Comparison of Proxmox and OpenNebula as Cyber Range platforms

Ange Simpalingabo

CYBER DEFENCE LAB

May 30, 2024

Comparison of Proxmox and OpenNebula as Cyber Range platforms
Ange Simpalingabo
May 30, 2024

Cyber Defence Lab
https://cylab.be

BibTeX citation:

```
@techreport{citekey,
title = {Comparison of Proxmox and OpenNebula as Cyber Range platforms},
author = {Ange Simpalingabo},
institution = {Cyber Defence Lab},
year = {2024},
month = {5},
}
```

Abstract

This report explores the comparison of two cyber range platforms, Proxmox VE and OpenNebula, with a particular focus on their application in the educational environments of the Royal Military Academy. It evaluates the technical capabilities of both systems in terms of installation, hypervisor management, and user interfaces, as well as their ability to integrate with existing virtual environments. The study highlights the strengths of Proxmox VE for centralized environments and OpenNebula for managing distributed and scalable configurations, emphasizing how each platform meets the specific training needs in cybersecurity.

Keywords: Proxmox VE, OpenNebula, cyber range, virtualisation, éducation, cybersécurité, gestion des hyperviseurs

Contents

1. Introduction to the cyber Range

1.1. Cyber Range Overview

Cyber Range is an advanced, highly complicated network environment and simulation platform for cybersecurity scenarios, which spans thousands of devices and participants. Cyber Range is highly important in cybersecurity training, as it is a very realistic yet highly controlled environment in which users can test their defense and attack skills separately or simultaneously against a real network. Such exercises might involve a small penetration test or a full-scale cyberattack simulation.

1.2. The role Of cyber Range in Cybersecurity

In the dynamic threat environment, cyber range is a critical framework in learning and testing operations in cybersecurity. It allows organizations, both small and large, to have better preparations against possible cyberattacks and practice personnel in acting under duress. In reality, its importance lies not only in boosting technical skills but also through its being the key position inside the overall cybersecurity resilience strategy of organizations.

1.3. Integration and Usability in Education

As part of the cybersecurity program at the Royal Military School, it is essential to implement a cyberattack simulation platform that not only accurately simulates a cyberattack environment but is also easy to use and highly accessible for students and instructors. A crucial aspect of this ease of use is the platform's ability to integrate and migrate virtual machines created on Oracle VirtualBox, the most widely used virtualization system, allowing for a seamless transition of developed scenarios to a larger and more complex simulated environment on the cyber range platform.

1.4. Illustration

After establishing the importance of cyber ranges in the current context of cybersecurity and their crucial role in training, it can be useful to visualize these concepts for better understanding. To this end, a detailed diagram will be presented to illustrate the main components of a cyber range and their integrated functioning. This visual aid will help to concretize the provided descriptions and show how these different elements interact to form a complete and effective simulation environment.

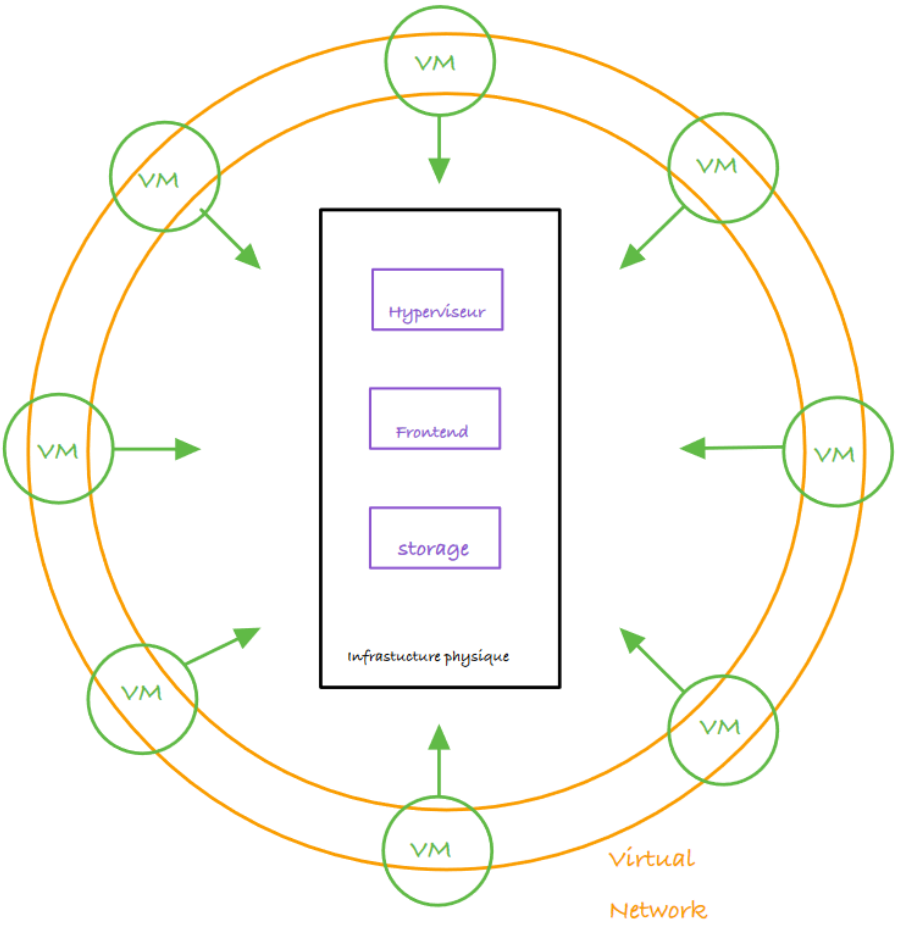


Figure 1.1. Illustration of a cyber range

Deployment of the cyber range will be through the installation of three key components, which include a frontend, hypervisors, and a storage solution. More so, it will be on a physical infrastructure, for instance, a PC among the others as previously discussed. This will then enable running of VMs in it. They need to connect through the bridge access towards the network interface to ensure smooth communication.

1.5. Objective and Scope of this Report

Therefore, this report will examine two cyber range platforms, Proxmox and OpenNebula, to determine which offers the best ease of use and integration of virtual machines in an educational context. The objective is to provide a clear recommendation on the most suitable platform to enhance the effectiveness of learning and the educational experience at the Royal Military School.

2. Methodology

2.1. Creation and Export

The virtual machines we created run Kali Linux and Metasploitable on Oracle VirtualBox. After creating and configuring the virtual machines, we exported the OVA file. The OVA package format contains a VMDK file, or a file that can be used to create virtual machines running on any compatible virtualization platform, containing not only the disk files but also all the virtual machine settings.

2.1.1. Disk Conversion

VMDK files in the OVAs are converted into the QCOW2 version which is the much better and recommended version of a disk file format to use for virtualization systems. This process is so important as this will guarantee us that whatever the current functionalities of the virtual machines are, will be possible inside the cyber range. [1]

2.2. Deployment on the Server

2.2.1. Migration and Cloning

Once the files are converted to QCOW2, they will be migrated to the cyber range server. The goal is to initially deploy two instances of each operating system (Kali Linux and Metasploitable). These instances will serve as templates for cloning.

2.2.2. Scalability and Access

We plan to clone these templates to create a total of 20 instances of each operating system on the server. These clones will be configured to ensure that each student has access to their own virtual machines (Kali Linux and Metasploitable)

to conduct their cybersecurity exercises.

2.3. Evaluation

The analysis for which the best one between the two offers better easy to use and, at the same time, performance and resource handling in an educational environment. A comparison of these results would guide in formulating a clear recommendation on the most appropriate platform for the cybersecurity program.

3. Proxmox VE

3.1. General Description

Proxmox VE stands for Proxmox Virtual Environment, an open-source server virtualization solution. It combines Linux kernel-based virtualization (KVM) and containerization (LXC), providing a comprehensive solution for managing virtual machines and containers on a single interface. The tool was developed by the Austrian company Proxmox Server Solutions GmbH, and over time, it has become popular due to its versatility, simplicity, and cost effectiveness.

Proxmox VE is actively used in data centers of small and medium-sized enterprises, as well as in large organizations. It is important for optimizing the use of hardware solutions and reducing infrastructure costs. Proxmox is popular among system administrators because with its help, managing a complex IT infrastructure is simplified, including a user-friendly Web interface, as well as the provisioning of functions required for cluster management and high uptime. Proxmox is a source for projects that need maximum flexibility for managing IT resources and maximum availability of services while minimizing infrastructure and hardware costs.

3.2. Installation

First of all, download the (ISO) media necessary to create a bootable USB key. This will allow us to install Proxmox on our server. It is always better to download the ISO directly from the editor's site : Proxmox Installer, to take no risk. Download it from the "Proxmox VE ISO installer" section. Then, you can build the bootable USB key with a tool like Rufus : rufus.exe. Select your USB key and the ISO file you have just downloaded.

