Why Every IT Professional Needs a Home Lab

April 2, 2026 / Gavin Jackson

homelab proxmox virtualization career infrastructure learning docker linux backup pdm pbs

Over my 25+ years in IT, I have learned that the difference between a good technologist and a great one often comes down to one thing: curiosity. The desire to tinker, break things, fix them, and understand how systems work at a fundamental level. And there is no better way to feed that curiosity than building and maintaining a home lab.

This is not just about having toys to play with. A home lab is an investment in your career, a sandbox for learning, and frankly - one of the first things I look for when hiring.

My Setup

My current home lab runs on an old HP Z440 workstation that I picked up second-hand. It is got 64GB of RAM and runs Proxmox VE - a free, open-source hypervisor that punches well above its weight.

This single box runs:

- **Media server** - Plex for the family's movies and TV shows
- **Development environments** - Isolated VMs and containers for different projects
- **Test servers** - Where I spin up technologies I need to learn for work
- **Security tooling** - Vulnerability scanners, SIEM experiments, network monitoring
- **Random experiments** - That new database I read about, the container orchestrator everyone is talking about, the CI/CD tool I want to evaluate

The beauty of this setup is flexibility. Need a Linux box to test a deployment script? Spin up an LXC container in 30 seconds. Want to try out a new configuration management tool? Clone a VM and go to town. Breaking things is encouraged - that is how you learn.

Why Proxmox?

I have run VMware vSphere, Hyper-V, and various bare-metal Linux setups over the years. Proxmox is genuinely impressive for what it offers at zero cost:

- **KVM virtualization** for full VMs with excellent performance
- **LXC containers** for lightweight Linux instances that share the host kernel
- **ZFS support** built-in for storage management
- **Web-based management** that is actually good
- **Clustering** if you want to add more nodes later
- **No licensing headaches** - truly open source

Installation is straightforward: download the ISO, boot it, answer a few questions, and you are done. Within an hour, you will have a working hypervisor with enterprise-grade features.

Community Scripts: The Secret Weapon

One of the best things about the Proxmox ecosystem is the Community Scripts project. This is a collection of helper scripts that automate the deployment of common applications and services.

Want a Pi-hole DNS server? One command. Home Assistant? One command. Nextcloud, Jenkins, GitLab, PostgreSQL, Redis, WireGuard - they have got you covered.

These scripts handle:

- Creating the container or VM with appropriate resources
- Installing and configuring the software
- Setting up proper networking
- Applying security hardening

What used to take hours of manual configuration now takes minutes. It is honestly one of the best community resources I have seen in the infrastructure space.

PVEScripts-local

For those managing multiple Proxmox nodes or a small cluster, the PVEScripts-local package extends this functionality with local cluster management capabilities. You can:

- Deploy containers across multiple nodes from a central location
- Maintain consistent configurations across your environment
- Automate backups and migrations
- Manage network and storage settings at scale

This turns a collection of individual Proxmox servers into a manageable platform - exactly what you need as your home lab grows.

The Broader Proxmox Ecosystem

Proxmox VE is just the beginning. The Proxmox ecosystem includes two other products that are worth knowing about, especially as your home lab grows beyond a single node.

Proxmox Datacenter Manager (PDM)

As your home lab expands to multiple physical machines, managing them individually becomes tedious. This is where Proxmox Datacenter Manager comes in.

PDM provides a centralized web interface to manage multiple Proxmox VE nodes from a single dashboard. Think of it as a lightweight alternative to VMware vCenter - without the licensing costs.

Key features include:

- **Centralized management** - View and control all your Proxmox nodes from one interface
- **Live migration** - Move running VMs between physical hosts with zero downtime (requires shared storage)
- **Resource monitoring** - See CPU, memory, and storage utilization across your entire cluster
- **Centralized authentication** - Single sign-on across all managed nodes

For home lab enthusiasts, this means you can start with one machine and seamlessly add more as your needs grow. When you find that old workstation on eBay, just install Proxmox VE, join it to your PDM instance, and you have instantly expanded your capacity. Moving VMs between nodes is as simple as a drag-and-drop operation - perfect for load balancing or performing hardware maintenance without interrupting services.

Proxmox Backup Server (PBS)

Backups are non-negotiable, and Proxmox Backup Server is Proxmox's purpose-built solution for backing up your virtual machines and containers.

PBS is particularly interesting from a technical perspective because of how it handles backups:

Snapshot-based backups: PBS uses QEMU's live snapshot capability to create consistent backups of running VMs without downtime. For containers, it uses LXC snapshot functionality. This means you can back up production workloads without service interruption.

Deduplication and incremental backups: PBS implements client-side deduplication. When you back up a VM, it only transfers the changed blocks since the last backup. This dramatically reduces storage requirements and backup times. The deduplication works across all backups in the datastore - if you have ten similar VMs, you only store the unique blocks once.

