GUIDE

Transform Your Old Gaming PC into the Ultimate Homelab NAS

What you'll have running: A powerful, cost-effective Network Attached Storage (NAS) solution with robust data redundancy and optional application hosting, built from your existing gaming PC. **Estimated time:** ~6 hours **Difficulty:** INTERMEDIATE **Power usage:** Highly variable (~50-150W idle depending on existing hardware, number of drives, and power management settings). Old gaming PCs can consume significantly more power than purpose-built NAS appliances, potentially increasing electricity costs.

Hardware needed:

- Existing Gaming PC (CPU, Motherboard, RAM, PSU, Case)
- Multiple Hard Disk Drives (HDDs) for storage (e.g., 2TB+ each, minimum 2 for redundancy)
- Solid State Drive (SSD) for OS boot drive (120GB-250GB recommended)
- SATA data cables (one per HDD/SSD)
- SATA power cables (ensure PSU has enough connectors or acquire splitters)
- USB drive (8GB+ for OS installation media)
- Optional: PCIe SATA expansion card (if motherboard lacks sufficient ports)
- Optional: Additional RAM (16GB+ recommended for TrueNAS, 8GB for Unraid/Linux)
- Optional: Low-power GPU (if existing GPU is power-hungry and not needed for transcoding)

Introduction: Why Repurpose Your Gaming PC for NAS?

We've all been there: that old gaming rig, once a beast, now gathering dust in a corner after an upgrade. Instead of letting it languish, why not give it a second life as the heart of your homelab? Transforming an old gaming PC into a Network Attached Storage (NAS) system is a fantastic way to gain centralized, redundant storage for all your files, media, and backups without breaking the bank.

This guide will walk you through turning that retired powerhouse into a robust NAS. We'll cover everything from hardware assessment and OS selection to setting up data redundancy and optimizing for power efficiency. You'll learn how to leverage powerful software like TrueNAS SCALE, Unraid, or a custom Linux setup to create a flexible, self-hosted data hub.

Why this matters: In an era of increasing cloud service costs and privacy concerns, owning your data is more important than ever. A self-hosted NAS gives you complete control, privacy, and often superior performance for local network access. Plus, it's a stellar learning experience in systems engineering, hardware management, and open-source software.

Hardware Assessment and Upgrades: Maximizing Storage and Efficiency

Before diving into software, let's take stock of your existing gaming PC and identify areas for improvement. The goal is to maximize storage capacity, ensure sufficient connectivity, and consider power efficiency for 24/7 operation.

Existing PC Evaluation

1. **CPU:** Most modern gaming CPUs (Intel i5/i7/i9 from 6th gen onwards, AMD Ryzen 3/5/7) are more than capable for NAS duties, even handling media transcoding for Plex.
2. **Motherboard:** Check the number of available SATA ports. Gaming motherboards often have 4-8 ports, which is a great start. Also, look for available PCIe slots for future SATA expansion cards.
3. **RAM:** While 8GB is a minimum for many NAS OSes, 16GB or more is highly recommended, especially for TrueNAS SCALE (which thrives on RAM for ZFS) or if you plan to run multiple applications (Docker containers, VMs).

4. **PSU (Power Supply Unit):** Gaming PSUs are typically high wattage. Ensure it has enough SATA power connectors for all your planned drives. If not, SATA power splitters are an option, but be mindful not to overload a single cable.
5. **Case:** A gaming case usually offers good airflow and space, but check if it has enough 3.5" drive bays for your HDDs. You might need drive cages or adapters.
6. **GPU:** Your powerful gaming GPU is likely a power hog. If your CPU has integrated graphics (Intel's iGPUs or AMD's APUs), consider removing the dedicated GPU to save significant power, especially if you don't need it for video transcoding.

Essential Upgrades

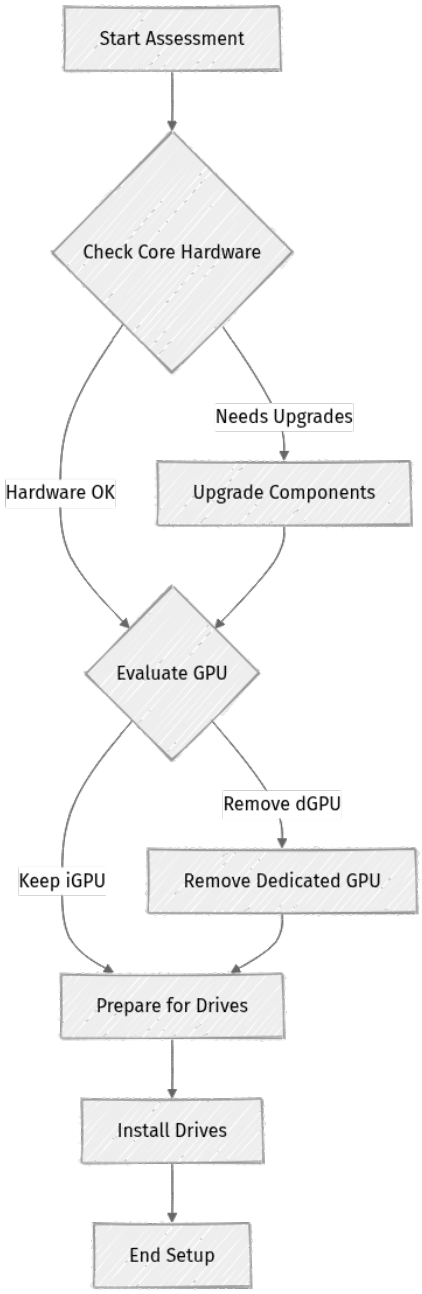
- **Storage HDDs (Minimum 2):** These are your primary data drives. Start with at least two drives for redundancy (e.g., 2x 4TB or 2x 8TB). You can add more later.
- **SSD for OS Boot Drive (120GB-250GB):** This keeps your operating system snappy and separate from your bulk storage. A small SATA SSD is perfect.
- **SATA Data Cables:** You'll need one for each HDD and your OS SSD.
- **SATA Power Cables/Splitters:** Ensure your PSU can power all drives.