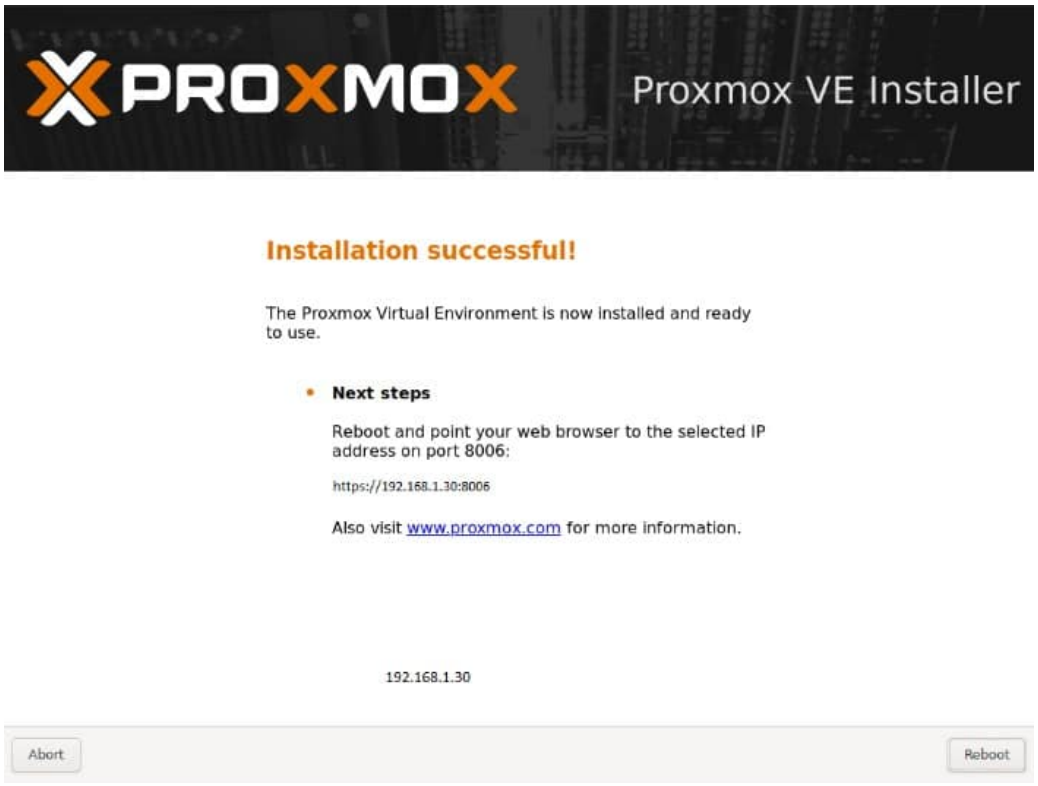
In order to minimize the risks, as much as possible, the modifications within the BIOS and/ or UEFI configuration should be carried out before initiating the installation on your server. All the more so on a system made of recent firmware. Be sure to go to your motherboard manufacturer's website and look up which key must be pressed. This most likely will be F1, F2, F10, Delete, or Esc.

You will often need to also validate the virtualization characteristics in the BIOS/UEFI as below:

- If you have an Intel processor, you have to turn on VT-x or Virtualization Technology.
- If you have an AMD processor, you have to turn on AMD-V or SVM (Secure Virtual Machine mode).

Afterwards, modify the boot sequence options so that the USB key comes up in the first position. That way, in order to install, it will just be that much easier to boot from the key.The Proxmox interface is now displayed! Select "Install Proxmox VE" from the list.

The user will be guided step-by-step through configuring the server to meet their specific needs. After configuration, access is provided to the server's web interface. This allows the user to manage and configure the system more intuitively and directly.



[2] [3]

3.3. Technical and Functional Specificities

3.3.1. Integration of KVM and LXC

KVM (Kernel-based Virtual Machine)

Proxmox VE supports both types of virtualization for its virtual machines, including KVM, a complete virtualization, and LXC, a containerization. KVM is a full virtualization solution for Linux hardware containing virtualization extensions. Thanks to KVM, Proxmox is capable of creating and controlling VMs which run their own operating systems in complete isolation. KVM's significance stems from the fact that it can provide both robust and efficient virtualization, allowing VMs to attain near-native performance.

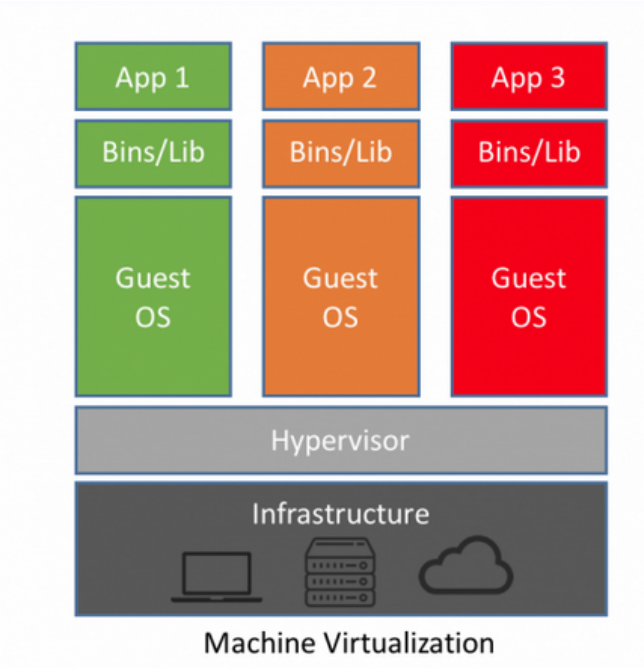


Figure 3.1. Kernel-based Virtual Machine represenation

The host has a hypervisor that allows it to share hardware resources among virtual machines, each acting as an independent system with its own operating system and software stack.

LXC (Linux Containers)

LXC, in turn, is also an operating system-level virtualization system that lets you run multiple independent Linux environments, or containers, on the same Linux host. LXC has less system overhead than KVM since it does not emulate hardware and has the same host system. When the main priority is server density and efficiency, containers are the best choice. They also allow for microservices applications since they have a faster boot time and consume fewer resources.

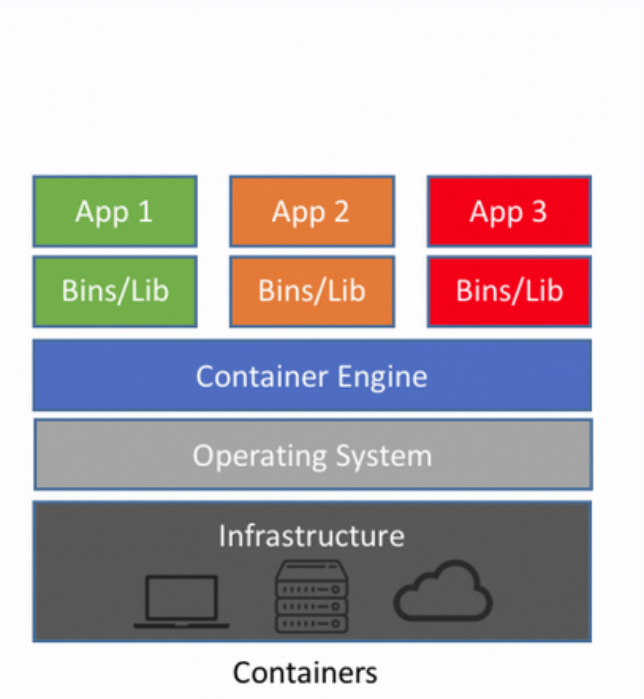


Figure 3.2. Linux Containers represenation

The interest lies in maintaining a certain level of isolation between them with usage flexibility (containers can be easily deployed and destroyed), without incurring the "cost" of full virtualization of a machine with its OS. This is particularly relevant regarding the occupied storage space or the "heaviness" of the hypervisor.

With Proxmox VE, the combination of KVM and LXC in Proxmox VE allows users to choose the most appropriate technology according to the specific needs. Whether to maximize performance with full virtual machines or optimize resources with containers, Proxmox delivers unparalleled flexibility in managing virtualized environments.

3.4. Cluster Management

A cluster is a group of servers, or nodes, that are supposed to work together and be treated as one entity. The basic idea behind the creation of a cluster is to improve the availability of resources and services in case of server failure, and also to improve the overall performance of a server by many means.

Proxmox VE uses clustering to allow multiple Proxmox servers act as a single one and share the virtualization tasks. This allows managing VMs and containers located on several physical hosts, making the system highly available, flexible for maintenance, and scalable for resources. This can easily be done through the Proxmox VE web interface, which offers an intuitive platform to manage and monitor the overall cluster.

