user.

If desired, `force create mode` and `force directory mode` may also be used.

5.4 The smb.conf From 5.1 Explained

`[global]`	Section header. Contains options that affect the server and all shares.
`log file = /var/log/samba/log.%m`	Log file location. `%m` is replaced with the name of the client.
`server role = standalone`	Sets server role. Standalone server, not part of a Windows Domain
`map to guest = bad user`	Attempts to login with an invalid username[11] and password are treated as guest accesses.

`[shared]`	Start of share definition. “shared” is the share name.
`path = /srv/shared`	Server side path to shared directory. Must be an absolute path.
`read only = no`	Writes to the share are permitted[12].
`guest ok = yes`	Guest access to the share is permitted.
`force create mode = 666`	Force permissions for files created by clients. 666 grants read and write permissions to all users and groups.
`force directory mode = 777`	Force permissions for directories created by clients. 777 grants read, write and execute permissions to all users and groups[13].

All other parameters will have their default values.

[11] Samba user not Linux user. Samba has its own user and password database.

[12] Subject to underlying file system permissions.

[13] Without execute permission directories cannot be entered

6 A More Secure Samba Server

6.1 Configure Samba

11. Stop the samba server:

    ```
    sudo systemctl stop smbd nmbd
    ```

12. If it does not exist, create a directory to share:

    ```
    sudo mkdir -p /srv/shared
    ```

13. Open permissions on it:

    ```
    sudo chmod 777 /srv/shared
    ```

14. Backup your current smb.conf:

    ```
    sudo cp /etc/samba/smb.conf /etc/samba/smb.bak
    ```

15. Open /etc/samba/smb.conf in your preferred text editor. You will need to be root or use sudo.

16. Replace its contents with the following:

    ```
    [global]
      log file = /var/log/samba/log.%m
      server role = standalone
      unix password sync = no

    [shared]
      path = /srv/shared
      read only = no
    ```

    See section 6.5 for a brief explanation of the above smb.conf.

17. Save and close.

18. Verify smb.conf:

    ```
    testparm
    ```

19. Fix any errors and reverify.

20. Start the samba server:

    ```
    sudo systemctl start smbd nmbd
    ```

6.2 Add Samba Users

Samba users must also be Linux users[14] but it is not required that they have a home directory or the ability to login to Linux.

6.2.1 Add Linux User

All steps must be performed as root or with sudo. This step can be skipped if the Linux user already exists.

For a standard user with a home directory and the ability to login:

```
sudo adduser username
```

e.g.

```
sudo adduser sam
```

and follow prompts.

For a user without a home directory but with the ability to login:

```
sudo adduser -M username
```

e.g.

```
sudo adduser -M sybil
```

and follow prompts.

To disable login on a user:

```
sudo usermod -L username
```

e.g.

```
sudo usermod -L nobby
```

Do not disable users that need to login whether locally, via ssh, via SFTP, or via SSHFS.

[14] This is a requirement of the default password backend.

6.2.2 Add The Samba User

User creation and management is done via `smbpasswd`. Most options require root or sudo though a user can change their own password.

The samba server must be running in order to add or modify a user.

If unix password sync = yes is present in smb.conf the Linux user’s password will be updated to match the new samba password.

To create a user:

```
sudo smbpasswd username
```

e.g.

```
sudo smbpasswd sam
```

and follow prompts.

6.2.3 Users For Windows Clients

While not necessary, it can somewhat ease access from Windows clients if the created samba user has the same username and password as the user on the windows machine.

6.3 Client Access

6.3.1 Linux Command Line

Remote shares can be made available via the `mount` command this normally requires root privileges. For example:

```
sudo mount -t cifs -o user=sam,rw,file_mode=666,dir_mode=777
//l-space/shared /mnt
```

Enter the samba user’s password when prompted.

Remote shares may also be added to /etc/fstab. For example[15]:

```
//l-space/shared /mnt cifs
defaults,nofail,_netdev,user=sam,password=bar,rw,file_mode=666,dir_mode=777 0 0
```

Permissions set in the above mount options are insecure as they will allow full access by all users on the client though files and directories will appear to be owned by root.

The meaning of the mount options are:

`user=sam`	Specify which remote samba user to connect as.
`password=bar`	Password for the above user
`rw`	Mount read/write.
`file_mode=666`	Override default file permissions.
`dir_mode=777`	Override default directory permissions.
`defaults`	Use default mount options
`nofail`	Allow boot to continue if the mount fails. Only useful; in fstab with the auto (and defaults) mount options.
`_netdev`	Flag to the system that the mount requires the network to be up.

For further mount options refer to the documentation for the `mount` and `mount.cifs` commands.

For further information on /etc/fstab please refer to its documentation or to my guide at https://github.com/thagrol/Guides.

fstab is readable by all users on a machine so the samba user’s password is not secure. Use of a credentials file is preferred. See `man mount.cifs`.

[15] An fstab entry must be on a single line. That may not be apparent here due to document formatting.

6.3.2 Linux Desktop

Mounts made via the command line or /etc/fstab will be available via the chosen mount point.

Most file managers can access shares by entering `smb://server/share` e.g. `smb://l-space/shared` in the address field.

Network browsing may be functional but only if both the server and client have the SMB v1 protocol enabled.

Please refer to the documentation for your chosen file manager for more details.

6.3.3 Microsoft Windows

Shares can be accessed by entering `\\server\\share` e.g. `\\l-space\shared` into the address filed of File Explorer or This PC. Shares may be assigned a drive letter via “Map network drive”.

Network browsing may be functional but only if both the server and client have the SMB v1 protocol enabled or the server has wsdd installed (see 14.2.2). SMB v1 is disabled by default on Windows 10.

Please refer to the documentation for your version of Windows and to section 2.4.

6.4 Sharing Users Home Directories

It may be desirable to allow access to a user’s home directory. Samba has built in support for this via the `[homes]` share.

1. Backup your smb.conf. You may wish to rename any existing backup first so it is not lost.
2. Open `/etc/samba/smb.conf` in your preferred text editor. You will need to be root or use sudo.
3. Add the following to the end of the file.

   ```
   [homes]
     comment = Home Directories
     browseable = no
     read only = no
     valid users = %S
   ```

4. Save and close
5. Verify your new smb.conf:

   ```
   testparm
   ```

6. Fix any errors and reverify.
7. Restart the samba server:

   ```
   sudo systemctl restart smbd nmbd
   ```

No `path =` is required and a single share definition will function for all users that have a home directory.

`valid users = %S` ensures that only the user whose home directory it is can access it.

