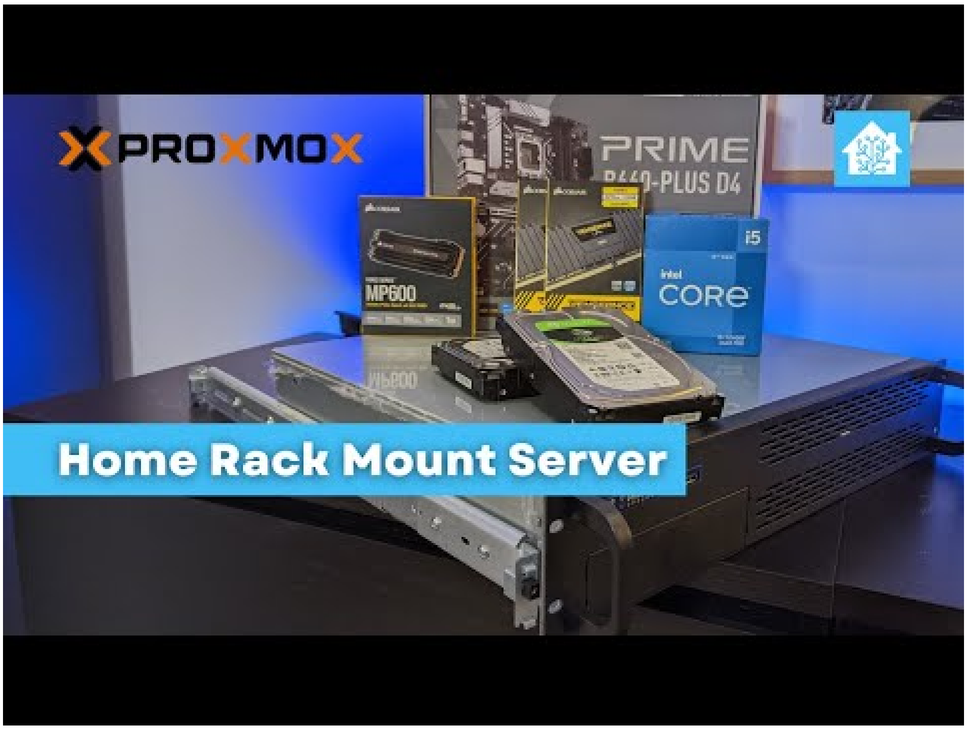
Installing Proxmox on my Home Server

homeautomationguy.io/blog/proxmox/installing-proxmox-on-my-home-server

February 17, 2023

My new smart home runs Proxmox, on a rack mounted self built server. Check out this video to see how I researched what to buy, how I built it, how I installed Proxmox, and how I eventually installed Home Assistant on a Virtual Machine using Prox Mox.



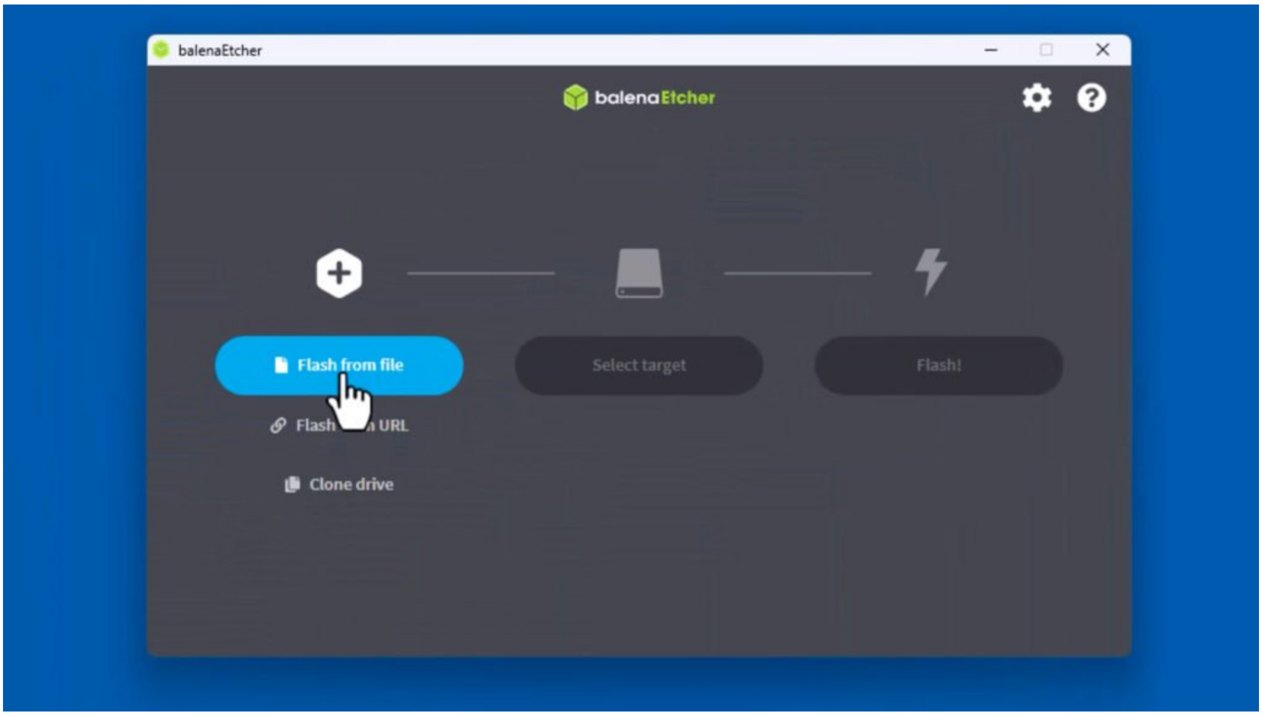
Watch Video At: https://youtu.be/FXoIy3x-GzU

This article documents, step by step, how I installed Proxmox on my home server, and then installed Home Assistant.

