E-ISSN: xxxx-xxxx
DOI: https://doi.org/10.64803/cessmuds.v1.68

Proxmox-Based Virtualization Techniques on Virtual Servers

Rahmadani[1*]

[1]Faculty of Computer Science and Digital Intelligence, Information Technology, Universitas Pembangunan Panca Budi, Medan, Indonesia
E-mail: [1]*rahm4dani@gmail.com
*E-mail Corresponding Author: rahm4dani@gmail.com

Abstract

Server virtualization is one of the key solutions in modern IT infrastructure management, offering resource efficiency, cost reduction, and operational flexibility. Proxmox Virtual Environment (Proxmox VE) combines KVM (Kernel-based Virtual Machine) technology for virtual machines (VMs) and LXC (Linux Containers) for containers. This study aims to evaluate the implementation of Proxmox VE in server virtualization, focusing on resource usage and service availability. Through experiments conducted on virtual infrastructure, it was shown that Proxmox VE is effective in managing multiple VMs and containers simultaneously. System evaluation has revealed that Proxmox VE can reduce the number of physical servers required, thereby saving costs and space. The high-availability clustering feature in Proxmox VE has been proven to improve service availability. This study concludes that Proxmox VE is an efficient, flexible, and cost-effective virtualization solution for environments that require server consolidation and improved service availability.

Keywords: *proxmox; virtual server; virtual machine; container; cloud computing*

I. INTRODUCTION

Server virtualization technology is one of the key technologies in modern network infrastructure management (Kummara, 2025). Cloud computing and virtualization technologies have revolutionized modern computing, offering organizations significant advantages in terms of flexibility, scalability, and operational efficiency (Doukha & Ez-zahout, 2025). With virtualization technology, a single physical server (on premise) can run multiple operating systems, which may be of different types, simultaneously, thereby making hardware usage more efficient (Amin et al., 2025). The resources used in virtualization come from the resources owned by the physical computer where the virtual computer is created (the host). Resource allocation can be done because of the hypervisor technology (Ariyanto, 2023). Virtualization technology can optimize the utilization of hardware resources and simplify centralized server infrastructure management. Virtualization also enables more flexible and scalable management. The use of virtualization technologies enables the creation of a flexible and scalable network infrastructure (Chepurna & Frolov, 2025).

Virtualization was developed for abstracting the hardware and system resources to provide simultaneous execution of several operating systems on a single hardware platform. Virtual Machine Hypervisor technology, also called Virtual Machine Monitor, has a long history since the 1960s and was widely used before the era of cloud computing. As shown in Figure. 1, a virtual machine hypervisor (for example, Virtual Box, Xen, KVM, VMware, etc.) is software that provides a virtual platform, so several guest operating systems can run on one system server. The hypervisor runs as a middleware between the virtual machine and the OS. Each virtual machine has its own guest OS (Radchenko et al., 2019).

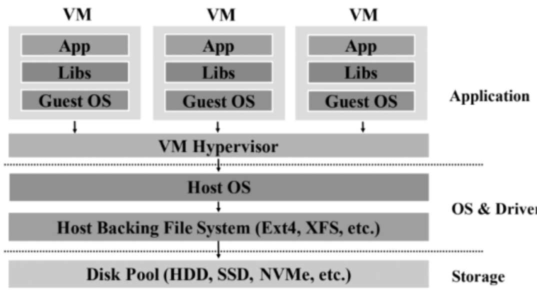

Figure 1. The architecture of the virtual machine hypervisor

Proxmox VE (Virtual Environment) is an open-source system capable of integrating two commonly used virtualization technologies: KVM (Kernel-based Virtual Machine) for creating virtual machines and LXC (Linux Containers) for creating containers. KVM (Kernel-based Virtual Machine) is an open-source virtualisation technology that facilitates hardware-level virtualisation directly within Linux, operating as an integral part of its kernel (Đorđević et al., 2025). Linux Containers (LXC) is one of the earliest containerization technologies, enabling multiple isolated Linux environments to run on a shared Linux kernel (Bompotas et al., 2025). Both technologies can be controlled through a web-based interface that is very easy to use and configure.

Proxmox VE's ability to manage various types of virtual servers with a centralized and flexible management system. Proxmox VE-based server virtualization techniques can reduce hardware usage, maximize CPU and memory capacity, and simplify resource management.

