Eleven Servers, One Hammer, and a Serious Look at Proxmox

September 10, 2026 / Gavin Jackson

proxmox vmware virtualization infrastructure linux veeam ansible

I have a slightly crazy idea: perhaps the next hypervisor we run at work will be Proxmox.

Three months ago, I would have been much less confident saying that. We were giving a graduate a useful research project and taking a sensible look around before the next VMware decision. I expected to learn something. I did not expect to come away thinking we might have found a serious contender.

Our environment sits in what I think of as the infrastructure sweet spot. Eleven physical servers, hundreds of cores across dozens of sockets, many terabytes of RAM, and more than a hundred virtual machines. Big enough that reliability, automation and recovery matter every day. Small enough that a small infrastructure team can still understand the whole thing.

The fleet is ageing, but there is significant capacity in those clusters. We run vSphere 8 with iSCSI-attached storage, and Veeam handles both on-site and off-site backups. Our requirements are fairly ordinary: run the servers, connect them to the right networks, move them when necessary, and recover them when something goes wrong. We have no software-defined networking and no particular desire to acquire it.

Then our Broadcom representative told us we were on a tight timeframe to start thinking about an upgrade to VMware Cloud Foundation 9.

That was a good reason to look up from the renewal paperwork.

What the VCF 9 conversation actually means

There are several decisions bundled into that conversation: the support deadline, the product we buy, the hardware it will support, and the work required to get there.

The published end of general support for ESXi 8.0 and vCenter Server 8.0 is **11 October 2027**, as shown in NEC's VMware support lifecycle table. Our contract renewal date is a separate deadline. As of September 2026, there is a planning window, but a year can disappear quickly when it includes procurement, hardware validation and migrating more than a hundred VMs.

VCF is VMware's broader private-cloud platform, encompassing the virtualisation layer alongside storage, networking, operations and automation capabilities. For us, the question is how much of that platform earns its place in an eleven-server environment. Broadcom's current VCF and vSphere Foundation comparison still documents VVF as a separate, narrower offering. We need to establish which products and upgrade rights are available to our account before treating VCF as the only possible destination.

The full VCF conversion workflow also brings operational changes. Broadcom's convergence guide describes additional management components, an NSX requirement for that workflow, and a move from baseline-based to image-based host lifecycle management. Even a team using ordinary VLANs needs to understand what the proposed deployment adds to its maintenance workload.

The ageing hardware makes this more than a software-budget exercise. Broadcom explicitly warns that hardware supported on vSphere 8 is not automatically supported on version 9. We need to check the actual servers, processors, network and storage controllers, firmware and drivers against the target release. An upgrade that brings forward a hardware refresh has a very different cost from one that uses the capacity we already own.

Storage deserves its own design check. The VCF comparison's storage table lists VMFS over iSCSI as supplemental storage, while Fibre Channel and NFSv3 can also serve as principal storage. That does not make iSCSI universally unsupported, but it means we cannot assume our existing layout drops straight into a full VCF deployment.

There is a licensing workflow to maintain as well. Version 9 introduces a rolling 180-day usage-reporting and license-refresh cycle; letting the local license file expire can affect management operations even during a multiyear subscription. Broadcom documents connected and disconnected workflows, so this is something to design and monitor, rather than a requirement to put every host directly on the internet.

Add a compatible Veeam release and a tested upgrade and rollback sequence, and staying with the same vendor still involves engineering work.

What has happened to the price?

Our last renewal was about two and a half years ago, around early 2024. Broadcom's move away from perpetual licensing was already under way then. The important change is the commercial model: subscription bundles, physical-core licensing and the possibility of paying for a wider platform than the one we currently use. The date alone does not tell us whether our last renewal was legacy support or already reflected the new model.

The core count deserves particular attention on older servers. Broadcom's VCF terms require a **minimum of 16 licensed cores per physical processor**. A two-socket server with eight cores per socket therefore counts as 32 licensed cores, despite containing only 16 physical cores. Hundreds of cores spread across dozens of sockets need a proper inventory before anyone starts multiplying prices. Broadcom's VCF program terms describe that counting rule.

There is no single public price that establishes what our renewal will cost. There are, however, useful recent examples. A publicly posted Insight quotation for West Virginia, dated 4 November 2025, prices **740 VCF cores at US$228.42 per core per year** on a three-year quote with annual billing. That is **US$169,030.80 annually for VCF alone**, covering annual periods through November 2028. It is one negotiated public-sector quote, but it gives the discussion some scale.