Optional Upgrades for the Future

- **PCIe SATA Expansion Card:** If your motherboard runs out of SATA ports, a PCIe card (e.g., an LSI HBA flashed to IT mode for TrueNAS) is a common and reliable solution.
- **Additional RAM:** If you're starting with 8GB, upgrading to 16GB or 32GB will provide a smoother experience, particularly with ZFS or multiple apps.
- **Low-Power GPU:** If your CPU lacks integrated graphics and your dedicated GPU is too power-hungry, a very basic, low-power GPU might be necessary just for initial setup and troubleshooting.

> ⚠️ **Warning:** Old gaming GPUs can draw 50-100W at idle. Removing it can drastically cut your NAS's power consumption.

Here's a simple flowchart illustrating the hardware assessment process:



Choosing Your NAS Operating System: TrueNAS SCALE vs. Unraid vs. Linux

This is a critical decision, as your chosen OS dictates the underlying storage technology, ease of management, and application ecosystem. Each has its strengths and ideal use cases.

TrueNAS SCALE

- **Core Technology:** ZFS (Zettabyte File System) on Debian Linux.

- **Data Redundancy:** RAID-Z (RAIDZ1, RAIDZ2, RAIDZ3) offers robust data integrity with checksums and self-healing capabilities. Requires drives of similar size within a vdev.
- **Flexibility:** Excellent for traditional NAS roles, block storage (iSCSI), and application hosting via Kubernetes (Apps).
- **Ease of Use:** Web UI is comprehensive and powerful, but can have a steeper learning curve for ZFS concepts.
- **RAM Requirement:** ZFS loves RAM. 16GB is a good starting point, 32GB+ for larger pools or heavy application use.
- **Cost:** Free and open source.
- **Ideal for:** Users prioritizing data integrity, performance, enterprise-grade features, and those comfortable with ZFS concepts.

Unraid

- **Core Technology:** Linux with a custom array management system (mdadm for individual drives, XFS/Btrfs for shares).
- **Data Redundancy:** Uses a dedicated parity drive (or two) to protect against single (or double) drive failures. Allows mixing drive sizes efficiently.
- **Flexibility:** Renowned for its flexibility in adding drives of different sizes and excellent Docker container and VM support.
- **Ease of Use:** Very user-friendly web UI, often considered easier to get started with than TrueNAS for mixed-drive arrays.
- **RAM Requirement:** 8GB is usually sufficient for a basic NAS with some Docker containers.
- **Cost:** Paid license (one-time purchase), but offers a trial.
- **Ideal for:** Users who want maximum flexibility with drive sizes, simple expansion, and robust Docker/VM hosting with a focus on media servers.

Linux (e.g., Ubuntu Server LTS) with ZFS/mdadm

- **Core Technology:** Standard Linux distribution with ZFS (OpenZFS) or mdadm (Linux software RAID).
- **Data Redundancy:** ZFS offers the same robust features as TrueNAS. mdadm provides traditional RAID levels (RAID0, RAID1, RAID5, RAID6, RAID10).
- **Flexibility:** Ultimate control and customization. You build everything from the ground up.

- **Ease of Use:** Highest learning curve, as you're primarily working from the command line for setup and management.
- **RAM Requirement:** Similar to TrueNAS for ZFS (16GB+), lower for mdadm (8GB+).
- **Cost:** Free and open source.
- **Ideal for:** Experienced Linux users, those who want granular control, deeply understand their system, or have very specific niche requirements.

> 💡 **Tip:** For most homelabbers repurposing a gaming PC, TrueNAS SCALE offers a great balance of power and a managed interface. If you value drive flexibility and extensive Docker support, Unraid is a strong contender. If you love the command line and want full control, Linux is your playground.

Preparing Your Hardware: Installation and BIOS Configuration

With your hardware assessed and OS chosen, it's time to get hands-on.

Physical Installation of Drives

1. **Power Down and Disconnect:** Completely shut down your PC and unplug it from the wall.
2. **Open Case:** Remove the side panel(s) of your PC case.
3. **Install OS SSD:** Mount your boot SSD in an available 2.5" bay or M.2 slot (if your motherboard supports it and you're using an M.2 NVMe SSD). Connect one SATA data cable from the SSD to an available SATA port on your motherboard, and one SATA power cable from your PSU to the SSD.
4. **Install HDDs:** Mount your storage HDDs in available 3.5" drive bays. Connect one SATA data cable from each HDD to an available SATA port on your motherboard. Connect one SATA power cable from your PSU to each HDD. Use splitters if necessary, but ensure even distribution of power.
5. **PCIe SATA Card (if applicable):** If you're using an expansion card, install it into an available PCIe slot and connect your additional HDDs to it.
6. **Cable Management:** Tidy up cables for better airflow and easier access later.
7. **Close Case:** Once all drives are securely mounted and cabled, close up your PC case.

BIOS/UEFI Configuration

The BIOS (Basic Input/Output System) or UEFI (Unified Extensible Firmware Interface) is crucial for proper drive detection and system behavior.

1. **Access BIOS:** Power on your PC and repeatedly press the designated key (commonly `DEL`, `F2`, `F10`, or `F12`) during boot-up to enter the BIOS/UEFI settings.
2. **Enable AHCI:**
    - Navigate to the "Storage Configuration," "SATA Configuration," or similar section.
    - Ensure the SATA Mode is set to **AHCI** (Advanced Host Controller Interface). **Do NOT use IDE or RAID mode** (unless you specifically know what you're doing with motherboard RAID, which is generally not recommended for TrueNAS/Unraid/ZFS). AHCI allows the OS to directly manage the drives.
3. **Configure Boot Order:**
    - Go to the "Boot Options" or "Boot Priority" section.
    - Set your USB drive (which will contain the OS installer) as the primary boot device.
    - Later, after OS installation, you'll change this to your OS SSD.
4. **Disable Unused Peripherals (Optional, for Power Efficiency):**
    - In sections like "Integrated Peripherals," "Onboard Devices," or "Advanced," consider disabling components you won't use:
        - Onboard Audio (if you're not using it)
        - Unused USB controllers
        - Serial/Parallel ports
    - This can slightly reduce idle power consumption.
5. **Enable Virtualization (Optional, for Apps/VMs):**
    - If you plan to run VMs or certain Docker applications, ensure `Intel VT-x` (for Intel CPUs) or `AMD-V` (for AMD CPUs) is enabled. This is usually found in "CPU Configuration" or "Advanced" settings.
6. **Save and Exit:** Save your changes and exit the BIOS. Your PC should now attempt to boot from the USB installer.