E-ISSN: xxxx-xxxx
DOI: https://doi.org/10.64803/cessmuds.v1.68

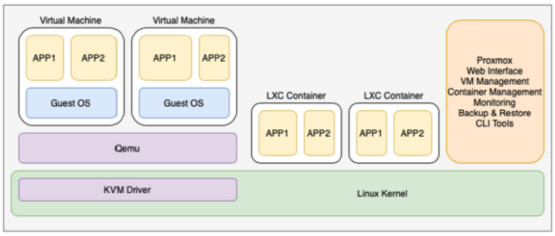

Figure 2. Proxmox VE Architecture (Šimon et al., 2023)

Server virtualization offers several benefits, including cost efficiency, dynamic resource management, and space and energy savings (Jin et al., 2012). Organizations can reduce the amount of hardware required, which directly impacts investment and maintenance costs. In addition, virtualization enables isolation between different services, reducing the risk of system failures that could disrupt all applications running on physical servers (Ketha et al., 2025).

Containerization is an operating system-level virtualization technology used to implement and run applications and cloud system services without the need to implement a virtual server for each solution. The use of containers allows multiple isolated cloud services to run on a single physical or virtual server while accessing the same operating system kernel.

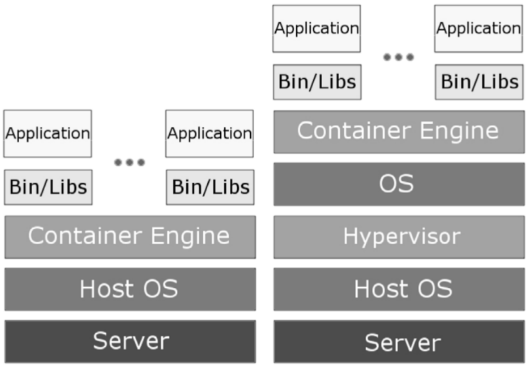

Figure 3. Containers and containers on VM (Mavridis & Karatza, 2019)

Container-based virtualization leverages operating system (OS) kernel features to create isolated environments at the application level, differing significantly from full virtualization, which virtualizes the entire computing platform from hardware up to the OS (Ferro et al., 2026). Containers are a type of operating system-level virtualization. The abstraction of a container can be seen in Figure 4. Within network slicing architectures, containers permit agile, lightweight, and scalable deployment of network functions. Unlike VMs, containers provide process-level isolation with minimal overhead, allowing for faster instantiation, reduced resource consumption, and improved portability across heterogeneous environments (Iliadis-Apostolidis et al., 2025).

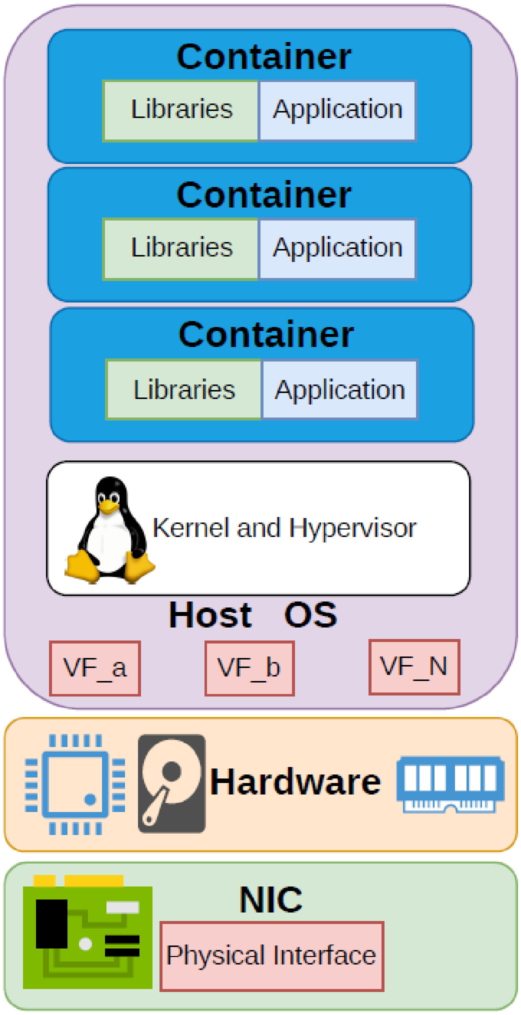

Figure 4. Abstraction of a container (Iliadis-Apostolidis et al., 2025)