More recently, US reseller AceMQ published these figures from proposals it issued in 2026, in an article updated on 4 September:

Product	Quoted size and commitment	Quoted price per core per year
vSphere Foundation	72 cores, three years, billed annually	$119.08
vSphere Foundation	72 cores, three years, billed annually	$126.29
VMware Cloud Foundation	96 cores, one year	$344.40

These are the reseller's reported customer quotes. The dollar amounts are reproduced as printed; unlike the Insight document, AceMQ's article does not explicitly identify the currency. They involve different customers and commitments, so they do not establish a like-for-like VCF premium or a percentage increase since our last renewal. They do show why the bundle and contract term deserve as much scrutiny as the discount.

For our comparison, the useful number will be the annual cost of a supported option for our actual hosts, with the term, support, currency and tax treatment made explicit. Then we can put hardware, migration effort and ongoing administration beside it. An internet horror story about somebody else's renewal cannot do that calculation for us.

Enter Thor, and some servers destined for disposal

For the last three months, we have had a graduate rotating through our small infrastructure team. Let's call him Thor.

Part of his project was to research the current on-premises hypervisor landscape. His research covered Microsoft Hyper-V, Nutanix AHV, HPE Morpheus VM Essentials and Proxmox VE. That gave us enough context to choose something to put our hands on. Proxmox got the first trial.

We had some old Elasticsearch servers waiting for somebody to arrange disposal. Thor turned three of them into a Proxmox cluster and built a separate physical Proxmox Backup Server.

The cluster runs Proxmox VE 9.2, with pve-manager 9.2.18.

Repurposing those machines gave us a practical way to test the things our team actually needs. The results have been much more interesting than I expected.

The things we have actually tested

Linux and Windows guests. Both worked in the trial. Proxmox uses QEMU/KVM for full virtual machines, and its VM documentation covers both Linux and Windows, including the VirtIO devices and drivers used for efficient disk and network access. We can bring a mixed server environment to this platform. Application support and guest configuration still need checking for each production workload.

Importing from ESXi. We successfully connected Proxmox to a running ESXi server and imported virtual machines. Being able to work directly with the existing host makes migration feel much more approachable than a pile of manual disk conversions.

There is an important distinction here: the ESXi **host** remains running, but Proxmox's documented importer requires the source **VM** to be shut down for a consistent import. Its optional live-import mode can start the destination VM while the remaining disk data transfers; that still involves a cutover. Our result demonstrates a usable import route, without claiming uninterrupted migration between hypervisors. The official migration guide explains the process and preparation required.

Live migration inside the Proxmox cluster. This worked too, and for this test we had **no shared attached storage**. Proxmox can transfer locally stored VM disks as part of moving a running guest between nodes. That is a useful capability to see working on the hardware in front of you.

The migration documentation sets out the conditions, including compatible CPUs and movable guest resources. Moving the disks means more traffic than migrating a guest whose storage is already shared. Production-sized workloads will give us a better picture of migration time and service impact.

Our existing VLANs. Thor configured a trunk port on the switch and successfully tested connectivity to different VLANs. Proxmox supports VLAN-aware Linux bridges and per-VM network tags, which fit naturally with the tagged networks we already operate. That meets a fairly fundamental requirement without giving us a new networking architecture to manage.

Ansible and templates. We tested building Ubuntu 26.04 virtual machines from Proxmox templates using Ansible. Repeatable provisioning matters to a small team: the hypervisor needs to fit the way we build and maintain servers.

The platform exposes the relevant building blocks. Ansible's `community.proxmox.proxmox_kvm` module supports VM management and cloning, and Proxmox also documents cloud-init templates for supplying initial guest configuration. Our template-and-Ansible test gives us a useful foundation for extending that workflow.

Proxmox Backup Server. We tested the free backup server on its own physical machine. PBS is free to download and use, with paid subscriptions available for enterprise updates and support. Dedicated hardware is also Proxmox's recommended deployment approach.

Its documented capabilities include incremental transfers, deduplication, compression, encryption, integrity verification and synchronisation to another backup server. Those make it worth evaluating seriously, particularly for off-site recovery, and give us plenty to explore as we work through the recovery tests.

Proxmox Datacenter Manager. We tested this as well. It provides a consolidated view of independent Proxmox clusters, nodes and backup servers, with resource information and management actions across them. I can see the appeal as soon as an environment grows beyond a single cluster. It also makes the broader Proxmox toolset feel much more complete.

Veeam backups and restores. This was a particularly important result: we successfully tested both. We already use Veeam for our on-site and off-site backups, so being able to retain that investment makes the hypervisor discussion much easier.

