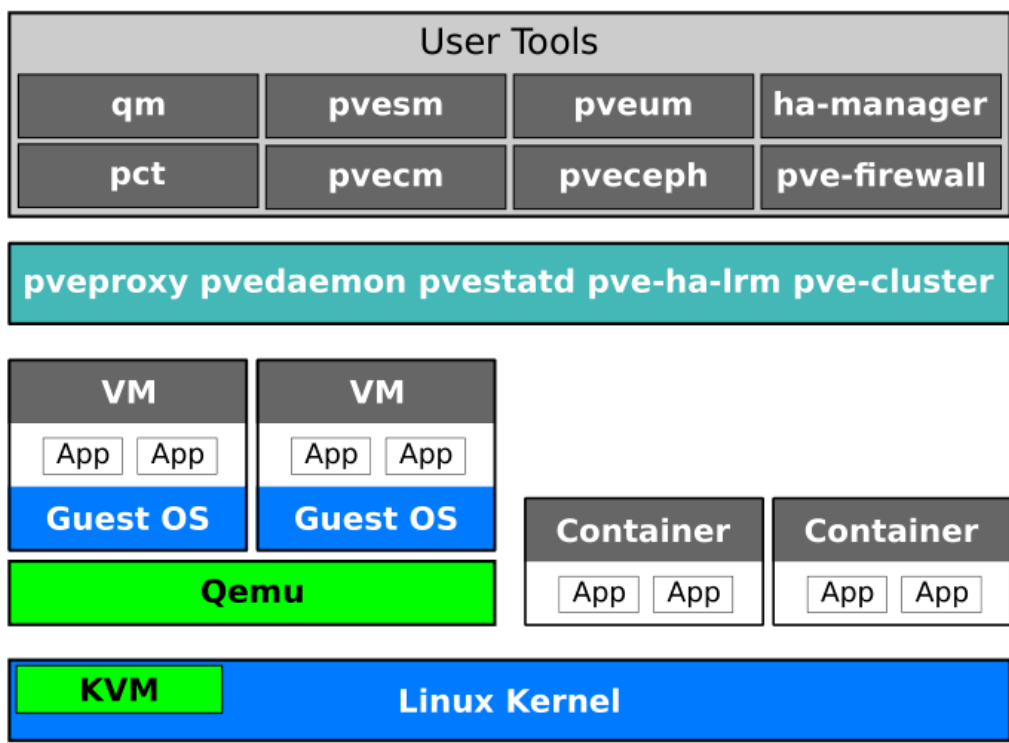


# 第1章 简介

Proxmox VE 是一个既可以运行虚拟机也可以运行容器的虚拟化平台。Proxmox VE 基于 Debian Linux 开发，并且完全开源。出于灵活性的考虑，Proxmox VE 同时支持两种虚拟化技术：KVM 虚拟机和 LXC 容器。

Proxmox VE 的一个重要设计目标就是尽可能简化管理员的工作。你既可以用单机模式使用 Proxmox VE，也可以组建多节点 Proxmox VE 集群。所有的管理工作都可以通过基于 web 页面的管理界面完成，即使是一个小白用户也可以在几分钟内上手安装使用 Proxmox VE。



## 1.1 集中管理

尽管很多人一开始都使用单机方式运行 Proxmox VE，但实际上 Proxmox VE 可以横向扩展为一个拥有大量节点的集群。Proxmox VE 的默认安装方式中就已经包含了全套的集群套件。

### 独特的多主集群架构

内嵌的 WebGUI 管理控制台可以让你纵览所有的 KVM 虚拟机、LXC 容器和整个集群。你也可以通过 WebGUI 轻松管理你的虚拟机、容器、存储和集群。完全没有必要另外安装单独的管理服务器。

### Proxmox 集群文件系统（pmxcfs）

Proxmox VE 使用专门设计的基于数据库的 Proxmox 文件系统（pmxcfs）保存配置文件。这个文件系统足以让你保存几千台虚拟机的配置信息，并且能够通过 corosync 将配置文件实时复制到 Proxmox VE 集群的所有节点。Proxmox 文件系统一方面将所有数据都保存在服务器磁盘的一个数据库文件上，以避免数据丢失，另一方面在内存里也复制了一个副本，以提高性能。其中内存副本的最大容量为 30M，虽然不是很大，但足以保存几千台虚拟机配置信息。

截至目前，Proxmox 是唯一利用这种集群文件系统管理配置信息的虚拟化平台。

### 基于 Web 的管理界面

Proxmox VE 的使用很简单。管理操作可以通过内嵌的 WebGUI 完成—不需要专门安装管理工具或基于大型数据库的管理服务器节点。多主集群架构能够让你通过任意节点管理整个集群。基于 JavaScript 框架（ExtJS）开发的集中 Web 管理界面不仅能够让你通过 GUI 界面控制一切功能，而且可以浏览每个节点的历史活动和 syslog 日志，例如虚拟机备份恢复日志、虚拟机在线迁移日志、HA 活动日志等。

