Servers

Here is where I'll discuss server hardware.

- Where to start
- Storage/File Server
    - General Info
    - How to test used/renewed hard drives
    - How to transfer files from old drive(s) to new one
- General purpose/Media Server
- Appliance Server
- AI Server
- Enterprise Server

Where to start

So, there's basically four types of servers most people will want in their home lab.

This first and most common is storage aka a file server. There's a range of hardware to discuss here.

The second is appliance type servers. They're usually single board computers with only one or a few functions being served.

The third is media servers. They often require the ability to transcode and can also be thought of as a general purpose server.

The fourth is an AI server. The entry level to this will cost as much as a moderate to high end gaming PC. In fact, most people will end up converting a gaming PC into their first AI server. You can technically run AI on lower end hardware but I wouldn't call it functional.

Now, in terms of actual hardware, these are the types of servers you are likely to find:

SBC - Single board computers. These are the Raspberry Pis and the ZimaBlades of the server world. They're typically what you find acting as an appliance server. Can be used for low performance/test clusters. Price range is about $200 or less.

Mini PCs - This is a relatively new entry to the field. They actually make great general purpose/media servers. They can also serve as a significant step up from SBCs for forming a cluster. Price range tends to be $200 or more. Honestly, the only reason they're a separate category from SBCs is because of the significant jump in price/performance/features that happens around the $200 mark.

Custom PCs - This is a desktop PC you build yourself from parts. This is only practical when you either have very specific requirements or when a mini PC is no longer practical for the performance you need. AI is the primary service I can think of that you'd want one for.

Enterprise Servers - These are a mixed bag and I generally don't recommend them in a home lab environment, unless you specifically need one to learn on, such as when pursuing a Cisco certification. You'll find almost exclusively used ones in home labs. Most normal people simply cannot afford a new one, much less consider it a practical expense. What's worse is proprietary hardware can become a major gotcha here.

Storage/File Server

General information about this type of server.

General Info

I'm starting here because this is, by far, the most common server people will want.

Simply put, it's a device that centralizes storage on your network. There's a lot to say about the software side but not actually a whole lot to discuss for the hardware.

Your best bet here would be either an old PC that you already have that's collecting dust or a new mini PC. You very well could even get by with just a SBC.

If you're looking to buy new hardware, I'd recommend going to the Mini PCs sub Reddit and look at the spreadsheet (posted here) comparing mini PCs. Alternatively, you could just get something like the ZimaBlade.

I do **not** recommend prebuilt NAS devices anymore, unless you are just super desperate to trade money for effort on setting it up. If you really need to go this route, Synology is currently the big dog in this arena but be sure to do your own research. Know that you will absolutely save money rolling your own storage solution.

Something worth noting is that, there's basically three routes to massive storage.

The absolute cheapest is USB enclosures. You can get 5 bay and 8 bay Syba USB enclosures for a lot cheaper than you can add that many hard drives to a server through literally any other method known to man. However, the caveat is that hardware RAID absolutely will not work with it. Software RAID still works just fine though. You might also encounter performance and reliability issues but the enclosures I recommend shouldn't really have much of an issue in that regard. It's worth noting that you need 5 (1 parity, 4 data) hard drives to max out single disk parity/raid and 8 (1 parity, 7 data) hard drives each for each additional disk worth of parity. I should mention that the second parity disk upgrades the total data disks from 4 max to 14 max. I don't know the specifics of why the first parity disk by itself only handles 4 data disks vs 7 data disks once you add the second parity disk.

The second cheapest is to use eSATA enclosures. This gets a little more pricey, even if the enclosure itself costs/is the same. Why? You need eSATA ports with port multipliers to make it work. This usually involves buying a PCI expansion card (eSATA or more likely HBA SAS controller/ eBay/YubTub) and a break out cable. There are a few variations you can make with the hardware but the result is roughly the same, even if the cost can vary quite a bit.

The last route is to just buy a massive storage server. Those things cost bank, tend to have a high power draw and noise level but they absolutely provide performance and reliability.

When it comes to hard drives, it's currently difficult to beat the 18 TB Seagate IronWolf Pro drives for sheer quantity of storage. You can save even more by buying them refurbished. You do have to be careful about buying refurb drives. There are reputable sites that specialize in selling them. (Found the links on Reddit but can't find them right now.)