6.5 The smb.conf From 6.1 Explained

`[global]`	Section header. Contains options that affect the server and all shares.
`log file = /var/log/samba/log.%m`	Log file location. `%m` is replaced with the name of the client.
`server role = standalone`	Sets server role. Standalone server, not part of a Windows Domain
`unix password sync = no`	If the samba user’s password changes do not update the Linux user’s password to match.
`[shared]`	Start of share definition. “shared” is the share name.
`path = /srv/shared`	Server side path to shared directory. Must be an absolute path.
`read only = no`	Writes to the share are permitted[16].

All other parameters will have their default values.

[16] Subject to underlying file system permissions.

7 Better Support For Linux Clients – NFS

While the SMB/CIFS protocol has some support for Linux/Unix file ownership, permissions, and other file system features use of them is not recommended as they require the obsolete and very insecure v1. Most modern OS, servers, and clients will not use v1 unless explicitly instructed to do so.

NFS[17] is Linux native with full support for Linux file system features.

Using both Samba and NFS on the same server directory is usually safe.

7.1 NFS And Users

Unlike with Samba/SMB/CIFS an NFS server does not require a separate user database nor does it require that any given user exist on both the client and server.

It does, however require that numeric user and group IDs are the same across all machines and that clients that do not have a particular user or group do not use the ID for a different user or group.

If user/group IDs are not consistent, file owner, group, and permissions will not be consistent across clients.

To illustrate:

Client device: Ankh

User: sam	ID: 1001	Group: sam	ID: 1001
User: sybil	ID 1002	Group: sybil	ID: 1002

Client device: Morpork

User: sam	ID: 1002	Group: sam	ID: 1002
User: sybil	ID: 1001	Group: sybil	ID: 1001

While logged in to Ankh, sam creates a file that appears in `ls -l` as:

```
-rw-r--r-- 1 sam sam 1024 Jan 22  2021 laws.txt
```

Later sam views the same file whist logged in to Morpork:

```
-rw-r--r-- 1 sybil sybil 1024 Jan 22  2021 laws.txt
```

Nothing has change on the file[18] but as the IDs are not consistent the owner and group appear to have changed giving sybil permissions that only sam should have.

[17] **N**etwork **F**ile **S**ystem

[18] The file system uses the numeric user and group IDs not the textual ones.

7.2 Quick, Dirty, And Somewhat Insecure

Quick because it’s a minimal configuration. Dirty because it’s a minimal configuration. Somewhat insecure because permissions on the exported directory are changed to allow full access by all users and the export definition allows access by all devices with access to your network. Normal Linux user, group, and permissions still apply.

That said, this may be good enough in some circumstances – where security is not required and all remote users and devices are trusted.

1. If not already done, install the NFS server per section 4.
2. If it does not exist, create a directory to export per section 5.1 steps 2 and 3.
3. Backup your current exports file:

   ```
   sudo cp /etc/exports /etc/exports.bak
   ```

4. Open /etc/exports in your preferred text editor. You will need to be root or use sudo.
5. Add the following on a new line at the end of the file:

   ```
   /srv/shared *(rw,sync,no_subtree_check)
   ```

6. Save and close.
7. Re-export all directories:

   ```
   sudo exportfs -r
   ```

7.3 Client Access

7.3.1 Linux Command Line

Remote exports can be made available via the `mount` command this normally requires root privileges. For example:

```
sudo mount -t nfs l-space:/srv/shared /mnt
```

`l-space` is the name or IP address of the NFS server. `/srv/shared` is the same path as entered into /etc/exports on the server.

Remote exports may also be added to /etc/fstab. For example[19]:

```
//l-space/srv/shared /mnt nfs defaults,nofail,_netdev,rw, 0 0
```

The meaning of the mount options are:

`defaults` Use default mount options

`nofail` Allow boot to continue if the mount fails. Only useful; in fstab with the auto (and defaults) mount options.

`_netdev` Flag to the system that the mount requires the network to be up.

For further mount options refer to the documentation for the `mount` and `mount.nfs` commands.

For further information on /etc/fstab please refer to its documentation or to my guide at https://github.com/thagrol/Guides.

7.3.2 Linux Desktop

Mounts made via the command line or /etc/fstab will be available via the chosen mount point.

Most file managers can access shares by entering nfs`://server/directory` e.g. nfs`://l-space/srv/shared` in the address field.

Please refer to the documentation for your chosen file manager for more details.

7.3.3 Microsoft Windows

Most versions of Windows have no native support for NFS[20]. For those that do, please refer to the documentation for your version of Windows.

[19] An fstab entry must be on a single line. That may not be apparent here due to document formatting.

[20] Third party drivers may exist however they are outside the scope of this guide.

7.4 Adding a User’s Home Directory

It may be desirable to allow access to a user’s home directory. As in 7.2, above, access is not permitted by any machine that can reach the server.

1. Backup your current exports file. You may wish to rename any existing backup first so it is not lost.

   ```
   sudo cp /etc/exports /etc/exports.bak
   ```

2. Open /etc/exports in your preferred text editor. You will need to be root or use sudo.

3. Add the following on a new line at the end of the file:

   ```
   /home/pi *(rw,sync,no_subtree_check)
   ```

4. Save and close.

5. Re-export all directories:

   ```
   sudo exportfs -r
   ```

To export the home directories of all users simply export `/home`.

7.5 The Export File And Options Explained

Each line in /etc/exports defines an exported directory. The format is:

```
/path/to/directory     host(export options)
```

For example:

```
/srv/shared *(rw,sync,no_subtree_check)
```

A full path must be used.

`*` Allow access by all clients regardless of hostname or IP address. Safe as long as your server is not directly exposed to the internet.

`rw` Allow writing by clients. Clients must still mount the export as rw.

`sync` Reply to client write requests only after changes have been written to disc. May impact performance but prevents data loss or corruption if the server crashes.

`no_subtree_check` Improves performance and reliability but has a minor security implication. See `man exports`.

8 Supporting Smart Devices As Clients

Most smart devices do not include support for SMB/CIFS and NFS instead preferring to use DLNA[21][22]. Allowing DLNA clients requires installing and configuring additional software such as miniDLNA.

1. Install miniDLNA:

   ```
   sudo apt update

   sudo apt install -y minidlna
   ```

2. Backup your existing minidlna.conf:

   ```
   sudo cp /etc/minidlna.conf /etc/minidlna.bak
   ```

3. Open /etc/exports in your preferred text editor. You will need to be root or use sudo.
4. Find the line starting `media_dir=`
5. Replace the path after “=” to the actual path to the directory you wish to share e.g.

   ```
   media_dir=/srv/shared
   ```

6. Add any additional directories, one `media_dir=` definition per line.
7. Save and close.
8. Restart minidlna

   ```
   sudo systemctl restart minidlna
   ```

Depending on the number of files, it may take miniDLNA some time to scan your media directories. Clients will not have access until this has completed.

