unRAID Server Build Guide

Created by Zack Fravel

The purpose of this guide is to assist others in building and setting up their own Unraid server. There are a ton of uses for Unraid, however, this guide will serve as a bare-bones introduction and should not be expected to go into depth on every possible implementation of advanced virtualization techniques or whatever it may be; there is a huge community on the internet waiting for those questions! This guide is meant as a jumping off point for beginners to get a server up and running for basic needs such as photo library backups, plex media server file hosting, etc. A large portion of this guide is adapted from https://wiki.unraid.net. Further investigation and experimentation is encouraged!

Contents

1 Overview

The following document is meant to serve as a jumping-off point for anyone looking to build their on Unraid server. Unraid OS allows sophisticated media aficionados, gamers, and other intensive data-users to have ultimate control over their data, media, applications, and desktops, using just about any combination of hardware. Common uses for an Unraid server include, but are not limited to:

- Data Storage
- Digital Media Server
- Game library caching
- Multi-OS Virtualization

Unraid is a file server system, based upon the Slackware distribution of Linux. Unraid allows you to build an 'array' of hard drives and share those drives across the local network for all users to access the files on those drives. In order to help combat physical drive failure, Unraid allows for the inclusion of a parity drive, which is used to contain parity bit information for all of the other data drives. This information can be used to rebuild the data on any drive if it has to be replaced (either for an upgrade or because it died). The benefits of Unraid versus other systems such as FreeNAS are:

- Ability to have different sized drives in the array (RAID normally requires you to have all drives in the system be of the same physical size)
- Ability to dynamically add drives to the array as they are acquired - most RAID arrays are very sensitive to "growing" the array and the operation typically involves a full backup before attempting to grow the array. If you've got 5 TB of data, this likely isn't something you're going to attempt!
- Non-striping of the data or parity - normally, the data and parity information in a RAID is striped across the disks, meaning portions of each file are stored on various drives. In Unraid, files are stored contiguously on a single drive. This means that if there is a catastrophic problem (e.g. multiple drives fail), you can likely retrieve the information from a single disk.

Of course there are some drawbacks to Unraid:

- Some limitations in hardware support - see Section 2.1: Compatibility Requirements.
- It's not free (for more than 3 drives) or open source.
- Since the data is not "striped" across multiple drives, read performance is limited to the speed of the individual drive. In most RAID implementations, data can be read much faster than an individual drive's performance because various drives are accessed simultaneously, thus aggregating the read performance.

In the following sections I will provide documentation on recommended hardware for Unraid, as well as detail my own Unraid build along with software installation. Additional information can be found online at http://www.wiki.Unraid.net and a vast community of others out there helping each other out on forum threads across the net.

2 Hardware and Build

Everyone will have their own set of goals and budget for their server. For my server, I was looking to spend around $200-300, excluding hard drives. There is tons of used server equipment for sale on ebay, and maybe even craigslist in your area! I chose to build with a combination of used parts on Ebay and a few new components off Newegg. In the following sections I'll detail Unraid's compatibility restrictions along with details on how to source parts and the build itself.

2.1 Compatibility Requirements

Detailed information available at https://wiki.Unraid.net/Hardware_Compatibility.

For the most part, Unraid is pretty loose with hardware requirements. One exception being Unraid must be run off a USB flash drive. An Unraid Server runs entirely from RAM, minimizing writes to the Flash device. However, it is also used to store vital system configuration data and must remain installed in your server during operation. USB Drive Requirements are:

- Must contain a unique GUID (Globally Unique Identifier)
- Minimum 1GB in size, maximum 32 GB in size

The minimum hardware requirements to install Unraid OS 6 for data storage usage are:

- A 64-bit capable processor, 1.0 GHz or better
- At least 2 GB of RAM
- Linux hardware driver support for storage, Ethernet, and USB controllers
- Two hard disk drives to ensure data protection with a parity disk

As you load more applications you load, you'll need to add additional resources. To run a wide gamut of applications your system should consist of:

- A quad-core CPU (2.4 GHz or faster)
- 4 - 8 GB of RAM
- GPU for hardware video transcoding

2.2 Build

For my build, I plan on the Unraid machine function soley as a file server. While I do plan on running a Plex Media Server, I am going to offload the hardware transcoding and server hosting duties to my living room PC. Others might want to consolidate these features into one machine (i.e. Run a Windows VM on the Unraid server itself for gaming). I purchased two SSDs, one to serve as a cache and another to hold Plex metadata, and three 4TB HDDs (1 parity) totaling 8TB of storage. Cache SSDs allow you to max out your internet speed as downloads write to the SSD and then on a schedule migrate to the HDD array.

