

# Complete Raspberry Pi 5 NAS Setup

A comprehensive guide for building a production-ready NAS using Raspberry Pi 5, complete with media streaming, photo management, and automated backups.

## Hardware Setup

### Components Used

- **Raspberry Pi 5** (8GB RAM recommended)
- **Radaxa Penta HAT** (5-bay SATA expansion)
- **4x 2TB TeamGroup SSDs** (SATA)
- **M.2 NVMe SSD** (for boot drive via USB adapter)
- **Cat 8 Ethernet cables** (gigabit networking)

### Storage Configuration

- **Boot Drive:** M.2 NVMe SSD (external USB adapter)
- **Data Storage:** 4x 2TB SSDs via Radaxa Penta HAT
- **Protection:** SnapRAID parity + content files
- **Total Usable:** ~6TB with 1x parity protection

## Performance Results

### Network Performance

- **Local throughput:** 38.9 Gbps (loopback)
- **Internet:** 1GB fiber connection
- **Streaming:** Supports multiple 4K streams

### Storage Performance

- **Individual SSD speed:** 486-520 MB/sec per drive
- **Cached reads:** 4.6-5.7 GB/sec
- **4K streaming:** Smooth, no buffering issues



## Software Stack

### Base System

- **OS:** Raspberry Pi OS (Debian 12 Bookworm)
- **NAS OS:** OpenMediaVault 7 (OMV)
- **Containerization:** Docker + Portainer
- **Storage Protection:** SnapRAID

### Applications

- **Media Server:** Plex Media Server
- **Photo Management:** Immich
- **Backup:** rclone to Google Drive
- **Monitoring:** Built-in OMV tools



## Installation Guide

### 1. Initial Setup

#### Boot from MicroSD First

```
bash
```

```
# Flash Raspberry Pi OS to microSD
```

```
# Boot and complete initial setup
```

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

#### Install OpenMediaVault

```
bash
```

```
wget -O - https://github.com/OpenMediaVault-Plugin-Developers/installScript/raw/master/install | s
```



## Setup SSH Keys (Recommended)

```
bash
```

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

```
# Copy public key to authorized_keys
```

## 2. Hardware Configuration

### Enable Radaxa Penta HAT

```
bash
```

```
# Add to /boot/firmware/config.txt
```

```
dtparam=pciex1
```

```
dtoverlay=pcie-32bit-dma
```

### Verify SATA Detection

```
bash
```

```
lsblk
```

```
# Should show sda, sdb, sdc, sde (your SSDs)
```

## 3. Storage Setup

### Configure SnapRAID via OMV

1. Install SnapRAID plugin in OMV

2. Configure drives:

- 3x data drives (movies, music, photos)

- 1x parity drive

3. Set content files on data drives

4. Schedule weekly scrub operations

## Format and Mount Drives

```
bash
```

```
# Done through OMV Web Interface
```

```
# Services > SnapRAID > Drives
```

## 4. Docker Installation

### Install Docker

```
bash
```

```
curl -fsSL https://get.docker.com -o get-docker.sh
```

```
sudo sh get-docker.sh
```

```
sudo usermod -aG docker $USER
```

### Install Portainer

```
bash
```

```
docker volume create portainer_data
```

```
docker run -d -p 8000:8000 -p 9000:9000 \
```

```
  --name=portainer --restart=always \
```

```
  -v /var/run/docker.sock:/var/run/docker.sock \
```

```
  -v portainer_data:/data \
```

```
  portainer/portainer-ce
```

## 5. Application Setup

### Plex Media Server

yaml

```
# docker-compose.yml
version: "3.8"
services:
  plex:
    image: lscr.io/linuxserver/plex:latest
    container_name: plex
    network_mode: host
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Denver
      - VERSION=docker
    volumes:
      - /path/to/plex/config:/config
      - /srv/dev-disk-by-uuid-xxx/movies:/movies
      - /srv/dev-disk-by-uuid-xxx/tvseries:/tv
      - /srv/dev-disk-by-uuid-xxx/music:/music
    restart: unless-stopped
```

### Immich Photo Management

yaml

```
# Use official Immich docker-compose.yml
# Configure storage paths to your photo drives
```