[21] **D**igital **L**iving **N**etwork **A**lliance

[22] Also know as UPnP.

9 Adding More Storage

Sooner or later you’ll run out of free space and need to add additional storage.

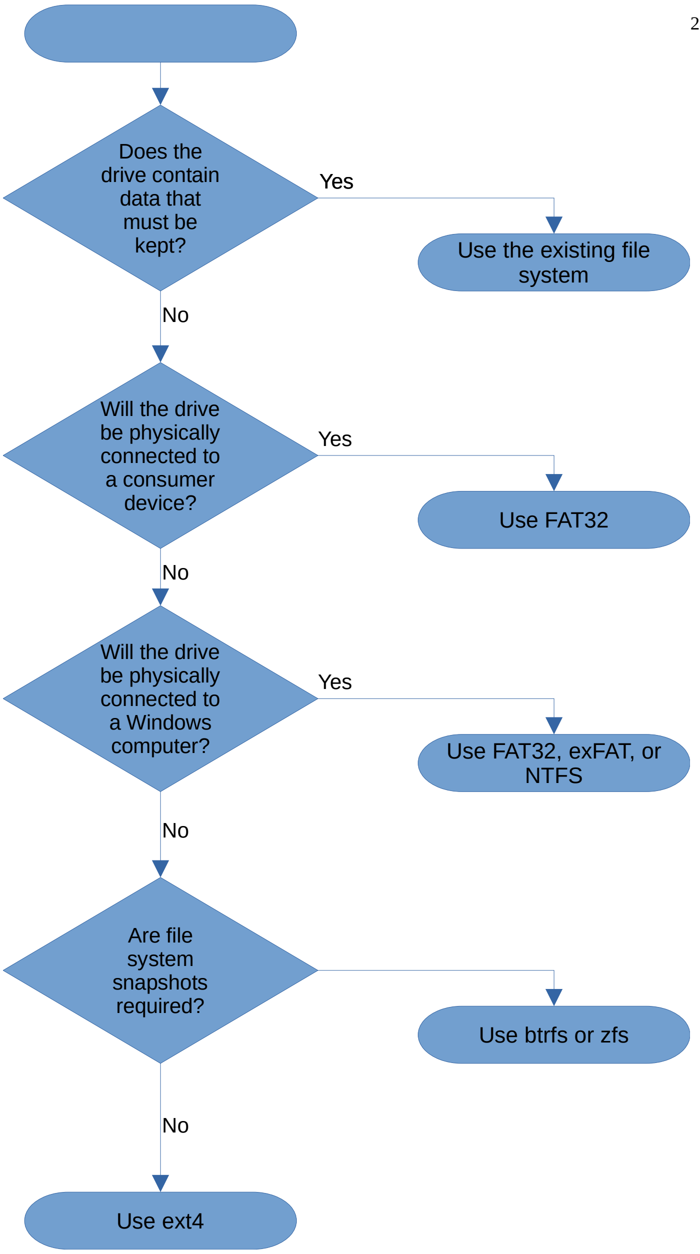
9.1 Selecting Drives

Some things to bear in mind when selecting drives:

- 3.5” desktop drives require both 5v and 12v so will need a powered enclosure or adapter with its own power supply.
- Most Pi can directly power one 2.5” drive. Using more than one requires powered enclosures, adapters, or a self powered USB hub.
- In terms of storage per £, traditional spinning rust is currently still the best value.
- Not all M.2 drives are NVMe. Some are SATA
- Not all M.2 adapters are for NVMe, some are SATA only.
- Not all M.2 adapters are for SATA, some are NVMe only.
- NVMe SSDs provide no performance benefit over SATA SSDs. Speed is governed by the slowest of network connection speed, USB Speed, and the single PCIe lane on some models. Both SATA and NVMe SSDs can saturate this as can sustained transfers from a decent HDD.
- M.2 SATA drives provide no performance benefit over 2.5” SATA drives.
- Avoid shingled (SMR) drives. Due to the additional housekeeping required write performance is reduced and these drive are not suited for RAID arrays.
- When selecting drives for a RAID array drives of similar capacity from different manufacturers or different batches are preferred as they are less likely to fail at the same time.

9.2 Selecting A File System

In almost all cases the file system in use on the server is both unknown and irrelevant to clients.[23] Given this it is usually best to use a Linux native file system such as ext4. If in doubt refer to the following chart.



Does the drive contain data that must be kept?
Yes
Use the existing file system
No
Will the drive be physically connected to a consumer device?
Yes
Use FAT32
No
Will the drive be physically connected to a Windows computer?
Yes
Use FAT32, exFAT, or NTFS
No
Are file system snapshots required?
Use btrfs or zfs
No
Use ext4

[24][25]

[23] The exception being iSCSI and similar block level protocols. Those are outside the scope of this guide.
[24] Snapshot management on btrfs, zfs, etc is outside the scope of this guide.
[25] Snapshots do not protect against hardware failure.

9.3 Preparing The Storage

Bare drives are not usually supplied pre formatted so will need to be initialised and partitioned first.

All of the following procedures in this section are destructive and cannot be easily undone. Be certain you are operating on the correct device and partition. I cannot and will not accept responsibility for any damage or data loss that may occur.

9.3.1 Desktop Users – Bare Drive

1. Connect the drive.
2. Open gparted (Menu then System Tools or `sudo gparted` in a terminal)
3. Enter your password if prompted.
4. From the drop down select the correct device. With a single drive connected it should be /dev/sda.
5. Open the “Device” menu.
6. Select “Create partition table”
7. Select partition type. Drives of 2TB or less can use either MS-DOS or GPT. Drives over 2TB must use GPT.
8. Double check that you have selected the correct drive. Writing a new partition table to a device cannot be easily undone.
9. Click “OK”
10. Open the “Partition” menu.
11. Select “New”
12. Accept the default options to use the full size of the drive in a single ext4 partition.
13. Click the green tick icon.
14. Wait for the format to complete.
15. Close gparted.

9.3.2 Desktop Users – Pre Formatted Drive

1. Connect the drive
2. If the drive has been automatically mounted, first unmount it via the file manager.
3. Open gparted (“Menu” then “System Tools” or `sudo gparted` in a terminal)
4. Enter your password if prompted.
5. From the drop down select the correct device. With a single drive connected it should be /dev/sda.
6. In the diagram or list find the correct partition. Most pre formatted drives only have one.
7. Right mouse click it.
8. In the menu open the “Format to” sub menu then select “ext4”.
9. Double check you have selected the correct drive and partition. Formatting the wrong one cannot easily be undone.
10. Click the green tick icon.
11. Wait for the format to complete.
12. Close gparted.

