Proxmox Datacenter Manager Documentation

Release 1.0.7

Proxmox Support Team

Tuesday, 26 May 2026

TABLE OF CONTENTS

1 Introduction **2**
- 1.1 What is Proxmox Datacenter Manager? . . . 2
- 1.2 Feature Overview . . . 2
- 1.3 Technology Stack . . . 3
- 1.4 Getting Help . . . 4
  - 1.4.1 Enterprise Support . . . 4
  - 1.4.2 Community Support Forum . . . 4
  - 1.4.3 Mailing Lists . . . 4
  - 1.4.4 Bug Tracker . . . 4
- 1.5 License . . . 4

2 Installation **5**
- 2.1 System Requirements . . . 5
  - 2.1.1 Minimum Server Requirements, for Evaluation . . . 5
  - 2.1.2 Recommended Server System Requirements . . . 5
  - 2.1.3 Supported Web Browsers for Accessing the Web Interface . . . 5
- 2.2 Installation Medium . . . 6
  - 2.2.1 Prepare Installation Medium . . . 6
  - 2.2.2 Prepare a USB Flash Drive as Installation Medium . . . 6
  - 2.2.3 Instructions for GNU/Linux . . . 6
  - 2.2.4 Instructions for macOS . . . 7
  - 2.2.5 Instructions for Windows . . . 7
  - 2.2.6 Use the Installation Medium . . . 8
  - 2.2.7 Install Proxmox Datacenter Manager Unattended . . . 8
  - 2.2.8 Install Proxmox Datacenter Manager on Debian . . . 8
- 2.3 Debian Package Repositories . . . 9
  - 2.3.1 Repository Formats . . . 9
  - 2.3.2 Debian Base Repositories . . . 9
  - 2.3.3 Proxmox Datacenter Manager Enterprise Repository . . . 10
  - 2.3.4 Proxmox Datacenter Manager No-Subscription Repository . . . 10
  - 2.3.5 Proxmox Datacenter Manager Test Repository . . . 10
  - 2.3.6 SecureApt . . . 11

3 Graphical User Interface **12**
- 3.1 Features . . . 12
- 3.2 Login . . . 13
- 3.3 User Interface Overview . . . 13
- 3.4 Sidebar . . . 13
  - 3.4.1 Dashboard . . . 14
  - 3.4.2 Views . . . 14
  - 3.4.3 Notes . . . 15
  - 3.4.4 Configuration . . . 15
  - 3.4.5 Administration . . . 15
  - 3.4.6 SDN . . . 15

3.4.7 Remotes . . . 16

4 SDN Integration **18**
4.1 Status Overview . . . 18
4.2 EVPN Integration . . . 18
4.2.1 Definitions . . . 18

5 Remotes **20**
5.1 Resource Operation . . . 20
5.2 Data Collection . . . 20
5.3 Proxmox VE Remote . . . 20
5.4 Proxmox Backup Server Remote . . . 21
5.5 Connection and Certificate Troubleshooting . . . 21

6 Automated Installations **22**
6.1 Overview . . . 22
6.2 Prepared Answers . . . 22
6.2.1 Target filter . . . 22
6.2.2 Templating . . . 23
6.3 Authentication token management . . . 23
6.4 Preparing an ISO . . . 23

7 Views **24**
7.1 Resource Selection . . . 24
7.1.1 Filter Types . . . 24
7.2 Customizable Dashboard . . . 24
7.2.1 Widgets . . . 25
7.2.2 Map Widget . . . 25
7.3 Access Control . . . 25

8 Subscription Registry **26**
8.1 Key Pool . . . 26
8.2 Node Subscription Status . . . 26
8.3 Assignment and Clearing . . . 27
8.4 Permissions . . . 28

9 Authentication & Access Control **29**
9.1 User Configuration . . . 29
9.2 API Tokens . . . 29
9.3 Access Control . . . 30
9.3.1 Privileges . . . 30
9.3.2 Access Roles . . . 30
9.3.3 Objects and Paths . . . 31
9.3.4 Configuration & Management . . . 31
9.3.5 API Token Permissions . . . 32
9.4 Two-Factor Authentication . . . 32
9.4.1 Introduction . . . 32
9.4.2 Available Second Factors . . . 32

10 Host System Administration **33**
10.1 ZFS on Linux . . . 33
10.1.1 Hardware . . . 34
10.1.2 ZFS Administration . . . 34
10.2 Host Bootloader . . . 39
10.2.1 Partitioning Scheme Used by the Installer . . . 39
10.2.2 Synchronizing the Content of the ESP with `proxmox-boot-tool` . . . 39
10.2.3 Determine which Bootloader is Used . . . 41
10.2.4 Grub . . . 42
10.2.5 Systemd-boot . . . 42

10.2.6 Editing the Kernel Commandline . . . 42
10.2.7 Override the Kernel-Version for next Boot . . . 43
10.2.8 Secure Boot . . . 44
10.3 Certificate Management . . . 46
10.3.1 Certificates for the API and SMTP . . . 46
10.3.2 Upload Custom Certificate . . . 47
10.3.3 Trusted certificates via Let's Encrypt (ACME) . . . 47
10.3.4 ACME HTTP Challenge Plugin . . . 48
10.3.5 ACME DNS API Challenge Plugin . . . 48
10.3.6 Automatic renewal of ACME certificates . . . 49
10.3.7 Manually Change Certificate over the Command Line . . . 49
10.4 Service Daemons . . . 50
10.4.1 `proxmox-datacenter-api` . . . 50
10.4.2 `proxmox-datacenter-privileged-api` . . . 50
10.5 Command-line Tools . . . 50
10.5.1 `proxmox-datacenter-manager-client` . . . 50
10.5.2 `proxmox-datacenter-manager-admin` . . . 50

11 FAQ **51**
11.1 What distribution is Proxmox Datacenter Manager (PDM) based on? . . . 51
11.2 Will Proxmox Datacenter Manager run on a 32-bit processor? . . . 51
11.3 How long will my Proxmox Datacenter Manager version be supported? . . . 51
11.4 How can I upgrade Proxmox Datacenter Manager to the next point release? . . . 51
11.5 Is there a dedicated subscription for the Proxmox Datacenter Manager? . . . 52
11.6 How can I get Enterprise Support for the Proxmox Datacenter Manager? . . . 52
11.7 How can I get access to the Proxmox Datacenter Manager Enterprise Repository? . . . 52

A Command Syntax **53**
A.1 `proxmox-datacenter-manager-admin` . . . 53
A.2 `proxmox-datacenter-manager-client` . . . 57

B Configuration Files **95**
B.1 `remotes.cfg` . . . 95
B.1.1 Options . . . 95
B.2 `views.cfg` . . . 96
B.2.1 Options . . . 96

C Roadmap **97**
C.1 Release History . . . 98
C.1.1 Proxmox Datacenter Manager 1.0 . . . 98

D Markdown Primer **109**
D.1 Markdown Basics . . . 109
D.1.1 Headings . . . 109
D.1.2 Emphasis . . . 109
D.1.3 Links . . . 109
D.1.4 Lists . . . 110
D.1.5 Tables . . . 110
D.1.6 Block Quotes . . . 110
D.1.7 Code and Snippets . . . 110

E GNU Free Documentation License **112**

Copyright (C) 2023 - 2025, Proxmox Server Solutions GmbH
Version 1.0.7 -- Tuesday, 26 May 2026

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled ”GNU Free Documentation License”.

CHAPTER

ONE

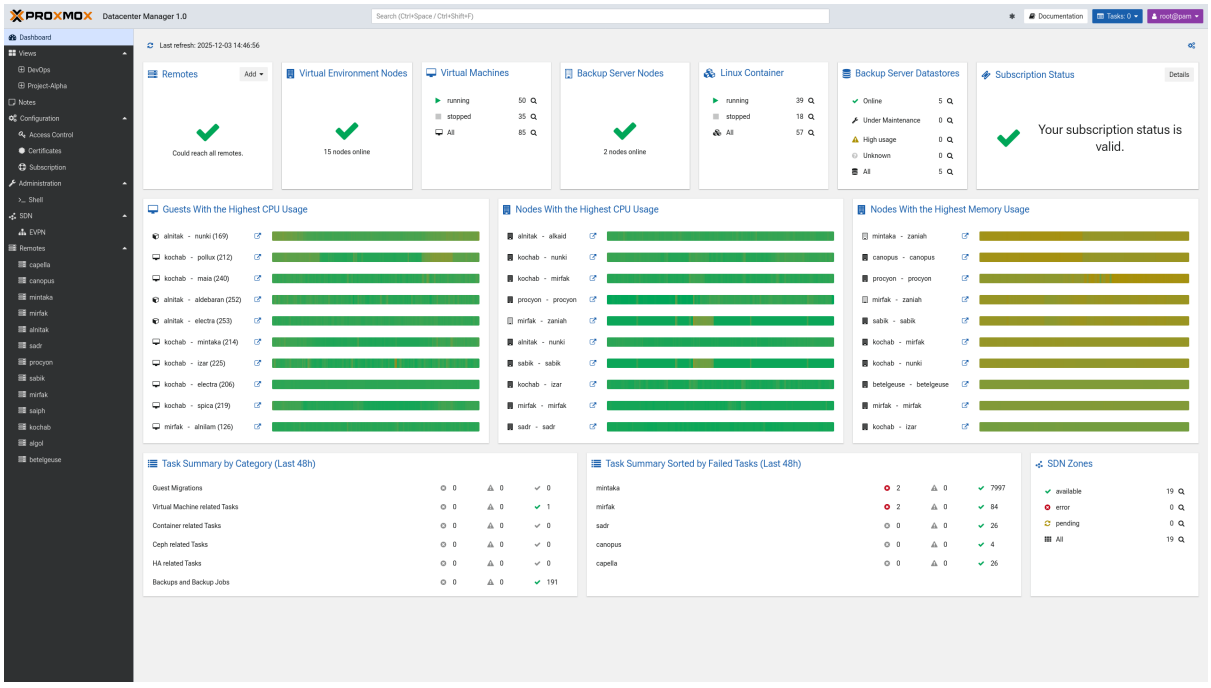
INTRODUCTION

1.1 What is Proxmox Datacenter Manager?

Proxmox Datacenter Manager is a centralized management platform designed to oversee Proxmox Virtual Environment and Proxmox Backup Server instances across disjointed locations. It provides a unified view of all registered resources, including nodes, virtual machines, containers, storages, and backup datastores, regardless of their geographic location.

This solution supports managing both single-node installations and complex clusters. In the context of Proxmox Datacenter Manager, these connected instances are referred to as **remotes**.

The platform allows for active resource control, enabling administrators to execute power operations, apply system updates, and perform live migrations across the infrastructure. It acts as a high-level control plane while providing a seamless transition ("escape hatch") to the native web interfaces of individual remotes for granular configuration. This architecture ensures a loosely coupled system where the central manager does not become a bottleneck or a single point of failure for the operation of the underlying remotes.



1.2 Feature Overview

Proxmox Datacenter Manager provides the following core capabilities:

- **Centralized Inventory**: Connect and manage a scalable number of independent nodes and clusters ("Datacenters") from a single interface.

- **Unified Resource Monitoring**: View the status, health, and load of global resources, including nodes, virtual guests, and storage backends.
- **Global Dashboard**: A high-level dashboard visualizes the state of all remotes, highlighting potential issues such as high resource consumption (CPU & memory) or failed tasks.
- **Task Aggregation**: Centralized access to task logs across the entire infrastructure for auditing and troubleshooting.
- **Lifecycle Management**: Perform basic power operations (start, stop, reboot, shutdown) on nodes and virtual guests directly from the central view.
- **Update Management**: Monitor available updates and security patches across the server fleet.
- **Cross-Cluster Migration**: Execute live migrations of virtual guests between nodes, supporting transfers within the same remote (cluster) or across different remotes.
- **Advanced Access Control**: Supports enterprise authentication standards, including LDAP/Active Directory and OpenID Connect (SSO), integrated with a granular permission system.
- **Certificate Management**: Integrated ACME support (e.g., Let's Encrypt) for automated certificate management.

1.3 Technology Stack

Proxmox Datacenter Manager relies on a modern, secure, and performant technology stack:

- **Core Language**: The project is primarily developed in the **Rust** programming Language, ensuring memory safety and high performance.
- **Backend Architecture**:
- The backend exposes a JSON-based REST API.
- It utilizes a dual-daemon architecture standard for Proxmox projects:
  1. **Main API Daemon**: Runs as an unprivileged user to handle external requests, minimizing the attack surface.
  2. **Privileged Daemon**: Runs as root and listens exclusively on a local UNIX socket to execute system-level operations.
- The backend leverages the existing, battle-tested REST/API stack from Proxmox Backup Server.
- Communication occurs over TCP port 443 (HTTPS).
- **Frontend Architecture**:
  - The web interface is a Single Page Application (SPA) written in **Rust** using the **Yew** framework.
  - It is compiled to **WebAssembly (Wasm)**, offering high performance and type safety similar to the backend.
  - The UI components are built upon the *proxmox-yew-widget-toolkit*, enabling a consistent look and feel across the Proxmox ecosystem.

1.4 Getting Help

1.4.1 Enterprise Support

Customers with an active Basic, Standard, or Premium subscription for their Proxmox remotes gain access to the Proxmox Datacenter Manager Enterprise Repository and technical support.

For more information, please visit https://www.proxmox.com or contact <mailto:sales@proxmox.com>.

1.4.2 Community Support Forum

The Proxmox Community Forum is a primary resource for user discussions and knowledge sharing. Moderated by the Proxmox support team, it connects a global user base and serves as an extensive database of solutions and configurations.

1.4.3 Mailing Lists

Proxmox Datacenter Manager is open-source software. The development mailing list is the primary communication channel for contributing developers:

Mailing list for developers
PDM Development List

1.4.4 Bug Tracker

Proxmox maintains a public issue tracker at https://bugzilla.proxmox.com. This system tracks bug reports and feature requests. Users can subscribe to issues to receive notifications regarding the status and resolution of submitted issues.

1.5 License

Copyright (C) 2023 - 2025, Proxmox Server Solutions GmbH

This software is written by Proxmox Server Solutions GmbH <support@proxmox.com>.

Proxmox Datacenter Manager is free and open-source software: you can use it, redistribute it, and/or modify it under the terms of the **GNU Affero General Public License** as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but **WITHOUT ANY WARRANTY**; without even the implied warranty of **MERCHANTABILITY** or **FITNESS FOR A PARTICULAR PURPOSE**. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see AGPL3.

CHAPTER

TWO

INSTALLATION

Proxmox Datacenter Manager can either be installed with a graphical installer or on top of Debian from the provided package repository.

2.1 System Requirements

We recommend using high quality server hardware when running Proxmox Datacenter Manager in production. While no managed remote or resource depends on Proxmox Datacenter Manager to run, you might find that Proxmox Datacenter Manager will become a convenient and critical tool in your operations fast enough.

2.1.1 Minimum Server Requirements, for Evaluation

These minimum requirements are for evaluation purposes only and should not be used in production.

- CPU: 64bit (*x86-64* or *AMD64*), 1+ Cores
- Memory (RAM): 1 GiB RAM
- Hard drive: more than 10 GB of space.
- Network card (NIC)

2.1.2 Recommended Server System Requirements

- CPU: Modern AMD or Intel 64-bit based CPU, with at least 2 cores
- Memory: minimum 4 GiB for the OS.
- OS storage:
  - 40 GB, or more, free storage space
  - Use a hardware RAID with battery protected write cache (*BBU*) or a redundant ZFS setup (ZFS is not compatible with a hardware RAID controller).
- Redundant Multi-GBit/s network interface cards (NICs)

2.1.3 Supported Web Browsers for Accessing the Web Interface

To access the server's web-based user interface, we recommend using one of the following browsers:

- Firefox, a release from the current year, or the latest Extended Support Release
- Chrome, a release from the current year
- Microsoft's currently supported version of Edge
- Safari, a release from the current year

2.2 Installation Medium

Proxmox Datacenter Manager can be installed via *different methods*. The recommended method is the usage of an installation medium, to simply boot the interactive installer.

2.2.1 Prepare Installation Medium

Download the installer ISO image from https://www.proxmox.com/downloads.

The Proxmox Datacenter Manager installation medium is a hybrid ISO image. It works in two ways:

- An ISO image file ready to burn to a DVD.
- A raw sector (IMG) image file ready to copy to a USB flash drive (USB stick).

Using a USB flash drive to install Proxmox Datacenter Manager is the recommended way since it is the faster and more frequently available option these days.

2.2.2 Prepare a USB Flash Drive as Installation Medium

The flash drive needs to have at least 2 GB of storage space.

Note

Do not use *UNetbootin*. It does not work with the Proxmox Datacenter Manager installation image.

Important

Existing data on the USB flash drive will be overwritten. Therefore, make sure that it does not contain any still needed data and unmount it afterwards again before proceeding.

2.2.3 Instructions for GNU/Linux

On Unix-like operating systems use the `dd` command to copy the ISO image to the USB flash drive. First find the correct device name of the USB flash drive (see below). Then run the `dd` command. Depending on your environment, you will need to have root privileges to execute `dd`.

```
# dd bs=1M conv=fdatasync if=./proxmox-datacenter-manager_*.iso of=/dev/XYZ
```

Note

Be sure to replace `/dev/XYZ` with the correct device name and adapt the input filename (*if*) path.

Caution

Be very careful, and do not overwrite the wrong disk!

Find the Correct USB Device Name

There are two ways to find out the name of the USB flash drive. The first one is to compare the last lines of the `dmesg` command output before and after plugging in the flash drive. The second way is to compare the output of the `lsblk` command. Open a terminal and run:

```
# lsblk
```

Then plug in your USB flash drive and run the command again:

```
# lsblk
```

A new device will appear. This is the one you want to use. To be on the extra safe side check if the reported size matches your USB flash drive.

2.2.4 Instructions for macOS

Open the terminal (query *Terminal* in Spotlight).

Convert the `.iso` file to `.dmg` format using the convert option of `hdiutil`, for example:

```
# hdiutil convert proxmox-datacenter-manager_*.iso -format UDRW -o proxmox-datacenter-manager_
↪*.dmg
```

Note

macOS tends to automatically add `.dmg` to the output file name.

To get the current list of devices run the command:

```
# diskutil list
```

Now insert the USB flash drive and run this command again to determine which device node has been assigned to it. (e.g., `/dev/diskX`).

```
# diskutil list
# diskutil unmountDisk /dev/diskX
```

Note

replace *X* with the disk number from the last command.

```
# sudo dd if=proxmox-datacenter-manager_*.dmg bs=1M of=/dev/rdiskX
```

Note

rdiskX, instead of *diskX*, in the last command is intended. It will increase the write speed.

2.2.5 Instructions for Windows

Using Etcher

Etcher works out of the box. Download Etcher from https://etcher.io. It will guide you through the process of selecting the ISO and your USB flash drive.

Using Rufus

Rufus is a more lightweight alternative, but you need to use the **DD mode** to make it work. Download Rufus from https://rufus.ie/. Either install it or use the portable version. Select the destination drive and the downloaded Proxmox ISO file.

Important

Once you click *Start*, you have to click *No* on the dialog asking to download a different version of Grub. In the next dialog select **DD mode**.

2.2.6 Use the Installation Medium

Insert the created USB flash drive (or DVD) into your server. Continue by reading the installer chapter, which also describes possible boot issues.

Using our provided disk image (ISO file) is the recommended installation method, as it includes a convenient installer, a complete Debian system as well as all necessary packages for the Proxmox Datacenter Manager.

Once you have created an installation_medium, the booted installer will guide you through the setup process. It will help you to partition your disks, apply basic settings such as the language, time zone and network configuration, and finally install all required packages within minutes.

As an alternative to the interactive installer, advanced users may wish to install Proxmox Datacenter Manager *unattended*.

With sufficient Debian knowledge, you can also install Proxmox Datacenter Manager *on top of Debian* yourself.

2.2.7 Install Proxmox Datacenter Manager Unattended

It is possible to install Proxmox Datacenter Manager automatically in an unattended manner. This enables you to fully automate the setup process on bare-metal. Once the installation is complete and the host has booted up, automation tools like Ansible can be used to further configure the installation.

The necessary options for the installer must be provided in an answer file. This file allows the use of filter rules to determine which disks and network cards should be used.

To use the automated installation, it is first necessary to prepare an installation ISO. For more details and information on the unattended installation see our wiki.

2.2.8 Install Proxmox Datacenter Manager on Debian

Proxmox ships as a set of Debian packages which can be installed on top of a standard Debian installation. After configuring the *Debian Package Repositories*, you need to run:

```
# apt update
# apt install proxmox-datacenter-manager proxmox-datacenter-manager-ui
```

The above commands keep the current (Debian) kernel and install a minimal set of required packages.

You can install the Proxmox default kernel with ZFS support by using:

```
# apt update
# apt install proxmox-default-kernel
```

Caution

Installing Proxmox Datacenter Manager on top of an existing Debian installation looks easy, but it assumes that the base system and local storage have been set up correctly. In general this is not trivial, especially when LVM or ZFS is used. The network configuration is completely up to you as well.

Note

You can access the web interface of the Proxmox Datacenter Manager with your web browser, using HTTPS on port 8443. For example at `https://<ip-or-dns-name>:8443`

2.3 Debian Package Repositories

All Debian based systems use **APT**_ as a package management tool. The lists of repositories are defined in `/etc/apt/sources.list` and the `.list` or `.sources` files found in the `/etc/apt/sources.d/` directory. Updates can be installed directly with the `apt` command-line tool, or via the GUI.