As mentioned above, I was looking to spend a few hundred dollars on a fairly budget file server through a combination of eBay and Newegg purchases. This build was inspired by the following guide:
https://forums.serverbuilds.net/t/guide-nas-killer-4-0-fast-quiet-power-efficient-and-flexible-starting-at-125/667

- CPU: Intel Core i3-8100 @ 3.60GHz

  Noctua NH-L9i, Premium Low-Profile CPU Cooler
- Motherboard: Asus Prime H310M-E R2.0
- RAM: Corsair Vengeance LPX 16GB (2x8GB) DDR4 DRAM 2400MHz
- Power Supply: EVGA 500 W1, 80+ WHITE 500W

- Case: Fractal Design Node 804
- Drives

  Boot Drive: Samsung MUF-32AB/AM FIT Plus 32GB - 200MB/s USB 3.1 Flash Drive

  Cache Drives: Pioneer 3D NAND Internal SSD 256GB (2x)

  Plex Metadata Drive: WD Blue 3D NAND 250GB Internal M.2 2280 SSD

  Data Drive: WD Red 4TB NAS Hard Drive - 5400 RPM (1x)

  Data Drive: Seagate Ironwolf 4TB NAS Hard Drive - 5900 RPM (2x)
- Miscellaneous

  SAS9211-8I 8PORT 6GB SATA+SAS PCI-e 2.0

  Benfei 15 Pin SATA Power Y-Splitter Cable 8 Inches - 2 Pack

  CyberPower CP850AVRLCD Intelligent LCD UPS System

The build will consist of a cache array in RAID0 for a total of 500GB. The cache helps with speeding up uploads to the server. A mover moves data from the cache to the data drives on a schedule. The M.2 drive is going to be for Plex metadata. I wanted to make sure the drive containing the metadata would have the fastest performance to make browsing the plex a breeze by preventing HDDs from being spun up. The data drives are a combination of a new WD Red drive and some used Seagate Ironwolf NAS drives I had in a previous storage solution.

3 Set Up and Software Installation

To give some background, two of the WD Red drives are already in use as a pair in a RAID1 configuration in a Terramaster box. When I create the initial array, it will be with one drive. Then I will transfer the data from my previous backups to the Unraid array. Once the data is transferred, I will add one of the old drives as a parity and allow the sync to complete as a safety measure. Once that is done I will add the last drive as the second data drive and the migration will be complete.

Upon completion of the build, I made sure and booted up the system into the BIOS to ensure each component was connected and recognized. The LSI controller I installed in the PCI-e slot takes about 5 minutes or so to boot up, which delays the BIOS boot time. Luckily since this machine is meant to be always on, that's not really an issue. The LSI controller has its own BIOS where you're able to see which devices it recognizes. My 4TB WD Red showed up there and my other drives showed up in the motherboard BIOS. Once I set up my fan curves and other miscellaneous BIOS options, I created my Unraid USB stick and booted into it.

3.1 Preparing USB Flash Drive

1. Plug the Flash into your PC and re-format it using Windows (Right-Click the Flash under Computer and select Format):

   For File system, leave it as Default

   For Volume label, enter Unraid (exactly 6 capital letters)

2. Download the Unraid USB Creator Tool: https://Unraid.net/download

   After opening the download, you can choose between running a stable or next release, customize your network configuration prior to booting, and opt for EFI booting if you prefer. This tool is an open source program digitally signed by Lime Technology, Inc.

3. Follow the wizard to create your boot disk.

3.2 Installing Unraid

Once you've booted into the Unraid installer, allow the operating system to unpack until you're presented with options. From this point you can press 'Enter' on unRAID OS. The server is now running headless and can be accessed from any machine on the network.