9.3.3 Command Line Users – Bare Drive

1. Connect the drive.
2. Find its device node[26]:

   ```
   lsblk
   ```

   A single drive will likely be `/dev/sda`

3. Start fdisk

   ```
   sudo fdisk /dev/sda
   ```

4. Create a new partition table:

   ```
   o
   ```

   or

   ```
   g
   ```

   For drives 2TB or less use `o` to create an MS-DOS partition table or `g` for a GPT partition table. Drives over 2TB must use `g`.

5. Create a new partition:

   ```
   n
   ```

6. Accept the default options to all prompts.
7. View the changes:

   ```
   p
   ```

8. If everything looks correct and you are certain that you are making changes to the correct drive, write the changes to disc:

   ```
   w
   ```

You should now have a drive containing a single, unformatted partition. Proceed to 9.3.4

[26] The item in /dev that refers to the physical disc.

9.3.4 Command Line Users – Pre Formatted Drive

1. Connect the drive.
2. Find the device node for the desired partition.

   ```
   lsblk
   ```

3. For a single drive with a single partition thus will likely be `/dev/sda1`.
4. Format the partition. Note: be certain you are working with the correct partition as this cannot easily be undone.

   ```
   sudo mkfs.ext4 /dev/sda1
   ```

9.4 Mounting The Drive[27]

9.4.1 Using the Desktop’s Automatic Mounting

Relying on the desktop to mount your drives/partitions, while possible, is not recommended for the following reasons:

- Mounting occurs only if the desktop is running and a user has logged in to it.
- Mounting occurs after the server software (NFS, Samba, miniDLNA) has started
- The mount point used is not guaranteed to be consistent across reboots.
- The is no control over which mount options are used.
- The is no control over which mount point is used.
- With the default permissions the mount point may not be accessible by other users.

The last point can be addressed by running `sudo chmod 766 /media/username` e.g. `sudo chmod 766 /media/pi`

No OS level configuration is required other than ensuring the OS boots to the desktop with automatic login enabled.

[27] In all but a few case partitions get mounted not entire drives.

9.4.2 Using /etc/fstab

fstab provides much finer control over where a drive/partition is mounted and with what options. However getting it wrong can often result in a system that will not boot correctly. To aid recovery, have a second SD card with a clean OS install to hand before making changes.

A full, in depth discussion of fstab is outside the scope of this guide. The following are the steps required to mount an ext4 partition.

1. Connect the drive to your Pi
2. Find the PARTUUID and FSTYPE of the desired partition:

   ```
   lsblk -o NAME,SIZE,TYPE,FSTYPE,MOUNTPOINT,PARTUUID
   ```

   Output will be similar to this:

   ```
   NAME          SIZE TYPE FSTYPE MOUNTPOINT PARTUUID
   sda         960.5M disk
   └─sda1      959.5M part                   83bd0d82-01
   mmcblk0      14.9G disk
   ├─mmcblk0p1   256M part vfat   /boot      738a4d67-01
   └─mmcblk0p2  14.6G part ext4   /          738a4d67-02
   ```

   If you’re unsure which item is the correct one, disconnect the drive, repeat the above command and compare the output.

3. Create a mount point[28] for the partition. For example:

   ```
   sudo mkdir -p /srv/external
   ```

4. Backup your existing fstab:

   ```
   sudo cp /etc/fstab /etc/fstab.bak
   ```

5. Open /etc/fstab in your preferred text editor. You will need to be root or use sudo.
6. On a new line at the end of the file add:

   ```
   PARTUUID=83bd0d82-01 /srv/external ext4 defaults,rw,nofail 0 0
   ```

   Replace “83bd0d82-01”, “/srv/external”, and “ext4” as required.

7. Save and close

[28] A mount point is simply a directory somewhere in your file system. Preferably empty but it does not have to be. If it is not empty, its contents will be masked once something has been mounted n it.

8. Test the fstab entry:

   ```
   sudo mount -a
   mountpoint /srv/external
   ```

   The `mount` command should return nothing and the `mountpoint` command should indicate that /srv/external is a mount point. If not fix the error and retry.

Note: A malformed fstab will prevent your system from booting. Without the `nofail` option a failed mount will do the same. Power off your Pi, swap to your second SD card, boot from that and fix the broken fstab by mounting the first SD card in a USB card reader.

9.4.3 Other Methods

Other methods of mounting a partition include a systemd mount unit, a manual mount command, a mount command in /etc/rc.local, a mount command called from cron, etc. None of these are recommended.

Systemd mount units are automatically created from fstab during boot.

A manual mount command will result in an empty directory being shared/exported until a user logs in and performs the mount. This may also result in files being saved by clients onto the SD card instead of the external storage device.

A mount command in rc.local is non standard and should it fail will prevent execution of anything after it in rc.local.

A mount command called from cron is non standard.

While the non standard forms will allow the system to boot should a mount fail, using them will cause problems with future maintenance simply because they’re not where they’re expected to be.

10 Further Improving Security

10.1 Restricting Access To A Resource[29] To A Subset Of Users

NFS exports do not have explicit configuration options to deny or allow certain users as it has full support for Linux owner, group, and permissions.

10.1.1Samba Guest Access

By default guest access is disabled. To enable it add the following to the [global] section of your smb.conf:

```
map to guest = Bad User
```

To disable either remove the above or change it to

```
map to guest = Never
```

Each share definition then needs to be configured to permit guest access. The default is to deny it. Add the following to the share definition to enable guest access:

```
guest ok = yes
```

Remove it or change it to the following to deny guest access:

```
guest ok = no
```

Refer to sections 5 and 6 for smb.conf examples.

In the event that a share should only be accessible by guest logins, include the following in the share definition:

```
guest ok = yes
guest only = yes
```

Users who login with a valid username and password will be denied access to the share.

[29] A Samba share or NFS export.

10.1.2Samba Invalid and Valid Users

The share specific parameters `invalid users` and `valid users` control which Samba users have access to the share.[30]

Each is a space separated list of Samba user names which default to empty lists. If a user name appears it both lists, access to the share will be denied. An example (incomplete) smb.conf:

```
[Share1]
path = /srv/share1

[Share2]
path = /srv/share2
invalid users = nobby colon

[Share3]
path = /srv/share3
valid users = sam sybil

[Share4]
path = /srv/share4
invalid users = sybil
valid users = sam sybil
```

Share1 can be accessed by any non guest user.

Share2 can be accessed by any non guest user except for nobby and colon.

Share3 can be accessed only by sam and sybil.

Share4 can only be accessed by sam.

