

Intro

- [What will I need to get started in homelab/homeserver? \(WIP\)](#)
- [What do \[I wish I knew\] / I think you should know *before* starting](#)
- [Overview: My biased opinion of what I think a base home lab requires](#)

What will I need to get started in homelab/homeserver? (WIP)

- **Bring Nothing:** If you are using less than 40% of your machines resources at all times. You have a really nice laptop or desktop that has 8 cores or more. In normal use your machine never sees its resources being pushed to the limit. We can use our personal PC for testing without having to set up a dedicated server.
 - On Windows we can use [WSL+Ubuntu / Hyper-V or least preferred Docker Desktop] to create a deployment environment for testing.
- **Dedicated Server:** This could be anything and there are tons of different choices for OS. I recommend Proxmox above all, then Ubuntu and Debian are also great though I like the simplicity of running a hypervisor as a base OS like Proxmox so we can run OS like Ubuntu or Debian as a Virtual Machine.
- **Server Cluster:** This can be again comprised of literally anything but if you are in the beginner side of the spectrum you might want to consider learning how to operate a dedicated server of some sort before learning about distributed compute, clustering or kubernetes. Proxmox has great clustering capabilities, then theres CEPH is the distributed filesystem that I personally use. Docker has a swarm mode that gives it resiliency and high availability features but I have put off learning that as I do intend to learn kubernetes soon and I have plenty of power with just proxmox's clustering features and single machine docker deployments.

Hardware (Work in Progress)

The used market and your old PCs are your best friends!

This is a general rundown of the main components that run the lab. I want you to at least have been made aware of these concepts though there is no demand for you to become any sort of educated about any of them unless you want to delve deeper. If you are just getting started I heavily recommend, dont get anything expensive. The reason most of my lab is retired enterprise server hardware is because we can get this stuff on deals. Be savvy, dont light your wallet on fire. I mean unless you want to join the club?! ☐☐

Overall Base Machine:

- You do not need any special hardware to begin your home server / homelab journey. A lot of us have special hardware. A lot of us will even claim that we need it. But the truth is that you do not need anything special.
- You can use a old PC or laptop you have laying around. A direct non Wifi network connection is best but that doesn't mean you can't make that work either.
- My personal favorite is used supermicro CSE 826 and CSE 847 units. You can get a 12 x 3.5" SATA chasis for \$200-400 with an x10drh-it rails and everything included depending

on market. These are big, make noise, heat, and use a lot of electricity. But I love them. Be weary if you get it in your head you want to do this. You might not be as crazy as me.

- Also mini PCs are popular like ProDesk 300s, Intel E100 mini pcs.
- You can do a custom build. One of my nodes is a custom built Ryzen machine in a lsv-4500U rosewill case.

NIC (Network interface controller):

- You will need a NIC installed in your machine. If you plan on doing routing (hosting a firewall or router like opnsense or pfSense) you will need more than one interface.
- If you are gonna be doing this for awhile I recommend investing in at least 10G networking components
- 2 x 10g x520-da2 are cheap: <https://www.ebay.com/itm/126876726221>
- 4 x 10g CX3's are available too: <https://www.ebay.com/itm/116656563401>
- I can't recommend anything else, anything should work good enough to learn and good luck.

HBA (Host Bus Adapter):

- The HBA is responsible for managing storage devices, drives. If you are going to be using any SAS drives especially on consumer motherboards... You will need at least one. There are multiple versions of SAS and internal and external variants. This is not meant to be a crash course for that. Just a mention so you know its a thing to look out for
- For learning, stick to SATA drives and your motherboard should be able to provide some SATA ports for storage; no worries.
- LSI LENOVO 9240-8i is cheap <https://www.ebay.com/itm/193280264480>
- Supermicro AOC-S3008L-L8E SAS3 12Gbps also super cheap for SAS3 <https://www.ebay.com/itm/405222212886>

GPU:

- Not necessary, but depending on what you want to run and especially if you want to use this for hosting multimedia or playing with self-hosted ai, security camera machine learning or game streaming, then you will probably want one. Generally Nvidia has the best all-around support and is the most likely to be developed for in self-hosted apps. However a lot of stuff still works on Intel and AMD.
- If you are intending on playing with AI 8-12GB of VRAM is probably a good place to be. At least a NVIDIA 800 series or higher. 10XX series or higher preferred. I'd assume 6xxx ryzen series or newer but I've only tested a 7900 xtx. From what I understand Intel is fresh enough if Intel is supported or not I'd hope the model may not be the biggest issue.