For SSDs, SAMSUNG and Crucial are usually the best deals considering price, performance and reliability. As usual, you should shop around yourself for the actual best deals. This counts for both SATA and NVME SSDs. The best bang per TB is currently fluctuating but I recommend sticking with at least 1 TB for any SSD. I couldn't find any good NVME SSDs for non-standard sizes. The standard size is 2280 but they can be smaller and require different (such as B and M key) slots. The last time I tried to buy a non-standard NVME, it didn't go well. If your PC includes a non-standard NVME slot, just don't put a SSD in it, would be my recommendation. If you're feeling adventurous, feel free to gamble on it but know that you'll have to figure that out yourself.

There's actually a lot of ways you can try to expand storage options on a PC but it's usually cheaper to either add more external USB enclosures until you reach the device limit of your PC's USB controller (which usually isn't that high and can be really low on older PCs) or to just buy another PC to use as a storage server.

If you don't mind paying a premium, there are mini PCs that are designed to work as NAS.

If you're wondering why I don't really recommend full sized desktops or used Enterprise servers, it's purely due to the power draw and noise. You can usually put in a bit of effort to silence them but there's really nothing that can be done about the power draw. A typical mini PC should be using about 1/10th the power of a full desktop or Enterprise server. Admittedly, this isn't always the case. So, be sure to shop around.

Now, if you have money to burn and noise isn't a concern, you can absolutely build yourself a network rack with tons of used enterprise servers with tons of built-in hot swappable drive bays. Otherwise, I don't suggest it.

NoteToSelf: Find and link more sources

Storage/File Server

How to test used/renewed hard drives

My Testing methodology (u/echogecko795)

This is something I developed to stress both new and used drives so that if there are any issues they will appear.
Testing can take anywhere from 4-7 days depending on hardware. I have a dedicated testing server setup.

I use a server with ECC RAM installed, but if your RAM has been tested with MemTest86+ then your are probably fine.

1) SMART Test, check stats

> `smartctl -i /dev/sdxx`
> "**-i, --info**Prints the device model number, serial number, firmware version, and ATA Standard version/revision information. Says if the device supports SMART, and if so, whether SMART support is currently enabled or disabled. If the device supports Logical Block Address mode (LBA mode) print current user drive capacity in bytes. (If drive is has a user protected area reserved, or is "clipped", this may be smaller than the potential maximum drive capacity.) Indicates if the drive is in the smartmontools database (see '-v' options below). If so, the drive model family may also be printed. If '-n' (see below) is specified, the power mode of the drive is printed."
>
> `smartctl -A /dev/sdxx`
> "**-a, --all**Prints all SMART information about the disk, or TapeAlert information about the tape drive or changer. For ATA devices this is equivalent to
>
> ```
> '-H -i -c -A -l error -l selftest -l selective'
> ```
>
> and for SCSI, this is equivalent to
>
> ```
> '-H -i -A -l error -l selftest'.
> ```

> Note that for ATA disks this does **not** enable the non-SMART options and the SMART options which require support for 48-bit ATA commands."
>
> `smartctl -t long /dev/sdxx`
> "**-T TYPE, --tolerance=TYPE**[ATA only] Specifies how tolerant **smartctl** should be of ATA and SMART command failures.
>
> The behavior of **smartctl** depends upon whether the command is "**optional**" or "**mandatory**". Here "**mandatory**" means "required by the ATA/ATAPI-5 Specification if the device implements the SMART command set" and "**optional**" means "not required by the ATA/ATAPI-5 Specification even if the device implements the SMART command set." The "**mandatory**" ATA and SMART commands are: (1) ATA IDENTIFY DEVICE, (2) SMART ENABLE/DISABLE ATTRIBUTE AUTOSAVE, (3) SMART ENABLE/DISABLE, and (4) SMART RETURN STATUS.
>
> The valid arguments to this option are:
>
> *normal* - exit on failure of any **mandatory** SMART command, and ignore all failures of **optional** SMART commands. This is the default. Note that on some devices, issuing unimplemented optional SMART commands doesn't cause an error. This can result in misleading **smartctl** messages such as "Feature X not implemented", followed shortly by "Feature X: enabled". In most such cases, contrary to the final message, Feature X is **not** enabled.
>
> *conservative* - exit on failure of any **optional** SMART command.
>
> *permissive* - ignore **failure**(s) of **mandatory** SMART commands. This option may be given more than once. Each additional use of this option will cause one more additional failure to be ignored. Note that the use of this option can lead to messages like "Feature X not implemented", followed shortly by "Error: unable to enable Feature X". In a few such cases, contrary to the final message, Feature X **is** enabled.
>
> *verypermissive* - equivalent to giving a large number of '-T permissive' options: ignore failures of **any number** of **mandatory** SMART commands. Please see the note above."

