**Proxmox Environment Development For CCSE**

IT Capstone 4983 – W01

Professor Donald Privitera

**Team 18-T1**

George Boddie, Abbey Morgan, Omar Rodriguez, Mikayla Russell, Henricksen Edouard

**Table Of Contents**

1. **Introduction..............................................................................3**
   * **Statement of Purpose......................................................3**
   * **Project Scope..................................................................3**
2. **Proxmox Test Environment.........................................................3**
3. **Proxmox Infrastructure...............................................................6**

* **Current Kennesaw VMware Infrastructure to Migrate.........6**
* **Architecture Overview......................................................6**
* **Hardware Requirements...................................................7**

1. **Network Design..........................................................................7**
   * **Bridges and VLANS...........................................................7**
   * **Firewall............................................................................8**
2. **Automation................................................................................8**
   * **Packer..............................................................................8**
   * **Terraform.........................................................................8**
   * **Telmate............................................................................9**
   * **Ansible.............................................................................9**
3. **Active Directory..........................................................................9**
4. **System Administration Tasks......................................................9**
   * **Monitoring and Logging.....................................................9**
   * **Back Up and Disaster Recovery.........................................9**
   * **Patching..........................................................................10**
   * **Server Hardening.............................................................10**
5. **Custom Dashboard Utilizing Heimdall........................................12**
6. **Glossary....................................................................................13**
7. **References................................................................................15**

**Introduction**

Statement of Purpose

The project aims to help the CCSE Department with research and testing sustainable and cost-effective virtual machine management software to replace their current VMware infrastructure. This project will investigate the viability of the open-source virtual environment software, Proxmox, in Kennesaw’s environment. Scalability and automation are two priority requests of the CCSE department and will be a key focus throughout the project. Through this research, CCSE will be provided with key information to determine if Proxmox is the best solution for their environment.

Project Scope

The scope of this project mainly surrounds the research and testing of scalability and automation within a Proxmox environment. Kennesaw CCSE currently has 600 VMs containing content for specific classes and were custom made in VMware Horizon 7 utilizing Terraform for scripting their VM templates. The goal of CCSE is to migrate their environment to Proxmox if it provides the desired features.

Team 18-T1 will research and develop documentation surrounding Proxmox and VM migrations. Testing of virtual machine compatibility within Proxmox will occur within environments such as Windows, Linux, Redhat, Ubuntu, etc. At its core, automation of VM assignments and scalability will be prioritized topics.

Team 18-T1 will also provide CCSE with deliverables as outlined by the Capstone course along with documentation, Proxmox 8.3 test environment and recommendation of Proxmox’s viability as a solution for CCSE.

This report will cover the proposed Proxmox infrastructure, network design, Terraform, ansible, Active Directory, and system administration tasks. Team 18-T1 will also be working jointly with Team 18-T2 of the Computer Science Capstone. While this is a joint venture, there will not be many interdependencies between both groups. Proof of concept and advisement to CCSE is the primary goal of this project.

**Proxmox Test Environment**

To test the viability of Proxmox as a VMware we’ve set up a test environment to mirror the requirements necessary for Kennesaw application. Highlight areas of our milestone 1 build include:

1. Installation:
   * Installed Proxmox VE 8.3 (Graphical) on personal hardware with 48GB of ram, intel i5 CPU, 1TB M2 SSD, and 250GB SSD.

