



Up and Running

by Brandon Lee

Blogger, consultant, and cloud architect at <https://www.virtualizationhowto.com>

Copyright © 2024 by Brandon Lee

All rights reserved.

No portion of this book may be reproduced in any form without written permission from the publisher or author, except as permitted by U.S. copyright law.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is distributed with the understanding that neither the author nor the publisher is engaged in rendering any type of professional services. While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The technical advice and strategies contained herein may not be suitable for your situation. You should consult with a technical professional when appropriate in your environment. Neither the publisher nor the author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, personal, or other damages.

Table of Contents

- [Section 1 - Nested Proxmox VMware installation in ESXi](#)
- [Section 2 - Proxmox 8.1 New Features and Download with Software-Defined Network and](#)
- [Section 3 -Secure Boot Upgrade Proxmox Host to 8.1: Tutorial & Steps](#)
- [Section 4 -Proxmox 8.2 New Features and Download](#)
- [Section 5 -Proxmox VE 8.3 New Features with OVA Import](#)
- [Section 6 -Proxmox Networking for VMware vSphere admins](#)
- [Section 7 -Proxmox Update No Subscription Repository Configuration](#)
- [Section 8 -Proxmox VLAN Configuration: Management IP, Bridge, and Virtual Machines](#)
- [Section 9 -Proxmox Management Interface VLAN tagging configuration](#)
- [Section 10 -Proxmox Create ISO Storage Location – disk space error](#)
- [Section 11 -Proxmox iSCSI target to Synology NAS](#)
- [Section 12 -Proxmox add disk storage space – NVMe drive](#)
- [Section 13 - Proxmox cluster installation and configuration](#)
- [Section 14 - Mastering Ceph Storage Configuration in Proxmox 8 Cluster](#)
- [Section 15 - CephFS Configuration in Proxmox Step-by-Step](#)
- [Section 16 - Proxmox HA Cluster Configuration for Virtual Machines](#)
- [Section 17 -Proxmox firewall setup and configuration](#)
- [Section 18 -Proxmox Container vs VM features and configuration](#)
- [Section 19 - Proxmox Containers with Fedora CoreOS Install](#)
- [Section 20 -Proxmox Helper Scripts you can use](#)
- [Section 21 -Proxmox scripts PowerShell Ansible and Terraform](#)
- [Section 22 -Proxmox Backup Server: Ultimate Install, Backup, and Restore Guide](#)
- [Section 23 -Proxmox SDN Configuration Step-by-Step](#)
- [Section 24 -New Proxmox Datacenter Manager Released Download and Install](#)
- [Section 25 -pfSense Proxmox Install Process and Configuration](#)
- [Section 26 - Nested ESXi install in Proxmox: Step-by-Step](#)

Nested Proxmox VMware installation in ESXi

In working with clients and different environments, you will definitely see many different hypervisors used across the landscape of enterprise organizations. While I recommend [VMware vSphere](#) for business-critical enterprise workloads to customers, there are use cases where I see other hypervisors used. Proxmox is a really great open-source, free hypervisor available for use and is even developed for use in enterprise applications. I also know of many in the community running Proxmox in their home lab environment. If you are like me and like to play around with technology, hypervisors, and other cool geeky stuff, I find I load a lot of different solutions in the lab. Let's take a look at nested Proxmox VMware installation in [ESXi](#) and see how you can easily spin up a Proxmox host in a vSphere VM.

What is Proxmox?

Proxmox is easily administered using a rich, fully-featured web interface that actually looks and feels nice. While it is not in my opinion where the vSphere client is in look and feel, it is quite nice and does the job needed to administer the Proxmox environment.

Proxmox VE is an open-source hypervisor platform for enterprise virtualization. It provides many features needed to run production workloads, including virtual machines, containers, software-defined storage, networking, clustering, and other capabilities out-of-the-box. It is based on Linux, so you get the pure Linux experience for virtualization, containers, and other facets. Note some of the benefits:

- Open-source software
- No vendor lock-in
- Linux kernel
- Fast and easy installation
- Easy-to-use with the intuitive web-based management interface
- Low administration costs and simple deployment
- Huge active community

Nested Proxmox VMware installation in ESXi

The first thing you need for your nested [Proxmox VMware](#) installation in ESXi is to download the Proxmox ISO for installation. You can download the Proxmox ISO here:

- [Get the free Proxmox VE ISO installer](#)
- Current version Proxmox VE 7.1

You will mount the ISO to your virtual machine in VMware vSphere like you would any other OS installation. Create a new VMware vSphere virtual machine with the following details:

- Guest OS Family – **Linux**
- Guest OS Version – **Debian GNU/Linux 11 (64-bit)**

Edit Settings | Proxmox

X

Virtual Hardware VM Options

General Options

VM Name	Proxmox
VM Config File	[ESX3DS01] Proxmox/Proxmox.vmx
VM Working Location	[ESX3DS01] Proxmox/
Guest OS Family	Linux
Guest OS Version	Debian GNU/Linux 11 (64-bit)
VMware Remote Console Options	<input type="checkbox"/> Lock the guest operating system when the last remote user disconnects
Encryption	Expand for encryption settings
Power management	Expand for power management settings
VMware Tools	Expand for VMware Tools settings
Boot Options	Expand for boot options
Advanced	Expand for advanced settings
Fibre Channel NPIV	Expand for Fibre Channel NPIV settings

CANCEL

OK

Proxmox VMware virtual machine settings

Next, make sure to expose hardware-assisted virtualization to the guest OS for your soon-to-be [Proxmox installation](#). As most of us are familiar with in our [nested ESXi](#) labs, this is a simple checkbox in the properties of your VMware ESXi virtual machine under the CPU settings.

Edit Settings | Proxmox

X

Virtual Hardware VM Options

ADD NEW DEVICE ▾

CPU		4	ⓘ
Cores per Socket	1	Sockets: 4	
CPU Hot Plug	<input type="checkbox"/> Enable CPU Hot Add		
Reservation	0	MHz	ⓘ
Limit	Unlimited	MHz	ⓘ
Shares	Normal	4000	ⓘ
Hardware virtualization	<input checked="" type="checkbox"/> Expose hardware assisted virtualization to the guest OS		
Performance Counters	<input type="checkbox"/> Enable virtualized CPU performance counters		
I/O MMU	<input type="checkbox"/> Enabled		
> Memory	8	GB	ⓘ
> Hard disk 1	40	GB	ⓘ
> SCSI controller 0	LSI Logic Parallel		
> Network adapter 1	DPG-Servers		<input checked="" type="checkbox"/> Connect...
> CD/DVD drive 1	Datastore ISO File		<input checked="" type="checkbox"/> Connect...
> Video card	Specify custom settings		

CANCEL

OK

Exposing CPU hardware virtualization to the guest OS

After booting from the ISO, the Proxmox VE 7.1 installation begins. Select to **Install Proxmox VE**.



Welcome to Proxmox Virtual Environment

[Install Proxmox VE](#)
[Install Proxmox VE \(Debug mode\)](#)
[Rescue Boot](#)
[Test memory \(Legacy BIOS\)](#)

Booting the Proxmos 7.1 VE installer

First things first. Accept the EULA to proceed.



Proxmox VE Installer

END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX VIRTUAL ENVIRONMENT (PROXMOX VE)

By using Proxmox VE software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintenance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox VE and any related update, source code and structure (the Programs), regardless of the delivery mechanism.

1. License. Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.
2. Limited Warranty. The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non-infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.
3. Limitation of Liability. To the maximum extent permitted under applicable law, under no

Abort

Previous

I agree

Accept the EULA for Proxmox VE 7.1

Next, you can customize the disk partition layout if you choose. However, for my [nested Proxmox](#) VMware installation, I am accepting the defaults.



Proxmox Virtual Environment (PVE)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data will be lost.

Press the Next button to continue the installation.

- **Please verify the installation target**
The displayed hard disk will be used for the installation.
Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface, via a web browser.

A screenshot of the Proxmox VE Installer interface. At the top, it says "Target Harddisk: /dev/sda (8GiB, Virtual disk) ▾ Options". Below that are two buttons: "Abort" on the left and "Previous" and "Next" on the right. The main area contains the text "Select the disk partitioning to be used with the Proxmox VE 7.1 installation".

Next up is setting your location and time zone configuration.

Location and Time Zone selection

The Proxmox Installer automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.



Configure the password for your **root** account. Also, Proxmox has you enter an email address.

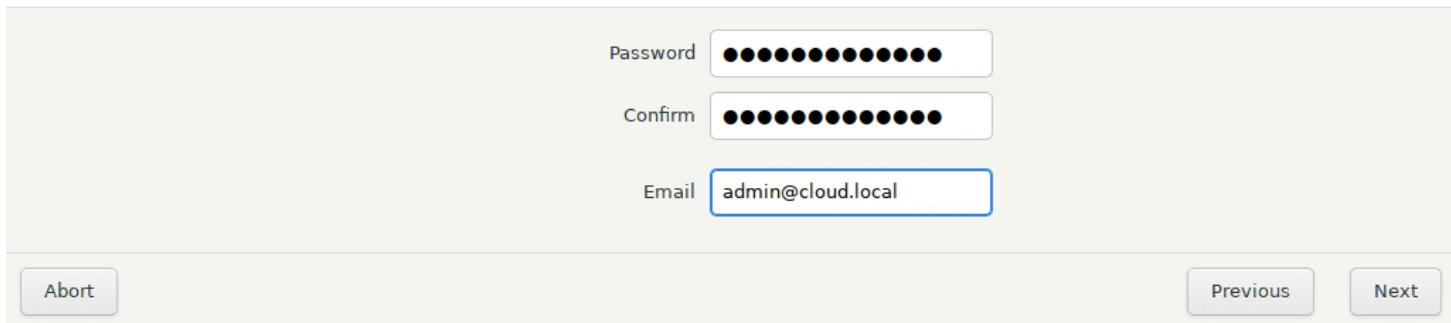
Administration Password and Email Address

Proxmox Virtual Environment is a full featured, highly secure GNU/Linux system, based on Debian.

In this step, please provide the *root* password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

Press the Next button to continue the installation.



A screenshot of the Proxmox VE Installer's configuration interface. It shows fields for entering a password and confirming it, as well as an email address. The 'Email' field contains 'admin@cloud.local'. Navigation buttons for 'Abort', 'Previous', and 'Next' are visible at the bottom.

Password	████████████████
Confirm	████████████████
Email	admin@cloud.local

Abort Previous Next

Set the administrator password and email address

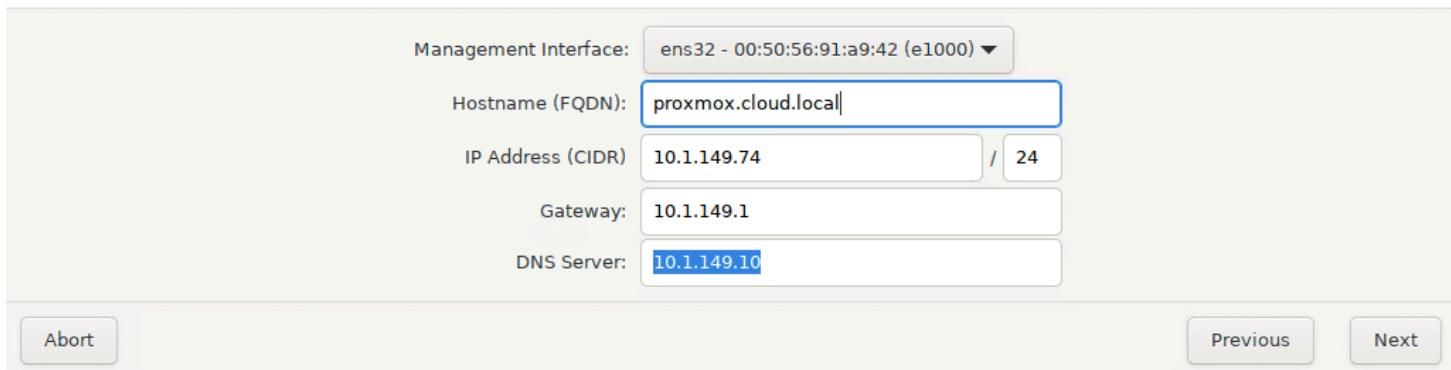
Configure the Proxmox hostname and your network configuration.

Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.



Management Interface: ens32 - 00:50:56:91:a9:42 (e1000)

Hostname (FQDN): proxmox.cloud.local

IP Address (CIDR): 10.1.149.74 / 24

Gateway: 10.1.149.1

DNS Server: 10.1.149.10

Abort Previous Next

Set the hostname and network configuration

Finally, we come to the Summary screen. Here, review the configuration and validate your settings. Then, click **Install**.



Proxmox VE Installer

Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	United States
Timezone:	America/Chicago
Keymap:	en-us
Email:	admin@cloud.local
Management Interface:	ens32
Hostname:	proxmox
IP CIDR:	10.1.149.74/24
Gateway:	10.1.149.1
DNS:	10.1.149.10

Automatically reboot after successful installation

[Abort](#)

[Previous](#)

[Install](#)

Summary of the Proxmox VE 7.1 installation

The installation process begins.



Proxmox VE Installer

Virtualize your IT Infrastructure

Proxmox VE is ready for enterprise deployments.

The role based permission management combined with the integration of multiple external authentication sources is the base for a secure and stable environment.

Visit www.proxmox.com for more information about commercial support subscriptions.

- **Commitment to Free Software**

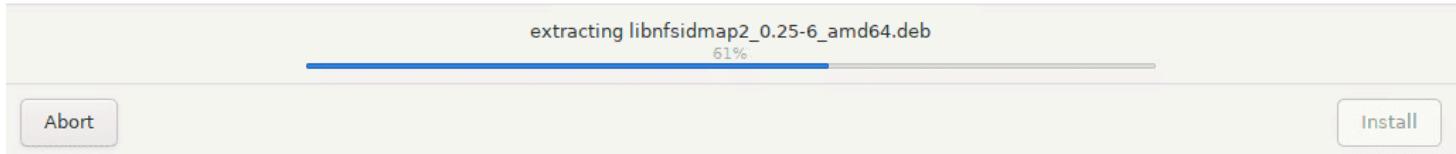
The source code is released under the GNU Affero General Public License.

- **RESTful web API**

Resource-oriented architecture (ROA) and declarative API definition using JSON Schema enable easy integration for third party management tools.

- **Virtual Appliances**

Pre-installed applications - up and running within a few seconds.



Proxmox VE 7.1 installation proceeds

After finishing the installation, the Proxmox server will reboot. Below is the boot screen captured as it reboots from the installation.

GNU GRUB version 2.04-20

```
*Proxmox VE GNU/Linux
Advanced options for Proxmox VE GNU/Linux
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Memory test (memtest86+, experimental multiboot)
Memory test (memtest86+, serial console 115200, experimental multiboot)
```

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 2s.

Proxmox VE 7.1 boots as a VMware ESXi VM

Finally, we are logged into the Proxmox web GUI using root and the password configured during the installation. Overall, the nested [Proxmox VMware](#) installation in ESXi was straightforward and easy. If you want to play around with Proxmox in a nested configuration, [VMware vSphere](#) provides a great way to do this using the basic functionality we have used for quite some time with nested ESXi installations.

The screenshot shows the Proxmox VE 7.1 web interface. The left sidebar is titled "Server View" and lists "Datacenter" and "proxmox". Under "proxmox", there are several sections: "local (proxmox)" and "local-lvm (proxmox)". The main content area is titled "Node 'proxmox'" and contains a "Summary" tab. The summary includes a "Package versions" section, a "CPU usage" chart, and system information like CPU(s), Kernel Version, PVE Manager Version, and Repository Status. A "Tasks" section at the bottom shows a single entry: "Jan 09 21:16:08 Jan 09 21:16:08 proxmox root@pam Start all VMs and Containers OK".

Logged into the Proxmox VE 7.1 web interface

Wrapping Up

Proxmox is a cool hypervisor that provides a lot of features in an open-source, freely available download. The latest Proxmox VE 7.1 release has a lot of out-of-the-box features and can be used to run production workloads. If you want to play around with Proxmox, running the hypervisor inside a nested virtual machine in VMware ESXi is a great way to gain experience with installing, operating, troubleshooting, and other aspects of the virtualization solution.

You can learn more about Proxmox from their official page found here:

- [Proxmox – Powerful open-source server solutions](#)

Proxmox 8.1 New Features and Download with Software-Defined Network and Secure Boot

The Proxmox 8.1 hypervisor has been released with great new features. The official information and documentation show it is a worthy upgrade for Proxmox 8 systems. Highlights include new software-defined network (SDN) features, secure boot, flexible notifications, and other new improvements. Let's dive into this release.

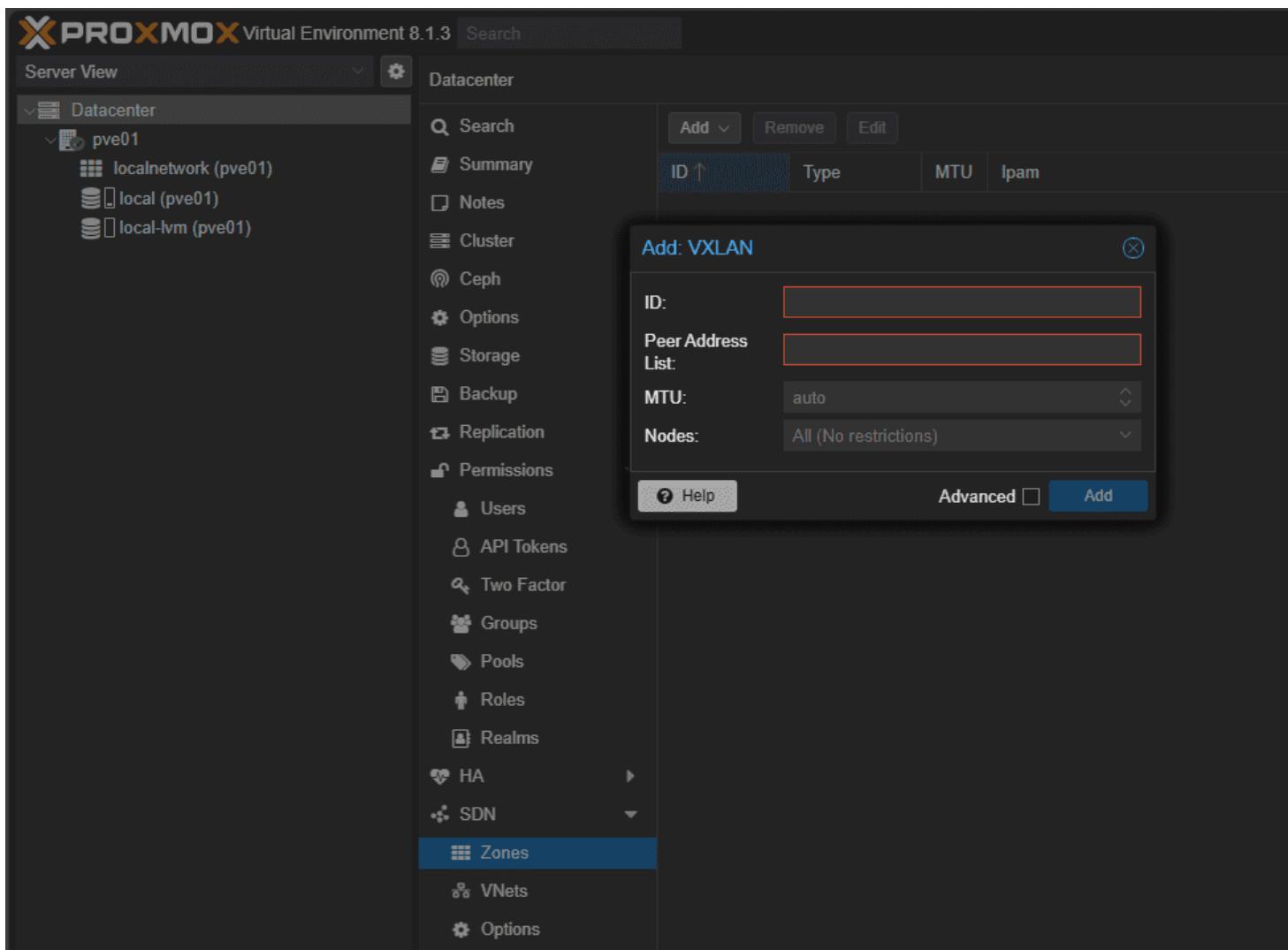
Table of contents

- [Software-Defined Networking in Proxmox VE 8.1](#)
- [Enhancing Security with Secure Boot Compatibility](#)
- [Introducing a Flexible Notification System support](#)
- [Kernel and Software Updates: Staying Ahead with Proxmox VE 8.1](#)
- [Comprehensive Support for Ceph Versions](#)
- [Simplifying Virtual Machine Management](#)
- [Download and Community Support](#)
- [Proxmox is Open Source with Professional Support available](#)
- [Great for home labs](#)
- [Wrapping up new Proxmox VE 8.1 features](#)

Software-Defined Networking in Proxmox VE 8.1

One of the top new features of [Proxmox VE 8.1 is its native support for software-defined](#) networking (SDN). Changes in this release, by default, the core SDN packages are now integrated into the Proxmox setup. This adds a more flexible, scalable networking solution within virtual environments and installations.

SDN in Proxmox VE 8.1 enables you to create virtual zones and networks, enabling you to manage and control complex networking configurations efficiently, right from the web interface. With this new feature, you can handle complex overlay [networks and enhance multi-tenancy setups](#).

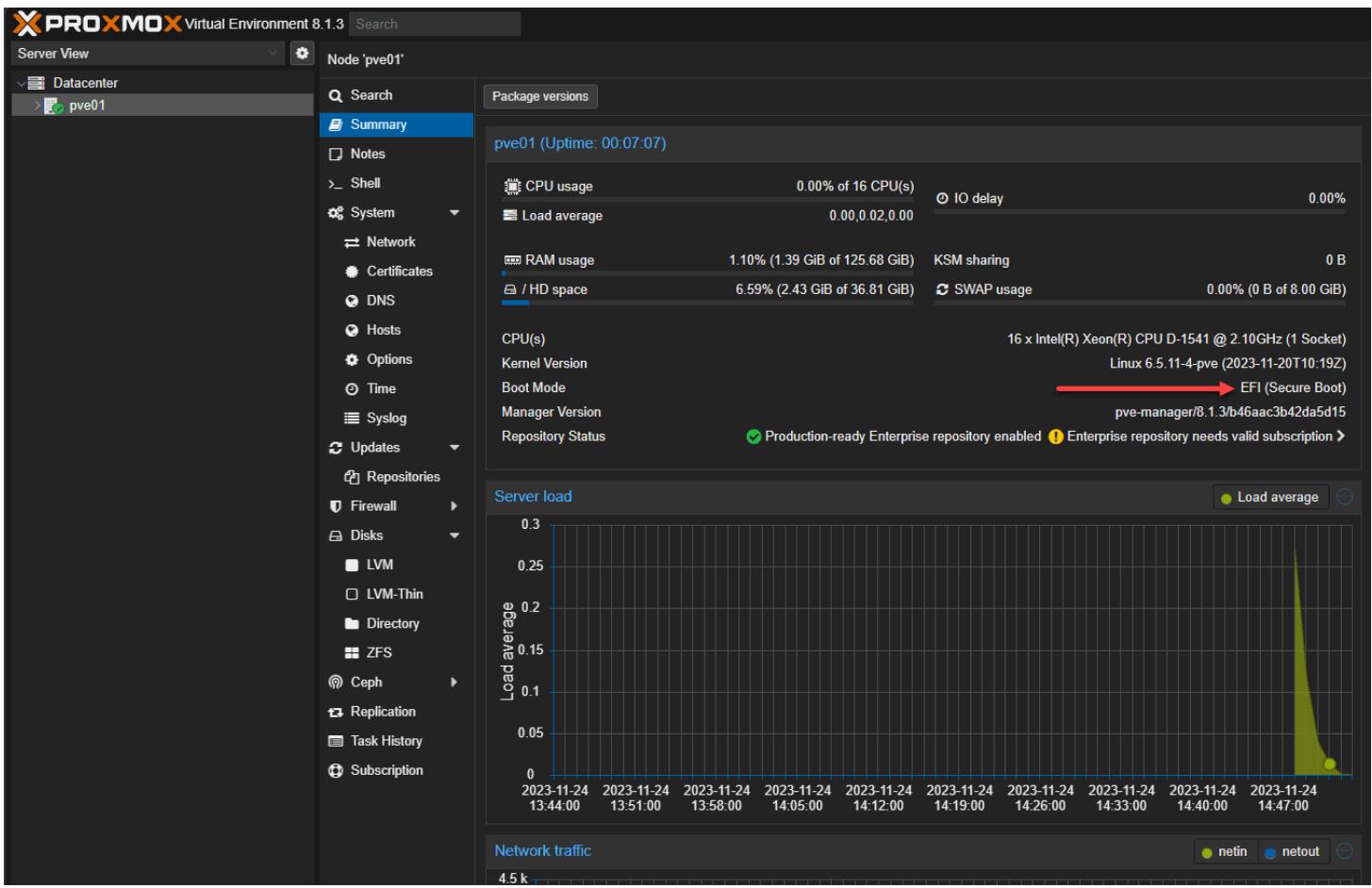


Software defined networking in proxmox 8.1

Enhancing Security with Secure Boot Compatibility

Security is enhanced in Proxmox VE 8.1 with the addition of support for Secure Boot. Secure Boot makes sure that only software with a valid digital signature is allowed to boot. This more secure boot process helps reduce the risk of unauthorized or malicious code execution in virtual machines.

Proxmox VE 8.1 now includes a signed shim bootloader, making it compliant with most hardware UEFI implementations. This feature is a great step forward in safeguarding virtualized data centers.



Efi secure boot enabled in proxmox 8.1

Introducing a Flexible Notification System support

Another new enhancement that many will be excited about is Proxmox VE 8.1 introduces a new, flexible [notification system](#) that employs a rules matcher-based approach to route notifications. This system allows users to specify various target types for receiving notifications.

It supports diverse notification channels, including local Postfix MTA, Gotify servers, and authenticated SMTP servers. The new granular control over notifications enhances system monitoring and response capabilities to system events.

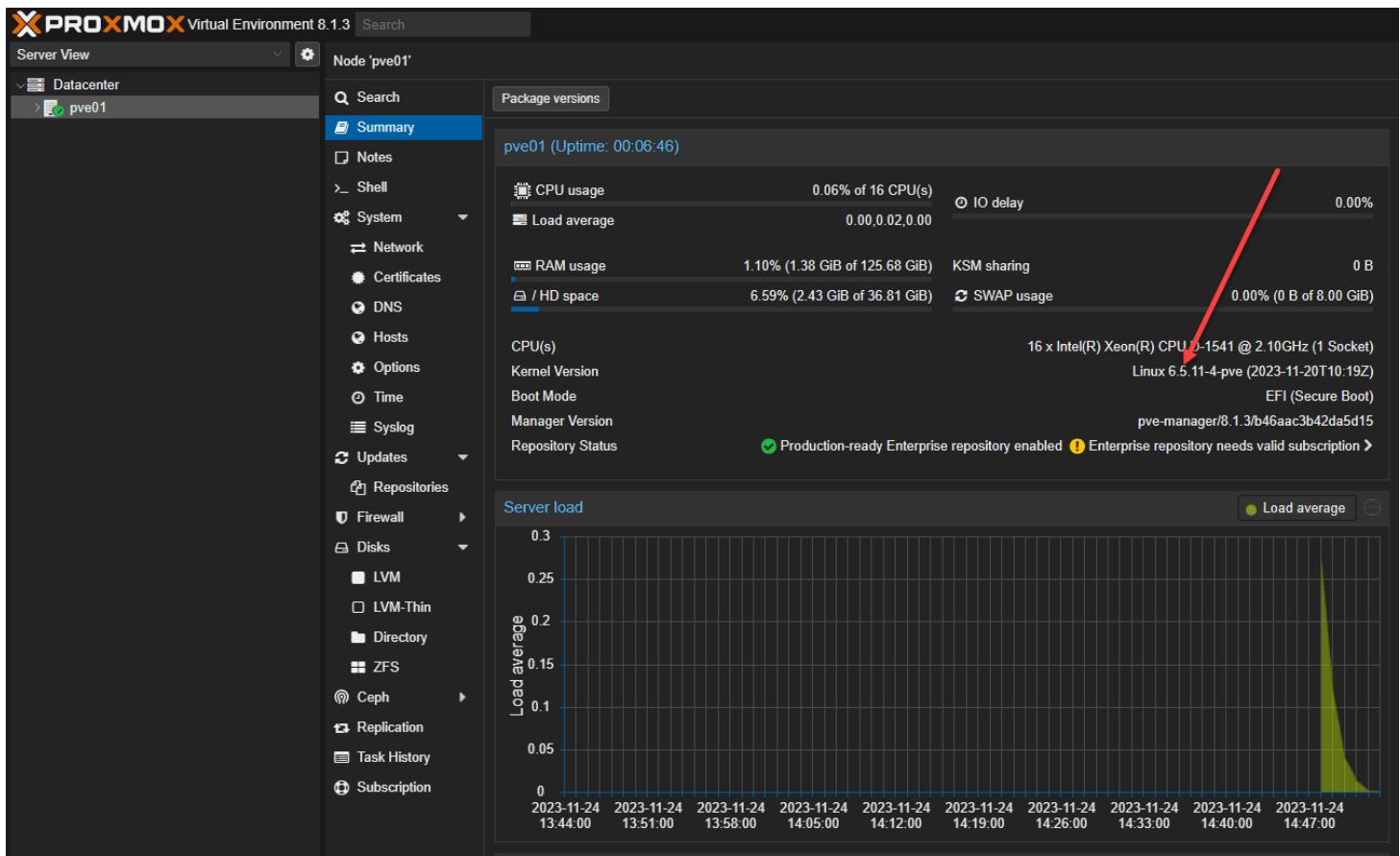
The screenshot shows the Proxmox VE 8.1 Datacenter interface. On the left, the navigation tree includes sections like Datacenter, pve01, localnetwork (pve01), local (pve01), local-lvm (pve01), and various system management options. A red box highlights the 'Notifications' section under the Datacenter menu. On the right, the main panel displays the 'Notification Targets' and 'Notification Matchers' sections. A red arrow points to the 'Add' button in the 'Notification Targets' header. The 'Notification Targets' table lists three entries: 'Notify' (Type: get Name ↑), 'Sendmail' (Type: mail-to-root), and 'SMTP' (Type: sendmail). The 'Notification Matchers' table lists one entry: 'default-matcher' (Enable checked, Comment: Route all notifications to mail-to-root).

New notification system support

Kernel and Software Updates: Staying Ahead with Proxmox VE 8.1

The new release is based on Debian 12.2, codenamed “Bookworm,” and includes a newer Linux kernel 6.5. Keeping up with the latest kernel helps ensure stability and performance. Proxmox VE 8.1 also includes updates to open-source technologies, such as QEMU 8.1, Ceph 18.2, and Open ZFS 2.2.

This will help to further enhance virtualization performance and storage technologies for virtualization tasks.



New linux kernel update with proxmox 8.1

Comprehensive Support for Ceph Versions

Proxmox VE 8.1 adds support for Ceph Reef 18.2.0 defaults and continues to provide compatibility with Ceph Quincy 17.2.7. This dual-version support provides flexibility in choosing the most appropriate Ceph version based on specific requirements and scenarios.

The screenshot shows the Proxmox VE 8.1 interface. On the left, the navigation tree shows 'Datacenter' and 'Node pve01'. Under 'pve01', there are several sub-nodes: 'localnetwork (pve01)', 'local (pve01)', and 'local-lvm (pve01)'. The main panel displays the 'Health' status of the node, which is currently 'Status' (indicated by a question mark icon). Below this, the 'Ceph Version:' field is empty. A modal dialog box is centered over the 'OSDs' section, containing the message: 'Ceph is not installed on this node. Would you like to install it now?'. A red arrow points from the text above to the 'Install Ceph' button in the dialog. The 'Status' section also includes 'Up' and 'Do' buttons. At the bottom of the screen, a 'Tasks' table is visible, listing recent activities:

Start Time	End Time	Node	User name	Description	Status
Nov 24 13:47:58		pve01	root@pam	Shell	OK
Nov 24 14:22:00	Nov 24 14:22:04	pve01	root@pam	Update package database	OK
Nov 24 13:46:28	Nov 24 13:46:32	pve01	root@pam	Update package database	OK
Nov 24 13:43:22	Nov 24 13:43:30	pve01	root@pam	Update package database	Error: command 'apt-get upd...
Nov 24 13:26:50	Nov 24 13:26:50	pve01	root@pam	Start all VMs and Containers	OK

Installing ceph in proxmox 8.1

Simplifying Virtual Machine Management

Proxmox VE 8.1 includes new bulk management features that make managing virtual machines more intuitive and efficient. It improves upon the “Bulk Actions” feature. These now include an option to suspend multiple guests simultaneously, adding new capabilities in streamlining administrative tasks.

Also, it adds a VirtIO [driver ISO image](#) that is now more straightforward and directly integrated into the VM creation wizard taking the heavy lifting out of this process.

Download and Community Support

Proxmox VE 8.1 is available for download from the official Proxmox website, complete with all features and capable of installation on bare-metal. The Proxmox community, with over 130,000 active members, continues to be a vibrant and supportive space for sharing knowledge and experiences.

Proxmox is Open Source with Professional Support available

As an open-source platform, Proxmox VE is licensed under the GNU Affero General Public License, v3, offering flexibility and freedom from vendor lock-in.

For enterprise users, Proxmox Server Solutions GmbH offers subscription-based support, ensuring access to tested updates and professional assistance.

Great for home labs

Many are already running Proxmox in their home lab environment. Proxmox is an excellent choice for home labbers who want a robust feature set for their lab VMs and self-hosted services and it is an open source virtualization platform. It

makes use of kernel based virtual machine (KVM).

The new Proxmox 8.1 features make it an even more appealing choice for running your critical self-hosted services. I have been running Proxmox in the [home lab for a few years now alongside other hypervisors](#) like vSphere. It is a great solution that allows you to run VMs and [LXC containers](#) without issue.

The web UI is fully-featured, and you can easily get to everything you need in the navigation links in the browser.

For me, I have had no major issues to report with great CPU performance and support for most project solutions I have installed. You can also passthrough your GPUs such as AMD and nVidia graphics cards. If you want to run a [Docker](#) container host, Proxmox makes for a great underlying hypervisor solution that you can also cluster with multiple hosts for HA, migration, and scalability purposes.

The Proxmox Backup server is also free to run and backup all your critical VM workloads. VM templates are available for quickly deploy various operating systems from the web-based console.

Wrapping up new Proxmox VE 8.1 features

Proxmox Virtual Environment 8.1 makes the Proxmox 8.x release even better with great new features and capabilities. The team at Proxmox is listening to what users and organizations need with Proxmox 8.1. [Features like secure](#) boot, SDWAN built-in, new kernel updates, better notification system and improved bulk operations, make this the best Proxmox VE release to date.

Upgrade Proxmox Host to 8.1: Tutorial & Steps

Let's look at the steps to upgrade your Proxmox host to Proxmox 8.1. In the example below, I will be upgrading an 8.0.3 host that I have running to 8.1 in the home lab.

Table of contents

- [New features](#)
- [No enterprise subscription prerequisites](#)
- [Proxmox 8.1 upgrade steps from the GUI](#)
- [Steps to upgrade from Proxmox 7.4 to Proxmox 8.1](#)
- [Mini PC running Proxmox](#)

New features

There are many new features to speak of in Proxmox 8.1. I just uploaded a post covering the new features. However, as a quick overview, the major new [features include](#):

- Software-defined networking
- Secure boot
- New bulk actions
- Upgraded Linux kernel
- A new flexible [notification system](#)
- [Upgraded Ceph Reef version](#)

No enterprise subscription prerequisites

If you are running [Proxmox in the home lab](#) and aren't running an enterprise subscription, which is how most home lab enthusiasts will be running, you need to reconfigure your update repositories. You may have already done this earlier for a lower-level Proxmox version. However, if you haven't already updated it to the "bookworm" repo, we will need to make that change, and then also change to the Ceph "reef" repo.

Update the following files in the comment lines:

```
#/etc/apt/sources.list.d/pve-enterprise.list
```

```
From: deb https://enterprise.proxmox.com/debian/pve bookworm enterprise
To: deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

```
#/etc/apt/sources.list.d/ceph.list
From: deb https://enterprise.proxmox.com/debian/ceph-quincy bookworm enterprise
To: deb http://download.proxmox.com/debian/ceph-reef bookworm no-subscription
```

Proxmox 8.1 upgrade steps from the GUI

After you have reconfigured the files above, you will need to **refresh** your updates. The following are Proxmox 8.1 [upgrade steps](#) using the GUI web interface.

First, click your Proxmox host in the GUI. Navigate to **System > Updates > Refresh**. When you click **Refresh**, it runs an "apt-get update".

The screenshot shows the Proxmox VE 8.0.3 interface. On the left, there's a sidebar with 'Server View' and a gear icon. Under 'Datacenter', 'pve' is selected, showing three sub-options: 'localnetwork (pve)', 'local (pve)', and 'local-lvm (pve)'. The main panel is titled 'Node 'pve'' and contains a 'Search' bar, a 'Summary' section, and a 'Notes' section. Below these are sections for 'Shell', 'System' (with sub-options like Network, Certificates, DNS, Hosts, Options, Time, and Syslog), 'Updates' (selected, with sub-options like Repositories, Firewall, Disks, and LVM), and 'Changelog'. The 'Updates' section displays a table of packages with columns for Package, Version, current, and new. The table includes packages like base-files, bind9-dnsutils, bind9-host, bind9-libs, curl, dbus, dbus-bin, dbus-daemon, dbus-session-bus-common, dbus-system-bus-common, debian-archive-keyring, debianutils, inetutils-telnet, krb5-locales, libc-bin, and libc-i18n. The 'Updates' section has a note at the bottom: 'Refresh updates after changing the repositories'.

Package ↑	Version	current	new
Origin: Debian (60 Items)			
base-files	12.4	12.4+deb1...	
bind9-dnsutils	1:9.18.12-1	1:9.18.19-1...	
bind9-host	1:9.18.12-1	1:9.18.19-1...	
bind9-libs	1:9.18.12-1	1:9.18.19-1...	
curl	7.88.1-10	7.88.1-10+...	
dbus	1.14.6-1	1.14.10-1~...	
dbus-bin	1.14.6-1	1.14.10-1~...	
dbus-daemon	1.14.6-1	1.14.10-1~...	
dbus-session-bus-common	1.14.6-1	1.14.10-1~...	
dbus-system-bus-common	1.14.6-1	1.14.10-1~...	
debian-archive-keyring	2023.3	2023.3+deb...	
debianutils	5.7-0.4	5.7-0.5~de...	
inetutils-telnet	2:2.4-2	2:2.4-2+de...	
krb5-locales	1.20.1-2	1.20.1-2+de...	
libc-bin	2.36-9	2.36-9+deb...	
libc-i18n	2.36-9	2.36-9+deb...	

Refresh updates after changing the repositories

You will see the **Task viewer** display the status of the apt-get update.

Task viewer: Update package database

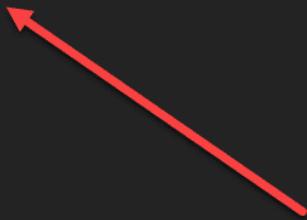


Output Status

Stop

Download

```
starting apt-get update
Hit:1 http://ftp.us.debian.org/debian bookworm InRelease
Get:2 http://security.debian.org bookworm-security InRelease [48.0 kB]
Get:3 http://ftp.us.debian.org/debian bookworm-updates InRelease [52.1 kB]
Get:4 http://security.debian.org bookworm-security/main amd64 Packages [104 kB]
Hit:5 http://download.proxmox.com/debian/ceph-reef bookworm InRelease
Hit:6 http://download.proxmox.com/debian/pve bookworm InRelease
Fetched 204 kB in 1s (301 kB/s)
Reading package lists...
TASK OK
```



The status of the apt get update from the GUI

After refreshing the updates, you can click the **Upgrade** button.

The screenshot shows the Proxmox VE 8.0.3 interface. On the left, there's a sidebar with 'Server View' at the top, followed by 'Datacenter' which is expanded to show 'pve' (selected), 'localnetwork (pve)', 'local (pve)', and 'local-lvm (pve)'. Below these are sections for 'Shell', 'System' (which is expanded to show 'Network', 'Certificates', 'DNS', 'Hosts', 'Options', 'Time', and 'Syslog'), 'Updates' (selected and highlighted in blue), 'Repositories', 'Firewall', 'Disks' (expanded to show 'LVM'), and 'LVM'. The main content area is titled 'Node 'pve'' and shows a table of updates. The table has columns for 'Package ↑', 'Version' (with 'current' and 'new' sub-columns), and 'Description'. A red arrow points to the 'Upgrade' button in the top right of this area. The table lists packages like 'base-files', 'bind9-dnsutils', 'bind9-host', etc., with their current and new versions and descriptions.

Package ↑	Version	Description	
	current	new	
Origin: Debian (60 Items)			
base-files	12.4	12.4+deb1...	Debian base system miscell...
bind9-dnsutils	1:9.18.12-1	1:9.18.19-1...	Clients provided with BIND 9...
bind9-host	1:9.18.12-1	1:9.18.19-1...	DNS Lookup Utility
bind9-libs	1:9.18.12-1	1:9.18.19-1...	Shared Libraries used by BIND 9...
curl	7.88.1-10	7.88.1-10+...	command line tool for transfer...
dbus	1.14.6-1	1.14.10-1~...	simple interprocess messaging...
dbus-bin	1.14.6-1	1.14.10-1~...	simple interprocess messaging...
dbus-daemon	1.14.6-1	1.14.10-1~...	simple interprocess messaging...
dbus-session-bus-common	1.14.6-1	1.14.10-1~...	simple interprocess messaging...
dbus-system-bus-common	1.14.6-1	1.14.10-1~...	simple interprocess messaging...
debian-archive-keyring	2023.3	2023.3+de...	GnuPG archive keys of the D...
debianutils	5.7-0.4	5.7-0.5~de...	Miscellaneous utilities specific...
inetutils-telnet	2:2.4-2	2:2.4-2+de...	telnet client
krb5-locales	1.20.1-2	1.20.1-2+d...	internationalization support fo...
libc-bin	2.36-9	2.36-9+deb...	GNU C Library: Binaries
libc-i18n	2.36-9	2.36-9+deb...	GNU C Library: localization fo...

Kicking off the upgrade from the proxmox GUI

It will launch another browser window displaying the prompt for you to enter **Y** to confirm you want to continue the [upgrade process](#).

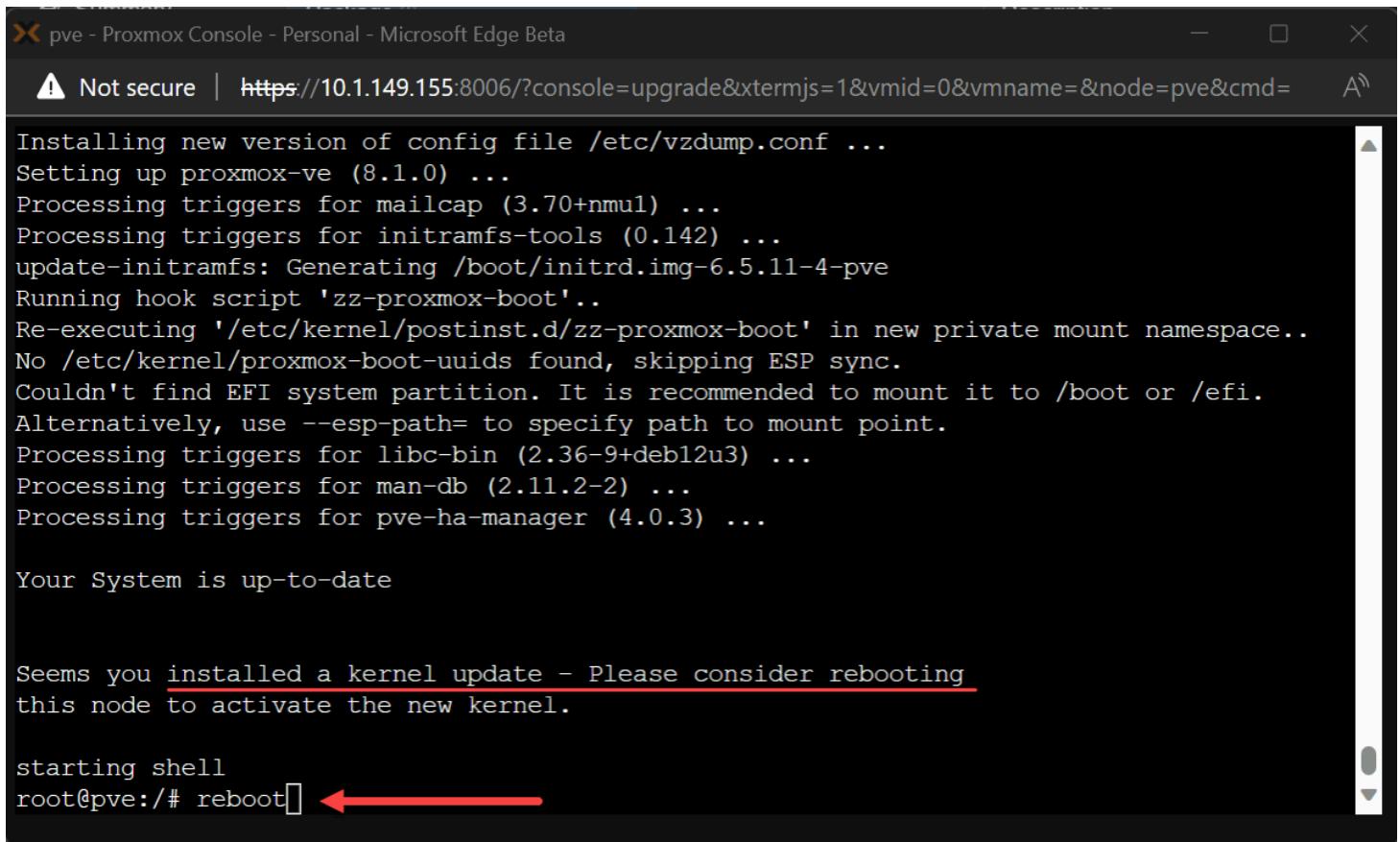
pve - Proxmox Console - Personal - Microsoft Edge Beta

⚠ Not secure | https://10.1.149.155:8006/?console=upgrade&xtermjs=1&vmid=0&vmname=&node=pve&cmd= A

```
libcephfs2 libcurl3-gnutls libcurl4 libdbus-1-3 libgssapi-krb5-2
libgstreamer-plugins-base1.0-0 libjs-extjs libk5crypto3 libknet1 libkrb5-3
libkrb5support0 libldb2 libnftables1 libnozze1 libnss-systemd libnvpair3linux
libpam-modules libpam-modules-bin libpam-runtime libpam-systemd libpam0g
libproxmox-acme-perl libproxmox-acme-plugins libproxmox-rs-perl libpve-access-control
libpve-cluster-api-perl libpve-cluster-perl libpve-common-perl
libpve-guest-common-perl libpve-http-server-perl libpve-rs-perl libpve-storage-perl
librados2 librados2-perl libradosstriper1 librbd1 librgw2 libsmclient libssl3
libsystemd-shared libsystemd0 libudev1 libunbound8 libutil3linux libwbclient0
libx11-6 libx11-data libx11-xcb1 libxml2 libzfs4linux libzpool5linux locales nftables
novnc-pve openssh-client openssh-server openssh-sftp-server openssl postfix
proxmox-backup-client proxmox-backup-file-restore proxmox-kernel-helper
proxmox-mail-forward proxmox-ve proxmox-widget-toolkit pve-cluster pve-container
pve-docs pve-edk2-firmware pve-firewall pve-firmware pve-ha-manager pve-i18n
pve-kernel-6.2 pve-manager pve-qemu-kvm pve-xtermjs python3-ceph-argparse
python3-ceph-common python3-cephfs python3-rados python3-rbd python3-rgw qemu-server
samba-common samba-libs smbclient spl ssh systemd systemd-boot systemd-boot-efi
systemd-sysv udev zfs-initramfs zfs-zed zfsutils-linux
122 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/436 MB of archives.
After this operation, 846 MB of additional disk space will be used.
Do you want to continue? [Y/n] [ ]
```

Press y to continue the upgrade process

After all the upgrade process is complete, you will [see the note that a new kernel was installed](#) and a reboot is required to instantiate the new kernel. Here I am typing **reboot** from the window.



```
pve - Proxmox Console - Personal - Microsoft Edge Beta
⚠ Not secure | https://10.1.149.155:8006/?console=upgrade&xtermjs=1&vmid=0&vmname=&node=pve&cmd=
Installing new version of config file /etc/vzdump.conf ...
Setting up proxmox-ve (8.1.0) ...
Processing triggers for mailcap (3.70+nmul) ...
Processing triggers for initramfs-tools (0.142) ...
update-initramfs: Generating /boot/initrd.img-6.5.11-4-pve
Running hook script 'zz-proxmox-boot'...
Re-executing '/etc/kernel/postinst.d/zz-proxmox-boot' in new private mount namespace..
No /etc/kernel/proxmox-boot-uuids found, skipping ESP sync.
Couldn't find EFI system partition. It is recommended to mount it to /boot or /efi.
Alternatively, use --esp-path= to specify path to mount point.
Processing triggers for libc-bin (2.36-9+deb12u3) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for pve-ha-manager (4.0.3) ...

Your System is up-to-date

Seems you installed a kernel update - Please consider rebooting
this node to activate the new kernel.

starting shell
root@pve:/# reboot[]
```

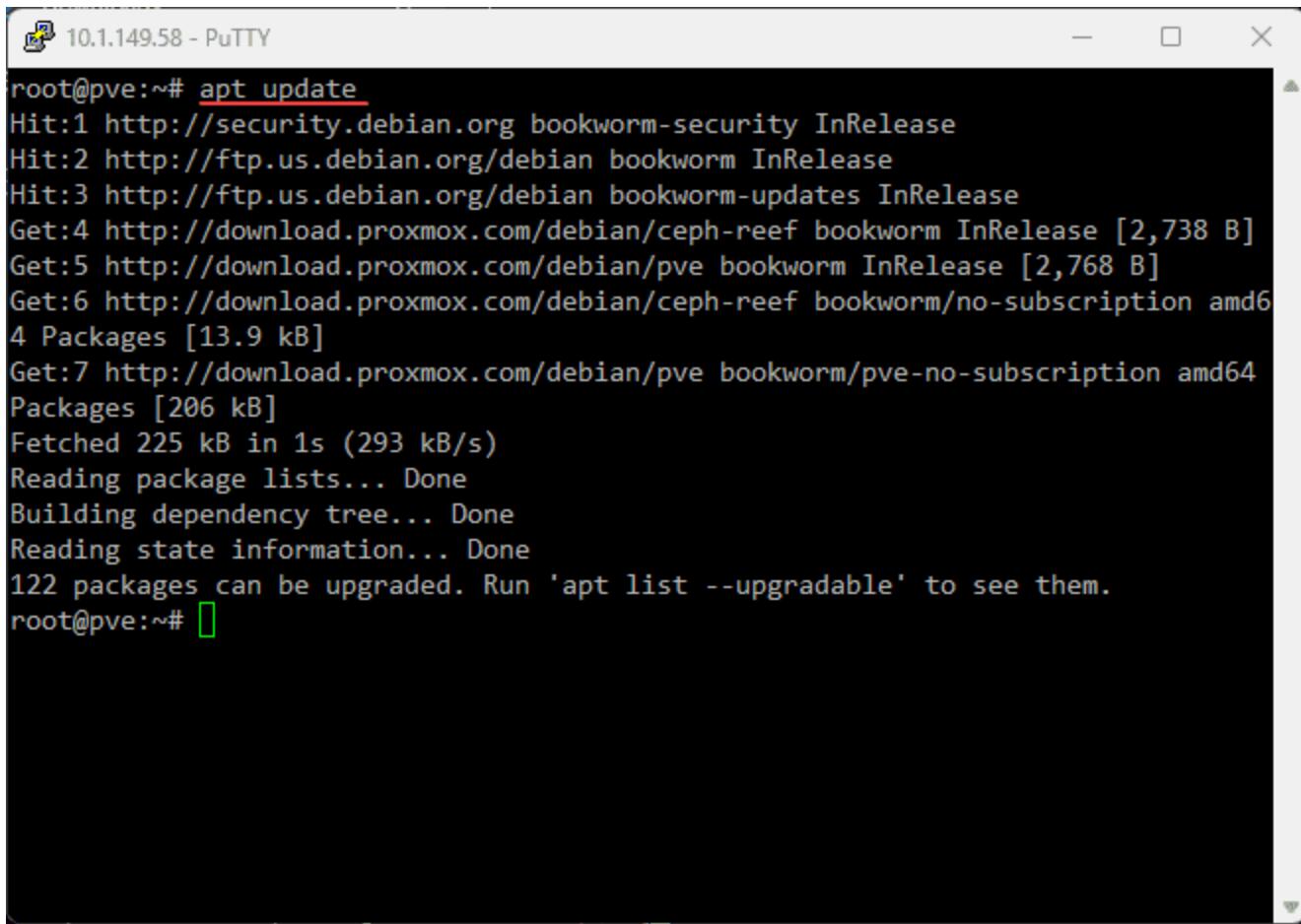
Reboot after the proxmox 8.1 upgrade and the kernel upgrade

Proxmox 8.1 upgrade steps from the command line

The upgrade steps from the command line are very simple. We just run the commands the GUI runs for us from the command line.

First we **refresh** the updates after we have updated the repository URLs. To do that, run the following commands:

```
apt update
```

A screenshot of a PuTTY terminal window titled "10.1.149.58 - PuTTY". The window shows the output of the "apt update" command being run on a root shell. The output includes details about hits from various repositories, file sizes, and package counts, followed by a summary message indicating 122 packages can be upgraded.

```
root@pve:~# apt update
Hit:1 http://security.debian.org bookworm-security InRelease
Hit:2 http://ftp.us.debian.org/debian bookworm InRelease
Hit:3 http://ftp.us.debian.org/debian bookworm-updates InRelease
Get:4 http://download.proxmox.com/debian/ceph-reef bookworm InRelease [2,738 B]
Get:5 http://download.proxmox.com/debian/pve bookworm InRelease [2,768 B]
Get:6 http://download.proxmox.com/debian/ceph-reef bookworm/no-subscription amd64
4 Packages [13.9 kB]
Get:7 http://download.proxmox.com/debian/pve bookworm/pve-no-subscription amd64
Packages [206 kB]
Fetched 225 kB in 1s (293 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
122 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@pve:~#
```

Running the apt update command to refresh the available updates

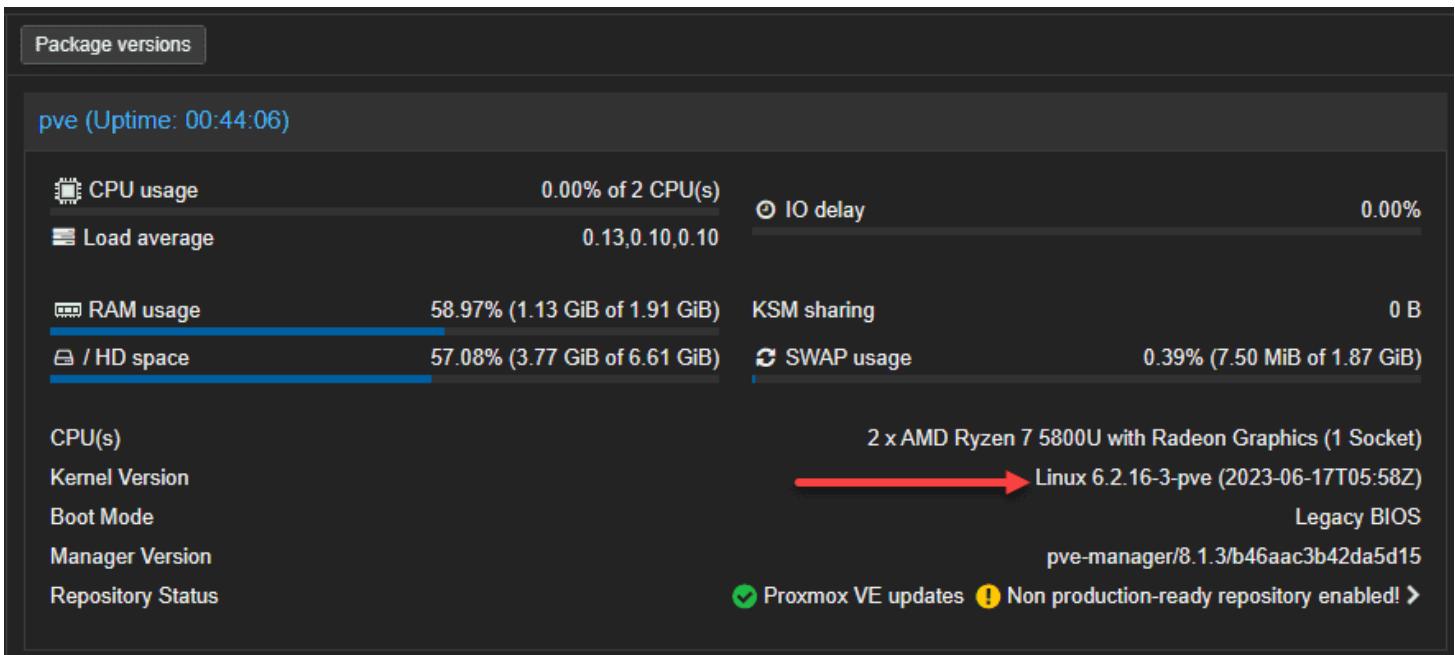
Next, we run the following command:

```
apt dist upgrade
```

```
root@pve:~# apt dist-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  pve-kernel-6.2
Use 'apt autoremove' to remove it.
The following NEW packages will be installed:
  fonts-font-logos libnet-subnet-perl libpve-network-perl libpve-notify-perl
  libsocket6-perl proxmox-default-kernel proxmox-kernel-6.2
  proxmox-kernel-6.2.16-19-pve proxmox-kernel-6.5
  proxmox-kernel-6.5.11-4-pve-signed proxmox-termproxy
  pve-edk2-firmware-legacy pve-edk2-firmware-ovmf
The following packages will be upgraded:
  base-files bind9-dnsutils bind9-host bind9-libs ceph-common ceph-fuse curl
  dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common
  debian-archive-keyring debianutils grub-common grub-efi-amd64-bin grub-pc
  grub-pc-bin grub2-common ifupdown2 inetutils-telnet krb5-locales libc-bin
  libc-110n libc6 libcephfs2 libcurl3-gnutls libcurl4 libdbus-1-3
  libgssapi-krb5-2 libgstreamer-plugins-base1.0-0 libjs-extjs libk5crypto3
  libknet1 libkrb5-3 libkrb5support0 libldb2 libnftables1 libnozle1
  libnss-systemd libnvpair3linux libpam-modules libpam-modules-bin
  libpam-runtime libpam-systemd libpam0g libproxmox-acme-perl
  libproxmox-acme-plugins libproxmox-rs-perl libpve-access-control
  libpve-cluster-api-perl libpve-cluster-perl libpve-common-perl
  libpve-guest-common-perl libpve-http-server-perl libpve-rs-perl
  libpve-storage-perl librados2 librados2-perl libradosstriper1 librbd1
  librgw2 libsmclient libssl13 libsystemd-shared libsystemd0 libudev1
  libunbound8 libutil3linux libwbclient0 libx11-6 libx11-data libx11-xcb1
  libxml2 libzfs4linux libzpool5linux locales nftables novnc-pve
  openssh-client openssh-server openssh-sftp-server openssl postfix
  proxmox-backup-client proxmox-backup-file-restore proxmox-kernel-helper
  proxmox-mail-forward proxmox-ve proxmox-widget-toolkit pve-cluster
  pve-container pve-docs pve-edk2-firmware pve-firewall pve-firmware
  pve-ha-manager pve-i18n pve-kernel-6.2 pve-manager pve-qemu-kvm pve-xtermjs
  python3-ceph-argparse python3-ceph-common python3_cephfs python3-rados
  python3-rbd python3-rgw qemu-server samba-common samba-libs smbclient spl
  ssh systemd systemd-boot systemd-boot-efi systemd-sysv udev zfs-initramfs
  zfs-zed zfsutils-linux
122 upgraded, 13 newly installed, 0 to remove and 0 not upgraded.
Need to get 436 MB of archives.
After this operation, 846 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Running the apt dist upgrade

After the upgrade is successful from the command line, if you look at your Proxmox host summary, you will see it has upgraded to 8.1.3, but the Linux kernel is still at version **6.2**. So, we need to reboot.



Before we reboot the kernel still shows 6.2

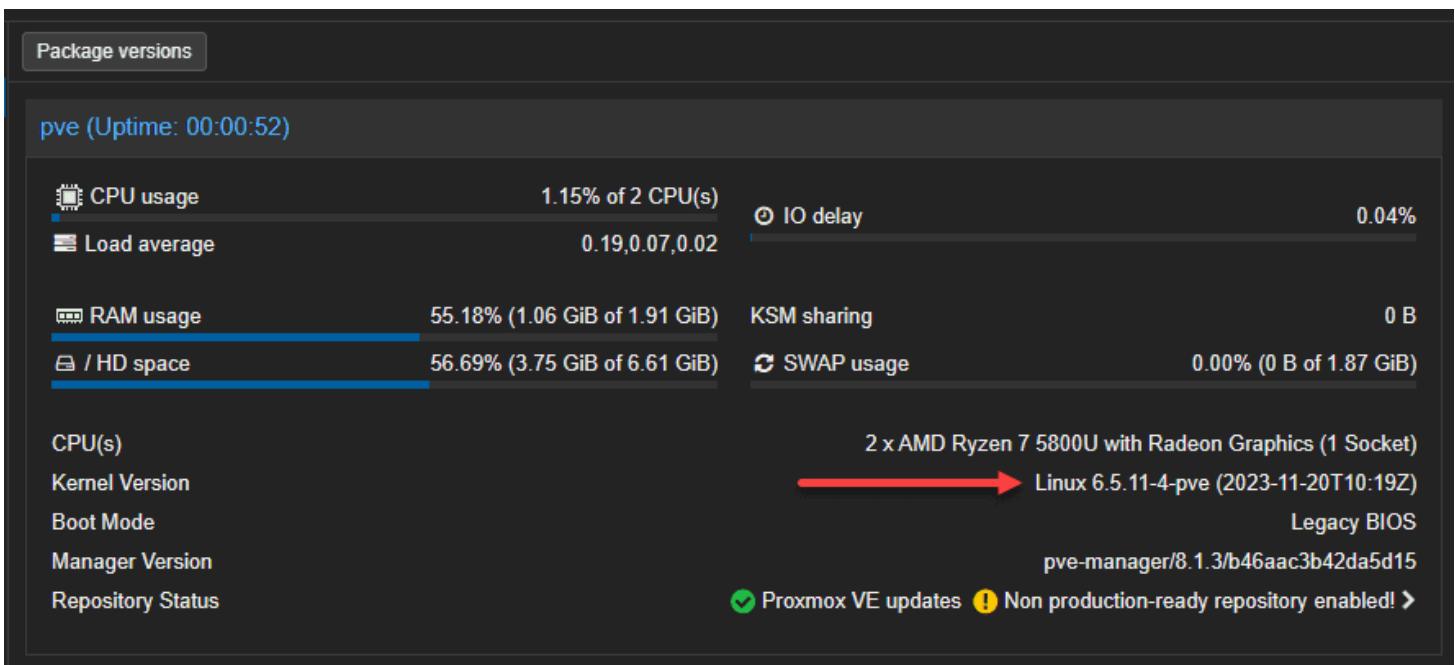
From the command line issue the reboot command:

```
reboot
```

```
root@pve:~# reboot ←
[  1.000000] Alternative, use --esp-path= to specify path to mount point.
[  1.000000] Processing triggers for libc-bin (2.36-9+deb12u3) ...
[  1.000000] Processing triggers for man-db (2.11.2-2) ...
[  1.000000] Processing triggers for pve-ha-manager (4.0.3) ...
[  1.000000] root@pve:~# reboot
[  1.000000] root@pve:~# [ ]
```

Running the reboot command to reboot proxmox and install the new kernel

Now, we can check the kernel version again and we see the Linux 6.5 kernel has been installed.



After the reboot the new linux 6.5 kernel has been installed

Steps to upgrade from Proxmox 7.4 to Proxmox 8.1

The [steps](#) to upgrade from Proxmox 7.4 to Proxmox 8.1 are fairly straightforward. However, it does involve more steps if you are currently running Ceph Quincy.

First, you will need to upgrade Ceph from Pacific to Quincy. The next step involves upgrading Proxmox VE from version 7.4 to 8.1. In the last step, once you have Proxmox VE 8.1 running, you will upgrade your Ceph installation to Reef.

Here are the links to the official documentation on those specific steps:

- [Ceph Pacific to Quincy Upgrade Guide](#)
- [Upgrading from Proxmox VE 7 to 8](#)
- [Ceph Quincy to Reef Upgrade Guide](#)

This page intentionally left blank.

Proxmox 8.2 New Features and Download

Proxmox has a lot of momentum behind it as of recently. We had the recent news of the new Proxmox import wizard to import VMware ESXi guests in Proxmox virtual environment. Also, Proxmox 8.1 wasn't released that long ago. However, Proxmox has now released Proxmox 8.2 new features and you can now download the ISO for installation. Let's take a look at the new features contained in this release.

Table of contents

- [Proxmox 8.2 New features](#)
 - [1. Automated and Unattended Installations](#)
 - [2. VMware ESX VM Import Wizard](#)
 - [3. nftables Firewall](#)
 - [4. LXC device passthrough](#)
 - [5. Advanced backup settings](#)
 - [6. Extended ACME-enabled CAs](#)
 - [7. User interface enhancements](#)
 - [8. Core technology updates](#)
- [What to do next? Download Proxmox 8.2](#)
- [Wrapping up Proxmox VE 8.2 new features](#)

Proxmox 8.2 New features

Proxmox 8.2 [includes multiple enhancements and new features](#), including the following:

1. Automated and unattended installations
2. VMware ESX VM Import Wizard
3. nftables Firewall
4. LXC device passthrough
5. Advanced backup settings
6. Extended ACME-enabled CAs
7. User interface enhancements
8. Core technology updates

1. Automated and Unattended Installations

One of the new features in Proxmox 8.2 is the new automated and unattended installation capabilities. Admins can now modify the Proxmox ISO and setup an installation answer file to automate the installation process. Admins can also place the answer file on a USB drive, or access it over the network.

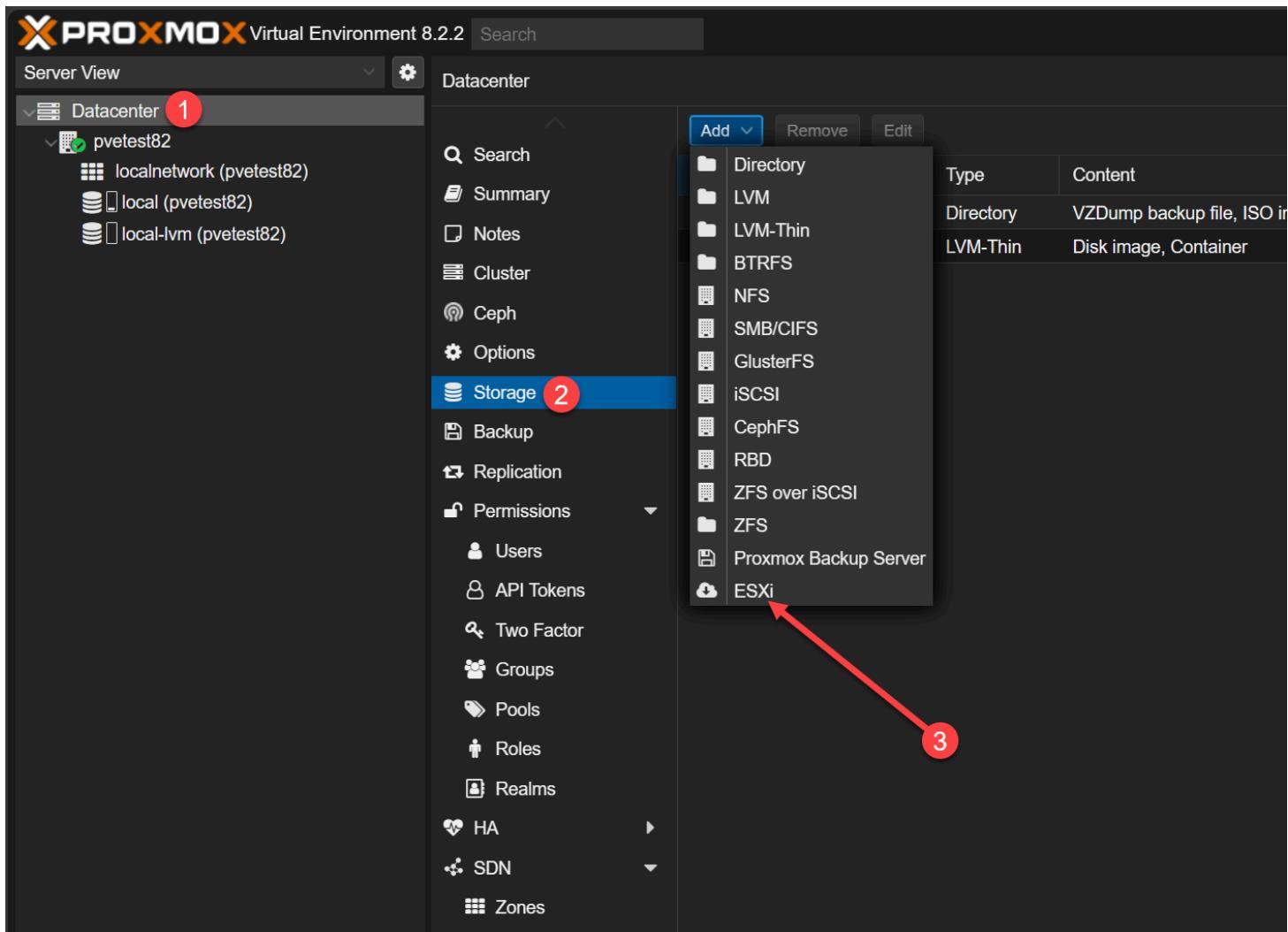
If you are deploying Proxmox [virtual environment](#) at scale and want to script the installation, the automated and unattended installation process will enhance Proxmox VE's configuration model to allow DevOps and admins alike to deploy Proxmox hosts consistently and in a hands off approach.

Why is the new feature helpful?

A: Many have complained that Proxmox is behind in the area of automation tools like are found in VMware vSphere. Proxmox is making a serious effort to improve some of those limitation of the platform. The unattended installation feature will help introduce a missing feature thus far and provide IT admins a new tool for rolling out Proxmox in mass, using automation.

2. VMware ESX VM Import Wizard

This new feature was actually made available before the official Proxmox 8.2 release. I blogged about the new VMware ESXi import wizard to migrate VMware ESXi guests in the blog post here: [Proxmox New Import Wizard for Migrating VMware ESXi VMs.](#)



The new vmware esxi import wizard

With the major upheaval in the VMware vSphere world, many organizations and home labbers alike have been looking at VMware alternatives to run virtual machines and containers, such as Proxmox server solutions.

Previously, [moving VMware ESXi](#) VMs to Proxmox was a bit of a manual process. However, now from the web based user interface in Proxmox, the new import wizard can be launched and allows a direct connection to the VMware ESXi host via a new storage plugin. It brings over most of the original [configuration settings mapped for the VM in Proxmox](#).

Why is this new feature helpful?

This integrated VM importer is a great step forward for those looking to [migrate from the increasingly unpalatable VMware](#) by Broadcom costs to a more cost-effective and open-source virtual environment. Previously, the import process was very manual with Proxmox from VMware ESXi. With the new feature, it will provide a more automated way to bring over VMware ESXi workloads.

With API access, this process could even be automated for those who want to use a script or other automation tools to import.

3. nftables Firewall

Proxmox has now introduced a new firewall implementation based on **nftables** written in Rust. It will replace the older IPtables [system in upcoming versions of the server virtualization management platform](#).

The new firewall feature is not on by default. Instead, it is available as an experimental feature that you must turn on to use. Proxmox is inviting community feedback on the new nftables functionality.

The screenshot shows the Proxmox VE 8.2.2 interface. On the left, the navigation tree is visible with 'Datacenter' expanded, showing 'pvetest82'. The main area is titled 'Node 'pvetest82''. On the right, there is a table of system settings. The 'Edit' button is at the top right of the table. Below it, the 'nftables (tech preview)' setting is listed as 'No'. A red circle labeled '2' is around this entry. A red arrow points from a red circle labeled '3' on the 'Edit' dialog to the 'No' value in the table.

Firewall	Yes
SMURFS filter	Yes
TCP flags filter	No
NDP	Yes
nf_conntrack_max	Default
nf_conntrack_tcp_timeout_established	Default
log_level_in	nolog
log_level_out	nolog
tcp_flags_log_level	nolog
smurf_log_level	nolog
nftables (tech preview)	No 2

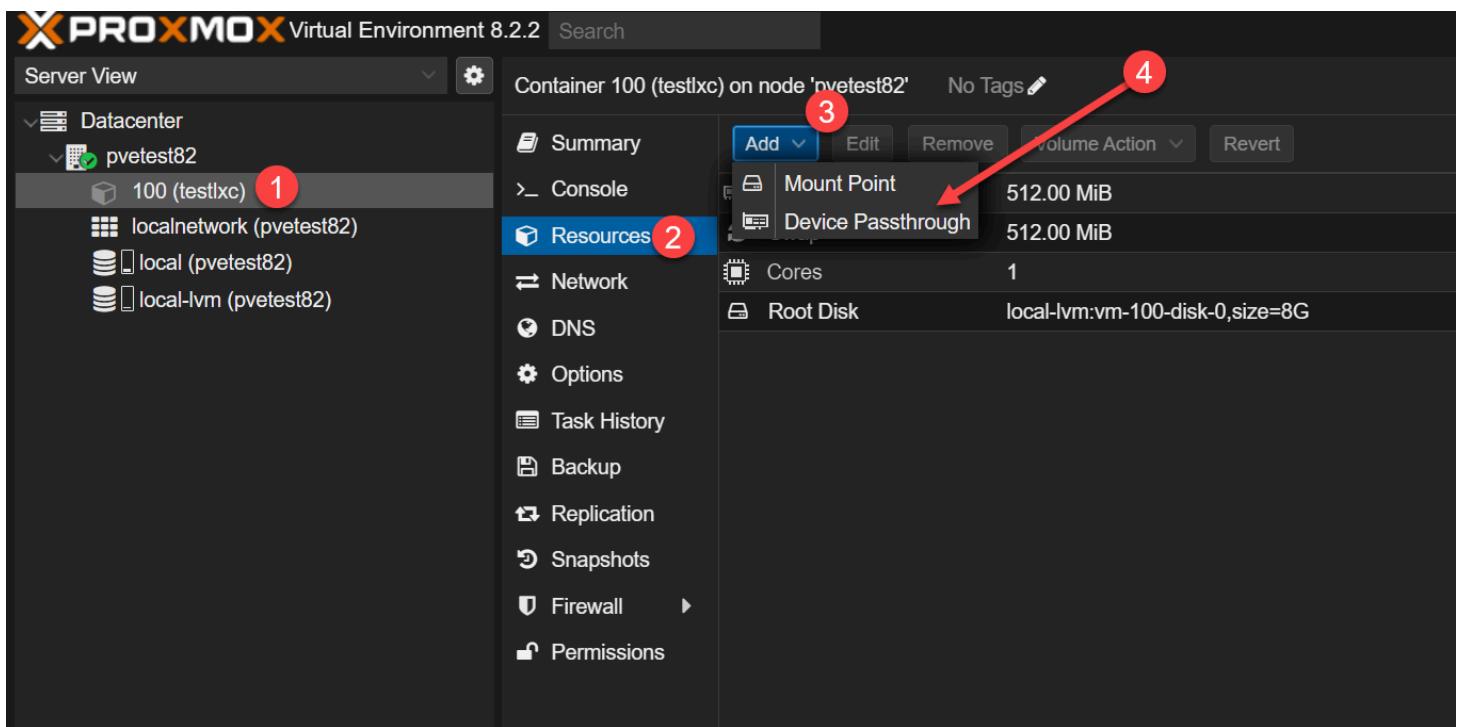
Nftables firewall in proxmox 8.2

Why is this new feature helpful?

New and improved firewall features and interface operation will ease the burden of managing security rules on Proxmox hosts.

4. LXC device passthrough

Proxmox VE 8.2 enhances [container management capabilities by integrating LXC](#) device passthrough directly within the user interface. This feature allows for precise configuration of host devices to containers, expanding the versatility and potential use cases of Proxmox in handling complex applications and workloads.



Device passthrough for lxc in proxmox 8.2 new features

Why is this new feature helpful?

Home labbers and production use cases alike may require passing through hardware devices to virtual environment. Having the ability to pass through a hardware device to LXC broadens what you can do with LXC containers and allows more powerful hardware processing for applications.

5. Advanced backup settings

With the new version of Proxmox virtual environment 8.2, admins have even more settings and control over [Proxmox backups](#). The new settings provide performance tweaks with backups, including the ability to configure bandwidth limitations in Proxmox Backup Server and decouple slower backup storage without affecting VM performance.

Why is this new feature helpful?

Having new backup features, including the ability to control backup performance and bandwidth limitations will give admins more control over their networks and help minimize impacts of backup operations on the overall network throughput and bandwidth available for production traffic.

6. Extended ACME-enabled CAs

Proxmox has already implemented ACME certificates in Proxmox versions prior to 8.2. However, in Proxmox 8.2, it now supports custom ACME-enabled certificate authorities.

Enhancing [SSL certificate](#) management, Proxmox VE 8.2 now supports custom ACME-enabled certificate authorities. It also allows configuring optional External Account Binding (EAB). It allows organizations to give [security priority and grants more control](#) over certificate provisioning and management.

Register Account

Account Name: default

E-Mail: !

ACME Directory: Custom ▼

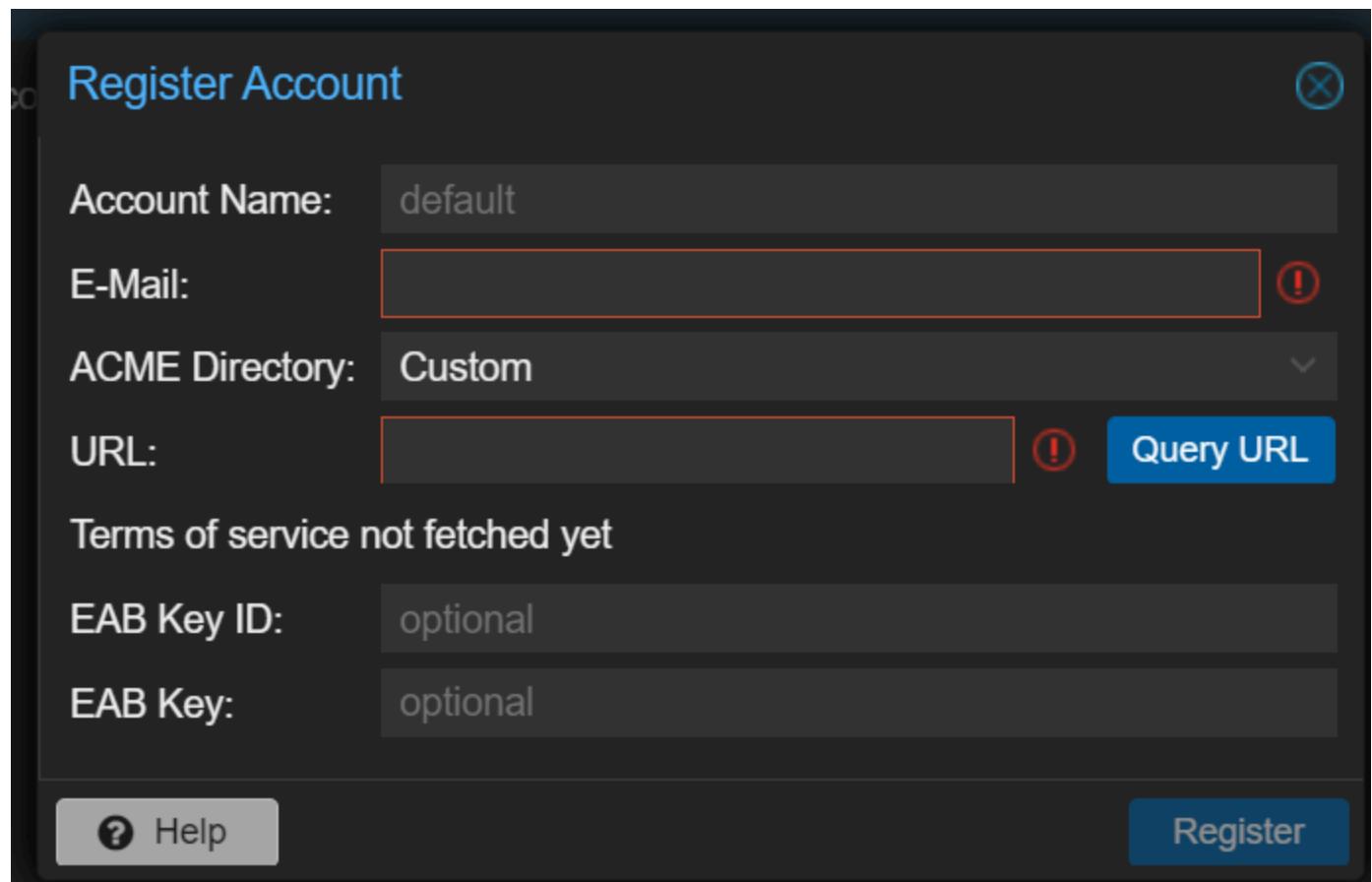
URL: ! Query URL

Terms of service not fetched yet

EAB Key ID: optional

EAB Key: optional

? Help Register



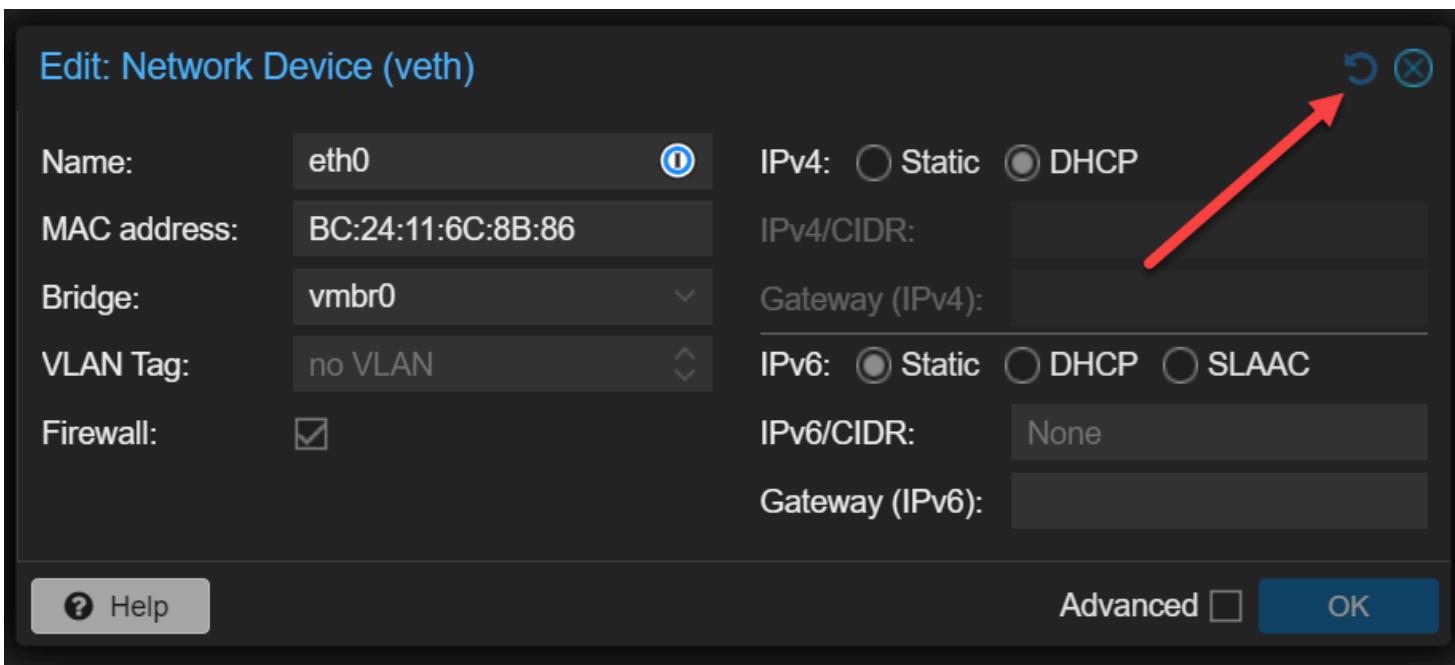
Custom acme enabled certificate option in proxmox 8.2

Why is this new feature helpful?

It gives further options for provisioning SSL certificates and using custom ACME-enabled CAs. More options will allow admins more flexibility in configuring SSL.

7. User interface enhancements

The new Proxmox 8.2 release helps refine the user interface of Proxmox to help reduce user errors. Proxmox has repositioned the **reset button** to prevent accidental clicks and modifications to input fields, such as in the notes field. These changes help to improve the overall user experience for admins and reduces frustration.



Repositioned reset button in proxmox 8.2

Why is this new feature helpful?

Quality of life improvements like the moved reset button will help to ease the workflow of provisioning, configuration, and management. It will also help prevent inadvertent mishaps in the interface.

8. Core technology updates

Proxmox VE 8.2 is based on Debian and version 8.2 implements a newer Linux kernel with Debian 12.5 (Bookworm), QEMU 8.1 LXC 6.0 Ceph 18.2, Linux Kernel 6.8, and Open ZFS 2.2. These updates to core technologies keep Proxox on the leading edge for performance and stability with the latest security fixes, etc.

Why is this new feature helpful?

Updating the core components helps keep Proxmox relevant, performant, secure, and capable of running modern applications.

What to do next? Download Proxmox 8.2

You can download the new Proxmox 8.2 ISO directly from the official website here: [Download Proxmox](#). The version and hash information are below:

- Version 8.2-1
- File size 1.39 GB
- Last Updated (at the time of this post) April 24, 2024
- SHA256SUM: d99d182a0df4ba94c27668d3e33d14cc286d775a7bdf571a86c24ea522009e93

Latest Releases

Proxmox VE 8.2 ISO Installer			
	Version 8.2-1	File Size 1.39 GB	Last Updated April 24, 2024
SHA256SUM d99d182a0df4ba94c27668d3e33d14cc286d775a7bdf571a86c24ea522009e93	Download Torrent		

Proxmox 8.2 download iso

Installing Proxmox 8.2 from ISO:



Installing proxmox 8.2

Wrapping up Proxmox VE 8.2 new features

Proxmox 8.2 is a great step forward for Proxmox VE. Many are becoming more interested in the platform in general, not just for home labs, but also for use in the SMB space or even the enterprise. It [contains a lot of great features and keeps](#) getting better with each new version released. Proxmox 8.2 makes the new VMware ESXi import wizard official out of the box. All the other changes are welcomed as well to continue to refine the product for production use and to make sure it is performant and stable for production workloads. Let me know in the comments if you plan on [updating your Proxmox servers](#) soon.

Proxmox VE 8.3 New Features with OVA Import

Proxmox has gained tremendous popularity in 2024. So many home lab enthusiasts, SMBs, and enterprise environments are looking at their options with the fallout from VMware by Broadcom. Proxmox has been introducing many new features with each release and are aggressively targeting ones looking at virtualization alternatives or looking to migrate from their existing solution. Proxmox VE 8.3 new [features will help to take Proxmox VE Server](#) functionality even further. Let's take a look at what's new in this release.

Table of contents

- [New features in Proxmox VE 8.3](#)
- [1. Improved OVA and OVF import](#)
- [2. New tag view](#)
- [3. New firewall features](#)
- [4. More efficient backup snapshots](#)
- [5. New language support](#)
- [Other new features](#)
- [How to upgrade to Proxmox VE 8.3](#)
- [Wrapping up](#)

New features in Proxmox VE 8.3

There are many new features with the 8.3 release, including the following:

1. Improved OVA and OVF import
2. New tag view
3. New firewall features
4. More efficient backup snapshots
5. New language support

[Proxmox Virtual Environment](#) (VE) 8.3 introduces an array of innovative features and updates aimed at enhancing usability, flexibility, and efficiency. Built on the robust foundation of Debian 12.8 (Bookworm), this release incorporates a modern Linux kernel, advanced virtualization tools, and a user-friendly interface. Here's a detailed look at what's new in Proxmox VE 8.3.

1. Improved OVA and OVF import

One of the features that many like about VMware vSphere is the ability to easily import OVA and OVF files. You can do this easily from the vSphere Client. Now, with Proxmox 8.3, admins can now import **Open Virtualization Format (OVF)** and **Open Virtualization Appliance (OVA)** files through the web interface.

Note the following new support with Proxmox VE Server 8.3:

- **File-Based Storage Support:** You can import directly from file-based storage locations like NFS
- **Upload Options:** You can upload OVA files from local machine storage or download them from a URL much like we have been able to do in VMware vSphere

PROXMOX Virtual Environment 8.3.0

Server View Datacenter

ID	Type	Content	Path/Target	Shared	Enabled	Bandwidth Limit
ESXi-Demo	ESXi	Import		No	No	
backupstore1	Proxmox Backup Server	VZDump backup file		Yes	Yes	
ceph	RBD (PVE)	Disk image, Container		Yes	Yes	
cephfs1	CephFS (PVE)	VZDump backup file, ISO image, Container template	/mnt/pve/cephfs1	Yes	Yes	
iso-templates	NFS	ISO image, Container template	/mnt/pve/iso-templates	Yes	Yes	
local	Directory	ISO image, Container template	/var/lib/vz	No	Yes	
local-lvm	LVM-Thin	Disk image, Container		No	Yes	

Search: Summary Notes Cluster Ceph Options Storage Backup Replication Permissions Users API Tokens Two Factor Groups Pools Roles Realms HA SDN Zones VNet Options IPAM VNet Firewall ACME Firewall

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Nov 20 10:27:46		pve-demo1	root@pam	Shell	OK
Nov 20 13:29:01	Nov 20 13:29:11	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:19	Nov 20 13:01:42	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:15	Nov 20 13:01:45	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:12	Nov 20 13:01:55	pve-demo1	root@pam	Shell	OK

Importing from nfs storage

PROXMOX Virtual Environment 8.3.0

Server View Datacenter (democuster)

Storage 'iso-templates' on node 'pve-demo1'

Upload | Download from URL Import Remove

Name

Import Options

- ISO image, Container template
- Disk image
- ISO image
- Container template
- VZDump backup file
- Container
- Snippets
- Import

Search: Name, Format Format Size

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Nov 20 10:27:46		pve-demo1	root@pam	Shell	OK
Nov 20 13:29:01	Nov 20 13:29:11	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:19	Nov 20 13:01:42	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:15	Nov 20 13:01:45	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:12	Nov 20 13:01:55	pve-demo1	root@pam	Shell	OK

Upload and download options for getting ova appliance files in proxmox ve server 8.3

The screenshot shows the Proxmox VE interface. On the left, the Datacenter (democluster) tree view is visible, showing various hosts like pve-demo1, pve-demo2, etc. In the center, the 'Storage iso-templates' screen is displayed. A modal dialog titled 'Download from URL' is open, prompting for a URL. The URL input field contains the value 'ant/operating-system/releases/download/13.2/haos_ova-13.2.ova'. Below the URL field, there's a note: 'Please (re-)query URL to get meta information'. The dialog also includes fields for 'File name', 'File size', 'Hash algorithm' (set to 'None'), 'Verify certificates' (checkbox checked), and 'Checksum' (set to 'none'). At the bottom right of the dialog are 'Advanced' and 'Download' buttons.