Cloud computing provides on-demand access to a shared pool of configurable computing resources, such as servers, storage, networks, applications, and so on, over the Internet. Cloud computing systems provide services by allocating underlying physical resource flexibly to users via virtualization technologies(Jawed & Sajid, 2022)(Lee, 2013). In recent years, container-based virtualization has emerged in cloud computing, driving widespread attention and adoption across both industry and academia due to the lightweight and flexibility of features. For example, major cloud providers (e.g., AWS, Google Cloud, and Azure) offer container orchestration services (e.g., Kubernetes and ECS) to support large-scale application deployment and management(Queiroz et al., 2024) (Guo et al., 2025).

This study aims to conduct an empirical study on the application of Proxmox VE for virtual server virtualization. This is done to assess how well and efficiently this system manages hardware and software resources so that it can maintain service availability under different conditions.

II. RESEARCH METHODOLOGY

This research was conducted using a structured experimental approach utilizing one physical computer (on premise). A virtual network of conditional OBI has been designed for experimental verification of the effectiveness of the proposed methodology for the formation of the OBI IS system (Lakhno et al., 2023). This is done to test the implementation of Proxmox Virtual Environment in a server virtualization environment to observe features, resource usage, and service availability.

In the initial stage of the research, the first thing to do is to install and configure virtualization devices on physical computers (on premise). The virtualization software used is Virtual Box. After Virtual Box is properly installed, settings and configurations are made as described in Figure 5.

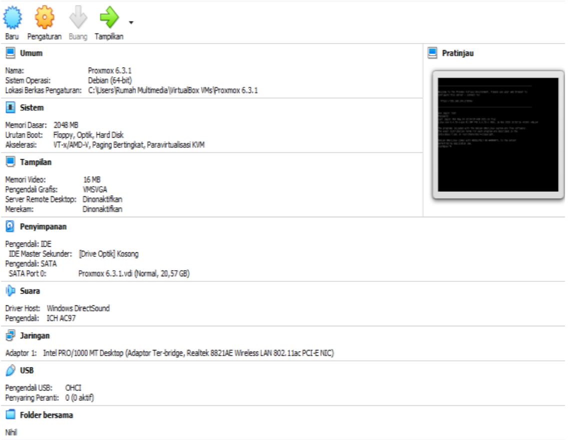

Figure 5. Physical Device Configuration (on premise) on Virtual Box

After the virtual server setup is complete, the next step is to install and configure Proxmox Virtual Environment, with the main configuration in the Management Network Configuration section as follows;

1. Hostname (FQDN) : pve.unpab.com
2. IP Address : 192.168.100.2
3. Netmask : 255.255.255.0
4. Gateway : 192.168.100.1
5. DNS Server : 192.168.100.1

The Proxmox VE installation process is carried out in stages according to the rules set by Proxmox VE. One of the most important parts of the installation process is performing Network Configuration Management on each virtual device. After the Proxmox VE installation process on the Virtualization Server is complete, a Proxmox VE-based Virtual Machine Dashboard will be displayed, as shown in Figure 6.

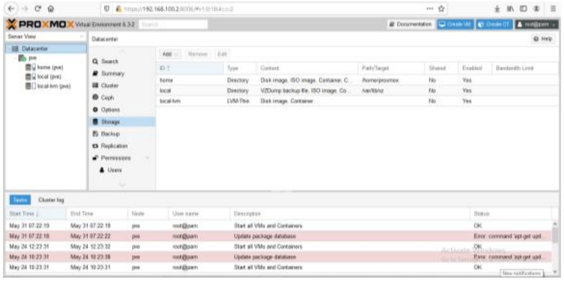

Figure 6. Dashboard Proxmox VE

There are several experimental steps to prepare a Proxmox VE-based virtual machine. For the initial steps after the Proxmox VE installation process is complete, the next step is to add storage to the Proxmox VE dashboard, as described in Figure 7.