2.3.1 Repository Formats

APT_ repositories can be configured in two distinct formats, the old single line format and the newer deb822 format. No matter what format you choose, `apt update` will fetch the information from all configured sources.

Single Line

Single line repositories are defined in `.list` files list one package repository per line, with the most preferred source listed first. Empty lines are ignored and a # character anywhere on a line marks the remainder of that line as a comment.

deb822 Style

The newer deb822 multiline format is used in `.sources` files. Each repository consists of a stanza with multiple key value pairs. A stanza is simply a group of lines. One file can contain multiple stanzas by separating them with a blank line. You can still use # to comment out lines.

Note

Modernizing your repositories is recommended under Debian Trixie, as `apt` will complain about older repository definitions otherwise. You can run the command `apt modernize-sources` to modernize your existing repositories automatically.

2.3.2 Debian Base Repositories

You will need a Debian base repository as a minimum to get updates for all packages provided by Debian itself:

Listing 1: File: `/etc/apt/sources.list.d/debian.sources`

```
Types: deb
URIs: http://deb.debian.org/debian/
Suites: trixie trixie-updates
Components: main contrib non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
Types: deb
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main contrib non-free-firmware
Signed-By: /usr/share/keyrings/debian-archive-keyring.gpg
```

In addition, you need a package repository from Proxmox to get Proxmox Datacenter Manager updates.

2.3.3 Proxmox Datacenter Manager Enterprise Repository

This is the stable, recommended repository. It is available for all users fulfilling the Proxmox Datacenter Manager *subscription requirements*. It contains the most stable packages, and is suitable for production use. The `pdm-enterprise` repository is enabled by default:

Listing 2: File: `/etc/apt/sources.list.d/pdm-enterprise.sources`

```
Types: deb
URIs: https://enterprise.proxmox.com/debian/pdm
Suites: trixie
Components: pdm-enterprise
Signed-By: /usr/share/keyrings/proxmox-archive-keyring.gpg
```

The change-log and details of each package can be viewed in the web UI.

Please note that you need a valid subscription key to access this repository, and for that your remote nodes need Basic or higher subscriptions. See the *FAQ* for details.

Note

You can disable this repository by adding the line `Enabled: false` to the stanza.

2.3.4 Proxmox Datacenter Manager No-Subscription Repository

As the name suggests, you do not need a subscription key to access this repository. It can be used for testing and non-production use. It is not recommended to use it on production servers, because these packages are not always heavily tested and validated.

We recommend to configure this repository in `/etc/apt/sources.list.d/proxmox.sources`.

Listing 3: File: `/etc/apt/sources.list.d/proxmox.sources`

```
Types: deb
URIs: http://download.proxmox.com/debian/pdm
Suites: trixie
Components: pdm-no-subscription
Signed-By: /usr/share/keyrings/proxmox-archive-keyring.gpg
```

2.3.5 Proxmox Datacenter Manager Test Repository

This repository contains the latest packages and is heavily used by developers to test new features.

Warning

the `pdm-test` repository should (as the name implies) only be used to test new features or bug fixes.

You can access this repository by adding the following stanza to `/etc/apt/sources.list.d/proxmox.sources`:

Listing 4: sources.list entry for `pdm-test`

```
Types: deb
URIs: http://download.proxmox.com/debian/pdm
Suites: trixie
Components: pdm-test
Signed-By: /usr/share/keyrings/proxmox-archive-keyring.gpg
```

2.3.6 SecureApt

The *Release* files in the repositories are signed with GnuPG. APT is using these signatures to verify that all packages are from a trusted source.

If you install Proxmox Datacenter Manager from an official ISO image, the verification key is already installed.

If you install Proxmox Datacenter Manager on top of Debian, download and install the key with the following commands:

```
# wget https://enterprise.proxmox.com/debian/proxmox-archive-keyring-trixie.gpg -O /usr/share/
↪keyrings/proxmox-archive-keyring.gpg
```

Note

The *wget* command above adds the keyring for Proxmox releases based on Debian Trixie. Once the *proxmox-archive-keyring* package is installed, it will manage this file. At that point, the hashes below may no longer match the hashes of this file, as keys for new Proxmox releases get added or removed. This is intended, *apt* will ensure that only trusted keys are being used. **Modifying this file is discouraged once 'proxmox-archive-keyring' is installed.**

Verify the SHA256 checksum afterwards with the expected output below:

```
# sha256sum /usr/share/keyrings/proxmox-archive-keyring.gpg
136673be77aba35dcce385b28737689ad64fd785a797e57897589aed08db6e45 /usr/share/keyrings/proxmox-
↪archive-keyring.gpg
```

and the md5sum, with the expected output below:

```
# md5sum /usr/share/keyrings/proxmox-archive-keyring.gpg
77c8b1166d15ce8350102ab1bca2fcbf /usr/share/keyrings/proxmox-archive-keyring.gpg
```

Note

Make sure that the path that you download the key to, matches the path specified in the `Signed-By:` lines in your repository stanzas from above.

CHAPTER

THREE

GRAPHICAL USER INTERFACE

Proxmox Datacenter Management offers an integrated, web-based interface to manage the server. This means that you can carry out all administration tasks through your web browser, and that you don't have to worry about installing extra management tools. The web interface also provides a built-in console, so if you prefer the command line or need some extra control, you have this option.

The web interface can be accessed via https://youripaddress:8443. The default login is *root*, and the password is either the one specified during the installation process or the password of the root user, in case of installation on top of Debian.

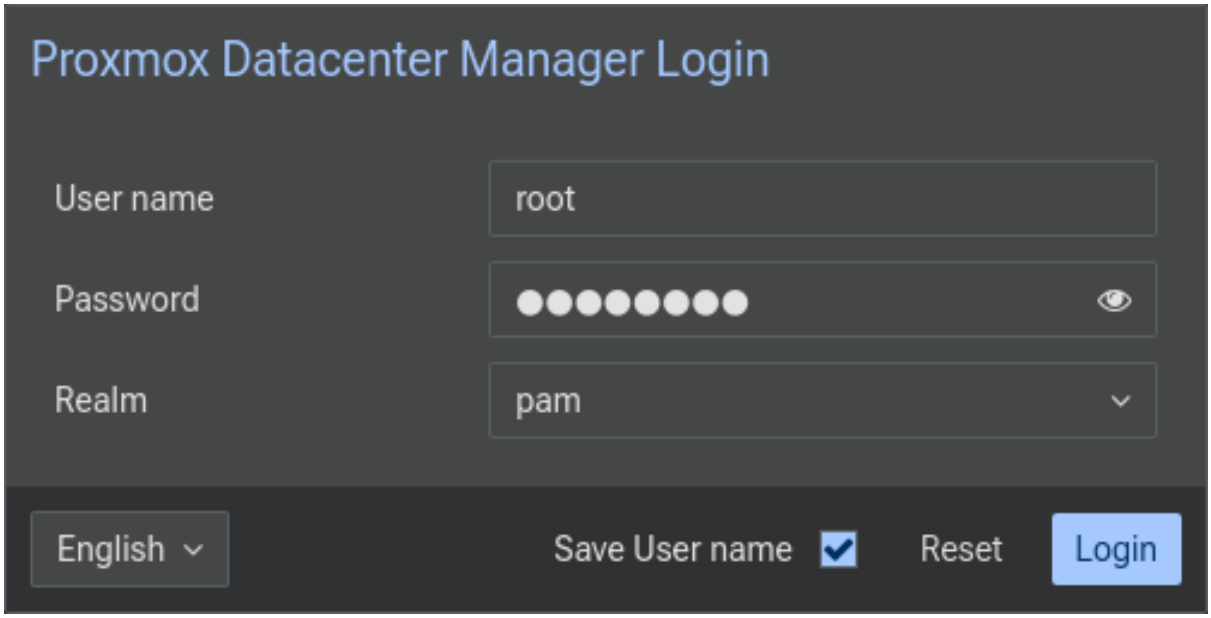
Note

Most of the descriptions below will focus on the user interface set to "English", which uses the left-to-right text direction. If you use a language with a text direction from right-to-left, then some elements will appear on the right when the description says "left" and vice versa.

3.1 Features

- Modern management interface for Proxmox Datacenter Manager
- Customizable Views.
- Management of remotes, resources, users, permissions, etc.
- Secure HTML5 console
- Support for multiple authentication sources
- Support for multiple languages
- Based on Yew, a modern Rust framework for creating multi-threaded, front-end web apps with WebAssembly.

3.2 Login



When you connect to the web interface, you will first see the login window. Proxmox Datacenter Manager supports various languages and authentication back ends (*Realms*), both of which can be selected here.

Note

For convenience, you can save the username on the client side, by selecting the "Save User name" checkbox at the bottom of the window.

3.3 User Interface Overview

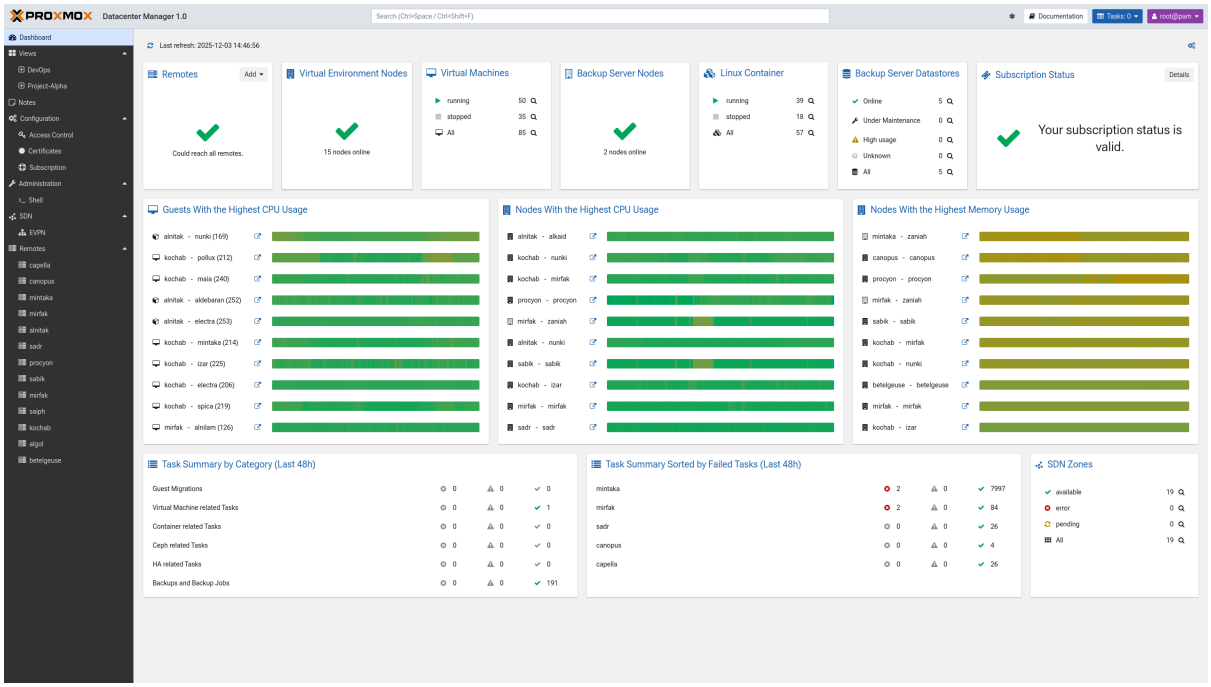
The Proxmox Datacenter Manager graphical interface can be roughly split into three different section:

- **Header**: On the left the header shows the current version information. In the middle a search bar is found that helps you find different remotes and resources. While on the right you can toggle the dark mode of the theme, go to the user documentation, open a list of tasks and open a menu to configure the theme and language that is used as well as log out.
- **Sidebar**: Below the header on the left the sidebar contains the main menu with the different menu options listed.
- **Main Panel**: The biggest part of the interface is taken up by the main panel. Its content will change depending on the menu selected in the sidebar. To start, the dashboard will be shown.

3.4 Sidebar

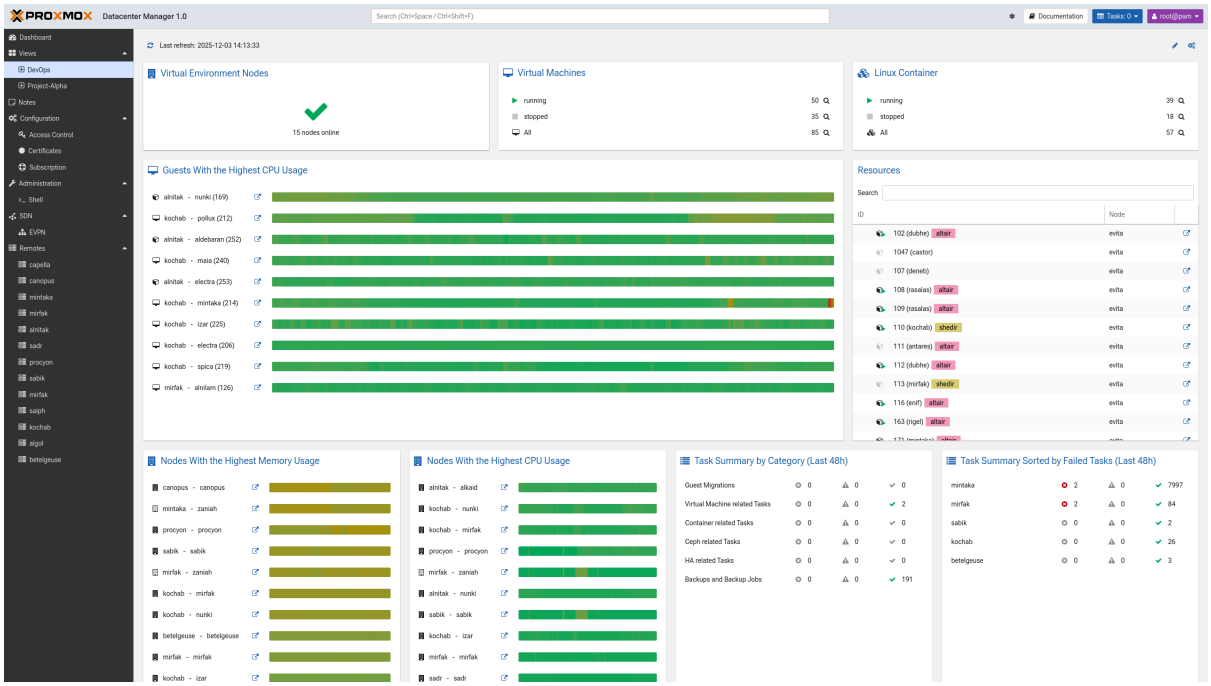
In the sidebar, on the left side of the page, you can see various items relating to specific management activities.

3.4.1 Dashboard



The dashboard gives an overview of all configured remotes and resources. Including whether remotes and their resources are up and running. Allowing you to tell at a glimpse how many VMs and CTs are running or stopped across your data center and the state of all configured datastores. Information on running tasks, CPU and memory usage, software defined networking (SDN) zones and the subscription status is shown as well.

3.4.2 Views



Views essentially allow you to create a custom dashboard. You can create a new view by copying an existing view or creating it as an empty view. Then a set of filters can be applied to a view, so it will only include information on specific resources or remotes. After creating a new view, select it from the sidebar to see what it looks like. From there you can also adjust its layout and widgets.

3.4.3 Notes

In the notes section you can keep track of information that might be useful to other administrators or yourself in the future. To format the notes, *Markdown* can be used. Notes are shared across all data center users that have been granted access to them.

3.4.4 Configuration

To configure your Proxmox Datacenter Manager, navigate to the ”Configuration” menu. It allows you to change the time and timezone, the DNS server and the its network interfaces. A second tab makes the WebAuthn Two Factor Authentication (TFA) settings available. There are also several sub-menus:

- **Access Control**: Manage users, API tokens, TFA settings, token and user permissions as well as authentication realms.
- **Certificates**: Set up custom TLS or automated ACME certificates.
- **Subscription**: To get proper support and access to the enterprise repository for your Proxmox Datacenter Manager, you can use this menu to check the subscription status.

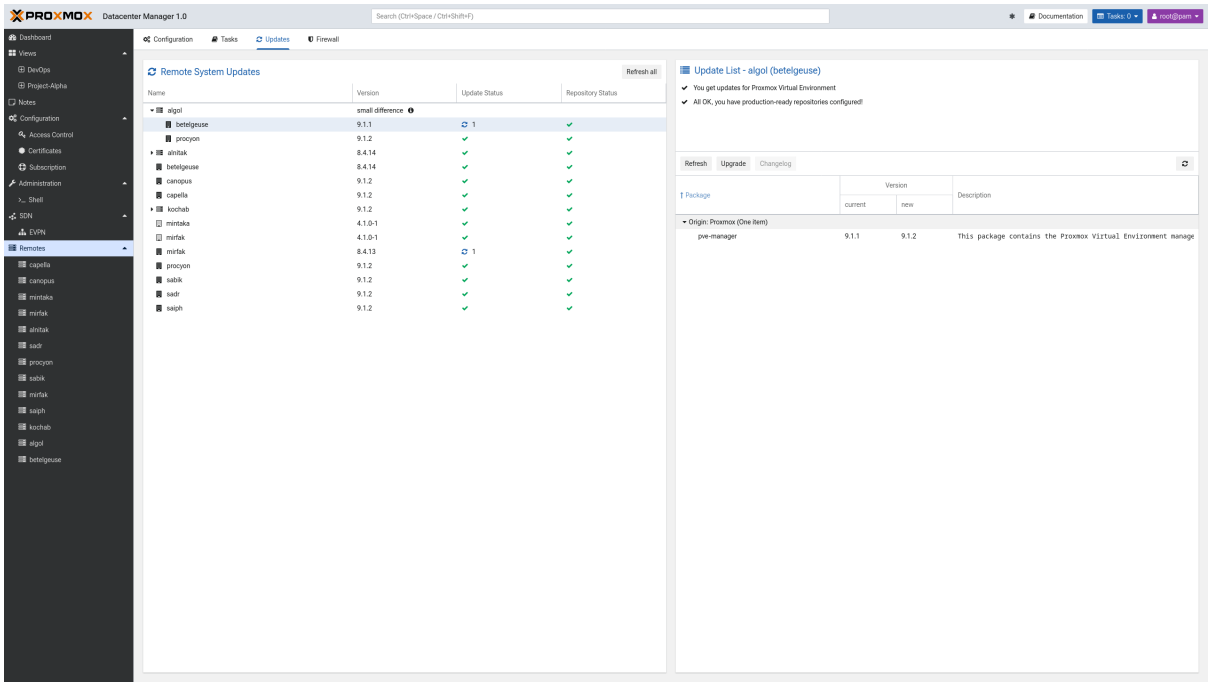
3.4.5 Administration

The administration menu can be used to get an overview of the Proxmox Datacenter Manager node itself. Through the sub-menu ”Shell” you can access the host’s shell. Several tabs allow you to manage different aspects of the node, such as:

- **Node Status**: See the CPU utilization, memory usage and other metrics of the Proxmox Datacenter Manager. Here you can also access package versions and the system report of your host, as well as reboot or turn of the host.
- **Updates**: Manage and install updates.
- **Repositories**: Add, enable and inspect update repositories.
- **Syslog**: Access the hosts system log.
- **Tasks**: An overview of all tasks of the host.

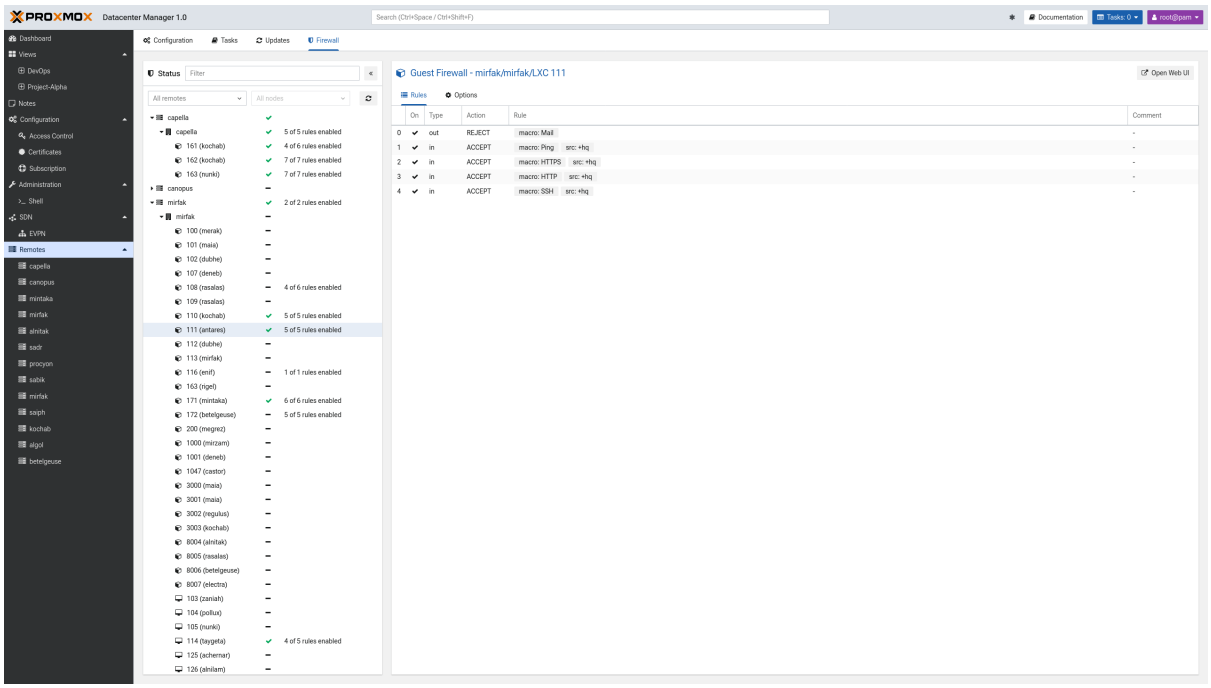
3.4.6 SDN

This menu provides an overview of all SDN zones across all configured Proxmox VE remotes. The EVPN menu can be used to set up EVPN zones across multiple remotes via a single interface. More detailed information on how to use Proxmox Datacenter Manager’s SDN integration can be found in the *SDN Integration* section.