Unraid's management interface (the webGui) is incompatible with most ad-blocker solutions. It is for this reason that they strongly suggest that users leveraging an ad-blocker in their browser first add the Unraid server to the ad-blocker whitelist to ensure the ad-blocker doesn't affect the webGui. Go to your laptop or PC, type in http://tower (or http://tower.local on MacOS) and you'll be presented with the registration page. Purchase the appropriate Unraid licence and you're ready to set up your array. You can change the name used for your Unraid server from the webGUi by going to Settings/System Settings/Identification.

3.3 Creating Array

After installing a registration key, you are ready to begin assigning devices for Unraid to manage. Click on the Main tab from the Unraid webGui and follow these guidelines when assigning disks:

- **Always pick the largest storage device available to act as your parity device(s).** When expanding your array in the future (adding more devices to data disk slots), you cannot assign a data disk that is larger than your parity device(s). For this reason, it is highly recommended to purchase the largest HDD available for use as your initial parity device, so future expansions aren't limited to small device sizes. If assigning dual parity disks, your two parity disks can vary in size, but the same rule holds true that no disk in the array can be larger than your smallest parity device.
- **SSD support in the array is experimental.** Some SSDs may not be ideal for use in the array due to how TRIM/Discard may be implemented. Using SSDs as data/parity devices may have unexpected/undesirable results. This does NOT apply to the cache / cache pool.
- **Using a cache will improve array performance.** It does this by redirecting write operations to a dedicated disk (or pool of disks in Unraid 6) and moves that data to the array on a schedule that you define (by default, once per day at 3:40AM). Data written to the cache is still presented through your user shares, making use of this function completely transparent.
- **Creating a cache-pool adds protection for cached data.** If you only assign one cache device to the system, data residing there before being moved to the array on a schedule is not protected from data loss. To ensure data remains protected at all times (both on data and cache disks), you must assign more than one device to the cache function, creating what is called a cache-pool. Cache pools can be expanded on demand, similar to the array.
- **SSD-based cache devices are ideal for applications and virtual machines.** Apps and VMs benefit from SSDs as they can leverage their raw IO potential to perform faster when interacting with them. Use SSDs in a cache pool for the ultimate combination of functionality, performance, and protection.

Once you have all your devices assigned, you can click the Start button under Array Operation. This will mount your devices and start the array. New devices added to disk or cache device slots will appear as 'Unformatted' and will be unusable for storing files until you format them. Unraid 6 defaults to using the XFS filesystem for all devices, but if you define a cache pool then BTRFS will automatically be used for those devices (you can change the default file system under Settings/System Settings/Disk Settings).

To format your devices for use, you must click the check box under 'Array Operation' that says Format, acknowledge the resulting prompt (read it carefully), and then click the Format button.

Even before the devices are formatted, a parity sync will be performing in the background to initialize the protection of the array. Until the sync is completed, the array will operate but in an unprotected state. It is recommended to wait until the initial parity sync completes before adding data to the array.

3.4 Adding Drive to Array

This is the normal case of expanding the capacity of the system by adding one or more new hard drives.

The capacity of any new disk(s) added must be the same size or smaller than your parity disk. If you wish to add a new disk which is larger than your parity disk, then you must instead first replace your parity disk. (You could use your new disk to replace parity, and then use your old parity disk as a new data disk).

1. Stop the array.
2. Power down the server.
3. Install your new disk(s).
4. Power up the server.
5. Assign the new storage device(s) to a disk slot(s) using the Unraid webGui.
6. Start the array.
7. Unraid will now automatically begin to clear the disk which is required before it can be added to the array.
8. Once the disk has been cleared, an option to format the disk will appear in the webGui. At this point the disk is added to the array and shows as unmountable and the option to format unmountable disks is shown.
9. Click the check box to confirm that you want to proceed with the format procedure.
10. The format button will now be enabled so you can click on it to start the formatting process.
11. The format should only take a few minutes and after the format completes the disk will show as mounted and ready for use.

You can add as many new disks to the array as you desire at one time, but none of them will be available for use until they are both cleared and formatted with a filesystem.

It is not mandatory for an ‘Unraid system to have a parity disk, but it is normal to provide redundancy. A parity disk can be added at any time, Each parity disk provides redundancy against one data drive failing. Any parity disk you add must be at least as large as the largest data drive (although it can be larger). If you have two parity drives then it is not required that they be the same size although it is required that they both follow the rule of being at least as large as the largest data drive.