Veeam's current support matrix includes Proxmox VE 8.2 through 9.2, with specific installation and backup-software requirements. That gives us a supported integration to investigate further. We still need to validate the production combinations, application recovery and off-site workflow; successful trial backups and restores are the beginning of that evidence.

> ***The bit that won over the Linux administrator in me***
>
> *Under the web interface, this thing feels like Debian. Because it is.*
>
> *Proxmox VE 9 is based on Debian 13, Trixie. Its package configuration uses Debian's Trixie, updates and security repositories alongside the Proxmox repositories. It has its own kernel packages and cluster-management components, but the operating environment is immediately familiar.*
>
> *It feels like **99% Debian** to administer. I can use `apt`, inspect services and logs, understand the interfaces, and follow a problem through the Linux networking stack. If I need tcpdump, it is an ordinary Debian package. The tools and habits I already use have somewhere to land.*
>
> *ESXi has capable diagnostic tools too, including `tcpdump-uw` and `pktcap-uw`. The difference for me is the breadth and familiarity of a normal Linux environment. ESXi feels alien by comparison. With Proxmox, an experienced Linux administrator can get below the management interface and make sense of what is happening.*

> *That familiarity lowers the effort of learning the platform. There is still Proxmox-specific knowledge to acquire, particularly around cluster quorum, HA, storage and upgrades, but we are building on skills the team already has.*

What still has to earn its place in production

iSCSI is the big one. Our current environment depends on it, and we have not tested it in this trial yet.

Proxmox has an Open-iSCSI storage backend, and shared LVM over iSCSI is a documented arrangement. The implementation details matter: LVM-thin cannot be shared between nodes, and LVM snapshots using volume chains are still described as a technology preview in the current documentation.

Our next storage tests need to cover the actual array and LUN layout, multipathing, concurrent host access, path failures, migration under load, and backup and restore on the chosen backend. Veeam adds another reason to test the whole arrangement: its storage and worker requirements depend on the layout. Its exclusion of iSCSI disks attached directly to VMs is a different case from ordinary virtual disks on an iSCSI-backed LVM datastore.

The successful migration of locally stored disks answers one question. It does not yet answer the production SAN question.

On-site package mirrors are also on the list. We want a controlled way to provide system updates locally. Proxmox Offline Mirror can mirror both Debian and Proxmox APT repositories and create repository snapshots. Mirroring the enterprise repositories through that tool requires the relevant product subscription and an Offline Mirror subscription, so repository access belongs in the support discussion as well as the technical trial.

Dynamic load balancing is another feature I am keen to test. My recollection that it was recent was correct: Proxmox VE 9.2 introduced the Dynamic Load Balancer on 21 May 2026. It uses current node and guest utilisation to inform placement and can automatically migrate **HA-managed guests**, while respecting the configured HA rules.

That is a meaningful addition. We need to enable and tune it, then see how it behaves with our workloads, spare capacity and migration network. The HA scheduler documentation describes those controls. Our manual live-migration test has not tested automatic balancing, and I would want that evidence before making any VMware DRS comparisons.

Beyond those three items, I want to see host failures, quorum loss, rolling maintenance and recovery from the off-site copy exercised deliberately. A functioning VM is encouraging. A team that can recover the service during a bad afternoon is what makes the platform usable at work.

Support also needs a realistic budget. Proxmox publishes annual VE subscription prices of **€370 for Basic, €550 for Standard and €1,100 for Premium, per occupied CPU socket**, before applicable tax. Cores do not change that price. Direct vendor support operates during Austrian business hours, which matters to an Australian team planning incident coverage. The subscription details make those boundaries explicit; local partner coverage and separate backup-server support belong in our comparison too.

A contender, and a very good graduate project

I did not go into this project expecting to find an alternative I could picture us running at work. I expected some interesting research and a useful learning exercise.

Instead, we have Linux and Windows guests running, a working path out of ESXi, live migration without shared storage, working VLAN connectivity, automated Ubuntu builds, a dedicated backup platform, a wider management view, and successful Veeam backups and restores.

There is still serious work ahead, especially around iSCSI and failure recovery. We have not made a production migration decision. But Proxmox has earned the next round of testing, and it has changed how I think about that renewal conversation.

Thor did a great job. He took the research, built something we could test, and gave the team evidence on which to base a decision.

He wielded his mighty hammer and smashed it out of the park.

Product details and pricing examples checked on 10 September 2026.

Downloaded from https://www.gavinj.net/post/eleven-servers-one-hammer-proxmox
Generated September 20, 2026. Copyright Gavin Jackson. All rights reserved.