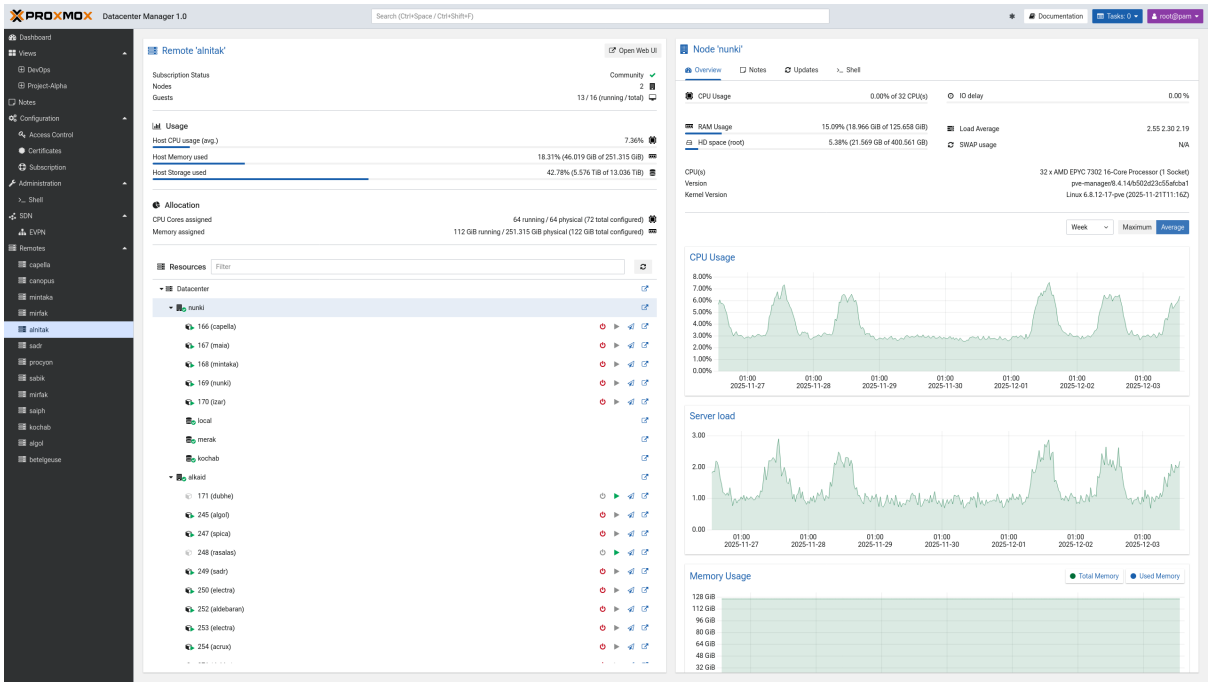
3.4.7 Remotes

Remotes allows you to configure new remotes as well as manage existing one. The "Remotes" menu itself provides different tabs to provide a unified view of your entire data center:

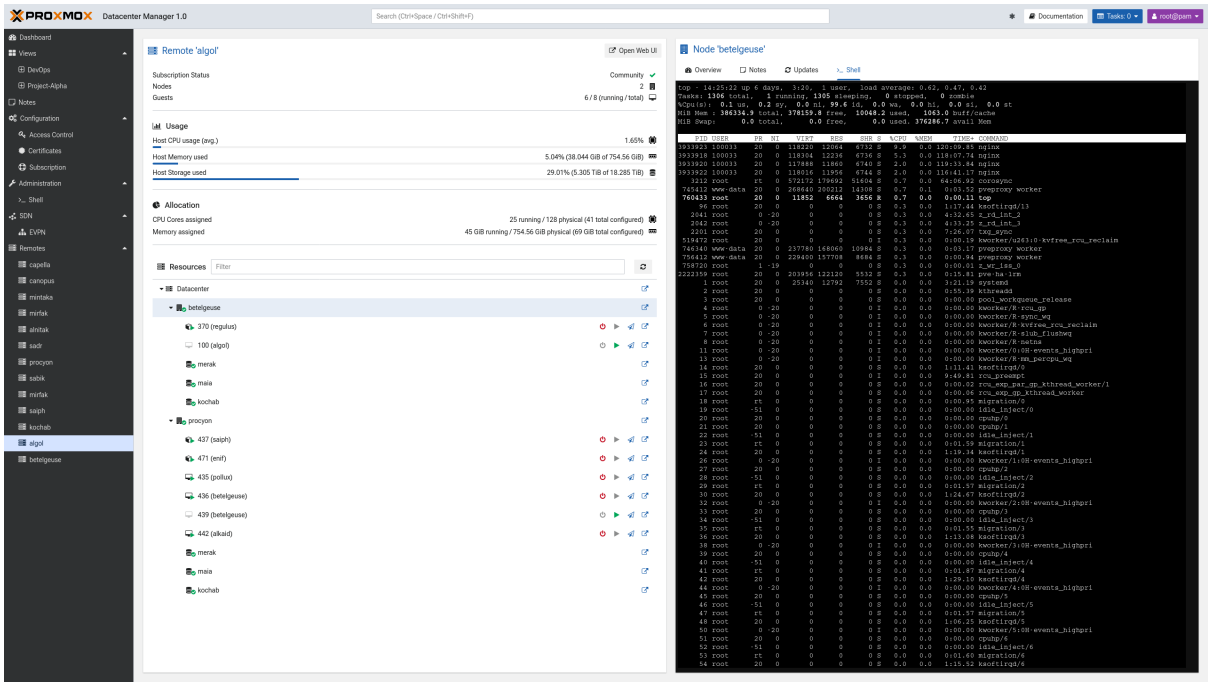


- **Configuration**: Shows a list of configured remotes and some basic information on them. It also allows configuring new remotes.
- **Tasks**: An overview of all tasks across the entire data center that can be filtered.
- **Updates**: Allows managing updates and package versions for all remotes.
- **Firewall**: Shows a list of configured firewall rules and settings for each Proxmox VE remote cluster, its nodes, and guests.

All configured remotes are also listed here in the sidebar. Each menu entry provides a split panel that contains the most important information for each remote in an easy to navigate interface.



For Proxmox VE remotes this includes an overview of all nodes and guests in a cluster. By selecting a guest or a node in the tree on the right, you can access its more detailed tabs on the left. For nodes you can get an overview of their metrics, a read-only view of their notes and updates as well as access to a "Shell" tab. This last tabs gives you access to the nodes shell, making management easy.



Proxmox Backup Server remotes show an overview of all their datastores on the left. If you select the node itself, you can see an overview of its most important metrics and a separate tab that shows available updates. When selecting a datastore its usage and disk I/O are shown first. A second tab provides a detailed view of its content by namespace.

CHAPTER

FOUR

SDN INTEGRATION

The Proxmox Datacenter Manager allows managing SDN zones and vnets across multiple remotes and provides an overview of the current state of SDN entities.

4.1 Status Overview

The status overview shows the current status (available / error / unknown) of all zones on all remotes. This is equivalent to the status shown in the SDN overview of the Proxmox VE Web UI. A summary is also shown on the dashboard, allowing users to quickly identify if there are any erroneous SDN zones on any remote.

4.2 EVPN Integration

The EVPN overview shows an aggregated overview of the contents of EVPN zones / routing table instances of all configured clusters.

Note

Currently, the integration operates under the assumption that EVPN controllers with the same ASN are interconnected and part of the same overlay network. Zones and Vnets with the same ASN:VNI tag will get automatically merged in the overview trees.

The EVPN integration respects the 'Route Target Import' field of an EVPN zone and assumes any Zones / Vnets with that Route Target are imported as well.

4.2.1 Definitions

Currently, the SDN stack in Proxmox VE uses the terms Zones and VNets, which are specific to the Proxmox VE stack. The following definitions try to make the relationship of those entities to the more commonly used definitions in RFC 7432 and RFC 9136 clearer:

A EVPN zone represents a routing table instance (identified by its ASN:VNI tag). This is also known as an IP-VRF It is associated with a VXLAN VNI (the VRF-VXLAN tag of a zone) and also referred to as L3VNI.

A vnet in an EVPN zone represents a bridging table (identified by its ASN:VNI tag). This is also known as a MAC-VRF. One IP-VRF can contain multiple MAC-VRFs. Analogous to a EVPN zone it is associated with a VXLAN VNI (the tag of a vnet) and also referred to as L2VNI.

Remotes

This view provides an overview of which zones are available on a remote and which vnets it contains. It shows the vnets that are locally configured on that remote, as well as the vnets that get imported either automatically (due to matching ASN:VNI tags) or manually (due to being specified in the 'Route Target Import' setting). Vnets that are not local to a remote are shown slightly greyed out, so they can be distinguished easily.

It contains the following columns:

- Name: The name of the remote / zone / vnet
- L3VNI: The VRF-VXLAN tag configured in the zone
- L2VNI: The tag configured in the vnet
- External: Whether this VNet is locally configured or from another remote
- Imported: Whether this VNet was manually imported, due to a respective 'Route Target Import' entry

IP-VRF

This view provides an overview of all available IP-VRFs and their contents. This view shows only VNets that are naturally part of an IP-VRF due to their zone having the same ASN:VNI combination. It can be used to see which VNets would get imported when specifying the respective ASN:VNI in the 'Route Target Import' field.

It contains the following columns:

- Name: The name of the remote / zone / vnet
- ASN: The VRF-VXLAN tag configured in the zone
- VNI: The L3VNI (for zones) or L2VNI (for vnets)
- Zone: The name of the zone that contains the vnet
- Remote: The name of the remote that contains the zone (and therefore vnet).

Status Panel

Selecting a zone or vnet shows the current status of the IP-VRF / MAC-VRF for the selected zone / vnet on a given node. The node can be selected via the dropdown in the EVPN status panel.

For zones it shows the contents of the IP-VRF, as seen by the kernel. This means that routes for guests located on the note do not show up in the IP-VRF status, since they are handled by the connected route for the subnet. For vnets it shows the type 2 routes, as learned via BGP, so all guests are included in this view.

The following properties are shown for entries in the zone:

- Destination: The CIDR of the destination for this routing table entry
- Nexthops: The nexthops for this route, for vnets this is usually the local bridge - for externally learned routes (e.g. default routes) the IP of the next hop
- Protocol: The protocol via which this routes was learned
- Metric: The metric (or cost) of a route, lower cost routes are preferred over higher cost routes

The following properties are shown for entries in the vnet:

- IP Address: The IP-Address from the type-2 route
- MAC Address: The MAC-Address from the type-2 route
- via: The nexthop for the type-2 route

CHAPTER

FIVE

REMOTES

Proxmox Datacenter Manager allows you to add arbitrary Proxmox VE nodes or clusters and Proxmox Backup Server instances as remotes. This allows for a structured, unified overview of every host, VM, container, and datastore across different locations.

5.1 Resource Operation

Through the Proxmox Datacenter Manager, administrators can manage the lifecycle of virtual workloads at scale. Supported operations include starting, stopping, and rebooting guests across the inventory without the need to log in to individual nodes.

Additionally, the platform supports live migration of guests. This capability extends to migrations between independent clusters, facilitating load balancing and planned maintenance while maintaining high availability.

5.2 Data Collection

Collecting data like RRD metrics, worker task status, logs, and other operational information is a primary function of Proxmox Datacenter Manager. The system aggregates metrics to provide insight into usage, performance, and infrastructure growth.

This allows for introspection into the server fleet, providing a central overview but also allowing you to explore specific remotes or resources. Dashboards and RRD graphs visualize this data to assist in detecting trends, optimizing resource allocation, and planning future capacity.

5.3 Proxmox VE Remote

Proxmox VE remotes integrate virtualization clusters and independent nodes into the central management view. Once added, the interface displays the hierarchy of hosts, virtual machines, containers, and storage resources, searchable via the central interface.

Specific management capabilities available for Proxmox VE remotes include:

- **Update Management**: A centralized panel provides an overview of available updates across the infrastructure and allows for the rollout of patches directly from the Datacenter Manager interface.
- **SDN Capabilities**: Administrators can configure EVPN zones and VNets across multiple remotes to manage network overlays and administrative tasks.

5.4 Proxmox Backup Server Remote

Proxmox Backup Server instances can be managed as remotes to oversee backup infrastructure alongside virtualization hosts. The interface provides a consolidated overview of different datastores, displaying content and storage utilization.

Metrics from Proxmox Backup Server remotes are integrated directly into the central dashboard widgets, including RRD graphs for performance and usage monitoring.

5.5 Connection and Certificate Troubleshooting

Proxmox Datacenter Manager validates a remote's TLS certificate against the system certificate store. If the remote presents a publicly trusted certificate, for example one issued by Let's Encrypt through ACME, no further trust configuration is needed and certificate renewals are handled transparently.

When a remote's certificate is not in the system trust store, as with the default self-signed Proxmox certificates, Proxmox Datacenter Manager instead pins the fingerprint you accepted when adding the remote. If such a remote later renews or rotates its certificate, the pinned fingerprint no longer matches the presented one and every connection to that remote fails. The web interface and the command-line tools surface this as an error such as:

```
connection failed: Could not establish a TLS connection. Check whether the fingerprint matches
or the certificate on the remote is valid. OpenSSL Error: error:0A000086:SSL routines:
tls_post_process_server_certificate:certificate verify failed
```

The most common cause is a legitimate certificate renewal on the remote. It can also indicate an expired or otherwise invalid certificate or, if the change is unexpected, a man-in-the-middle attack, so confirm the new certificate through a trusted channel before accepting it.

To recover, re-probe the certificate that the remote currently presents and update the stored fingerprint:

- In the web interface, open the affected remote and use the **Check Certificate** action. It contacts the node, shows the certificate presented now, and lets you update the pinned fingerprint once you have confirmed the change. If the remote now uses a certificate that the system trust store accepts, you can instead clear the stored fingerprint to rely on that trust.
- On the command line, inspect the presented certificate with `proxmox-datacenter-manager-client remote probe-certificate <remote> <node>`, then store the verified value with `proxmox-datacenter-manager-client remote set-fingerprint <remote> <node> <fingerprint>` (omit the fingerprint to clear the pin).

The system journal on the Proxmox Datacenter Manager host records additional detail, including the fingerprint that the remote presented and the one that was expected.

CHAPTER

SIX

AUTOMATED INSTALLATIONS

The Proxmox Datacenter Manager provides integration with the automated installer for all Proxmox products.

A detailed documentation of all available options can be found on our dedicated wiki page.

6.1 Overview

The overview shows all past and ongoing installations done using the Proxmox Datacenter Manager. It allows access to the raw system information data as sent by the automated installer before the actual installation, and (if configured) post-installation notification hook data, containing extensive information about the newly installed system.

6.2 Prepared Answers

This view provides an overview over all defined answer files and allows editing, copying into new answers and deleting them. For a quick overview, it shows whether an answer is the default and what target filters have been defined for that particular configuration.

6.2.1 Target filter

Target filter allow you to control what systems should match.

Filters are key-value pairs in the format `key=format`, with keys being JSON Pointers, and match systems based the identifying information sent by the installer as JSON document. An example of such a document is provided on the wiki.

JSON Pointers allow for identifying specific values within a JSON document. For example, to match only Proxmox VE installations by the product name, a filter entry like `/product/product=pve` can be used.

Values are *globs* and use the same syntax as the automated installer itself. The following special characters can be used in filters:

- ? -- matches any single character
- * -- matches any number of characters, can be none
- `[a]`, `[abc]`, `[0-9]` -- matches any single character inside the brackets, ranges are possible
- `[!a]` -- negate the filter, any single character but the ones specified

A prepared answer can be also set as default, in which case it will be used if no other more specific answer matches based on its configured target filters.

6.2.2 Templating

Certain fields support templating via MiniJinja (a Jinja2-inspired templating engine) and sequential *counters*. Counters are automatically incremented each time an answer file is served to a client, allowing for easy provisioning of unique fields, such as per-system hostnames.

The following counter is automatically defined when creating a new prepared answer configuration:

- `installation_nr` - Counter of the number of installations done with this particular answer configuration.

This mechanism allows templating on the following fields for prepared answer configurations:

- **Administrator email address**
- **Hostname/FQDN**
- **Network IP address (CIDR)**
- **Network gateway**
- **DNS Server address**

The templating context provided for each field contains the system information data as sent by the automated installer on answer retrieval, as well as all template counters.

For example, to provide a unique hostname to each target system, the following template can be used for the **Hostname/FQDN** field:

```
{{ product.product }}{{ installation_nr }}.example.com
```

MiniJinja features a wide range of built-in filters, which are enabled by default, similar to Jinja2.

6.3 Authentication token management

To use the automated installer integration of Proxmox Datacenter Manager, an installation process must authenticate itself. This also provides for an additional scoping mechanism for prepared answer configurations.

The automated installer integration uses a dedicated token mechanism, separate from the normal API tokens. See the example under *Preparing an ISO* on how to include such a token in the ISO when preparing it.

6.4 Preparing an ISO

To use an installation ISO of a Proxmox product with the Proxmox Datacenter Manager functionality, the ISO must be appropriately prepared to fetch an answer via HTTP from the Proxmox Datacenter Manager using the `proxmox-auto-install-assistant` tool, available from the Proxmox VE package repositories.

The target URL for the automated installer must point to `https://<pdm>/api2/json/auto-install/answer`, where `<pdm>` is the address under which the Proxmox Datacenter Manager is reachable from the systems to be installed.

For example:

```
proxmox-auto-install-assistant prepare-iso /path/to/source.iso \
  --fetch-from http \
  --url 'https://datacenter.example.com/api2/json/auto-install/answer' \
  --cert-fingerprint
↪'ab:cd:ef:12:34:56:78:90:a1:b2:c3:d4:e5:f6:7a:8b:9c:0d:aa:bb:cc:dd:ee:ff:21:43:65:87:09:af:bd:ce
↪' \
  --answer-auth-token 'mytoken:ee2a5901-1910-4eb0-b0a2-c914f4adbb75'
```

CHAPTER

SEVEN

VIEWS

Views allow you to add an interactive view on a selected set of resources.

7.1 Resource Selection

The resource selection is controlled by an include-exclude filter system.

You define what resources to consider for including which then get passed through an exclude list to single specific types out again.

This way you can, for example, easily configure to include all virtual machine resources, but then exclude any such VM that resides on a specific remote.

7.1.1 Filter Types

The following lists of filter types are available to be used in include or exclude lists.

- The *resource-type* filter allows you to filter by a specific resource type. The following types are available:
  - *datastore*: A Proxmox Backup Server datastore.
  - *lxc*: A LXC container.
  - *node*: A Proxmox VE or Proxmox Backup Server node.
  - *qemu*: A QEMU virtual machine.
  - *sdn-zone*: A SDN zone.
  - *storage*: A Proxmox VE storage
- The *resource-pool* filter allows you to include or exclude only resources that are located in a specific resource pool-name.
- The *tag* filter allows you to filter resources that are tagged with a specific tag-name.
- The *remote* filter allows you to filter resources located on a specific remote.
- The *resource-id* filter allows you to filter resources with a specific ID.

Each filter can be prefixed with an optional *<match-behavior>:* prefix. Currently there is only the *exact* matching behavior available. This behavior is the default if no prefix is provided.

7.2 Customizable Dashboard

You can create customizable dashboards for a views from a set of pre-defined widgets. Only resources matching your include minus the ones matching your exclude filters will be displayed in these widgets.

7.2.1 Widgets

The following widgets are available:

- The *nodes* widget shows a status overview of the Proxmox VE and Proxmox Backup Server nodes, and can be limited to a single remote type.
- The *guests* widget shows a status overview of the virtual guests, and can be limited to QEMU virtual machines or LXC containers.
- The *pbs-datastores* widget shows the usage and status of Proxmox Backup Server datastores.
- The *remotes* widget lists the configured remotes and their status. It can also show a wizard for adding a new remote.
- The *subscription* widget shows the subscription status of the remotes.
- The *sdn* widget shows the status of the Software-Defined Networking (SDN) zones.
- The *leaderboard* widget ranks resources by a metric, such as guest or node CPU or node memory usage, and lists the top consumers.
- The *task-summary* widget summarizes recent tasks, grouped by a chosen criterion.
- The *resource-tree* widget shows the selected resources in a hierarchical tree.
- The *node-resource-gauge* widget displays a single node resource, such as CPU, memory, or storage, as a gauge chart, and can be limited to a single remote type.
- The *map* widget plots the remotes on an interactive world map at their configured geographic location. Markers are colored by the remote's status, and markers that are close together are clustered as you zoom out.