2) BadBlocks -This is a complete write and read test, will destroy all data on the drive

> “ `badblocks -b 4096 -c 65535 -wsv /dev/sdxx > $disk.log`

3) Real world surface testing, Format to ZFS -Yes you want compression on, I have found checksum errors, that having compression off would have missed. (I noticed it completely by accident. I had a drive that would produce checksum errors when it was in a pool. So I pulled and ran my test without compression on. It passed just fine. I would put it back into the pool and errors would appear again. The pool had compression on. So I pulled the drive re ran my test with compression on. And checksum errors. I have asked about. No one knows why this happens but it does. This may have been a bug in early versions of ZOL that is no longer present.)

> `zpool create -f -o ashift=12 -O logbias=throughput -O compress=lz4 -O dedup=off -O atime=off -O xattr=sa TESTR001 /dev/sdxx`
>
> `zpool export TESTR001`
>
> `sudo zpool import -d /dev/disk/by-id TESTR001`
>
> `sudo chmod -R ugo+rw /TESTR001`

4) Fill Test using F3 + 5) ZFS Scrub to check any Read, Write, Checksum errors.

> `sudo f3write /TESTR001 && f3read /TESTR001 && zpool scrub TESTR001`

If everything passes, drive goes into my good pile, if something fails, I contact the seller, to get a partial refund for the drive or a return label to send it back. I record the wwn numbers and serial of each drive, and a copy of any test notes

> 8TB wwn-0x5000cca03bac1768 -Failed, 26 -Read errors, non recoverable, drive is unsafe to use.
>
> 8TB wwn-0x5000cca03bd38ca8 -Failed, CheckSum Errors, possible recoverable, drive use is not recommend.

References:

https://www.reddit.com/r/homelab/comments/17oobde/what_do_you_use_to_test_your_drives_that_you_buy/

https://linux.die.net/man/8/smartctl

...

How to transfer files from old drive(s) to new one

This is technically a software topic but, since you're only likely to use this when you get a new hard drive, I'm leaving it in the hardware section.

The first step is to go to Storage -> Disks and find out which device (/dev/sd#) your new drive was assigned.

Go to Storage -> S.M.A.R.T. -> Devices and Edit the drive to enable monitoring.

If you want to test the drive, see the previous page about testing used drives.

Go to Storage -> File Systems and either (triangle play button) mount the existing file system or (plus button) create a new one and mount it. EXT4 or BTRFS are preferred. BTRFS is mostly for bit rot protection but is completely redundant with snapraid.

Go to Storage -> Shared Folders and share the new drive by itself. Repeat this step for any old drives you wish to retire.

When you go to pick it under Shared Folders, don't just accept what OMV puts in for the path/directory. Instead, all the way to the right, there's a "folder tree" icon. Click that and browse to the root of where you want to copy the files from (should just be "/" to get everything on the drive, but if you are copying from a particular directory, it will be like "path/to/dir").

Find your way to the command line. The guide tells you how to use WeTTY. Login as root (you can use `sudo -i` to switch to root) and use the following command.

rsync -avP /path/to/dir1/ /path/to/dir2/ /path/to/destination/ - This example merges two sources to one destination.

-a (archive): Preserves symbolic links, permissions, ownership, and timestamps. Identical to -dR --preserve=all, preserving recursive structure, symlinks, and attributes.
-P: Combines --partial (keeps partially transferred files if the sync is interrupted) and --progress (shows a progress bar during the transfer).
-v (Verbose): Increases the information provided during the transfer, listing each file as it is processed.

v and P are optional. Use P, if you want to be able to resume large files. On resume attempts, use the --append or --append-verify switch to actually resume. See this page for further details on the command. https://docs.jbcoder.us/books/linux/page/useful-commands

Be sure to use the "Absolute Path" (it has a convenient copy button) from Storage -> File Systems to find the correct paths.

Once you've entered the correct command, you're likely to be in for a bit of a wait.

Once it's done copying, you can add the new drive to your mergerfs pool or share it however you wish. Likewise, you can remove old drives from the pool, unshare, unmount and disconnect them. Be sure to remove all references before you disconnect any drives. This includes other services like File Browser. If you don't get an error when trying to unmount the file system, you most likely succeeded in removing any other references.

Don't forget to delete any temporary shares. Keep in mind that new shares will create a new folder (with the name of the share) on each drive it applies to. You have to use a root "/" share to delete any empty share folders you created by mistake. You can use the `rm <folder>` command to delete an empty folder. Use `rm -R <folder>` to delete non-empty folders. Obviously, be cautious about deleting things since there's no easy way to undo it.