Download from url example in proxmox 8.3

The screenshot shows the Proxmox VE interface. The left sidebar shows the Datacenter (democluster) tree. The central area shows the 'Storage iso-templates' screen. The 'ISO Images' tab is selected. A red arrow points to a file entry named 'haos_ova-13.2.ova' in the list. To the right of the list, file details are shown: Name 'haos_ova-13.2.ova', Format 'ova', and Size '434.96 MB'. At the bottom of the screen, the 'Tasks' and 'Cluster log' panes are visible, showing system activity logs.

Ova file in inventory

2. New tag view

In this release, there is a new **Tag View** that has been added to the resource tree. Using this view, it provides users with a quick way to organize and monitor VMs and containers according to their tags.

- **Custom tags:** Admins can add custom tags to their VMs and containers for better categorization of virtual resources
- **Grouped display:** The new view arranges your virtual guests by tags and makes it easier to navigate and manage large environments with many virtual resources

This feature simplifies administrative tasks and provides a clear visual representation of your infrastructure. Below shows where you choose the new Tag View.

The screenshot shows the Proxmox VE 8.3.0 interface. The left sidebar has a 'Tag View' section highlighted, containing items like pve-demo4, pve-demo5, Production, and Testing. The main dashboard includes sections for Health (Status: 5 Online, 0 Offline, Ceph HEALTH_OK), Resources (CPU: 6%, Memory: 19%, Storage: 44%), Guests (Virtual Machines: 0 Running, 1 Stopped; LXC Container: 1 Running, 5 Stopped), Nodes (list of five nodes with details like ID, Online status, Support level, Server Address, CPU usage, Memory usage, and Uptime), and Subscriptions (Community status valid). At the bottom, a 'Tasks' table lists recent log entries from a node named pve-demo1.

Start Time	End Time	Node	User name	Description	Status
Nov 20 10:27:46		pve-demo1	root@pam	Shell	OK
Nov 20 13:29:01	Nov 20 13:29:11	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:19	Nov 20 13:01:42	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:15	Nov 20 13:01:45	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:12	Nov 20 13:01:55	pve-demo1	root@pam	Shell	OK

New tag view in proxmox 8.3

3. New firewall features

New firewall features in the Proxmox VE 8.3 update include support for forwarded network traffic at the host and VNet level with Proxmox SDN.

Note the following:

- **IP Sets for VNets:** The Software-Defined Networking (SDN) in Proxmox now automatically generates IP sets for each VNet. This includes subnet and DHCP ranges.
- **Improved management** of security features: Users can fine-tune firewall rules to secure their environment comprehensively.

Admins will need to enable the firewall to use these features. Below you see the IPsets:

The screenshot shows the Proxmox VE 8.3 interface with the 'VNet Firewall' section selected. A modal window titled 'Add: Rule' is open, showing a table of existing rules. One rule is selected: 'ipset +Demo-gateway SDN'. The table columns are Type, Name ↑, Scope, and Comment.

Type	Name ↑	Scope	Comment
ipset	+Demo-all	SDN	All subnets of VNet D...
ipset	+Demo-dhcp	SDN	DHCP ranges of VNe...
ipset	+Demo-gateway	SDN	All gateways of VNet ...
ipset	+Demo-no-gateway	SDN	All subnets of VNet D...
ipset	+guest-ipam-103	SDN	
ipset	+guest-ipam-107	SDN	

Adding firewall rules

Adding a new rule.

The screenshot shows the Proxmox VE 8.3 interface with the 'VNet Firewall' section selected. A modal window titled 'Add: Rule' is open, showing a table of existing rules. A new rule is being added with the source '+Demo-no-gateway' and destination '+Demo-gateway'. The 'Add' button is highlighted.

Type	Name ↑	Scope	Comment
ipset	+Demo-all	SDN	All subnets of VNet D...
ipset	+Demo-dhcp	SDN	DHCP ranges of VNe...
ipset	+Demo-gateway	SDN	All gateways of VNet ...
ipset	+Demo-no-gateway	SDN	All subnets of VNet D...
ipset	+guest-ipam-103	SDN	
ipset	+guest-ipam-107	SDN	

Firewall rules 2 in proxmox 8.3

4. More efficient backup snapshots

Backups have gotten better in Proxmox 8.3. Now the **metadata** and **data** of the Proxmox backups are separated into two different archives.

Note the benefits of doing this:

- **Incremental backup optimization:** This introduces changed block tracking in Proxmox by referencing metadata from the previous backup snapshot and identifying unchanged files. Once those files are identified, these are skipped which means backup runtime is significantly less and much more efficient.
- **Less time:** This optimization is extremely welcome in large environments and with frequently backed-up VMs

Below, we are editing a backup job.

The screenshot shows the Proxmox Virtual Environment 8.3.0 interface. In the center, a modal window titled "Edit: Backup Job" is open, showing the "General" tab selected. The configuration includes:

- Node: All
- Storage: backupsstore1
- Schedule: */2:00
- Selection mode: Include selected VMs
- Notification mode: Default (Auto)
- Send email to: martin@proxmox.com
- Send email: Always
- Compression: ZSTD (fast and good)
- Mode: Snapshot
- Enable: checked

Below the modal, a table titled "Job Comment" lists VMs and LXC containers:

ID	Node	Status	Name	Type
100	pve-demo4	stopped	Win10-en	Virtual Machine
102	pve-demo2	stopped	debianbuster	LXC Container
103	pve-demo3	stopped	centosstream	LXC Container
104	pve-demo1	running	centos	LXC Container
105	pve-demo1	stopped	Rocky9	LXC Container
106	pve-demo1	stopped	Arch	LXC Container
107	pve-demo1	stopped	AlmaLinux	LXC Container

At the bottom of the modal is an "OK" button. At the bottom of the main interface, there is a table titled "Cluster.log" showing system logs:

Start Time	End Time	Node	User name	Description	Status
Nov 20 10:27:46		pve-demo1	root@pam	Shell	OK
Nov 20 13:36:55	Nov 20 13:37:07	pve-demo1	root@pam	File haos_ova-13.2.ova - Download	OK
Nov 20 13:29:01	Nov 20 13:29:11	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:19	Nov 20 13:01:42	pve-demo1	root@pam	Shell	OK
Nov 20 13:01:15	Nov 20 13:01:45	pve-demo1	root@pam	Shell	OK

Editing a backup job in proxmox backup server

Here you can see the PBS change detection mode set to **metadata**.

Setting the pbs change detection mode to use metadata

5. New language support

Proxmox 8.3 extends language support to **30 languages**. This will definitely help organizations who are adopting Proxmox to be able to effectively use the web interface and administer the product.

Other new features

There are other new features to note in this release as well:

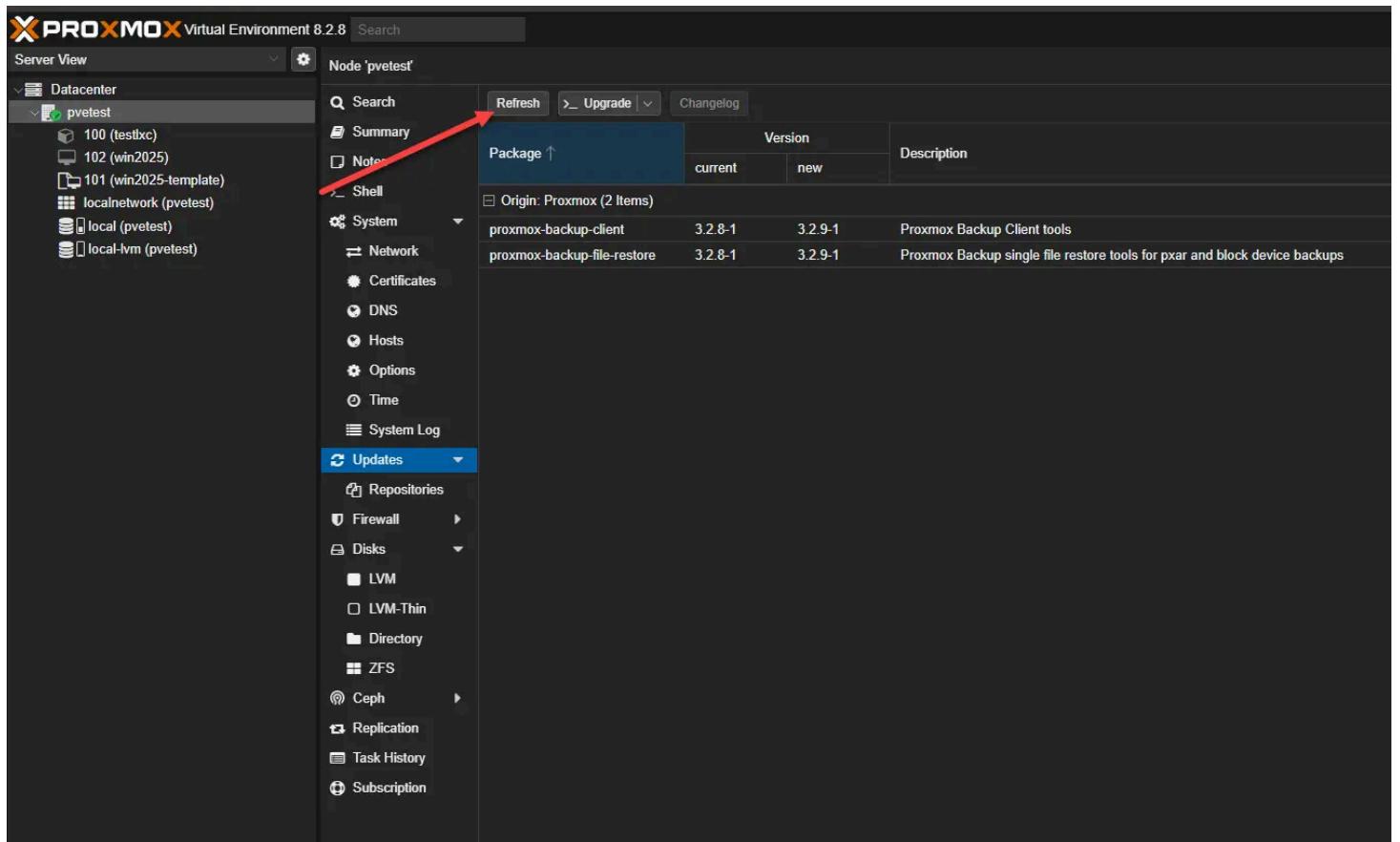
- **Linux Kernel:** It includes the Linux Kernel Version 6.8 as the default stable kernel and 6.11 available as an opt-in.
- **Ceph:** You can now choose between Ceph Reef 18.2 and Ceph Squid 19.2 for HCI storage
- **QEMU 9.0:** New virtualization features with QEMU 9.0
- **OpenZFS 2.2.6:** The latest OpenZFS filesystem and volume management

How to upgrade to Proxmox VE 8.3

If you are already running Proxmox VE 8.x, how can you upgrade your Proxmox server to version 8.3? Simple. Just make sure you have your repos pointed to the non-subscription sources if you don't have a subscription. If you want to read how to do that, check out my detailed blog post here:

- [Proxmox Update No Subscription Repository Configuration](#)

Once you are pointed to the non-subscription repos, you can refresh your updates, and then perform the upgrade on your Proxmox host. Click your Proxmox VE host, then click **Updates > Refresh** to refresh and pull the latest updates available.



Refreshing updates in proxmox web interface

Next, once the updates are pulled, click the **Upgrade** button to actually apply the updates.

Package ↑	Version	Description
	current new	
ifupdown2	3.2.0-1+pmx9	Network Interface Management tool similar to ifupdown
libjs-extjs	7.0.0-4	cross-browser JavaScript library
libpve-access-control	8.1.4	Proxmox VE access control library
libpve-cluster-api-perl	8.0.8	Proxmox Virtual Environment cluster Perl API modules.
libpve-cluster-perl	8.0.8	Proxmox Virtual Environment cluster Perl modules.
libpve-common-perl	8.2.8	Proxmox VE base library
libpve-guest-common-perl	5.1.4	Proxmox VE common guest-related modules
libpve-network-api-perl	0.10.0	API endpoints for Proxmox VE's SDN stack
libpve-network-perl	0.9.8	Proxmox VE's SDN (Software Defined Network) stack
libpve-notify-perl	8.0.8	Notify helper module.
libpve-rs-perl	0.8.11	PVE parts which have been ported to Rust - Rust source code
libpve-storage-perl	8.2.6	Proxmox VE storage management library
proxmox-backup-client	3.2.8-1	Proxmox Backup Client tools
proxmox-backup-file-restore	3.2.8-1	Proxmox Backup single file restore tools for pxar and block device backups
proxmox-firewall	0.5.0	Proxmox's nftables-based firewall written in rust
proxmox-mail-forward	0.2.3	Proxmox mail forward helper
proxmox-ve	8.2.0	Proxmox Virtual Environment
proxmox-widget-toolkit	4.3.0	Core Widgets and ExtJS Helper Classes for Proxmox Web UIs
pve-cluster	8.0.8	"pmxcls" distributed cluster filesystem for Proxmox Virtual Environment.
pve-container	5.2.1	Proxmox VE Container management tool
pve-docs	8.2.4	Proxmox VE Documentation
pve-firewall	5.0.7	Proxmox VE Firewall
pve-ha-manager	4.0.5	Proxmox VE HA Manager
pve-i18n	3.2.4	Internationalization support for Proxmox VE
pve-manager	8.2.8	Proxmox Virtual Environment Management Tools
qemu-server	8.2.6	Qemu Server Tools

Upgrade your proxmox ve server you can see the 8.3 modules

You will be prompted to confirm the upgrade in the bash shell dialog that comes up.

```

Starting system upgrade: apt-get dist-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
  libpve-network-api-perl
The following packages will be upgraded:
  ifupdown2 libjs-extjs libpve-access-control libpve-cluster-api-perl libpve-cluster-perl libpve-common-perl libpve-guest-common-perl libpve-networ
  proxmox-mail-forward proxmox-ve proxmox-widget-toolkit pve-cluster pve-container pve-docs pve-firewall pve-ha-manager pve-i18n pve-manager qemu-s
25 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 33.4 MB of archives.
After this operation, 3700 kB of additional disk space will be used.
Do you want to continue? [Y/n] 

```

Confirm the upgrade to proxmox ve server 8.3

After the updates are applied and the host is rebooted, upon logging back into Proxmox, we see the version is Proxmox 8.3.

The screenshot shows the Proxmox VE 8.3.0 interface. At the top left is the Proxmox logo and "Virtual Environment 8.3.0". A red arrow points from the text above to the search bar at the top right. The main window is titled "Datacenter". On the left is a sidebar with various management options: Server View, Datacenter (selected), Notes, Cluster, Ceph, Options, Storage, Backup, Replication, Permissions (with sub-options: Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms), HA, SDN (with sub-options: Zones, VNets), and a gear icon for settings. The main pane displays a table of resources. The columns are: Type, Description, Disk usage..., Memory us..., CPU usage, and Uptime. The data includes:

Type	Description	Disk usage...	Memory us...	CPU usage	Uptime
Ixc	100 (testlxc)	-	-	-	-
node	pvetest	38.7 %	6.1 %	0.1% of 22 ...	09:10:09
qemu	102 (win2025)	-	-	-	-
qemu	101 (win2025-template)	-	-	-	-
sdn	localnetwork (pvetest)	-	-	-	-
storage	local (pvetest)	38.7 %	-	-	-
storage	local-lvm (pvetest)	4.1 %	-	-	-

After upgrading and rebooting to get the new kernel we see proxmox ve server 8.3

Wrapping up

I dare say that Proxmox is arguably the most popular open source virtualization solution on the market today. With all the fallout from the [Broadcom acquisition of VMware](#), many organizations are exploring their options on this front. Proxmox 8.3 adds even more features that will attract VMware vSphere administrators, like the ability to easily ingest OVA and OVF files into your Proxmox environment. Version 8.3 gives us all the same ways we can work with OVA files in VMware vSphere, so this will be a great addition.

Let me know your thoughts on the upgrade and new features found in this latest release of Proxmox and if you have already taken the plunge to upgrade in your home lab?

Proxmox Networking for VMware vSphere admins

One of the challenges that we run into when we are more familiar with one vendor over another is the difference in the technologies, how they work for customers, what they are called, and how to configure them. On the networking side of things, this can be the case as well. If you are familiar with VMware vSphere and looking to work with and play around with Proxmox, Proxmox networking may be foreign to work with when trying to compare it with VMware ESXi networking, etc. In this Proxmox networking for vSphere admins post for the community, we will look at the equivalent networking configurations in Proxmox compared to vSphere.

Table of contents

- [1. Proxmox Linux Bridge equivalent to the VMware vSwitch](#)
- [2. Proxmox Linux VLANs equivalent to VMware vSwitch Port Groups](#)
- [3. Distributed Switches](#)
- [4. Network Adapters](#)
- [5. Network I/O Control](#)
- [6. Software-defined networking \(SDN\)](#)
- [7. Troubleshooting](#)
- [Wrapping up](#)

1. Proxmox Linux Bridge equivalent to the VMware vSwitch

The most equivalent network construct out of the box with Proxmox is the default Proxmox Linux bridge in the Proxmox environment. With the Linux Bridge in Proxmox, you establish the initial connectivity to your [Proxmox host](#) with a management IP address. Also, the default Linux Bridge is backed by a physical network adapter(s).

Below:

- Ports/Slaves – This shows the physical network adapter **ens192** assigned to the default Linux bridge
- CIDR – Shows the IP address and mask associated with the Linux bridge

The screenshot shows the Proxmox VE 8.1.3 interface. The left sidebar is titled "Server View" and shows a tree structure with "Datacenter" expanded, containing "pve". The right panel is titled "Node 'pve'" and has a "Network" tab selected. The main area is a table with columns: Name ↑, Type, Active, Autostart, VLAN a..., Ports/Slaves, Bond Mode, and CIDR. There are two rows: "ens192" (Network Device, Active Yes, Autostart No, VLAN a... No, Ports/Slaves ens192, Bond Mode, CIDR 10.1.149.155/24) and "vmbr0" (Linux Bridge, Active Yes, Autostart Yes, VLAN a... No, Ports/Slaves ens192, Bond Mode, CIDR 10.1.149.155/24). A red arrow points to the "vmbr0" row.

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR
ens192	Network Device	Yes	No	No			
vmbr0	Linux Bridge	Yes	Yes	No	ens192		10.1.149.155/24

Viewing the default proxmox linux bridge

This is very similar to the default **vSwitch0** created in VMware ESXi right out of an install. As you can see below, we have a physical network adapter backing vSwitch0.

ESXi Host Client

root@10.3.33.200 | Help | Search

vSwitch0

Type: Standard vSwitch
Port groups: 1
Uplinks: 1

vSwitch Details

- MTU: 1500
- Ports: 2060 (2055 available)
- Link discovery: Listen / Cisco discovery protocol (CDP)
- Attached VMs: 0 (0 active)
- Beacon interval: 1

NIC teaming policy

- Notify switches: Yes
- Policy: Route based on originating port ID
- Reverse policy: Yes
- Fallback: Yes

Security policy

- Allow promiscuous mode: No
- Allow forged transmits: No
- Allow MAC changes: No

vSwitch topology

VM Network (VLAN ID: 0) connects to Physical adapters (vmnic0, 1000 Mbps, ...).

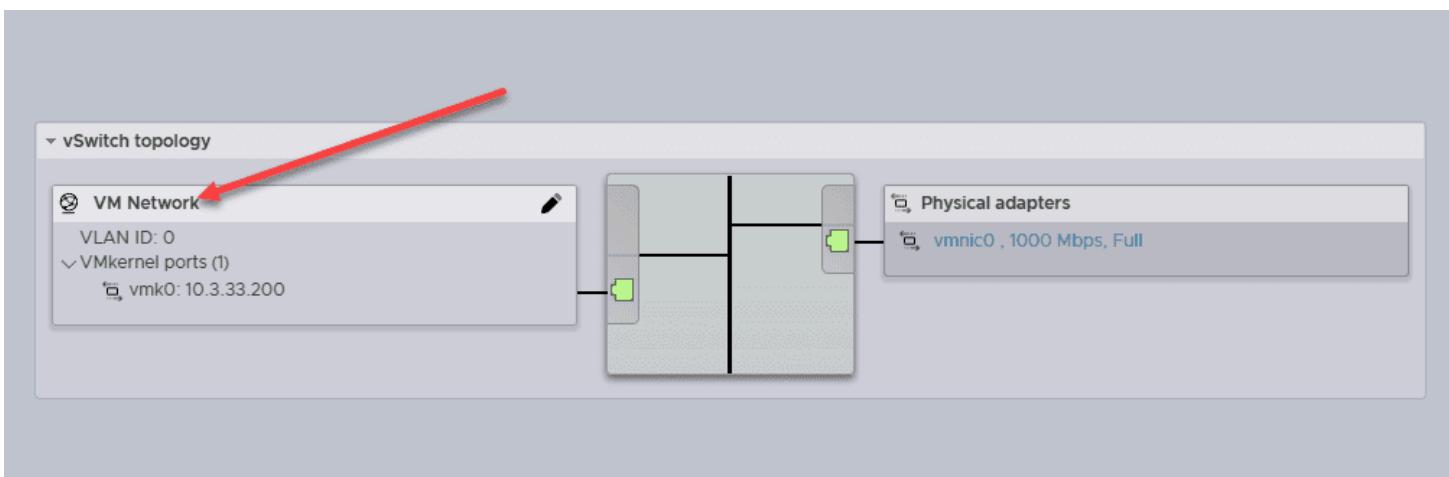
Recent tasks

Task	Target	Initiator	Queued	Started	Result	Completed
Refresh Network System	localhost.cloud.local	root	12/27/2023 14:42:56	12/27/2023 14:42:56	Completed successfully	12/27/2023 14:42:56
Update Network Config	localhost.cloud.local	root	12/27/2023 14:42:56	12/27/2023 14:42:56	Completed successfully	12/27/2023 14:42:56
Refresh Network System	localhost.cloud.local	dcul	12/27/2023 14:42:46	12/27/2023 14:42:46	Completed successfully	12/27/2023 14:42:46
Refresh Network System	localhost.cloud.local	root	12/27/2023 14:42:46	12/27/2023 14:42:46	Completed successfully	12/27/2023 14:42:46
Update Network Config	localhost.cloud.local	root	12/27/2023 14:42:41	12/27/2023 14:42:41	Completed successfully	12/27/2023 14:42:43
Refresh Network System	localhost.cloud.local	root	12/27/2023 14:42:18	12/27/2023 14:42:18	Completed successfully	12/27/2023 14:42:18

Vmware esxi default vswitch0

2. Proxmox Linux VLANs equivalent to VMware vSwitch Port Groups

As most vSphere admins are aware, you have the vSwitch0 and the default **VM Network port group** as shown above. Port groups in [vSphere allow you to create VLAN-tagged network labels in vSphere to assign VLAN tags to the virtual machines](#) connected to them.



Default vsphere port group on vswitch0

In Proxmox networking with the default Linux Bridge, the port group construct carries over to what is known as a **Linux VLAN**. By default, when you create a VM in Proxmox without any other configuration, you can attach the VM to the default **vmbr0** bridge, which is essentially **VLAN 0** and assumes untagged traffic.

Unlike the [VMware default vSwitch0 and VM Network](#) port group, the default Proxmox Linux Bridge is not VLAN-aware out of the box. You have to enable this.

When you edit the default Linux Bridge, you will see the checkbox **VLAN aware** available on the Linux Bridge properties. Also, you will see [basic networking](#) configurations like the IP address and subnet, gateway for routing, etc. Place a check box in the VLAN aware checkbox.

The screenshot shows the Proxmox VE web interface. On the left, there's a sidebar with various system management options like Summary, Notes, Shell, System, Network, Certificates, DNS, Hosts, Options, Time, Syslog, Updates, Repositories, Firewall, Disks, LVM, LVM-Thin, Directory, ZFS, Ceph, Replication, Task History, and Subscription. The 'Network' option is selected. In the main area, a table lists network interfaces: ens192 (Network Device, Active: Yes, Autostart: No, VLAN a...: No) and vmbr0 (Linux Bridge, Active: Yes, Autostart: Yes, VLAN a...: No). The IP configuration for vmbr0 is shown as 10.1.149.155/24. A modal dialog titled 'Edit: Linux Bridge' is open for vmbr0. It contains fields for Name (vmbr0), Autostart (checked), IPv4/CIDR (10.1.149.155/24), Gateway (IPv4) (10.1.149.1), Bridge ports (ens192), and Comment. The 'VLAN aware' checkbox is highlighted with a red arrow. At the bottom of the dialog are 'Advanced' (checkbox), 'OK', and 'Reset' buttons.

Making the proxmox linux bridge vlan aware

Now we can apply the configuration. Click the **Apply Configuration** button. Also, in the preview of the Pending changes, you will see the new VLAN bridge-ports configuration set to auto, containing the configuration lines:

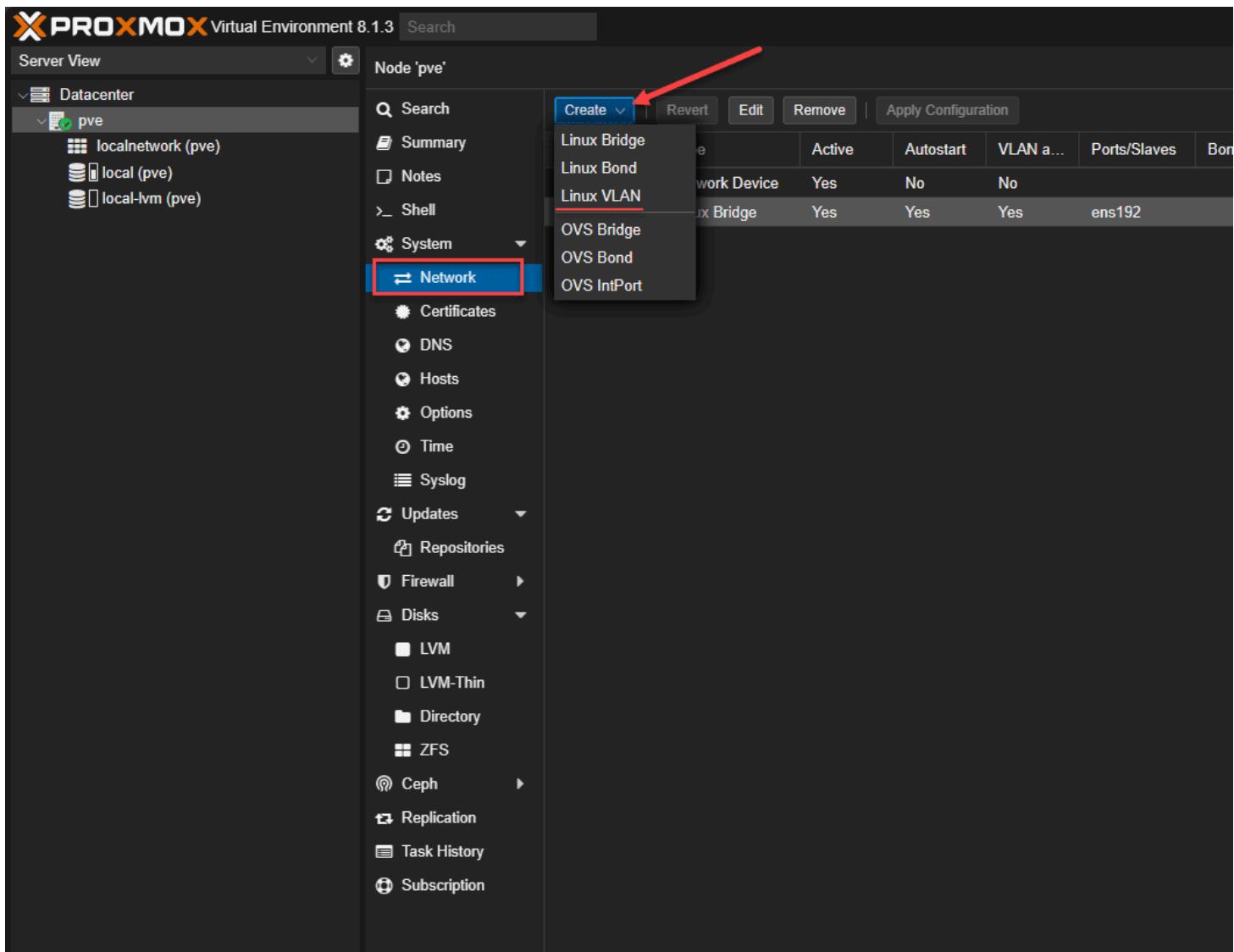
```
bridge-ports ens192
bridge-stp off
bridge-vlan-aware yes
bridge-vids 2-4094
```

The screenshot shows the Proxmox VE interface for Node 'pve'. The left sidebar contains various system management options like Certificates, DNS, Hosts, Options, Time, Syslog, Updates, Repositories, Firewall, Disks, LVM, LVM-Thin, Directory, ZFS, Ceph, Replication, Task History, and Subscription. The main panel is titled 'Node 'pve'' and is currently viewing the 'Network' section. It displays two network interfaces: 'ens192' (Network Device, Active: Yes, Autostart: No, VLAN aware: No) and 'vmbr0' (Linux Bridge, Active: Yes, Autostart: Yes, VLAN aware: Yes, Ports/Slaves: ens192). Below the table, a message says 'Pending changes (Either reboot or use 'Apply Configuration' (needs ifupdown2) to activate)'. A diff output shows the changes made to /etc/network/interfaces:

```
--- /etc/network/interfaces      2023-12-15 10:10:55.496227374 -0600
+++ /etc/network/interfaces.new 2023-12-27 15:04:35.696287346 -0600
@@ -21,4 +21,6 @@
          bridge-ports ens192
          bridge-stp off
          bridge-fd 0
+         bridge-vlan-aware yes
+         bridge-vids 2-4094
```

Apply the vlan aware configuration

This configuration essentially makes the default Linux Bridge able to understand VLANs and VLAN traffic, so we can add Linux VLANs.



Creating a linux vlan

Now we can populate the new Linux VLAN with the appropriate configuration. Once you name the VLAN with the parent **vmbr0** interface, you will see the **VLAN raw device** and **VLAN Tag** greyed out. This essentially says we are creating a new Linux VLAN on the parent Linux Bridge interface, vmbr0.

Under the **Advanced** checkbox, you can set the MTU value in case you are wondering.

Create: Linux VLAN

Name:	vmbr0.333	Autostart:	<input checked="" type="checkbox"/>
IPv4/CIDR:		Vlan raw device:	vmbr0
Gateway (IPv4):		VLAN Tag:	333
IPv6/CIDR:		Comment:	
Gateway (IPv6):		Either add the VLAN number to an existing interface name, or choose your own name and set the VLAN raw device (for the latter ifupdown1 supports vlanXY naming only)	
		Help	Advanced <input type="checkbox"/> Create

Creating the linux vlan from the vmbr0 interface

Now that we have created the child VLAN interface on the vmbr0 Linux bridge, you can see the **vmbr0.333** interface listed now under the network configuration in the navigation tree of **System > network**.

The screenshot shows the Proxmox VE 8.1.3 interface. The left sidebar shows the navigation tree with 'Datacenter' selected, followed by 'pve' which has 'localnetwork (pve)', 'local (pve)', and 'local-lvm (pve)' as children. Under 'pve', 'Network' is selected. The main pane displays a table of network interfaces. The table has columns: Name ↑, Type, Active, Autostart, VLAN a..., Ports/Slaves, Bond Mode, and CIDR. The table contains three rows:

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR
ens192	Network Device	Yes	No	No			
vmbr0	Linux Bridge	Yes	Yes	Yes	ens192		10.1.149.
vmbr0.333	Linux VLAN	No	Yes	No			

A red arrow points to the 'vmbr0.333' row. Below the table, a section titled 'Pending changes (Either reboot or use 'Apply Configuration' (needs ifupdown2) to activate)' shows the diff between /etc/network/interfaces and /etc/network/interfaces.new. The diff highlights the addition of a new interface entry:

```

--- /etc/network/interfaces      2023-12-27 15:04:35.696287346 -0600
+++ /etc/network/interfaces.new 2023-12-27 15:13:21.815396349 -0600
@@ -24,3 +24,6 @@
        bridge-vlan-aware yes
        bridge-vids 2-4094
+
+auto vmbr0.333
+iface vmbr0.333 inet manual
+

```

Viewing the newly created linux vlan

3. Distributed Switches

In case you are wondering, Proxmox doesn't have an equivalent construct like the vSphere Distributed Switch. The [vSphere distributed switch is managed](#) and controlled from vCenter Server. Proxmox doesn't have a centralized management platform like vCenter Server.

The screenshot shows the vSphere Client interface. On the left, the navigation tree is expanded to show 'vcsa.cloud.local' and its subfolders: 'CloudLocal' (containing 'AzureStackHCI', 'VM Network', 'VSS-Mgmt-333', 'VSS-Servers', and 'VDS-AzureStackHCI'), and 'VDS-Mgmt' (containing 'DGP-Mgmt-333', 'DPG-Cameras-222', 'DPG-Cluster', 'DPG-DMZ', 'DPG-Frontend', 'DPG-Internet1-1000', 'DPG-iscsi', 'DPG-LAN', 'DPG-Mgmt', 'DPG-Monitoring', 'DPG-pfsense-trunk', 'DPG-Servers', 'DPG-TestNetwork', and 'DPG-vMotion'). A red arrow points from the 'VDS-Mgmt' folder in the tree to the 'VDS-Mgmt' section in the main content area. The main content area has tabs: 'Summary' (selected), 'Monitor', 'Configure', 'Permissions', 'Ports', and 'Hosts'. The 'Summary' tab displays 'Switch Details' with the following information:

	Manufacturer	Version	Networks	Hosts	Virtual machines	Ports
	VMware, Inc.	8.0.0	20	4	84	354

Below this is a 'Custom Attributes' section containing the entry 'com.vrlcm.snapshot'.

Vmware vsphere distributed switch

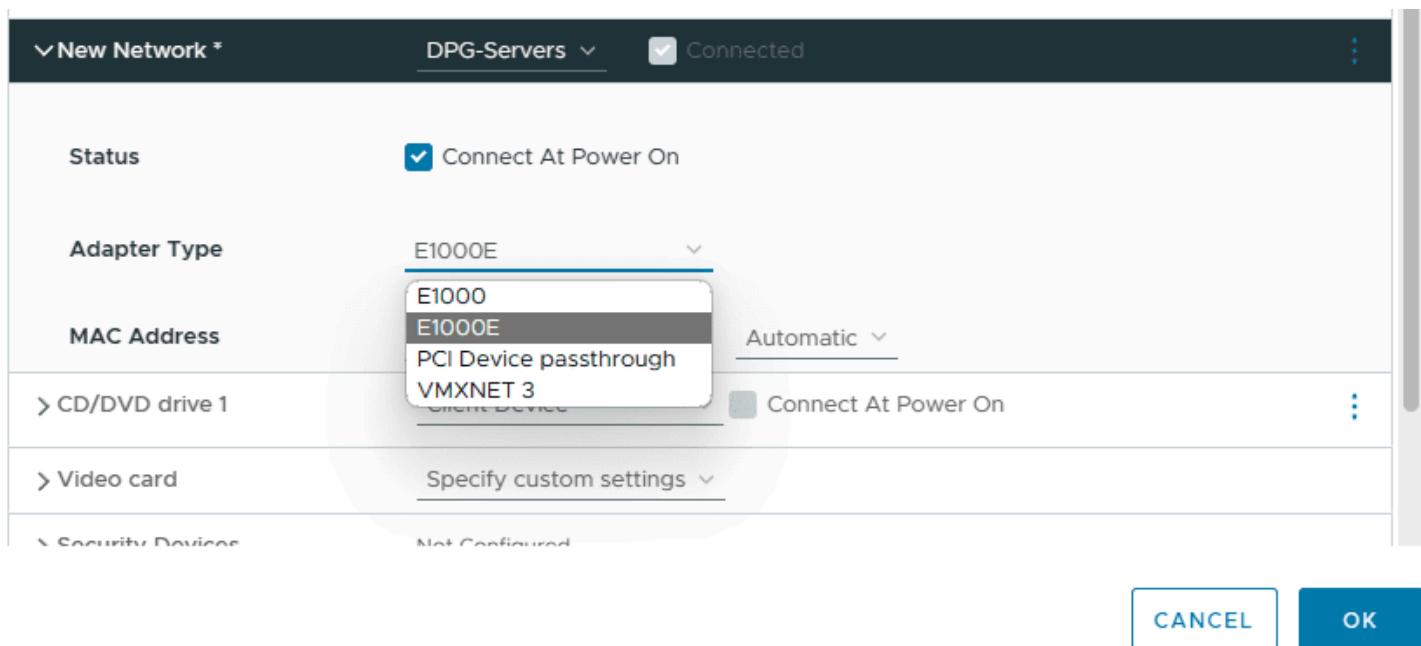
Proxmox admins would need to manage complex network setups manually with scripting or use third-party tools available for Proxmox for centralized network management.

4. Network Adapters

When it comes to network adapters for a virtual machine or container, both [Proxmox and VMware vSphere support different types of network](#) adapters. These include:

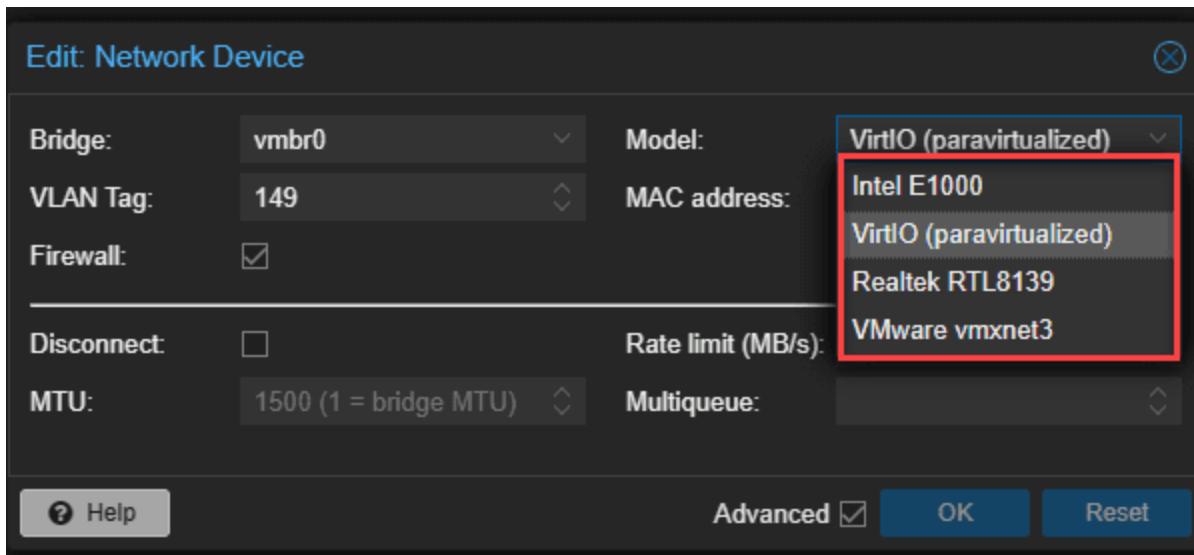
- VMXNET3
- E1000
- PCI Passthrough
- VirtIO (Proxmox)

Below you see the vnic options for a virtual machine.



Vmware network adapters

Below are the options when creating a new virtual machine in Proxmox.



Proxmox virtual network adapters

5. Network I/O Control

VMware has many tools for network traffic priority, bandwidth allocation, and other network capabilities. Proxmox does not contain network [I/O control features built-in that are found in VMware vSphere](#). You can use Linux tools in Proxmox to help control network bandwidth.

With Proxmox, you can take advantage of one or many of the following Linux networking tools:

1. Traffic Control (tc):

- tc is a tool in the Linux iproute2 package. It allows you to control the traffic going through your network interfaces. You can create rules for traffic shaping and prioritization. You can also use it for Quality of Service (QoS) rules.

2. iptables:

- iptables can also be used for basic network traffic control. It allows packet filtering and can be combined with tc for more granular control.

3. **ethtool:**

- ethtool is used for querying and controlling network driver and hardware settings. It can be used to adjust settings like speed, duplex, and autonegotiation on Ethernet interfaces, which can indirectly influence network performance.

4. **nftables:**

- nftables can be used to set up basic traffic control mechanisms.

5. **Wondershaper:**

- Wondershaper is a simpler tool [designed to limit the bandwidth of specific network](#) interfaces. It's a good choice for basic bandwidth management.

6. **VLAN Configuration:**

- Configuring VLANs like we discussed above, can help segment network traffic for more efficient network utilization. Linux's native [VLAN configuration tools can be used with Proxmox](#) for this purpose.

7. **Network Namespaces:**

- Linux network namespaces can be used to isolate network environments for different VMs or containers. This can help manage network traffic and ensure that different services don't interfere with each other's network resources.

8. **Monitoring Tools (like iftop, nload, bmon):**

- While not directly involved in controlling traffic, monitoring tools are crucial for understanding network usage and identifying bottlenecks. Tools like iftop, nload, or bmon provide real-time network bandwidth usage information.

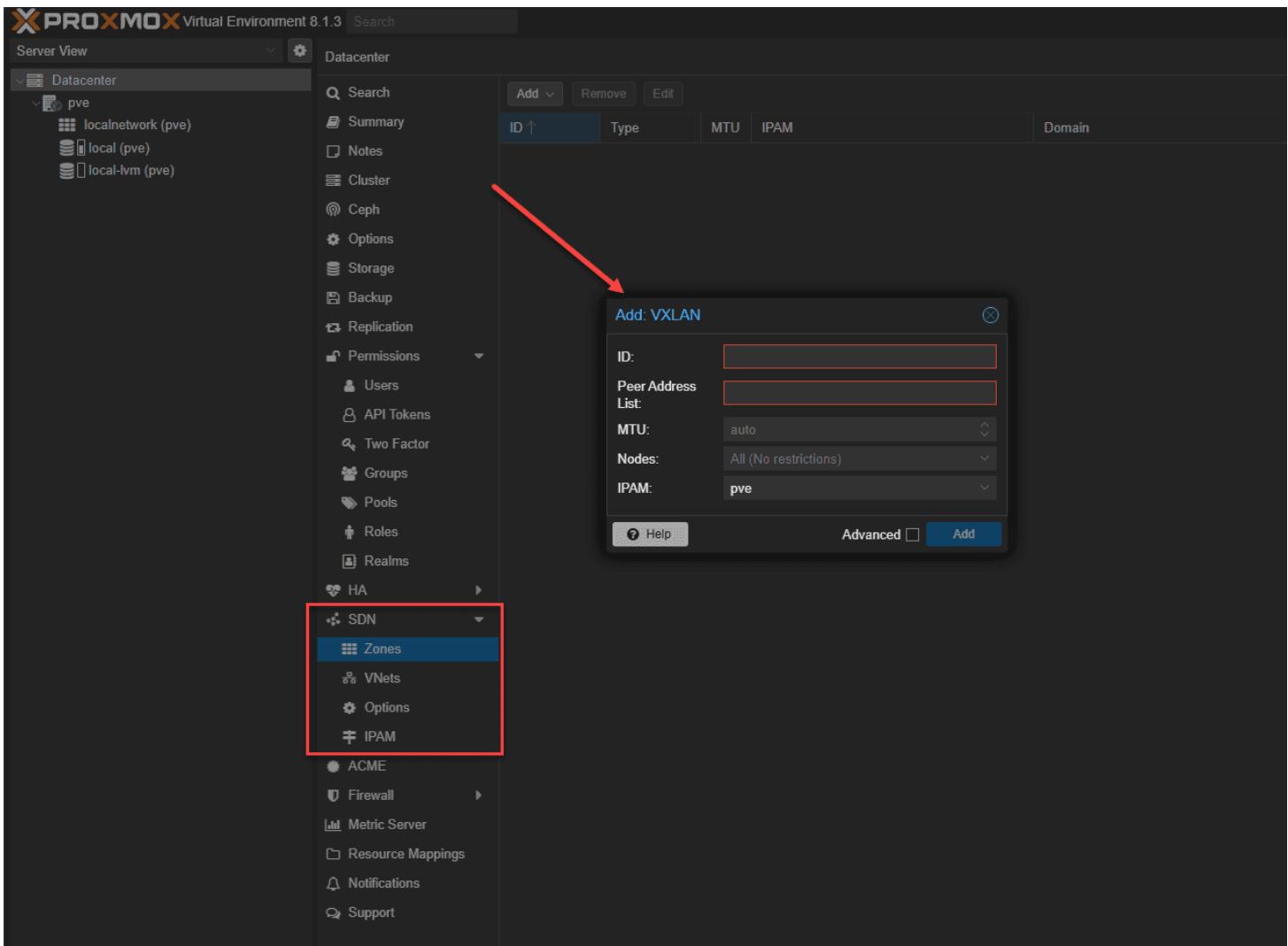
9. **Cgroups (Control Groups):**

- Cgroups, a Linux kernel feature, can be used to limit and prioritize network bandwidth usage among different processes and VMs. While more commonly used for managing CPU and memory resources, cgroups can also be configured to control network I/O.

6. Software-defined networking (SDN)

VMware has a very well-known platform for software-defined networking, known as VMware NSX. The NSX platform is a paid solution on top of vSphere that allows admins to create logical software-defined overlay networks on top of the physical underlay network.

New with Proxmox 8.1 is the introduction of software-defined networking capabilities. You can read the official documentation here. The latest version of Proxmox VE comes with core SDN packages pre-installed. You now have the option for SDN technology in Proxmox VE, allowing admins to create virtual zones and networks (VNets). SDN can also be used for advanced load balancing, NAT, and other features.



Software defined networking in proxmox 8.1

Admins can administer intricate network configurations and multi-tenant environments directly through the web interface at the datacenter level in Proxmox. It allows creating network infrastructure that is more adaptive and responsive and can scale in line with evolving business requirements.

7. Troubleshooting

As you start to work with Proxmox networking, there may be a need for troubleshooting things when networking isn't working correctly. Checking the obvious things like VLANs, VLAN tagging configuration, both in Proxmox, and on your physical network switch are important. If you are using DHCP and DNS to connect to the host, is DHCP handing out the correct IP, and do you have records to resolve the Proxmox host?

Wrapping up

No doubt you have seen various posts and content thread posts from the search [forums and community](#) support forums like the Proxmox support forum related to networking issues. These can be challenging, especially when coming from another hypervisor. Hopefully, this post will help visitors understand Proxmox networking and the security enhancements available like VLANs, SDN, and others. Proxmox networking isn't so difficult to setup once you understand the equivalents from other virtualization environments you may be familiar with.

Proxmox Update No Subscription Repository Configuration

If you are delving into running Proxmox VE for your home lab or other use cases and are coming from other hypervisors you may have been playing around with, like me, you may struggle a bit with some of the basics when getting started learning the platform. One of those tasks is updating Proxmox to the latest Proxmox VE version. Let's take a look at how to update repositories and perform a dist upgrade to the latest version without having a Proxmox subscription.

Learn how to [install Proxmox](#) VE in VMware vSphere:

- [Nested Proxmox VMware installation in ESXi](#)

What is Proxmox VE?

Proxmox VE is an enterprise hypervisor that I have seen really gaining popularity among the home usage community and elsewhere as it provides a readily available and Proxmox works with most hardware that other hypervisors work with.

Proxmox VE is a complete open-source virtualization platform for enterprise virtualization. With PVE you can run [virtual machines and even containers with your Proxmox](#) VE installation.

It also includes a free Proxmox Backup Server that provides an enterprise backup solution for backing up and recovering your virtual machines, containers, and physical hosts, all in one solution.



You can learn more about and download Proxmox VE from here:

- [Proxmox – Powerful open-source server solutions](#)

Updating Proxmox is important

Like any good lifecycle management, upgrading Proxmox VE is best practice to apply and patch any out of date packages that may be on your nodes in your cluster. Updates can also add new features and is part of the general maintenance of your servers. The last thing you want to do is neglect to upgrade your hypervisor platform that you are running your critical virtual machines and containers on. Performing a dist upgrade ensures getting the latest security and other updates for your Proxmox VE solution.

There are a couple of ways you upgrade your Proxmox VE installation, using the Proxmox web interface, or using the `apt get update proxmox ve` and `apt get upgrade` commands from the command line, either at the console or from an SSH connection.

Proxmox VE no subscription upgrade challenges

One of the challenges when starting off with Proxmox VE you may run into is Proxmox VE asks for a valid subscription to upgrade the platform. If you have a PVE no subscription installation, how do you perform a run disk upgrade on the hypervisor for non-enterprise use?

The good news is even if you have a non-licensed version, non-PVE enterprise installation that is not a paid version, you can still retrieve software upgrades on your non-enterprise version to update Proxmox.

Like all other Linux distributions, upgrades and updates pull from a repository for update content. Proxmox VE by default is geared towards production use, and the update and upgrade repositories are pointed to the enterprise repository locations

accordingly.

You can also easily see this in the Proxmox web GUI. Below, we see a freshly installed Proxmox 8.X server that is aside from being pointed to bookworm-updates for security updates and other release configuration, it is pointed to enterprise repositories.

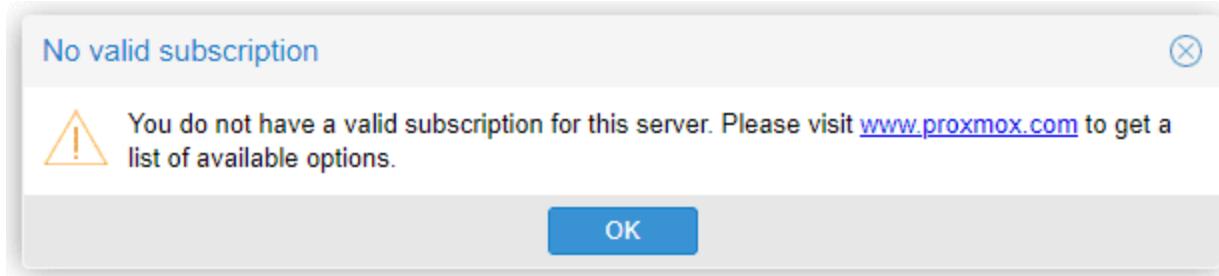
The screenshot shows the Proxmox VE 8.1.3 web interface. In the left sidebar, under 'Updates', the 'Repositories' tab is selected. On the right, the 'APT Repositories' section displays a list of repositories. A red arrow points to a yellow warning box at the top of the list that says 'The enterprise repository is enabled, but there is no active subscription!'. The list includes:

File	Type	URI	Suites	Components
/etc/apt/sources.list (3 repositories)	deb	http://ftp.us.debian.org/debian	bookworm	main contrib
	deb	http://ftp.us.debian.org/debian	bookworm-updates	main contrib
	deb	http://security.debian.org	bookworm-security	main contrib
/etc/apt/sources.list.d/ceph.list (1 repository)	deb	https://enterprise.proxmox.com/debian/ceph-quincy	bookworm	enterprise
/etc/apt/sources.list.d/pve-enterprise.list (1 repository)	deb	https://enterprise.proxmox.com/debian/pve	bookworm	pve-enterprise

Viewing update repositories in the pve console

Default Proxmox VE upgrade settings

The default Proxmox VE upgrade settings point to enterprise repositories. So, when you run software upgrades using the run dist upgrade command you may see the error that you are running PVE no subscription.



No valid subscription configuration

This is because, by default, Proxmox VE points to the enterprise repo to pull down package lists. So, when you download and install Proxmox VE, it is set up for PVE enterprise and the PVE no subscription configuration is something you can introduce. Let's work on the PVE no subscription repository subscription repository.

Task viewer: Update package database

X

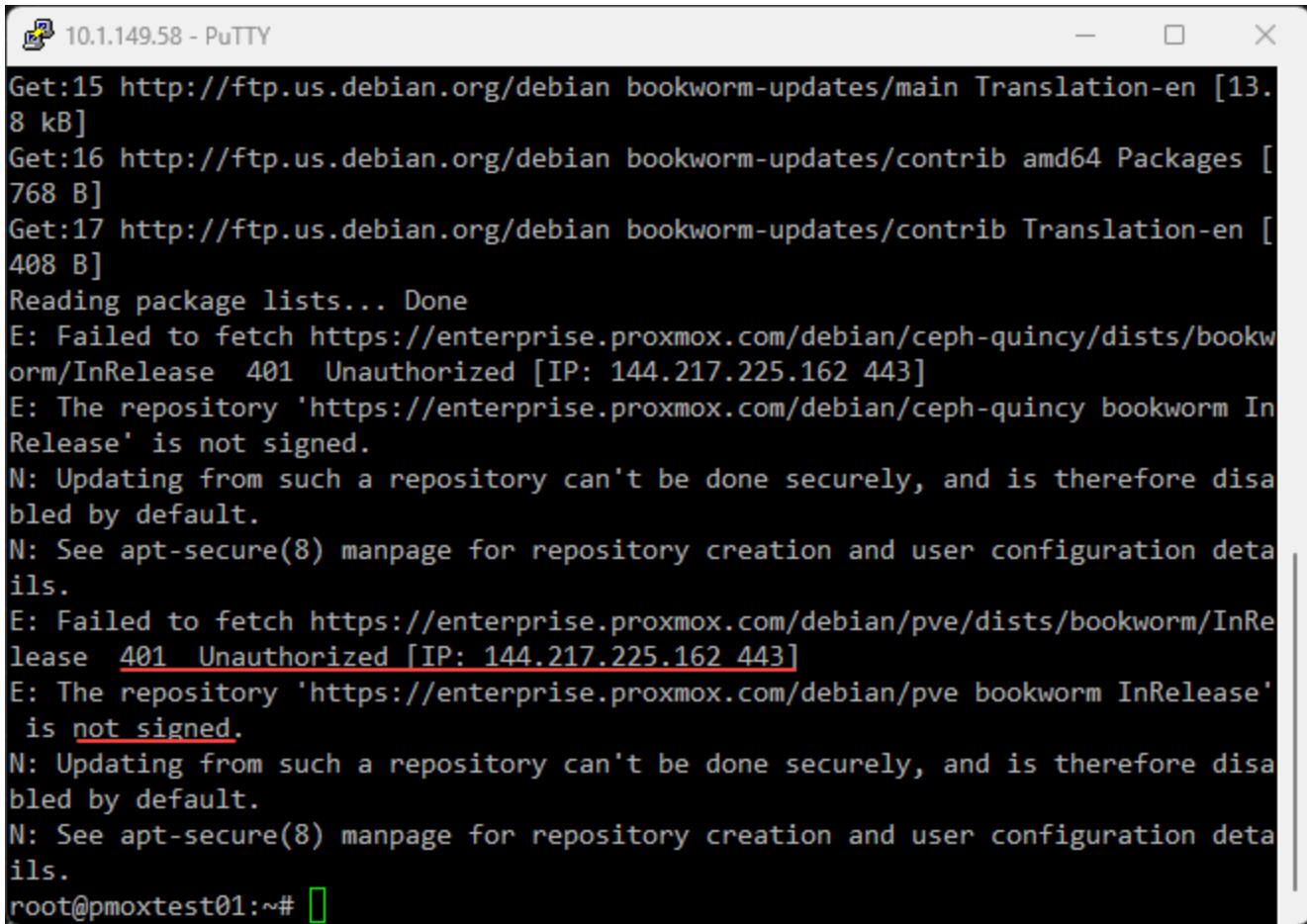
Output Status

Stop

Status	stopped: command 'apt-get update' failed: exit code 100
Task type	aptupdate
User name	root@pam
Node	proxmox
Process ID	1935333
Start Time	2022-07-07 02:52:15
End Time	2022-07-07 02:52:16
Duration	1s
Unique task ID	UPID:proxmox:001D87E5:04DA093B:62C690AF:aptupdate::root@pam:

Update package database error

If you run the updates from the command line, you will see messages about signatures not being valid as well as not being authorized in reading from the repository with various failure or warning messages. The problem relates to which repo it is pointed to.



A screenshot of a PuTTY terminal window titled "10.1.149.58 - PuTTY". The window displays the output of an "apt update" command. The log shows several errors related to failed repository fetches from enterprise.proxmox.com. It specifically mentions "E: Failed to fetch https://enterprise.proxmox.com/debian/ceph-quincy/dists/bookworm/InRelease 401 Unauthorized [IP: 144.217.225.162 443]" and "E: The repository 'https://enterprise.proxmox.com/debian/ceph-quincy bookworm InRelease' is not signed." The terminal ends with the prompt "root@pmonitest01:~#".

```
Get:15 http://ftp.us.debian.org/debian bookworm-updates/main Translation-en [13.8 kB]
Get:16 http://ftp.us.debian.org/debian bookworm-updates/contrib amd64 Packages [768 B]
Get:17 http://ftp.us.debian.org/debian bookworm-updates/contrib Translation-en [408 B]
Reading package lists... Done
E: Failed to fetch https://enterprise.proxmox.com/debian/ceph-quincy/dists/bookworm/InRelease 401 Unauthorized [IP: 144.217.225.162 443]
E: The repository 'https://enterprise.proxmox.com/debian/ceph-quincy bookworm InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
E: Failed to fetch https://enterprise.proxmox.com/debian/pve/dists/bookworm/InRelease 401 Unauthorized [IP: 144.217.225.162 443]
E: The repository 'https://enterprise.proxmox.com/debian/pve bookworm InRelease' is not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
root@pmonitest01:~#
```

Signature error and not authorized

Proxmox Update No Subscription Repository Configuration

What steps are needed to pivot from the enterprise [repository to a no subscription configuration](#) with Proxmox VE? The enterprise repository is defined in the etc apt sources.list.d configuration file. The PVE no subscription repository configuration is defined in the repository files.

Proxmox 8 and higher

The files changed a little with Proxmox 8 and higher. Note the following changes you need to make:

```
#/etc/apt/sources.list.d/pve-enterprise.list
```

```
From: deb https://enterprise.proxmox.com/debian/pve bookworm enterprise
To: deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

```
#/etc/apt/sources.list.d/ceph.list
```

```
#For Ceph Quincy
```

```
From: deb https://enterprise.proxmox.com/debian/ceph-quincy bookworm enterprise
To: deb http://download.proxmox.com/debian/ceph-quincy bookworm no-subscription
```

```
#For Ceph Reef
```

```
From: deb https://enterprise.proxmox.com/debian/ceph-reef bookworm enterprise
To: deb http://download.proxmox.com/debian/ceph-reef bookworm no-subscription
```

Before Proxmox 8

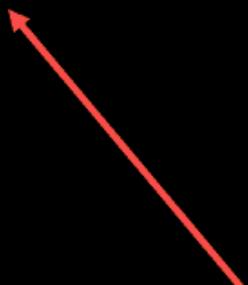
The [steps](#) for setting up a PVE no subscription configuration is configured using the etc apt sources.list.d file found at:

/etc/apt/sources.list

Add the following line in the /etc/apt/sources.list file:

deb http://download.proxmox.com/debian/pve bullseye pve-no-subscription

```
GNU nano 5.4                               /etc/apt/sources.list
deb http://ftp.us.debian.org/debian bullseye main contrib
deb http://ftp.us.debian.org/debian bullseye-updates main contrib
# security updates
deb http://security.debian.org bullseye-security main contrib
# non production use updates
deb http://download.proxmox.com/debian/pve bullseye pve-no-subscription
```



```
^G Help      ^O Write Out   ^W Where Is   [ Read 9 lines ]
^X Exit      ^R Read File    ^N Replace    ^K Cut          ^T Execute
              ^U Paste       ^J Justify    ^C Location    M-U Undo
              ^L Go To Line   M-B Redo
```

Adding the pve no subscription line in the configuration file

Now, we just need to comment out a line in another file located here:

/etc/apt/sources.list.d/pve-enterprise.list

```
GNU nano 5.4          /etc/apt/sources.list.d/pve-enterprise.list
#deb https://enterprise.proxmox.com/debian/pve bullseye pve-enterprise
```

```
^G Help      ^O Write Out  ^W Where Is  [ Read 1 line ]
^X Exit      ^R Read File  ^Y Replace   ^K Cut        ^T Execute
              ^U Paste     ^J Justify   ^C Location  M-U Undo
                                      ^_ Go To Line M-B Redo
```

Editing the pve enterprise.list file

After editing and saving both of the above files, we need to run an **apt-get update** proxmox VE command at the command line.

```
root@proxmox:~# apt-get update
Hit:1 http://ftp.us.debian.org/debian bullseye InRelease
Hit:2 http://ftp.us.debian.org/debian bullseye-updates InRelease
Hit:3 http://security.debian.org bullseye-security InRelease
Hit:4 http://download.proxmox.com/debian/pve bullseye InRelease
Reading package lists... Done
root@proxmox:~#
```

Running an apt get update

After updating the repository with the non enterprise repo, we can perform a non pve enterprise repository upgrade using the command:

```
apt dist-upgrade
```

As you can see below, I have an upgrade that is available for the [Proxmox VE server ready to install after configuring](#) the upgrade to bypass the subscription requirement.

```

Hit:2 http://ftp.us.debian.org/debian bullseye-updates InRelease
Hit:3 http://security.debian.org bullseye-security InRelease
Hit:4 http://download.proxmox.com/debian/pve bullseye InRelease
Reading package lists... Done
root@proxmox:~# apt dist-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following NEW packages will be installed:
 libdrm-common libdrm2 libepoxy0 libgbm1 libproxmox-rs-perl libvirglrenderer1 libwayland-server0
 proxmox-websocket-tunnel pve-kernel-5.13.19-6-pve pve-kernel-5.15 pve-kernel-5.15.39-4-pve
The following packages will be upgraded:
 base-files bash bind9-dnsutils bind9-host bind9-libs bsdextrautils bsutils btrfs-progs chrony
 cifs-utils curl dirmngr distro-info-data dpkg e2fsprogs eject fdisk gnupg gnupg-110n gnupg-utils
 gnutls-bin gpg gpg-agent gpg-wks-client gpg-wks-server gpgconf gpgsm gpgv gzip libarchive13
 libblkid1 libc-bin libc-110n libc6 libcom-err2 libcryptsetup12 libcurls2 libcurl3-gnutls libcurl4
 libexpat1 libext2fs2 libfdisk1 libflac8 libfreetype6 libfribidi0 libgnmp10 libgnutls-dane0
 libgnutls30 libgnutlsxx28 libknet1 libldap-2.4-2 libldb2 liblzma5 libmount1 libnozle1
 libnss-systemd libnss3 libnvpair3linux libpam-systemd libproxmox-acme-perl
 libproxmox-acme-plugins libproxmox-backup-qemu0 libpve-access-control libpve-cluster-api-perl
 libpve-cluster-perl libpve-common-perl libpve-guest-common-perl libpve-http-server-perl
 libpve-rs-perl libpve-storage-perl libpve-u2f-server-perl libsasl2-2 libsasl2-modules-db
 libseccomp2 libsmartcols1 libsmclient libss2 libss11.1 libsystemd0 libtirpc-common libtirpc3
 libtpms0 libudev1 libuuid1 libutil13linux libwbclient0 libxml2 libzfs4linux libzpool5linux
 locales logrotate logsave lxc-pve lxcfs mount nano novnc-pve openssh-client openssh-server
 openssh-sftp-server openssl postfix procmail proxmox-backup-client proxmox-backup-file-restore
 proxmox-ve proxmox-widget-toolkit pve-cluster pve-container pve-docs pve-edk2-firmware
 pve-firmware pve-ha-manager pve-i18n pve-kernel-5.13 pve-kernel-helper pve-lxc-syscalld
 pve-manager pve-qemu-kvm pve-xtermjs python3-1db qemu-server rsyslog samba-common samba-libs
 smartmontools smbclient spl ssh swtpm swtpm-libs swtpm-tools systemd systemd-sysv sysvinit-utils
 tasksel tasksel-data tcpdump tzdata udev util-linux vim-common vim-tiny wget xxd xz-utils
 zfs-initramfs zfs-zed zfsutils-linux zlib1g
150 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 336 MB of archives.
After this operation, 786 MB of additional disk space will be used.
Do you want to continue? [Y/n] 

```

Running an apt dist upgrade command from the command line

Disable Enterprise repo and add non-production from GUI

We can do the same thing from the Proxmox VE web GUI. Navigate to **Updates > Repositories**. Highlight the **enterprise repos** and click the **Disable** button.

The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the navigation tree is expanded to show the Datacenter and the selected node 'pmoxtest01'. Under 'pmoxtest01', there are three local networks: 'localnetwork (pmoxtest01)', 'local (pmoxtest01)', and 'local-lvm (pmoxtest01)'. The 'Repositories' section is selected under 'Updates'. The main panel displays the 'APT Repositories' configuration. At the top, there is a warning icon with the word 'Warning' and a message: 'The enterprise repository is enabled, but there is no active subscription!'. Below this, there are buttons for 'Reload', 'Add', and 'Disable'. The 'Disable' button is highlighted with a red arrow. The table lists repositories from '/etc/apt/sources.list' and '/etc/apt/sources.list.d/ceph.list' and '/etc/apt/sources.list.d/pve-enterprise.list'. The 'enterprise' repository from the pve-enterprise list is currently enabled, indicated by a checkmark.

Disable enterprise from GUI

The enterprise repos are disabled.

This screenshot shows the same APT Repositories page after the enterprise repositories have been disabled. The 'enterprise' repository from the pve-enterprise list is now marked with a minus sign (-) instead of a checkmark, indicating it is disabled. The rest of the repositories listed in the other files remain enabled. A red box highlights the disabled 'enterprise' entry in the pve-enterprise list.

Enterprise repos disabled

Now, click **Add**.

The screenshot shows the Proxmox VE 8.1.3 Server View interface. In the left sidebar, under the 'Datacenter' section, 'pmoxtest01' is selected. Under 'Repositories', the 'APT Repositories' tab is active. A yellow warning icon with an exclamation mark is displayed, and the word 'Warning' is written below it. On the right, a message states: 'The enterprise repository is enabled, but there is no active subscription'. Below this message, there is a table titled 'APT Repositories' with columns for 'Enabled', 'Types', 'URLs', 'Suites', and 'Components'. The table lists several repositories, including 'deb' entries for 'http://ftp.us.debian.org/debian', 'http://ftp.us.debian.org/debian', and 'http://security.debian.org'. It also lists 'File: /etc/apt/sources.list.d/ceph.list' and 'File: /etc/apt/sources.list.d/pve-enterprise.list'. The last entry is 'deb https://enterprise.proxmox.com/debian/pve bookworm pve-enterprise'. At the bottom of the table, there are buttons for 'Reload', 'Add', and 'Disable'.

Add a repo from the GUI

Add the **No-Subscription** repo and then either the **Ceph Quincy** or **Ceph Reef no-subscription** repo.

The screenshot shows the 'Add: Repository' dialog box. The 'Repository:' field contains 'Enterprise'. The 'Description:' field is empty. The 'Status:' field contains 'Test'. A red arrow points to the 'No-Subscription' option in the dropdown menu, which is highlighted in blue. Other options in the dropdown include 'Enterprise', 'Ceph Quincy Enterprise', 'Ceph Quincy No-Subscription', 'Ceph Quincy Test', 'Ceph Reef Enterprise', 'Ceph Reef No-Subscription', and 'Ceph Reef Test'. A 'Help' button is located at the bottom left of the dialog.

Add no subscription repos from GUI

The no-subscription repos have been added.

APT Repositories					
			Reload	Add	Disable
Enabled	Types	URIs	Suites	Components	Options
<input type="checkbox"/> File: /etc/apt/sources.list (4 repositories)					
✓	deb	http://ftp.us.debian.org/debian	bookworm	main contrib	
✓	deb	http://ftp.us.debian.org/debian	bookworm-updates	main contrib	
✓	deb	http://security.debian.org	bookworm-security	main contrib	
✓	deb	http://download.proxmox.com/debian/pve	bookworm	pve-no-subscription !	
<input type="checkbox"/> File: /etc/apt/sources.list.d/ceph.list (2 repositories)					
—	deb	https://enterprise.proxmox.com/debian/ceph-quincy	bookworm	enterprise	
✓	deb	http://download.proxmox.com/debian/ceph-reef	bookworm	no-subscription	
<input type="checkbox"/> File: /etc/apt/sources.list.d/pve-enterprise.list (1 repository)					
—	deb	https://enterprise.proxmox.com/debian/pve	bookworm	pve-enterprise	

No subscription repos enabled

Proxmox VE upgrade FAQs

What is Proxmox VE?

Proxmox VE is an open-source server management platform for enterprise virtualization. It provides integration with the [KVM hypervisor](#) and Linux Containers (LXC), software-defined storage and networking functionality, on a single platform. You can use the web-based user interface to manage virtual machines, LXC containers, Proxmox clusters, or integrate disaster recovery tools.

Why am I getting a Proxmox subscription error about updates?

By default, Proxmox VE is pointed to the enterprise repositories which requires a subscription to perform updates. However, this is a minor configuration change to bypass the enterprise repo and point to the non enterprise repo for pulling down updates.

How do I configure Proxmox for the non enterprise repository?

There are essentially two files that you need to edit, the `/etc/apt/sources.list` file and the `/etc/apt/sources.list.d/pve-enterprise.list` file. After editing the files with the configuration listed above, you run an `apt-get update` and then the command `apt dist-upgrade`.

Is Proxmox free?

Yes, Proxmox is free to download and install in your environment. Additionally, as shown, you can change from the enterprise version of the update proxmox repository to the non enterprise version.

Wrapping Up

Proxmox VE is a great platform for the home lab or for enterprise use and provides many great capabilities to run virtual machines and containerized workloads in your environment. In this article, we have seen that by editing just a few minor configuration files, you can easily bypass the subscription requirement when updating Proxmox VE installations with the latest upgrades. This allows keeping Proxmox installations up to date with the latest security patches and other upgrades from Proxmox.

Be sure to let me know in the comments if you see any issues with the instructions or you are seeing something different. Sign up for the VHT forums and also keep the Proxmox Support forum threads in mind as it is a great resource to look at for Proxmox issues.

Proxmox VLAN Configuration: Management IP, Bridge, and Virtual Machines

If you are running VLANs in your network, you may want your Proxmox VE management IP on your management VLAN, or you may need to connect your virtual machines to separate VLANs. Let's look and see how we can do this.

Table of contents

- [What are VLANs?](#)
- [Network and VLAN terms](#)
- [Proxmox default network configuration](#)
- [Make the default Proxmox VE Linux bridge VLAN-aware](#)
- [Physical network switch tagging](#)
- [Setting the Proxmox Management interface IP on a different VLAN](#)
- [Change Proxmox VE host file reference to old IP](#)
- [Configuring VLANs in Proxmox VE web interface](#)
 - [Web Interface Configuration](#)
- [Advanced Configurations](#)
 - [Trunk Ports](#)
 - [VLAN Aware Bridges](#)
 - [Routing between VLANs](#)
- [Troubleshooting](#)

What are VLANs?

First, let's get a quick primer on what VLANs are exactly. VLANs (**virtual local area networks**) are logical networks you can create that separate devices and broadcast domains. They enable isolating network traffic between [devices all without having separate physical](#) network infrastructure. VLAN-aware network switches can assign a VLAN tag that identifies a unique network and broadcast domain.

Network and VLAN terms

Before diving into VLANs, let's review some essential networking concepts:

network device: A network device is really anything (physical or virtual) that can connect to a computer network

Linux Bridge: A Linux bridge enables more than one network interface to act as a single network device.

Virtual Machine (VM): A virtualized instance of an operating system running on a hypervisor

IP Address: A numeric identifier of network devices on a network. These must be unique.

Default Configuration: The initial settings applied to a device or software.

Networking Service: Software that manages [network connections](#) and traffic flow

Management Interface: In Proxmox VE this is the network interface that allows you to access the web UI and command line interface of your Proxmox host.

Physical Network Interface (NIC): The physical connection from a computer to a physical network switch port.

Network Interfaces File: In Linux systems this is where you setup the network configuration for your network interfaces.

Proxmox default network configuration

In the below screenshots, I am using one of my Supermicro hosts that is configured with (2) 1 GbE connections and (2) 10 GbE connections.

In the [Proxmox network](#) connections, you will see the individual physical adapters and then you will see the Proxmox Linux bridge configured by default.



Proxmox ve 1

Below:

- Individual physical adapters are named **eno1, eno2, eno3, eno4**
- The Linux bridge is called **vmbr0**

A screenshot of the Proxmox Virtual Environment 8.1.3 web interface. The left sidebar shows a tree view of nodes and storage pools. The main area is titled 'Node 'pve01''. A table lists network interfaces. The 'Network' tab is selected. A red box highlights the table header and the first four rows (eno1, eno2, eno3, eno4). The last row, 'vmbr0', is also highlighted with a blue border.

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves
eno1	Network Device	No	No	No	
eno2	Network Device	No	No	No	
eno3	Network Device	Yes	No	No	
eno4	Network Device	No	No	No	
vmbr0	Linux Bridge	Yes	Yes	No	eno3

Viewing the default proxmox network configuration after installation

You can look at the low-level configuration in the following file:

/etc/network/interfaces

10.3.33.14 - PuTTY

```
GNU nano 7.2          /etc/network/interfaces
auto lo
iface lo inet loopback

iface eno3 inet manual

auto vmbr0
iface vmbr0 inet static
    address 10.3.33.14/24
    gateway 10.3.33.1
    bridge-ports eno3
    bridge-stp off
    bridge-fd 0

iface eno1 inet manual
iface eno2 inet manual
iface eno4 inet manual

[ Read 21 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut        ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify  ^/ Go To Line
```

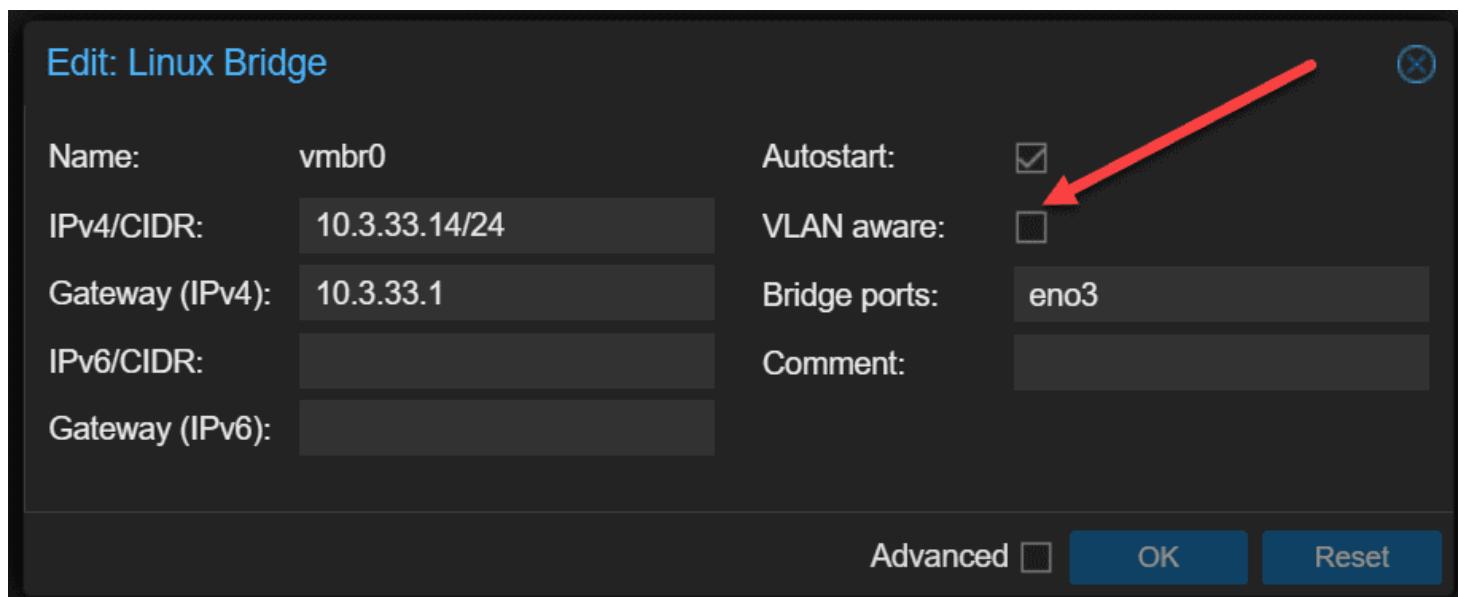
Viewing the network configuration from the network interfaces file

Make the default Proxmox VE Linux bridge VLAN-aware

One of the easiest configurations to implement Proxmox VLANs is called **bridge VLAN aware**. With this configuration, you are simply enabling VLANs on the default **vmbr0** interface.

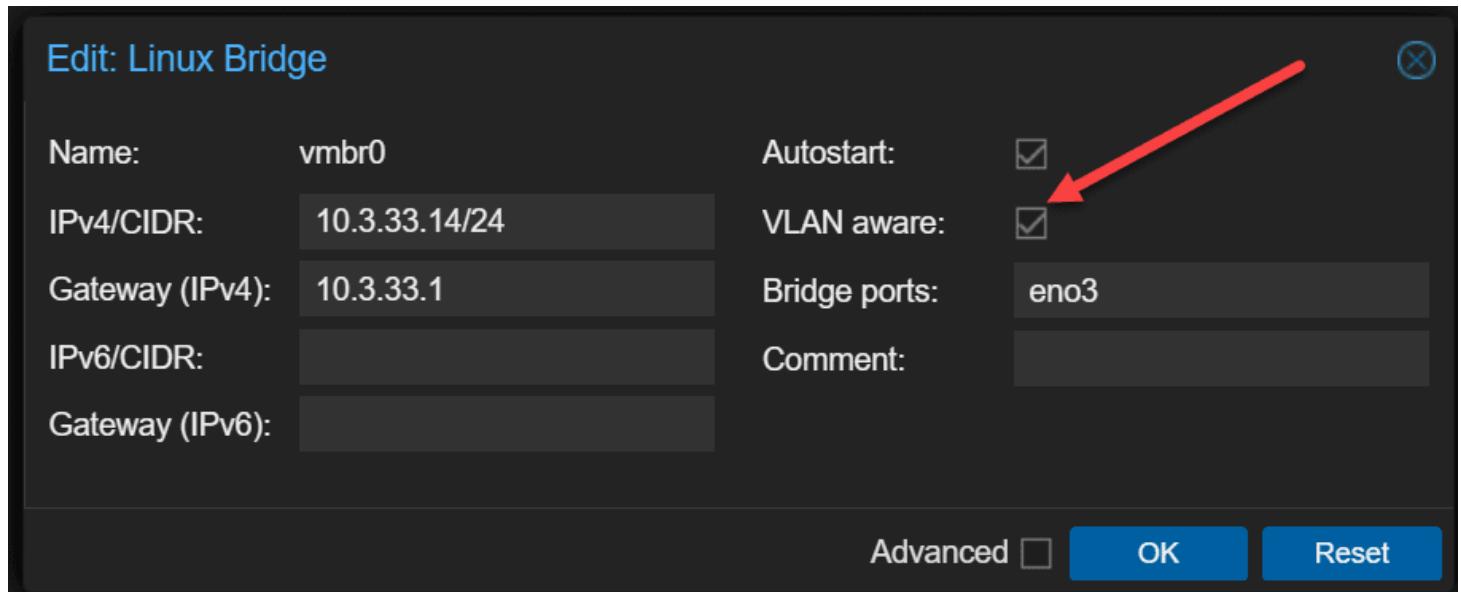
To do this, open the properties of the **vmbr0** interface under your proxmox host properties **Network > vmbr0 > Edit**.

You will see this by default. The **VLAN aware** setting will be unchecked. The bridge port is assigned with the interface that is uplinked.



Vlan aware check box

Now, to make our bridge VLAN-aware, place a check in the VLAN aware box. Click **OK**.



Enabling vlan aware on the default bridge

After you make the change, reboot your Proxmox VE host:

reboot

How does this change the **/etc/network/interfaces** file?

```
GNU nano 7.2          /etc/network/interfaces
iface lo inet loopback

iface eno3 inet manual

iface eno1 inet manual

iface eno2 inet manual

iface eno4 inet manual

auto vmbr0
iface vmbr0 inet static
    address 10.3.33.14/24
    gateway 10.3.33.1
    bridge-ports eno3
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094
```

Configuration added to the network interfaces file after making vlan aware

You will see the configuration change and add the VLAN stanzas in the configuration, as you can see in my configuration.

```

iface lo inet loopback
iface eno3 inet manual
iface eno1 inet manual
iface eno2 inet manual
iface eno4 inet manual

auto vmbr0
iface vmbr0 inet static
address 10.3.33.14/24
gateway 10.3.33.1
bridge-ports eno3
bridge-stp off
bridge-fd 0
bridge-vlan-aware yes
bridge-vids 2-4094

```

```

auto vmbr0
iface vmbr0 inet manual
    address 10.3.33.14/24
    gateway 10.3.33.1
    bridge-ports eno3
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094 ←

```

Trunking configuration allowing multiple VLANs

By default, Proxmox will enable the [Linux bridge with a “trunk port” configuration](#) that accepts all VLANs from 2-4094. You can remove all the VLANs aside from specific VLANs you want to tag, using the following configuration:

```

auto vmbr0
iface vmbr0 inet static
address 10.3.33.14/24
gateway 10.3.33.1
bridge-ports eno3
bridge-stp off
bridge-fd 0
bridge-vlan-aware yes
bridge-vids 10,149,222

```

Below, I have removed the IP from the default bridge, but you can see the restricted bridge-VIDs to specific VLANs.

```

auto vmbr0
iface vmbr0 inet manual
    bridge-ports eno3
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 10,149,222 ←

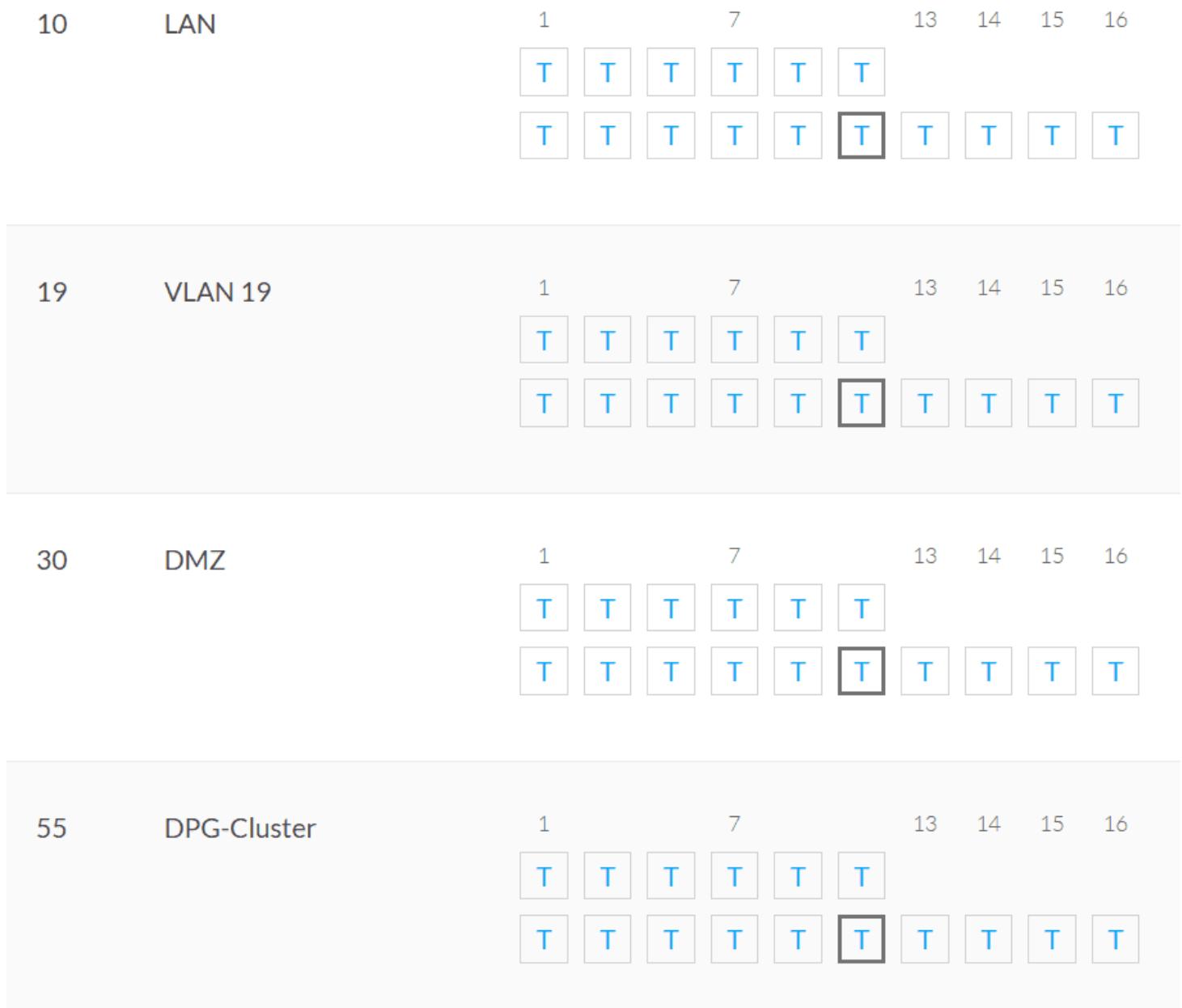
```

Restricting VLANs on the default bridge

Physical network switch tagging

One point to note at this point is that you need to make sure your physical network switch plinking your Proxmox host is tagged for all the VLANs you want the Proxmox bridge to communicate with. If we are tagging frames from the Proxmox side with VLAN IDs that the physical [network switch does not have configured](#), the frames will be discarded.

Below is a screenshot of [VLANs configuration](#) and VLAN setup on my Ubiquiti 10 GbE switch. You can see the [VLANs tagging and trunking configured](#) on the switch. The T stands for "tagged". As you can see below, I have VLANs 10, 19, and 30 tagged on all ports.



Viewing tagged interfaces on a physical network switch

Setting the Proxmox Management interface IP on a different VLAN

What if we want to change the management interface IP and set the management interface IP on a different VLAN? We can do that with the following configuration. As we can see, I have removed the **address** and **gateway** configuration lines from the **vmbr0** configuration.

Instead, I have created a VLAN tagged interface, tagged with VLAN 149 for the management interface.

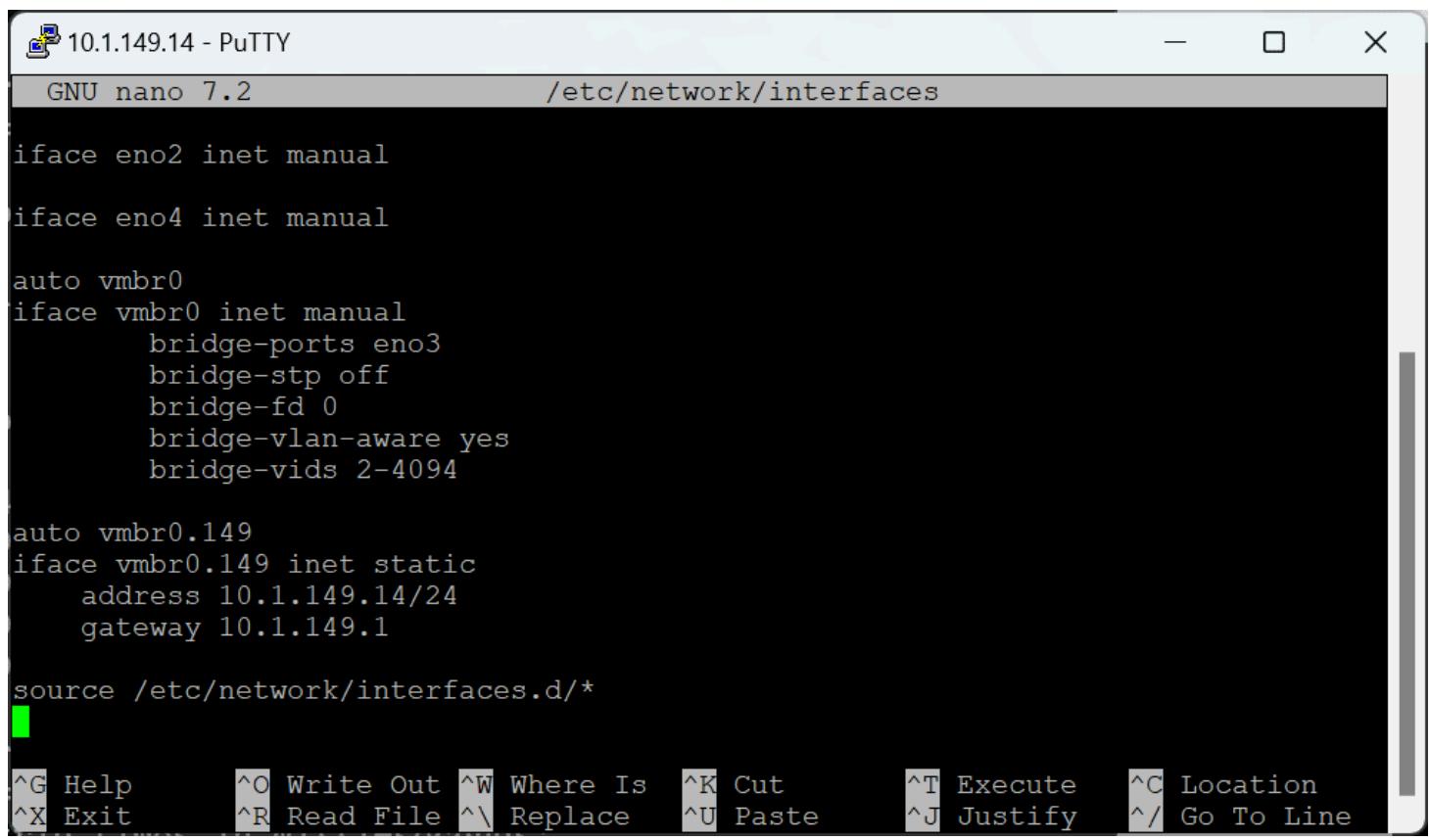
```

iface eno3 inet manual
iface eno1 inet manual
iface eno2 inet manual
iface eno4 inet manual

auto vmbr0
iface vmbr0 inet manual
bridge-ports eno3
bridge-stp off
bridge-fd 0
bridge-vlan-aware yes
bridge-vids 2-4094

auto vmbr0.149
iface vmbr0.149 inet static
address 10.1.149.14/24
gateway 10.1.149.1

```



```

10.1.149.14 - PuTTY
GNU nano 7.2          /etc/network/interfaces

iface eno2 inet manual
iface eno4 inet manual

auto vmbr0
iface vmbr0 inet manual
    bridge-ports eno3
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094

auto vmbr0.149
iface vmbr0.149 inet static
    address 10.1.149.14/24
    gateway 10.1.149.1

source /etc/network/interfaces.d/*

```

Toolbar:

- ^G Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^T Execute
- ^C Location
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Paste
- ^J Justify
- ^/ Go To Line

Management ip for proxmox ve host on a different vlan

Save your configuration. You can reboot to make the configuration take effect, or you can run the command:`ifup -a`

Once you have rebooted or ran the `ifup` command, you should be able to run the **ip address** command to see the IP address and `interfaces:ip a`

```

1000
    link/ether ac:1f:6b:6c:1c:dd brd ff:ff:ff:ff:ff:ff
    altname enp8s0f1
4: eno3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master vmbr0 state UP group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
    altname enp3s0f0
5: eno4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
    altname enp3s0f1
6: vmbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
    inet6 fe80::aelf:6bff:fe6c:222c%64 scope link
        valid_lft forever preferred_lft forever
7: vmbr0.149@vmbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether [REDACTED] brd ff:ff:ff:ff:ff:ff
    inet 10.1.149.14/24 scope global vmbr0.149
        valid_lft forever preferred_lft forever
    inet6 fe80::aelf:6bff:fe6c:222c%64 scope link
        valid_lft forever preferred_lft forever
root@pve01:~#

```

Viewing the ip a command to verify the ip address

We can also verify the configuration in the Proxmox VE GUI, looking at the properties of the Proxmox host > **Network**.

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	B
eno1	Network Device	No	No	No		
eno2	Network Device	No	No	No		
eno3	Network Device	Yes	No	No		
eno4	Network Device	No	No	No		
vmbr0	Linux Bridge	Yes	Yes	Yes	eno3	
vmbr0.149	Linux VLAN	Yes	Yes	No		

Viewing the new linux vlan from the proxmox gui

We can also check external connectivity with a simple ping of the new management IP address we have placed on the new VLAN.