7.2.2 Map Widget

The locations shown by the *map* widget come from the remotes and cannot be edited in Proxmox Datacenter Manager. Configure them on the remote side: on Proxmox VE in the datacenter configuration as a cluster-wide default, optionally overridden per node, and on Proxmox Backup Server in the node configuration.

The map background is drawn from public-domain vector data by Natural Earth, shipped in the *proxmox-geojson-data* package.

7.3 Access Control

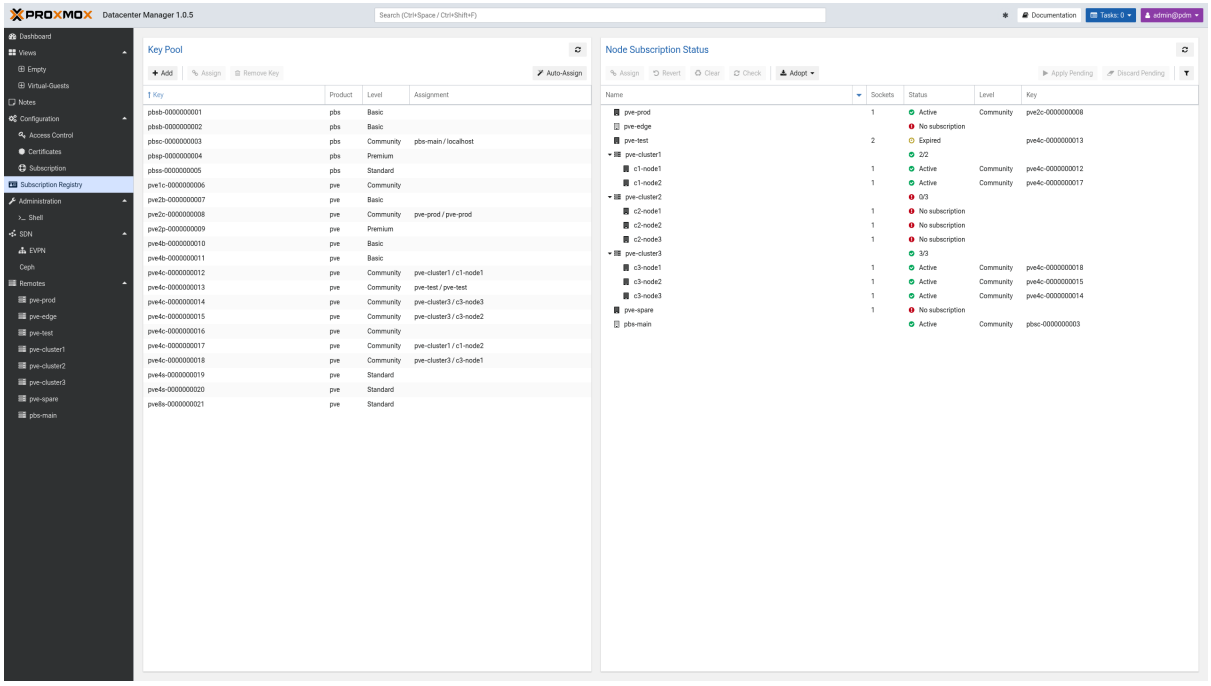
You can grant permissions on specific views. With such a permission the user can operate on the view and all its selected resources.

CHAPTER

EIGHT

SUBSCRIPTION REGISTRY

The subscription registry maintains a central pool of Proxmox VE and Proxmox Backup Server subscription keys and lets an administrator assign them to remote nodes from a single place, without having to select and configure a key for all remote nodes individually.



8.1 Key Pool

The pool accepts Proxmox VE and Proxmox Backup Server keys; other key prefixes are rejected so that a new product type is noticed instead of silently parking unusable entries. Each entry records its origin and the optional remote node it has been assigned to.

Keys can be added in bulk from the web interface or with the `proxmox-datacenter-client subscriptions add-keys` command. The Add dialog takes multiple keys, separated by newlines or commas, and validates the whole batch atomically.

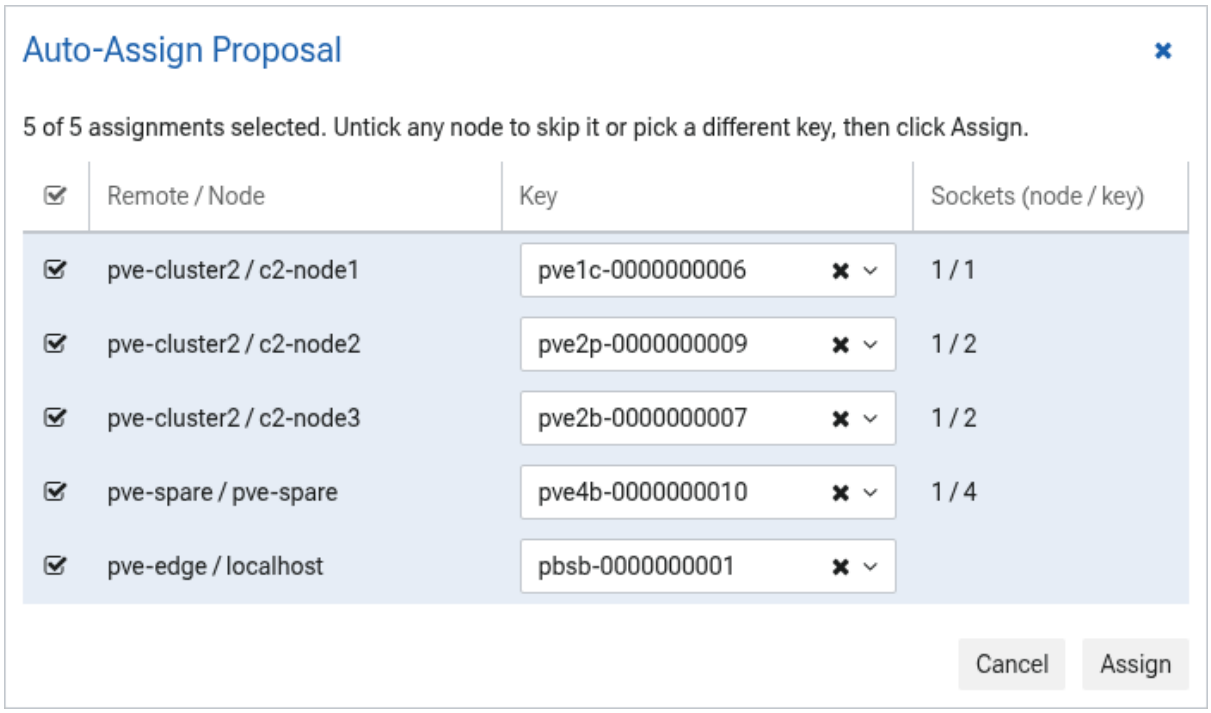
8.2 Node Subscription Status

The Node Subscription Status panel shows the live subscription state of every node behind a configured remote alongside any pending plan from the pool. Nodes that already hold a key the registry assigned appear with the live level; nodes with a pending pool assignment show a clock icon until the change is pushed to the remote. The filter button narrows the list by subscription status, or by a free-text remote or node name search.

From this view an operator can revert a pending change on the selected node (an unpushed assignment or a queued Clear Key) or queue a new Clear Key. Clear Key frees the live subscription key from a node so it can be reassigned elsewhere. The action is queued until it is committed via Apply Pending or reverted on a per-node basis.

8.3 Assignment and Clearing

A key can be pinned to a single node manually.



The Auto-Assign action proposes a plan that fills unsubscribed nodes from free pool keys. For Proxmox VE, the smallest covering key by socket count is chosen, so a 4-socket key is not used on a 2-socket host while a larger host stays unsubscribed.

The Clear Key action queues the live subscription on the selected node for removal. The action requires the (remote, node) to already be tracked by the pool. Apply Pending later issues the removal on the remote and releases the pool binding so the key becomes available for reassignment. Discard Pending drops the queued clear without touching the remote; the binding stays intact and the operator can retry.

The Adopt Key action imports the live subscription on a remote node into the pool as a bound entry, without touching the remote. Use it to bring a pre-existing subscription -- one installed on a node before PDM took over its pool management -- under the registry so that pool actions such as Clear Key and Auto-Assign can act on it. Nodes that are eligible for adoption are highlighted with a download hint icon in the Node Subscription Status tree; the pool grid carries a hidden-by-default Source column distinguishing manually-added from adopted entries, which can be enabled via the column picker if the distinction matters.

The Adopt All action runs the same import across every remote the operator can audit in one transaction. Use it after first connecting an existing fleet of nodes to PDM so the pool catches up with the live subscriptions already deployed, without having to click through Adopt Key for each node. Candidates the operator has no modify privilege on, whose key is already bound elsewhere in the pool, whose (remote, node) target is already bound by another pool entry, or whose key or node name fails schema validation are skipped silently.

The proposed plan can be reviewed and adjusted before it is applied: untick a row to skip that node, or pick a different free key for a row from its key dropdown. Apply Pending walks the queue in order and attempts every entry; any that fail are reported and stay pending, so one unreachable

node does not strand the rest and a later Apply Pending retries only the failures. Discard Pending drops the plan without touching any remote.

The Check Subscription action triggers a fresh shop-side verification of the live subscription on the selected node, equivalent to the per-product ”Check” button on PVE / PBS. Useful for promoting a stale `Invalid` or `Expired` verdict to `Active` once the underlying issue is fixed at the shop, without having to wait for the next periodic check. The Status column tooltip surfaces the last-checked timestamp and the next-due-date as reported by the remote, where available.

8.4 Permissions

Listing the pool and the node status view follows the regular audit privileges on each affected remote. Pool entries pinned to a remote the operator has no audit privilege on are hidden from the listing; unbound entries stay visible to anyone with the system-scope audit privilege.

Adding or removing pool entries requires the system-scope MODIFY privilege. Any action that drives a change on a remote, such as assigning or clearing a key, adopting a live subscription, or applying the pending plan, additionally requires the matching resource privilege on that remote, so an operator with global system access alone cannot drive changes against remotes they have no other authority on. Auto-Assign skips remotes the caller cannot modify, so a previewed plan never silently commits an assignment on a remote the operator only had audit on.

CHAPTER

NINE

AUTHENTICATION & ACCESS CONTROL

9.1 User Configuration

Proxmox Datacenter Manager supports several authentication realms, and you need to choose the realm when you add a new user. Possible realms are:

pam
: Linux PAM standard authentication. Use this if you want to authenticate as a Linux system user. The users needs to already exist on the host system.

pdm
: Proxmox Datacenter Manager realm. This type stores hashed passwords in `/etc/proxmox-datacenter-manager/access/shadow.json`.

openid
: OpenID Connect server. Users can authenticate against an external OpenID Connect server.

ldap
: LDAP server. Users can authenticate against external LDAP servers.

ad
: Active Directory server. Users can authenticate against external Active Directory servers.

The *root@pam* superuser has full administration rights on everything, so it's recommended to add other users with less privileges.

9.2 API Tokens

Any authenticated user can generate API tokens, which can in turn be used to configure various clients, instead of directly providing the username and password.

API tokens serve two purposes:

1. Easy revocation in case client gets compromised
2. Limit permissions for each client/token within the users' permission

An API token consists of two parts: an identifier consisting of the user name, the realm and a tokenname (`user@realm!tokenname`), and a secret value. Both need to be provided to the client in place of the user ID (`user@realm`) and the user password, respectively.

The API token is passed from the client to the server by setting the `Authorization` HTTP header with method `PDMAPIToken` to the value `TOKENID:TOKENSECRET`.

9.3 Access Control

By default, new users and API tokens do not have any permissions. Instead you need to specify what is allowed and what is not.

Proxmox Datacenter Manager uses a role- and path-based permission management system. An entry in the permissions table allows a user, group or token to take on a specific role when accessing an 'object' or 'path'. This means that such an access rule can be represented as a triple of '(path, user, role)', '(path, group, role)' or '(path, token, role)', with the role containing a set of allowed actions, and the path representing the target of these actions.

9.3.1 Privileges

Privileges are the building blocks of access roles. They are internally used to enforce the actual permission checks in the API.

System.Audit
: Allows knowing about the system and its status.

System.Modify
: Allows modifying system-level configuration.

Sys.Console
: Allows access to the system's console

Sys.PowerManagement
: Allows powering off or rebooting the system.

Resource.Audit
: Allows auditing guests, storages and other resources.

Resource.Manage
: Allows managing resources, like starting or stopping guests.

Resource.Modify
: Allows modifying resources, like making configuration changes.

Resource.Create
: Allows creating a guest.

Resource.Delete
: Allows deleting a guest.

Resource.Migrate
: Allows remote migration of a guest.

Access.Audit
: Allows auditing permissions and users.

Access.Modify
: Allows modifying permissions and users.

Realm.Allocate
: Allows viewing, creating, modifying and deleting realms

9.3.2 Access Roles

An access role combines one or more privileges into something that can be assigned to a user or API token on an object path.

Currently, there are only built-in roles, meaning you cannot create your own, custom role.

The following roles exist:

NoAccess
: Disable Access - nothing is allowed.

Administrator
: Can do anything, on the object path assigned.

Auditor
: Can view the status and configuration of things, but is not allowed to change settings.

9.3.3 Objects and Paths

Access permissions are assigned to objects, such as a datastore, namespace or some system resources.

We use filesystem-like paths to address these objects. These paths form a natural tree, and permissions of higher levels (shorter paths) can optionally be propagated down within this hierarchy.

Paths can be templated, meaning they can refer to the actual id of a configuration entry. When an API call requires permissions on a templated path, the path may contain references to parameters of the API call. These references are specified in curly brackets.

Some examples are:

`/resource`	Access to *all* resources managed by a Proxmox Datacenter Manager.
`/resource/{id}`	Access to resources on a specific remote.
`/resource/{id}/guest`	Access to *all* virtual guest resources on a specific remote.
`/resource/{id}/guest/{vmid}`	Access to a specific virtual guest on a specific remote.
`/resource/{id}/node`	Access to *all* nodes resources on a specific remote.
`/resource/{id}/node/{name}`	Access to a specific node on a specific remote.
`/views/`	Access to views.
`/views/{id}`	Access to a specific view.
`/system/network`	Access to configure the host network.
`/access/users`	User administration.
`/access/domains`	Administrative access to realms.

Inheritance

As mentioned earlier, object paths form a file system like tree, and permissions can be inherited by objects down that tree through the propagate flag, which is set by default. We use the following inheritance rules:

- Permissions for API tokens are always limited to those of the user.
- Permissions on deeper, more specific levels replace those inherited from an upper level.

9.3.4 Configuration & Management

Access permission information is stored in `/etc/proxmox-datacenter-manager/access/acl.cfg`. The file contains 5 fields, separated using a colon (':') as a delimiter. A typical entry takes the form:

```
acl:1:/datastore:john@pdm:Administrator
```

The data represented in each field is as follows:

1. `acl` identifier
2. A `1` or `0`, representing whether propagation is enabled or disabled, respectively
3. The object on which the permission is set. This can be a specific object (like a single view) or a top level object, which with propagation enabled, represents all children of the object also.

4. The user(s)/token(s) for which the permission is set
5. The role being set

You can manage permissions via **Configuration -> Access Control -> Permissions** in the web interface.

9.3.5 API Token Permissions

API token permissions are calculated based on ACLs containing their ID, independently of those of their corresponding user. The resulting permission set on a given path is then intersected with that of the corresponding user.

In practice this means:

1. API tokens require their own ACL entries
2. API tokens can never do more than their corresponding user

9.4 Two-Factor Authentication

9.4.1 Introduction

With simple authentication, only a password (single factor) is required to successfully claim an identity (authenticate), for example, to be able to log in as *root@pam* on a specific instance of Proxmox Datacenter Manager. In this case, if the password gets leaked or stolen, anybody can use it to log in - even if they should not be allowed to do so.

With two-factor authentication (TFA), a user is asked for an additional factor to verify their authenticity. Rather than relying on something only the user knows (a password), this extra factor requires something only the user has, for example, a piece of hardware (security key) or a secret saved on the user's smartphone. This prevents a remote user from gaining unauthorized access to an account, as even if they have the password, they will not have access to the physical object (second factor).

9.4.2 Available Second Factors

You can set up multiple second factors, in order to avoid a situation in which losing your smartphone or security key locks you out of your account permanently.

Proxmox Datacenter Manager supports three different two-factor authentication methods:

- TOTP (Time-based One-Time Password). A short code derived from a shared secret and the current time, it changes every 30 seconds.
- WebAuthn (Web Authentication). A general standard for authentication. It is implemented by various security devices, like hardware keys or trusted platform modules (TPM) from a computer or smart phone.
- Single use Recovery Keys. A list of keys which should either be printed out and locked in a secure place or saved digitally in an electronic vault. Each key can be used only once. These are perfect for ensuring that you are not locked out, even if all of your other second factors are lost or corrupt.

CHAPTER

TEN

HOST SYSTEM ADMINISTRATION

Proxmox Datacenter Manager is based on the famous Debian Linux distribution. This means that you have access to the entire range of Debian packages, and that the base system is well documented. The Debian Administrator's Handbook is available online, and provides a comprehensive introduction to the Debian operating system.

A standard Proxmox Datacenter Manager installation uses the default repositories from Debian, so you get bug fixes and security updates through that channel. In addition, we provide our own package repository to roll out all Proxmox related packages. This includes updates to some Debian packages when necessary.

We also deliver a specially optimized Linux kernel, based on the Ubuntu kernel. This kernel includes drivers for ZFS.

The following sections will explain things which are different on Proxmox Datacenter Manager, or tasks which are commonly used on Proxmox Datacenter Manager. For other topics, please refer to the standard Debian documentation.

10.1 ZFS on Linux

ZFS is a combined file system and logical volume manager, designed by Sun Microsystems. There is no need to manually compile ZFS modules - all packages are included.

By using ZFS, it's possible to achieve maximum enterprise features with low budget hardware, and also high performance systems by leveraging SSD caching or even SSD only setups. ZFS can replace expensive hardware raid cards with moderate CPU and memory load, combined with easy management.

General advantages of ZFS:

- Easy configuration and management with GUI and CLI.
- Reliable
- Protection against data corruption
- Data compression on file system level
- Snapshots
- Copy-on-write clone
- Various raid levels: RAID0, RAID1, RAID10, RAIDZ-1, RAIDZ-2 and RAIDZ-3
- Can use SSD for cache
- Self healing
- Continuous integrity checking
- Designed for high storage capacities
- Asynchronous replication over network

- Open Source
- Encryption

10.1.1 Hardware

ZFS depends heavily on memory, so it's recommended to have at least 8GB to start. In practice, use as much you can get for your hardware/budget. To prevent data corruption, we recommend the use of high quality ECC RAM.

If you use a dedicated cache and/or log disk, you should use an enterprise class SSD (for example, Intel SSD DC S3700 Series). This can increase the overall performance significantly.

Important

Do not use ZFS on top of a hardware controller which has its own cache management. ZFS needs to directly communicate with disks. An HBA adapter or something like an LSI controller flashed in `IT` mode is recommended.

10.1.2 ZFS Administration

This section gives you some usage examples for common tasks. ZFS itself is really powerful and provides many options. The main commands to manage ZFS are `zfs` and `zpool`. Both commands come with extensive manual pages, which can be read with:

```
# man zpool
# man zfs
```

Create a new zpool

To create a new pool, at least one disk is needed. The `ashift` should have the same sector-size (2 power of `ashift`) or larger as the underlying disk.

```
# zpool create -f -o ashift=12 <pool> <device>
```

Create a new pool with RAID-0

Minimum 1 disk

```
# zpool create -f -o ashift=12 <pool> <device1> <device2>
```

Create a new pool with RAID-1

Minimum 2 disks

```
# zpool create -f -o ashift=12 <pool> mirror <device1> <device2>
```

Create a new pool with RAID-10

Minimum 4 disks

```
# zpool create -f -o ashift=12 <pool> mirror <device1> <device2> mirror <device3> <device4>
```

Create a new pool with RAIDZ-1

Minimum 3 disks

```
# zpool create -f -o ashift=12 <pool> raidz1 <device1> <device2> <device3>
```

Create a new pool with RAIDZ-2

Minimum 4 disks

```
# zpool create -f -o ashift=12 <pool> raidz2 <device1> <device2> <device3> <device4>
```

Create a new pool with cache (L2ARC)

It is possible to use a dedicated cache drive partition to increase the read performance (use SSDs).

For `<device>`, you can use multiple devices, as is shown in "Create a new pool with RAID*".

```
# zpool create -f -o ashift=12 <pool> <device> cache <cache_device>
```

Create a new pool with log (ZIL)

It is possible to use a dedicated cache drive partition to increase the write performance (use SSDs).

For `<device>`, you can use multiple devices, as is shown in "Create a new pool with RAID*".

```
# zpool create -f -o ashift=12 <pool> <device> log <log_device>
```

Add cache and log to an existing pool

You can add cache and log devices to a pool after its creation. In this example, we will use a single drive for both cache and log. First, you need to create 2 partitions on the SSD with `parted` or `gdisk`

> **Important**
>
> Always use GPT partition tables.

The maximum size of a log device should be about half the size of physical memory, so this is usually quite small. The rest of the SSD can be used as cache.

```
# zpool add -f <pool> log <device-part1> cache <device-part2>
```

Changing a failed device

```
# zpool replace -f <pool> <old device> <new device>
```

Changing a failed bootable device

Depending on how Proxmox Datacenter Manager was installed, it is either using `grub` or `systemd-boot` as a bootloader.