In the general case, if `valid users` is given but `invalid users` is not, only users in the list will have access; if `invalid users` is given but `valid users` is not, all users except those listed will have access; if both are given and a user appears in both lists that user will be denied access; if neither are given, all users will have access.

10.1.3DLNA

DLNA has no concept of users so restriction by user name is not possible.

[30] If you want to deny access to a user to your entire server, delete or disable their account.

10.2 Restricting Access To A Resource[31] To A Subset Of Machines

10.2.1Samba

Samba provides two parameters `hosts allow` and `host deny` to control access by machines. By default both lists are empty allowing access from all hosts.

Each is a comma, space, or tab separated list of hosts.

Hosts may be specified by hostname, IP address, an IP address prefix, or a subnet and netmask. IF a host appears in both lists, access will be granted.

The `EXCEPT` keyword may be used to exclude a host from a range. The ALL[32] keyword can be used with `hosts deny` to deny access from all hosts except those listed in `hosts allow`.

The loopback address (127.0.0.1) will always be allowed access unless explicitly denied.

An example (incomplete) smb.conf:

```
[Share1]
path = /srv/share1

[Share2]
path = /srv/share2
hosts deny = ankh, morpork

[Share3]
path = /srv/share3
hosts allow = 10.0.0. EXCEPT 10.0.0.1

[Share4]
path = /srv/share4
hosts deny = ALL
hosts allow = ankh, morpork
```

Share1 can be accessed from all hosts

Share2 can be accessed from any host except ankh and morpork.

Share3 can be accessed from all hosts with an IP address starting 10.0.0. but not from 10.0.0.1

Share4 can only be accessed from ankh and morpork.

[31] A Samba share or NFS export.

[32] 0.0.0.0/0 may also be used.

10.2.2NFS

The host portion of an NFS export controls which machines can access it and can be specified in one of the following ways:

- hostname e.g. ankh or morpork.disc
- Wildcards:
  - * matches zero or more characters
  - ? matches exactly one characters
  - [square brackets] indicate lists of characters to match against e.g. [abc]
- IP address e.g. 10.0.0.1
- IP networks e.g. 10.0.0.0/255.255.255.0 or 192.168.0.0/24

An example exports file:

```
/srv/1      ankh(rw)
/srv/2      *(rw)
/srv/3      *.disc(rw)
/srv/4      clacks[123](rw)
/srv/5      10.0.0.1(rw)
/srv/6      192.168.0.0/24(rw)
```

/srv/1 can only be accessed from ankh.

/srv/2 can be accessed from any machine.

/srv/3 can only be accessed from any machine with a hostname in the .disc domain including any subdomains it may have.

/srv/4 can only be accessed from clacks1, clacks2, and clacks3.

/srv/5 can only be accessed from a machine with an IP address of 10.0.0.1.

/srv/6 can be accessed from any machine with an IP address in the 192.168.0.0/24 subnet.

10.2.3DLNA

DLNA does not support restricting access to specific hosts.

11 User Management

11.1 Samba

Samba requires both a Samba specific user and a corresponding Linux user. The Linux user is created and managed via the normal Linux tools (adduser and usermod), the Samba user is created and managed by smbpasswd.

Normal (i.e. non root) user are only able to change their password. All other options require root or sudo.

If the [global] parameter `unix password sync = yes` is in force the matching Linux user’s password will be updated to match the new Samba password. No other changes are propagated.

To add a user:

```
sudo smbpasswd -a username
```

for example

```
sudo passwd -a sam
```

To delete a user:

```
sudo smbpasswd -x username
```

for example

```
sudo smbpasswd -x nobby
```

To disable but not delete a user:

```
sudo smbpasswd -d username
```

for example

```
sudo smbpasswd -d sam
```

To change a user’s password:

```
sudo smbpasswd username
```

for example

```
sudo smbpasswd colon
```

To change your own password:

```
smbpasswd
```

11.2 NFS And sftp/sshfs

NFS user management is done with the normal Linux user management tools with one caveat: the numeric user and group IDs for a user/group must be the same a cross all clients and the server or if the user/group does not exist on a machine that the numeric ID must be unused.

Linux users for Samba, sftp, and sshfs are managed with the same tools without the requirement on numeric IDs.

See also 7.1

Adding or changing a user requires root or sudo. See 6.2.1 and the man pages for `adduser` and `usermod`.

Users can change their own passwords with `passwd`. This will not update the password for the corresponding Samba user should one exist.

Root (or users with appropriate sudo rights) can update the password of any user via

```
passwd username
```

for example:

```
passwd sybil
```

or

```
sudo passwd sybil
```

11.3 DLNA

DLNA has no concept of users so user management is neither needed nor possible.

11.4 Recovering A User’s Password

Passwords are one way encrypted and cannot be decrypted. If a user has lost their password, all that can be done is to create a new one for them. Usual practise is to ask them to change it immediately so that only they know what it is.

12 Advanced Usage

12.1 Access From The Internet

Use a VPN. Directly exposing Samba, NFS and/or DLNA is not safe. Detailed instructions on setting up a VPN are outside the scope of this guide however pivpn[33], openvpn[34], and wireguard[35] may be of interest.

sftp and sshfs may be used over the internet but you should be aware of and take steps to mitigate the security issues, particularly those relating to the well know default user name and password on Raspberry Pi OS.

Port forward port 22[36] on your router to port 22 on your NAS.

12.2 Download Only Access From Web Browsers

This is quick and easy but insecure. Data transfers are not encrypted, no login is required, and all file access on the server is done as the user starting the process. As such starting it as root is not recommended as, while files cannot be changed, files that should not be accessible to normal users may be accessed.

The following command will start python’s http.server module which will allow access to the current directory and its sub directories:

```
python3 -m http.server
```

To use a specific directory:

```
cd /path/to/directory ; python3 -m http.server
```

e.g.

```
cd /srv/shared ; python3 -m http.server
```

This can be started at boot using any of the normal methods.

Clients can connect via the URL `http://server:8000/` e.g. `http://l-space:8000/`

[33] https://pivpn.io/

[34] https://en.wikipedia.org/wiki/OpenVPN

[35] https://en.wikipedia.org/wiki/WireGuard

[36] Some routers allow different ports on the internal and external sides. Using a non standard port on the outside of your router is not a substitute for proper security measures.

12.3 A Resource With Read Only Access To All And Write Access to Some Users

NFS handles this natively using the underlying file system properties. For Samba define a share as read only then add the write list parameter with the names of the Samba user who should be able to write tot he share. For example:

```
[shared]
   path = /srv/shared
   read only = yes
   write list = sam, sybil
```

Users sam and sybil will be able to write to the share. All other users will not. This includes guest logins should those be enabled.