```

Pinging 10.1.149.14 with 32 bytes of data:
Reply from 10.1.149.14: bytes=32 time=3ms TTL=63
Reply from 10.1.149.14: bytes=32 time=2ms TTL=63
Reply from 10.1.149.14: bytes=32 time=2ms TTL=63

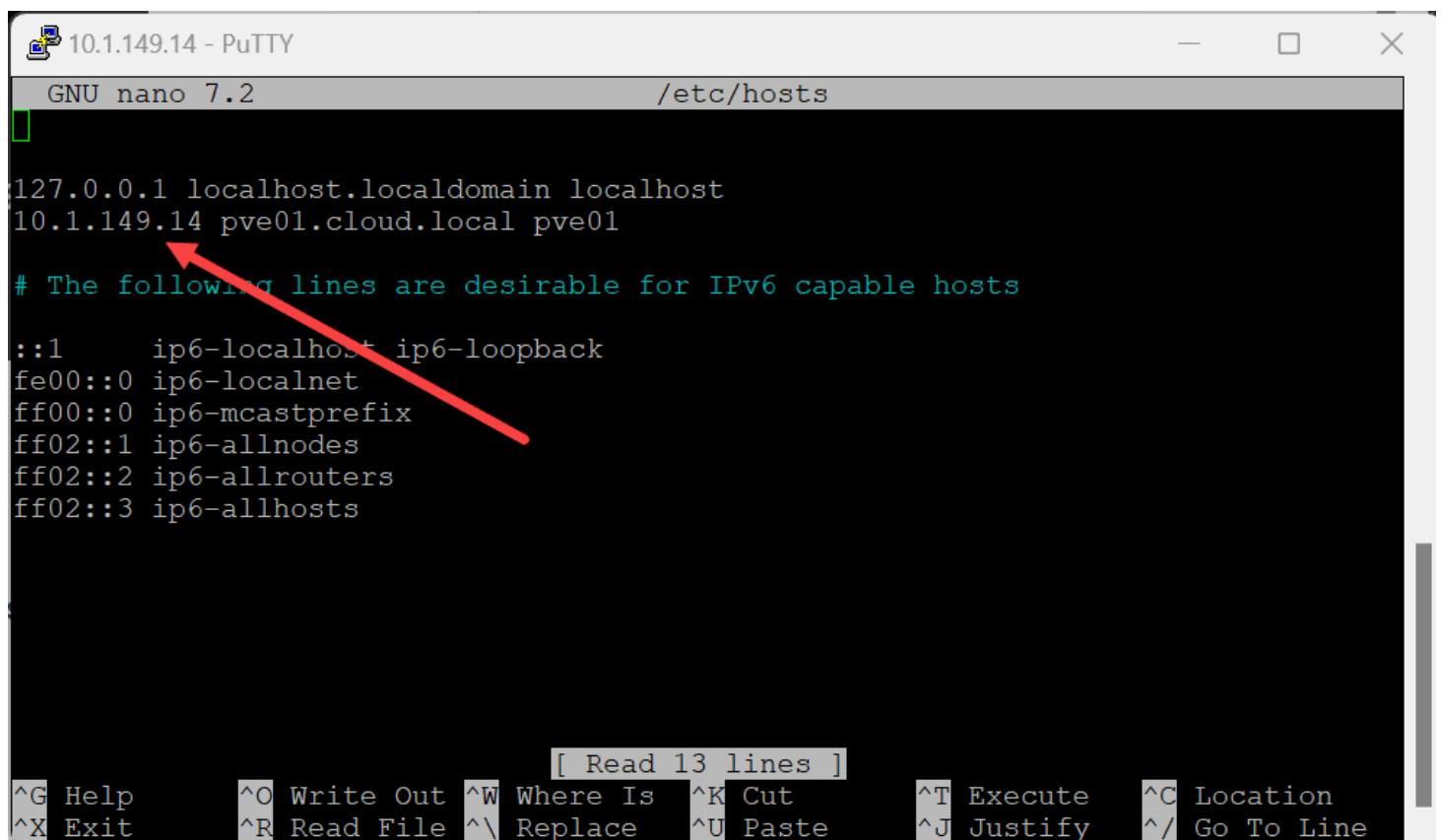
Ping statistics for 10.1.149.14:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 3ms, Average = 2ms

```

Pinging the new management ip on the proxmox host

Change Proxmox VE host file reference to old IP

If you change your Proxmox VE management IP address, you will want to go into your `/etc/hosts` file and change the IP reference for the Proxmox host.`nano /etc/hosts`



```

10.1.149.14 - PuTTY
GNU nano 7.2          /etc/hosts

127.0.0.1 localhost.locaLdomain localhost
10.1.149.14 pve01.cloud.local pve01
# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
ff02::3  ip6-allhosts

[ Read 13 lines ]
^G Help      ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^C Location
^X Exit      ^R Read File  ^\ Replace   ^U Paste    ^J Justify  ^/ Go To Line

```

Changing the linux hosts file

Configuring VLANs in Proxmox VE web interface

To configure VLANs in Proxmox VE on the default bridge using the web interface, we can follow the below.

Web Interface Configuration

The Proxmox VE web interface simplifies VLAN configuration through its GUI.

Create a new Linux VLAN:

- Go to the **Network** section in the web interface.
- Click on **Create** and select **Linux VLAN**.
- Enter a **VLAN ID** and a **Name**.
- Optionally, you can configure other settings, such as the bridge and the VLAN tag.
- Click on **Create** to save the changes.

The screenshot shows the 'Create: Linux VLAN' dialog. It has fields for Name (vmbr0.222), Autostart (checked), IPv4/CIDR, Vlan raw device (vmbr0), Gateway (IPv4), VLAN Tag (222), IPv6/CIDR, and Comment. Below the form is a note: 'Either add the VLAN number to an existing interface name, or choose your own name and set the VLAN raw device (for the latter ifupdown1 supports vlanXY naming only)'. At the bottom are 'Help', 'Advanced' (unchecked), and 'Create' buttons.

Adding a new linux vlan from the proxmox gui

Advanced Configurations

Now that you understand the [basics of VLAN configuration in Proxmox](#) VE, we can explore some advanced topics:

Trunk Ports

A trunk port is a network interface that can carry multiple VLANs traffic. It is a useful configuration for connecting multiple VLANs and VMs to multiple VLANs. To configure a trunk port on Proxmox VE, you need to:

- Make the bridge VLAN aware
- Add the VLAN ID to the bridge configuration. By default it will be a trunking configuration when you make it VLAN aware. Proxmox automatically configures VLAN 2-4094 on the default bridge.
- Configure the VM network interface with the right VLAN tag

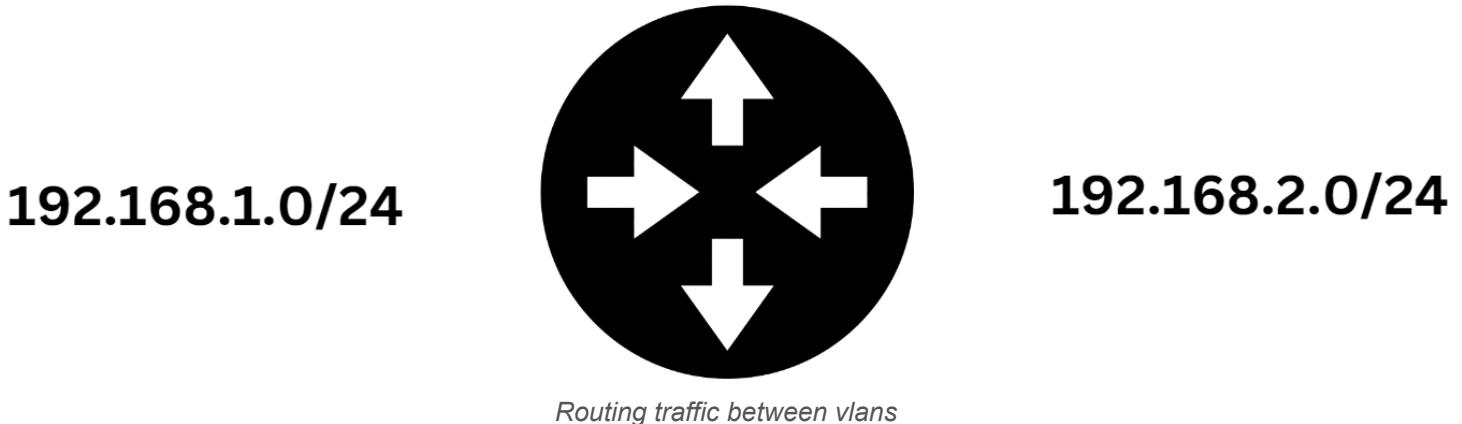
VLAN Aware Bridges

The VLAN bridge is a network type in Proxmox that comes from Linux networking and it allows the Proxmox VE host to be able to understand VLAN tags and forward traffic to the correct VLAN. This is required for communicating between VMs on different VLANs. To configure a VLAN-aware bridge on Proxmox VE, you need to:

- Enable the `vlan_filtering` option in the bridge configuration.
- Add the VLAN ID to the bridge configuration.

Routing between VLANs

When you setup new VLANs, devices on one VLAN can't talk to the devices on the other subnet by default. Generally, according to best practice, a VLAN will house 1 subnet. So it means your devices on each VLAN will have different IP addresses on different subnets. You will need to configure a router or firewall that can do routing (like pfSense) between the devices on different VLANs/subnets so these can communicate.



Troubleshooting

What if you have issues with your Proxmox VLANs?

- Check the syntax you have used in the **/etc/network/interfaces** file
- Make sure there isn't a mismatch between the VLAN tagging in Proxmox and the untagged VLAN on the Switch port. This could result in double-tagging frames (meaning your host is tagging a VLAN, and the Switch is also trying to tag it via untagged traffic)
- If you are restricting specific VLANs in your bridge VIDS configuration, make sure you have allowed the VLANs you are expecting to be tagged
- Make sure your network switch is tagged with all the VLANs on the physical uplink(s) for your Proxmox VE host
- If you are having trouble with one VLAN subnet talking to another VLAN subnet, make sure the appropriate routes are in place to make this happen

Wrapping up

Creating and configuring VLANs in Proxmox is not too difficult. Once you understand the concepts and where to implement the configuration, it is actually quite simple. Adding VLANs to your Proxmox VE host will allow you to connect your virtualized workloads to the various [networks](#) that may be running in your network environment and enable traffic to flow and connect as expected.

Proxmox Management Interface VLAN tagging configuration

If you have configured your Proxmox server in the home lab, most want to segregate their management traffic from the other types of traffic in their lab environment as part of their network configuration. Making the management interface VLAN aware ensures your Proxmox server can be connected to a trunk port and carry traffic for various VLANs. Let's see how to set up the Proxmox management interface VLAN tagging configuration and the steps involved.

Why segment your Promox VE management traffic?

First, why do you want to segment [Proxmox](#) VE management VLAN traffic from the rest of the traffic? Having [management traffic on the same VLAN](#) interface as virtual machines and other types of traffic is a security risk.

You never want to be able to manage the hypervisor host on the same network on which other clients and servers exist. as you can imagine, if an attacker has compromised the network where a client resides, you don't want them to have easy Layer 2 access to the management interface of your hypervisor.

What are VLANs?

VLAN traffic refers to “virtual local area network” traffic that essentially allows creating of many virtual networks on the same Ethernet wire. It is a layer 2 construct. Multiple VLANs allow segmenting traffic across the same physical network switch. Physical interfaces on switch ports are configured with VLAN-aware trunk port configuration, allowing the switch to see the VLAN tags added to Ethernet frames. You can essentially have one port that carries all the different VLAN networks. Two VLANs will flow across the same physical cabling and port.

There are many types of VLAN configurations. You can configure “untagged traffic,” meaning traffic that does not have VLAN tagging, automatically get a specific VLAN tag. Generally, for many, the default VLAN is used for untagged traffic.

VLAN tagging can happen at the switch port level, or the network interface level, as well as the network interface tags VLAN traffic as it traverses the network. We can tag VLANs from the Proxmox side of things so that traffic is correctly tagged with the appropriate VLAN.

VLAN for Guest networks

Below is straight from the [Proxmox documentation](#) from the Proxmox server when you click the Linux vlan help button:

Proxmox VE supports this setup out of the box. You can specify the VLAN tag when you create a VM. The VLAN tag is part of the guest network configuration. The networking layer supports different modes to implement VLANs, depending on the bridge configuration:

VLAN awareness on the Linux bridge: In this case, each guest's virtual network card is assigned to a VLAN tag, which is transparently supported by the Linux bridge. Trunk mode is also possible, but that makes configuration in the guest necessary.

“traditional” VLAN on the Linux bridge: In contrast to the VLAN awareness method, this method is not transparent and creates a VLAN device with associated bridge for each VLAN. That is, creating a guest on VLAN 5 for example, would create two interfaces eno1.5 and vmbr0v5, which would remain until a reboot occurs.

Open vSwitch VLAN: This mode uses the OVS VLAN feature.

Guest configured VLAN: VLANs are assigned inside the guest. In this case, the setup is completely done inside the guest and can not be influenced from the outside. The benefit is that you can use more than one VLAN on a single virtual NIC.

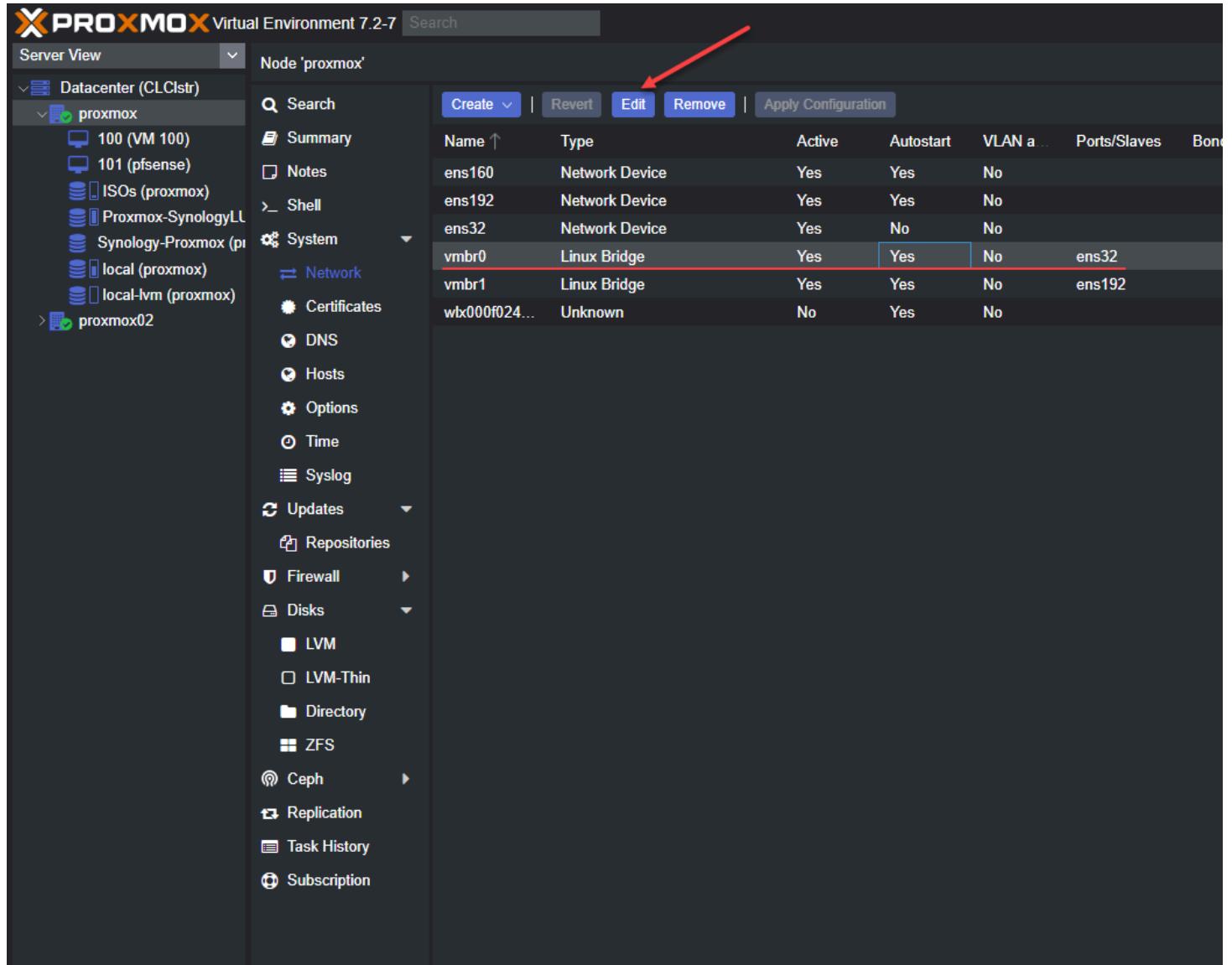
Proxmox management interface VLAN tagging

A couple of steps are involved to correctly tag VLANs from the Proxmox VE side of things. First, we must make the Linux bridge in Proxmox Server VLAN aware. This first step can be completed from the web interface.

Configuring the Linux Bridge to be VLAN aware

We need to navigate to the Promox host > **System > Network** and then **Edit** the properties of the default linux bridge interface in Promox.

Navigating to the default Linux bridge interface in Proxmox server and editing the default bridge in the Proxmox GUI, we click the Edit button in the user interface.



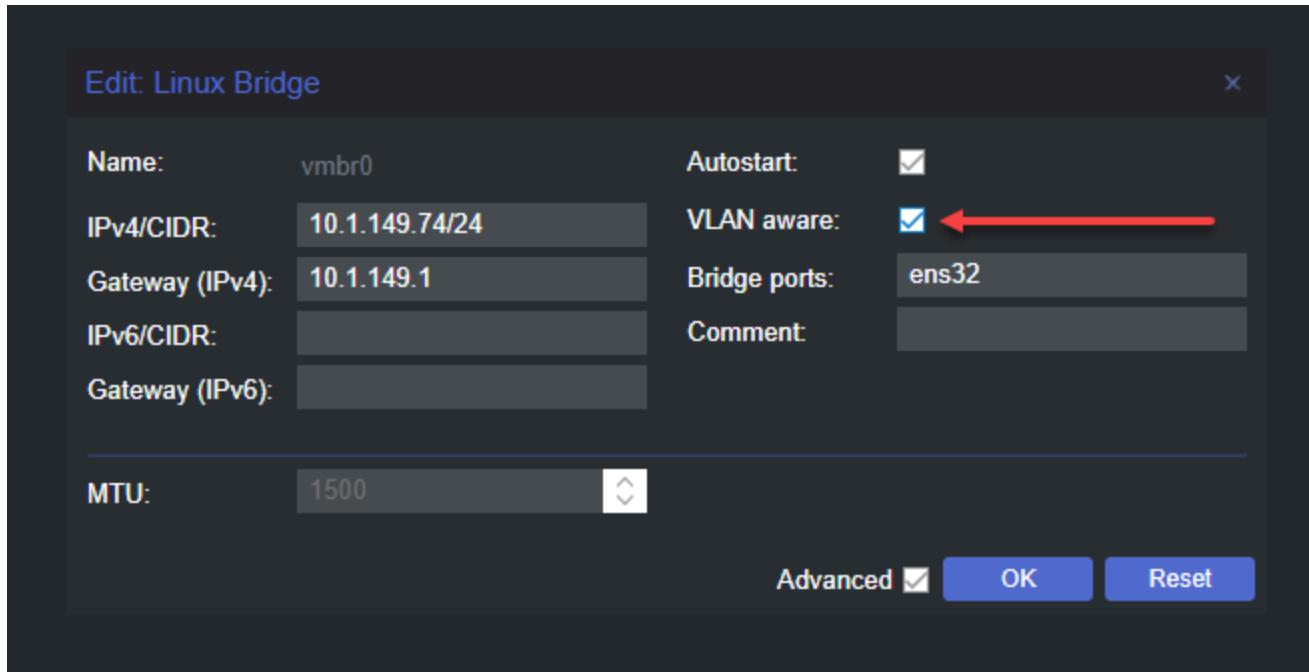
The screenshot shows the Proxmox Web Interface with the title "PROXMOX Virtual Environment 7.2-7". The left sidebar shows a tree view of the system, including "Datacenter (CLC1str)" and "proxmox" (selected). Under "proxmox", there are several items: "100 (VM 100)", "101 (pfSense)", "ISOs (proxmox)", "Proxmox-SynologyLL", "Synology-Proxmox (pr)", "local (proxmox)", "local-lvm (proxmox)", and "proxmox02". The "Network" section is expanded, showing "vmbr0" and "vmbr1" under "Linux Bridge". The main panel shows a table of network interfaces:

Name ↑	Type	Active	Autostart	VLAN a ...	Ports/Slaves	Bond
ens160	Network Device	Yes	Yes	No		
ens192	Network Device	Yes	Yes	No		
ens32	Network Device	Yes	No	No		
vmbr0	Linux Bridge	Yes	Yes	No		ens32
vmbr1	Linux Bridge	Yes	Yes	No		ens192
wlx000f024...	Unknown	No	Yes	No		

A red arrow points to the "Edit" button in the top right of the table header. Below the table, the text "Edit the default bridge in proxmox" is displayed.

Checking the box next to VLAN aware

The first change we need to make is small. We need to tick the box next to VLAN aware. This allows us to configure Proxmox and the Linux bridge to be aware of vlan tagging for the Linux bridge interface.



Checking the box next to vlan aware in proxmox

Applying the configuration

When we edit the network configuration of the Proxmox node, we need to **Apply configuration** to the network changes. This will apply the changes and restart networking services.

The Proxmox Server displays a preview of the etc network interfaces file, which shows the changes made to the default bridge interface:

```
bridge-vlan-aware yes
bridge-vids 2-4094
```

The screenshot shows the Proxmox VE 7.2-7 interface. On the left, the 'Server View' sidebar lists nodes and datacenters. The main panel is titled 'Node 'proxmox'' and shows a table of network interfaces. The 'Apply Configuration' button is highlighted with a red arrow. Below the table, a text area displays pending configuration changes, specifically the addition of 'bridge-vlan-aware yes' to the /etc/network/interfaces file. This line is also highlighted with a red arrow.

Name	Type	Active	Autostart	VLAN a...	Port
ens160	Network Device	Yes	Yes	No	
ens192	Network Device	Yes	Yes	No	
ens32	Network Device	Yes	No	No	
vmbr0	Linux Bridge	Yes	Yes	Yes	ens160
vmbr1	Linux Bridge	Yes	Yes	No	ens192
wlx000f024...	Unknown	No	Yes	No	

```
Pending changes (Either reboot or use 'Apply Configuration' (needs ifupdown2) to activate)

auto ens160
iface ens160 inet static
@0 -35,6 +35,8 @@
    bridge-ports ens32
    bridge-stp off
    bridge-fd 0
+   bridge-vlan-aware yes
+   bridge-vids 2-4094

auto vmbr1
iface vmbr1 inet static
```

Applying the new vlan configuration for the management interface

Making changes to etc network interfaces file for the new Linux bridge interface

This is the first part of the Proxmox server configuration for VLAN-aware traffic on the management VLAN for the Proxmox system. Now we need to make some low-level changes to the etc network interfaces file on the Proxmox host.

Editing the default Linux bridge

We need to edit the file to set the VLAN for the management VLAN and IP address, which is a static address to the new bridge interface tagged with a VLAN.

Below is an example of the default configuration **after** we have turned on the VLAN aware setting.

Default configuration

```
auto vmbr0

iface vmbr0 inet static

    address 10.1.149.74/24
    gateway 10.1.149.1
    bridge-ports ens32
```

```

bridge-stp off
bridge-fd 0
bridge-vlan-aware yes
bridge-vids 2-4094

auto vmbr1
iface vmbr1 inet static
    address 172.16.16.254/24
    bridge-ports ens192
    bridge-stp off
    bridge-fd 0

```

New Linux Bridge VLAN configuration

However, we want to add VLAN tagging from the management interface bridge interface. To do this, we need to change the configuration to the following. Note below, we take the IP address off the iface vmbr0 configuration or iface eno1 inet manual config. However, we leave the VLAN configuration intact. We then create another network interface that is very similar to “subinterface” configuration syntax. We create a vmbr0.<vlan tag> configuration. It is where we place the IP address configuration for the Linux bridge network device.

With this configuration, the Proxmox IP will now be the static IP address and subnet mask is configured in the new bridge interface, since these are virtual interfaces off the main Linux bridge shown with the iface vmbr0 inet manual stanza. You also place the default gateway on the new Linux bridge. You can configure multiple IP addresses across different bridges configured on your Proxmox server.

VLAN config

```

auto vmbr0
iface vmbr0 inet manual
    bridge-ports ens192
    bridge-stp off
    bridge-fd 0
    bridge-vlan-aware yes
    bridge-vids 2-4094

auto vmbr0.333
iface vmbr0.333 inet static
    address 10.3.33.16/24
    gateway 10.3.33.1

```

Network Switch network configuration

After configuring your new Linux Bridge virtual interface, we need to make sure the physical interface of the network switch port is configured as a trunk port to “understand” the [VLAN tagging](#) coming across from the Promox server. The physical port of the switch allows carrying the tagged VLAN traffic to the rest of the network.

The VLAN ID is part of the Layer 2 ethernet frame. If the physical interface of the switch port is not configured correctly, VLAN traffic for the VLAN ID is discarded.

Virtual machines VLAN traffic

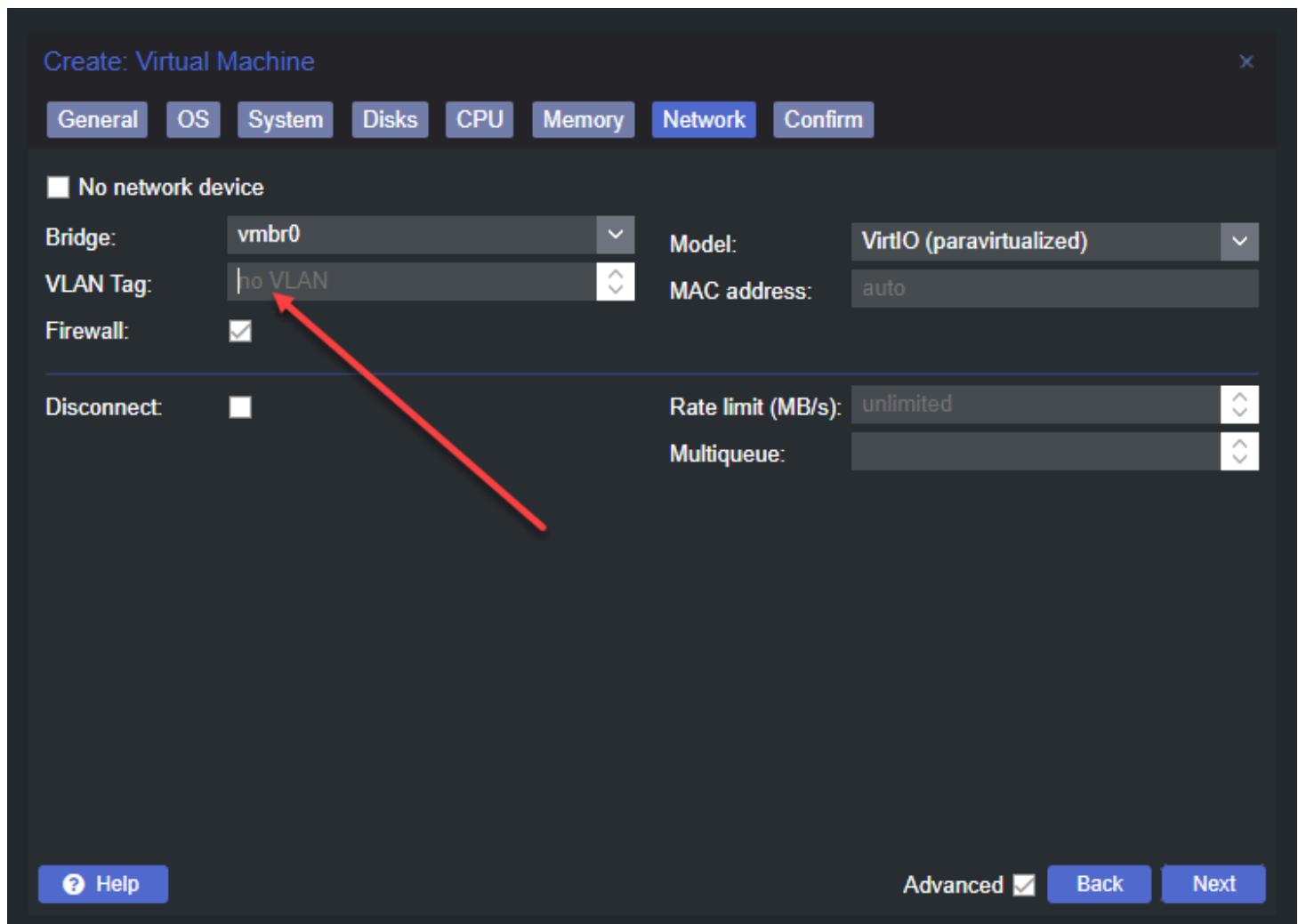
Once we have made the default Linux bridge VLAN aware, virtual machines can also have a [VLAN tag associated with their network configuration](#). It allows the virtual machine to tag VLAN traffic and be placed on that particular VLAN.

Creating VM with the VLAN tag

When you create VMs, you can choose to tag the network traffic with a VLAN ID. This allows sending the virtual machine traffic through the physical device VLAN interface to the rest of the physical network.

The beauty of the VLAN aware bridge is you can have other VLANs configured on other virtual machines, and each can communicate on the required VLAN interface.

Below is an example of the screen to create a new VM and the networking screen. The VLAN tag field allows typing in your VLAN interface number.



Configuring the vlan tag field

Home Lab network configuration

Proxmox is a great choice for building a home lab environment and run VMs on your home network. Once you create your new VLAN in Proxmox server and on your network device, you can start building out your lab environment and have traffic flow as expected. You can make sure your virtual machines are connected to the appropriate networks.

You can also ensure you have Internet access via inter-VLAN routing on your network switch, firewall, router, etc. VLANs create a lot of flexibility from a physical cabling, ports, and virtual configuration, providing many opportunities to allow traffic to flow from your VM guests or physical hosts.

Proxmox resources

Take a look at my Proxmox resources that I have written about below:

- [Proxmox vs ESXi – ultimate comparison 2022](#)
- [pfSense Proxmox Install Process and Configuration](#)
- [Proxmox Update No Subscription Repository Configuration](#)
- [Proxmox iSCSI target to Synology NAS](#)
- [Nested Proxmox VMware installation in ESXi](#)

Proxmox Create ISO Storage Location – disk space error

If you are working with Proxmox in your home lab or otherwise, one of the first things you will want to do is upload ISO installation media to your Proxmox host. You can mount a physical CD to your Proxmox host, of course. However, this is cumbersome and not feasible for remote configurations and installing a wide range of operating systems across the board in the Proxmox environment.

Uploading ISO installation media to your Proxmox host is the way forward for most. If you are like me, you may run into issues with a default installation of Proxmox and the partition size configured for ISO images by default. Let's talk about Proxmox create ISO [storage](#) location and see how this is completed.

Proxmox VE Server and installing operating system guests

[Promox VE Server](#) is a great open-source hypervisor that provides many capabilities and features. It has the capability as a native feature to upload [ISO files](#). You can then select file to select the ISO image you want to use to install guest operating system virtual machine instances.

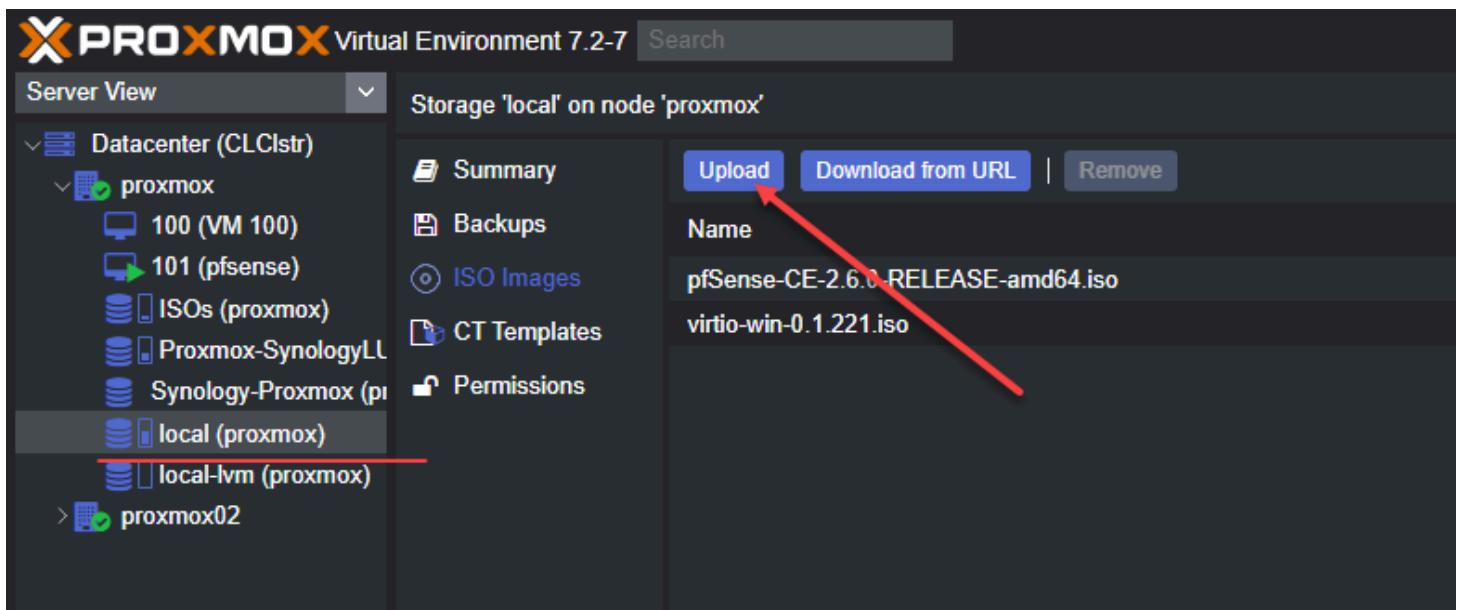
An ISO file is a disk image that most software vendors provide to install operating systems. This includes Linux operating systems like Ubuntu and also Microsoft Windows operating system variants.

When you configure the virtual machine in Proxmox VE, you select the disc image and the ISO image is used as part of the virtual machine installation process.

Uploading Proxmox VE ISO images to the server

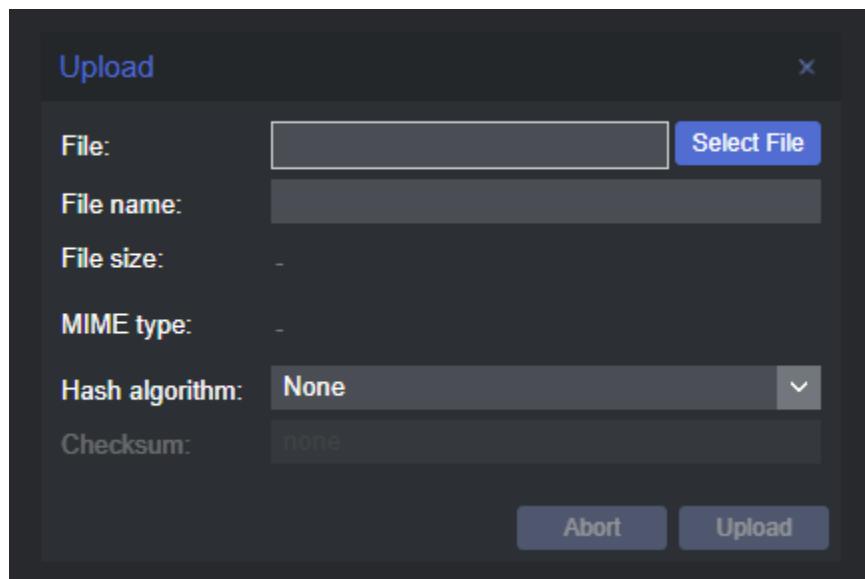
Like [VMware](#) and other hypervisor solutions, the [Proxmox VE server](#) has the means to upload ISO files using the Server view interface.

In the Server View, click the storage pool location > **ISO images** > **Upload**.



Uploading an iso to proxmox ve server

It will launch the box to upload the ISO files. Browse to your ISO file and click the **Select file** button to point to the ISO image you want to use and click finish.

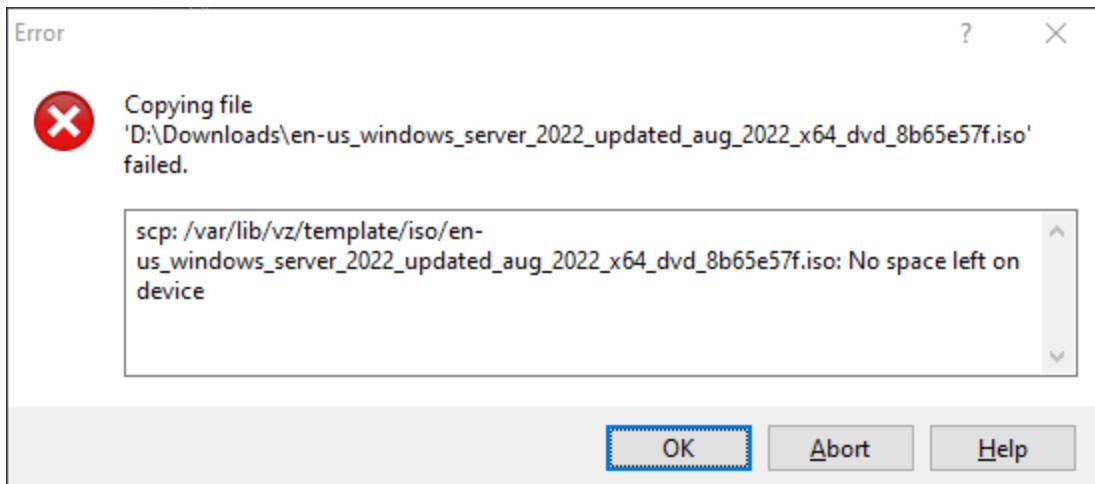


Select the iso file and then upload

As you can see, the upload can be performed from the browser, like other hypervisors, such as VMware.

Proxmox VE ISO files storage disk space error

However, the following disk space error may be one issue with a default Proxmox VE installation.



Disk space error when uploading iso images to proxmox ve

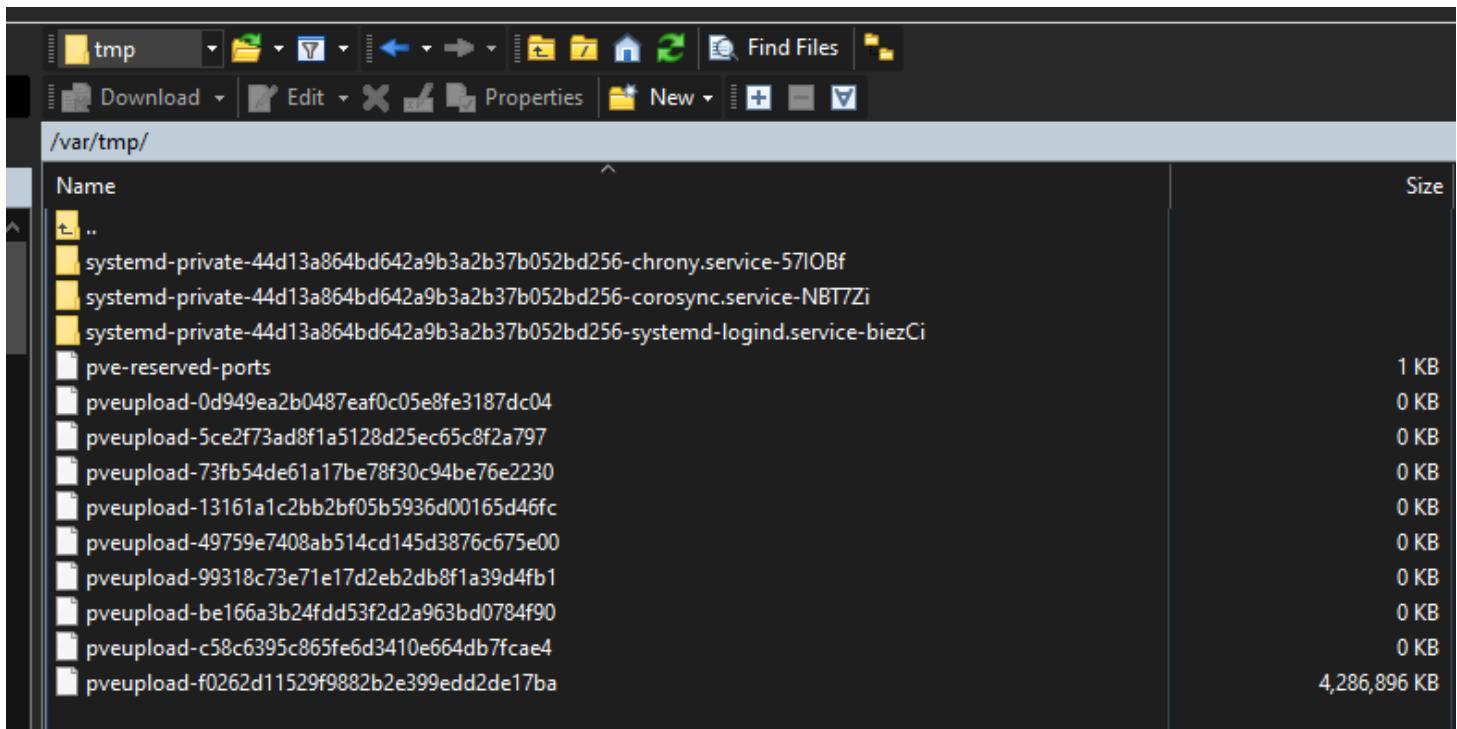
The default Proxmox VE installation storage pool space only included a 10 GB partition for uploading ISO files. When you upload an ISO image to a Proxmox VE server, it will first attempt to upload the ISO image file to the **/var**/

The screenshot shows a PuTTY terminal session connected to '10.1.149.58'. The session starts with root login and password entry. It then displays the standard Debian/GNU Linux welcome message. Following this, the user runs the 'df -h' command to view disk space usage. The output shows the following filesystems and their usage:

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	3.9G	0	3.9G	0%	/dev
tmpfs	796M	976K	795M	1%	/run
/dev/mapper/pve-root	9.5G	9.5G	0	100%	/
tmpfs	3.9G	60M	3.9G	2%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
/dev/sdb1	30G	3.9G	24G	14%	/mnt/pve/ISOs
/dev/fuse	128M	24K	128M	1%	/etc/pve
tmpfs	796M	0	796M	0%	/run/user/0

Viewing disk space on a proxmox ve host

As you can see below, the image file is first uploaded to **/var/tmp** before they are staged into the permanent location found at **/var/lib/vz/template/iso** folder.



Uploaded image files in the tmp directory in proxmox ve

Proxmox Create ISO image Storage Location

How do we create a custom location for ISO storage location for storing your ISO image files in Proxmox VE?

First of all, we need to go through a directory creation process to create a custom location for uploading your operating system ISO files to your Proxmox VE server for creating your Server VM installation or other operating system VM installations.

The process involves the following steps:

1. Add a new hard disk to your Proxmox host
2. Create a new Proxmox directory
3. View the storage location in your Proxmox server web interface

Add a new hard disk to your Proxmox host

The first step in adding an ISO image storage location in Proxmox is to add an additional hard disk where we can create a directory. Below, I have added a new hard disk to the Proxmox host. We can use this in the configuration of the new ISO image storage location, instead of the /var/lib/vz location on the screen.

The screenshot shows the Proxmox VE interface. On the left, the 'Server View' sidebar lists nodes: Datacenter (CLCstr), proxmox (selected), 100 (VM 100), Proxmox-SynologyLL, Synology-Proxmox (p), local (proxmox), local-lvm (proxmox), and proxmox02. The main panel, titled 'Node 'proxmox'', displays a 'Disk' section with a search bar and buttons for Reload, Show S.M.A.R.T. values, Initialize Disk with GPT, and Wipe Disk. Below these are tabs for Search, Summary, Notes, Shell, System, Network, Certificates, DNS, and Hosts. A table lists disk devices:

Device	Type	Usage	Size
/dev/sda	SSD	partitions	42.95 GB
/dev/sda1	partition	BIOS boot	1.03 MB
/dev/sda2	partition	EFI	536.87 MB
/dev/sda3	partition	LVM	42.41 GB
/dev/sdb	SSD	No	32.21 GB

Adding a new hard disk to proxmox ve server host

Viewing the new disk from the command line

You can view the new disk from the Proxmox command line using the command and switch:

df -h

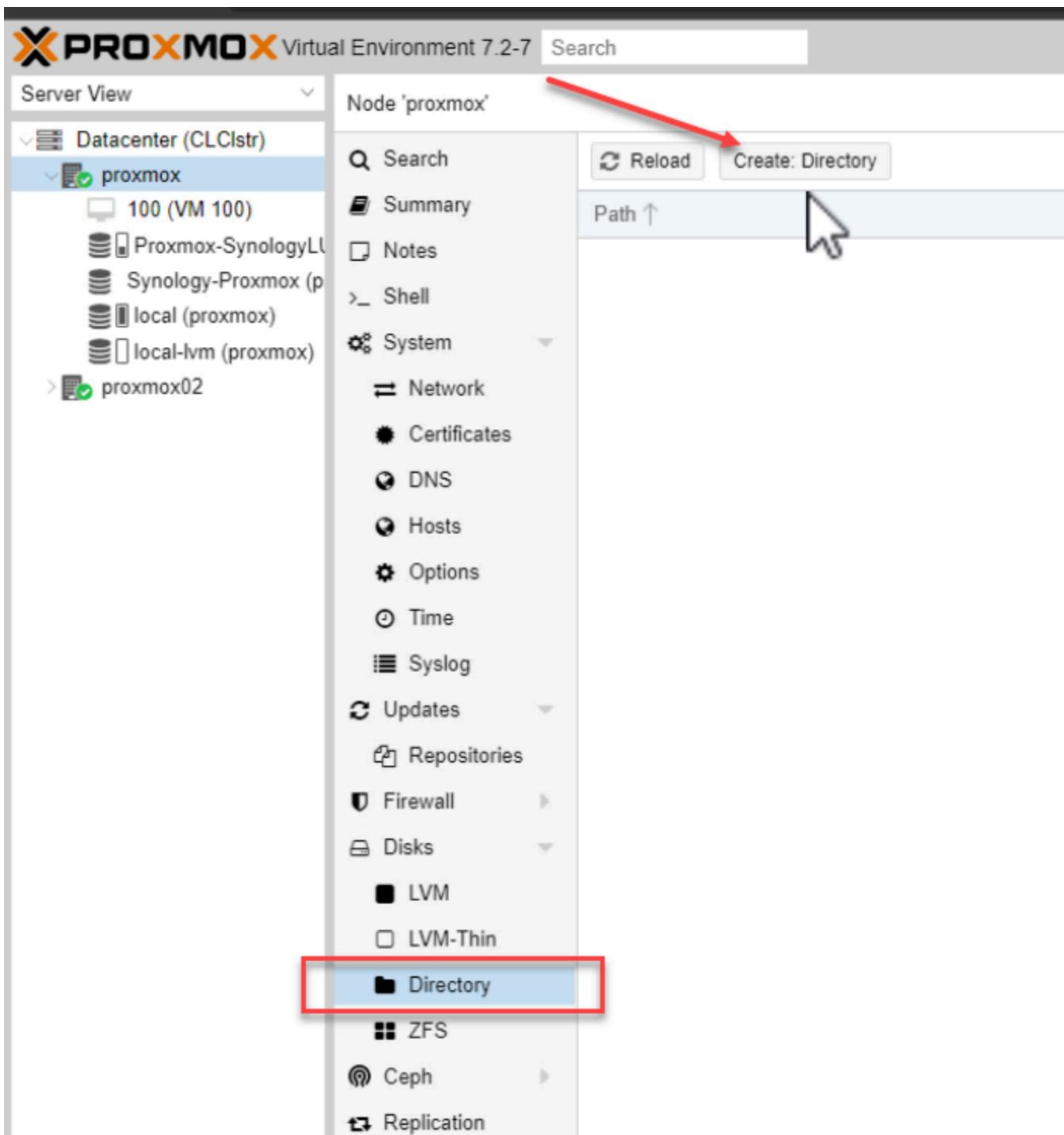
This is the same command you would use in Ubuntu or other Linux distribution for viewing disk space for the store.

```
root@proxmox:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0    3.9G  0% /dev
tmpfs           796M  984K  795M  1% /run
/dev/mapper/pve-root  9.5G  5.4G  3.7G  60% /
tmpfs           3.9G   66M  3.9G  2% /dev/shm
tmpfs           5.0M   0    5.0M  0% /run/lock
/dev/sdb1        30G  4.2G  24G  15% /mnt/pve/ISOs
/dev/fuse        128M  24K  128M  1% /etc/pve
tmpfs           796M   0    796M  0% /run/user/0
root@proxmox:~#
```

Viewing the local storage on a proxmox ve server

Create a new Proxmox Directory for ISO image upload iso files

Next, we navigate to the **Disks > Directory > Create Directory** button in Proxmox. Here we can create the directory we need and format the file storage for uploading ISO files.



Create a new directory in proxmox for iso storage

When you click the **Create directory** button, you will see the following Create Directory dialog box. Select the disk, filesystem, name, and check the box to **Add storage**. Then click the Create button.

Below is an example of [creating a new ISO image storage location](#) on a Proxmox server host. When you create the new directory, you can then ensure your ISO images are stored in the new location when uploading ISOs for creating your new VM hosted in Proxmox.

Create: Directory

Disk: /dev/sdb

Filesystem: ext4

Name: ISOs

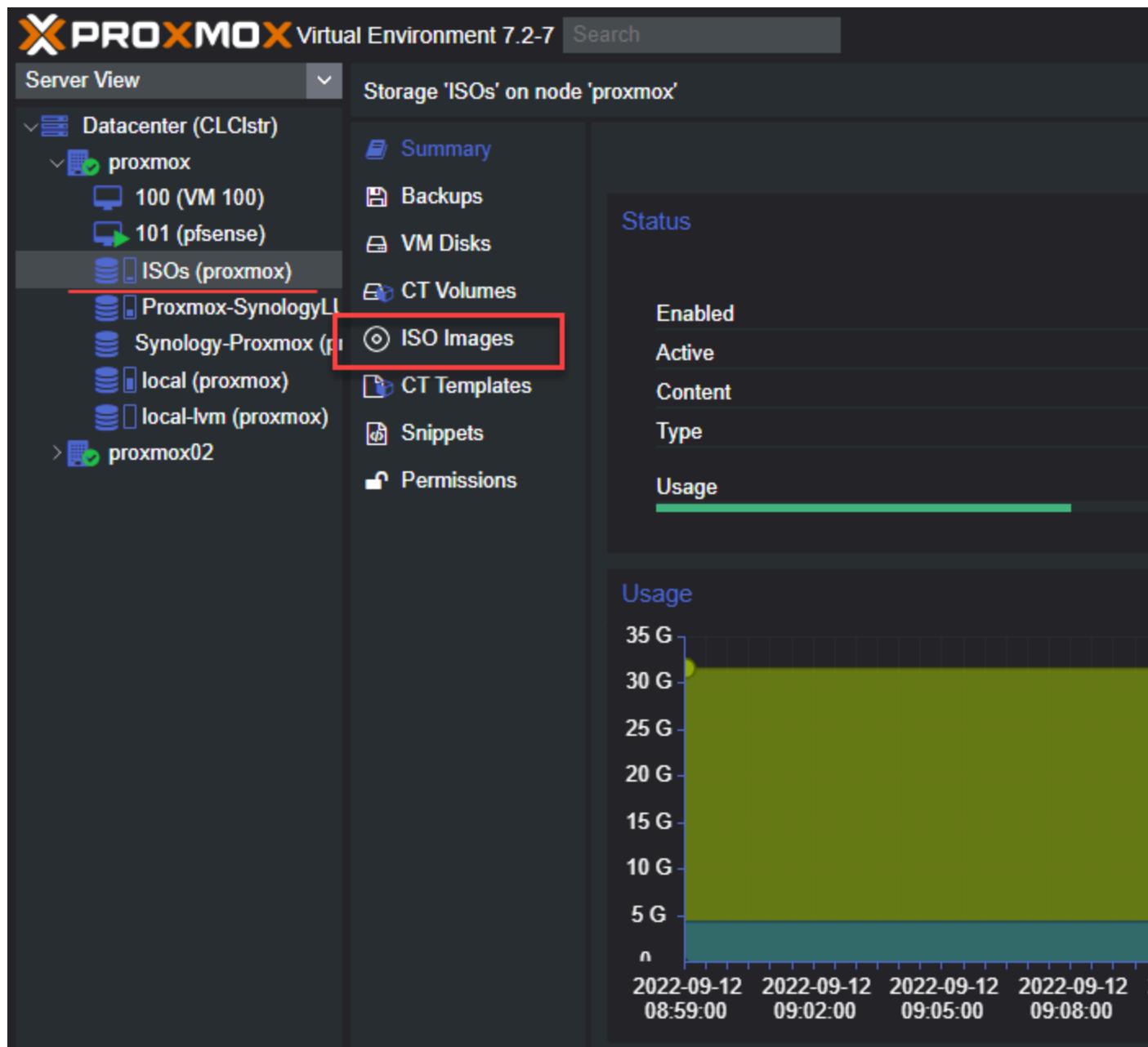
Add Storage:

[Help](#) [Create](#)

Create the directory for proxmox iso image files

Viewing the new Proxmox ISO image location

After creating the new storage location, you will see the ISO image location listed in your Proxmox browser interface on the left-hand side.



New proxmox iso storage location for storing iso image files

Proxmox storage FAQs

What is Proxmox? Proxmox is an open-source hypervisor that allows easily running virtual machines and containers.

What is Proxmox ISO storage? This is storage in Proxmox allowing you to upload ISO files to storage and use these to install VM guests in Proxmox.

How do you upload ISO files to [Proxmox server](#)? You can do this using a web browser logged into your Proxmox host, or you can use SCP and an SCP utility like WinSCP.

Why might you get a disk upload error? If you use the default ISO storage location, you may receive the error on screen when uploading a large iso file operating system installations such as Windows Server 2022.

Wrapping Up

When learning about [Proxmox](#), uploading ISO files to your Proxmox VE server is one of the first steps you will take when loading operating systems on your Proxmox host. If you want to learn about installing Proxmox as a virtual machine in VMware, you can look at [my previous article covering that topic here](#).

Be sure to comment if you have alternative ways of handling the uploading and creation of ISO image file storage in Proxmox.

Proxmox iSCSI target to Synology NAS

Not long ago, I wrote a quick blog post detailing how to install Proxmox inside a VMware virtual machine. However, to really play around with the hypervisor, it is great to have storage to work with. I could've added a local disk to the VM. However, iSCSI sounded way more interesting, especially with the new addition of the Synology DS1621xs+ in the home lab environment. Let's take a look at adding [Proxmox](#) iSCSI target to Synology NAS LUN and see what this process looks like.

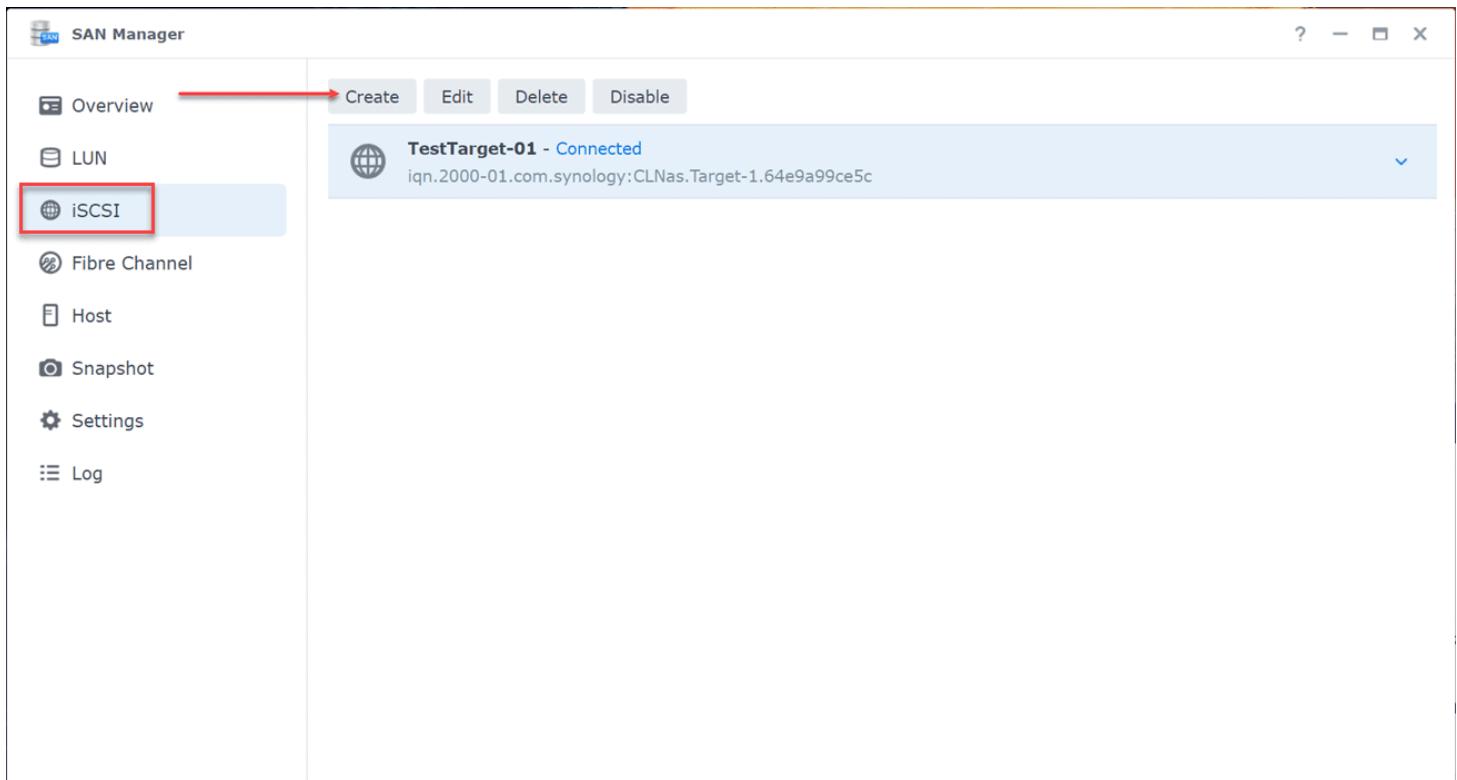
Proxmox iSCSI target to Synology NAS

The steps to complete adding a Synology on Proxmox [hypervisor](#) server looks like the following:

1. Create the iSCSI target on the [Synology NAS](#)
2. Add a dedicated interface to your Proxmox server (if you don't have already)
3. Add the iSCSI target to Proxmox
4. Create the LVM to the Synology iSCSI target

1. Create the iSCSI target on the Synology NAS

Let's first create the iSCSI target on the Synology NAS device. This process is carried out in the Synology SAN Manager. Launch SAN Manager and click **iSCSI > Create**.



Create a new iSCSI target in the Synology SAN Manager

Configure a name for the iSCSI target and configure CHAP if you are using CHAP to secure the connections. For this test, I am leaving CHAP unchecked.

Create a new iSCSI target

Name:

Proxmox-01

IQN:

iqn.2000-01.com.synology:CLNas.Tar



Enable CHAP

Name:

Password:

 Enable Mutual CHAP

Name:

Password:

Next

Name the new iSCSI target and choose CHAP options

From the new iSCSI target wizard, it will prompt you to create or map to a LUN. I am creating a new LUN here.

Set up LUN mapping

You can choose to map a target to the LUN now or after the creation of the LUN.

- Create a new LUN

Create a new LUN and map it to this target.

- Map an existing LUN

Mapped LUN will not be shown in the list.

- Map later

Back

Next

Set up LUN mapping in Synology SAN Manager

Name the new LUN and configure the **Capacity** and the **Space allocation** method (thick or thin).

Set up LUN properties

Name:

ProxmoxLUN-01

Description:

Location:

Volume 1 (Available capacity: 26804 GE ▾)

Total capacity (GB):

100

Space allocation:

Thick Provisioning (better performance) ▾



Thick provisioned LUNs do not support snapshots and space reclamation.

[Back](#)[Next](#)

Set up LUN properties for the new LUN

Review the settings configured and click **Done**.

Confirm Settings

Item	Value
Target name	Proxmox-01
IQN	iqn.2000-01.com.synology:CLNas.Target-11.64e9a99...
Authentication	None
LUN name	ProxmoxLUN-01
Description	
Location	Volume 1 (Available capacity: 26804 GB) - btrfs
Total capacity	100 (GB)
Space allocation	Thick Provisioning
Space reclamation	Disabled

[Back](#)[Done](#)*Confirm the settings for the new iSCSI target and LUN properties*

You will see your new LUN displayed in the list.

The screenshot shows the SAN Manager application window. On the left, there's a sidebar with icons for Overview, LUN, iSCSI (which is selected and highlighted in blue), Fibre Channel, Host, Snapshot, Settings, and Log. At the top right are buttons for Create, Edit, Delete, and Disable. The main pane displays two entries: 'Proxmox-01 - Ready' (with the tooltip 'iqn.2000-01.com.synology:CLNas.Target-11.64e9a99ce5c') and 'TestTarget-01 - Connected' (with the tooltip 'iqn.2000-01.com.synology:CLNas.Target-1.64e9a99ce5c').

The new LUN is displayed in SAN Manager

3. Add a dedicated interface to your Proxmox server (if you don't have already)

On the Proxmox virtual machine, I have added a secondary NIC to the VM for dedicated iSCSI traffic. Now, we need to configure the NIC with an IP address. To do this, in the Proxmox GUI, click your host > Network > <your network adapter> > Edit.

Server View

Node 'proxmox'

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR	Gateway	Co...
ens160	Network Device	No	No	No					
ens32	Network Device	Yes	No	No					
vmbr0	Linux Bridge	Yes	Yes	No	ens32		10.1.149.74/24	10.1.149.1	

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jan 18 21:39:12	Jan 18 21:39:31	proxmox	root@pam	Shell	OK
Jan 18 21:37:11	Jan 18 21:37:11	proxmox	root@pam	Start all VMs and Containers	OK
Jan 14 02:11:45	Jan 14 02:11:47	proxmox	root@pam	Update package database	Error: command 'apt-get upd...
Jan 13 03:24:46	Jan 13 03:24:48	proxmox	root@pam	Update package database	Error: command 'apt-get upd...
Jan 12 22:07:44	Jan 12 22:07:44	proxmox	root@pam	Start all VMs and Containers	OK

Editing the network adapter properties in Proxmox GUI

Enter the IP address you want to configure to communicate with your iSCSI target on the Synology NAS.

Edit: Network Device

Name:	ens160	Autostart:	<input checked="" type="checkbox"/>
IPv4/CIDR:	192.168.77.74/24	Comment:	iSCSI
Gateway (IPv4):	<input type="text"/>		
IPv6/CIDR:	<input type="text"/>		
Gateway (IPv6):	<input type="text"/>		
MTU:	9000		
		Advanced <input checked="" type="checkbox"/>	OK
		Reset	

Configuring the IP address mask and comment

The IP address has been configured. However, the configuration needs to be applied as the Adapter Active status shows No. Click the **Apply Configuration** button at the top.

Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR
ens160	Network Device	No	Yes	No			192.168.77.74/24
ens32	Network Device	Yes	No	No			
vmbr0	Linux Bridge	Yes	Yes	No	ens32		10.1.149.74/24

Pending changes (Either reboot or use 'Apply Configuration' (needs ifupdown2) to activate)

```
--- /etc/network/interfaces      2022-01-09 21:12:51.392943623 -0600
+++ /etc/network/interfaces.new 2022-01-18 21:46:15.510082062 -0600
@@ -1,8 +1,25 @@
+## network interface settings; autogenerated
+## Please do NOT modify this file directly, unless you know what
+## you're doing.
+#
+## If you want to manage parts of the network configuration manually,
+## please utilize the 'source' or 'source-directory' directives to do
+## so.
+## PVE will preserve these directives, but will NOT read its network
+## configuration from sourced files, so do not attempt to move any of
+## the PVE managed interfaces into external files!
+
+auto lo
iface lo inet loopback

iface ens32 inet manual

+auto ens160
+iface ens160 inet static
+    address 192.168.77.74/24
+    mtu 9000
+    #ISCSI
```

IP address is successfully configured

Confirm the new network settings.

@@
terface settings; autogenerated
NOT modify this file directly, unless you know what
ng.

Confirm



Do you want to apply pending network changes?



Yes

No

+ 1 loopback

Confirm to apply the pending network changes

The new network adapter with the configured IP address now shows **Active**.

Create	Revert	Edit	Remove	Apply Configuration			
Name ↑	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond M...	CIDR
ens160	Network Device	Yes	Yes	No			192.168.77.74/24
ens32	Network Device	Yes	No	No			
vmbr0	Linux Bridge	Yes	Yes	No	ens32		10.1.149.74/24

The network adapter now shows to be active

From your Proxmox server, ping your Synology iSCSI address to ensure you have connectivity.

```
Linux proxmox 5.13.19-2-pve #1 SMP PVE 5.13.19-4 (Mon, 29 Nov 2021 12:10:09 +0100) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.
```

```
Last login: Tue Jan 18 21:39:12 CST 2022 on pts/0
```

```
root@proxmox:~# ping 192.168.77.7 ↙
```

```
PING 192.168.77.7 (192.168.77.7) 56(84) bytes of data.  
64 bytes from 192.168.77.7: icmp_seq=1 ttl=64 time=0.373 ms  
64 bytes from 192.168.77.7: icmp_seq=2 ttl=64 time=0.233 ms  
64 bytes from 192.168.77.7: icmp_seq=3 ttl=64 time=0.274 ms  
64 bytes from 192.168.77.7: icmp_seq=4 ttl=64 time=0.215 ms  
64 bytes from 192.168.77.7: icmp_seq=5 ttl=64 time=0.258 ms  
64 bytes from 192.168.77.7: icmp_seq=6 ttl=64 time=0.249 ms  
64 bytes from 192.168.77.7: icmp_seq=7 ttl=64 time=0.256 ms
```

```
^C
```

```
--- 192.168.77.7 ping statistics ---
```

```
7 packets transmitted, 7 received, 0% packet loss, time 6123ms  
rtt min/avg/max/mdev = 0.215/0.265/0.373/0.047 ms
```

```
root@proxmox:~#
```

Verify you have connectivity to your iSCSI portal target of the Synology NAS

3. Add the iSCSI target to Proxmox

Next, lets add the Synology iSCSI target to Proxmox. Click your **Datacenter > Storage > Add**.

The screenshot shows the Proxmox Virtual Environment 7.1-7 interface. On the left, there's a sidebar with 'Server View' and a tree view showing 'Datacenter' and a node named 'proxmox'. The main area is titled 'Datacenter' and contains a 'Storage' section. In the 'Storage' section, there's a dropdown menu with options: 'Add' (highlighted with a red arrow), 'Remove', and 'Edit'. Below the dropdown is a list of storage types: 'Directory', 'LVM', 'LVM-Thin' (which is selected and highlighted with a blue background), 'BTRFS', 'NFS', 'SMB/CIFS', 'GlusterFS', 'iSCSI' (highlighted with a yellow background), 'CephFS', 'RBD', 'ZFS over iSCSI', 'ZFS', and 'Proxmox Backup Server'. To the right of the storage types, there are two columns: 'Type' and 'Content'. The 'LVM-Thin' row shows 'Type' as 'VM-Thin' and 'Content' as 'Disk image, Container'. The 'iSCSI' row is currently empty.

Add a new iSCSI target in Proxmox

Configure the iSCSI ID, Portal, and Target.

Add: iSCSI



General

Backup Retention

ID:	iSCSI-synology	Nodes:	All (No restrictions)
Portal:	192.168.77.7	Enable:	<input checked="" type="checkbox"/>
Target:	iqn.2000-01.com.synolo	Use LUNs directly:	<input checked="" type="checkbox"/>

Help

Add

Fill in the configuration for your Synology iSCSI NAS target

After adding the target, you will see it in your **Storage** list.

Datacenter					
Search Summary Notes Cluster Ceph Options Storage Backup Replication	Add	Remove	Edit		
	ID ↑	Type	Content	Path/Target	Shared
	iSCSI-synology	iSCSI	Disk image	iqn.2000-01.com.syno...	Yes
	local	Directory	VZDump backup file, ISO image, Cont...	/var/lib/vz	No
	local-lvm	LVM-Thin	Disk image, Container		No

New iSCSI target has been added in Proxmox

4. Create the LVM to the Synology iSCSI target

Now that we have the target added, we need to add an LVM to use the iSCSI storage. Click **Storage > Add > LVM**.

The screenshot shows the Proxmox VE web interface under the 'Datacenter' tab. On the left sidebar, the 'Storage' option is selected. In the main area, there is a table listing storage types. The 'LVM' option is highlighted with a yellow background. At the top of the table, there are buttons for 'Add', 'Remove', and 'Edit'. A red arrow points to the 'Add' button.

Type	Content	Path/Target	Shared
SCSI	Disk image	iqn.2000-01.com.syno...	Yes
Directory	VZDump backup file, ISO image, Cont...	/var/lib/vz	No
VM-Thin	Disk image, Container		No

Add a new LVM in Proxmox

Add an ID, Base storage (choose from dropdown), Base volume (choose from dropdown), Volume Group (name this something intuitive), and Content as Disk image, Container.

The screenshot shows the 'Add: LVM' configuration dialog. The 'General' tab is selected. The form fields are:

- ID: Synology-LUN01
- Nodes: All (No restrictions)
- Base storage: iSCSI-synology (iSCSI)
- Enable:
- Base volume: CH 00 ID 0 LUN 1
- Shared:
- Volume group: Synology\SCSI
- Content: Disk image, Container

At the bottom right is a blue 'Add' button.

Configure the base storage pointed to the Synology iSCSI target

You will now see the new iSCSI LUN displayed in your list of storage.

The screenshot shows the Proxmox VE 7.1-7 interface. In the left sidebar, under 'Datacenter', the 'Storage' option is selected. The main area displays a table of storage resources:

ID ↑	Type	Content	Path/Target	Shared	Enabled
Synology-LUN01	LVM	Disk image, Container		Yes	Yes
iSCSI-synology	iSCSI	Disk image	iqn.2000-01.com.syno...	Yes	Yes
local	Directory	VZDump backup file, ISO image, Cont...	/var/lib/vz	No	Yes
local-lvm	LVM-Thin	Disk image, Container		No	Yes

New iSCSI LUN successfully added to Proxmox

Now, when you create a new Virtual Machine, you will see the iSCSI LUN listed as available to select.

The screenshot shows the 'Create: Virtual Machine' dialog box. The 'System' tab is active. The 'EFI Storage:' dropdown menu is open, showing 'Synology-LUN01' as an option. A red arrow points from the text above to this dropdown. Other settings visible include:

- General:** Graphic card: Default, Machine: q35
- System:** SCSI Controller: VirtIO SCSI, Qemu Agent:
- Firmware:**
 - BIOS: OVMF (UEFI)
 - Add EFI Disk:
 - EFI Storage: Synology-LUN01 (highlighted with a yellow box)
 - Format: Raw disk image (raw)
 - Pre-Enroll keys:
- Advanced Options:** Add TPM: (with a warning icon), TPM Storage: (empty dropdown), Version: v2.0

Creating a new Proxmox virtual machine you can choose the Synology iSCSI LUN

Wrapping Up

Hopefully, this quick walkthrough of setting up a [Proxmox iSCSI target to Synology NAS](#) helps to remove any uncertainty of how this is configured. From the Synology NAS side, the process is the same no matter which hypervisor you are using. Generally, the only change in how you add the iSCSI storage comes from the vendor side that you are adding the storage from.

This page intentionally left blank

Proxmox add disk storage space – NVMe drive

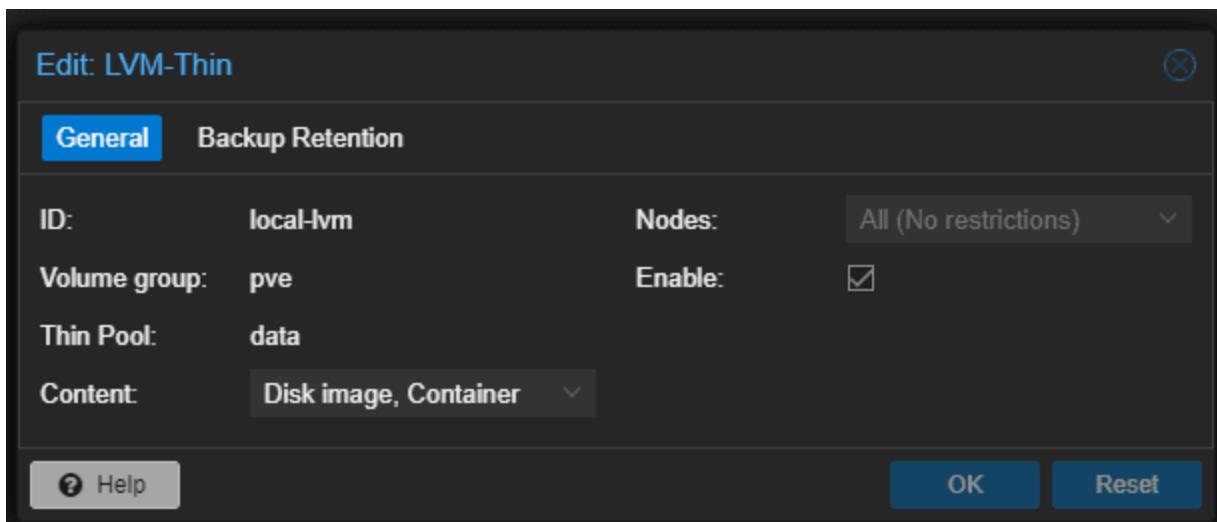
Proxmox and NVMe storage is a great combination for the home lab or production environments. NVMe drives have gotten extremely inexpensive and you can buy large drives for low cost. However, how do you add an NVMe drive to Proxmox disk storage space?

Table of contents

- [Why add disk space to Proxmox?](#)
- [Add the physical drive to your Proxmox host](#)
- [Get the NVMe Drive ready](#)
- [Creating the Primary Partition](#)
- [Mounting the New Partition](#)
- [Adding Storage Space to Proxmox](#)
- [Wrapping up](#)

Why add disk space to Proxmox?

You may need more hard [disk space](#) for a storage drive for virtual machines, containers, images, etc.



You may already have [Proxmox](#) installed on one disk and need to add storage to have more local storage. This process is as simple as adding a drive to Proxmox.

NVMe disks are cheap and great for adding speedy virtualization storage to your [Proxmox host](#).

A screenshot of an Amazon product listing for a Samsung 980 PRO SSD 1TB PCIe 4.0 NVMe M.2 Internal Solid State Drive. The product image shows a black M.2 SSD with "SAMSUNG" and "980 PRO" branding. A red arrow points from the SSD image to the price of \$89.99. The product details include:

- Up arrow icon
- SAMSUNG 980 PRO SSD 1TB PCIe 4.0 NVMe Gen 4 Gaming M.2 Internal Solid State Drive Memory Card, Maximum Speed, Thermal Control, MZ-V8P1T0B
- Visit the SAMSUNG Store
- ★★★★★ 24,900 ratings | 1000+ answered questions
- Amazon's Choice for "evo 980"
- \$89.99
- prime
- FREE Returns

Thank you for being a Prime Member. Pay \$89.99 **\$0.00** for this order. Get a \$125 Amazon Gift Card upon approval for the [Amazon Business Prime Card](#) with an eligible Prime membership. Terms apply.

Add the physical drive to your Proxmox host

The first step is, of course, adding the new drive to your Proxmox host. Most current hosts (mini PCs and servers) have a hard drive M.2 slot for NVMe storage. Once added, we can begin the process to provision the storage.

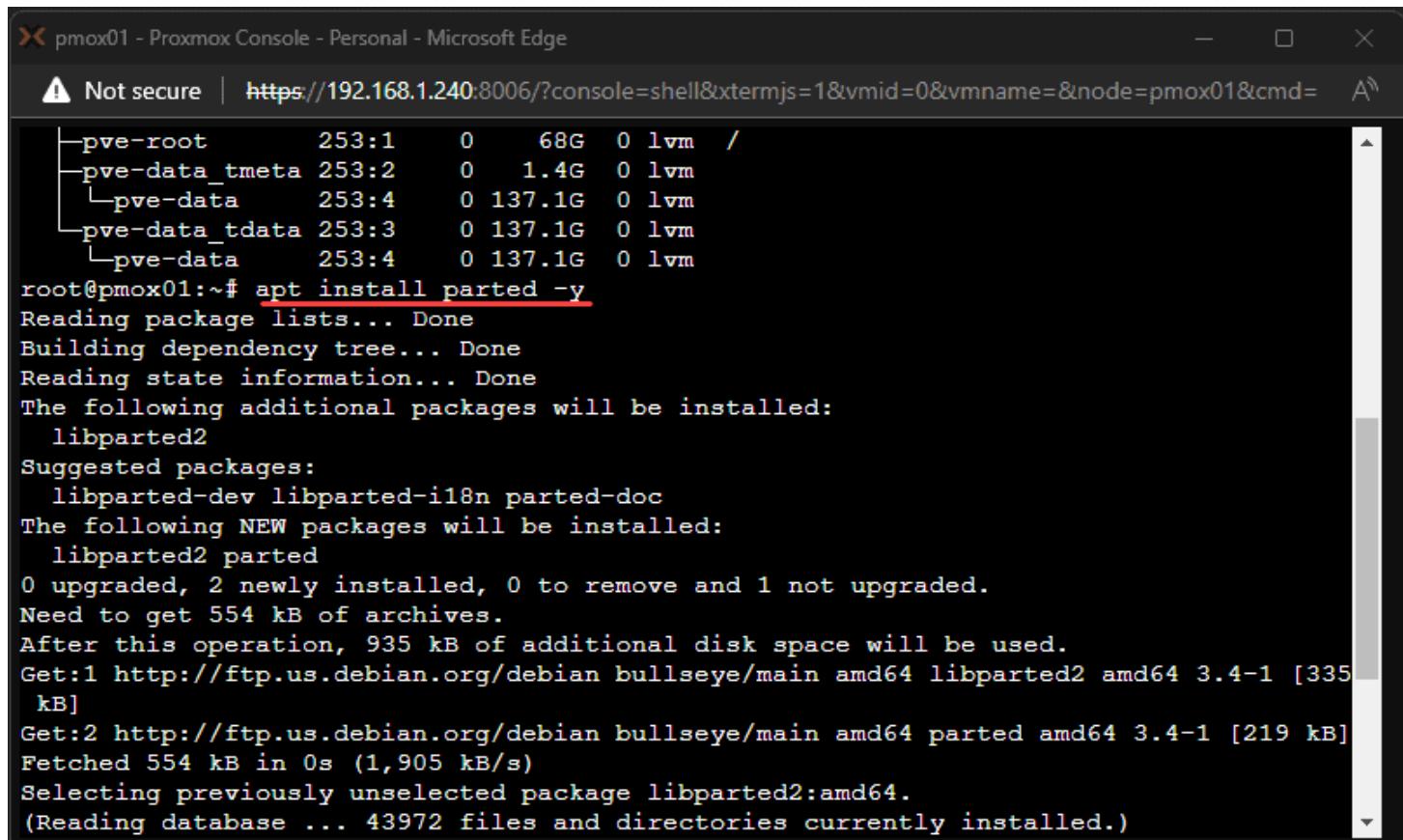
Get the NVMe Drive ready

Before adding storage space to your Proxmox server, you must get the new disk ready. This involves installing necessary tools, creating a new partition table, and setting up the primary partition.

Install Parted:

To begin, install the parted utility on your Proxmox server. Select **Shell** in the web interface to launch the command shell. This tool is used working with your block devices and creating, managing, and destroying partitions. To install Parted, run the following command:

```
apt install parted
```



```
pve-root      253:1      0    68G  0 lvm   /
└─pve-data_tmeta 253:2      0    1.4G  0 lvm
  └─pve-data    253:4      0  137.1G  0 lvm
    └─pve-data_tdata 253:3      0  137.1G  0 lvm
      └─pve-data    253:4      0  137.1G  0 lvm
root@pmax01:~# apt install parted -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libparted2
Suggested packages:
  libparted-dev libparted-i18n parted-doc
The following NEW packages will be installed:
  libparted2 parted
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 554 kB of archives.
After this operation, 935 kB of additional disk space will be used.
Get:1 http://ftp.us.debian.org/debian bullseye/main amd64 libparted2 amd64 3.4-1 [335 kB]
Get:2 http://ftp.us.debian.org/debian bullseye/main amd64 parted amd64 3.4-1 [219 kB]
Fetched 554 kB in 0s (1,905 kB/s)
Selecting previously unselected package libparted2:amd64.
(Reading database ... 43972 files and directories currently installed.)
```

List block devices:

Next, we can use the **lsblk** command to pinpoint the new NVMe drive we want to add to our Proxmox installation. For example, the drive may appear as **dev/nvme0n1**. You will see the normal disks such as dev sda dev sda1.

pmax01 - Proxmox Console - Personal - Microsoft Edge

⚠ Not secure | https://192.168.1.240:8006/?console=shell&xtermjs=1&vmid=0&vmname=&node=pmax01&cmd=A

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Apr 10 15:46:49 CDT 2023 on tty1
root@pmax01:~# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    1  58.9G  0 disk
└─sda1     8:1    1   224K  0 part
sda2       8:2    1   2.8M  0 part
sda3       8:3    1   1G   0 part
└─sda4     8:4    1   300K  0 part
nvme0n1   259:0   0 931.5G  0 disk
└─nvme0n1p1 259:2   0   2M  0 part
└─nvme0n1p2 259:3   0 931.5G  0 part
nvme1n1   259:1   0 232.9G  0 disk
└─nvme1n1p1 259:4   0 1007K  0 part
└─nvme1n1p2 259:5   0   1G  0 part
└─nvme1n1p3 259:6   0 231.9G  0 part
    ├─pve-swap 253:0   0   8G  0 lvm  [SWAP]
    ├─pve-root  253:1   0   68G  0 lvm  /
    └─pve-data_tmeta 253:2   0   1.4G 0 lvm
        └─pve-data  253:4   0 137.1G 0 lvm
        └─pve-data_tdata 253:3   0 137.1G 0 lvm
            └─pve-data  253:4   0 137.1G 0 lvm
root@pmax01:~# 
```

Create a new partition table:

Once you have pinpointed the new disk, you can create a new partition table. Run the following command to create a type GPT partition table on the new NVMe drive:

```
parted /dev/nvme0n1 mklabel gpt
```

```
root@pmax01:~# ^C
root@pmax01:~# parted /dev/nvme0n1 mklabel gpt
Warning: The existing disk label on /dev/nvme0n1 will be destroyed and all data on
this disk will be lost. Do you want to continue?
Yes/No? Yes
Information: You may need to update /etc/fstab.

root@pmax01:~# 
```

Creating the Primary Partition

With the new admin-prepared disk that is ready to go for storage, the next step you need as an admin is to create the primary partition on the NVMe drive. The partition will actually store the data.

Create a primary partition

Run this command to create the partition we need on the NVMe drive.

```
parted /dev/nvme0n1 mkpart primary ext4 0% 100%
```

```
root@pmax01:~# parted /dev/nvme0n1 mkpart primary ext4 0% 100%
Information: You may need to update /etc/fstab.
```

With this command we are creating a **new primary ext4 partition** that will span across the entire NVMe drive and will remove existing partitions which is a destructive process. So be careful and ensure this is what you would like to do.

Format the partition:

Next, format the new partition with the ext4 filesystem:

```
mkfs.ext4 /dev/nvme0n1p1
```

Name the storage for your data

You can name the storage something intuitive using the command:

```
mkfs.ext4 -L vmstorage /dev/nvme0n1
```

```
root@pmox01:~# mkfs.ext4 -L vmstorage /dev/nvme0n1
mke2fs 1.46.5 (30-Dec-2021)
Found a gpt partition table in /dev/nvme0n1
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 244190646 4k blocks and 61054976 inodes
Filesystem UUID: 61cf39ed-a308-4eae-a9d2-a019eaaa1697
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
     4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
    102400000, 214990848

Allocating group tables: done
Writing inode tables: done
Creating journal (262144 blocks): done
Writing superblocks and filesystem accounting information: done

root@pmox01:~# 
```

Mounting the New Partition

Now that we have the primary partition created, you must now mount it to a mount point on your Proxmox server.

Create a new directory:

First, create a new directory to serve as the mount point for the new partition. For example:

```
mkdir /mnt/vmstorage
```

Edit the /etc/fstab file:

Next, edit the /etc/fstab file to ensure the new partition will **auto mount upon reboot**. Open the file with a text editor like Nano:

```
nano /etc/fstab
```

```
root@pmox01:~# mkdir -p /mnt/vmstorage
root@pmox01:~# nano /etc/fstab
root@pmox01:~# 
```

Add the following line to the file, replacing `/dev/nvme0n1p1` with the appropriate device identifier for your NVMe drive:

```
LABEL=vmstorage /mnt/vmstorage ext4 defaults 0 2
```

GNU nano 5.4

```
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/pve/root / ext4 errors=remount-ro 0 1
/dev/pve/swap none swap sw 0 0
proc /proc proc defaults 0 0
LABEL=vmstorage /mnt/vmstorage ext4 defaults 0 2
```

Save the changes and exit the text editor.

Mount the new partition:

Finally, mount the new partition to the mount point:

```
mount /mnt/vmstorage
```

Adding Storage Space to Proxmox

Now that the new partition is mounted, you can add it as storage space in the Proxmox web interface.

1. Browse out to your Proxmox web interface and login.
 2. Go in the interface to the “Datacenter” tab and click on “Storage.”
 3. Click on the “Add” button. Then you will want to “Directory” from the dropdown menu. There will be a few options including **lvm thin**, think provisioning, volume group and others.

The screenshot shows the Proxmox Virtual Environment 7.4-3 interface. The left sidebar is titled "Server View" and shows a tree structure with "Datacenter" selected, which is expanded to show "pmox01" and its sub-storages: "local (pmox01)", "local-lvm (pmox01)", and "vmstorage (pmox01)". The main area is titled "Datacenter" and contains a navigation menu on the left with the following items: Search, Summary, Notes, Cluster, Ceph, Options, Storage (which is highlighted with a blue background), Backup, Replication, Permissions (with sub-items Users and API Tokens), Two Factor, Groups, and Pools. To the right of the menu is a table with a red border. The table has two columns: "Type" and "Storage". The "Type" column lists the storage types: Directory, LVM, LVM-Thin, BTRFS, NFS, SMB/CIFS, GlusterFS, iSCSI, CephFS, RBD, ZFS over iSCSI, ZFS, and Proxmox Backup Server. The "Storage" column lists the corresponding storage names: local (pmox01), local-lvm (pmox01), vmstorage (pmox01), and pmox01 respectively.

Type	Storage
Directory	local (pmox01)
LVM-Thin	local-lvm (pmox01)
Directory	vmstorage (pmox01)
Directory	pmox01
LVM	
BTRFS	
NFS	
SMB/CIFS	
GlusterFS	
iSCSI	
CephFS	
RBD	
ZFS over iSCSI	
ZFS	
Proxmox Backup Server	

4. Enter what it wants as a new unique ID value for the new storage. Then set the “Directory” field to the folder mount point you created earlier (e.g., `/mnt/vmstorage`). Finally, you just need to create and select the content types like “Disk image,” “Container template,” or “Backup.” Click “Add” to save the configuration.

The screenshot shows the Proxmox Virtual Environment 7.4-3 interface. The left sidebar has 'Server View' selected, with 'Datacenter' expanded. Under 'Datacenter', there is a node named 'pmox01' with two storage volumes: 'local (pmox01)' and 'local-lvm (pmox01)'. The 'Storage' option in the sidebar is highlighted. In the main content area, the 'Datacenter' tab is selected. At the top, there are 'Add', 'Remove', and 'Edit' buttons. Below them is a table with columns 'ID ↑', 'Type', and 'Content'. Two entries are listed: 'local' (Type: Directory, Content: VZDump backup file, ISO image, Container template) and 'local-lvm' (Type: LVM-Thin, Content: Disk image, Container). A modal dialog titled 'Add: Directory' is open. It has two tabs: 'General' (selected) and 'Backup Retention'. The 'General' tab contains fields: 'ID' (set to 'vmstorage'), 'Directory' (set to '/mnt/vmstorage'), 'Nodes' (set to 'All (No restrictions)'), 'Enable' (checkbox checked), 'Content' (dropdown set to 'Disk image, ISO image, ...'), and 'Shared' (checkbox unchecked). There are 'Help' and 'Advanced' buttons at the bottom, and a blue 'Add' button. A red arrow points from the text above to the 'ID' field in the dialog.

