LIBRO CASTELLANO PROXMOX VE

GUÍA DE ADMINISTRACIÓN

RELEASE 9.2.4

4 de agosto de 2026

PARTE 1

Capítulo 1 · Introducción



Palentino.es

PROXMOX VE 9.2 — GUÍA EN CASTELLANO

Traducción no oficial

Esta obra es una traducción no oficial al castellano y versión adaptada de la documentación de Proxmox VE, realizada con el objetivo de facilitar el estudio, aprendizaje y consulta técnica de Proxmox VE a la comunidad hispanohablante.

Documentación original: Proxmox VE Administration Guide
Titular de la documentación original: © Proxmox Server Solutions GmbH
Traducción, revisión y estructuración en castellano: © 2026 Óscar de la Cuesta
Edición: 2026

Esta traducción ha sido realizada de forma independiente y no constituye documentación oficial de Proxmox Server Solutions GmbH, ni implica aprobación, patrocinio o respaldo por parte de dicha entidad.

Los nombres Proxmox y Proxmox VE, así como las marcas y logotipos relacionados, pertenecen a sus respectivos titulares.

Licencia

La documentación original de Proxmox VE se distribuye bajo los términos de la GNU Free Documentation License, versión 1.3 o cualquier versión posterior publicada por la Free Software Foundation, sin Secciones Invariantes, sin Textos de Portada y sin Textos de Contraportada.

Esta traducción constituye una versión modificada de la documentación original a los efectos de la GNU Free Documentation License y se distribuye bajo los mismos términos.

Se permite copiar, distribuir y/o modificar este documento de acuerdo con las condiciones establecidas en la GNU Free Documentation License, versión 1.3 o posterior.

Se conserva el aviso de copyright correspondiente a la documentación original de Proxmox Server Solutions GmbH y se reconoce separadamente la autoría del trabajo de traducción, revisión y estructuración realizado para esta edición en castellano.

La GNU Free Documentation License completa y sin modificaciones debe incluirse como parte de esta obra.

Aunque se ha realizado un trabajo de traducción, revisión terminológica y estructuración destinado a mantener la máxima fidelidad posible respecto de la documentación original, ante cualquier discrepancia técnica deberá considerarse como referencia la documentación oficial publicada por Proxmox Server Solutions GmbH.

Fuente de referencia: Proxmox VE Administration Guide.

Nota del traductor

Esta edición en castellano de la **Proxmox VE Administration Guide 9.2.4** ha sido traducida y preparada por **Oscar de la Cuesta — palentino.es**, con el objetivo de facilitar el acceso a la documentación técnica de Proxmox VE a administradores de sistemas, profesionales IT, estudiantes y usuarios de habla hispana.

Durante la traducción se ha intentado **respetar fielmente la estructura, el contenido técnico y el significado del documento original**, manteniendo sin traducir cuando resulta necesario los nombres de comandos, parámetros, rutas, tecnologías, opciones de configuración y demás elementos cuya modificación podría alterar su significado o utilidad práctica.

Esta traducción **no pretende sustituir la documentación oficial de Proxmox**, sino ofrecer una versión en castellano que facilite su estudio y consulta. Proxmox, Proxmox VE y las denominaciones de sus productos pertenecen a sus respectivos titulares.

Sobre la licencia

El documento original se distribuye bajo los términos de la **GNU Free Documentation License, versión 1.3 o cualquier versión posterior publicada por la Free Software Foundation**, sin Secciones Invariantes, sin Textos de Portada y sin Textos de Contraportada. pve-admin-guide-9.2

La propia licencia considera la traducción como una forma de modificación y permite distribuir traducciones bajo sus condiciones. Por este motivo, esta edición mantiene el aviso de licencia correspondiente y conserva también la **versión original en inglés de la GNU Free Documentation License**, que prevalece en caso de discrepancia con una traducción.

Traducción y preparación de la edición en castellano:
Oscar de la Cuesta - @oscardelacuesta
01-10-2026
palentino.es

Proxmox VE Administration Guide — Release 9.2.4
Documento original: **Proxmox Server Solutions GmbH**

Aviso de copyright y licencia

Copyright © 2026 Proxmox Server Solutions GmbH

Se concede permiso para copiar, distribuir y/o modificar este documento bajo los términos de la GNU Free Documentation License, versión 1.3 o cualquier versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes, sin Textos de Portada y sin Textos de Contraportada.

Se incluye una copia de la licencia en la sección titulada "GNU Free Documentation License".

Capítulo 1 · Introducción

Proxmox VE es una plataforma para ejecutar máquinas virtuales y contenedores. Está basada en Debian GNU/Linux y es completamente de código abierto. Para ofrecer la máxima flexibilidad, Proxmox VE proporciona dos tecnologías de virtualización: Kernel-based Virtual Machine (KVM) y Linux Containers (LXC).

Uno de los objetivos principales de diseño es hacer que la administración resulte lo más sencilla posible. Puede utilizar Proxmox VE en un único nodo o montar un clúster formado por muchos nodos. Todas las tareas de gestión pueden realizarse mediante la interfaz de administración web, y hasta un usuario principiante puede configurar e instalar Proxmox VE en pocos minutos.

Visión general de herramientas y servicios

Herramientas de usuario	qm · pvesm · pveum · ha-manager
	pct · pvecm · pveceph · pve-firewall
Servicios	pveproxy · pvedaemon · pvestatd · pve-ha-lrm · pve-cluster
Virtualización	VMs mediante QEMU/KVM · Contenedores mediante LXC
Base	Linux Kernel · AppArmor · cgroups

1.1 Gestión centralizada

Aunque muchas personas comienzan con un único nodo, Proxmox VE puede escalar hasta un gran conjunto de nodos agrupados en clúster. La pila de clúster está totalmente integrada y se incluye con la instalación predeterminada.

Diseño Multi-Master único

Cada nodo de un clúster Proxmox VE puede realizar todas las tareas de administración. No existe un servidor central de gestión que constituya un único punto de fallo: cualquier nodo puede administrar todo el clúster. Esta arquitectura multi-master mantiene administrable el clúster incluso cuando nodos individuales no están disponibles temporalmente.

Proxmox Cluster File System (pmxcfs)

Proxmox VE utiliza Proxmox Cluster File System (pmxcfs), un sistema de archivos basado en base de datos para almacenar los archivos de configuración. Puede guardar la configuración de miles de máquinas virtuales. Corosync replica estos archivos en tiempo real a todos los nodos del clúster. El sistema de archivos almacena todos los datos dentro de una base de datos persistente en disco, mientras que una copia reside en RAM, proporcionando un tamaño máximo de almacenamiento de 128 MiB, más que suficiente para miles de VMs.

Proxmox VE es la única plataforma de virtualización que utiliza este sistema de archivos de clúster.

Interfaz de administración basada en web

Proxmox VE es sencillo de utilizar. La interfaz web incluida, construida sobre el framework JavaScript ExtJS, ofrece una visión clara de todos los invitados KVM y contenedores Linux, así como control completo de todos los recursos del clúster. Desde la GUI puede administrar VMs, contenedores, almacenamiento y red; consultar el historial y syslog de cada nodo; ejecutar tareas de backup o restauración; realizar migraciones en vivo y gestionar actividades activadas por HA. No es necesario instalar una herramienta de administración independiente ni un nodo adicional de gestión.

Línea de comandos

Para usuarios avanzados que prefieren la comodidad de un shell Unix o Windows PowerShell, Proxmox VE proporciona una interfaz de línea de comandos para gestionar todos los componentes del entorno virtual. Esta interfaz dispone de autocompletado inteligente mediante tabulador y documentación completa en forma de páginas man.

API REST

Proxmox VE utiliza una API RESTful con JSON como formato de datos principal. Toda la API está definida formalmente mediante JSON Schema, lo que permite una integración rápida y sencilla con herramientas de administración de terceros, como entornos de hosting personalizados.

Administración basada en roles

Puede definir accesos granulares para todos los objetos, como VMs, almacenamiento y nodos, mediante la gestión de usuarios y permisos basada en roles. Cada permiso especifica un sujeto, usuario o grupo, y un rol, conjunto de privilegios, sobre una ruta concreta, siguiendo el modelo de listas de control de acceso.

Dominios de autenticación

Proxmox VE admite múltiples fuentes de autenticación, entre ellas Microsoft Active Directory, LDAP, la autenticación estándar Linux PAM y el servidor de autenticación integrado de Proxmox VE.

1.2 Almacenamiento flexible

El modelo de almacenamiento de Proxmox VE es muy flexible. Las imágenes de las máquinas virtuales pueden almacenarse en uno o varios almacenamientos locales, o en almacenamiento compartido como NFS o SAN. Puede configurar tantas definiciones de almacenamiento como desee, utilizando todas las tecnologías de almacenamiento disponibles para Debian GNU/Linux.

Una de las principales ventajas de almacenar las VMs en almacenamiento compartido es la posibilidad de migrar en vivo máquinas en ejecución sin tiempo de inactividad, ya que todos los nodos del clúster tienen acceso directo a las imágenes de disco de las VMs.

Se admiten los siguientes tipos de almacenamiento de red:

- Grupo LVM (backend de red con targets iSCSI)
- Target iSCSI
- LUNs iSCSI (uso directo)
- Recurso compartido NFS
- Recurso compartido CIFS
- Ceph RBD
- CephFS
- Proxmox Backup Server

Se admiten los siguientes tipos de almacenamiento local:

- Grupo LVM (dispositivos de backend local como dispositivos de bloques, dispositivos Fibre Channel y DRBD)
- Directorio (almacenamiento sobre un sistema de archivos existente)
- ZFS

1.3 Backup y restauración integrados

La herramienta de backup integrada (vzdump) crea snapshots coherentes de contenedores en ejecución e invitados KVM. Genera un archivo con los datos de la VM o del contenedor, incluidos los archivos de configuración.

Además, la integración con Proxmox Backup Server proporciona funciones avanzadas, entre ellas la posibilidad de combinar backups completos y deduplicación de datos, realizar cifrado en el lado del cliente, hacer copias a cinta o almacenamiento de objetos S3 y sincronizar con otras instalaciones remotas de Proxmox Backup Server.

El backup en vivo de KVM funciona con todos los tipos de almacenamiento, incluidas imágenes de VM sobre NFS, CIFS, LUN iSCSI y Ceph RBD. El formato de backup está optimizado para almacenar de forma eficiente las copias de VMs, utilizando archivos sparse, datos fuera de orden y E/S minimizada.

1.4 Clúster de alta disponibilidad

Un clúster HA multinodo de Proxmox VE permite disponer de máquinas virtuales y contenedores en alta disponibilidad. El gestor High Availability (HA) integrado detecta automáticamente los fallos de nodos mediante fencing basado en watchdog y reinicia los servicios afectados en nodos sanos. No se necesitan herramientas externas como Pacemaker: la pila HA se incluye con Proxmox VE y se configura completamente desde la interfaz web o la línea de comandos.

1.5 Redes flexibles

Proxmox VE utiliza un modelo de red basado en bridges. Todas las VMs pueden compartir un mismo bridge como si los cables de red virtuales de cada invitado estuvieran conectados al mismo switch. Para conectar las VMs con el exterior, los bridges se asocian a tarjetas de red físicas y se les asigna una configuración TCP/IP.

Para obtener mayor flexibilidad, se dispone de Virtual LANs (VLANs, IEEE 802.1q) y bonding/agregación de red, lo que permite construir redes virtuales complejas y flexibles para los hosts Proxmox VE y aprovechar toda la potencia de la pila de red de Linux.

Para entornos más grandes o multi-tenant, la pila Software Defined Network (SDN) proporciona aislamiento de red basado en zonas y admite protocolos como VXLAN y EVPN. SDN está completamente integrada en la interfaz web y se instala de forma predeterminada.

1.6 Firewall integrado

El firewall integrado permite filtrar paquetes de red en cualquier interfaz de VM o contenedor. Los conjuntos comunes de reglas de firewall pueden agruparse en “security groups”.

1.7 Infraestructura hiperconvergente

Proxmox VE es una plataforma de virtualización que integra estrechamente recursos de computación, almacenamiento y red, y que gestiona clústeres de alta disponibilidad, backup/restauración y recuperación ante desastres. Todos los componentes están definidos por software y son compatibles entre sí.

Por ello, es posible administrarlos como un único sistema mediante la interfaz web centralizada. Estas capacidades convierten a Proxmox VE en una opción idónea para desplegar y gestionar una infraestructura hiperconvergente de código abierto.

1.7.1 Ventajas de una infraestructura hiperconvergente (HCI) con Proxmox VE

Una infraestructura hiperconvergente (HCI) resulta especialmente útil en despliegues donde una alta demanda de infraestructura coincide con un presupuesto de administración reducido, en configuraciones distribuidas como entornos de oficinas remotas o sucursales, o para nubes privadas y públicas.

HCI ofrece las siguientes ventajas:

- Escalabilidad: ampliación fluida de los recursos de computación, red y almacenamiento; por ejemplo, aumentar servidores y almacenamiento de forma rápida e independiente.
- Bajo coste: Proxmox VE es de código abierto e integra todos los componentes necesarios, como computación, almacenamiento, red, backup y centro de administración. Puede sustituir una infraestructura costosa de computación/almacenamiento.
- Protección de datos y eficiencia: servicios como backup y recuperación ante desastres están integrados.
- Simplicidad: configuración sencilla y administración centralizada.
- Código abierto: sin dependencia de un único proveedor.

1.7.2 Infraestructura hiperconvergente: almacenamiento

Proxmox VE ofrece soporte estrechamente integrado para desplegar una infraestructura de almacenamiento hiperconvergente. Por ejemplo, puede desplegar y gestionar únicamente desde la interfaz web las dos tecnologías de almacenamiento siguientes:

- Ceph: sistema de almacenamiento compartido fiable, altamente escalable, autorreparable y autogestionado. Consulte cómo administrar servicios Ceph en nodos Proxmox VE.
- ZFS: sistema combinado de archivos y gestor de volúmenes lógicos con amplia protección frente a corrupción de datos, varios modos RAID y snapshots rápidos y económicos, entre otras funciones. Consulte cómo aprovechar la potencia de ZFS en nodos Proxmox VE.

Además de lo anterior, Proxmox VE permite integrar una amplia gama de tecnologías de almacenamiento adicionales. Puede consultarlas en el capítulo Storage Manager.

1.8 Por qué código abierto

Proxmox VE utiliza un kernel Linux y está basado en la distribución Debian GNU/Linux. El código fuente de Proxmox VE se publica bajo la GNU Affero General Public License, versión 3. Esto significa que puede inspeccionar el código fuente en cualquier momento o contribuir personalmente al proyecto.

Proxmox se compromete a utilizar software de código abierto siempre que sea posible. El software de código abierto garantiza acceso completo a toda la funcionalidad, además de alta seguridad y fiabilidad. Todo el mundo debería tener derecho a acceder al código fuente del software para ejecutarlo, construir sobre él o enviar cambios de vuelta al proyecto. Se anima a todos a contribuir, mientras Proxmox garantiza que el producto cumpla siempre criterios de calidad profesional.

El software de código abierto también ayuda a mantener bajos los costes y hace que la infraestructura central sea independiente de un único proveedor.

1.9 Sus ventajas con Proxmox VE

- Software de código abierto
- Sin dependencia de proveedor
- Kernel Linux
- Instalación rápida y facilidad de uso
- Interfaz de administración basada en web
- API REST
- Gran comunidad activa
- Bajos costes de administración y despliegue sencillo

1.10 Obtener ayuda

1.10.1 Wiki de Proxmox VE

La principal fuente de información es la Wiki de Proxmox VE. Combina la documentación de referencia con contenido aportado por usuarios.

1.10.2 Foro de soporte de la comunidad

Proxmox VE es completamente de código abierto, por lo que siempre se anima a los usuarios a debatir y compartir conocimientos mediante el Proxmox VE Community Forum. El foro está moderado por el equipo de soporte de Proxmox y cuenta con una gran base de usuarios de todo el mundo. Un foro de este tamaño es un excelente lugar para obtener información.

1.10.3 Listas de correo

Es una forma rápida de comunicarse con la comunidad Proxmox VE por correo electrónico.

- Lista de correo para usuarios: Proxmox VE User List
- Proxmox VE es completamente de código abierto y las contribuciones son bienvenidas. El principal canal de comunicación para desarrolladores es:
- Lista de correo para desarrolladores: Proxmox VE development discussion

1.10.4 Soporte comercial

Proxmox Server Solutions GmbH también ofrece soporte empresarial mediante los Proxmox VE Subscription Service Plans. Todos los usuarios con suscripción obtienen acceso al Proxmox VE Enterprise Repository y, con una suscripción Basic, Standard o Premium, también al Proxmox Customer Portal. El portal de clientes proporciona ayuda y soporte con tiempos de respuesta garantizados por los desarrolladores de Proxmox VE.

Para descuentos por volumen o para obtener más información en general, contacte con sales@proxmox.com.

1.10.5 Bug Tracker

Proxmox mantiene un sistema público de seguimiento de errores en https://bugzilla.proxmox.com. Si aparece un problema, puede registrar allí su informe. Un problema puede ser un error, una solicitud de nueva función o una mejora. El bug tracker ayuda a realizar seguimiento y enviará una notificación cuando se haya resuelto.

1.11 Historia del proyecto

El proyecto comenzó en 2007, seguido de una primera versión estable en 2008. En aquel momento utilizaba OpenVZ para contenedores y KVM para máquinas virtuales. Las funciones de clúster eran limitadas y la interfaz de usuario era sencilla, una página web generada por el servidor.

El proyecto desarrolló rápidamente nuevas funciones utilizando la pila de clúster Corosync, y la introducción de Proxmox Cluster File System (pmxcfs) supuso un gran avance, ya que oculta completamente al usuario la complejidad del clúster. Administrar un clúster de 16 nodos es tan sencillo como administrar un único nodo.

La introducción de la API REST, con una especificación declarativa completa escrita en JSON Schema, permitió a otras personas integrar Proxmox VE en su infraestructura y facilitó la prestación de servicios adicionales.

La API REST también hizo posible sustituir la interfaz de usuario original por una moderna aplicación de una sola página ejecutada en el cliente mediante JavaScript. El antiguo código de consola VNC basado en Java fue reemplazado por noVNC, por lo que solo se necesita un navegador web para administrar las VMs.

El soporte para diversos tipos de almacenamiento es otro gran logro. Cabe destacar que Proxmox VE fue la primera distribución en incluir ZFS sobre Linux de forma predeterminada en 2014. Otro hito fue la capacidad de ejecutar y administrar almacenamiento Ceph en los propios nodos del hipervisor. Estas configuraciones son extremadamente rentables.

Cuando comenzó el proyecto, Proxmox fue una de las primeras empresas en ofrecer soporte comercial para KVM. El propio proyecto KVM ha evolucionado de forma continua y hoy es un hipervisor ampliamente utilizado. Con cada versión llegan nuevas funciones. La función de backup en vivo de KVM, desarrollada por Proxmox, permite crear copias snapshot sobre cualquier tipo de almacenamiento.

El cambio más destacado de la versión 4.0 fue el paso de OpenVZ a LXC. Los contenedores quedaron profundamente integrados y pueden utilizar las mismas funciones de almacenamiento y red que las máquinas virtuales. Al mismo tiempo se introdujo el gestor High Availability (HA), fácil de utilizar, simplificando la configu ración y administración de entornos de alta disponibilidad.

Durante el desarrollo de Proxmox VE 5 se introdujeron, entre muchas otras funciones, la replicación asíncrona de almacenamiento y la gestión automatizada de certificados mediante ACME/Let's Encrypt.

La pila Software Defined Network (SDN) se desarrolló en colaboración con la comunidad. Se integró en la interfaz web como función experimental en la versión 6.2, simplificando la administración de configuraciones de red sofisticadas. Desde la versión 8.1, la integración SDN está plenamente soportada y se instala de forma predeterminada.

En 2020 se publicó un nuevo proyecto, Proxmox Backup Server, escrito en el lenguaje de programación Rust. Proxmox Backup Server está profundamente integrado con Proxmox VE y mejora significativamente las capacidades de backup mediante copias incrementales, deduplicación y muchas otras funciones.

En 2022 se publicó otra nueva herramienta, Proxmox Offline Mirror, que permite utilizar suscripciones en sistemas sin conexión a Internet pública.

El muy solicitado tema oscuro para la interfaz web se introdujo en 2023. Más tarde ese mismo año, la versión 8.0 integró acceso al repositorio empresarial de Ceph. El acceso al repositorio de Ceph más estable está incluido con cualquier suscripción de Proxmox VE.

La instalación automatizada y desatendida para el instalador ISO oficial se introdujo en la versión 8.2, simplificando considerablemente los despliegues grandes de Proxmox VE.

Con el asistente de importación, también introducido en la versión 8.2, los usuarios pueden migrar invitados directa y eficientemente desde otros hipervisores como VMware ESXi. Además, los archivos en Open Virtualization Format (OVF/OVA) ahora pueden importarse directamente desde almacenamientos basados en archivos en la interfaz web.

Referencia del manual: Migrate to Proxmox VE - https://pve.proxmox.com/wiki/Migrate_to_Proxmox_VE

1.12 Mejorar la documentación de Proxmox VE

Las contribuciones y mejoras a la documentación de Proxmox VE son siempre bienvenidas. Existen varias formas de contribuir.

Si encuentra errores u otros aspectos mejorables en esta documentación, registre un error en el bug tracker de Proxmox para proponer una corrección.

