

Home Lab Overview

This environment reflects my hands-on experience with deploying, securing, and managing modern infrastructure components. It also provides a controlled space to test updates, troubleshoot issues, and evaluate new technologies before applying them in production environments.

Whether I'm fine-tuning service availability, setting up HA, or exploring integrations between tools, my home lab is where curiosity meets practical implementation.

- [Description](#)
- [Virtualization: Proxmox VE](#)
- [Home Lab Hardware Specifications](#)

Description

Welcome to my Home Lab documentation! This book serves as a centralized knowledge base for the design, configuration, and ongoing management of my personal IT infrastructure. It includes detailed notes, guides, troubleshooting steps, and configurations for the various technologies and services I'm running in my lab environment.

The purpose of this home lab is to deepen my understanding of enterprise technologies, experiment with new tools, and simulate real-world IT scenarios. From virtualization and networking to security, automation, and containerization, this space is where I build, break, and learn.

Whether it's setting up a Proxmox cluster, configuring a pfSense firewall, deploying Docker containers, managing endpoints with Tanium, or monitoring systems with tools like Zabbix and Wazuh, everything I do here helps sharpen my skills and stay current with the latest industry trends.

This book documents the architecture, configuration, and ongoing management of my personal home lab — a self-hosted environment that mirrors real-world enterprise infrastructure. It serves both as a technical reference for myself and as a showcase of my skills and experience for recruiters and industry professionals.

My lab is designed for continuous learning and hands-on experience with systems administration, virtualization, networking, endpoint management, security, and automation.

?? Core Platforms & Services

- **Virtualization:** Proxmox VE for efficient VM management and high availability
- **Containerization:** Docker Compose for deploying and orchestrating containerized applications
- **Networking & Security:** pfSense for firewall, routing, and VPN services
- **Identity & Access Management:** Authentik for centralized authentication and single sign-on (SSO)
- **Monitoring & Management:**
 - **Zabbix** for infrastructure monitoring and alerting
 - **Tanium** for real-time endpoint visibility and security
 - **SCCM (System Center Configuration Manager)** for imaging, patching, and software deployment

? Windows Infrastructure:

- **Active Directory Domain Services (AD DS)** for centralized authentication
- **DNS, DHCP, and Group Policy** to manage network configuration and user environments

- **WSUS** for Windows Update Services
 - **SCCM (System Center Configuration Manager)** for endpoint management, imaging, and software deployment
-

? Productivity, Collaboration & Documentation

- **Nextcloud** for self-hosted file sync, sharing, and collaboration
 - **Mattermost** as an internal messaging and communication platform
 - **Mailcow** as a complete mail server suite for email testing and administration
 - **Snipe-IT** for asset and license tracking
 - **Paperless-NGX** for document digitization and automation
 - **BookStack** as a structured internal wiki and knowledge base
 - **Audiobookshelf** for personal media streaming
-

This lab reflects my passion for IT, my commitment to self-improvement, and my ability to design, deploy, and manage production-grade infrastructure. It provides a sandbox for testing updates, validating new tools, and simulating enterprise-level scenarios — all while refining best practices in security, automation, and documentation.

Virtualization: Proxmox VE

This page outlines how I installed, configured, and currently use **Proxmox VE** as the foundation of my home lab virtualization platform. It includes steps for installing Proxmox, setting up storage, creating virtual machines and LXC containers, and using templates for rapid deployment.

? 1. Proxmox VE Installation

? Requirements:

- CPU with virtualization support (Intel VT-x / AMD-V)
- 8GB+ RAM (16GB recommended)
- SSD/HDD storage
- Bootable USB with Proxmox ISO

? Steps:

1. Download the latest ISO: <https://www.proxmox.com/en/downloads>
 2. Create a bootable USB drive with Rufus or the `dd` command.
 3. Boot from USB and follow the installation wizard:
 - Accept EULA
 - Select target disk (use ZFS if using multiple disks or for snapshots)
 - Set country, time zone, and keyboard
 - Set root password and email
 - Assign a static IP, hostname (e.g., `proxmox.local`), and gateway
 4. Reboot and access the web GUI:
`https://<proxmox-ip>:8006`
-

?? 2. Initial Configuration

? Post-Install Checklist:

- Update system:

```
apt update && apt full-upgrade -y
```

- (Optional) Remove subscription nag:

```
sed -i.bak "s|data.status != 'Active'|false|" /usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js
```

```
systemctl restart pveproxy
```