In either case, the first steps of copying the partition table, reissuing GUIDs and replacing the ZFS partition are the same. To make the system bootable from the new disk, different steps are needed which depend on the bootloader in use.

```
# sgdisk <healthy bootable device> -R <new device>
# sgdisk -G <new device>
# zpool replace -f <pool> <old zfs partition> <new zfs partition>
```

Note

Use the *zpool status -v* command to monitor how far the resilvering process of the new disk has progressed.

With `systemd-boot`:

```
# proxmox-boot-tool format <new ESP> # proxmox-boot-tool init <new ESP>
```

Note

ESP stands for EFI System Partition, which is setup as partition #2 on bootable disks by the Proxmox Datacenter Manager installer. For details, see *Setting up a new partition for use as synced ESP*.

With `grub`:

Usually `grub.cfg` is located in `/boot/grub/grub.cfg`

```
# grub-install <new disk> # grub-mkconfig -o /path/to/grub.cfg
```

Activate e-mail notification

ZFS comes with an event daemon, ZED, which monitors events generated by the ZFS kernel module. The daemon can also send emails upon ZFS events, such as pool errors. Newer ZFS packages ship the daemon in a separate package `zfs-zed`, which should already be installed by default in Proxmox Datacenter Manager.

You can configure the daemon via the file `/etc/zfs/zed.d/zed.rc`, using your preferred editor. The required setting for email notification is ZED_EMAIL_ADDR, which is set to `root` by default.

```
ZED_EMAIL_ADDR="root"
```

Please note that Proxmox Datacenter Manager forwards mails to *root* to the email address configured for the root user.

Limit ZFS memory usage

It is good to use at most 50 percent (which is the default) of the system memory for ZFS ARC, to prevent performance degradation of the host. Use your preferred editor to change the configuration in `/etc/modprobe.d/zfs.conf` and insert:

```
options zfs zfs_arc_max=8589934592
```

The above example limits the usage to 8 GiB ('8 * 2^30^').

Important

In case your desired `zfs_arc_max` value is lower than or equal to `zfs_arc_min` (which defaults to 1/32 of the system memory), `zfs_arc_max` will be ignored. Thus, for it to work in this case, you must set `zfs_arc_min` to at most `zfs_arc_max - 1`. This would require updating the configuration in `/etc/modprobe.d/zfs.conf`, with:

```
options zfs zfs_arc_min=8589934591
options zfs zfs_arc_max=8589934592
```

This example setting limits the usage to 8 GiB ('8 * 2^30^') on systems with more than 256 GiB of total memory, where simply setting `zfs_arc_max` alone would not work.

Important

If your root file system is ZFS, you must update your initramfs every time this value changes.

```
# update-initramfs -u
```

Swap on ZFS

Swap-space created on a zvol may cause some issues, such as blocking the server or generating a high IO load.

We strongly recommend using enough memory, so that you normally do not run into low memory situations. Should you need or want to add swap, it is preferred to create a partition on a physical disk and use it as a swap device. You can leave some space free for this purpose in the advanced options of the installer. Additionally, you can lower the *swappiness* value. A good value for servers is 10:

```
# sysctl -w vm.swappiness=10
```

To make the swappiness persistent, create a new file `/etc/sysctl.d/99-swappiness.conf` with an editor of your choice and add the following line:

```
vm.swappiness = 10
```

Table 1: Linux kernel *swappiness* parameter values

Value	Strategy
vm.swappiness = 0	The kernel will swap only to avoid an 'out of memory' condition
vm.swappiness = 1	Minimum amount of swapping without disabling it entirely.
vm.swappiness = 10	Sometimes recommended to improve performance when sufficient memory exists in a system.
vm.swappiness = 60	The default value.
vm.swappiness = 100	The kernel will swap aggressively.

ZFS compression

To activate compression:

```
# zpool set compression=lz4 <pool>
```

We recommend using the `lz4` algorithm, since it adds very little CPU overhead. Other algorithms such as `lzjb`, `zstd` and `gzip-N` (where `N` is an integer from `1-9` representing the compression ratio, where 1 is fastest and 9 is best compression) are also available. Depending on the algorithm and how compressible the data is, having compression enabled can even increase I/O performance.

You can disable compression at any time with:

```
# zfs set compression=off <dataset>
```

Only new blocks will be affected by this change.

ZFS special device

Since version 0.8.0, ZFS supports *special* devices. A *special* device in a pool is used to store metadata, deduplication tables, and optionally small file blocks.

A *special* device can improve the speed of a pool consisting of slow spinning hard disks with a lot of metadata changes. For example, workloads that involve creating, updating or deleting a large number of files will benefit from the presence of a *special* device. ZFS datasets can also be configured to store small files on the *special* device, which can further improve the performance. Use fast SSDs for the *special* device.

Important

The redundancy of the *special* device should match the one of the pool, since the *special* device is a point of failure for the entire pool.

Warning

Adding a *special* device to a pool cannot be undone!

To create a pool with *special* device and RAID-1:

```
# zpool create -f -o ashift=12 <pool> mirror <device1> <device2> special mirror <device3>
↪<device4>
```

Adding a *special* device to an existing pool with RAID-1:

```
# zpool add <pool> special mirror <device1> <device2>
```

ZFS datasets expose the `special_small_blocks=<size>` property. `size` can be `0` to disable storing small file blocks on the *special* device, or a power of two in the range between 512B to 128K. After setting this property, new file blocks smaller than *size* will be allocated on the *special* device.

Important

If the value for `special_small_blocks` is greater than or equal to the `recordsize` (default 128K) of the dataset, *all* data will be written to the *special* device, so be careful!

Setting the `special_small_blocks` property on a pool will change the default value of that property for all child ZFS datasets (for example, all containers in the pool will opt in for small file blocks).

Opt in for all files smaller than 4K-blocks pool-wide:

```
# zfs set special_small_blocks=4K <pool>
```

Opt in for small file blocks for a single dataset:

```
# zfs set special_small_blocks=4K <pool>/<filesystem>
```

Opt out from small file blocks for a single dataset:

```
# zfs set special_small_blocks=0 <pool>/<filesystem>
```

Troubleshooting

Corrupt cache file

zfs-import-cache.service imports ZFS pools using the ZFS cache file. If this file becomes corrupted, the service won't be able to import the pools that it's unable to read from it.

As a result, in case of a corrupted ZFS cache file, some volumes may not be mounted during boot and must be mounted manually later.

For each pool, run:

```
# zpool set cachefile=/etc/zfs/zpool.cache POOLNAME
```

then, update the *initramfs* by running:

```
# update-initramfs -u -k all
```

and finally, reboot the node.

Another workaround to this problem is enabling the `zfs-import-scan.service`, which searches and imports pools via device scanning (usually slower).

10.2 Host Bootloader

Proxmox Datacenter Manager currently uses one of two bootloaders, depending on the disk setup selected in the installer.

For EFI Systems installed with ZFS as the root filesystem `systemd-boot` is used, unless Secure Boot is enabled. All other deployments use the standard `grub` bootloader (this usually also applies to systems which are installed on top of Debian).

10.2.1 Partitioning Scheme Used by the Installer

The Proxmox Datacenter Manager installer creates 3 partitions on all disks selected for installation.

The created partitions are:

- A 1 MB BIOS Boot Partition (gdisk type EF02)
- A 1 GB EFI System Partition (ESP, gdisk type EF00)
- A third partition spanning the configured `hdsize` parameter or the remaining space available for the chosen storage type

Systems using ZFS as a root filesystem are booted with a kernel and initrd image stored on the 1 GB EFI System Partition. For legacy BIOS systems, and EFI systems with Secure Boot enabled, `grub` is used, for EFI systems without Secure Boot, `systemd-boot` is used. Both are installed and configured to point to the ESPs.

`grub` in BIOS mode (`--target i386-pc`) is installed onto the BIOS Boot Partition of all selected disks on all systems booted with `grub` (that is, all installs with root on `ext4` or `xfs`, and installs with root on ZFS on non-EFI systems).

10.2.2 Synchronizing the Content of the ESP with `proxmox-boot-tool`

`proxmox-boot-tool` is a utility used to keep the contents of the EFI System Partitions properly configured and synchronized. It copies certain kernel versions to all ESPs and configures the respective bootloader to boot from the `vfat` formatted ESPs. In the context of ZFS as root filesystem, this means that you can use all the optional features on your root pool, instead of the subset which is also present in the ZFS implementation in `grub` or having to create a small, separate boot-pool (see: Booting ZFS on root with grub).

In setups with redundancy, all disks are partitioned with an ESP by the installer. This ensures the system boots, even if the first boot device fails or if the BIOS can only boot from a particular disk.

The ESPs are not kept mounted during regular operation. This helps to prevent filesystem corruption in the `vfat` formatted ESPs in case of a system crash, and removes the need to manually adapt `/etc/fstab` in case the primary boot device fails.

`proxmox-boot-tool` handles the following tasks:

- Formatting and setting up a new partition
- Copying and configuring new kernel images and initrd images to all listed ESPs
- Synchronizing the configuration on kernel upgrades and other maintenance tasks
- Managing the list of kernel versions which are synchronized
- Configuring the boot-loader to boot a particular kernel version (pinning)

You can view the currently configured ESPs and their state by running:

```
# proxmox-boot-tool status
```

Setting up a New Partition for use as Synced ESP

To format and initialize a partition as synced ESP, for example, after replacing a failed vdev in an rpool, `proxmox-boot-tool` from `proxmox-kernel-helper` can be used.

Warning

the `format` command will format the `<partition>`. Make sure to pass in the right device/partition!

For example, to format an empty partition `/dev/sda2` as ESP, run the following:

```
# proxmox-boot-tool format /dev/sda2
```

To setup an existing, unmounted ESP located on `/dev/sda2` for inclusion in Proxmox Datacenter Manager's kernel update synchronization mechanism, use the following:

```
# proxmox-boot-tool init /dev/sda2
```

or

```
# proxmox-boot-tool init /dev/sda2 grub
```

to force initialization with Grub instead of systemd-boot, for example for Secure Boot support.

Following this, *'/etc/kernel/proxmox-boot-uuids'* should contain a new line with the UUID of the newly added partition. The `init` command will also automatically trigger a refresh of all configured ESPs.

Updating the Configuration on all ESPs

To copy and configure all bootable kernels and keep all ESPs listed in `/etc/kernel/proxmox-boot-uuids` in sync, you just need to run:

```
# proxmox-boot-tool refresh
```

(Equivalent to running `update-grub` on systems with `ext4` or `xfs` on root).

This is necessary after making changes to the kernel commandline, or if you want to sync all kernels and initrds.

Note

Both `update-initramfs` and `apt` (when necessary) will automatically trigger a refresh.

Kernel Versions Considered by `proxmox-boot-tool`

The following kernel versions are configured by default:

- The currently running kernel
- The version being newly installed on package updates
- The two latest, already installed kernels
- The latest version of the second-to-last kernel series (e.g. 5.0, 5.3), if applicable
- Any manually selected kernels

Manually Keeping a Kernel Bootable

Should you wish to add a certain kernel and initrd image to the list of bootable kernels, use `proxmox-boot-tool kernel add`.

For example, run the following to add the kernel with ABI version `6.14.1-1-pve` to the list of kernels to keep installed and synced to all ESPs:

```
# proxmox-boot-tool kernel add 6.14.1-1-pve
```

`proxmox-boot-tool kernel list` will list all kernel versions currently selected for booting:

```
# proxmox-boot-tool kernel list
Manually selected kernels:
6.14.1-1-pve
Automatically selected kernels:
6.17.2-2-pve
```

Run `proxmox-boot-tool kernel remove` to remove a kernel from the list of manually selected kernels, for example:

```
# proxmox-boot-tool kernel remove 6.14.1-1-pve
```

Note

It's required to run `proxmox-boot-tool refresh` to update all EFI System Partitions (ESPs) after a manual kernel addition or removal from above.

10.2.3 Determine which Bootloader is Used

The simplest and most reliable way to determine which bootloader is used, is to watch the boot process of the Proxmox Datacenter Manager node.

You will either see the blue box of `grub` or the simple black on white `systemd-boot`.

Determining the bootloader from a running system might not be 100% accurate. The most reliable way is to run the following command:

```
# efibootmgr -v
```

If it returns a message that EFI variables are not supported, `grub` is used in BIOS/Legacy mode.

If the output contains a line that looks similar to the following, `grub` is used in UEFI mode.

```
Boot0005* proxmox     [...] File(\EFI\proxmox\grubx64.efi)
```

If the output contains a line similar to the following, `systemd-boot` is used.

```
Boot0006* Linux Boot Manager  [...] File(\EFI\systemd\systemd-bootx64.efi)
```

By running the following command, you can find out if `proxmox-boot-tool` is configured, which is a good indication of how the system is booted:

```
# proxmox-boot-tool status
```

10.2.4 Grub

`grub` has been the de facto standard for booting Linux systems for many years and is quite well documented (see the Grub Manual).

Configuration

Changes to the `grub` configuration are done via the defaults file `/etc/default/grub` or via config snippets in `/etc/default/grub.d`. To regenerate the configuration file after a change to the configuration, run:

```
# update-grub
```

Note

Systems using `proxmox-boot-tool` will call `proxmox-boot-tool refresh` upon `update-grub`

10.2.5 Systemd-boot

`systemd-boot` is a lightweight EFI bootloader. It reads the kernel and initrd images directly from the EFI Service Partition (ESP) where it is installed. The main advantage of directly loading the kernel from the ESP is that it does not need to reimplement the drivers for accessing the storage. In Proxmox Datacenter Manager, *proxmox-boot-tool* is used to keep the configuration on the ESPs synchronized.

Configuration

`systemd-boot` is configured via the file `loader/loader.conf` in the root directory of an EFI System Partition (ESP). See the `loader.conf(5)` manpage for details.

Each bootloader entry is placed in a file of its own, in the directory `loader/entries/`

An example entry.conf looks like this (`/` refers to the root of the ESP):

```
title    Proxmox
version  5.0.15-1-pve
options   root=ZFS=rpool/ROOT/pve-1 boot=zfs
linux    /EFI/proxmox/5.0.15-1-pve/vmlinuz-5.0.15-1-pve
initrd   /EFI/proxmox/5.0.15-1-pve/initrd.img-5.0.15-1-pve
```

10.2.6 Editing the Kernel Commandline

You can modify the kernel commandline in the following places, depending on the bootloader used:

Grub

The kernel commandline needs to be placed in the variable `GRUB_CMDLINE_LINUX_DEFAULT` in the file `/etc/default/grub`. Running `update-grub` appends its content to all `linux` entries in `/boot/grub/grub.cfg`.

systemd-boot

The kernel commandline needs to be placed as one line in `/etc/kernel/cmdline`. To apply your changes, run `proxmox-boot-tool refresh`, which sets it as the `option` line for all config files in `loader/entries/proxmox-*.conf`.

10.2.7 Override the Kernel-Version for next Boot

To select a kernel that is not currently the default kernel, you can either:

- Use the boot loader menu that is displayed at the beginning of the boot process
- Use the `proxmox-boot-tool` to `pin` the system to a kernel version either once or permanently (until pin is reset).

This should help you work around incompatibilities between a newer kernel version and the hardware.

Note

Such a pin should be removed as soon as possible, so that all recent security patches from the latest kernel are also applied to the system.

For example, to permanently select the version `5.15.30-1-pve` for booting, you would run:

```
# proxmox-boot-tool kernel pin 5.15.30-1-pve
```

Tip

The pinning functionality works for all Proxmox Datacenter Manager systems, not only those using `proxmox-boot-tool` to synchronize the contents of the ESPs, if your system does not use `proxmox-boot-tool` for synchronizing, you can also skip the `proxmox-boot-tool refresh` call in the end.

You can also set a kernel version to be booted on the next system boot only. This is useful, for example, to test if an updated kernel has resolved an issue, which caused you to `pin` a version in the first place:

```
# proxmox-boot-tool kernel pin 5.15.30-1-pve --next-boot
```

To remove any pinned version configuration, use the `unpin` subcommand:

```
# proxmox-boot-tool kernel unpin
```

While `unpin` has a `--next-boot` option as well, it is used to clear a pinned version set with `--next-boot`. As that happens already automatically on boot, invoking it manually is of little use.

After setting or clearing pinned versions, you also need to synchronize the content and configuration on the ESPs by running the `refresh` subcommand.

> **Tip**
>
> You will be prompted to automatically do for `proxmox-boot-tool` managed systems if you call the tool interactively.

```
# proxmox-boot-tool refresh
```

10.2.8 Secure Boot

Proxmox Datacenter Manager supports Secure Boot out of the box via signed packages and integration in `proxmox-boot-tool`.

The following packages need to be installed for Secure Boot to be enabled:

- `shim-signed` (shim bootloader signed by Microsoft)
- `shim-helpers-amd64-signed` (fallback bootloader and MOKManager, signed by Proxmox)
- `grub-efi-amd64-signed` (Grub EFI bootloader, signed by Proxmox)
- `proxmox-kernel-6.X.Y-Z-pve-signed` (Kernel image, signed by Proxmox)

Only Grub as bootloader is supported out of the box, since there are no other pre-signed bootloader packages available. Any new installation of Proxmox Datacenter Manager will automatically have all of the above packages included.

More details about how Secure Boot works, and how to customize the setup, are available in our wiki.

Switching an Existing Installation to Secure Boot

> **Warning**
>
> This can lead to an unbootable installation in some cases if not done correctly. Reinstalling the host will setup Secure Boot automatically if available, without any extra interactions. **Make sure you have a working and well-tested backup of your Proxmox Datacenter Manager host!**

An existing UEFI installation can be switched over to Secure Boot if desired, without having to reinstall Proxmox Datacenter Manager from scratch.

First, ensure all your system is up-to-date. Next, install all the required pre-signed packages as listed above. Grub automatically creates the needed EFI boot entry for booting via the default shim.

systemd-boot

If `systemd-boot` is used as a bootloader (see *Determine which Bootloader is used*), some additional setup is needed. This is only the case if Proxmox Datacenter Manager was installed with ZFS-on-root.

To check the latter, run:

```
# findmnt /
```

If the host is indeed using ZFS as root filesystem, the FSTYPE column should contain `zfs`:

```
TARGET SOURCE           FSTYPE OPTIONS
/      rpool/ROOT/pdm-1 zfs    rw,relatime,xattr,noacl
```

Next, a suitable potential ESP (EFI system partition) must be found. This can be done using the `lsblk` command as following:

```
# lsblk -o +FSTYPE
```

The output should look something like this:

```
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS FSTYPE
sda      8:0    0   32G  0 disk
├─sda1   8:1    0 1007K  0 part
├─sda2   8:2    0    1G  0 part             vfat
└─sda3   8:3    0   31G  0 part             zfs_member
sdb      8:16   0   32G  0 disk
├─sdb1   8:17   0 1007K  0 part
├─sdb2   8:18   0    1G  0 part             vfat
└─sdb3   8:19   0   31G  0 part             zfs_member
```

In this case, the partitions `sda2` and `sdb2` are the targets. They can be identified by the their size of 1G and their FSTYPE being `vfat`, in this case on a ZFS RAID-1 installation.

These partitions must be properly set up for booting through Grub using `proxmox-boot-tool`. This command (using `sda2` as an example) must be run separately for each individual ESP:

```
# proxmox-boot-tool init /dev/sda2 grub
```

Afterwards, you can sanity-check the setup by running the following command:

```
# efibootmgr -v
```

This list should contain an entry looking similar to this:

```
[..]
Boot0009* proxmox       HD(2,GPT,..,0x800,0x100000)/File(\EFI\proxmox\shimx64.efi)
[..]
```

Note

The old `systemd-boot` bootloader will be kept, but Grub will be preferred. This way, if booting using Grub in Secure Boot mode does not work for any reason, the system can still be booted using `systemd-boot` with Secure Boot turned off.

Now the host can be rebooted and Secure Boot enabled in the UEFI firmware setup utility.

On reboot, a new entry named `proxmox` should be selectable in the UEFI firmware boot menu, which boots using the pre-signed EFI shim.

If, for any reason, no `proxmox` entry can be found in the UEFI boot menu, you can try adding it manually (if supported by the firmware), by adding the file `\EFI\proxmox\shimx64.efi` as a custom boot entry.

Note

Some UEFI firmwares are known to drop the `proxmox` boot option on reboot. This can happen if the `proxmox` boot entry is pointing to a Grub installation on a disk, where the disk itself is not a boot option. If possible, try adding the disk as a boot option in the UEFI firmware setup utility and run `proxmox-boot-tool` again.

Tip

To enroll custom keys, see the accompanying Secure Boot wiki page.

Using DKMS/Third Party Modules With Secure Boot

On systems with Secure Boot enabled, the kernel will refuse to load modules which are not signed by a trusted key. The default set of modules shipped with the kernel packages is signed with an ephemeral key embedded in the kernel image which is trusted by that specific version of the kernel image.

In order to load other modules, such as those built with DKMS or manually, they need to be signed with a key trusted by the Secure Boot stack. The easiest way to achieve this is to enroll them as Machine Owner Key (MOK) with `mokutil`.

The `dkms` tool will automatically generate a keypair and certificate in `/var/lib/dkms/mok.key` and `/var/lib/dkms/mok.pub` and use it for signing the kernel modules it builds and installs.