Si desea proponer contenido nuevo, elija una de las siguientes opciones:

- La wiki: para configuraciones específicas, guías prácticas o tutoriales, la wiki es la opción adecuada para contribuir.
- La documentación de referencia: para contenido general que resulte útil a todos los usuarios, proponga su contribución para la documentación de referencia. Esto incluye información sobre cómo instalar, configurar, utilizar y solucionar problemas de las funciones de Proxmox VE. La documentación de referencia está escrita en formato asciidoc. Para editarla debe clonar el repositorio git en git://git.proxmox.com/git/pve-docs.git y después seguir el documento README.adoc.

Nota: Si está interesado en trabajar sobre el código de Proxmox VE, el artículo Developer Documentation de la wiki le mostrará por dónde empezar.

1.13 Traducir Proxmox VE

La interfaz de usuario de Proxmox VE está en inglés de forma predeterminada. Sin embargo, gracias a las contribuciones de la comunidad también existen traducciones a otros idiomas. Se agradece cualquier ayuda para añadir nuevos idiomas, traducir las funciones más recientes y mejorar traducciones incompletas o incoherentes.

Se utiliza gettext para gestionar los archivos de traducción. Herramientas como Poedit ofrecen una interfaz agradable para editar esos archivos, aunque puede utilizar el editor con el que se sienta más cómodo. No se necesitan conocimientos de programación para traducir.

1.13.1 Traducir con git

Los archivos de idioma están disponibles como repositorio git. Si está familiarizado con git, contribuya siguiendo la Developer Documentation.

Puede crear una nueva traducción de la siguiente forma, sustituyendo <LANG> por el identificador del idioma:

```
# git clone git://git.proxmox.com/git/proxmox-i18n.git
# cd proxmox-i18n
# make init-<LANG>.po
```

También puede editar una traducción existente utilizando el editor que prefiera:

```
# poedit <LANG>.po
```

1.13.2 Traducir sin git

Aunque no esté familiarizado con git, puede ayudar a traducir Proxmox VE. Para empezar, puede descargar los archivos de idioma. Busque el idioma que desee mejorar, haga clic con el botón derecho en el enlace "raw" de ese archivo de idioma y seleccione "Save Link As...". Realice los cambios y después envíe la traducción final directamente a office(at)proxmox.com junto con un contributor license agreement firmado.

1.13.3 Probar la traducción

Para que la traducción pueda utilizarse en Proxmox VE, primero debe convertir el archivo .po en un archivo .js. Puede hacerlo ejecutando el siguiente script, que se encuentra en el mismo repositorio:

```
# ./po2js.pl -t pve xx.po >pve-lang-xx.js
```

El archivo resultante pve-lang-xx.js puede copiarse al directorio /usr/share/pve-i18n del servidor Proxmox para probarlo.

Como alternativa, puede construir un paquete deb ejecutando el siguiente comando desde la raíz del repositorio:

```
# make deb
```

Nota: Para que cualquiera de estos métodos funcione, necesita tener instalados los siguientes paquetes Perl en el sistema. Para Debian/Ubuntu:

```
# apt-get install perl liblocale-po-perl libjson-perl
```

1.13.4 Enviar la traducción

Puede enviar la traducción terminada, archivo .po, al equipo de Proxmox en la dirección office(at)proxmox.com junto con un contributor license agreement firmado. Como alternativa, si tiene experiencia como desarrollador, puede enviarla como parche a la lista de correo de desarrollo de Proxmox VE. Consulte la Developer Documentation.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

RELEASE 9.2.4

4 de agosto de 2026

PARTE 2

Capítulo 2 · Instalación de Proxmox VE

Requisitos, medio de instalación, instalador y opciones avanzadas

Capítulo 2 · Instalación de Proxmox VE

Proxmox VE está basado en Debian GNU/Linux. Las imágenes de disco de instalación (archivos ISO) proporcionadas por Proxmox incluyen un sistema Debian completo, así como todos los paquetes necesarios de Proxmox VE.

Consejo: Consulte la tabla de soporte de las FAQ para conocer la relación entre las versiones de Proxmox VE y las versiones de Debian.

El instalador guía a través de la configuración, permitiendo particionar los discos locales, aplicar ajustes básicos del sistema -por ejemplo, zona horaria, idioma y red- e instalar todos los paquetes necesarios. Este proceso no debería tardar más que unos minutos. La instalación mediante la ISO proporcionada es el método recomendado tanto para usuarios nuevos como existentes.

Como alternativa, Proxmox VE puede instalarse sobre un sistema Debian ya existente. Esta opción solo se recomienda para usuarios avanzados, ya que requiere un conocimiento detallado de Proxmox VE.

2.1 Requisitos del sistema

Se recomienda utilizar hardware de servidor de alta calidad al ejecutar Proxmox VE en producción. Para reducir aún más el impacto de un fallo de host, puede ejecutar Proxmox VE en un clúster con máquinas virtuales y contenedores de alta disponibilidad (HA).

Proxmox VE puede utilizar almacenamiento local (DAS), SAN, NAS y almacenamiento distribuido como Ceph RBD. Para más detalles, consulte el capítulo de almacenamiento.

2.1.1 Requisitos mínimos, para evaluación

Estos requisitos mínimos son únicamente para evaluación y no deben utilizarse en producción.

- CPU: x86 de 64 bits (Intel 64 o AMD64). Como alternativa, ARM de 64 bits (arm64, aarch64) con una CPU ARMv8-A o posterior; consulte 2.1.3 Soporte ARM de 64 bits (arm64).
- Para hosts x86, CPU y placa base compatibles con Intel VT o AMD-V para virtualización KVM completa. En las plataformas arm64 compatibles, la virtualización asistida por hardware KVM no necesita un indicador de CPU equivalente.
- RAM: 1 GB de RAM, más la RAM adicional necesaria para los invitados.
- Disco duro.
- Una tarjeta de red (NIC).

2.1.2 Requisitos de sistema recomendados

- Intel 64 o AMD64 con indicador de CPU Intel VT o AMD-V. Como alternativa, un host ARM de 64 bits con una CPU ARMv9-A o posterior; consulte 2.1.3.
- Memoria: mínimo 2 GB para el sistema operativo y los servicios de Proxmox VE, más la memoria asignada a los invitados. Para Ceph y ZFS se necesita memoria adicional; aproximadamente 1 GB de memoria por cada TB de almacenamiento utilizado.
- Almacenamiento rápido y redundante; los mejores resultados se obtienen con SSD.
- Almacenamiento del sistema operativo: utilice RAID por hardware con caché de escritura protegida por batería (BBU), o una configuración sin RAID con ZFS (SSD opcional para ZIL).
- Almacenamiento de VMs:
  - Para almacenamiento local, utilice RAID por hardware con caché de escritura protegida por batería (BBU), o sin RAID para ZFS y Ceph. Ni ZFS ni Ceph son compatibles con un controlador RAID por hardware.
  - Es posible utilizar almacenamiento compartido y distribuido.
  - Se recomiendan SSD con Power-Loss-Protection (PLP) para obtener buen rendimiento. Se desaconsejan los SSD de consumo.
- NICs redundantes de varios Gbit, con NICs adicionales según la tecnología de almacenamiento y la configuración del clúster.
- Para passthrough PCI(e), la CPU y la placa base deben admitir una unidad de gestión de memoria de entrada/salida (IOMMU): Intel VT-d o AMD-Vi en x86, o System Memory Management Unit (SMMU) en arm64.

Consejo: Normalmente IOMMU también debe habilitarse en la BIOS/UEFI. Consulte el capítulo de passthrough PCI(e) para obtener más detalles.

2.1.3 Soporte ARM de 64 bits (arm64)

Proxmox VE admite ARM nativo de 64 bits (arm64, aarch64) en las plataformas NVIDIA Grace Hopper y NVIDIA Vera. En otro hardware basado en UEFI con ARMv9-A o posterior, el soporte se ofrece en modalidad best-effort. No se admiten ordenadores monoplaca que dependan exclusivamente de device tree, como Raspberry Pi.

Un host arm64 debe arrancar mediante UEFI y describir su hardware por medio de ACPI (Advanced Configuration and Power Interface). Proxmox VE no admite arranque únicamente mediante device tree ni BIOS heredada en arm64.

Varias funciones específicas de x86 no están disponibles en arm64:

- Las máquinas virtuales no pueden utilizar SeaBIOS. En arm64, los invitados siempre arrancan mediante UEFI, proporcionado por la compilación ARM de OVMF, también denominada AAVMF.
- El cifrado de memoria AMD SEV (Secure Encrypted Virtualization) y las vGPU mediadas GVT-g de Intel son exclusivos de x86.
- No existe un paquete de microcódigo de CPU a nivel de sistema operativo equivalente a intel-microcode y amd64-microcode de x86.

Un clúster puede contener nodos x86 y arm64, pero los invitados no pueden moverse entre nodos de arquitecturas distintas mientras están en ejecución:

- La migración en vivo solo es posible entre nodos de la misma arquitectura, porque el estado de CPU de un invitado en ejecución no puede transferirse entre arquitecturas.
- Un invitado, ya sea máquina virtual o contenedor, solo se ejecuta en un nodo cuya arquitectura coincida con la del invitado. La migración offline, el backup y restauración o el almacenamiento compartido pueden mover los datos de un invitado a un nodo con otra arquitectura, pero el invitado debe reinstalarse o reconfigurarse para esa arquitectura antes de iniciarse.

2.1.4 Vista general sencilla del rendimiento

Para obtener una visión general del rendimiento de CPU y disco duro en un sistema Proxmox VE ya instalado, ejecute la herramienta incluida pveperf.

Nota: Se trata únicamente de un benchmark muy rápido y general. Se recomiendan pruebas más detalladas, especialmente respecto al rendimiento de E/S del sistema.

2.1.5 Navegadores web compatibles para acceder a la interfaz web

Para acceder a la interfaz de usuario basada en web se recomienda utilizar uno de los siguientes navegadores:

- Firefox, una versión del año actual o la última Extended Support Release.
- Chrome, una versión del año actual.
- La versión de Microsoft Edge actualmente soportada.
- Safari, una versión del año actual.

Al acceder desde un dispositivo móvil, Proxmox VE mostrará una interfaz ligera basada en controles táctiles.

2.2 Preparar el medio de instalación

Descargue la imagen ISO del instalador desde:

```
https://www.proxmox.com/en/downloads/proxmox-virtual-environment/iso
```

El medio de instalación de Proxmox VE es una imagen ISO híbrida. Puede utilizarse de dos formas:

- Como archivo de imagen ISO listo para grabarse en un CD o DVD.
- Como archivo de imagen de sectores sin procesar (IMG), listo para copiarse a una memoria USB.

El uso de una memoria USB para instalar Proxmox VE es la opción recomendada porque es más rápida.

2.2.1 Preparar una memoria USB como medio de instalación

La memoria USB debe ser mayor que la imagen ISO.

Nota: No utilice UNetbootin. No funciona con la imagen de instalación de Proxmox VE.

Importante: Asegúrese de que la memoria USB no esté montada y de que no contenga datos importantes.

2.2.2 Instrucciones para GNU/Linux

En sistemas operativos tipo Unix utilice el comando dd para copiar la imagen ISO a la memoria USB. Primero determine el nombre correcto del dispositivo USB y después ejecute dd:

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

Nota: Sustituya /dev/XYZ por el nombre correcto del dispositivo y adapte la ruta del archivo de entrada (if).
Precaución: Tenga mucho cuidado para no sobrescribir el disco equivocado.

Encontrar el nombre correcto del dispositivo USB

Hay dos formas de averiguar el nombre de la memoria USB. La primera consiste en comparar las últimas líneas de la salida de dmesg antes y después de conectar la memoria. La segunda es comparar la salida de lsblk.

```
# lsblk
```

A continuación conecte la memoria USB y ejecute el comando de nuevo:

```
# lsblk
```

Aparecerá un dispositivo nuevo. Ese es el dispositivo que debe utilizar. Para mayor seguridad, compruebe que el tamaño indicado coincide con el de la memoria USB.

2.2.3 Instrucciones para macOS

Abra Terminal, por ejemplo buscándolo con Spotlight.

Convierta el archivo .iso a formato .dmg utilizando la opción convert de hdiutil:

```
# hdiutil convert proxmox-ve_*.iso -format UDRW -o proxmox-ve_*.dmg
```

Consejo: macOS suele añadir automáticamente .dmg al nombre del archivo de salida.

Para obtener la lista actual de dispositivos ejecute:

```
# diskutil list
```

Inserte ahora la memoria USB y vuelva a ejecutar el comando para determinar qué nodo de dispositivo se le ha asignado, por ejemplo /dev/diskX.

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

Nota: Sustituya X por el número de disco obtenido con el comando anterior.

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

Nota: El uso de rdiskX, en lugar de diskX, en el último comando es intencionado y aumenta la velocidad de escritura.

2.2.4 Instrucciones para Windows

Uso de Etcher

Etcher funciona directamente. Descárguelo desde https://etcher.io. La aplicación le guiará para seleccionar la ISO y la memoria USB.

Uso de Rufus

Rufus es una alternativa más ligera, pero debe utilizarse el modo DD. Descargue Rufus desde https://rufus.ie/. Puede instalarlo o utilizar la versión portable. Seleccione la unidad de destino y el archivo ISO de Proxmox VE.

Importante: Después de pulsar Start debe hacer clic en No en el diálogo que pregunta si desea descargar una versión diferente de GRUB. En el siguiente diálogo seleccione el modo DD.

2.3 Uso del instalador de Proxmox VE

La imagen ISO del instalador incluye lo siguiente:

- Sistema operativo completo Debian GNU/Linux de 64 bits.
- Instalador de Proxmox VE, que particiona los discos locales con ext4, XFS, BTRFS (technology preview) o ZFS e instala el sistema operativo.
- Kernel Linux de Proxmox VE con soporte KVM y LXC.
- Conjunto completo de herramientas para administrar máquinas virtuales, contenedores, el sistema host, clústeres y todos los recursos necesarios.
- Interfaz de administración basada en web.

Nota: Todos los datos existentes en las unidades seleccionadas se eliminarán durante la instalación. El instalador no añade entradas de arranque para otros sistemas operativos.

Inserte el medio de instalación preparado, por ejemplo una memoria USB o CD-ROM, y arranque desde él.

Consejo: Asegúrese de que el arranque desde el medio de instalación, por ejemplo USB, esté habilitado en el firmware del servidor. Secure Boot debe estar deshabilitado al arrancar un instalador anterior a Proxmox VE 8.1.

Tras seleccionar la entrada correcta, por ejemplo Boot from USB, aparece el menú de Proxmox VE y puede elegir entre las siguientes opciones:

Install Proxmox VE (Graphical)

Inicia la instalación normal.

Consejo: Es posible utilizar el asistente de instalación únicamente con el teclado. Los botones pueden activarse pulsando ALT junto con el carácter subrayado del botón correspondiente. Por ejemplo, ALT + N para pulsar Next.

Install Proxmox VE (Terminal UI)

Inicia el asistente de instalación en modo terminal. Proporciona la misma experiencia general que el instalador gráfico, pero suele ofrecer mejor compatibilidad con hardware muy antiguo o muy reciente.

Install Proxmox VE (Terminal UI, Serial Console)

Inicia el asistente en modo terminal y además configura el kernel Linux para utilizar el primer puerto serie de la máquina para entrada y salida. Puede utilizarse cuando la máquina es totalmente headless y solo dispone de consola serie.

Ambos modos utilizan la misma base de código para el proceso de instalación real, beneficiándose de más de una década de correcciones de errores y garantizando paridad funcional.

Consejo: La opción Terminal UI puede utilizarse si el instalador gráfico no funciona correctamente, por ejemplo por problemas de controladores. Consulte también 2.3.6 sobre el parámetro de kernel nomodeset.

Advanced Options: Install Proxmox VE (Graphical, Debug Mode)

Inicia la instalación en modo depuración. Se abrirá una consola en varios pasos de la instalación. Esto ayuda a depurar la situación si algo falla. Para salir de una consola de depuración pulse CTRL-D. Esta opción también puede utilizarse para arrancar un sistema live con todas las herramientas básicas disponibles; por ejemplo, para reparar un rpool ZFS degradado o corregir el bootloader de una instalación existente de Proxmox VE.

Advanced Options: Install Proxmox VE (Terminal UI, Debug Mode)

Igual que el modo gráfico de depuración, pero preparando el sistema para ejecutar el instalador basado en terminal.

Advanced Options: Install Proxmox VE (Serial Console Debug Mode)

Igual que el modo de depuración basado en terminal, pero además configura el kernel Linux para utilizar el primer puerto serie de la máquina para entrada y salida.

Advanced Options: Install Proxmox VE (Automated)

Inicia el instalador en modo desatendido incluso si la ISO no se ha preparado adecuadamente para una instalación automatizada. Esta opción puede utilizarse para recopilar detalles de hardware o para depurar una configuración de instalación automatizada. Consulte 2.4 Instalación desatendida para obtener más información.

Advanced Options: Rescue Boot

Permite arrancar una instalación existente. Busca en todos los discos duros conectados. Si encuentra una instalación existente, arranca directamente desde ese disco utilizando el kernel Linux de la ISO. Puede ser útil si existen problemas con el bootloader (GRUB/systemd-boot) o si la BIOS/UEFI no puede leer el bloque de arranque del disco.

Advanced Options: Test Memory (memtest86+)

Ejecuta memtest86+. Resulta útil para comprobar que la memoria funciona correctamente y no contiene errores. Secure Boot debe estar desactivado en la configuración de firmware UEFI para utilizar esta opción.

Nota: Esta entrada solo está disponible en medios de instalación x86, ya que memtest86+ no admite arm64.

Normalmente se selecciona Install Proxmox VE (Graphical) para iniciar la instalación.

El primer paso consiste en leer el EULA (End User License Agreement). Después puede seleccionar los discos duros de destino para la instalación.

Precaución: De forma predeterminada se utiliza todo el servidor y se eliminan todos los datos existentes. Asegúrese de que no exista información importante en el servidor antes de continuar.

El botón Options permite seleccionar el sistema de archivos de destino, cuyo valor predeterminado es ext4. El instalador utiliza LVM si selecciona ext4 o xfs, y ofrece opciones adicionales para limitar el espacio LVM.

Proxmox VE también puede instalarse sobre ZFS. Como ZFS ofrece varios niveles de RAID por software, es una opción para sistemas que no dispongan de controlador RAID por hardware. Los discos de destino deben seleccionarse en el diálogo Options. Otros ajustes específicos de ZFS pueden modificarse en Advanced Options.

Advertencia: ZFS sobre cualquier RAID por hardware no está soportado y puede provocar pérdida de datos.

La página siguiente solicita opciones básicas de configuración como ubicación, zona horaria y distribución de teclado. La ubicación se utiliza para seleccionar un servidor de descargas cercano y aumentar así la velocidad de las actualizaciones. El instalador normalmente puede detectar automáticamente estos ajustes, por lo que solo es necesario cambiarlos en situaciones poco habituales en las que falle la detección automática o cuando quiera utilizar una distribución de teclado poco común en su país.

También debe especificar la contraseña del superusuario (root) y una dirección de correo electrónico. La contraseña debe tener al menos 8 caracteres. Se recomienda encarecidamente utilizar una contraseña más robusta. Algunas recomendaciones son:

- Utilizar una longitud mínima de al menos 12 caracteres.
- Incluir caracteres alfabéticos en minúscula y mayúscula, números y símbolos.
- Evitar repetición de caracteres, patrones de teclado, palabras comunes de diccionario, secuencias de letras o números, nombres de usuario, nombres de familiares o mascotas, relaciones sentimentales actuales o pasadas e información biográfica, por ejemplo números de identificación, nombres de antepasados o fechas.

La dirección de correo se utiliza para enviar notificaciones al administrador del sistema. Por ejemplo:

- Información sobre actualizaciones de paquetes disponibles.
- Mensajes de error de trabajos cron periódicos.

El sistema envía todos los correos de notificación a la dirección indicada.

El último paso es la configuración de red. Las interfaces de red que están UP muestran un círculo relleno delante de su nombre en el menú desplegable. Durante la instalación puede especificar una dirección IPv4 o IPv6, pero no ambas. Para configurar un nodo dual-stack, añada las direcciones IP adicionales después de la instalación.

A continuación se muestra un resumen de las opciones seleccionadas anteriormente. Revise de nuevo cada ajuste y utilice el botón Previous si necesita cambiar alguno.

Después de hacer clic en Install, el instalador comienza a formatear los discos y copiar los paquetes a los discos de destino . Espere hasta que finalice; después retire el medio de instalación y reinicie el sistema.

La copia de paquetes normalmente tarda varios minutos, dependiendo principalmente de la velocidad del medio de instalación y del rendimiento del disco de destino.

Cuando termine la copia y configuración de los paquetes, puede reiniciar el servidor. De forma predeterminada, esto se realizará automáticamente después de unos segundos.

Fallo de instalación

Si la instalación falla, revise los errores concretos en la segunda TTY (CTRL + ALT + F2) y asegúrese de que el sistema cumple los requisitos mínimos.

Si la instalación sigue sin funcionar, consulte el capítulo sobre cómo obtener ayuda.