5. The new storage space will now be available in the Proxmox web interface for virtual machines and containers.

PROXMOX Virtual Environment 7.4-3 Search

Documentation Create VM Create CT root@pam Help

Server View Datacenter

Datacenter

- pmox01
 - local (pmox01)
 - local-lvm (pmox01)
 - vmstorage (pmox01)

Add Remove Edit

ID ↑	Type	Content	Path/Target	Shared	Enab...	Bandwidth Limit
local	Direc...	VZDump backup file, ISO i...	/var/lib/vz	No	Yes	
local-lvm	LVM...	Disk image, Container		No	Yes	
vmstorage	Direc...	Disk image, ISO image, Co...	/mnt/vmstorage	No	Yes	

Storage

Backup Replication Permissions

- Users
- API Tokens
- Two Factor
- Groups
- Pools
- Roles
- Realms

HA ACME Firewall Metric Server Support

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Apr 10 16:04:19	Apr 10 16:16:50	pmox01	root@pam	Shell	OK
Apr 10 15:46:42	Apr 10 15:46:42	pmox01	root@pam	Start all VMs and Containers	OK

Now, when you create a new virtual machine or container you will see the storage available to select in the storage drop down.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

Use CD/DVD disc image file (iso)

Guest OS:

Storage: local

Type:

Linux

ISO image: Name ↑

Type

Avail

Capacity

Use physical C:

local

64.20 GB

71.25 GB

Do not use any

vmstorage

dir

933.32 GB

983.35 GB



Advanced

Back

Next

Wrapping up

Combining NVMe storage with Proxmox will make for a very fast virtualization environment. As we have seen, the process to add an NVMe drive to your Proxmox VE host isn't too difficult. We just need to run a few commands from the Linux Bash command shell and we are good to.

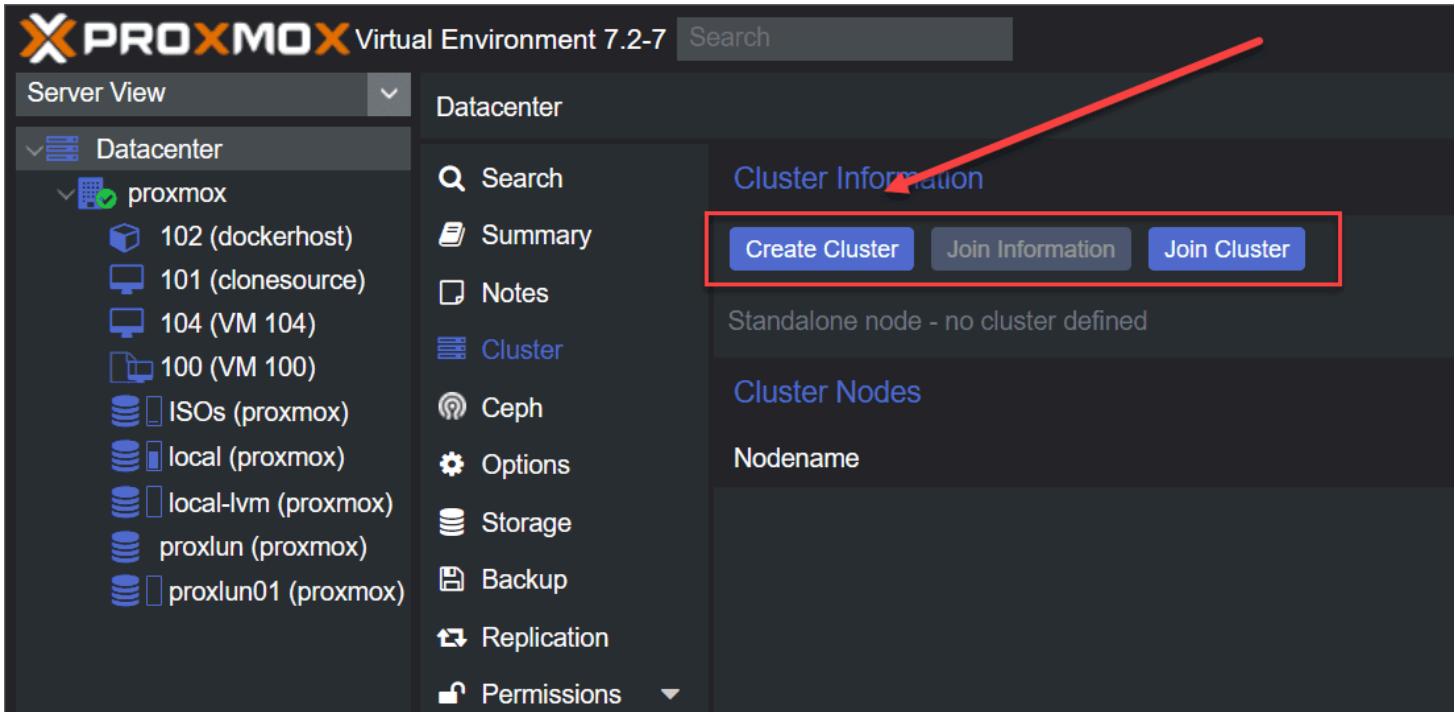
Proxmox cluster installation and configuration

After running a Proxmox single node server in your home lab or development environment, you probably want to graduate to using multiple nodes for high-availability. Creating a Proxmox cluster is a great way to have HA and have the option to move workloads around for maintenance and failover. Let's look at Proxmox cluster installation and configuration.

What is Proxmox Cluster?

A Proxmox cluster is multiple Proxmox hosts that work together as a single logical unit. They usually have some type of shared storage between them. This way, if a host fails, the other hosts can run the VMs and containers the failed host was running and restart them on a healthy host.

The Proxmox cluster uses hosts loaded and installed with Proxmox Virtual Environment (VE). After you install one Proxmox VE host, you can navigate to your datacenter settings > **Cluster > Create Cluster**.



Proxmox cluster benefits

A Proxmox Cluster provides many benefits, including

- high availability of VMs and LXC containers
- Automated failover
- Making sure VMs are always available
- Bolstering maintenance operations making it easy to keep hosts updated or taking them down for maintenance.

Minimum Requirements for Building a Proxmox Cluster

There are requirements for a Proxmox cluster like there are for an ESXi cluster, Hyper-V cluster, etc. To create a Proxmox VE cluster, you need to have at least 2 Proxmox nodes for the cluster. Like other cluster configurations, it is best practice to have identical hardware for the cluster to make sure you have compatibility between them and everything is uniform and consistent when it comes to resources and running VMs.

The nodes communicate back and forth to carry out management tasks and make sure everything is up and running and reachable.

As mentioned, shared storage is really a requirement for HA and other production workloads or in the home lab. With shared storage, it means the storage or data doesn't really move if a host fails. The healthy hosts just spin up the failed VM

or containers from the failed host and bring them back online.

Datacenter			
	Add	Remove	Edit
	ID ↑	Type	Content
ISOs	Directory	VZDump backup file, Disk image, ISO ...	
local	Directory	VZDump backup file, ISO image, Cont...	
local-lvm	LVM-Thin	Disk image, Container	
proxlun	iSCSI	Disk image	
proxlun01	LVM	Disk image, Container	

Firewall Requirements

There are also firewall requirements. The cluster uses TCP ports to communicate between nodes. So, you want to make sure the ports are open on the firewall or any firewalls in between.

The screenshot shows the Proxmox Virtual Environment 7.2-7 interface. On the left, there's a tree view of the server configuration. The main area is titled "Datacenter". At the top right of the main area, there are buttons for "Add", "Copy", "Insert: Security Group", "Remove", and "Edit". Below these are filters for "On", "Type", "Action", "Macro", "Interface", and "Protocol". The main content area lists various cluster management options: Search, Summary, Notes, Cluster, Ceph, Options, Storage, Backup, Replication, Permissions (with sub-options for Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms), HA, ACME, Firewall (highlighted by a red arrow), Options, and Security Group.

IP addresses

As with networking best practices and functionality, your Proxmox nodes each have their own unique IP address. Also, with a cluster configuration, there will be other IP addresses these may have, including for cluster communication, and for things like running Ceph storage.

Node 'proxmox'								Reboot	Shutdown	Shell	Bulk Actions
	Name ↑	Type	Active	Autostart	VLAN a ..	Ports/Slaves	Bond Mode	CIDR	Gateway		
Search	ens160	Network Device	Yes	Yes	No			10.3.33.74/24			
Summary	ens192	Network Device	Yes	Yes	No						
Notes	ens32	Network Device	Yes	No	No						
Shell	vmbr0	Linux Bridge	Yes	Yes	Yes	ens32		10.1.149.74/24	10.1.149.1		
System	vmbr1	Linux Bridge	Yes	Yes	No	ens192		172.16.16.254/24			
Network											

Config files and Corosync

The [configuration files for a Proxmox Cluster](#) can be stored on either on local storage that is directly attached to a Proxmox node or shared storage accessible from multiple nodes.

Just make sure to choose the right type of storage for your needs, hardware requirements, and other factors that are going to be unique in each environment.

The **corosync** communication protocol is the protocol that manages Proxmox cluster communication between nodes in a [Proxmox Cluster](#). It is responsible for making sure the nodes in the cluster can communicate with each other. It is also responsible for the transfer of information and data between nodes.

Failure, Cluster Manager, and Node types

Let's talk about node failures. When a node fails, the remaining healthy node or nodes will continue to work and operate normally in the cluster. These other nodes will be the target of failover operations for high-availability so that your VMs and LXC containers are restarted over on healthy hosts.

The corosync cluster manager is responsible for automatically failing over to the remaining nodes in the event of a failure.

The **cluster manager** is the node that is responsible for performing these management tasks. These include live migrations of VMs and automated failover operations. It is one of the important roles that is found inside the cluster management plane that handles events and coordinates during failures.

In a Proxmox Cluster, there are two types of nodes: the **main node** and the **slave node** or second node.

- The main node is responsible for performing management tasks
- the slave node is responsible for running virtual machines.

In the event of a failure of the main node, the slave node will take over and perform management tasks until the main node is restored.

Can you run a single node?

Yes. A single-node cluster is a [Proxmox](#) cluster that has only one node and is used in smaller environments for testing or home labs. It is not used for production.

You can setup a single node cluster in Proxmox. You will need to have [install Proxmox](#) on a single node and configure the network settings. Once Proxmox is installed, you can create a new single-node cluster using the Proxmox Web GUI or the command line.

Home Lab Environments

A home lab environment has a small number of physical servers. Most run often only one or two, and is used for testing and learning purposes.

Using a Proxmox cluster in your home lab can provide the same benefits of high availability and easy migration of virtual machines. This doesn't have to be a large cluster, just a cluster of two nodes.

Start Time	End Time	Node	User name	Description	Status
Feb 12 21:22:09	Feb 12 21:22:10	proxmox	root@pam	VM 104 - Stop	OK
Feb 12 21:20:53	Feb 12 21:22:09	proxmox	root@pam	VM/CT 104 - Console	OK
Feb 12 21:20:47	Feb 12 21:20:48	proxmox	root@pam	VM 104 - Start	OK
Feb 12 21:18:16	Feb 12 21:18:18	proxmox	root@pam	Create Cluster	OK
Feb 12 03:47:05	Feb 12 03:47:10	proxmox	root@pam	Update package database	OK
Feb 11 04:45:36	Feb 11 04:46:11	proxmox	root@pam	Update package database	OK

Clusters introduce some complexity in the lab environment, such as more network requirements and properly configuring the firewall. However, the extra complexity is usually worth it for a good learning environment that is stable and one you can easily move things around.

Setting up a cluster itself is also a great way to learn about virtual environments and to gain hands-on experience with Proxmox. Everything you configure in a lab is nothing more than real world experience that you can take into production environments.

Create a Proxmox cluster

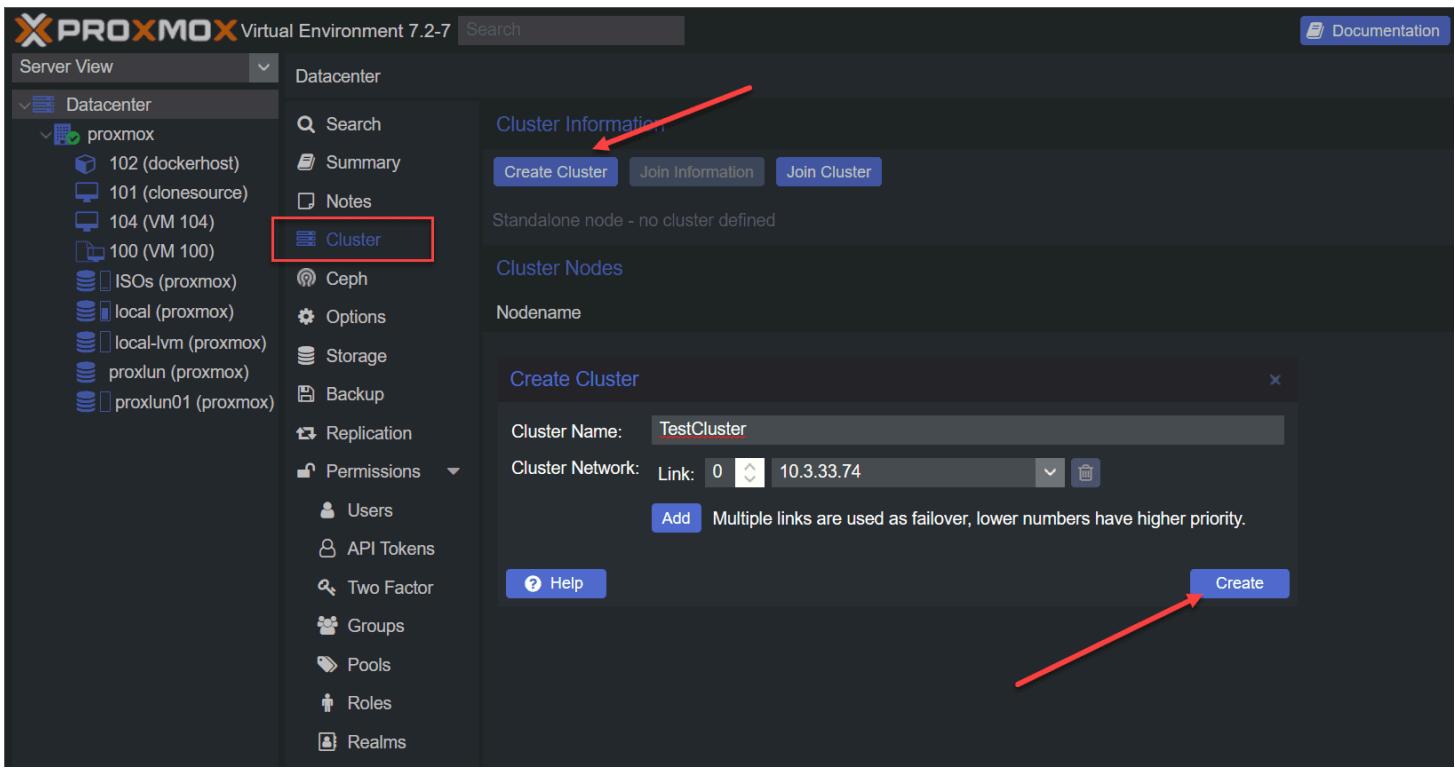
Step 1: Log in to the Proxmox Web GUI

To create a Proxmox Cluster using the Proxmox Web GUI, you will need to log in to the Proxmox Web GUI on one of the nodes in the cluster.

The Proxmox Web GUI can be accessed by navigating to <https://<node-ip-address>:8006> in a web browser.

Step 2: Create a new cluster

To create a new cluster, click on the “Cluster” tab in the Proxmox Web GUI and then click on the “Create Cluster” button. This will open a dialog where you can enter the name of the new cluster.

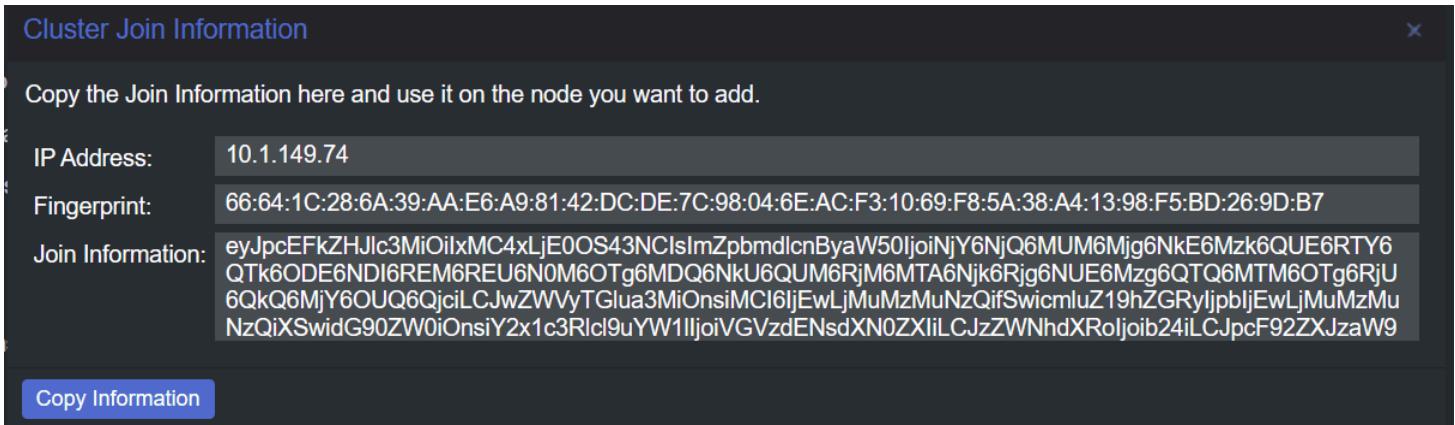


You can also create this from the command line:

```
pvecm create <cluster-name>
```

Step 3: Add nodes to the cluster

Once the new cluster has been created, you can add additional nodes to the cluster. To do this, click on the “Cluster” tab in the Proxmox Web GUI and then click on the “Add Node” button. This will open a dialog where you can enter the node’s IP address you want to add to the cluster.



You will use this join information to join cluster on the second, and third node.

You can also do this from the command line:

```
pvecm join <ip-address-of-the-main-node>
```

Step 4: Configure the corosync communication protocol

To configure the corosync communication protocol, click on the “Cluster” tab in the Proxmox Web GUI and then click on the “Edit” button next to the cluster you want to configure.

This will open a dialog where you can modify the settings for the corosync communication protocol, including the communication port and the number of votes required to reach quorum.

Task viewer: Create Cluster

Output

Status

Stop

Writing corosync config to /etc/pve/corosync.conf

Restart corosync and cluster filesystem

TASK OK

Step 5: Add virtual machines to the cluster

Once the cluster has been configured, you can add virtual machines to the cluster. To do this, click on the “Virtual Machines” tab in the Proxmox Web GUI and then click on the “Create VM” button.

This will open a dialog where you can create and configure virtual machines, including specifying the virtual machine name, the operating system, and the storage location.

Step 6: Monitor the cluster

This can be done using the Proxmox Web GUI by clicking on the “Cluster” tab and then clicking on the “Monitor” button. Make sure to keep updates as part of your regular schedule of lifecycle updates.

Cluster cold start

Cluster cold start refers to the process of starting a Proxmox Cluster from scratch, without any configuration or state information. A cluster cold start usually happens in the following

1. **After a complete failure of the cluster:** In the event of a complete failure of the cluster, all configuration information and state information are lost, and a cluster cold start is necessary to rebuild the cluster from scratch.
2. **When setting up a new Proxmox Cluster:** When setting up a new Proxmox Cluster, a cluster cold start is necessary to create a new cluster and configure the cluster from scratch.
3. **When changing the cluster configuration:** When changing the configuration of an existing Proxmox Cluster, such as adding or removing nodes, a cluster cold start may be necessary to properly reconfigure the cluster.

A cluster cold start in Proxmox Clusters involves installing Proxmox on each node, configuring the network settings, creating a new cluster, adding nodes to the cluster, and configuring the corosync communication protocol. This process can be performed using the Proxmox Web GUI or by using the command line.

It is important to note that a cluster cold start can result in data loss, as all virtual machines and configurations will need to be recreated. As such, it is important to plan properly and back up all virtual machines and configurations prior to performing a cluster cold start.

Wrapping Up

Proxmox is a great platform for running home lab workloads and production environments. With Proxmox clusters, you can set up a high-availability environment to protect your virtual machines from a single node failure in the data center.

If you follow all the steps listed to create a Proxmox cluster, you can easily create a Proxmox cluster using the web UI and CLI.

Mastering Ceph Storage Configuration in Proxmox 8 Cluster

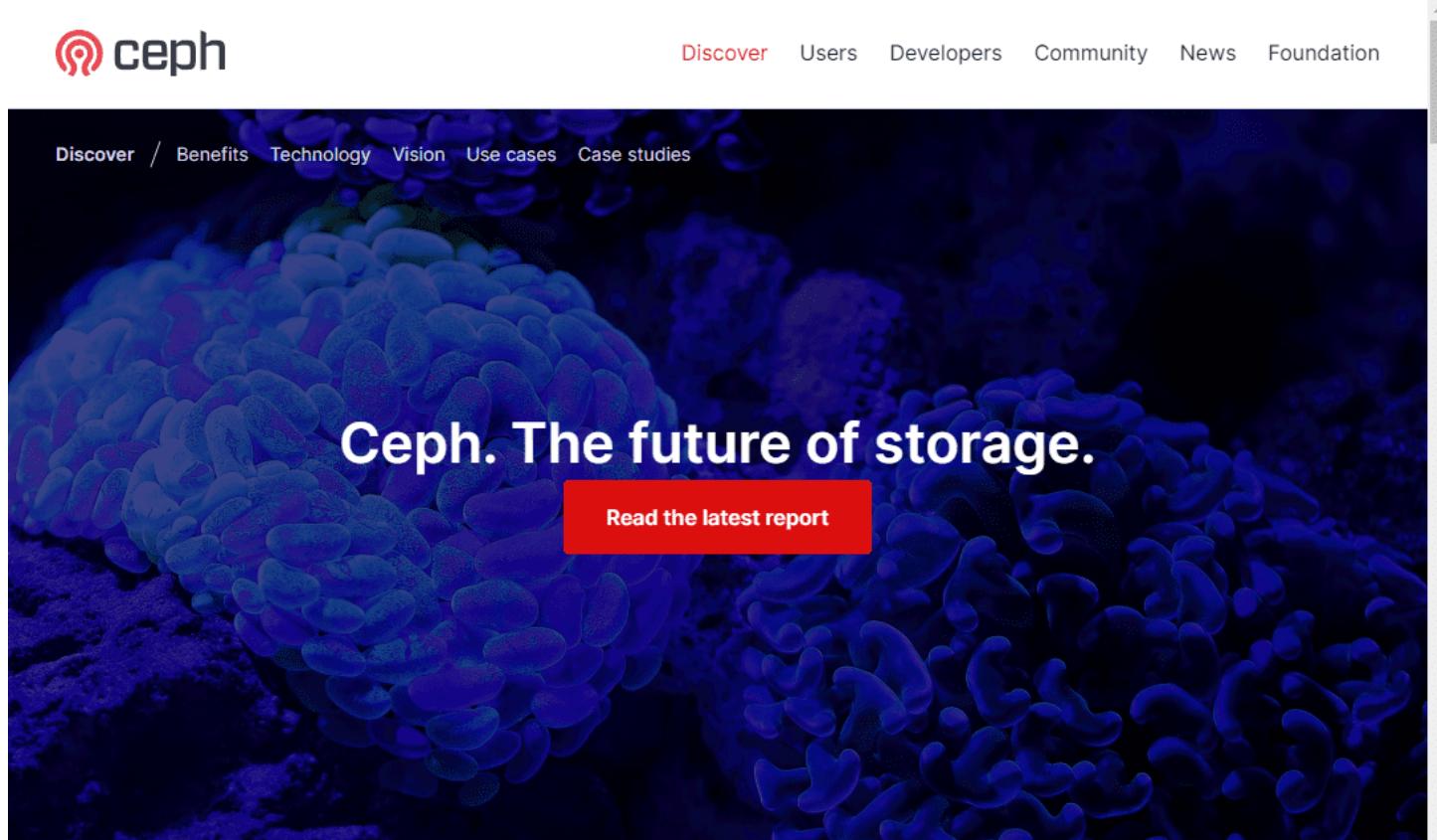
Ceph [Storage](#) is an excellent storage platform because it's designed to run on commodity hardware, providing an enterprise-level deployment experience that's both cost-effective and highly reliable. Let's look at mastering Ceph Storage [configuration in Proxmox](#) 8 Cluster.

Table of contents

- [What is Ceph Storage?](#)
- [Install and configure Ceph in Proxmox](#)
 - [Setting up Ceph OSD Daemons and Ceph Monitors](#)
 - [Creating Ceph Monitors](#)
- [Creating a Ceph Pool for VM and Container storage](#)
- [Utilizing Ceph storage for Virtual Machines and Containers](#)
- [Wrapping up](#)
- [Other links you may like](#)

What is Ceph Storage?

Ceph Storage is an open-source solution designed to provide object storage devices, block devices, and file storage within the same cluster. A key characteristic of Ceph storage is its intelligent data placement method. An algorithm called CRUSH (Controlled Replication Under Scalable Hashing) decides where to store and how to retrieve data, avoiding any single point of failure and effectively providing fault-tolerant storage



Ceph is the future of storage; where traditional systems fail to deliver, Ceph is designed to

excel. Leverage your data for better business decisions and achieve operational

Ceph Storage is an open source and robust storage solution

Install and configure Ceph in Proxmox

Start by installing the Ceph packages in your Proxmox environment. These packages include essential Ceph components like Ceph OSD daemons, Ceph Monitors (Ceph Mon), and Ceph Managers (Ceph Mgr).

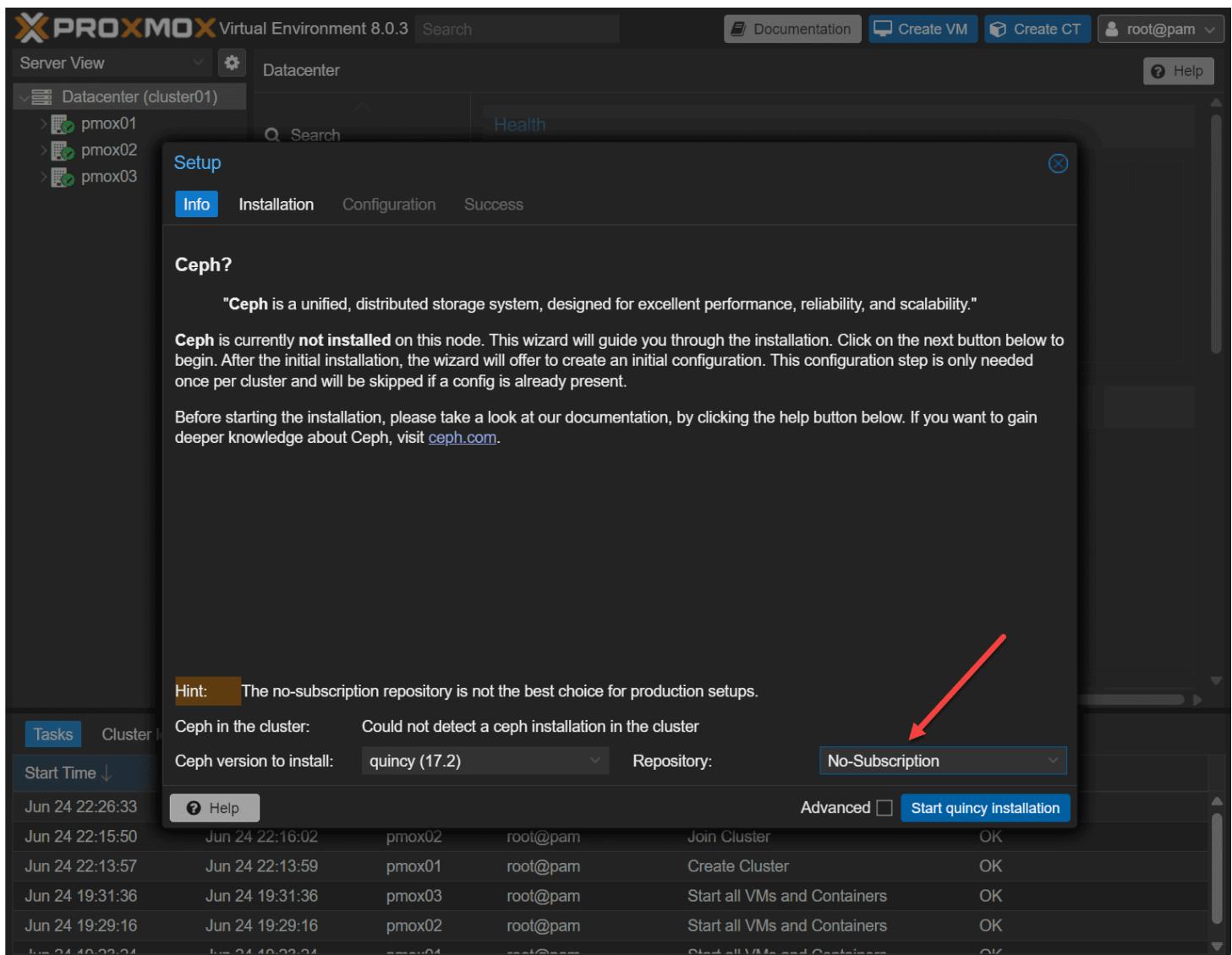
Click on one of your Proxmox nodes, and navigate to **Ceph**. When you click Ceph, it will prompt you to install Ceph.

The screenshot shows the Proxmox VE 8.0.3 interface. In the left sidebar under 'Datacenter', 'Ceph' is selected. A modal dialog box is open over the main content area, asking 'Ceph is not installed on this node. Would you like to install it now?'. Below the question is a blue 'Install Ceph' button. A red arrow points from the text 'Install Ceph on each Proxmox 8 cluster node' at the bottom of the page to this button. At the bottom of the main window, there is a table titled 'Tasks' showing recent system events.

Start Time	End Time	Node	User name	Description	Status
Jun 24 22:26:33	Jun 24 22:26:46	pmpox03	root@pam	Join Cluster	OK
Jun 24 22:15:50	Jun 24 22:16:02	pmpox02	root@pam	Join Cluster	OK
Jun 24 22:13:57	Jun 24 22:13:59	pmpox01	root@pam	Create Cluster	OK
Jun 24 19:31:36	Jun 24 19:31:36	pmpox03	root@pam	Start all VMs and Containers	OK
Jun 24 19:29:16	Jun 24 19:29:16	pmpox02	root@pam	Start all VMs and Containers	OK
Jun 24 19:29:24	Jun 24 19:29:24	-----	-----	Start all VMs and Containers	OK

Install Ceph on each Proxmox 8 cluster node

This begins the setup wizard. First, you will want to choose your **Repository**. This is especially important if you don't have a subscription. You will want to choose the **No Subscription** option. For production environments, you will want to use the **Enterprise** repository.



Choosing the Ceph repository and beginning the installation

You will be asked if you want to continue the installation of Ceph. Type **Y** to continue.

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam Help

Server View Datacenter Health

Setup

Info Installation Configuration Success

```

python-cryptography-doc python3-cryptography-vectors python-mako-doc
python3-beaker python-natsort-doc python-openssl-doc python3-openssl-dbg
libapache2-mod-python python-pecan-doc python-waitress-doc python-webob-doc
python-webtest-doc ipython3 python-werkzeug-doc python3-lxml python3-watchdog
Recommended packages:
btrfs-tools python3-lxml python3-routes python3-simplejson python3-pastescript
python3-pyinotify
The following NEW packages will be installed:
ceph ceph-base ceph-mds ceph-mgr ceph-mgr-modules-core ceph-mon ceph-osd
ceph-volume cryptsetup-bin libnvme1 libparted2 libpython3.11
libsqllite3-mod-ceph nvme-cli parted python3-autocommand python3-bcrypt
python3-bs4 python3-cffi-backend python3-cheroot python3-cherrypy3
python3-cryptography python3-dateutil python3-inflct python3-jaraco.classes
python3-jaraco.collections python3-jaraco.context python3-jaraco.functools
python3-jaraco.text python3-logutils python3-mako python3-markupsafe
python3-more-itertools python3-natsort python3-openssl python3-paste
python3-pastedeploy python3-pastedeploy-tpl python3-pecan python3-portend
python3-simplegeneric python3-singledispatch python3-soupsieve python3-tempita
python3-tempora python3-tz python3-waitress python3-webob python3-webtest
python3-werkzeug python3-zc.lockfile sudo uuid-runtime
0 upgraded, 53 newly installed, 0 to remove and 0 not upgraded.
Need to get 54.6 MB of archives.
After this operation, 252 MB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

Tasks Cluster Start Time Advanced Next

Start Time					
Jun 24 22:29:46					
Jun 24 22:26:33	Jun 24 22:26:46	pmox03	root@pam	Join Cluster	OK
Jun 24 22:15:50	Jun 24 22:16:02	pmox02	root@pam	Join Cluster	OK
Jun 24 22:13:57	Jun 24 22:13:59	pmox01	root@pam	Create Cluster	OK
Jun 24 19:31:36	Jun 24 19:31:36	pmox03	root@pam	Start all VMs and Containers	OK
Jun 24 19:30:46	Jun 24 19:30:46	pmox02	root@pam	Start all VMs and Containers	OK

Verify the installation of Ceph storage modules

The screenshot shows the Proxmox VE 8.0.3 interface. In the top navigation bar, there are links for Documentation, Create VM, Create CT, and a user session for root@pam. Below the navigation bar, the main interface shows a tree view of Datacenter (cluster01) containing nodes pmox01, pmox02, and pmox03. A search bar and a Health tab are also present.

A modal window titled "Setup" is open, specifically the "Installation" tab. It displays a terminal log of the Ceph installation process:

```
Setting up ceph-osd (17.2.6-pve1+3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ceph-osd.target -> /lib/systemd/system/ceph-osd.target.
Created symlink /etc/systemd/system/ceph.target.wants/ceph-osd.target -> /lib/systemd/system/ceph-osd.target.
Setting up ceph-mon (17.2.6-pve1+3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mon.target -> /lib/systemd/system/ceph-mon.target.
Created symlink /etc/systemd/system/ceph.target.wants/ceph-mon.target -> /lib/systemd/system/ceph-mon.target.
Setting up ceph-mgr (17.2.6-pve1+3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mgr.target -> /lib/systemd/system/ceph-mgr.target.
Created symlink /etc/systemd/system/ceph.target.wants/ceph-mgr.target -> /lib/systemd/system/ceph-mgr.target.
Setting up ceph-volume (17.2.6-pve1+3) ...
Setting up ceph (17.2.6-pve1+3) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9) ...

installed ceph quincy successfully! ←
```

Below the terminal log, it says "reloading API to load new Ceph RADOS library..." followed by a progress bar.

At the bottom of the modal, there are "Advanced" and "Next" buttons. The "Tasks" tab in the main interface shows a list of recent tasks:

Start Time	End Time	User	Action	Status
Jun 24 22:29:46				
Jun 24 22:26:33	Jun 24 22:26:46	pmox03	root@pam	Join Cluster
Jun 24 22:15:50	Jun 24 22:16:02	pmox02	root@pam	Join Cluster
Jun 24 22:13:57	Jun 24 22:13:59	pmox01	root@pam	Create Cluster
Jun 24 19:31:36	Jun 24 19:31:36	pmox03	root@pam	Start all VMs and Containers
Jun 24 19:29:46	Jun 24 19:29:46	pmox02	root@pam	Start all VMs and Containers

Ceph installed successfully

Next, you will need to choose the **Public Network** and the **Cluster Network**. Here, I don't have dedicated networks configured since this is a nested installation. So I am just choose the same subnet for each.

The screenshot shows the Proxmox VE 8.0.3 setup interface. In the top left, the navigation bar includes "Server View", "Documentation", "Create VM", "Create CT", and a user session "root@pam". The main area is titled "Datacenter" with tabs for "Search" and "Health". A modal window titled "Setup" is open, specifically the "Configuration" tab. It displays "Ceph cluster configuration:" with fields for "Public Network IP/CIDR" set to "10.1.149.120/24" and "Cluster Network IP/CIDR" set to "10.1.149.120/24". To the right, it shows "First Ceph monitor:" with "Monitor node: pmox01" and a note: "Additional monitors are recommended. They can be created at any time in the Monitor tab." Below the modal, the "Tasks" tab is selected in the navigation bar, showing a list of recent tasks:

Start Time					
Jun 24 22:29:46	Jun 24 22:26:46	pmox03	root@pam	Join Cluster	OK
Jun 24 22:26:33	Jun 24 22:16:02	pmox02	root@pam	Join Cluster	OK
Jun 24 22:15:50	Jun 24 22:13:59	pmox01	root@pam	Create Cluster	OK
Jun 24 22:13:57	Jun 24 19:31:36	pmox03	root@pam	Start all VMs and Containers	OK
Jun 24 19:31:36	Jun 24 19:31:36	pmox03	root@pam	Start all VMs and Containers	OK
Jun 24 19:30:46	Jun 24 19:30:46	pmox03	root@pam	Start all VMs and Containers	OK

Configuring the public and cluster networks

If you click the **Advanced** checkbox, you will be able to setup the **Number of replicas** and **Minimum replicas**.

The screenshot shows the Proxmox VE 8.0.3 setup interface for creating a Ceph cluster. In the 'Configuration' tab, the 'Ceph cluster configuration:' section is displayed. It includes fields for 'Public Network IP/CIDR' (10.1.149.120/24), 'Cluster Network IP/CIDR' (10.1.149.120/24), 'Number of replicas' (3), and 'Minimum replicas' (2). To the right, the 'First Ceph monitor:' section shows 'Monitor node: pmox01' and a note: 'Additional monitors are recommended. They can be created at any time in the Monitor tab.' At the bottom right of the configuration tab, there is an 'Advanced' checkbox and a 'Next' button. A red arrow points from the 'Advanced' checkbox to the text 'Advanced configuration including the number of replicas' located below the configuration tabs.

Tasks Cluster

Start Time ↓

Jun 24 22:29:46	Jun 24 22:26:46	pmox03	root@pam	Join Cluster	OK
Jun 24 22:26:33	Jun 24 22:16:02	pmox02	root@pam	Join Cluster	OK
Jun 24 22:15:50	Jun 24 22:13:59	pmox01	root@pam	Create Cluster	OK
Jun 24 22:13:57	Jun 24 19:31:36	pmox03	root@pam	Start all VMs and Containers	OK
Jun 24 19:31:36	Jun 24 19:31:36	pmox03	root@pam	Start all VMs and Containers	OK

?

Help

Documentation Create VM Create CT root@pam

Server View Datacenter Health

Setup Info Installation Configuration Success

Ceph cluster configuration:

First Ceph monitor:

Monitor node: pmox01

Additional monitors are recommended. They can be created at any time in the Monitor tab.

Public Network IP/CIDR: 10.1.149.120/24

Cluster Network IP/CIDR: 10.1.149.120/24

Number of replicas: 3

Minimum replicas: 2

Advanced Next

Advanced configuration including the number of replicas

At this point, Ceph has been successfully installed on the Proxmox node.

The screenshot shows the Proxmox VE 8.0.3 interface. In the top left, the navigation bar includes "Server View", "Documentation", "Create VM", "Create CT", and a user session "root@pam". The main area is titled "Datacenter" with tabs for "Search" and "Health". A modal window titled "Setup" is open, showing the "Success" tab. It displays the message "Installation successful!" and instructions: "The basic installation and configuration is complete. Depending on your setup, some of the following steps are required to start using Ceph: 1. Install Ceph on other nodes 2. Create additional Ceph Monitors 3. Create Ceph OSDs 4. Create Ceph Pools". Below the modal, a "Tasks" section lists recent operations: "Jun 24 22:32:29 Jun 24 22:32:29 pmox01 root@pam Ceph Monitor mon.pmx01 - Create OK", "Jun 24 22:29:46 Jun 24 22:30:32 pmox01 root@pam Shell OK", "Jun 24 22:26:33 Jun 24 22:26:46 pmox03 root@pam Join Cluster OK", "Jun 24 22:15:50 Jun 24 22:16:02 pmox02 root@pam Join Cluster OK", and "Jun 24 22:12:57 Jun 24 22:12:59 pmox01 root@pam Create Cluster OK". A "Help" button is visible in the bottom right of the tasks table.

Ceph configured successfully and additional setup steps needed

Repeat these steps on the remaining [cluster nodes in your Proxmox](#) cluster configuration.

Setting up Ceph OSD Daemons and Ceph Monitors

Ceph OSD Daemons and Ceph Monitors are crucial to the operation of your Ceph storage cluster. The OSD daemons handle data storage, retrieval, and replication on the storage devices, while Ceph Monitors maintain the cluster map, tracking active and failed cluster nodes.

You'll need to assign several Ceph OSDs to handle data storage and maintain the redundancy of your data.

PROXMOX Virtual Environment 8.0.3 Search Documentation Create VM Create CT root@pam

Server View Node 'pmox01'

1 Datacenter (cluster01)
pmox01
pmox02
pmox03

Updates
Repositories
Firewall
Disks
LVM
LVM-Thin
Directory
ZFS
Ceph
Configuration
Monitor
OSD
CephFS
Pools
Log
Replication

2 Create: OSD
Manage Global Flags
No OSD selected
Details Start Stop

Create: Ceph OSD

Disk: /dev/sdb (highlighted with red box)

DB Disk: use OSD disk
DB size (GiB): Automatic

Encrypt OSD:
Device Class: auto detect
WAL Disk: use OSD/DB disk
WAL size (GiB): Automatic

Note: Ceph is not compatible with disks backed by a hardware RAID controller. For details see [the reference documentation](#).

Help Advanced Create 3

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jun 24 22:34:23	Jun 24 22:34:58	pmox03	root@pam	Shell	OK
Jun 24 22:33:59	Jun 24 22:34:53	pmox02	root@pam	Shell	OK
Jun 24 22:32:29	Jun 24 22:32:36	pmox01	root@pam	Ceph Manager mgr.pmx01 - Create	OK
Jun 24 22:32:27	Jun 24 22:32:29	pmox01	root@pam	Ceph Monitor mon.pmx01 - Create	OK
Jun 24 22:29:46	Jun 24 22:30:32	pmox01	root@pam	Shell	OK
Jun 24 22:26:33	Jun 24 22:26:46	pmox02	root@pam	Join Cluster	OK

Adding an OSD in Proxmox Ceph storage

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam

Server View Node 'pmox01'

Updates Reload Create: OSD Manage Global Flags No OSD selected Details Start Stop

Datacenter (cluster01) pmox01 pmox02 pmox03

Updates Repositories Firewall Disks LVM LVM-Thin Directory ZFS Ceph Configuration Monitor OSD CephFS Pools Log Replication

Task: Ceph OSD sdb - Create running... Details

Name	Class	OSD Type	Status	Version	weight
default					

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jun 24 22:42:50		pmox01	root@pam	Ceph OSD sdb - Create	
Jun 24 22:34:23	Jun 24 22:34:58	pmox03	root@pam	Shell	OK
Jun 24 22:33:59	Jun 24 22:34:53	pmox02	root@pam	Shell	OK
Jun 24 22:32:29	Jun 24 22:32:36	pmox01	root@pam	Ceph Manager mgr.pmx01 - Create	OK
Jun 24 22:32:27	Jun 24 22:32:29	pmox01	root@pam	Ceph Monitor mon.pmx01 - Create	OK
Jun 24 22:32:16	Jun 24 22:32:22	pmox01	root@pam	Shell	OK

The OSD begins configuring and adding

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam

Server View Node 'pmox01'

Updates Reload Create: OSD Manage Global Flags No OSD selected Details Start Stop

Repositories Firewall Disks LVM LVM-Thin Directory ZFS Ceph Configuration Monitor OSD CephFS Pools Log Replication

Name Class OSD Type Status Version weight

default pmox01 osd.0 ssd bluestore up / in 17.2.6 17.2.6 0.0488

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jun 24 22:42:50	Jun 24 22:42:56	pmox01	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:34:23	Jun 24 22:34:58	pmox03	root@pam	Shell	OK
Jun 24 22:33:59	Jun 24 22:34:53	pmox02	root@pam	Shell	OK
Jun 24 22:32:29	Jun 24 22:32:36	pmox01	root@pam	Ceph Manager mgr.pmx01 - Create	OK
Jun 24 22:32:27	Jun 24 22:32:29	pmox01	root@pam	Ceph Monitor mon.pmx01 - Create	OK
Jun 24 22:32:16	Jun 24 22:32:22	pmox01	root@pam	Shell	OK

The OSD is successfully added to the Proxmox host

Also, set up more than one Ceph Monitor to ensure high availability and fault tolerance.

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam

Server View Node 'pmox03'

Repositories Reload Create: OSD Manage Global Flags No OSD selected Details Start Stop

Firewall Disks LVM LVM-Thin Directory ZFS Ceph Configuration Monitor OSD CephFS Pools Log Replication Task History

Name	Class	OSD Type	Status	Version	weight
default					
pmox03	ssd	bluestore	up / in	17.2.6	0.0488
pmox02	ssd	bluestore	up / in	17.2.6	0.0488
pmox01	ssd	bluestore	up / in	17.2.6	0.0488
osd.2					
osd.1					
osd.0					

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jun 24 22:45:48	Jun 24 22:45:54	pmox03	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:45:05	Jun 24 22:45:11	pmox02	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:42:50	Jun 24 22:42:56	pmox01	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:34:23	Jun 24 22:34:58	pmox03	root@pam	Shell	OK
Jun 24 22:33:59	Jun 24 22:34:53	pmox02	root@pam	Shell	OK
Jun 24 22:32:30	Jun 24 22:32:36	pmox01	root@pam	Ceph Manager Backend - Create	OK

OSDs added to all three Proxmox nodes

At this point, if we visit the Ceph storage dashboard , we will see the status of the Ceph storage cluster.

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam Help

Server View Datacenter

Datacenter (cluster01)

- pmox01
- pmox02
- pmox03

Ceph

Search Summary Notes Cluster Options Storage Backup Replication Permissions

Users API Tokens Two Factor Groups Pools Roles

Health Status

Severity Summary

No Warnings/Errors

HEALTH_OK Ceph Version: 17.2.6

Status OSDs PGs

	Up	In	Out	Down	Total	active+clean:
Up	3	0	0	0	3	1
Down	0	0	0	0	0	0
Total: 3						

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jun 24 22:45:48	Jun 24 22:45:54	pmox03	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:45:05	Jun 24 22:45:11	pmox02	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:42:50	Jun 24 22:42:56	pmox01	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:34:23	Jun 24 22:34:58	pmox03	root@pam	Shell	OK
Jun 24 22:33:59	Jun 24 22:34:53	pmox02	root@pam	Shell	OK
Jun 24 22:32:20	Jun 24 22:32:26	pmox01	root@pam	Ceph Manager mon01 - Create	OK

Healthy Ceph storage status for the cluster

Creating Ceph Monitors

Let's add additional Ceph Monitors, as we have only configured the first node as a Ceph monitor. What is a Ceph Monitor?

A Ceph Monitor, often abbreviated as Ceph Mon, is the part that maintains and manages the cluster map, a crucial data structure that keeps track of the entire cluster's state, including the location of data, the cluster topology, and the status of other daemons in the system.

Here we are adding the 2nd Proxmox node as a monitor. I added the 3rd one as well.

The screenshot shows the Proxmox VE 8.0.3 interface. In the left sidebar under 'Datacenter (cluster01)', the nodes pmox01, pmox02, and pmox03 are listed. The node pmox02 is selected. The main panel shows the 'Monitor' section with a table listing one monitor named 'mon....'. A red circle labeled '1' highlights the 'Monitors' icon in the sidebar. A red arrow labeled '2' points to the 'Create' button in the top right of the monitor table. A red circle labeled '3' highlights the 'Create' button in the 'Create: Monitor' dialog.

Name	Host	Status	Address	Version	Quorum
mon....	pmo...	running	10.1.149.120:6789/0	17.2.6	Yes

Create: Monitor

Host: pmox02

Create 3

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jun 24 22:45:48	Jun 24 22:45:54	pmox03	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:45:05	Jun 24 22:45:11	pmox02	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:42:50	Jun 24 22:42:56	pmox01	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:34:23	Jun 24 22:34:58	pmox03	root@pam	Shell	OK
Jun 24 22:33:59	Jun 24 22:34:53	pmox02	root@pam	Shell	OK
Jun 24 22:32:30	Jun 24 22:32:36	pmox01	root@pam	Ceph Manager mon - Create	OK

Adding Ceph Monitors to additional Proxmox hosts

Now, each node is a monitor.

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam

Server View Node 'pmox03'

Monitor

Name ↑	Host	Status	Address	Version	Quorum
mon...	pmox01	running	10.1.149.120:6789/0	17.2.6	Yes
mon...	pmox02	running	10.1.149.121:6789/0	17.2.6	Yes
mon...	pmox03	running	10.1.149.122:6789/0	17.2.6	Yes

Manager

Name ↑	Host	Status	Address	Version
mgr.p...	pmox01	active	10.1.149.120	17.2.6

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jun 24 22:48:28	Jun 24 22:48:28	pmox03	root@pam	Ceph Monitor mon.pmx03 - Create	OK
Jun 24 22:48:14	Jun 24 22:48:15	pmox02	root@pam	Ceph Monitor mon.pmx02 - Create	OK
Jun 24 22:45:48	Jun 24 22:45:54	pmox03	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:45:05	Jun 24 22:45:11	pmox02	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:42:50	Jun 24 22:42:56	pmox01	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:24:22	Jun 24 22:24:58	pmox03	root@pam	Shell	OK

All three Proxmox hosts are running the Ceph Monitor

Creating a Ceph Pool for VM and Container storage

Now that we have the OSDs and Monitors configured, we can create our Ceph Pool. Below we can see the replicas and minimum replicas.

The screenshot shows the Proxmox VE 8.0.3 interface. In the top navigation bar, there are links for Documentation, Create VM, Create CT, and a user session (root@pam). Below the navigation, the main menu shows 'Server View' and a gear icon. The left sidebar lists nodes: Datacenter (cluster01) containing pmox01 (with localnetwork, local, and local-lvm), pmox02, and pmox03. The pmox01 node is selected. The right panel displays a table for a Ceph pool named '.mgr' with 3/2 data distributed across 1 placement group, with optimal and autoscale modes both set to 'on'. A red arrow points from the 'Create' button in the top right of the main area to the 'Create' button in the 'Create: Ceph Pool' dialog.

Create: Ceph Pool

Name:	Pool01	PG Autoscale Mode:	on
Size:	3	Add as Storage:	<input checked="" type="checkbox"/>
Min. Size:	2	Target Ratio:	0.0
Crush Rule:	replicated_rule	Target Size:	0 GiB
# of PGs:	128	Target Ratio takes precedence.	
		Min. # of PGs:	0

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jun 24 22:51:12	Jun 24 22:51:16	pmox01	root@pam	File ubuntu-22.04-standard_22.04-1...	OK
Jun 24 22:48:28	Jun 24 22:48:28	pmox03	root@pam	Ceph Monitor mon.pmx03 - Create	OK
Jun 24 22:48:14	Jun 24 22:48:15	pmox02	root@pam	Ceph Monitor mon.pmx02 - Create	OK
Jun 24 22:45:48	Jun 24 22:45:54	pmox03	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:45:05	Jun 24 22:45:11	pmox02	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:42:50	Jun 24 22:42:56	pmox01	root@pam	Ceph OSD sdb - Create	OK

Creating a Ceph Pool

Now, the Ceph Pool is automatically added to the Proxmox cluster nodes.

PROXMOX Virtual Environment 8.0.3

Server View | Documentation | Create VM | Create CT | root@pam | Help

Node 'pmox01'

Hosts | Create | Edit | Destroy

Options (highlighted)

Pool #	Name	Size/min	# of Plac...	Optimal ...	Autoscal...	CRU...
1	.mgr	3/2	1	1	on	repl
2	Pool01	3/2	128	32	on	repl

localnetwork (pmox01)

Pool01 (pmox01) (highlighted)

local (pmox01)

local-lvm (pmox01)

pmox02

localnetwork (pmox02)

Pool01 (pmox02) (highlighted)

local (pmox02)

local-lvm (pmox02)

pmox03

localnetwork (pmox03)

Pool01 (pmox03) (highlighted)

local (pmox03)

local-lvm (pmox03)

Updates

Repositories (highlighted)

Time

Syslog

Firewall

Disks

LVM

LVM-Thin

Directory

ZFS

Ceph

Configuration

Monitor

OSD

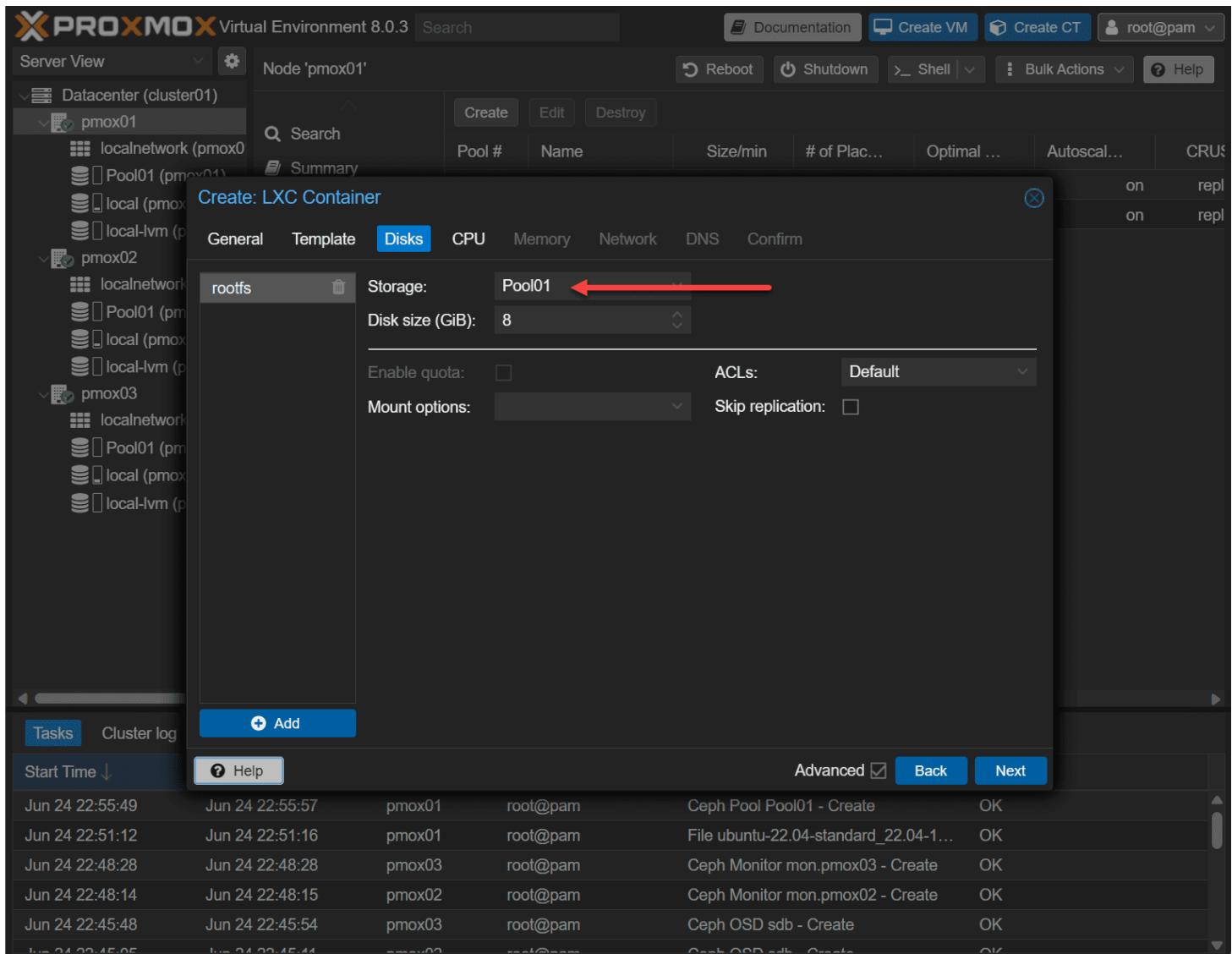
Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jun 24 22:55:49	Jun 24 22:55:57	pmox01	root@pam	Ceph Pool Pool01 - Create	OK
Jun 24 22:51:12	Jun 24 22:51:16	pmox01	root@pam	File ubuntu-22.04-standard_22.04-1... - Create	OK
Jun 24 22:48:28	Jun 24 22:48:28	pmox03	root@pam	Ceph Monitor mon.pmx03 - Create	OK
Jun 24 22:48:14	Jun 24 22:48:15	pmox02	root@pam	Ceph Monitor mon.pmx02 - Create	OK
Jun 24 22:45:48	Jun 24 22:45:54	pmox03	root@pam	Ceph OSD sdb - Create	OK
Jun 24 22:45:55	Jun 24 22:45:54	pmox02	root@pam	Ceph OSD sdb - Create	OK

Pool added to all three Proxmox nodes

Utilizing Ceph storage for Virtual Machines and Containers

Now that we have the Ceph Pool configured, we can use it for backing storage for Proxmox Virtual Machines and Containers. Below, I am creating a new LXC container. Note how we can choose the new Ceph Pool as the container storage.



Choosing the new pool for Proxmox LXC container storage

The LXC container creates successfully with no storage issues which is good.

PROXMOX Virtual Environment 8.0.3 Search

Documentation Create VM Create CT root@pam

Server View Node 'pmox01'

Datacenter (cluster01) pmox01 100 (ubuntu-ct01) localnetwork (pmox01) Pool Local Local

pmox02 Local Pool Local Local

pmox03 Local Pool Local Local

Task viewer: CT 100 - Create

Output Status

Stop Download

```
/dev/rbd0
Creating filesystem with 2097152 4k blocks and 524288 inodes
Filesystem UUID: 0b54e104-013f-42d1-8217-61ed699d4526
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
extracting archive '/var/lib/vz/template/cache/ubuntu-22.04-standard_22.04-1_amd64.tar.zst'
Total bytes read: 508579840 (486MiB, 239MiB/s)
Detected container architecture: amd64
Creating SSH host key 'ssh_host_dsa_key' - this may take some time ...
done: SHA256:9Y2MYSIDYp7EJFvVOu/f59ekdFxYEnerWUQSSb6Zsgg root@ubuntu-ct01
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:4poNj/CjEBMktHbRrRwCHuQxjxufIDuLo9ObKH3d3Do root@ubuntu-ct01
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:juyTLnkMfNVYsam9vrZL4gyYnulXyGUYIYNJehts74 root@ubuntu-ct01
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:EDicgfO0peJqpviUSJLRR3JwOY1sbjvxaAtbXoaJBs root@ubuntu-ct01
TASK OK
```

Tasks Cluster Start Time ↓

Start Time	End Time	Host	User	Description	Status
Jun 24 23:01:47	Jun 24 23:01:58	pmox01	root@pam	CT 100 - Create	OK
Jun 24 22:55:49	Jun 24 22:55:57	pmox01	root@pam	Ceph Pool Pool01 - Create	OK
Jun 24 22:51:12	Jun 24 22:51:16	pmox01	root@pam	File ubuntu-22.04-standard_22.04-1...	OK
Jun 24 22:48:28	Jun 24 22:48:28	pmox03	root@pam	Ceph Monitor mon.pmx03 - Create	OK
Jun 24 22:48:14	Jun 24 22:48:15	pmox02	root@pam	Ceph Monitor mon.pmx02 - Create	OK
Jun 24 22:45:48	Jun 24 22:45:54	pmox02	root@pam	Ceph OSD Path - Create	OK

The new LXC container is created successfully on Ceph storage

We can see we have the container up and running without issue. Also, I was able to migrate the LXC container to another node without issue.

PROXMOX Virtual Environment 8.0.3 Search Documentation Create VM Create CT root@pam

Server View

Container 100 (ubuntu-ct01) on node 'pmox01' No Tags Start Shutdown Migrate Console M

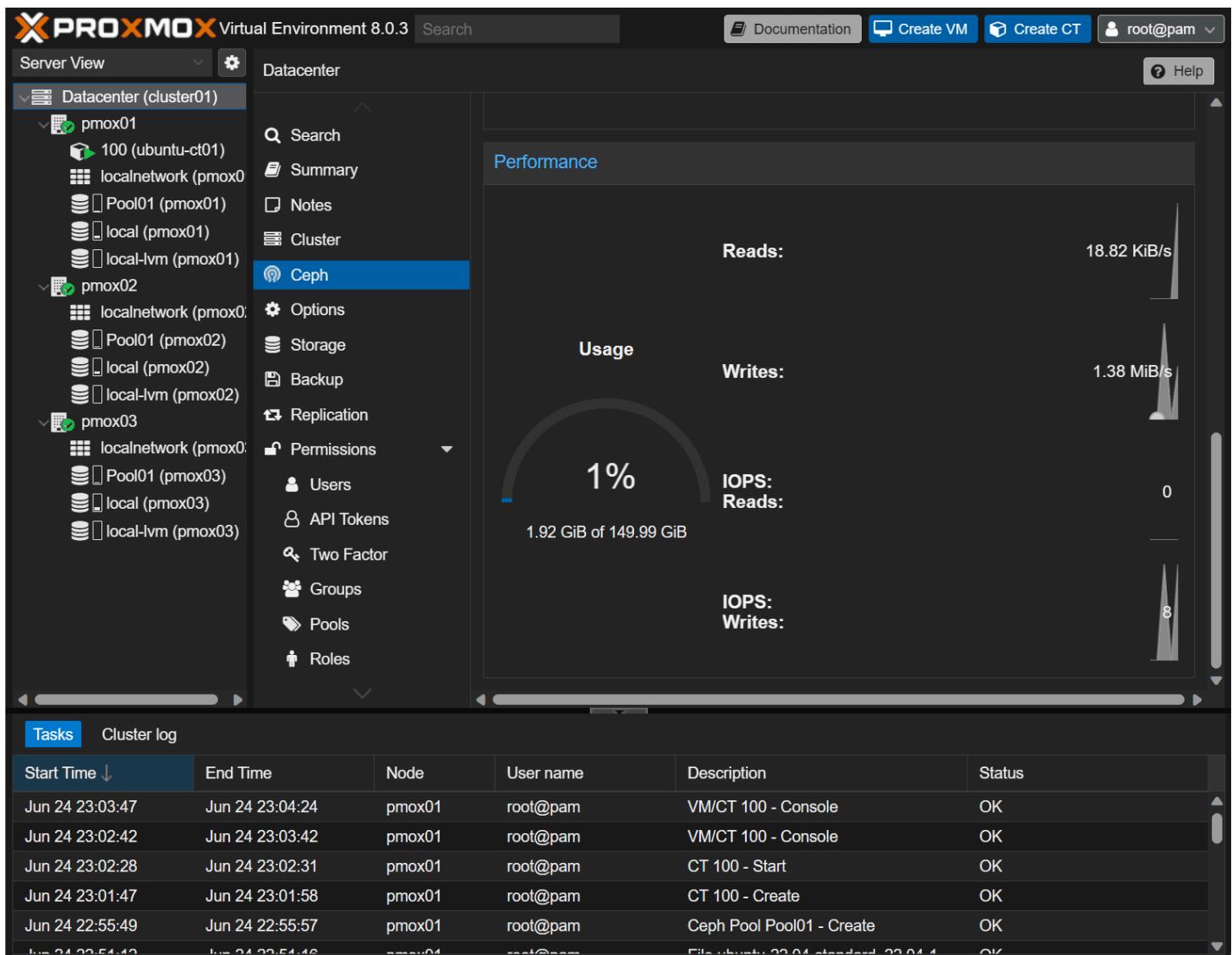
Summary
Console Resources Network DNS Options Task History Backup Replication Snapshots Firewall Permissions

```
root@ubuntu-ct01:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 5e:ab:28:78:a7:67 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.1.149.166/24 metric 1024 brd 10.1.149.255 scope global dynamic
      eth0
        valid_lft 172717sec preferred_lft 172717sec
        inet6 fe80::5cab:28ff:fe78:a767/64 scope link
            valid_lft forever preferred_lft forever
root@ubuntu-ct01:~# ping 10.1.149.1
PING 10.1.149.1 (10.1.149.1) 56(84) bytes of data.
64 bytes from 10.1.149.1: icmp_seq=1 ttl=64 time=1.35 ms
64 bytes from 10.1.149.1: icmp_seq=2 ttl=64 time=2.79 ms
^V64 bytes from 10.1.149.1: icmp_seq=3 ttl=64 time=2.62 ms
64 bytes from 10.1.149.1: icmp_seq=4 ttl=64 time=2.61 ms
```

Tasks Cluster log

Start Time	End Time	Node	User name	Description	Status
Jun 24 23:03:47		pmox01	root@pam	VM/CT 100 - Console	
Jun 24 23:02:42	Jun 24 23:03:42	pmox01	root@pam	VM/CT 100 - Console	OK
Jun 24 23:02:28	Jun 24 23:02:31	pmox01	root@pam	CT 100 - Start	OK
Jun 24 23:01:47	Jun 24 23:01:58	pmox01	root@pam	CT 100 - Create	OK
Jun 24 22:55:49	Jun 24 22:55:57	pmox01	root@pam	Ceph Pool Pool01 - Create	OK
Jun 24 22:54:12	Jun 24 22:54:46	pmox01	root@pam	File /ubuntu-22.04-standard-22.04.1	OK

The LXC container operating on the Ceph Pool



Ceph performance dashboard in Proxmox

Wrapping up

Ceph is totally free HCI storage that you can install and configure in Proxmox. It provides a way for you to have HCI storage totally free and have resilient software-defined storage for your virtualization environment to make sure your VMs are resilient without having to have a storage shelf and making use of local storage on each Proxmox server host.

This page intentionally left blank

CephFS Configuration in Proxmox Step-by-Step

Since working with Ceph in Proxmox VE lately, one of the cool features that I wanted to try out was Proxmox CephFS, which allows you to work with your Ceph installation directly from your clients. It allows mounting file storage to your clients on top of your Ceph storage pool with some other really cool benefits. Let's look at CephFS configuration in Proxmox and see how you can install and configure it.

Table of contents

- [What is CephFS \(CephFS file system\)?](#)
- [CephFS configuration in Proxmox: An Overview of the lab](#)
- [Installation steps](#)
- [Installing Ceph client tools in Linux](#)
 - [Ceph fuse](#)
- [Things you will need for your CephFS configuration in Proxmox](#)
 - [1. The admin keyring](#)
 - [2. The name of the Ceph file system](#)
 - [3. The monitor addresses of your Proxmox CephFS servers](#)
 - [4. A ceph.config file](#)
- [Connect a Linux client to CephFS running on Proxmox](#)
 - [Run the mount command to mount the Ceph file system](#)
 - [Troubleshooting and support](#)

What is CephFS (CephFS file system)?

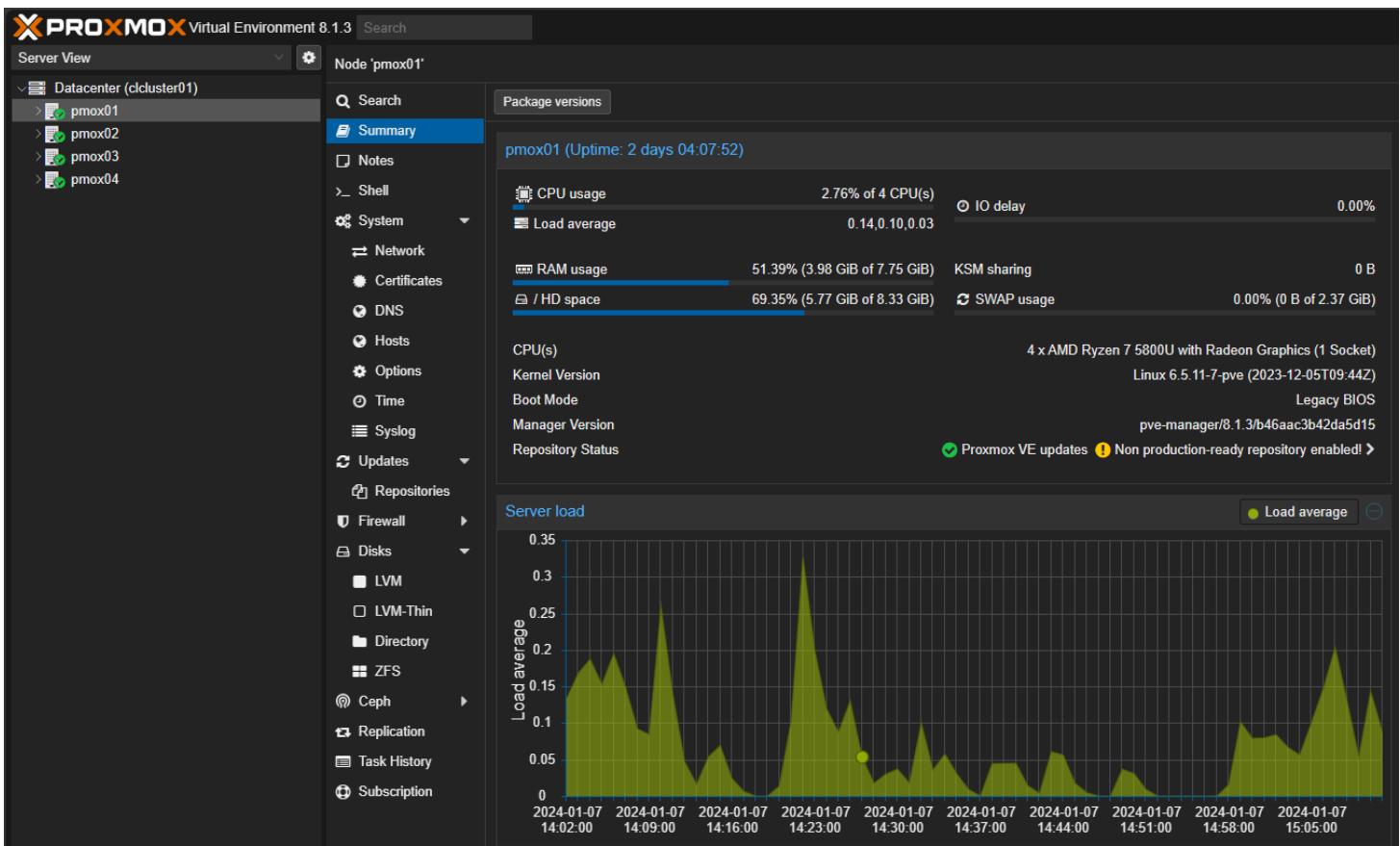
CephFS is a POSIX-compliant [file system](#) that offers a scalable and reliable solution for managing file data. CephFS is not specific to Proxmox. However, in [Proxmox environments when you configure a Ceph storage](#) pool, it uses the same file system that Proxmox uses for writing file data blocks and keeping replica data for resiliency from top to bottom.

CephFS can handle vast amounts of file metadata and data and be installed on commodity virtualization hardware. It is an excellent solution for many use cases, especially when integrated with a Ceph storage cluster, as we can do in Proxmox.

CephFS configuration in Proxmox: An Overview of the lab

After you have a working Ceph cluster on top of a [Proxmox installation](#), including Ceph mgr, cluster monitors (Ceph mon), Ceph OSDs, daemons, cluster network, and a Ceph storage pool, how do you enable the Ceph file system on top of that? It is super easy to do in Proxmox, especially since everything is integrated. We will see this integration in the menus below.

As a note, in this example I am running Proxmox VE version 8.1.3 and Ceph Quincy, which are the latest updates to the platform from the official site with various security enhancements and features. For the lab, I am running a simple 4 node member cluster (started with 3 but was doing other testing and added a node) in nested [virtual machines on an SSD disk](#) with 3/2 Crush rule. You can [configure different rules](#) based on your needs and infrastructure.

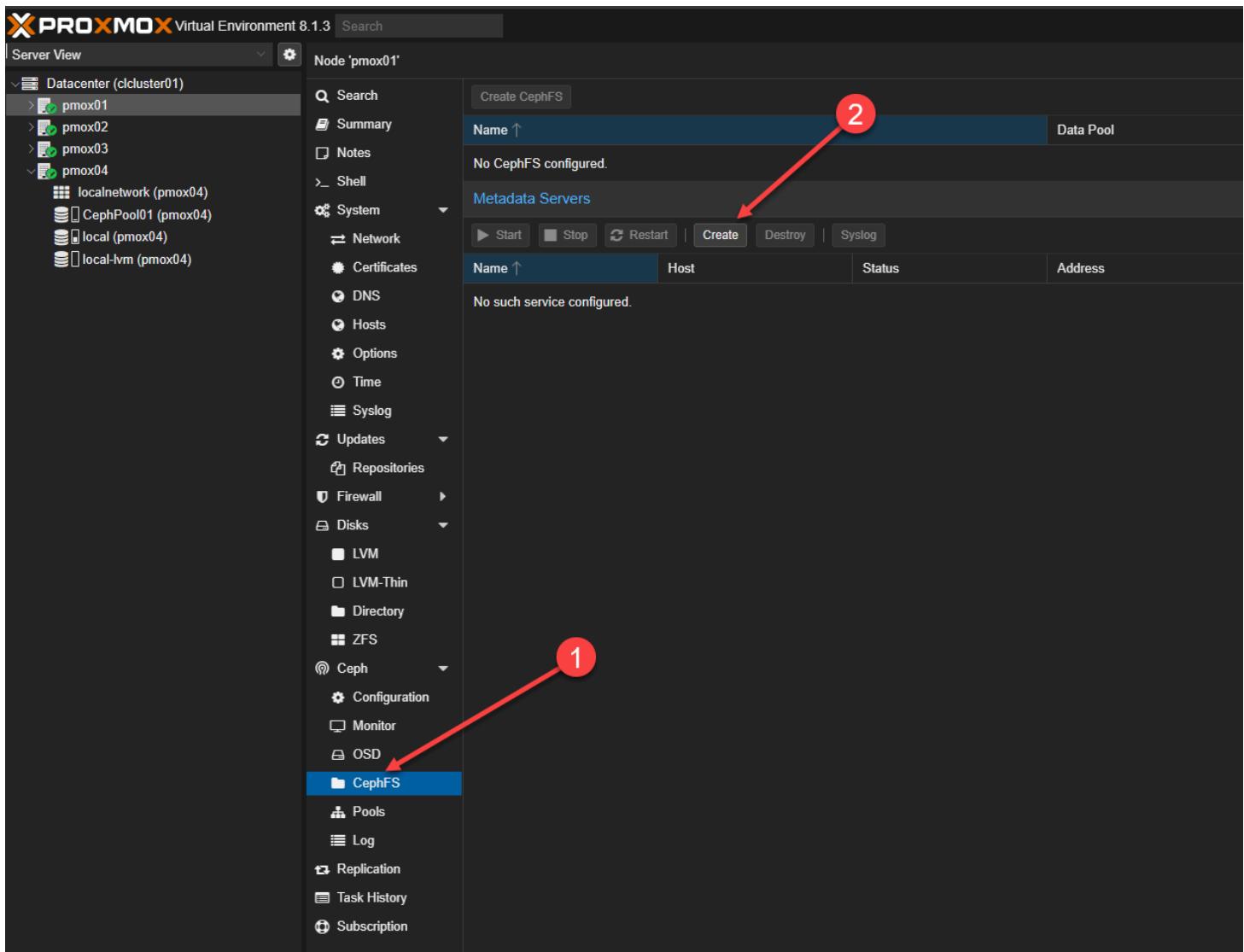


Cephfs home lab in proxmox

I set to replicated rule and a single NIC (multiple NICs and networks are recommended) for each machine running pveceph. In this small configuration, it leads to a significant amount of space used with replicas taking up 75% of the capacity in order to created the replicated data and additional writes with changes.

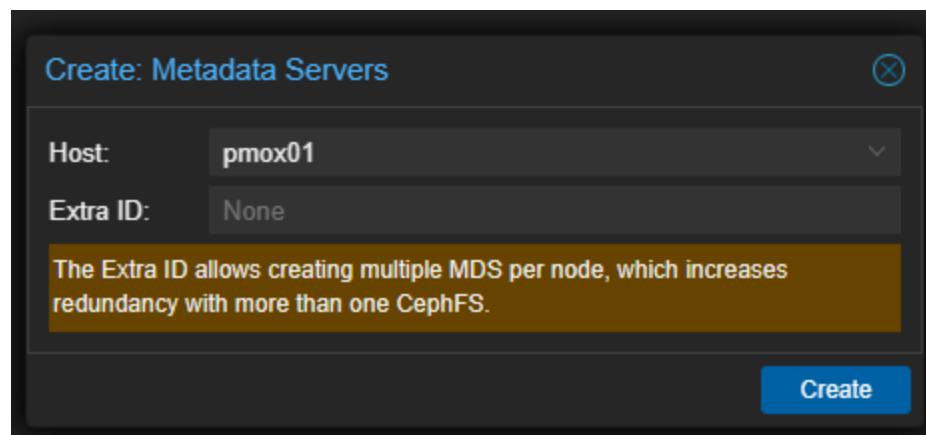
Installation steps

First, click the **CephFS** menu under Ceph for your Proxmox host. Next, you click the **Create** button in the Proxmox web app.



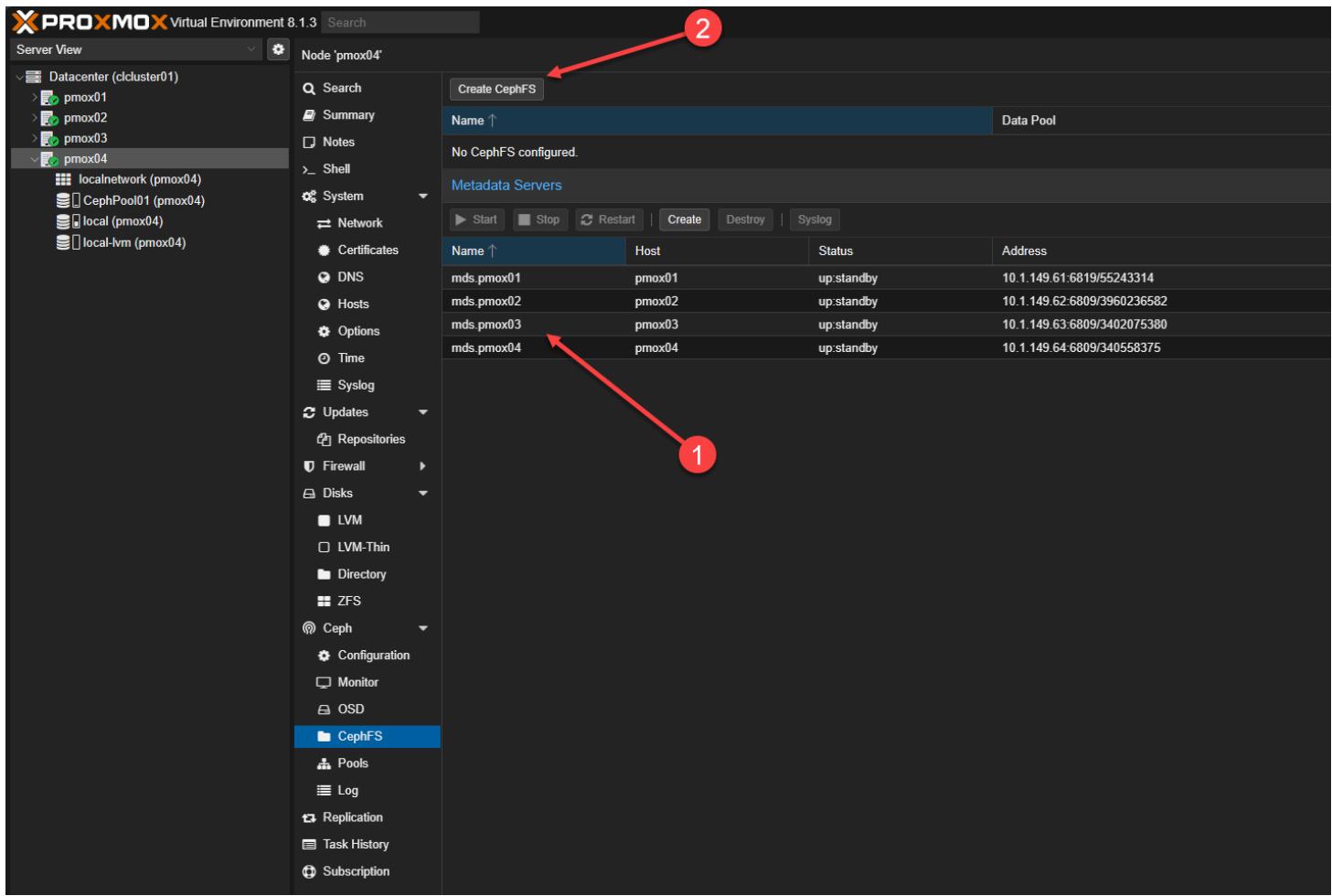
Beginning to create cephfs in proxmox

This will launch the dialog box to **Create: Metadata Servers**.



Create metadata servers dialog box

- 1) In my lab, I made each Proxmox host a Metadata server. 2) Click the **Create CephFS** button at the top.



Click the create cephfs button

I left all the default options here:

- Name
- Placement groups: default 128
- Add as Storage checked

Click **Create**.

The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the 'Server View' sidebar is open, showing a tree structure of nodes: Datacenter (clcluster01) containing pmox01, pmox02, pmox03, and pmox04, which further contains localnetwork (pmox04), CephPool01 (pmox04), local (pmox04), and local-lvm (pmox04). The 'CephFS' option under the 'Ceph' section is selected. In the main pane, the 'Create CephFS' dialog is open, prompting for a 'Name' (set to 'cephfs') and 'Placement Groups' (set to '128'). A red arrow points to the 'Create' button at the bottom right of the dialog. Above the dialog, a table lists four 'Metadata Servers' (mds.pmx01 to mds.pmx04) with their respective hosts (pmox01 to pmox04), status (up:standby), and addresses.

Name	Host	Status	Address
mds.pmx01	pmox01	up:standby	10.1.149.61:6819/55243314
mds.pmx02	pmox02	up:standby	10.1.149.62:6809/3960236582
mds.pmx03	pmox03	up:standby	10.1.149.63:6809/3402075380
mds.pmx04	pmox04	up:standby	10.1.149.64:6809/340558375

Create cephfs after creating metadata servers

In the Task viewer you will see the status of the task which should complete successfully.

Task viewer: CephFS cephfs - Create



Output Status

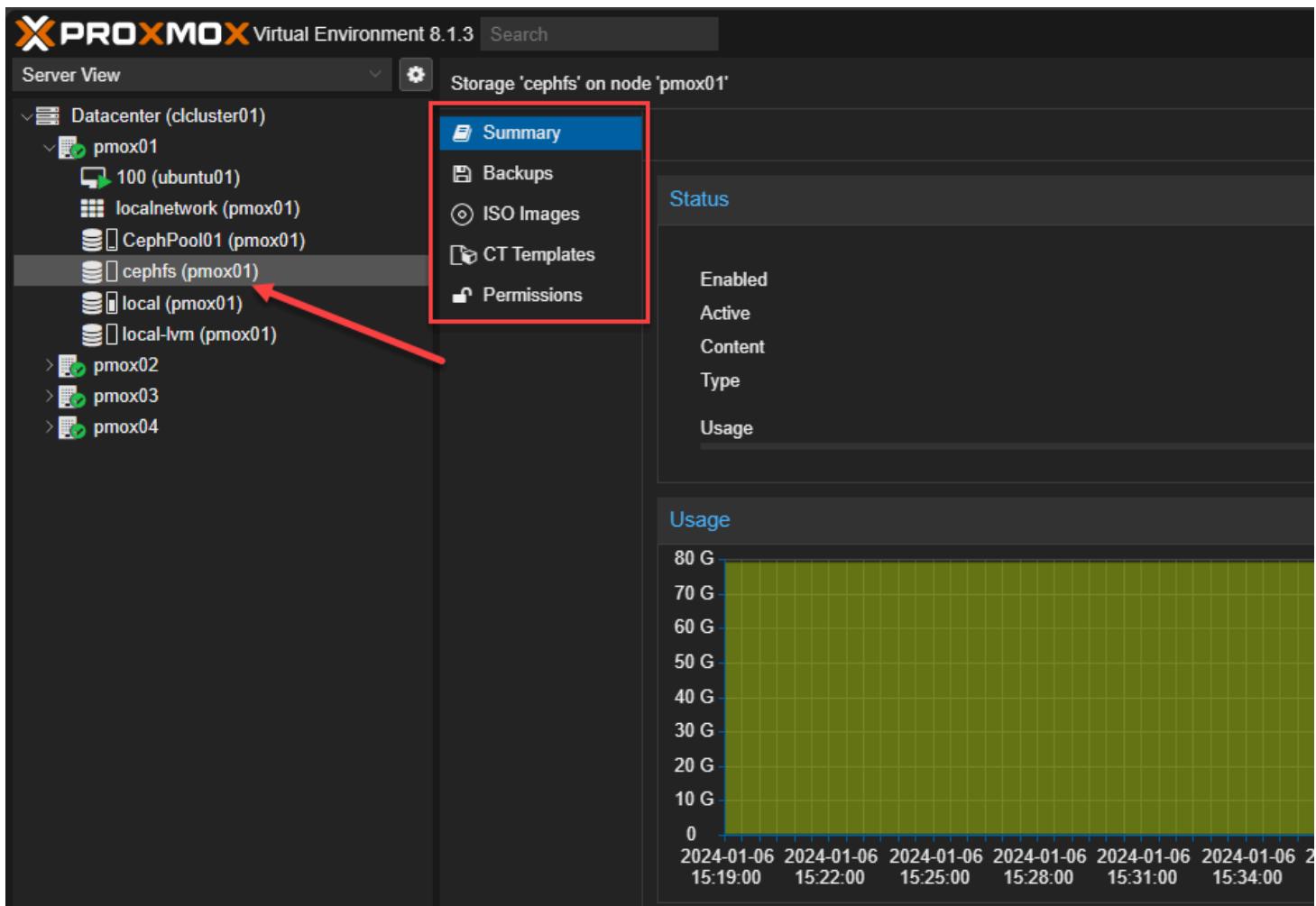
Stop

Download

```
creating data pool 'cephfs_data'...
pool cephfs_data: applying application = cephfs
pool cephfs_data: applying pg_num = 128
creating metadata pool 'cephfs_metadata'...
pool cephfs_metadata: applying pg_num = 32
configuring new CephFS 'cephfs'
Successfully create CephFS 'cephfs'
Adding 'cephfs' to storage configuration...
Waiting for an MDS to become active
Waiting for an MDS to become active
TASK OK
```

Viewing the create cephfs task

If you choose to mount as storage, you will see the CephFS storage listed under your Proxmox host(s). Also, the great thing about the CephFS storage is you can use it to store things like ISOs, etc on top of your Ceph storage pools. Note in the navigation, we see the types of resources and content we can store, including ISO disks, etc.



Viewing the cephfs storage in proxmox

Installing Ceph client tools in Linux

To work with Ceph FS on Linux client nodes (Ceph clients), you install the Ceph client tools software packages from the CLI.
sudo apt install ceph-common

```

linuxadmin@cldockertest3:~$ sudo apt install ceph-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ibverbs-providers libbabeltrace1 libboost-context1.74.0 libboost-filesystem1.74.0 libboost-iostreams1.74.0 libibverbs1 liblua5.3-0 libndctl16 libnl-route-3-200 liboauth0 libpmem1 libpmemobj1 librabbitmq4 librados2 libradosstripe1 python3-prettytable python3-rados python3-rbd python3-wcwidth
Suggested packages:
  ceph ceph-mds
The following NEW packages will be installed:
  ceph-common ibverbs-providers libbabeltrace1 libboost-context1.74.0 libboost-filesystem1.74.0 libboost-iostreams1.74.0 libibverbs1 liblua5.3-0 libndctl16 libnl-route-3-200 liboauth0 libpmem1 libpmemobj1 librabbitmq4 librados2 libradosstripe1 python3-cephfs python3-prettytable python3-rados python3-rbd python3-wcwidth
0 upgraded, 32 newly installed, 0 to remove and 11 not upgraded.
Need to get 35.0 MB of archives.
After this operation, 141 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libboost-iostreams1.74.0 amd64 1.74.0-14ubuntu3 [24.4 MB]
Get:2 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libboost-thread1.74.0 amd64 1.74.0-14ubuntu3 [262 KB]
Get:3 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libnl-route-3-200 amd64 3.5.0-0.1 [180 kB]
Get:4 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libibverbs1 amd64 39.0-1 [69.3 kB]
Get:5 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 librdmacm1 amd64 39.0-1 [71.2 kB]
Get:6 http://gb.archive.ubuntu.com/ubuntu jammy-updates/main amd64 librados2 amd64 17.2.6-0ubuntu0.22.04.2 [3,540 kB]
Get:7 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libdaxctl1 amd64 72.1-1 [19.8 kB]
Get:8 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libndctl16 amd64 72.1-1 [57.7 kB]
Get:9 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libpmem1 amd64 1.11.1-3build1 [81.4 kB]
Get:10 http://gb.archive.ubuntu.com/ubuntu jammy/main amd64 libpmemobj1 amd64 1.11.1-3build1 [124 kB]
Get:11 http://gb.archive.ubuntu.com/ubuntu jammy-updates/main amd64 librbd1 amd64 17.2.6-0ubuntu0.22.04.2 [3,540 kB]

```

Installing ceph common components on a linux client

Ceph fuse

Also, you can install the **ceph fuse** package. The ceph-fuse package is an alternate way of mounting CephFS. The difference is it mounts it in the userspace. The performance of ceph-fuse is not as good as the more traditional mounting of a CephFS file system.

However, it does allow you to connect to a Ceph distributed file system from a user's perspective, without the need to integrate it deeply into the system's core.

You can specify which Ceph file system to connect to either through a [command line](#) option (-m) or by using a configuration file (ceph.conf). This tool mounts the Ceph file [system at a designated location on your system](#).[sudo apt install ceph-fuse](#)

```
~# sudo apt install ceph-fuse
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  ceph-fuse
0 upgraded, 1 newly installed, 0 to remove and 11 not upgraded.
Need to get 856 kB of archives.
After this operation, 2,669 kB of additional disk space will be used.
Get:1 http://gb.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 ceph-fuse amd64 17.2.6-0ubuntu0.22.04.2 [856 kB]
Fetched 856 kB in 1s (571 kB/s)
Selecting previously unselected package ceph-fuse.
(Reading database ... 110666 files and directories currently installed.)
Preparing to unpack .../ceph-fuse_17.2.6-0ubuntu0.22.04.2_amd64.deb ...
Unpacking ceph-fuse (17.2.6-0ubuntu0.22.04.2) ...
Setting up ceph-fuse (17.2.6-0ubuntu0.22.04.2) ...
Created symlink /etc/systemd/system/remote-fs.target.wants/ceph-fuse.target → /lib/systemd/system/ceph-fuse.target.
Created symlink /etc/systemd/system/ceph.target.wants/ceph-fuse.target → /lib/systemd/system/ceph-fuse.target.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Restarting services...
Service restarts being deferred:
  systemctl restart ModemManager.service
  systemctl restart docker.service
  systemctl restart unattended-upgrades.service

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@cldockertest3:~#
```

Installing ceph fuse components

Things you will need for your CephFS configuration in Proxmox

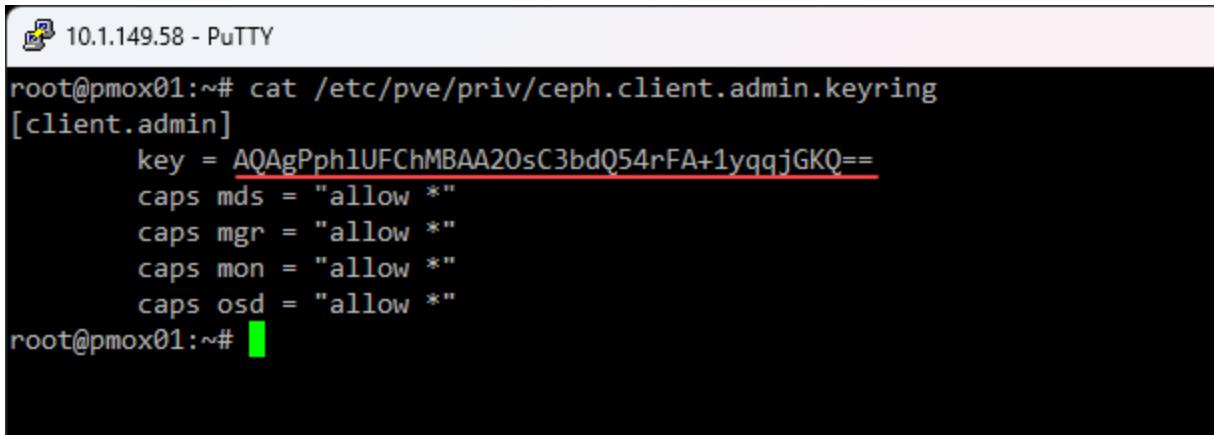
There are a few steps you will need to [connect to your Promox CephFS installation](#):

1. The admin keyring
2. The name of the Ceph file system
3. The monitor addresses of your CephFS servers
4. A ceph.config file

1. The admin keyring

To see the admin credentials that you need to mount the CephFS file system, you need to get your **key** from the **ceph.client.admin.keyring** file. To get this, run the command:`cat /etc/pve/priv/ceph.client.admin.keyring`

You will see the value in the **key** section of the file. Note the user is **admin** and not **root**.



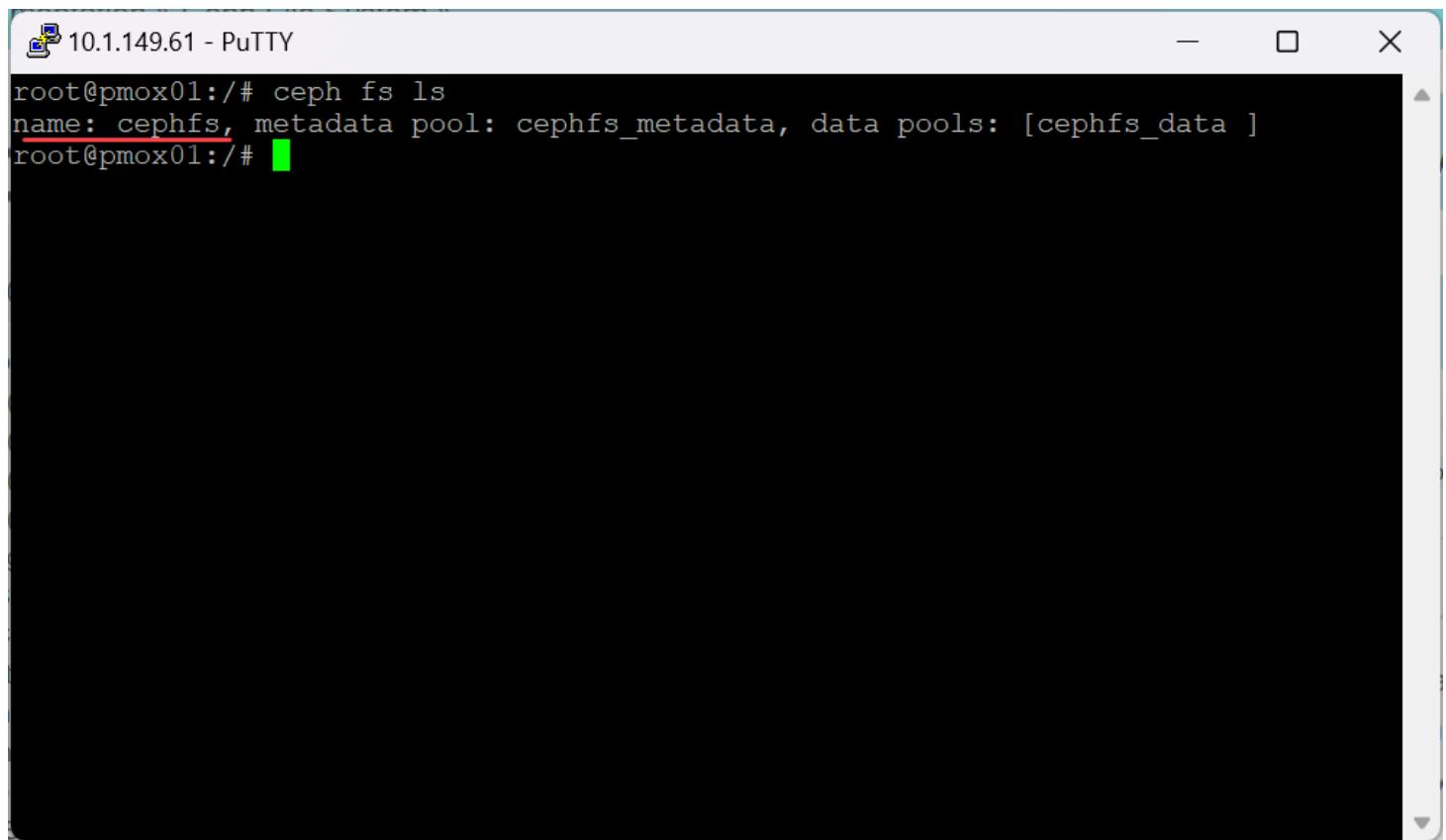
```
root@pmox01:~# cat /etc/pve/priv/ceph.client.admin.keyring
[client.admin]
key = AQAgPph1UFChMBA20sC3bdQ54rFA+1yqqjGKQ==
caps mds = "allow *"
caps mgr = "allow *"
caps mon = "allow *"
caps osd = "allow *"
root@pmox01:~#
```

Viewing the admin key in proxmox for cephfs

2. The name of the Ceph file system

The next piece of information you need is the name of the Ceph file system. To get that, you can run this command on your Proxmox host: ceph fs ls

You will see the name of the file system. The default name is **cephfs**.