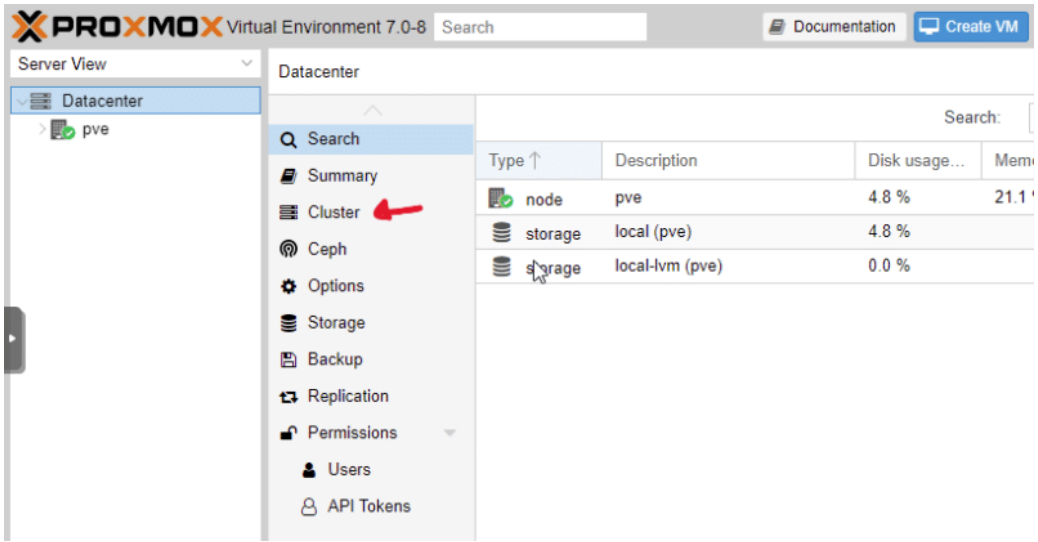


Figure 3.3. Set up a cluster

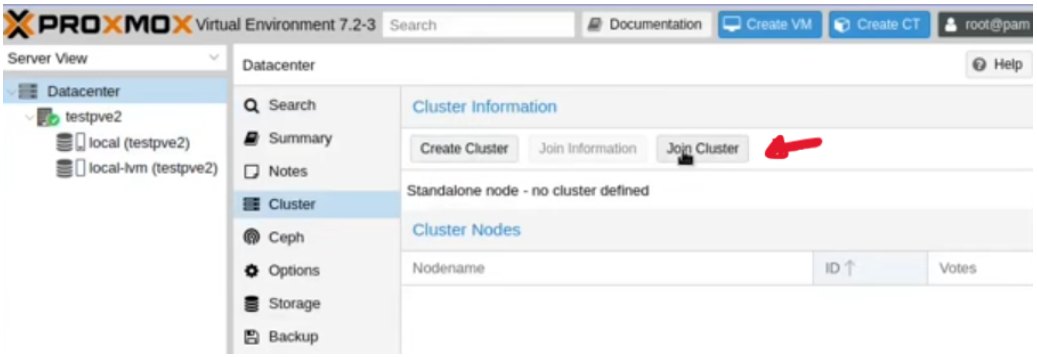


Figure 3.4. Join a cluster

Next, you need to retrieve the information from the "Join information" section, which serves as the code to allow other nodes to join the cluster.

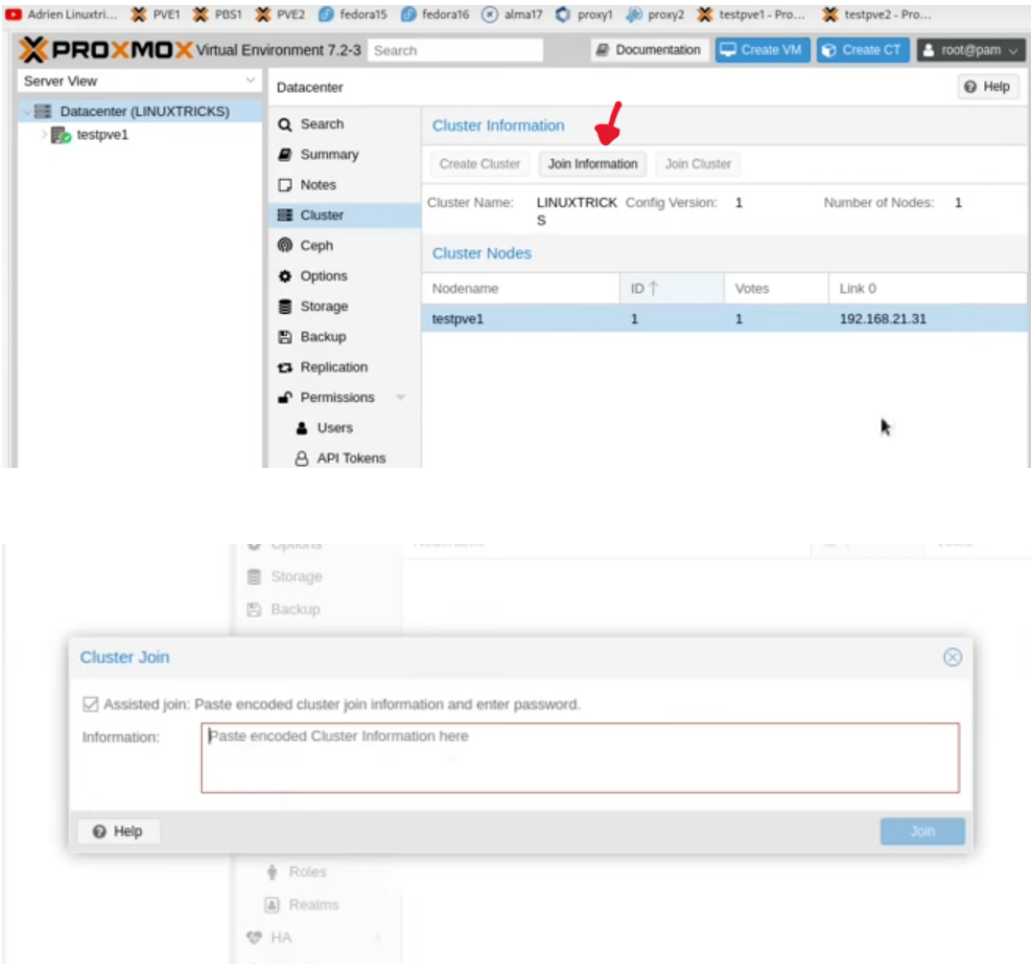


Figure 3.5. Join iformation

[4]

3.4.1. Main Features of Clustering in Proxmox VE

1. High Availability (HA)
    - Proxmox VE provides high availability for virtual machines and containers. This means that, in case of failure of any server in that cluster, the VMs or containers running over that server are automatically restarted over another server in the cluster, which tremendously reduces service interruption as well as downtime.
2. Centralised Management
    - All the nodes in the cluster can be managed through a single web interface. As a result, this simplifies administrative tasks, such as software updates, network configuration and performance monitoring, among others. which can all be effected from a single center point.
3. Live Migration
    - Clustered Proxmox VE enables uninterrupted services while performing live migrations of VMs and containers. For instance, the administrator performs VM migration as part of the maintenance task, avoiding downtime for end users, by moving one VM from one physical server to another.

We have mentioned that high availability is important. Then, it is necessary to define via the web interface all the above high availability settings for each of the virtual machines, including failover options and boot priorities. It is also essential to advance with the test that a high availability configuration really works. This includes checking server failure scenarios.In addition, it is important to use shared files on all servers - Ceph, NFS, and iSCSI. We'll look at this in more detail immediately.

3.5. Storage

The storage framework in Proxmox VE is an integral part directly associated with the performance and flexibility of the virtualized environment. Proxmox offers a variety of storage options that could be configured, optimized, and managed as per the requirement of the infrastructure in place.

3.5.1. Local storage

- Hard disks and SSDs : As they are installed directly on the servers with Proxmox, these virtual disks can, in turn, be used by them directly. As a result, they are suitable for environments that require fast data access, such as databases or applications with a high degree of transactionality.
- LVM (Logical Volume Manager) : Provides flexible management of physical storage volumes, enabling dynamic volume resizing and efficient snapshot management for rapid recovery and testing.

3.5.2. Shared Storage

In a shared storage environment, shared storage is facilitated using the NFS (Network File System), especially in the environment in which several servers have to access a similar set of data, such as server clusters or high availability.

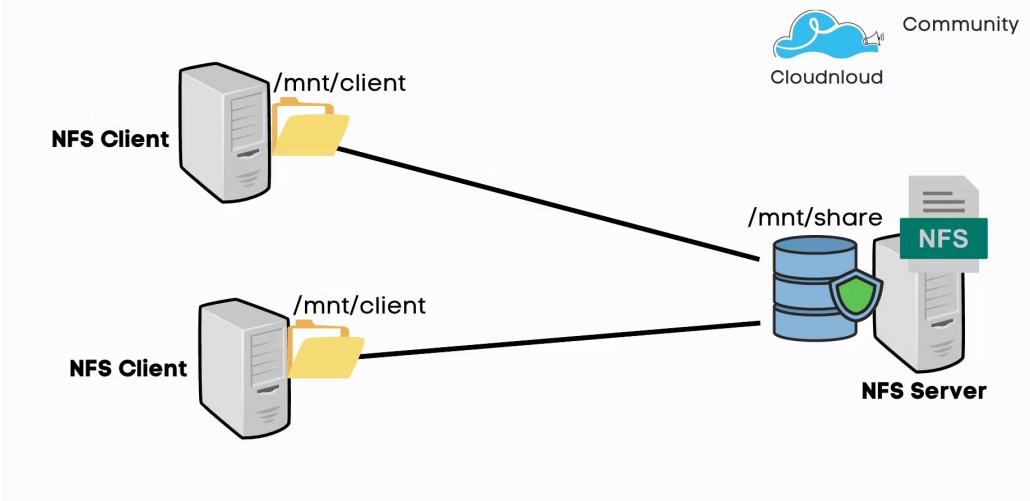


Figure 3.6. NFS Server

3.5.3. Distributed Storage

Ceph is a distributed storage solution inherently integrated to the Proxmox platform, ensuring none other can outperform in terms of performance, availability, and scalability. Particularly, Ceph provides a good fit for large-scope cloud and virtualization environments, where availability of data durability is in the first place. Distributed clusters in which disks of virtual machines get stored allow them to replicate data and to perform live migrations of virtual machines.