You can view the certificate contents with

```
# openssl x509 -in /var/lib/dkms/mok.pub -noout -text
```

and enroll it on your system using the following command:

```
# mokutil --import /var/lib/dkms/mok.pub
input password:
input password again:
```

The `mokutil` command will ask for a (temporary) password twice, this password needs to be entered one more time in the next step of the process! Rebooting the system should automatically boot into the `MOKManager` EFI binary, which allows you to verify the key/certificate and confirm the enrollment using the password selected when starting the enrollment using `mokutil`. Afterwards, the kernel should allow loading modules built with DKMS (which are signed with the enrolled MOK). The MOK can also be used to sign custom EFI binaries and kernel images if desired.

The same procedure can also be used for custom/third-party modules not managed with DKMS, but the key/certificate generation and signing steps need to be done manually in that case.

10.3 Certificate Management

Access to the API and thus the web-based administration interface is always encrypted through `https`. Each Proxmox Datacenter Manager host creates by default its own (self-signed) certificate. This certificate is used for encrypted communication with the host's `proxmox-datacenter-api` service, for any API call between a user or backup-client and the web-interface.

Certificate verification when sending backups to a Proxmox Datacenter Manager is either done based on pinning the certificate fingerprints in the storage/remote configuration, or by using certificates, signed by a trusted certificate authority.

10.3.1 Certificates for the API and SMTP

Proxmox Datacenter Manager stores its certificate and key in:

- `/etc/proxmox-datacenter-manager/auth/api.pem`
- `/etc/proxmox-datacenter-manager/auth/api.key`

You have the following options for the certificate:

1. Keep using the default self-signed certificate in `/etc/proxmox-datacenter-manager/auth/api.pem`.
2. Use an externally provided certificate (for example, signed by a commercial Certificate Authority (CA)).
3. Use an ACME provider like Let's Encrypt to get a trusted certificate with automatic renewal; this is also integrated in the Proxmox Datacenter Manager API and web interface.

Certificates are managed through the Proxmox Datacenter Manager web-interface or API.

10.3.2 Upload Custom Certificate

If you already have a certificate which you want to use for a Proxmox Datacenter Manager host, you can simply upload that certificate over the web interface.

Note that any certificate key files must not be password protected.

10.3.3 Trusted certificates via Let's Encrypt (ACME)

Proxmox Datacenter Manager includes an implementation of the **A**utomatic **C**ertificate **M**anagement **E**nvironment (**ACME**) protocol, allowing Proxmox Datacenter Manager admins to use an ACME provider like Let's Encrypt for easy setup of TLS certificates, which are accepted and trusted by modern operating systems and web browsers out of the box.

Currently, the two ACME endpoints implemented are the Let's Encrypt (LE) production and staging environments. Our ACME client supports validation of `http-01` challenges using a built-in web server and validation of `dns-01` challenges using a DNS plugin supporting all the DNS API endpoints acme.sh does.

ACME Account

You need to register an ACME account per cluster, with the endpoint you want to use. The email address used for that account will serve as the contact point for renewal-due or similar notifications from the ACME endpoint.

You can register or deactivate ACME accounts over the web interface `Certificates -> ACME Accounts`.

> **Tip**
>
> Because of rate-limits you should use LE `staging` for experiments or if you use ACME for the very first time until all is working there, and only then switch over to the production directory.

ACME Plugins

The ACME plugin's role is to provide automatic verification that you, and thus the Proxmox Datacenter Manager Server under your operation, are the real owner of a domain. This is the basic building block of automatic certificate management.

The ACME protocol specifies different types of challenges, for example the `http-01`, where a web server provides a file with a specific token to prove that it controls a domain. Sometimes this isn't possible, either because of technical limitations or if the address of a record is not reachable from the public internet. The `dns-01` challenge can be used in such cases. This challenge is fulfilled by creating a certain DNS record in the domain's zone.

Proxmox Datacenter Manager supports both of those challenge types out of the box, you can configure plugins either over the web interface under `Certificates -> ACME Challenges`.

ACME Plugin configurations are stored in `/etc/proxmox-datacenter-manager/acme/plugins.cfg`.

Domains

You can add new or manage existing domain entries under `Certificates`.

After configuring the desired domain(s) for a node and ensuring that the desired ACME account is selected, you can order your new certificate over the web-interface. On success, the interface will reload after roughly 10 seconds.

Renewal will happen *automatically*

10.3.4 ACME HTTP Challenge Plugin

There is always an implicitly configured `standalone` plugin for validating `http-01` challenges via the built-in web server spawned on port 80.

> **Note**
>
> The name `standalone` means that it can provide the validation on its own, without any third party service.

There are a few prerequisites to use this for certificate management with Let's Encrypts ACME.

- You have to accept the ToS of Let's Encrypt to register an account.
- **Port 80** of the node needs to be reachable from the internet.
- There **must** be no other listener on port 80.
- The requested (sub)domain needs to resolve to a public IP of the Proxmox Datacenter Manager host.

10.3.5 ACME DNS API Challenge Plugin

On systems where external access for validation via the `http-01` method is not possible or desired, it is possible to use the `dns-01` validation method. This validation method requires a DNS server that allows provisioning of TXT records via an API.

Configuring ACME DNS APIs for validation

Proxmox Datacenter Manager reuses the DNS plugins developed for the `acme.sh`[1] project. Please refer to its documentation for details on configuration of specific APIs.

The easiest way to configure a new plugin with the DNS API is using the web interface (`Certificates -> ACME Accounts/Challenges`).

Here you can add a new challenge plugin by selecting your API provider and entering the credential data to access your account over their API.

> **Tip**
>
> See the acme.sh How to use DNS API wiki for more detailed information about getting API credentials for your provider. Configuration values do not need to be quoted with single or double quotes; for some plugins that is even an error.

As there are many DNS providers and API endpoints, Proxmox Datacenter Manager automatically generates the form for the credentials, but not all providers are annotated yet. For those you will see a bigger text area, into which you simply need to copy all the credential's KEY=VALUE pairs.

DNS Validation through CNAME Alias

A special `alias` mode can be used to handle validation on a different domain/DNS server, in case your primary/real DNS does not support provisioning via an API. Manually set up a permanent `CNAME` record for `_acme-challenge.domain1.example` pointing to `_acme-challenge.domain2.example`, and set the `alias` property in the Proxmox Datacenter Manager node configuration file `/etc/proxmox-datacenter-manager/acme/domains.cfg` to `domain2.example` to allow the DNS server of `domain2.example` to validate all challenges for `domain1.example`.

[1] acme.sh https://github.com/acmesh-official/acme.sh

Wildcard Certificates

Wildcard DNS names start with a `*.` prefix and are considered valid for all (one-level) subdomain names of the verified domain. So a certificate for `*.domain.example` is valid for `foo.domain.example` and `bar.domain.example`, but not for `baz.foo.domain.example`.

Currently, you can only create wildcard certificates with the DNS challenge type.

Combination of Plugins

Combining `http-01` and `dns-01` validation is possible in case your node is reachable via multiple domains with different requirements / DNS provisioning capabilities. Mixing DNS APIs from multiple providers or instances is also possible by specifying different plugin instances per domain.

> **Tip**
>
> Accessing the same service over multiple domains increases complexity and should be avoided if possible.

10.3.6 Automatic renewal of ACME certificates

If a node has been successfully configured with an ACME-provided certificate, the certificate will be renewed automatically by the `proxmox-datacenter-manager-daily-update.service`. Currently, renewal is triggered if the certificate either has already expired or if it will expire in the next 30 days.

10.3.7 Manually Change Certificate over the Command Line

If you want to get rid of certificate verification warnings, you have to generate a valid certificate for your server.

Log in to your Proxmox Datacenter Manager via ssh or use the console:

```
openssl req -newkey rsa:2048 -nodes -keyout key.pem -out req.pem
```

Follow the instructions on the screen, for example:

```
Country Name (2 letter code) [AU]: AT
State or Province Name (full name) [Some-State]:Vienna
Locality Name (eg, city) []:Vienna
Organization Name (eg, company) [Internet Widgets Pty Ltd]: Proxmox GmbH
Organizational Unit Name (eg, section) []:Proxmox Datacenter Manager
Common Name (eg, YOUR name) []: yourproxmox.yourdomain.com
Email Address []:support@yourdomain.com
Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []: not necessary
An optional company name []: not necessary
```

After you have finished the certificate request, you have to send the file `req.pem` to your Certification Authority (CA). The CA will issue the certificate (BASE64 encoded), based on your request – save this file as `cert.pem` to your Proxmox Datacenter Manager.

To activate the new certificate, do the following on your Proxmox Datacenter Manager

```
cp key.pem /etc/proxmox-datacenter-manager/auth/api.key
cp cert.pem /etc/proxmox-datacenter-manager/auth/api.pem
```

Then restart the API servers:

```
systemctl restart proxmox-datacenter-api.service
```

Test your new certificate, using your browser.

Note

To transfer files to and from your Proxmox Datacenter Manager, you can use secure copy: If your desktop runs Linux, you can use the `scp` command-line tool. If your desktop PC runs windows, please use an scp client like WinSCP (see https://winscp.net/).

10.4 Service Daemons

10.4.1 `proxmox-datacenter-api`

This daemon exposes the whole Proxmox Datacenter Manager API on TCP port 8443 using HTTPS. It runs as user `www-data` and has very limited permissions. Operations requiring more permissions are forwarded to the local `proxmox-datacenter-privileged-api` service.

10.4.2 `proxmox-datacenter-privileged-api`

This daemon exposes the Proxmox Datacenter Manager management API through a restricted UNIX socket at `/run/proxmox-datacenter-manager/priv.sock`. The daemon runs as `root` and has permission to do all privileged operations.

NOTE: The daemon listens to a local UNIX socket address only, so you cannot access it from outside. The `proxmox-datacenter-api` daemon exposes the API to the outside world.

10.5 Command-line Tools

10.5.1 `proxmox-datacenter-manager-client`

This tool implements a datacenter manager client, i.e. it can connect to any datacenter manager with the same major version and issue management commands, query status and control remotes and their resources.

10.5.2 `proxmox-datacenter-manager-admin`

This tool exposes some of the datacenter managers administrative API on the command line.

CHAPTER

ELEVEN

FAQ

11.1 What distribution is Proxmox Datacenter Manager (PDM) based on?

Proxmox Datacenter Manager is based on Debian GNU/Linux.

11.2 Will Proxmox Datacenter Manager run on a 32-bit processor?

Proxmox Datacenter Manager only supports 64-bit CPUs (AMD or Intel). There are no future plans to support 32-bit processors.

11.3 How long will my Proxmox Datacenter Manager version be supported?

Proxmox Datacenter Manager	Debian Version	First Release	Debian EOL	Proxmox Datacenter Manager EOL
Proxmox Datacenter Manager 1	Debian 13 (Trixie)	2025-12	TBA	TBA

11.4 How can I upgrade Proxmox Datacenter Manager to the next point release?

Minor version upgrades, for example upgrading from Proxmox Datacenter Manager in version 1.0 to 1.1 or 1.3, can be done just like any normal update.

But, you should still check the release notes for any relevant notable, or breaking change.

For the update itself use either the Web UI *Administration -> Updates* panel or through the CLI with:

```
apt update
apt full-upgrade
```

Note

Always ensure you correctly setup the *package repositories* and only continue with the actual upgrade if *apt update* did not hit any error.

11.5 Is there a dedicated subscription for the Proxmox Datacenter Manager?

No, there is not. However, your existing Basic or higher subscription for Proxmox VE and Proxmox Backup Server remotes includes access to the Proxmox Datacenter Manager Enterprise Repository and support at no extra cost.

11.6 How can I get Enterprise Support for the Proxmox Datacenter Manager?

Existing customers with active Basic or higher subscriptions for their Proxmox remotes also gain access to the Proxmox Datacenter Manager enterprise repository and support.

11.7 How can I get access to the Proxmox Datacenter Manager Enterprise Repository?

The Proxmox Datacenter Manager can use the enterprise repository if at least 80% of the configured remote nodes have a valid Basic or higher subscription.

APPENDIX

A

COMMAND SYNTAX

A.1 proxmox-datacenter-manager-admin

`proxmox-datacenter-manager-admin acme account deactivate <name> [OPTIONS]`

Deactivate an ACME account.

<name>
: [<string>] ACME account name.

Optional parameters:

--force <boolean> (default=true)
: Delete account data even if the server refuses to deactivate the account.

`proxmox-datacenter-manager-admin acme account info <name> [OPTIONS]`

Show ACME account information.

<name>
: [<string>] ACME account name.

Optional parameters:

--output-format text|json|json-pretty
: Output format.

`proxmox-datacenter-manager-admin acme account list [OPTIONS]`

List ACME accounts.

Optional parameters:

--output-format text|json|json-pretty
: Output format.

`proxmox-datacenter-manager-admin acme account register <name> <contact> [OPTIONS]`

Register an ACME account.

<name>
: [<string>] ACME account name.

<contact>
: [<string>] List of email addresses.

Optional parameters:

--directory <string>
The ACME Directory.

`proxmox-datacenter-manager-admin acme account update <name> [<contact>]`

Update an ACME Account.

<name>
[<string>] ACME account name.

<contact>
[<string>] List of email addresses.

`proxmox-datacenter-manager-admin acme certificate order [OPTIONS]`

Order a new ACME certificate.

Optional parameters:

--force <boolean> (default=false)
Force renewal even if the certificate does not expire soon.

`proxmox-datacenter-manager-admin acme certificate revoke`

Revoke ACME certificate.

`proxmox-datacenter-manager-admin acme plugin add <type> <id> --api <string> --data <string> [OPTIONS]`

Add ACME plugin configuration.

<type>
[<string>] The ACME challenge plugin type.

<id>
[<string>] ACME Challenge Plugin ID.

--api <string>
DNS API Plugin Id.

--data <string>
File containing the plugin data.

Optional parameters:

--disable <boolean> (default=false)
Flag to disable the config.

--validation-delay <integer> (0 - 172800) (default=30)
Extra delay in seconds to wait before requesting validation.

Allows to cope with long TTL of DNS records.

`proxmox-datacenter-manager-admin acme plugin config <id> [OPTIONS]`

Show ACME plugin information.

<id>
[<string>] Plugin ID

Optional parameters:

--output-format **text|json|json-pretty**
Output format.

`proxmox-datacenter-manager-admin acme plugin list [OPTIONS]`

List ACME plugins.

Optional parameters:

--output-format **text|json|json-pretty**
Output format.

`proxmox-datacenter-manager-admin acme plugin remove <id>`

Delete an ACME plugin configuration.

<id>
[<string>] ACME Challenge Plugin ID.

`proxmox-datacenter-manager-admin acme plugin set <id> [OPTIONS]`

Update an ACME plugin configuration.

<id>
[<string>] ACME Challenge Plugin ID.

Optional parameters:

--api **<string>**
DNS API Plugin Id.

--data **<string>**
DNS plugin data (base64 encoded with padding).

--delete **disable|validation-delay**
List of properties to delete. Can be specified more than once.

--digest **<string>**
Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

--disable **<boolean> (default=false)**
Flag to disable the config.

--validation-delay **<integer> (0 - 172800) (default=30)**
Extra delay in seconds to wait before requesting validation.

Allows to cope with long TTL of DNS records.

`proxmox-datacenter-manager-admin help [{<command>}] [OPTIONS]`

Get help about specified command (or sub-command).

<command>
[<string>] Command. This may be a list in order to specify nested sub-commands. Can be specified more than once.

Optional parameters:

--verbose **<boolean>**
Verbose help.

`proxmox-datacenter-manager-admin remote add --authid <string> --id <string> --nodes <string> --token <string> --type pve|pbs [OPTIONS]`

Add a new remote.

`--authid <string>`
Authentication ID

`--id <string>`
Remote ID.

`--nodes <string>`
A list of cluster node addresses. Can be specified more than once.

`--token <string>`
The access token's secret.

`--type pve|pbs (default=pve)`
The type of a remote entry.

Optional parameters:

`--web-url <string>`
Configuration for the Web UI URL link generation.

`proxmox-datacenter-manager-admin remote list [OPTIONS]`

List all the remotes this instance is managing.

Optional parameters:

`--output-format text|json|json-pretty`
Output format.

`proxmox-datacenter-manager-admin remote remove <id> [OPTIONS]`

Remove a remote.

`<id>`
[<string>] Remote ID.

Optional parameters:

`--delete-token <boolean> (default=true)`
If set to false, token deletion on the remote is skipped.

`proxmox-datacenter-manager-admin remote subscriptions [OPTIONS]`

Get the status of all the remotes this instance is managing.

Optional parameters:

`--output-format text|json|json-pretty`
Output format.

`proxmox-datacenter-manager-admin remote update <id> [OPTIONS]`

Update a remote.

`<id>`
[<string>] Remote ID.

Optional parameters:

--authid <string>
Authentication ID

--nodes <string>
A list of cluster node addresses. Can be specified more than once.

--token <string>
The access token's secret.

--web-url <string>
Configuration for the Web UI URL link generation.

`proxmox-datacenter-manager-admin remote version <id>`

Add a new remote.

<id>
[<string>] Remote ID.

`proxmox-datacenter-manager-admin report`

Generate the system report.

`proxmox-datacenter-manager-admin support-status get`

Get the support status information.

`proxmox-datacenter-manager-admin support-status update`

Update the support status information.

`proxmox-datacenter-manager-admin versions [OPTIONS]`

List package versions for important Proxmox Datacenter Manager packages.

Optional parameters:

--output-format text|json|json-pretty
Output format.

--verbose <boolean> (default=false)
Output verbose package information. It is ignored if output-format is specified.

A.2 proxmox-datacenter-manager-client

Options available for command group `proxmox-datacenter-manager-client`:

--color no|always|auto (default=auto)
Control terminal color output.

--output-format text|json|json-pretty
Output format.

--fingerprint <string>
Certificate fingerprint to expect.

--host <string>
Server to connect to, or *user@realm@host* triple.

--password-command <string>
Command to run to get the password.

--password-file <string>
File to read the password from.

--port <integer> (0 - 65535) (default=8443)
Port to connect.

--user <string>
User ID

`proxmox-datacenter-manager-client acl delete <path> <role> --auth-id <string> [OPTIONS]`

Delete an ACL entry.

<path>
[<string>] Access control path.

<role>
[<role>] Enum representing roles via their [PRIVILEGES] combination.

Since privileges are implemented as bitflags, each unique combination of privileges maps to a single, unique *u64* value that is used in this enum definition.

--auth-id <string>
Authentication ID

Optional parameters:

--digest <string>
Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client acl list [OPTIONS]`

List all users or show a single user's information.

Optional parameters:

--exact <boolean> (default=false)
If set, returns only ACL for the exact path.

--path <string>
Access control path.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client acl update <path> <role> --auth-id <string> [OPTIONS]`

Add or update an ACL entry.

<path>
: [<string>] Access control path.

<role>
: [<role>] Enum representing roles via their [PRIVILEGES] combination.

  Since privileges are implemented as bitflags, each unique combination of privileges maps to a single, unique *u64* value that is used in this enum definition.

--auth-id <string>
: Authentication ID

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

--propagate <boolean> (default=true)
: Allow to propagate (inherit) permissions.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client help [{<command>}] [OPTIONS]`

Get help about specified command (or sub-command).

<command>
: [<string>] Command. This may be a list in order to specify nested sub-commands. Can be specified more than once.

Optional parameters:

--verbose <boolean>
Verbose help.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client login

Log into a server.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client metric-collection status

Show metric collection status.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client metric-collection trigger [OPTIONS]

Trigger metric collection. If a remote is passed, only this remote will be collected, otherwise all.

Optional parameters:

--remote <string>
Remote ID.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pbs datastore list <remote>

List all the remotes this instance is managing.

<remote>
[<string>] Remote ID.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pbs datastore rrddata <remote> <datastore> <mode> <timeframe>

Return a PBS datastore's metric data.

<remote>
[<string>] Remote ID.

<datastore>
[<string>] Datastore name.

<mode>
[MAX|AVERAGE] RRD consolidation mode

<timeframe>
[hour|day|week|month|year|decade] RRD time frame

Inherited group parameters:

--color

--fingerprint

--host

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pbs node rrddata <remote> <mode> <timeframe>`

Return a PBS node's metric data.

<remote>
: [<string>] Remote ID.

<mode>
: [MAX|AVERAGE] RRD consolidation mode

<timeframe>
: [hour|day|week|month|year|decade] RRD time frame

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pbs snapshot list <remote> <datastore> [OPTIONS]`

List all the remotes this instance is managing.

<remote>
: [<string>] Remote ID.

<datastore>
: [<string>] Datastore name.

Optional parameters:

--ns <string>
: Namespace.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

--port

--user

---

proxmox-datacenter-manager-client pbs task list <remote>

List the tasks of a cluster.

<remote>
: [<string>] Remote ID.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pbs task status <remote> <upid>

Query the status of a task.

<remote>
: [<string>] Remote ID.

<upid>
: [<string>] The task UPID, optionally with the remote name prefix

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve lxc config <remote> <vmid> [OPTIONS]

Query the configuration of a container.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
Node name (or 'localhost')

--snapshot <string>
The name of the snapshot

--state pending|active (default=pending)
Guest configuration access.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve lxc list <remote> [OPTIONS]

List the LXC containers of a cluster.

<remote>
[<string>] Remote ID.

Optional parameters:

--node <string>
Node name (or 'localhost')

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve lxc migrate <remote> <vmid> <target> [OPTIONS]

Migrate a container to a different node in the same cluster.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<target>
[<string>] Remote ID.