CPU/RAM: Very convoluted to discuss. Every PC needs them obviously. The amount of RAM you need will be determined by how much RAM your workload needs. Can't really guess that. If you have a CPU heavy workload you might need a beefy CPU or a server with multiple CPUs, if you are not CPU heavy; having a big CPU could just raise the utility bill and make noise and heat for no reason. I'm massively overspec and wasting electricity. Can't give the best advice in this area. I am

a mess.

Networking

Switches

- I am a big fan of mikrotik gear. Router OS is extremely flexible and will allow you to implement most any networking concept you want to learn.
- It is really nice if your switches can do POE, some devices can be Powered Over Ethernet; this can be helpful in not having to use an extra cable to get certain equipment installed. All you need with POE is to install the network cable for the device to power on and come up on the network.
- If you are going to be around for awhile if you buy anything I recommend going for 10G stuff. Anything more than 16 ports is super overkill starting out. A 16 port might be perfect if you are planning to grow into it and I currently am using most of the ports.
- If you get gear that has SFP/SFP+ ports do note that its not using the same port you are used to for internet. Typical equipment has RJ45 connectors or CAT, this is what typically ports use that are marked "LAN" or "WAN" in consumer devices. You can get transceivers that convert sfp to rj45, this may be necessary. Do note that devices get a lot more hot using these, it shouldn't be your go to. Aim to connect devices using SFP or RJ45 without adapters. I personally seek to avoid RJ45 as much as I can going forward for some time now.
- Mostly the goal here is just to be able to connect all the devices with something that allows them to intercommunicate with speed and low latency.

Firewall / Router

- A firewall / router is the device responsible for directing the flow of network traffic. In most households this is a device that was supplied to the household by the internet provider, most times a dual purpose device that is both a modem which uplinks with the ISP, and also a router which supplies a local network that provides a place for all the devices on your network to communicate between each other and reach the internet.
- This is a homelab guide. I recommend running pfsense/opnsense. I recommend for you to ditch any consumer routers as soon as possible, if you are even semi curious about this hobby. I have ambitions to learn vyOS in the near future, but hard to recommend something I haven't tried and also only has a CLI and may be discouraging for new users. You can use other routers and firewalls. You don't need one to get started. But these are the best two for flexibility if you want to learn networking and have great community support. You can find answers anywhere imaginable opnsense/pfsense are super popular.
- You can run opnsense/pfsense inside of a VM or even baremetal. Some users suggest its bad to run in a VM. Its COMPLETELY FINE. ☐ Just make sure you have at least 2 ports ideally in addition to the one you plan to use for the host OS (proxmox, etc). Best case 3, pfsenseWAN pfsenseLAN and proxmox LAN. WAN and shared LAN will work with "linux bridges" in proxmox.

- To host services you should at least have admin access to the router/firewall or have communication to the admin where they are hopefully willing to allow you to expose your services to the outside world by allowing you a few ports for outbound communication. This also is not necessary but makes things far easier. If you cant forward ports you can use a VPN like Zerotier or Tailscale, or you can rent a VPS for a server or just to host your own VPN to use to communicate with devices that otherwise would be unable to communicate past your network. How to set this up could be a future topic. Feel free to ask if you have trouble finding this info out on your own.

Access Points (APs)

A access point is just a device like a wifi router that provides a point you can connect to your main network from.

- I got a unifi AC Pro for around \$50 USD shipped. Its working out amazing, it allows me to have 4 unique networks on it. If you need a AP and have more flexible requirements it may be great for you too. I get 400MBP/s to my phone with this. That exceeds my needs, *for Wifi at least.*
- You can literally use anything including old wifi routers you have laying around and isolate each to a single VLAN if you do setup VLANs any point in the future.

UPS (Uninterruptible Power Supplies)

- A UPS is a device that allows for your PC to be prevented from loosing power in the case that your location is without power. It allows for your machines to safely shutdown or be transitioned to a generator in the event of an extended outage. Or it allows for them to remain on during a case where there is a momentarily loss of power.
- I really like my Eaton 5PX 3000. You can find deals on ebay for used ones if you look hard enough. I got mine for less than \$300, they also still sell for way more.

Domain Name

If you want to route services to the internet or host your own websites, while a domain name isn't necessary (you can access your stuff via your public IP and the service port) it does make things so much simpler just having a domain to access everything.