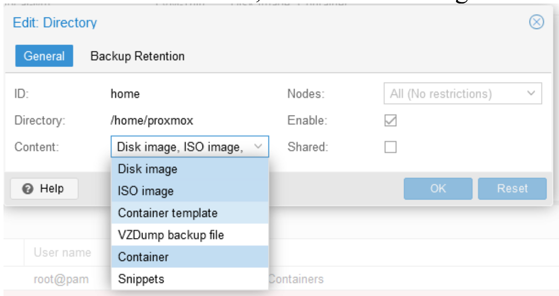

Figure 7. Add Storage

To add storage, click on the Server View window, then click Add on the General tab, then fill in the ID with home, then Directory: /home/proxmox. In the Container section, select Disk Image, ISO, Container, and Container Template, make sure enable is selected, then click OK.

The next step of the experiment is to add users and groups to Proxmox VE, as described in Figure 8.

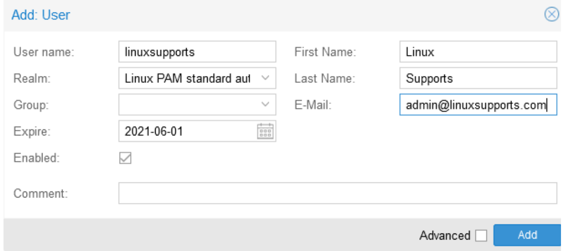

Figure 8. Add User

The next step in this research was to add a group to Proxmox VE, as described in Figure 9.

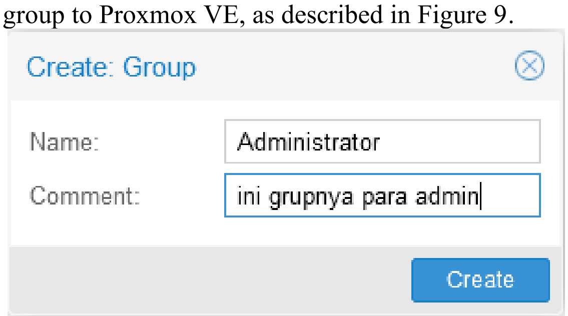

Figure 9. Add Group

E-ISSN: xxxx-xxxx
DOI: https://doi.org/10.64803/cessmuds.v1.68

The next step is to add permissions, which are needed to manage user roles in the Proxmox VE system, as explained in Figure 10.

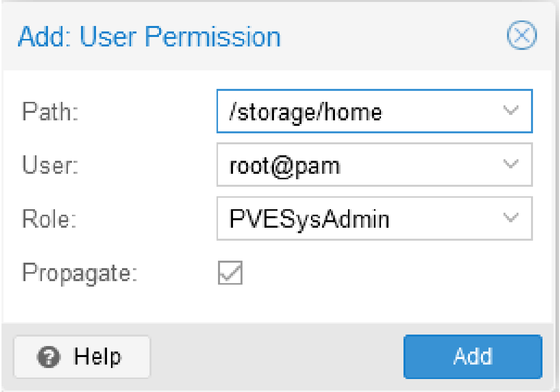

Figure 10. Add Permission

For permission configuration, click on the Server View window, then select Permission, then select User or Groups, for example; Fill in the Path, User name, and Role of the permission to be created. Then click Create Add. Next, adjust the permission group as well.

The next step is to upload the ISO file to create virtualization as described in Figure 11.

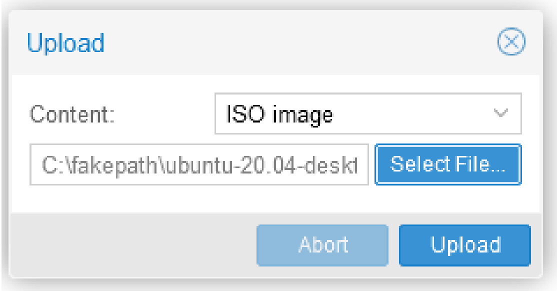

Figure 11. Upload ISO Linux Ubuntu

Click on the Server View window, then select pve, then home (pve). In the dashboard section, select ISO images and in the content section, make sure ISO Image is selected, then browse for the ISO file to be installed from your computer directory. Click Upload. Wait until the upload process is complete and OK.

Next is to set up the pool on the virtualization machine as described in Figure 12.