2.3.1 Acceso a la interfaz de administración después de la instalación

Tras una instalación correcta y el reinicio del sistema puede utilizar la interfaz web de Proxmox VE para continuar la configuración:

1. Dirija el navegador a la dirección IP indicada durante la instalación y al puerto 8006, por ejemplo: https://youripaddress:8006
2. Inicie sesión con el usuario root (realm PAM) y la contraseña elegida durante la instalación.
3. Suba su clave de suscripción para obtener acceso al repositorio Enterprise. En caso contrario, deberá configurar uno de los repositorios públicos, menos probados, para obtener actualizaciones de seguridad, correcciones de errores y nuevas funciones.
4. Compruebe la configuración IP y el nombre de host.
5. Compruebe la zona horaria.
6. Compruebe la configuración del Firewall.

2.3.2 Opciones avanzadas de configuración LVM

Si se utiliza ext4 o xfs, el instalador crea un Volume Group (VG) denominado pve y Logical Volumes (LV) adicionales denominados root, data y swap. Para controlar el tamaño de estos volúmenes utilice:

hdsize

Define el tamaño total de disco duro que se utilizará. De este modo puede reservar espacio libre en el disco para un particionado posterior, por ejemplo para un PV y VG adicionales en el mismo disco que puedan utilizarse para almacenamiento LVM.

swapsize

Define el tamaño del volumen swap. El valor predeterminado es el tamaño de la memoria instalada, con un mínimo de 4 GB y un máximo de 8 GB. El valor resultante no puede ser mayor que una octava parte del tamaño del disco duro (hdsize / 8).

Nota: Si se establece en 0, no se creará ningún volumen swap.

maxroot

Define el tamaño máximo del volumen root, que almacena el sistema operativo. Con más de 48 GiB disponibles, el valor predeterminado es una cuarta parte del tamaño del disco (hdsize / 4), con un máximo de 96 GiB. Con menos de 48 GiB disponibles, el tamaño del volumen root es al menos la mitad del tamaño del disco (hdsize / 2).

maxvz

Define el tamaño máximo del volumen data. El tamaño real del volumen data es:

```
datasize = hdsize - rootsize - swapsize - minfree
```

donde datasize no puede ser mayor que maxvz.

Nota: En el caso de LVM thin, el pool de datos solo se creará si datasize es mayor de 4 GB.
Nota: Si se establece en 0, no se creará ningún volumen data y la configuración de almacenamiento se adaptará en consecuencia.

minfree

Define la cantidad de espacio libre que debe dejarse en el grupo de volúmenes LVM pve. Con más de 128 GB disponibles, el valor predeterminado es 16 GB; en caso contrario se utiliza hdsize/8.

Nota: LVM necesita espacio libre en el VG para crear snapshots; no es necesario para snapshots lvmthin.

2.3.3 Opciones avanzadas de configuración ZFS

Si se utiliza ZFS, el instalador crea el pool ZFS rpool. No se crea espacio swap, aunque puede reservar espacio sin particionar en los discos de instalación para swap. También puede crear un zvol de swap después de la instalación, aunque esto puede ocasionar problemas; consulte las notas sobre swap en ZFS.

ashift

Define el valor ashift del pool creado. ashift debe establecerse como mínimo de acuerdo con el tamaño de sector de los discos subyacentes -2 elevado a ashift es el tamaño de sector- o de cualquier disco que pueda introducirse posteriormente en el pool, por ejemplo como sustitución de un disco defectuoso.

compress

Define si la compresión está habilitada para rpool.

checksum

Define qué algoritmo de checksum debe utilizarse para rpool.

copies

Define el parámetro copies para rpool. Consulte la página man zfs(8) para conocer su semántica y por qué no sustituye la redundancia a nivel de disco.

ARC max size

Define el tamaño máximo al que puede crecer ARC y, por tanto, limita la cantidad de memoria que utilizará ZFS. Consulte también la sección sobre cómo limitar el uso de memoria de ZFS para obtener más detalles.

hdsize

Define el tamaño total de disco duro que se utilizará. Resulta útil para reservar espacio libre en los discos para particiona do posterior, por ejemplo para crear una partición swap. hdsize solo se aplica a discos arrancables: únicamente el primer disco o mirror para RAID0, RAID1 o RAID10, y todos los discos en RAID-Z[123].

2.3.4 Opciones avanzadas de configuración BTRFS

No se crea espacio swap cuando se utiliza BTRFS, aunque puede reservar espacio sin particionar en los discos de instalación para swap. Puede crear una partición independiente, un subvolumen BTRFS o un swapfile mediante el comando btrfs filesystem mkswapfile.

compress

Define si la compresión está habilitada para el subvolumen BTRFS. Se admiten diferentes algoritmos de compresión: on (equivalente a zlib), zlib, lzo y zstd. El valor predeterminado es off.

hdsize

Define el tamaño total de disco duro que se utilizará. Resulta útil para reservar espacio libre en los discos para un particionado posterior, por ejemplo para crear una partición swap.

2.3.5 Consejos de rendimiento para ZFS

ZFS funciona mejor con mucha memoria. Si pretende utilizar ZFS, asegúrese de disponer de RAM suficiente. Una buena estimación es 4 GB más 1 GB de RAM por cada TB de espacio RAW en disco.

ZFS puede utilizar una unidad dedicada como caché de escritura, denominada ZFS Intent Log (ZIL). Utilice una unidad rápida, como un SSD. Puede añadirse después de la instalación con el siguiente comando:

```
# zpool add <pool-name> log </dev/path_to_fast_ssd>
```

2.3.6 Añadir el parámetro de kernel nomodeset

Pueden aparecer problemas en hardware muy antiguo o muy reciente debido a los controladores gráficos. Si la instalación se bloquea durante el arranque, puede intentar añadir el parámetro nomodeset. Esto impide que el kernel Linux cargue controladores gráficos y obliga a continuar utilizando el framebuffer proporcionado por BIOS/UEFI.

En el menú del bootloader de Proxmox VE, desplácese hasta Install Proxmox VE (Terminal UI) y pulse e para editar la entrada. Con las teclas de cursor, vaya a la línea que comienza por linux, mueva el cursor al final y añada el parámetro nomodeset separado por un espacio del último parámetro existente.

Después pulse Ctrl-X o F10 para arrancar esa configuración.

2.4 Instalación desatendida

El método de instalación automatizada permite instalar Proxmox VE sin intervención. Esto permite automatizar por completo el proceso de configuración sobre bare-metal. Una vez terminada la instalación y arrancado el host, pueden utilizarse herramientas de automatización como Ansible para continuar configurando la instalación.

Las opciones necesarias para el instalador deben proporcionarse en un archivo de respuestas. Este archivo permite utilizar reglas de filtrado para determinar qué discos y tarjetas de red deben utilizarse.

Para utilizar la instalación automatizada, primero elija una fuente desde la que obtener el archivo de respuestas y después prepare una ISO de instalación con esa elección.

Una vez preparada la ISO, el menú de arranque inicial mostrará una nueva entrada denominada Automated Installation, que se seleccionará automáticamente tras un timeout de 10 segundos.

Consulte la wiki para obtener más detalles e información sobre la instalación desatendida.

2.5 Instalar Proxmox VE sobre Debian

Proxmox VE se distribuye como un conjunto de paquetes Debian y puede instalarse sobre una instalación Debian estándar.

Después de configurar los repositorios debe ejecutar los siguientes comandos:

```
# apt-get update
# apt-get install proxmox-ve
```

Instalar sobre una instalación Debian existente parece sencillo, pero presupone que el sistema base se ha instalado correctamente y que sabe cómo desea configurar y utilizar el almacenamiento local. También debe configurar la red manualmente.

En general, esto no es trivial, especialmente cuando se utiliza LVM o ZFS.

En la wiki puede encontrarse una guía detallada paso a paso.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 - Traducción al castellano

PARTE 3A · Capítulo 3: Administración del sistema host

Secciones 3.1 a 3.7 · Repositorios, actualizaciones, firmware, red, tiempo, métricas y discos

Basado en “Proxmox VE Administration Guide”, Release 9.2.4
4 de agosto de 2026 · Proxmox Server Solutions GmbH

Aviso sobre esta parte

Esta Parte 3A continúa la traducción progresiva del manual oficial y cubre las secciones 3.1 a 3.7 del Capítulo 3, correspondientes a las páginas 28 a 56 de la numeración interna del documento original.

La traducción mantiene la organización, advertencias, comandos, rutas, nombres de paquetes, parámetros, ejemplos y nivel de detalle del original. Los comandos y bloques de configuración se conservan sin traducir. Los diagramas de red se han redibujado en castellano manteniendo el sentido técnico del original.

La edición completa se publicará como versión modificada bajo la GNU Free Documentation License (GFDL), conservando los avisos de copyright y la licencia original conforme a sus condiciones.

Contenido de la Parte 3A

- 3.1 Repositorios de paquetes
- 3.2 Actualizaciones del software del sistema
- 3.3 Actualizaciones de firmware
- 3.4 Configuración de red
- 3.5 Sincronización horaria
- 3.6 Servidor externo de métricas
- 3.7 Supervisión del estado de los discos

Capítulo 3 · Administración del sistema host

Las siguientes secciones se centran en tareas habituales de virtualización y explican las particularidades de Proxmox VE relacionadas con la administración y gestión de la máquina host.

Proxmox VE se basa en Debian GNU/Linux y añade repositorios propios para proporcionar los paquetes relacionados con Proxmox VE. Esto significa que está disponible toda la gama de paquetes de Debian, incluidas las actualizaciones de seguridad y las correcciones de errores. Proxmox VE proporciona su propio kernel Linux basado en el kernel de Ubuntu. Este kernel tiene habilitadas las funciones necesarias de virtualización y contenedores e incluye ZFS y diversos controladores de hardware adicionales.

Para otros temas no incluidos en las siguientes secciones, consulte la documentación de Debian. El Debian Administrator's Handbook está disponible en línea y ofrece una introducción completa al sistema operativo Debian (véase [Hertzog13]).

3.1 Repositorios de paquetes

Proxmox VE utiliza APT como herramienta de gestión de paquetes, al igual que cualquier otro sistema basado en Debian.

Proxmox VE comprueba automáticamente cada día si hay actualizaciones de paquetes. El usuario root@pam recibe por correo electrónico avisos sobre las actualizaciones disponibles. Desde la interfaz gráfica, el botón Changelog permite consultar más información sobre una actualización seleccionada.

3.1.1 Repositorios en Proxmox VE

Los repositorios son colecciones de paquetes de software. Se utilizan para instalar software nuevo y también son esenciales para recibir nuevas actualizaciones.

NOTA: Es necesario disponer de repositorios válidos de Debian y Proxmox para obtener las últimas actualizaciones de seguridad, correcciones de errores y nuevas funciones.

Los repositorios APT se definen en el archivo /etc/apt/sources.list con el formato heredado de una sola línea y en archivos .sources ubicados en /etc/apt/sources.list.d/ para el formato moderno multilínea deb822. Consulte «Formatos de repositorio» para más información.

Gestión de repositorios

Desde Proxmox VE 7, el estado de los repositorios puede comprobarse desde la interfaz web. El panel de resumen del nodo muestra una visión general del estado, mientras que el panel independiente Repositories muestra el estado detallado y la lista de todos los repositorios configurados.

También se admite la gestión básica de repositorios, por ejemplo, activarlos o desactivarlos.

Los paquetes disponibles en un repositorio se obtienen ejecutando apt update. Las actualizaciones pueden instalarse directamente mediante apt o desde la interfaz gráfica (Node -> Updates).

Formatos de repositorio

Los repositorios de paquetes pueden configurarse en /etc/apt/sources.list y en los archivos contenidos en /etc/apt/sources.list.d/. Se admiten dos formatos:

Una sola línea. En un archivo sources.list de una sola línea, cada línea define un repositorio de paquetes. Las líneas vacías se ignoran. El carácter # en cualquier punto de una línea hace que el resto de esa línea se considere un comentario. Es el formato heredado. Desde Debian 13 Trixie, apt mostrará avisos si se utiliza este formato. La mayoría de los repositorios pueden migrarse automáticamente con el comando apt modernize-sources.

deb822. En el formato multilínea de los archivos repo.sources, cada entrada está formada por varias líneas con pares clave-valor. Un archivo puede incluir varias entradas separando cada grupo con una línea en blanco. Este es el formato moderno.

Repositorios disponibles

Además de requerir los repositorios base de Debian, Proxmox VE proporciona tres repositorios de paquetes diferentes.

3.1.2 Repositorios base de Debian

Archivo: /etc/apt/sources.list.d/debian.sources

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

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

3.1.3 Repositorio Proxmox VE Enterprise

Es el repositorio recomendado y está disponible para todos los usuarios con una suscripción de Proxmox VE. Contiene los paquetes más estables y es adecuado para entornos de producción. El repositorio pve-enterprise está activado de forma predeterminada:

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

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

Para acceder al repositorio pve-enterprise es necesaria una clave de suscripción válida. Proxmox ofrece distintos niveles de soporte; puede consultar sus detalles en https://proxmox.com/en/proxmox-virtual-environment/pricing.

NOTA: Las claves de suscripción están vinculadas a la arquitectura del host. Un host arm64 necesita una clave de suscripción específica para arm64; una clave x86 no es válida para él, y viceversa.

NOTA: Puede desactivar este repositorio añadiendo una línea Enabled: no a la entrada correspondiente del archivo. Esto evitará mensajes de error si el host no dispone de una clave de suscripción. En ese caso, configure el repositorio pve-no-subscription.

3.1.4 Repositorio Proxmox VE No-Subscription

Como indica su nombre, no se necesita una clave de suscripción para acceder a este repositorio. Puede utilizarse para pruebas y para usos que no sean de producción. No se recomienda utilizarlo en servidores de producción, ya que estos paquetes no siempre se han probado y validado con el mismo nivel de intensidad.

Se recomienda configurar este repositorio en /etc/apt/sources.list.d/proxmox.sources.

Archivo: /etc/apt/sources.list.d/proxmox.sources

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

NOTA: Recuerde que siempre necesitará los repositorios base de Debian además de un repositorio de Proxmox VE.

3.1.5 Repositorio de pruebas de Proxmox VE

Este repositorio contiene los paquetes más recientes y se utiliza principalmente por los desarrolladores para probar nuevas funciones. Para configurarlo, añada la siguiente entrada al archivo /etc/apt/sources.list.d/proxmox.sources:

Archivo: /etc/apt/sources.list.d/proxmox.sources

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

ADVERTENCIA: El repositorio pve-test, como su nombre indica, solo debería utilizarse para probar nuevas funciones o correcciones de errores.

3.1.6 Repositorio de símbolos de depuración de Proxmox VE

Los paquetes de símbolos de depuración, denominados <package>-dbgsym, contienen la información de depuración separada correspondiente al paquete binario. No son necesarios para el funcionamiento normal, pero permiten que herramientas como gdb, systemd-coredump o crash generen trazas útiles al analizar un bloqueo o una suspensión. Se proporcionan en un repositorio dedicado, independiente de los repositorios normales, para las versiones de Proxmox VE basadas en Debian Bookworm y Trixie, y posteriores.

Archivo: /etc/apt/sources.list.d/pve-dbgsym.sources

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

Después de activar el repositorio, instale el paquete -dbgsym que corresponda al binario que desea depurar.

3.1.7 Repositorios Ceph

Los paquetes relacionados con Ceph se mantienen actualizados mediante un repositorio Enterprise de Ceph preconfigurado. Los paquetes preinstalados permiten conectarse a un clúster Ceph externo y añadir sus pools RBD o CephFS como almacenamiento. También es posible construir una infraestructura hiperconvergente (HCI) ejecutando Ceph sobre el clúster Proxmox VE.

La información de este apartado resulta útil en los siguientes casos:

Instalación de Ceph para construir una HCI. Elija uno de los repositorios descritos a continuación y una versión reciente de Ceph; después podrá seleccionarlos en el asistente web o mediante la herramienta CLI.

HCI ya existente y actualización a la siguiente versión principal de Ceph. Siga la guía de actualización de Ceph correspondiente.

HCI ya existente y actualización a la siguiente versión principal de Proxmox VE. En una HCI, cada versión principal de Proxmox VE requiere una versión principal mínima de Ceph; siga la guía de actualización de Proxmox VE correspondiente.

Sin HCI, pero utilizando un clúster Ceph externo. Para instalar paquetes más recientes de conexión a Ceph, aplique las actualizaciones del sistema disponibles, elija uno de los repositorios y versiones de Ceph indicados a continuación, añádalo al nodo desde el panel Repositories, aplique las nuevas actualizaciones disponibles, verifique el resultado ejecutando ceph --version y desactive el repositorio Ceph antiguo.

Versiones de Ceph disponibles en Proxmox VE 9

Para consultar la versión más reciente de la guía de administración correspondiente a su versión de Proxmox VE, asegúrese de que todas las actualizaciones del sistema estén instaladas y vuelva a cargar la página.

Versión	Fin de vida estimado	enterprise	no-subscription	test
ceph-tentacle	2027-11 (v20.2)	recomendado	disponible	disponible
ceph-squid	2026-09 (v19.2)	disponible	disponible	disponible

Repositorios Ceph para Proxmox VE 9

El contenido del archivo ceph.sources que se muestra a continuación sirve como referencia (antes de Proxmox VE 9 se utilizaba el archivo ceph.list). Para realizar cambios, siga el caso aplicable a su situación descrito al comienzo de este subcapítulo. Si ha desactivado un repositorio desde la interfaz web y también desea eliminarlo de la lista, puede borrar manualmente la entrada correspondiente del archivo.

enterprise. Este repositorio se recomienda para producción y contiene las versiones de paquetes más estables. Es accesible si el nodo Proxmox VE tiene una suscripción válida de cualquier nivel. Para conocer los detalles y niveles de soporte incluidos, consulte https://proxmox.com/en/proxmox-virtual-environment/pricing.

Archivo: /etc/apt/sources.list.d/ceph.sources

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

no-subscription. Este repositorio es adecuado para pruebas y usos que no sean de producción. Es de acceso libre y no requiere una suscripción válida. Transcurrido un tiempo, sus versiones de paquetes también pasan a estar disponibles en el repositorio Enterprise.

Archivo: /etc/apt/sources.list.d/ceph.sources

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

test. Este repositorio contiene las versiones de paquetes más recientes y se utiliza principalmente por los desarrolladores para probar nuevas funciones y correcciones de errores.

Archivo: /etc/apt/sources.list.d/ceph.sources

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

ADVERTENCIA: El repositorio de pruebas de Ceph solo debería utilizarse para probar nuevas funciones o correcciones de errores.

3.1.8 Repositorio de firmware de Debian

A partir de Debian Bookworm (Proxmox VE 8), el firmware no libre, según la definición de las DFSG, se trasladó al nuevo componente de repositorio de Debian non-free-firmware.

Desde Proxmox VE 9, este repositorio está habilitado de forma predeterminada en las instalaciones nuevas para que puedan recibir actualizaciones tempranas de microcódigo del sistema operativo.

También permite obtener archivos de firmware en tiempo de ejecución adicionales que no estén ya incluidos en el paquete preinstalado pve-firmware.

Para poder instalar paquetes de este componente en configuraciones antiguas de repositorios, ejecute editor /etc/apt/sources.list, añada non-free-firmware al final de cada línea de repositorio .debian.org y ejecute apt update.

Si actualizó su instalación de Proxmox VE 9 desde una versión anterior y ya modernizó los repositorios al formato deb822, deberá adaptar /etc/apt/sources.list.d/debian.sources. Ejecute editor /etc/apt/sources.list.d/debian.sources y añada non-free-firmware a las líneas que empiezan por Components: en cada bloque.

NOTA: Se recomienda modernizar los repositorios de paquetes. De lo contrario, apt en Debian Trixie mostrará avisos. Puede hacerlo ejecutando apt modernize-sources.

3.1.9 SecureApt

Los archivos Release de los repositorios están firmados con GnuPG. APT utiliza estas firmas para verificar que todos los paquetes proceden de una fuente de confianza.

Si instala Proxmox VE desde una imagen ISO oficial, la clave de verificación ya está instalada.

Si instala Proxmox VE sobre Debian, descargue e instale la clave mediante el siguiente comando:

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

NOTA: El comando wget anterior añade el llavero para las versiones de Proxmox basadas en Debian Trixie. Una vez instalado el paquete proxmox-archive-keyring, este gestionará el archivo. A partir de ese momento, los hashes que aparecen a continuación pueden dejar de coincidir con el archivo, ya que se añadirán o eliminarán claves para nuevas versiones de Proxmox. Es el comportamiento previsto: apt garantizará que solo se utilicen claves de confianza. Se desaconseja modificar el archivo una vez instalado proxmox-archive-keyring.

Compruebe después la suma de verificación con la herramienta de línea de comandos sha512sum:

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

O con md5sum:

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

NOTA: Asegúrese de que la ruta en la que instala la clave coincida con las líneas Signed-By: de las entradas de sus repositorios.

3.2 Actualizaciones del software del sistema

Proxmox publica actualizaciones periódicamente para todos los repositorios. Para instalarlas, utilice la interfaz web o los siguientes comandos:

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

ADVERTENCIA: Como Proxmox VE utiliza un modelo de publicación continua sobre Debian, ejecutar apt upgrade puede dejar el sistema en un estado de paquetes parcialmente actualizado o inconsistente. El motivo es que apt full-upgrade puede eliminar paquetes existentes si es necesario para satisfacer dependencias, mientras que apt upgrade no puede hacerlo.

Para actualizar ocasionalmente Ceph a su siguiente versión principal, consulte el apartado «Repositorios Ceph».

NOTA: El sistema de gestión de paquetes APT es muy flexible y ofrece muchas funciones; consulte man apt o [Hertzog13] para obtener información adicional.