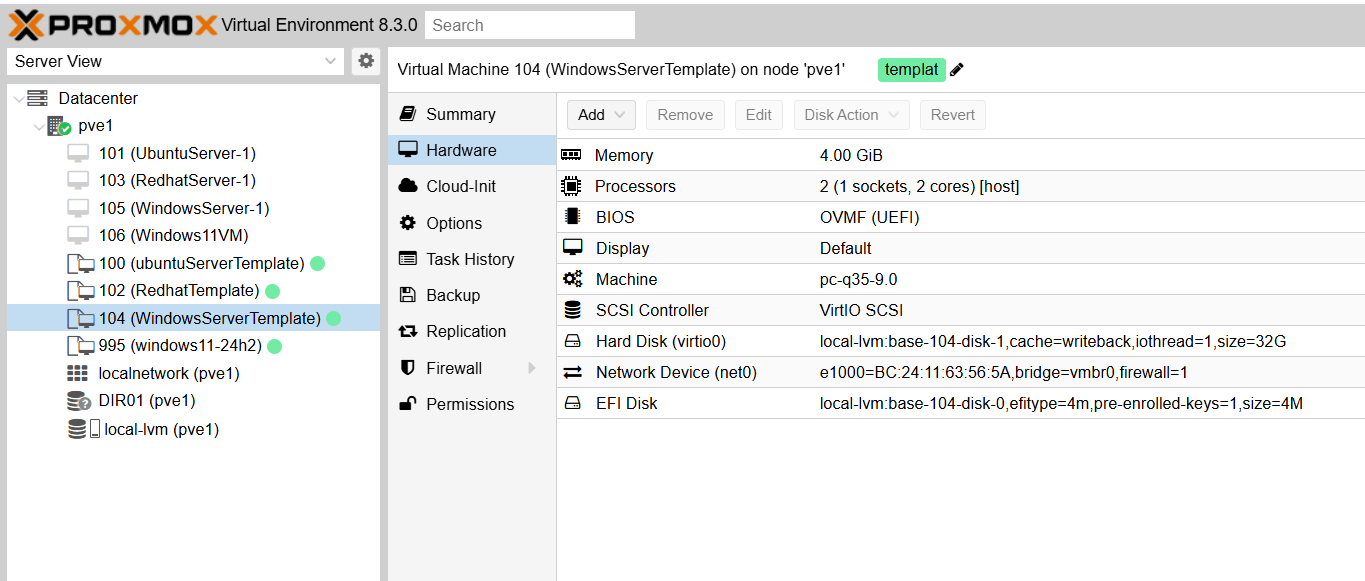
A screenshot of a computer

AI-generated content may be incorrect.

1. Virtual Machine Templates:

Created four VM templates to streamline provisioning:

* + Windows Server 2019

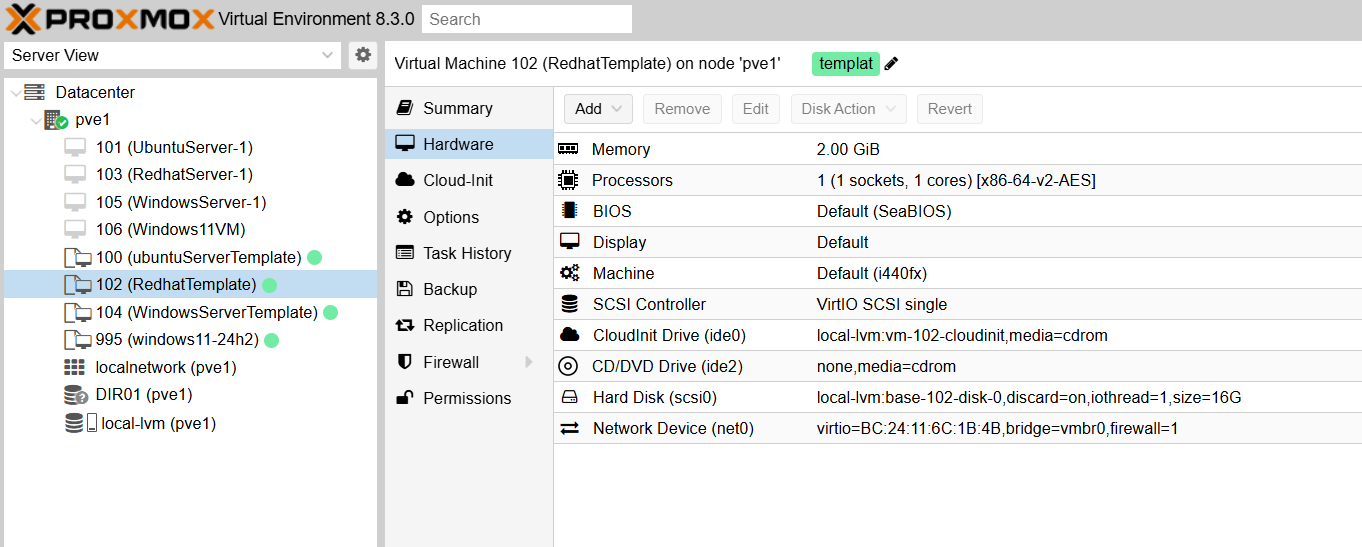


* + Ubuntu 22.04 LTS

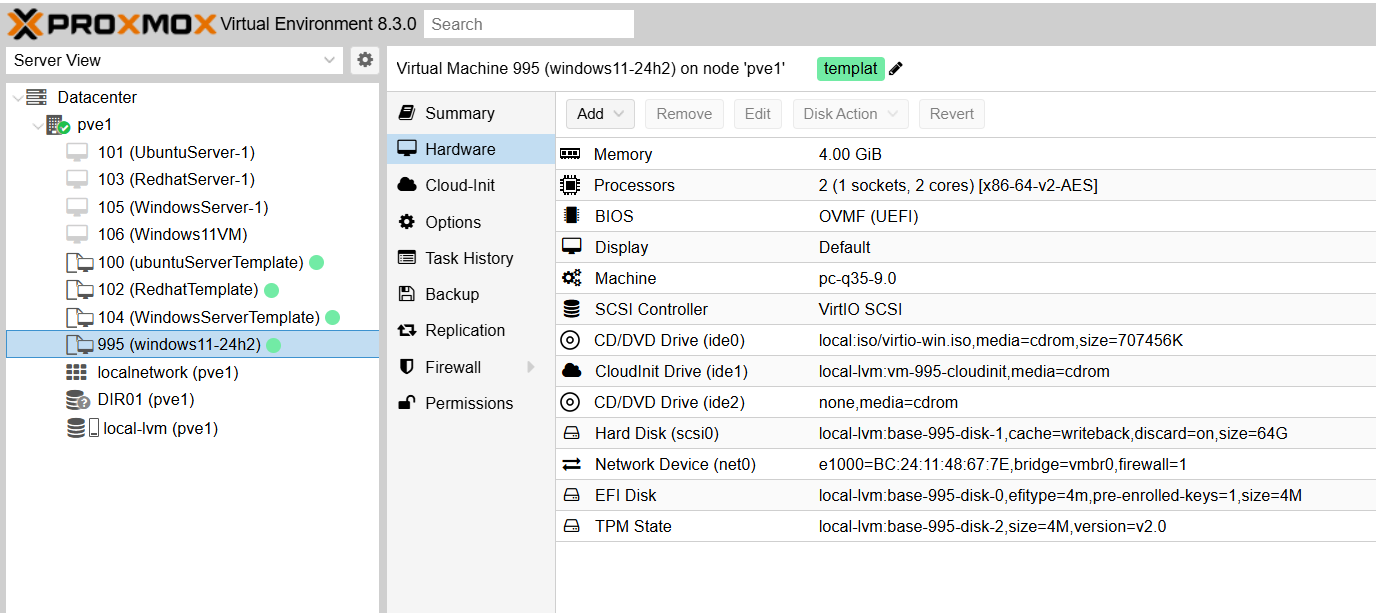
A screenshot of a computer

AI-generated content may be incorrect.

* + Red Hat Enterprise Linux 9



* + Windows 11



1. Storage Pool Setup:
   * Primary pool as LVM-Thin to store VMs and boot drive.
   * A secondary pool as a mounted directory for backups and ISO files.

**Proxmox Infrastructure**

Current Kennesaw VMware Infrastructure to Migrate

Our goal is to transition the current VMware infrastructure at Kennesaw State University’s CCSE department to a Proxmox-based solution. The existing setup supports approximately 600 VMs primarily utilized for academic purposes. These VMs are deployed via VMware Horizon 7 and automated using Terraform scripts. As part of our migration strategy, we will focus on converting from a VMware format to a compatible Proxmox format while ensuring that performance standards are met. VMware's current configuration supports Active Directory integration, allowing automated user management and VM assignments. We aim to evaluate the migration process through benchmarking and performance analysis to confirm that Proxmox can handle the required VM load efficiently.