Optional parameters:

--bwlimit <integer> (0 - N)
Override I/O bandwidth limit (in KiB/s).

--map-storage FROM:TO,...
Mapping of source storages to ones on the target cluster. Can be specified more than once.

--node <string>
Node name (or 'localhost')

--online <boolean>
Perform an online migration if the vm is running.

--restart <boolean> (default=false)
Perform a restart-migration.

--timeout <integer> (0 - N)
Add a shutdown timeout for the restart-migration.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

```
proxmox-datacenter-manager-client pve lxc remote-migrate <remote> <vmid>
<target> --map-bridge FROM:TO,... --map-storage FROM:TO,... [OPTIONS]
```

Migrate a container to a different cluster.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<target>
[<string>] Remote ID.

--map-bridge FROM:TO,...
Mapping of source network bridges to ones on the target cluster. Can be specified more than once.

--map-storage FROM:TO,...
Mapping of source storages to ones on the target cluster. Can be specified more than once.

Optional parameters:

--bwlimit <integer> (0 - N)
Override I/O bandwidth limit (in KiB/s).

--delete <boolean>
Delete the original VM and related data after successful migration.

--node <string>
Node name (or 'localhost')

--online <boolean>
Perform an online migration if the vm is running.

--restart <boolean> (default=false)
Perform a restart-migration.

--target-vmid <integer> (1 - N)
A guest ID

--timeout <integer> (0 - N)
Add a shutdown timeout for the restart-migration.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc rrddata <remote> <vmid> <mode> <timeframe>`

Return a CT's metric data.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<mode>
[MAX|AVERAGE] RRD consolidation mode

<timeframe>
[hour|day|week|month|year|decade] RRD time frame

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc shutdown <remote> <vmid> [OPTIONS]`

Shutdown a container.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
Node name (or 'localhost')

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve lxc snapshot create <remote> <vmid> <snapname> [OPTIONS]

Create a snapshot of an LXC guest.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<snapname>
[<string>] The name of the snapshot

Optional parameters:

--description <string>
A textual description or comment.

--node <string>
Node name (or 'localhost')

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

`proxmox-datacenter-manager-client pve lxc snapshot delete <remote> <vmid> <snapname> [OPTIONS]`

Delete a snapshot of an LXC guest.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc snapshot list <remote> <vmid> [OPTIONS]`

List the snapshots of an LXC guest.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc snapshot rollback <remote> <vmid> <snapname> [OPTIONS]`

Roll back an LXC guest to a snapshot (destructive: reverts disk and config to it).

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--node <string>
: Node name (or 'localhost')

--start <boolean>
: Start the container after a successful rollback.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc snapshot update-description <remote> <vmid> <snapname> [OPTIONS]`

Update an LXC snapshot's description.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--description <string>
: The new description; pass an empty string to clear.

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc start <remote> <vmid> [OPTIONS]`

Start a container.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve lxc stop <remote> <vmid> [OPTIONS]`

Stop a container abruptly.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve node list <remote>`

List all the nodes of a pve cluster.

<remote>
: [<string>] Remote ID.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve node rrddata <remote> <node> <mode> <timeframe>`

Return a PVE node's metric data.

<remote>
: [<string>] Remote ID.

<node>
: [<string>] Node name (or 'localhost')

<mode>
: [MAX|AVERAGE] RRD consolidation mode

<timeframe>
: [hour|day|week|month|year|decade] RRD time frame

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve qemu config <remote> <vmid> [OPTIONS]`

Query the configuration of a VM.

<remote>
: [<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
Node name (or 'localhost')

--snapshot <string>
The name of the snapshot

--state pending|active (default=pending)
Guest configuration access.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve qemu list <remote> [OPTIONS]

List the QEMU VMs of a cluster.

<remote>
[<string>] Remote ID.

Optional parameters:

--node <string>
Node name (or 'localhost')

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve qemu migrate <remote> <vmid> <target> [OPTIONS]

Migrate a VM to a different node of the same cluster.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<target>
[<string>] Remote ID.

Optional parameters:

--bwlimit <integer> (0 - N)
Override I/O bandwidth limit (in KiB/s).

--force <boolean>
Perform an online migration if the vm is running.

--map-storage FROM:TO,...
Mapping of source storages to ones on the target cluster. Can be specified more than once.

--migration-network <string>
CIDR of the (sub) network that is used for migration.

--migration-type secure|insecure
Migration traffic is encrypted using an SSH tunnel by default. On secure, completely private networks this can be disabled to increase performance.

--node <string>
Node name (or 'localhost')

--online <boolean>
Perform an online migration if the vm is running.

--with-local-disks <boolean>
Enable live storage migration for local disks.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

```
proxmox-datacenter-manager-client pve qemu remote-migrate <remote> <vmid>
<target> --map-bridge FROM:TO,... --map-storage FROM:TO,... [OPTIONS]
```

Migrate a VM to a different cluster.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<target>
[<string>] Remote ID.

--map-bridge FROM:TO,...
Mapping of source network bridges to ones on the target cluster. Can be specified more than once.

--map-storage FROM:TO,...
Mapping of source storages to ones on the target cluster. Can be specified more than once.

Optional parameters:

--bwlimit <integer> (0 - N)
Override I/O bandwidth limit (in KiB/s).

--delete <boolean>
Delete the original VM and related data after successful migration.

--node <string>
Node name (or 'localhost')

--online <boolean>
Perform an online migration if the vm is running.

--target-vmid <integer> (1 - N)
A guest ID

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve qemu rrddata <remote> <vmid> <mode> <timeframe>`

Return a VM's metric data.

<remote>
[<string>] Remote ID.

<vmid>
[<integer> (1 - N)] A guest ID

<mode>
[MAX|AVERAGE] RRD consolidation mode

<timeframe>
[hour|day|week|month|year|decade] RRD time frame

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve qemu shutdown <remote> <vmid> [OPTIONS]`

Shutdown a VM.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve qemu snapshot create <remote> <vmid> <snapname> [OPTIONS]`

Create a snapshot of a QEMU guest.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--description <string>
: A textual description or comment.

--node <string>
: Node name (or 'localhost')

--vmstate <boolean>
: Include the VM's RAM state.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve qemu snapshot delete <remote> <vmid> <snapname> [OPTIONS]

Delete a snapshot of a QEMU guest.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve qemu snapshot list <remote> <vmid> [OPTIONS]

List the snapshots of a QEMU guest.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve qemu snapshot rollback <remote> <vmid> <snapname> [OPTIONS]

Roll back a QEMU guest to a snapshot (destructive: reverts disk and config to it).

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--node <string>
: Node name (or 'localhost')

--start <boolean>
: Start the guest after a successful rollback.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client pve qemu snapshot update-description <remote> <vmid> <snapname> [OPTIONS]

Update a QEMU snapshot's description.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

<snapname>
: [<string>] The name of the snapshot

Optional parameters:

--description <string>
: The new description; pass an empty string to clear.

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve qemu start <remote> <vmid> [OPTIONS]`

Start a VM.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve qemu stop <remote> <vmid> [OPTIONS]`

Stop a VM abruptly.

<remote>
: [<string>] Remote ID.

<vmid>
: [<integer> (1 - N)] A guest ID

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve resources <remote> [<kind>]`

Query the cluster resources.

<remote>
: [<string>] Remote ID.

<kind>
: [vm|storage|node|sdn] Resource type.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve task list <remote> [OPTIONS]`

List the tasks of a cluster.

<remote>
: [<string>] Remote ID.

Optional parameters:

--node <string>
: Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client pve task status <remote> <upid>`

Query the status of a task.

<remote>
: [<string>] Remote ID.

<upid>
: [<string>] The task UPID, optionally with the remote name prefix

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client remote add <type> <id> --authid <string> --nodes <string> --token <string> [OPTIONS]`

Add a new remote.

<type>
: [pve|pbs (default=pve)] The type of a remote entry.

<id>
: [<string>] Remote ID.

--authid <string>
: Authentication ID

--nodes <string>
: A list of cluster node addresses. Can be specified more than once.

--token <string>
: The access token's secret.

Optional parameters:

--create-token <string>
: If given, create this token on the remote and use it.

--web-url <string>
: Configuration for the Web UI URL link generation.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client remote delete <id> [OPTIONS]

Delete a remote.

<id>
: [<string>] Remote ID.

Optional parameters:

--delete-token <boolean> (default=true)
: If set to false, token deletion on the remote is skipped.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client remote list

List all the remotes this instance is managing.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client remote probe-certificate <id> <node>

Re-probe a configured node's TLS certificate, ignoring the pinned fingerprint.

Useful to detect a rotated certificate. Apply the new fingerprint with *remote set-fingerprint <id> <node> <fingerprint>*, or clear the pin by omitting it.

<id>
: [<string>] Remote ID.

<node>
[<string>] Hostname of the configured node to probe.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client remote set-fingerprint <id> <node> [OPTIONS]

Set or clear a node's stored TLS certificate fingerprint.

Omit the fingerprint to clear the pin and rely on the system trust store. Run *remote probe-certificate* first to read the fingerprint the node currently presents.

<id>
[<string>] Remote ID.

<node>
[<string>] Hostname of the configured node whose fingerprint to set.

Optional parameters:

--fingerprint <string>
X509 certificate fingerprint (sha256).

Inherited group parameters:

--color

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client remote update <id> [OPTIONS]

Update a remote.

<id>
[<string>] Remote ID.

Optional parameters:

--authid <string>
Authentication ID

--delete <string>
List of properties to clear, e.g. 'group' or 'web-url'. Can be specified more than once.

--nodes <string>
A list of cluster node addresses. Can be specified more than once.

--token <string>
The access token's secret.

--web-url <string>
Configuration for the Web UI URL link generation.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client remote version <id>`

Show a remote's version.

<id>
[<string>] Remote ID.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client resources [OPTIONS]`

List all the remotes this instance is managing.

Optional parameters:

--max-age <integer> (0 - N)
Maximum age of cached remote resources.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions add-keys {<keys>} [OPTIONS]

Add one or more subscription keys to the pool.

<keys>
: [<string>] Subscription keys to add to the pool. Can be specified more than once.

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions adopt-all [OPTIONS]

Adopt every foreign live subscription into the pool in one transaction.

Walks all remotes the caller can audit, imports any (remote, node) with a live current key and no pool binding. Candidates the caller has no modify privilege on, or whose key is already bound elsewhere in the pool, are silently skipped.

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

`proxmox-datacenter-manager-client subscriptions adopt-key <remote> <node> [OPTIONS]`

Adopt the live subscription on a remote node into the pool.

Brings a foreign subscription under PDM management without touching the remote: the live current key on *remote*/*node* is imported as a pool entry bound to that node. Refuses if the (remote, node) target already has a pool-managed binding.

<remote>
: [<string>] Remote ID.

<node>
: [<string>] Node name (or 'localhost')

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client subscriptions apply-pending [OPTIONS]`

Push all pending key assignments to remotes as a worker task.

Blocks until the worker finishes so the operator sees the exit status of the actual push run, not just a UPID they would have to chase down by hand.

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client subscriptions assign-key <key> --node <string> --remote <string> [OPTIONS]`

Assign a key from the pool to a remote node.

<key>
: [<string>] Subscription key.

--node <string>
: Node name (or 'localhost')

--remote <string>
: Remote ID.

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client subscriptions auto-assign [OPTIONS]`

Propose (and optionally apply) automatic key-to-node assignments.

Optional parameters:

--apply <boolean> (default=false)
: Commit the proposal immediately via bulk-assign. Without this, only a preview is printed.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client subscriptions check <remote> <node>`

Trigger a fresh shop-side subscription check on a remote node.

Equivalent to the per-product ”Check” button: re-verifies the live subscription status against the shop. Useful for promoting a stale Invalid/Expired verdict to Active once the underlying issue is fixed at the shop, without waiting for the next periodic check.

<remote>
: [<string>] Remote ID.

<node>
: [<string>] Node name (or 'localhost')

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client subscriptions clear-assignment <key> [OPTIONS]`

Clear the assignment of a key (unassign from its remote node).

<key>
: [<string>] Subscription key.

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client subscriptions clear-key <remote> <node> [OPTIONS]`

Queue a Clear Key on a remote node so its subscription can be removed at next Apply Pending.

Refuses if no pool entry is bound to (remote, node): foreign live subscriptions must first be imported via the explicit Adopt Key action, never as a side effect of queueing a clear.

<remote>
: [<string>] Remote ID.

<node>
[<string>] Node name (or 'localhost')

Optional parameters:

--digest <string>
Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions clear-pending [OPTIONS]

Clear every pending assignment in one bulk transaction.

Optional parameters:

--digest <string>
Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions list-keys

List all subscription keys in the pool.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions remove-key <key>

Remove a key from the pool entirely.

<key>
: [<string>] Subscription key.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions revert-clear <remote> <node> [OPTIONS]

Drop a queued Clear Key on a remote node while keeping the pool binding.

<remote>
: [<string>] Remote ID.

<node>
: [<string>] Node name (or 'localhost')

Optional parameters:

--digest <string>
: Prevent changes if current configuration file has different SHA256 digest. This can be used to prevent concurrent modifications.

Inherited group parameters:

--color

--fingerprint

--host

--output-format

--password-command

--password-file

--port

--user

---

proxmox-datacenter-manager-client subscriptions status [OPTIONS]

Show the subscription status of all remotes.

Optional parameters:

--max-age <integer> (0 - N) (default=86400)
Maximum age (in seconds) of cached remote subscription state.

--verbose <boolean> (default=false)
If true, includes subscription information per node (with enough privileges)

--view <string>
View name.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user create <userid> [OPTIONS]`

List all users or show a single user's information.

<userid>
[<string>] User ID

Optional parameters:

--comment <string>
Comment.

--email <string>
E-Mail Address.

--enable <boolean> (default=true)
Enable the account (default). You can set this to '0' to disable the account.

--expire <integer> (0 - N) (default=0)
Account expiration date (seconds since epoch). '0' means no expiration date.

--firstname <string>
First name.

--lastname <string>
Last name.

--password <string>
Password.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user delete <userid>`

List all users or show a single user's information.

<userid>
[<string>] User ID

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user list`

List all users or show a single user's information.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user passwd <userid> [OPTIONS]`

Change a user's password. If no password is provided, it will be prompted for interactively.

<userid>
[<string>] User ID

Optional parameters:

--password <string>
Password.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user tfa add <type> <description> [OPTIONS]`

Add a TFA method to a user (currently only recovery keys).

`<type>`
: [`totp|u2f|webauthn|recovery|yubico`] A TFA entry type.

`<description>`
: [`<string>`] a description for the tfa entry

Optional parameters:

`--userid <string>`
: userid

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user tfa delete <id> [OPTIONS]`

Delete a TFA entry by id.

`<id>`
: [`<string>`] the tfa id to remove

Optional parameters:

`--userid <string>`
: userid

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user tfa list [OPTIONS]`

List all the remotes this instance is managing.

Optional parameters:

`--userid <string>`
: User ID

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

`--password-file`

`--port`

`--user`

`proxmox-datacenter-manager-client user update <userid> [OPTIONS]`

Change user information.

`<userid>`
: [<string>] User ID

Optional parameters:

`--comment <string>`
: Comment.

`--delete comment|enable|expire|firstname|lastname|email`
: Clear/reset user properties. Can be specified more than once.

`--email <string>`
: E-Mail Address.

`--enable <boolean> (default=true)`
: Enable the account (default). You can set this to '0' to disable the account.

`--expire <integer> (0 - N) (default=0)`
: Account expiration date (seconds since epoch). '0' means no expiration date.

`--firstname <string>`
: First name.

`--lastname <string>`
: Last name.

Inherited group parameters:

`--color`

`--fingerprint`

`--host`

`--output-format`

`--password-command`

```
--password-file
--port
--user
```

APPENDIX

B

CONFIGURATION FILES

The Proxmox Datacenter Manager configuration files are stored at `/etc/proxmox-datacenter-manager/` directory.

B.1 `remotes.cfg`

B.1.1 Options

Section type 'pbs': The information required to connect to a remote instance.

Required properties:

`authid`
: [<string>] Authentication ID

`nodes`
: [<string>] A list of cluster node addresses. Can be specified more than once.

`token`
: [<string>] The access token's secret.

`type`
: [pve|pbs (default=pve)] The type of a remote entry.

Optional properties:

`web-url`
: [<string>] Configuration for the Web UI URL link generation.

Section type 'pve': The information required to connect to a remote instance.

Required properties:

`authid`
: [<string>] Authentication ID

`nodes`
: [<string>] A list of cluster node addresses. Can be specified more than once.

`token`
: [<string>] The access token's secret.

`type`
: [pve|pbs (default=pve)] The type of a remote entry.

Optional properties:

`web-url`
: [<string>] Configuration for the Web UI URL link generation.

B.2 views.cfg

B.2.1 Options

Optional properties:

exclude
: [[exact:]resource-type=<storage|qemu|lxc|sdn-zone|datastore|node>|[exact:]resource-p
List of filter rules. Can be specified more than once.

include
: [[exact:]resource-type=<storage|qemu|lxc|sdn-zone|datastore|node>|[exact:]resource-p
List of filter rules. Can be specified more than once.

include-all
: [<boolean>] Include all resources by default.

layout
: [<string>] The configured layout, encoded as json

APPENDIX

C

ROADMAP

- Simplify adding remotes by implementing a remote-join information endpoint. This allows copying key data to provide the Proxmox Datacenter Manager with all initial information required to connect to a remote, ensuring trust and safety.
  - This concept is similar to the "Join Information" API and UI used in Proxmox VE clusters but functions independently, as Proxmox Datacenter Manager does not rely on cluster communication.
- Management of core configurations:
  - Backup jobs and their status.
  - Firewall management (building upon the existing visualization capabilities).
- Off-site replication of guests for manual recovery in case of datacenter failure.
- Evaluate an active-standby architecture for the Proxmox Datacenter Manager to avoid a single point of failure.
  - Currently, users can operate two instances, which results in doubled metric collection but minimal overhead otherwise.
- Integration of Proxmox Mail Gateway.
- Bulk actions, such as starting, stopping, or migrating multiple virtual guests at once.
- Implementation of a notification system:
  - Standard system notifications and update alerts.
  - Evaluation of Proxmox Datacenter Manager acting as a notification target for remotes.
- User Interface improvements:
  - Handling Multi-Factor Authentication (MFA) for the initial "Probe Remote" connection.
  - Evaluate a Pool View where hierarchical resource pools from all remotes are merged.
- Improvements for customizable Views:
  - Provide more card widgets that one can add, including ones that provide some direct control over the included resources.
  - Allow to add the pre-defined Updates, Firewall or Task tabs from the Remotes panel.
  - Allow to create tabs to organize complex views.
  - Evaluate other layouting options for rendering the card widgets.
  - Extending the filter capability by providing more types and evaluate more flexible comparisons operators.

Please note that this list outlines general goals and potential ideas rather than fixed promises. If you have a substantial use case you're willing to describe in detail, we encourage you to open enhancement requests for items listed here, as your feedback helps us prioritize work and understand specific needs.

C.1 Release History

C.1.1 Proxmox Datacenter Manager 1.0

Released 04. December 2025

- Based on Debian Trixie (13.2)
- Latest 6.17.2-2 Kernel as stable default
- ZFS 2.3.4

Features (Highlights)

- First stable release.
- Add Support for Proxmox Backup Server.

  Allows managing Proxmox Backup Server remotes similarly to Proxmox VE remotes.

  An overview shows the contents of different datastores alongside RRD graphs.

  The dashboard now includes metrics from Proxmox Backup Server remotes in its widgets.
- Custom Views allow creating custom overviews of all remotes.

  Views allow providing an overview similar to the dashboard but with custom layouts and filters.

  The data a view has access to can be filtered by remotes, resources, resource type, or tags.

  Users can be granted access to specific views without granting them access to the underlying remotes or resources directly.
- Add support for accessing a remote's shell.

  With the release of Proxmox VE 9.1 and Proxmox Backup Server 4.1, API tokens can now request shell access.

  Proxmox Datacenter Manager leverages this capability to allow accessing shells of supported remotes through one unified control plane.
- Global package repository and pending updates status.

  A new panel offers an overview of the status of all package repositories and available updates from remotes.

  Updates can be applied from within Proxmox Datacenter Manager by leveraging the new remote shell features.
- Improved authentication functionality allows easier user management.

  Proxmox Datacenter Manager now supports LDAP, Active Directory and OpenID Connect realms for authentication.

  Tokens allow granting more fine-grained access to other applications that want to use the API.

Changelog Overview

Enhancements in the Web Interface (GUI)

- Views allow for custom overviews of all or a specific set of remotes and resources.

A drag and drop editor allows easy adjustment of any widget.

The data that a view displays can be easily tweaked via filters.

The default dashboard is provided as an initial view.

- Add a panel for adding and managing new realms.

  LDAP, Active Directory, and OpenID connect realms can be added to allow easy authentication management.

  LDAP and Active Directory realms can also be synced using this panel.

  These realms can be configured as default realms. Default realms are used by the login mask by default instead of the PAM realm.