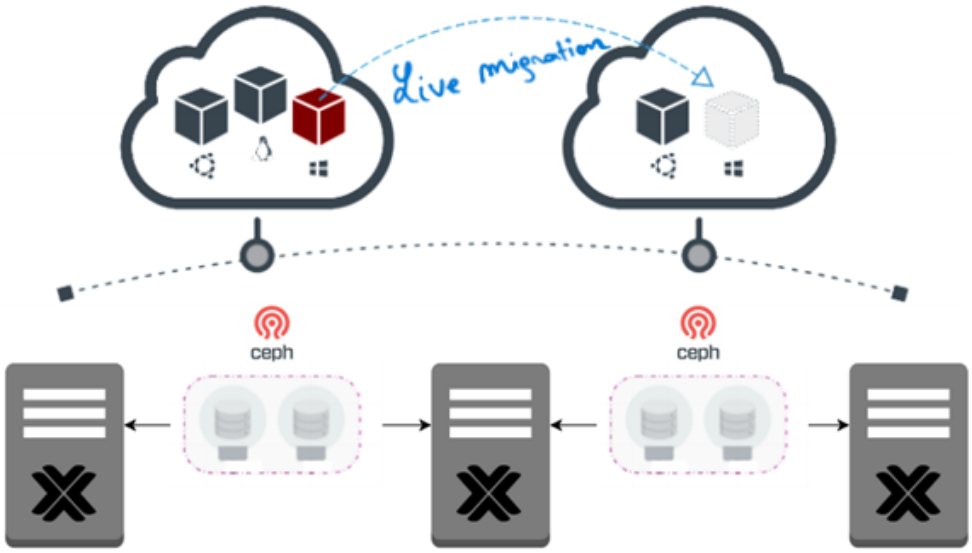


Figure 3.7. Ceph Server

3.5.4. NFS vs Ceph

NFS:

- **Configuration:** Simple and quick to set up.
- **Data Centralization:** Centralized, which can be a single point of failure.
- **Service Interruptions:** Risk of interruption during maintenance, requiring careful planning to minimize impact.

Ceph:

- **Configuration:** More complex and requires technical expertise.
- **Data Centralization:** Distributed, eliminating single points of failure and providing high availability.
- **Service Interruptions:** Minimal, due to the distributed and redundant nature, allowing maintenance operations without major interruptions.

3.6. Web Interface

Let's now take a deeper look into the Proxmox VE web interface, fully packed with ease of use and that makes the management of virtual machines, containers, and

system resources quite easier. The interface is not only to create and configure virtual machines, but myriad other features, amongst them some advanced tools, including those for cloning and performance monitoring. Below are some features and how they can improve the day-in, day-out management of your virtual infrastructure.

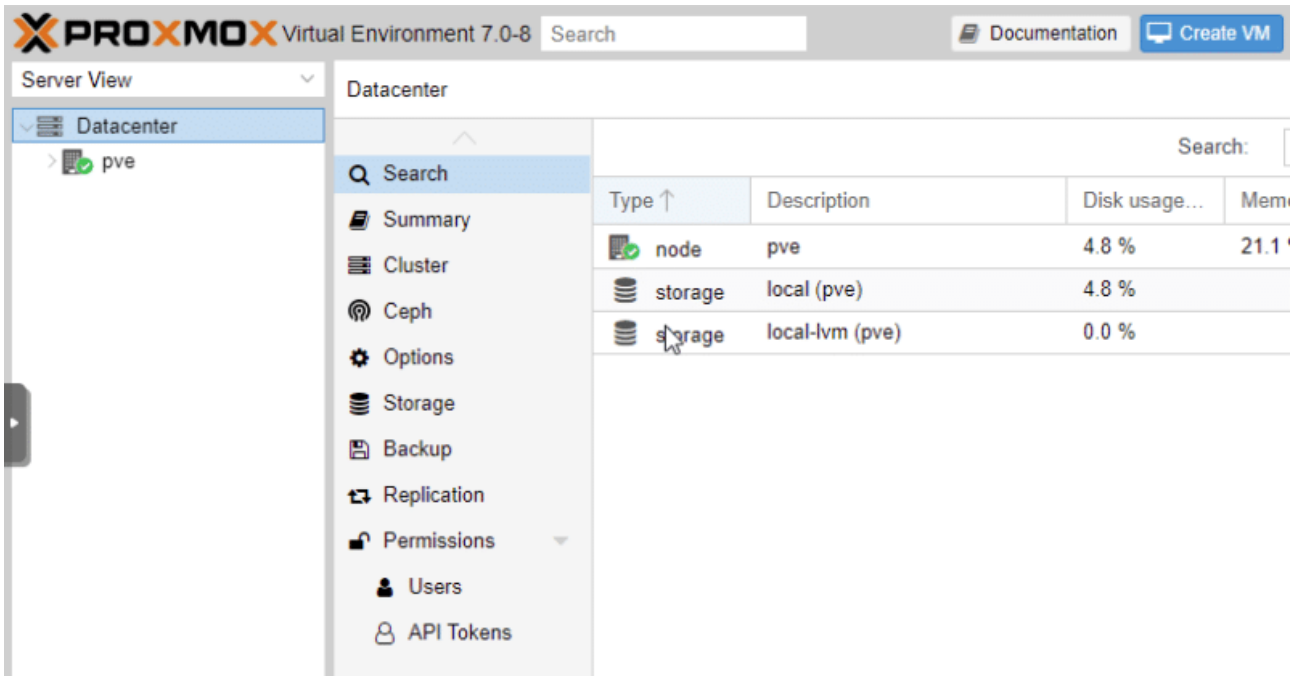


Figure 3.8. Web Interface

The default user for logging in is "root" and the password is the one configured during installation.

3.6.1. VM creation and configuration

We can easily upload ISO files from our computer and add them directly to a VM via the interface, but this does not apply to qcow2 or raw files. In our context, this whole process includes command line use and the Proxmox web interface, which is very nice and user-friendly. So, first of all, copy the disk file, for example, a qcow2 file converted from a vmdk disk, export from VirtualBox, we want to create a VM Proxmox to the Proxmox server by SCP or any other file transfer protocol. Now you will create a new VM through the web interface but, of course, create it empty, or remove this disk afterward. In this step, you will define the CPU, memory, and network configuration available. A second command will be needed for the integration of the already imported disk into the specified new VM. But the interface includes at least a shell prompt to execute this command lines.

```
-scp VMname.vmdk root@ipadrss:/folder
-qemu-img convert convert -f vmdk VMname.vmdk -O qwow2 VMname.
   qcow2
#Then go to the interface create the VM
#Then go to right folder with the prompt shell
- qm importdisk ID_VM VMname.qcwo2 local-lvm
```

3.6.2. Cloning

Proxmox VE allows administrators to quickly duplicate existing virtual machines by using the advanced cloning capabilities. This feature is beneficial for creating similar test or development environments. Another very useful aspect is that system allows for the creation and management of VM templates. Therefore, the creation of new virtual machines with a similar configuration is straightforward.

However, the automation of the remote access configuration by ensuring that the VNC/SPICE ports always follow a particular scheme or the automatic retrieval of the virtual machine's IP address requires the use of scripts. This allows for much more effective management of the system on a day-to-day basis by removing repetitive tasks or ensuring that all operations follow a uniform configuration pattern. Thus, the use of custom scripts is imperative for the effective management of a Proxmox VE virtual environment, as it allows for much faster operation and creates a smoother experience with fewer errors.

3.6.3. Management, Updates and Maintenance

The interface also provides real-time monitoring tools on a technical level, showing you useful metrics such as CPU time, memory use, and network bandwidth, as well as which storage you have. I find that the platform also makes software and hardware administration significantly easier. Particularly in the context of the web interface, you can always organize updates and then administer them without significantly inhibiting services. I think this offers a good combination of security and efficiency.