```
root@pmox01:/# ceph fs ls
name: cephfs, metadata pool: cephfs_metadata, data pools: [cephfs_data ]
root@pmox01:/#
```

Running the ceph fs ls command

3. The monitor addresses of your Proxmox CephFS servers

You will need to have the Ceph monitor server addresses. There should be multiple [servers configured as monitors](#) for reliability and so you don't have a single point of failure.

You can file these hosts addresses under the **Ceph > Monitor** menu in the Proxmox GUI in the browser. Make sure your router or routers have the [routes configured](#) to allow your client devices to have connectivity to these IP addresses and port configurations.

The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the 'Server View' sidebar lists nodes: Datacenter (clcluster01), pmox01, pmox02, pmox03, and pmox04. The 'Node 'pmox01'' details panel is open. In the 'Monitor' section, there is a table:

Name	Host	Status	Address
mon.pmx01	pmox01	running	10.1.149.61:6789/0
mon.pmx02	pmox02	running	10.1.149.62:6789/0
mon.pmx03	pmox03	running	10.1.149.63:6789/0

A red arrow points to the 'Address' column for mon.pmx03.

Viewing the proxmox ceph monitor addresses

4. A ceph.config file

You will also need a **ceph.config** file. Like the admin keyring, we can also [copy the file](#) from the Proxmox server. But we will trim some of the information out of the Proxmox server file. This file is located here on your Proxmox server:/etc/pve/ceph.config

Mine has the following contents for my Proxmox server environment.

```
[global]
auth_client_required = cephx
auth_cluster_required = cephx
auth_service_required = cephx
cluster_network = 10.1.149.61/24
fsid = 75a2793d-00b7-4da5-81ce-48347089734d
mon_allow_pool_delete = true
mon_host = 10.1.149.61 10.1.149.63 10.1.149.62
ms_bind_ipv4 = true
ms_bind_ipv6 = false
osd_pool_default_min_size = 2
osd_pool_default_size = 3
public_network = 10.1.149.61/24

[client]
keyring = /etc/pve/priv/$cluster.$name.keyring

[mds]
keyring = /var/lib/ceph/mds/ceph-$id/keyring

[mds.pmx01]
```

```

host = pmox01
mds_standby_for_name = pve

[mds.pmx02]
host = pmox02
mds_standby_for_name = pve

[mds.pmx03]
host = pmox03
mds_standby_for_name = pve

[mds.pmx04]
host = pmox04
mds_standby_for_name = pve

[mon.pmx01]
public_addr = 10.1.149.61

[mon.pmx02]
public_addr = 10.1.149.62

[mon.pmx03]
public_addr = 10.1.149.63

```

Connect a Linux client to CephFS running on Proxmox

To connect a Linux client to our CephFS configuration in Proxmox, we need to create a couple of files. First, make the following directory:

```
mkdir /etc/ceph
```

In that directory create the files:

- admin.keyring
- ceph.conf

```
linuxadmin@cldockertest3:/etc/ceph$ tree
.
├── admin.keyring
└── ceph.conf

0 directories, 2 files
linuxadmin@cldockertest3:/etc/ceph$ 
```

Running the tree command on the directory housing the cephfs configuration files

In the **admin.keyring** file, just put the **key** value in the file, nothing else. It will be a value as we had shown above that looks similar to this:

```
AQAgPphlUFChMBAA20sC3bdQ54rFA+1yqqjGKQ==
```

Then, you will need the following in your **ceph.conf** file. As you can see below, I have updated the keyring location to point to our **admin.keyring** file.

```

[global]
auth_client_required = cephx
auth_cluster_required = cephx
auth_service_required = cephx
cluster_network = 10.1.149.0/24
fsid = 75a2793d-00b7-4da5-81ce-48347089734d
mon_allow_pool_delete = true
mon_host = 10.1.149.61 10.1.149.63 10.1.149.62
ms_bind_ipv4 = true
ms_bind_ipv6 = false
osd_pool_default_min_size = 2
osd_pool_default_size = 3
public_network = 10.1.149.0/24

[client]
keyring = /etc/ceph/admin.keyring

```

Run the mount command to mount the Ceph file system

We need to make a directory for the mount operation.

```
mkdir /mnt/cephfs
```

Now that we have a directory, we can run the following command to mount the CephFS file system for a connection to the IP address of each monitor node.

```
sudo mount -t ceph admin@75a2793d-00b7-4da5-81ce-48347089734d.cephfs=/ /mnt/cephfs -o
'secretfile=/etc/ceph/admin.keyring,mon_addr=10.1.149.61/10.1.149.62/10.1.149.63'
```

The command will complete without any return if it is successful. We can run the following to see our mounted Ceph file system:

```
df -h
```

```

linuxadmin@cldockertest3:/etc/ceph$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           393M  1.4M  392M  1% /run
/dev/sda2        40G   11G   27G  30% /
tmpfs           2.0G    0  2.0G  0% /dev/shm
tmpfs           5.0M    0  5.0M  0% /run/lock
tmpfs           393M   4.0K  393M  1% /run/user/0
10.1.149.61,10.1.149.62,10.1.149.63:/ 74G    0  74G  0% /mnt/cephfs
linuxadmin@cldockertest3:/etc/ceph$ in a

```

Cephfs mounted in linux

Troubleshooting and support

Like any technology, there may be times when you need to troubleshoot something with CephFS. CephFS does not require a subscription license as it is free and open-source and can be pulled from the no-subscription repository.

Customers can of course, opt for enterprise support for your [Proxmox cluster](#) with the customer portal from the Proxmox team. If you still go the open-source route, the Proxmox support [forum](#) on the Internet is a great source of help for visitors across tens of thousands of threads thanks to activity members in the community. In addition, you can search forums for a wide variety of topics, instructions, question-answer type posts, etc.

There are a number of other home forums and websites, links, wiki sites and thread search titles where you can find people with experience to help with troubleshooting warning messages and errors, and share log data.

Wrapping up installing CephFS in Proxmox

CephFS configuration in Proxmox is an extremely powerful storage solution you can run in Proxmox for most dev and prod environments. However, not only does it allow you to have hyper-converged storage for vm instances and [LXC container](#) instances, it allows you to have file storage you can mount for clients that runs on top of the Ceph storage pool, with all the benefits that come with Ceph in terms of resiliency, scalability, and availability.

There are a lot of community support resources if you need help troubleshooting issues or figuring out the details and settings, even with the open-source no-subscription channels for Proxmox and Ceph Quincy.

Proxmox HA Cluster Configuration for Virtual Machines

If you are learning the Proxmox hypervisor or want high-availability cluster resources for learning and self-hosting services with some resiliency, building a cluster is not too difficult. Also, you can easily create Proxmox HA virtual machine clustering once you create cluster nodes. Let's look at Proxmox HA virtual machine deployment and how to ensure your VM is protected against failure and increase uptime, much like VMware HA.

Table of contents

- [Proxmox cluster: the starting point](#)
 - [Shared storage](#)
- [Setting Up Your Proxmox Cluster](#)
 - [Key Steps in Creating a Proxmox Cluster](#)
- [Configuring Virtual machine HA](#)
 - [High Availability Setup Requirements](#)
- [Configuring HA groups \(optional\)](#)
- [Fencing device configuration](#)
- [Rebooting Proxmox Servers running HA](#)
- [Wrapping Up](#)

Proxmox cluster: the starting point

The starting point for a high availability solution with Proxmox is the Proxmox cluster. Most start with a single Proxmox server in the [home lab](#). However, building a cluster requires a 2nd and third node. There are ways to increase the vote of one node if you have two Proxmox servers in a cluster if one goes down. However, for “production” Proxmox VE, having 3 nodes is the standard for configuring a minimum Proxmox clusters for scalability and availability.

Below, I have three nodes in a Proxmox cluster running Proxmox 8.1.3 in the Proxmox UI.

Start Time	End Time	Node	User name	Description	Status
Jan 05 11:33:11	Jan 05 11:33:21	pmox03	root@pam	Join Cluster	OK
Jan 05 11:32:53	Jan 05 11:33:05	pmox02	root@pam	Join Cluster	OK
Jan 05 11:32:23	Jan 05 11:32:26	pmox01	root@pam	Create Cluster	OK
Jan 05 11:04:15	Jan 05 11:04:15	pmox01	root@pam	Bulk start VMs and Containers	OK
Jan 05 11:03:54	Jan 05 11:03:54	pmox1	root@pam	Bulk shutdown VMs and Containers	OK

Proxmox cluster running three nodes in an ha configuration

A Proxmox cluster includes multiple Proxmox servers or nodes that operate together as a logical unit to run your workloads. Understanding how to set up and manage the PVE cluster service effectively is important to ensure your VM [data is protected](#) and you have containers hardware redundancy.

Remember that this doesn't replace all the other best practices with hardware configurations, such as redundant network hardware and power supplies in your [Proxmox hosts](#) and UPS battery backup as the basics.

Shared storage

When you are thinking about a [Proxmox cluster and virtual machine](#) high availability, you need to consider integration with shared storage as part of your design. Shared storage is a requirement so that all Proxmox cluster hosts have access to the data for your VMs. If a Proxmox host goes down, the other Promox hosts can pick up running the VM with the data they already have access to.

You can run a Proxmox cluster where each node has local storage, but this will not allow the VM to be highly available.

For my test cluster, I [configured Proxmox Ceph storage](#). However, many other types of [shared storage can work such as an iSCSI](#) or other connection to a ZFS pool, etc. Below, we are navigating to Ceph and choosing to [Install Ceph](#).

The screenshot shows the Proxmox VE 8.1.3 interface. In the left sidebar under 'Datacenter', 'pmox01' is expanded, showing 'localnetwork (pmox01)', 'local (pmox01)', and 'local-lvm (pmox01)'. Below them are 'pmox02' and 'pmox03'. The 'Ceph' option is selected in the sidebar menu. In the main content area, the 'Health' section shows a large question mark icon and 'No Warnings/Errors'. The 'Status' section has tabs for 'OSDs' and 'PGs'. An 'Install Ceph' button is visible in a modal dialog box. Below these are sections for 'Services', 'Monitors', 'Managers', and 'Meta Data Servers'. At the bottom, there is a 'Tasks' table showing cluster log entries:

Start Time	End Time	Node	User name	Description	Status
Jan 05 11:33:11	Jan 05 11:33:21	pmox03	root@pam	Join Cluster	OK
Jan 05 11:32:53	Jan 05 11:33:05	pmox02	root@pam	Join Cluster	OK
Jan 05 11:32:23	Jan 05 11:32:26	pmox01	root@pam	Create Cluster	OK
Jan 05 11:04:15	Jan 05 11:04:15	pmox01	root@pam	Bulk start VMs and Containers	OK
Jan 05 11:03:54	Jan 05 11:03:54	pmox1	root@pam	Bulk shutdown VMs and Containers	OK

Getting started installing ceph

This launches the **Info** screen. Here I am choosing to install Ceph Reef and using the **No-Subscription** repo.

PROXMOX Virtual Environment 8.1.3 Search

Documentation Create VM Create CT root@pam Help

Server View Datacenter

pmox01 localnetwork (pmox01) local (pmox01) local-lvm (pmox01)

pmox02 pmox03

Setup

Info Installation Configuration Success

Ceph?

"Ceph is a unified, distributed storage system, designed for excellent performance, reliability, and scalability."

Ceph is currently not installed on this node. This wizard will guide you through the installation. Click on the next button below to begin. After the initial installation, the wizard will offer to create an initial configuration. This configuration step is only needed once per cluster and will be skipped if a config is already present.

Before starting the installation, please take a look at our documentation, by clicking the help button below. If you want to gain deeper knowledge about Ceph, visit ceph.com.

Hint: The no-subscription repository is not the best choice for production setups.

Ceph in the cluster: Could not detect a ceph installation in the cluster

Ceph version to install: reef (18.2) Repository: No-Subscription

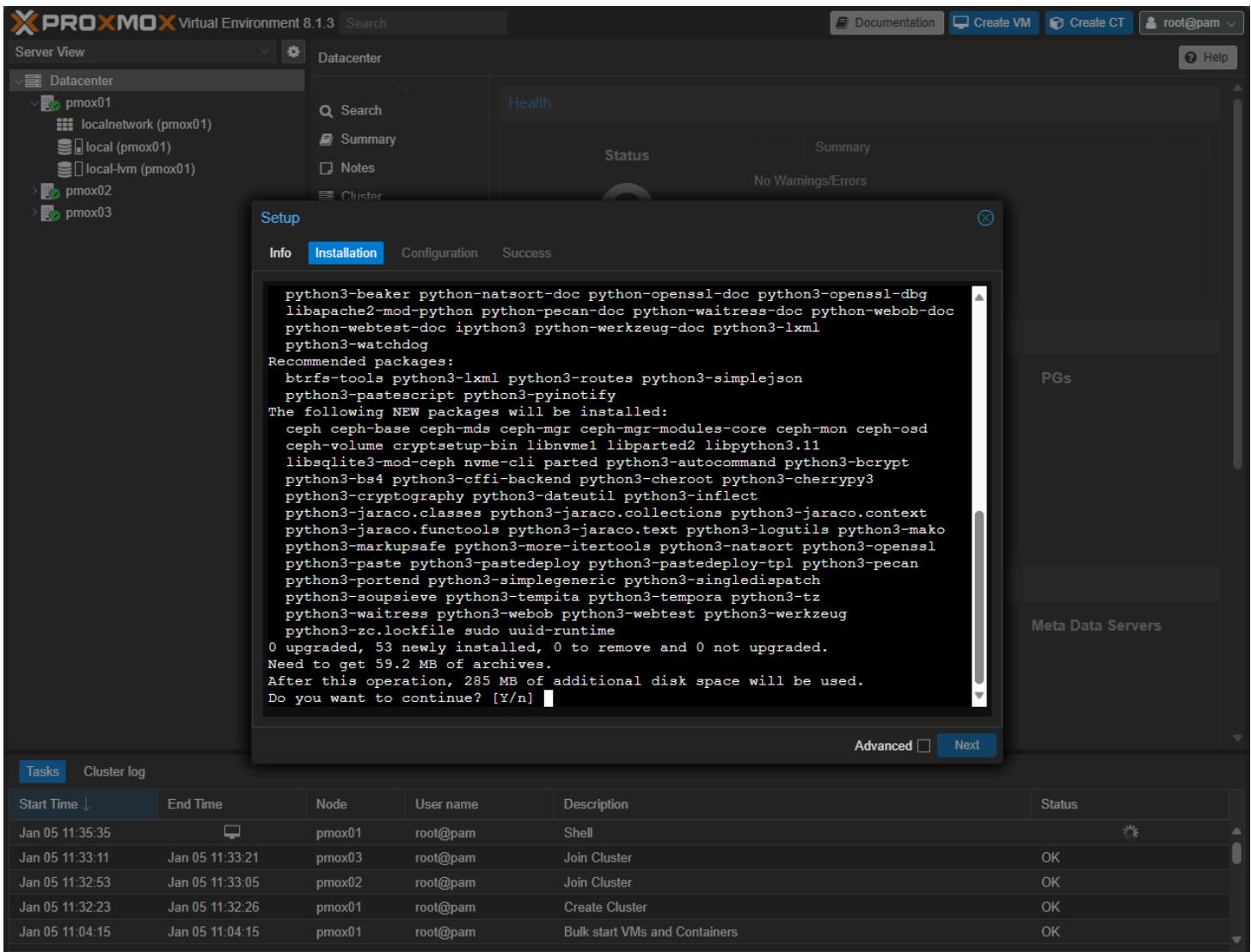
Advanced Start reef installation

Tasks Cluster log

Start Time ↓	End Time	Node	User name	Description	Status
Jan 05 11:33:11	Jan 05 11:33:21	pmox03	root@pam	Join Cluster	OK
Jan 05 11:32:53	Jan 05 11:33:05	pmox02	root@pam	Join Cluster	OK
Jan 05 11:32:23	Jan 05 11:32:26	pmox01	root@pam	Create Cluster	OK
Jan 05 11:04:15	Jan 05 11:04:15	pmox01	root@pam	Bulk start VMs and Containers	OK
Jan 05 11:03:54	Jan 05 11:03:54	pmox1	root@pam	Bulk shutdown VMs and Containers	OK

Starting the ceph setup

Type **Y** to begin the installation of Ceph.



Confirm the ceph installation

Create the Ceph Pool, including configuring the:

- Name
- Size
- Min Size
- Crush Rule
- # of PGs
- PG autoscale mode

Below is the default value for the configuration.

Create: Ceph Pool

Name:	CephPool01	PG Autoscale Mode:	on
Size:	3	Add as Storage:	<input checked="" type="checkbox"/>
Min. Size:	2	Target Ratio:	0.0
Crush Rule:	replicated_rule	Target Size:	0 GiB
# of PGs:	128	Target Ratio takes precedence.	
		Min. # of PGs:	0

Help **Advanced** **Create**

Creating the ceph pool

A healthy Ceph pool after installing Ceph on all three nodes, creating OSDs, Managers, Monitors, etc.

The screenshot shows the Proxmox Virtual Environment 8.1.3 interface. On the left, the navigation tree under 'Datacenter' shows three nodes: pmox01, pmox02, and pmox03. The 'Ceph' tab is selected in the sidebar. The main panel displays the 'Health' status as 'HEALTH_OK' with 'No Warnings/Errors' and 'Ceph Version: 18.2.0'. Below it, the 'Status' section shows 3 OSDs (all Up) and 122 PGs (all active+clean). The 'Services' section lists Monitors (pmox01, pmox02, pmox03), Managers (pmox01, pmox02, pmox03), and Meta Data Servers (pmox01, pmox02, pmox03). At the bottom, a table titled 'Tasks' shows recent operations: 'Ceph Pool CephPool01 - Create', 'Ceph Manager mgr.pmx03 - Create', 'Ceph Manager mgr.pmx02 - Create', 'Ceph OSD sdb - Create', and 'Ceph OSD sdb - Create'. All tasks have a status of 'OK'.

A healthy ceph storage configuration and pool

Setting Up Your Proxmox Cluster

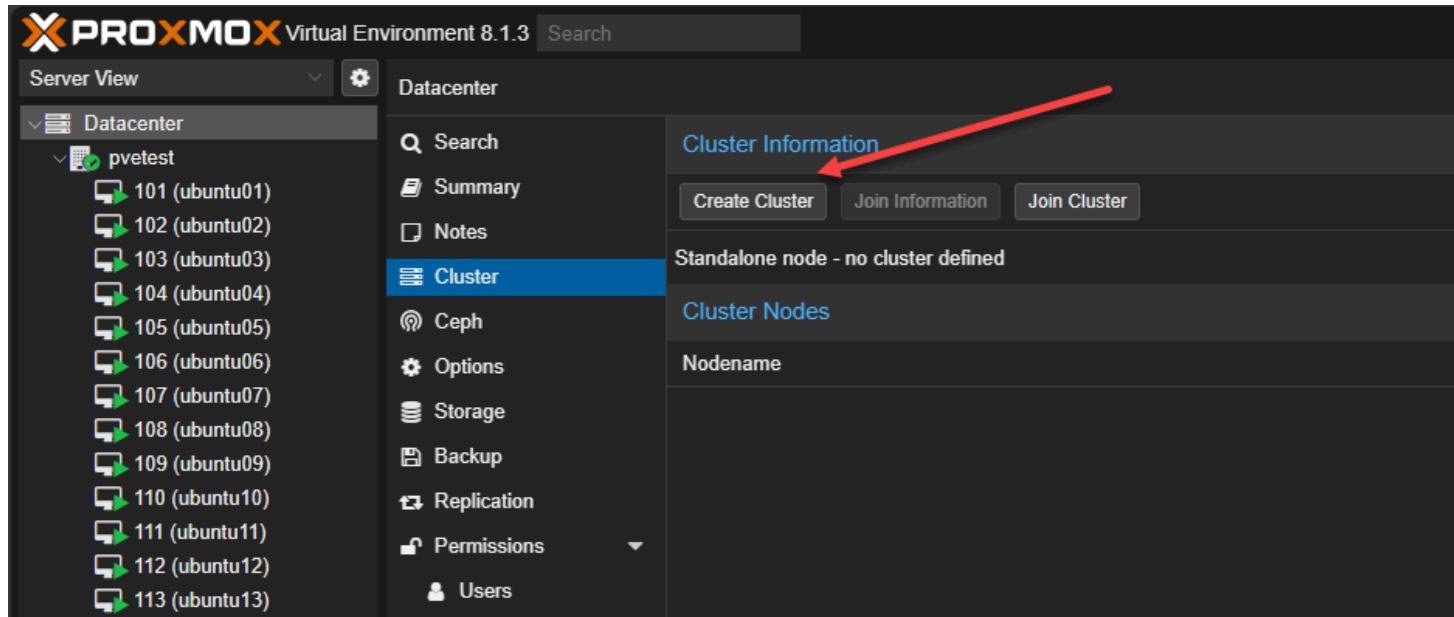
The journey to high availability begins with the [creation of a Proxmox cluster](#). Here, we'll guide you through the process of joining multiple nodes to form a unified system. Each Proxmox node will contribute to the cluster's overall strength, offering

redundancy and reliability.

Key Steps in Creating a Proxmox Cluster

- Choosing Cluster Nodes:** Selecting the right Proxmox nodes is the first step. Ensure that each node is equipped with redundant network hardware and adequate storage capabilities.
- Configuring the Network:** A stable cluster network is vital. We'll explore how to set up a network that supports HA, focusing on IP address configuration and avoiding common pitfalls like split brain scenarios.
- Cluster Formation:** The process of forming a cluster involves initializing the first node and then adding additional nodes with the **join cluster** function. We'll walk you through the commands and steps necessary to create your cluster.

Let's look at screenshots of creating a Proxmox cluster and joining nodes to the cluster. Navigate to **Datacenter > Cluster > Create Cluster**.



Beginning to create the cluster

This will launch the **Create Cluster** dialog box. Name your cluster. It will default to your primary network link. You can **Add** links as a failover. Click **Create**.



Create the new cluster

The task will begin and should complete successfully.

Task viewer: Create Cluster

Output Status

Stop Download

```
Corosync Cluster Engine Authentication key generator.  
Gathering 2048 bits for key from /dev/urandom.  
Writing corosync key to /etc/corosync/authkey.  
Writing corosync config to /etc/pve/corosync.conf  
Restart corosync and cluster filesystem  
TASK OK
```

The cluster creation completes successfully

PROXMOX Virtual Environment 8.1.3

Server View Datacenter

v Datacenter pvetest 101 (ubuntu01) 102 (ubuntu02) 103 (ubuntu03) 104 (ubuntu04) 105 (ubuntu05) 106 (ubuntu06) 107 (ubuntu07) 108 (ubuntu08) 109 (ubuntu09) 110 (ubuntu10) 111 (ubuntu11) 112 (ubuntu12) 113 (ubuntu13) 114 (ubuntu14) 115 (ubuntu15) 116 (ubuntu16)

Search Summary Notes Cluster Ceph Options Storage Backup Replication Permissions Users API Tokens Two Factor

Cluster Information

Create Cluster Join Information Join Cluster

Cluster Name: TestCluster

Cluster Nodes

Nodename

pvetest

Get the cluster join information

You can then click the **Cluster join information** to display the information needed to join the cluster for the other nodes. You can click the copy information button to easily copy the join information to the clipboard.

Cluster Join Information

Copy the Join Information here and use it on the node you want to add.

IP Address:	10.1.149.52
Fingerprint:	14:F7:03:B0:B1:89:15:7A:30:8C:08:64:88:A2:37:A4:8E:C0:EC:77:8B:5C:ED:31:F4:17:79:4D:82:52:9B:25
Join Information:	eyJpcEFkZHJlc3MiOlxMC4xLjE0OS41MilsImZpbmdlcnByaW50IjoiMTQ6Rjc6MDM6QjA6QjE6ODk6MTU6N0E6MzA6OEM6MDg6NjQ6ODg6QTl6Mzc6QTQ6OEU6QzA6RUM6Nzc6OEI6NUM6RUQ6MzE6RjQ6MTc6Nzk6NEQ6ODl6NTl6OUI6MjUlCJwZWVvTGlua3MiOnsiMCi6ljEwLjEuMTQ5LjUyIn0slnJpbmdfYWRkcil6WylxMC4xLjE0QS41Mi.IdI.C.I0h3RlhSI6ev.lncF927X.lzaW9ulniaXR2NC.02liwiaW507X.lmYVNllin7liAiOnsibGlua251hW.llcrl6li

Copy Information

Viewing the join information

On the target node, we can click the **Join Cluster** button under the **Datacenter > Cluster** menu.

The screenshot shows the Proxmox VE 8.1.3 interface. The top navigation bar has the title "PROXMOX Virtual Environment 8.1.3". Below it, the left sidebar shows "Server View" and "Datacenter". Under "Datacenter", there is a tree view with "Datacenter" expanded, showing "pve" as a child. The "Cluster" item in the list is highlighted with a blue selection bar. To the right, the main panel displays "Cluster Information" with tabs for "Create Cluster", "Join Information", and "Join Cluster". The "Join Cluster" tab is highlighted with a blue bar. Below the tabs, the text "Standalone node - no cluster defined" is shown. A red arrow points from the bottom right towards the "Join Cluster" tab.

Join the cluster from another node

Now we can use the join information from our first node in the cluster to join additional nodes to the cluster. You will also need the root password of the cluster node to join the other Proxmox nodes.

Cluster Join

Assisted join: Paste encoded cluster join information and enter password.

Information: EQ6ODI6NTI6OUI6MjUiLCJwZWVytGlua3MiOnsiMCi6ljEwLjEuMTQ5LjUyIn0sInJpbmdfYWRkcil6WylxMC4xLjE0OS41MiJdLCJ0b3RlbSl6eyJpcF92ZXJzaW9uljoiaXB2NC02liwiaW50ZXJmYWNlIjp7ijAiOnsibGlua251bWJlcil6ljAifX0slnNIY2F1dGgiOjJvbilsInZlcnPb24iOilyliwibGlua19tb2RlljoicGFzc2lZSIsImNsdXN0ZXJfbmFtZSj6llRlc3RDdbHVzdGVyliwiY29uZmlnX3ZlcnPb24iOilxln19

Peer Address: 10.1.149.52 **Password:** 

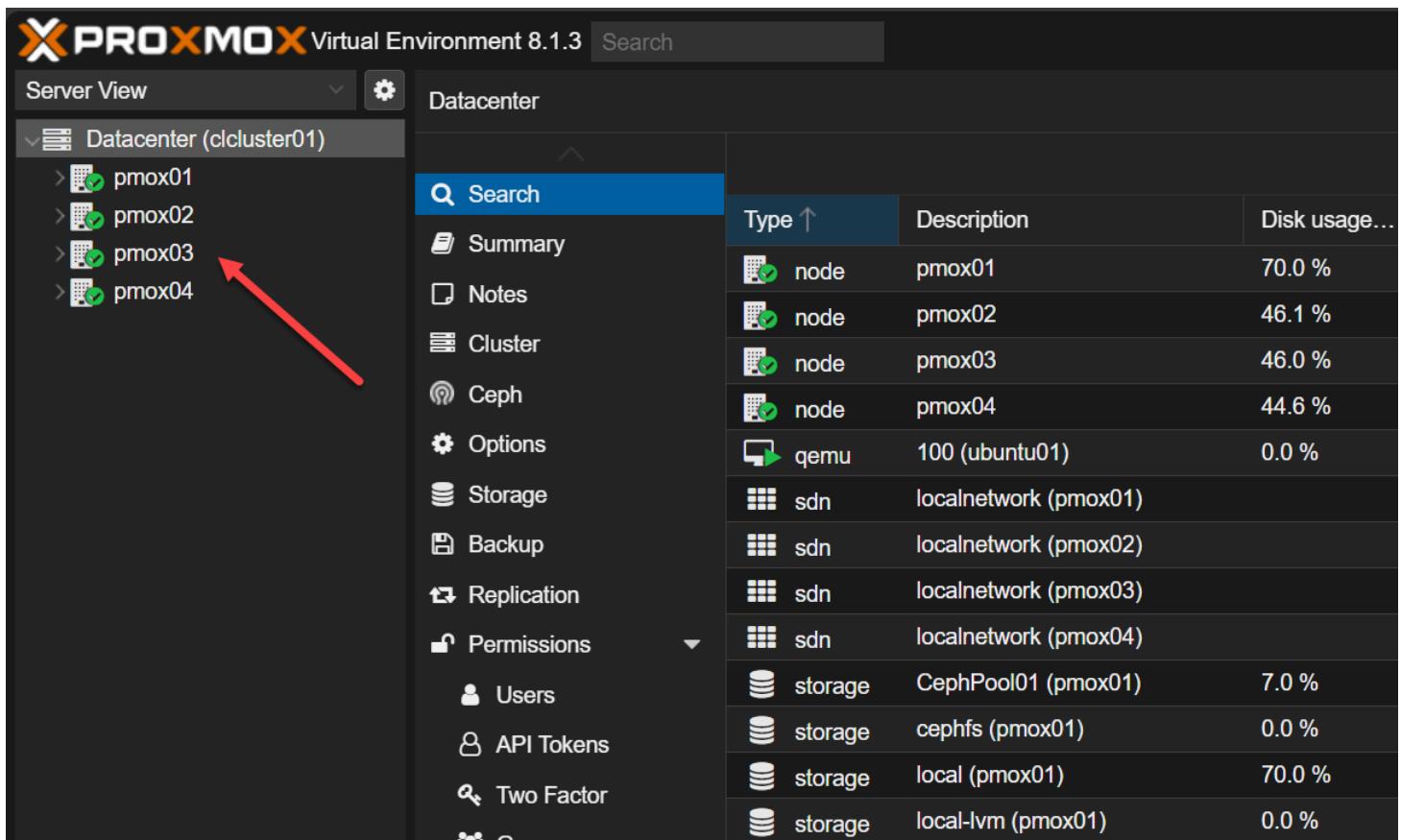
Fingerprint: 14:F7:03:B0:B1:89:15:7A:30:8C:08:64:88:A2:37:A4:8E:C0:EC:77:8B:5C:ED:31:F4:17:79:4D:82:52:9B:25

Cluster Network: Link: 0 IP resolved by node's hostname peer's link address: 10.1.149.52

Entering the join information

Below, I have created a cluster with 4 Proxmox hosts running Ceph shared storage.



The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the 'Server View' sidebar shows a tree structure with 'Datacenter (clcluster01)' expanded, revealing four nodes: pmox01, pmox02, pmox03, and pmox04. A red arrow points from the text 'A 4 node proxmox ve cluster' to this list. The main panel is titled 'Datacenter' and contains a search bar and a list of resources. The list includes:

Type ↑	Description	Disk usage...
node	pmox01	70.0 %
node	pmox02	46.1 %
node	pmox03	46.0 %
node	pmox04	44.6 %
qemu	100 (ubuntu01)	0.0 %
sdn	localnetwork (pmox01)	
sdn	localnetwork (pmox02)	
sdn	localnetwork (pmox03)	
sdn	localnetwork (pmox04)	
storage	CephPool01 (pmox01)	7.0 %
storage	cephfs (pmox01)	0.0 %
storage	local (pmox01)	70.0 %
storage	local-lvm (pmox01)	0.0 %

A 4 node proxmox ve cluster

Configuring Virtual machine HA

Once your Proxmox cluster is operational, the next step is configuring HA for your virtual machines. The [Virtual Machine](#) HA config provides automation for restarting a VM that is owned by a failed host, on a healthy host. This involves setting up shared storage, understanding HA manager, and defining HA groups administration.

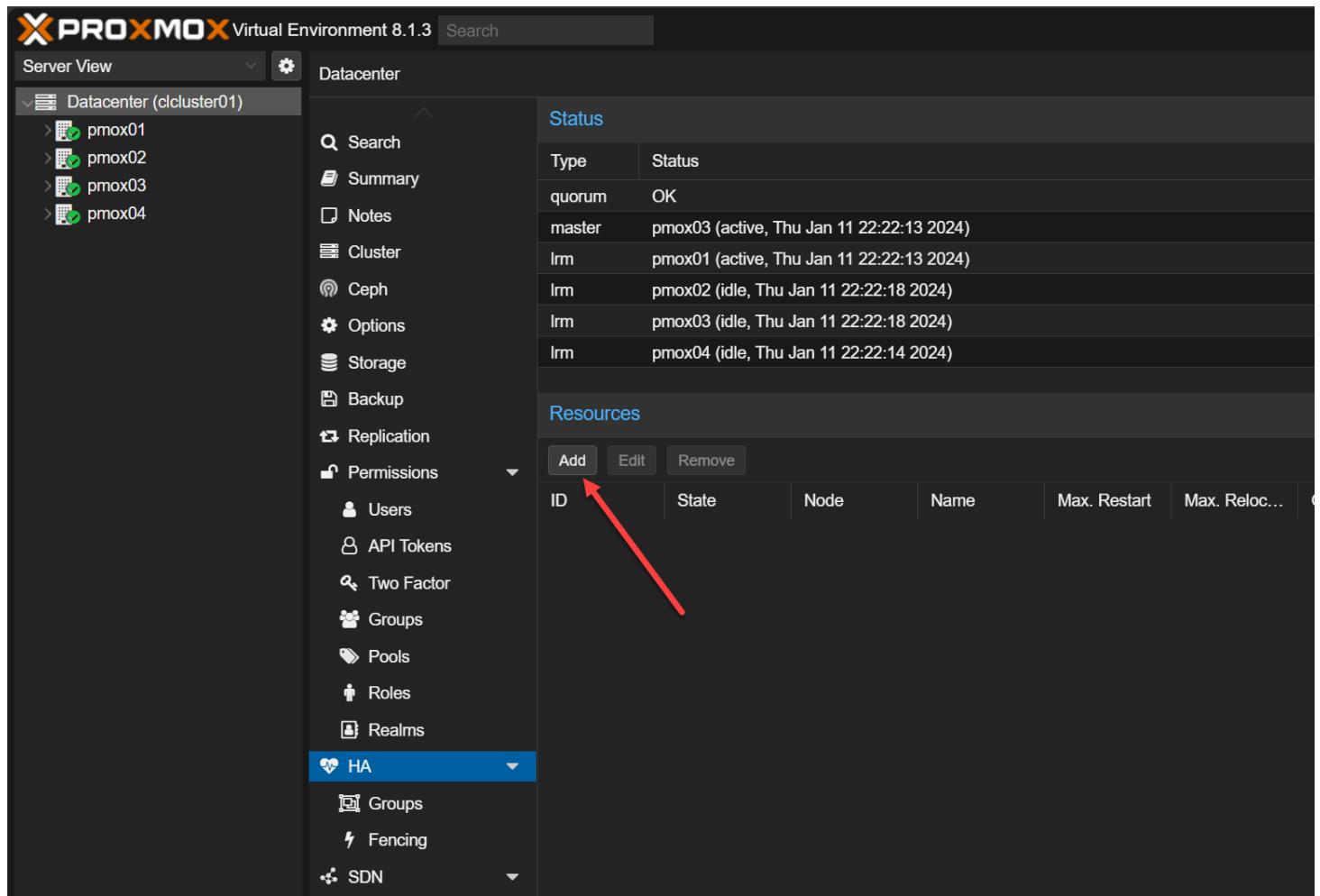


High Availability Setup Requirements

When provisioning Proxmox high availability, there are a number of infrastructure requirements.

- 1. Shared Storage Configuration:** For VMs to migrate seamlessly between nodes, shared storage is a necessity as we have mentioned above so data does not have to move during a failover.
- 2. The HA Manager:** Proxmox's HA manager plays a critical role in monitoring and managing the state of VMs across the cluster. It works like an automated sysadmin. After you configure the resources it should oversee, such as VMs and containers, the ha-manager [monitors their performance](#) and manages the failover of services to another node if errors occur. Also, the ha-manager can process regular user commands, including starting, stopping, relocating, and migrating services.
- 3. Defining HA Groups (optional) :** HA groups determine how VMs are distributed across the cluster.

Let's look at a [basic example of configuring](#) a single VM for high availability. Below, in the web interface we have navigated to the **Datacenter > HA > Resources > Add** button. Click the Add button.



The screenshot shows the Proxmox VE 8.1.3 web interface. The left sidebar shows a tree view of 'Datacenter (clcluster01)' containing nodes 'pmox01', 'pmox02', 'pmox03', and 'pmox04'. The main panel has two sections: 'Status' and 'Resources'. The 'Status' section lists the quorum as 'OK' and details for four hosts: 'pmox03' (master, active), 'pmox01' (active), 'pmox02' (idle), and 'pmox04' (idle). The 'Resources' section has a table with columns 'ID', 'State', 'Node', 'Name', 'Max. Restart', and 'Max. Reloc...'. At the top of this table is a row with buttons 'Add', 'Edit', and 'Remove'. A red arrow points from the text 'Select the VM ID to create the HA resource.' to the 'Add' button.

ID	State	Node	Name	Max. Restart	Max. Reloc...

Add resources for ha

Select the VM ID to create the HA resource.

Add: Resource: Container/Virtual Machine

VM:	100	Group:	
Max. Restart:	1	Request State:	started
Max. Relocate:	1	Comment:	
		Help	Add

Add the resource id for the vm you want to be ha

This will configure a service for the VM to make the VM highly available. The service will start and enter the **started** state. Now, we have the VM configured for HA.

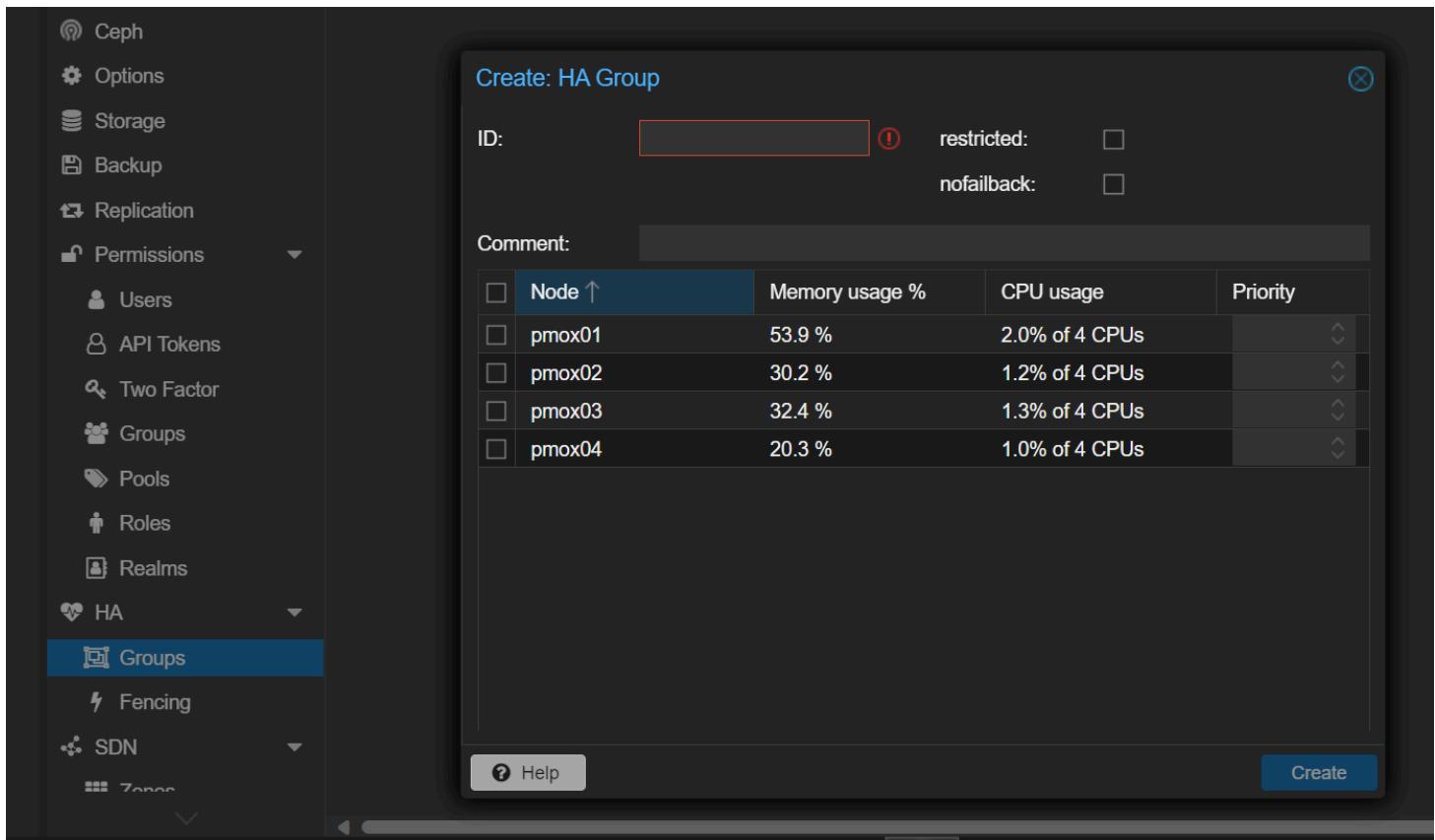
The screenshot shows the Proxmox VE web interface with the 'HA' tab selected in the sidebar. The main area displays a table of resources under the 'Resources' section. One row in the table represents a VM with the following details:

ID	State	Node	Name	Max. Restart	Max. Reloc...	Group
vm:100	started	pmox01	ubuntu01	1	1	

The ha service for the vm is started

Configuring HA groups (optional)

The HA group [configuration file /etc/pve/ha/groups.cfg defines groups of cluster](#) nodes and how resources are spread across the nodes in the cluster. You can configure resources to only run on the members of a certain group. You can use this to give priority to certain VMs, on certain hosts. Below is the **Create HA Group** configuration dialog box.



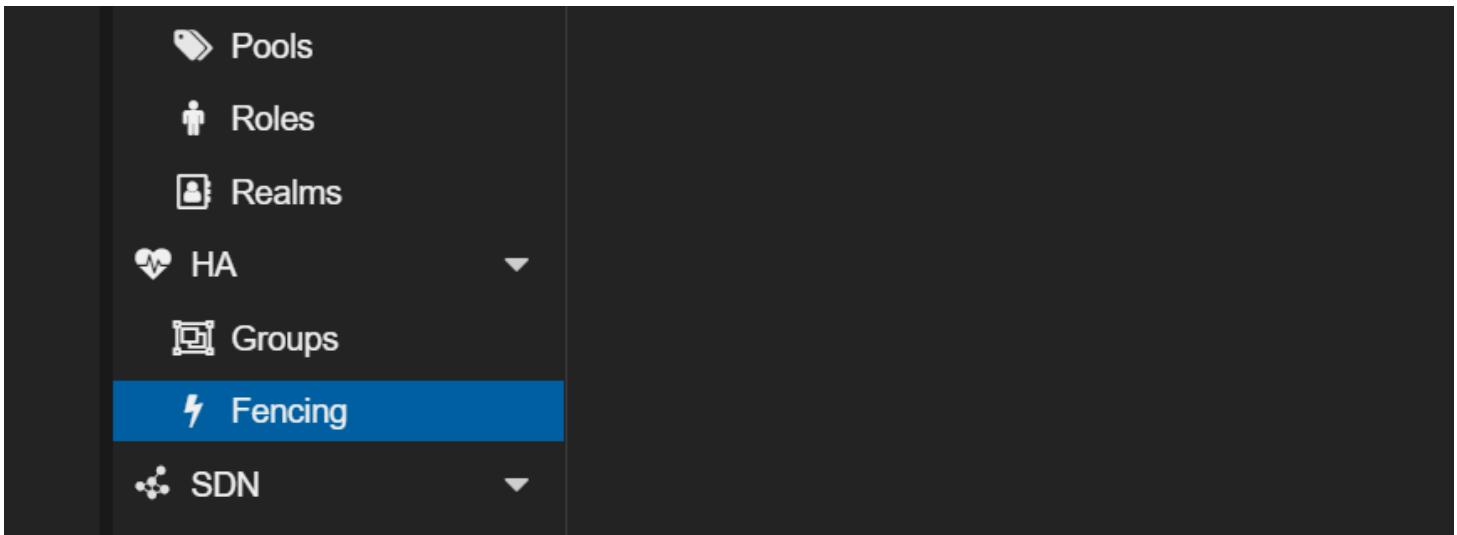
Create an ha group for resources

Fencing device configuration

Fencing is important for managing node failures in Proxmox VE. When a host goes down and is completely offline, it prevents resource duplication during recovery. Without fencing, resources could run on multiple nodes simultaneously which can corrupt data.

Unfenced nodes can access shared resources, posing a risk. For instance, a VM on an unfenced node might still write to shared storage even if it's unreachable from the public network, causing race conditions and potential data loss if the VM is started elsewhere.

Proxmox VE employs various fencing methods, including traditional ones like power cutoffs and network isolation, as well as self-fencing using watchdog timers. These timers, [integral in critical systems](#), reset regularly to prevent system malfunctions. If a malfunction occurs, the timer triggers a server reboot. Proxmox VE utilizes built-in hardware watchdogs on modern servers or falls back to the Linux Kernel softdog when necessary.



Fencing configuration in proxmox ve

Now, I simulated a failure of the Proxmox host by disconnecting the network connection. The pings to the VM start timing out.

```
Reply from 10.1.149.167: bytes=32 time<1ms TTL=64
Request timed out.
```

Virtual machine going down after a failed proxmox host

The host is now taking down the VM resource.

The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the navigation bar is visible with 'Server View' selected. Under 'Datacenter', there is a tree view with 'Datacenter (clcluster01)' expanded, showing three hosts: 'pmox01' (green), 'pmox02' (red), and 'pmox03' (green). To the right of the tree is a sidebar with various management options like Search, Summary, Notes, Cluster, Ceph, Options, Storage, Backup, Replication, Permissions, Users, API Tokens, Two Factor, Groups, Pools, Roles, and Realms. Below the sidebar is a dropdown menu with 'HA' selected. The main content area displays the 'Status' table:

Type	Status
quorum	OK
master	pmox03 (active, Fri Jan 5 13:01:32 2024)
lrm	pmox01 (idle, Fri Jan 5 13:01:37 2024)
lrm	pmox02 (dead?, Fri Jan 5 13:00:09 2024)
lrm	pmox03 (idle, Fri Jan 5 13:01:37 2024)

Below the status table is a 'Resources' section with an 'Add' button and a table:

ID	State	Node	Name	Max. Restart	Max. Reloc...	Group
vm:100	fence	pmox02	ubuntu01	1	1	

Proxmox ha viewing the server as dead

The HA process will restart the VM on a healthy host.

The screenshot shows the Proxmox Virtual Environment 8.1.3 interface. The left sidebar is titled "Server View" and lists three hosts: "Datacenter (clcluster01)", "pmox01", and "pmox02". Under "pmox01", a red arrow points to the entry "100 (ubuntu01)". The main panel is titled "Datacenter" and contains several sections: "Status" (listing "quorum" as OK, "master" as pmox03 active, and "lrm" entries for pmox01, pmox02, and pmox03), "Resources" (with tabs for "Add", "Edit", and "Remove"), and "Permissions" (listing "Users", "API Tokens", "Two Factor", "Groups", "Pools", "Roles", and "Realms"). A dropdown menu under "HA" is open, showing options like "Groups", "Fencing", "SDN", and "Zones".

Virtual machine restarted on a healthy proxmox host

After just a couple of minutes, the VM restarts and starts pinging on a different host.

The proxmox ha virtual machine configuration has brought the vm back up

Best Practices for Cluster Health

- 1. Regular Updates and Backups:** Keeping your Proxmox servers and VMs up-to-date is critical. High availability is not a replacement for VM and container backup. Always protect your data with something like Proxmox Backup Server.
 - 2. Monitoring Tools and Techniques:** Proxmox has several tools for monitoring the health and performance of your cluster. Keep a [check on your cluster health](#). Make sure monitor your nodes from the GUI and ensure things like shared storage are in a healthy state.
 - 3. Handling Node Failures:** Even with HA, node failures can happen. We'll cover the steps to recover from a failed node and how to ensure minimal impact on your [virtual machines](#).
 - 4. Documentation:** Be sure to document the configuration of the cluster, including IPs, storage configuration, etc.

Rebooting Proxmox Servers running HA

If you want to reboot a Proxmox server for maintenance or other that is part of an HA cluster, you need to stop the following service on the node, either from the [command line or GUI](#):

```
/etc/init.d/rgmanager stop
```

Wrapping Up

Proxmox virtualization has some great features, including high-availability configuration for [virtual machines](#). In this article, we have considered the configuration of a high-availability Proxmox VE cluster and then configuring high availability for VMs. In the comments, let me know if you are running a Proxmox VE cluster, what type of storage you are using, and any other details you would like to share.

Proxmox firewall setup and configuration

The Proxmox firewall is a powerful part of the Proxmox virtualization solution and can do many things that admins can use to their advantage, such as controlling traffic and making sure certain things like allowing clients or hosts to connect to certain VMs or otherwise.

Proxmox Firewall Rules Configuration

Proxmox firewall is based on the Linux **iptables** firewall. IPtables filters network traffic at the hypervisor layer. This allows filtering based on the normal firewall capabilities on things like IPs, ports, and those types of network objects.

You can manage the Proxmox firewall using the Proxmox web interface (web GUI) or command-line interface (CLI). These can be used to [configure firewall rules and implement cluster-wide firewall configuration in your Proxmox cluster](#).

Zones configuration

You can divide the firewall into zones. Zones combine network interfaces and IP addresses. By default, notice the four zones available in Proxmox VE.

1. **Input** – handles incoming traffic from external networks
2. **Output** – handles outgoing traffic to external networks
3. **Forward** – handles traffic between [virtual machines](#) and containers
4. **Host** – handles traffic to and from the Proxmox host

Cluster Wide Firewall Rules

With clustered Proxmox configurations, you can have firewall rules to apply to all nodes in the cluster. This is done by configuring the underlying iptables rules automatically and using the same firewall configuration files on all nodes. You can also configure a central firewall solution for the entire cluster by creating a firewall cluster.

Proxmox VE Firewall Zones

To manage the firewall, you need to enable the firewall service. Once enabled, you can configure the firewall zones using the web interface or the command line.

You can also assign IP addresses to zones and create firewall rules that allow or block traffic based on the zone.

Enabling the Firewalls

The Proxmox firewall is disabled by default. To enable the firewall service, you can use the following command on the CLI:

```
pve-firewall enable
```

This will start the firewall service and load the firewall configuration files.

Ports used by Proxmox

There are a few admin ports to be aware of when working with the Proxmox firewall. These include the well-known ports of SSH, HTTP, and VNC. These are open out of the fox, but you can create firewall rules to block access to these ports if you want to do that for security purposes.

Firewall Rules Configuration Direction

You need to specify the direction of the traffic you want to filter with the [Proxmox firewall configuration](#). You can choose to filter incoming traffic, outgoing traffic, or both.

The screenshot shows the Proxmox VE web interface. On the left, there's a sidebar with various management options like Datacenter, Cluster, Ceph, Options, Storage, Backup, Replication, Permissions, Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms, HA, ACME, and Firewall. The 'Firewall' option is highlighted with a red box and has a dropdown menu with Options, Security Group, Alias, IPSet, Metric Server, and Support. The main panel is titled 'Datacenter' and shows a table for firewall rules. The table has columns for On, Type, Action, Macro, Interface, Protocol, and Source. At the top of the table, there are buttons for Add, Copy, Insert: Security Group, Remove, and Edit. A red arrow points to the 'Add' button. Below the table, there's a form for adding a new rule with fields for Direction (set to 'in'), Action (set to 'ACCEPT'), Enable (unchecked), Interface, Source, Destination, Comment, and Log level (set to 'nolog'). There are also 'Advanced' and 'Add' buttons at the bottom of the form.

Enable the Firewall Service from the Command Line

You can enable the Proxmox [firewall service from the command line](#) with the following command:

```
systemctl enable pve-firewall.service
```

This will start the firewall service on [boot and load the firewall configuration files](#).

Proxmox VE Firewall Configurations via Files

You can also configure the Proxmox firewall using configuration files. These files are located in the directory found here:

```
/etc/pve/firewall
```

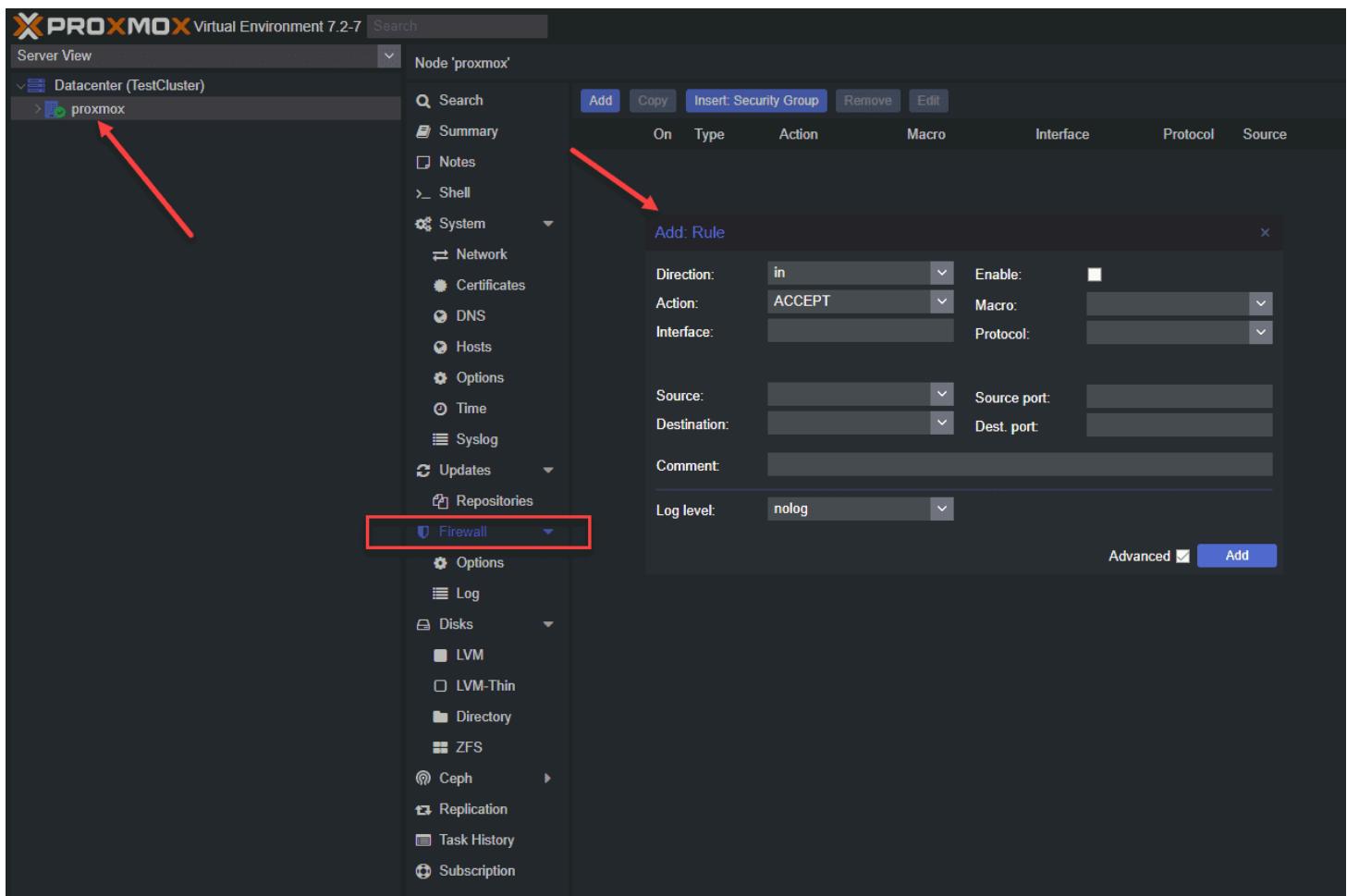
Enabling the Firewall for VMs and Containers

The firewall is disabled by default for virtual machines and containers. However, you can enable the firewall service for a VM or container. To do this, you must add a firewall configuration file to that VM or container's directory:

```
/etc/pve/firewall
```

Host, VMs, and Containers Configuration Files

Below is a look at host-specific firewall rules.



See the Generated IPTables Rules

To view the underlying iptables rules generated by the Proxmox firewall:

```
iptables -L
```

This will display the current iptables rules managed by the Proxmox firewall service.

Check VM Network Device

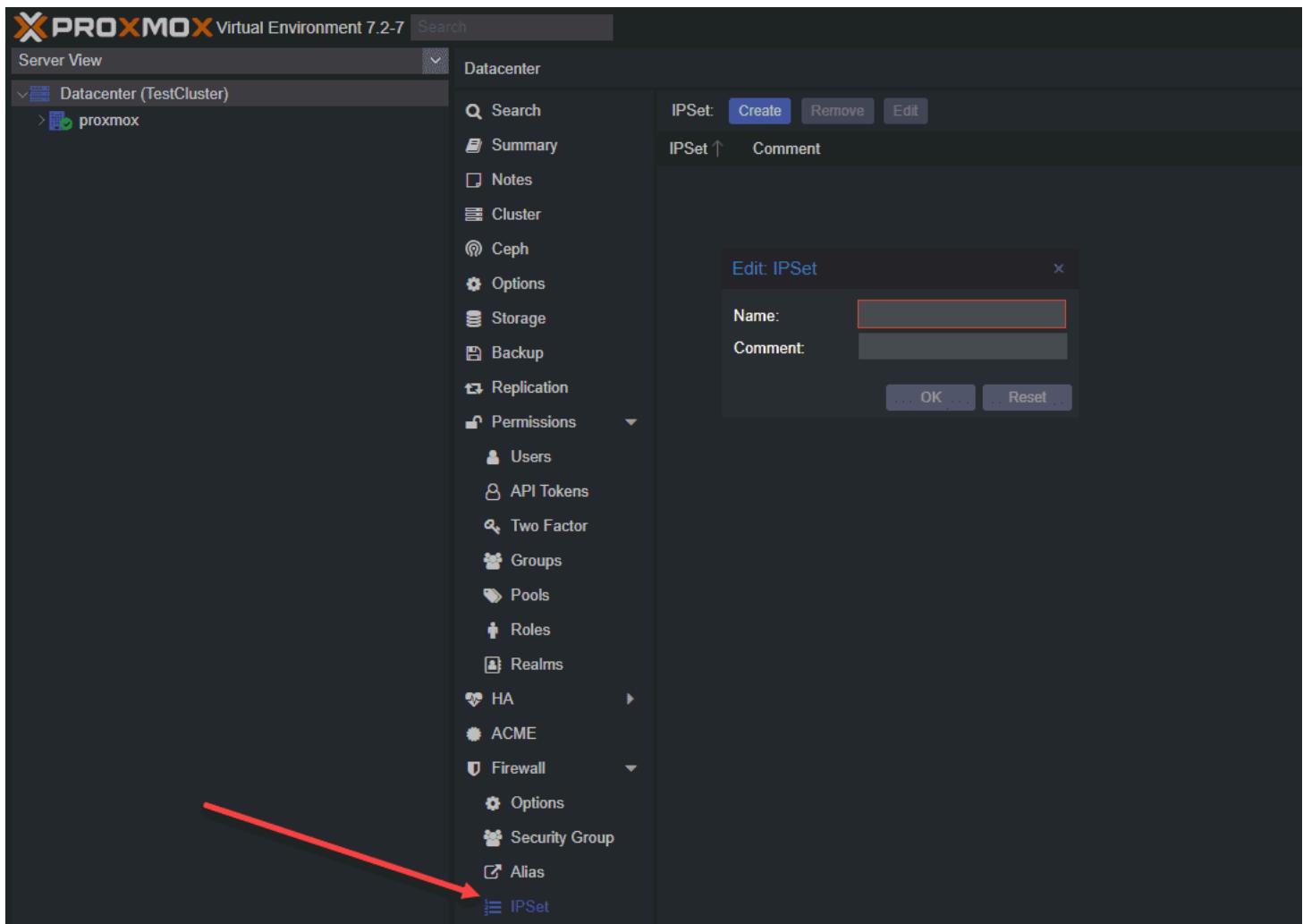
You need to know the name of the virtual device when you configure Proxmox firewall rules to filter traffic for a Proxmox virtual machine. You can find this by using the following command:

```
qm config <VM ID> | grep net
```

This will display the name of the virtual network device used by the VM.

IP Sets

IP sets define a set of IP addresses that can be used in firewall rules.



Default Firewall Rules

A set of default firewall rules out of the box allows incoming and outgoing traffic for certain services. These include traffic types such as SSH and HTTP. You can view the default firewall rules using the following command:

```
iptables -L
```

```
root@proxmox:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@proxmox:~#
```

Standard IP Alias local network

An example of the default IP alias is the local_network standard IP alias defined in the [Proxmox](#) firewall configuration files. It represents the IP addresses assigned to the Proxmox host and is used in firewall rules to allow traffic between the host and virtual machines/containers.

Adding Security Groups and Rules

Security groups and IP aliases can be used in the Proxmox firewall configuration files. These can then be used in firewall rules to allow or block traffic based on the group or alias.

Note the following to define security groups and IP aliases using the following syntax in the configuration files:

```
group <group name> { <ip addresses> }
```

```
alias <alias name> { <ip addresses> }
```

Below is a look at creating security group-based firewall rules.

The screenshot shows the Proxmox Web Interface with the 'Datacenter' tab selected. On the left, there is a sidebar with various navigation items. In the center, a modal dialog titled 'Add: Rule' is open. The dialog has fields for 'Security Group' (set to 'testgroup'), 'Enable' (unchecked), 'Interface' (empty), and 'Comment' (empty). At the bottom right of the dialog is a blue 'Add' button. A red arrow points from the 'Firewall' item in the sidebar to the 'Add' button in the dialog.

Proxmox Firewall CLI Commands to Know

Several CLI commands are useful when configuring the Proxmox firewall:

1. **pve-firewall enable** – Enables the firewall service
2. **pve-firewall disable** – Disables the firewall service
3. **pve-firewall status** – Displays the current status of the firewall service
4. **pve-firewall reload** – Reloads the firewall configuration files
5. **pve-firewall log** – Displays the firewall log

Adding a Proxmox Firewall Rule

You can add a Proxmox Firewall Rule using the command:

```
iptables -A <zone> -p <protocol> --dport <port> -s <source address> -d <destination address> -j <action>
```

PVE Firewall Status

You can also check the status of the Proxmox PVE firewall service using the command:

```
pve-firewall status
```

```
USAGE: pve-firewall <COMMAND> [ARGS] [OPTIONS]

      pve-firewall help [<extra-args>] [OPTIONS]

      pve-firewall compile
      pve-firewall localnet
      pve-firewall restart
      pve-firewall simulate [OPTIONS]
      pve-firewall start [OPTIONS]
      pve-firewall status
      pve-firewall stop
root@proxmox:~#
```

Define Rules

The syntax for adding a firewall rule is as follows with the command that admins can run:

```
iptables -A <zone> -p <protocol> --dport <port> -s <source address> -d <destination address> -j <action>
```

PVE Firewall Stop

To stop the Proxmox firewall service use the command:

```
pve-firewall stop
```

CLI Commands

Several CLI commands can be used to manage the Proxmox firewall service, such as

```
pve-firewall enable
pve-firewall disable
pve-firewall status
pve-firewall reload
```

Remote IPs

You can manage access from remote IP addresses to bolster security. You can configure firewall rules for remote IPs using the

remote.fw file located in the directory:

```
/etc/pve/firewall
```

Cluster Specific Firewall

Using a Proxmox cluster, you can configure a cluster-specific firewall using the `/etc/pve/firewall/cluster.fw` configuration file.

Corosync Cluster Traffic

You can manage traffic for Corosync cluster traffic. You can configure firewall rules for Corosync cluster traffic using the file `corosync.fw` located in the directory:

`/etc/pve/firewall`

HTTP Traffic

You can also filter HTTP traffic in your Proxmox environment. You can configure firewall rules for HTTP connections using the file `http.fw` located in the directory:

`/etc/pve/firewall`

Create Rules

When creating a firewall rule you need to edit the appropriate configuration file in the `/etc/pve/firewall` directory. The syntax for creating a firewall rule is as follows:

```
iptables -A <zone> -p <protocol> --dport <port> -s <source address> -d <destination address> -j <action>
```

Wrapping up

The Proxmox firewall is a tool that admins can use if they need to filter or block or otherwise channel the traffic coming into the Proxmox VE host or virtual machines. Security often starts by scoping down which clients and servers can connect to and from other clients and servers. As shown, it is easy to work with the Proxmox firewall and filter traffic as needed. Let me know in the comments if you guys work with the Proxmox firewall to scope traffic down or are you using something else.

Proxmox Container vs VM features and configuration

Proxmox is an extremely versatile hypervisor that provides the ability to run a Proxmox Container vs VM as part of the native functionality built into the platform. This is a great way to have the best of both worlds. It allows for solving multiple applications challenges and multiple purposes with a single hypervisor. You can use these for production or test environments.

Instead of running multiple virtual machines (VM workloads) to host services, you can run the LXC containers on the host system for more efficient environments. Let's explore the topic of [Proxmox containers vs VM](#) instances and see how running virtual machines in Proxmox differs from Proxmox containers.

Take note of my latest Promox posts:

- [Proxmox Management Interface VLAN tagging configuration](#)
- [Proxmox vs ESXi – ultimate comparison 2022](#)
- [Proxmox Create ISO Storage Location – disk space error](#)
- [pfSense Proxmox Install Process and Configuration](#)
- [Proxmox Update No Subscription Repository Configuration](#)

What is the difference between LXC containers and Docker containers?

First of all, many will recognize that we are describing Proxmox VE containers as LXC containers and not Docker. What is the difference between the two? LXC containers are known as Linux Containers and are an OS-level virtualization technology.

It enables running multiple Linux OS'es on a single LXC host. LXC containers are much smaller than a full virtual machine but often larger than Docker containers.

This can help with the performance of spinning up applications and setup access much more quickly to resources. There are many reasons why one is preferred over the other. However, depending on the use case, one may be the best choice over the other.

Docker is most popular

Docker containers are arguably the most popular container technology used in the enterprise today. They focus on running applications and all their dependencies in a seamless, self-contained way and allow provisioning a single-purpose application environment for running applications.

LXC containers are more like virtual machines

LXC containers are very much like a virtual machine, but significantly lighter weight since it is sharing the host kernel with the LXC host. It does not require the disk space or other resources as full VMs.

LXC containers aim to align with a specific distribution of Linux. However, Docker containers aim to be distro-less and focus on the applications and dependencies. Virtual machines have their own kernel instance as opposed to the shared kernel instance with containers.

Allow hosting multiple applications

With multiple Docker containers, you can host multiple applications on your container host. LXC containers provide the traditional resources you would find in a virtual machine running in the same environment. However, you can't run different operating systems like Windows in an LXC container, only different Linux distributions.

Nesting Docker containers inside LXC containers

One of the really cool things about running LXC [containers on a Proxmox](#) host is you can actually install Docker inside an LXC container. In fact, you can run Kubernetes in a lab environment using LXC containers as your Kubernetes hosts.

Many may not realize that Docker is actually a fork of Linux LXC containers. Both LXC and Docker share the same kernel with the container host.

Proxmox container vs virtual machine

[Proxmox](#), unlike many commercial hypervisors, has the ability out of the box to run containers on top of the hypervisor directly. You can choose to create either a container vs VM.

Container vs VM

A virtual machine can load any operating system you want inside the VM with its kernel instance and provides the best isolation for running a server for resources. Containers share the kernel instance with the physical server Linux instance.

So the container operating system is shared with the host. Both have the hardware abstracted using virtualization technologies. The user does not know they are accessing virtual machines or containers when accessing resources.

Overhead

The overhead of running multiple virtual machines is much more than the overhead of running multiple containers. If users need access to a desktop or desktop resources, virtual machines are needed for this purpose. The speed to provision containers is faster, and the effort involved is generally less involved.

Persistence

Virtual machines are generally considered persistent and have to maintain lifecycle management, etc. Whereas containers offer the ability to have ephemeral resources. The time to boot a container is minimal.

You will see the choice in the menu for Create VM or Create CT on the host system. Again the main difference is you are creating a full virtual machine or an LXC container.

Backups

In terms of backups, you can backup both containers vs VM in Proxmox VE. This is a great option since many solutions allow backing up virtual machines but do not support containers.

Creating new Proxmox containers

Let's look at the configuration [steps to create Proxmox](#) containers and see what configuration is involved. Incidentally, the screens for creating a virtual machine are basically the same. so, we will look at the containers screens since these are probably the least familiar Again, with containers, we are using a virtualization option that shares the same kernel instance with the Proxmox host.

The screenshot shows the Proxmox VE 7.2-11 interface. On the left, the 'Server View' sidebar is open, showing the 'Datacenter' section. Under 'Datacenter', there is a node named 'proxmox'. Two options, 'Create VM' and 'Create CT', are highlighted with a red box. A red arrow points from the 'Create VM' option to the main content area. The main content area displays a table of storage resources:

Type	Description	Disk usage %
lxc	102 (khost01)	9.2 %
lxc	103 (khost02)	9.2 %
lxc	104 (khost03)	9.2 %
node	proxmox	50.8 %
qemu	101 (win2k22clone)	0.0 %
qemu	100 (WindowsServer2022)	
storage	Proxmox-Synology (proxmox)	
storage	isobackups (proxmox)	50.8 %
storage	local (proxmox)	50.8 %
storage	local-lvm (proxmox)	37.3 %
storage	zfs-nvme01 (proxmox)	0.3 %

Creating either a new virtual machine or containers in proxmox

When you choose the New CT option, you will begin the **Create: LXC Container** wizard. Below you will see the first screen has you define:

Node

CTID

Hostname

Privileges

Nesting

Resource Pool

Password

SSH public key

General Tab

This screen helps establish the basics of connectivity, authentication, and a few other data configurations for the container instance.

Create: LXC Container

X

General Template Disks CPU Memory Network DNS Confirm

Node: proxmox

CT ID: 105

Hostname:

Unprivileged container:

Nesting:

Resource Pool:

Password:

Confirm password:

SSH public key:

Load SSH Key File

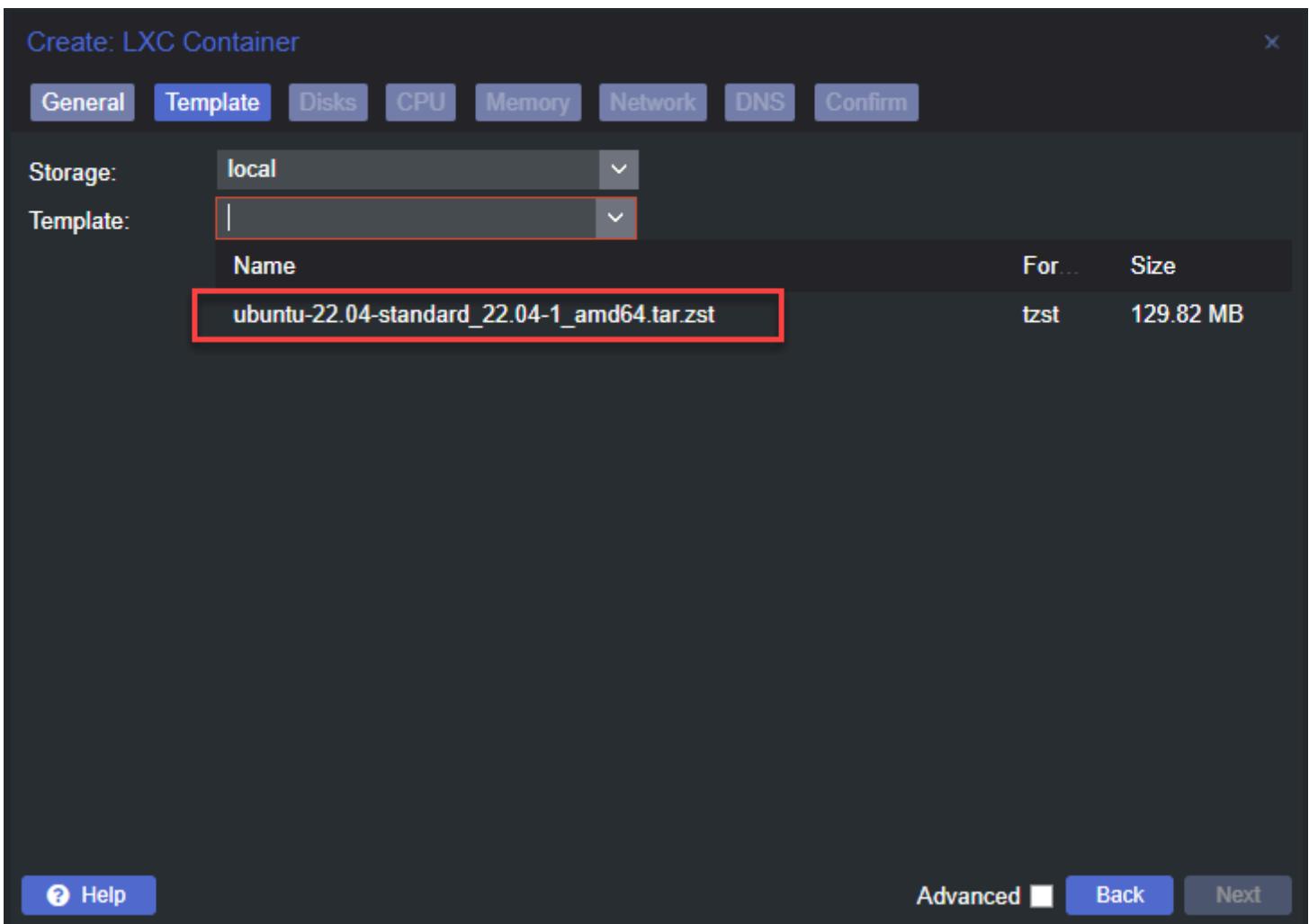
Help Advanced Back Next

The screenshot shows the 'Create: LXC Container' dialog box. The 'General' tab is selected. On the left, there are fields for 'Node' (set to 'proxmox'), 'CT ID' (set to '105'), 'Hostname' (empty), 'Unprivileged container' (checked), and 'Nesting' (checked). On the right, there are fields for 'Resource Pool' (empty), 'Password' (a masked password), 'Confirm password' (a masked password), and 'SSH public key' (empty). A red box highlights the 'Node' and 'CT ID' fields. At the bottom, there are buttons for 'Help', 'Advanced' (highlighted in orange), 'Back', and 'Next'.

Proxmox containers settings on the general tab

Choosing your container template

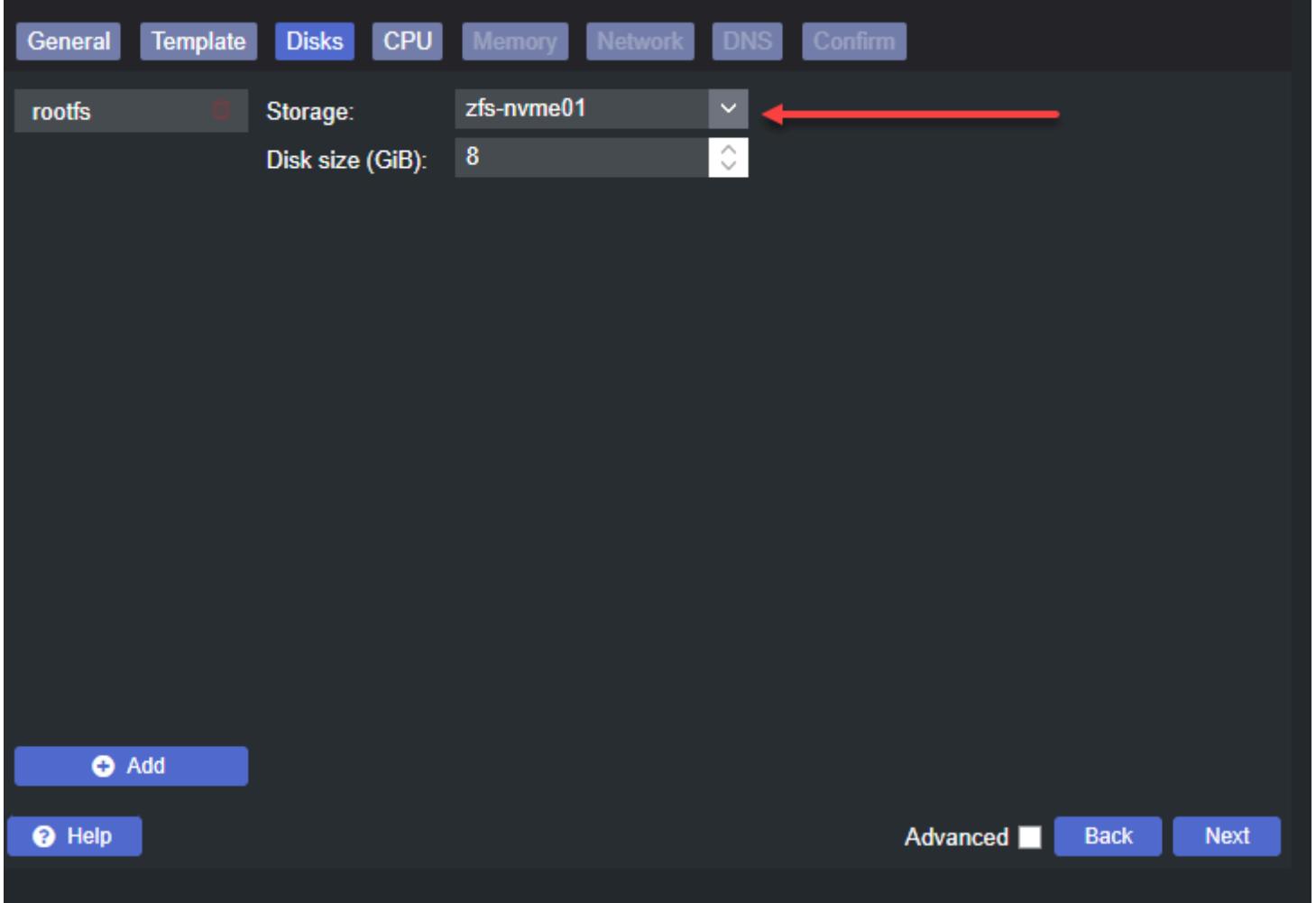
On the next screen, you choose the Proxmox containers template that will be used for spinning up the LXC container. As you can see below, I have pulled down an Ubuntu 22.04 container image to spin up a new system.



Selecting the container template on the proxmox container configuration

Choosing storage

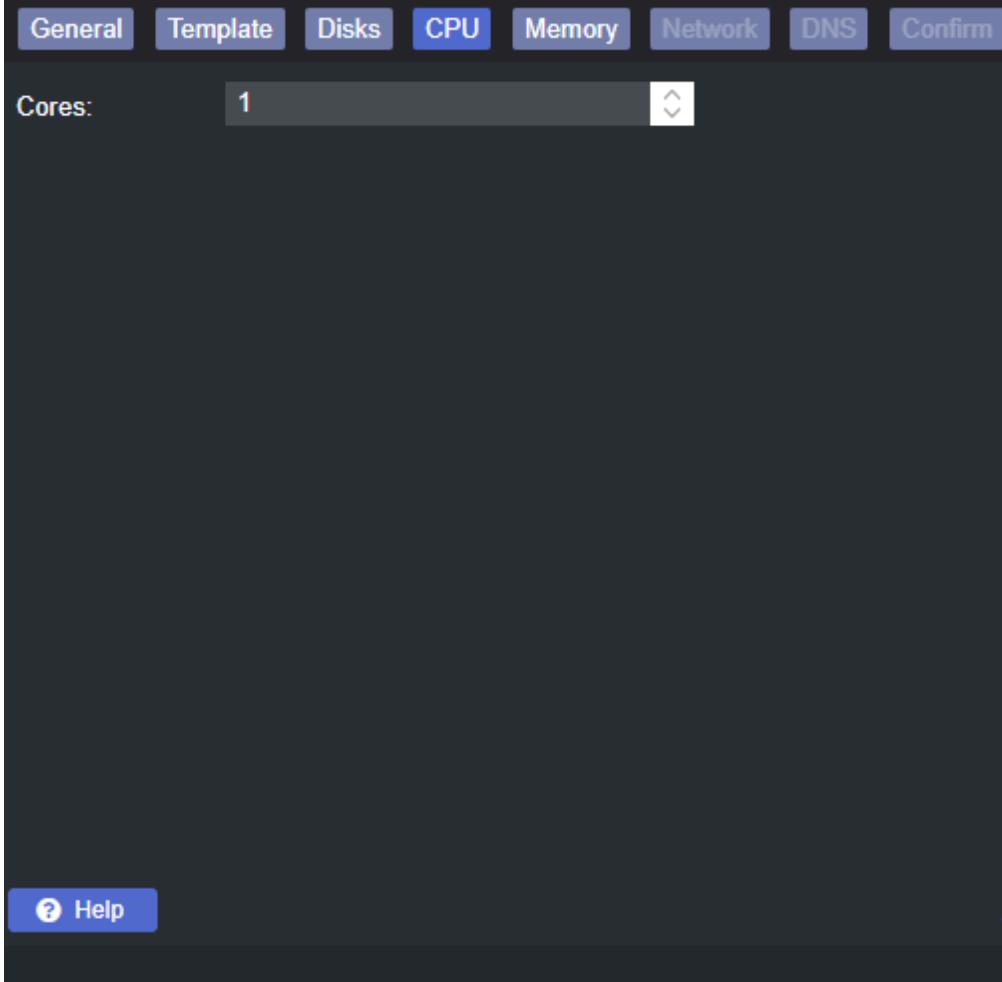
Next, we select the disk storage needed for the LXC container. Below, I have selected the storage for the container file storage using the Proxmox tool.



Configuring proxmox container storage for the new lxc container

Configuring the CPU settings

Next, we select the CPU resources, needed for the container. We can select the core value needed for the new container.



Selecting the cores for the new lxc container

Configuring memory

We need to assign the memory value for the new container in Proxmox.

General Template Disks CPU **Memory** Network DNS Confirm

Memory (MiB): ^ ⌄

Swap (MiB): ^ ⌄

? Help Advanced Back Next

Assigning the memory values to the new lxc container

Network configuration

Now, we create network resources for the new LXC container running in Proxmox. The [Proxmox containers](#) can have all of the normal virtual machines configuration we are used to, such as assigning a VLAN tag, IP address configuration, such as static or DHCP and others as you would any other computer system running on Proxmox VE.

General Template Disks CPU Memory Network DNS Confirm

Name:	eth0	IPv4:	<input type="radio"/> Static <input checked="" type="radio"/> DHCP
MAC address:	auto	IPv4/CIDR:	None
Bridge:	vmbr0	Gateway (IPv4):	
VLAN Tag:	no VLAN	IPv6:	<input type="radio"/> Static <input checked="" type="radio"/> DHCP <input type="radio"/> SLAAC
Rate limit (MB/s):	unlimited	IPv6/CIDR:	None
Firewall:	<input checked="" type="checkbox"/>	Gateway (IPv6):	

[? Help](#) Advanced Back Next

Configuring network resources for the new lxc container

DNS configuration

Going along with the network configuration on the next screen we have the DNS configuration.

General Template Disks CPU Memory Network DNS Confirm

DNS domain:

DNS servers:

Advanced Back

Defining dns settings for lxc container

Confirming the creation of the new LXC container

Finally, we get to the point of finishing out the Proxmox VE configuration. Here we can review the

General	Template	Disks	CPU	Memory	Network	DNS	Confirm
Key ↑	Value						
cores	1						
features	nesting=1						
hostname	testcontainer						
memory	512						
net0	name=eth0,bridge=vmbr0,firewall=1						
nodename	proxmox						
ostemplate	local:vztmpl/ubuntu-22.04-standard_22.04-1_amd64.tar.zst						
pool							
rootfs	zfs-nvme01:8						
swap	512						
unprivileged	1						
vmid	105						

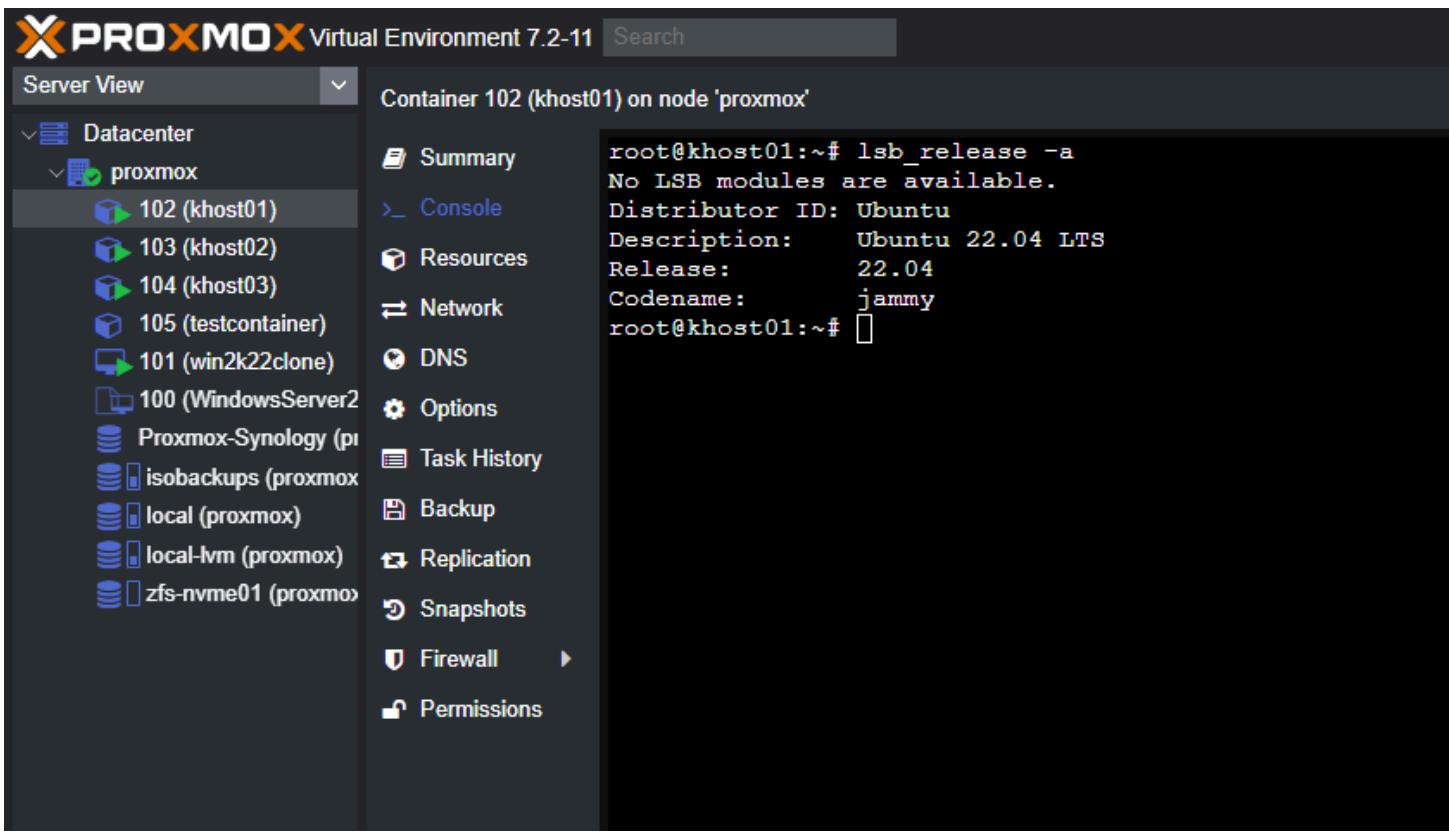
Start after created

Advanced Back

Finishing the proxmox container configuration

Accessing the console of the container for command line access

Below, you can easily access the container's command line from the Proxmox VE web interface.



Command line access with a proxmox server accessing a container

Converting virtual machines and containers to templates

In Proxmox VE, you can convert both virtual machines and containers to templates. Templates are a way to easily save a copy with the configuration included for a virtual machine or a container so these can be quickly spun up from the template.

You can have Windows, Linux, and other operating systems converted to template and easily spin these up for quick deployment from a common mount point.

Proxmox container vs VM FAQs

What are Proxmox containers? [Proxmox containers](#) are LXC containers that are very similar to virtual machines in terms of features and behaviors. These are heavier containers generally speaking than Docker containers. Docker containers focus on applications, whereas LXC containers focus on Linux distributions.

What are Proxmox containers vs VM? Containers vs VM in Proxmox VE provides very robust and diverse capabilities that allow solving many different challenges from a technical and business perspective.

What is the difference between Docker vs. LXC containers? Docker is focused on applications and LXC containers are focused on distributions and more VM-specific functionality.

Wrapping Up

Proxmox has a wide range of features. When looking at Proxmox container vs VM functionality, it covers it all. Using LXC containers you can quickly spin up environments. Virtual Machines allow spinning up isolated environments with their own kernel instance for the most isolation. However, containers are still a secure way to run applications and spin up environments for users to access applications and resources.

Proxmox Containers with Fedora CoreOS Install

I recently looked at the installation of Fedora CoreOS on VMware. In the home lab, many are running Proxmox and maybe more coming up will be switching from VMware over the course of 2024. Let's take a look at the topic of running Proxmox Containers with Fedora CoreOS setup.

Table of contents

- [Proxmox and Fedora CoreOS](#)
 - [Fedora CoreOS](#)
 - [Proxmox](#)
- [Fedora CoreOS install on Proxmox](#)
- [1. Clone the community repo](#)
- [2. Enable snippets on a Proxmox VE storage repository](#)
- [3. Run the included shell script](#)
- [4. Configure the cloud-init settings for the created template](#)
- [5. Clone the template VM to a new Fedora CoreOS virtual machine](#)
- [Wrapping up Proxmox containers Fedora CoreOS install](#)

Proxmox and Fedora CoreOS

Combining an excellent hypervisor for virtualization and an operating system platform that is **purpose-built for containerization and Kubernetes**, is a great combination. We can do this by running Fedora CoreOS distribution on top of Proxmox VE with KVM for the purpose of running containers.

Fedora CoreOS

While you can run LXC container (Linux containers) configurations and container template machines inside of Proxmox natively, many developers and DevOps guys need access to Docker containers. In the [home lab](#), most solutions you want to self-host are readily available as Docker containers also without running full virtual machine instances with full operating systems. So, running Docker is a great way to have access to these solutions.

Fedora CoreOS has as its focus, containerized infrastructure. If you look at the documentation, It offers an automatically updating, minimal, container-centric operating system that natively runs Docker, Podman, and can run Kubernetes as well also, with good support across the board. It is also **immutable which means it has [security enhancements](#)** for customers running it for their container cluster.

So if you are looking for a clean, secure, and immutable OS to run your containers on top of Proxmox, CoreOS is a great solution!

Proxmox

Running it on top of Proxmox has other benefits such as the ability to run Proxmox Backup Server solution to [backup the host virtual machines](#). It has [powerful networking](#) and monitoring. Businesses can even choose to have enterprise support and [home labbers](#) can become a member of the Proxmox support forum (for search forums threads, requests, and share things with others in the home forums such as issues and troubleshooting) and access to other Proxmox solutions, like Proxmox Mail Gateway.

Fedora CoreOS install on Proxmox

Let's look at the Fedora CoreOS installation on Proxmox VE and see what steps are involved. There is actually a really great [community](#) project that will allow getting up and running with Fedora CoreOS on Proxmox. We will take a look at this below. In addition, Fedora CoreOS can be installed on [bare metal](#) using a live ISO installation as well as OVA appliance for VMware.

Note the steps we will cover:

1. Clone the community repo
2. Enable snippets on a Proxmox VE storage repository
3. Run the included shell script

4. Configure **cloud-init** options for the created template
5. Clone the template VM to a new Fedora CoreOS virtual machine

1. Clone the community repo

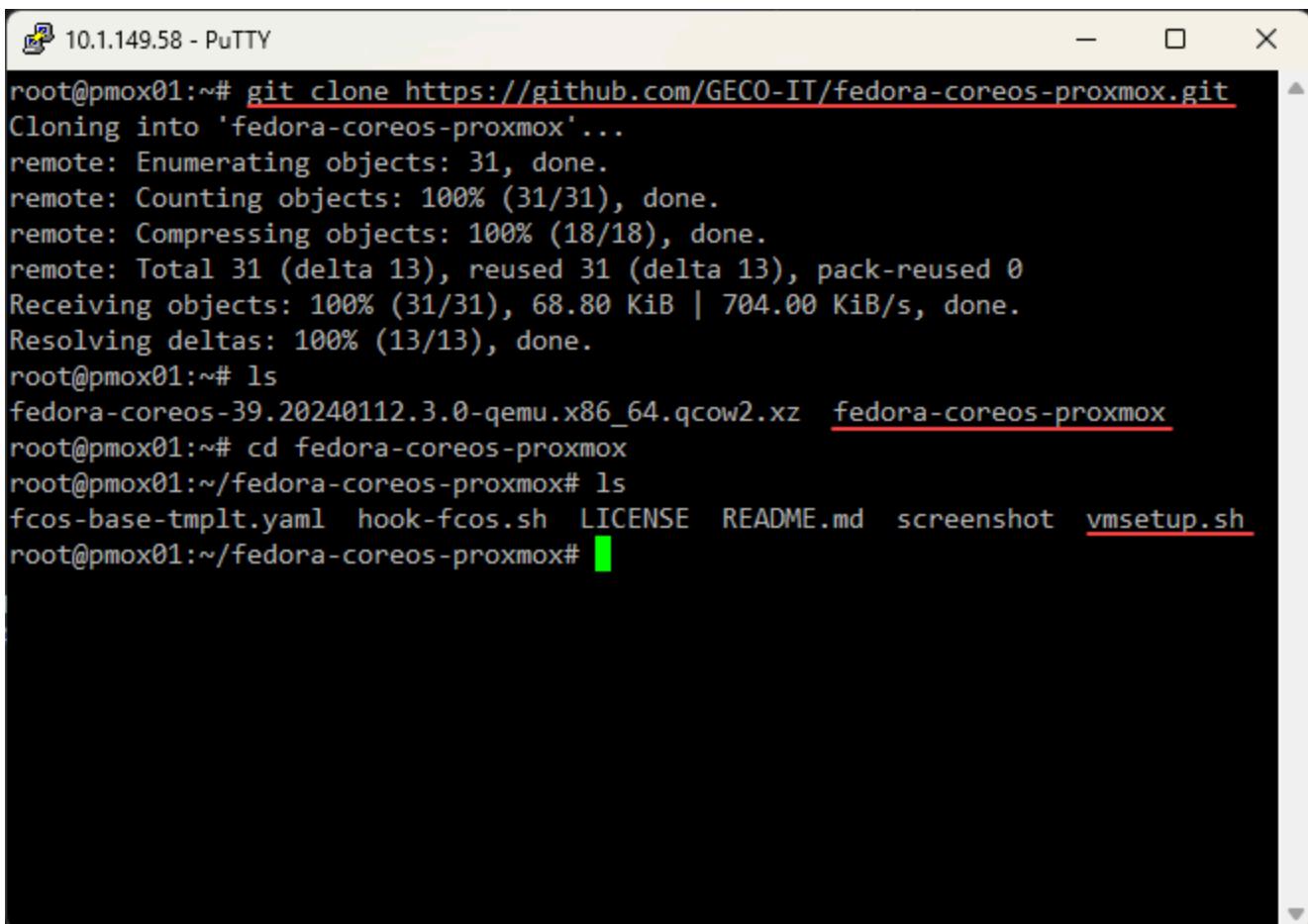
Before creating containers with Fedora CoreOS, ensure your Proxmox VE setup is ready. This involves checking available [disk space](#), configuring network settings, and making sure your Promox host is up-to-date. If you can navigate to the Proxmox web interface to manage containers (view Proxmox container toolkit settings) and virtual machines you should be good to go.

We need to clone the repository that contains the community script for deploying the Fedora CoreOS installation. You can clone the following repository:

- <https://github.com/GECO-IT/fedora-coreos-proxmox.git>

I did this directly from my Proxmox VE host. Just install the git tools if you haven't already: apt install git -y

You will see the **fedora-coreos-proxmox** folder. If you cd inside the folder, you will see a **vmsetup.sh** wrapper script that is the script we will run on the Proxmox VE server. The **fedora-coreos-<version>.yaml** serves as the ignition file for CoreOS. The Ignition file configures all the required settings..