Installing Proxmox

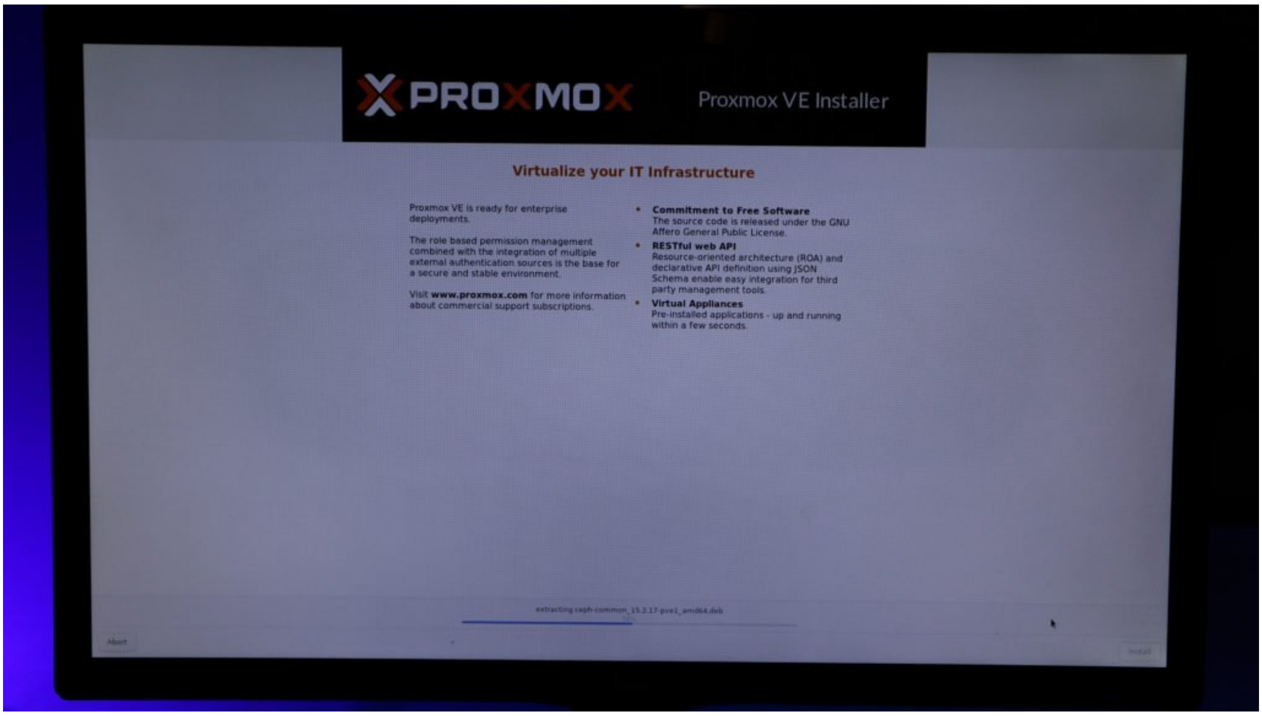
Start off by going over to the Proxmox website and downloading the latest version of the Proxmox VE software ISO installer.

Next, flash the ISO image to a USB thumbdrive using Balena Etcher.



Once it's flashed, plug the USB drive into your computer and boot from it.

Follow the setup wizard to install Proxmox VE onto your server's internal hard disk. I installed mine onto a 1TB solid state NVME m.2 drive. If possible, you should try and install your operating system onto a solid state disk as well, but any old hard drive will do if you've got it.



Virtualize your IT Infrastructure

Proxmox VE is ready for enterprise deployments.

The role based permission management combined with the integration of multiple external authentication sources is the base for a secure and stable environment.

Visit **www.proxmox.com** for more information about commercial support subscriptions.

- **Commitment to Free Software**
  The source code is released under the GNU Affero General Public License.
- **RESTful web API**
  Resource-oriented architecture (ROA) and declarative API definition using JSON Schema enable easy integration for third party management tools.
- **Virtual Appliances**
  Pre-installed applications - up and running within a few seconds.

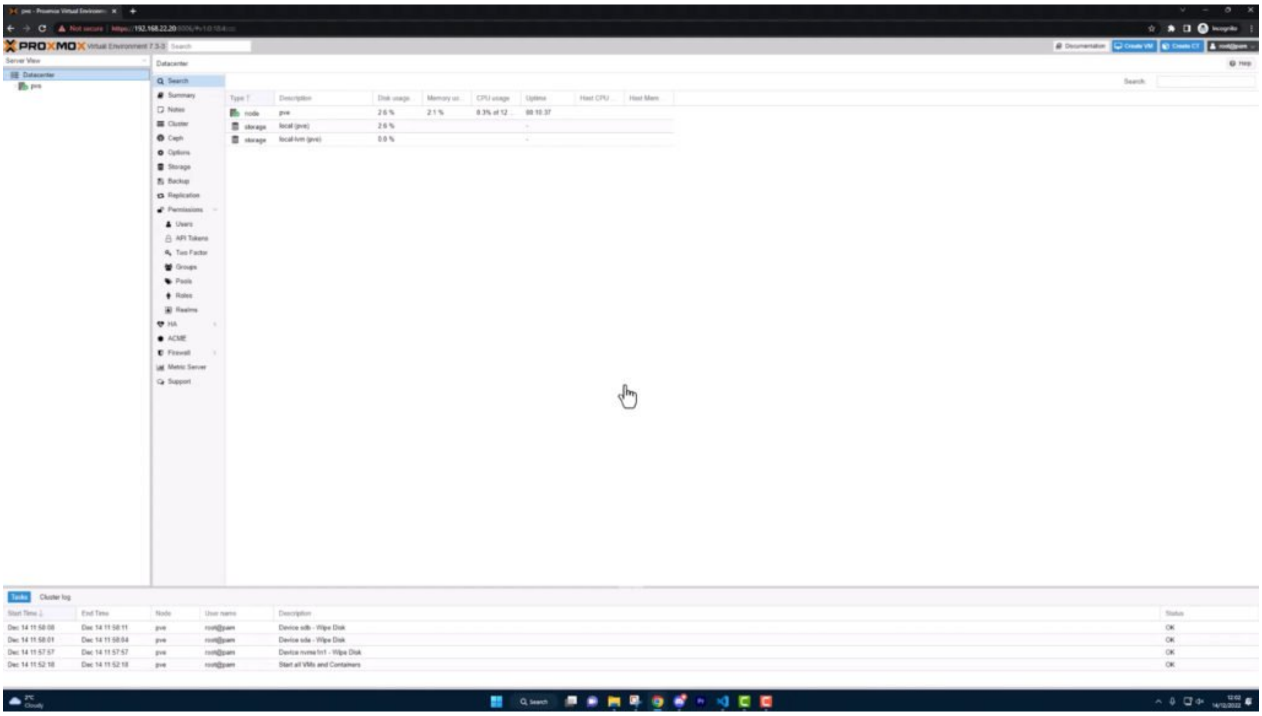
Once the setup is done, you can unplug the USB drive and it should boot up the Proxmox VE operating system!