12.4 Samba – Enable Printing

This does not configure the printing subsystem of the OS on your NAS[37]. It simply enables the relevant Samba shares and features.

1. Add the following share definitions to your smb.conf:

   ```
   [printers]
      comment = All Printers
      browseable = no
      path = /var/spool/samba
      printable = yes
      guest ok = no
      read only = yes
      create mask = 0700

   [print$]
      comment = Printer Drivers
      path = /var/lib/samba/printers
      browseable = yes
      read only = yes
      guest ok = no
   ```

2. Restart Samba:

   ```
   sudo systemctl restart smbd nmbd
   ```

[37] That’s outside the scope of this guide.

12.5 Samba – Fully Disable Printing

To fully disable printing support in Samba:

1. Removes the share definitions listed in 12.4
2. Add the following to the [global] section of your smb.conf:

   ```
    load printers = no
   printing = bsd
   printcap name = /dev/null
   disable spoolss = yes
   ```

3. Restart Samba:

   ```
   sudo systemctl restart smbd nmbd
   ```

12.6 Samba – Additional smb.conf Parameters

For a comprehensive list see https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html

12.6.1[global] Parameters

`log file =` Log file location.

`server role =` Sets server role. One of `auto`, `standalone`, `member server`, `classic primary domain controller`, `classic backup domain controller`, or `active directory domain controller`.
Default: `auto`.

`map to guest =` When to map a failed login to a guest login. One of `never`, `bad user`, `bad password`, or `bad uid`.
Default: `never`

`guest account =` Linux user to use for all guest access.
Default: `nobody`

`server string =` Text shown to clients when browsing the network. `%v` is replaced by the samba version number, `%h` is replaced by the server’s hostname.
Default: `Samba %v`

`workgroup =` Workgroup the server will belong to.
Default: `WORKGROUP`

12.6.2[share] Parameters

`path =`
Server side path to shared directory. Must be an absolute path.
No default and must be specified.

`comment =`
Text sent to client when browsing the network.
No default.

`read only =`
Whether writes to the share are denied[38]. `yes` or `no`
Default: `yes`

`guest ok =`
Whether guest access to the share is permitted. `yes` or `no`.
Default: `no`

`guest only =`
Whether a share is restricted to guest logins only. `yes` or `no`. No effect if guest access is not permitted.
Default: `no`

`force create mode =`
Force permissions for files created by clients. Standard Linux octal bit mask.
Default: `0000`

`force directory mode =`
Force permissions for directories created by clients. Standard Linux octal bit mask.
Default: `0000`

`create mode =`
Permissions for files created by clients. Standard Linux octal bit mask. Permissions not set here will not be present on the file unless set in `force create mode`.
Default: `0744`

`directory mode =`
Permissions for directories created by clients. Standard Linux octal bit mask. Permissions not set here will not be present on the directory unless set in `force directory mode`.
Default: `0755`

`hosts deny =`
`hosts allow =`
`See 10.2.1`

`invalid users =`
`valid users =`
See 10.1.2

`write list =`
See 12.3

hide files =
List of files and/or directories that are not visible but are still accessible. Clients see the DOS “hidden” attribute is set.
“/” is used as a separator so cannot be used as part of a file or directory name. “*” and “?” wildcards are permitted. Names are case sensitive.
No default. May impact performance.
Example: `hide files = /.*/foo/ba?/`

[38] Subject to underlying file system permissions.

12.7 Samba – Unix Extensions

When both client and server are using SMB/CIFS protocol v1, Samba can implement the CIFS UNIX extensions. However, v1 is old, insecure, and deprecated and should not be used.

They are also only useful with Linux and UNIX clients.

If you must use them:

1. Add the following to the [global] section of your smb.conf:

   ```
   unix extensions = yes
   server min protocol = NT1
   ```

2. Restart samba.
3. Remount the share on the client using the `vers=1.0` mount option.

12.8 NFS – Different Export Options For Different Clients

Each exported directory in /etc/exports can have multiple sets of options for different hosts. Each host(options) group must be separated from the next by a space and be on the same line.

For example:

```
/srv/shared *(ro) ankh(rw)
```

All clients have read only access except ankh which has read/write access.

12.9 NFS – Export Options

For a comprehensive list see `man exports`.

`ro`	Default. Only allow reading.
`rw`	Allow both reading and writing
`sync`	Default. Only reply to clients after changes have been written to disc.
`async`	Reply to client write requests only after changes have been written to disc. May impact performance but prevents data loss or corruption if the server crashes.
`no_subtree_check`	Default. Improves performance and reliability but has a minor security implication. See `man exports`.
`subtree_check`	Opposite of `no_subtree_check`. “tends to cause more problems than it is worth.”[39]

`mountpoint=path`
`mp=path`
	Only export the directory if `path` is a mount point.
	If `path` is not given, the path of the export must be a mount point.

`root_squash`	Map all access as root (UID 0, GID 0) to the anonymous user instead (usually nobody and nogroup).
`no_root_squash`	Turn off `root_squash`. Normally only really needed for network booted clients.
`all_squash`	Map all UIDs/GIDs to the anonymous ones.
`anonuid`	UID to use when squashing
`anongid`	GID to use when squashing

[39] Source: `man exports`

12.10 Backup And Restore Server Configuration

This does not backup your data and mount points just the server and user configuration. Restart all services or reboot the server following a restore.

On anything other than a clean OS a manual merge of the existing and backed up files is a safer option than simply replacing them.

12.10.1 Linux User Database

Backup the following files:

- `/etc/passwd`
- `/etc/shadow`

To restore, replace theses with the backed up version and ensure they have the correct owner, group. and permissions:

File	**Permissions**	**Owner**	**Group**
`/etc/passwd`	644	root	root
`/etc/shadow`	620	root	shadow

12.10.2 Samba Configuration

If your Samba users have matching Linux users you must also backup and restore the Linux user database (see above).

Backup the following files:

- `/etc/samba/smb.conf`
- `/var/lib/samba/private/passdb.tdb`
- Any additional files you have used with `include =` in your smb.conf

To restore, replace theses with the backed up version and ensure they have the correct owner, group. and permissions:

File	**Permissions**	**Owner**	**Group**
`/etc/samba/smb.conf`	644	root	root
`/var/lib/samba/private/passdb.tdb`	600	root	root
Additional files	As before	As before	As before

12.10.3 NFS

Optionally backup the Linux user database (see 12.10.1).