```
10.1.149.58 - PuTTY
root@pmax01:~# git clone https://github.com/GECO-IT/fedora-coreos-proxmox.git
Cloning into 'fedora-coreos-proxmox'...
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 31 (delta 13), reused 31 (delta 13), pack-reused 0
Receiving objects: 100% (31/31), 68.80 KiB | 704.00 KiB/s, done.
Resolving deltas: 100% (13/13), done.
root@pmax01:~# ls
fedora-coreos-39.20240112.3.0-qemu.x86_64.qcow2.xz  fedora-coreos-proxmox
root@pmax01:~# cd fedora-coreos-proxmox
root@pmax01:~/fedora-coreos-proxmox# ls
fcos-base-tmplt.yaml  hook-fcos.sh  LICENSE  README.md  screenshot  vmsetup.sh
root@pmax01:~/fedora-coreos-proxmox#
```

Cloning the repo down from the community to install fedora coreos in proxmox

2. Enable snippets on a Proxmox VE storage repository

Next, following the instructions on the GitHub repo, we need to enable **snippets** on a local storage repository. It defaults to the **local** default storage in Proxmox. However, you can edit the **vmsetup.sh** script to point to a different storage location for both the snippet storage and template storage which is also needed.

You will see the lines I have changed below in the top part of the script. I have changed the **TEMPLATE_VMSTORAGE** and **SNIPPET_STORAGE** to the locations I wanted. Also, the **vmsetup.sh** script was quite

a bit behind on the Fedora CoreOS version it was looking to deploy. So, I updated that to the latest at the time of writing in the file below.

```
#!/bin/bash

#set -x # debug mode
set -e

# =====
# global vars

# force english messages
export LANG=C
export LC_ALL=C

# template vm vars
TEMPLATE_VMID="900"
TEMPLATE_VMSTORAGE="CephPool01"
SNIPPET_STORAGE="cephfs"
VMDISK_OPTIONS=",discard=on"

TEMPLATE_IGNITION="fcos-base-tmplt.yaml"

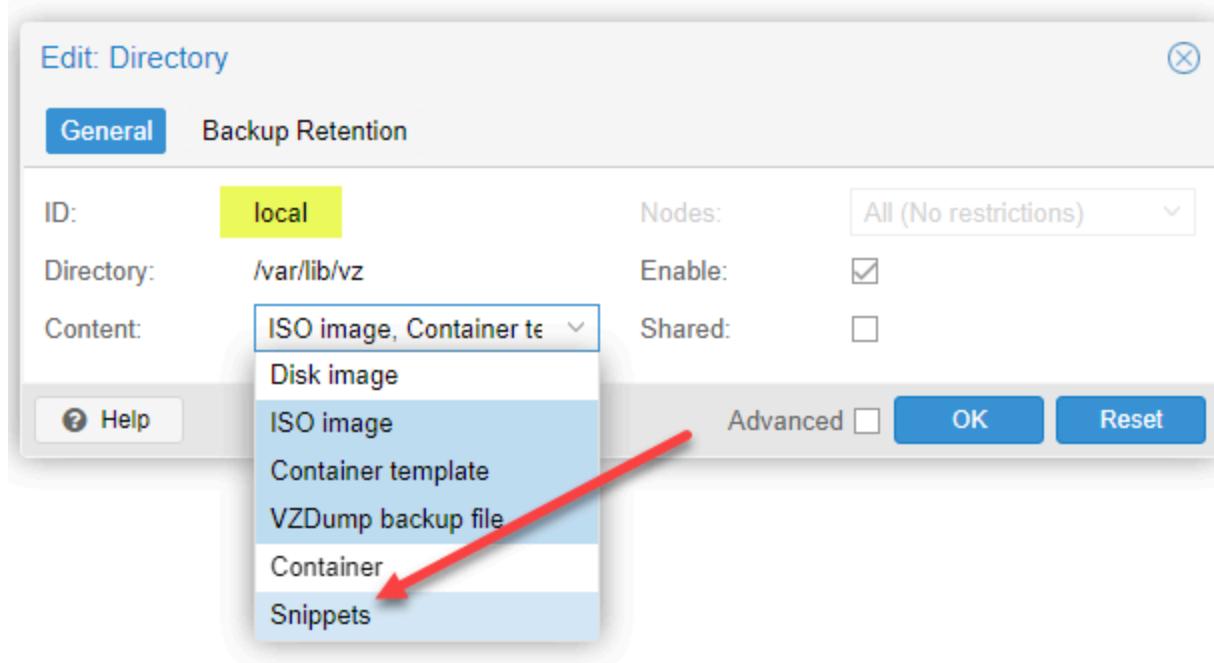
# fcos version
STREAMS=stable
VERSION=39.20240112.3.0
PLATFORM=qemu
BASEURL=https://builds.coreos.fedoraproject.org

# =====
# main()
```

Changing the template and snippet storage in the shell script

Make a connection in a browser to the URL of your Proxmox web UI if you want to go with the default **local** location, or any other location, navigate to **Datacenter > Storage > “your storage”** in the menu navigation, then click **EDIT**.

Add **Snippets** to the **Content** dropdown. Then click **OK**. You can also create a new folder and enable it with the snippets content type if you want something specific set aside for this use case. Just create a new folder, enter a description if you like, and make changes to the vmsetup.sh file.

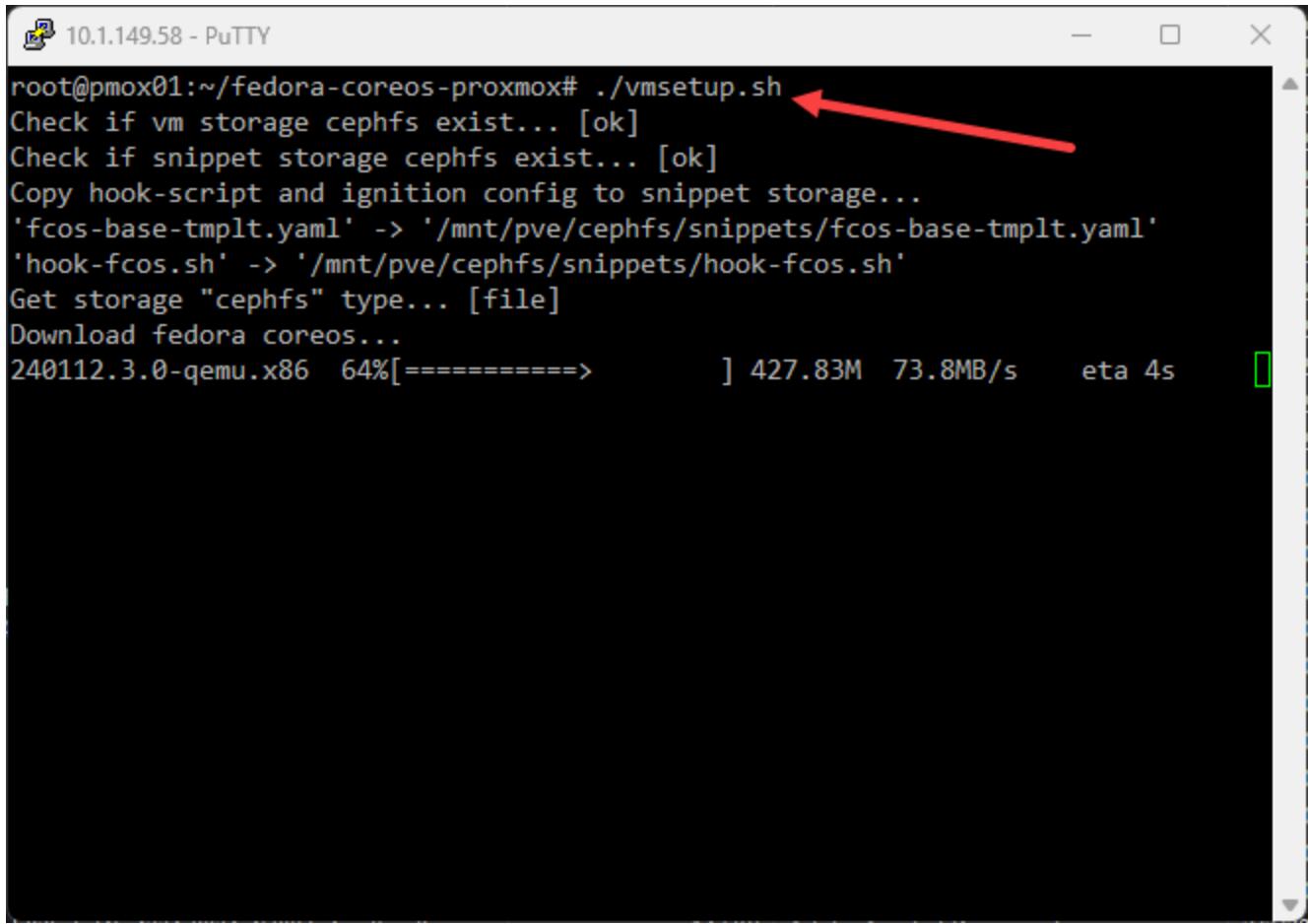


Enabling snippets on the local storage repository

3. Run the included shell script

Now that we have the snippets enabled on the storage of our choice, we can run the `vmsetup.sh` script included in the cloned repo.

It will automatically download the Fedora CoreOS image in the QEMU QCOW format needed. The Fedora CoreOS container templates in Proxmox streamline the deployment process. The Fedora CoreOS template allows for new container creations and uses the configuration files for settings specific to Fedora CoreOS to ensure smooth operation within the Proxmox environment.



```
root@pmox01:~/fedora-coreos-proxmox# ./vmsetup.sh
Check if vm storage cephfs exist... [ok]
Check if snippet storage cephfs exist... [ok]
Copy hook-script and ignition config to snippet storage...
'fcos-base-tmplt.yaml' -> '/mnt/pve/cephfs/snippets/fcos-base-tmplt.yaml'
'hook-fcos.sh' -> '/mnt/pve/cephfs/snippets/hook-fcos.sh'
Get storage "cephfs" type... [file]
Download fedora coreos...
240112.3.0-qemu.x86 64%[=====] 427.83M 73.8MB/s eta 4s
```

Running the vmsetup.sh script

The process should complete with the message at the bottom: **Convert VM 900 in proxmox vm template.**

```

10.1.149.58 - PuTTY
transferred 8.4 GiB of 10.0 GiB (83.91%)
transferred 8.5 GiB of 10.0 GiB (84.94%)
transferred 8.6 GiB of 10.0 GiB (85.95%)
transferred 8.7 GiB of 10.0 GiB (87.07%)
transferred 8.8 GiB of 10.0 GiB (88.14%)
transferred 8.9 GiB of 10.0 GiB (89.16%)
transferred 9.0 GiB of 10.0 GiB (90.22%)
transferred 9.1 GiB of 10.0 GiB (91.23%)
transferred 9.2 GiB of 10.0 GiB (92.30%)
transferred 9.3 GiB of 10.0 GiB (93.32%)
transferred 9.4 GiB of 10.0 GiB (94.35%)
transferred 9.5 GiB of 10.0 GiB (95.37%)
transferred 9.6 GiB of 10.0 GiB (96.37%)
transferred 9.7 GiB of 10.0 GiB (97.39%)
transferred 9.8 GiB of 10.0 GiB (98.44%)
transferred 9.9 GiB of 10.0 GiB (99.44%)
transferred 10.0 GiB of 10.0 GiB (100.00%)
transferred 10.0 GiB of 10.0 GiB (100.00%)
Successfully imported disk as 'unused0:CephPool01:vm-900-disk-0'
update VM 900: -scsi0 CephPool01:vm-900-disk-0,discard=on -scsihw virtio-scsi-pci
i
update VM 900: -hookscript cephfs:snippets/hook-fcos.sh
Convert VM 900 in proxmox vm template... [done]
root@pox01:~/fedora-coreos-proxmox#

```

The vm is converted to a template

If you hop over to the Proxmox web interface, you will see the new virtual machine template.

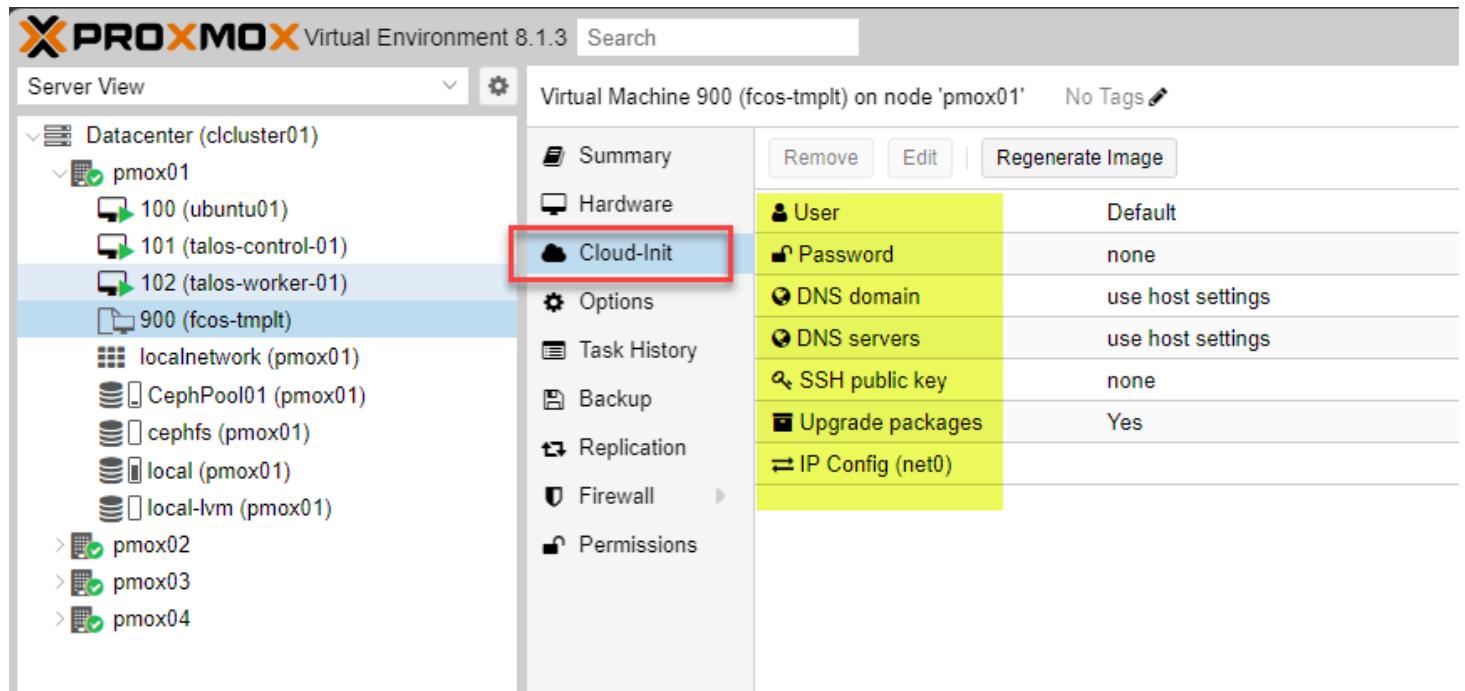
The screenshot shows the Proxmox Virtual Environment 8.1.3 web interface. The left sidebar shows a tree view of Datacenter (clcluster01) nodes: pmox01, pmox02, pmox03, and pmox04. Under pmox01, there are several VMs: 100 (ubuntu01), 101 (talos-control-01), 102 (talos-worker-01), and 900 (fcos-tmplt). The 900 entry is selected. The main panel displays the details for Virtual Machine 900 (fcos-tmplt) on node 'pmox01'. The 'Summary' tab is active. The summary panel shows the name 'fcos-tmplt' and links to HA State, Node, Processors, Memory, and Bootdisk size. Other tabs include Hardware, Cloud-Init, Options, Task History, Backup, Replication, Firewall, and Permissions.

The new template vm

4. Configure the cloud-init settings for the created template

Click the VM and then look at the **Cloud-Init** settings. Here you will find settings you can customize for:

- User
- Password (passwd)
- DNS domain
- [DNS servers](#)
- SSH public key
- Upgrade packages
- IP Config (defaults to the default Linux bridge)



Configuring cloud init parameters for proxmox fedora coreos install

Below, I have configured custom settings for Cloud-Init.

Virtual Machine 900 (fcos-tmplt) on node 'pmox01' No Tags

User	linuxadmin
Password	*****
DNS domain	cloud.local
DNS servers	10.1.149.10
SSH public key	none
Upgrade packages	Yes
IP Config (net0)	

Entering custom cloud init parameters for fedora coreos in proxmox

5. Clone the template VM to a new Fedora CoreOS virtual machine

Now that we have the Cloud-Init settings configured, we can [clone a new virtual machine](#) from the new template.

Cloning a new VM called **pmcos01** from the newly created template.

Clone VM Template 900

Target node:	pmox01	Mode:	Full Clone
VM ID:	103	Target Storage:	Same as source
Name:	pmcos01	Format:	QEMU image format (qc)
Resource Pool:		Clone	

Cloning a new virtual machine for coreos installation

The clone task completes successfully. As you can see, the process to spin up quick Dev workloads for app development, websites, working with source, etc, is easy.

Task viewer: VM 900 - Clone



Output Status

Stop

Download

```
transferred 7.9 GiB of 10.0 GiB (79.22%)
transferred 8.0 GiB of 10.0 GiB (80.23%)
transferred 8.1 GiB of 10.0 GiB (81.25%)
transferred 8.2 GiB of 10.0 GiB (82.27%)
transferred 8.3 GiB of 10.0 GiB (83.28%)
transferred 8.4 GiB of 10.0 GiB (84.30%)
transferred 8.5 GiB of 10.0 GiB (85.31%)
transferred 8.6 GiB of 10.0 GiB (86.33%)
transferred 8.7 GiB of 10.0 GiB (87.34%)
transferred 8.8 GiB of 10.0 GiB (88.36%)
transferred 8.9 GiB of 10.0 GiB (89.38%)
transferred 9.0 GiB of 10.0 GiB (90.39%)
transferred 9.1 GiB of 10.0 GiB (91.41%)
transferred 9.2 GiB of 10.0 GiB (92.42%)
transferred 9.3 GiB of 10.0 GiB (93.44%)
transferred 9.4 GiB of 10.0 GiB (94.45%)
transferred 9.5 GiB of 10.0 GiB (95.47%)
transferred 9.6 GiB of 10.0 GiB (96.48%)
transferred 9.8 GiB of 10.0 GiB (97.50%)
transferred 9.9 GiB of 10.0 GiB (98.52%)
transferred 10.0 GiB of 10.0 GiB (99.53%)
transferred 10.0 GiB of 10.0 GiB (100.00%)
transferred 10.0 GiB of 10.0 GiB (100.00%)
TASK OK
```

The cloning task is successful for cloning a new proxmox fedora coreos installation

Now, we boot the new virtual machine and it boots. We can see the OS loading the config from the Ignition file.

QEMU (pmcos01) - noVNC - Personal - Microsoft Edge

Not secure | https://10.1.149.61:8006/?console=kvm&novnc=1&vmid=103&vmname=pmcos01&... A

```
activated successfully.
[ 17.042352] igni[1291]: ignition-mount.service: Referenced but unset environment variable evaluates to an empty string: IGNITION_ARGS
[ 17.047786] systemd[1]: Stopped ignition-ostree-transposefs-autosave-xfs.service - Ignition OSTree: Autosave XFS Rootfs Partition.
[ 17.051123] systemd[1]: ignition-ostree-growfs.service: Deactivated successfully.
[ 17.055257] systemd[1]: Stopped ignition-ostree-growfs.service - Ignition OSTree: Grow Root Filesystem.
[ 17.058092] ignition[1291]: Ignition 2.17.0
[ 17.059909] ignition[1291]: Stage: umount
[ 17.061084] systemd[1]: ignition-ostree-uuid-root.service: Deactivated successfully.
[ 17.064906] ignition[1291]: reading system config file "/usr/lib/ignition/base.d/00-core.ign"
[ 17.067270] ignition[1291]: reading system config file "/usr/lib/ignition/base.d/30-afterburn-sshkeys-core.ign"
[ 17.069835] ignition[1291]: no config dir at "/usr/lib/ignition/base.platform.d/qemu".
[ 17.071861] ignition[1291]: umount: umount passed
[ 17.073098] ignition[1291]: Ignition finished successfully
[ 17.081006] systemd[1]: Stopped ignition-ostree-uuid-root.service - Ignition OSTree: Regenerate Filesystem UUID (root).
[ 17.085791] systemd[1]: ignition-mount.service: Deactivated successfully.
```

Booting the new fedora coreos installation in proxmox

After the machine fully boots and grabs an IP address, I log into the VM on the console, and I used the **linuxadmin** user I had specified in the cloud-init settings.

Success! I can login with the new **linuxadmin** user, showing the VM has used our cloud-init settings.

```
Fedora CoreOS 39.20240112.3.0
Kernel 6.6.9-200.fc39.x86_64 on an x86_64 (tty1)

SSH host key: SHA256:p12X2231eTWN9E4dLz9vgBKu/XU6Ka0D1s0SLVwJXkg (ED25519)
SSH host key: SHA256:xe8PhGPDfVoI6fEK2QzrVxUWkAmUreHK0numBwPx7Bg (ECDSA)
SSH host key: SHA256:wX+xJnCfc6AYIEBwsefQModWIxEYuZS19lcB+Jfta6c (RSA)
ens18: 10.1.149.203 fe80::7207:2000:f181:6325
Ignition: ran on 2024/02/12 20:38:27 UTC (this boot)
Ignition: user-provided config was applied
No SSH authorized keys provided by Ignition or Afterburn
pmcos01 login: linuxadmin
Password:
Fedora CoreOS 39.20240112.3.0
[systemd]
Failed Units: 1
  geco-motd.service
[linuxadmin@pmcos01 ~]$ _
```

Logging into the fedora coreos virtual machine

The Fedora CoreOS installation already has Docker preinstalled, so we can create containers, including system containers and application containers immediately after cloning over new VMs. Now all we need to do is start spinning up our containers.

Commands:

```

attach      Attach local standard input, output, and error streams to a running container
commit     Create a new image from a container's changes
cp          Copy files/folders between a container and the local filesystem
create     Create a new container
diff        Inspect changes to files or directories on a container's filesystem
events     Get real time events from the server
export     Export a container's filesystem as a tar archive
history    Show the history of an image
import     Import the contents from a tarball to create a filesystem image
inspect    Return low-level information on Docker objects
kill       Kill one or more running containers
load       Load an image from a tar archive or STDIN
logs       Fetch the logs of a container
pause      Pause all processes within one or more containers
port       List port mappings or a specific mapping for the container
rename    Rename a container
restart   Restart one or more containers
rm        Remove one or more containers
rmi      Remove one or more images
save      Save one or more images to a tar archive (streamed to STDOUT by default)
start     Start one or more stopped containers
stats     Display a live stream of container(s) resource usage statistics
stop      Stop one or more running containers
tag       Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top       Display the running processes of a container
unpause  Unpause all processes within one or more containers
update   Update configuration of one or more containers
wait     Block until one or more containers stop, then print their exit codes

```

Global Options:

--config string	Location of client config files (default "/var/home/core/.docker")
-c, --context string	Name of the context to use to connect to the daemon (overrides DOCKER_HOST context use")
-D, --debug	Enable debug mode
-H, --host list	Daemon socket to connect to
-l, --log-level string	Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls	Use TLS; implied by --tlsverify
--tlscacert string	Trust certs signed only by this CA (default "/var/home/core/.docker/ca.pem")
--tlscert string	Path to TLS certificate file (default "/var/home/core/.docker/cert.pem")

Fedora coreos comes out of the box ready to run docker containers

Wrapping up Proxmox containers Fedora CoreOS install

There are many advantages of using Fedora CoreOS with Proxmox for managing containers and virtual machines. Proxmox is gaining popularity in the [home lab](#) and even the business realm. Especially with the recent Broadcom breakup with the VMware product portfolio, I think many others will be switching over to Proxmox and other options if they are currently running VMware. Fedora CoreOS provides the resources needed to go all in on an operating system set for container mode, giving dev and DevOps users what they need for development and a platform for running containers across the board. You don't have to have a subscription for either and you can freely use many of the forum, wiki, and support thread resources out there. Let me know in the comments what you think about Fedora CoreOS and also be sure to sign up for the forums.

Proxmox Helper Scripts you can use

Let's take a look at Proxmox helper scripts that can save you as a Proxmox admin tons of time and effort in doing manual tasks through the GUI. Let's see how they work, and some examples of Proxmox helper scripts you can use to automate tasks in your environment.

Table of contents

- [What are Scripts?](#)
- [How do Scripts work?](#)
- [Examples of Scripts](#)
 - [Backup script](#)
 - [Migration Script](#)
 - [Firewall Script](#)
- [Benefits of Scripts](#)
- [Scripts FAQs](#)
- [Wrapping up](#)

What are Scripts?

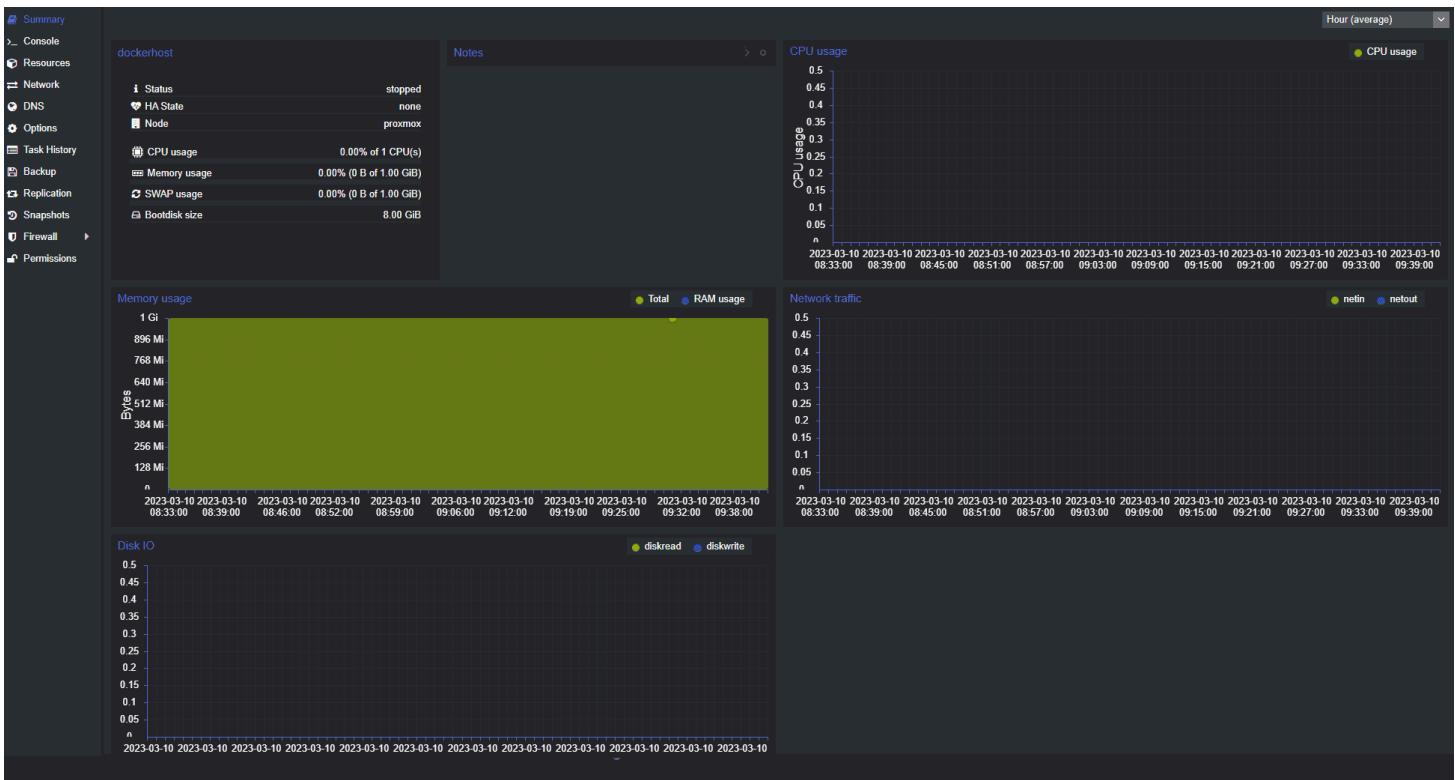
Scripts are tiny programs that are usually simple in what they contain but that can be extremely helpful such as providing automation for some type of mundane task.

Type	Description	Disk usage %	Memory us ...	CPU usage	Uptime	Host CPU ...	Host Memo...
lxc	102 (dockerhost)						
node	proxmox	70.4 %	15.9 %	3.4% of 4 ...	3 days 00:14...		
qemu	101 (clonesource)						
qemu	104 (VM 104)						
qemu	100 (VM 100)						
storage	ISOs (proxmox)	7.7 %					
storage	local (proxmox)	70.4 %					
storage	local-lvm (proxmox)	0.0 %					
storage	proxbackup (proxmox)						
storage	proxlun (proxmox)						
storage	proxlun01 (proxmox)	0.0 %					

helper scripts can be run by an admin manually or scheduled by using something like a CRON job to run these automatically at specific times. You can integrate these into other tools like monitoring software so they are triggered for instance if a condition or threshold is met.

How do Scripts work?

Proxmox scripts connect to an API session on your Proxmox node that you specify for the script to connect to. The API is a RESTful API and provides the automation endpoint. Helper scripts can use the API to perform tasks like mass creating or cloning virtual machines. You can also do things like modifying network configurations, and managing backups of VMs and their configuration.



Helper scripts can be run from the host or another machine on the network. They can prompt for user input, such as the virtual machine's name to be created or the backup file name. Once the necessary information is provided, the script interacts with the API to perform the rest of the task using automation.

Examples of Scripts

Let's take a look at a few Proxmox helper scripts you can use in your environment for automating tasks.

Backup script

Backup scripts are very good to use in the environment as they can often help to make sure backups get created for virtual machines. There is nothing worse than thinking you have a VM backup of a crashed VM only to find that one was never created!

The below scripts are just examples and may need modification to work in your specific environment without error when loading. It is important to always test scripts in a non-production environment before running them in a production environment to fully understand that your logic is going to do what you think it will do.

Backup Script:

```
#!/bin/bash

read -p "Enter the name of the virtual machine to be backed up: " vm_name
read -p "Enter the backup file name: " backup_file

# Backup the specified virtual machine to the specified backup file
qm backup $vm_name $backup_file
```

You can then restore from the backup file.

Migration Script

If you want to migrate virtual machines between Proxmox hosts, this is another task that can benefit from the automation possibilities of helper scripts and interacting with the API via a automated session.

The admin can get prompted for the name of the virtual machine to be migrated and the name of the target host. Once the necessary information is provided, the script uses the API to migrate the specified virtual machine to the specified target

host.

Migration Script:

```
#!/bin/bash

read -p "Enter the name of the virtual machine to be migrated: " vm_name
read -p "Enter the name of the target host: " target_host

# Migrate the specified virtual machine to the specified target host
qm migrate $vm_name $target_host
```

It will also be interesting to see what new helper scripts will be used with the new Proxmox VMware migration functionality contained in the new versions of Proxmox. Undoubtedly there will be room to automate the migration of VMware virtual machines to Proxmox using the API for the functionality.

Firewall Script

[Proxmox](#) Firewall Script is a script that automates the configuration of the firewall rules by prompting the admin for the IP address and port number to be blocked or allowed. Then the script implements this using the API.

Firewall Script:

```
#!/bin/bash

read -p "Enter the IP address to be blocked/allowed: " ip_address
read -p "Enter the port number to be blocked/allowed: " port_number
read -p "Enter 'block' to block the IP address/port combination or 'allow' to allow it: " action

if [ $action == "block" ]; then
    # Block the specified IP address and port number
    pvesh set /cluster/firewall/iptables -ipfilter
    "in,${ip_address},tcp,dport=${port_number},j=DROP"
    echo "IP address ${ip_address} blocked on port ${port_number}"
elif [ $action == "allow" ]; then
    # Allow the specified IP address and port number
    pvesh set /cluster/firewall/iptables -ipfilter
    "in,${ip_address},tcp,dport=${port_number},j=ACCEPT"
    echo "IP address ${ip_address} allowed on port ${port_number}"
else
    echo "Invalid action specified. Please enter 'block' or 'allow'."
fi
```

Benefits of Scripts

Admins can save a ton of time and streamline their processes by using helper scripts. You can create custom scripts that meet your specific needs with some programming knowledge. Now with resources like Google Gemini and ChatGPT, writing scripts is much easier. If you get stumped you can have AI help you with these tasks.

```

PS C:\Users\Administrator> get-pvenodeshosts

cmdlet Get-PveNodesHosts at command pipeline position 1
Supply values for the following parameters:
Node: proxmox

Response          : @{data=}
StatusCode        : 200
ReasonPhrase     :
IsSuccessStatusCode : True
RequestResource   : /nodes/proxmox/hosts
Parameters        :
Method            : Get
ResponseType      : json

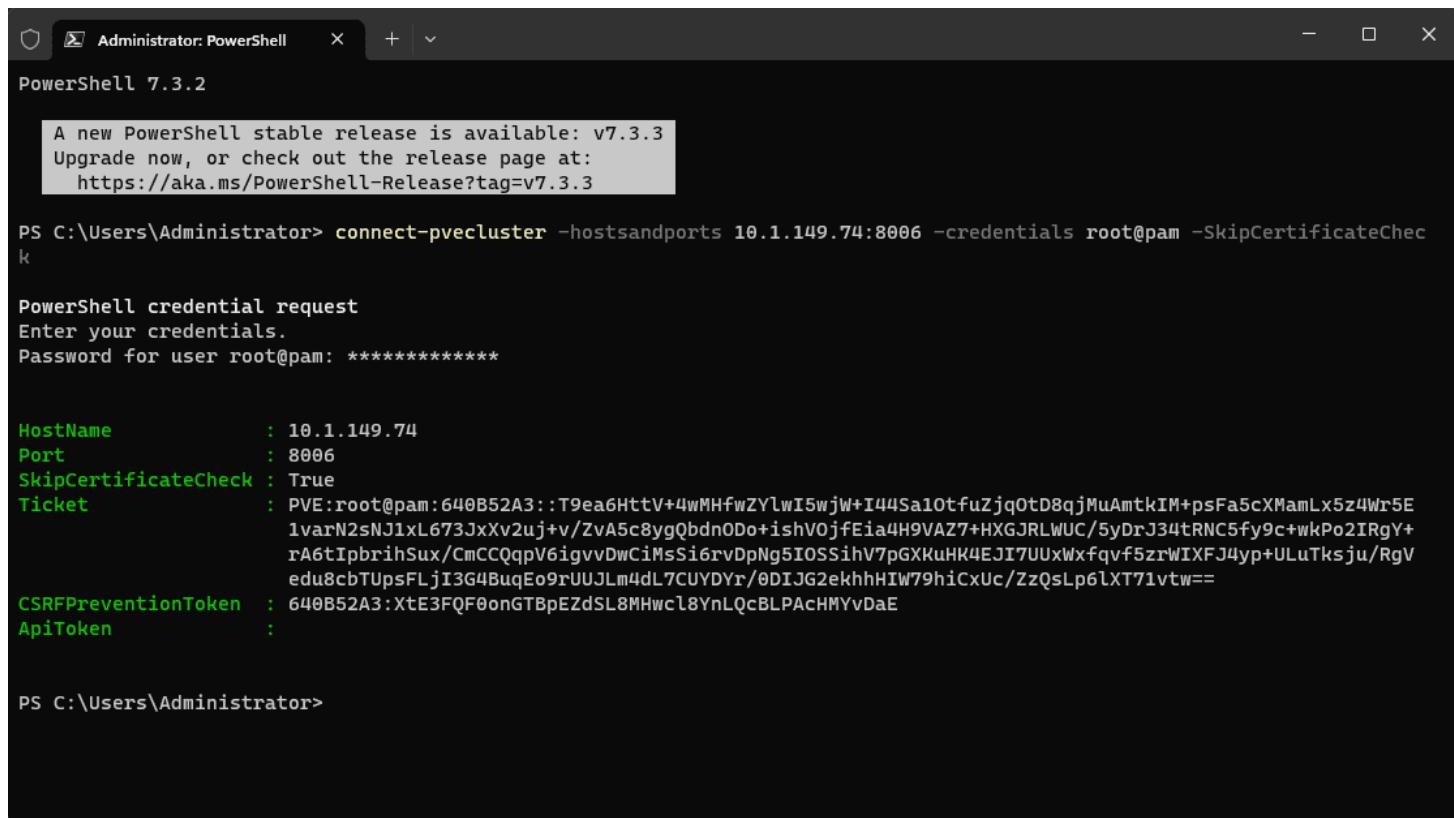
```

Also, many online resources provide pre-built scripts that you can use to automate tasks. [Proxmox provides a GitHub repository containing](#) a collection of useful helper scripts you can use as a starting point for your own scripts. Additionally, there are many online communities, such as the forum, where users share their scripts and offer support to others.

Scripts FAQs

Why are helper scripts important? Scripts are a great way to introduce automation into your environment. Using scripting and automated tasks helps to make operations much more streamlined, effective, and repeatable.

What technologies can you use for Scripts? You can use built-in Bash scripting for automation, Ansible configuration management, or even PowerShell works well for automated environments.



A screenshot of a Windows PowerShell window titled "Administrator: PowerShell". The window shows the following command and its output:

```

PowerShell 7.3.2

A new PowerShell stable release is available: v7.3.3
Upgrade now, or check out the release page at:
https://aka.ms/PowerShell-Release?tag=v7.3.3

PS C:\Users\Administrator> connect-pvecluster -hostsandports 10.1.149.74:8006 -credentials root@pam -SkipCertificateCheck

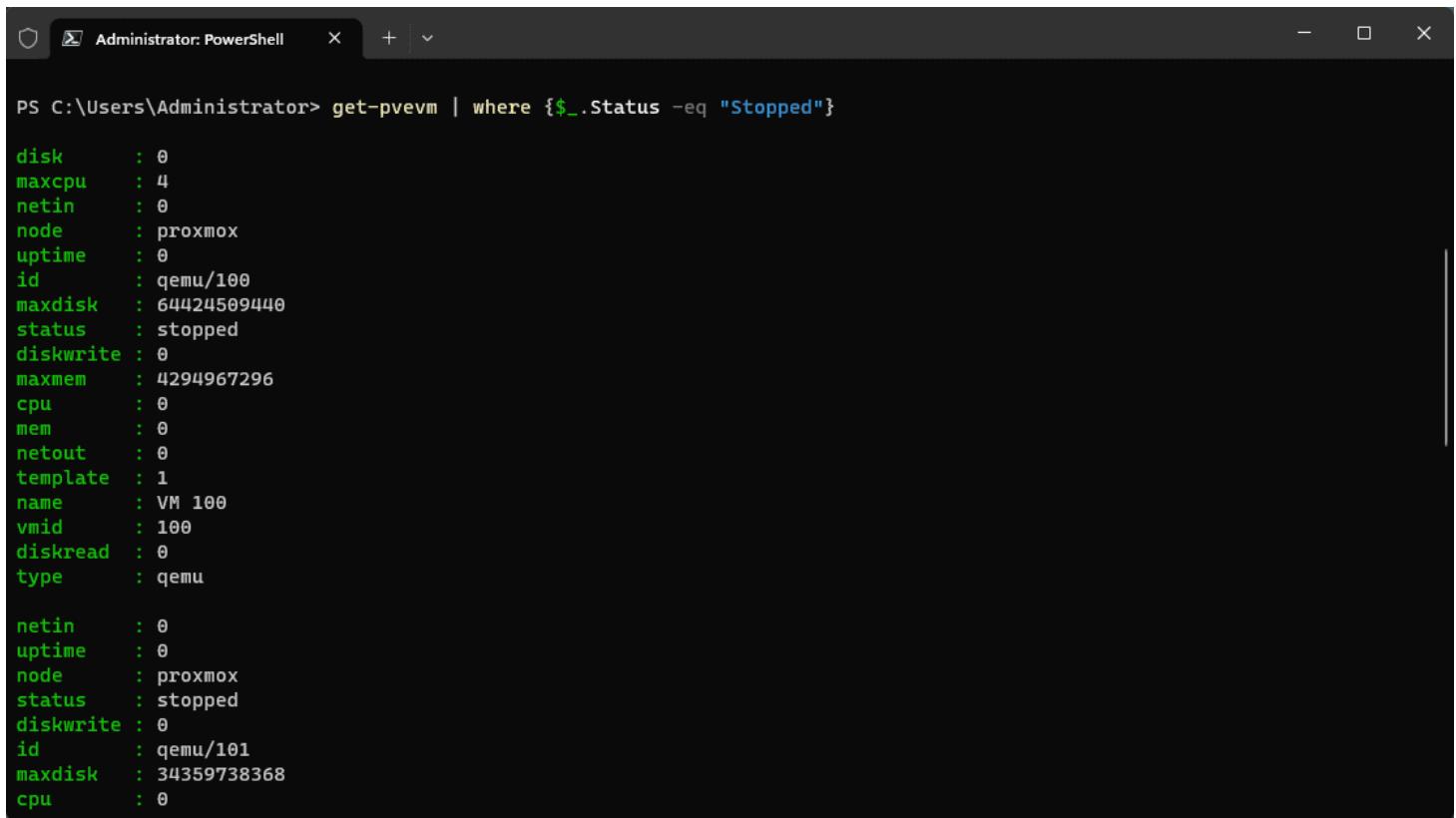
PowerShell credential request
Enter your credentials.
Password for user root@pam: *****

HostName       : 10.1.149.74
Port           : 8006
SkipCertificateCheck : True
Ticket         : PVE:root@pam:640B52A3::T9ea6HttV+4wMHfwZYlwI5wjW+I44Sa10tfuZjq0tD8qjMuAmtkIM+psFa5cXHamLx5z4Wr5E
                 1varN2sNJ1xL673Jxxv2uj+v/ZvA5c8ygQbdnODo+ishV0jfEia4H9VAZ7+HXGJRLWUC/5yDrJ34tRNC5fy9c+wkPo2IRgY+
                 rA6tIpbihSux/CmCCQqpV6igvvDwCiMsSi6rvDpNg5IOSSihV7pGXkuHK4EJI7UxWxfqvF5zrWIXFJ4yp+ULuTksju/RgV
                 edu8cbTUpesFLjI3G4BuqEo9rUUJLm4dL7CUYDYr/0DIJG2ekhhHIW79hiCxUc/ZzQsLp6lXT71vtw==

CSRFPreventionToken : 640B52A3:XtE3FQF0onGTBpEZdSL8MHwcl8YnLQcBLPAcHMYvDaE
ApiToken         :

PS C:\Users\Administrator>

```



```
PS C:\Users\Administrator> get-pvevm | where {$_.Status -eq "Stopped"}  
disk      : 0  
maxcpu   : 4  
netin     : 0  
node      : proxmox  
uptime    : 0  
id        : qemu/100  
maxdisk   : 64424509440  
status    : stopped  
diskwrite : 0  
maxmem   : 4294967296  
cpu       : 0  
mem      : 0  
netout    : 0  
template  : 1  
name      : VM 100  
vmid      : 100  
diskread  : 0  
type      : qemu  
  
netin      : 0  
uptime    : 0  
node      : proxmox  
status    : stopped  
diskwrite : 0  
id        : qemu/101  
maxdisk   : 34359738368  
cpu       : 0
```

Why use Proxmox in your environment? It is a great hypervisor with many features and capabilities for running home labs or production workloads.

Wrapping up

Keep in mind these Proxmox helper scripts are only a few of the possibilities when working with Proxmox. Admins can greatly benefit in efficiency and time involved with day-to-day operations by using these types of helper scripts working with the API interface. What scripts are you using in your home lab or production environments?

Proxmox scripts PowerShell Ansible and Terraform

Proxmox is growing more and more popular, especially for home lab enthusiasts and those looking to spin up labs based on totally free and open-source software. [Proxmox](#) has a great API that allows throwing automation tasks at the solution and creating Proxmox helper scripts for automating your Proxmox environment.

Why scripting and automation are important

For many reasons, scripting and automation are essential in today's infrastructure environments. IT admins and DevOps engineers must move quickly and provision, configure, and interact with infrastructure effectively and efficiently. This certainly involves automation.

Type ↑	Description	Disk usage %	Memory us...	CPU us...
node	proxmox	70.8 %	16.7 %	3.0 % of
node	proxmox02			
qemu	100 (VM 100)			
qemu	101 (pfSense)			
storage	ISOs (proxmox)	14.3 %		
storage	Proxmox-SynologyLUN (pro...	100.0 %		
storage	Synology-Proxmox (proxmox)			
storage	local (proxmox)	70.8 %		
storage	local-lvm (proxmox)	0.0 %		
storage	Proxmox-SynologyLUN (pro...			
storage	Synology-Proxmox (proxmox...			
storage	local (proxmox02)			
storage	local-lvm (proxmox02)			

Scripting and automation improve your effectiveness as an administrator and virtualization engineer. If you accept the challenge of learning scripting, it will pay off dividends.

Infrastructure as code

Due to the massive shift to cloud-based technologies, today's infrastructure services is driven by infrastructure as code. It allows admins to commit code to a code repository location, version of that code, and other resources it manages.

Scripting automation tasks

This infrastructure-as-code approach includes creating VM environments and [Docker](#) containers as code. LXC containers can also easily be provisioned in Proxmox VE environments. Using simple and easy script-based approaches, admins can, with due diligence, create scripts to manage the environment.

Create: Virtual Machine

X

General OS System Disks CPU Memory Network Confirm

Node:	proxmox	Resource Pool:
VM ID:	102	
Name:		

Help

Advanced

Back

Next

Don't reinvent the wheel!

Also, there have been so many great scripts and code is already written admins don't have to reinvent the wheel. Sourcing scripts from free and open-source sites is easy to do, along with other learning tools like [YouTube](#), blog posts, and other third-party sites etc. You can tap into many great learning resources and support forums.

Create: LXC Container

General **Template** **Disks** **CPU** **Memory** **Network** **DNS** **Confirm**

Node:	proxmox	Resource Pool:
CT ID:	102	Password:
Hostname:		Confirm password:
Unprivileged container:	<input checked="" type="checkbox"/>	SSH public key:
Nesting:	<input checked="" type="checkbox"/>	Load SSH Key File

Help **Advanced** **Back** **Next**

Running into error messages along the way is part of the learning process. However, the effort will outweigh the challenges with massive time and effort savings that automation provides.

Proxmox automated REST API interface

The Proxmox VE solution uses an interface known as RESTful API. The API uses the HTTPS protocol and the server listens to port 8006. So the base URL for that API is: <https://your.server:8006/api2/json/>

Proxmox VE uses a ticket or token-based authentication. All requests to the API need to include a ticket inside a Cookie (header) or send an API token through the Authorization header.

Proxmox PowerShell scripts

I have found many great repositories working with Proxmox scripts to allow changing settings, install updates, backup your configuration, and loading configurations to give you an idea of what is possible. Check out this Proxmox PowerShell repository, which provides a VMware PowerCLI approach to managing Proxmox with [Proxmox helper scripts](#).

[PowerShell Gallery | Corsinvest.ProxmoxVE.Api 7.3.0](#)

This is a great way to create [scripts with Proxmox and PowerShell](#).

```

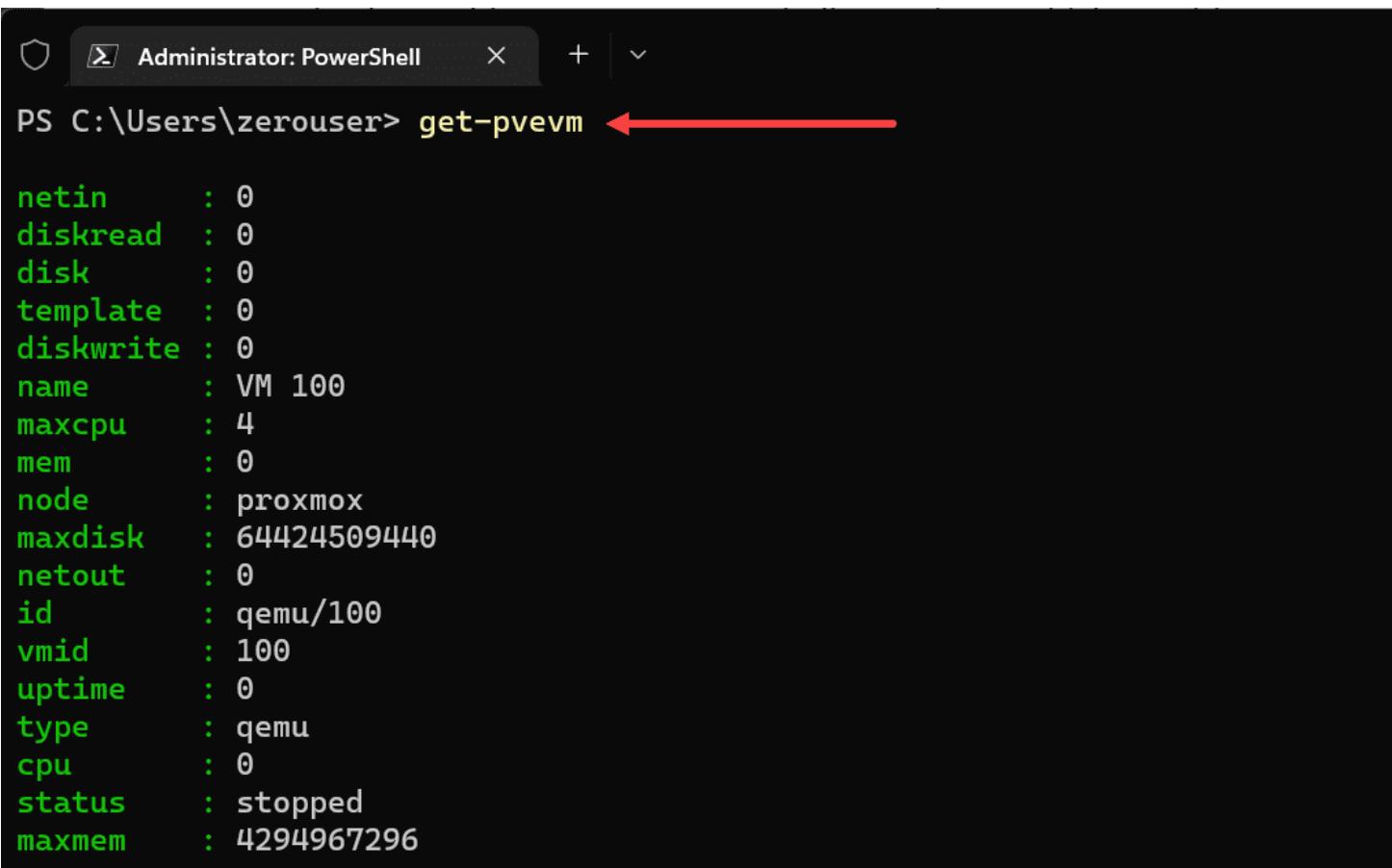
PS C:\Users\zerouser> Connect-PveCluster -hostsandports 10.1.149.74:8006 -skipcertificatecheck
PowerShell credential request
Proxmox VE Username and password, username formated as user@pam, user@pve, user@yourdomain or user (default domain pam).
User: root@pam
Password for user root@pam: *****

HostName      : 10.1.149.74
Port          : 8006
SkipCertificateCheck : True
Ticket        : PVE:root@pam:63D72E82::pqw75uh2oEcfxuvo8E/jLdWhCr7cxV0ONILom/mD8bMaRhfHpgyi0i/UWjVNR1rijVSZc0YG1
                [REDACTED]
CSRFPreventionToken : 63D72E82:4t7bUqn7BdoSS9DazKSc1WwDfk80FE57W2MbEAGm63Q
ApiToken      :

```

Connecting to Proxmox with the PowerShell module

Below, we are getting Proxmox VMs using the **get-pvevm** cmdlet.



```

Administrator: PowerShell
PS C:\Users\zerouser> get-pvevm ←

```

```

netin      : 0
diskread   : 0
disk       : 0
template   : 0
diskwrite  : 0
name       : VM 100
maxcpu    : 4
mem        : 0
node       : proxmox
maxdisk   : 64424509440
netout    : 0
id         : qemu/100
vmid      : 100
uptime    : 0
type       : qemu
cpu        : 0
status     : stopped
maxmem    : 4294967296

```

Getting Proxmox virtual machines using PowerShell

Proxmox Ansible and Terraform scripts

Proxmox is easy to work with using Ansible and Terraform and also allows great scripting capabilities and functionality. Check out these repositories:

[community.general.proxmox module – Management of instances in Proxmox VE cluster — Ansible Documentation](#)

[Docs overview | Telmate/proxmox | Terraform Registry](#)

Proxmox helper scripts

[Proxmox Helper Scripts | Proxmox Scripts For Home Automation \(tteck.github.io\)](#)

Wrapping up

Hopefully, this quick [guide to Proxmox](#) scripts with PowerShell, Ansible, and Terraform shows there are many great ways to automate Proxmox and create infrastructure as code in your Proxmox VE environment. With the RESTful API driven automation provided by Proxmox, you can create quick and easy infrastructure as code.

Proxmox Backup Server: Ultimate Install, Backup, and Restore Guide

Backups are essential to running Proxmox VE in the home lab or production to avoid data loss. Proxmox Backup Server is a free solution to back up and recover Proxmox VE VMs and containers.

Table of contents

- [What is Proxmox Backup Server?](#)
- [Proxmox Backup Server installation](#)
- [Logging into the Proxmox Backup](#)
- [Adding a datastore for storing backups](#)
- [Add the Proxmox Backup Server instance to your Proxmox VE server](#)
- [Creating a backup job](#)
- [Restoring a Proxmox virtual machine](#)
 - [Granular file restore](#)
- [Wrapping up](#)

What is Proxmox Backup Server?

Proxmox Backup Server or PBS is a free solution from Proxmox. It is free to download and install on a server just like Proxmox VE server is. It allows you to have a totally free and open source solution for backing up your Proxmox Server virtual machines and LXC containers without any licensing costs associated. For those running a home lab environment, this is a great solution to make sure you have things backed up.

Also, with organizations that are deciding to migrate from VMware vSphere with the Broadcom changes, it means you will not only save on licensing for the hypervisor, if you use PBS, you can save on the licensing costs for enterprise backup solutions you may have needed with VMware.

You can read more about the official features from the Proxmox documentation located here: [Proxmox Backup Server – Open-Source Enterprise Backup Solution](#).

Proxmox Backup Server installation

Like installing Proxmox VE virtualization server, installing PBS is extremely easy and looks very much like installing Proxmox VE. Let's install PBS and configure backups of Proxmox virtual machines.

First, you will need to download the PBS release ISO image from Proxmox here: [Proxmox Backup Server](#).

Once you have the [ISO file](#), “burn” the software to a USB flash drive or upload it to your Proxmox VE host if you are hosting your backup server as a virtual machine.

Below is a screenshot of the Proxmox Backup Server virtual machines booting from the ISO installation.

```
Welcome to the Proxmox Backup Server 3.0 installer ←  
initial setup startup  
mounting proc filesystem  
mounting sys filesystem  
boot cmdline: BOOT_IMAGE=/boot/linux26 ro ramdisk_size=16777216 rw quiet spla  
loading drivers: i2c_piix4 pata_acpi vmgenid floppy qemu_fw_cfg mac_hid uhci_hc  
pkr aesni_intel sha512_ssse3  
searching for block device containing the ISO proxmox-backup-server-3.0-1  
with ISO ID '5d145420-1501-11ee-a40c-7fb0fe0bd0d3'  
testing device '/dev/sr0' for ISO  
found Proxmox Backup Server ISO  
switching root from initrd to actual installation system  
Starting Proxmox installation  
Installing additional hardware drivers  
Starting hotplug events dispatcher: systemd-udevd.  
Synthesizing the initial hotplug events (subsystems)...done.  
Synthesizing the initial hotplug events (devices)...done.  
Waiting for /dev to be fully populated...done.  
mount: devpts mounted on /dev/pts.  
  in/dbus-daemon  
▶ starting D-Bus daemon  
Attempting to get DHCP leases... Internet Systems Consortium DHCP Client 4.4.3-P1  
Copyright 2004-2022 Internet Systems Consortium.  
All rights reserved.  
For info, please visit https://www.isc.org/software/dhcp/  
  
Listening on LPF/ens18/bc:24:11:5c:5a:cb  
Sending on  LPF/ens18/bc:24:11:5c:5a:cb  
Sending on  Socket/fallback  
DHCPDISCOVER on ens18 to 255.255.255.255 port 67 interval 6  
DHCPOffer of 10.3.33.230 from 10.3.33.1  
DHCPREQUEST for 10.3.33.230 on ens18 to 255.255.255.255 port 67  
DHCPACK of 10.3.33.230 from 10.3.33.1  
bound to 10.3.33.230 -- renewal in 297690 seconds.  
done  
Starting chrony for opportunistic time-sync...  
Starting a root shell on tty3.  
trying to detect country...
```

Running the PBS installer

Accept the end user license agreement (EULA).



END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX BACKUP SERVER

By using Proxmox Backup Server software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintenance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox Backup Server and any related update, source code and structure (the Programs), regardless of the delivery mechanism.

1. **License.** Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.

2. **Limited Warranty.** The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non-infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.

3. **Limitation of Liability.** To the maximum extent permitted under applicable law, under no

Abort

Previous

I agree

PBS eula

Select the installation drive.



Proxmox Backup Server (PBS)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data will be lost.

To continue the installation, press the Next button.

- **Please verify the installation target**
The displayed hard disk will be used for the installation.
Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface, via a web browser.

Target Harddisk: /dev/sda (32.00GiB, QEMU HARDDISK) ▾ Options

Abort Previous Next

PBS disk configuration

Set your country, time zone, and keyboard layout language.

Location and Time Zone selection

The **Proxmox Installer** automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.



Location timezone and keyboard layout

Create an administrative password and email address.

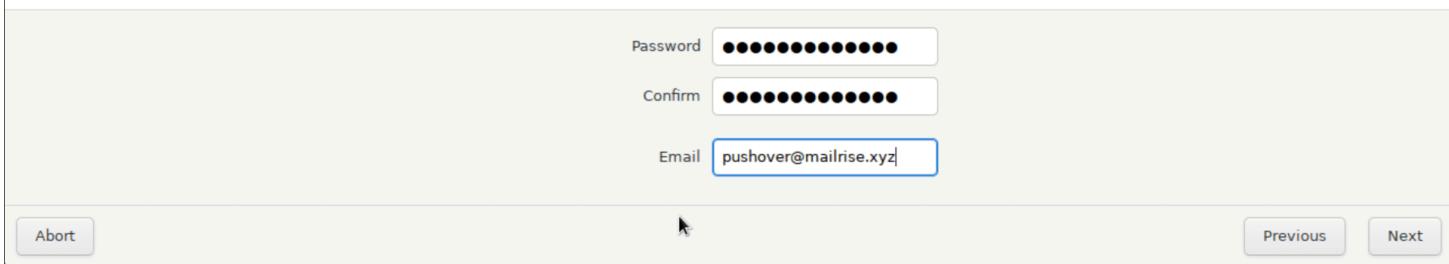
Administration Password and Email Address

Proxmox Backup Server is a full-featured, highly secure system, based on Debian GNU/Linux.

In this step, please provide the *root* password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Your Proxmox Backup Server will send important alert notifications to this email account (all emails for 'root').

To continue the installation, press the Next button.



A screenshot of a software interface for setting up a Proxmox Backup Server. The window title is 'Administration Password and Email Address'. It contains three input fields: 'Password' (filled with dots), 'Confirm' (also filled with dots), and 'Email' (containing 'pushover@mailrise.xyz'). Below the fields are buttons for 'Abort', 'Previous', and 'Next'.

Abort	Next
Previous	

Setting the password and email address

Enter the FQDN and IP address configuration.



Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	United States
Timezone:	America/Chicago
Keymap:	en-us
Email:	pushover@mailrise.xyz
Management Interface:	ens18
Hostname:	pbs01
IP CIDR:	10.3.33.230/24
Gateway:	10.3.33.1
DNS:	10.1.149.10

Automatically reboot after successful installation

Abort

Previous

Install

2023 11 26 22 06 06

Review the summary screen.



Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	United States
Timezone:	America/Chicago
Keymap:	en-us
Email:	pushover@mailrise.xyz
Management Interface:	ens18
Hostname:	pbs01
IP CIDR:	10.3.33.230/24
Gateway:	10.3.33.1
DNS:	10.1.149.10

Automatically reboot after successful installation

Abort

Previous

Install

Summary screen for the installation process with PBS

Below is a screenshot of the Proxmox Backup Server running on Proxmox VE.

PROXMOX Virtual Environment 8.1.3 Search

Server View

Node 'pve01'

Package versions

Reboot Shutdown Shell Bulk Actions Help

Search

Summary

Notes

Shell

System

Network

Certificates

DNS

Hosts

Options

Time

Syslog

Updates

Repositories

Firewall

Disks

Hour (average)

pve01 (Uptime: 2 days 07:19:45)

CPU usage: 9.26% of 16 CPU(s) IO delay: 0.09%

Load average: 0.41, 0.39, 0.23

RAM usage: 2.84% (3.57 GiB of 125.68 GiB) KSM sharing: 0 B

/ HD space: 9.34% (3.44 GiB of 36.81 GiB) SWAP usage: 0.00% (0 B of 8.00 GiB)

CPU(s): 16 x Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz (1 Socket)

Kernel Version: Linux 6.5.11-4-pve (2023-11-20T10:19Z)

Boot Mode: EFI (Secure Boot)

Manager Version: pve-manager/8.1.3/b46aac3b42da5d15

Repository Status: Production-ready Enterprise repository enabled Enterprise repository needs valid subscription

CPU usage: CPU usage (blue), IO delay (yellow)

Start Time	End Time	Node	User name	Description	Status
Nov 26 22:03:45		pve01	root@pam	VM/CT 100 - Console	
Nov 26 22:03:30	Nov 26 22:03:35	pve01	root@pam	VM 100 - Start	OK
Nov 26 22:03:28	Nov 26 22:03:30	pve01	root@pam	VM 100 - Create	OK
Nov 26 22:00:15	Nov 26 22:01:36	pve01	root@pam	LVM-Thin Storage NVMePool01 - Create	OK
Nov 26 21:59:59	Nov 26 22:00:02	pve01	root@pam	Thinpool NVMPool01-NVMPool01 - Remove	OK
Nov 26 21:59:42	Nov 26 21:59:52	pve01	root@pam	LVM Thin Storage NVMePool01 - Create	OK

Running the proxmox backup server on proxmox ve

Booting the Proxmox Backup Server after installation.

GNU GRUB version 2.06-13

```
*Proxmox Backup Server GNU/Linux
Advanced options for Proxmox Backup Server GNU/Linux
Memory test (memtest86+x64.bin)
Memory test (memtest86+x64.bin, serial console)
```

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands before booting or 'c'
for a command-line.
The highlighted entry will be executed automatically in 1s.

Booting the proxmox backup server after installation

After the Proxmox Backup Server boots, you will see the default text splash screen directing you to open a browser to start [managing the server](#). Note the port 8007, which is different from the Proxmox VE port 8006 for accessing the GUI. This is the command line interface (CLI) from the console.

```
Welcome to the Proxmox Backup Server. Please use your web browser to  
configure this server - connect to:
```