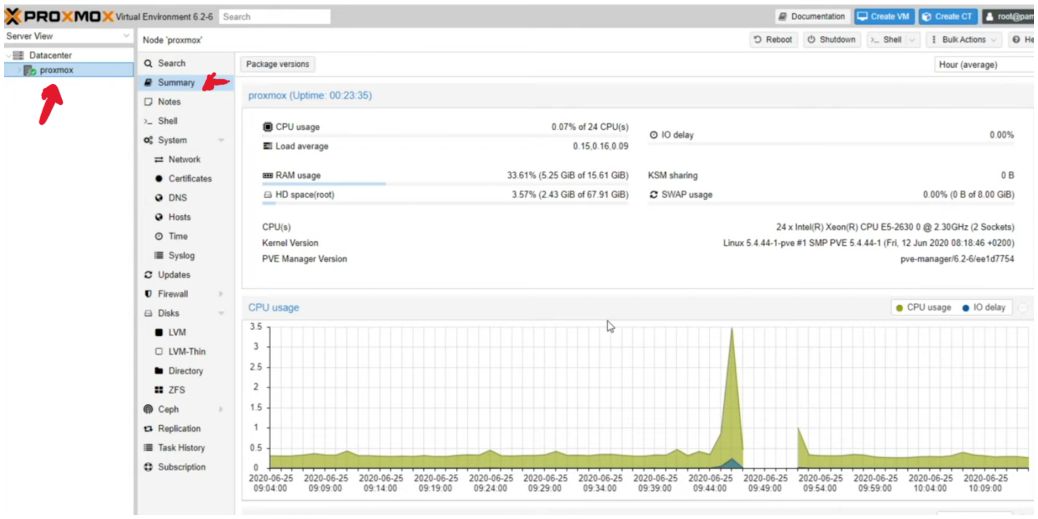


Figure 3.9. Node summary

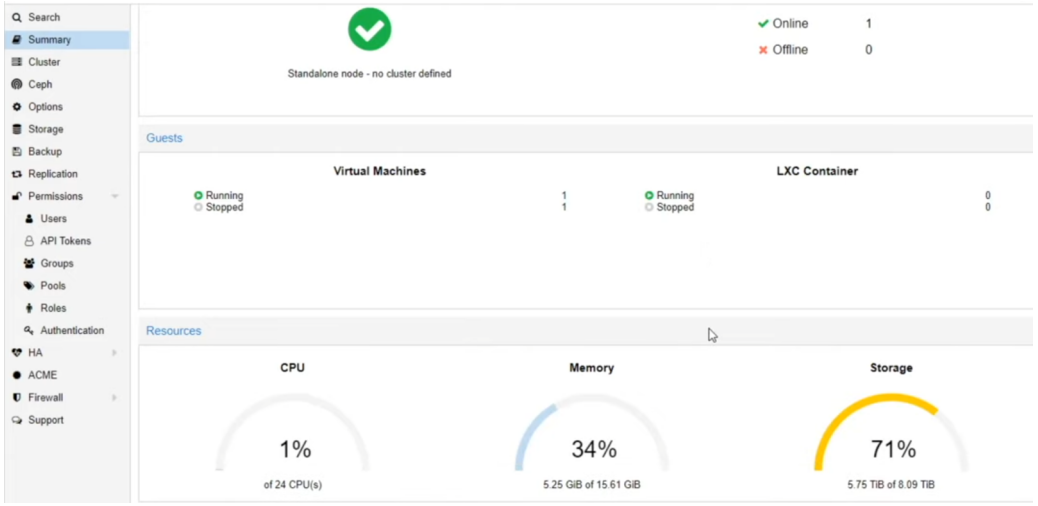


Figure 3.10. Data center summary

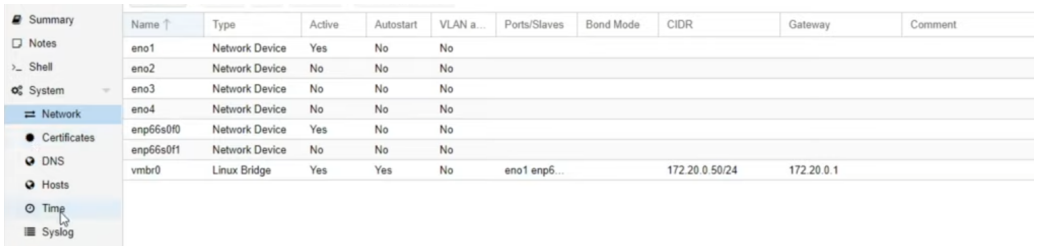


Figure 3.11. Network configuration

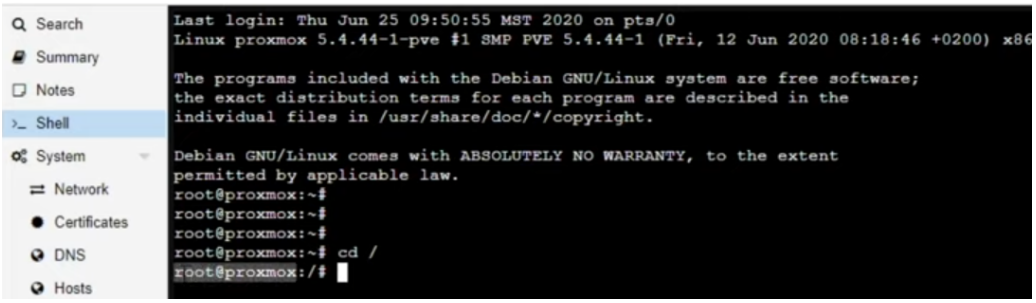


Figure 3.12. Shell prompt

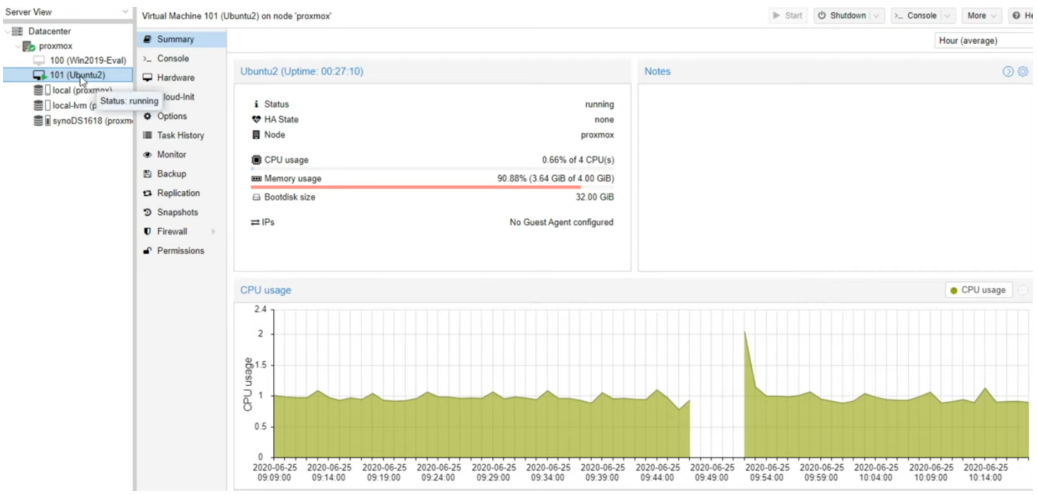


Figure 3.13. VM summary

The Proxmox VE web interface is proving to be an essential and powerful tool for the efficient management of virtualisation environments. Through the screenshots and details presented, we were able to see how this interface facilitates complex operations, from creating and managing VMs and containers to monitoring and optimising system performance. Proxmox VE combines remarkable accessibility with functional depth, enabling users of all levels to quickly master their virtualised infrastructures. The interface serves not only to simplify technology management, but also to reinforce the security and reliability of the entire system, ensuring that businesses can exploit the full potential of virtualisation to meet their operational and strategic needs. With Proxmox VE, administrators have a robust platform that adapts cutting-edge technology to simplified day-to-day use, demonstrating the transformative impact of well-designed solutions on organisational efficiency and productivity.

Throughout this chapter, we have explored in detail how Proxmox VE facilitates advanced and efficient management of virtualization environments. From the initial installation to the complex management of clusters, through innovative storage strategies and the use of the web interface, Proxmox VE stands out as a robust solution for system administrators seeking to optimize their IT resources.

The provided screenshots and explanations clearly illustrate the power and flexibility of Proxmox's web interface, which simplifies complex operations such as creating, managing, and cloning VMs, as well as monitoring and optimizing system performance. These tools not only reduce technical complexity but also enhance the overall security and reliability of the infrastructure, ensuring that businesses can fully leverage the potential of virtualization to meet their operational and strategic needs.

Ultimately, Proxmox VE demonstrates that cutting-edge technology can be adapted for simplified daily use, showcasing the transformative impact of well-designed solutions on organizational efficiency and productivity. This makes Proxmox VE an essential platform for any administrator looking to master modern IT challenges while maintaining a high-performance and secure infrastructure.

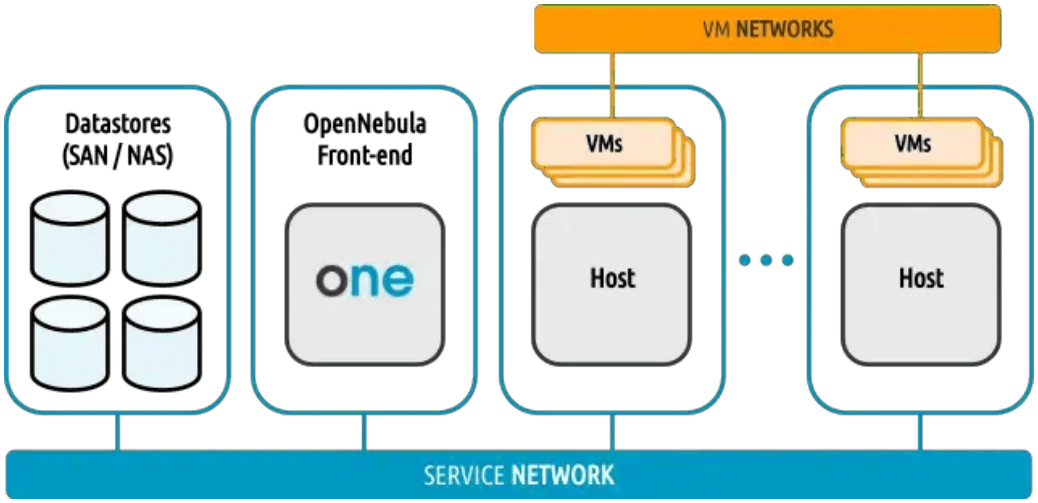
4. OpenNebula

4.1. General Description

OpenNebula, an open-source server virtualization solution founded in 2008, enables the deployment of public, private, and hybrid cloud computing infrastructures. OpenNebula receives a lot of appreciation due to its flexibility, simplicity, and cost efficiency. This platform, developed by OpenNebula Systems, is gaining popularity among companies of every size, right from small to medium-sized enterprises and big infrastructures. Performance should be extracted to the maximum using hardware solutions, mostly coming with higher prices, for optimizing infrastructure cost.