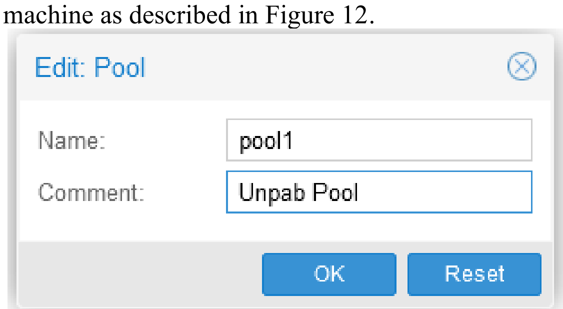

Figure 12. Konfigurasi Pool

Click on the Server View window. Select Permission, then select Pools. In the name and comment sections, enter the pool name and comment.

The final step in the initial configuration is to add or create a virtual machine, as described in Figure 13.

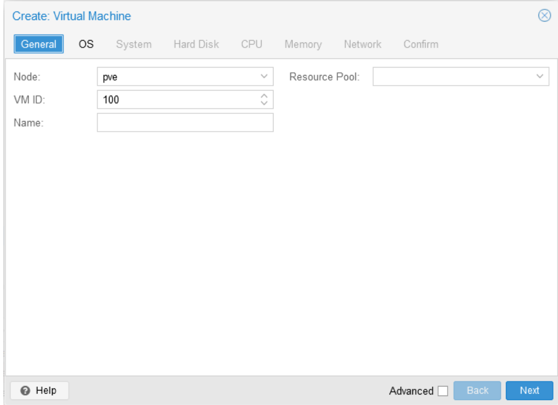

Figure 13. Create Virtualization

Click on the Server View window. In the upper right corner, select Create VM. A configuration window for creating a new VM will appear. Set the configuration according to your needs. Then install the OS based on the uploaded ISO, using the Console Menu. Wait until the OS installation process on the VM is complete.

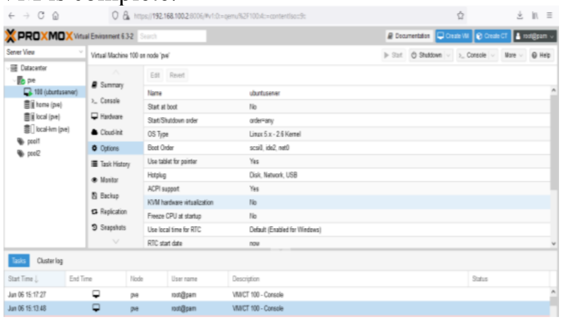

Figure 14. Machine Virtualization

Before creating and configuring a new Virtual Machine (VM), it is recommended to perform general configurations on Proxmox VE, such as user and group settings, permissions, and pools. The newly created Virtual Machine (VM) will then be used to maximize virtual server services according to user needs.

The configured Virtual Machine can then be installed with various operating systems according to requirements. In this study, the operating system used on the virtual machine is Linux Ubuntu.

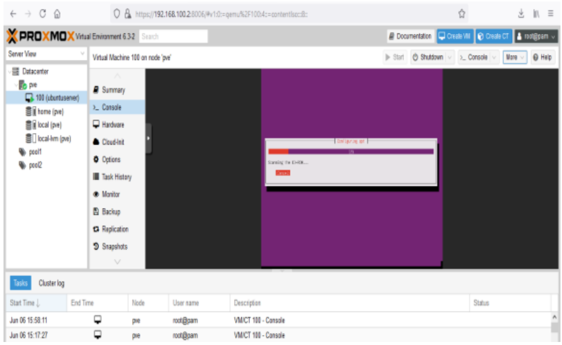

Figure 15. Operating System Installation

After the virtual machine has been configured, the next step is to prepare the container template. On the Proxmox VE page, select the server view, then click home (pve). Next, select CT Templates. Then select the Templates button. Next, select the available Packages, for example, ownCloud, then select Download, making sure you are connected to the internet. Wait until the download process is complete.

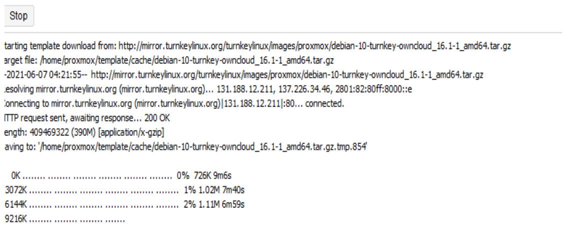

Figure 16. Template Container

Next, to create a container, do the following: Click the Create CT button. Fill in the information needed to create a new container on each tab until you are finished. At the end of the tabs, a summary related to the container settings that have been created will be displayed. Then click Finish. The downloaded container template file will then be extracted automatically. Wait until it is finished or until you see "TASK OK.").

