

# Server OS

Will discuss various server operating systems, distros and anything pretending to be an OS.

- [File Server](#)
  - [Introduction](#)
  - [OMV Setup](#)
  - [OMV Install Notes](#)
- [Ubuntu Server](#)
- [Proxmox](#)
- [Linux Mint Debian Edition](#)
- [Windows](#)
- [CasaOS, ZimaOS, Cosmos Cloud OS, RunTipi, etc.](#)

# File Server

# Introduction

So, you can use nearly any OS to host a file server. However, some are purpose built for the task, some do it as a side job and some are just a bare bones server designed to host anything.

I'm going to start with [Open Media Vault](#) because it's really simple, basic, free and completely focused on making it easy to host files. In spite of this, it doesn't actually prevent you from hosting whatever else you want on it too.

Pros:

- Easy to use
- Focused
- Few limitations
- Built on Debian
- Free

Cons:

Not really meant for anything other than file sharing

Summary:

It's a web UI built on Debian with a set of pre-installed tools and plugin expansion focused on letting you host files. Notably, there's also a plugin for running Docker containers. Thanks to the Debian base and the ability to SSH in, you can actually host anything you want on it, so long as you know how to install/run it from command line. If you want to get up and running with a file server at minimal cost (free) and minimal effort, without any real restrictions on what it can do, this is a good option. The community support is decent but there aren't many staff working on it, so responses may be slow.

Now, I'm going to list [Ubuntu Server](#), to give a baseline comparison.

Pros:

- Optimized Server OS that can run anything.
- Community Support
- Free

Cons:

Aside from some pre-installed apps that make remote management easier, it does absolutely nothing to hold your hand.

Summary:

If you want to roll a server from scratch, this is pretty much the de-facto starting point. You can use other distros but this one has the best community support and only the very minimum of apps needed to get started. If you have a decent amount of Linux experience and don't care about hand holding, this is actually a good option. The vast majority of guides are either made for it or compatible with it. As with most of my recommendations, it is based on Debian.

[Proxmox](#) can host a file server VM or can act as one directly but it's going to come with a lot of restrictions and will generally make you work a lot harder in most scenarios. Built on Debian LTS.

Pros:

- Easy to spin up VMs
- Easily share resources across PCs
- Good for setting up clusters
- Free\*

Cons:

- File sharing is more difficult and limited
- Not great for individual servers

Summary:

It can host a file server VM but it's not ideal for running a file server. It's really made for running in a cluster and I wouldn't really recommend it for any other situation. However, it can function individually.

The free version restricts you a little but the most annoying restriction is that you get treated like a peasant by being denied access to the "premium" repos. Repos are the servers you download apps and app updates from.

[TrueNAS/FreeNAS](#) is actually a NAS OS but very different from OMV. Built on FreeBSD.

Pros:

- Easy to setup file sharing
- Scalable
- Mostly Free

Cons:

No mergerfs or snapraid support  
Limited support for USB drives

If you've got the desire to have scalable storage, this is probably the best free option. People running Proxmox tend to set it up in a VM. The software might be free but you're going to spend extra on the hardware because you're going to need something that supports hardware raid. There's not much in the way of software raid for this OS. Don't even bother trying to use it for something other than file sharing.

[unRAID](#) is a paid solution with software raid features.

Pros:

Easy to use  
Scalable

Cons:

Expensive  
Limited use outside of file sharing

Summary:

There are plenty of people that like it but I find it too expensive and too limiting in what it can do. If you really want to pay to have your hand held for one of the most basic server tasks in the existence of servers, this isn't a bad option.

[HexOS](#) is an overpriced alternative to unRAID built on TrueNAS. I do not recommend. If you want to pay to have your hand held, go with unRAID.

File Server

# OMV Setup

### Make proper instructions later

# OMV Install Notes

## Further reading:

List of (mostly outdated) video guides - <https://forum.openmediavault.org/index.php?thread/23005-installation-and-setup-videos-beginning-intermediate-and-advanced/&postID=175857#post175857>

Written guides - <https://forum.openmediavault.org/index.php?thread/27073-new-user-guide-getting-started-with-openmediavault-6-5-4/>

Intel Video compatibility -

[https://en.wikipedia.org/wiki/Intel\\_Quick\\_Sync\\_Video#Hardware\\_decoding\\_and\\_encoding](https://en.wikipedia.org/wiki/Intel_Quick_Sync_Video#Hardware_decoding_and_encoding)

Samba share stuff - <https://superuser.com/questions/115337/windows-7-connecting-to-samba-shares>