4.2. Installation

Installation of OpenNebula differs significantly from that of Proxmox. You remember that a cyber range consists of essential three components whith a frontend, a hypervisor, and a storage. In the case of Proxmox, those three components are usually installed on one physical infrastructure. In the case of OpenNebula, it is recommended that you install the three components (frontend, hypervisor, and storage) on different physical infrastructures. This way, which is the most documented one, is OpenNebula's favorite, especially for a deployment in a production environment where the choice of multi-infrastructure best allows better performance, greater scalability, and enhanced security. [5] [6]



4.3. Multi-Hypervisor Support

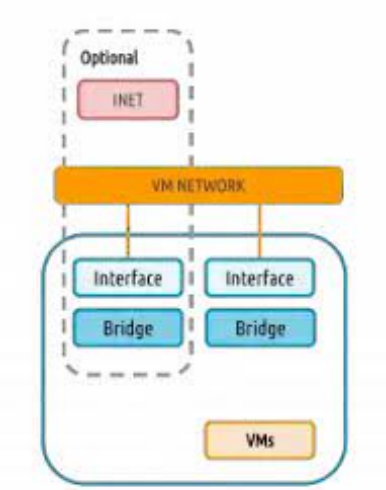
OpenNebula supports both KVM for full virtualization and LXC for container-based virtualization. Additionally, OpenNebula integrates with VMware vCenter, enabling users to manage vSphere environments alongside KVM and LXC. This integration offers great flexibility for hybrid cloud deployments. Although less commonly used, OpenNebula also supports Hyper-V, thereby facilitating the management of Windows- based virtualization.

4.3.1. VM creation, Configuration and Cloning

Unlike Proxmox, the OpenNebula web interface offers the ability to directly upload a qcow2 file as an image and use it as a template for virtual machines (VMs). Additionally, the cloning process is simplified, as all configurations such as VNC/SPICE ports and VM IP addresses are automated by OpenNebula. We only need to specify the number of VMs we want to instantiate.

4.4. Networking

OpenNebula makes advanced networking even more advanced with the integration of a broad set of Networking technologies to build flexible, high-performance network architectures. This is really useful for virtualized environments with Open vSwitch and VMware NSX.. It is especially through OVS that we can manage virtual bridges, allowing the connection of multiple virtual machines on the same hypervisor or across different hypervisors. OVS improves network performance and ensures better isolation between virtual machines compared to traditional bridging solutions.



4.5. Storage

- Shared Storage : Just like Proxmox, OpenNebula supports multiple shared storage solutions, such as NFS, iSCSI, and also GlusterFS.
- Distributed Storage : The system also integrates with Ceph to offer highly scalable and resilient storage solutions.
- Block and file storage : OpenNebula offers flexible support for different storage backends, ensuring high performance and continuous availability. This flexibility allows users to choose the storage solution that best meets their specific needs.

4.6. Cluster

OpenNebula does not form a virtualized environment automatically just because there are several nodes (In Proxmox a cluster is a configuration where two or more nodes are working together and participating in the same cluster under one management interface). The creation of clusters in the OpenNebula platform has to be explicit on the part of the user. A cluster is composed of hosts that share common storage and networking resources. A cluster is commonly defined in OpenNebula and Proxmox VE as a number of nodes that operate together to provide a unified

abstraction for management and operation. The setup and management of the clusters are, by design, different between the two. Cluster creation and management are part of the explicit work of OpenNebula, but in the case of Proxmox VE, when a number of nodes are defined to work together, a cluster is automatically created.

4.7. Web Interface

Let's now take a look at the OpenNebula web interface to start comparing and identifying the information it offers. This will give us a better understanding of the functionality available and the ease of use of the platform.