The process for adding a parity disk is identical to that for adding a data disk except that when you start the array after adding it Unraid will start to build parity on the drive that you have just added. If you wish to upgrade your parity device(s) to a larger one(s) so you can start using larger sized disks in the array or to add an additional parity drive, the procedure is as follows:

1. Stop the array.
2. Power down the server.
3. Install larger parity disks.
4. Power up the unit.
5. Assign a larger disk to the parity slot (replacing the former parity device).
6. Start the array.

When you start the array, the system will once again perform a parity sync to the new parity device and when it completes the array will once again be in a protected state. If you have a dual parity system and wish to upgrade both of your parity disks, it is recommended to perform this procedure one parity disk at a time, as this will allow for your array to still be in a protected state throughout the entire upgrade process. Once you’ve completed the upgrade process for a parity disk, the former parity disk can be considered for assignment and use in the array as an additional data disk (depending on age and durability).

3.5 Replacing Disks

There are two primary reasons why you may wish to replace disks in the array:

- A disk needs to be replaced due to failure or scheduled retirement (out of warranty / support / serviceability).
- The array is nearly full and you wish to replace existing data disk(s) with larger ones (out of capacity).

In either of these cases, the procedure to replace a disk is roughly the same, but one should be aware of the risk to data loss during a disk replacement activity. Parity device(s) protect the array from data loss in the event a disk failure. A single parity device protects against a single failure, whereas two parity devices can protect against losing data when two disks in the array fail. A detailed chart is on the following page:

Data Protection During Disk Replacements		
	With Single Parity	*With Dual Parity*
Replacing a single disk	Array cannot tolerate a disk failure without potential data loss to both the disk being replaced and the additional disk that has failed.	Array can tolerate up to one additional disk failure without potential data loss
Replacing two disks	Not possible!	Array cannot tolerate a disk failure without potential data loss to both the disk(s) being replaced and the additional disk that has failed.

As noted previously, with a single parity disk, you can replace up to one disk at a time, but during the replacement process, you are at risk for data loss should an additional disk failure occur. With two parity disks, you can replace either one or two disks at a time, but during a two disk replacement process, you are also at risk for data loss. Another way to visualize the previous chart:

Array Tolerance to Disk Failure Events			
	Without Parity	*With Single Parity*	*With Dual Parity*
A single disk failure	Data from that disk is lost	Data is still available and the disk can be replaced	Data is still available and the disk can be replaced
A dual disk failure	Data on both disks are lost	Data on both disks are lost	Data is still available and the disks can be replaced

It is important to realize what is meant by the term failed drive:

- It is typically used to refer to a drive that is marked with a red 'x' in the Unraid GUI.
- It does NOT necessarily mean that there is a physical problem with the drive (although that is always a possibility). More often than not the drive is OK and an external factor caused the write to fail.
- If you have sufficient parity drives then Unraid will emulate the failed drive using the combination of the parity drive(s) and the remaining 'good' drives. From a user perspective this results in the system reacting as if the failed drive is still present.

When a disk is marked as disabled and Unraid indicates it is being emulated then the following points apply:

- Unraid will stop writing to the physical drive. Any writes to the 'emulated' drive will not be reflected on the physical drive but will be reflected in parity so from the end-user perspective then the array seems to be updating data as normal.
- When you rebuild a disabled drive the process will make the physical drive correspond to the emulated drive. You can, therefore, check that the emulated drive contains the content that you expect before starting the rebuild process
- If a drive is being emulated then you can carry out recovery actions on the emulated drive before starting the rebuild process. This can be important as it keeps the physical drive untouched for potential data recovery processes if the emulated drive cannot be recovered.

A replacement drive does not need to be the same size as the disk it is replacing. It cannot be smaller but it can be larger. If the replacement drive is not larger than any of your parity drives then the simpler procedure below can be used. In the special case where you want to use a new disk that is larger than at least one of your parity drives then please refer to the Parity Swap procedure that follows instead.

If you have purchased a replacement drive, many users like to pre-clear the drive to stress test the drive first, to make sure it's a good drive that won't fail for a few years at least. The Preclearing is not strictly necessary as replacement drives don't have to be cleared since they are going to be completely overwritten., but Preclearing new drives one to three times provides a thorough test of the drive, eliminates 'infant mortality' failures. You can also carry out stress tests in other ways such as running an extended SMART test or using tools supplied by the disk manufacturer that run on Windows or MacOS.