The "ls" command - <https://www.freecodecamp.org/news/the-linux-ls-command-how-to-list-files-in-a-directory-with-options/>

<https://briantward.github.io/convert-windows-to-kvm/>

<https://docs.openstack.org/image-guide/convert-images.html>

<https://dannyyda.com/2020/06/25/how-to-use-qemu-img-command-to-convert-between-vmdk-raw-qcow2-vdi-vhd-vhdx-formats-disk-images-qemu-img-create-snapshot-resize-etc/>

USB Limit for VirtualBox - <https://forums.virtualbox.org/viewtopic.php?f=1&t=93436>

<https://jellyfin.org/docs/general/administration/installing/>

[https://www.reddit.com/r/Plex/comments/mpwc8r/linuxserverplex\\_limited\\_to\\_indirect\\_connection/](https://www.reddit.com/r/Plex/comments/mpwc8r/linuxserverplex_limited_to_indirect_connection/)

## Requirements:

Windows - <https://www.microsoft.com/en-us/windows/windows-11-specifications>

Manjaro - <https://manjaro.org/download/>

OMV - <https://en.wikipedia.org/wiki/OpenMediaVault>

Snapraid - <https://www.snapraid.it/faq#howmanypar>

Ventoy - <https://www.ventoy.net/en/faq.html>

## What I'm using:

Intel NUC11TNKi3

Western Digital 1TB WD Green SN350 NVMe Internal SSD

G.Skill RipJaws Series 32GB (2 x 16GB) 260-Pin SO-DIMM PC4-25600 DDR4 3200

Corsair Flash Survivor Stealth 64GB USB 3.0 Flash Drive

## Optional Hardware:

Mediasonic PROBOX 8 Bay 3.5-inch Hard Drive Enclosure - Only if you need eSata

Syba 8 Bay Tool Less Tray Hot Swappable 2.5" 3.5" SATA Non Raid External USB 3.0 Enclosure - Preferred for USB 3.0

## Seagate IronWolf Pro 18TB NAS Internal Hard Drive HDD

### OS Setup:

You can use whatever OS but I picked Manjaro, so that's what I'm going to focus on.

Download Ventoy and install it to your USB drive.

<https://www.ventoy.net/en/download.html>

Copy your OS ISOs to the USB drive.

Manjaro - <https://manjaro.org/download/>

OMV - [https://www.openmediavault.org/?page\\_id=77](https://www.openmediavault.org/?page_id=77)

Install OMV direct to HDD or install to a VM

Installing direct to HDD requires nothing special, you can just follow this guide from the beginning. The relevant parts are from 2:57 to 4:57 for initial install.

<https://www.youtube.com/watch?v=z5CqnmuqyHk>

Skip to Setup OMV.

Installing to a VM requires setting up Virtualbox.

Found a video for this but didn't follow it. <https://www.youtube.com/watch?v=kSMCpeQOGg4>

First, install the Host OS. In my case, that's Manjaro but any OS works. I'm going to assume you know how to install an OS on your own.

For Manjaro, I recommend installing the following optional tools.

SSH - Helpful, in the event the Desktop stops working.

I can't find the instructions I used but you just need to enter the following commands in "Konsole" to enable it.

```
"sudo pacman -S openssh"
```

```
"sudo systemctl enable --now sshd"
```

NoMachine - Useful for remote admin of the Host OS. <https://techviewleo.com/install-and-use-nomachine-rdp-on-arch-manjaro/>

Commands used:

```
"sudo pacman -Syu"
```

```
"sudo pacman -S git"
```

```
"sudo pacman -S --needed git base-devel"
```

```
"git clone https://aur.archlinux.org/nomachine.git"  
"cd ~/Downloads"  
"cd nomachine"  
"makepkg -s"  
"ls -lh"  
"sudo pacman -U nomachine-*-x86_64.pkg.tar.zst"  
"pacman -Qi nomachine"
```

Barrier - KVM software based on Synergy. Honestly, Synergy works better but isn't great for other reasons.

Can't find the original install directions I followed but this is what I did.

Click Add/Remove Software

Click the three horizontal bars on the top right of the screen.

Click Preferences.

Enter password.

Click "Third Party" tab

Click "Enable AUR support"

Click Search icon

Type "Barrier" and install it

It has a setting to start with the OS but it never works. I have to use NoMachine to remote connect and launch it.