> ⚠️ **Warning:** Changing SATA mode from IDE/RAID to AHCI after an OS is installed on that drive can cause boot issues. Ensure AHCI is set before OS installation.

Installing Your Chosen NAS OS

This section provides a high-level overview. Always refer to the official documentation for the most up-to-date and detailed installation steps.

General Steps for All OSes

1. **Download ISO:** Download the latest stable ISO image for your chosen OS.
    - **TrueNAS SCALE:** truenas.com/download-truenas-scale
    - **Unraid:** unraid.net/download (requires creating an account)
    - **Ubuntu Server LTS:** ubuntu.com/download/server
2. **Create Bootable USB:** Use a tool like Rufus (Windows) or Etcher (Windows/macOS/Linux) to write the ISO image to your 8GB+ USB drive.

TrueNAS SCALE Installation

1. **Boot from USB:** Insert the bootable USB into your gaming PC and power it on. It should boot into the TrueNAS SCALE installer.
2. **Follow On-Screen Prompts:**
    - Select `Install/Upgrade`.
    - Choose your **OS SSD** as the installation target. **Crucially, do NOT select any of your large storage HDDs.**
    - Set a strong root password.
    - Confirm the installation.
3. **Reboot:** Once the installation is complete, remove the USB drive and reboot. Your system should now boot into TrueNAS SCALE.

Verification: After reboot, you should see a console screen displaying local IP addresses. Look for an output similar to:

```
TrueNAS SCALE
Web UI: http://192.168.1.100 (example IP)
```

Access this IP from another computer on your network to continue configuration via the web UI.

Unraid Installation

1. **Prepare USB (Special Tool):** Unraid uses a specific USB creator tool. Download it from the Unraid website, run it, select your USB drive, and write the Unraid OS to it.
2. **Boot from USB:** Insert the Unraid USB into your PC and power it on. Select `Unraid OS (GUI Mode)` or `Unraid OS (Text Mode)` from the boot menu.
3. **Access Web UI:** Once Unraid boots (it runs entirely from the USB drive), it will display an IP address on the console. Look for an output similar to:

```
    My IP address is 192.168.1.101 (example IP)

```

```
Navigate to this IP in your web browser from another machine to access the
Unraid web UI.
```

Verification: You should be greeted by the Unraid web interface login page. The default username is `root` with no password initially.

Linux (Ubuntu Server LTS) Installation

1. **Boot from USB:** Insert the bootable Ubuntu Server USB and power on.
2. **Follow Installer:**
    - Choose your language.
    - Select `Install Ubuntu Server`.
    - Configure network (DHCP is fine for now, we'll set static later).
    - For storage, choose `Custom storage layout`. Select your **OS SSD** (e.g., `/dev/sda`) and configure it for the OS installation (e.g., ext4 filesystem, mount point `/`). **Do NOT touch your large HDDs at this stage.**
    - Create a user account (this will be your primary non-root user).
    - Choose to install `OpenSSH server` for remote access.
    - Complete the installation and reboot when prompted, removing the USB drive.

Verification: After reboot, you should see a login prompt on the console. Log in with the user you created. To verify network connectivity:

```
ip a
```

Expected output will show your network interfaces and their assigned IP addresses.

Initial Configuration: Network, Users, and Basic Settings

Regardless of your chosen OS, some initial setup steps are universal for a robust homelab NAS.

Setting a Static IP Address

A static IP ensures your NAS is always reachable at the same address, making it easier for clients to connect and for you to manage.

Where to find it (on your router):

1. Log into your home router's administration interface (usually `192.168.0.1`, `192.168.1.1`, or `192.168.X.1`).
2. Look for sections like "DHCP Reservation," "Static Leases," or "Address Reservation."
3. Find your NAS's current IP address and its MAC address. Most routers allow you to "reserve" the current IP for that MAC address. This is the simplest method as the NAS itself can remain on DHCP.

What to change (on the NAS, if not using DHCP reservation):

TrueNAS SCALE (Web UI)

1. Log into the TrueNAS SCALE web UI.
2. Navigate to `Network` -> `Global Configuration`.
3. You can set the hostname and DNS servers here.
4. Go to `Network` -> `Interfaces`.
5. Edit your primary network interface (e.g., `enpXsX`).
6. Uncheck `DHCP` and manually enter `IPv4 Address`, `Netmask`, and `Default Gateway`.
7. Click `Apply` and `Test Changes`.

Unraid (Web UI)

1. Log into the Unraid web UI.

2. Navigate to `Settings` -> `Network Settings`.
3. Change `Bonding mode` to `Active-Backup` or `None` if you only have one NIC.
4. Set `IPv4 Protocol` to `Static`.
5. Enter `IPv4 Address`, `Netmask`, `Gateway`, and `DNS Servers`.
6. Click `Apply`.

Linux (Ubuntu Server LTS - `netplan`)

1. Identify your network interface name (e.g., `enp0s3`) using `ip a`.
2. Edit the `netplan` configuration file. It's usually in `/etc/netplan/`.

```
sudo nano /etc/netplan/00-installer-config.yaml
```

1. Modify the file to look like this (replace with your interface name and desired IP details):

```
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3: # Replace with your actual interface name
      dhcp4: no
      addresses: [192.168.1.100/24] # Your desired static IP and subnet
mask
      routes:
        - to: default
          via: 192.168.1.1 # Your router's IP
      nameservers:
        addresses: [192.168.1.1, 8.8.8.8] # Your router and a public DNS
```

1. Apply the changes:

```
sudo netplan apply
```

Verification (all OSes): From your NAS console or via SSH, ping your router and a public website.

```
ping 192.168.1.1 # Replace with your router's IP
ping google.com
```

Both should return successful replies.

Creating Users

It's best practice to create specific users for accessing shares rather than using `root`.

TrueNAS SCALE (Web UI)

1. Navigate to `Credentials` -> `Local Users`.
2. Click `Add`.
3. Enter `Username`, `Full Name`, and a strong `Password`.
4. Ensure `New Primary Group` is checked, and `Home Directory` is `/nonexistent` unless you want a home directory.
5. Click `Save`.