Configuring Proxmox

Once Proxmox is running on your server you need to go to another computer, open up a web browser and go to the IP address of your Proxmox server and the port 8006.

```
https://<IP ADDRESS OF PROXMOX>:8006/
```

This will bring up the Proxmox web interface. You can log into it using the username *root* and the password you entered during the installation wizard.

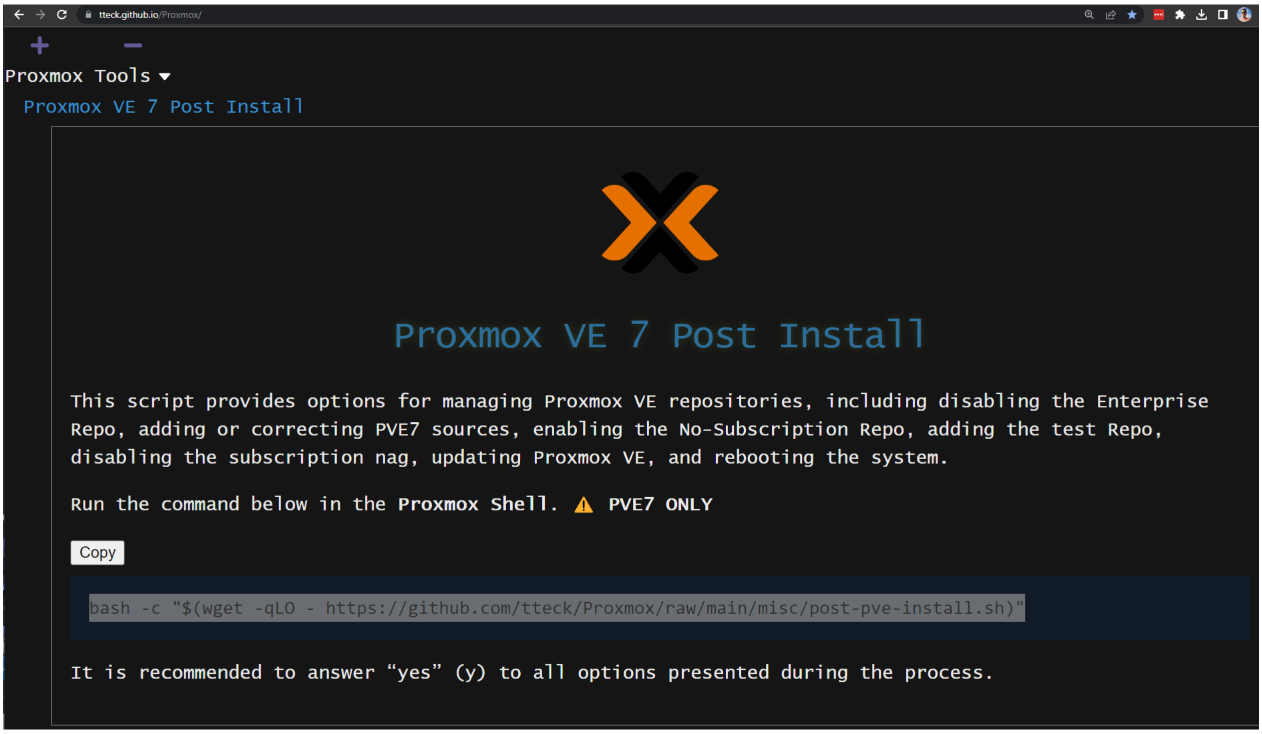


Type ↑	Description	Disk usage	Memory us...	CPU usage	Uptime	Host CPU	Host Mem
node	pve	2.6 %	2.1 %	8.3% of 12 ...	00:10:37		
storage	local (pve)	2.6 %			-		
storage	local-lvm (pve)	0.0 %			-		

Start Time ↓	End Time	Node	User name	Description	Status
Dec 14 11:58:08	Dec 14 11:58:11	pve	root@pam	Device sdb - Wipe Disk	OK
Dec 14 11:58:01	Dec 14 11:58:04	pve	root@pam	Device sda - Wipe Disk	OK
Dec 14 11:57:57	Dec 14 11:57:57	pve	root@pam	Device nvme1n1 - Wipe Disk	OK
Dec 14 11:52:18	Dec 14 11:52:18	pve	root@pam	Start all VMs and Containers	OK

I would now recommend running the proxmox post install script which cleans a bunch of things up and enables some other things (Don't ask me what exactly, but they seemed sensible)

I found this script as part of a handy collection of Proxmox tools in this github library.



Proxmox VE 7 Post Install

This script provides options for managing Proxmox VE repositories, including disabling the Enterprise Repo, adding or correcting PVE7 sources, enabling the No-Subscription Repo, adding the test Repo, disabling the subscription nag, updating Proxmox VE, and rebooting the system.