Security Keys

Another thing that isn't necessary but enhances everything. Security keys allow you to have a more secure way of authenticating (logging in) to services. You tie a pinlock code to a device that stores a key more complicated than your passwords might be and you can have more than one. This helps enforce stronger security though it takes some getting used to remembering to bring your yubikey with you everywhere. Mine literally lives on my actual keyring where I keep my house and car keys!

Smart Devices (lights, sensors, locks, cameras etc.)

- I use zigbee protocol for most of my stuff. I have a SMLight brand one. It uses POE and connects to my Home Assistant instance via LAN so I can control my lights from any machine that accesses the zigbee to mqtt service.

Desk phones & SIP trunks

If you want to host your own phone service these are things you will need.

What do [I wish I knew] / I think you should know *before* starting

Dockerfiles: I wish I knew how easy it is to use Dockerfiles to create your own custom container images from the beginning! I wish I had attempted sooner, it is so powerful being able to do that! You can often find dockerfiles on github when you are browsing open source project code alongside the docker-compose.yaml that is used to deploy the app.

Kubernetes:

Its not infinitely hard! If you want the most flexible deployment system its the way to go. You can configure services to stay online during reboots of servers and maintenance. Kubernetes has built in features to support replicating application data between multiple remote sites and the cloud. Kubernetes has a modern design that considers everything that you might need in the modern world it just has a learning curve that is steepest at the beginning.

I sometimes wonder if I should've skipped docker for kubernetes. I was super intimidated with a lot of the tech sector and later found things are easier than I assumed. I have a good enough understanding of kubernetes now to say if I put the time in I'd probably be pretty fluent with it by now even if its somewhat more painful to learn, its just about putting in the time. I definitely plan to learn more soon.

I have spent very little time configuring a cluster consisting of 10 Nodes; 5 masters, 5 workers. I was recommended to scale down to 3 masters max even if I increase workers. Apparently lots of masters isnt really needed lol. I have been planning on switching the deployment to Cilium for a CNI, I am running Calico and metallb and I found after deploying kubernetes and learning more that Cilium would be my preference. Many users I have discussed with are running Cilium as well after making this decision so it seems more unanimous of a decision. Though metallb and calico is just fine!

I wish I ever thought to research the purpose of base linux directories:

<https://www.pathname.com/fhs/pub/fhs-2.3.html>

Directory	Description
bin	Essential command binaries
boot	Static files of the boot loader
dev	Device files
etc	Host-specific system configuration

Directory	Description
lib	Essential shared libraries and kernel modules
media	Mount point for removeable media
mnt	Mount point for mounting a filesystem temporarily
opt	Add-on application software packages
sbin	Essential system binaries
srv	Data for services provided by this system
tmp	Temporary files
usr	Secondary hierarchy
var	Variable data

I used to store my docker deployments in /mnt/app_data/Server/_docker-stack/<appname> ☐☐
Now I store them at /opt/docker/<app name> I only put data in the /mnt or /media folders if it relates to a different disk or medium than the primary disk that the OS is installed on. Its super tidy, and the way it was designed to be used.

Overview: My biased opinion of what I think a base home lab requires

A good homelab base environment should have all of these items sorted out in some regard, this is the core *in my opinion*. There are many ways to achieve these goals, I am sharing my biased opinion of what was best from my perspective. I also will be sorting by order I think we should tackle these goals.

1. ****A hypervisor (Proxmox)**:** (optional) We need at least one machine to build an environment. Things will be best in the future in my opinion with a hypervisor. I have been using Proxmox VE for quite some time now. Its what I recommend. I personally run a 5 node proxmox cluster, the way clustering works is pretty intuitive. All the machines are managed thru the same ui and you can send Virtual Machines in between the individual host machines while they are actively running and without loss of connection. The best part is this is a functionality you can add later, you can add more hosts in the future when we start with just 1. Please aim to have 3+ total machines in the cluster if you ever do intend to cluster, otherwise you will get a suboptimal experience regardless of whether it is possible or not.
2. ****A VM or host machine that can run container images**:** There may be services that you encounter going forward that you might run as a separate VM in proxmox but the bulk of this stuff will require you to either install Ubuntu/Debian or your preferred linux OS on a VM or you can just run it in WSL on Windows or the equivalent whatever that be on MacOS?!
3. We need an easy to use and intuitive way to manage the environment
4. We need authentication for the environment

Version Control:

Backups

Oversight

UPS

Netbird

nginx

truenas

High Availability:

SSO

Yubikeys

Automation

zitadell

bitwarden

2fauth

linkwarden