Unraid (Web UI)

1. Navigate to `Users` tab.
2. Click `Add User`.
3. Enter `Username` and a strong `Password`.
4. Click `Add`.

Linux (CLI)

```
sudo adduser mynasuser
sudo usermod -aG sudo mynasuser # Optional: Add to sudo group if needed for administration
```

Follow the prompts to set a password and user information.

Enabling and Securing SSH

SSH (Secure Shell) allows you to remotely access your NAS's command line.

TrueNAS SCALE (Web UI)

1. Navigate to `System Settings` -> `Services`.
2. Find the `SSH` service and toggle its `Running` switch to `On`.
3. Ensure `Start Automatically` is also enabled.
4. Click the `pencil icon` next to SSH to configure it.
    - It's recommended to disable `Login as Root with Password` and enable `Allow Password Authentication` only for specific users, or better yet, use `Allow PKCS11Provider` for key-based authentication.

Unraid (Web UI)

1. Navigate to `Settings` -> `SSH`.
2. Set `Enable SSH` to `Yes`.
3. Optionally, change the `SSH port` from the default 22.
4. Click `Apply`.

Linux (CLI)

OpenSSH server should have been installed during Ubuntu Server setup.

1. Ensure the service is running:

```
    sudo systemctl status ssh
```

```
If not running, start it:
```

```
    sudo systemctl enable ssh --now
```

1. **Basic Firewall (UFW):** Allow SSH access.

```
    sudo ufw allow ssh
    sudo ufw enable
    sudo ufw status
```

```
The status should show `SSH` allowed.
```

Verification (all OSes): From your client machine (Windows, macOS, Linux terminal):

```
ssh mynasuser@192.168.1.100 # Replace with your username and NAS IP
```

You should be prompted for a password and then gain shell access to your NAS.

Storage Pool/Array Creation and Data Redundancy Setup

This is where your raw drives become usable, redundant storage.

TrueNAS SCALE: ZFS Pool Creation (UI Steps)

TrueNAS uses ZFS pools, which are collections of virtual devices (vdevs) made from your physical drives. RAID-Z levels provide redundancy.

1. **Log in to TrueNAS SCALE Web UI.**
2. Navigate to `Storage` -> `Pools`.
3. Click `Add` -> `Create New Pool`.
4. Enter a `Pool Name` (e.g., `data_pool`).
5. Under `Data VDEVs`, click `Add VDEV`.
6. Select your desired HDDs from the `Available Disks` list. **Do NOT select your OS boot drive.**
7. Choose your `VDEV Type`:
    - **Stripe:** No redundancy (not recommended).
    - **Mirror:** 2 drives for 1 drive's capacity, 1 drive failure tolerance.
    - **RAIDZ1:** Minimum 3 drives, 1 drive failure tolerance.
    - **RAIDZ2:** Minimum 4 drives, 2 drive failure tolerance.
    - **RAIDZ3:** Minimum 5 drives, 3 drive failure tolerance.

    > 💡 **Tip:** For a typical homelab with 2-4 drives, RAIDZ1 is a good balance. For 5+ drives or critical data, RAIDZ2 is highly recommended.

8. Adjust `Disk Size Warning` if you're mixing sizes (though ZFS performs best with same-sized drives in a vdev).
9. Review the `Layout` and `Capacity`.
10. Check `Confirm` and click `Create Pool`.
11. Confirm the warning about data erasure.

Verification: After creation, the `Pools` dashboard should show your new pool, its capacity, and health status.

Unraid: Array Setup and Parity Drive Assignment (UI Steps)

Unraid uses a unique array where a dedicated parity drive protects other data drives.

1. **Log in to Unraid Web UI.**
2. Navigate to the `Main` tab.

3. Under `Array Devices`, you will see slots for `Parity` and `Data disks`.
4. **Assign Parity Drive:**
    - Click the dropdown next to `Parity-1`.
    - Select your largest HDD to be the parity drive. **This drive must be equal to or larger than any other data drive.**
5. **Assign Data Drives:**
    - Click the dropdowns next to `Disk 1`, `Disk 2`, etc.
    - Select your remaining HDDs as data drives.
6. **Start Array:**
    - Check `I understand that all data on the following devices will be OVERWRITTEN when the array is Started:`
    - Click `Start` to begin the array initialization.
    - The parity sync process will begin. This can take many hours depending on drive size. You can use the array during this time, but performance might be reduced, and redundancy is not fully active until the sync completes.

Verification: The `Main` tab will show the status of the array, including the parity sync progress. All drives should show as `Online`.

Linux with ZFS: `zpool create`, `zfs create`

For Linux, you'll use the command line to create ZFS pools and datasets.

1. **Install OpenZFS:**

```
sudo apt update
sudo apt install zfsutils-linux
```

1. **Identify Drives:** Use `lsblk -f` or `sudo fdisk -l` to identify your storage HDDs (e.g., `/dev/sdb`, `/dev/sdc`, `/dev/sdd`). **Double-check these names carefully!**

> ⚠️ **Warning:** Using the wrong device name will wipe the wrong drive. Use `/dev/disk/by-id/` paths for maximum safety.

2. **Create ZFS Pool (e.g., RAIDZ1 with three drives):**

```
    sudo zpool create -f data_pool raidz1 /dev/disk/by-id/ata-
WDC_WD40EZRZ-00WN9B0_WD-WCC7K4HXXXXX /dev/disk/by-id/ata-
WDC_WD40EZRZ-00WN9B0_WD-WCC7K4HXXXXY /dev/disk/by-id/ata-
WDC_WD40EZRZ-00WN9B0_WD-WCC7K4HXXXXZ
```

```
- Replace `data_pool` with your desired pool name.
- Replace `raidz1` with `mirror` or `raidz2` as appropriate.
- Replace `/dev/disk/by-id/...` with the actual `by-id` paths for your drives.
```

1. **Create ZFS Datasets (Filesystems):**

```
    sudo zfs create data_pool/media
    sudo zfs create data_pool/backups
    sudo zfs create data_pool/documents
```

```
These datasets are automatically mounted under `/data_pool/media`, `/
data_pool/backups`, etc.
```

1. **Set Permissions (Optional, but recommended for sharing):**

```
    sudo chown -R mynasuser:mynasuser /data_pool/media
    sudo chmod -R 770 /data_pool/media
```

Verification:

```
sudo zpool status data_pool
sudo zfs list
```

The `zpool status` output should show your pool in `ONLINE` state with no errors. `zfs list` will show your datasets.