Run the command below in the **Proxmox Shell**. ⚠ **PVE7 ONLY**

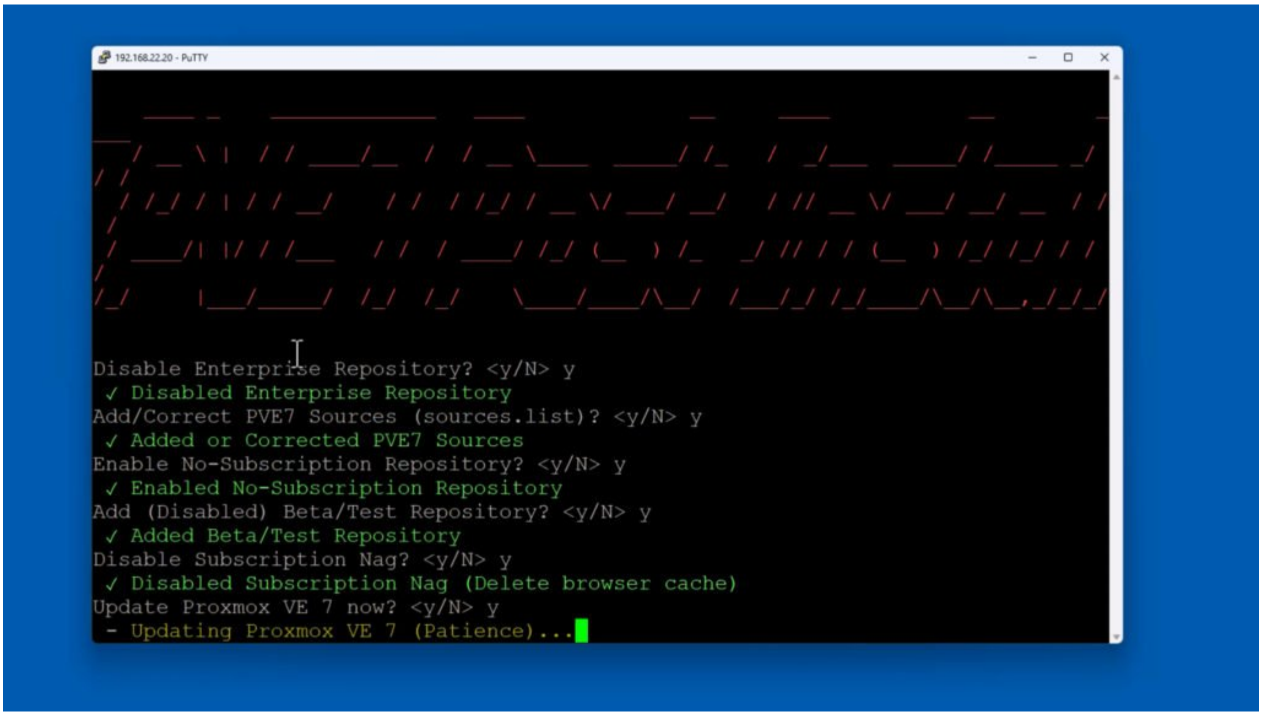
```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/post-pve-install.sh)"
```

It is recommended to answer “yes” (y) to all options presented during the process.

To run the script, SSH into your Proxmox server and run this script in your terminal

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/misc/post-pve-install.sh)"
```

This will make the changes, update Proxmox and then reboot.



```
Disable Enterprise Repository? <y/N> y
 ✓ Disabled Enterprise Repository
Add/Correct PVE7 Sources (sources.list)? <y/N> y
 ✓ Added or Corrected PVE7 Sources
Enable No-Subscription Repository? <y/N> y
 ✓ Enabled No-Subscription Repository
Add (Disabled) Beta/Test Repository? <y/N> y
 ✓ Added Beta/Test Repository
Disable Subscription Nag? <y/N> y
 ✓ Disabled Subscription Nag (Delete browser cache)
Update Proxmox VE 7 now? <y/N> y
 - Updating Proxmox VE 7 (Patience)...
```

My next task was to configure my storage. I have 4 hard disks in my Proxmox server and I am going to set them up like this

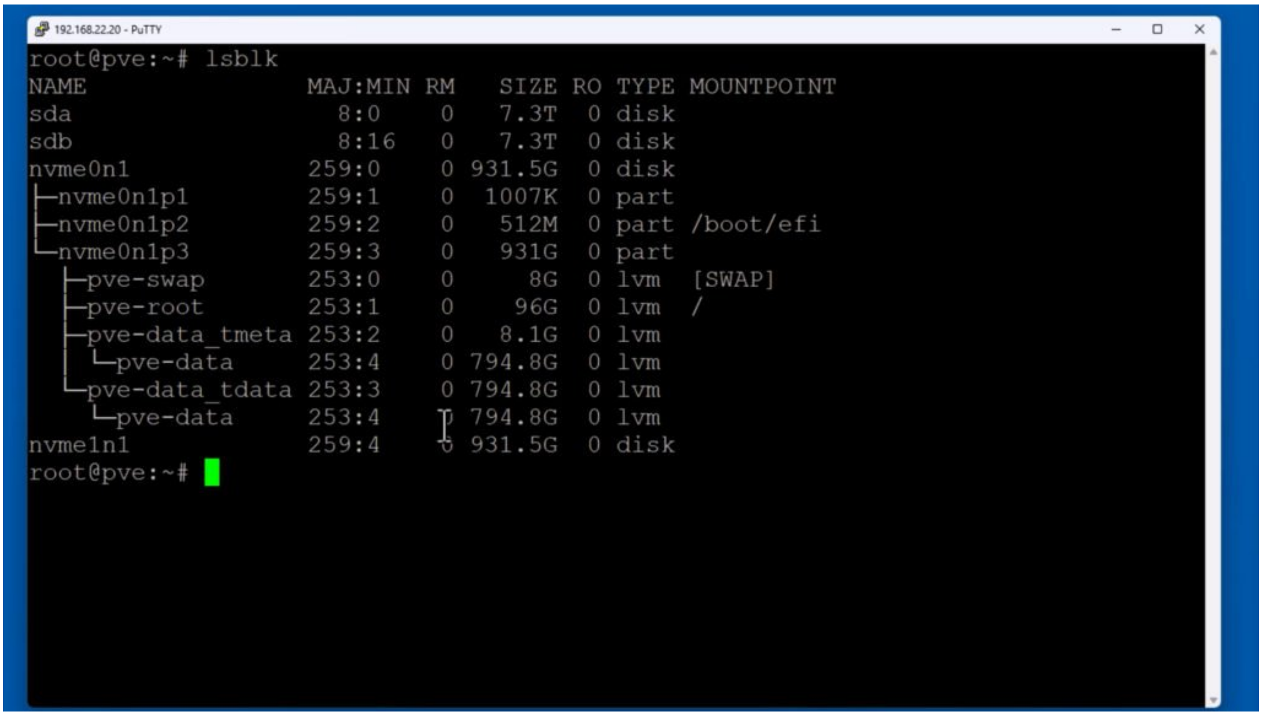
1TB Solid State m.2 NVME	Proxmox Operating System and Hypervisor	This drive is reserved for the operating system and Proxmox application. I chose a fast Solid State disk for this.
1TB Solid State m.2 NVME	Virtual Machines	This is another fast solid state disk that will store my virtual machines. You could always store these on the same hard disk as your operating system, but I've heard horror stories on the internet from people who have done this, where a virtual machine has eaten up all the storage space, leaving Proxmox with no disk space and then it no longer boots up.
8TB 3.5″ SATA Drive	Part of my Data Mirror	This is where I will store my Proxmox templates, backups, ISO Images as well as my media and NAS data.
8TB 3.5″ SATA Drive	Part of my Data Mirror	This is where I will store my Proxmox templates, backups, ISO Images as well as my media and NAS data.

The installer took care of setting up my Proxmox disk, so there was nothing to do there.

The next thing I needed to do was mirror my two 8TB disks. Mirroring the disks means that it acts like one disk, but the files and data are stored across both disks. Unfortunately you lose the capacity of one of the drives (So two 8TB disks still only gives you 8TB of total storage) but the benefit is that you don't lose any data if one of them breaks.

I couldn't figure out how to mirror these in Proxmox in a way that still let me use them for things other than Proxmox, so I did it in the terminal using SSH.

I started by running the *lsblk* command to see how Proxmox and the underlying Linux operating sytsem saw the disks.



