

# mini

Ancillary Proxmox box for running VMs accessible via a KVM switch

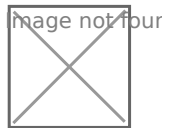
- [Physical Hardware](#)
- [Host Configuration](#)
  - [Base Install](#)
  - [Networking \(out-of-date\)](#)
  - [Common Software](#)
  - [Setup PCI Passthrough](#)
  - [Storage & Backups \(out-of-date\)](#)
- [VM / macOSAMD](#)

# Physical Hardware

## Basic Components

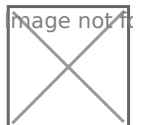
### Compute

#### Processor



[AMD Ryzen 3 3200G](#) - Economical CPU that has 4 cores and integrated graphics so a PCIe slot can be saved since a GPU isn't necessary.

<b>Cores / Threads</b>	4 / 4
<b>Base Frequency</b>	3.6 GHz
<b>Burst Frequency</b>	4.0 GHz
<b>Cache</b>	4MB L3 Cache
<b>TDP</b>	65W







#### Motherboard

[Asrock B450M Steel Legend](#) - Small form factor mATX board that is being reused from a prototyping project. Being an mATX board it is very limited in expansion.

<b>Manufacturer</b>	Asrock
<b>Model</b>	B450M Steel Legend
<b>CPU</b>	AMD AM4 Socket
<b>Chipset</b>	AMD Promontory B450
<b>Memory</b>	<ul style="list-style-type: none"><li>• 4x DDR4 DIMM supporting up to 64GB</li><li>• Dual channel memory architecture</li></ul>

Display	<ul style="list-style-type: none"><li>• 1x HDMI 1.4</li><li>• 1x DisplayPort 1.2</li></ul>
Networking	<ul style="list-style-type: none"><li>• 1x Realtek RTL8111H 1GbE LAN</li></ul>
Expansion	<ul style="list-style-type: none"><li>• 1x PCIe 3.0 x16</li><li>• 1x PCIe 2.0 x16</li><li>• 1x PCIe 2.0 x1</li></ul>
Storage	<ul style="list-style-type: none"><li>• 1x M.2 PCIe x4/x2</li><li>• 1x M.2 SATA x4/x2</li><li>• 4x SATA3</li></ul>
USB	<ul style="list-style-type: none"><li>• 1x USB 3.1 Gen 2 (Type-C)</li><li>• 1x USB 3.1 Gen 2 (Type-A)</li><li>• 4x USB 3.1 Gen 1 (Type-A)</li><li>• 2x USB 2.0</li></ul>

# Memory

Slot 1	 <p>image not found or type unknown</p>	<p>Corsair Vengeance LPX 8GB DDR4 2666MHz (1x8GB)</p> <ul style="list-style-type: none"><li>• 2Rx8 Dual Rank</li><li>• CAS Latency 16</li><li>• timing 16-18-18-35</li><li>• 1.2V</li></ul>
Slot 2	 <p>image not found or type unknown</p>	<p>Corsair Vengeance LPX 8GB DDR4 2666MHz (1x8GB)</p> <ul style="list-style-type: none"><li>• 2Rx8 Dual Rank</li><li>• CAS Latency 16</li><li>• timing 16-18-18-35</li><li>• 1.2V</li></ul>
Slot 3	 <p>image not found or type unknown</p>	<p>Corsair Vengeance LPX 8GB DDR4 2666MHz (1x8GB)</p> <ul style="list-style-type: none"><li>• 2Rx8 Dual Rank</li><li>• CAS Latency 16</li><li>• timing 16-18-18-35</li><li>• 1.2V</li></ul>
Slot 4	 <p>image not found or type unknown</p>	<p>Corsair Vengeance LPX 8GB DDR4 2666MHz (1x8GB)</p> <ul style="list-style-type: none"><li>• 2Rx8 Dual Rank</li><li>• CAS Latency 16</li><li>• timing 16-18-18-35</li><li>• 1.2V</li></ul>







# Case


Fractal Design - Define Mini C (Blackout) - A fantastic case with an attractive minimalistic design that in a mATX form factor.