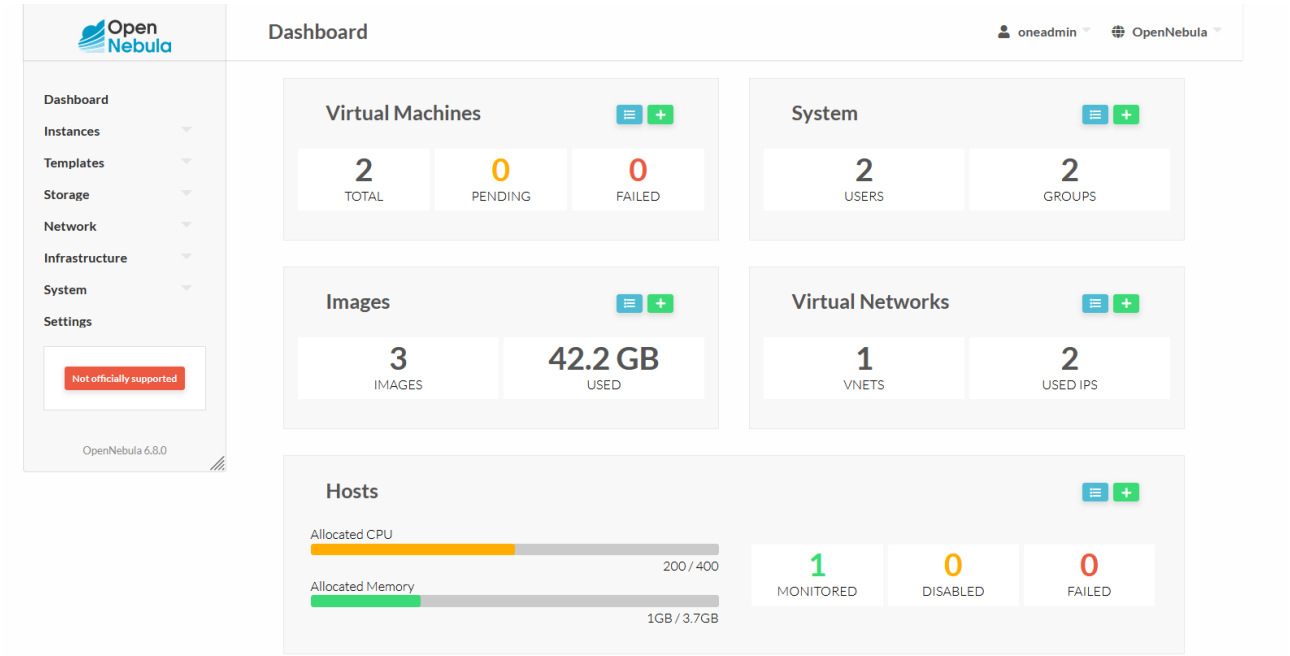


Figure 4.1. Dashboard

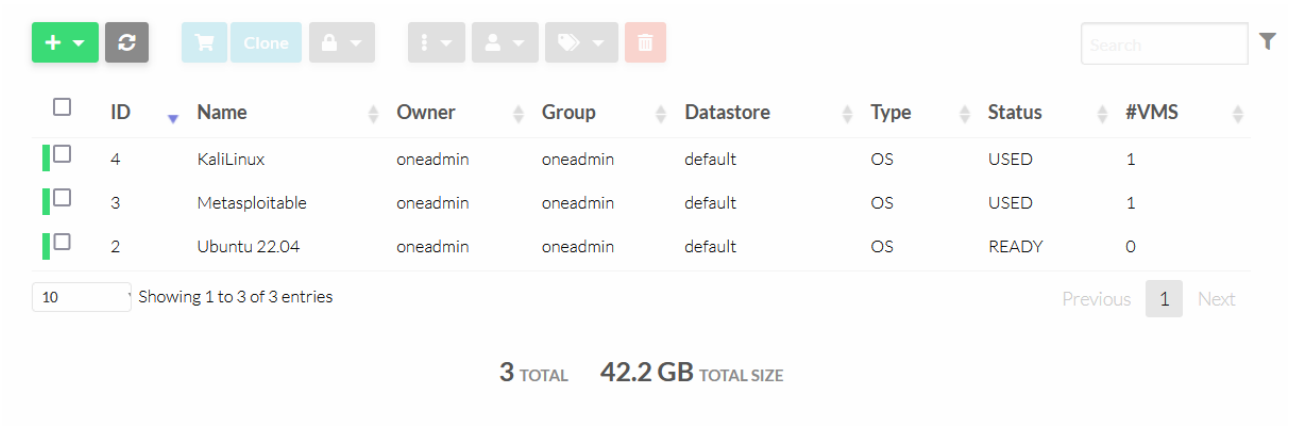


Figure 4.2. Image Iso

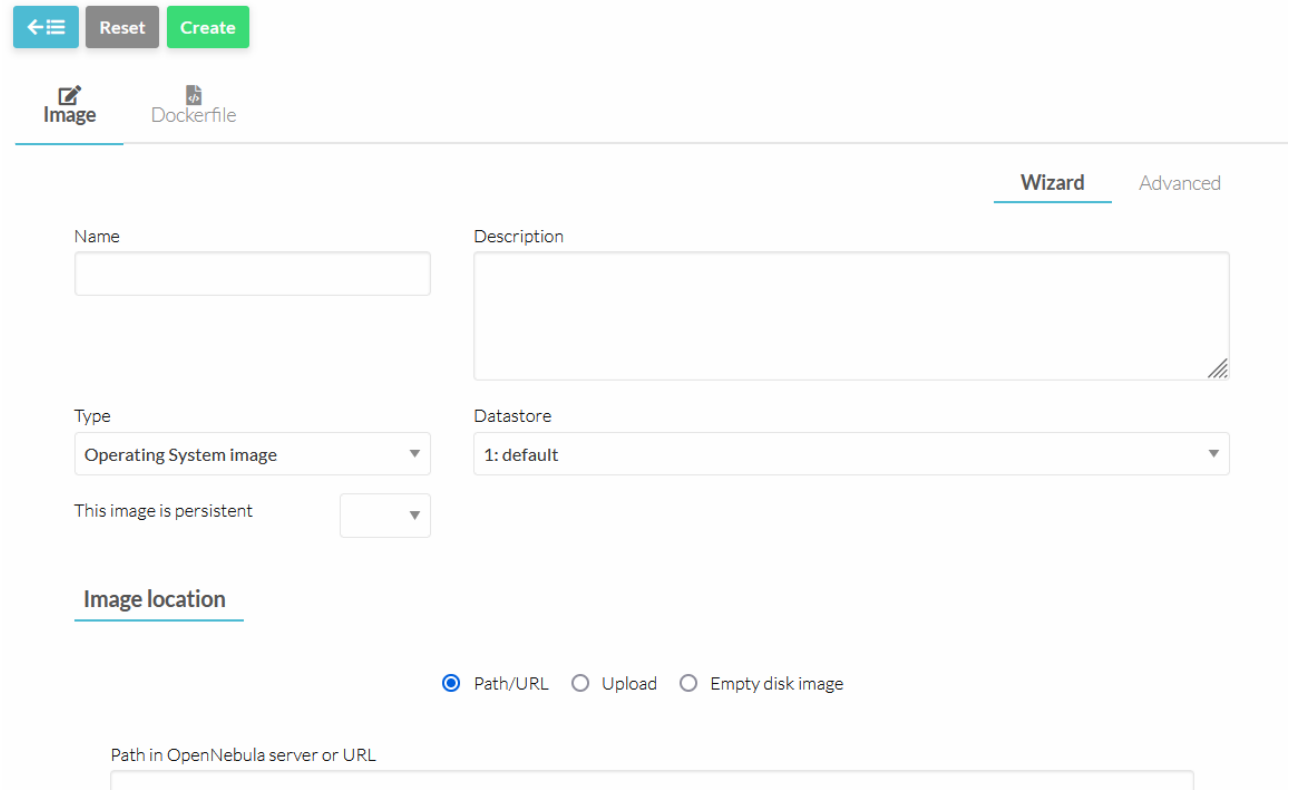


Figure 4.3. Upload image qcow2

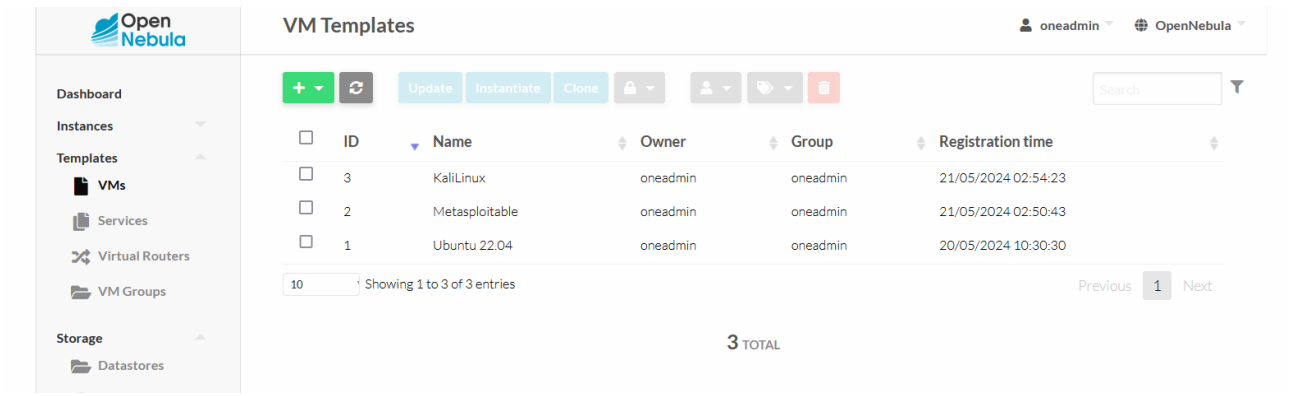


Figure 4.4. VM template based on image

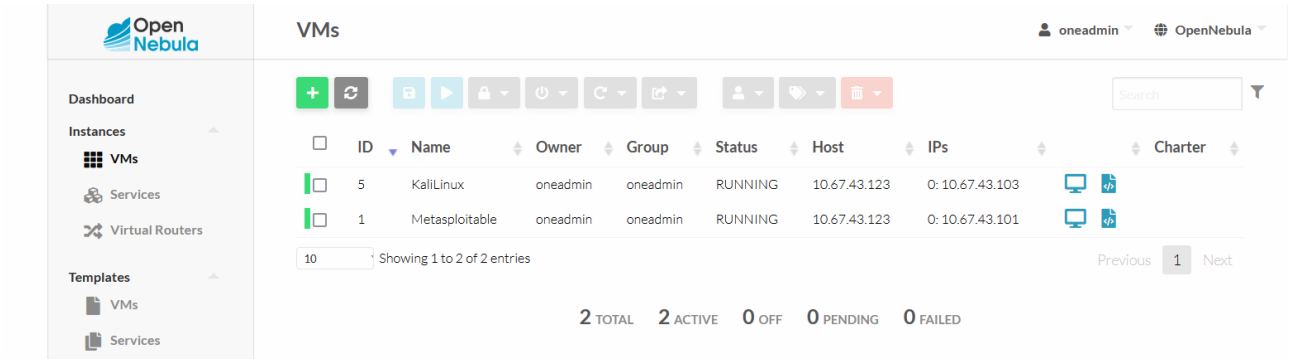


Figure 4.5. Create VM

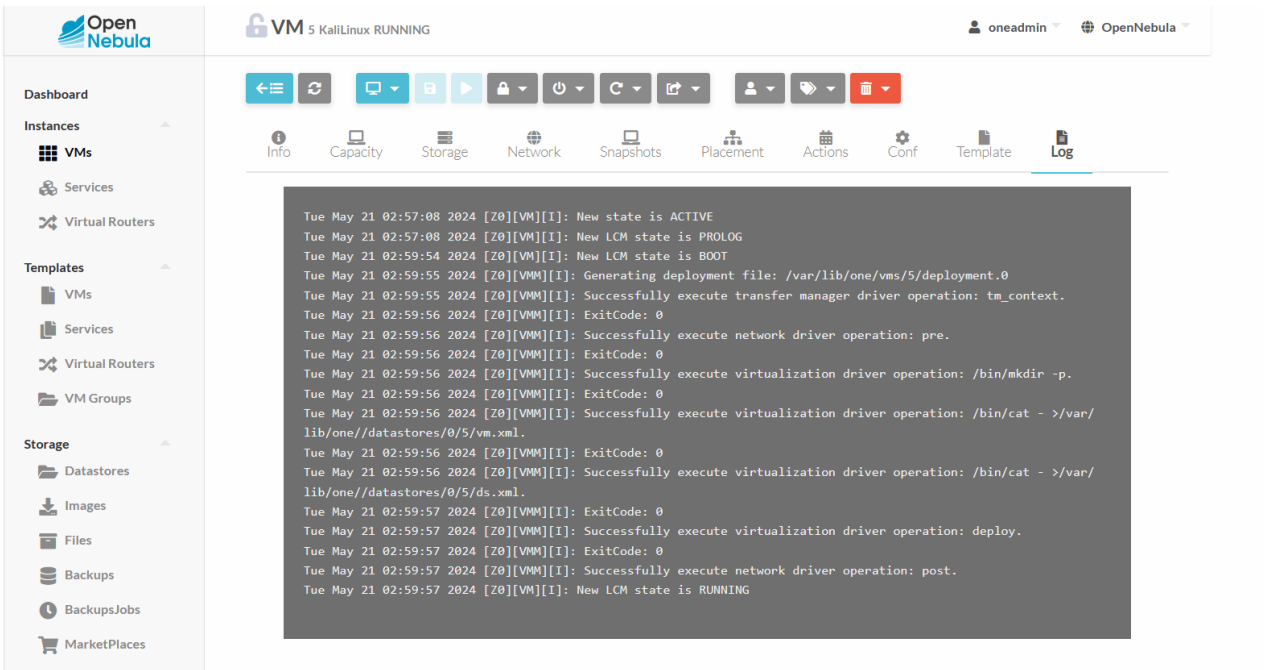


Figure 4.6. Log information

5. Comparison

We're now going to compare a few key points that will help us choose the most suitable platform in the context in which we plan to use these two systems.

5.1. Installation

Being one of the easy-to-install virtualization platforms, Proxmox VE is usually deployed on a single machine, so the management of its system is also simple and rather centralized. Such a setup is perfect for habitual environments that require a resilient, integrated virtualization solution with easy management. On the other hand, OpenNebula is designed to be set up on many machines, which reflects its nature as a tool to orchestrate and manage distributed environments at a large scale. OpenNebula in this multi-node setup is more resilient and flexible, better in fault tolerance and load distribution from several servers. Proxmox, on the other side, is more likely to be preferred because of its ease in a smaller or centralized environment, while OpenNebula is more attractive for advanced management of large, available infrastructures in the cloud.