It is worth emphasising that Unraid must be able to reliably read every bit of parity PLUS every bit of ALL other disks in order to reliably rebuild a missing or disabled disk. This is one reason why you want to fix any disk related issues with your Unraid server as soon as possible.

If you only have a single parity device in your system and a disk failure occurs during a data-rebuild event, the data rebuild will be cancelled as parity will no longer be valid. However, if you have dual parity disks assigned in your array, you have options. You can either let the first disk rebuild complete before starting the second, or you can cancel the first rebuild, stop the array, replace the second failed disk, then start the array again. If the first disk being rebuilt is nearly complete, it's probably better to let that finish, but if you only just began rebuilding the first disk when the second disk failure occurred, you may decide rebuilding both at the same time is a better solution.

3.6 Creating Shares

Unraid uses "shares" to organize your file system. Shares can be configured in a number of different ways including, but not limited to:

- Allocation Method
- Split level

  Manage how Unraid splits your files across drives (i.e. set rules to define what subfolders to keep on the same disk and what folders can be split across disks).
- Export Settings

To create a share, click the "Shares" tab in the Unraid WebGUI and click "Add Share" at the bottom of the list. From here, you'll be taken to all the various configuration options you have on this share, user access, etc. Once you're ready, hit "Done" and the share is now ready have files written to it. The following is a list of default Unraid shares:

- appdata

  Contains all the necessary appdata for Docker containers.
- domains

  Contains the storage for Virtual Disks.
- isos

  Contains the ISOs for Virtual Machines.
- system

  Used as the default locations for the docker image and the libvirt image.

3.7 Third Party Software Packages

There is a huge collection of third party applications available for Unraid. The easiest way to get access to all of them is installing a plugin called "Community Applications." This plugin can be found at https://forums.unraid.net/topic/38582-plug-in-community-applications/.

To install this plugin, paste the following URL into the Plugins / Install Plugin section:

https://raw.githubusercontent.com/Squidly271/community.applications/master/plugins/community.applications.plg

After installation, a new tab called "Apps" will appear on your unRaid webGUI. To see what the various icons do, simply press Help or the (?) on unRaid's Tab Bar. CA also has a dedicated Settings section (click Settings) which will let you fine tune certain aspects of its operation. I installed the following packages to assist bringing my server up to production:

- Unassigned Devices - Mount Non-Array Drives

  This plugin uses UDEV to automount and share disks that are not part of your Unraid array. Available devices are listed under the "Main/Unassigned Devices" tab. You can also mount remote SMB and NFS shares on other servers.

  Support for HFS+ and exFAT disk formats, and enabling destructive mode.

- FileZilla - FTP Transfers

  FileZilla is a cross-platform graphical FTP, SFTP, and FTPS file management tool with a vast list of features. The GUI of the application is accessed through a modern web browser (no installation or configuration needed on client side) or via any VNC client. Also, being based on Alpine Linux, size of this container is very small (less than 100MB in download size).

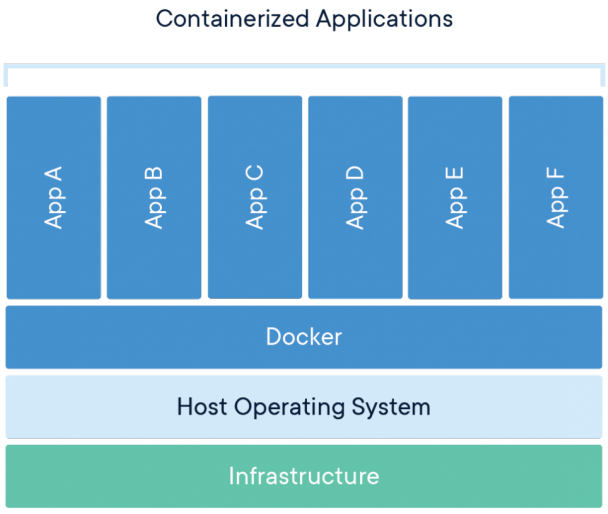
- Krusader - File Management

  Krusader is an advanced orthodox file manager for KDE and other desktops in the Unix world. It is similar to the console-based GNU Midnight Commander, GNOME Commander for the GNOME desktop environment, or Total Commander for Windows, all of which can trace their paradigmatic features to the original Norton Commander for DOS.