```
root@pve:~# lsblk
NAME                  MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                     8:0    0   7.3T  0 disk
sdb                     8:16   0   7.3T  0 disk
nvme0n1               259:0    0 931.5G  0 disk
├─nvme0n1p1           259:1    0  1007K  0 part
├─nvme0n1p2           259:2    0   512M  0 part /boot/efi
└─nvme0n1p3           259:3    0   931G  0 part
  ├─pve-swap          253:0    0     8G  0 lvm  [SWAP]
  ├─pve-root          253:1    0    96G  0 lvm  /
  ├─pve-data_tmeta    253:2    0   8.1G  0 lvm
  │ └─pve-data        253:4    0 794.8G  0 lvm
  └─pve-data_tdata    253:3    0 794.8G  0 lvm
    └─pve-data        253:4    0 794.8G  0 lvm
nvme1n1               259:4    0 931.5G  0 disk
root@pve:~#
```

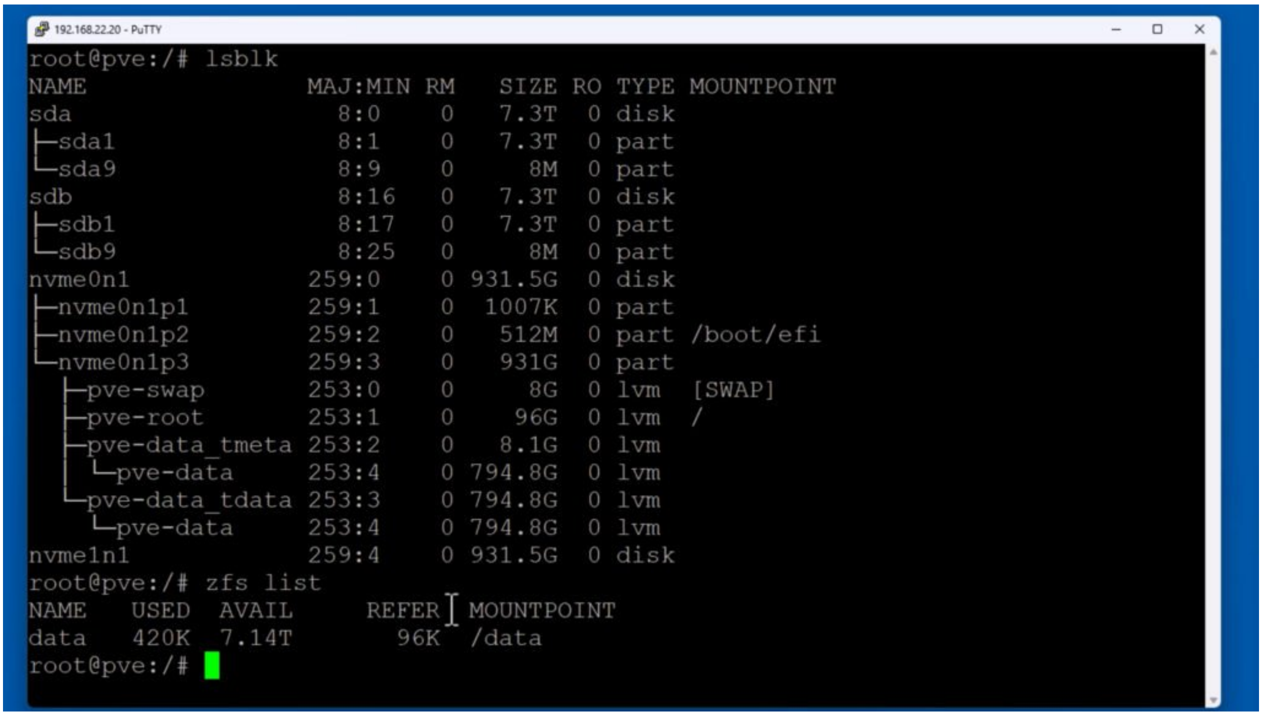
You can see it saw my two 8TB disks as *sda* and *sdb*. The other two 1TB disks were listed as *nvme0n1* and *nvme1n1*.

I created a mirror, called *data*, from the two 8TB disks using the zpool command like this