Architecture Overview

To successfully replace the existing VMware environment, we will design and implement a Proxmox-based infrastructure that meets CCSE’s scalability, automation, and performance requirements. The architecture will be structured as follows:

* Proxmox VE Cluster Deployment: The new infrastructure will consist of a Proxmox cluster deployed for the CCSE department. This cluster will provide high availability, load balancing, and redundancy to support approximately 600 virtual machines used for academic purposes (“Features,” 2019).
* Virtual Machine Provisioning: Our approach will involve deploying Proxmox-managed VMs that replace VMware instances while maintaining compatibility with course requirements and academic workloads (Hoke, 2024).
* Active Directory Integration: To maintain seamless authentication, we will configure Proxmox to integrate with Active Directory, allowing automated VM assignment based on student groups and user roles.
* Automation and Disaster Recovery: Our implementation will leverage Telmate for automated VM provisioning and configuration management. Additionally, we will establish backup and disaster recovery solutions to ensure data integrity and minimize downtime risks (“Proxmox VE Administration Guide,” 2024).

Hardware Requirements

Proxmox VE needs a 64-bit Intel or AMD processor that supports virtualization (VT or AMD-V) and at least 2 GB of memory for the system, plus more for virtual machines and storage setups like Ceph or ZFS (1 GB extra for every TB of storage). Fast SSDs are recommended for best performance, and you can use hardware RAID with battery backup or ZFS for storage (but ZFS and Ceph don’t work with RAID controllers). It also needs reliable network connections with multiple Gbit NICs, or faster if your setup requires it. For PCI(e) passthrough, the processor must support VT-d or AMD-d. For testing purposes, Proxmox VE needs a 64-bit Intel or AMD processor that supports virtualization (Intel VT or AMD-V). This allows it to run full virtual machines. It also requires at least 1 GB of RAM, though more is recommended for better performance. You’ll need a hard drive to install the system and at least one network card (NIC) to connect to other devices. These are the basic hardware requirements to get started with Proxmox VE.

**Network Design**

Bridges and VLANS

In Proxmox VE, network bridges act like virtual switches, connecting virtual machines (VMs) to physical networks. These bridges allow VMs to communicate as if they are on the same local network. To organize and isolate network traffic, Proxmox supports VLANs (Virtual Local Area Networks). VLANs create separate sub-networks within the same physical network. “Proxmox VE supports this setup out of the box. You can specify the VLAN tag when you create a VM.” You can set up VLANs using different methods, like VLAN-aware Linux bridges, where you assign VLAN tags directly to each VM’s network interface. Another option is creating separate VLAN devices for each VLAN, such as eno1.5 for VLAN 5. Proxmox also supports advanced setups with Open vSwitch (OVS) and allows VLANs to be configured inside the VM itself. These features help keep networks secure and organized.

Firewall

The Proxmox VE firewall helps secure your system by letting you set rules for network traffic across the cluster, as well as for individual virtual machines (VMs) or containers. It organizes the network into zones (Host, VM, and VNet), so “...you can define rules for incoming, outgoing or forwarded traffic.” The firewall uses iptables on each node, ensuring secure isolation between VMs without slowing down the network. It works with both IPv4 and IPv6 traffic automatically. You can manage it easily through the Proxmox web interface or by editing configuration files, with tools like macros and security groups simplifying setup.

**Automation**

Various tools were used in our automation pipeline. Our pipeline is packer -> terraform -> ansible. Packer creates a “golden image”, this image is the base for all our VMs to be created from. From this image we can create new images with slight variations. Packer can create each of these and give them differing names for version control. Terraform can target these images and clones them to any desired scale then manage the VMs it's created. Finally, Ansible can do post build custom configurations. This is mostly for running scripts on active systems to make some changes to already existing applications. These tools were also leveraged in creation and automation of snapshots on the VMs.

Packer

Packer creates preconfigured golden images to server as a reliable reusable design for all VMs. Packer reduces setup times and ensures consistency across all VM’s

Terraform

Terraform is an open-source infrastructure as code tool that allows you to automate and manage the lifecycle of your infrastructure (Hashicorp). When integrated with Proxmox, Terraform allows you to create, manage, and provision virtual machines through automation. Terraform can manage the underlying logic of Proxmox configuration allowing you to define your configurations in .tf files and Terraform will handle the details work to accomplish your goal.