Creating Network Shares (SMB/NFS) and Access Permissions

Now that you have storage, let's make it accessible over the network. SMB (Server Message Block) is for Windows and macOS clients, while NFS (Network File System) is typically for Linux/Unix clients.

TrueNAS SCALE: SMB Share Configuration (UI Steps)

1. **Log in to TrueNAS SCALE Web UI.**
2. Navigate to `Shares` -> `SMB Shares`.
3. Click `Add`.
4. **Path:** Browse to the dataset you want to share (e.g., `/mnt/data_pool/media`).
5. **Name:** Enter a descriptive name for the share (e.g., `Media`).
6. **Description (Optional):** Add a brief description.
7. **Purpose (Optional):** Select a purpose (e.g., `Default Share Parameters`).
8. **Advanced Options (Optional):**
    - `Allow Guest Access`: Generally **discouraged** for security.
    - `Apply Permissions Recursively`: Often useful for media shares.
    - `ACL Mode`: `RESTRICTED` is usually fine.
9. **Permissions:** Click `Save` and then `Edit ACL` for the newly created share.
    - Set `Owner` and `Owner Group` to the user/group you created earlier (e.g., `mynasuser`).
    - Set `Permissions Type` to `OPEN`.
    - Set `Default Permissions` to `rwx` for owner and group, `---` for others.
    - Check `Apply permissions recursively`.
    - Click `Save Access Control List`.
10. **Enable SMB Service:** Ensure the `SMB` service is running under `System Settings` -> `Services`.

Verification: From a Windows PC, open File Explorer and type `\\192.168.1.100\Media` (replace with your NAS IP and share name) into the address bar. You should be prompted for credentials and then see the share contents.

Unraid: Creating Shares (UI Steps)

1. **Log in to Unraid Web UI.**
2. Navigate to the `Shares` tab.
3. Click `Add Share`.

4. **Share Name:** Enter a name (e.g., `Media`).
5. **Comments (Optional):** Add a description.
6. **Allocation Method:** `High water` is common.
7. **Minimum free space:** Leave default or set if needed.
8. **Included/Excluded Disks:** Select which disks this share can use.
9. **SMB Security:**
    - `Export`: `Yes`.
    - `Security`: `Private` (requires user authentication).
    - `SMB User Access`: Assign read/write or read-only access to your created users.
10. Click `Add Share`.

Verification: From a Windows PC, open File Explorer and type `\\192.168.1.101\Media` (replace with your NAS IP and share name) into the address bar. You should be prompted for credentials and then see the share contents.

Linux with ZFS: Samba Configuration (`/etc/samba/smb.conf`)

1. **Install Samba:**

```
sudo apt update
sudo apt install samba samba-common-bin
```

1. **Backup original config:**

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

1. **Edit `smb.conf`:**

```
sudo nano /etc/samba/smb.conf
```

1. **Add your share definition at the end of the file:**

```
[global]
   workgroup = WORKGROUP
   server string = %h server (Samba, Ubuntu)
   log file = /var/log/samba/log.%m
```

```
        max log size = 1000
        logging = file
        panic action = /usr/share/samba/panic-action %d
        server role = standalone server
        obey pam restrictions = yes
        unix password sync = yes
        passwd program = /usr/bin/passwd %u
        passwd chat = *Enter\snew\s*\spassword:* %n\n
*Retype\snew\s*\spassword:* %n\n *password\supdated\ssuccessfully* .
        pam password change = yes
        map to guest = bad user
        dns proxy = no
        usershare allow guests = no
        # This is important for ZFS shares to allow proper permissions
        vfs objects = acl_xattr
        map acl inherit = yes
        store dos attributes = yes

    [Media]
        path = /data_pool/media # Your ZFS dataset mount point
        read only = no
        browsable = yes
        guest ok = no
        valid users = mynasuser # User you created earlier
        create mask = 0664
        directory mask = 0775
```

1. **Create Samba User:** The Linux user needs a Samba password.

```
    sudo smbpasswd -a mynasuser
```

```
Enter a password for the Samba share (can be different from Linux login).
```

1. **Restart Samba:**

```
    sudo systemctl restart smbd nmbd
```

Verification: From a Windows PC, open File Explorer and type `\\192.168.1.100\Media` (replace with your NAS IP and share name) into the address bar. You should be prompted for credentials and then see the share contents.

Optimizing for Power Efficiency: BIOS, OS Settings, and Drive Spin-Down

Gaming PCs are not built for low power consumption. While you won't match a purpose-built NAS appliance, you can significantly reduce power draw.

> ⚠️ **Warning:** Frequent drive spin-up/spin-down cycles can reduce HDD lifespan. Balance power savings with drive health. It's often better to let drives spin down after several hours of inactivity, rather than minutes.

BIOS Settings for Power Savings

Revisit your BIOS/UEFI settings:

1. **C-States/EIST:** Ensure CPU power-saving features like `Intel SpeedStep Technology (EIST)` and `C-States` (C1E, C3, C6, C7) are enabled. These allow the CPU to enter lower power states when idle.
2. **Disable Unused Hardware:** As mentioned earlier, disable onboard audio, unused USB controllers, COM/LPT ports, and remove the dedicated GPU if possible.
3. **PCIe Power Management:** Look for `PCIe ASPM (Active State Power Management)` and enable it if available.
4. **Wake-on-LAN (WoL):** If you don't use WoL, disable it in the BIOS and your network card settings, as it keeps the NIC partially powered.

OS-Level Power Management

TrueNAS SCALE (Web UI)

1. Navigate to `System Settings` -> `Advanced`.
2. Look for `Power Management` options. TrueNAS typically handles CPU power states automatically.
3. **HDD Standby:** Navigate to `Storage` -> `Disks`.
    - For each HDD, click the `pencil icon` to edit.
    - Set `HDD Standby` to a suitable value (e.g., 60 minutes or 120 minutes). This will spin down the drive after that period of inactivity.