```
zpool create -f -o ashift=12 data mirror sda sdb
```

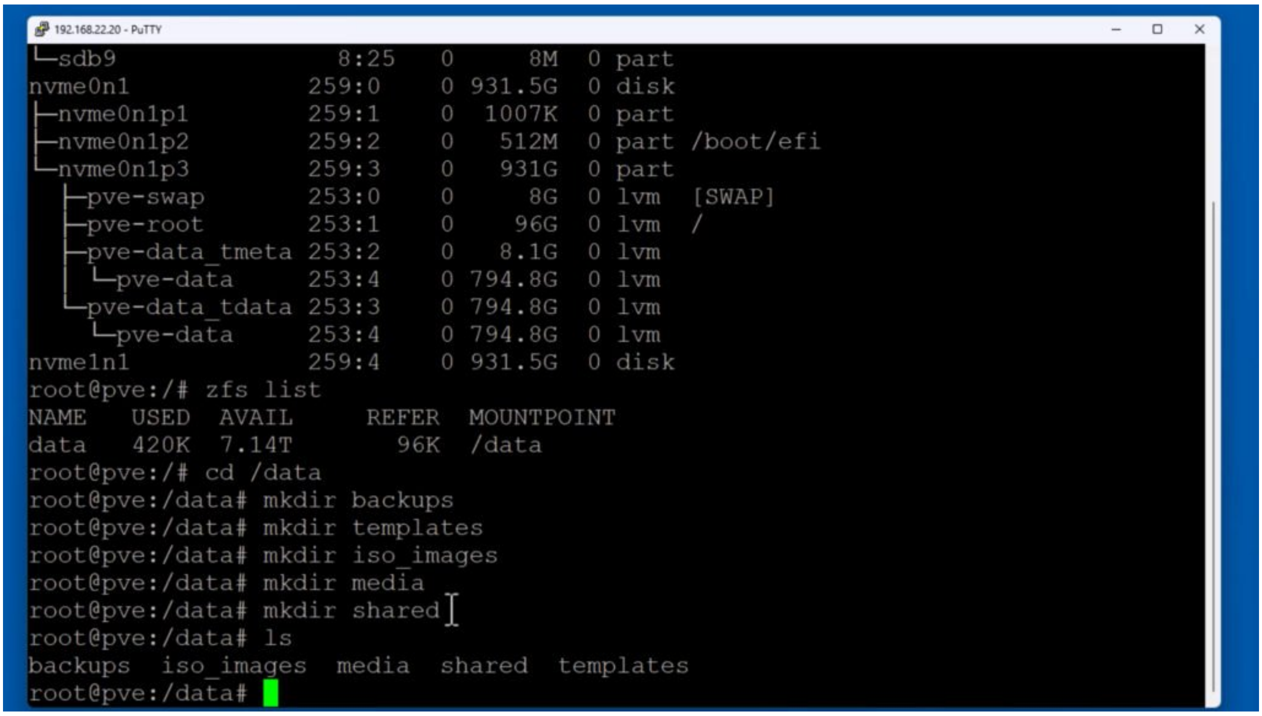
You'll need to replace *data* with whatever you want to call your storage volume, and substitute your own disk IDs in for *sda* and *sdb*.

If you now run the *zfs list* command you'll see your new mirror listed, and the mountpoint you gave it which in my case is /data



```
root@pve:/# lsblk
NAME                  MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                     8:0    0   7.3T  0 disk
├─sda1                  8:1    0   7.3T  0 part
└─sda9                  8:9    0     8M  0 part
sdb                     8:16   0   7.3T  0 disk
├─sdb1                  8:17   0   7.3T  0 part
└─sdb9                  8:25   0     8M  0 part
nvme0n1               259:0    0 931.5G  0 disk
├─nvme0n1p1           259:1    0  1007K  0 part
├─nvme0n1p2           259:2    0   512M  0 part /boot/efi
└─nvme0n1p3           259:3    0   931G  0 part
  ├─pve-swap          253:0    0     8G  0 lvm  [SWAP]
  ├─pve-root          253:1    0    96G  0 lvm  /
  ├─pve-data_tmeta    253:2    0   8.1G  0 lvm
  │ └─pve-data        253:4    0 794.8G  0 lvm
  └─pve-data_tdata    253:3    0 794.8G  0 lvm
    └─pve-data        253:4    0 794.8G  0 lvm
nvme1n1               259:4    0 931.5G  0 disk
root@pve:/# zfs list
NAME   USED  AVAIL     REFER  MOUNTPOINT
data   420K  7.14T       96K  /data
root@pve:/#
```

You can now *cd* to */data* and I created a bunch of directories in here to store the Proxmox stuff, as well as my media and shared files.

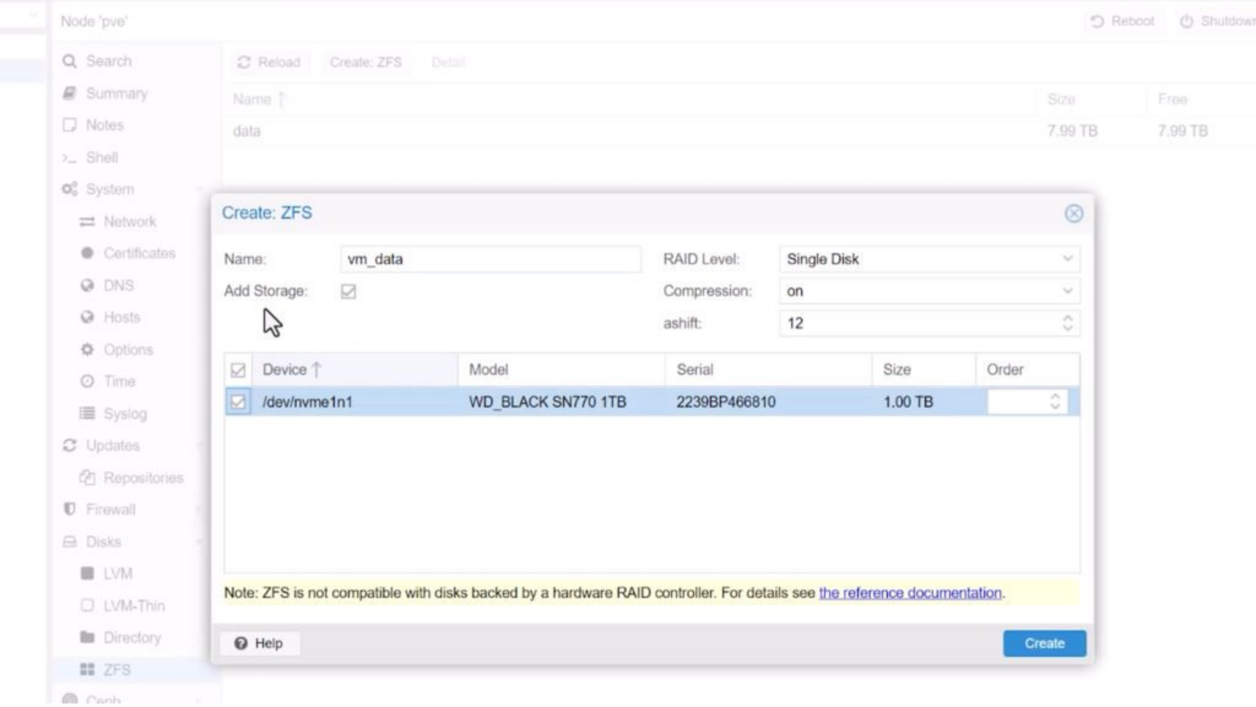