- unBALANCE - Disk Management

  This plugin frees up space from one of the disks in the array by moving folders and files to other disks.

After reading that you might be asking what is a container? A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. Unraid uses an application called Docker to manage apps and containers on your server.



A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings. Container images become containers at runtime and in the case of Docker containers - images become containers when they run on Docker Engine. Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging.

Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient. Docker containers are (for all intents and purposes) completely separate from the rest of the Unraid system. A container has no idea that its running on Unraid (or any other system), and has no concept of any other containers that may also be running. This allows us to easily allocate systme resources to specific third party applications without lots of conflict between them.

4 Accessing and Modifying File System

The following sections are a reference for all the information needed and steps to take to access Unraid shares, whether it be locally or remotely.

4.1 Local

- Filezilla FTP
  1. From the Unraid Dashboard, click the FileZilla icon in the "Docker Containers" section.
  2. Click "WebGUI." From here you have access to your entire file system and can connect to hosts.
- Krusader
  1. From the Unraid Dashboard, click the Krusader icon in the "Docker Containers" section.
  2. Click "WebGUI." From here you are presented with a dual panel, commander-style file system managment interface where you can perform all sorts of operations on your files.
- unBALANCE
  1. From the Unraid Dashboard, click the unBALANCE icon in the "Plugins" section and enable the plugin.
  2. Navigate to https://host_name:6238/ (default)
  3. From here, you can "scatter" and "gather" folders and files on your unraid array to and from specific disks.
- Windows or MacOS
  1. Open "This PC" on Windows or "Finder" on Mac OS.
  2. Click "Map Network Drive" Windows or "CMD+K" to Connect to Server on Mac OS.
  3. Type "\\tower\drive" where 'drive' is the name of the share you want to access and 'tower' is the name of your server. On Mac OS put "smb:\\tower\drive"

4.2 Remote

There are a few things to setup for remote server access:

1. DuckDNS - Dynamic DNS Server that tracks you public IP address. This is then defined in your VPN server so your VPN client knows where to connect.
2. OpenVPN Server - VPN Server hosted from an Unraid docker container that allows remote connections to the file system / home network.

Below I will detail the basic outline of what it takes to setup these two services. Everyone's configuration is a little differnet so I invite anyone reading to do their own research and also follow "Spaceinvader One" on YouTube for awesome in-depth tutorials of everything I lay out in this document. Overall, these services basically track your router's public IP address and assigns it a hostname so you can reliably use the same domain to connect to your VPN server.

4.2.1 DuckDNS

4.2.2 OpenVPN Server

4.3 Migrating Existing Data

The following section will go over in detail the strategy I used to move my existing data and drives to my Unraid machine. To give some background, I had already been using a Terramaster RAID1 enclosure with two 4TB drives to store my files. My plan was to move these files to the Unraid server while also moving the drives in to gain the 8TB of storage. The steps I took were the following:

1. Initialize unRAID array with one 4TB data drive.
2. Transfer data from Terramaster RAID1 enclosure by USB with the unassigned devices plugin.

   I was able to transfer the files to my unRAID shares using the FileZilla WebGUI.
3. Once data transfer is complete, add one of the old drives to array as a parity drive. Let the parity sync complete and your data is secure.
4. Finally, add the third drive to the array.

   All drives are now moved and data is migrated.

5 Uninterruptible Power Supplies (UPS)

Setting up a UPS for unRAID is an incredibly straight-forward process (provided you are using a compatible device). unRAID uses APCUPSD, a built in daemon for controlling APC UPS devices. Many other brands of smart UPS work as well (I used Cyberpower). More about APCUPSD here: http://www.apcupsd.org/wordpress/.

Once you've plugged in your UPS and connected the USB cable to your server:

1. Open the 'Settings' Tab in unRAID.
2. Click 'UPS Settings.'
3. Set 'Start UPS Daemon' to 'Yes.'
4. Set 'Runtime left to initiate shutdown' and 'Battery level to initiate shutdown.'
5. Once your settings have been applied, double check unRAID has recognized your UPS and you're good.