VNC Viewer - Only needed it for a short time but just in case. Either scroll a lot or ctrl+f "Manjaro"  
<https://linuxhint.com/install-realvnc-vnc-viewer-linux/>

Commands used:

```
"sudo pacman -Sy"  
"sudo pacman -S git fakeroot binutils"  
"cd ~/Downloads"  
"git clone https://aur.archlinux.org/realvnc-vnc-viewer.git"  
"cd realvnc-vnc-viewer"  
"sudo pacman -U realvnc-vnc-viewer-*-x86_64.pkg.tar.zst"
```

Other commands I used:

```
"su" - elevate prompt so you don't need sudo  
"pwd" - show current directory  
"sudo fdisk -l" - ?  
"mhwd -li" - show ??  
"pamac install gvfs-smb" - install samba  
"rsync -avr /path/to/dir1 /path/to/destination" - how to copy with rsync
```

To install Virtualbox - <https://wiki.manjaro.org/index.php/VirtualBox>

Installing the expansionpack with AUR didn't work, just download it directly using the links below that section and install it from Virtualbox itself. Use Tools -> Expansion Pack Manager to install it.

Note: Techno Dad Life recommends VHDs instead of VDIs for the virtual drives.

Once you have Virtualbox and the expansionpack installed, setup the VM. The automagic install failed, so I had to manually install it. For the VM itself, I setup a 20 GB VDI, 2 CPUs, 16 GB (16384) of RAM for the initial setup. I went with half of my available RAM because I'll likely need at least 12 GB just based on software requirements. See <https://www.snapraid.it/faq#mem> and the SnapRAID HDD chart below for details. After installing OVM, I shutdown the VM and made the following changes under settings. You can follow this video from 2:57 to 4:57 for initial install.

<https://www.youtube.com/watch?v=z5CqnmuqyHk>

First, I add a second VDI for app data.

Storage -> Controller: SATA -> HDD+ Icon (Adds hard disk.) -> 200 GB

Then, I switch to bridge mode, so the VM can have its own IP#. This requires a second Ethernet port on your device. Use the "ip a" command to help identify which Ethernet port you want. The last part is the name of my actual adapter on my dock, yours will probably be different.

Network -> Attached to: "Bridged Adapter" -> "enp47s0u1u4u3"

Finally, I attached the external drives.

USB -> Check "USB 3.0 (xHCI) Controller" -> USB Connector+ Icon for each drive (Adds new USB filter with all fields set to the values of the selected USB device attached to the host PC.)

Hard drive info based on SnapRAID.

Max HDDs:

8x 16 TB = 128 TB

4x 8 TB = 32 TB

160 TB Total

10 GB RAM Minimum

Current HDDs:

3x 16 TB = 48 TB

3x 12 TB = 36 TB

4x 8 TB = 32 TB

116 TB Total

8 GB RAM Minimum

11 Drives Total

2 Parity Required

## Parity

1x 16 TB

1x 12 TB

28 TB Parity

## Data

2x 16 TB = 32 TB

2x 12 TB = 24 TB

4x 8 TB = 32 TB

96 TB Data

## Rules:

1 GB RAM per 16 TB Data

Parity : Data

1 : 1-4

2 : 5-14

3 : 15-21

4 : 22-28

## Setup OMV

Once you have the VM fully setup, it's time to setup OMV. We'll be once again following this video. <https://www.youtube.com/watch?v=z5CqnmuqyHk>

You can resume following the video where we left off at 4:47 until about 7:40, where he tells you how to change your password. I suggest going ahead and changing your password then, so you don't forget.

You'll need this link soon. <https://wiki.omv-extras.org/>

Resume the video until 11:18. At this point, you'll have omv-extras installed and return to the plugins page, install the mergerfs, snapraid and filebrowser (filebrowser is optional but recommended; you can follow this video to set it up [https://www.youtube.com/watch?v=\\_gBnhZiWuNw](https://www.youtube.com/watch?v=_gBnhZiWuNw)) plugins and resume the video until about 15:26.

At this point, you can either wipe the drives the way he says or, if you already have drives formatted for Linux, you can skip this part. Resume until 16:45.

If you wiped the drives, keep following the video until 17:08. If you already have a file system, click Mount instead of Create and resume at 16:58 until 17:08. Remember to repeat the steps for each data drive. The 200 GB VDI will need a file system and it will need mounting but there's no point in enabling SMART monitoring for it. Once you have all of your drives mounted, go to Storage -> mergerfs to setup your drive pool.