Unraid (Web UI)

1. Navigate to `Settings` -> `Disk Settings`.

2. **Spin Down Delay:** Set `Default spin down delay` to your desired inactivity period (e.g., `120 minutes`).
3. You can also configure individual disk spin-down delays.

Linux (CLI - `hdparm`, `powertop`, `tlp`)

1. **Install `hdparm` and `powertop`:**

```
    sudo apt update
    sudo apt install hdparm powertop
```

1. **Drive Spin-Down with `hdparm`:** Identify your drives (e.g., `/dev/sdb`, `/dev/sdc`).

```
    sudo hdparm -S 120 /dev/sdb # Spin down after 10 minutes of inactivity
(120 * 5 seconds)
    sudo hdparm -S 240 /dev/sdc # Spin down after 20 minutes of inactivity
```

```
To make this persistent, you'd typically add these commands to a systemd
service or an appropriate startup script (e.g., `/etc/hdparm.conf` if present,
or `/etc/rc.local` for older systems).
> 💡 **Tip:** A value of `120` in `hdparm -S` means 10 minutes (120 * 5
seconds). `240` is 20 minutes. `241-251` map to 30 minutes to 3.5 hours. `252`
is 21 minutes. `253` is a specific manufacturer-defined timeout. `254`
disables standby.
```

1. **CPU/System Power Management with `powertop`:** Run `powertop` to see power consumption and suggestions.

```
    sudo powertop
```

```
Press `Tab` to go to the `Tunables` tab. You can set recommended tunables to
`Good` to apply power savings.
To auto-tune on startup (use with caution, can cause issues with some
hardware):
```

```
    sudo powertop --auto-tune
```

```
For persistent auto-tuning, you might need to create a systemd service.
```

1. **Install `tlp` (Advanced Power Management):** `tlp` is a powerful tool for laptop power management but can also benefit desktops.

```
    sudo apt install tlp tlp-rdw
    sudo systemctl enable tlp --now
```

```
Edit `/etc/default/tlp` to configure specific settings.
```

Implementing Data Backup and Snapshot Strategies

Data redundancy (RAIDZ, Unraid parity) protects against drive failure, but it's not a backup. Backups protect against accidental deletion, ransomware, software bugs, and catastrophic events.

ZFS Snapshots (TrueNAS/Linux)

ZFS snapshots are incredibly powerful, creating read-only copies of a dataset at a specific point in time with minimal overhead.

TrueNAS SCALE (Web UI)

1. Navigate to `Storage` -> `Pools`.
2. Find your pool/dataset (e.g., `data_pool/media`).
3. Click the `three dots` next to the dataset and select `Add Periodic Snapshot Task`.
4. Configure the task:
    - `Dataset`: Select the target dataset.
    - `Recursive`: Check if you want to snapshot sub-datasets.
    - `Naming Schema`: Use default or customize (e.g., `auto-%Y%m%d%H%M`).
    - `Schedule`: Set frequency (e.g., `Every Day` at `02:00`).
    - `Lifetime`: How long to keep snapshots (e.g., `7 days`).
5. Click `Save`.

Verification: After the first scheduled snapshot, navigate to `Storage` -> `Snapshots`. You should see your snapshots listed.

Linux (CLI)

1. **Manual Snapshot:**

```
sudo zfs snapshot data_pool/media@$(date +%Y%m%d%H%M)
```

1. **List Snapshots:**

```
sudo zfs list -t snapshot
```

1. **Automatic Snapshots (e.g., with `zfs-auto-snapshot` or cron):** For a simple cron job for daily snapshots:

```
sudo nano /etc/cron.daily/zfs_snapshot
```

```
Add the following (make executable with `sudo chmod +x /etc/cron.daily/
zfs_snapshot`):
```

```
#!/bin/bash
/usr/bin/zfs snapshot -r data_pool@$(date +%Y%m%d%H%M)
/usr/bin/zfs destroy -r data_pool@$(date -d '7 days ago' +%Y%m%d%H%M) #
Delete snapshots older than 7 days
```

The 3-2-1 Backup Rule

- **3 copies of your data:** The original and at least two backups.
- **2 different media types:** E.g., internal NAS drives, external HDD, cloud storage.
- **1 offsite copy:** Protects against fire, theft, or local disaster.

Consider an external USB drive for local backups, or a cloud service like Backblaze B2 (with rclone) for offsite.

Optional: Installing and Configuring Applications (e.g., Plex, Docker)

This is where your NAS truly becomes a homelab server, extending beyond just file storage.

TrueNAS SCALE: Apps Catalog

TrueNAS SCALE leverages Kubernetes for containerized applications, accessible via a user-friendly UI.

1. **Log in to TrueNAS SCALE Web UI.**
2. Navigate to `Apps`.
3. Click `Available Applications`.
4. Search for your desired app (e.g., `Plex Media Server`, `Jellyfin`, `Nextcloud`, `Docker` [if you want a raw Docker environment]).
5. Click `Install` on the chosen app.
6. Follow the wizard, configuring settings like:
    - `Application Name`
    - `Storage` (map host paths to container paths for data persistence, e.g., `/mnt/data_pool/media` -> `/data`)
    - `Network` (port mappings, static IP if needed)
    - `Resources` (CPU/RAM limits)
7. Click `Install`.

Verification: After installation, the app will appear under `Installed Applications`. Its status should change to `Running`. You can then access the app via its configured IP and port (e.g., `<http://192.168.1.100:32400` > for Plex).

Unraid: Docker Containers

Unraid has excellent Docker integration, with a vast community applications plugin.

1. **Log in to Unraid Web UI.**
2. Navigate to `Apps`.
3. If it's your first time, install the `Community Applications` plugin.
4. Once installed, browse the `Apps` tab for applications.
5. Search for your desired app (e.g., `Plex`, `Jellyfin`, `Nextcloud`).
6. Click the `Download` icon next to the app.

7. Configure the template:
    - `Container Name`
    - `Host Path 1` (e.g., `/mnt/user/Media/` for your media files) mapped to `Container Path` (e.g., `/data`).
    - `Port mappings` (e.g., `32400` for Plex).
    - `Plex Claim Token` (for Plex setup).
8. Click `Apply`.