But all of these bring associated costs : as the number of machines hosting OpenNebula increases, so, too, does complexity, and the risks of systems failure become magnified. Proxmox is installed usually in one machine, to avoid points of failure. OpenNebula multi-nodes demand more careful handling and greater amounts of coordination to ensure system stability.

An OpenNebula node failure may seriously damage resource availability and cluster performance if there are not enough mechanisms in place for failover, especially when loads are not balanced or data is not properly replicated. It demands solid settings for high availability, unyielding monitoring, and well-defined disaster recovery plans. Proxmox VE supports resilience and redundancy with clusters, unlike OpenNebula. Cluster architecture joins up several servers so that even if one fails, service will be offered by the virtual servers without them being down, based on the other nodes that offer service in the cluster. This kind of architecture helps in effective load distribution. With the live migration utility, one can smoothly migrate machines without causing downtime. Proxmox, with its user-friendly GUI and tools integrated into cluster management, tends to be a bit easier

to configure with high scores of availability in environments that are more uniform or centralized.

5.2. Hypervisor

OpenNebula has wide hypervisor support that includes KVM, VMware, and LXD, among others, which in part makes it wide for a variety of uses. This way, users can pick each hypervisor as required : the native performance of KVM, broad compatibility of VMware, or the efficiency of LXD containers. VMProx has put its major focus on KVM and LXC-based virtualization. With this approach, the solution has to deal with fewer hypervisor choices to work with, but much tighter integration with those concrete technologies, which generally dictates deep optimization and ease of management. So, OpenNebula will be a solution for many preferences for versatility and interoperability, and Proxmox will be a more specialized and consistent solution for whoever focuses on logical simplicity and optimized performance.

5.3. Interface

The web interfaces of Proxmox VE and OpenNebula are both rich in features and provide a wealth of valuable information for managing virtualized environments. Some features would clearly provide a lot of added useful information in managing virtualized environments at an enterprise level. However, there are a few specific advantages which OpenNebula boasts over the others, and all are useful when enterprise scenarios are considered. A clear example can be brought out in this case: OpenNebula can directly upload qcow2 from its web interface without the necessity to juggle command-line operations for this purpose. In literal terms, enough for the task, ease of use is pretty enough in the real sense for adding new disk image files to your virtualized environment.

In addition, OpenNebula provides easy access to VM logs through its web interface and allows monitoring and troubleshooting of virtual instances. The same interface provides administrators with a general view of the server to easily glance at the general status and performance of the infrastructure without getting lost in the details of the single components.

Furthermore, for dealing with direct file uploads and access to VM logs, OpenNebula makes it somewhat difficult. At the same time, Proxmox VE has a good

user interface. To summarize, Proxmox has an integrated terminal in the host's web interface, enabling the execution of system commands and the possibility of managing one's infrastructure without leaving the graphic interface in order to do a quick setup or solve a problem of a technical nature.

It is here that this becomes particularly valuable to those administrators who prefer a more hands-on approach or who, because of their needs, have to execute some of the operations not provided by the graphical interface. It offers extended operational flexibility and the users the ability to seamlessly switch from visual management to command-line control, satisfying demands in the best possible manner from both worlds. While OpenNebula centers around assisting the user in making certain operations simpler in an easy-to-process manner through an intuitive interface, Proxmox VE stands out specifically because it integrates an easy-to-use graphical interface with full access to command-line tools. In that duality, Proxmox is going to seem appealing to people in search of a virtualization platform that can combine visual simplicity with deep technical control.

5.4. Network

Regarding the configuration of networks in a virtualized environment, Proxmox VE and OpenNebula have taken different approaches to the implementation, which potentially affects the ease of use by end users. In the Proxmox VE environment, the process is made easier by the automatic creation of a network bridge—Linux Bridge—when setting up. Showcasing an inbuilt feature in the solution, the number of manual configuration steps lessens, which would otherwise be required to prepare an environment in readiness for VMs to communicate on the network. In so doing, an immediate installation can let an administrator start preparing, deploying, and managing virtual machines with no concern over the network configuration underlying the VMs.

OpenNebula's solution is a bit more flexible. However, it might also add a bit more complexity in case network bridges have to be set up manually by administrators for some not-so-clear reason. This high level of flexibility will serve very well in custom environments or massive deployment cases where needs around the network are not standard. On the other hand, this may be getting just a bit too complex for newbies or customers who would rather have a ready-to-use solution with the least possible initial effort in network configuration.

More specifically, the automatic Linux Bridge in Proxmox VE makes it easily and with maximum efficiency possible to perform the setting that administrators would

like not to carry out manually and to delegate such work to other dimensions of infrastructure management; on the other hand, OpenNebula, while not automated in setting network bridges, offers advanced customization options, which is quite important for more complicated deployments that may have their requirements for network settings. [7] This difference in the initial setup of the network shows a difference in priority between the two platforms: Proxmox VE keeps its attention on offering an integrated, ready-to-use solution, where time and effort to start getting productive with the system are kept at a minimum. OpenNebula, on the other hand, lays excess emphasis on flexibility and adaptability in its approach to allow setup in a freer way, which might be important for more sophisticated scenarios. To conclude, Proxmox VE is powerful due to its easiness and built-in approach, making a network configuration right away, while OpenNebula is flexible, enabling it to cross complex network configuration, although it might involve more work from the administrator.

5.5. Access to students

In an educational context, it is crucial to provide students with secure and direct access to their virtual machines while reserving full management access for teachers. Proxmox VE and OpenNebula integrate remote connection features such as noVNC, accessible directly through the web interface. This ease of access is essential for enabling students to work efficiently in a virtual environment while keeping administrative controls protected and reserved for teachers.

Proxmox VE and OpenNebula also offer the ability to configure granular access rights. This capability is particularly useful for limiting students' access to the management interface while allowing them to connect directly to their VMs via VNC clients. This system requires that each VM be configured with specific and secure VNC ports, and that students receive the necessary addresses and authentication information to access their machines. By ensuring that each VM assigned to a student has its own VNC or SPICE port, students can connect directly to their machines without needing access to the main Proxmox or OpenNebula interface.

It is also important to provide students with clear guides on how to use external VNC clients to connect to their virtual machines. These instructions help simplify the process for less experienced users and enhance students' autonomy in managing their virtual learning environment.

A notable advantage of Proxmox lies in its integrated shell command functionality and the scripts often accompanying the platform. These tools allow easy retrieval

of information files containing the IP addresses and VNC ports for each VM. This information can then be shared with students, facilitating access while maintaining a high level of security. Proxmox's ability to simplify the distribution of critical information to users while preserving security and ease of use makes it a beneficial choice in educational environments requiring efficient and secure management of virtual computing resources. [8]

5.6. Conclusion

After analyzing the cyber range platforms Proxmox VE and OpenNebula, it is clear that each platform offers specific advantages suited to different usage contexts. Proxmox VE stands out for its ability to operate efficiently on a single machine, offering simplicity and centralized management. This characteristic is particularly advantageous for environments that require a robust and integrated virtualization solution. With its easy installation and user-friendly interface, Proxmox is ideal for small environments or those that prioritize simplified management.

On the other hand, OpenNebula excels in managing large distributed infrastructures. It offers great resilience and flexibility, which are essential for good fault tolerance and efficient load distribution. However, this capability to operate on multiple machines can also complicate system management and require closer coordination, potentially increasing the risk of failures.

For the specific needs of the Royal Military Academy, which seeks to create an educational environment that is both effective and secure, Proxmox VE seems particularly well-suited. Its ease of use, combined with the richness of its management tools directly accessible through the graphical interface, makes Proxmox the ideal solution. This platform minimizes complex manipulations and allows flexible management between the visual interface and technical commands, a major asset in an educational context where ease of administration and reliability are paramount.

In conclusion, thanks to its simple installation, intuitive user interface, and advanced cluster management and network configuration features, Proxmox VE is particularly well-positioned to enhance the quality of learning and the educational experience at the Royal Military Academy.

Bibliography

[1] "Import virtualbox and virt manager vms into proxmox - ova, vmdk, qcow2 import," [Online]. Available: https://www.youtube.com/watch?v=clSyL2-5TtM&t=1017s.

[2] "Proxmox 8.1.x, installation pas à pas," [Online]. Available: https://www.youtube.com/watch?v=twRPCJrWzck&t=1084s.

[3] "Proxmox virtual environment," [Online]. Available: https://www.proxmox.com/en/proxmox-virtual-environment/overview.

[4] "Proxmox : Créer un cluster et démonstrations !," [Online]. Available: https://www.youtube.com/watch?v=UDqetfm6T0A&t=432s.

[5] J. Mutai, "Install opennebula frontend on debian 12/11/10," 2023. [Online]. Available: https://computingforgeeks.com/install-opennebula-frontend-on-debian-buster/.

[6] "Opennebula installation," [Online]. Available: https://docs.opennebula.io/6.8/installation_and_configuration/frontend_installation/index.html.

[7] J. Mutai, "Create and use bridged networks in opennebula vms," 2023. [Online]. Available: https://computingforgeeks.com/create-and-use-bridged-networks-in-opennebula/.

[8] "Vnc client access," [Online]. Available: https://pve.proxmox.com/wiki/VNC_Client_Access.