### 命令行工具

对于那些用惯了 Unix Shell 或 Windows Powershell 的高级用户，Proxmox VE 提供了一个命令行界面，可以管理虚拟化环境里的全部组件。这个命令行工具不仅有 Tab 键补全功能，而且提供了完善的 Unix man 形式的技术文档。

### REST API

Proxmox VE 使用了 RESTful 形式的 API。开发人员选用 JSON 作为主要数据格式，所有的 API 定义均采用 JSON 语法。第三方管理工具很容易就可以将 Proxmox VE 的 API 集成进去。

## 基于角色的权限管理

在 Proxmox VE 中你可以用基于角色的方法对所有对象（包括虚拟机、存储、节点等等）设置用户管理权限。你可以定义权限，并控制对每个对象的访问。Proxmox VE 的权限管理方式实际上类似于访问控制列表：每个权限都针对特定主体（用户或用户组），每个角色（一组权限）都被限制在特定目录。

## 多种身份认证

Proxmox VE 支持多种用户身份认证方法，具体包括 Microsoft 活动目录，LDAP，Linux 系统用户认证，Proxmox VE 内嵌身份认证。

## 1.2 支持多种存储类型

Proxmox VE 支持多种存储技术。虚拟机镜像既可以保存在服务器本地存储，也可以保存在基于 NFS 或 SAN 的共享存储设备上。你可以根据需要自由地为 Proxmox VE 配置多种存储。实际上，Debian Linux 支持的所有类型的存储技术都可以用于 Proxmox VE。

用共享存储来保存虚拟机镜像有一个很大的好处，那就是 Proxmox VE 集群中的所有节点都可以直接访问到该虚拟机镜像，虚拟机就可以从一个 Proxmox VE 节点在线迁移到其他节点运行，并且虚拟机在迁移过程中可以保持连续运行，无需关机。

Proxmox VE 目前支持的网络共享存储类型如下：

- LVM 卷组（基于 iSCSI 网络存储）
- iSCSI 网络存储设备
- NFS 共享存储
- Ceph RBD
- iSCSI 卷
- GlusterFS

支持的本地存储类型如下：

- LVM 卷组（基于本地磁盘、FC 磁盘、DRBD 等）
- 目录（本地文件系统）
- ZFS

## 1.3 虚拟机备份与恢复

Proxmox VE 内嵌了虚拟机备份工具（vzdump），可以在线创建 KVM 虚拟机和 LXC 容器的快照备份。创建的备份不仅包括虚拟机和容器的完整镜像数据，同时包含了相应的配置文件信息。

KVM 虚拟机在线备份功能兼容所有的存储类型，对于保存在 NFS、iSCSI LUN、Ceph RBD 或者 Sheepdog 上的虚拟机镜像，均可以进行备份。目前新的备份文件格式进行过特别优化，确保备份过程的高效和快速（优化内容包括稀疏磁盘镜像文件、非连续镜像文件数据、最小化 I/O 等）。

## 1.4 高可用集群

多节点 Proxmox VE HA 集群支持用户自定义配置高可用的虚拟机。Proxmox VE HA 集群基于久经考验的 Linux HA 技术，能够提供稳定可靠的 HA 服务。

## 1.5 支持多种虚拟网络技术

Proxmox VE 支持基于桥接模式的虚拟网络。在该模式下，所有的虚拟机共享一个虚拟交换机，效果就相当于所有的虚拟机同时接入了同一个交换机一样。虚拟交换机还可以和 Proxmox VE 的物理网卡桥接，以便相关虚拟机和外部网络进行 TCP/IP 通讯。

此外，Proxmox VE 还支持 VLANs（802.1q）和网卡绑定/链路聚合技术。用户可以充分利用 Linux 网络组件的强大功能，在 Proxmox VE 服务器上构建非常复杂和多样的虚拟网络环境。

## 1.6 内嵌防火墙

你可以利用 Proxmox VE 的内嵌防火墙对任意虚拟机和容器的网络通信流量进行过滤。还可以利用“Security groups”把常用防火墙策略和集合分组管理。

## 1.7 开源的原因

Proxmox VE 使用 Linux 作为内核，并且基于 Debian GNU/Linux 构建用户空间组件。Proxmox VE 的源代码基于 [GNU Affero General Public License, version 3](#) 发布。这确保你可以在任何时候都可以自由查看 Proxmox VE 源代码，并向该项目共享代码。

在 Proxmox 上我们坚持使用开源软件。使用开源软件不仅能确保能使用所有功能，还保证了软件的安全和可靠。我们认为每个人都有权访问软件的源代码，以便于更好的运行软件、打包软件、为软件提交新的代码。我们鼓励每个人向 Proxmox VE 贡献代码，同时我们将确保 Proxmox VE 始终保持专业软件的品质。

开源软件还能帮助你节约成本，避免你的基础设施产生单一厂商依赖问题。

## 1.8 Proxmox VE 的优势

- 开源软件
- 没有单一厂商依赖
- Linux 内核
- 快速安装，易于使用
- 基于 Web 的管理界面
- REST API
- 庞大而活跃的社区
- 很低的管理和部署成本