Verification: The container will appear under the `Docker` tab. Its status should be `Running`. Click on the container icon and select `WebUI` to access the application.

Linux: Docker and Docker Compose

For Linux, Docker is the standard for containerization. Docker Compose simplifies multi-container applications.

1. **Install Docker Engine:**

```
    sudo apt update
    sudo apt install apt-transport-https ca-certificates curl software-
properties-common
    curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --
dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
    echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/
keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /
dev/null
    sudo apt update
    sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-
plugin docker-compose-plugin
```

1. **Add user to `docker` group:**

```
    sudo usermod -aG docker mynasuser
    newgrp docker # Apply group change without logging out
```

1. **Install Docker Compose (v2):** It's now included with `docker-ce-cli`.

```
    docker compose version
```

1. **Example: Plex Media Server with Docker Compose:** Create a directory for your Plex configuration:

```
    mkdir -p ~/docker/plex
    cd ~/docker/plex
```

```
Create a `docker-compose.yml` file:
```

```
    # ~/docker/plex/docker-compose.yml
    version: "3.8"
    services:
      plex:
        image: lscr.io/linuxserver/plex:latest
        container_name: plex
        network_mode: host # Required for Plex's discovery and remote access
        environment:
          - PUID=1000 # Replace with your user ID (id -u mynasuser)
          - PGID=1000 # Replace with your group ID (id -g mynasuser)
          - VERSION=docker
          - TZ=Europe/London # Replace with your timezone
          - PLEX_CLAIM= # Optional: Get a claim token from https://
www.plex.tv/claim
        volumes:
          - ./config:/config
          - /data_pool/media/movies:/movies # Map your movie library
          - /data_pool/media/tvshows:/tvshows # Map your TV show library
         # Add more volume mappings for other media types as needed
        restart: unless-stopped
```

```
> 💡 **Tip:** Find your PUID/PGID with `id -u mynasuser` and `id -g mynasuser`
on your Linux NAS.
```

1. **Start Plex:**

```
    docker compose up -d
```

Verification:

```
docker ps
```

You should see the `plex` container running. Access Plex via `<http://192.168.1.100:32400/web>` (replace with your NAS IP).

Security Best Practices for Your Homelab NAS

Your NAS holds your valuable data, making security paramount.

1. **Strong, Unique Passwords:** Use complex passwords for all user accounts (admin, SSH, share users). Consider a password manager.
2. **SSH Key Authentication:** For remote SSH access, disable password authentication and use SSH keys.
    - Generate a key pair on your client machine (`ssh-keygen`).
    - Copy the public key to your NAS (`ssh-copy-id mynasuser@192.168.1.100`).
    - Disable password login in `/etc/ssh/sshd_config` on Linux (`PasswordAuthentication no`) or via the TrueNAS/Unraid UI.
3. **Firewall Configuration:**
    - **TrueNAS/Unraid:** Their built-in firewalls are configured via the UI. Ensure only necessary ports are open (e.g., 80/443 for web UI, 22 for SSH, 139/445 for SMB, 2049 for NFS, app-specific ports like 32400 for Plex).
    - **Linux (UFW):**

```
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow ssh          # Or sudo ufw allow 22/tcp
sudo ufw allow 139/tcp      # SMB
sudo ufw allow 445/tcp      # SMB
sudo ufw allow 2049/tcp     # NFS (if using)
sudo ufw allow 32400/tcp    # Plex (if using)
sudo ufw enable
sudo ufw status verbose
```

1. **Disable Root Login:** Never allow direct SSH login as `root`. Use a regular user and `sudo` for administrative tasks.
2. **Keep Software Updated:** Regularly apply OS, application, and firmware updates to patch security vulnerabilities.
3. **Network Segmentation (Advanced):** If your router supports VLANs, consider placing your NAS on a separate VLAN from your general home network to isolate it.

4. **Physical Security:** Keep your NAS in a secure location, away from unauthorized access.
5. **Regular Backups:** As discussed, backups are your last line of defense.

Maintenance, Monitoring, and Updates

A homelab isn't a "set it and forget it" system. Regular maintenance ensures reliability and longevity.

System Updates

- **TrueNAS SCALE (Web UI):**
  - Navigate to `System Settings` -> `Update`.
  - Check for updates and follow the prompts to apply them.
- **Unraid (Web UI):**
  - The `Main` tab will often show a notification if an update is available.
  - Navigate to `Tools` -> `Update OS`.
- **Linux (CLI):**

```
    sudo apt update
    sudo apt upgrade -y
    sudo apt autoremove -y
    sudo reboot # Reboot if kernel or critical system components were updated
```

```
For Docker containers:
```

```
    cd ~/docker/plex # Or your docker compose project directory
    docker compose pull
    docker compose up -d --remove-orphans
```

Drive Health Monitoring (SMART)

All modern HDDs include SMART (Self-Monitoring, Analysis, and Reporting Technology) data, which can predict impending drive failures.

- **TrueNAS SCALE (Web UI):**
  - Navigate to `Storage` -> `Disks`.
  - For each disk, click the `three dots` and select `SMART Test`. You can set up periodic tests here.
  - You can also configure email alerts for SMART errors under `System Settings` -> `Email` and `Alert Services`.
- **Unraid (Web UI):**
  - The `Main` tab shows a summary of drive health.
  - Click on individual disks to view detailed SMART attributes and run tests.
- **Linux (CLI - `smartmontools`):**

```
sudo apt install smartmontools
sudo smartctl -a /dev/sdb # Replace /dev/sdb with your drive
```

```
To set up automatic checks and email alerts, you'll need to configure `/etc/
smartd.conf`.
```

System Logs

Regularly check system logs for unusual activity or errors.

- **TrueNAS SCALE/Unraid (Web UI):** Both provide dedicated log viewers in their interfaces.
- **Linux (CLI):**

```
journalctl -xe # View systemd journal
tail -f /var/log/syslog # Real-time system log
dmesg # Kernel ring buffer messages
```

Backups

Verify your backup strategy regularly. Test restoring a file from a snapshot or backup to ensure it works.

Troubleshooting Common Issues

Even the most carefully built homelab can encounter issues. Here are some common problems and their solutions.