Encryption: Backups can be encrypted client-side before transmission, ensuring your data remains secure even if the backup storage is compromised.

File-level restore: While backups are snapshot-based, PBS allows you to browse and restore individual files from within the backup images. Accidentally deleted a config file? You can mount the backup and extract just that file without restoring the entire VM.

Retention policies: PBS handles complex retention schedules (grandfather-father-son) automatically, keeping daily backups for a week, weekly backups for a month, and monthly backups for a year - all configurable to your needs.

For a home lab, PBS can run as a VM or LXC container on your existing Proxmox host, backing up to a separate disk, a NAS, or even external USB storage. It is another example of enterprise-grade features available at zero cost.

What I Use My Lab For

Systems Administration

The best sysadmins I know run complex home setups because it is the only way to stay current. My lab lets me:

- Practice Linux administration skills
- Test backup and restore procedures
- Experiment with monitoring and alerting stacks
- Understand networking at a deeper level
- Try security hardening techniques

Software Development

Having dedicated development environments that mirror production (or experiment with different architectures) is invaluable:

- Isolated test databases
- CI/CD pipeline experimentation
- Microservices architecture testing
- Performance testing without cloud bills

Cyber Security

Security is increasingly everyone is responsibility. My lab includes:

- Vulnerability scanning with OpenVAS
- Network monitoring with Zeek and Suricata
- Log aggregation experiments
- Penetration testing practice environments
- Certificate management and PKI experimentation

What I Look for When Hiring

Here is something I tell every junior person who asks for career advice: **build a home lab.**

When I am interviewing candidates - especially for infrastructure, DevOps, or systems roles - one of the first questions I ask is what they run at home. The answer tells me a lot:

- **Nothing, I just use cloud services** - They have missed an opportunity to build deep understanding
- **I have got a Raspberry Pi running Pi-hole** - Good start, they are curious
- **I run a Proxmox cluster with VMs for testing and a NAS for storage** - Now we are talking

A candidate who maintains a home lab demonstrates:

1. **Genuine interest in technology** - They do not just do this for a paycheck; they actually care
2. **Self-directed learning** - They take initiative to learn outside formal structures
3. **Hands-on experience** - They have touched real systems, not just read documentation
4. **Problem-solving skills** - Home labs break constantly; maintaining one builds resilience
5. **Breadth of knowledge** - They understand how systems interconnect

This has been a constant in my 25+ year career. The best people I have worked with - the ones who really understand their craft - almost always have some kind of home setup. It might be modest, it might be elaborate, but it is there.

The AI Revolution and Home Labs

Here is where things get really interesting. The intersection of AI tooling and home lab management is opening up possibilities that did not exist even a year ago.

Tools like OpenClaw (which I have written about previously) bring AI assistants directly into your terminal. This changes the game for home lab administration:

- **Natural language infrastructure management** - Create a new Ubuntu container with 4GB RAM and install Docker
- **Intelligent troubleshooting** - AI can analyze logs, suggest fixes, and even implement them
- **Configuration generation** - Describe what you want to build, get working configs
- **Security analysis** - Automated review of your setup for common vulnerabilities

I am increasingly using AI to help manage my home lab - from writing Ansible playbooks to analyzing performance metrics to suggesting architecture improvements. The productivity gains are substantial, and the learning experience is fascinating.

I will be blogging more about this intersection in future posts. The combination of accessible virtualization (Proxmox), community automation (Community Scripts), and AI assistance is creating a new paradigm for how technical professionals build and manage their personal infrastructure.

Getting Started

If you do not have a home lab yet, here is my advice:

Start Small

You do not need enterprise hardware. That old desktop in your garage? Perfect. A used workstation from eBay? Even better. A Raspberry Pi 4 or 5? Absolutely viable for many use cases.

Start with Proxmox

Download the ISO, install it on whatever hardware you have, and spend a weekend clicking around. Create a VM. Create an LXC container. Break something and fix it.

Pick a Project

Do not just build a lab for the sake of it. Have a goal:

- Set up a Pi-hole DNS server for your network
- Run a local Git server
- Build a media server for your movies
- Create a development environment for learning

Document Everything

Start a blog. Write notes. Create diagrams. Teaching others (or your future self) solidifies your learning and creates a record of your progress.

Conclusion

A home lab is not just a collection of hardware and software. It is a statement about who you are as a technologist. It says: I am curious. I am self-motivated. I learn by doing.

In an industry that changes as fast as ours, those qualities are priceless. Whether you are just starting out or have been in the game for decades, a home lab will make you a better technologist.

So go dust off that old computer, download Proxmox, and start building. Your future self will thank you.

Downloaded from https://www.gavinj.net/post/importance-of-home-lab-it-professional
Generated October 10, 2026. Copyright Gavin Jackson. All rights reserved.