- Add no-subscription repo in `/etc/apt/sources.list.d/pve-enterprise.list`:

```
deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

- Add SSH key for secure login
- Set up storage (local-lvm, additional disks via ZFS or ext4)

? 3. Creating a Virtual Machine (VM)

Steps:

1. Upload ISO to `local` or `local-lvm` storage
2. Go to **Create VM**
3. General:
 - Name: `ubuntu-server`
4. OS:
 - Select ISO image
5. System:
 - Use default (UEFI or BIOS)
6. Hard Disk:
 - Choose `local-lvm`, disk size
7. CPU & Memory:
 - 2 cores, 4GB RAM (example)
8. Network:
 - Select `vmbbr0` (bridge to LAN)
9. Confirm and finish

Post-VM:

- Start VM
- Install OS (e.g., Ubuntu, Debian, Windows)
- Set static IP, install guest agent:

```
apt install qemu-guest-agent  
systemctl enable --now qemu-guest-agent
```

? 4. Creating LXC Containers

LXC Benefits:

- Lightweight, fast
- Ideal for services like DNS, Pi-hole, NGINX, Git, etc.

Steps:

1. Go to **Create CT**
2. General:
 - Set hostname and password
3. Template:
 - Download from Proxmox template library
4. Root Disk:
 - Choose size and storage
5. CPU & Memory:
 - Allocate as needed
6. Network:
 - Assign static IP or DHCP
7. Confirm and finish

Example Services via LXC:

- Pi-hole
 - NGINX reverse proxy
 - Node Exporter
 - Lightweight Docker host (with nesting enabled)
-

? 5. Using and Creating Templates

VM Templates:

1. Create and configure a VM (e.g., Ubuntu minimal)
2. Shutdown VM
3. Convert to template:

```
qm template <vmid>
```

LXC Templates:

1. Download from **CT Templates** in Proxmox GUI
2. Or upload custom `.tar.gz` template to:

```
/var/lib/vz/template/cache/
```

?? 6. Best Practices & Tips

- **Backups:** Schedule daily/weekly backups to external storage (e.g., NFS or USB)
- **Snapshots:** Take before major updates or reconfigurations

- **Networking:** Use VLANs and firewall rules via pfSense
- **High Availability (Optional):** Use Proxmox cluster for HA with shared storage
- **Resource Monitoring:** Integrate with **Zabbix** or built-in usage graphs

Home Lab Hardware Specifications

This section outlines the physical hardware powering my virtualization environment. It includes both compute and network components that support VM hosting, containerization, and service reliability.

?? Primary Host (Proxmox Node)

pve

Primary Node for all Production VMs and CTs

- **CPU:** Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz (8 Cores / 16 threads)
- **RAM:** 80GB DDR4 (2x32GB & 2x8GB DDR4 UDIMM)
- **Storage:**
 - 500GB NVMe SSD (Proxmox OS + Storage)
 - 2TB NVMe SSD (Storage)
- **Network:** 1000 Mbps Ethernet NICs
- **Power Supply:** 650W Gold-rated PSU
- **Chassis:** Dell Precision 3450 Small Form Factor

pve1

Node for Tanium Appliance (TanOS)

- **CPU:** Intel(R) Core(TM) i7-10610U CPU @ 1.80GHz (4 Cores)
- **RAM:** 64GB DDR4
- **Storage:** 500GB NVMe SSD (Proxmox OS + Storage)
- **Network:** 1000 Mbps Ethernet NICs
- **Device:** Dell Latitude 5410

pve2

Node for testing endpoints

- **CPU:** Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz (4 Cores)
- **RAM:** 16GB DDR3
- **Storage:** 500GB SATA SSD (Proxmox OS + Storage)
- **Network:** 1000 Mbps Ethernet NICs
- **Device:** ThinkPad L460

? Networking Equipment

- **Firewall / Router:** pfSense running on Protectli VP2410
- **Switch:** TP-Link TL-SG108E (8-Port Gigabit Smart Switch)
- **Access Point:** Ubiquiti UniFi U6-Lite (Wi-Fi 6)

?? Storage & Backup

- **NAS:** TrueNAS Core running on Mini-PC with:
 - Intel i5 CPU
 - 16GB RAM
 - 2x 4TB HDDs (ZFS mirror)
- **External Backup:** 2TB USB 3.0 external drive (rotated offsite weekly)

“ *Note:* This setup is optimized for cost-efficiency, power savings, and scalability, allowing me to run dozens of services simultaneously in a virtualized and containerized environment.