```
https://10.3.33.230:8007/
```

```
pbs01 login: _
```

Proxmox backup server login

Logging into the Proxmox Backup

Next, let's navigate to the web UI and log in to the web UI for the Proxmox Backup Server.

Proxmox Backup Server Login

User name: 

Password:

Realm: Linux PAM standard authentication 

Language: Default (English) 

Save User name:

Login

Pbs web interface login

Below, we see the overview of the PBS tool.

The screenshot shows the PBS dashboard with the following sections:

- System Status:** pbs (Uptime: 00:21:11) with a "Show Fingerprint" button. It displays CPU usage (0.31% of 2 CPU(s)), IO delay (0.00%), RAM usage (9.25% / 181.27 MiB of 1.91 GiB), Load average (0,0,0), HD space(root) (12.94% / 1.85 GB of 14.28 GB), and SWAP usage (0.00% / 0 B of 1.87 GiB).
- System Info:** CPU(s) (2 x AMD Ryzen 7 5800U with Radeon Graphics (1 Socket)), Kernel Version (Linux 6.5.11-6-pve (2023-11-29T08:32Z)), Boot Mode (Legacy BIOS), and Repository Status (Production-ready Enterprise repository enabled, Enterprise repository needs valid subscription).
- Datastore Usage:** A table with columns Name ↑, Size, Used, Available, Usage %, Estimated Full, and History... showing "No Data".
- Longest Tasks (30 days):** A table showing Log Rotation with a duration of <0.1s.

Logged into PBS seeing the dashboard

Adding a datastore for storing backups

Next, let's add a datastore for storing Proxmox backups. After logging into the Proxmox Backup Server, click the **Add Datastore** option under **Datastore**.

The screenshot shows the PBS dashboard with various system metrics and configuration options. A red arrow points from the bottom-left towards the 'Add Datastore' button in the sidebar.

System Metrics:

- CPU usage: 0.00% of 2 CPU(s)
- IO delay: 0.00%
- RAM usage: 9.08% (177.92 MiB of 1.91 GiB)
- Load average: 0,0,0
- HD space(root): 12.95% (1.85 GB of 14.28 GB)
- SWAP usage: 0.00% (0 B of 1.87 GiB)

System Information:

- CPU(s): 2 x AMD Ryzen 7 5800U with Radeon Graphics (1 Socket)
- Kernel Version: Linux 6.5.11-6-pve (2023-11-29T08:32Z)
- Boot Mode: Legacy BIOS
- Repository Status: Production-ready Enterprise repository enabled (Enterprise repository needs valid subscription)

Datastore Usage:

Name ↑	Size	Used	Available	Usage %	Estimated Full	History...
No Data						

Longest Tasks (30 days):

- Log Rotation: <0.1s (status: green)

Beginning the process to add a datastore to the PBS

This will launch the **Add Datastore** dialog box. Name the new datastore and then enter the backing path. You don't have to create the backing path location as the process will do this for you. Click **Add**. This will add the backup storage location to *local storage* on your Proxmox Backup Server.

The screenshot shows the Proxmox Backup Server 3.1-2 interface. On the left, a sidebar lists various management options: Dashboard, Notes, Configuration, Access Control, Remotes, Traffic Control, Certificates, Subscription, Administration, Shell, Storage / Disks, Tape Backup, Datastore, and Add Datastore. The main area is titled "Dashboard" and displays system statistics for "pbs (Uptime: 00:31:33)". Below the stats is a "Show Fingerprint" button. A central modal window is open for "Add: Datastore". The "General" tab is active, showing fields for "Name" (set to "Backups") and "Backing Path" (set to "/backups"). Both of these fields are highlighted with a red border. Other tabs in the modal include "Prune Options" and "History...". At the bottom of the modal are "Help" and "Add" buttons.

Naming the datastore and setting the path

Now, we see the datastore displaying in our Proxmox Backup Server.

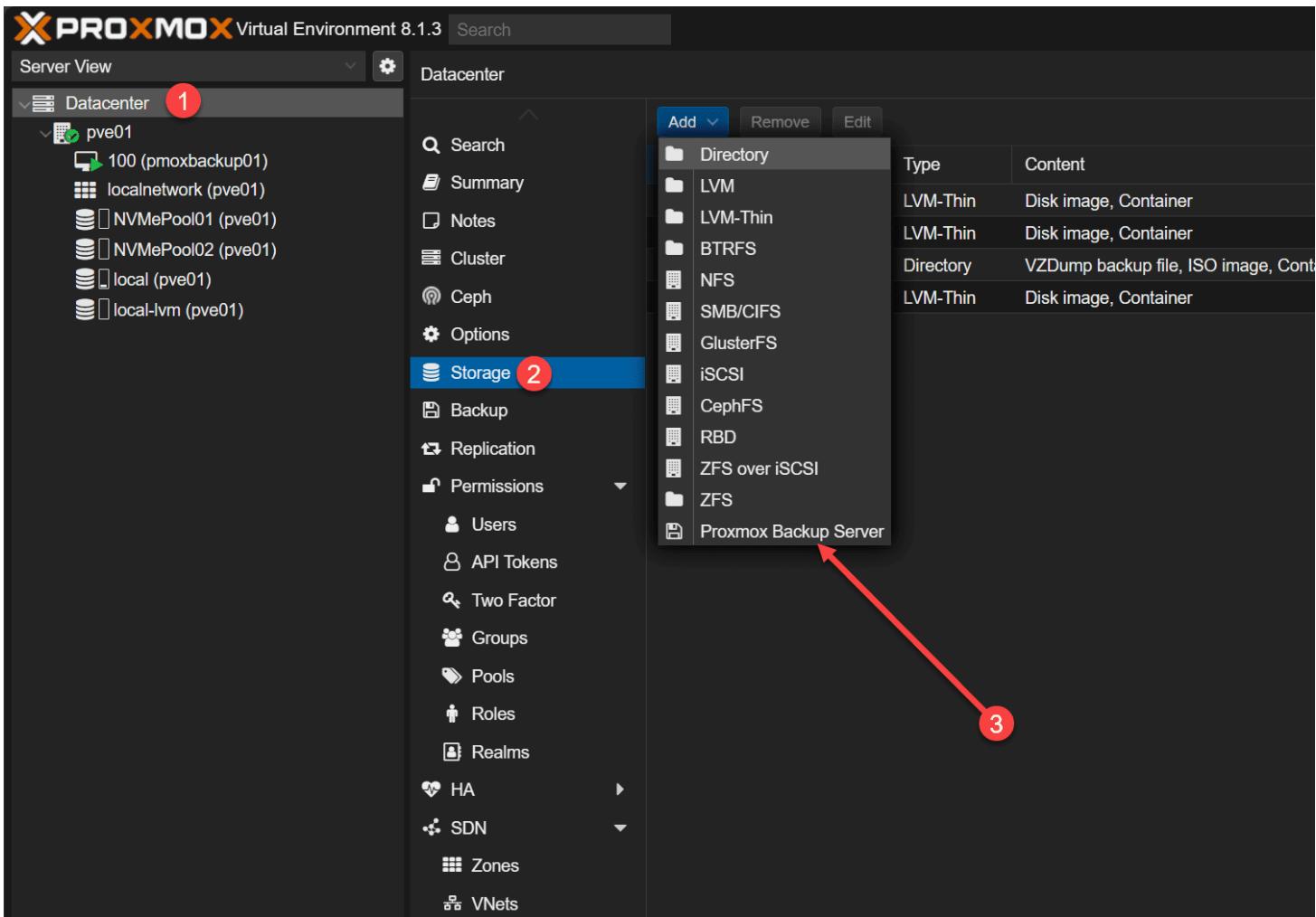
The screenshot shows the Proxmox Backup Server 3.1-2 dashboard. The left sidebar contains navigation links: Dashboard, Notes, Configuration, Access Control, Remotes, Traffic Control, Certificates, Subscription, Administration (Shell, Storage / Disks), Tape Backup, Datastore, Backups (highlighted with a red arrow), and Add Datastore. The main content area displays system metrics for 'pbs' (Uptime: 00:32:11) including CPU usage (1.36% of 2 CPU(s)), IO delay (0.00%), RAM usage (15.02% of 1.91 GiB), Load average (0.07, 0.02, 0), HD space(root) (14.84% of 14.28 GB), and SWAP usage (0.00% of 1.87 GiB). It also shows hardware details: 2 x AMD Ryzen 7 5800U with Radeon Graphics (1 Socket), Kernel Version (Linux 6.5.11-6-pve (2023-11-29T08:32Z)), Boot Mode (Legacy BIOS), and Repository Status (Production-ready Enterprise repository enabled, Enterprise repository needs valid subscription).

Viewing the datastore in PBS

Add the Proxmox Backup Server instance to your Proxmox VE server

On our Proxmox VE host, the Proxmox Backup Server is viewed as **Storage**. So, we first need to add it as storage to our Proxmox VE host.

Navigate to **Datacenter > Storage > Add > Proxmox Backup Server**.



Adding PBS storage to your proxmox ve instance

This will launch the **Add: Proxmox Backup Server** dialog box. Here we enter the following information:

- ID
- Server
- Username
- Password
- Datastore
- Fingerprint

You may wonder where we get the fingerprint.

Add: Proxmox Backup Server

[General](#)[Backup Retention](#)[Encryption](#)

ID:	pbs01	Nodes:	All (No restrictions)
Server:	10.1.149.155	Enable:	<input checked="" type="checkbox"/>
Username:	root@pam	Content:	backup
Password:	*****	Datastore:	Backups
		Namespace:	Root

Fingerprint: Server certificate SHA-256 fingerprint, required for self-signed certificates

[? Help](#)[Add](#)

Add the PBS certificate thumbprint

Navigate back to your Proxmox Backup Server and click on the **Dashboard > Show Fingerprint** button.

Dashboard

- Notes
- Configuration
 - Access Control
 - Remotes
 - Traffic Control
 - Certificates
 - Subscription
- Administration
 - Shell
 - Storage / Disks
 - Tape Backup
 - Datastore
 - Backups
 - Add Datastore

Dashboard

pbs (Uptime: 00:44:08)



Show Fingerprint

 CPU usage	0.00% of 2 CPU(s)	 IO delay	0.00%
 RAM usage	15.00% (293.81 MiB of 1.91 GiB)	 Load average	0,0,0
 HD space(root)	14.84% (2.12 GB of 14.28 GB)	 SWAP usage	0.00% (0 B of 1.87 GiB)

CPU(s)
2 x AMD Ryzen 7 5800U with Radeon Graphics (1 Socket)

Kernel Version
Linux 6.5.11-6-pve (2023-11-29T08:32Z)

Boot Mode
Legacy BIOS

Repository Status
 Production-ready Enterprise repository enabled  Enterprise repository needs valid subscription 

Datastore Usage

Name ↑	Size	Used	Available	Usage %	Estimated Full	History...
Backups	13.53 GB	2.12 GB	11.42 GB	 15.66%	Not enough data	

Show fingerprint on PBS

Here, we can copy the fingerprint.

The screenshot shows the Proxmox Backup Server 3.1-2 interface. On the left, a sidebar lists various management options like Dashboard, Notes, Configuration, Access Control, Remotes, Traffic Control, Certificates, Subscription, Administration, Shell, Storage / Disks, Tape Backup, Datastore, Backups, and Add Datastore. The main area is titled "Dashboard" and displays system status for a host named "pbs" (Uptime: 00:47:22). It includes sections for CPU usage (0.00% of 2 CPU(s)), IO delay (0.00%), RAM usage (15.06% of 295.05 MiB of 1.91 GiB), Load average (0.0), HD space (root) (14.84% of 2.12 GB of 14.28 GB), and SWAP usage (0.00% of 0 B of 1.87 GiB). A red arrow points to a "Copy" button in a modal window titled "Fingerprint" that displays the hex string "0d:9f:c5:ec:b3:42:e4:3a:ec:ae:a3:b7:cb:24:4b:0f:2a:85:13:b0:4f:da:06:d1:75:1a:60:75:1b:80:4d:3c". Below the modal, a message states "Production-ready Enterprise repository enabled" and "Enterprise repository needs valid subscription".

Copy the fingerprint of the PBS server

Now, we can paste in the fingerprint.

Add: Proxmox Backup Server

General Backup Retention Encryption

ID:	pbs01	Nodes:	All (No restrictions)
Server:	10.1.149.155	Enable:	<input checked="" type="checkbox"/>
Username:	root@pam	Content:	backup
Password:	*****	Datastore:	Backups
		Namespace:	Root
Fingerprint:	0d:9f:c5:ec:b3:42:e4:3a:ec:ae:a3:b7:cb:24:4b:0f:2a:85:13:b0:4f:da:06:d1:75:1a:f		

[? Help](#) Add

Thumbprint added and ready to save to connect to pbs

After adding the fingerprint and clicking the **Add** button, we see the Proxmox Backup Server listed.

Datacenter

ID ↑	Type	Content	Path/Target
NVMePool01	LVM-Thin	Disk image, Container	
NVMePool02	LVM-Thin	Disk image, Container	
local	Directory	VZDump backup file, ISO image, Cont...	/var/lib/vz
local-lvm	LVM-Thin	Disk image, Container	
pbs01	Proxmox Backup Server	VZDump backup file	



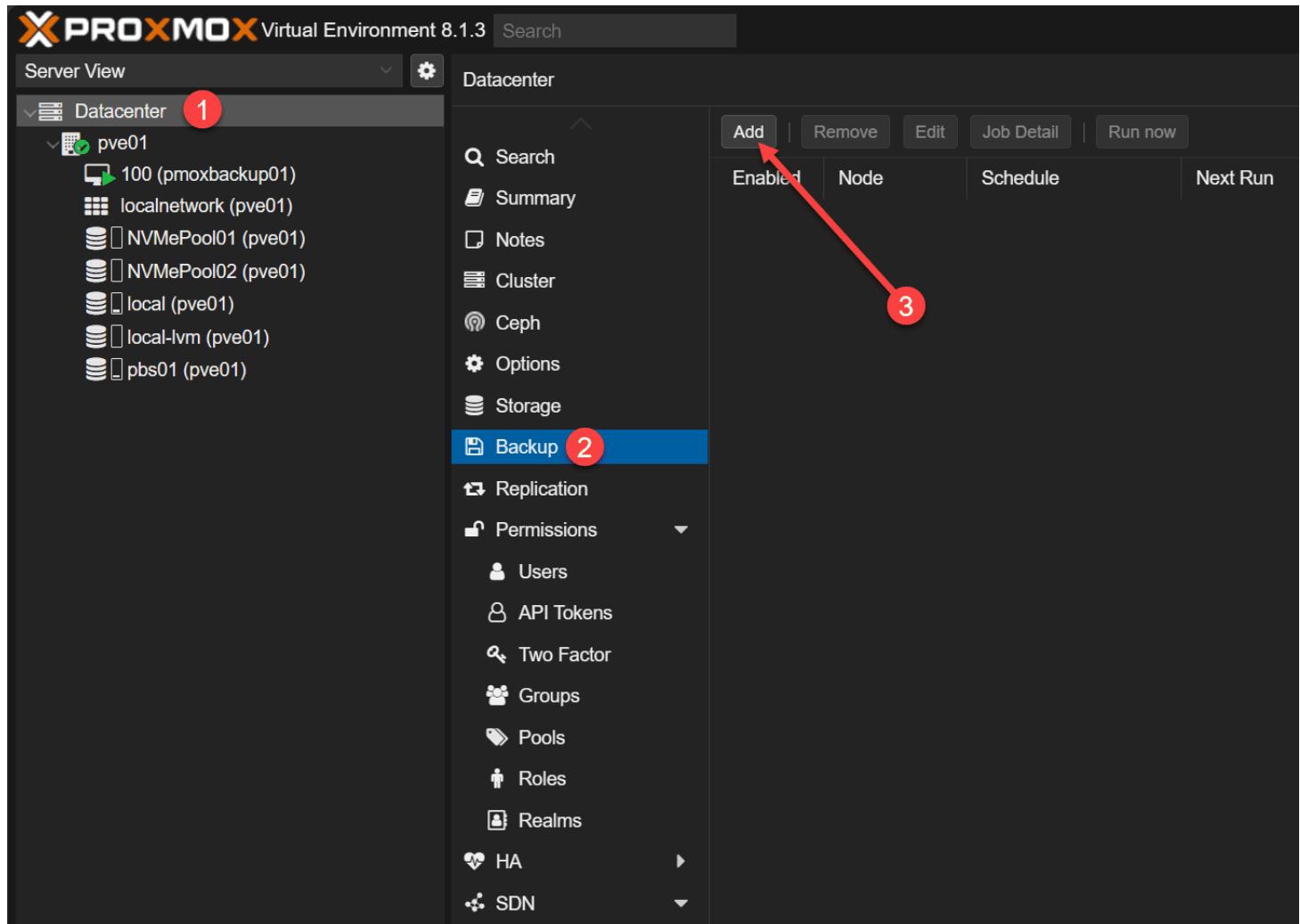
Proxmox backup server storage added successfully

Creating a backup job

Now that we have the Proxmox Backup Server added as storage to our Proxmox VE host, let's create a backup job.

You might think we would do this from the Proxmox Backup Server side. However, we create the Proxmox backup job from the Proxmox VE host. Proxmox Backup Server offers advanced features like snapshot mode and customizable backup retention policies.

Navigate to **Datacenter > Backup > Add**.



Beginning to add a new proxmox backup job

On the general tab, we can set all the main backup option selection for our Proxmox backup job. This includes:

- Source nodes from which to backup
- Storage that we want to target for the backup
- Schedule when your backup job will runReading the previous sentence, it becomes evident that determining the storage and backup schedule are crucial aspects when setting up a Proxmox backup server.
- Selection mode (which VMs)
- Notification mode
- Send email options
- Send email to for configuring an email address
- Backup mode – snapshot, suspend, stop

Create: Backup Job (X)

General Retention Note Template

Node:	-- All --	Notification mode:	Default (Auto)
Storage:	pbs01	Send email:	Always
Schedule:	21:00	Send email to:	
Selection mode:	Include selected VMs	Compression:	ZSTD (fast and good)
		Mode:	Snapshot
		Enable:	<input checked="" type="checkbox"/>

Job Comment:

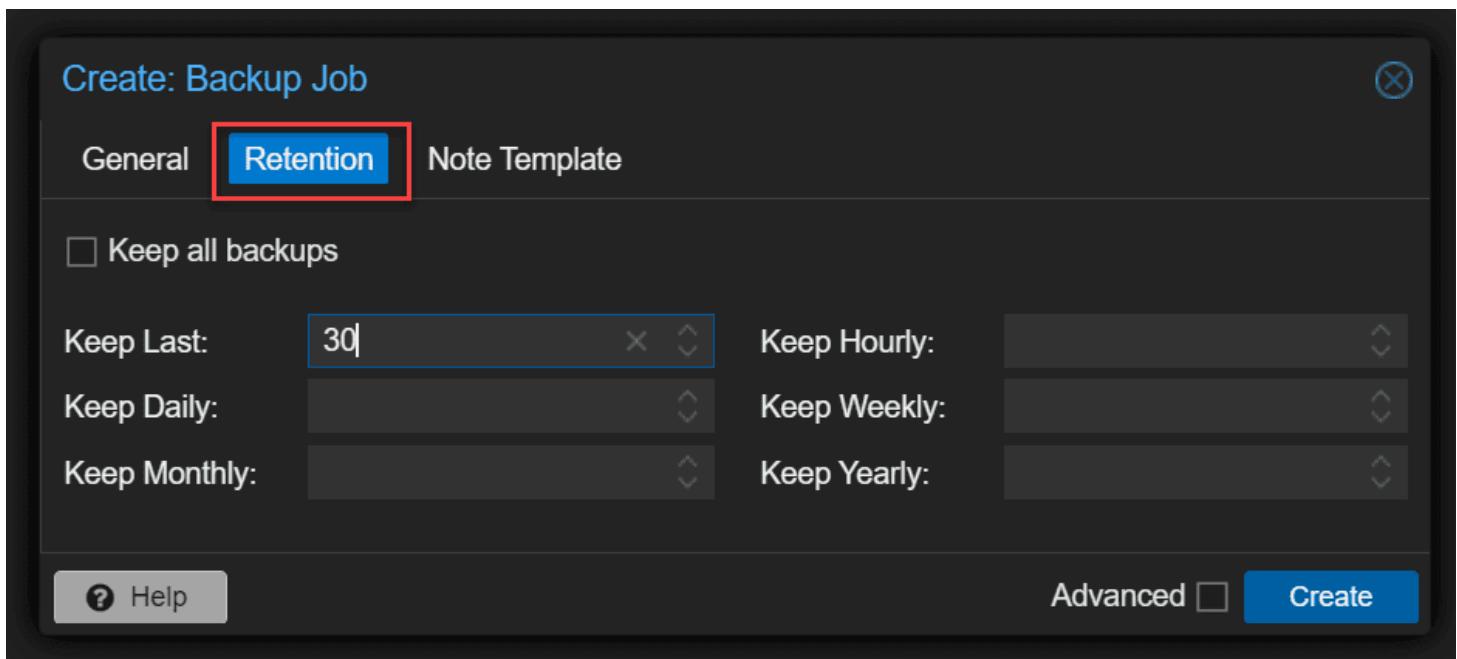
<input type="checkbox"/>	ID ↑	<input type="checkbox"/>	Node	Status	Name	Type
100	pve01	running	pmoxbackup01	Virtual Machine		
<input checked="" type="checkbox"/>	101	pve01	running	ubuntu2204	Virtual Machine	

? Help Advanced Create

The general tab on the create backup job box

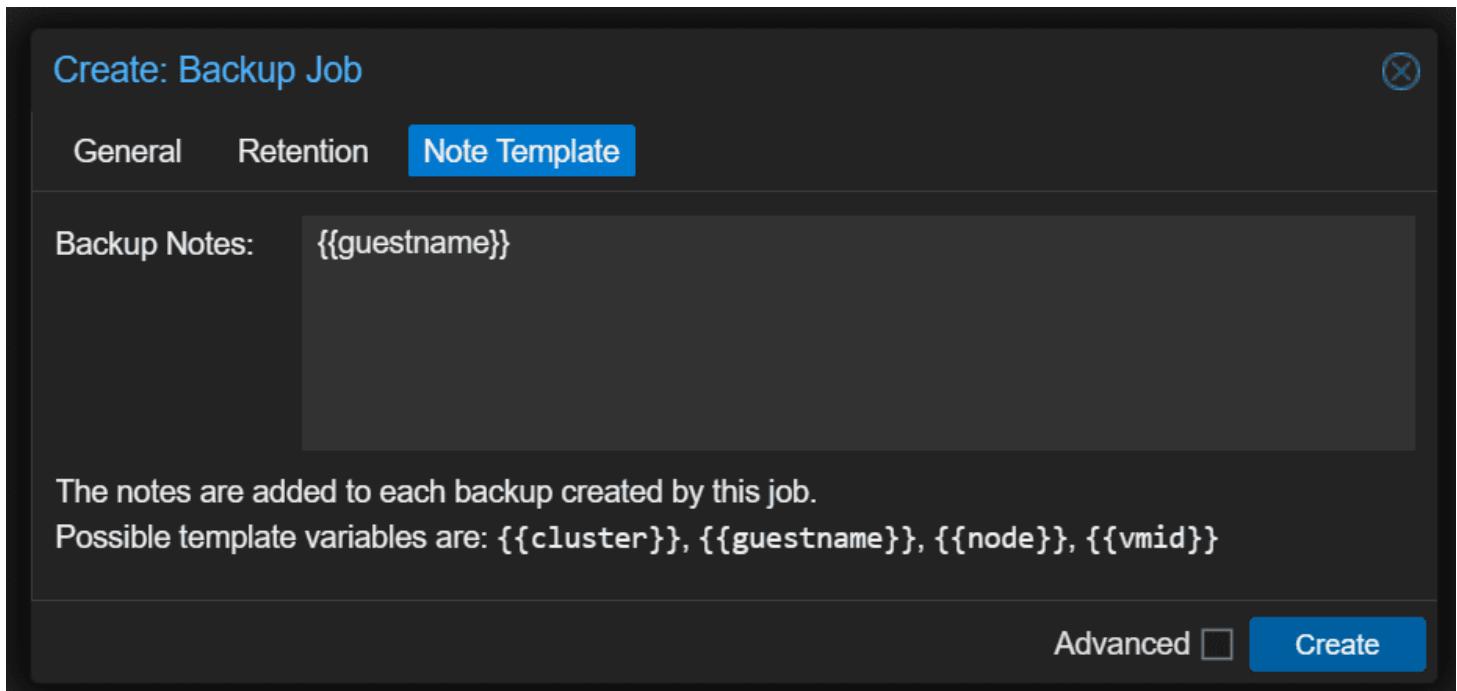
On the retention screen, you can configure all things retention and archive related, including:

- Keep all backups
- Keep last
- Keep daily
- Keep monthly
- Keep hourly
- Keep yearly



The retention tab on the new proxmox backup job box

You can also configure the **Note template**.



The note template for the new proxmox backup job

After creating the backup job.

The screenshot shows the Proxmox VE 8.1.3 interface. The left sidebar is titled "Server View" and contains a tree view with "Datacenter" selected, which has "pve01" expanded. The main area is titled "Datacenter" and shows a table of backup jobs. The table has columns: Enabled, Node, Schedule, Next Run, Storage, C., R., Sel... . A single row is visible with the following values: Enabled (checked), Node (- All --), Schedule (21:00), Next Run (2023-12-02 21:00:00), Storage (pbs01), C. (k...), R. (101). A red arrow points from the text "The newly created backup job proxmox virtual machine" to the "Schedule" column of the table. The bottom of the screen shows a "Logs" tab and a circular progress indicator.

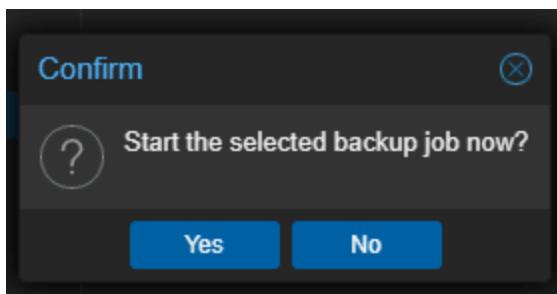
The newly created backup job proxmox virtual machine

We can choose to run the backup job now:

The screenshot shows the Proxmox VE 8.1.3 interface. In the left sidebar, under 'Datacenter', the 'Backup' option is selected. On the right, a table lists backup jobs. The 'Run now' button is highlighted with a red arrow. The table columns include: Enabled, Node, Schedule, Next Run, Storage, C..., R..., and Selection. One row is visible with the following values: Enabled (checked), Node (pve01), Schedule (All), Next Run (2023-12-02 21:00:00), Storage (pbs01), and C... (k...) R... (101).

Choosing to run the new proxmox backup job now

Confirm you want to run the backup now.



Confirm you want to run the new proxmox backup job

You will also note, that you can navigate to the Proxmox VE host virtual machine and select **Backup** and you will have the option to backup your VM from here. Make sure you choose your PBS storage location.

The screenshot shows the Proxmox VE interface. In the left sidebar, under 'Datacenter', 'pve01' is selected, and '101 (proxmoxbackup01)' is highlighted with a red box. The 'Backup' button in the sidebar is also highlighted with a red box. A red arrow points from the 'Backup' button to the 'Storage' dropdown in the central 'Backup VM 101' dialog. The dialog shows 'Storage: pbs01', 'Mode: Snapshot', and 'Protected: false'. The 'Backup' button at the bottom right of the dialog is also highlighted with a red box.

Confirm your backup storage

After you run the backup, you can see the vzdump backups details for the VM by selecting your PBS storage location.

The screenshot shows the Proxmox Virtual Environment 8.1.3 interface. On the left, the navigation tree shows 'Datacenter' with 'pve01' selected, containing '100 (pmoxbackup01)', '101 (ubuntu2204)', and several storage pools like 'localnetwork', 'NVMePool01', etc. The 'Backup' tab is selected in the center-left sidebar. The main area displays a table of backups for VM 101, located on node 'pve01'. The table has columns: Name, Notes, Date, Format, Size, Encrypted, and Verify State. Three backup entries are listed, all pointing to 'ubuntu2204' and stored in 'pbs-vm' format. The 'Storage: pbs01' dropdown menu is highlighted with a red arrow.

Name	Notes	Date	Format	Size	Encrypted	Verify State
vm/101/2023-12-03T01:47:01Z	ubuntu2204	2023-12-02 19:47:01	pbs-vm	8.59 GB	No	None
vm/101/2023-12-03T01:42:13Z	ubuntu2204	2023-12-02 19:42:13	pbs-vm	8.59 GB	No	None
vm/101/2023-12-03T01:40:19Z	ubuntu2204	2023-12-02 19:40:19	pbs-vm	8.59 GB	No	None

Look at the backups on your remote PBS datastore

Restoring a Proxmox virtual machine

After selecting your PBS backup location, you can select one of the restore points and select **Restore**

The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the 'Server View' sidebar is open, showing the Datacenter structure with nodes pve01 and pbs01. Under pve01, there is a VM named 101 (ubuntu2204). In the main pane, 'Virtual Machine 101 (ubuntu2204) on node 'pve01'' is selected. The 'Backup' tab is active. A red arrow points to the 'File Restore' button in the top navigation bar. Below it, a modal window titled 'Overwrite Restore: VM 101' is displayed. The modal contains the following fields:

Setting	Value
Source	vm/101/2023-12-03T01:47:01Z
Storage	From backup configuration
VM	101
Bandwidth Limit	Defaults to target storage restore limit
Unique	<input type="checkbox"/>
Start after restore	<input checked="" type="checkbox"/>
Live restore:	
Name	ubuntu2204
Memory	2048
Cores	1
Sockets	1

A blue 'Restore' button is at the bottom right of the modal.

Beginning the restore process for a proxmox virtual machine

You can select to start the VM and also perform a Live restore.

PROXMOX Virtual Environment 8.1.3 Search

Server View

Virtual Machine 101 (ubuntu2204) on node 'pve01' No Tags

Backup now | Restore | File Restore | Show Configuration | Edit Notes | Change Protection | Remove | Storage: pbs01 | Filter >

Name	Notes	Date ↓	Format	Size	Encrypted	Verify State
vm/101/2023-12-03T01:47:01Z	ubuntu2204	2023-12-02 19:47:01	pbs-vm	8.59 GB	No	None
vm/101/2023-12-03T01:42:13Z	ubuntu2204	2023-12-02 19:42:13	pbs-vm	8.59 GB	No	None
vm/101/2023-12-03T01:40:19Z	ubuntu2204	2023-12-02 19:40:19	pbs-vm	8.59 GB	No	None

Overwrite Restore: VM 101

Source: vm/101/2023-12-03T01:47:01Z
Storage: From backup configuration
VM: 101
Bandwidth Limit: Defaults to target storage restore limit MiB/s
Unique: Start after restore:
Live restore:

Note: If anything goes wrong during the live-restore, new data written by the VM may be lost.

Override Settings:
Name: ubuntu2204 Memory: 2048
Cores: 1 Sockets: 1

Restore

Choosing to overwrite with a live restore and power on the vm

Confirm you want to restore the VM.

PROXMOX Virtual Environment 8.1.3 Search

Server View

Virtual Machine 101 (ubuntu2204) on node 'pve01' No Tags

Backup now | Restore | File Restore | Show Configuration | Edit Notes | Change Protection | Remove | Storage: pbs01 | Filter >

Name	Notes	Date ↓	Format	Size	Encrypted	Verify State
vm/101/2023-12-03T01:47:01Z	ubuntu2204	2023-12-02 19:47:01	pbs-vm	8.59 GB	No	None
vm/101/2023-12-03T01:42:13Z	ubuntu2204	2023-12-02 19:42:13	pbs-vm	8.59 GB	No	None
vm/101/2023-12-03T01:40:19Z	ubuntu2204	2023-12-02 19:40:19	pbs-vm	8.59 GB	No	None

Overwrite Restore: VM 101

Source: vm/101/2023-12-03T01:47:01Z
Storage: From backup configuration

VM: Confirm

Unit: B/s

Live: Yes | No

Note: This will permanently erase current VM data.

Override Settings:

Name: ubuntu2204	Memory: 2048
Cores: 1	Sockets: 1

Restore

Confirm you want to overwrite the vm

The task will progress and should finish successfully. The speed will depend on the network bandwidth you have between your Proxmox VE host and your Proxmox Backup Server.

The screenshot shows the Proxmox VE 8.1.3 interface. On the left, the navigation tree includes 'Datacenter' (pve01), 'Virtual Machine 101 (ubuntu2204) on node 'pve01'', and various system components like 'localnetwork', 'NVMePool01', 'NVMePool02', 'local', 'local-lvm', and 'pbs01'. The 'Backup' section is selected. In the center, a table lists three backup snapshots for VM 101. The top snapshot is selected. A 'Task viewer' window titled 'VM 101 - Restore' is open, showing the progress of a restore task. The log output in the viewer shows multiple 'restore-drive-scsi0' entries indicating data transfer, followed by 'restore-drive-scsi0: stream-job finished' and 'restore-drive jobs finished successfully, removing all tracking block devices to disconnect from Proxmox Backup Server'. A red arrow points to the 'TASK OK' message at the bottom of the log. The status bar at the bottom of the viewer window also displays 'TASK OK'.

The restore task completes successfully

Granular file restore

You can also click the **File Restore** button to restore individual files from the backups.

Name ↑	Size	Modified	Type
- drive-scsi0.img.fidx	8.00 GiB		Virtual
- lvm			Virtual
- ubuntu-vg			Directory
- ubuntu-lv	6.25 GiB		Directory
+ bin			Directory
+ boot			Directory
+ dev			Directory
+ etc			Directory
+ home			Directory
+ lib			Directory
+ lib32			Directory
+ lib64			Directory
+ libx32			Directory
+ lost+found			Directory
+ media			Directory
+ mnt			Directory
+ opt			Directory
+ proc			Directory
+ root			Directory
+ run			Directory
+ sbin			Directory

[Download as](#) *Running a file restore for a proxmox virtual machine*

Wrapping up

Proxmox Backup Server allows you to have a way to protect your virtual machines and containers inside Proxmox. It is also great for those with [home lab](#) environments to protect VMs and containers they don't want to lose or configurations that are hard to reproduce. But it is also good for production and is ready for that use case. As shown, the solution is not hard to install. It is easy to set up backup jobs, and you can quickly restore your backups in the Proxmox VE environment.

Proxmox SDN Configuration Step-by-Step

With the release of Proxmox 8.1, Proxmox introduced new networking features in the way of Proxmox SDN, or “software defined networking” that is fully integrated out of the box for use in the datacenter. Thanks to virtualization infrastructure, Software defined networking allows taking networking into software without having the need for physical network devices to spin up new networks, subnets, IP ranges, DHCP servers, etc. Proxmox SDN allows creating these virtualized network infrastructures. This post will look at Proxmox SDN configuration step-by-step and how it is setup.

Table of contents

- [Introduction to Proxmox SDN](#)
- [Comparison with VMware NSX](#)
- [Use Cases of Proxmox SDN](#)
- [Prerequisites](#)
- [Setting Up Proxmox SDN](#)
 - [1. Create a Simple SDN Zone](#)
 - [2. Create a VNet](#)
 - [3. Create a Subnet and DHCP range](#)
 - [4. Apply the SDN configuration](#)
- [Connect Virtual Machines and Containers to the SDN network](#)
- [Key points to remember](#)
- [Wrapping up Proxmox SDN configuration](#)

Introduction to Proxmox SDN

[Virtualization is not just for compute and storage](#) or SD-WAN. Proxmox SDN is a new feature in Proxmox VE that allows you to create virtualized networks and isolated private network configurations in code. Think of it like creating your own little switch in software. These network are made up of virtual zones and networks (VNets) for communication. Using SDN, admins have much better control over networking management and virtual networks that are attached to VM guests and it is all free and open-source.

Note the following components of [Proxmox software-defined network](#):

- **Zones** – a virtually separated network configuration or area
- [Virtual networks \(VNets\)](#) – Virtual network that is part of a zone
- **Subnets** – The network IP space inside a VNet.

Comparison with VMware NSX

You have probably heard about [VMware's SDN solution called VMware NSX](#). There are many similarities with NSX and Proxmox SDN in capabilities. Arguably VMware NSX is a more robust solution that is a paid add-on to VMware vSphere. However, the Proxmox SDN solution is not as mature as VMware NSX that has been around for years now. I would like to see some of the additional micro-segmentation firewall features added to Proxmox SDN that we have in VMware NSX to create any number of connectivity rules and it can be integrated with ID sources for users, like AD domain configurations.

Use Cases of Proxmox SDN

What is the application of this technology? Using these components, you can create complex overlay networks on top of your existing network. The SDN network is a layer above the physical IP network where physical devices and hosts are connected.

Also, you can create your own isolated [private network](#) on each Proxmox VE server and span this to networks across multiple Proxmox VE clusters in many different locations.

Prerequisites

While Proxmox version 8.1 has the SDN components preloaded and the integration is available, according to the documentation, you will need to load the SDN package in Proxmox 7.X for every node in the cluster config:

```
apt update  
apt install libpve-network-perl
```

After installation, you need to ensure that the following line is present at the end of the **/etc/network/interfaces configuration** file on all nodes:

```
source /etc/network/interfaces.d/*
```

Proxmox requires the dnsmasq package for SDN functionality to enable features like DHCP management and network addressing. To install the DNSmasq packages:

```
apt update  
apt install dnsmasq  
# disable default instance  
systemctl disable --now dnsmasq
```

For advanced routing:

```
apt update  
apt install frr-pythontools
```

Setting Up Proxmox SDN

Let's take a look at setting up software defined networking SDN on a [Proxmox host](#) and enabling an existing local Linux machine to connect. In this overview, we will enable automatic DHCP on the network interface so the machine can pull an IP from the IP range.

To Install Proxmox SDN as a simple network, we will do that in the following order:

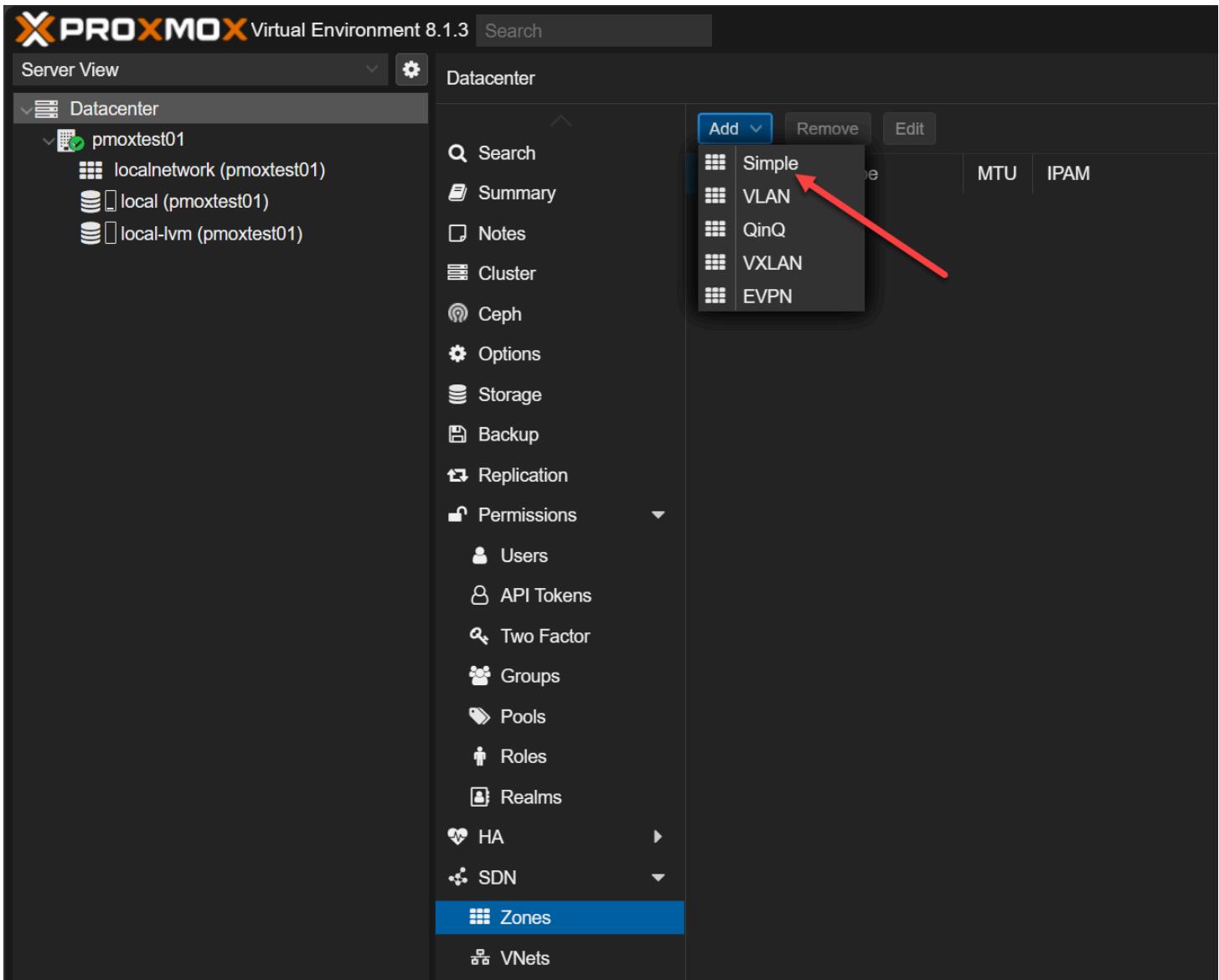
1. Create a **Simple** SDN Zone
2. Create a VNet
3. Create a Subnet and DHCP range
4. Apply the SDN configuration

1. Create a Simple SDN Zone

There are a few types of Zones you can create. These include:

- **Simple**: The simple configuration is an Isolated Bridge that provides a simple layer 3 routing bridge (NAT)
- **VLAN**: Virtual LANs enable the traditional method of dividing up a LAN. The VLAN zone uses an existing local Linux or OVS bridge to connect to the Proxmox VE host's NIC
- **QinQ**: Stacked VLAN (IEEE 802.1ad)
- **VXLAN**: Layer 2 VXLAN network that is created using a UDP tunnel
- **EVPN** (BGP EVPN): VXLAN that uses BGP to create Layer 3 routing. In this config, you create exit nodes to force traffic through a primary exit node instead of using [load balancing](#) between nodes.

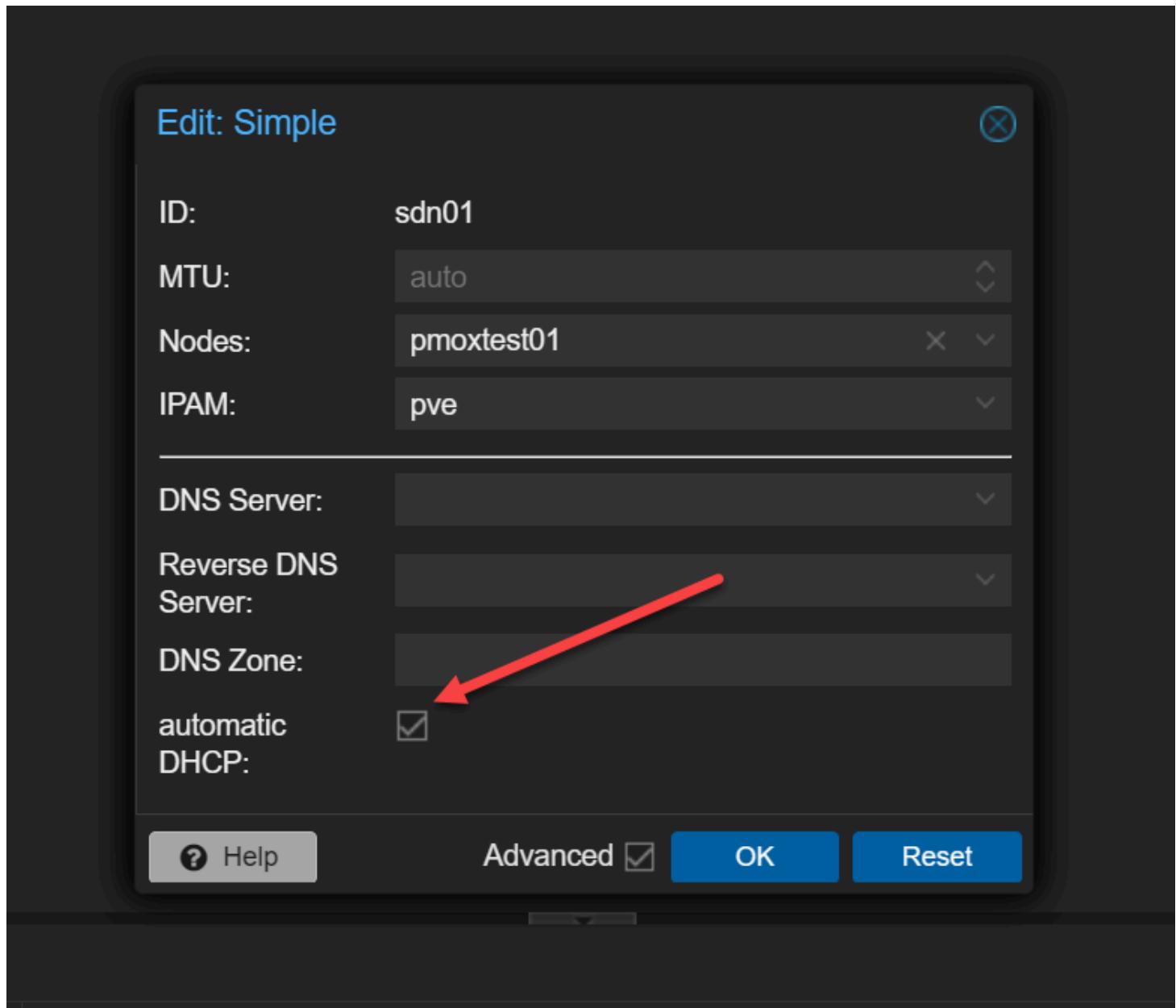
First, we need to create a new Zone. For this walkthrough, we will just be creating a **Simple** Zone. Login to your Proxmox node in a browser as root for the proper permissions. At the datacenter level, navigate to **SDN > Zones > Add**.



Creating a new zone in proxmox sdn

The SDN Zone configuration also allows you to set the zone for **automatic DHCP** configuration that will allow your VMs to pull an IP address from the VNet and Subnet configuration we will setup below. You can also set the MTU value for the size of the ethernet frames (packet), and DNS configuration, including DNS server, DNS zone, etc. In this example, I am creating a SDN Zone called **sdn01**.

The MTU value is important to note as with VXLAN, it uses 50 bytes to encapsulate the packet, you need to reduce the size by 50 bytes less than the normal MTU value. Optional will default to a size of 1450 on auto. In the case of VXLAN with IPSEC security, customers will need to reduce the MTU by 60 with IPv4, or 60 for IPv6 for guest traffic or you will see an [issue with connectivity](#) that may be a problem that is hard to uncover.



Enabling automatic dhcp

After clicking OK above, we see the new **sdn01** Simple Zone.

The screenshot shows the Proxmox VE web interface under the 'Datacenter' section. On the left, there's a sidebar with various management options like Search, Summary, Notes, Cluster, Ceph, Options, Storage, Backup, Replication, Permissions, and SDN. The SDN menu is expanded, showing sub-options for HA and SDN. The main area is titled 'SDN' and contains a table with the following data:

ID	Type	MTU	IPAM	Domain	DNS	Reverse DNS	Nodes	State
sdn01	simple		pve				pmoxtest01	new

A red arrow points to the 'simple' row in the table.

Viewing the simple zone in proxmox sdn

2. Create a VNet

Next, we need to create a VNet in PVE. Navigate to the VNet menu under the SDN menu and click to **Create** a new VNet.

The screenshot shows the Datacenter interface of a cloud provider. On the left, a sidebar lists various management options like Storage, Backup, Replication, Permissions, HA, SDN, Zones, and VNets. The VNets option is currently selected and highlighted in blue. The main area is titled 'VNets' and contains a table with columns: ID ↑, Alias, Zone, Tag, VL..., and State. At the top of this table are three buttons: 'Create', 'Remove', and 'Edit'. A red arrow points from the text 'Beginning the process to create a new vnet' down to the 'Create' button.

Beginning the process to create a new vnet

Create a name for the VNet and select the Zone we created above. You also have the option to make these VLAN aware with a tag and also create an alias.

8.1.3 Search Documentation

Datacenter

Storage
Backup
Replication
Permissions
Users
API Tokens
Two Factor
Groups
Pools
Roles
Realms
HA
SDN
Zones
VNets
Options
IPAM
ACME
Firewall

VNets

Create Remove Edit

ID ↑	Alias	Zone	Tag	VL...	State
------	-------	------	-----	-------	-------

Create: VNet

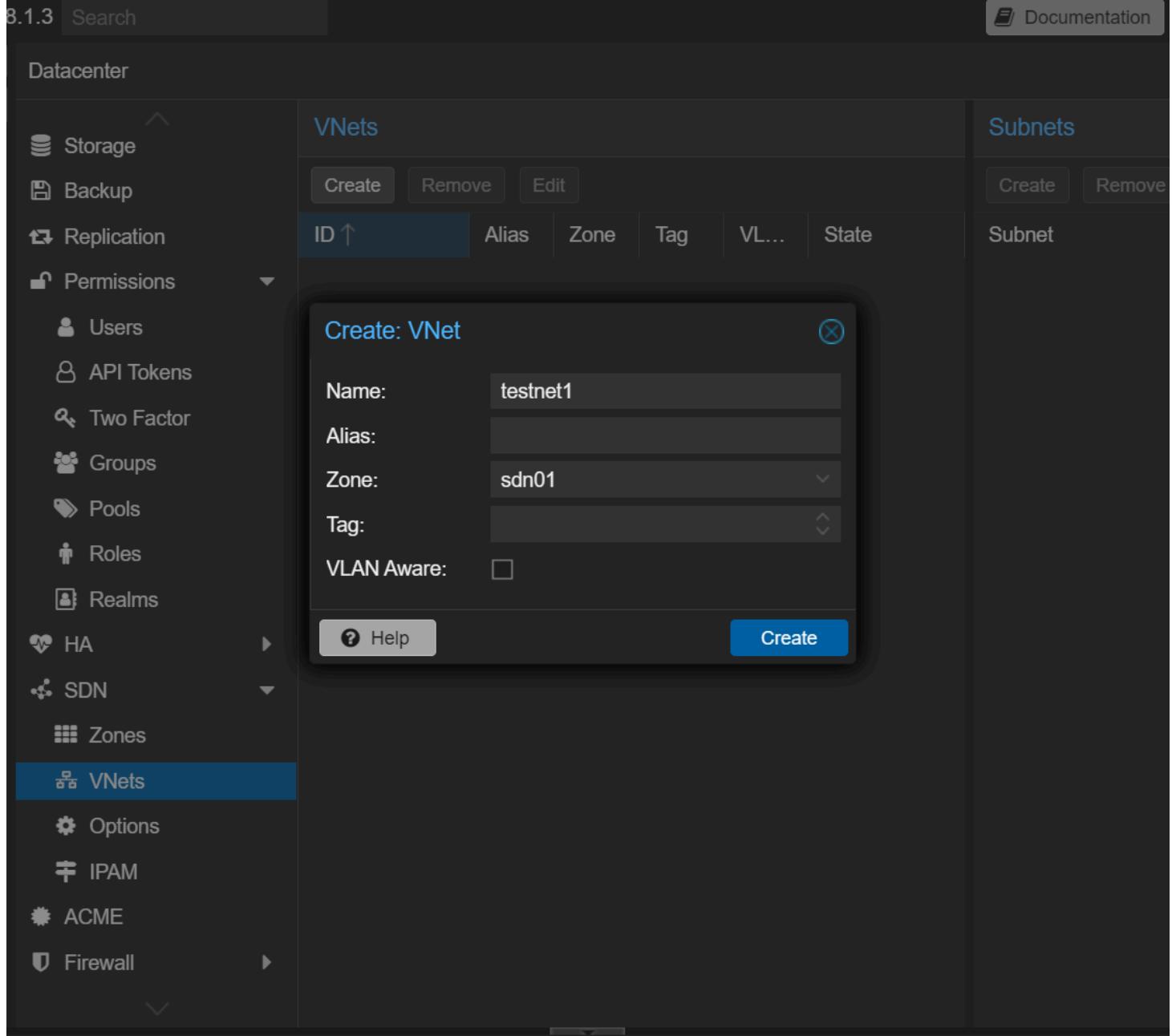
Name: testnet1
Alias:
Zone: sdn01
Tag:
VLAN Aware:

? Help Create

Subnets

Create Remove

Subnet

A screenshot of the Proxmox VE web interface. On the left, there's a sidebar with various menu items like Storage, Backup, Replication, and VNets. The VNets item is currently selected and highlighted in blue. In the main content area, there's a 'VNets' section with buttons for Create, Remove, and Edit. Below that is a table with columns for ID, Alias, Zone, Tag, VLAN, and State. A modal window titled 'Create: VNet' is open in the center, prompting for 'Name' (set to 'testnet1'), 'Alias' (empty), 'Zone' (set to 'sdn01'), 'Tag' (empty), and 'VLAN Aware' (unchecked). At the bottom of the modal are 'Help' and 'Create' buttons.

Configuring the new vnet in proxmox sdn

3. Create a Subnet and DHCP range

After creating the VNet, we can create a Subnet. Click the **Create** button on the **Subnets** screen.

The screenshot shows the Proxmox VE web interface. On the left, a sidebar lists various management sections: Storage, Backup, Replication, Permissions, Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms, HA, SDN, Zones, VNets (which is selected and highlighted in blue), Options, IPAM, ACME, and Firewall. The main content area is divided into two tabs: 'VNets' and 'Subnets'. The 'VNets' tab shows a table with one entry: 'testnet1' (ID), 'sdn...' (Alias), 'new' (State). The 'Subnets' tab shows a table with one entry: 'Subnet' (Label), 'Gate...' (Gateway), 'SNAT' (SNAT checkbox checked), 'DNS ...' (DNS checkbox checked), and 'State' (new). A red arrow points to the 'Create' button in the 'Subnets' header.

Creating a new subnet in proxmox sdn

Enter your IP address CIDR information and Gateway. If you populate the Gateway here, your Proxmox server will assume this IP address. Also, you can check the **SNAT** box. This will allow your VMs connected to the SDN network to easily connect to external networks beyond the SDN network (aka the Internet and your physical network) by masquerading as the IP and MAC of the host. Click **Create**.

8.1.3 Search Documentation

Datacenter

- Storage
- Backup
- Replication
- Permissions
 - Users
 - API Tokens
 - Two Factor
- Groups
- Pools
- Roles
- Realms
- HA
- SDN
- Zones
- VNets
- Options
- IPAM
- ACME
- Firewall

VNets

ID ↑	Alias	Zone	Tag	VL...	State
testnet1	sdn...				new

Create Subnet

General DHCP Ranges

Subnet: 192.168.55.0/24

Gateway: 192.168.55.1

SNAT:

DNS Zone Prefix:

Create



Creating a new subnet

Click on the **DHCP Ranges** and enter your start and end address for the DHCP range. It will hand out addresses from this range of IPv4 IPs.

8.1.3 Search Documentation

Datacenter

- Storage
- Backup
- Replication
- Permissions
 - Users
 - API Tokens
 - Two Factor
 - Groups
 - Pools
 - Roles
 - Realms
- HA
- SDN
 - Zones
 - VNets
 - Options
 - IPAM
 - ACME
 - Firewall

VNets

ID ↑	Alias	Zone	Tag	VL...	State
testnet1	sdn...				new

Subnets

Subnet	...
192.168.55.0/...	...

Edit: Subnet

General DHCP Ranges

Start Address	End Address	...
192.168.55.100	192.168.55.200	

Add

OK Reset



Creating a dhcp range in proxmox sdn

After clicking OK, we will see the new VNet and Subnet displayed.

The screenshot shows the Proxmox VE web interface. On the left, a sidebar menu includes options like Storage, Backup, Replication, Permissions, Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms, HA, SDN, and Zones. The 'VNets' option is selected and highlighted in blue. The main content area is divided into two tables: 'VNets' and 'Subnets'. The 'VNets' table has columns for ID (testnet1), Alias (sdn...), Zone, Tag, VL... (new), and State. The 'Subnets' table has columns for Subnet (192.168.55.0/24), Gateway (192.168.55.1), SNAT (1), DN..., and State (new).

Looking at the vnets and subnets created

We are not setting anything in the **Options** screen or **IPAM**. However, let's take a look at what those screens look like. Under the Options screen and the **Controllers** section, we can add [network controllers](#) for more advanced configurations like VXLAN to configure network tunnel configurations between peers, which are the Proxmox nodes. Under the Controllers section, we can add **EVPN**, **EBGP**, and **ISIS**.

For BGP controllers, these are not used directly by a zone. You can configure FRR to manage BGP peers. BGP-EVPN configuration define a different ASN by node. When you click the controller dropdown, you will see a list of options.

The screenshot shows the Proxmox SDN interface under the 'Datacenter' menu. The left sidebar is open, showing various management options like 'Search', 'Summary', 'Notes', 'Cluster', 'Ceph', 'Options', 'Storage', 'Backup', 'Replication', 'Permissions', 'Users', 'API Tokens', 'Two Factor', 'Groups', 'Pools', 'Roles', 'Realms', 'HA', 'SDN' (with sub-options 'Zones' and 'VNets'), 'Options' (selected), 'IPAM', 'ACME', 'Firewall', 'Metric Server', 'Resource Mappings', 'Notifications', and 'Support'. The main area is titled 'Options' and contains three sections: 'Controllers', 'IPAM', and 'DNS'.
Controllers: Shows a table with one entry: 'evpn' (Type: PVE). An 'Add' button is highlighted with a red arrow.
IPAM: Shows a table with one entry: 'pve' (Type: PVE).
DNS: Shows an empty table.

Looking at controllers and options available in proxmox sdn

The screenshot shows the Datacenter interface with the following details:

- Top Bar:** Version 8.1.3, Search bar, Documentation, Create VM, Create CT, root@pam, Help.
- Left Sidebar:** Storage, Backup, Replication, Permissions (Users, API Tokens, Two Factor, Groups, Pools, Roles, Realms), HA, SDN (Zones, VNets, Options), IPAM (selected), ACME, Firewall.
- Table Header:** Reload, Name / VMID ↑, IP Address ↑, MAC, Gateway, Actions.
- Table Body:** (Empty table area)

Looking at ipam

4. Apply the SDN configuration

It is very important to understand that creating the configuration we have created **does not apply** the configuration. It only **stages** the configuration so to speak. You need to click the **SDN** parent menu and click the **Apply** button.

The screenshot shows the Proxmox VE web interface. On the left, there's a sidebar with various management options like Datacenter, Replication, Permissions, and SDN. The SDN section is currently selected. The main content area is titled 'Status' and contains a table with three columns: SDN, Node, and Status. There is one row in the table: 'localnet...' under SDN, 'pmoxte...' under Node, and 'ok' under Status. Below the table is a large 'Apply' button. A red arrow points from the word 'Status' to this 'Apply' button.

SDN	Node	Status
localnet...	pmoxte...	ok

Apply the proxmox sdn configuration

Now we see the new SDN [network status after the configuration is applied](#) and the Proxmox networking services are restarted.

Management 8.1.3 Search Documentation Create VM

Datacenter

- Backup
- Replication
- Permissions
 - Users
 - API Tokens
 - Two Factor
 - Groups
 - Pools
 - Roles
 - Realms
- HA
- SDN**
 - Zones
 - VNets
 - Options
 - IPAM
 - ACME
 - Firewall
 - Metric Server

Status

SDN	Node	Status
localnet...	pmon...	ok
sdn01	pmon...	available

Node User name Description Status

pmon...	root@pam	SRV networking - Reload	OK
---------	----------	-------------------------	----

Viewing the new configuration applied in proxmox

Connect Virtual Machines and Containers to the SDN network

Now that we have the configuration for SDN in place on our [virtual switches](#) bridge in the hypervisor, we can connect the virtual machine or container (CT) to the new SDN network.

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

No network device

Bridge: testnet1  Model: VirtIO (paravirtualized)

VLAN Tag: no VLAN MAC address: auto

Firewall:

Help Advanced Back Next

Connecting a new virtual machine to the proxmox sdn network

Below, you see the summary screen of creating a new virtual machine and we see I have connected it to the new SDN network.

Create: Virtual Machine



General OS System Disks CPU Memory Network **Confirm**

Key ↑	Value
cores	2
cpu	x86-64-v2-AES
ide2	local:iso/ubuntu-22.04.4-live-server-amd64.iso,media=cdrom
memory	2048
net0	<u>virtio,bridge=testnet1,firewall=1</u>
nodename	pmoxtest01
numa	0
ostype	I26
scsi0	local-lvm:20,iothread=on
scsihw	virtio-scsi-single
sockets	1
vmid	100

Start after created

Advanced

Back

Finish

Node

User name

Description

Summary of new vm creation details

After installing Ubuntu, the VM correctly grabs a DHCP address from the range configured. Also, we can ping the gateway that was established in the configuration. Keep in mind how cool this really is. We have a network with total separation from the other physical network technologies for VM traffic and it is totally defined in software.

```

linuxadmin@ubuntu01:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:01:d6:ab brd ff:ff:ff:ff:ff:ff
    altnet enp0s18
    inet 192.168.55.100/24 metric 100 brd 192.168.55.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe01:d6ab/64 scope link
        valid_lft forever preferred_lft forever
linuxadmin@ubuntu01:~$ ping 192.168.55.1 ←
PING 192.168.55.1 (192.168.55.1) 56(84) bytes of data.
64 bytes from 192.168.55.1: icmp_seq=1 ttl=64 time=0.709 ms
64 bytes from 192.168.55.1: icmp_seq=2 ttl=64 time=0.359 ms
64 bytes from 192.168.55.1: icmp_seq=3 ttl=64 time=0.390 ms
64 bytes from 192.168.55.1: icmp_seq=4 ttl=64 time=0.438 ms
64 bytes from 192.168.55.1: icmp_seq=5 ttl=64 time=0.383 ms
64 bytes from 192.168.55.1: icmp_seq=6 ttl=64 time=0.395 ms
64 bytes from 192.168.55.1: icmp_seq=7 ttl=64 time=0.497 ms
64 bytes from 192.168.55.1: icmp_seq=8 ttl=64 time=0.348 ms

```

New virtual machine pulls a dhcp address from proxmox sdn

Key points to remember

Let's consider a few key points to remember about the Proxmox SDN solution.

Network Interfaces and VLAN Configuration

Network interfaces are the gateways between your [virtual machines](#) and the broader network (Internet). Make sure to give attention to detail to configure these correctly for proper connectivity and optimal performance.

VLANs enable you to segment your network into isolated sections. With VLANs you can [create a secure](#), organized network zones.

VXLAN Zone Implementation

VXLAN zones extend VLAN capabilities and create overlay networks across even different physical network locations. With VXLAN, you can build a complex, scalable network architecture.

Advanced Proxmox SDN Features

Some of the advanced Proxmox SDN features include automatic DHCP assignment to IP address management. Understand how you can use these features to enhance your network management.

Virtual Zones and Traffic Isolation

Creating virtual zones within Proxmox SDN allows network traffic segregation. This enhances the [security and performance of your network](#). Traffic isolation is crucial for security.

Wrapping up Proxmox SDN configuration

The new Proxmox SDN features in Proxmox 8.1 and above are a great new feature that allows you to create new networks quickly and easily in software. Networking has traditionally been a challenge to configure quickly and easily since physical network devices and configurations have to be changed. With SDN, all of this changes with the network overlay created. The underlying physical network no longer has to be updated, like [network switches](#), or changed for new networks and connectivity to be created.

Proxmox SDN is easy to configure and you can create a simple new network as shown in the walkthrough to start playing around with the new feature in your home lab. Let me know in the comments or VHT forum if you have played around with Proxmox SDN as of yet and what use cases you are finding in the home lab.

New Proxmox Datacenter Manager Released Download and Install

In case you haven't heard, Proxmox has released a new Proxmox DataCenter Manager project that is set to help admins have a centralized view of all their individual Proxmox VE server nodes and Proxmox clusters. You can also do basic operations with it at this point from a central console. Let's look at the new Proxmox Datacenter Manager download and install and get more info on this cool new tool!

Table of contents

- [What is Proxmox Datacenter Manager?](#)
- [Installing Proxmox Datacenter Manager](#)
- [Adding a Proxmox VE Server Node to PDM](#)
- [Viewing remote node in Proxmox Datacenter Manager](#)
- [Proxmox Datacenter Manager permissions management](#)
- [Thoughts about the new tool](#)

What is Proxmox Datacenter Manager?

Well, it is currently in ALPHA stage at this point so still rough around the edges as can be expected. But, it provides that centralized view of your Proxmox estate which is something that has been needed for quite some time now, especially for admins managing multiple standalone nodes or clusters.

You can also perform basic operations management for your Proxmox environment using the tool. You can do things like migrate virtual machines and can do this without some of the cluster requirements needed.

Proxmox has made note that in the ALPHA stage there will be features that won't make it into the release until after the 1.0 GA release is final. They make note many of the features being considered will help with "**pain points that one currently faces when managing big and/or physically distributed Proxmox VE and Proxmox Backup Server setups.**"

Here are some of the details of the release itself:

Released 19. December 2024

- Debian Bookworm (12.8) platform
- Latest 6.8.12-5 Kernel as stable default
- Newer 6.11 Kernel as optional
- ZFS: 2.2.6 patches for Kernel 6.11)

Download Proxmox Datacenter Manager here:

- [ISO Download](#)

Installing Proxmox Datacenter Manager

Let's look at the process of installing Proxmox Datacenter Manager. The process is very much akin to [installing Proxmox VE Server](#), Proxmox Backup Server, etc as the screens will be very familiar.

Select **Install Proxmox Datacenter Manager (Graphical)**.



Welcome to Proxmox Datacenter Manager

[Install Proxmox Datacenter Manager \(Graphical\)](#)

[Install Proxmox Datacenter Manager \(Console\)](#)

[Advanced Options](#)

enter: select, arrow keys: navigate, e: edit entry, esc: back

Beginning the installation of proxmox datacenter manager

Accept the end user license agreement.

PROXMOX Datacenter Manager Installer

END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX DATACENTER MANAGER

By using Proxmox Datacenter Manager software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintenance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox Datacenter Manager and any related update, source code and structure (the Programs), regardless of the delivery mechanism.

1. **License.** Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.

2. **Limited Warranty.** The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non-infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.

3. **Limitation of Liability.** To the maximum extent permitted under applicable law, under no

Abort

Previous

I agree

Accept the eula

The machine will boot and should grab a DHCP address to get started.

```
Welcome to the Proxmox Datacenter Manager 0.1 installer
initial setup startup
mounting proc filesystem
mounting sys filesystem
boot commandline: BOOT_IMAGE=/boot/linux26 ro ramdisk_size=16777216 rw quiet spla
loading drivers: i2c_piix4 pata_acpi ahci vmw_pvscsi mac_hid input_leds serio_r
ssse3 ghash_clmulni_intel polyval_clmulni crc32_pcimul crct10dif_pcimul intel_ra
searching for block device containing the ISO proxmox-datacenter-manager-0.1-ALP
with ISO ID '087a1226-be09-11ef-b5e8-cbd47817244d'
testing device '/dev/sr0' for ISO
found Proxmox Datacenter Manager ISO
preparing installer mount points and working environment
switching root from initrd to actual installation system
Starting Proxmox installation
Installing additional hardware drivers
Starting hotplug events dispatcher: systemd-udevd.
Synthesizing the initial hotplug events (subsystems)...done.
Synthesizing the initial hotplug events (devices)...done.
Waiting for /dev to be fully populated...done.
mount: devpts mounted on /dev/pts.
/bin/dbus-daemon
starting D-Bus daemon
Attempting to get DHCP leases... Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/ens192/00:50:56:91:78:97
Sending on   LPF/ens192/00:50:56:91:78:97
Sending on   Socket/fallback
DHCPDISCOVER on ens192 to 255.255.255.255 port 67 interval 8
DHCPOffer of 10.1.149.150 from 10.1.149.10
DHCPREQUEST for 10.1.149.150 on ens192 to 255.255.255.255 port 67
DHCPACK of 10.1.149.150 from 10.1.149.10
```

Booting proxmox datacenter manager for the first time

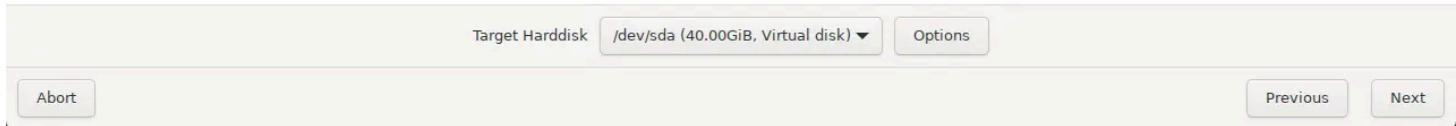
Select the target disk for installation.

Proxmox Datacenter Manager (PDM)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data on the selected disks will be lost.

To continue the installation, press the Next button.

- **Please verify the installation target**
The displayed hard disk will be used for the installation.
Warning: All existing partitions and data on selected disks will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface, via a web browser.



Select the target hard disk for installation

Choose the country, time zone, and keyboard layout for the installation. Click Next.

PROXMOX Datacenter Manager Installer

Location and Time Zone selection

The Proxmox Installer automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.

A screenshot of the Proxmox Datacenter Manager Installer's configuration interface. It shows three dropdown menus for selecting location parameters. The 'Country' dropdown is set to 'United States'. The 'Time zone' dropdown is set to 'America/Chicago'. The 'Keyboard Layout' dropdown is set to 'U.S. English'. Below these fields are two buttons: 'Abort' on the left and 'Previous' and 'Next' on the right. A descriptive text 'Choose the country time zone and keyboard layout' is centered above the form.

Country	United States
Time zone	America/Chicago
Keyboard Layout	U.S. English

Abort Previous Next

Choose the country time zone and keyboard layout

Configure your root password and email address. Click Next.

Administration Password and Email Address

Proxmox Datacenter Manager is a full-featured, highly secure system, based on Debian GNU/Linux.

In this step, please provide the *root* password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Proxmox Datacenter Manager will send important alert notifications to this email account (all emails for 'root').

To continue the installation, press the Next button.

Password █████████████████████

Confirm █████████████████████

Email admin@cloud.local

Abort Previous Next

Set the password for proxmox datacenter manager pdm

Set your hostname, IP address, Gateway, and [DNS server](#). Click Next.

Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

The screenshot shows a configuration form for a Proxmox Datacenter Manager. It includes fields for Management Interface (set to ens192 - 00:50:56:91:78:97 (vmxnet3)), Hostname (FQDN) (set to pdm.cloud.local), IP Address (CIDR) (set to 10.1.149.150 / 24), Gateway (set to 10.1.149.1), and DNS Server (set to 10.1.149.10). Buttons for Abort, Previous, and Next are at the bottom.

Management Interface	ens192 - 00:50:56:91:78:97 (vmxnet3)
Hostname (FQDN)	pdm.cloud.local
IP Address (CIDR)	10.1.149.150 / 24
Gateway	10.1.149.1
DNS Server	10.1.149.10

Abort Previous Next

Configure your proxmox datacenter manager hostname ip address gateway and dns server

View the summary screen and by default it will leave the automatically reboot after successful installation checked. Click **Install**.

PROXMOX Datacenter Manager Installer

Summary

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	United States
Timezone:	America/Chicago
Keymap:	en-us
Email:	admin@cloud.local
Management Interface:	ens192
Hostname:	pdm
IP CIDR:	10.1.149.150/24
Gateway:	10.1.149.1
DNS:	10.1.149.10

Automatically reboot after successful installation

Abort Previous Install

Proxmox datacenter manager installation summary

The installation is successful. Click **Reboot**.

PROXMOX Datacenter Manager Installer

Installation successful!

Proxmox Datacenter Manager is now installed and ready to use.

- **Next steps**

Reboot and point your web browser to the selected IP address on port 8443:

<https://10.1.149.150:8443>

Also visit www.proxmox.com for more information.

Automatic reboot scheduled in 2 seconds.

[Abort](#)

[Reboot](#)

Successful installation of proxmox datacenter manager

The Proxmox Datacenter Manager will boot.

GNU GRUB version 2.06-13+pmx2

*Proxmox Datacenter Manager GNU/Linux
Advanced options for Proxmox Datacenter Manager GNU/Linux
Memory test (memtest86+x64.bin)
Memory test (memtest86+x64.bin, serial console)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 2s.

Booting up proxmox datacenter manager pdm

Below is a view of the console. Looks familiar.

Welcome to the Proxmox Datacenter Manager. Please use your web browser to configure this server - connect to:

<https://10.1.149.150:8443/>

pdm login: _

Pdm console

Below, we are logging into the Proxmox Datacenter Manager as our configured root account.

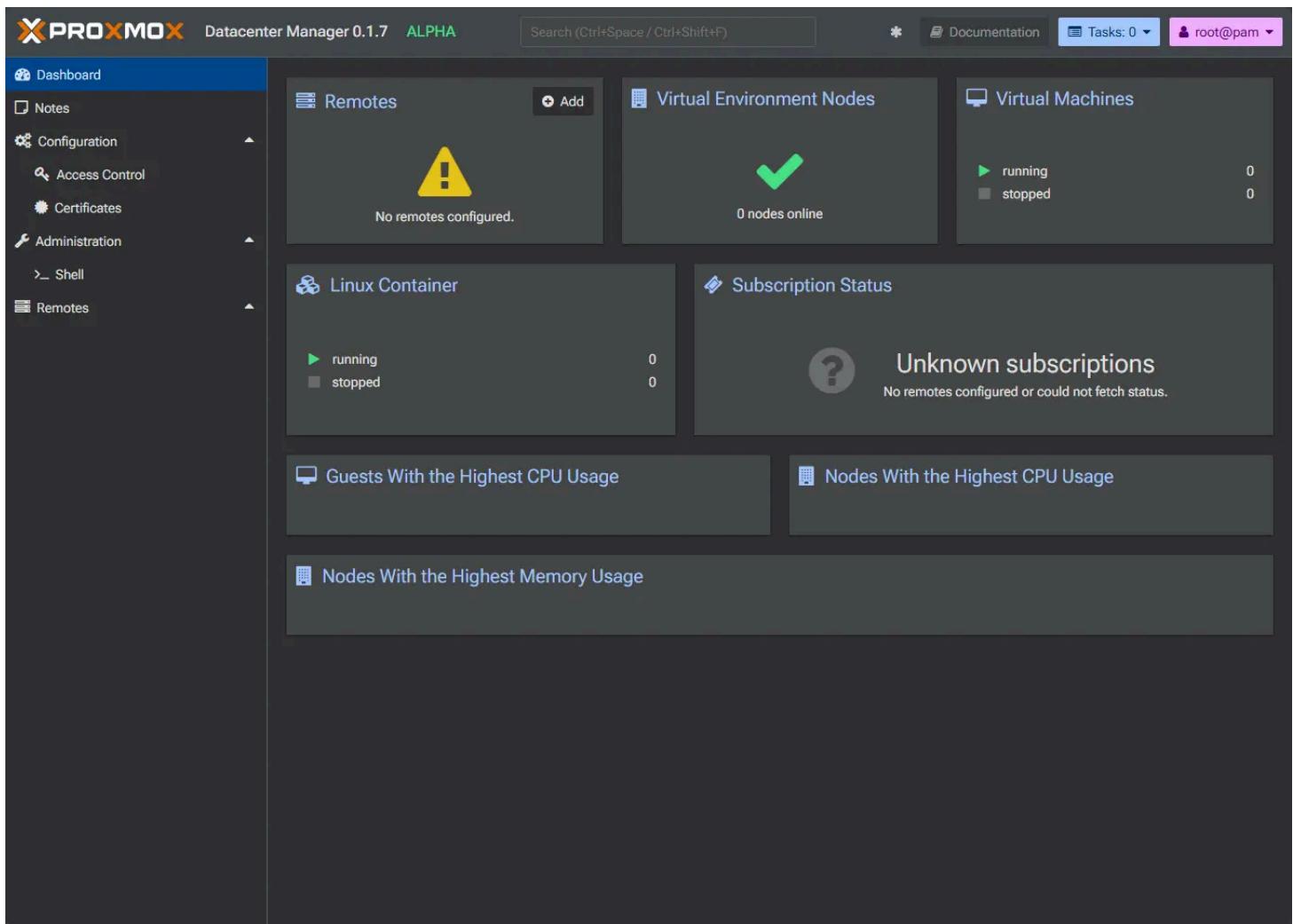
Proxmox Datacenter Manager Login

User name	root
Password	***** 
Realm	pam

English ▾ Save User name Reset **Login**

Logging into pdm for the first time

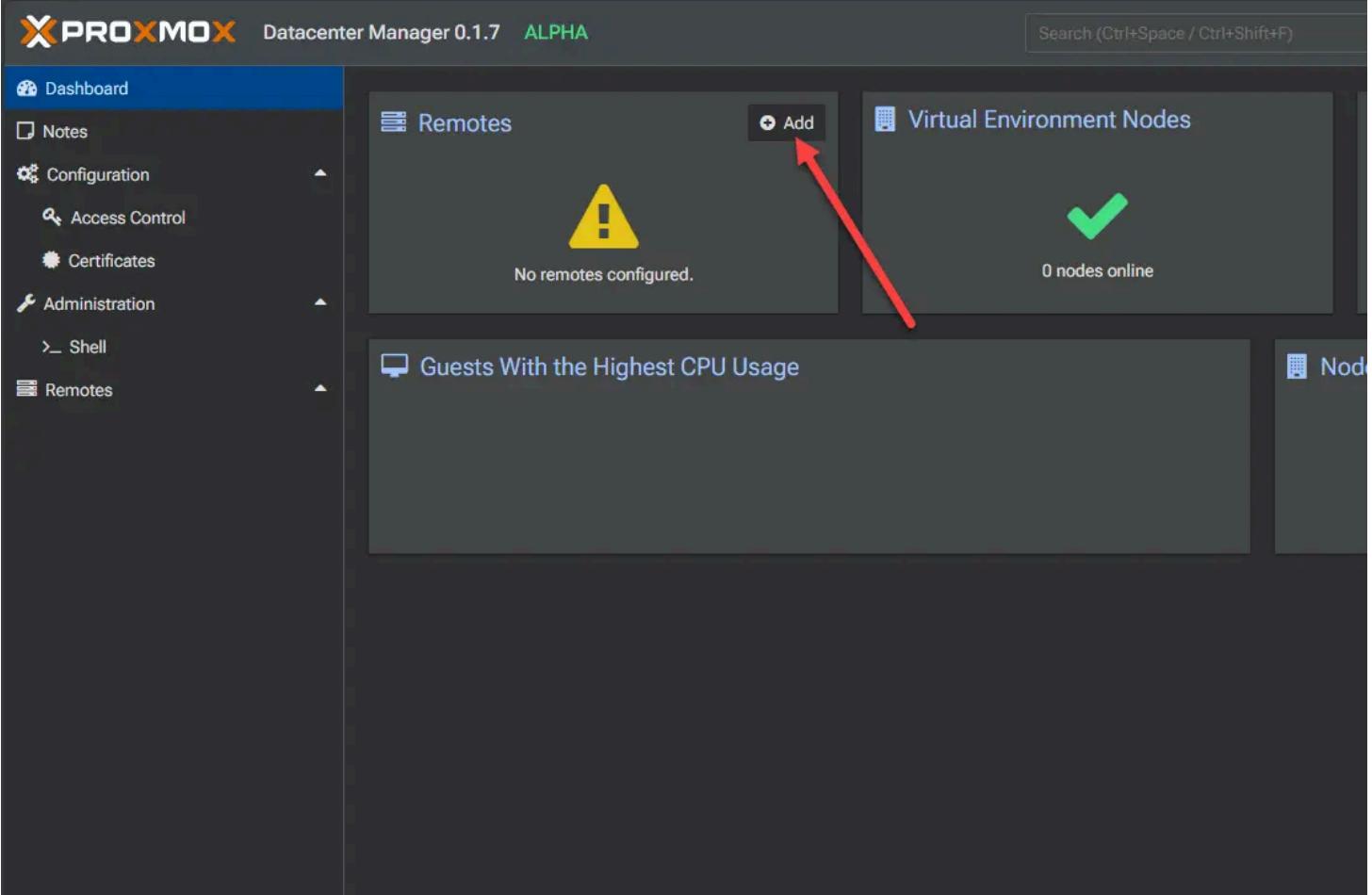
Viewing the dashboard for the first time.



Viewing the pdm dashboard for the first time

Adding a Proxmox VE Server Node to PDM

Now let's look at how to add a Proxmox node to our Proxmox Datacenter Manager instance. In the **Remotes** tile, you will see an **Add** button in the top right-hand corner.



Click add on the remotes tile

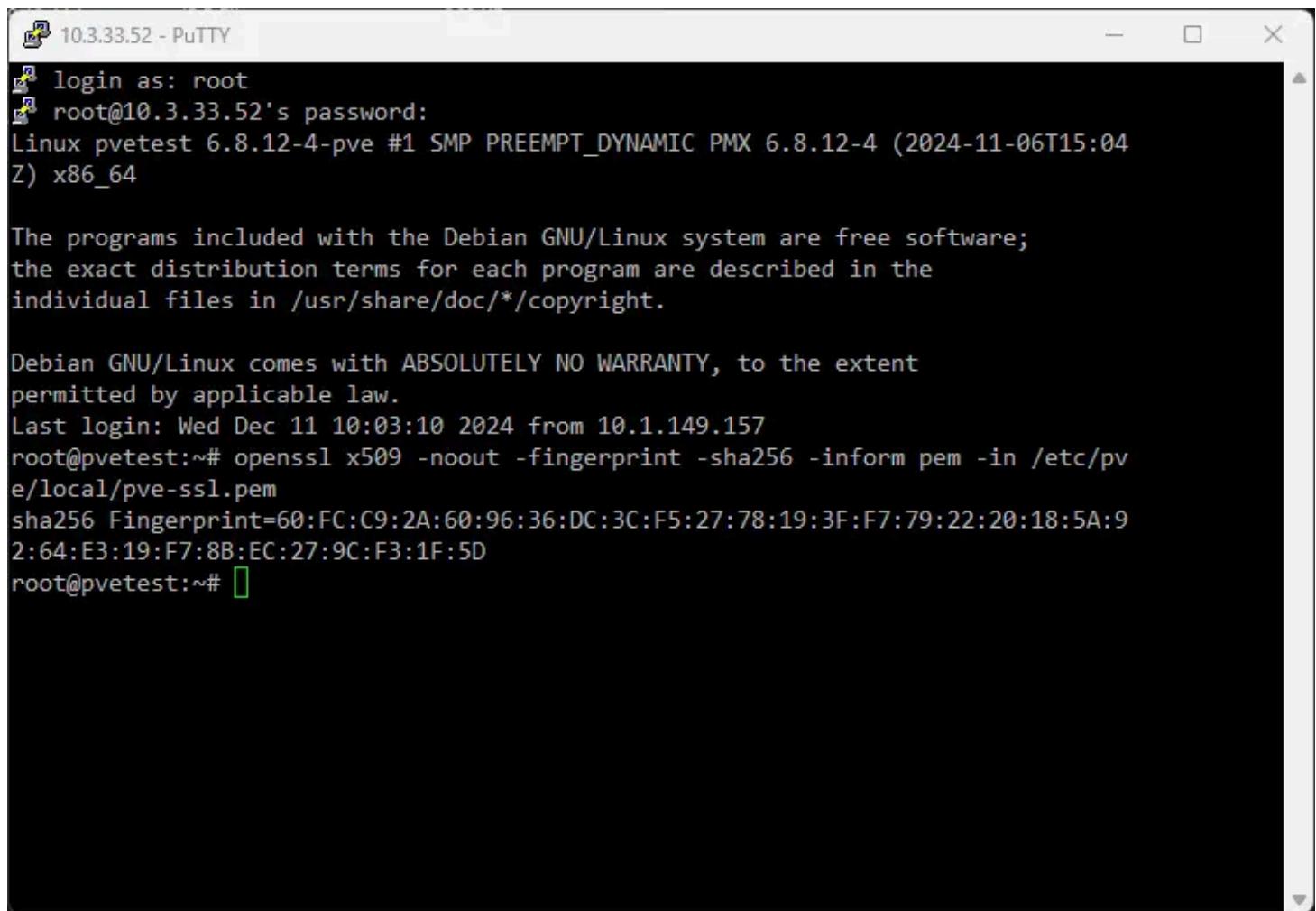
This will launch the **Add Remote** dialog box. The first tab is **Probe Remote**. There are a couple of fields here of information we need to populate to connect to a node. The server address and Fingerprint.

The screenshot shows the Proxmox Datacenter Manager interface. The top bar displays the logo and the version "Datacenter Manager 0.1.7 ALPHA". A search bar is located at the top right. On the left, a sidebar menu includes "Dashboard", "Notes", "Configuration", "Access Control", "Certificates", "Administration", "Shell", and "Remotes". The main area features three cards: "Remotes" (warning icon, "No remotes configured."), "Virtual Environment Nodes" (green checkmark, "0 nodes online"), and "Guests With the Highest CPU Usage". A modal dialog box titled "Add Remote" is open in the foreground. It has tabs for "Probe Remote", "Settings", "Endpoints", and "Summary", with "Probe Remote" selected. The "Server Address" field contains "<IP/Hostname>:Port". The "Fingerprint" field is described as "Server certificate SHA-256 fingerprint, required for self-signed certificates". Buttons for "Connect" and "Next" are visible.

Viewing the add remote dialog box in proxmox datacenter manager

You can get the fingerprint of the certificate you will need for the above screen using this command:

```
openssl x509 -noout -fingerprint -sha256 -inform pem -in /etc/pve/local/pve-ssl.pem
```



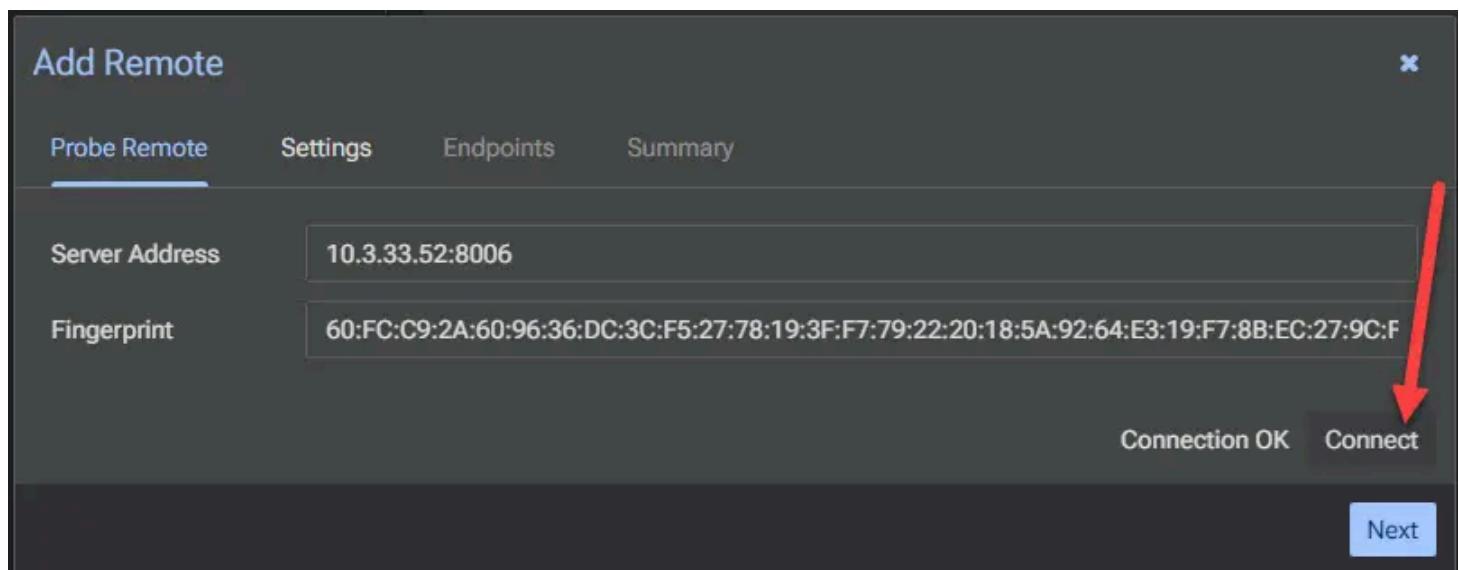
```
10.3.33.52 - PuTTY
login as: root
root@10.3.33.52's password:
Linux pvetest 6.8.12-4-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-4 (2024-11-06T15:04
Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 11 10:03:10 2024 from 10.1.149.157
root@pvetest:~# openssl x509 -noout -fingerprint -sha256 -inform pem -in /etc/pv
e/local/pve-ssl.pem
sha256 Fingerprint=60:FC:C9:2A:60:96:36:DC:3C:F5:27:78:19:3F:F7:79:22:20:18:5A:9
2:64:E3:19:F7:8B:EC:27:9C:F3:1F:5D
root@pvetest:~#
```

Getting the fingerprint of the certificate



Add Remote

Probe Remote Settings Endpoints Summary

Server Address: 10.3.33.52:8006

Fingerprint: 60:FC:C9:2A:60:96:36:DC:3C:F5:27:78:19:3F:F7:79:22:20:18:5A:92:64:E3:19:F7:8B:EC:27:9C:F3:1F:5D

Connection OK Connect

Next

Entering the server address and fingerprint

Add Remote

Probe Remote **Settings** Endpoints Summary

Remote ID	pvetest01
<input checked="" type="radio"/> Login and create Token	<input type="radio"/> Use existing Token
User	root
Password	*****
Realm	pam
API Token Name	pdm-admin

Scan OK Scan

Back Next

Scan the node

At this point, if we look at our PVE node that we are adding directly, navigate to **Datacenter > API Tokens**, we will see the API token that is autogenerated by the PDM add node process for onboarding.

PROXMOX Virtual Environment 8.3.0

Server View Datacenter Documentation Create VM Create CT root@pam

 Datacenter

Datacenter				
Add Edit Remove Show Permissions				
User name	Token Name	Expire	Comment	Privilege Sep...
root	@pam	packer	never	No
root	@pam	pdm-admin	never	auto-generated by PDM host 'pdm' on Fri, 20 Dec 20 ..

 Datacenter

- Search
- Summary
- Notes
- Cluster
- Ceph
- Options
- Storage
- Backup
- Replication
- Permissions
- Users
- API Tokens**
- Two Factor
- Groups
- Pools
- Roles
- Realms
- HA
- SDN
- Zones
- VNets
- Options

Pdm token is autogenerated by credentials you populate when adding a node

Going back to the Add Remote dialog, next we go to **Endpoints**. here we can just click Next.

Add Remote

x

Probe Remote Settings Endpoints Summary

Define a set of addresses that Proxmox Datacenter Manager can use to reach the cluster or single node. Fingerprints are required for self-signed certificates.

Hostname	Fingerprint	
10.3.33.52:8006	60:FC:C9:2A:60:96:36:DC:3C:F5:27:78:19:3F:F7:79:22:20:18:5A:92:64:E3:...	
pvetest	60:FC:C9:2A:60:96:36:DC:3C:F5:27:78:19:3F:F7:79:22:20:18:5A:92:64:E3:...	

Add

Back

Next

Viewing the endpoints tab

Finally, the Summary screen. Click **Finish**.

Add Remote

x

Probe Remote Settings Endpoints Summary

Remote ID pvetest01 Create Token Yes (pdm-admin)

Auth ID root@pam

Connections

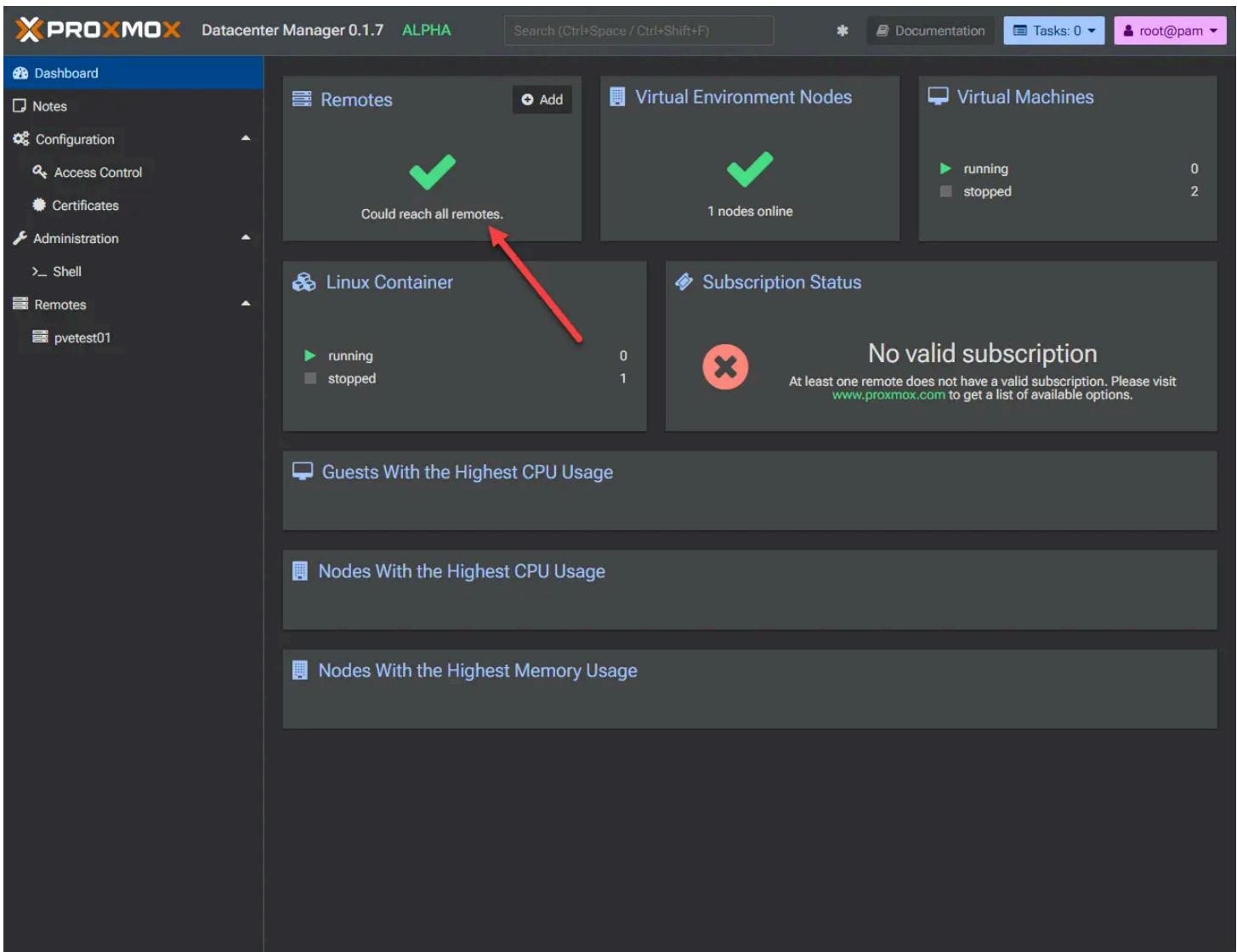
Hostname/Address	Fingerprint
10.3.33.52:8006	60:FC:C9:2A:60:96:36:DC:3C:F5:27:78:19:3F:F7:79:22:20:18:5A:92:64:E3:...
pvetest	60:FC:C9:2A:60:96:36:DC:3C:F5:27:78:19:3F:F7:79:22:20:18:5A:92:64:E3:...

Back

Finish

View the summary screen for adding the node

On the dashboard we see the message **Could reach all remotes**.



Nodes can be reached

Viewing remote node in Proxmox Datacenter Manager

Now that we have a node added to Proxmox Datacenter Manager, we can click the node and see various information about the node, including the node details like CPU usage and Server load. As a side point, I like the color scheme better in this UI than the node UI for the proxmox host. Just my personal preference and observation.

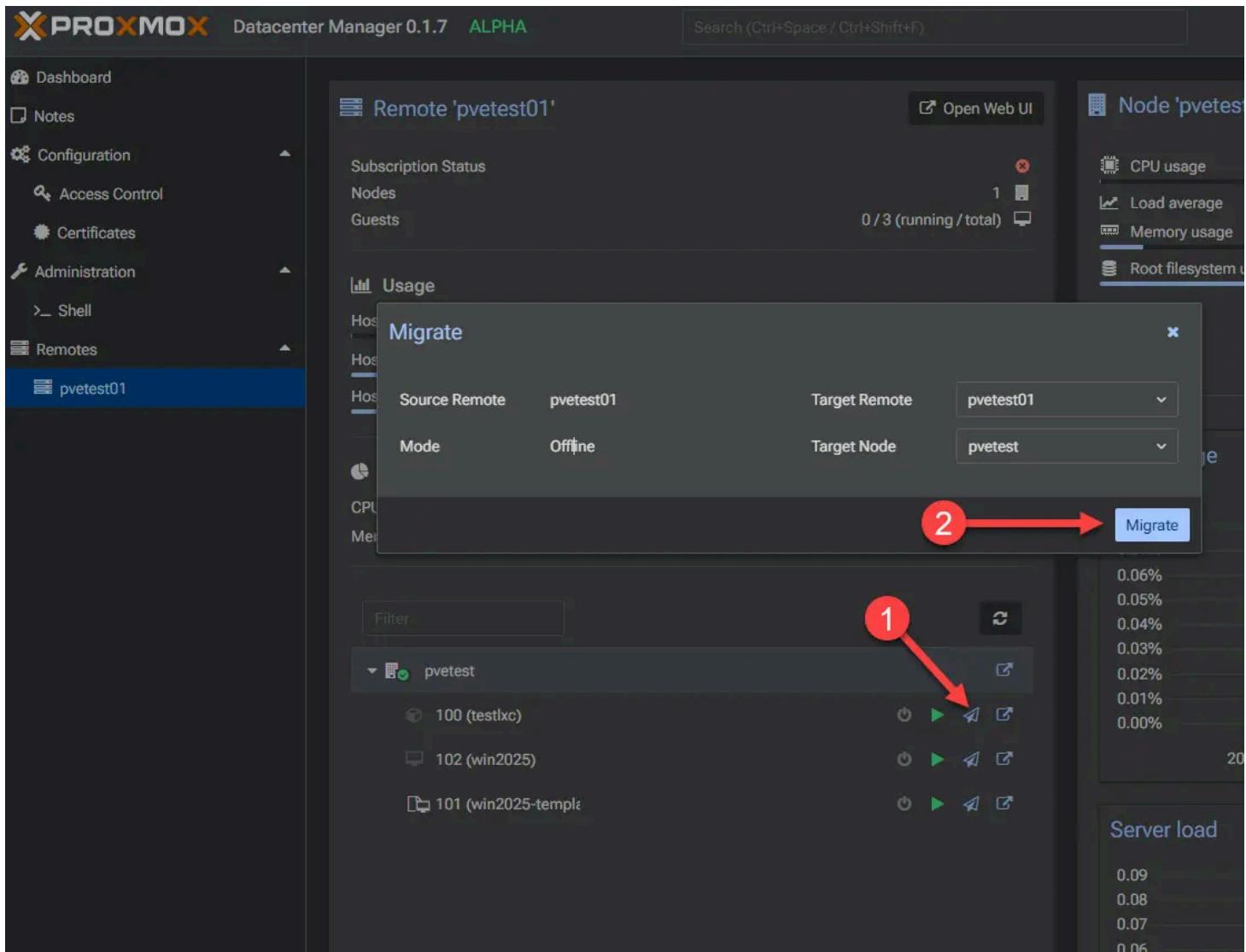
By the VMs shown under the node, you will see the little power button, play button, “airplane” button, and maximize looking button. These are:

1. Power status
2. Start, stop, etc
3. Migration
4. Shortcut to go directly to the VM in the PVE host interface

The screenshot shows the Proxmox Datacenter Manager interface. On the left sidebar, under the 'Remotes' section, the node 'pvetest01' is selected, indicated by a blue background and a red arrow pointing to it. The main content area displays the 'Remote 'pvetest01'' page, which includes sections for Subscription Status, Usage (Host CPU, Memory, Storage), Allocation (CPU Cores assigned, Memory assigned), and a list of virtual machines (100 (testlxc), 102 (win2025), 101 (win2025-template)). A red box highlights the actions (Power, Start, Stop, Migrate, Detach) for each VM. To the right, the 'Node 'pvetest'' page shows detailed resource usage statistics and three performance graphs: CPU Usage, Server load, and Memory Usage.

Viewing the remote proxmox node and virtual machine actions

If we click the migrate button you will see the Migrate dialog box. We can see some of the capabilities here. I just have a [single host](#) added so don't really have a target in this case. But Proxmox has noted this tool allows for migrating between nodes that don't meet cluster requirements.



Migrating a virtual machine functionality

Proxmox Datacenter Manager permissions management

One of the advantages of having a tool like this is centralized permissions management. Using Proxmox Datacenter Manager, admins will be able to add users and hopefully manage access to various nodes and resources using the permissions model here instead of granting access to individual Proxmox nodes.

One of the things I really like that Proxmox has natively is two-factor authentication. You can also set that up directly in the Proxmox host web UI, but this looks to also be available in Proxmox Datacenter Manager which is great to see.

The screenshot shows the Proxmox Datacenter Manager 0.1.7 ALPHA interface. On the left is a sidebar with navigation links: Dashboard, Notes, Configuration, Access Control (selected), Certificates, Administration, Shell, Remotes, and pvetest01. The main area has tabs for User Management and Two Factor Authentication, with User Management selected. Below the tabs is a toolbar with Add, Edit, Remove, Change Password, and Show Permissions buttons. A table lists users with columns: Username, Realm, Enabled, Expire, Name, and Email. One row shows 'root' with 'pam' as the realm, 'Enabled' checked, 'Expire' set to 'never', 'Name' as 'root', and 'Email' as 'admin@cloud.local'. A modal dialog titled 'Add: User' is open, containing fields for User name (with validation error), First name, Realm (set to 'pam'), Last name, Password, Confirm password, Expire (date field), Enabled (checked), and Comment. An 'Add' button is at the bottom right of the dialog.

Viewing access control and permissions capabilities

Thoughts about the new tool

The new Proxmox Datacenter Manager looks to be Proxmox taking things in the right direction. With its explosive growth and interest in the platform, they are no doubt looking to provide the types of tools needed to take Proxmox as a platform to the next level of enterprise adoption. Having a centralized management tool is a great step for that.

Up until now, [VMware had vCenter Server](#), XCP-ng had Xen Orchestra, but Proxmox didn't really have a tool. Now the Proxmox Datacenter Manager (PDM) will provide that missing link for managing your Proxmox environment at scale.

pfSense Proxmox Install Process and Configuration

Many great open source solutions are available these days for many use cases, including security, networking, routing, etc. Two of those include pfSense and Proxmox server. Proxmox VE is an open-source solution that you can easily download for free and run a pfSense VM for routing, virtual network interfaces, firewall capabilities, etc. Let's deep dive into the process of pfSense [Proxmox install process and configuration and see what steps](#) are involved.

pfSense on Proxmox installation and configuration - Step-by-step

<https://youtube.com/watch?v=mwDv790YoZ0>



What is Proxmox VE?

Proxmox VE is a open-source virtualization solution allowing you to run virtual machines, including pfSense VM solutions. This is great as it allows you to run a pfSense virtual machine that can perform routing, firewalling, VPN, and all the great features that pfSense includes as part of the solution. In addition, you can run other virtual machines along with pfSense in Proxmox.

You can also run a [Proxmox cluster](#) for the highest availability requirements and for failover purposes.

Running pfSense on Proxmox VE

Running pfSense on Proxmox server, pfSense Proxmox, is a great way to have powerful features for no cost, running on commodity bare metal hardware. Proxmox provides many enterprise hypervisor features, including backups that can be enabled for newly created virtual machine boxes running in Proxmox server.

Run on bare metal or virtual machine

Proxmox hosts can run on a bare metal server or run as a virtual machine itself. If you would like to see how to run Proxmox Server as a nested VMware virtual machine, check out my post here: [Nested Proxmox VMware installation in ESXi – Virtualization Howto](#)

What is pfSense?

First of all, what is pfSense? The pfSense solution is a secure and widely used firewall distribution that is available as a virtual machine appliance or running on hardware platforms from Netgate.

Either way, you can get network interfaces either in hardware or virtual machine network interfaces, allowing you to route, firewall, and connect traffic to your network as you would any other enterprise firewall solution.

Netgate hardware



Netgate hardware firewall appliances

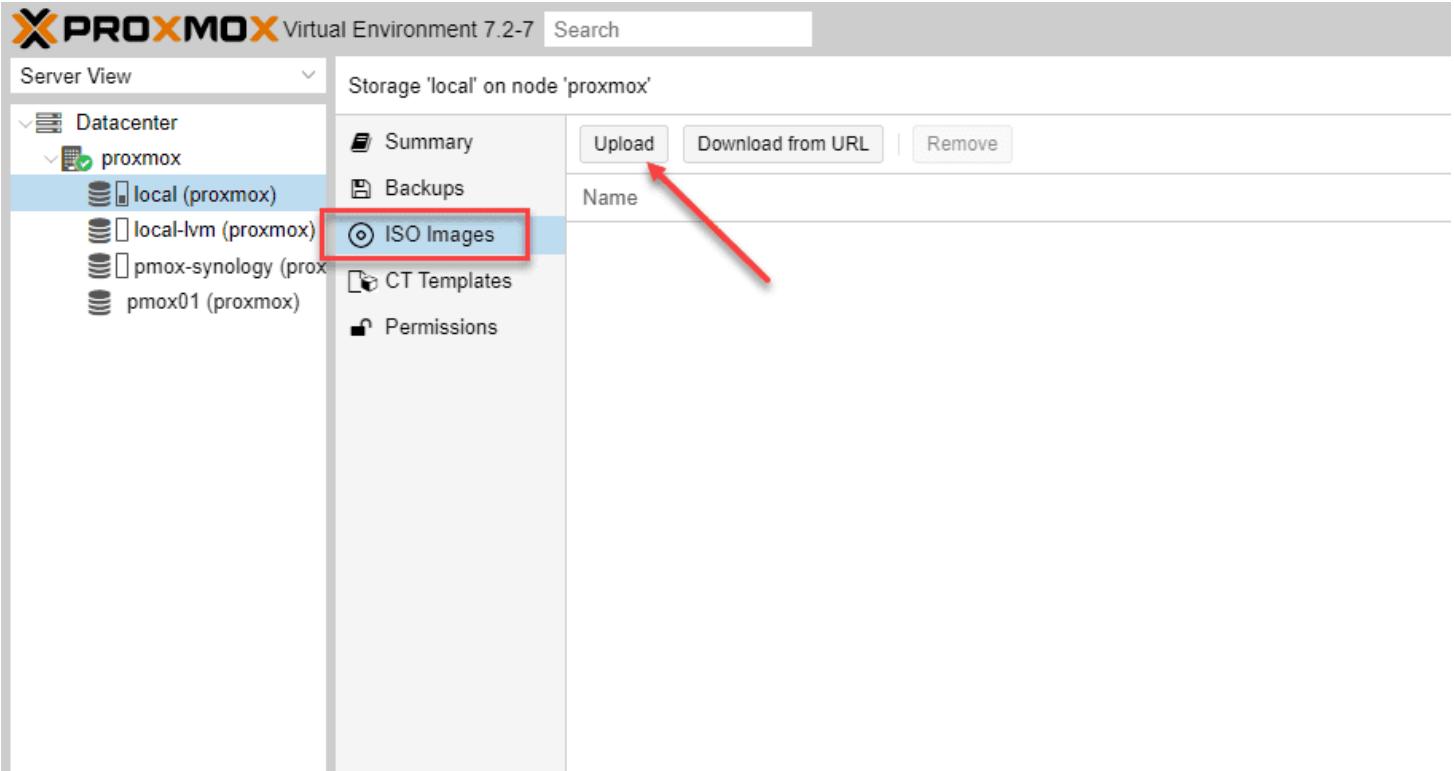
Download pfSense ISO image

You can download the pfSense ISO image here for the Community edition:

[Download pfSense Community Edition](#)

Upload ISO image to Proxmox server

Before we can run our pfSense VM installation on Proxmox VE, we need to get the [installation ISO image for pfSense VM uploaded to Proxmox](#). To do that, we log into Proxmox VE and browse to our local Proxmox storage, select ISO images and click the **Upload** button



Beginning process to upload pfSense ISO to Proxmox

Once you click the Upload button, you will have the ability to click the **Select File** button. Click the Select File button and browse to your downloaded pfSense ISO image. Then, click **Upload**.

Upload

File: C:\fakepath\pfSense-CE-2.6.0-F

File name: pfSense-CE-2.6.0-RELEASE-amd64.iso

File size: 731.91 MiB

MIME type: -

Hash algorithm: None

Checksum: none

Choose the pfSense ISO for upload to Proxmox VE

After you click Upload, you will see the upload progress. Then, the screen below should display, noting the upload of the ISO image was successful for pfSense.

Output Status

Stop