I only setup one big pool for my library but you can make as many as you want. Remember to save blank drives to use as parity in SnapRAID later, if you plan to use it. To setup a pool, just click the + icon, enter a name, add your drives and pick your policy. I found "Least used space" to be the least frustrating policy but feel free to read more about the policies at the following site. <https://github.com/trapexit/mergerfs> You'll also want to read this page. <https://github.com/trapexit/mergerfs-tools>

Once you have your drive pools setup, it's time to resume the video. Remember to setup shares for your appdata mount and each drive pool. While you can try to setup a detailed user policy, be warned that it can cause a mess of your permissions. I ended up making my entire library read only thanks to that. I'll explain how to fix that problem later on though. Continue the video until the end.

If you want to setup user home directories, you can check "Enabled" and, optionally, "Enable recycle bin" under Services -> SMB/CIFS -> Settings. However, you have to first enable User home directory and give it a destination under Users -> Settings. I was doing exactly this and installing Plex when my library became read only. I feel like the Plex install had more to do with it but I can't be sure this had no impact. It's an old version of OMV but this video basically explains user home folders. <https://www.youtube.com/watch?v=oEbba-eaIDU>

If you want to solve permission problems with the shares themselves, this video will help. <https://www.youtube.com/watch?v=G1KuoC1b6GY>

However, if your files themselves have all been restricted to root access like mine were, you'll want to use one of the following.

Using WeTTY or SSH:

<https://linux.die.net/man/1/chown>  
<https://linux.die.net/man/1/chmod>

Install webmin GUI:

<https://www.webmin.com/udeb.html>

You have to login as "root" to make webmin work. I haven't figured out how to actually use it, yet.

You can setup SnapRAID whenever you want. I haven't set it up because I'm currently shuffling files around and don't have the extra space, yet. However, you can probably follow this video starting at 4:18 to do it. <https://www.youtube.com/watch?v=FYkdPyCt5FU>

To setup Plex in Yacht, you can reference this video but it will not be enough on its own.  
<https://www.youtube.com/watch?v=oLnwyZx-HqQ>

The main problem I ran into with Plex is it doesn't like storing its appdata on a mergerfs pool.  
That's why I setup the 200 GB VDI for appdata.

So, I confirmed that it was Plex that write protected my whole library. This page has a guide on how to setup Plex in a docker and it specifically mentions that it will write protect folders, unless you tell it not to. <https://github.com/plexinc/pms-docker>

I'm also learning how to use Intel Quick Sync in a VM with the following guide.  
[https://github.com/intel/gvt-linux/wiki/GVTg\\_Setup\\_Guide](https://github.com/intel/gvt-linux/wiki/GVTg_Setup_Guide)

To get the Plex Claim Token, visit this site. <https://www.plex.tv/claim/>

I ended up using the premade docker container to get Plex running because no amount of editing could get the stack version to stop giving syntax errors. I had to wait a minute for Plex to show up after setting it up correctly with a claim token. <https://hub.docker.com/r/linuxserver/plex>

To fix the problem with not being able to create an NFS share, add the following in front of each disk in the mergerfs mount.

```
,x-systemd.requires=
```

So, you get something like: `,x-systemd.requires=/srv/dev-disk-by-uuid-1a9b42c6-b9c2-4b5a-ae75-a227643e51b7`

To Remount mergerfs pool:

Services -> SMB/CIFS -> Shares

Delete share, Apply changes

Storage -> Shared Folders

Delete share, Apply changes

Storage -> mergerfs

Delete pool, Apply changes (Takes a while)

Recreate pool (Should show as mounted under Storage -> File Systems)

Recreate share

Recreate SMB share

```
/srv/dev-disk-by-uuid-1a9b42c6-b9c2-4b5a-ae75-a227643e51b7
```

```
/srv/dev-disk-by-uuid-e0e83b37-2074-4b89-b861-1aa952bfdc0f
```

```
/srv/dev-disk-by-uuid-afa38c07-7b53-4a6f-a70a-004ddb78e3ed
```

```
/srv/dev-disk-by-uuid-6604d48b-dd2e-4c6a-867b-a5fe12e0feae
```

```
/srv/dev-disk-by-uuid-87c8acab-a768-438e-a36a-6ae6ae17dfef
```

```
/srv/dev-disk-by-uuid-75c522d4-0f07-48fc-83c4-3a339bac5867
```

Enabled  
Media  
Guests allowed  
Browseable  
Hide dot files

Mergerfs and Remote Mount: <https://forum.openmediavault.org/index.php?thread/47442-mergerfs-and-remote-mount/>

KVM:  
[https://www.reddit.com/r/OpenMediaVault/comments/10ndmdl/omv\\_6\\_kvm\\_plugin\\_performance/](https://www.reddit.com/r/OpenMediaVault/comments/10ndmdl/omv_6_kvm_plugin_performance/)

Problems:

<https://forum.openmediavault.org/index.php?thread/53452-super-long-error-code-when-creating-nfs-share/&postID=396861#post396861>

Converting Windows bare metal install to KVM.