6 Scheduled Backups

There are a ton of different solutions out there for making scheduled backups of your computers to an unRAID server. My personal preference is using Macrium Reflect for my backups. Macrium reflect allows you to backup your entire machine as a disk image that you could reload at anytime. Think of it as saving the 'state' of your computer on a given date. Software found here: https://www.macrium.com/reflectfree. Macrium Reflect has a ton of different paid versions but the free version should work for most people's use cases.

I created a share on my unRAID array called "Computer Backups," which is where I will tell Macrium Reflect to upload its images to. Once you open Macrium Reflect, you're going to want to create a new backup by selecting "Image Selected Disks on this Computer." You will be brought to a dialogue wizard that will walk you through what you want included in the backup, how many to keep, when to do housekeeping, etc. Once you've got your backup definition file set with your schedules, all you need to ensure is Macrium is given privileges to wake your computer from sleep and run while locked.

Now to verify that the Windows Task Scheduler is set to wake the computer if it's sleeping. Click Start, then type in "schedule"; wait for Task Scheduler to appear in the list and click it. The Task Scheduler window should now appear; on the top left of the screen, it will say "Task Scheduler (Local)" and under that, a folder that says "Task Scheduler Library". Click on the Task Scheduler Library to highlight it. Next, scroll through the "Name" heading (near the top middle of the screen) until you see "Macrium-Backup-XXX" in the list, and double click to

edit it. The backup settings schedule will appear; click the Conditions tab and look under the heading "Power" and ensure that "Wake the computer to run this task" has a check mark beside it.

Next, it's time to check the computer's Sleep / Hibernate settings. Preferably you should only enable Sleep and not Hibernate, since Macrium Reflect may have issues with the Hibernate feature. To do so: click Start and type in "power options"; wait for the Power Options icon to appear and click it. Under "Preferred Plans", click on the "Change plan settings" link, then click on the "Change advanced power settings" link. A new "Power Options" window will appear; scroll through the list of options until you see the "Sleep" setting, and expand its list of options. Expand the "Allow hybrid sleep" and ensure the setting is Off. Do the same for "Hibernate after" - set it to Never. For "Allow wake timers", select Enable, then click Apply, then OK.

If you're using a laptop, you will also want to change the way your computer behaves when the lid is closed - you will need to disable the sleep there as well. To do so, click Start and type in "power options", then click on the Power Options icon when it appears. Near the top left of the screen, click the link that says "Choose what closing the lid does". The System Settings window will appear; in the middle of the screen, look for the heading "When I closed the lid:" and select "Do nothing" for both "On Battery" and "Plugged in".

7 Google Photos

I wanted my server to have an automated script setup to backup my Google Photos library. This is arguably unnecessary, but I think it'll be nice to have the option to clear out my library instead of paying Google to obtain more storage in the future. Luckily, a user known as "Logix" on the Linux Uprising forum has detailed a script that utilizes the google photos developer API to do just this (link below). You can skip the beginning section, for our purposes we're most interested in the section titled "Create a Google Cloud project, enable the Photos Library API and create an OAuth client ID:"

https://www.linuxuprising.com/2019/06/how-to-backup-google-photos-to-your.html

On the unRAID side, luckily a user known as "rix1337" has recently created a docker container to install the necessary dependencies to run the backup script. The setup instructions after installing the docker container are as follows:

1. Place your client_secret.json at /config
   How to get it is best described at Logix's Article at Linux Uprising
2. Afterwards you need to sign into the application once which cannot be done headlessly (using the "Syncing" command)
3. Afterwards you can call the "Syncing" command any time you wish, as long as the container is running (e.g. by using cron).

Syncing: docker exec -it GooglePhotosSync gphotos-sync /storage

/storage is the path defined in the setup of the docker container where you want the script to backup your library. I suggest creating a separate folder and/or share for this purpose. Before all this you should follow the instructions laid out on the Logix article to get your client_secret.json file. Place the file in the GooglePhotosBackup folder in the appdata share.

Once the plugin is installed, open the Unraid terminal window and run the "Syncing" command laid out above. You'll be asked to sign into your google account once and then you'll be able to schedule the script to run automatically. To do this, I downloaded the "User Scripts" plugin and added the "Syncing" command as my own script titled "gphotobackup" and set it to run weekly. Once I set it I ran the script and verified the files were downloaded in the correct place.