## 1.9 获取支持

### 1.9.1 Proxmox VE Wiki

Proxmox VE 技术资料的一个主要来源就是 [Proxmox VE Wiki](#)。其中既有官方参考文档，也有用户贡献的内容。

### 1.9.2 社区支持论坛

Proxmox VE 本身是完全开源的。我们鼓励用户在 [Proxmox VE Community Forum](#) 讨论和分享关于 Proxmox VE 的知识。这个论坛完全由 Proxmox 支持团队主持，并且拥有来自全世界的众多用户群体。毫无疑问，这个论坛是一个非常棒的获取 Proxmox VE 相关信息的途径。

### 1.9.3 电子邮件列表

通过电子邮件快速访问 Proxmox VE 社区的方式如下。

- 用户电子邮件列表：[PVE User List](#)

Proxmox VE 开发者的主要通讯频道如下。

- 开发者电子邮件列表：[PVE development discussion](#)

### 1.9.4 商业支持

Proxmox Server Solutions Gmbh 提供了 Proxmox VE 商业支持服务 [Proxmox VE Subscription Service Plan](#)。在发生问题时，订阅了 Proxmox VE 服务的系统管理员可以通过专门的支持渠道寻求支持，并可以在规定的响应时间内获得 Proxmox VE 开发人员的帮助。要获取更多具体信息以及咨询折扣，请直接联系 Proxmox 销售团队 [Proxmox sales team](#)。

### 1.9.5 Bug 提交及跟踪

我们有一个公开的 Bug 跟踪系统 <http://bugzilla.proxmox.com>。如果你遇到了 bug，可以在这里创建相关 bug 记录。你可以通过这个系统跟踪 bug 状态，也可以在第一时间获得 bug 修复的消息。

## 1.10 项目历程

Proxmox VE 项目始于 2007 年，并在 2008 年发布了第一个 stable 版本。当时我们采用了 OpenVZ 容器技术和 KVM 虚拟机技术。但集群功能十分有限，用户界面也很简陋（页面由服务器生成）。

但我们很快用 [Corosync](#) 集群组件开发了新的集群功能，并通过引入新的 Proxmox 集群文件系统（pmxcfs）很好地对用户屏蔽了集群的复杂性。这是一个很大的进步。用户管理一个有 16 个节点的集群就和管理一个单机系统一样简单。

我们还引入了新的 REST API，并用 JSON 定义了所有的 API。借助 REST API，第三方不仅可以将 Proxmox VE 集成到他们现有的 IT 基础设施中，并且可以很容易地开发新的服务。

此外，利用新的 REST API，我们用基于 Javascript 的 HTML5 应用替换了原有的用户界面。我们还用 [noVNC](#) 替换了原来基于 Java 的 VNC 控制台组件。现在你只需要通过浏览器就可以直接访问虚拟机桌面。

支持不同类型的存储技术是 Proxmox VE 另一个重要特性。其中值得一提的是，在 2014 年 Proxmox VE 就默认支持 ZFS，这在 Linux 发行版中是第一个。另一个重要的里程碑是在 Proxmox VE 服务器上运行 [Ceph](#) 存储服务，从而提供了一种性价比极高的部署方式。

我们是最早提供 KVM 虚拟机商业软件技术服务的公司之一。KVM 技术本身在不断演进，目前已经是被广泛使用的虚拟机技术，而且随着每个新版本发布都会有新功能推出。我们开发了 KVM 虚拟机在线备份功能，从而可以对保存在任何存储设备上的 KVM 虚拟机进行快照式备份。

Proxmox VE 4.0 的一个重大变化就是舍弃了 OpenVZ 容器并转向了 LXC 容器技术。目前容器技术已经深度整合到了 Proxmox VE 当中，并且可以和虚拟机在同一个存储和网络环境中同时使用。

## 1.11 参与完善 Proxmox VE 文档

根据你想要改进的内容主题，可以用不同的方式提交给 Proxmox VE 开发人员。

如果你想修正当前文档中的错误，可以使用 [Proxmox bug tracker](#) 提交更正后的文档。

如果你想增加新的内容主题，则取决于你希望增加的文档内容类型：

- 如果想增加的内容仅限于你自己的部署环境，那么添加到 wiki 上是最合适的。例如，你想增加的内容是关于特定虚拟机的配置信息，比如是针对一个冷门操作系统的最佳 Qemu 驱动组合，那么就非常适合用 wiki 文章的形式来记录。
- 如果想增加的内容是关于一般性问题，并且对所有用户都会有帮助，你可以尝试添加到参考文档。参考文档使用 asciidoc 文档格式编写。你可以先克隆参考文档

代码仓库 `git://git.proxmox.com/git/pve-docs.git`，然后按照 [README.adoc](#) 中的指示来编写新的内容。

参与完善 Proxmox VE 文档就和在 Wikipedia 上写文章一样简单，并且也是参与大型开源软件项目的一种有趣的尝试。

---

➤ 注意

如果你有兴趣了解 Proxmox VE 代码仓库的使用，那么可以先从 wiki 文章 [Development Documentation](#) 开始。

---