Download Disk2VHD (<https://learn.microsoft.com/en-us/sysinternals/downloads/disk2vhd>) to convert the OS drive into a VHDX image.

Download qemu-img for Windows (<https://cloudbase.it/qemu-img-windows/>) to convert it to qcow2 format. Use the command "qemu-img.exe convert -O qcow2 <image.vhdx> <image.qcow2>"

Setup VM using Windows VM image

<https://forum.openmediavault.org/index.php?thread/39510-how-to-use-the-openmediavault-kvm-plugin/#post277368>

# Ubuntu Server

As I mentioned on the File Server page, this is the de-facto server for anyone wanting to host services. It's stripped down to only the most necessary apps for the purpose of remote connecting and setting things up. It's free and it has great community support. There's not much more you can ask for, so long as you know what you're doing.

It's not the best for beginners because it doesn't hold your hand or even provide a web UI. You can install your own web UI and gain all of the conveniences of any other server OS through various apps. There's not much of a limit to what you can run on it.

This is what I call the baseline for any server. Everything else is a trade off of advantages and limitations.

# Proxmox

Proxmox is purpose built for running VMs and containers. There's a reason most people pair it with Kubernetes.

This is in my top three (with OMV and Ubuntu Server) for a server OS and my pick for clusters and scalability.

The biggest limitation is that you're generally going to be running things in either a VM or a container. So, it doesn't make a great choice for basic/cheap file sharing.

Note that there will be a learning curve to it. It does some hand holding in how it streamlines certain tasks but it's not going to necessarily be simple. Even setting up containers can require extra steps due to the nature of how Proxmox handles things.

# Linux Mint Debian Edition

Okay, you got me, this one isn't actually a server OS.

However, it can actually do just fine acting as a server. It comes with a full desktop UI and graphical tools. It resembles Windows. It's generally the most user friendly and hand holding version of Linux under the Sun. It also inherits the rock solid performance and stability of Debian.

The drawback, of course, will be a performance hit from the desktop and related apps. I don't recommend it for a low cost SBC but it should still work on one. Honestly, the performance hit shouldn't be super significant in most cases.

This is actually the OS I'm currently hosting most of my Docker containers on and it just works. The setup has been super easy. The only real issue I had was hardware related. It has turned my little Intel NUC into my "do everything and a bag of chips" machine.

If I were determined enough, I could uninstall the desktop and the GUI apps and basically make it function similar to Ubuntu Server but I'd be better off installing Ubuntu Server, in that case.

# Windows

You're not going to get more hand holding than you'll get with Windows. This is regardless of whether you talk about Windows 11 or Windows Server.

Know that Windows Server comes with limitations that the desktop version doesn't, even though it has a desktop environment that is pretty much the same. So, don't expect to have a fun time running Crisis or whatever on it.

Expect it to have significant performance hits compared to Linux. Expect to have to either pay extra or put in extra effort getting some things to work with it.

You **can** run containerized apps through Docker on it and do most other Linux tasks on it.

Honestly, if you're already neck deep in the Windows ecosystem and just want your hand held to the maximum extent, it's probably the best choice. Don't expect to have a fun time running VMs or scaling stuff.

I really don't recommend using Windows as a server OS. There's a reason Linux absolutely dominates that market. I can't overstate the inconvenience and performance hit you'll take running this over literally any flavor of Linux.

However, it's worth noting that there are certain circumstances where you might want to pick Windows due to certain hardware or software. Now, most of that hardware and software isn't going to matter much in a self hosting environment but maybe you have really specific needs.

# CasaOS, ZimaOS, Cosmos Cloud OS, RunTipi, etc.

So, none of these are a true OS. At best, some of them are trying to be an app suite built on top of a Linux distro. Some are literally just an app you install on whatever Linux OS you want.

The thing they all have in common is they try to simplify setting up Docker apps through the use of an "app store" where you just click install and off you go.

They all have some serious limitations and disadvantages to anyone wanting even a slightly customized install for any Docker app. Even trying to by-pass the store to install a Docker container directly can result in a mess.

Unless you want to run a bunch of Docker apps with the most generic settings possible and almost zero effort on your end, I say stay away. Stay far, far away from these. They're mostly going to hamper your efforts in the long run.

That said, if you absolutely need/want to use one, I'd probably recommend going with [Tipi](#).