```
└─sdb9              8:25    0     8M  0 part
nvme0n1           259:0     0 931.5G  0 disk
├─nvme0n1p1       259:1     0  1007K  0 part
├─nvme0n1p2       259:2     0   512M  0 part /boot/efi
└─nvme0n1p3       259:3     0   931G  0 part
  ├─pve-swap      253:0     0     8G  0 lvm  [SWAP]
  ├─pve-root      253:1     0    96G  0 lvm  /
  ├─pve-data_tmeta 253:2    0   8.1G  0 lvm
  │ └─pve-data    253:4     0 794.8G  0 lvm
  └─pve-data_tdata 253:3    0 794.8G  0 lvm
    └─pve-data    253:4     0 794.8G  0 lvm
nvme1n1           259:4     0 931.5G  0 disk
root@pve:/# zfs list
NAME   USED  AVAIL     REFER  MOUNTPOINT
data   420K  7.14T       96K  /data
root@pve:/# cd /data
root@pve:/data# mkdir backups
root@pve:/data# mkdir templates
root@pve:/data# mkdir iso_images
root@pve:/data# mkdir media
root@pve:/data# mkdir shared
root@pve:/data# ls
backups  iso_images  media  shared  templates
root@pve:/data#
```

Next I set up my last disk, the second 1TB SSD, in the Proxmox user interface.

I went to the Proxmox node, then to *Disks*, then to *ZFS*. In here I saw the *data* volume I just created via the command line and clicked *Create*: ZFS in the top menu bar.

I now created a *Single Disk* storage pool called *vm_data* out of the remaining disk that was left.



Name ↑	Size	Free
data	7.99 TB	7.99 TB

Create: ZFS

Name: vm_data

Add Storage: ☑

RAID Level: Single Disk

Compression: on

ashift: 12

☑	Device ↑	Model	Serial	Size	Order
☑	/dev/nvme1n1	WD_BLACK SN770 1TB	2239BP466810	1.00 TB	

Note: ZFS is not compatible with disks backed by a hardware RAID controller. For details see the reference documentation.

If you check the *Add Storage* box, it will assign this disk to house your VM and container image files.

Finally, I had to tell Proxmox where to store the other bits and pieces, using the directories I created in the command line earlier.

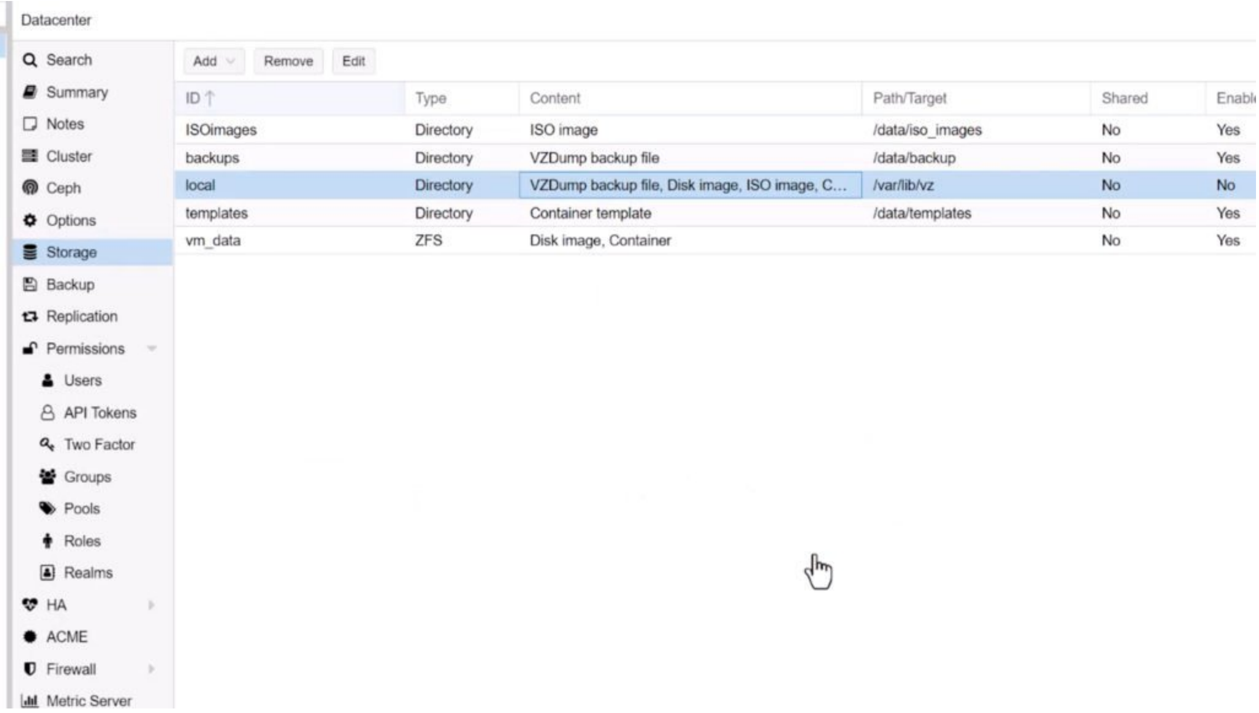
Go to the *Datacenter* node in the Proxmox web interface, and then to *Storage*.

Click *Add* in the top menu bar, and then choose *Directory*.