## 6. Clone to NVMe Boot Drive

### Prepare NVMe Drive

bash

*# Connect M.2 NVMe via USB adapter*  
*# Use rpi-clone to copy SD card to NVMe*  
`sudo rpi-clone sda` *# assuming sda is your NVMe*

## Update Boot Configuration

bash

*# Modify /boot/firmware/cmdline.txt*  
*# Change root=PARTUUID=xxx to point to NVMe*

## Test Boot from NVMe

bash

*# Remove SD card and reboot*  
*# Verify faster boot times and performance*

## Optimization & Troubleshooting

### Plex Performance Optimization

#### Scheduled Tasks Configuration

- **Maintenance Window:** 4:00 AM - 6:00 AM
- **Disable:** Extensive media analysis during maintenance
- **Disable:** Chapter thumbnail generation during maintenance
- **Keep:** Database optimization, cache cleanup

### Hardware Transcoding

bash

*# Enable in Plex Settings > Transcoder*  
*# "Use hardware acceleration when available"*

## Network Optimization

### Bandwidth Limiting for Backups

bash

*# Use `--bwlimit` in rclone commands*  
rclone **sync** /source gdrive:dest `--bwlimit 50M`

## Common Issues & Solutions

### "Insufficient Bandwidth" Streaming Errors

**Cause:** Background tasks competing with streaming **Solution:**

- Move Plex maintenance to off-hours (4-6 AM)
- Disable intensive thumbnail generation
- Limit backup bandwidth

### Slow Boot Times

**Cause:** Booting from microSD **Solution:** Clone to NVMe SSD boot drive

### Storage Access Issues

**Cause:** Incorrect permissions or mount points **Solution:** Use OMV web interface for proper configuration

## Project Structure

```
pi5-nas-setup/
├── README.md
├── docs/
│   ├── hardware-setup.md
│   ├── software-installation.md
│   └── troubleshooting.md
├── scripts/
│   ├── backup/
│   │   ├── nas_backup.sh
│   │   └── rclone.conf.example
│   ├── installation/
│   │   ├── install_docker.sh
│   │   └── setup_omv.sh
│   └── maintenance/
│       └── system_health.sh
├── docker/
│   ├── plex/
│   │   └── docker-compose.yml
│   ├── immich/
│   │   └── docker-compose.yml
│   └── portainer/
│       └── docker-compose.yml
└── configs/
    ├── snapraid.conf.example
    └── omv-settings.json.example
```

## Automated Backup Solution

### Features

- **Automated daily backups** to Google Drive
- **Bandwidth limiting** to prevent network saturation
- **Comprehensive logging** with timestamps



- **Multiple backup targets** (photos, configs, docker)

## Setup Instructions

1. Configure rclone with Google Drive
2. Copy backup script to `/home/pi/scripts/`
3. Setup cron job for 4 AM daily execution
4. Test backup and monitor logs



## Monitoring & Maintenance

### System Health Checks

- **SnapRAID scrub**: Weekly integrity checks
- **Docker containers**: Auto-restart policies
- **Storage usage**: OMV dashboard monitoring
- **Network performance**: Regular speed tests

### Automated Tasks Schedule

- **4:00 AM**: Backup scripts start
- **4:00-6:00 AM**: Plex maintenance window
- **Weekly**: SnapRAID scrub operations
- **Monthly**: System updates and reboots



## Results & Benefits

### Performance Achievements

- **4K streaming**: Multiple simultaneous streams
- **Photo access**: Fast Immich performance
- **Backup speed**: 50MB/sec sustained uploads

- **Boot time:** <30 seconds from NVMe

## Reliability Features

- **Data protection:** SnapRAID parity + multiple content files
- **Automatic recovery:** Container restart policies
- **Remote backup:** Complete Google Drive sync
- **Monitoring:** Comprehensive logging and alerts

## Contributing

Contributions welcome! Please:

1. Fork the repository
2. Create a feature branch
3. Submit a pull request
4. Include detailed testing results

## License

MIT License - See LICENSE file for details

## Acknowledgments

- OpenMediaVault team for excellent NAS software
- Radaxa for the Pi 5 compatible SATA HAT
- Plex and Immich communities for media software
- Raspberry Pi Foundation for the amazing Pi 5

---

**Total Project Cost:** ~\$400-500 USD

**Setup Time:** 4-6 hours

**Skill Level:** Intermediate

**Maintenance:** Minimal (automated)