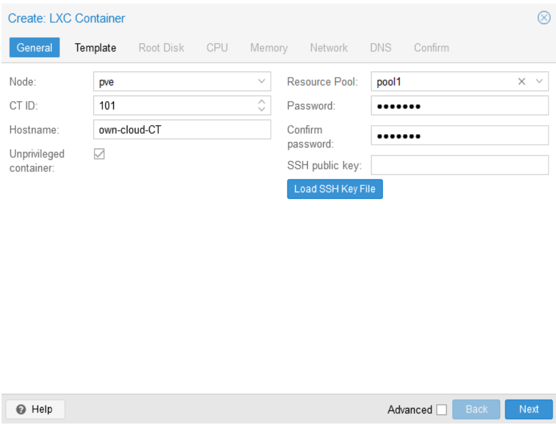

Figure 17. Create LXC Container

III. RESULTS AND DISCUSSION

To evaluate the results of Proxmox VE implementation and configuration for available services, using clustering techniques and virtual machine backups that have been created. This is done to protect the virtual machine infrastructure on the virtual server. Firewall settings for all virtual machines within the cluster, or defining rules for virtual machines and containers. All configurations will be stored in the cluster file system, and the iptables-based firewall service will run on each cluster node, providing complete isolation between multiple virtual machines.

During the firewall evaluation stage, you can select the VM whose firewall will be tested. Then, set the chain on the firewall to be created

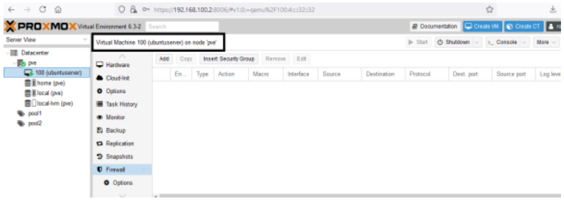

Figure 18. Server View VM

Add a Rule to the Firewall that will be activated and tested. After adding the Rule to the firewall, the firewall configuration results will be displayed, as shown in Figure 19.

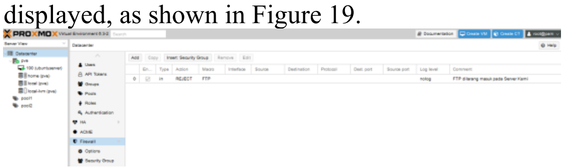

Figure 19. Firewall Rules

Then, for the clustering process, assume that there are two Proxmox VE server nodes, each with an IP address of 192.168.100.30 and 192.168.100.101, which will be referred to as proxmox1 and proxmox2.

Proxmox1 will act as the master and proxmox2 as the slave. Activate the console of each Proxmox to start creating the Proxmox server cluster. On the 192.168.100.30 server in the browser and directly on the VirtualBox virtual server, use the following command:

root@pve:~#pvecm create cluster1

To view the status of the newly created cluster, type the following command;

root@pve:~#pvecm status

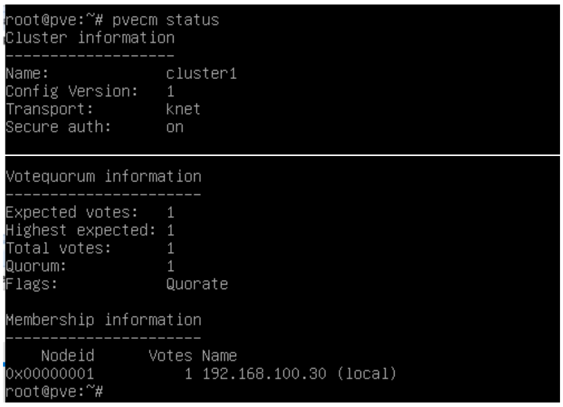

Figure 20. Status Cluster

On the terminal server 192.168.100.101 on the virtual box server, log in as root. Ensure that the corosync service is running, then to view its status, type the command as described in Figure 21.

Figure 21. Running *service corosync*

Type the next command to view the service status, as shown in Figure 22.