I created a directory called *templates*, specified it to store data in the */data/templates* directory I created earlier and then told it to store *Container Template* content in there.

I repeated this process for my *backups* and *iso images* as well.

To make sure I didn't accidentally create any VMs, containers or store any data on my OS Drive I deleted the local-lvm storage option and disabled the default local option as well.

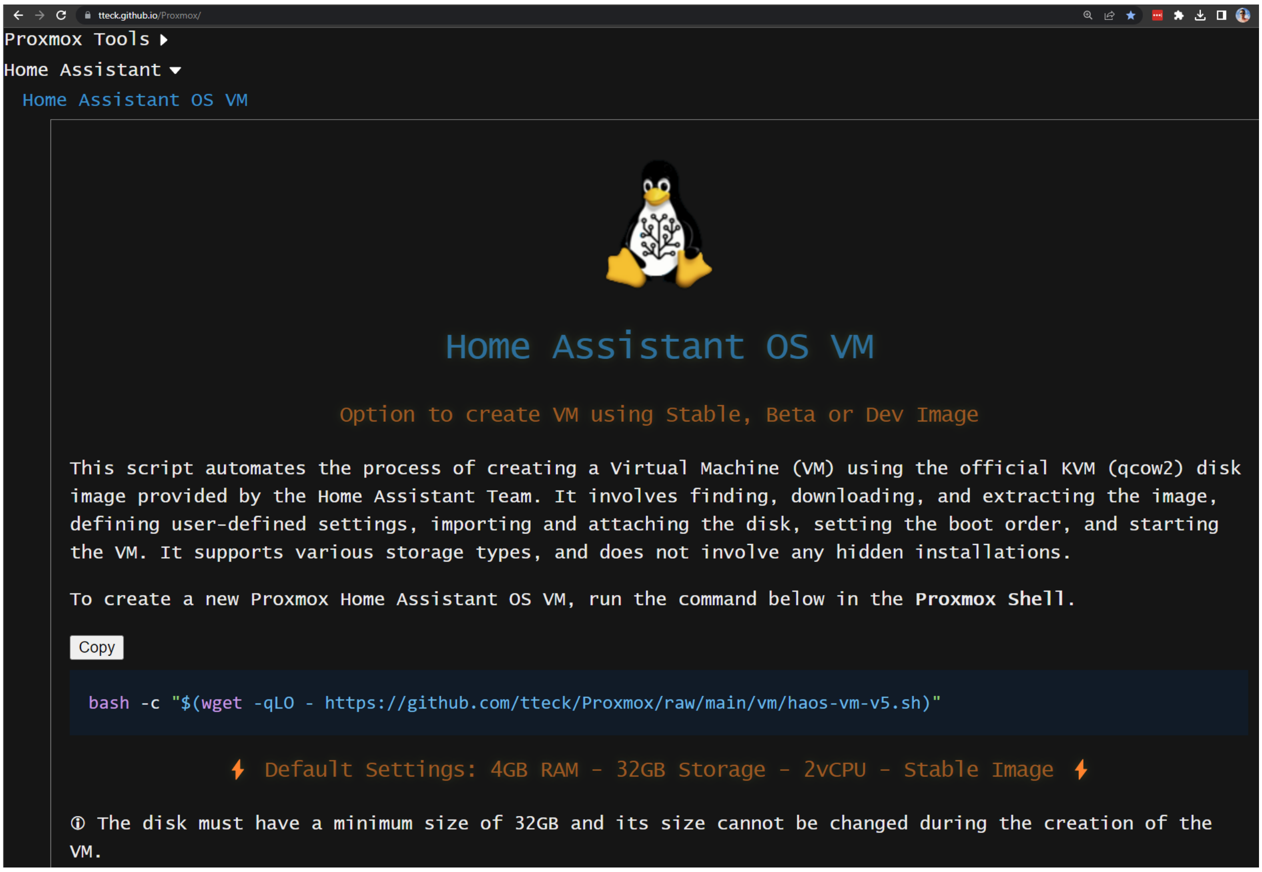


ID ↑	Type	Content	Path/Target	Shared	Enabl
ISOimages	Directory	ISO image	/data/iso_images	No	Yes
backups	Directory	VZDump backup file	/data/backup	No	Yes
local	Directory	VZDump backup file, Disk image, ISO image, C...	/var/lib/vz	No	No
templates	Directory	Container template	/data/templates	No	Yes
vm_data	ZFS	Disk image, Container		No	Yes

I'm a total newbie when it comes to Proxmox, so this may not be the most ideal way to set this up, but it's been working for me so far!

Installing Home Assistant on Proxmox

To install Home Assistant on Proxmox once again used the helper script library.



tteck.github.io/Proxmox/

Proxmox Tools ▸

Home Assistant ▾

Home Assistant OS VM

Home Assistant OS VM

Option to create VM using Stable, Beta or Dev Image

This script automates the process of creating a Virtual Machine (VM) using the official KVM (qcow2) disk image provided by the Home Assistant Team. It involves finding, downloading, and extracting the image, defining user-defined settings, importing and attaching the disk, setting the boot order, and starting the VM. It supports various storage types, and does not involve any hidden installations.

To create a new Proxmox Home Assistant OS VM, run the command below in the **Proxmox Shell**.

```
bash -c "$(wget -qLO - https://github.com/tteck/Proxmox/raw/main/vm/haos-vm-v5.sh)"
```

⚡ Default Settings: 4GB RAM - 32GB Storage - 2vCPU - Stable Image ⚡

ⓘ The disk must have a minimum size of 32GB and its size cannot be changed during the creation of the VM.

It was just a matter of SSHing into Proxmox again, and then running the script in the command line.