8 Plex Media Server

I'll begin this section just saying there are a lot of different ways to configure a Plex media server, this is just how I chose to do it! The Plex media server itself will be running on an older gaming PC I had already with an

Nvidia GTX 1070 which I will be using for hardware accelerated stream transcoding. A lot of different GPUs can be used for hardware accelerated transcoding, this is just what I had. The file for the libraries themselves will be hosted on the Unraid server, along with a host of other features. I'll lay out below my intentions for the Plex server.

1. I want friends to be able to share a library and stream content.
2. I want a method for securely connecting to add new content or to download a local copy.
3. I want all downloads to be secure and hidden from my ISP.
4. I want anyone to be able to restart / perform basic troubleshooting remotely.

I'll be using a host of automation services to serve as the middleman between my downloaders (Deluge for torrents, Sabnzbd for Usenet). If you don't know what Usenet is and are interested, read more here: https://en.wikipedia.org/wiki/Usenet. These middlemen are Unraid docker containers known as Sonarr, Radarr, and Lidarr and they are built for managing TV Series, Movies, and Music libraries respectively.

8.1 Basic Setup

So, before setting up any complex automated downloading system I wanted to make sure the functionality was there with my configuration. I downloaded the Plex Media Server app on my living room PC and mapped a network drive to my Plex server share. There is a specific way Plex wants you to name your files and folders, I won't be detailing that here but there are plenty of resources online showing how it needs to be done.

https://support.plex.tv/articles/naming-and-organizing-your-tv-show-files/

Once the Server app is installed, go ahead and launch it and add a new "Movies" library. Once you have confirmed you can stream content, congratulations! You're ready to move onto the next steps. If something isn't working, check out the "Dashboard" and go to your "Remote Access" tab. There are a few steps to try here for troubleshooting. Most people's setups should work just fine if your router has uPnP enabled. If you aren't interested in automation services and will add content manually to your own personal server, you're pretty much done now! However, if you'd like advanced features like remote access to troubleshoot / restart your server, or automated searching and downloading then read on!

8.2 Advanced Setup

Before moving forward I will say it is worth going on youtube, searching "Spaceinvader One" and watching all the videos he has on topics related to what I'm about to describe in detail and whatever else interests you. He is a great resource for learning all things Unraid and I always find having a video tutorial can make a real difference in learning. With that out of the way, onto the advnaced plex setup.

8.2.1 Setting up Remote Access

8.2.2 Securely Downloading and Managing New Content

9 Summary and Conclusions

Section in progress.

10 References

The following is a list of works cited. Special thanks to the Tech Pod (with Brad and Will) Discord community for answering beginner questions I had as I put this document together.

1. Unraid Documentation - https://wiki.unraid.net/Official_Documentation
2. Build Guide - https://forums.serverbuilds.net/t/guide-nas-killer-4-0-fast-quiet-power-efficient-and-flexible-starting-at-125/667
3. Community Applications - https://forums.unraid.net/topic/38582-plug-in-community-applications/
4. Docker FAQ - https://forums.unraid.net/topic/57181-docker-faq
5. Docker Website - https://www.docker.com/
6. Diskmv FAQ - https://forums.unraid.net/topic/34547-diskmv-a-set-of-utilities-to-move-files-between-disks/
7. FileZilla - https://forums.unraid.net/topic/59047-support-djoss-filezilla/
8. Krusader - https://forums.unraid.net/topic/71764-support-binhex-krusader/
9. Macrium Reflect - https://www.macrium.com/reflectfree
10. rClone - https://rclone.org/googlephotos/
11. Unassigned Devices - https://forums.unraid.net/topic/44104-unassigned-devices-managing-disk-drives-and-remote-shares-outside-of-the-unraid-array/
12. unBALANCE - https://forums.unraid.net/topic/43651-plug-in-unbalance/
13. Macrium Reflect - https://www.macrium.com/reflectfree
14. Macrium Support - https://www.infopackets.com/news/9783/how-fix-macrium-reflect-backup-wont-wake-sleep
15. Google Photos - https://www.linuxuprising.com/2019/06/how-to-backup-google-photos-to-your.html
16. Usenet - https://en.wikipedia.org/wiki/Usenet