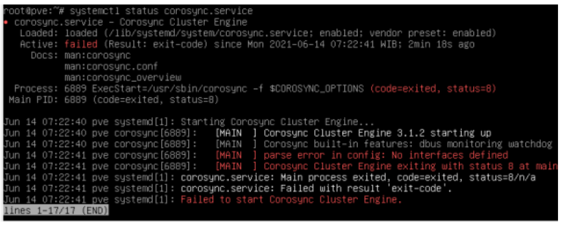

Figure 22. Status service corosync

Next, run the command, as shown in Figure 23.

Figure 23. Service corosync Configuration

Finally, run the command to add or merge on the destination server, as shown in Figure 24.

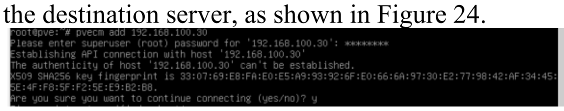

Figure 24. Pvecm Merge

The result will show two PVEs from the clustering process on one dashboard. All nodes (Proxmox servers) are connected within one network, as shown in Figure 25.

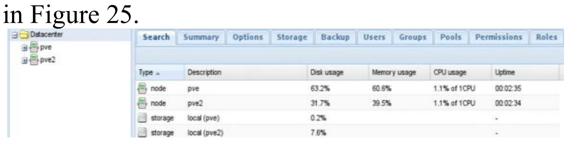

Figure 25. Results of merging 2 Proxmox servers

The data backup process is an important requirement in every good network infrastructure implementation. Proxmox VE provides an integrated solution that can be utilized for the capabilities of each storage system and each type of guest system.

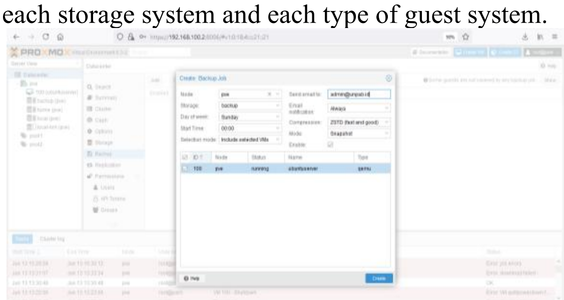
Figure 26. Backup Configuration

For the backup process, you can do this by selecting Add Directory. The results of the Add Directory backup are displayed as shown in Figure 27.

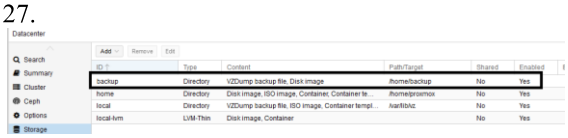

Figure 25. Storage Backup

To restore backup files on a Proxmox server, the process is basically the same as performing a backup, except that restore is the opposite (restore will use data overwrite mode). Click on the VM to be restored. Click on the main backup menu. Select the Restore tab.

Then select the backup file to be restored, then click restore, click Restore (wait until the restore process is complete).

IV. CONCLUSION

Proxmox VE as a server virtualization platform has proven to be effective in providing virtual machines on virtual servers with resource efficiency, flexibility, and better service availability. Physical server consolidation and centralized management capabilities can be the best solution for institutions or organizations with limited resources. Therefore, Proxmox VE is ideal for use as a virtual server infrastructure base, such as for education, startups, or institutions with limited budgets but requiring high flexibility in server management.

V. RECOMMENDATIONS

For further research suggestions and recommendations, explore aspects of performance under higher loads (many tenants), security and isolation between VMs or containers, and the use of distributed storage and networks to support large-scale implementation in accordance with organizational needs.

VI. REFERENCES

Amin, N. U., Shahzad, A. D., Hamza Abid, M., Yamin Muiz, M., Suleimanov, T., & Abdul Razick, M. S. (2025). *The Role and Application of Hypervisors in Modern Organizations*. 0–14. https://doi.org/10.20944/preprints202501.1355.v1

Ariyanto, Y. (2023). Single Server-Side and Multiple Virtual Server-Side Architectures: Performance Analysis on Proxmox Ve for E-Learning Systems. *Journal of Engineering and Technology for Industrial Applications*, *9*(44), 25–34. https://doi.org/10.5935/jetia.v9i44.903

Bompotas, A., Kalogeropoulos, N. R., & Makris, C. (2025). CommC: A Multi-Purpose COMModity Hardware Cluster. *Future Internet*, *17*(3). https://doi.org/10.3390/fi17030121

