

Host Configuration

- [Base Install](#)
- [Networking \(out-of-date\)](#)
- [Common Software](#)
- [Setup PCI Passthrough](#)
- [Storage & Backups \(out-of-date\)](#)

Base Install

Operating System

Proxmox Virtual Environment 6.x

Configuration

Proxmox configuration has been transitioned to being automated by an [Ansible Role](#)

Networking (out-of-date)

Configuration

Because I don't want my main management interface to ever change names, I explicitly give it a name based on its MAC address.

```
# /etc/systemd/network/10-management-net.link
+ [Match]
+ MACAddress=70:85:c2:fe:4c:b7
+
+ [Link]
+ Name=man0
```

Bridges

Master	Bridge	IP Address	Gateway	Description
man0	vibr0	10.0.2.5/21	10.0.2.1	Main Interface (slower Realtek NIC)
enp6s0	--	--	--	Intel 10GbE SFP+ (used for PCI passthrough)

Common Software

Install fail2ban

This blocks connections that make repeated failed attempts to authenticate. SSH is covered by default which is what I am interested in, and I'll add additional config to similarly block too many repeated auth failures against the Proxmox web interface.

```
apt install fail2ban
```

```
# /etc/fail2ban/jail.local
+ [proxmox]
+ enabled = true
+ port = https,http,8006
+ filter = proxmox
+ logpath = /var/log/daemon.log
+ maxretry = 3
+ # 1 hour
+ bantime = 3600
```

Install sysfsutils

Sysfs is a virtual file system in Linux kernel 2.5+ that provides a tree of system devices. This package provides the program 'systool' to query it: it can list devices by bus, class, and topology.

In addition this package ships a configuration file /etc/sysfs.conf which allows one to conveniently set sysfs attributes at system bootup (in the init script etc/init.d/sysfsutils).

```
apt install sysfsutils
```

Install Netdata Monitoring

Install [Netdata](#) so that I can get a detailed view of system metrics. It will also be used as a datasource for [LXC / Conception / Prometheus](#) so I can look at metrics over a larger timeframe.

```
apt update
```

```
apt install curl
```

```
bash <(curl -Ss https://my-netdata.io/kickstart.sh)
```

Setup PCI Passthrough

See [PCI Passthrough](#) for more detail as to why I am doing these things.

Proxmox doesn't need a GPU, so blacklist the GPU and prepare it to be passed for a guest machine.

Enable Kernel Modules

```
# /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
+ vfio_pci
+ vfio
+ vfio_iommu_type1
+ vfio_virqfd
```

Bind `vfio-pci` Driver to Devices

```
# /etc/modprobe.d/vfio.conf
+ # AMD Radeon RX 560 [1002:67ff,1002:aae0]
+ alias pci:v00001002d000067FFsv00001458sd000022FFbc03sc00i00 vfio-pci
+ alias pci:v00001002d0000AAE0sv00001458sd0000AAE0bc04sc03i00 vfio-pci
+
+ options vfio-pci ids=1002:aae0,1002:67ff disable_vga=1
```

Rebuild `initramfs`

The `initramfs` needs to be rebuilt to reflect the changes I just did.

```
update-initramfs -u
```

Update Bootloader

Proxmox uses `systemd-boot` as the bootloader so I have to make sure to update the boot entries

Update Kernel Parameters

```
# /etc/kernel/cmdline  
- root=ZFS=rpool/ROOT/pve-1 boot=zfs  
+ root=ZFS=rpool/ROOT/pve-1 boot=zfs amd_iommu=on iommu=on video=efib:off  
pcie_acs_override=multifunction
```

Rebuild Bootloader Options

```
pve-efiboot-tool refresh
```

Storage & Backups (out-of-date)

Setup ZFS Scrub (Data Integrity)

Automate [ZFS scrubbing](#) so the data integrity on disks is actively monitored, repaired if necessary, and I'm alerted if there is a problem with my disks.

Create systemd Service/Timer ([source](#))

Create a simple systemd service template for scrubbing ZFS pools.

```
# /etc/systemd/system/zpool-scrub@.service
+ [Unit]
+ Description=Scrub ZFS Pool
+ Requires=zfs.target
+ After=zfs.target
+
+ [Service]
+ Type=oneshot
+ ExecStartPre=-/usr/sbin/zpool scrub -s %I
+ ExecStart=/usr/sbin/zpool scrub %I
```

Then create a systemd timer template for periodically running that service. I am running the scrub weekly, but semi-monthly or monthly would almost certainly be ok too.

```
# /etc/systemd/system/zpool-scrub@.timer
+ [Unit]
+ Description=Scrub ZFS pool weekly
+
+ [Timer]
+ OnCalendar=weekly
+ Persistent=true
+
```

```
+ [Install]
+ WantedBy=timers.target
```

Enable ZFS Scrub

```
systemctl daemon-reload
systemctl enable --now zpool-scrub@rpool.timer
```

Setup Sanoid/Synoid (Data Backup)

Run [Sanoid](#) for automating snapshots and Synoid for remote backups. Unfortunately this isn't available in repositories so you have to build it yourself. However the author makes it fairly simple.

Install ([source](#))

```
apt-get install build-essential debhelper dpkg-buildpackage libcapture-tiny-perl libconfig-inifiles-perl pv lzop
mbuffer
sudo git clone https://github.com/jimsalterjrs/sanoid.git
cd sanoid
ln -s packages/debian .
dpkg-buildpackage -uc -us
apt install ../sanoid*_all.deb
```

Configure Sanoid

I want to take hourly snapshots of both of my ZFS pools because sometimes I am not as careful or thoughtful as I should be about what I am doing at any given moment.

```
# /etc/sanoid/sanoid.conf
+ [template_proxmox]
+   frequently = 0
+   hourly = 24
+   daily = 7
+   weekly = 4
+   monthly = 1
+   yearly = 0
+   autosnap = yes
+   autoprune = yes
+
```

```
+ [rpool]
+   use_template = template_proxmox
+   process_children_only = yes
+   recursive = yes
+
+ [rpool/ROOT]
+   use_template = rpool
+   process_children_only = yes
+   recursive = yes
+
+ [rpool/data]
+   use_template = template_proxmox
+   weekly = 1
+   monthly = 1
+   process_children_only = yes
+   recursive = yes
```

Maybe this is a sin, but I'd like my snapshots to be in local time so I don't have to do the (admittedly simple) conversion in my head.

```
# /usr/lib/systemd/system/sanoid.service
[Service]
- Environment=TZ=UTC
+ Environment=TZ=EST
```

Configure Syncoid

I haven't decided where I want to replicate to yet so I haven't configured syncoid yet.