Backup /etc/exports

To restore replace the file with the backup and ensure it has the correct owner, group, and permissions:

File	Permissions	Owner	Group
`/etc/exports`	644	root	root

12.10.4 sftp And sshfs

No backup required other than 12.10.1

12.10.5 miniDLNA

Backup /etc/minidlna.conf

To restore replace the file with the backup and ensure it has the correct owner, group, and permissions:

File	Permissions	Owner	Group
`/etc/minidlna.conf`	644	root	root

12.10.6 Miscellaneous

It can be useful to backup the following files particularly where they have been customised and/or when using storage beyond the default two partitions:

- `/boot/config.txt`
- `/boot/cmdline.txt`
- `/etc/fstab`

To restore replace the file with the backup and ensure it has the correct owner, group, and permissions:

File	Permissions	Owner	Group
`/boot/config.txt`	Any	Any	Any
/boot/cmdline.txt	Any	Any	Any
/etc/fstab	644	root	root

13 RAID

This is not a guide to RAID on Linux. Enough information has been provided for you to configure a RAID array but there is not, and will never be, any coverage of maintaining, administering, or repairing RAID arrays.

Ensure you are aware of and comfortable with the tools and steps required to repair and rebuild a broken RAID array before proceeding.

While every effort has been made to ensure the validity of this section including testing commands and configuration I do not use software RAID and therefore cannot guarantee its fitness for any particular purpose or that theses instructions will work in all circumstances.

13.1 What Is RAID?

RAID stands for **R**edundant **A**rray of **I**nexpensive **D**iscs. Its purpose is to provide reliability and availability in the event of a disc failure. It is not a substitute for proper backups.

If you do not require high availability, RAID levels 1 and above may not be worth the added complexity and potential impact to performance.

13.2 Software Or Hardware

In general hardware RAID is preferable but likely more expensive.

Software	**Hardware**
Runs inside the OS kernel.	Runs inside the controller or enclosure
Additional CPU load.	Little to no additional CPU load.
Additional data transfers over the USB bus are required.	Little to no additional data transfers required.
Requires additional software and OS configuration	No additional OS configuration required. Enclosure and/or controller configuration are vendor specific.
Pi cannot boot from a software RAID array.	Pi can boot from a FAT partition on the RAID array.
Root partition can be on the RAID array but significant additional configure is required including creation and maintaining an initrd.[40]	Root partition can be on the RAID array.
	External enclosures are often transparent to the OS and present as a single device.
	PCIe RAID controllers[41] may require additional drivers and may need configuration tools (or their own BIOS or firmware) that are not compatible with Pi.

[40] Such configuration is outside the scope of this guide.

[41] Only usable with a CM4 at time of writing.

13.3 Common RAID Levels

13.3.1Linear/Span/JBOD[42] Span

Concatenates all drive and present a single device.

Minimum Number of Drives:	2[43]
Usable Capacity:	Total capacity of all drives in the array.
Fault Tolerance:	None
Can Be Online During Rebuild:	No
Performance:	No change

13.3.2RAID 0

Data is striped[44] across all drives.

Minimum Number of Drives:	2
Usable Capacity:	Total capacity of all drives in the array.
Fault Tolerance:	None
Can Be Online During Rebuild:	No
Performance:	May be improved but subject to limitations of the Pi’s USB controller.

13.3.3RAID 1

Data is mirrored[45] across two or more drives.

Minimum Number of Drives:	2
Usable Capacity:	Limited to the capacity of the smallest drive.
Fault Tolerance:	All but one drive
Can Be Online During Rebuild:	Yes
Performance:	Reduced write speed particularly with software RAID.

[42] **J**ust **A** **B**unch **O**f **D**iscs

[43] 2 to be useful. Can be forced to 1 but there is little point in doing so.

[44] First block of a file is written to the first drive, second to the second, etc. Wraps if more data blocks than drives.

[45] Identical data is written to all drives.

13.3.4RAID 10, RAID 1+0

Combinations of striping and mirroring.

Minimum Number of Drives:	4
Usable Capacity:	Half the total capacity of all drives or less depending on precise configuration.
Fault Tolerance:	One drive. Two if they are in different stripe sets and different positions in the array.
Can Be Online During Rebuild:	Depends on configuration.
Performance:	Reduced write speed particularly with software RAID.

13.3.5RAID 5

Striping with distributed parity.

Minimum Number of Drives:	3
Usable Capacity:	Total capacity of N – 1 drives. Where N is the total number of drives and all drives are the same size.
Fault Tolerance:	One drive
Can Be Online During Rebuild:	Yes
Performance:	Reduced write speed due to the need to calculate and write parity data particularly with software RAID.

13.3.6RAID 6

Striping with double distributed parity.

Minimum Number of Drives:	4
Usable Capacity:	Total capacity of N – 2 drives. Where N is the total number of drives and all drives are the same size.
Fault Tolerance:	Two drives
Can Be Online During Rebuild:	Yes
Performance:	Reduced write speed due to the need to calculate and write parity data particularly with software RAID.

13.3.7Additional RAID Levels And Combinations

There are many additional RAID levels and combinations of levels. For further reading see https://en.wikipedia.org/wiki/RAID and others.

Not all RAID levels are supported in the software RAID driver or in all RAID controllers/enclosures.

13.4 Configuration – Hardware RAID

Refer to the documentation for your RAID controller and/or drive enclosure.

Once the controller and/or enclosure have been configured set up the drive(s) per 9

13.5 Configuration – Software RAID

All instructions and examples below assume the following:

- Booting from SD card with the root partition on the SD card.
- All device names (/dev/sda, /dev/md0, etc.) are examples and may be different on your system.

Care must be taken to ensure operations are performed on the correct devices. Using the wrong device will almost certainly result in data loss.

I cannot and will not accept responsibility for any data loss however caused.

13.5.1Partitions or Raw Drives

While software RAID can be configured using raw drives (/dev/sda etc.) this is generally more trouble than it is worth. Partitions provide several advantages:

- Drives differing in size by more than 1% can be used. Unused capacity can be placed in a separate partition and used outside the array.
- It’s much easier to compensate for variance in nominal capacity vs actual capacity and between capacity of different drives. Just ensure all partitions are sized the same.
- It’s more flexible when replacing drives as drives of different capacity to the others can be used.
- No risk of conflicts between the RAID superblock[46] and the drive’s partition table.

[46] The superblock provides metadata about the RAID array.

13.5.2All RAID Levels

1. Update your package lists:

   ```
   sudo apt update
   ```

2. Optionally update your installe dpackages:

   ```
   sudo apt full-upgrade -y
   ```

3. Install the required packages:

   ```
   sudo apt install mdadm -y
   ```

4. Create partitions per 9.3 except partitions can be left unformatted and partition type must be set to `fd`.

   fdisk: `t` then `fd` then `w`.

   gparted: right click then “Manage Flags”

5. Create RAID array(s) (see below).

6. Partition, format and mount RAID array(s). See 9

   As with any partitions it is preferable to use UUID or PARTUUID when mounting as the device node can change across boots. Use the UUID or PARTUUID of the partition(s) created inside the RAID array not those the array was created from.