```
starting file import from: /var/tmp/pveupload-85148f7d63fbcd23bf7b1f71e7534c02
target node: proxmox
target file: /var/lib/vz/template/iso/pfSense-CE-2.6.0-RELEASE-amd64.iso
file size is: 767463424
command: cp -- /var/tmp/pveupload-85148f7d63fbcd23bf7b1f71e7534c02 /var/lib/vz/template/iso/pfSense-CE-2.6.0-RELEASE-amd64.iso
finished file import successfully
TASK OK
```

pfSense ISO image successfully uploaded to Proxmox VE server

Creating the pfSense VM in Proxmox VE

We first need to create the pfSense VM in [Proxmox VE that will be used to install](#) pfSense.

After you access Proxmox through port 8006, right-click your Proxmox VE server in the Proxmox web GUI and select Create VM.

The screenshot shows the Proxmox VE 7.2-7 interface. In the left sidebar under 'Datacenter', the node 'proxmox' is selected. A red arrow points to the 'Create VM' option in the context menu that appears when right-clicking on the node. The main panel displays the 'Node 'proxmox'' configuration with tabs for 'Summary', 'Notes', 'Shell', 'System', 'Network', 'Certificates', 'DNS', 'Hosts', 'Options', 'Time', 'Syslog', 'Updates', and 'Repositories'. On the right, a table lists network interfaces: ens160, ens192, ens32, vmbr0, and vmbr1. Below the table, a message says 'Create a new VM in Proxmox VE'.

Name ↑	Type	Active	Autostart
ens160	Network Device	Yes	Yes
ens192	Network Device	Yes	Yes
ens32	Network Device	Yes	No
vmbr0	Linux Bridge	Yes	Yes
vmbr1	Linux Bridge	Yes	Yes

Configuring the new pfSense VM

Note the following tabs and how they are [configured with the new pfSense VM](#).

General tab Settings

On the general tab, configure a name for the new pfSense VM.

Create: Virtual Machine



General

OS

System

Disks

CPU

Memory

Network

Confirm

Node:

proxmox



Resource Pool:



VM ID:

100



Name:

pfSense

Help

Advanced

Back

Next

Configure a name for the new pfSense VM

OS settings

On the OS tab, here is where we select the ISO image that we uploaded earlier.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

Use CD/DVD disc image file (iso)

Guest OS:

Storage: local

Type:

Linux

ISO image: pfSense-CE-2.6.0-RELEASE-arm

Version:

5.x - 2.6 Kernel

Use physical CD/DVD drive

Name

For...

Size

Do not use any

pfSense-CE-2.6.0-RELEASE-amd64.iso

iso

767.46 MB

Advanced

Back

Next

Select the pfSense ISO image

System tab

On the system tab, we can leave the default settings.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

Graphic card: Default SCSI Controller: VirtIO SCSI

Machine: Default (i440fx) Qemu Agent:

Firmware

BIOS: Default (SeaBIOS) Add TPM:

Help

Advanced

Back

Next

Accept the defaults on System

Disks

On the disk screen, you select where you want to install pfSense, the disk size, bus device information, etc.

Create: Virtual Machine



General OS System **Disks** CPU Memory Network Confirm

scsi0 ✖	Disk Bandwidth
Bus/Device: SCSI 0 Cache: Default (No cache)	
SCSI Controller: VirtIO SCSI Discard: <input type="checkbox"/>	
Storage: local-lvm	
Disk size (GiB): 32	
Format: Raw disk image (raw)	

+ Add

? Help Advanced Back Next

Select the storage location for the pfSense VM in Proxmox

CPU tab

On the CPU tab, you can configure the number of CPU sockets and cores.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

Sockets:	<input type="text" value="1"/>	Type:	<input type="text" value="Default (kvm64)"/>
Cores:	<input type="text" value="1"/>	Total cores:	<input type="text" value="1"/>

Help

Advanced

Back

Next

Configure CPU settings

Memory tab

On the memory tab, you configure how much memory you want to allocate to the pfSense VM.

Create: Virtual Machine



General OS System Disks CPU **Memory** Network Confirm

Memory (MiB):

2048



Help

Advanced

Back

Next

Configure memory settings for pfSense

Networking Tab

On the network tab, you configure the network interfaces you want to use for your pfSense VM running on your Proxmox host. There are differences to think about depending on whether you are running pfSense on physical hardware with physical interface ports or a virtual machine running pfSense.

Here, on the creation screen, we can just accept the defaults and then we will change a couple of settings once we have the VM created. Note on the screen the settings you can configure, including bridge ports, VLAN tag, firewall, model, MAC address, etc.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

No network device

Bridge: Model:
VLAN Tag: MAC address:
Firewall:

Help

Advanced

Back

Next

Configure the network settings for the pfSense VM

Confirm tab

On the confirm tab, we can confirm the settings used to create VM for pfSense.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

Key ↑	Value
cores	1
ide2	local:iso/pfSense-CE-2.6.0-RELEASE-amd64.iso,media=cdrom
memory	2048
name	pfsense
net0	virtio,bridge=vmbr0,firewall=1
nodename	proxmox
numa	0
ostype	I26
scsi0	pmox-synology:32
scsihw	virtio-scsi-pci
sockets	1
vmid	100

Start after created

Advanced

Back

Finish

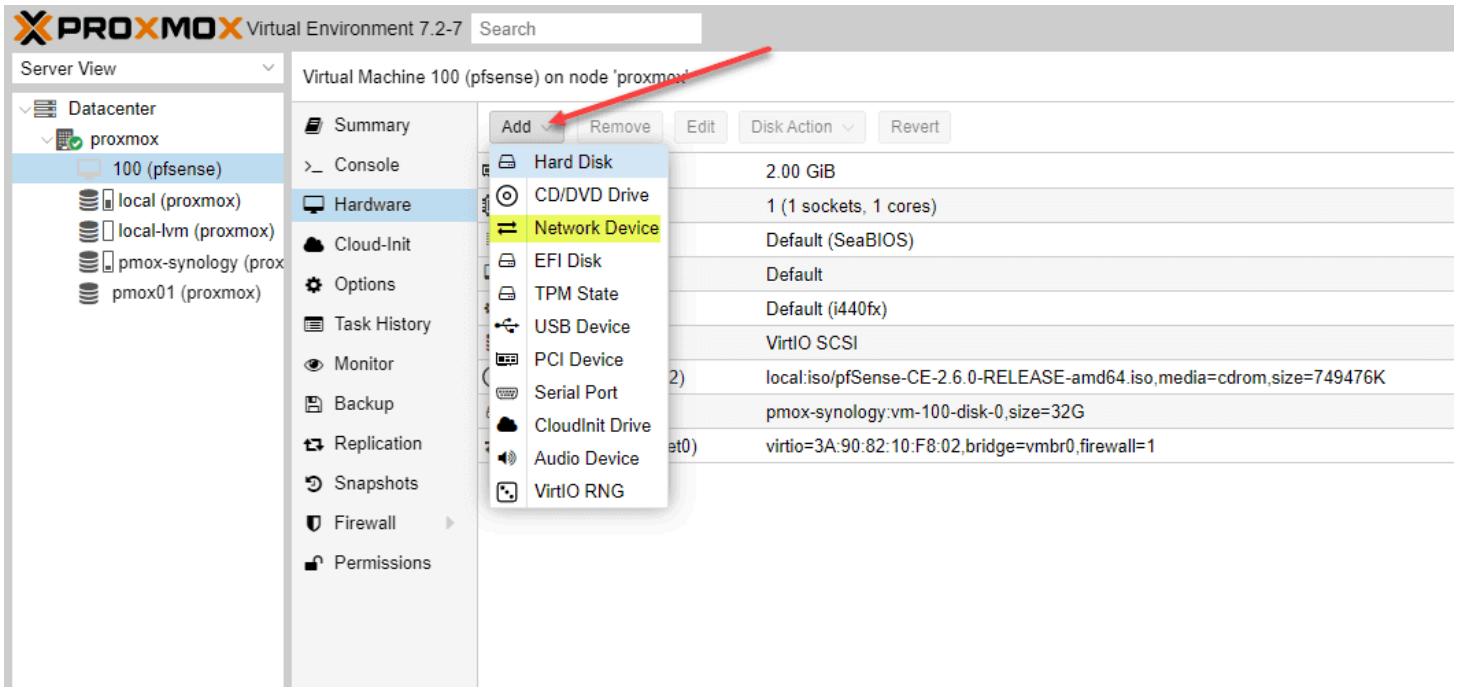
Confirm the configuration settings for the new pfSense VM

After you click create VM of the pfSense VM, this essentially creates the pfSense virtual machine so we can install pfSense as a guest OS on in the Proxmox box VM.

Changing a few settings on the pfSense VM

If you noticed on the network device screen above, it only configured one network device. However, for the pfSense VM to route traffic as expected, we need both a LAN port, or LAN interface, and a WAN port, or WAN interface.

The WAN interface will house the WAN IP address that will provide connectivity from the outside inward for accessing internal resources and provide Internet connectivity. These WAN and LAN interface connections will allow successfully routing traffic as expected and benefiting from the pfSense firewall.



Add a new network adapter

On the new interface, select the bridge ports, VLAN tag, and other settings for the second network adapter. By default, it will add virtIO interfaces when you add a new adapter. You may need to play around with this when adding. I had to go back and change my installation to Intel Pro 1000 adapters for it to work correctly in my nested lab.

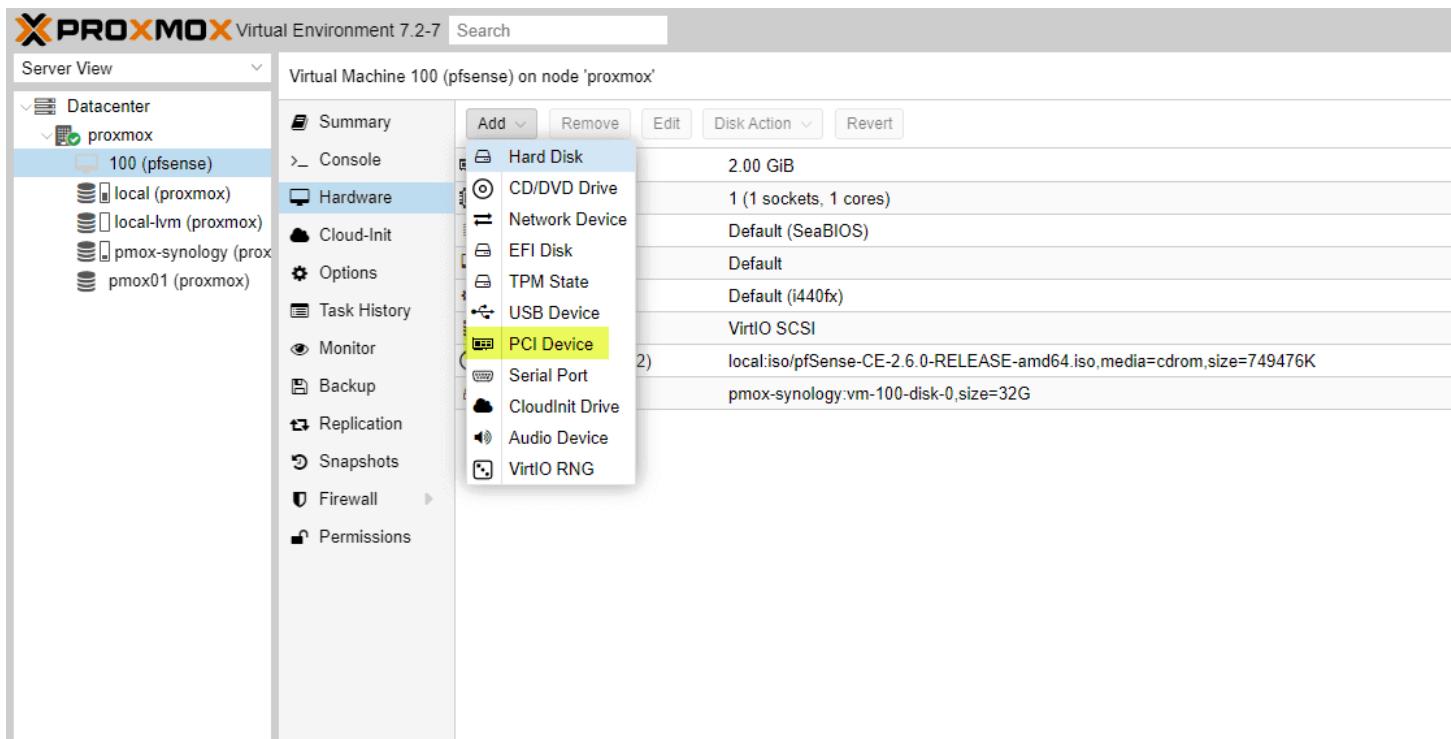
I also added an additional network bridge where you can choose a new Linux bridge configuration.

Add a new network adapter after creating the pfSense VM

After adding an additional network device, we now have two network devices configured with the pfSense VM.

Viewing the network adapters after adding to the pfSense VM

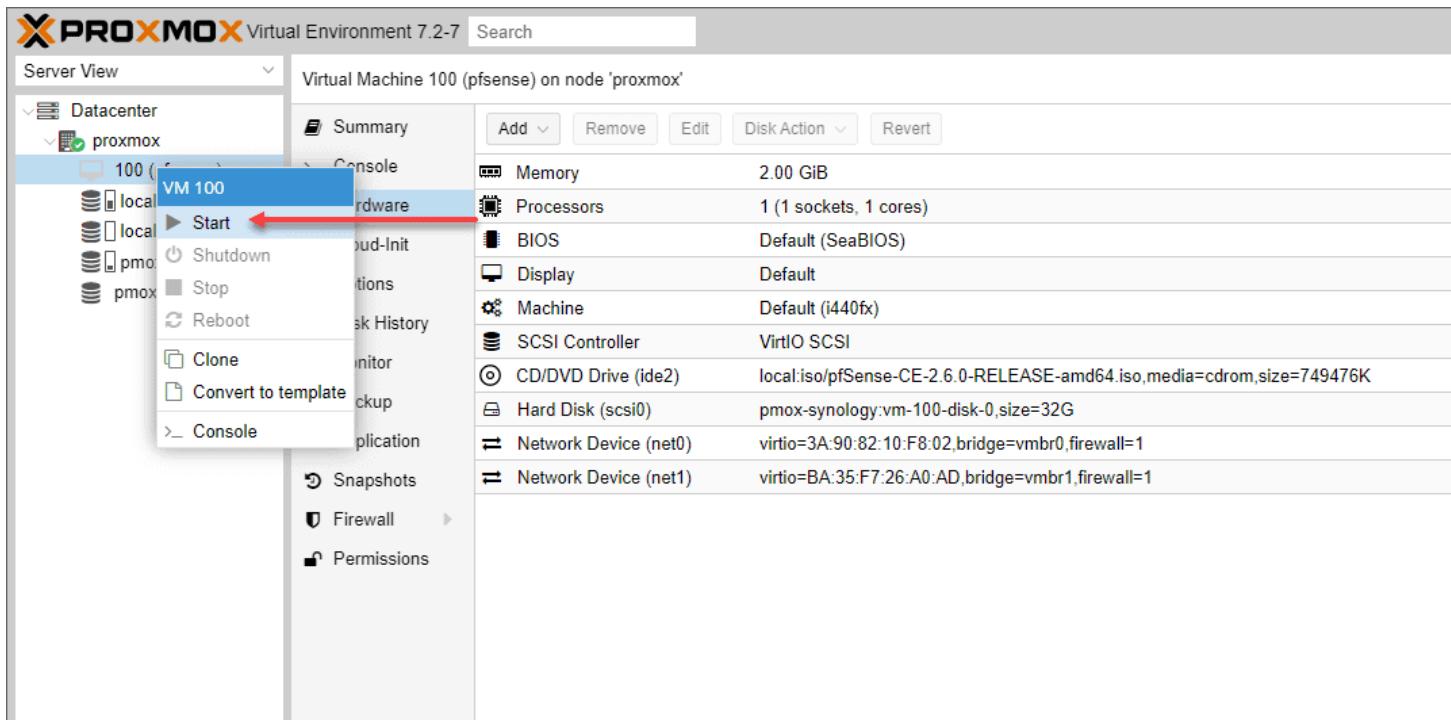
As a note, depending on what type of hardware you are running on top of for your Proxmox host, some may need to instead not add a network adapter but instead add a PCI device that is passed through to physical NICs.



Adding a new PCI device in Proxmox VE

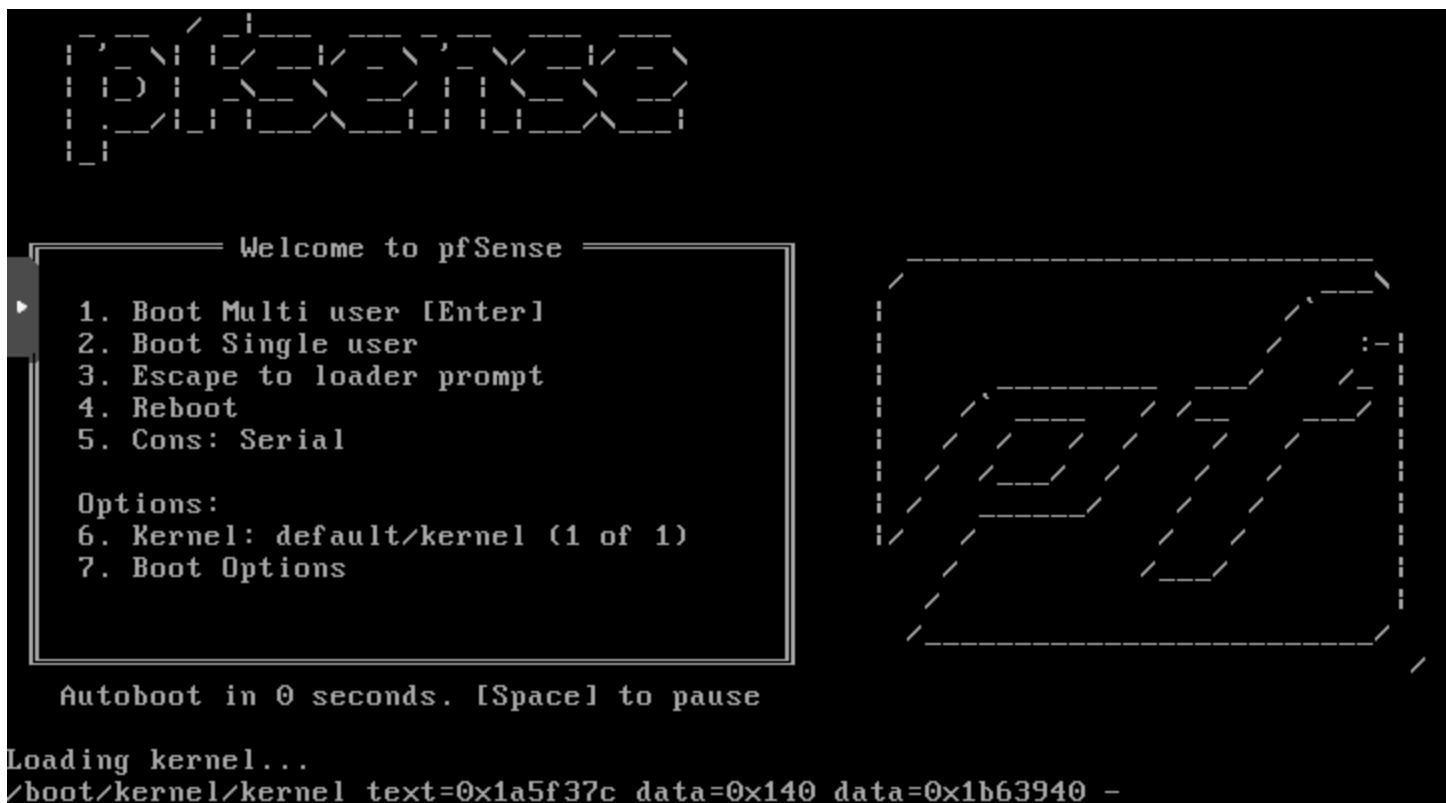
Install pfSense VM

Now we can actually install pfSense and configure the virtual machine appliance. Right-click the pfSense VM shown on your Proxmox host and select start.



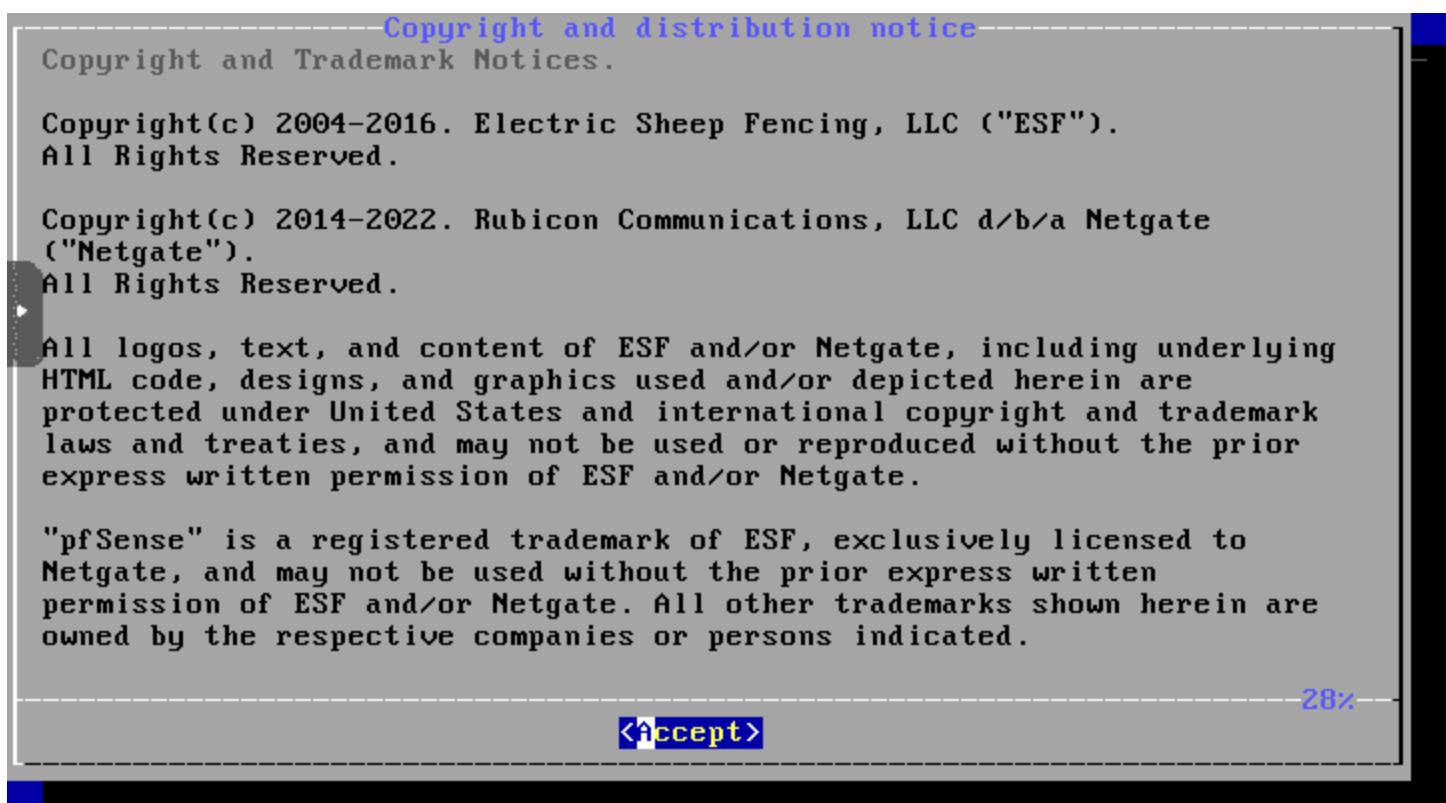
Power on the new pfSense virtual machine

After powering on and pfSense running as a VM, we can begin the process to run pfSense as an installed pfSense version.



Boot screen of the pfSense VM running in Proxmox VE

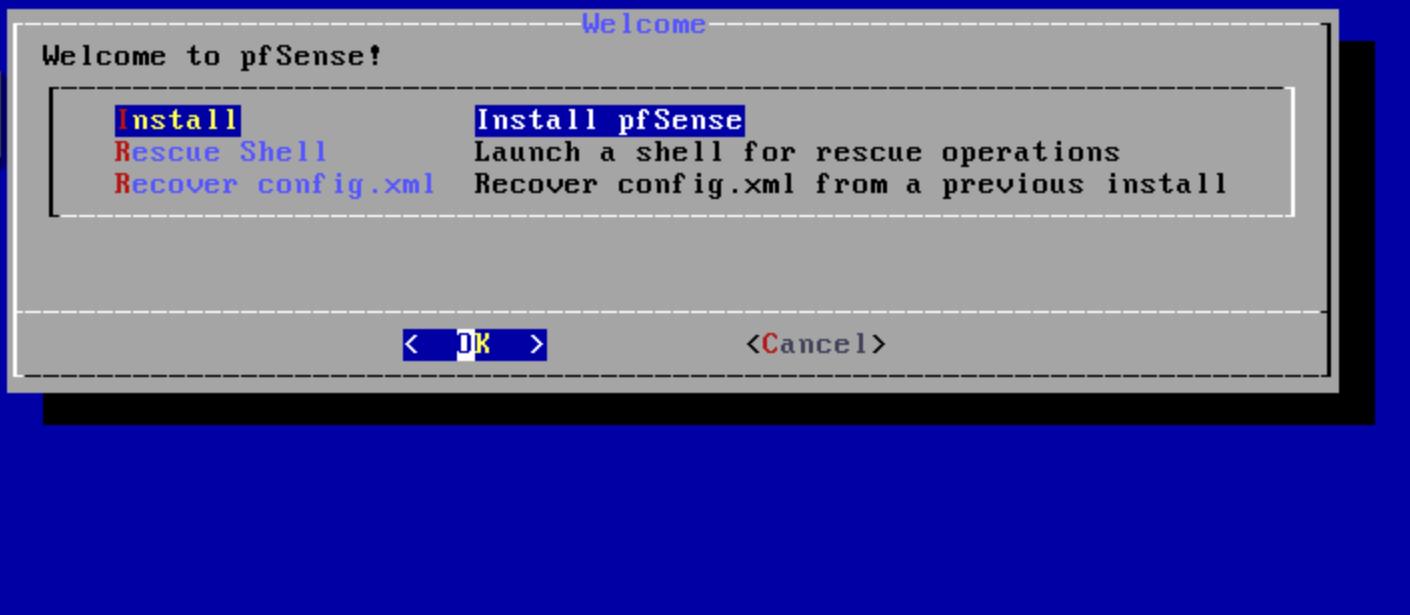
This begins the “text” install pfSense VM process. Accept the EULA displayed.



Accept the EULA agreement

Choose to install pfSense on the next screen.

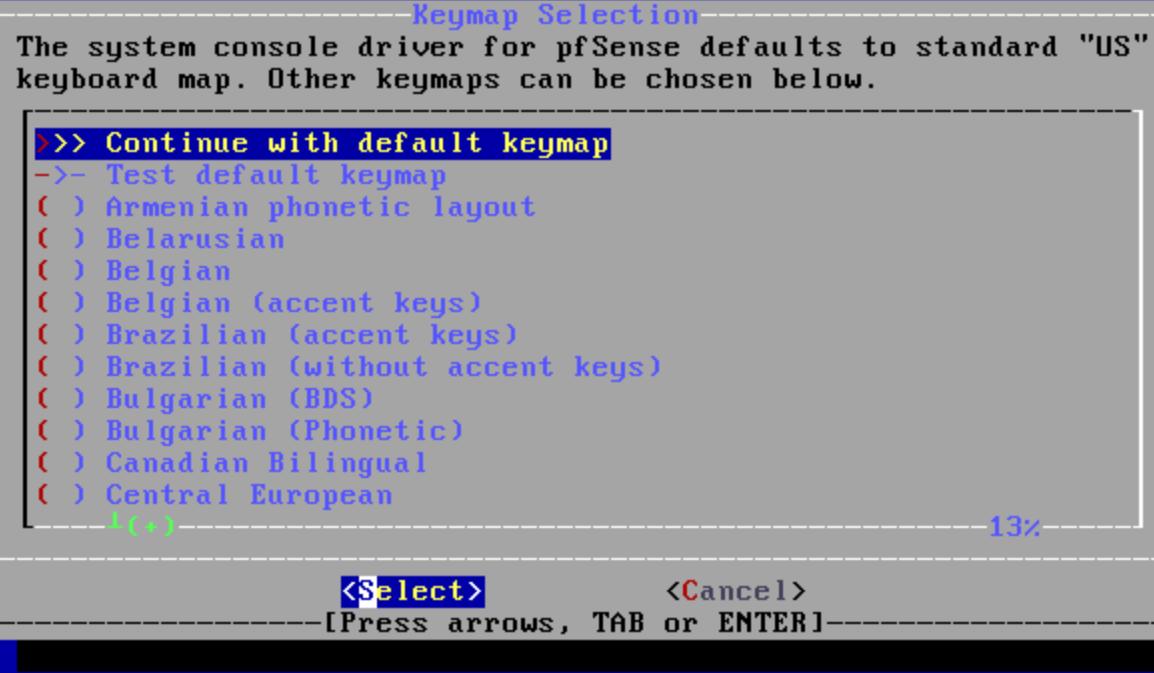
pfSense Installer



Selecting the *Install* option on the text installer screen

Continue with the default keymap for the keyboard layout.

pfSense Installer



Configuring the default keymap

Choose to configure the partitioning unless you need a custom layout. Automatically. Here I am choosing ZFS configuration.

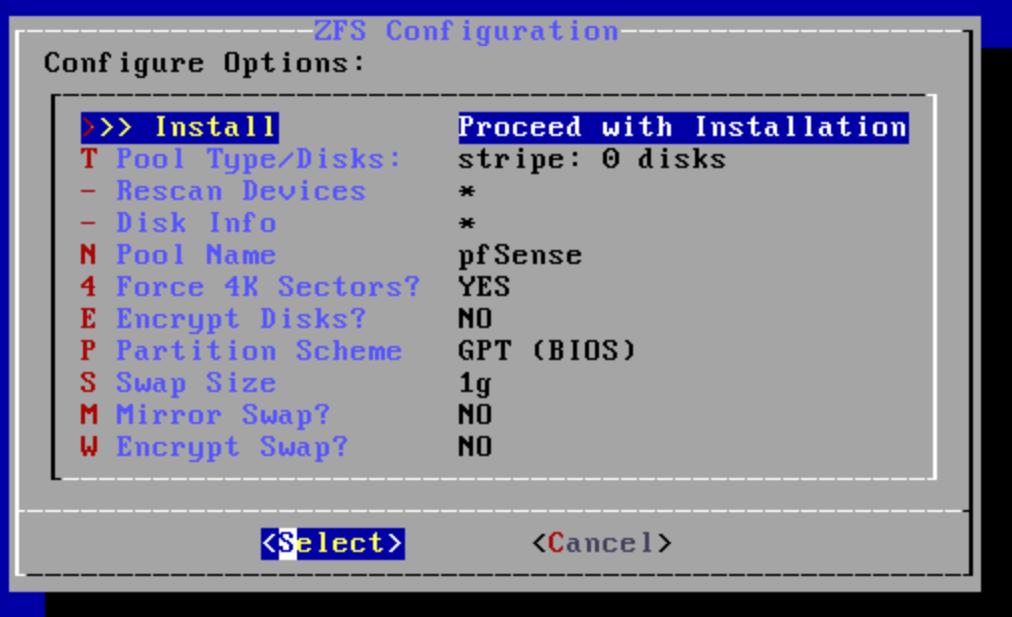
pfSense Installer



Selecting how you would like to partition your disk in Proxmox VE

Proceed with the install pfSense process.

pfSense Installer



Create ZFS boot pool with displayed options

Confirming to proceed with the installation

Choose the virtual device type. Here I am selecting the Stripe no redundancy.



[1+ Disks] Striping provides maximum storage but no redundancy

Select the virtual disk type

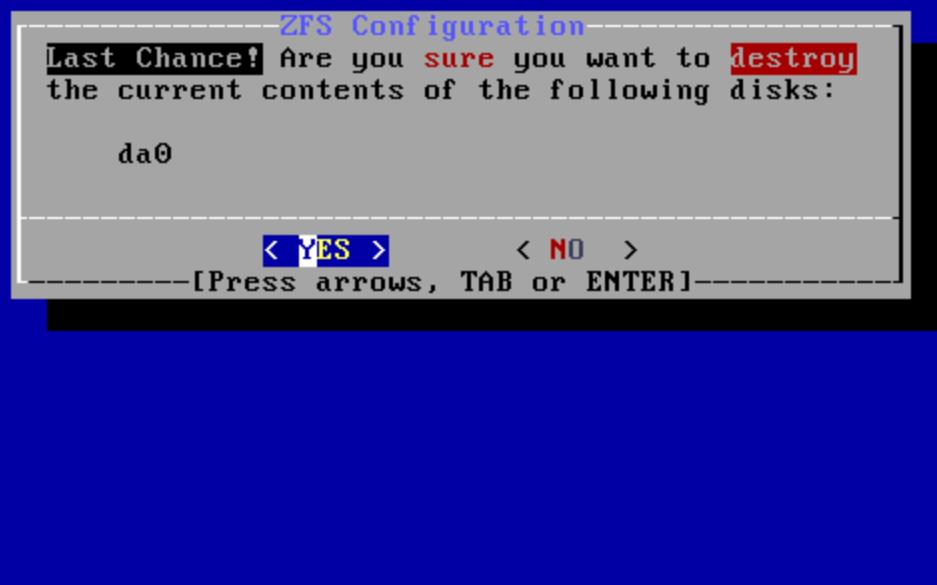
On the ZFS configuration screen, click OK.



Confirm your ZFS configuration

Click Yes on the ZFS configuration screen.

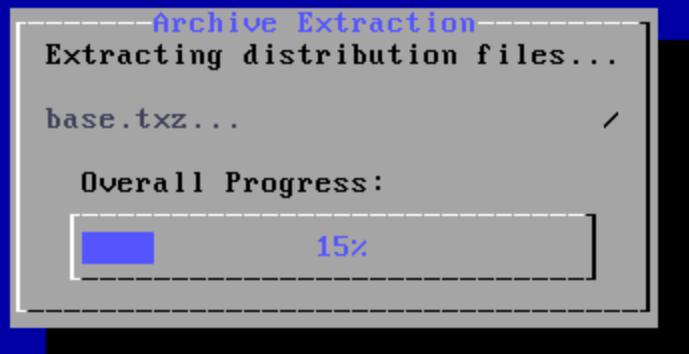
pfSense Installer



Confirming you want to format the disk and destroy data

The install pfSense process begins.

pfSense Installer

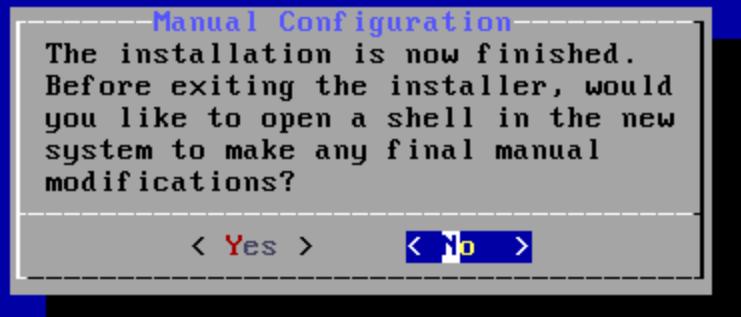


4094 files read @ 454.0 files/sec.

pfSense installation on Proxmox begins

You will be asked if you have any manual configuration you want to perform. If not, select No.

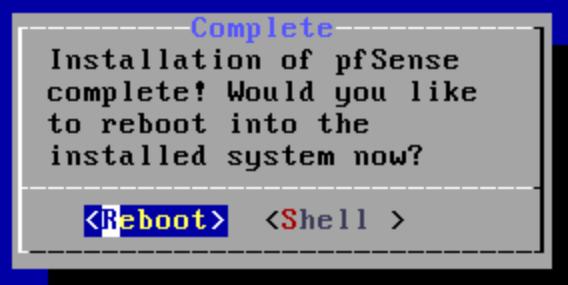
pfSense Installer



Installation is finished and choosing no custom modifications

The installation is complete. Reboot your pfSense VM.

pfSense Installer



Choosing to reboot after the installation

After the pfSense VM boots for the first time, you should see your WAN and LAN interfaces come up and show IP addresses for the WAN and LAN ports. As you can see, these are not on the same network or same subnet.

Most configurations will see the WAN IP address [configured from the ISP via DHCP server](#). You will want to have a static IP address configured on the LAN interface since this will be used as the gateway address for clients connected to the LAN port of the pfSense VM.

The pfSense LAN address is configurable and you will want to configure the address to match your clients. The LAN port also doubles as the management port for pfSense VM by default. You can't manage pfSense from the WAN port by default, only the LAN port. This can be changed later, but is something to note as you run the pfSense virtual machine on your Proxmox box.

The pfSense firewall will also be the default gateway for the clients on the network. The pfSense WAN is the address used for incoming traffic that will be NAT'ed inward to internal IP addresses on the network. For management, specifically note the LAN ip address.

Below, you will note I have private IPs on both the WAN and LAN port. This is because I have this configured in a lab environment. In production, you will have a public IP address configured on the WAN port for true edge firewall capabilities.

```
Starting syslog...done.
Starting CRON... done.
pfSense 2.6.0-RELEASE amd64 Mon Jan 31 19:57:53 UTC 2022
Bootup complete

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

KVM Guest - Netgate Device ID: e025443f9e90aa86fb59
* Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***
WAN (wan)      -> em0          -> v4/DHCP4: 10.1.149.165/24
LAN (lan)      -> em1          -> v4: 192.168.1.1/24

0) Logout (SSH only)          9) pfTop
1) Assign Interfaces           10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable Secure Shell (sshd)
6) Halt system                 15) Restore recent configuration
7) Ping host                   16) Restart PHP-FPM
8) Shell

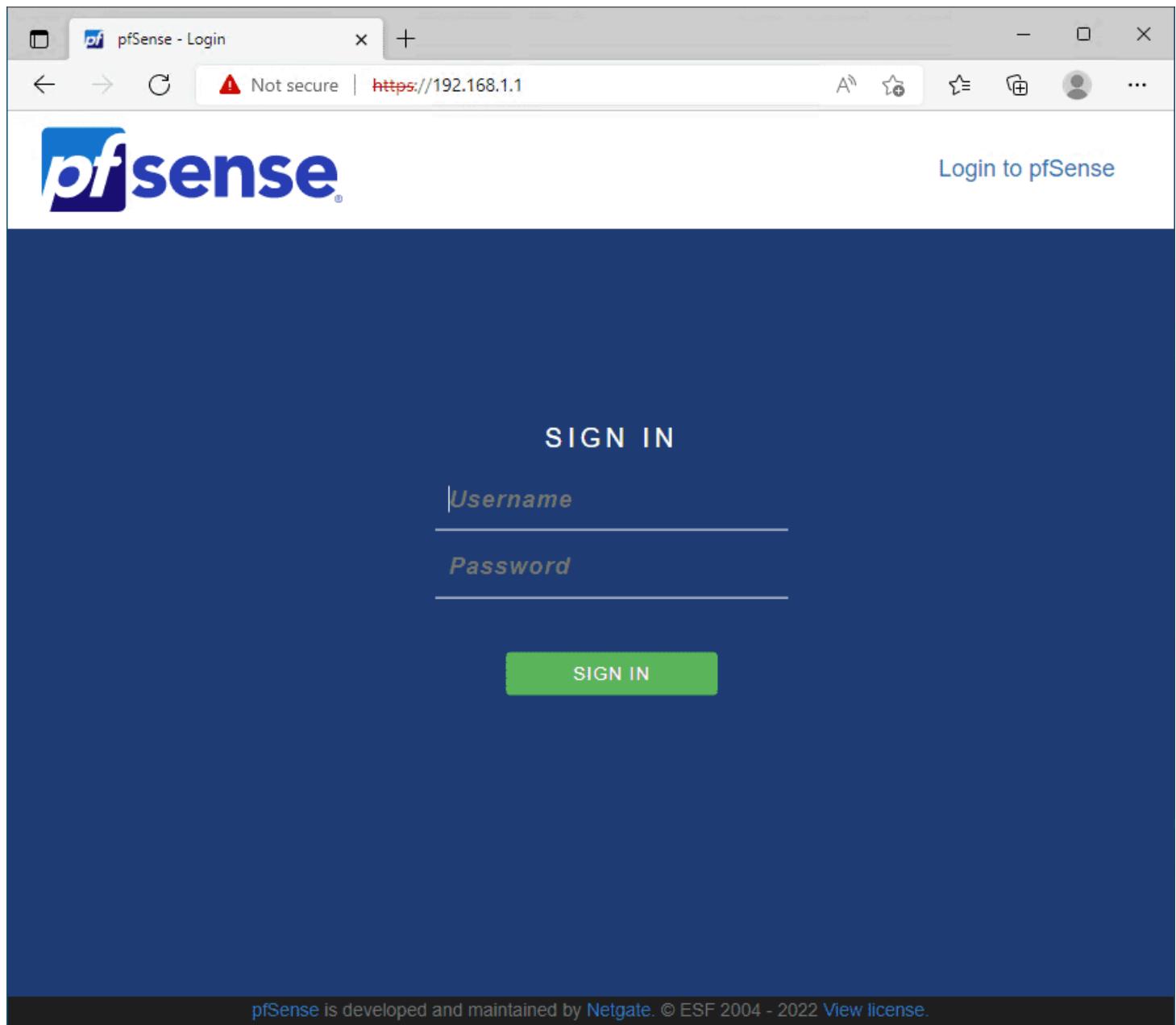
Enter an option: [
```

Viewing the interface DHCP address and internal LAN

Configure pfSense VM on Proxmox

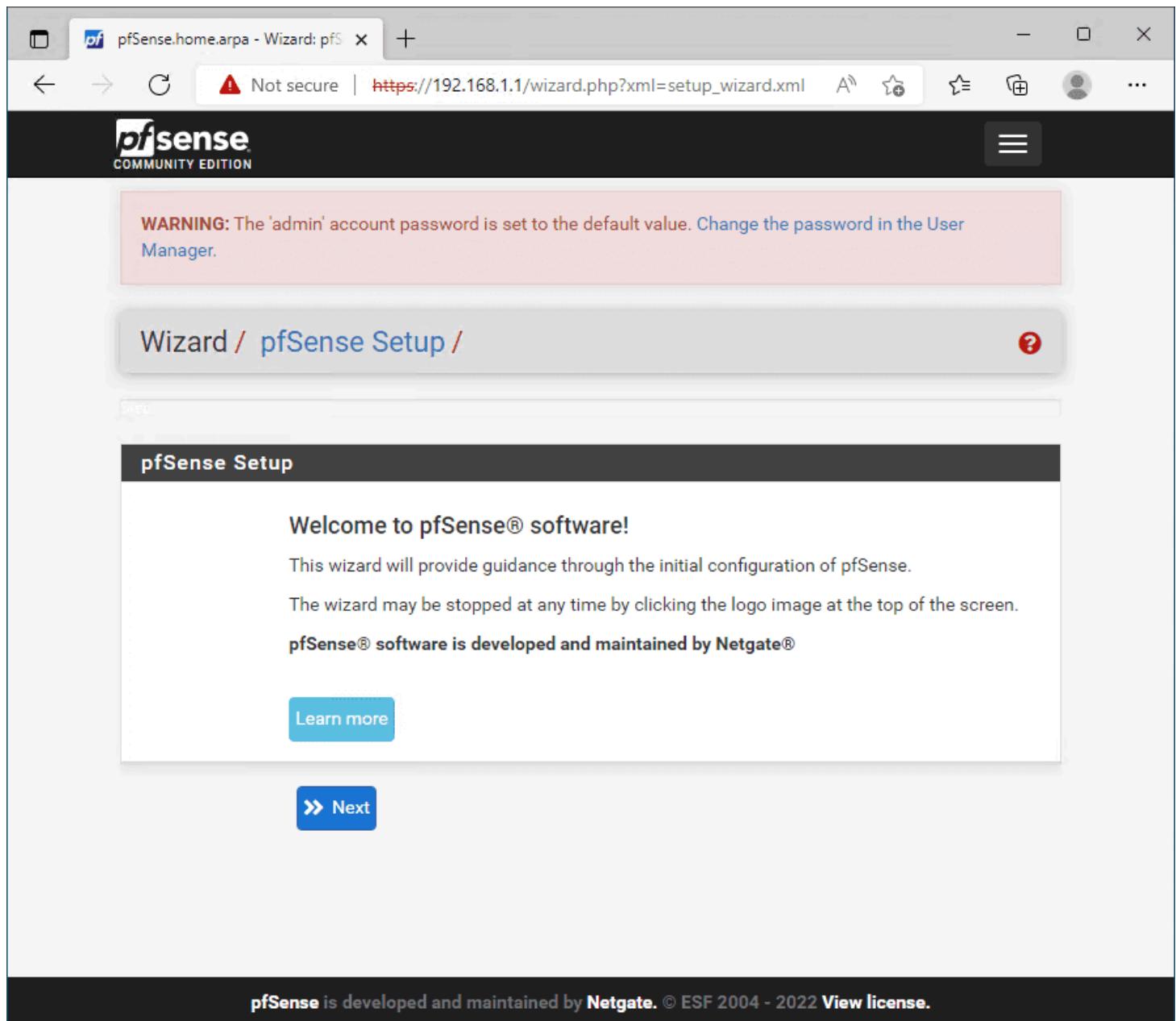
Now, we need to browse out to the pfSense web GUI found on the IP address of the LAN port after installing in Proxmox. The default password will be needed as you log into the pfSense LAN and is:

admin/pfsense



Logging into pfSense VM for the first time

After logging in with the default admin password, the configuration wizard will begin to run pfSense, including the pfSense firewall capabilities.



Beginning the pfSense web UI setup wizard

Click next past the Netgate support message.

The screenshot shows a web browser window for the pfSense setup wizard. The address bar indicates the URL is https://192.168.1.1/wizard.php?xml=setup_wizard.xml. A red warning message in a pink box at the top states: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." Below this, a blue header bar says "Wizard / pfSense Setup / Netgate® Global Support is available 24/7". A red button labeled "Step 1 of 9" is visible. The main content area has a dark header "Netgate® Global Support is available 24/7". It contains text about Netgate's 24/7 support team and subscription plans, followed by a bulleted list of support features. A blue "Learn more" button is present, and a blue "» Next" button is at the bottom. The footer of the page includes the pfSense logo, a note about Netgate support, and a link to the license.

WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.

Wizard / pfSense Setup / Netgate® Global Support is available 24/7

Step 1 of 9

Netgate® Global Support is available 24/7

Our 24/7 worldwide team of support engineers are the most qualified to diagnose your issue and resolve it quickly, from branch office to enterprise – on premises to cloud.

We offer several support subscription plans tailored to fit different environment sizes and requirements. Many companies around the world choose Netgate support because:

- Support is available 24 hours a day, seven days a week, including holidays.
- Support engineers are located around the world, ensuring that no support call is missed.
- Our support engineers hold many prestigious network engineer certificates and have years of hands-on experience with networking.

[Learn more](#)

[» Next](#)

pfSense is developed and maintained by [Netgate](#). © ESF 2004 - 2022. [View license](#).

Note the message on Netgate support

Set the pfSense hostname and domain name.

The screenshot shows a web browser window for the pfSense setup wizard. The address bar indicates the URL is https://192.168.1.1/wizard.php?xml=setup_wizard.xml. A warning message at the top states: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." The main title is "Wizard / pfSense Setup / General Information". Below it, a red bar indicates "Step 2 of 9". The "General Information" section contains fields for "Hostname" (set to "pfSense") and "Domain" (set to "cloud.local"). A note explains the DNS resolver behavior. There are also fields for "Primary DNS Server" and "Secondary DNS Server".

WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager.

Wizard / pfSense Setup / General Information

Step 2 of 9

General Information

On this screen the general pfSense parameters will be set.

Hostname pfSense
EXAMPLE: myserver

Domain cloud.local
EXAMPLE: mydomain.com

The default behavior of the DNS Resolver will ignore manually configured DNS servers for client queries and query root DNS servers directly. To use the manually configured DNS servers below for client queries, visit Services > DNS Resolver and enable DNS Query Forwarding after completing the wizard.

Primary DNS Server

Secondary DNS Server

Configure the pfSense hostname

Configure the NTP time server configuration.

The screenshot shows the pfSense setup wizard on a web browser. The title bar reads "pfSense.home.arpa - Wizard: pfSense". A warning message in a pink box says: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." The main heading is "Wizard / pfSense Setup / Time Server Information". A red bar at the top indicates "Step 3 of 9". The section title is "Time Server Information". It asks for time, date, and time zone information. The "Time server hostname" field contains "2.pfsense.pool.ntp.org" with the placeholder "Enter the hostname (FQDN) of the time server.". The "Timezone" dropdown is set to "Etc/UTC". A blue "» Next" button is at the bottom.

pfSense is developed and maintained by **Netgate**. © ESF 2004 - 2022 [View license](#).

Configure NTP settings in pfSense

Configure the WAN interface. Even though we have already configured this, the pfSense wizard gives you another opportunity to configure the WAN port.

The screenshot shows a web browser window for pfSense setup. The title bar reads "pfSense.home.arpa - Wizard: pfS". A warning message in a red box says: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." The main page title is "Wizard / pfSense Setup / Configure WAN Interface". A progress bar at the top indicates "Step 4 of 9". The main section is titled "Configure WAN Interface" and contains the following configuration fields:

- Selected Type:** DHCP
- General configuration**
 - MAC Address:** (Input field) This field can be used to modify ("spoof") the MAC address of the WAN interface (may be required with some cable connections). Enter a MAC address in the following format: xx:xx:xx:xx:xx:xx or leave blank.
 - MTU:** (Input field) Set the MTU of the WAN interface. If this field is left blank, an MTU of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed.
 - MSS:** (Input field) If a value is entered in this field, then MSS clamping for TCP connections to the value entered above minus 40 (TCP/IP header size) will be in effect. If this field is left blank, an MSS of 1492 bytes for PPPoE and 1500 bytes for all other connection types will be assumed.

Configure WAN interface in pfSense

Same with the LAN port. You can reconfigure if needed here.

The screenshot shows a web browser window for pfSense setup. The address bar indicates the URL is https://192.168.1.1/wizard.php?xml=setup_wizard.xml. A warning message in a pink box states: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." The main title is "Wizard / pfSense Setup / Configure LAN Interface". A progress bar at the top shows "Step 5 of 9". The section title "Configure LAN Interface" is displayed. Below it, a note says: "On this screen the Local Area Network information will be configured." There are two input fields: "LAN IP Address" containing "192.168.1.1" and "Subnet Mask" containing "24". A note next to the IP address field says: "Type dhcp if this interface uses DHCP to obtain its IP address." At the bottom is a blue "» Next" button. The footer of the page includes the pfSense logo and copyright information: "pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 [View license](#)".

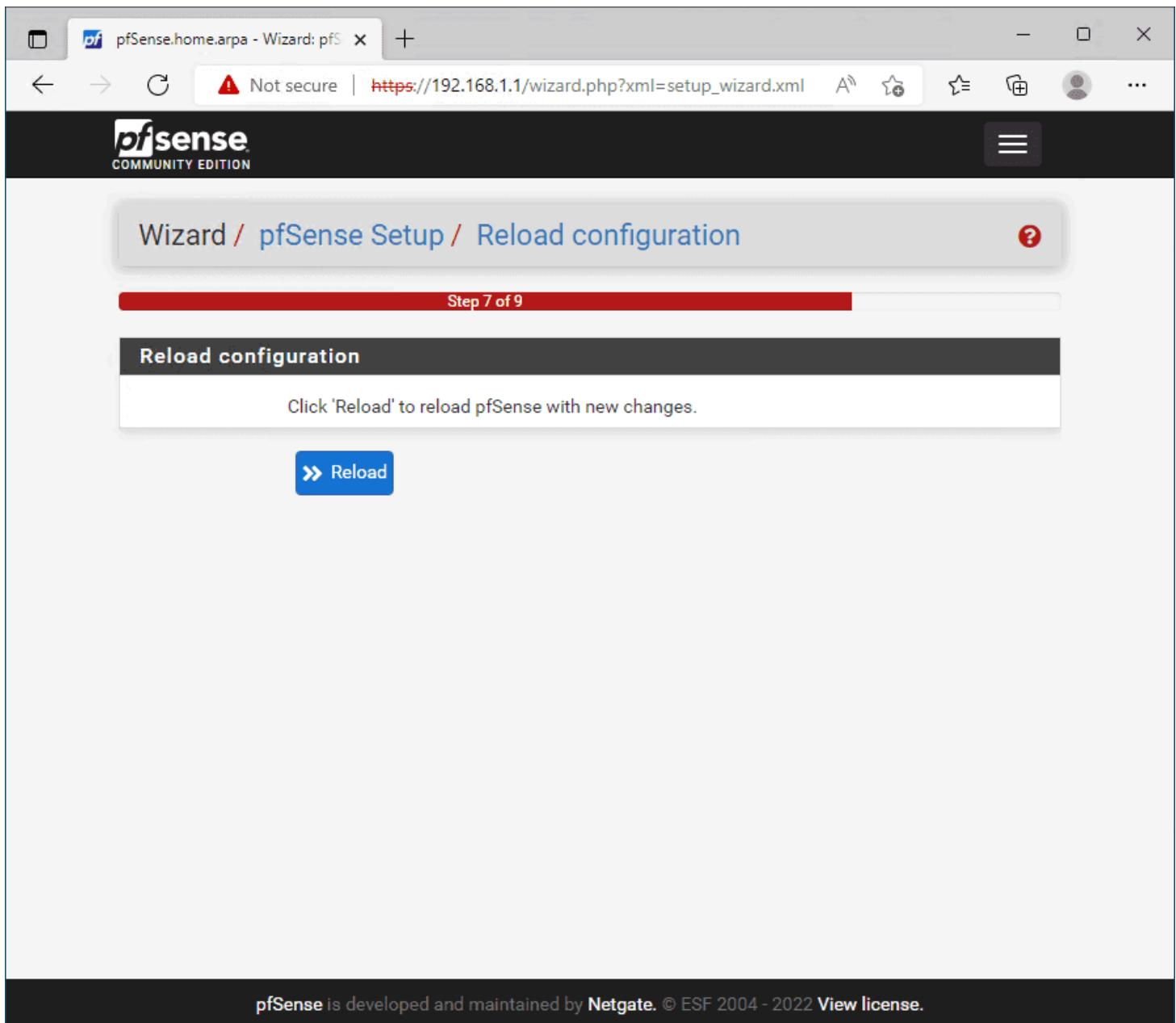
Configure the LAN interface

Change the admin password on the next screen.

The screenshot shows a web browser window for the pfSense setup wizard. The address bar indicates the URL is https://192.168.1.1/wizard.php?xml=setup_wizard.xml. A warning message in a pink box states: "WARNING: The 'admin' account password is set to the default value. Change the password in the User Manager." The main title is "Wizard / pfSense Setup / Set Admin WebGUI Password". A progress bar at the top shows "Step 6 of 9". The section title is "Set Admin WebGUI Password". It explains that the admin password will be set for access to the WebGUI and SSH services. Two password fields are present: "Admin Password" and "Admin Password AGAIN". Both fields contain placeholder dots. A blue "Next" button is at the bottom. At the bottom of the page, a footer notes: "pfSense is developed and maintained by Netgate. © ESF 2004 - 2022 [View license](#)".

Change the default admin password

Ready to reload pfSense to finalize the configuration.



Reload pfSense with the new configuration

At this point after the reload, the install pfSense process is now complete.

The screenshot shows a web browser window for pfSense.cloud.local - Wizard: pfs. The address bar indicates the URL is https://192.168.1.1/wizard.php?xml=setup_wizard.xml... with a 'Not secure' warning. The page title is "Wizard / pfSense Setup / Wizard completed." A green progress bar at the top right shows "Step 9 of 9". Below it, a dark bar says "Wizard completed.". The main content area has a heading "Congratulations! pfSense is now configured." followed by a message about checking for software updates. It includes a green button labeled "Check for updates". Below this, there's a section titled "User survey" with a message encouraging users to answer a short survey. A blue link "Anonymous User Survey" is provided. At the bottom, there's a section titled "Useful resources." with a bulleted list: "Learn more about Netgate's product line, services, and pfSense software from our website", "To learn about Netgate appliances and other offers, visit our store", and a partially visible link "Become part of the pfSense community. Visit our forums".

Wizard completes after the reload of pfSense

Congratulations, the install pfSense process is now complete!

Wrapping Up

The pfSense Proxmox installation procedure is straightforward and consists of creating a new Proxmox virtual machine with the correct network adapter settings. Then you power on the VM, run through the initial text configuration setup to install pfSense and establish basic networking connectivity. Afterward, using the pfSense web GUI, you finalize the pfSense installation on [Proxmox using the configuration](#) wizard. Proxmox makes for a great platform to [install](#) pfSense as Proxmox provides many of the settings and configuration capabilities needed to customize your installation of pfSense Proxmox.

Nested ESXi install in Proxmox: Step-by-Step

If you have a Proxmox VE server in your home lab or production environment and want to play around with VMware ESXi, you can easily do that with Proxmox nested virtualization. Let's look at the steps required for a nested ESXi server install in Proxmox.

Table of contents

- [Nested Virtualization in Proxmox](#)
- [Preparing your Proxmox VE host to enable nested virtualization for ESXi](#)
- [Creating the ESXi VM in Proxmox](#)
- [Step-by-Step Installation of Nested ESXi](#)
- [Managing Virtual Machines in a Nested Setup](#)
 - [Using advanced features in nested VMs](#)
- [Troubleshooting Common Issues in Nested Environments](#)

Nested Virtualization in Proxmox

Nested virtualization in Proxmox VE is easy to set up and has real benefits in learning and setting up rather complex architectures without the physical hardware that would otherwise be needed to set them up physically.

Now, you can use something like VMware Workstation to easily nest ESXi. However, if you already have a dedicated Proxmox host, it is a better platform for a dedicated lab experience. There is always running it on VMware ESXi if you have a physical [VMware host](#).

Proxmox nested virtualization allows exposing the CPU's hardware virtualization characteristics to a nested hypervisor. This process to expose hardware assisted virtualization to the [guest ESXi VM is required so the nested](#) hypervisor can run virtual machines.

Preparing your Proxmox VE host to enable nested virtualization for ESXi

If you don't know how to configure Proxmox Nested Virtualization or enabling hardware assisted virtualization, you can see my recent guide to do that here: [How to Enable Proxmox Nested Virtualization](#).

An overview of the few [steps exist to enable nested virtualization for Proxmox](#) and run a nested VM hypervisor are as follows:

- Make sure your CPU supports hardware-assisted virtualization
- Enable hardware-assisted virtualization if it isn't enabled already
- Enable nested virtualization on the nested ESXi installation VM

Creating the ESXi VM in Proxmox

VMware hypervisors are extremely popular in the enterprise. Let's look at the process to create the VMware [ESXi VM in Proxmox](#). This is a normal creation process for the most part. I will show you guys one option I chose that didn't work, surprisingly when creating the VM running ESXi.

Create: Virtual Machine



General

OS

System

Disks

CPU

Memory

Network

Confirm

Node:

pve01

Resource Pool:

VM ID:

104

Name:

esxionpve



Start at boot:



Start/Shutdown
order:

any

Startup delay:

default

Shutdown timeout:

default

Tags

No Tags



Help

Advanced

Back

Next

Beginning the create virtual machine wizard

Upload your VMware ESXi 8.0 U2 or other ESXi ISO to your Proxmox server and select this in the wizard. On the type, choose **Other** for the guest operating system.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

Use CD/DVD disc image file (iso)

Storage: local

ISO image: VMware-VMvisor-Installer-8.0U2

Use physical CD/DVD Drive

Do not use any media

Guest OS:

Type:

Other

Version:

-



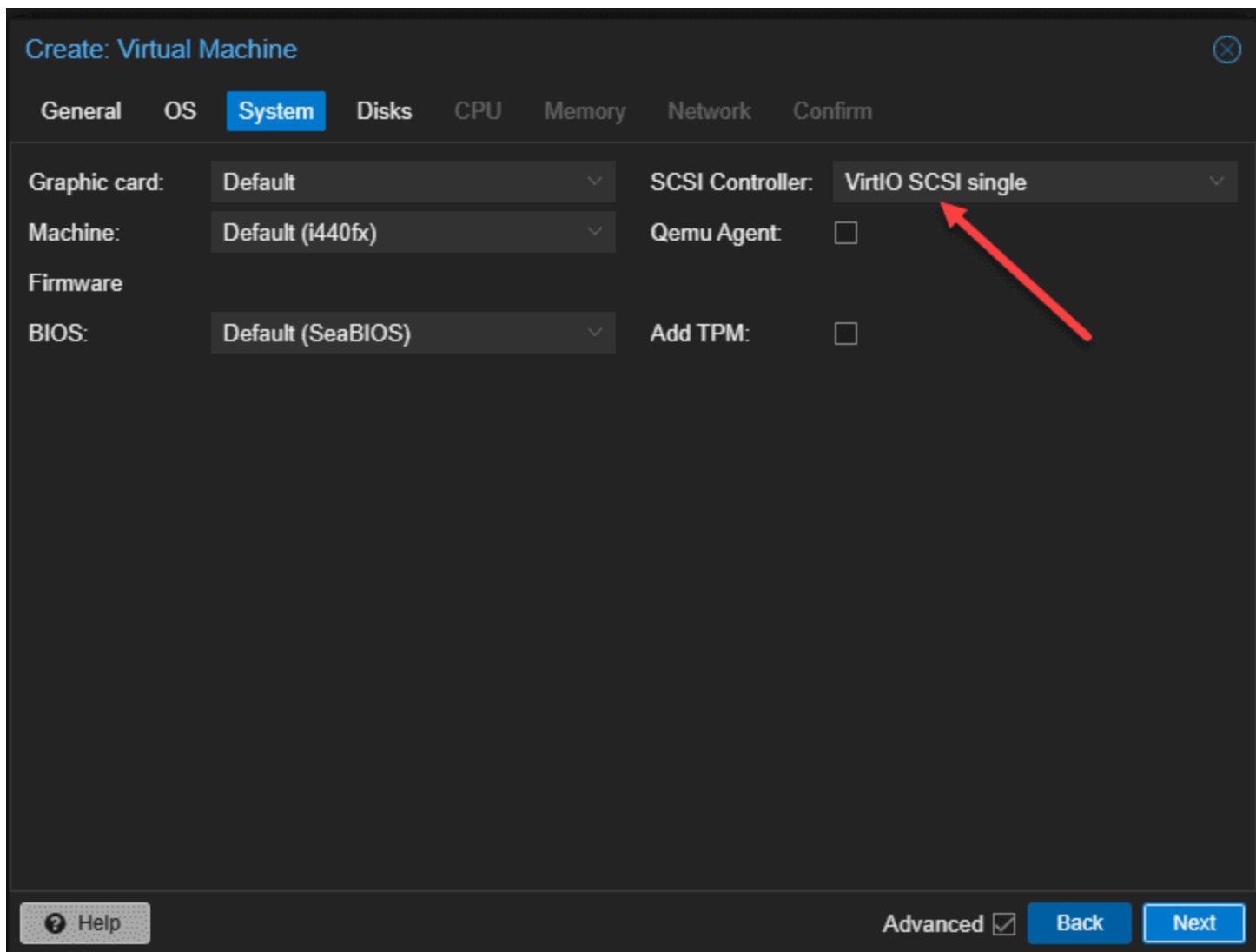
Advanced

Back

Next

Select your esxi iso image under the os tab

Here I left **VirtIO SCSI single** selected for SCSI controller.



Leaving the defaults under system for esxi nested

On the **Disks** screen, configure the disk size you want and also the Storage location for your VM files and hit **Next**.

Create: Virtual Machine



General OS System **Disks** CPU Memory Network Confirm

ide0	
------	--

Disk Bandwidth

Bus/Device:	IDE	0	Cache:	Default (No cache)
Storage:	NVMePool01		Discard:	<input type="checkbox"/>
Disk size (GiB):	256		IO thread:	<input type="checkbox"/>
Format:	Raw disk image (raw)			
SSD emulation:	<input type="checkbox"/>	Backup:	<input checked="" type="checkbox"/>	
Read-only:	<input type="checkbox"/>	Skip replication:	<input type="checkbox"/>	
	Async IO:	Default (io_uring)		

Add

Help

Advanced

Back

Next

Setting up the storage for the esxi vm

Choose your CPU options.

Create: Virtual Machine



General OS System Disks **CPU** Memory Network Confirm

Sockets: 1 Type: x86-64-v2-AES

Cores: 4 Total cores: 4

VCPUs: 4 CPU units: 100

CPU limit: unlimited Enable NUMA:

CPU Affinity: All Cores

Extra CPU Flags:

Default	- ○○○ +	md-clear	Required to let the guest OS know if MDS is mitigated correctly
Default	- ○○○ +	pcid	Meltdown fix cost reduction on Westmere, Sandy-, and IvyBridge Intel CPUs
Default	- ○○○ +	spec-ctrl	Allows improved Spectre mitigation with Intel CPUs
Default	- ○○○ +	ssbd	Protection for "Speculative Store Bypass" for Intel models
Default	- ○○○ +	ibpb	Allows improved Spectre mitigation with AMD CPUs
Default	- ○○○ +	virt-ssbd	Basis for "Speculative Store Bypass" protection for AMD models

Advanced Back Next

Cpu settings for the esxi nested vm

Configure your memory.

Create: Virtual Machine X

General OS System Disks CPU **Memory** Network Confirm

Memory (MiB): ▼ ▲

Minimum memory (MiB): ▼ ▲

Shares: Default (1000) ▼ ▲

Ballooning Device:

? Help Advanced Back Next

Memory configuration

Ok, so this is the step that surprised me a bit. I here selected **Intel E1000** which is a standard Intel driver. But I will show you what happens during the install.

Create: Virtual Machine



General OS System Disks CPU Memory Network Confirm

No network device

Bridge: vmbr0 Model: Intel E1000

VLAN Tag: 149 MAC address: auto

Firewall:

Disconnect: Rate limit (MB/s): unlimited

MTU: 1500 (1 = bridge MTU) Multiqueue:

Help

Advanced

Back

Next

Setting the network adapter to e1000

Confirm your configuration and click **Finish**.

Create: Virtual Machine (X)

General OS System Disks CPU Memory Network **Confirm**

Key ↑	Value
cores	4
cpu	x86-64-v2-AES
ide0	NVMePool01:256
ide2	local:iso/VMware-VMvisor-Installer-8.0U2-22380479.x86_64_1_.iso,media=cdrom
memory	16384
name	esxionpve
net0	e1000,bridge=vmbr0,tag=149,firewall=1
nodename	pve01
numa	0
ostype	other
scsihw	virtio-scsi-single
sockets	1
vmid	104

Start after created

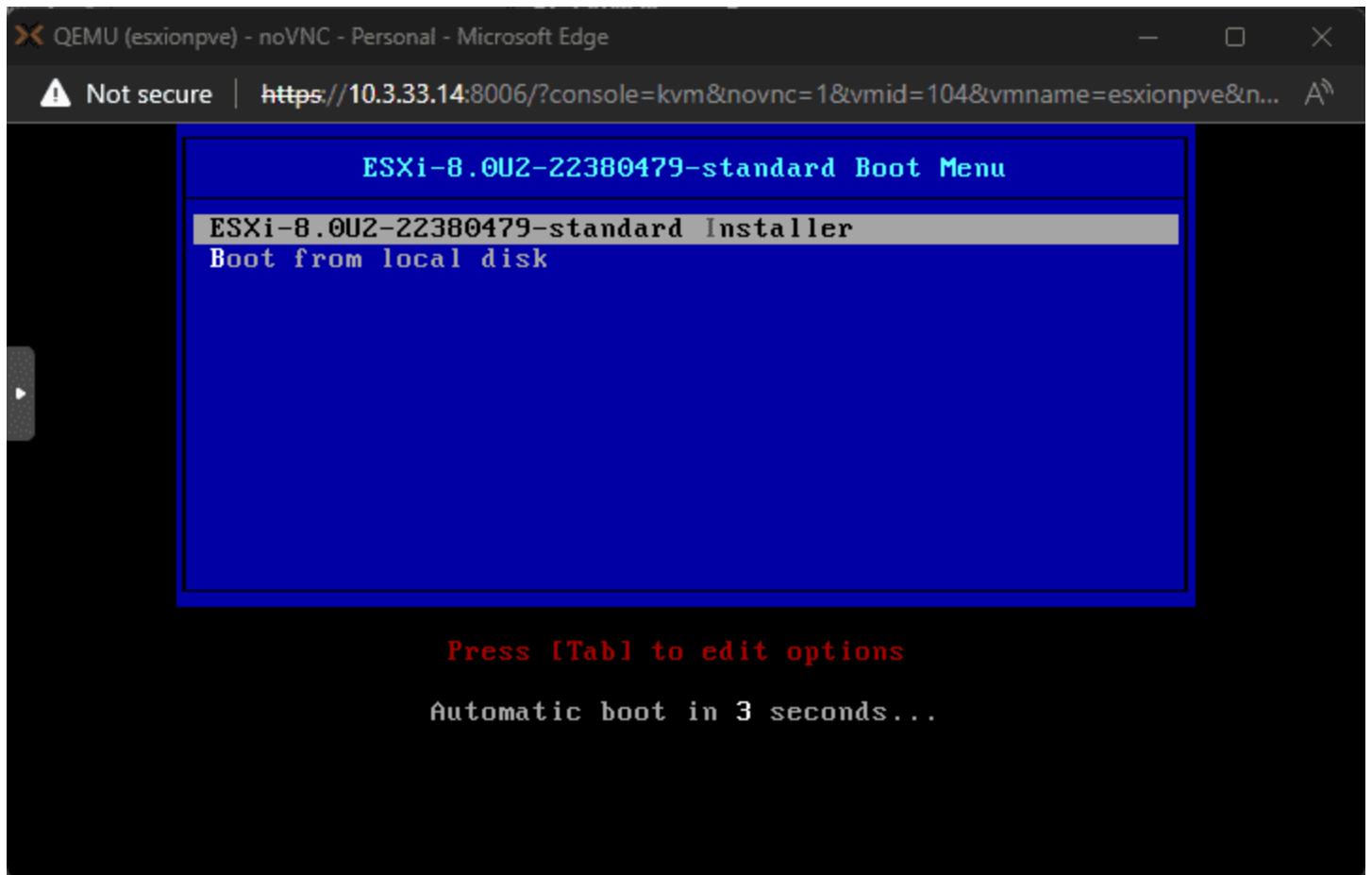
[Advanced](#) [Back](#) [Finish](#)

Confirm the installation options

Step-by-Step Installation of Nested ESXi

Let's look at how to [install ESXi](#) in Proxmox after we have created the Proxmox virtual machine to house the nested virtual machine install.

Below is booting the VMware VM guest OS in Proxmox.



Beginning the esxi 8.0 u2 installation

VMware ESXi 8.0.2 (VMKernel Release Build 22380479)

QEMU Standard PC (i440FX + PIIX, 1996)

Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
16 GiB Memory

Uncompressing boot modules...

vmx.v00
vim.v00
sb.v00
s.v00

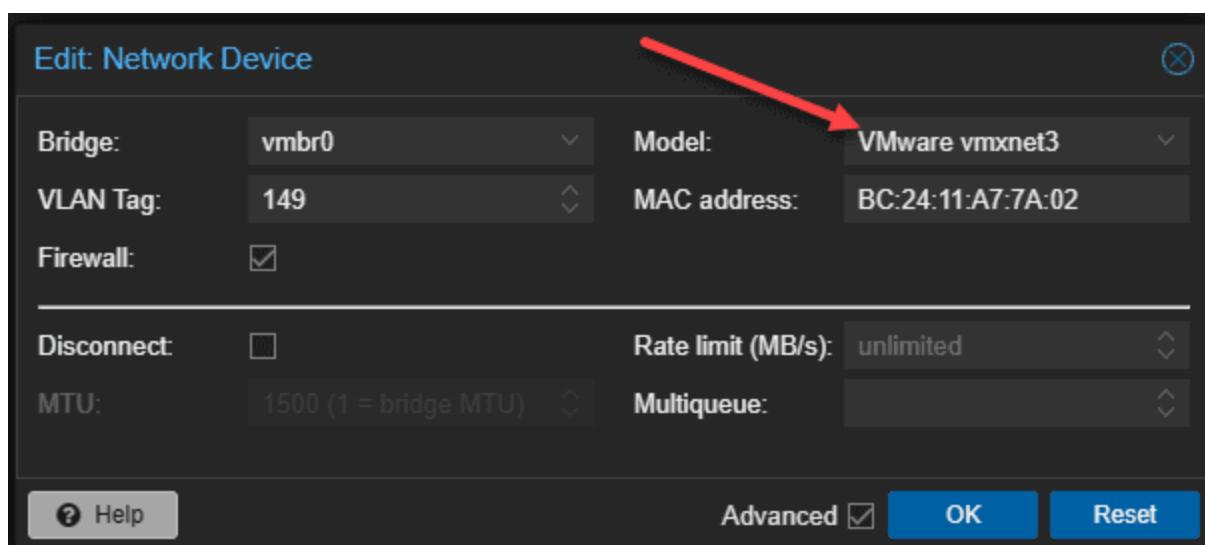
The esxi nested vm boots into the installation

OK, so I told you there was something unexpected happen with the Intel E1000 driver. It didn't detect the network adapter in ESXi.



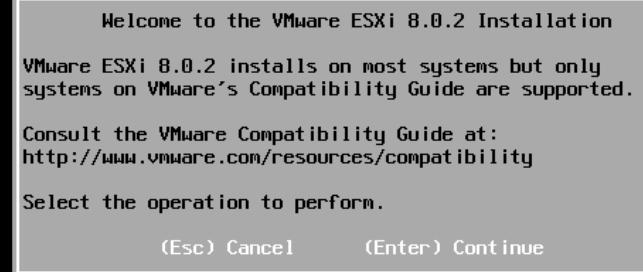
No network card detected in esxi

I powered the ESXi VM down and went back and selected **VMware vmxnet3** adapter for the model.



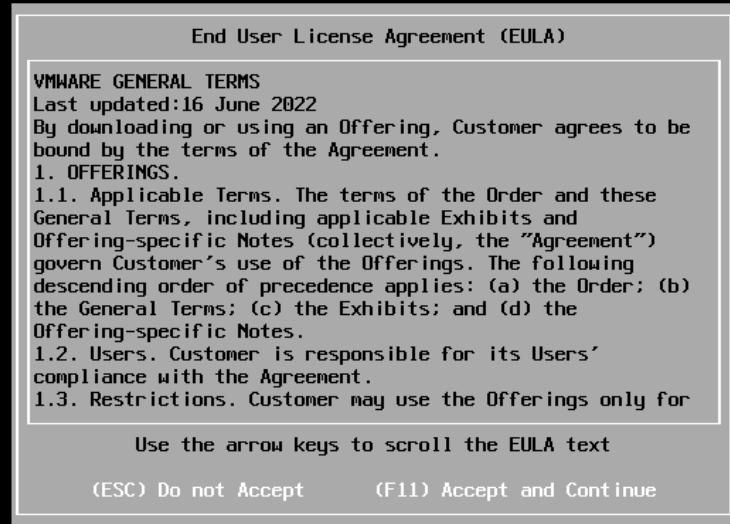
Changing the network adapter model to vmware vmxnet3

Now, the network adapter was recognized and the installation proceeded.



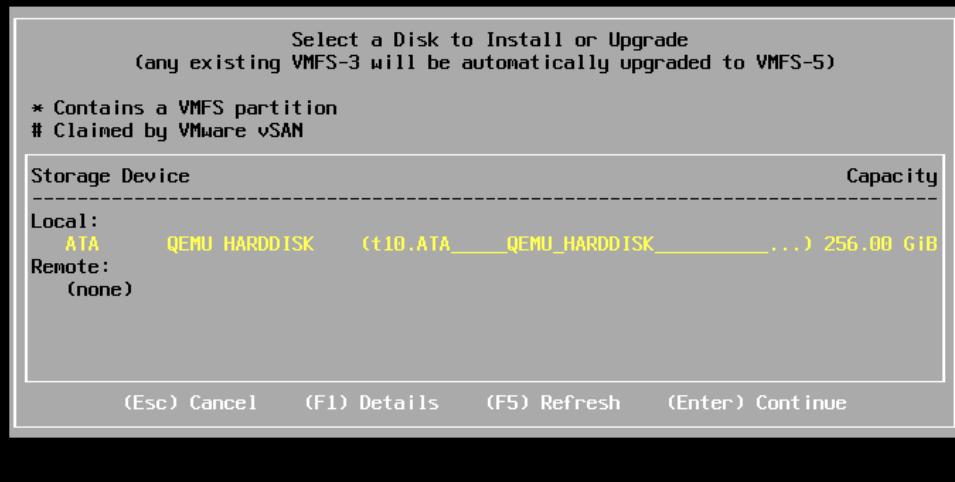
The installation of nested esxi continues

Now for the standard screens, but we will show them anyway. Accept the EULA.



Accept the eula 1

Select the target storage for the installation.



Select the installation target storage for nested esxi

Select the location for the keyboard layout.



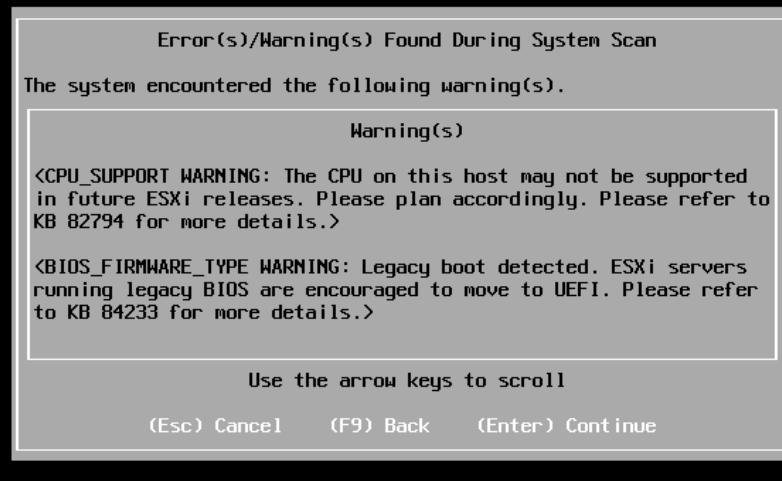
Select the esxi keyboard layout

Enter and confirm your password.



Configure the root password for esxi

I am running on an older Xeon D processor so we see the alert about an outdated processor that may not be supported in future releases. You will see the same error on [bare metal](#).

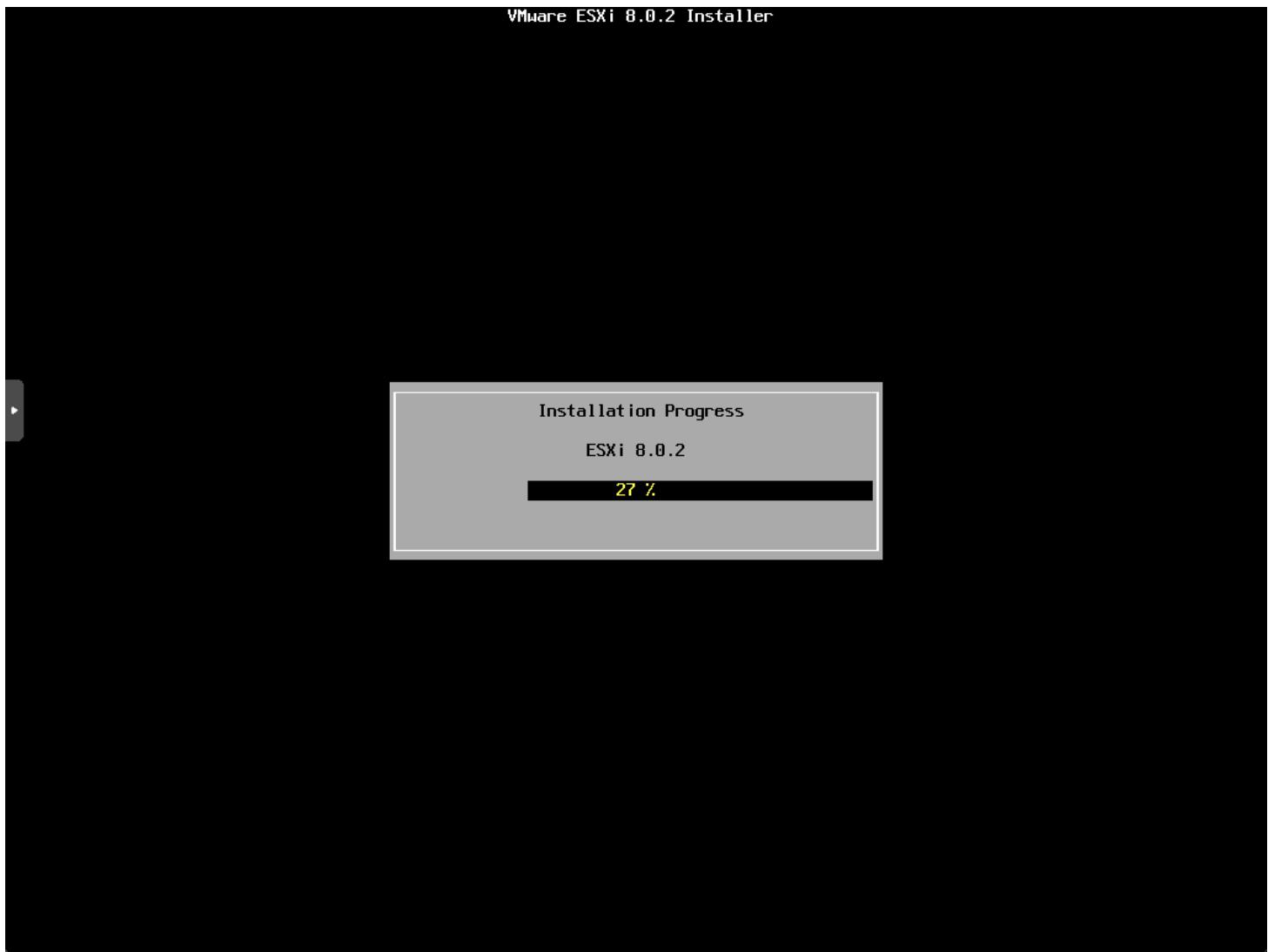


Warning about older cpu support in esxi 8.0 update 2



Confirm the installation of esxi and repartitioning

The installation begins.



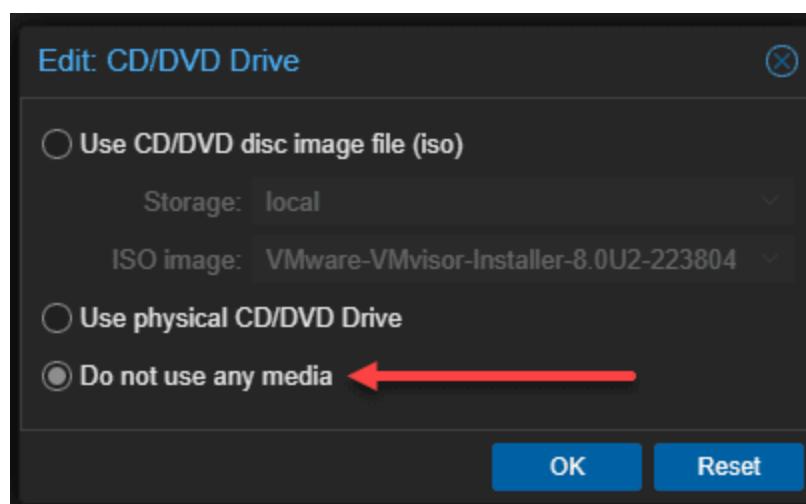
Esxi installation progress begins

Finally, we are prompted to remove the installation media and reboot.



Installation finished remove installation media

Hopping back over to Proxmox, I remove the ESXi ISO before rebooting.



Removing the iso from the esxi vm in proxmox

After initiating a reboot.

Rebooting Server
The server will shut down and reboot.
The process will take a short time to complete.

Rebooting the esxi installation

After the nested ESXi installation boots, we see it has correctly pulled an IP address from DHCP so the network adapter is working as expected.

VMware ESXi 8.0.2 (VMKernel Release Build 22380479)

QEMU Standard PC (i440FX + PIIX, 1996)

Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
16 GiB Memory

To manage this host, go to:

[https://10.1.149.152/ \(DHCP\)](https://10.1.149.152/)

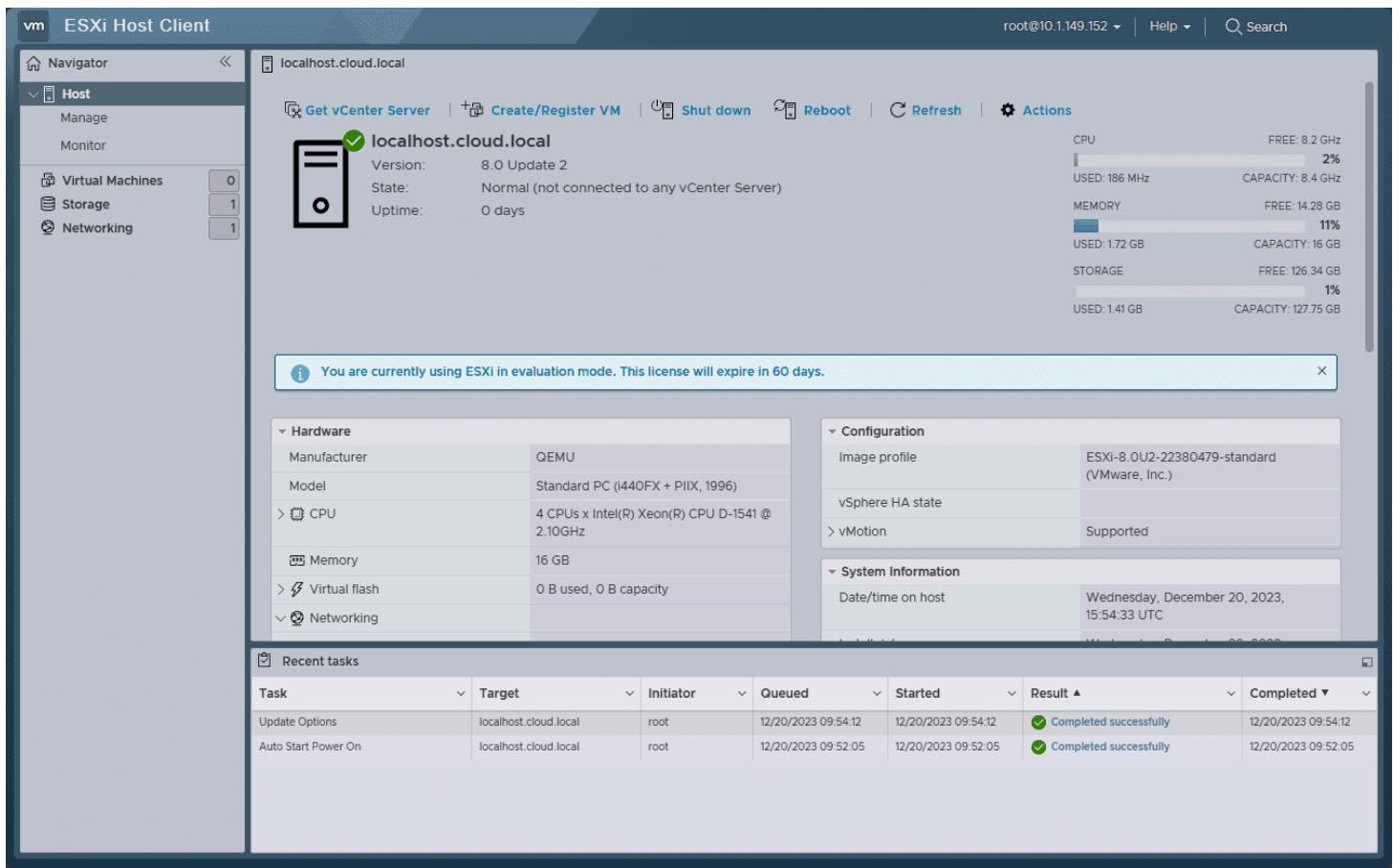
[https://\[fe80::be24:11ff:fea7:7a02\]/ \(STATIC\)](https://[fe80::be24:11ff:fea7:7a02]/)

<F2> Customize System/View Logs

<F12> Shut Down/Restart

Vmware esxi vm in proxmox boots and it correctly pulls a dhcp address

Below, I logged into the VMware host client to manage the ESXi host running in Proxmox.



Logged into the esxi host client

Managing Virtual Machines in a Nested Setup

The cool thing about working with ESXi that is nested in a Proxmox VM is that, for the most part, you won't notice much difference if you are used to accessing the ESXi host client or adding the ESXi host to the vCenter Server and managing it with vCenter.

Using advanced features in nested VMs

The great thing about running ESXi as a nested hypervisor, is you won't see any difference in the advanced features for nested VMs. You will still be able to do things like installing VMware Tools in Linux and your [Windows Server operating system](#) instances.

If you are configuring a cluster of ESXi hosts with vCenter, you can utilize features like vMotion and DRS within a nested VMware [vSphere cluster](#).

Troubleshooting Common Issues in Nested Environments

Running nested ESXi in Proxmox can be a bit of a mind-bender on the networking side. However, this is not unique to Proxmox, as running nested ESXi on a physical ESXi host can be the same challenge.

First, though, you need to understand Proxmox VLANs. I just covered this recently as well. So, check out my post on Proxmox VLANs to first understand how to configure VLANs in Proxmox.

Just remember, on the nested VMware ESXi side, you can't tag VLANs on your port groups as this will lead to "double tagging". They will instead assume the tag from the Proxmox side.

What I like to do is set up the Proxmox Linux Bridge as a trunk bridge, which is the default configuration when you make it VLAN aware. Then, you can change the tag on your network adapter configured for your VMware ESXi VM to tag the traffic from the ESXi VM.

Wrapping up

Hopefully, this blog post has been a help to any who are running Proxmox as your hypervisor running your environments. It is easy to get a virtual machine running with [VMware ESXi in a Proxmox nested environment](#). Keep in mind the need to use the VMware vmxnet3 adapter and the note on Proxmox VLAN tagging. If you are running guest VMs in your ESXi VM, you will also need to keep in mind the need to enable promiscuous mode for your Proxmox bridge.