CONSEJO: Las actualizaciones periódicas son esenciales para obtener los últimos parches y correcciones de seguridad. Las actualizaciones principales del sistema se anuncian en el foro de la comunidad de Proxmox VE.

3.3 Actualizaciones de firmware

Las actualizaciones de firmware descritas en este capítulo deben aplicarse cuando Proxmox VE se ejecuta en un servidor bare metal. La conveniencia de configurar actualizaciones de firmware dentro de los invitados, por ejemplo al utilizar passthrough de dispositivos, depende en gran medida de cada configuración y queda fuera del alcance de esta sección.

Además de las actualizaciones habituales de software, las actualizaciones de firmware también son importantes para un funcionamiento fiable y seguro. Al obtener y aplicar firmware se recomienda combinar las opciones disponibles para recibir las actualizaciones lo antes posible, o para poder recibirlas en absoluto.

El término firmware suele dividirse, en la práctica, en microcódigo para las CPU y firmware para otros dispositivos.

3.3.1 Firmware persistente

Este apartado es aplicable a todos los dispositivos. El microcódigo actualizado, normalmente incluido en una actualización de BIOS/UEFI, se almacena en la placa base, mientras que el resto del firmware se almacena en el dispositivo correspondiente. Este método persistente es especialmente importante para la CPU, ya que permite cargar el microcódigo actualizado lo antes posible durante el arranque.

PRECAUCIÓN: Con algunas actualizaciones, por ejemplo de BIOS/UEFI o del controlador de almacenamiento, la configuración del dispositivo puede restablecerse. Siga cuidadosamente las instrucciones del fabricante y realice una copia de seguridad de la configuración actual.

Consulte con el fabricante qué métodos de actualización están disponibles.

- Entre los métodos cómodos de actualización para servidores se encuentran Dell Lifecycle Manager o los Service Packs de HPE.
- En algunos casos también existen utilidades para Linux, como mlxup para NVIDIA ConnectX o bnxtnvm/niccli para tarjetas de red Broadcom.
- LVFS también es una opción cuando existe colaboración con el fabricante y se utiliza hardware compatible. El requisito técnico es que el sistema haya sido fabricado después de 2014 y arranque mediante UEFI.

Proxmox VE incluye su propia versión del paquete fwupd para habilitar la compatibilidad con Secure Boot usando la clave de firma de Proxmox. En esta versión se eliminó deliberadamente la recomendación de dependencia del paquete udisks2 debido a problemas observados al utilizarlo en hipervisores.

CONSEJO: Si las instrucciones de actualización requieren reiniciar el host, asegúrese de que el reinicio pueda realizarse de forma segura. Consulte también el apartado sobre mantenimiento del nodo.

3.3.2 Archivos de firmware en tiempo de ejecución

Este método almacena el firmware en el sistema operativo Proxmox VE y lo entrega al dispositivo si el firmware persistente de este es más antiguo. Es compatible con dispositivos como tarjetas de red y tarjetas gráficas, pero no con aquellos que dependen de firmware persistente, como la placa base y los discos duros.

En Proxmox VE, el paquete pve-firmware ya está instalado de forma predeterminada. Por tanto, mediante las actualizaciones normales del sistema con APT, el firmware incluido para hardware común se mantiene automáticamente al día.

Existe además un repositorio adicional de firmware de Debian, pero no está configurado de forma predeterminada. Si intenta instalar un paquete de firmware adicional que entra en conflicto con otro paquete, APT cancelará la instalación. Puede que ese firmware concreto pueda obtenerse de otra forma.

3.3.3 Actualizaciones de microcódigo de CPU

NOTA: Las actualizaciones de microcódigo de CPU descritas en este apartado son específicas de hosts x86. En arm64 no existe un paquete equivalente de microcódigo a nivel de sistema operativo.

Las actualizaciones de microcódigo están destinadas a corregir vulnerabilidades de seguridad descubiertas y otros errores graves de la CPU. Aunque el rendimiento de la CPU puede verse afectado, un microcódigo corregido suele seguir ofreciendo mejor rendimiento que un microcódigo sin corregir cuando el propio kernel debe aplicar mitigaciones. Dependiendo del tipo de CPU, es posible que el rendimiento del estado de fábrica defectuoso ya no pueda alcanzarse sin ejecutar deliberadamente la CPU en un estado inseguro.

Para obtener una visión general de las vulnerabilidades presentes en la CPU y de sus mitigaciones, ejecute lscpu. Las vulnerabilidades reales conocidas en ese momento solo pueden aparecer correctamente si el host Proxmox VE está actualizado, su versión no ha llegado al fin de vida y se ha reiniciado al menos una vez desde la última actualización del kernel.

Además de la actualización recomendada del microcódigo mediante actualizaciones persistentes de BIOS/UEFI, existe un método independiente denominado Early OS Microcode Updates. Es cómodo de utilizar y especialmente útil cuando el fabricante de la placa base ya no proporciona actualizaciones de BIOS/UEFI. Con independencia del método utilizado, siempre es necesario reiniciar para aplicar una actualización de microcódigo.

Configurar actualizaciones tempranas de microcódigo del sistema operativo

Para configurar actualizaciones de microcódigo que el kernel Linux aplique al comienzo del arranque:

7. Habilite el repositorio de firmware de Debian.
8. Obtenga los paquetes disponibles más recientes con apt update, o utilice la interfaz web en Node -> Updates.
9. Instale el paquete de microcódigo correspondiente al fabricante de la CPU: Intel: apt install intel-microcode; AMD: apt install amd64-microcode.
10. Reinicie el host Proxmox VE.

Cualquier futura actualización de microcódigo también requerirá un reinicio para poder cargarse.

Versión del microcódigo

Para consultar la revisión de microcódigo actualmente en ejecución, por ejemplo para compararla o depurar un problema:

```
# grep microcode /proc/cpuinfo | uniq
microcode : 0xf0
```

Un paquete de microcódigo contiene actualizaciones para muchas CPU diferentes, pero las actualizaciones específicas para su CPU pueden ser poco frecuentes. Por ello, la fecha del paquete no indica necesariamente cuándo publicó el fabricante una actualización para su modelo de CPU concreto.

Si ha instalado un paquete de microcódigo nuevo y ha reiniciado el host Proxmox VE, y ese microcódigo es más reciente tanto que el integrado en la CPU como que el proporcionado por el firmware de la placa base, el registro del sistema mostrará un mensaje indicando que el microcódigo se actualizó de forma temprana.

```
# dmesg | grep microcode
[    0.000000] microcode: microcode updated early to revision 0xf0, date = 2021-11-12
[    0.896580] microcode: Microcode Update Driver: v2.2.
```

Solución de problemas

Para tareas de depuración, la actualización temprana de microcódigo aplicada normalmente durante el arranque puede desactivarse temporalmente del siguiente modo:

11. Asegúrese de que el host puede reiniciarse de forma segura.
12. Reinicie el host para acceder al menú GRUB; mantenga pulsada SHIFT si está oculto.
13. En la entrada de arranque de Proxmox VE que desee, pulse E.
14. Vaya a la línea que comienza por linux y añada, separado por un espacio, dis_ucode_ldr.
15. Pulse CTRL-X para arrancar esta vez sin aplicar una actualización temprana de microcódigo.

Si sospecha de un problema relacionado con una actualización reciente de microcódigo, debería considerar degradar el paquete en lugar de eliminarlo con apt purge <intel-microcode|amd64-microcode>. De lo contrario, podría cargarse un microcódigo persistente demasiado antiguo aunque una versión más reciente funcionase correctamente.

Es posible realizar una degradación si en el repositorio de Debian existe una versión anterior del paquete de microcódigo. Ejemplo:

```
# apt list -a intel-microcode
Listing... Done
intel-microcode/stable-security,now 3.20230808.1~deb12u1 amd64 [installed]
intel-microcode/stable 3.20230512.1 amd64

# apt install intel-microcode=3.202305*
...
Selected version '3.20230512.1' (Debian:12.1/stable [amd64]) for 'intel-microcode'
...
dpkg: warning: downgrading intel-microcode from 3.20230808.1~deb12u1 to 3.20230512.1
...
intel-microcode: microcode will be updated at next boot
...
```

Asegúrese de nuevo de que el host puede reiniciarse de forma segura. Para aplicar un microcódigo más antiguo incluido en el paquete para su tipo de CPU, reinicie el sistema.

CONSEJO: Puede ser razonable mantener retenido durante un tiempo el paquete degradado e intentar versiones más recientes posteriormente. Incluso aunque la versión del paquete sea la misma en el futuro, otras actualizaciones del sistema podrían haber resuelto entretanto el problema experimentado.

```
# apt-mark hold intel-microcode
intel-microcode set on hold.

# apt-mark unhold intel-microcode
# apt update
# apt full-upgrade
```

3.4 Configuración de red

Proxmox VE utiliza la pila de red de Linux. Esto proporciona mucha flexibilidad para configurar la red de los nodos Proxmox VE. La configuración puede realizarse desde la interfaz gráfica o editando manualmente /etc/network/interfaces, archivo que contiene toda la configuración de red. La página de manual interfaces(5) describe el formato completo. Las herramientas de Proxmox VE intentan preservar las modificaciones realizadas directamente por el usuario, aunque sigue siendo preferible utilizar la interfaz gráfica, ya que ayuda a evitar errores.

Se necesita una interfaz de puente Linux, normalmente denominada vmbrX, para conectar los invitados con la red física subyacente. Puede entenderse como un switch virtual al que se conectan tanto los invitados como las interfaces físicas. Esta sección muestra ejemplos de configuración para distintos casos de uso, como redundancia mediante bond, VLAN, redes enrutadas y configuraciones NAT.

Para redes virtuales más complejas en clústeres Proxmox VE puede utilizarse la funcionalidad Software Defined Network (SDN).

ADVERTENCIA: Si no está seguro de su funcionamiento, se desaconseja utilizar las herramientas tradicionales de Debian ifup e ifdown. Presentan algunos riesgos, por ejemplo: ifdown vmbrX puede interrumpir todo el tráfico de los invitados y un posterior ifup sobre el mismo puente no vuelve necesariamente a conectar esos invitados.

3.4.1 Aplicar cambios de red

Proxmox VE no escribe los cambios directamente en /etc/network/interfaces. En su lugar, los guarda en un archivo temporal llamado /etc/network/interfaces.new. De este modo pueden realizarse varios cambios relacionados de una sola vez y comprobarse antes de aplicarlos, algo especialmente importante porque una configuración de red incorrecta puede dejar el nodo inaccesible.

Recarga en caliente de la red con ifupdown2

Con el paquete recomendado ifupdown2, predeterminado en las instalaciones nuevas desde Proxmox VE 7.0, es posible aplicar cambios de configuración de red sin reiniciar. Si modifica la configuración desde la GUI, puede pulsar Apply Configuration. Esto mueve los cambios desde el archivo temporal interfaces.new a /etc/network/interfaces y los aplica en vivo.

Si realizó cambios manuales directamente en /etc/network/interfaces, puede aplicarlos ejecutando:

```
ifreload -a
```

NOTA: Si instaló Proxmox VE sobre Debian, o actualizó a Proxmox VE 7.0 desde una instalación anterior, asegúrese de que ifupdown2 esté instalado: apt install ifupdown2.

Reiniciar el nodo para aplicar los cambios

Otra forma de aplicar una nueva configuración de red es reiniciar el nodo. En ese caso, el servicio systemd pvenetcommit activará el archivo temporal interfaces.new antes de que el servicio de red aplique la configuración.

3.4.2 Convenciones de nombres

Actualmente se utilizan las siguientes convenciones para los nombres de dispositivos:

- Dispositivos Ethernet: en*, nombres de interfaces de red de systemd. Este esquema se utiliza en nuevas instalaciones de Proxmox VE desde la versión 5.0.
- Dispositivos Ethernet: eth[N], donde N >= 0 (eth0, eth1, ...). Este esquema se utiliza en hosts Proxmox VE instalados antes de la versión 5.0. Al actualizar a 5.0, los nombres se conservan.
- Puentes: normalmente vmbr[N], donde 0 <= N <= 4094 (vmbr0 - vmbr4094), aunque puede utilizar cualquier cadena alfanumérica que comience por una letra y tenga un máximo de 10 caracteres.
- Bonds: bond[N], donde N >= 0 (bond0, bond1, ...).
- VLAN: añada simplemente el número de VLAN al nombre del dispositivo, separado por un punto (eno1.50, bond1.30).

Esta convención facilita la depuración de problemas de red, ya que el nombre del dispositivo permite inferir su tipo.

Nombres de interfaces de red de systemd

systemd define un esquema versionado para los nombres de dispositivos de red. El esquema utiliza el prefijo de dos caracteres en para dispositivos Ethernet. Los caracteres siguientes dependen del controlador, la ubicación del dispositivo y otros atributos. Algunos patrones posibles son:

- o<index>[n<phys_port_name>|d<dev_port>] - dispositivos integrados en placa.
- s<slot>[f<function>][n<phys_port_name>|d<dev_port>] - dispositivos identificados por hotplug.
- [P<domain>]p<bus>s<slot>[f<function>][n<phys_port_name>|d<dev_port>] - dispositivos identificados por bus.
- x<MAC> - dispositivos identificados por dirección MAC.

Ejemplos habituales: eno1 es la primera NIC integrada en placa; enp3s0f1 es la función 1 de la NIC situada en el bus PCI 3, ranura 0. Para la lista completa de patrones consulte systemd.net-naming-scheme(7).

Una versión nueva de systemd puede definir una versión nueva del esquema de nombres y utilizarla de forma predeterminada. Por ello, una actualización a una versión más reciente de systemd, por ejemplo durante una actualización principal de Proxmox VE, puede cambiar los nombres de los dispositivos de red y requerir ajustes en la configuración. Para evitar cambios provocados por una nueva versión del esquema puede fijarse manualmente una versión concreta.

Aun fijando una versión del esquema, los nombres pueden cambiar por actualizaciones del kernel o de controladores. Para impedir completamente el cambio de nombre de un dispositivo concreto, puede sobrescribirlo manualmente mediante un archivo .link.

Fijar una versión concreta del esquema de nombres

Puede fijar una versión específica del esquema añadiendo el parámetro net.naming-scheme=<version> a la línea de comandos del kernel. Consulte systemd.net-naming-scheme(7) para conocer las versiones disponibles.

Por ejemplo, para fijar v252, que era la versión más reciente del esquema en una instalación nueva de Proxmox VE 8.0:

```
net.naming-scheme=v252
```

Consulte también el apartado dedicado a editar la línea de comandos del kernel. Es necesario reiniciar para que el cambio surta efecto.

Sobrescribir nombres de dispositivos de red con pve-network-interface-pinning

Proxmox VE proporciona una herramienta que genera automáticamente archivos .link para sobrescribir el nombre de los dispositivos de red. Además, sustituye automáticamente las apariciones del nombre antiguo de la interfaz en los siguientes archivos:

- /etc/network/interfaces
- /etc/pve/nodes/<nodename>/host.fw
- /etc/pve/sdn/controllers.cfg
- /etc/pve/sdn/fabrics.cfg

NOTA: Como el mapeo generado es local al nodo en el que se crea, los nombres de interfaz contenidos en la configuración de firewall del Datacenter (/etc/pve/firewall/cluster.fw) no se actualizan automáticamente.

Los archivos .link generados se almacenan en /usr/local/lib/systemd/network. Para los archivos de configuración se genera un nuevo archivo en la misma ubicación con el sufijo .new. Así puede inspeccionar los cambios con diff, o con otro visor de diferencias:

```
diff -y /etc/network/interfaces /etc/network/interfaces.new
```

Si detecta cambios problemáticos o desea revertir lo realizado por la herramienta antes de reiniciar, elimine todos los archivos .new y los archivos .link correspondientes de /usr/local/lib/systemd/network.

El siguiente comando genera un archivo .link para todas las interfaces físicas que aún no dispongan de uno y actualiza determinados archivos de configuración de Proxmox VE. Los nombres generados utilizan por defecto el prefijo nic, de forma que las interfaces resultantes serán nic1, nic2, etc.:

```
pve-network-interface-pinning generate
```

Puede cambiar el prefijo predeterminado con --prefix:

```
pve-network-interface-pinning generate --prefix myprefix
```

También puede fijar únicamente una interfaz concreta:

```
pve-network-interface-pinning generate --interface enp1s0
```

Al fijar una interfaz concreta puede especificar el nombre exacto que desea asignarle:

```
pve-network-interface-pinning generate --interface enp1s0 --target-name if42
```

Para aplicar a la configuración de red los cambios realizados por pve-network-interface-pinning, es necesario reiniciar el nodo.

Creación manual de archivos .link

Puede asignar manualmente un nombre a un dispositivo de red concreto mediante un archivo systemd.link personalizado. Esto sobrescribe el nombre que se asignaría según la versión más reciente del esquema de nombres y permite evitar cambios debidos a actualizaciones del kernel, de controladores o del propio esquema.

Los archivos .link personalizados deben colocarse en /etc/systemd/network/ y denominarse <n>-<id>.link, donde n es una prioridad inferior a 99 e id es un identificador. Un archivo link contiene dos secciones: [Match] determina a qué interfaces se aplicará; [Link] define cómo se configurarán esas interfaces, incluido su nombre.

Para asignar un nombre a un dispositivo concreto es necesario identificarlo de forma única y permanente en [Match]. Una posibilidad es utilizar su dirección MAC con la opción MACAddress, ya que es poco probable que cambie.

La sección [Match] también debería incluir Type para asegurarse de que solo coincide con la interfaz física prevista y no con interfaces bridge, bond o VLAN que compartan la misma MAC. En la mayoría de configuraciones, Type debe establecerse en ether para seleccionar únicamente dispositivos Ethernet, aunque algunos escenarios pueden requerir otros valores. Consulte systemd.link(5).

A continuación puede asignarse el nombre mediante la opción Name de la sección [Link]. Los archivos link se copian al initramfs, por lo que se recomienda actualizarlo después de añadir, modificar o eliminar uno:

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

Ejemplo para asignar el nombre enwan0 al dispositivo Ethernet con dirección MAC aa:bb:cc:dd:ee:ff, creando /etc/systemd/network/10-enwan0.link:

```
[Match]
MACAddress=aa:bb:cc:dd:ee:ff
Type=ether

[Link]
Name=enwan0
```

No olvide adaptar /etc/network/interfaces para utilizar el nuevo nombre y actualizar el initramfs como se ha descrito. Debe reiniciar el nodo para aplicar el cambio.

NOTA: Se recomienda asignar un nombre que comience por en o eth para que Proxmox VE reconozca la interfaz como un dispositivo de red físico configurable desde la GUI. También debe evitarse que el nombre pueda entrar en conflicto con otras interfaces futuras. Una opción es escoger un nombre que no coincida con los patrones que utiliza systemd, como enwan0 en el ejemplo anterior.

Para más información sobre archivos link, consulte systemd.link(5).

3.4.3 Elección de una configuración de red

Según la organización de su red y los recursos disponibles, puede elegir entre una configuración con bridge, enrutada o con masquerading.

Servidor Proxmox VE en una LAN privada con una puerta de enlace externa

En este caso, el modelo con bridge es el más adecuado y es también el modo predeterminado de las nuevas instalaciones de Proxmox VE. Cada sistema invitado dispone de una interfaz virtual conectada al bridge de Proxmox VE. El efecto es similar a conectar directamente la tarjeta de red del invitado a un switch de la LAN, siendo el host Proxmox VE el que desempeña el papel del switch.

Servidor Proxmox VE en un proveedor de hosting con rangos de IP públicas para los invitados

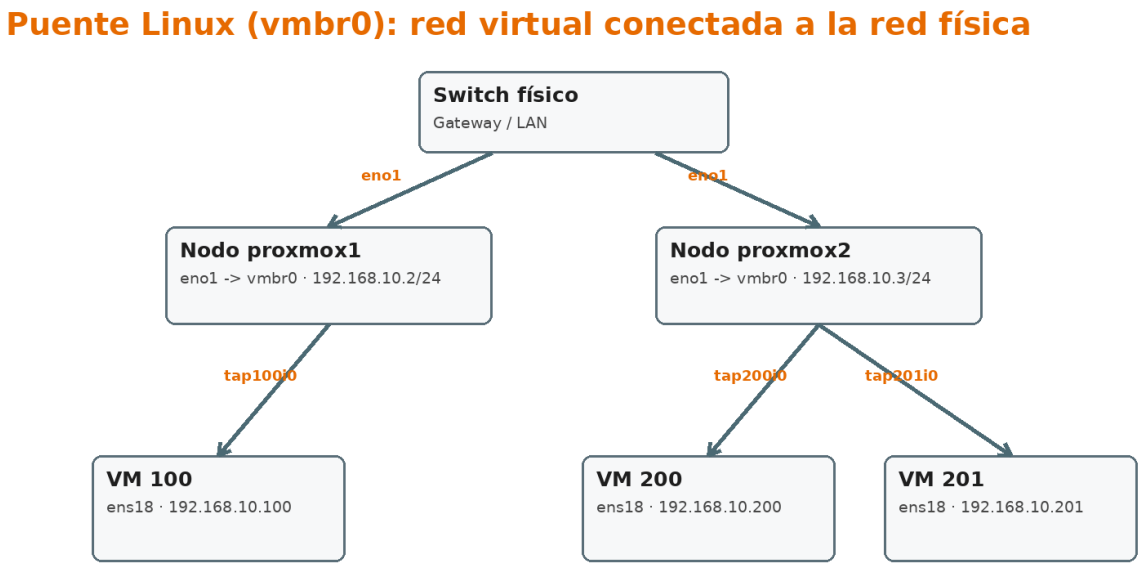
Puede utilizar un modelo con bridge o un modelo enrutado, dependiendo de lo que permita el proveedor.