Reference:
https://www.reddit.com/r/OpenMediaVault/comments/1k3tbme/fastest_way_to_transfer_data/mwg44u3/

General purpose/Media Server

Honestly, you can use almost anything for this. However, I'm going to give pretty much the same recommendations as I did with the file server.

The biggest difference is, for the sake of performance, efficiency and cost, I'm going to strongly steer you towards a $200-$500 (bare bones) mini PC with a decent core/thread (4/8 minimum) count, at least 32 GB of RAM and a 1 TB NVME SSD. You can go bigger but, going smaller would put you in appliance territory and that's a separate section of this guide.

Something else to consider is to make sure it has an integrated CPU with a feature like Intel Quick Sync. It's the minimum of what you'll need to handle transcoding on something like Plex. That said, unless you've got more than a half dozen people watching streams at the same time, that's all you should really need.

A high core/thread count and high RAM quantity are important, if you want to host a lot of services at once. I'll cover this more in the software section but containerized apps like Docker/Kubernetes will greatly benefit from that. Most of your storage should be on your NAS, so the 1 TB NVME SSD should be plenty for local storage. The speed/timings of your RAM are only important in that they should match what your motherboard supports. Faster RAM speeds are pretty much negligible when it comes to performance. As time goes on, I continue to feel like there's no such thing as too much RAM, as long as your PC supports it. I'd consider 32 GB the minimum for anything running at least a half dozen services. Even then, some services are just plain resource hogs and may need their own hardware, just to keep them from choking out other services. qBittorrent is especially the app I think of when I say this. Oddly, the Windows version of the app doesn't suffer this problem nearly so much. On the other hand, Plex on Windows was the resource hog. Of course, you can always try to directly restrict resources for the apps to try to get them to play nicely.

As usual, Crucial is my go-to recommendation for RAM and SSDs. There are way too many choices for mini PCs to even begin to recommend one. So, I'll once again point you to the Reddit thread and spreadsheet, for that.

Note that these are also the servers you'd most likely want in a cluster. The exception being something far more beefy like trying to host AI.

Appliance Server

This could be anything from smart home control panels to HTPCs. Pretty much, any app or small suite of lightweight apps will fall into this category.

In most cases, you'll want a cheap SBC to cover you here. Since Raspberry Pis are actually more meant for learning, we're most likely talking about OrangePis, ZimaBlades or something like an Arduino board. In rare cases, you might have to go for a full blown mini PC. It really depends on the purpose of the PC.

I don't have any clear winners here or even some "works in most cases" winners because, as with mini PCs, there's just too many competitors and things just change way too often. Ultimately, you'll need to decide what features are vital to the task and shop around based on that. Again, you can always ask Reddit, if you're struggling.

AI Server

I actually don't have a lot of advice on this because there just isn't a lot known about optimal specs for hosting AI. I'm lumping in other high end servers into this category because high end servers are actually fairly niche. Personally, the best I could do is try to write up a basic build guide for a gaming PC and hope it is somehow relevant to what your server actually needs.

So, I'm just going to link this one Reddit post and hope that's helpful. (Can't find link ??? It was either on SelfHosted or LocalLLaMA or Homelab)

Enterprise Server

If you're buying one for your home lab, it's most likely going to be used. Even at the price of a used one off of eBay, they stretch the definition of viable for a home lab.

There are only two cases where I'd really bother to recommend one and that's if you're trying to learn for the sake of a certification (such as the various Cisco certs) or if you've got money to burn and just want to do it for the giggles.

There's really not much I can say for a buying guide, aside from saying to do your own research on which models will actually serve your needs and that you'll probably find your best deals on eBay. If you're lucky, you might be able to scavenge some for yourself at a business/school that is either closing down permanently or in the middle of a hardware refresh. However, know that you'll likely be competing with professional resellers to grab any. This comes down to connections and luck.

Some particular caveats worth mentioning are:

Power Hungry - These things are designed for 100% uptime and the cost of electricity rarely enters the equation for their design.

Noise - These are typically shoved into closets where they can emulate jet engine noises all they want.

Proprietary - Businesses that make these often come up with contrived hardware DRM just to charge extra. They get away with it because the businesses that buy this stuff are used to paying in the order of 5 to 6 figures for even just one server. Don't be surprised if the server you buy has some kind of hardware gotcha hidden inside that dramatically ups the cost of running it. Some of these can be circumvented cheaply and some can't.

Compatibility - Related to the previously mentioned propriety drawback. Due to the proprietary nature of these servers, they are often severely limited in what kind of hardware works with them.