- Add a panel that allows managing tokens and allow configuring ACL entries for tokens.
- Enable the documentation button in the top navigation bar.
- Link to proper builtin documentation instead of Beta documentation.
- A new tab under the "Administration" menu shows the status of the Proxmox Datacenter Manager host and allows shutting it down or rebooting it (issue 6300).
- Add presentation of subscription status of remotes:
  - The remote subscription status can now be refreshed manually.
  - Remote subscriptions can now be inspected by clicking on the subscription status panel in the dashboard (issue 6797).
  - Add a "Details" button in the subscription panel to show the subscription status dialog.
- Tags of Proxmox VE guests are now shown in the resource tree.
- Add a panel displaying the notes of Proxmox VE nodes and datacenters.
- Align available functionality for Proxmox VE guests with the version of the remote.
- Allow the UI to render components based on the user's privileges.
- Remove a duplicate entry from the permission path selector.
- Improve Proxmox Backup Server datastore panel by making the labels translatable.
- Proxmox Backup Server remote tasks are handled correctly now.
- The remote setup wizard now validates the remote's ID.
- Add a title to the Proxmox VE remote tree toolbar.
- Remove unnecessary "enabled" status line for Proxmox VE storages.
- Do not show storage entries in the Proxmox VE resource tree unconditionally.
- Add a button to allow navigating to a Proxmox VE guest directly in their respective details views.
- Tabs for Proxmox VE and Proxmox Backup Server remotes now properly support history navigation.
- Add a window to display and copy the system report.
- A new panel shows the Proxmox Datacenter Manager's subscription information.
- When adding a remote via the setup wizard, the token name will now include the Proxmox Datacenter Manager host. This ensures multiple instances of Proxmox Datacenter Manager can be connected to the same remote.
- Mask remote shells if the remote version is too old to support the feature.
- Fix an issue that prevented realms from being deleted (issue 6885).
- Fix an issue where updating a storage's status did not trigger correctly.

- Fix an issue that prevented users in the PAM realm from being added as Proxmox Datacenter Manager users (issue 6787).
- The UI now properly respects the text direction for Arabic, Persian (Farsi) and Hebrew.
- Fix an issue where the resource tree for a search was not loaded correctly.
- Make navigating to network resources work properly again.
- Updated translations, among others:
  - Czech
  - French
  - Georgian
  - German
  - Italian
  - Japanese
  - Korean
  - Polish
  - Spanish
  - Swedish
  - Traditional Chinese
  - Ukrainian

Resource Management

- Remote shells for Proxmox VE and Proxmox Datacenter Manager can be accessed directly from the UI.

  Proxmox VE remotes make this shell available through a new tab in a node's details panel.

  For Proxmox Backup Server remotes, a button was added in the top bar of the overview to open a new window with the shell.
- A new panel shows hardware and options configuration for Proxmox VE remote's guests.
- Make search terms case-insensitive.
- Allow searching for resources by remote type.
- Extend matching to properties of resources.
- Views can now be searched for.

  Resources can specify a list of properties that can then be searched for.
- Add support for new Proxmox VE network resource type.
- Allow searching for resources by network type.
- Fix an issue that needlessly kept polling the API when users were logged out.

  This could trigger a bug where users were instantly logged out again after a fresh login.
- Show VMs and CTs overviews in a tab panel for Proxmox VE remotes.

Remotes Management

- Proxmox Backup Server remotes can now be added similarly to Proxmox VE remotes.

  A wizard can be used to add new Proxmox Backup Server remotes.

  This includes the ability to inspect the TLS certificate of the remote from within the wizard, enabling trust-on-first-use.

  An overview panel shows the status of a datastore, such as usage and I/O information, and its contents as a tree of backup snapshots.

  The content of datastores can be inspected, including namespaces and backup snapshots they contain.

  The dashboard has also been improved to include new functionality for Proxmox Backup Server remotes:

  - Proxmox Backup Server remotes can be added directly from the dashboard.
  - The status of all Proxmox Backup Server remotes can be inspected from a dedicated panel.
  - A new panel shows datastores and their statistics.
- Implement a view that displays a global overview of all available updates for all remotes.

  This includes version information as well as repository status information.
- Add an update panel for Proxmox Backup Server remotes.
- The subscription status endpoint now marks clusters with nodes that all have an unknown subscription status as ”unknown” instead of ”mixed” subscription status.
- Top entities now include Proxmox Backup Server remotes.
- Show more status information on Proxmox VE nodes in the node overview panel.

Firewall and Software Defined Network

- Add basic support to gather information on a Proxmox VE remote’s firewall setup.

  An overview panel shows which remote nodes and remote guests have an active firewall and how many rules are enabled.

  Detailed rules can be inspected by selecting an entity from the overview panel.
- The IP-VRF and MAC-VRF of a EVPN VNet can now be queried.
- Show the status of an IP-VRF and MAC-VRF in new panels in the EVPN panel.
- Show unknown zones if there are any.
- Show fabrics on Proxmox VE remotes in addition to zones.
- Show SDN zones with pending changes as status “pending” instead of “unknown”.

Backend Improvements

- Allow filtering API responses based on a `view` parameter.

  A view can filter the results of an API endpoint based on resource ID, resource pool, resource type, remote, and tags.

  By granting a user permissions to a view, users can query an API endpoint based on the view’s filter regardless of their own permissions.

  Currently, views can be used when listing resources, querying top entities, status of resources, subscription status of remotes, and remote tasks.

- Add endpoints that allow proxying a remote's shell via a web socket.
- Backend support for Proxmox Backup Server remotes:
  - Add TLS probing for Proxmox Backup Server remotes.
  - Allow scanning Proxmox Backup Server remotes analogous to Proxmox VE remotes.
  - Assign an ACL with admin role on "/" for newly created Proxmox Backup Server tokens when adding them as a remote.
  - Allow querying a Proxmox Backup Server's remote status.
  - Add a new API endpoint that returns the namespaces of a remote datastore.
  - Add API endpoints to query Proxmox Backup Server tasks.
  - Improve information collection on Proxmox Backup Server datastores by including configuration properties and more status types.
  - Support Proxmox Backup Server remote update information collection.
  - Request latest metrics for Proxmox Backup Server when using hourly timeframe.
  - Fix an issue where some Proxmox Backup Server remotes wrongly signaled HttpOnly cookie support, leading to an issue when querying them.
- Add an endpoint for listing Proxmox VE and Proxmox Backup Server remotes under `/pve/remotes` and `/pbs/remotes` respectively.
- Add an API endpoint for retrieving and refreshing the remote update summary.
- Cache results for remote update availability.
- Poll the remote update status via a periodic task.
- Implement LDAP and Active Directory realm support.
- Add support for OpenID Connect realms.
- When collecting the remote status, keep track of all remotes that collection has failed for.
- Allow non-root users to access several endpoints, such as:
  - Querying top entities (issue 6794).
  - Proxmox Backup Server RRD endpoints and overview (issue 6901).
  - Listing SDN controllers, VNets and zones for all configured Proxmox VE hosts (issue 6901).
- Improve permissions on the remote tasks endpoint.
- The node update summary now includes information for package version and repository status.
- Add an endpoint that allows querying remote APT repository status.
- Remove entries of a user in the ACL tree when the user is removed.
- Logs will now include the API path when an API call fails. Unknown errors will be logged too.
- Add endpoints for querying the Proxmox Datacenter Manager host's status and shutting it down or rebooting it.
- Fix an issue where only active tasks were included in the remote task list instead of all other tasks.
- Fix an issue that broke migration of remote guests.
- Improve documentation of API endpoints and their return type.
- Task, auth, and access logs will now be rotated.
- Split remote configuration and token storage into separate files.

- Add endpoints for querying the subscription status of Proxmox Datacenter Manager and connected Proxmox VE and Proxmox Backup Server remotes.
- New endpoints allows querying the configuration of a Proxmox VE node and cluster options.
- Add an API endpoint to get the cached version information of a remote.

Command Line Interface Enhancements

- The CLI client can now list the status and task list for Proxmox Backup Server remotes.
- The type of remote UPID can be inferred by the client instead of having to be explicitly specified.
- Add a command for getting all remote subscriptions to `proxmox-datacenter-manager-admin`.
- A new sub-command to show the subscription status of all remotes was added.
- Fix a bug that prevented the `proxmox-datacenter-manager-admin` to function as intended.

Documentation and Support for Troubleshooting

- Add initial Proxmox Datacenter Manager documentation.
- Add a system report to make supporting Proxmox Datacenter Manager setups easier.
- Include an API viewer.

Known Issues & Breaking Changes

- The API was restructured:
  - Endpoints under `/remotes/{id}` were moved to `/remotes/remote/{id}`.
  - API Endpoints for `remote-tasks`, `remote-update`, and `metrics-collection` were moved under `/remotes`.
- Some API endpoints will now correctly return 403 Forbidden error codes when a user has insufficient permissions instead of 401 Unauthorized.

  API users relying on the previous erroneous return code may break. Affected are the following endpoints:

  - `POST /api2/json/pve/remotes/remote/{remote}/lxc/{vmid}/remote-migrate`
  - `GET /api2/json/pve/remotes/remote/{remote}/resources`
  - `GET /api2/json/pve/remotes/remote/{remote}/lxc`
  - `GET /api2/json/pve/remotes/remote/{remote}/qemu`
  - `POST /api2/json/pve/remotes/remote/{remote}/qemu/{vmid}/remote-migrate`
  - `GET /api2/json/resources/list`
  - `GET /api2/json/resources/status`
- Some Alpha releases did not ship with the new HttpOnly authentication flow, API users that relied on it may need to adapt.

  Ideally new API users would be switched to use tokens wherever possible.
- A minimum password length of eight characters is now enforced on users of the “pdm” realm.

- Move the file storing the LDAP password from `/etc/proxmox-datacenter-manager/ldap_passwords.json` to `/etc/proxmox-datacenter-manager/access/ldap-passwords.json`

Proxmox Datacenter Manager 0.9 BETA

Released 11. September 2025

- Based on Debian Trixie (13)
- Latest 6.14.11-1 Kernel as stable default
- ZFS 2.3.4

Features (Highlights)

- New release based on the great Debian Trixie.
- Seamless upgrade from Proxmox Datacenter Manager Alpha, see Proxmox Datacenter Manager Upgrade from Alpha to Beta.
- EVPN configuration for Software-Defined Networking between clusters.

  A new panel provides an overview of the state of all EVPN zones across all remotes.

  Create EVPN Zones and VNets across multiple remotes from a single interface.

  A more detailed explanation of Proxmox Datacenter Manager's SDN capabilities can be found in the documentation.
- Improved search functionality to find resources quicker.

  Allows filtering by resource type (remote, virtual machine, container...), status (stopped, running...) and much more.

  The query syntax is inspired by Elasticsearch and GitHub's query language.

  Please refer to the documentation for a more thorough explanation of the syntax.
- More efficient metric collection logic.

  Metrics are now collected concurrently.
- Integrate privilege management in the access control UI.

  Allow managing the permissions of Proxmox Datacenter Manager users.

Changelog Overview

Enhancements in the Web Interface (GUI)

- Add a time frame selector for RRD graphs to allow users to select the displayed time frame.
- Display new metrics such as Pressure Stall Information (PSI) for Proxmox VE 9 hosts.
- Improve the remote URL list of a remote by adding a placeholder, clear trigger and clearer column headers.
- Enhancements to the Proxmox VE remote setup wizard.

  Probe hosts for fingerprint settings, to verify a provided fingerprint or to enable trust on first use (TOFU).

  Try matching the provided host against the host list that was queried from the remote to avoid duplicates.

  Reset later pages when previous pages have been changed, as they need to be revisited.

- Make the “remote loading” icon nicer.
- Correctly show a “cube” icon for container guests.
- Add a panel that allows adding and editing permissions.
- Move the node overview to a tab and add a tab that displays available updates.
- Add a button linking the user to a remote’s upgrade page.
- Add descriptions for Software Defined Networking tasks.
- Provide an EVPN overview panel for displaying EVPN Zones and Vnets.
- Add a view for showing EVPN VRF instances across all remotes.
- Allow creating EVPN VNets.
- Open the search panel when clicking different panels in the dashboard and pre-fill it with appropriate filters.
- Add a clear trigger to the search bar.
- Provide a search icon in the guest panel for better discoverability of the search function.
- Include a summary of all tasks in the dashboard.
- Render status icons with a shadow instead of a solid background for a cleaner look.
- Enhance the reloading logic for the dashboard.
- Show tasks from the last 48 hours in the dashboard’s task summary.
- Close the search box if a user navigated to an entry.
- Display a list of storages and their status in the resource tree of a Proxmox VE remote.
- Change the warning and critical thresholds to 90% and 97.5% respectively.
- Don’t show a start or shutdown button for templates (issue 6782).
- The dashboard now includes a panel showing the SDN status report.
- Show an overview of all SDN zones and their status as a tree.

  The EVPN section is now moved below the SDN menu to mimic Proxmox VE’s menu structure.
- Route to correct panels when navigating between components.
- Allow filtering tasks in the task list by remote.
- Show the remote tasks when selecting the root node of the resource tree for a Proxmox VE remote.
- Allow navigating to an SDN zone and SDN panel of a remote from the zone tree overview.
- Show failed tasks only in task summary.
- Add support for initial translations:
  - Arabic
  - Bulgarian
  - Catalan
  - Chinese (Simplified)
  - Chinese (Traditional)
  - Croatian
  - Czech
  - Danish

  - Dutch
  - Euskera (Basque)
  - French
  - Georgian
  - German
  - Hebrew
  - Italian
  - Japanese
  - Korean
  - Norwegian (Bokmal)
  - Norwegian (Nynorsk)
  - Persian (Farsi)
  - Polish
  - Portuguese (Brazil)
  - Russian
  - Slovenian
  - Spanish
  - Swedish
  - Turkish
  - Ukrainian

Remotes Management

- Enable Proxmox Backup Server Integration, CLI only for now.
- Enable connection tracking when live migrating VMs on remotes.

  Whether connection tracking actually persists after migration also depends on the environment and especially on whether third party firewalls are used.
- Enable trust on first use (TOFU) prompts when adding remotes.
- Include templates in status counts.
- Add an API endpoint that allows querying updates and changelogs from remotes.
- Add the API infrastructure for the initial Software Defined Networking integration.

Backend Improvements

- Improve robustness of incoming connection handling.
- Improve size requirements and performance for remote tasks cache.
- More intelligently query remote tasks.
- Fix an issue where the ACME configuration would not be constructed properly for the default account.
- Collect metrics from remotes concurrently to improve performance.
- Persist metric collection state after a run to allow reusing it after a daemon restart.

  This should allow more efficient metric collection runs after restarts.
- Metrics that should have been collected already, but were not due to collection timing changes, will now be collected.

- Keep track of the time it took to collect metrics from each single remote and all remotes together.

  This provides better insights into the performance of metric collection runs.
- Add an API endpoint to trigger metric collection.
- Trigger immediate metric collection when a remote is added.
- When a metric collection task is delayed skip it instead of triggering it quicker.
- Add a more complex filter and search syntax inspired by Elasticsearch and GitHub query language.
- When querying the remote task list treat a limit of "0" as unbounded and return the entire list.
- Allow filtering remote tasks by remote.
- Add an API endpoint that allows querying remote task statistics.
- Add API endpoints for querying Proxmox VE storage's RRD data and status.
- Add a `resource-type` parameter to the resources API endpoints.

  This allows more efficient filtering when querying the API for tasks and resource statuses.
- Don't match templates when searching by remote.
- Improve search when searching by remotes.

  For example, searching for all VMs of a specific remote is now possible.
- When encountering an error, return the root cause not the top level error when fetching remotes.

  This makes the reported error messages more specific.

Command Line Interface Enhancements

- Allow query the status and RRD data from remotes via `proxmox-datacenter-manager-client`.
- Add an upgrade checking script (`pdmAtoB`) to make upgrades more seamless.
- The utility `proxmox-datacenter-manager-admin` can now display the currently running version.

Miscellaneous Improvements

- Log an error when a task to query remote tasks fails instead of cancelling all tasks.
- Fix the order filters are applied when requesting a filtered task list.
- Use the new deb822 format for package repositories.
- Add a CLI command to allow querying the metric collection status and triggering a metric collection run.
- Handle a missing journal file error more gracefully when querying the task list.

Known Issues & Breaking Changes

- The API endpoint for listing realms was changed from a `POST` to a `GET` request.

Proxmox Datacenter Manager 0.1 ALPHA

Released 19. December 2024

- Based on Debian Bookworm (12.8)
- Latest 6.8.12-5 Kernel as stable default

- Newer 6.11 Kernel as opt-in
- ZFS: 2.2.6 (with compatibility patches for Kernel 6.11)

Features (Highlights)

- Connect to and view any number of independent nodes or clusters (”Datacenters”)
- View the basic resource usage of all nodes and their guests.

  Saves and caches the list of resources (mainly guests and storage) and their usage metrics to provide a quick overview of all resources and the last-seen state for offline/unresponsive ones.
- Basic management of the resources: shutdown, reboot, start, …

  For more complex management tasks, it provides a direct link to the full web interface of Proxmox VE/Proxmox Backup Server/…
- Remote migration of virtual guests between different datacenters.
- Support for the standard Proxmox feature set including complex Multi-Factor Authentication or ACME/Let’s Encrypt from the beginning.

Changelog Overview

Not applicable for the first alpha release.

Known Issues & Breaking Changes

This is an alpha release, there might be lots of stuff that is broken, gets reworked and fixed somewhat frequently.

APPENDIX

D

MARKDOWN PRIMER

> "Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML)."
>
> —John Gruber, https://daringfireball.net/projects/markdown/

The "Notes" panel of the *Proxmox Datacenter Manager* web-interface supports rendering Markdown text.

Proxmox Backup Server supports CommonMark with most extensions of GFM (GitHub Flavoured Markdown), like tables or task-lists.

D.1 Markdown Basics

Note that we only describe the basics here. Please search the web for more extensive resources, for example on https://www.markdownguide.org/

D.1.1 Headings

```
# This is a Heading h1
## This is a Heading h2
##### This is a Heading h5
```

D.1.2 Emphasis

Use `*text*` or `_text_` for emphasis.

Use `**text**` or `__text__` for bold, heavy-weight text.

Combinations are also possible, for example:

```
_You **can** combine them_
```

D.1.3 Links

You can use automatic detection of links. For example, `https://forum.proxmox.com/` would transform it into a clickable link.

You can also control the link text, for example:

```
Now, [the part in brackets will be the link text](https://forum.proxmox.com/).
```

D.1.4 Lists

Unordered Lists

Use * or - for unordered lists, for example:

```
* Item 1
* Item 2
* Item 2a
* Item 2b
```

You can create nested lists by adding indentation.

Ordered Lists

```
1. Item 1
1. Item 2
1. Item 3
   1. Item 3a
   1. Item 3b
```

NOTE: The integer of ordered lists does not need to be correct, they will be numbered automatically.

Task Lists

Task lists use a empty box [ ] for unfinished tasks and a box with an *X* for finished tasks.

For example:

```
- [X] First task already done!
- [X] Second one too
- [ ] This one is still to-do
- [ ] So is this one
```

D.1.5 Tables

Tables use the pipe symbol | to separate columns, and - to separate the table header from the table body. In that separation, you can also set the text alignment, making one column left-, center-, or right-aligned.

```
| Left columns  | Right columns |  Some   | More | Cols.| Centering Works Too
| ------------- |--------------:|---------|------|------|:------------------:|
| left foo      | right foo     | First   | Row  | Here | >center<           |
| left bar      | right bar     | Second  | Row  | Here | 12345              |
| left baz      | right baz     | Third   | Row  | Here | Test               |
| left zab      | right zab     | Fourth  | Row  | Here | ▰▰▰                |
| left rab      | right rab     | And     | Last | Here | The End            |
```

Note that you do not need to align the columns nicely with white space, but that makes editing tables easier.

D.1.6 Block Quotes

You can enter block quotes by prefixing a line with >, similar as in plain-text emails.

```
> Markdown is a lightweight markup language with plain-text-formatting syntax,
> created in 2004 by John Gruber with Aaron Swartz.
>
>> Markdown is often used to format readme files, for writing messages in online discussion␣
 ↪forums,
>> and to create rich text using a plain text editor.
```

D.1.7 Code and Snippets

You can use backticks to avoid processing a group of words or paragraphs. This is useful for preventing a code or configuration hunk from being mistakenly interpreted as markdown.

Inline Code

Surrounding part of a line with single backticks allows you to write code inline, for examples:

```
This hosts IP address is `10.0.0.1`.
```

Entire Blocks of Code

For code blocks spanning several lines, you can use triple-backticks to start and end such a block, for example:

````
```
# This is the network config I want to remember here
auto vmbr2
iface vmbr2 inet static
        address 10.0.0.1/24
        bridge-ports ens20
        bridge-stp off
        bridge-fd 0
        bridge-vlan-aware yes
        bridge-vids 2-4094
```
````

APPENDIX

E

GNU FREE DOCUMENTATION LICENSE

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <https://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document ”free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ”copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ”Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as ”you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ”Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ”Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ”Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ”Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ”Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ”Transparent” is called ”Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The ”Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ”Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The ”publisher” means any person or entity that distributes copies of the Document to the public.

A section ”Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ”Acknowledgements”, ”Dedications”, ”Endorsements”, or ”History”.) To ”Preserve the Title” of such a section when you modify the Document means that it remains a section ”Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled ”Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled ”Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled ”Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ”History” in the various original documents, forming one section Entitled ”History”; likewise combine any sections Entitled ”Acknowledgements”, and any sections Entitled ”Dedications”. You must delete all sections Entitled ”Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ”aggregate” if the

copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/licenses/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those

works. A public wiki that anybody can edit is an example of such a server. A ”Massive Multiauthor Collaboration” (or ”MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

”CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

”Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is ”eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.