13.5.3Linear/Span/JBOD Span

Drives and partitions used for the array do not need to be the same size.

```
sudo mdadm --create /dev/md0 --level=linear --raid-devices=2 /dev/sda1
/dev/sdb1
```

Replace `/dev/md0`, `2`, and `/dev/sda1` etc. as required. More than two devices may be used.

13.5.4RAID 0

Drives and partitions used for the array do not need to be the same size.

```
sudo mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sda1 /dev/sdb1
```

Replace `/dev/md0`, `2`, and `/dev/sda1` etc. as required. More than two devices may be used.

13.5.5RAID 1

Drives and partitions used for the array do not need to be the same size but when this is the case capacity will be limited to that of the smallest and any excess capacity will be lost.

```
sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
```

Replace `/dev/md0`, `2`, and `/dev/sda1` etc. as required.

13.5.6RAID 10

For simplicity all drives and partitions used to form the array should be the same size.

```
sudo mdadm --create /dev/md0 --level=10 --raid-devices=4 /dev/sda1 /dev/sdb1
/dev/sdc1 /dev/sdd1
```

Replace `/dev/md0`, `4`, and `/dev/sda1` etc. as required.

13.5.7RAID 5

For simplicity all drives and partitions used to form the array should be the same size.

```
sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sda1 /dev/sdb1
/dev/sdc1
```

Replace `/dev/md0`, `3`, and `/dev/sda1` etc. as required.

13.5.8RAID 6

For simplicity all drives and partitions used to form the array should be the same size.

```
sudo mdadm --create /dev/md0 --level=6 --raid-devices=4 /dev/sda1 /dev/sdb1
/dev/sdc1 /dev/sdd1
```

Replace `/dev/md0`, `4`, and `/dev/sda1` etc. as required.

14 Troubleshooting

14.1 Basics

- Check the server is switched on.
- Check all network hardware (routers, switches, Wi-Fi access points, etc.) are switched on and working.
- Check the server is connected to the network.
- Check the server has an IP address for the Ethernet or Wi-Fi interface in use.
- Check the server software (smbd, nmbd, etc.) is running.
- Check the client is connected to the same network as the server.
- Check the client has an IP address for the Ethernet or Wi-Fi interface in use.
- Check the client can `ping` the server using the server's IP address.
- Check the client can `ping` the server using the server's hostname.

If any of the above fail, troubleshoot your network issues and come back to NAS issues once your network is functioning. General network troubleshooting is outside the scope of this guide.

14.2 Samba Shares Not Showing In Windows Explorer

Browsing Samba shares and/or servers in windows explorer requires either that both Windows and the server have v1 of the protocol enabled or that the Samba server have wsdd[47] installed. Recent version of Windows and Samba do not have v1 enabled by default.

Shares can still be accessed by \\servername\share e.g. `\\morpork\shades`.

14.2.1Enabling SMB v1

Enabling SMBv1 is not recommended as there are serious security issues with it.

To enable SMBv1 on the server:

1. Add the following to the `[global]` section of your smb.conf:

   ```
   server min protocol = NT1
   ```

2. Restart Samba:

   ```
   sudo systemctl restart smbd nmbd
   ```

To enable SMBv1 on Windows clients:

1. Open `Control Panel`
2. Open `Programs and Features`
3. Click `Turn Windows features on of off`
4. Find and tick `SMB 1.0/CIFS File Sharing Support`
5. Click `OK`

[47] **W**eb **S**ervice **D**iscovery **D**aemon

14.2.2Installing wsdd

wssd can be found at https://github.com/christgau/wsdd. Installation on Raspberry Pi OS is as follows:

Ensure you have python 3.7 or later installed. Raspberry Pi OS Buster and later this will be the case.

1. Install git

   ```
   sudo apt update
   sudo apt install -y git
   ```

2. Clone the wsdd repository:

   ```
   git clone https://github.com/christgau/wsdd
   ```

3. Copy the script:

   ```
   sudo cp wsdd/src/wsdd.py /usr/bin/wsdd
   ```

4. Ensure it is executable:

   ```
   sudo chmod a+x /usr/bin/wsdd
   ```

5. Copy the systemd service:

   ```
   sudo cp wsdd/etc/systemd/wsdd.service /etc/systemd/system
   ```

6. Open `/etc/systemd/system/wsdd.service` in your preferred text editor. You will need to be root or use sudo.

7. Replace `Group=nobody` with `Group=nogroup`

8. Save and close

9. Enable the service:

   ```
   sudo systemctl enable wsdd
   ```

10. Start the service:

    ```
    sudo systemctl start wsdd
    ```

If desired, the directory created in step 2 can now be removed and git uninstalled.

14.3 Problems Writing to a Resource

In order to write to a Samba share, NFS export, or SFTP server all of the following must be true:

- The client must have mounted the resource read/write.
- The local user on the client must have write permission on the mounted resource.
- The Samba share, NFS export or SFTP server must allow writing by the user who has logged in.
- The Linux user on the server must have write permission to both the file and the directory containing it.
- The driver for the file system in use on the server must allow writing. Drivers for some non-Linux file systems do not.
- The disc partition must be mounted read/write on the server.

Clients cannot override server permissions.[48]

14.4 Samba Reports The Wrong Amount Of Free Space To Clients

This is most likely when your server shares a directory whose subdirectories are mount points. Share the mount points rather than the directory containing them.

14.5 Unable To Upload Large Files

Assuming there is sufficient free space and no disc space quotas have been set up, check the file system in use on the server can handle files of the required size. FAT32 has a maximum file size of 4GB. NTFS, ext4, and exFAT for practical purposes have no upper limit[49].

14.6 Some Clients Can Connect Others Cannot

Perform basic network troubleshooting (see 14.1) then check the following:

- For Samba, check the `hosts allow =`, `hosts deny =`, `invalid users=`, and `valid users =` share settings
- For NFS, check the client name/IP section of the export settings.

14.7 RAID

Troubleshooting RAID issues can be complex and as such is outside the scope of this guide.

[48] NFS clients of servers exporting Linux native file systems may be able to change permissions and ownership subject to the usual rules.

[49] They do, in fact, have an upper limit but it’s several orders of magnitude bigger than the biggest available HDDs.

15 Change Log

2023-06-06

New section 9.2

Renumber pre-existing 9.2 and later.

2022-08-02

Added advice on drive selection

Added basic RAID information

Moved Troubleshooting to section 14

2022-04-21

Added Change Log.

Added 12.10

Added 11.4