Benefits:

* Automates VM provisioning.
* Creates consistent reproducible infrastructure setup.
* Effective for scalability of your Proxmox environment.

Terraform automation in Proxmox simplifies infrastructure management by utilizing Infrastructure as Code (IaC) to provision and manage resources within a Proxmox Virtual Environment (PVE). The Terraform Proxmox provider allows users to define virtual machines, storage, networks, and other resources through declarative configuration files, enabling consistent and repeatable deployments. This automation reduces manual intervention, minimizes errors, and enhances scalability, making it ideal for managing on-premises or hybrid cloud environments. As highlighted by Spacelift, the Terraform Proxmox provider bridges the gap between Terraform's IaC capabilities and Proxmox's virtualization platform, offering a powerful solution for DevOps teams to automate infrastructure workflows efficiently (Spacelift, 2023). This integration is particularly valuable for organizations seeking to optimize resource management and accelerate deployment processes.

Telmate

In our implementation, we utilized Telmate which acts as a bridge between Terraform and Proxmox. Telmate allows for efficient automation of VM provisioning, storage, and networking within Proxmox (Terraform Registry, 2025). With Telmate, we were able to declare infrastructure configurations in Terraform’s declarative syntax, which ensures consistent and repeatable deployments (Cude, 2022). This integration simplifies management, reduces manual intervention, and improves scalability, making it an essential tool for automating Proxmox-based environments.

Ansible

Ansible can be utilized for several tasks including, post-deployment customization, software installs, configuring services, and application deployment

**Active Directory**

Traditional Active Directory may not migrate smoothly from the VMware environment to the Proxmox environment due to differences in architecture, compatibility, and management tools. Direct migration, third-party software solutions, and manual user creation within Proxmox will be researched and tested to determine the most efficient and reliable option. The team will conduct a proof of concept to demonstrate functionality, focusing on core integration aspects rather than delving deeply into granular user management systems unless they directly impact other tasks and deliverables.

A critical aspect of this migration is the automated assignment of VM pools to students or students to VM pools, which may rely on the successful integration of Active Directory or an alternative user management solution. CCSE currently used group permissions and/or role-based access control (RBAC) for assignment. Automating the deployment of VMs from a pool to authorized users is a key priority for CCSE, as it streamlines resource allocation, reduces administrative overhead, and ensures a seamless experience for students and faculty. The team will explore how Proxmox can support these automation goals, potentially leveraging its API, scripting capabilities, or third-party tools to achieve the desired outcomes. Additionally, considerations around scalability, security, and performance will be evaluated to ensure the solution meets the CCSE group requirements.

**System Administration Tasks**

Monitoring and Logging

Proxmox VE makes it easy to monitor and manage your system with built-in tools. “All configuration changes and tasks performed via the GUI or API are logged with time and user information.” The web interface shows real-time information about virtual machines (VMs), containers, and cluster activities, so you can track performance without extra software. It also includes smartmontools to check the health of hard drives and alert you to any problems. Proxmox keeps detailed logs of tasks and events, which are helpful for troubleshooting and tracking system activity. You can also log and monitor network traffic using the firewall’s logging features, keeping your environment secure and organized.

Back Up and Disaster Recovery

Backing up the servers is a vital part of this process. We’re recommending the use of the Proxmox Backup Server. It is compatible with the environment and hosts many features such as the option to do incremental or full backups, encryption, and methods to maintain data integrity all without dragging the network down. This option is also open source like the main Proxmox environment and stores the critical data in a space efficient manner. Regarding Disaster Recovery, we recommend the usage of Proxmox Backup server, but also the use of the 3-2-1 backup solution. While it may be an older method it is still a highly effective way of ensuring there are multiple ways to gain access to data when and if it is needed. We recommend that three copies of the data be made and made on at least two diverse types of storage media and at least one copy be sent offsite. We feel given the amount of data and the importance of it that this is done regularly and tested at a minimum annually. The offsite copy can be sent to a third-party cloud storage/disaster recovery company such as Veeam. Annual testing of backups, disaster recovery, and their respective plans should be tested with CCSE and KSU leadership to ensure that everything is functional, and revisions are made as needed. In both cases, we feel these measures will provide a robust backup and disaster recovery solution should Kennesaw ever need it.