Manufacturer	Fractal Design
Model	Define Mini C (Blackout)
Features	<ul style="list-style-type: none"><li>• Sound dampening panels</li><li>• Excellent build quality</li><li>• 2x 3.5" Drive Bays</li><li>• 2x 2.5" Drive Bays</li></ul>

# Storage

#	Capacity	Interface	Type	Manufacturer & Model	Speed
1x 	256GB	NVMe	SSD	Western Digital Black WDS256G1X0C	PCIe 3.0 x2
2x 	512GB	SATA	SDD	Crucial MX100	SATA3 6.0Gb/s
1x 	1TB	SATA	HDD	Western Digital WD10EZEX	SATA3 6.0Gb/s
1x 	1TB	SATA	HDD	Seagate ST1000DM003	SATA3 6.0Gb/s

# Cooling

CPU 	AMD Wraith Spire
---	------------------

Case (Front)	 image not found or type unknown	Noctua NF-A14 PWM 140mm
Case (front)	 image not found or type unknown	Noctua NF-F12 PWM 120mm
Case (rear)	 image not found or type unknown	Noctua NF-F12 PWM 120mm



# Power Supply

Manufacturer	EVGA
Model	SuperNOVA 550 G2
Features	<ul style="list-style-type: none"><li>• 550W</li><li>• fully module</li></ul>

# UPS

n/a

# Add-On Cards

PCI 3.0 x16	 Radeon RX 560 Gaming OC 4G (rev. 1.0) <ul style="list-style-type: none"><li>• PCIe Gen3 x8</li></ul>
PCI 3.0 x1	 Inateck 4 Ports PCIe to USB 3.0 <ul style="list-style-type: none"><li>• PCIe Gen3 x1</li></ul>

**PCI 2.0 x16**



**10Gtek Intel 82599ES SFP+ PCIe x8**

- PCIe Gen3 x8
- SFP+ 10GbE port
- SR-IOV

# Host Configuration

Host Configuration

# 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	vmbr0	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 [Netadata](#) 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=efifb: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/Syncoid (Data Backup)

Run [Sanoid](#) for automating snapshots and Syncoid 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.



# VM / macOSAMD

## Description

This VM is for running macOS via dedicated hardware so I have something faster than my laptop.

## Configuration

### Resources

Hostname	CPU	Memory
MiMac	12 vCPU	16GB

### Storage

Disk	Controller	Size	Purpose
local-zfs:vm-100-disk-1	ide0	1M	NVRAM
local-zfs:vm-100-disk-0	virtio	200MB	EFI boot loader
local-zfs:vm-100-disk-2	virtio	1TB	boot disk

### PCI Passthrough

Name	BDF	Settings
Fresco USB 3.0 Controller	04:00	n/a
AMD RX 560 GPU	08:00	pcie=1,x-vga=1
Intel 10GbE SFP+ NIC	06:00	n/a

### Networking

# Interfaces

ID	Name	Bridge	IP Address	Comments
eno0	n/a	vmbr0	10.0.2.5/21 (DHCP)	1GbE
eno1	n/a	--	10.0.10.5/24 (manual)	10GbE

## Configuration

Because we want macOS to route all traffic destined for [blackbox.hermz](#) over our 10GbE network we need to adjust our routing table to redirect all traffic to [LXC / Routeman](#) first.

Knowing that blackbox.hermz has ip addresses `10.0.2.2`, and `10.0.2.3` and all of its services run with `10.0.4.x` we can easily setup some updated routes.

```
# 10.0.2.2 (main interface) and 10.0.2.3 (admin interface)
ip route add 10.0.2.2/31 dev en1 via 10.0.10.6

# 10.0.4.x (services running on blackbox)
ip route add 10.0.4.0/24 dev en1 via 10.0.10.6
```

This works perfectly except it isn't persistent.