Chepurna, I., & Frolov, D. (2025). a Method for Increasing the Productivity of a Distributed Firewall Based on Proxmox in Corporate Computer Networks. *Innovative Technologies and Scientific Solutions for Industries*, *2025*(3), 180–188. https://doi.org/10.30837/2522-9818.2025.3.180

Đorđević, B., Janjić, K., & Kraljević, N. (2025). Mathematical Modelling and Case Study with File System Performance Comparison for Linux-based Hypervisors. *Acta Polytechnica Hungarica*, *22*(1), 101–121. https://doi.org/10.12700/aph.22.1.2025.1.6

Doukha, R., & Ez-zahout, A. (2025). Enhanced Virtual Machine Resource Optimization in Cloud Computing Using Real-Time Monitoring and Predictive Modeling. *International Journal of Advanced Computer Science and Applications*, *16*(2), 658–664. https://doi.org/10.14569/IJACSA.2025.0160267

Ferro, L., Bravi, E., Sisinni, S., & Lioy, A. (2026). Privacy-Preserving Container Attestation. *Journal of Network and Systems Management*, *34*(1), 1–28. https://doi.org/10.1007/s10922-025-09982-5

Guo, Y., Wei, L., Fan, C., Ma, Y., Zhao, X., & He, H. (2025). Joint Optimization of Container Resource Defragmentation and Task Scheduling in Queueing Cloud Computing: A DRL-Based Approach. *Future Internet*, *17*(11). https://doi.org/10.3390/fi17110483

Iliadis-Apostolidis, D., Lawo, D. C., Kosta, S., Monroy, I. T., & Olmos, J. J. V. (2025). QRoNS: Quantum Resilience over IPsec Tunnels for Network Slicing †. *Electronics (Switzerland)*, *14*(21), 1–26. https://doi.org/10.3390/electronics14214234

Jawed, M. S., & Sajid, M. (2022). A Comprehensive Survey on Cloud Computing: Architecture, Tools, Technologies, and Open Issues. *International Journal of Cloud Applications and Computing*, *12*(1), 1–33. https://doi.org/10.4018/IJCAC.308277

Jin, Y., Wen, Y., & Chen, Q. (2012). Energy efficiency and server virtualization in data centers: An empirical investigation. *Proceedings - IEEE INFOCOM*, 133–138. https://doi.org/10.1109/INFCOMW.2012.6193474

Ketha, K. S., Song, G., & Zhu, T. (2025). *Analysis of Security in OS-Level Virtualization*. 2–7. http://arxiv.org/abs/2501.01334

Kummara, R. (2025). Server Virtualization: Transforming Modern IT Infrastructure. *European Journal of Computer Science and Information Technology*, *13*(13), 1–13. https://doi.org/10.37745/ejcsit.2013/vol13n13113

Lakhno, V., Alimseitova, Z., Kalaman, Y., Kryvoruchko, O., Desiatko, A., & Kaminskyi, S. (2023). Development of an Information Security System Based on Modeling Distributed Computer Network Vulnerability Indicators of an Informatization Object. *International Journal of Electronics and Telecommunications*, *69*(3), 475–483. https://doi.org/10.24425/ijet.2023.146495

Lee, J. (2013). A view of cloud computing. *International Journal of Networked and Distributed Computing*, *1*(1), 2–8. https://doi.org/10.2991/ijndc.2013.1.1.2

Mavridis, I., & Karatza, H. (2019). Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing. *Future Generation Computer Systems*, *94*, 674–696. https://doi.org/10.1016/j.future.2018.12.035

Queiroz, R., Cruz, T., Mendes, J., Sousa, P., & Simões, P. (2024). Container-based Virtualization for Real-time Industrial Systems - A Systematic Review. *ACM Computing Surveys*, *56*(3). https://doi.org/10.1145/3617591

Radchenko, G. I., Alaasam, A. B. A., & Tchernykh, A. N. (2019). Comparative analysis of virtualization methods in Big Data processing. *Supercomputing Frontiers and Innovations*, *6*(1), 48–79. https://doi.org/10.14529/jsfi190107

Šimon, M., Huraj, L., & Búčik, N. (2023). A Comparative Analysis of High Availability for Linux Container Infrastructures. *Future Internet*, *15*(8). https://doi.org/10.3390/fi15080253