Patching

To ensure a secure and efficient Proxmox environment, it is important for us to implement a structured patching strategy. Regular updates can be applied to maintain system stability and security. The Proxmox environment will be kept up to date by scheduling routine patching for the Proxmox VE software and associated packages (Proxmox VE Administration Guide, 2024; Proxmox and Automated Security Patching, 2024). A key focus will be minimizing downtime by leveraging live patching solutions in which KernalCare may come into play. Security patching will be prioritized by the need of regular monitoring and applying fixes promptly.

Server Hardening

Server hardening is a critical security practice for the resiliency of our server against attacks. By focusing on key areas during the development of our environment, we can ensure a robust and secure infrastructure.

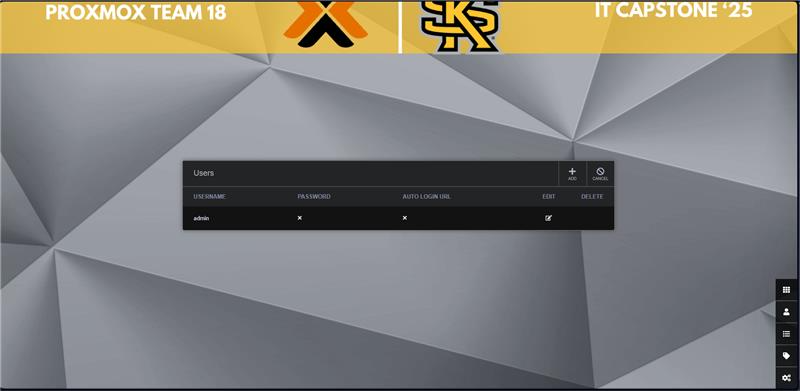
* Installation: Install a minimal Debian Proxmox OS with only necessary packages. Separate critical directories into partitions to limit damage due to a breach.
* Access control: Require a strong password policy. Use least access practices so users can only touch what is required by their role, and disable root logins (Buenning, 2024).
* Network Security: Use Proxmox built in firewall to control traffic, and limit traffic to only relevant ports such as 8006. Change the SSH port from the default port 22 to something non-standard.
* Patching: Keep Proxmox up to date with regular patching which would require a paid subscription.
* Monitoring: Configure appropriate logging and employ an intrusion detection system.
* Backups: schedule regular backups of VMs and Proxmox configurations. Encrypt backups so they aren’t stored as plain text.

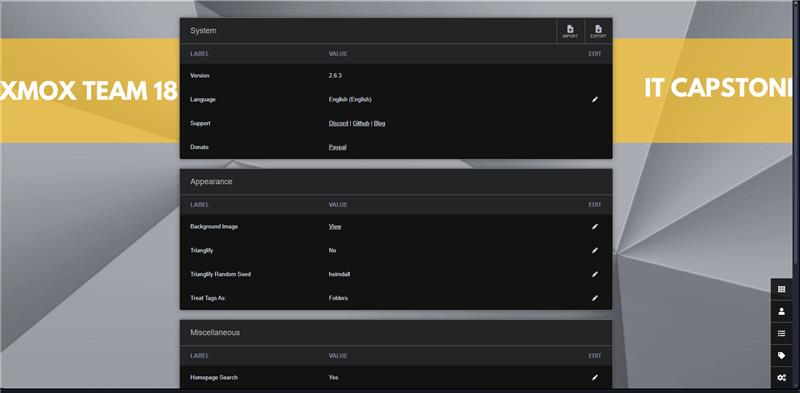
**Custom Dashboard Utilizing Heimdall**

Monitoring and Logging

Heimdall was installed and set up for our Proxmox connection to create dashboard functionality for the customer dashboard buildout. Heimdall presents our initial homepage and can be utilized for adding new apps to dashboard. Some apps that can be added for Proxmox administration are the server itself via its URL, Prometheus + Grafana to collect system metrics and create dashboards for visualizing the Proxmox server health and performance, ntopng to monitor network traffic, WireGuard to setup a VPN to securely connect to Proxmox server remotely.