1. NAS Not Accessible on Network

Symptom: Cannot ping the NAS, web UI doesn't load, shares are unavailable.

Cause:

- Incorrect IP address.
- Network cable unplugged or faulty.
- Firewall blocking access.
- NAS OS didn't boot correctly.

Fix:

1. **Check Physical Connection:** Ensure the Ethernet cable is securely plugged into both the NAS and the router/switch. Check the link lights on the NIC.
2. **Verify IP Address:**
    - On the NAS console, log in and use `ip a` (Linux/TrueNAS) or check the Unraid console output to confirm the IP.
    - Ensure your client device is on the same network subnet.
3. **Check Firewall:**
    - **Linux:** `sudo ufw status`. If `inactive` or blocking, allow necessary ports (`sudo ufw allow 80`, `sudo ufw allow 443`, `sudo ufw allow 22`, `sudo ufw allow 139`, `sudo ufw allow 445`).
    - **TrueNAS/Unraid:** Temporarily disable the firewall in the UI if you suspect it's the culprit, then re-enable and configure properly.
4. **Reboot:** A simple reboot of the NAS might resolve temporary network glitches.

2. Cannot Access SMB/NFS Shares

Symptom: "Network path not found," "Access denied," or share contents are empty.

Cause:

- Incorrect share path or name.
- Incorrect user permissions.
- SMB/NFS service not running.
- Firewall blocking SMB/NFS ports.

Fix:

1. **Verify Share Path/Name:** Double-check the share name and path in your NAS OS configuration.
2. **Check Service Status:**
    - **TrueNAS:** `System Settings` -> `Services`. Ensure `SMB` (and `NFS` if used) is running.
    - **Unraid:** `Settings` -> `SMB`. Ensure it's enabled.
    - **Linux:** `sudo systemctl status smbd nmbd` (for SMB), `sudo systemctl status nfs-kernel-server` (for NFS). Restart if needed.
3. **Permissions:**
    - **TrueNAS:** `Shares` -> `SMB Shares` -> `Edit ACL` for the share. Ensure the accessing user has appropriate read/write permissions.
    - **Unraid:** `Shares` -> `[Share Name]` -> `SMB Security`. Verify user access.
    - **Linux:** Check file system permissions on the shared directory (`ls -ld /data_pool/media`). Ensure the Linux user associated with the Samba user has correct permissions. Also, verify `smbpasswd -a <user>` was run.
4. **Firewall:** Ensure ports 139/445 (SMB) or 2049 (NFS) are open on the NAS firewall.

3. Drive Failure / Pool Degradation

Symptom: NAS UI reports a degraded pool, a drive shows as "offline" or "faulted," or SMART alerts.

Cause:

- Physical drive failure.
- Bad SATA cable or power connection.

Fix:

1. **Identify Failed Drive:**
    - **TrueNAS/Unraid:** The UI will clearly indicate which drive has failed.
    - **Linux:** `sudo zpool status <pool_name>` will show the faulted drive. `sudo smartctl -a /dev/sdX` may also show errors.
2. **Check Physical Connections:** Power down the NAS, open the case, and reseat the SATA data and power cables for the problematic drive. Power on and recheck. If it's still faulted, the drive itself is likely bad.
3. **Replace Drive:**
    - **TrueNAS:** Power down, replace the physical drive, power on. In the UI, navigate to `Storage` -> `Pools`, click the `three dots` next to the degraded vdev, and select `Replace`. Choose the new drive.
    - **Unraid:** Power down, replace the physical drive, power on. In the `Main` tab, assign the new drive to the failed slot and start the array. Unraid will automatically rebuild the data onto the new drive using parity.
    - **Linux (ZFS):** Power down, replace the physical drive, power on. Identify the new drive's `by-id` path.

```
sudo zpool replace <pool_name> <old_drive_id> <new_drive_id>
```

```
(e.g., `sudo zpool replace data_pool ata-WDC_OLD_SERIAL ata-WDC_NEW_SERIAL`).
```

1. **Monitor Rebuild:** The rebuild process can take many hours. Monitor the pool status (`zpool status` or UI) until it's healthy.

4. High Power Consumption

Symptom: Electricity bill is higher than expected, or a power meter shows high idle wattage.

Cause:

- Dedicated GPU still installed.
- CPU C-states/EIST not enabled in BIOS.
- Drives not spinning down.
- High base power draw of gaming motherboard/CPU.

Fix:

1. **Remove Dedicated GPU:** If your CPU has integrated graphics and you don't need the dedicated GPU for transcoding, remove it. This is often the biggest power saver.
2. **BIOS Settings:** Revisit BIOS for C-states, EIST, and PCIe ASPM.
3. **Drive Spin-Down:** Configure drive spin-down in your OS (TrueNAS/Unraid UI, `hdparm` for Linux).
4. **Consider Hardware Upgrade (Long-term):** If power consumption remains unacceptably high, the underlying gaming hardware might simply be too inefficient. Consider a low-power CPU (e.g., Intel N-series, AMD Ryzen APUs) and a more efficient motherboard for a dedicated NAS build in the future.

```
# Example error message for SMB access denied on Linux
# Client output:
#   mount error(13): Permission denied
#   Refer to the mount.cifs(8) manual page (e.g. man mount.cifs) and kernel
log messages (dmesg)
#
# Server /var/log/samba/log.smbd output:
#   [2026/06/03 14:30:05.123456,  0] ../../source3/smbd/
service.c:1070(create_connection_session_info)
#     create_connection_session_info: guest user (from client 192.168.1.5) not
allowed to access share Media (user mynasuser)
#
# Cause: The client tried to connect as a guest or with incorrect credentials,
and the share is configured for specific users.
# Fix: Ensure the client is providing the correct username and password
(mynasuser and its smbpasswd).
#      Alternatively, if guest access is desired (less secure), enable 'guest
ok = yes' in smb.conf for the share.
```

This guide should give you a solid foundation for transforming your old gaming PC into a powerful and versatile homelab NAS. Remember, the journey of self-hosting is about continuous learning and iteration. Enjoy your new data freedom!

Community Resources:

- r/homelab
- r/truenas
- r/unRAID
- TrueNAS Community Forums