In our use case it was set up for an initial homepage, the connection, and for adding/managing dashboard users and for configuring additional settings.





**Glossary**



Active Directory (AD)

Advanced Micro Devices (AMD)



College of Computer Science and Engineering (CCSE)

Central Processing Unit (CPU)

G.

Gigabyte (GB)

I.

Infrastructure as a Code (IaC)

International Organization for Standardization (ISO)

L.

Long Term Support (LTS)

Logical Volume Management (LVM)

O.

Operating System (OS)

P.

Proxmox Virtual Environment (PVE)

R.

Random Access Memory (RAM)

Role-Based Access Control (RBAC)

S.

Solid State Drive (SSD)

Secure Shell (SSH)

T.

Terabyte (TB)

V.

Virtual Environment (VE)

Virtual Local Area Network (VLAN)

Virtual Machine (VM)

Z.

Zettabyte File System (ZFS)

**References**

*Buenning, M. (2024, November 21). Steps of a server hardening process. NinjaOne. Retrieved February 15, 2025, from* <https://www.ninjaone.com/blog/steps-of-a-server-hardening-process/>

*Castagna, R. (2024, June). What is the 3-2-1 backup strategy? Definition from WhatIs.com. Retrieved from* <https://www.techtarget.com/searchdatabackup/definition/3-2-1-Backup-Strategy>

*Cude, T. (2022, December 2). Using Terraform with Proxmox. Retrieved March 23, 2025, from Tanner’s Tech website* <https://tcude.net/using-terraform-with-proxmox/>

*Features. (2019). Retrieved from* <https://www.proxmox.com/en/products/proxmox-virtual-environment/features>

*Firewall - Proxmox VE. (2022). Retrieved February 15, 2025, from* <https://pve.proxmox.com/wiki/Firewall>

*HashiCorp. (n.d.). Introduction to Terraform. Retrieved February 15, 2025, from* <https://developer.hashicorp.com/terraform/intro>

*Hoke, K. (2024, September 21). Overview of Proxmox Virtual Environment. Retrieved February 15, 2025, from* <https://docs.netapp.com/us-en/netapp-solutions/proxmox/proxmox-overview.html#network>

*Network Configuration - Proxmox VE. (2024, November 28). Retrieved from* <https://pve.proxmox.com/wiki/Network_Configuration>

*Proxmox and Automated Security Patching. (2024, May 9). Retrieved February 15, 2025, from* <https://forum.proxmox.com/threads/proxmox-and-automated-security-patching.146718/>

*Proxmox Backup Server. (2019). Retrieved from* <https://www.proxmox.com/en/products/proxmox-backup-server/overview>

*Proxmox VE Administration Guide. (2024, November 20). Retrieved from* <https://pve.proxmox.com/pve-docs/pve-admin-guide.html>

*Serrano, L. E. (2024, January 13). Proxmox Virtual Environment: Architecture, Services, and User Tools. Retrieved February 15, 2025, from* <https://cloudnroll.com/2024/07/11/understanding-proxmox-ve-architecture-services-user-tools/>

*Spacelift. (2023). Terraform Proxmox Provider: A Comprehensive Guide. Retrieved February 15, 2025, from* <https://spacelift.io/blog/terraform-proxmox-provider>

*Terraform Registry. (2025). Retrieved March 23, 2025, from* <https://registry.terraform.io/providers/Telmate/proxmox/latest/docs?ref=tcude.net>

### **Appendix A: Acronym Glossary**

This appendix provides definitions for key acronyms used throughout the report.

AD – Active Directory

AMD – Advanced Micro Devices

CCSE – College of Computer Science and Engineering

CPU – Central Processing Unit

GB – Gigabyte

IaC – Infrastructure as Code

ISO – International Organization for Standardization

LTS – Long Term Support

LVM – Logical Volume Management

OS – Operating System

PVE – Proxmox Virtual Environment

RAM – Random Access Memory

RBAC – Role-Based Access Control

SSD – Solid State Drive

SSH – Secure Shell

TB – Terabyte

VLAN – Virtual Local Area Network

VM – Virtual Machine

ZFS – Zettabyte File System