Servidor Proxmox VE en un proveedor de hosting con una única dirección IP pública

En este caso, la única forma de proporcionar acceso saliente a la red a los sistemas invitados es utilizar masquerading. Para acceso entrante hacia los invitados será necesario configurar redirección de puertos (Port Forwarding).

Para disponer de mayor flexibilidad, también puede configurar VLAN (IEEE 802.1Q) y bonding de red, también denominado link aggregation. De este modo es posible construir redes virtuales complejas y flexibles.

3.4.4 Configuración predeterminada mediante un bridge



Esquema traducido: bridge Linux vmbr0 conectando invitados con la red física.

Los bridges son equivalentes a switches de red físicos implementados por software. Todos los invitados virtuales pueden compartir un único bridge, o pueden crearse varios bridges para separar dominios de red. Cada host puede tener hasta 4094 bridges.

El programa de instalación crea un único bridge denominado vmbr0 conectado a la primera tarjeta Ethernet. La configuración correspondiente en /etc/network/interfaces podría ser:

```
auto lo
iface lo inet loopback

iface eno1 inet manual

auto vmbr0
iface vmbr0 inet static
        address 192.168.10.2/24
        gateway 192.168.10.1
        bridge-ports eno1
        bridge-stp off
        bridge-fd 0
```

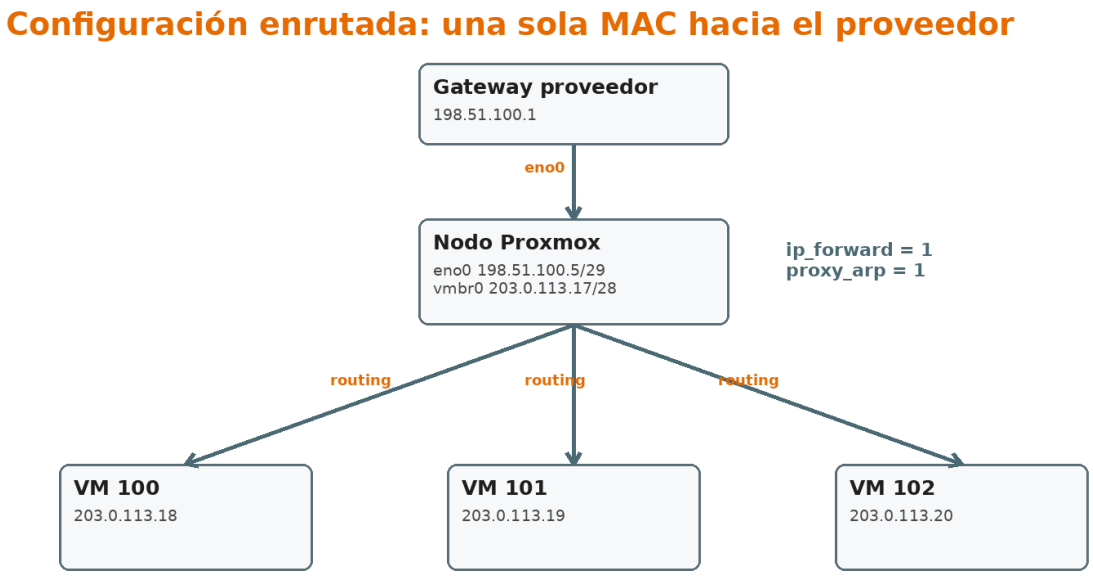
Las máquinas virtuales se comportan como si estuvieran conectadas directamente a la red física. La red ve cada máquina virtual con su propia dirección MAC, aunque exista un único cable de red conectando todas esas VM con la red.

3.4.5 Configuración enrutada

Muchos proveedores de hosting no admiten la configuración anterior. Por motivos de seguridad, desactivan la conectividad cuando detectan varias direcciones MAC en una misma interfaz.

CONSEJO: Algunos proveedores permiten registrar direcciones MAC adicionales desde su panel de gestión. Esto evita el problema, aunque puede resultar incómodo porque hay que registrar una MAC por cada VM.

El problema puede evitarse enrutando todo el tráfico a través de una única interfaz. De esta forma, todos los paquetes de red utilizan la misma dirección MAC.



Esquema traducido: ejemplo de red enrutada con un bloque público adicional para las VM.

Un escenario habitual consiste en disponer de una IP pública, por ejemplo 198.51.100.5, y un bloque IP adicional para las máquinas virtuales, por ejemplo 203.0.113.16/28. Para este caso se recomienda una configuración similar a la siguiente:

```
auto lo
iface lo inet loopback

auto eno0
iface eno0 inet static
        address 198.51.100.5/29
        gateway 198.51.100.1
        post-up echo 1 > /proc/sys/net/ipv4/ip_forward
        post-up echo 1 > /proc/sys/net/ipv4/conf/eno0/proxy_arp

auto vmbr0
iface vmbr0 inet static
        address 203.0.113.17/28
        bridge-ports none
        bridge-stp off
        bridge-fd 0
```

La dirección de vmbr0, 203.0.113.17, es la primera dirección de host utilizable del bloque 203.0.113.16/28. La dirección de red del bloque, 203.0.113.16, no puede asignarse a una interfaz.

3.4.6 Masquerading (NAT) con iptables

El masquerading permite que invitados que solo disponen de una dirección IP privada accedan a la red utilizando la dirección IP del host para el tráfico saliente. iptables reescribe cada paquete de salida para que parezca proceder del host y reescribe las respuestas de forma correspondiente para encaminarlas al remitente original.

```
auto lo
iface lo inet loopback

auto eno1
#real IP address
iface eno1 inet static
        address 198.51.100.5/24
        gateway 198.51.100.1

auto vmbr0
#private sub network
iface vmbr0 inet static
        address 10.10.10.1/24
        bridge-ports none
        bridge-stp off
        bridge-fd 0
        post-up echo 1 > /proc/sys/net/ipv4/ip_forward
        post-up iptables -t nat -A POSTROUTING -s '10.10.10.0/24' -o eno1 -j MASQUERADE
        post-down iptables -t nat -D POSTROUTING -s '10.10.10.0/24' -o eno1 -j MASQUERADE
```

NOTA: En algunas configuraciones con masquerading y firewall activado puede ser necesario utilizar zonas conntrack para las conexiones salientes. De lo contrario, el firewall podría bloquearlas porque preferirán el POSTROUTING del bridge de la VM y no la regla MASQUERADE.

Añadir las siguientes líneas a /etc/network/interfaces puede resolver el problema:

```
post-up iptables -t raw -I PREROUTING -i fwbr+ -j CT --zone 1
post-down iptables -t raw -D PREROUTING -i fwbr+ -j CT --zone 1
```

El manual original remite como referencias adicionales al flujo de paquetes de Netfilter, al parche de netdev-list que introdujo las zonas conntrack y a una explicación basada en TRACE sobre la tabla raw.

3.4.7 Linux Bond

Bonding, también denominado NIC teaming o Link Aggregation, es una técnica para agrupar varias NIC en un único dispositivo de red. Puede utilizarse para proporcionar tolerancia a fallos, aumentar el rendimiento o conseguir ambas cosas a la vez.

El hardware de alta velocidad, como Fibre Channel y la electrónica de conmutación asociada, puede resultar costoso. Mediante agregación de enlaces, dos NIC pueden presentarse como una interfaz lógica, dando como resultado el doble de velocidad. Es una función nativa del kernel Linux compatible con la mayoría de los switches. Si los nodos tienen varios puertos Ethernet, los cables pueden distribuirse entre distintos switches para reducir puntos únicos de fallo; en caso de problema de red, la conexión agregada puede conmutar de un cable a otro.

Los enlaces agregados pueden reducir los tiempos de migración en vivo y mejorar la velocidad de replicación de datos entre nodos del clúster Proxmox VE.

Existen siete modos de bonding:

- Round-robin (balance-rr): transmite paquetes secuencialmente desde la primera interfaz esclava disponible hasta la última. Proporciona balanceo de carga y tolerancia a fallos.
- Active-backup (active-backup): solo una NIC esclava está activa. Otra se activa únicamente si falla la actual. La MAC de la interfaz lógica agregada solo es visible externamente en una NIC para evitar distorsiones en el switch. Proporciona tolerancia a fallos.
- XOR (balance-xor): transmite en función de [(MAC origen XOR MAC destino) módulo número de esclavos]. Selecciona el mismo esclavo para cada MAC de destino. Proporciona balanceo de carga y tolerancia a fallos.
- Broadcast (broadcast): transmite los paquetes por todas las interfaces esclavas. Proporciona tolerancia a fallos.
- IEEE 802.3ad Dynamic link aggregation (802.3ad/LACP): crea grupos de agregación que comparten velocidad y configuración dúplex y utiliza todas las interfaces esclavas del grupo agregador activo según la especificación 802.3ad.
- Adaptive transmit load balancing (balance-tlb): modo del controlador bonding de Linux que no requiere soporte especial del switch. El tráfico saliente se distribuye según la carga actual, calculada en relación con la velocidad de cada interfaz. El tráfico entrante lo recibe una interfaz esclava designada; si falla, otra asume la MAC de la interfaz receptora fallida.
- Adaptive load balancing (balance-alb): incluye balance-tlb más balanceo de carga de recepción (rlb) para tráfico IPv4 y tampoco requiere soporte especial del switch. El balanceo de recepción se consigue mediante negociación ARP; el controlador modifica la dirección hardware de origen de las respuestas ARP para que distintos pares de red utilicen distintas MAC del bond.

Si el switch admite LACP (IEEE 802.3ad), se recomienda utilizar el modo de bonding correspondiente, 802.3ad. En caso contrario, generalmente debe utilizarse active-backup.

Para la red del clúster Corosync se recomienda configurar varias redes. Corosync no necesita un bond para obtener redundancia, ya que puede cambiar por sí mismo entre redes si una deja de ser utilizable. Algunos modos de bond pueden resultar problemáticos para Corosync; consulte el apartado «Corosync over Bonds».

La siguiente configuración de bond puede utilizarse como red de almacenamiento distribuido o compartido. Su ventaja es combinar más velocidad con tolerancia a fallos.

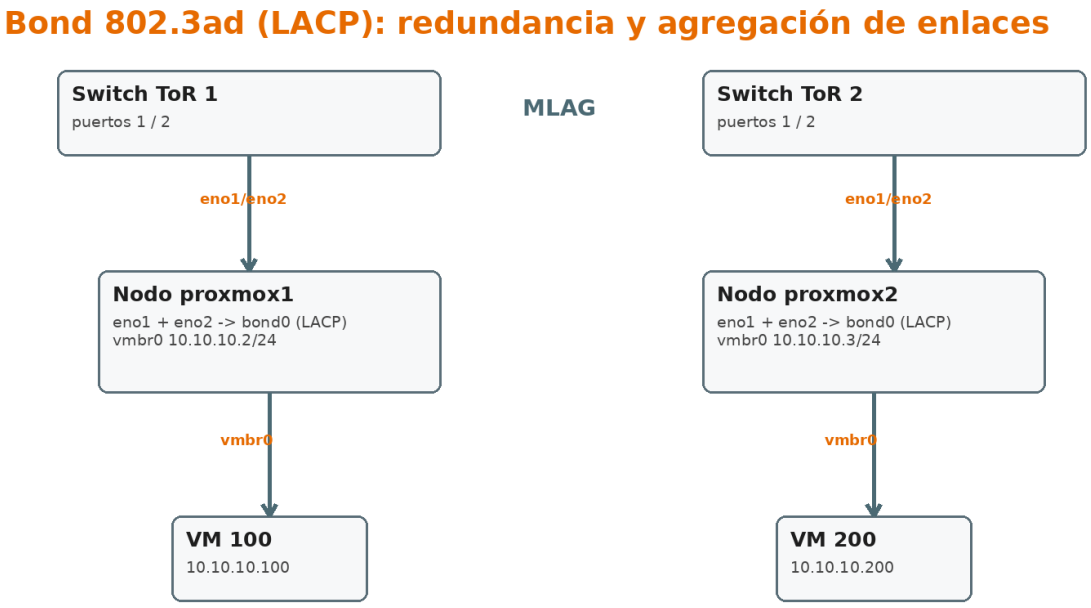
Ejemplo: bond con dirección IP fija

```
auto lo
iface lo inet loopback

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

auto bond0
iface bond0 inet static
        bond-slaves eno1 eno2
        address 192.168.1.2/24
        bond-miimon 100
        bond-mode 802.3ad
        bond-xmit-hash-policy layer2+3

auto vmbr0
iface vmbr0 inet static
        address 10.10.10.2/24
        gateway 10.10.10.1
        bridge-ports eno3
        bridge-stp off
        bridge-fd 0
```



Esquema traducido: bond 802.3ad/LACP con dos switches y dos nodos.

Otra posibilidad es utilizar el bond directamente como puerto del bridge. Puede emplearse para aportar tolerancia a fallos a la red de los invitados.

Ejemplo: utilizar un bond como puerto del bridge

```
auto lo
iface lo inet loopback

iface eno1 inet manual
iface eno2 inet manual

auto bond0
iface bond0 inet manual
        bond-slaves eno1 eno2
        bond-miimon 100
        bond-mode 802.3ad
        bond-xmit-hash-policy layer2+3

auto vmbr0
iface vmbr0 inet static
        address 10.10.10.2/24
        gateway 10.10.10.1
        bridge-ports bond0
        bridge-stp off
        bridge-fd 0
```

3.4.8 VLAN 802.1Q

Una LAN virtual (VLAN) es un dominio de broadcast particionado y aislado en la capa 2 de la red. De este modo pueden existir múltiples redes, hasta 4096, sobre una misma red física, independientes unas de otras. Cada VLAN se identifica mediante un número denominado habitualmente tag; los paquetes se etiquetan para indicar a qué red virtual pertenecen.

VLAN para redes de invitados

Proxmox VE admite esta configuración de forma nativa. Puede especificar la etiqueta VLAN al crear una VM. La etiqueta forma parte de la configuración de red del invitado. La capa de red admite distintos modos para implementar VLAN según la configuración del bridge:

- VLAN awareness en el bridge Linux: a cada tarjeta de red virtual del invitado se le asigna una etiqueta VLAN que el bridge Linux gestiona de forma transparente. También es posible el modo trunk, aunque en ese caso es necesario configurar las VLAN dentro del invitado.
- VLAN «tradicional» en el bridge Linux: a diferencia del método VLAN aware, no es transparente y crea un dispositivo VLAN y un bridge asociado por cada VLAN. Por ejemplo, crear un invitado en VLAN 5 generaría las interfaces eno1.5 y vmbr0v5, que permanecerían hasta el siguiente reinicio.
- VLAN en Open vSwitch: utiliza la funcionalidad VLAN de OVS.
- VLAN configurada en el invitado: las VLAN se asignan dentro del propio invitado. Toda la configuración se realiza en él y no puede controlarse desde el exterior. La ventaja es que pueden utilizarse varias VLAN en una sola NIC virtual.

VLAN en el host

Para permitir que el host se comunique con una red aislada pueden aplicarse etiquetas VLAN a cualquier dispositivo de red: NIC, bond o bridge. En general, la VLAN debería configurarse en la interfaz que tenga el menor número de capas de abstracción entre ella y la NIC física.

Por ejemplo, en una configuración predeterminada puede colocarse la dirección de gestión del host en una VLAN independiente.

Ejemplo: VLAN 5 para la IP de gestión con bridge Linux tradicional

```
auto lo
iface lo inet loopback

iface eno1 inet manual
iface eno1.5 inet manual

auto vmbr0v5
iface vmbr0v5 inet static
        address 10.10.10.2/24
        gateway 10.10.10.1
        bridge-ports eno1.5
        bridge-stp off
        bridge-fd 0

auto vmbr0
iface vmbr0 inet manual
        bridge-ports eno1
        bridge-stp off
        bridge-fd 0
```

Ejemplo: VLAN 5 para la IP de gestión con bridge Linux VLAN aware

```
auto lo
iface lo inet loopback

iface eno1 inet manual

auto vmbr0.5
iface vmbr0.5 inet static
        address 10.10.10.2/24
        gateway 10.10.10.1

auto vmbr0
iface vmbr0 inet manual
        bridge-ports eno1
        bridge-stp off
        bridge-fd 0
        bridge-vlan-aware yes
        bridge-vids 2-4094
```

El siguiente ejemplo utiliza la misma idea, pero añade un bond para que la red sea tolerante a fallos.

Ejemplo: VLAN 5 con bond0 para la IP de gestión y bridge Linux tradicional

```
auto lo
iface lo inet loopback

iface eno1 inet manual
iface eno2 inet manual

auto bond0
iface bond0 inet manual
        bond-slaves eno1 eno2
        bond-miimon 100
        bond-mode 802.3ad
        bond-xmit-hash-policy layer2+3

iface bond0.5 inet manual

auto vmbr0v5
iface vmbr0v5 inet static
        address 10.10.10.2/24
        gateway 10.10.10.1
        bridge-ports bond0.5
        bridge-stp off
        bridge-fd 0

auto vmbr0
iface vmbr0 inet manual
        bridge-ports bond0
        bridge-stp off
        bridge-fd 0
```

3.4.9 Desactivar IPv6 en el nodo

Proxmox VE funciona correctamente en cualquier entorno, con independencia de si IPv6 está desplegado o no. Se recomienda mantener los valores predeterminados.

Si aun así necesita desactivar IPv6 en un nodo, hágalo creando un fragmento adecuado de sysctl.conf(5), por ejemplo /etc/sysctl.d/disable-ipv6.conf, con el siguiente contenido:

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
```

Este método es preferible a desactivar la carga del módulo IPv6 desde la línea de comandos del kernel.

3.4.10 Desactivar el aprendizaje MAC en un bridge

De forma predeterminada, el aprendizaje de direcciones MAC está habilitado en un bridge para ofrecer un funcionamiento sencillo con los invitados virtuales y sus redes.

Sin embargo, en algunos entornos puede no ser deseable. Desde Proxmox VE 7.3 puede desactivarse el aprendizaje MAC estableciendo bridge-disable-mac-learning 1 en un bridge de /etc/network/interfaces. Ejemplo:

```
# ...
auto vmbr0
iface vmbr0 inet static
        address 10.10.10.2/24
        gateway 10.10.10.1
        bridge-ports ens18
        bridge-stp off
        bridge-fd 0
        bridge-disable-mac-learning 1
```

Una vez habilitada esta opción, Proxmox VE añadirá manualmente a la base de datos de reenvío de los bridges la dirección MAC configurada de las máquinas virtuales y los contenedores, de modo que los invitados puedan seguir utilizando la red, pero únicamente cuando usen su dirección MAC real.

3.5 Sincronización horaria

La propia pila de clúster de Proxmox VE depende en gran medida de que todos los nodos tengan la hora sincronizada con precisión. Otros componentes, como Ceph, tampoco funcionarán correctamente si la hora local de todos los nodos no está sincronizada.

La sincronización entre nodos puede conseguirse mediante Network Time Protocol (NTP). Desde Proxmox VE 7, chrony es el daemon NTP predeterminado, mientras que Proxmox VE 6 utiliza systemd-timesyncd. Ambos vienen preconfigurados para utilizar un conjunto de servidores públicos.

IMPORTANTE: Si actualiza un sistema a Proxmox VE 7, se recomienda instalar manualmente chrony, ntp u openntpd.

3.5.1 Uso de servidores NTP personalizados

En algunos casos puede ser conveniente utilizar servidores NTP distintos de los predeterminados. Por ejemplo, si los nodos Proxmox VE no tienen acceso a Internet debido a reglas de firewall restrictivas, será necesario disponer de servidores NTP locales e indicar al daemon NTP que los utilice.

Sistemas que utilizan chrony

Indique en /etc/chrony/chrony.conf los servidores que debe utilizar chrony:

```
server ntp1.example.com iburst
server ntp2.example.com iburst
server ntp3.example.com iburst
```

Reinicie chrony:

```
# systemctl restart chronyd
```

Compruebe el journal para confirmar que se están utilizando los servidores NTP recién configurados:

```
# journalctl --since -1h -u chrony
...
Aug 26 13:00:09 node1 systemd[1]: Started chrony, an NTP client/server.
Aug 26 13:00:15 node1 chronyd[4873]: Selected source 10.0.0.1 (ntp1.example.com)
Aug 26 13:00:15 node1 chronyd[4873]: System clock TAI offset set to 37 seconds
...
```

Sistemas que utilizan systemd-timesyncd

Indique en /etc/systemd/timesyncd.conf los servidores que debe utilizar systemd-timesyncd:

```
[Time]
NTP=ntp1.example.com ntp2.example.com ntp3.example.com ntp4.example.com
```

Después, reinicie el servicio de sincronización con systemctl restart systemd-timesyncd y compruebe en el journal que se están utilizando los nuevos servidores NTP con journalctl --since -1h -u systemd-timesyncd:

```
...
Oct 07 14:58:36 node1 systemd[1]: Stopping Network Time Synchronization...
Oct 07 14:58:36 node1 systemd[1]: Starting Network Time Synchronization...
Oct 07 14:58:36 node1 systemd[1]: Started Network Time Synchronization.
Oct 07 14:58:36 node1 systemd-timesyncd[13514]: Using NTP server 10.0.0.1:123 (ntp1.example.com).
Oct 07 14:58:36 node1 systemd-timesyncd[13514]: interval/delta/delay/jitter/drift 64s/-
0.002s/0.020s/0.000s/-31ppm
...
```

3.6 Servidor externo de métricas

En Proxmox VE pueden definirse servidores externos de métricas que recibirán periódicamente distintas estadísticas sobre los hosts, los invitados virtuales y los almacenamientos.

Actualmente se admiten:

- Graphite - https://graphiteapp.org
- InfluxDB - https://www.influxdata.com/time-series-platform/influxdb/
- OpenTelemetry - https://opentelemetry.io/

Las definiciones de los servidores externos de métricas se guardan en /etc/pve/status.cfg y pueden editarse desde la interfaz web.

3.6.1 Configuración de un servidor Graphite

El puerto predeterminado es 2003 y la ruta predeterminada de Graphite es proxmox.

De forma predeterminada, Proxmox VE envía los datos mediante UDP, por lo que el servidor Graphite debe configurarse para aceptarlos. También puede configurarse la unidad máxima de transmisión (MTU) en entornos que no utilicen la MTU estándar de 1500.

El complemento también puede configurarse para utilizar TCP. Para evitar bloquear el importante daemon de recopilación de estadísticas pvestatd, se requiere un timeout que permita manejar problemas de red.

3.6.2 Configuración del complemento InfluxDB

Proxmox VE envía los datos mediante UDP, por lo que el servidor InfluxDB debe configurarse para ello. También puede configurarse la MTU si es necesario.

Ejemplo de configuración de InfluxDB en el servidor InfluxDB:

```
[[udp]]
   enabled = true
   bind-address = "0.0.0.0:8089"
   database = "proxmox"
   batch-size = 1000
   batch-timeout = "1s"
```

Con esta configuración, el servidor escucha en todas las direcciones IP por el puerto 8089 y escribe los datos en la base de datos proxmox.

Como alternativa, el complemento puede configurarse para utilizar la API http(s) de InfluxDB 2.x. InfluxDB 1.8.x incluye un endpoint compatible hacia delante con esta API v2.

Para utilizarla, establezca influxdbproto en http o https, según su configuración. De forma predeterminada, Proxmox VE utiliza la organización proxmox y el bucket/base de datos proxmox; pueden cambiarse mediante organization y bucket, respectivamente.

Como la API v2 de InfluxDB solo está disponible con autenticación, debe generar un token con permisos de escritura sobre el bucket correcto y configurarlo.

En la API compatible con v2 de InfluxDB 1.8.x puede utilizar user:password como token, si es necesario, y omitir organization, ya que no tiene significado en InfluxDB 1.x.

También puede configurar el timeout HTTP, cuyo valor predeterminado es 1 s, mediante timeout, así como el tamaño máximo de lote, cuyo valor predeterminado es 25000000 bytes, mediante max-body-size. Este último corresponde al ajuste del mismo nombre de InfluxDB.

3.7 Supervisión del estado de los discos

Aunque se recomienda utilizar almacenamiento robusto y redundante, puede resultar muy útil supervisar el estado de los discos locales.

Desde Proxmox VE 4.3, el paquete smartmontools está instalado y es obligatorio. Se trata de un conjunto de herramientas para supervisar y controlar el sistema S.M.A.R.T. de los discos duros locales.

Puede obtener el estado de un disco con:

```
# smartctl -a /dev/sdX
```

donde /dev/sdX es la ruta de uno de los discos locales.

Si la salida indica:

```
SMART support is: Disabled
```

puede habilitar S.M.A.R.T. con:

```
# smartctl -s on /dev/sdX
```

Para más información sobre smartctl, consulte man smartctl.

De forma predeterminada, el daemon smartd de smartmontools está activo y habilitado. Examina cada 30 minutos los dispositivos que coincidan con los siguientes patrones para detectar errores y advertencias, y envía un correo electrónico a root si encuentra un problema:

- /dev/sd[a-z]
- /dev/sd[a-z][a-z]
- /dev/hd[a-t]
- /dev/nvme[0-99]

Para más información sobre la configuración de smartd, consulte man smartd y man smartd.conf.

Si utiliza los discos duros a través de un controlador RAID por hardware, lo más probable es que existan herramientas específicas para supervisar los discos del array y el propio array. Consulte la documentación del fabricante del controlador RAID.

Referencia del manual original: smartmontools - https://www.smartmontools.org

FIN DE LA PARTE 3A: La siguiente entrega continúa el Capítulo 3 desde la sección 3.8 (Logical Volume Manager - LVM), seguida de ZFS, BTRFS, gestión del nodo, certificados, bootloader, Secure Boot y KSM.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 - Traducción al castellano

PARTE 3B · Capítulo 3: Administración del sistema host

Secciones 3.8 a 3.14 · LVM, ZFS, BTRFS, nodos, certificados, bootloader, Secure Boot y KSM

Basado en “Proxmox VE Administration Guide”, Release 9.2.4
4 de agosto de 2026 · Proxmox Server Solutions GmbH

Aviso sobre esta parte

Esta Parte 3B continúa la traducción progresiva del manual oficial y completa el Capítulo 3, desde la sección 3.8 hasta la 3.14. Se corresponde con las páginas 57 a 97 de la numeración interna del manual original.

La traducción mantiene la organización, advertencias, comandos, rutas, nombres de paquetes, parámetros, ejemplos y nivel de detalle del documento fuente. Los bloques de comandos se reproducen sin traducir; únicamente se han normalizado algunos saltos de línea y elementos gráficos para mejorar la legibilidad en castellano.

La edición completa se publicará como versión modificada bajo la GNU Free Documentation License (GFDL), conservando los avisos de copyright y la licencia original en inglés conforme a sus condiciones.

Contenido de la Parte 3B

- 3.8 Logical Volume Manager (LVM)
- 3.9 ZFS en Linux
- 3.10 BTRFS
- 3.11 Gestión de nodos Proxmox
- 3.12 Gestión de certificados
- 3.13 Bootloader del host
- 3.14 Kernel Samepage Merging (KSM)

Capítulo 3 · Administración del sistema host

3.8 Logical Volume Manager (LVM)

La mayoría de las personas instala Proxmox VE directamente sobre un disco local. El medio de instalación de Proxmox VE ofrece varias opciones para gestionar los discos locales y la configuración predeterminada actual utiliza LVM. El instalador permite seleccionar un único disco para esta configuración y utiliza ese disco como volumen físico del Volume Group (VG) pve. La siguiente salida procede de una instalación de prueba con un disco pequeño de 8 GB:

```
# pvs
  PV             VG    Fmt Attr PSize PFree
  /dev/sda3      pve   lvm2 a-- 7.87g 876.00m

# vgs
  VG       #PV #LV #SN Attr   VSize VFree
  pve        1   3   0 wz--n- 7.87g 876.00m
```

El instalador asigna tres Logical Volumes (LV) dentro de este VG:

```
# lvs
  LV   VG        Attr       LSize   Pool Origin Data%  Meta%
  data pve       twi-a-tz--  4.38g             0.00   0.63
  root pve       -wi-ao----  1.75g
  swap pve       -wi-ao---- 896.00m
```

Volumen	Uso
root	Formateado como ext4; contiene el sistema operativo.
swap	Partición de intercambio (swap).
data	Utiliza LVM-thin y se emplea para almacenar imágenes de máquinas virtuales. LVM-thin es preferible para esta tarea porque ofrece un soporte eficiente para snapshots y clones.

Hasta Proxmox VE 4.1 inclusive, el instalador creaba un volumen lógico estándar denominado “data”, montado en /var/lib/vz. Desde la versión 4.2, el volumen lógico “data” es un thin pool de LVM utilizado para almacenar imágenes de invitados basadas en bloques, y /var/lib/vz es simplemente un directorio del sistema de archivos raíz.

3.8.1 Hardware

Para este tipo de configuraciones se recomienda encarecidamente utilizar un controlador RAID por hardware con BBU. Esto aumenta el rendimiento, proporciona redundancia y facilita la sustitución de discos, incluso en caliente.

LVM por sí mismo no necesita ningún hardware especial y sus requisitos de memoria son muy bajos.

3.8.2 Bootloader

De forma predeterminada se instalan dos mecanismos de arranque. La primera partición contiene el bootloader GRUB estándar. La segunda es una EFI System Partition (ESP), que permite arrancar en sistemas EFI y aplicar actualizaciones persistentes de firmware desde el espacio de usuario.

3.8.3 Crear un Volume Group

Supongamos que disponemos de un disco vacío /dev/sdb en el que queremos crear un grupo de volúmenes denominado “vmdata”.

PRECAUCIÓN: Los comandos siguientes destruirán todos los datos existentes en /dev/sdb.

Primero cree una partición:

```
# sgdisk -N 1 /dev/sdb
```

Cree un Physical Volume (PV) sin confirmación y con 250K de tamaño de metadatos:

```
# pvcreate --metadatasize 250k -y -ff /dev/sdb1
```

Cree un grupo de volúmenes llamado “vmdata” sobre /dev/sdb1:

```
# vgcreate vmdata /dev/sdb1
```

3.8.4 Crear un LV adicional para /var/lib/vz

Puede hacerse fácilmente creando un nuevo thin LV:

```
# lvcreate -n <Name> -V <Size[M,G,T]> <VG>/<LVThin_pool>
```

Ejemplo real:

```
# lvcreate -n vz -V 10G pve/data
```

A continuación debe crearse un sistema de archivos sobre el LV:

```
# mkfs.ext4 /dev/pve/vz
```

Por último, debe montarse.

ADVERTENCIA: Asegúrese de que /var/lib/vz está vacío. En una instalación predeterminada no lo está.

Para que siempre esté accesible, añada la siguiente línea a /etc/fstab:

```
# echo '/dev/pve/vz /var/lib/vz ext4 defaults 0 2' >> /etc/fstab
```

3.8.5 Redimensionar el thin pool

Redimensione el LV y el pool de metadatos con el siguiente comando:

```
# lvresize --size +<size[M,G,T]> --poolmetadatasize +<size[M,G]> <VG>/<LVThin_pool>
```

NOTA: Al ampliar el data pool, también debe ampliarse el pool de metadatos.

3.8.6 Crear un pool LVM-thin

Un thin pool debe crearse sobre un grupo de volúmenes. Para saber cómo crear un Volume Group, consulte la sección LVM.

```
# lvcreate -L 80G -T -n vmstore vmdata
```

3.9 ZFS en Linux

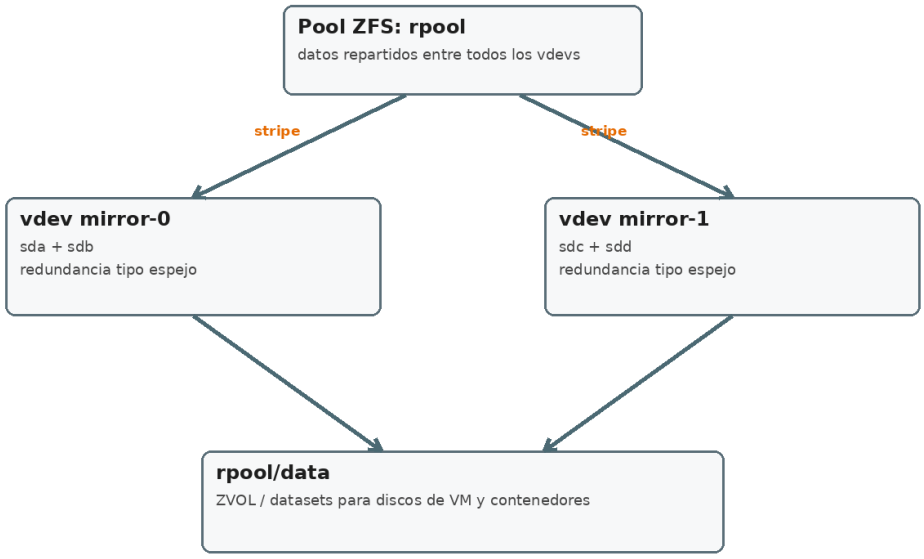
ZFS es un sistema de archivos y gestor de volúmenes lógicos combinado, diseñado por Sun Microsystems. Desde Proxmox VE 3.4, el port nativo para el kernel Linux del sistema de archivos ZFS se ofrece como sistema de archivos opcional y también como opción adicional para el sistema de archivos raíz. No es necesario compilar manualmente módulos ZFS: todos los paquetes están incluidos.

Con ZFS es posible obtener funciones de nivel empresarial utilizando hardware de coste moderado, y también construir sistemas de alto rendimiento mediante caché SSD o configuraciones formadas únicamente por SSD. ZFS puede sustituir costosas controladoras RAID por hardware a cambio de una carga moderada de CPU y memoria, junto con una administración sencilla.

Ventajas generales de ZFS

- Configuración y administración sencillas mediante la GUI y la CLI de Proxmox VE.
- Fiabilidad.
- Protección frente a corrupción de datos.
- Compresión de datos a nivel de sistema de archivos.
- Snapshots.
- Clones copy-on-write.
- Diversos niveles RAID: RAID0, RAID1, RAID10, RAIDZ-1, RAIDZ-2, RAIDZ-3, dRAID, dRAID2 y dRAID3.
- Posibilidad de utilizar SSD como caché.
- Autorrecuperación.
- Comprobación continua de integridad.
- Diseñado para grandes capacidades de almacenamiento.
- Replicación asíncrona a través de la red.
- Código abierto.
- Cifrado.
- ...

ZFS: pool, vdevs y almacenamiento de invitados



Esquema didáctico: un pool ZFS distribuye los datos entre sus vdevs; los datasets/ZVOL pueden utilizarse para los invitados.

3.9.1 Hardware

ZFS depende en gran medida de la memoria, por lo que se necesitan al menos 8 GB para empezar. En la práctica, utilice tanta memoria como permitan el hardware y el presupuesto. Para prevenir la corrupción de datos, se recomienda utilizar memoria ECC de alta calidad.

Si utiliza un disco dedicado para caché y/o log, debería emplear un SSD de clase empresarial. Esto puede aumentar significativamente el rendimiento global.

IMPORTANTE: No utilice ZFS sobre una controladora RAID por hardware que gestione su propia caché. ZFS necesita comunicarse directamente con los discos. Es más apropiado utilizar un adaptador HBA o una controladora, por ejemplo LSI, configurada/flasheada en modo "IT".

Si está experimentando con una instalación de Proxmox VE dentro de una máquina virtual (virtualización anidada), no utilice virtio para los discos de esa VM, ya que no están soportados por ZFS. Utilice IDE o SCSI en su lugar; también funciona con el tipo de controladora virtio SCSI.

3.9.2 Instalación como sistema de archivos raíz

Cuando se instala Proxmox VE mediante su instalador, puede elegirse ZFS como sistema de archivos raíz. El tipo de RAID debe seleccionarse durante la instalación:

Nivel	Descripción
RAID0	También denominado striping. La capacidad es la suma de las capacidades de todos los discos. No aporta redundancia; el fallo de una sola unidad inutiliza el volumen.
RAID1	También denominado mirroring. Los datos se escriben de forma idéntica en todos los discos. Requiere al menos 2 discos del mismo tamaño. La capacidad resultante es la de un único disco.
RAID10	Combinación de RAID0 y RAID1. Requiere al menos 4 discos.
RAIDZ-1	Variación de RAID-5 con paridad simple. Requiere al menos 3 discos.
RAIDZ-2	Variación de RAID-5 con doble paridad. Requiere al menos 4 discos.
RAIDZ-3	Variación de RAID-5 con triple paridad. Requiere al menos 5 discos.

El instalador particiona automáticamente los discos, crea un pool ZFS denominado rpool e instala el sistema de archivos raíz en el subvolumen ZFS rpool/ROOT/pve-1.

También se crea el subvolumen rpool/data para almacenar imágenes de máquinas virtuales. Para utilizarlo con las herramientas de Proxmox VE, el instalador crea la siguiente entrada en /etc/pve/storage.cfg:

```
zfspool: local-zfs
        pool rpool/data
        sparse
        content images,rootdir
```

Después de la instalación, puede consultar el estado del pool ZFS mediante zpool:

```
# zpool status
  pool: rpool
 state: ONLINE
  scan: none requested
config:

        NAME             STATE   READ WRITE CKSUM
        rpool            ONLINE     0     0     0
          mirror-0       ONLINE     0     0     0
            sda2         ONLINE     0     0     0
            sdb2         ONLINE     0     0     0
          mirror-1       ONLINE     0     0     0
            sdc          ONLINE     0     0     0
            sdd          ONLINE     0     0     0

errors: No known data errors
```

El comando zfs se utiliza para configurar y administrar los sistemas de archivos ZFS. El siguiente comando enumera todos los sistemas de archivos después de la instalación:

```
# zfs list
NAME                     USED   AVAIL  REFER  MOUNTPOINT
rpool                   4.94G   7.68T    96K  /rpool
rpool/ROOT               702M   7.68T    96K  /rpool/ROOT
rpool/ROOT/pve-1         702M   7.68T   702M  /
rpool/data                96K   7.68T    96K  /rpool/data
rpool/swap              4.25G   7.69T    64K  -
```

3.9.3 Consideraciones sobre niveles RAID de ZFS

Al elegir la disposición de un pool ZFS deben considerarse varios factores. El bloque básico de construcción de un pool ZFS es el dispositivo virtual, o vdev. Todos los vdevs de un pool se utilizan por igual y los datos se distribuyen entre ellos mediante striping (RAID0). Consulte la página de manual zpoolconcepts(7) para obtener más detalles sobre los vdevs.

Rendimiento

Cada tipo de vdev presenta comportamientos de rendimiento distintos. Los dos parámetros de interés son los IOPS (operaciones de entrada/salida por segundo) y el ancho de banda con el que pueden escribirse o leerse los datos.

Un vdev en espejo (RAID1) se comportará aproximadamente como un solo disco, tanto en IOPS como en ancho de banda, al escribir. Al leer, el rendimiento escalará linealmente con el número de discos del espejo.

Una situación habitual es disponer de 4 discos. Si se configuran como 2 vdevs en espejo (RAID10), el pool tendrá, para escritura, características equivalentes a dos discos individuales en IOPS y ancho de banda. Para lectura se parecerá a 4 discos individuales.

Un RAIDZ, con cualquier nivel de redundancia, se comportará aproximadamente como un único disco en IOPS, pero con un ancho de banda elevado. El ancho de banda concreto depende del tamaño del vdev RAIDZ y de su nivel de redundancia.

Un pool dRAID debería ofrecer un rendimiento equivalente al de un pool RAIDZ comparable.

Para ejecutar máquinas virtuales, los IOPS suelen ser la métrica más importante.

Tamaño, uso del espacio y redundancia

Aunque un pool formado por vdevs en espejo ofrece las mejores características de rendimiento, el espacio utilizable será el 50 % del total disponible. Será inferior si un vdev en espejo contiene más de 2 discos, por ejemplo en un espejo de tres vías . Para que el pool continúe funcionando se necesita al menos un disco sano en cada espejo.

El espacio utilizable de un vdev RAIDZ con N discos es aproximadamente N -P, donde P es el nivel RAIDZ. El nivel RAIDZ indica cuántos discos arbitrarios pueden fallar sin perder datos. Un caso especial es un pool de 4 discos con RAIDZ2: en esta situación suele ser mejor utilizar 2 vdevs en espejo por su mayor rendimiento, ya que el espacio utilizable será el mismo.

Otro factor importante al utilizar cualquier nivel RAIDZ es el comportamiento de los datasets ZVOL, que se utilizan para los discos de las máquinas virtuales. Por cada bloque de datos, el pool necesita datos de paridad cuyo tamaño es, como mínimo, el tamaño de bloque mínimo definido por el valor ashift del pool. Con ashift=12, el tamaño de bloque del pool es 4K. El tamaño de bloque predeterminado de un ZVOL es 8K. Por tanto, en RAIDZ2 cada bloque de 8K escrito provocará la escritura de dos bloques de paridad adicionales de 4K: 8K + 4K + 4K = 16K. Se trata de una simplificación; en una situación real también intervienen metadatos, compresión y otros factores.

Este comportamiento puede observarse consultando las propiedades volsize, refreservation -si el pool no utiliza thin provisioning- y used -si el pool utiliza thin provisioning y no existen snapshots-:

```
# zfs get volsize,refreservation,used <pool>/vm-<vmid>-disk-X
```

volsize es el tamaño del disco tal como se presenta a la VM, mientras que refreservation muestra el espacio reservado en el pool, incluido el espacio esperado para los datos de paridad. Si el pool utiliza thin provisioning, refreservation será 0. Otra forma de observarlo es comparar el espacio utilizado dentro de la VM con la propiedad used. Tenga en cuenta que los snapshots alterarán este valor.

Existen varias opciones para contrarrestar el aumento del uso de espacio:

- Aumentar volblocksize para mejorar la relación entre datos y paridad.
- Utilizar vdevs en espejo en lugar de RAIDZ.
- Utilizar ashift=9, correspondiente a bloques de 512 bytes.

La propiedad volblocksize solo puede establecerse al crear un ZVOL. Su valor predeterminado puede modificarse en la configuración de almacenamiento. En tal caso, el invitado debe ajustarse en consecuencia y, dependiendo del uso, el problema de amplificación de escritura puede trasladarse simplemente de la capa ZFS al invitado.

Utilizar ashift=9 al crear el pool puede producir un rendimiento deficiente dependiendo de los discos subyacentes y no puede modificarse posteriormente.

Los vdevs en espejo (RAID1, RAID10) presentan un comportamiento favorable para cargas de trabajo de máquinas virtuales. Utilícelos salvo que su entorno tenga necesidades y características concretas para las que sean aceptables las características de rendimiento de RAIDZ.

3.9.4 ZFS dRAID

En un ZFS dRAID (declustered RAID), las unidades hot spare participan en el RAID. Su capacidad de reserva queda apartada y se utiliza para la reconstrucción cuando falla una unidad. Dependiendo de la configuración, esto proporciona una reconstrucción más rápida que RAIDZ en caso de fallo de disco. Puede encontrarse más información en la documentación oficial de OpenZFS.

NOTA: dRAID está pensado para configuraciones con más de 10-15 discos. Para un número menor de discos, una configuración RAIDZ suele ser mejor en la mayoría de los casos.

NOTA: La GUI requiere un disco más que el mínimo: por ejemplo, dRAID1 necesita 3. La interfaz espera que también se añada un disco de reserva.

- dRAID1 o dRAID: requiere al menos 2 discos; puede fallar uno antes de perder datos.
- dRAID2: requiere al menos 3 discos; pueden fallar dos antes de perder datos.
- dRAID3: requiere al menos 4 discos; pueden fallar tres antes de perder datos.

Información adicional:

```
# man zpoolconcepts
```

Spares y Data

El número de spares indica al sistema cuántos discos debe mantener preparados para un fallo. El valor predeterminado es 0. Sin spares, la reconstrucción no obtiene ventajas de velocidad.

data define el número de dispositivos de un grupo de redundancia. El valor predeterminado es 8. La excepción se produce cuando discos - paridad - spares da como resultado un valor inferior a 8; entonces se utiliza ese valor menor. En general, un número menor de dispositivos de datos proporciona más IOPS, mejores ratios de compresión y resilvering más rápido, pero reduce la capacidad de almacenamiento disponible del pool.

Referencia del manual original: OpenZFS dRAID - https://openzfs.github.io/openzfs-docs/Basic%20Concepts/dRAID%20Howto.html

3.9.5 Bootloader

Proxmox VE utiliza proxmox-boot-tool para gestionar la configuración del bootloader. Consulte el apartado dedicado a los bootloaders del host Proxmox VE para obtener más detalles.

3.9.6 Administración de ZFS

Este apartado proporciona ejemplos de uso para tareas habituales. ZFS es muy potente y ofrece numerosas opciones. Los principales comandos para administrarlo son zfs y zpool. Ambos disponen de páginas de manual detalladas:

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

Crear un zpool nuevo

Para crear un pool nuevo se necesita al menos un disco. El valor ashift debe corresponder al mismo tamaño de sector, o a uno mayor, que el disco subyacente (2 elevado a ashift).

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

CONSEJO: Los nombres de pool deben empezar por una letra; contener únicamente caracteres alfanuméricos, -, _, ., : o espacio; no empezar por mirror, raidz, draid o spare; y no ser log.

Para activar la compresión, consulte también la sección “Compresión en ZFS”:

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

Crear un pool RAID-0

Mínimo 1 disco:

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

Crear un pool RAID-1

Mínimo 2 discos:

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

Crear un pool RAID-10

Mínimo 4 discos:

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

Crear un pool RAIDZ-1

Mínimo 3 discos:

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

Crear un pool RAIDZ-2

Mínimo 4 discos:

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

Consulte la sección sobre consideraciones de niveles RAID de ZFS para obtener una estimación aproximada de los IOPS y del ancho de banda antes de configurar un pool, especialmente si desea utilizar un modo RAID-Z.

Ampliar RAIDZ-N

NOTA: Esta función solo está disponible a partir de Proxmox VE 9 (ZFS 2.3.3).

Si dispone de un <pool> existente con un vdev RAIDZ-N <raidzN-M>, puede añadir un nuevo disco físico <device> con la siguiente sintaxis:

```
zpool attach <pool> <raidzN-M> <device>
```

Puede verificar el éxito general ejecutando zpool status <pool>, obtener salida detallada para todos los discos conectados con zpool list <pool> -v y comprobar la nueva capacidad del pool con zfs list <pool>.

Crear un pool con caché (L2ARC)

Puede utilizarse un dispositivo o una partición dedicados como caché de segundo nivel para aumentar el rendimiento. Este dispositivo de caché ayuda especialmente en cargas de lectura aleatoria de datos mayoritariamente estáticos. Al actuar como capa adicional entre el almacenamiento real y la ARC en memoria, también puede ayudar cuando la ARC debe reducirse por restricciones de memoria.

Crear un pool ZFS con caché en disco:

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

El ejemplo utiliza un único <device> y un único <cache-device>, pero pueden utilizarse más dispositivos, igual que en los ejemplos de pools RAID. Para los dispositivos de caché no existen modos mirror o RAID: simplemente se acumulan. Si un dispositivo de caché produce errores de lectura, ZFS desviará transparentemente la petición hacia la capa de almacenamiento subyacente.

Crear un pool con log (ZIL)

Puede utilizarse una unidad o partición dedicada para el ZFS Intent Log (ZIL). Se utiliza principalmente para proporcionar transacciones síncronas seguras, por lo que suele estar presente en rutas críticas de rendimiento, como bases de datos u otros programas que ejecutan operaciones fsync con frecuencia.

El pool se utiliza como ubicación predeterminada del ZIL. Desviar la carga de E/S del ZIL a un dispositivo separado puede ayudar a reducir las latencias de transacción y descargar el pool principal, aumentando al mismo tiempo el rendimiento global.

Para los discos utilizados como log, directamente o mediante una partición, se recomienda:

- Utilizar SSD rápidos con protección frente a pérdida de alimentación, ya que presentan latencias de commit mucho menores.
- Utilizar al menos unos pocos GB para la partición o el dispositivo completo; utilizar más de la mitad de la memoria instalada no aporta una ventaja real.

Crear un pool ZFS con un dispositivo de log independiente:

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

En el ejemplo se utiliza un único <device> y un único <log-device>, pero también puede combinarse con otras variantes RAID. El dispositivo de log puede duplicarse en varios dispositivos, principalmente para evitar que el rendimiento se degrade inmediatamente si falla uno de ellos. Si fallan todos los dispositivos de log, volverá a utilizarse el propio pool principal de ZFS hasta que sean sustituidos.

Añadir caché y log a un pool existente

Si dispone de un pool sin caché ni log, puede añadir ambos, o solo uno, en cualquier momento. Por ejemplo, si dispone de un SSD empresarial con protección frente a pérdida de alimentación, puede utilizar una parte para el ZIL y el resto como caché. Como el tamaño máximo de un dispositivo de log debería ser aproximadamente la mitad de la memoria física instalada, el ZIL normalmente ocupará una parte relativamente pequeña del SSD.

Primero cree dos particiones GPT en el SSD con parted o gdisk. Después añádalas al pool:

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

Sustituya <pool>, <device-part1> y <device-part2> por el nombre del pool y las rutas /dev/disk/by-id/ de las dos particiones. También pueden añadirse ZIL y caché por separado.

Añadir un dispositivo de log a un pool ZFS existente:

```
# zpool add <pool> log <log-device>
```

Sustituir un dispositivo averiado

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

Sustituir un dispositivo de arranque averiado

Dependiendo de cómo se instaló Proxmox VE, puede estar utilizando systemd-boot o GRUB mediante proxmox-boot-tool, o GRUB sin esa herramienta. Puede comprobarlo con:

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

Los primeros pasos -copiar la tabla de particiones, generar nuevos GUID y sustituir la partición ZFS- son los mismos. Para que el sistema pueda arrancar desde el disco nuevo, los pasos posteriores dependen del bootloader utilizado:

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

NOTA: Utilice zpool status -v para supervisar el progreso del resilvering del nuevo disco.

Con proxmox-boot-tool:

```
# proxmox-boot-tool format <new disk's ESP>
# proxmox-boot-tool init <new disk's ESP> [grub]
```

NOTA: ESP significa EFI System Partition. Desde Proxmox VE 5.4, el instalador la crea como partición número 2 en los discos de arranque. Consulte el apartado sobre cómo preparar una partición nueva como ESP sincronizada.

NOTA: Pase grub como modo a proxmox-boot-tool init si proxmox-boot-tool status indica que los discos actuales utilizan GRUB, especialmente si Secure Boot está habilitado.

Con GRUB sin proxmox-boot-tool:

```
# grub-install <new disk>
```

NOTA: GRUB sin proxmox-boot-tool solo se utiliza en sistemas instalados con Proxmox VE 6.3 o anterior que no hayan sido migrados manualmente a proxmox-boot-tool.

3.9.7 Configurar notificaciones por correo electrónico

ZFS incluye el daemon de eventos ZED, que supervisa los eventos generados por el módulo ZFS del kernel. El daemon también puede enviar correos electrónicos ante eventos de ZFS, como errores en un pool. Las versiones recientes de los paquetes ZFS distribuyen el daemon en el paquete independiente zfs-zed, que debería estar instalado de forma predeterminada en Proxmox VE.

Puede configurarlo mediante /etc/zfs/zed.d/zed.rc. El ajuste necesario para las notificaciones por correo es ZED_EMAIL_ADDR, cuyo valor predeterminado es root:

```
ZED_EMAIL_ADDR="root"
```

Proxmox VE reenvía los mensajes destinados a root a la dirección de correo configurada para el usuario root.

3.9.8 Limitar el uso de memoria de ZFS

De forma predeterminada, ZFS utiliza para Adaptive Replacement Cache (ARC) el 10 % de la memoria del host, limitado a un máximo de 16 GiB. Este valor se escribe en /etc/modprobe.d/zfs.conf durante la instalación. Consulte man 4 zfs para obtener más información sobre los parámetros del módulo ZFS del kernel.

Antes de Proxmox VE 8.1, el límite de uso de ARC no se establecía durante la instalación y coincidía con el valor predeterminado de ZFS: 62,5 % a partir de ZFS 2.3.0 o 50 % en versiones anteriores. En instalaciones existentes anteriores a Proxmox VE 8.1, deben realizarse pasos manuales para reducir el límite como se describe a continuación.

Asignar memoria suficiente a ARC es crucial para el rendimiento de E/S, por lo que debe reducirse con precaución. Como regla general, asigne al menos 2 GiB de base + 1 GiB por TiB de almacenamiento. Por ejemplo, para un pool con 8 TiB de espacio disponible debería utilizar 10 GiB de memoria para ARC. ZFS también impone un valor mínimo de 64 MiB.

Puede cambiar el límite de ARC para el arranque actual -se restablecerá tras reiniciar- escribiendo directamente en el parámetro zfs_arc_max:

```
echo "$[10 * 1024*1024*1024]" >/sys/module/zfs/parameters/zfs_arc_max
```

Para cambiar de forma permanente los límites de ARC, añada o modifique la siguiente línea en /etc/modprobe.d/zfs.conf:

```
options zfs zfs_arc_max=8589934592
```

Este ejemplo limita el uso a 8 GiB.

IMPORTANTE: Si el valor deseado de zfs_arc_max es menor o igual que zfs_arc_min -cuyo valor predeterminado es 1/32 de la memoria del sistema-, zfs_arc_max se ignorará a menos que también establezca zfs_arc_min como máximo en zfs_arc_max - 1.

```
echo "$[8 * 1024*1024*1024 - 1]" >/sys/module/zfs/parameters/zfs_arc_min
echo "$[8 * 1024*1024*1024]" >/sys/module/zfs/parameters/zfs_arc_max
```

Este ejemplo limita temporalmente el uso a 8 GiB en sistemas con más de 256 GiB de memoria total, donde establecer únicamente zfs_arc_max no funcionaría.

IMPORTANTE: Si el sistema de archivos raíz utiliza ZFS, debe actualizar initramfs cada vez que cambie este valor: # update-initramfs -u -k all. Debe reiniciar para activar los cambios.

3.9.9 SWAP sobre ZFS

El espacio swap creado sobre un zvol puede generar problemas, como bloquear el servidor o producir una carga de E/S elevada; esto suele observarse al iniciar un backup hacia un almacenamiento externo.

Se recomienda encarecidamente disponer de memoria suficiente para no llegar habitualmente a situaciones de memoria escasa. Si necesita o desea añadir swap, es preferible crear una partición en un disco físico y utilizarla como dispositivo swap. En las opciones avanzadas del instalador puede dejar espacio libre para este propósito.

También puede reducirse el valor de swappiness. Un valor adecuado para servidores es 10:

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

Para hacerlo persistente, cree /etc/sysctl.d/99-swappiness.conf y añada:

```
vm.swappiness = 10
```

Valor	Estrategia
vm.swappiness = 0	El kernel solo utilizará swap para evitar una condición de falta de memoria.
vm.swappiness = 1	Cantidad mínima de swapping sin desactivarlo por completo.
vm.swappiness = 10	Valor que en ocasiones se recomienda para mejorar el rendimiento cuando el sistema dispone de memoria suficiente.
vm.swappiness = 60	Valor predeterminado.
vm.swappiness = 100	El kernel utilizará swap de forma agresiva.

3.9.10 Datasets ZFS cifrados

ADVERTENCIA: El cifrado nativo de ZFS en Proxmox VE es experimental. Entre las limitaciones y problemas conocidos se incluyen la replicación con datasets cifrados y errores de checksum al utilizar snapshots o ZVOL.

Referencias del manual original: https://bugzilla.proxmox.com/show_bug.cgi?id=2350 · https://github.com/openzfs/zfs/issues/11688

ZFS on Linux 0.8.0 introdujo soporte para cifrado nativo de datasets. Tras actualizar desde versiones anteriores, la función de cifrado puede habilitarse por pool:

```
# zpool get feature@encryption tank
NAME PROPERTY             VALUE      SOURCE
tank feature@encryption disabled     local

# zpool set feature@encryption=enabled

# zpool get feature@encryption tank
NAME PROPERTY             VALUE      SOURCE
tank feature@encryption enabled      local
```

ADVERTENCIA: Actualmente no existe soporte para arrancar desde pools con datasets cifrados utilizando GRUB y solo hay soporte limitado para desbloquear automáticamente datasets cifrados durante el arranque. Versiones antiguas de ZFS sin soporte de cifrado no podrán descifrar los datos almacenados.

NOTA: Se recomienda desbloquear manualmente los datasets de almacenamiento después del arranque o escribir una unidad personalizada que proporcione a zfs load-key el material de clave necesario para desbloquearlos.

ADVERTENCIA: Establezca y pruebe un procedimiento de backup antes de habilitar el cifrado de datos de producción. Si se pierde el material de clave, passphrase o keyfile asociado, ya no será posible acceder a los datos cifrados.

El cifrado debe configurarse al crear los datasets o zvols y, de forma predeterminada, lo heredan los datasets hijos. Por ejemplo, para crear un dataset cifrado tank/encrypted_data y configurarlo como almacenamiento en Proxmox VE:

```
# zfs create -o encryption=on -o keyformat=passphrase tank/encrypted_data
Enter passphrase:
Re-enter passphrase:

# pvesm add zfspool encrypted_zfs -pool tank/encrypted_data
```

Todos los volúmenes/discos de invitados creados en este almacenamiento quedarán cifrados con el material de clave compartido del dataset padre.

Para utilizar el almacenamiento, debe cargarse el material de clave asociado y montarse el dataset. Puede hacerse en un solo paso:

```
# zfs mount -l tank/encrypted_data
Enter passphrase for 'tank/encrypted_data':
```

También puede utilizarse un keyfile aleatorio en lugar de solicitar una passphrase, configurando las propiedades keylocation y keyformat al crear el dataset o mediante zfs change-key en datasets existentes:

```
# dd if=/dev/urandom of=/path/to/keyfile bs=32 count=1

# zfs change-key -o keyformat=raw -o keylocation=file:///path/to/keyfile tank/encrypted_data
```

ADVERTENCIA: Cuando se utiliza un keyfile debe protegerse especialmente frente a acceso no autorizado o pérdida accidental. Sin el keyfile no es posible acceder a los datos en texto claro.

Un volumen de invitado creado dentro de un dataset cifrado tendrá configurada su propiedad encryptionroot. El material de clave solo necesita cargarse una vez por cada encryptionroot para estar disponible para todos los datasets cifrados que cuelguen de él.

Para obtener más detalles y usos avanzados, consulte las propiedades encryptionroot, encryption, keylocation, keyformat y keystatus; los comandos zfs load-key, zfs unload-key y zfs change-key; y la sección Encryption de man zfs.

3.9.11 Compresión en ZFS

Cuando la compresión está habilitada en un dataset, ZFS intenta comprimir todos los bloques nuevos antes de escribirlos y los descomprime al leerlos. Los datos ya existentes no se comprimen retroactivamente.

Para habilitar la compresión:

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

Se recomienda utilizar el algoritmo lz4 porque añade muy poca sobrecarga de CPU. También están disponibles otros algoritmos como lzjb y gzip-N, donde N es un entero de 1 -más rápido- a 9 -mejor relación de compresión-. Dependiendo del algoritmo y de lo compresibles que sean los datos, habilitar compresión puede incluso aumentar el rendimiento de E/S.

Puede desactivar la compresión en cualquier momento:

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

De nuevo, el cambio solo afectará a los bloques nuevos.

3.9.12 ZFS Special Device

Desde la versión 0.8.0, ZFS admite dispositivos special. Un special device de un pool se utiliza para almacenar metadatos, tablas de deduplicación y, opcionalmente, bloques de archivos pequeños.

Un special device puede mejorar la velocidad de un pool formado por discos mecánicos lentos cuando existe una gran cantidad de cambios en metadatos. Por ejemplo, las cargas de trabajo que crean, actualizan o eliminan muchos archivos se benefician de su presencia. Los datasets ZFS también pueden configurarse para almacenar archivos pequeños completos en el special device, mejorando aún más el rendimiento. Utilice SSD rápidos para este dispositivo.

IMPORTANTE: La redundancia del special device debería ser equivalente a la del pool, ya que este dispositivo constituye un punto de fallo para todo el pool.

ADVERTENCIA: Añadir un special device a un pool no puede deshacerse.

Crear un pool con special device en RAID-1:

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

Añadir un special device a un pool existente con RAID-1:

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

Los datasets ZFS exponen la propiedad special_small_blocks=<size>. size puede ser 0 para deshabilitar el almacenamiento de bloques pequeños en el special device o una potencia de dos entre 512B y 1M. Tras establecerla, los nuevos bloques de archivos menores que size se asignarán al special device.

IMPORTANTE: Si special_small_blocks es mayor o igual que recordsize -128K de forma predeterminada-, todos los datos se escribirán en el special device. Utilícelo con precaución.

Establecer special_small_blocks en un pool cambia el valor predeterminado de esa propiedad para todos los datasets ZFS hijos; por ejemplo, todos los contenedores del pool optarán por almacenar bloques de archivos pequeños.

Activar a nivel de pool todos los archivos menores que bloques de 4K:

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

Activarlo para un único dataset:

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

Desactivarlo para un único dataset:

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

3.9.13 Funciones de los pools ZFS

Los cambios en el formato en disco de ZFS solo se realizan entre cambios de versión principal y se especifican mediante funciones o features. Todas las funciones y el mecanismo general están documentados en la página de manual correspondiente a zpool-features.

Como habilitar nuevas funciones puede hacer que un pool no pueda importarse con una versión antigua de ZFS, esta operación debe realizarse explícitamente por el administrador ejecutando zpool upgrade; consulte la página de manual zpool-upgrade(8).

Salvo que necesite utilizar alguna función nueva, habilitarlas no ofrece ninguna ventaja. De hecho, tiene varias desventajas:

- Un sistema con raíz sobre ZFS que todavía arranque mediante GRUB dejará de poder arrancar si se activa una función nueva en rpool, debido a la implementación incompatible de ZFS en GRUB.
- El sistema no podrá importar un pool actualizado cuando arranque con un kernel antiguo que todavía incluya módulos ZFS antiguos.
- Arrancar una ISO antigua de Proxmox VE para reparar un sistema que no inicia tampoco funcionará.

IMPORTANTE: No actualice rpool si su sistema todavía arranca mediante GRUB, ya que podría dejarlo sin posibilidad de arranque. Esto incluye sistemas instalados antes de Proxmox VE 5.4 y sistemas que arrancan mediante BIOS heredado.

Habilitar nuevas funciones para un pool ZFS:

```
# zpool upgrade <pool>
```

3.10 BTRFS

ADVERTENCIA: La integración de BTRFS se encuentra actualmente en estado de technology preview en Proxmox VE.

BTRFS es un sistema de archivos moderno copy-on-write soportado nativamente por el kernel Linux. Implementa funciones como snapshots, RAID integrado y autorrecuperación mediante checksums de datos y metadatos. Desde Proxmox VE 7.0, BTRFS se ofrece como opción para el sistema de archivos raíz.

Ventajas generales de BTRFS

- Configuración principal del sistema casi idéntica a la configuración tradicional basada en ext4.
- Snapshots.
- Compresión de datos a nivel de sistema de archivos.
- Clones copy-on-write.
- RAID0, RAID1 y RAID10.
- Protección frente a corrupción de datos.
- Autorrecuperación.
- Soporte nativo por parte del kernel Linux.

Limitaciones

- Los niveles RAID 5/6 son experimentales y peligrosos; consulte el estado de BTRFS indicado por el proyecto.

3.10.1 Instalación como sistema de archivos raíz

Al instalar mediante el instalador de Proxmox VE, puede elegirse BTRFS como sistema de archivos raíz. El tipo de RAID debe seleccionarse durante la instalación:

Nivel	Descripción
RAID0	Striping. La capacidad es la suma de las capacidades de todos los discos, pero no aporta redundancia; el fallo de una sola unidad inutiliza el volumen.
RAID1	Mirroring. Los datos se escriben de forma idéntica en todos los discos. Requiere al menos 2 discos del mismo tamaño. La capacidad resultante es la de un único disco.
RAID10	Combinación de RAID0 y RAID1. Requiere al menos 4 discos.

El instalador particiona automáticamente los discos y crea un subvolumen adicional en /var/lib/pve/local-btrfs. Para utilizarlo con las herramientas de Proxmox VE, crea la siguiente configuración en /etc/pve/storage.cfg:

```
dir: local
        path /var/lib/vz
        content iso,vztmpl,backup
        disable

btrfs: local-btrfs
        path /var/lib/pve/local-btrfs
        content iso,vztmpl,backup,images,rootdir
```

Esto deshabilita explícitamente el almacenamiento local predeterminado en favor de una entrada de almacenamiento específica para BTRFS sobre el subvolumen adicional.

El comando btrfs se utiliza para configurar y administrar el sistema de archivos. Después de la instalación, el siguiente comando enumera los subvolúmenes adicionales:

```
# btrfs subvolume list /
ID 256 gen 6 top level 5 path var/lib/pve/local-btrfs
```

3.10.2 Administración de BTRFS

Este apartado ofrece algunos ejemplos de uso para tareas habituales.

Crear un sistema de archivos BTRFS

Para crear sistemas de archivos BTRFS se utiliza mkfs.btrfs. Los parámetros -d y -m establecen el perfil de datos y metadatos respectivamente. Con el parámetro opcional -L puede establecerse una etiqueta. En general se admiten los modos single, raid0, raid1 y raid10.

Crear un sistema de archivos BTRFS en un único disco /dev/sdb con la etiqueta My-Storage:

```
# mkfs.btrfs -m single -d single -L My-Storage /dev/sdb
```

Crear un RAID1 en las particiones /dev/sdb1 y /dev/sdc1:

```
# mkfs.btrfs -m raid1 -d raid1 -L My-Storage /dev/sdb1 /dev/sdc1
```

Montar un sistema de archivos BTRFS

El nuevo sistema de archivos puede montarse manualmente:

```
# mkdir /my-storage
# mount /dev/sdb /my-storage
```

También puede añadirse a /etc/fstab, como cualquier otro punto de montaje, para montarlo automáticamente al arrancar. Se recomienda evitar rutas de dispositivos de bloque y utilizar el valor UUID que mostró mkfs.btrfs, especialmente cuando hay más de un disco en la configuración BTRFS.

Archivo: /etc/fstab

```
# ... se omiten otros puntos de montaje por brevedad

# se recomienda utilizar el UUID mostrado por mkfs.btrfs
UUID=e2c0c3ff-2114-4f54-b767-3a203e49f6f3 /my-storage btrfs defaults 0 0
```

CONSEJO: Si ya no dispone del UUID, puede utilizar la herramienta blkid para enumerar las propiedades de los dispositivos de bloques.

Después puede provocar el primer montaje ejecutando:

```
mount /my-storage
```

A partir del siguiente reinicio, el sistema lo montará automáticamente durante el arranque.

Añadir un sistema de archivos BTRFS a Proxmox VE

Puede añadir un sistema de archivos BTRFS existente a Proxmox VE mediante la interfaz web o con la CLI:

```
pvesm add btrfs my-storage --path /my-storage
```

Crear un subvolumen

Crear un subvolumen lo vincula a una ruta del sistema de archivos BTRFS, donde aparecerá como un directorio normal:

```
# btrfs subvolume create /some/path
```

Después, /some/path se comportará como un directorio normal.

Eliminar un subvolumen

A diferencia de los directorios eliminados mediante rmdir, los subvolúmenes no tienen que estar vacíos para poder eliminarlos con btrfs:

```
# btrfs subvolume delete /some/path
```

Crear un snapshot de un subvolumen

BTRFS no distingue realmente entre snapshots y subvolúmenes normales, por lo que tomar un snapshot también puede entenderse como crear una copia arbitraria de un subvolumen. Por convención, Proxmox VE utiliza el indicador de solo lectura al crear snapshots de discos o subvolúmenes de invitados, aunque después puede modificarse.

```
# btrfs subvolume snapshot -r /some/path /a/new/path
```

Esto crea en /a/new/path un "clon" de solo lectura del subvolumen /some/path. Cualquier modificación posterior de /some/path provoca que los datos modificados se copien antes de la modificación. Si se omite la opción -r, ambos subvolúmenes serán escribibles.

Habilitar compresión

De forma predeterminada, BTRFS no comprime los datos. Para activar la compresión puede añadirse la opción de montaje compress. Los datos ya escritos no se comprimirán retroactivamente.

De forma predeterminada, rootfs aparecerá en /etc/fstab así:

```
UUID=<uuid of your root file system> / btrfs defaults 0 1
```

Puede añadir compress=zstd, compress=lzo o compress=zlib a defaults:

```
UUID=<uuid of your root file system> / btrfs defaults,compress=zstd 0 1
```

El cambio tendrá efecto después de reiniciar.

Comprobar el uso de espacio

La herramienta clásica df puede mostrar valores confusos en algunas configuraciones BTRFS. Para obtener una estimación mejor utilice btrfs filesystem usage /PATH, por ejemplo:

```
# btrfs fi usage /my-storage
```

3.11 Gestión de nodos Proxmox

La herramienta de gestión de nodos de Proxmox VE, pvenode, permite controlar ajustes y recursos específicos de cada nodo.

Actualmente pvenode permite establecer la descripción de un nodo, ejecutar diversas operaciones masivas sobre sus invitados, consultar el historial de tareas del nodo y gestionar sus certificados SSL, utilizados por la API y la GUI web a través de pveproxy.

3.11.1 Wake-on-LAN

Wake-on-LAN (WoL) permite encender a través de la red un equipo suspendido enviándole un magic packet. Al menos una NIC debe soportar esta función y la opción correspondiente debe estar habilitada en la configuración de firmware BIOS/UEFI. El nombre de la opción puede variar desde Enable Wake-on-Lan hasta Power On By PCIE Device; consulte el manual del fabricante de la placa base si tiene dudas.

ethtool permite comprobar la configuración WoL de <interface>:

```
ethtool <interface> | grep Wake-on
```

pvenode permite despertar mediante WoL miembros suspendidos de un clúster:

```
pvenode wakeonlan <node>
```

Este comando difunde el magic packet WoL por el puerto UDP 9 e incluye la dirección MAC de <node> obtenida de la propiedad wakeonlan. Puede establecerse con:

```
pvenode config set -wakeonlan XX:XX:XX:XX:XX:XX
```

La interfaz utilizada para enviar el paquete WoL se determina a partir de la ruta predeterminada. Puede sobrescribirse con bind-interface:

```
pvenode config set -wakeonlan XX:XX:XX:XX:XX:XX,bind-interface=<iface-name>
```

La dirección de broadcast, cuyo valor predeterminado es 255.255.255.255, también puede modificarse explícitamente:

```
pvenode config set -wakeonlan XX:XX:XX:XX:XX:XX,broadcast-address=<broadcast-address>
```

3.11.2 Historial de tareas

Al resolver problemas del servidor -por ejemplo, backups fallidos-, puede ser útil disponer de un registro de tareas ejecutadas anteriormente. Proxmox VE permite consultar el historial del nodo mediante pvenode task.

Puede obtener una lista filtrada de tareas finalizadas con el subcomando list. Para listar, por ejemplo, las tareas relacionadas con la VM 100 que terminaron con error:

```
pvenode task list --errors --vmid 100
```

Después puede imprimirse el log de una tarea utilizando su UPID:

```
pvenode task log UPID:pve1:00010D94:001CA6EA:6124E1B9:vzdump:100:root@pam:
```

3.11.3 Gestión masiva de energía de invitados

Cuando existen muchas VMs o contenedores, puede iniciarlos o detenerlos mediante operaciones masivas utilizando los subcomandos startall y stopall de pvenode. Por defecto, pvenode startall solo inicia los invitados configurados para arrancar automáticamente con el host, pero este comportamiento puede anularse mediante --force. Ambos comandos incluyen además la opción --vms para limitar la operación a VMID concretos.

Para iniciar las VMs 100, 101 y 102 independientemente de si tienen onboot configurado:

```
pvenode startall --vms 100,101,102 --force
```

Para detener estos invitados -y cualquier otro invitado que esté ejecutándose-:

```
pvenode stopall
```

NOTA: stopall intenta primero realizar un apagado limpio y espera hasta que todos los invitados se hayan apagado correctamente o expire un timeout configurable -3 minutos de forma predeterminada-. Después, salvo que force-stop se establezca explícitamente en 0 (false), se detendrán de forma forzada todos los invitados que sigan ejecutándose.

3.11.4 Retraso antes de arrancar el primer invitado

Si sus VMs o contenedores dependen de recursos externos que tardan en arrancar, como un servidor NFS, puede configurar por nodo un retraso entre el arranque de Proxmox VE y el inicio del primer invitado configurado para autoarranque. En el siguiente ejemplo, 10 representa el retraso en segundos:

```
pvenode config set --startall-onboot-delay 10
```

3.11.5 Migración masiva de invitados

Cuando una actualización requiere migrar todos los invitados de un nodo a otro, pvenode ofrece el subcomando migrateall. De forma predeterminada migrará todos los invitados del sistema al nodo de destino, aunque puede limitarse a un conjunto de invitados.

Para migrar las VMs 100, 101 y 102 al nodo pve2 y permitir live migration de discos locales:

```
pvenode migrateall pve2 --vms 100,101,102 --with-local-disks
```

3.11.6 Objetivo de uso de RAM para ballooning

El porcentaje objetivo para la asignación automática de memoria es 80 % de forma predeterminada. Puede personalizarse por nodo mediante la propiedad ballooning-target. Para establecer, por ejemplo, un objetivo del 90 % de uso de la memoria del host:

```
pvenode config set --ballooning-target 90
```

3.12 Gestión de certificados

3.12.1 Certificados para la comunicación interna del clúster

Cada clúster Proxmox VE crea de forma predeterminada su propia Certificate Authority (CA) autofirmada y genera un certificado para cada nodo, firmado por dicha CA. Estos certificados se utilizan para la comunicación cifrada con el servicio pveproxy del clúster y para la función Shell/Console cuando se utiliza SPICE.

El certificado y la clave de la CA se almacenan en Proxmox Cluster File System (pmxcfs).

Para distribuir la CA del clúster a los puestos de administración y evitar avisos del navegador, copie /etc/pve/pve-root-ca.pem desde cualquier nodo del clúster por un canal alternativo -por ejemplo, scp o una sesión SSH existente- e impórtelo en el almacén de confianza del sistema operativo o del navegador. No existe un endpoint de API ni de interfaz web para descargar el archivo.

3.12.2 Certificados para la API y la GUI web

La API REST y la GUI web las proporciona el servicio pveproxy, que se ejecuta en cada nodo.

Existen las siguientes opciones para el certificado utilizado por pveproxy:

16. De forma predeterminada se utiliza el certificado específico del nodo ubicado en /etc/pve/nodes/NODENAME/pve-ssl.pem. Está firmado por la CA del clúster y, por tanto, los navegadores y sistemas operativos no confían en él automáticamente.
17. Utilizar un certificado proporcionado externamente, por ejemplo firmado por una CA comercial.
18. Utilizar ACME -Let's Encrypt- para obtener un certificado de confianza con renovación automática; esta función está integrada en la API y en la interfaz web de Proxmox VE.

Para las opciones 2 y 3 se utilizan /etc/pve/local/pveproxy-ssl.pem y /etc/pve/local/pveproxy-ssl.key; la clave no debe estar protegida por contraseña.

NOTA: /etc/pve/local es un enlace simbólico específico del nodo hacia /etc/pve/nodes/NODENAME.

Los certificados se gestionan con el comando de gestión de nodos Proxmox VE; consulte la página de manual pvenode(1).

ADVERTENCIA: No sustituya ni modifique manualmente los certificados de nodo generados automáticamente en /etc/pve/local/pve-ssl.pem y /etc/pve/local/pve-ssl.key, ni los archivos de la CA del clúster /etc/pve/pve-root-ca.pem y /etc/pve/priv/pve-root-ca.key.

3.12.3 Cargar un certificado personalizado

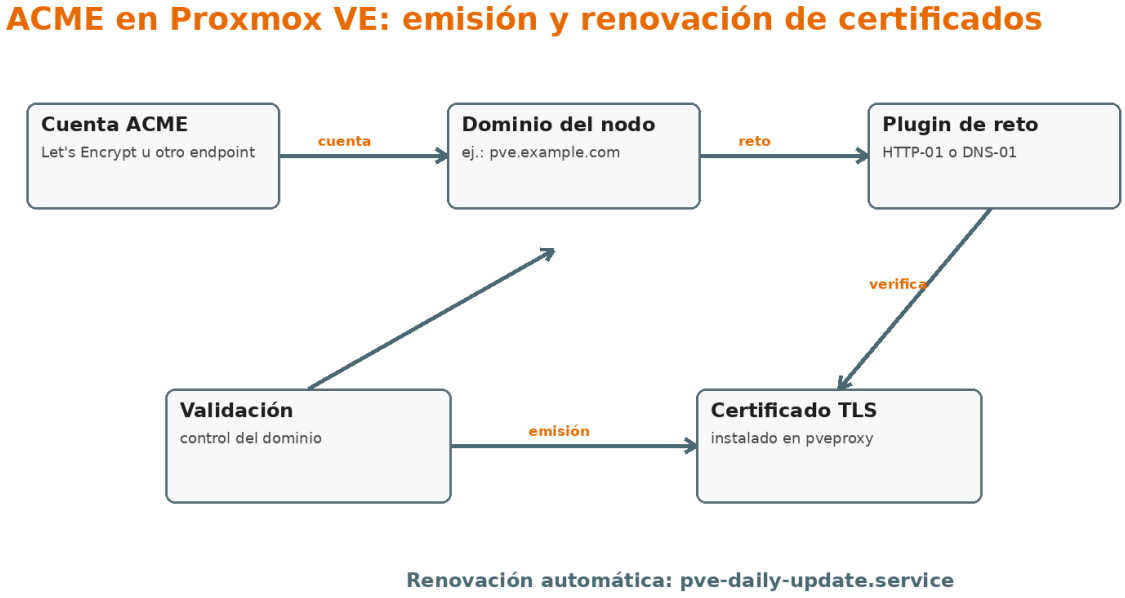
Si ya dispone de un certificado que desea utilizar para un nodo Proxmox VE, puede cargarlo directamente desde la interfaz web.

NOTA: El archivo de clave del certificado, si se proporciona, no debe estar protegido por contraseña.

3.12.4 Certificados de confianza mediante Let's Encrypt (ACME)

Proxmox VE incluye una implementación del protocolo Automatic Certificate Management Environment (ACME), que permite utilizar un proveedor ACME como Let's Encrypt para configurar fácilmente certificados TLS aceptados y considerados de confianza por los sistemas operativos y navegadores web modernos.

Actualmente se implementan dos endpoints ACME de Let's Encrypt: producción y staging. El cliente ACME admite la validación de retos http-01 mediante un servidor web integrado y la validación dns-01 mediante un plugin DNS compatible con los endpoints de API DNS soportados por acme.sh.



Flujo simplificado de ACME en Proxmox VE: cuenta, dominio, reto, validación, certificado y renovación automática.

Cuenta ACME

Debe registrar una cuenta ACME por clúster en el endpoint que quiera utilizar. La dirección de correo asociada servirá como punto de contacto para avisos de renovación pendiente y notificaciones similares del endpoint ACME.

Puede registrar y desactivar cuentas ACME desde Datacenter -> ACME en la interfaz web o mediante pvenode:

```
pvenode acme account register account-name mail@example.com
```

CONSEJO: Debido a los límites de solicitudes, utilice el entorno staging de Let's Encrypt para pruebas o la primera vez que configure ACME.

Plugins ACME

La función de los plugins ACME es proporcionar una verificación automática de que usted -y, por tanto, el clúster Proxmox VE que administra- es el propietario real de un dominio. Es el bloque básico de la gestión automática de certificados.

El protocolo ACME define distintos tipos de retos. En http-01, un servidor web proporciona un archivo con un contenido concreto para demostrar que controla un dominio. A veces esto no es posible, por limitaciones técnicas o porque la dirección del registro no es accesible desde Internet. En esos casos puede utilizarse dns-01, que se completa creando un registro DNS concreto en la zona del dominio.

Proxmox VE admite ambos tipos de reto de forma nativa. Puede configurar plugins desde Datacenter -> ACME o mediante pvenode acme plugin add. Las configuraciones se almacenan en /etc/pve/priv/acme/plugins.cfg y un plugin queda disponible para todos los nodos del clúster.

Dominios del nodo

Cada dominio es específico de un nodo. Puede añadir nuevas entradas o administrar las existentes desde Node -> Certificates o mediante pvenode config.

Después de configurar los dominios deseados y asegurarse de que está seleccionada la cuenta ACME apropiada, puede solicitar el certificado desde la interfaz web. Si la operación finaliza correctamente, la interfaz se recargará tras 10 segundos. La renovación se realizará automáticamente.

3.12.5 Plugin ACME para reto HTTP

Siempre existe un plugin standalone configurado implícitamente para validar retos http-01 mediante el servidor web integrado que escucha en el puerto 80.

NOTA: standalone significa que puede proporcionar por sí solo la validación, sin depender de servicios de terceros. Por ello, este plugin también funciona en nodos de clúster.

Para utilizarlo con Let's Encrypt deben cumplirse los siguientes requisitos:

- Aceptar los términos de servicio de Let's Encrypt al registrar la cuenta.
- El puerto 80 del nodo debe ser accesible desde Internet.
- No debe existir ningún otro servicio escuchando en el puerto 80.
- El (sub)dominio solicitado debe resolver a una IP pública del nodo.

3.12.6 Plugin ACME para reto DNS mediante API

En sistemas donde no es posible o no se desea el acceso externo para validar mediante http-01, puede utilizarse dns-01. Este método requiere un servidor DNS que permita aprovisionar registros TXT mediante una API.

Configurar APIs DNS ACME para la validación

Proxmox VE reutiliza los plugins DNS desarrollados para el proyecto acme.sh. Consulte su documentación para obtener detalles sobre la configuración de APIs concretas. La forma más sencilla de configurar un plugin nuevo con API DNS es utilizar Datacenter -> ACME en la interfaz web.

Seleccione DNS como tipo de reto. Después elija el proveedor de API e introduzca las credenciales necesarias para acceder a la cuenta mediante su API. El retraso de validación determina el número de segundos entre la creación del registro DNS y la solicitud de validación al proveedor ACME, ya que a menudo los proveedores necesitan tiempo para propagar el registro en su infraestructura.

CONSEJO: Consulte la wiki "How to use DNS API" de acme.sh para obtener información más detallada sobre cómo conseguir las credenciales API de su proveedor.

Como existen numerosos proveedores DNS y endpoints de API, Proxmox VE genera automáticamente el formulario de credenciales para algunos proveedores. Para el resto se mostrará un área de texto más grande; copie en ella los pares de credenciales KEY=VALUE.

Referencia del manual original: acme.sh - https://github.com/acmesh-official/acme.sh

Validación DNS mediante alias CNAME

Puede utilizarse un modo especial de alias para gestionar la validación en otro dominio o servidor DNS cuando el DNS principal no admite aprovisionamiento mediante API. Configure manualmente un CNAME permanente de _acme-challenge.domain1.example hacia _acme-challenge.domain2.example y establezca la propiedad alias en la clave acmedomainX correspondiente del archivo de configuración del nodo Proxmox VE con el valor domain2.example. Así, el servidor DNS de domain2.example podrá validar los retos de domain1.example.

Combinación de plugins

Es posible combinar validación http-01 y dns-01 cuando un nodo es accesible mediante varios dominios con requisitos o capacidades de aprovisionamiento DNS distintos. También pueden mezclarse APIs DNS de distintos proveedores o instancias especificando una instancia de plugin distinta para cada dominio.

CONSEJO: Acceder al mismo servicio mediante varios dominios aumenta la complejidad y debería evitarse siempre que sea posible.

3.12.7 Renovación automática de certificados ACME

Si un nodo se ha configurado correctamente con un certificado proporcionado por ACME -mediante pvenode o mediante la GUI-, pve-daily-update.service lo renovará automáticamente. Actualmente se intenta renovar cuando el certificado ya ha caducado o va a caducar en los próximos 30 días.

NOTA: Si utiliza un directorio personalizado que emite certificados de corta duración, puede ser aconsejable deshabilitar el retraso aleatorio de la unidad pve-daily-update.timer para evitar perder una renovación después de un reinicio.

3.12.8 Ejemplos ACME con pvenode

Ejemplo: utilizar certificados Let's Encrypt

```
root@proxmox:~# pvenode acme account register default mail@example.invalid
Directory endpoints:
0) Let's Encrypt V2 (https://acme-v02.api.letsencrypt.org/directory)
1) Let's Encrypt V2 Staging (https://acme-staging-v02.api.letsencrypt.org/directory)
2) Custom
Enter selection: 1

Terms of Service: https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf
Do you agree to the above terms? [y|N]y
...
Task OK
root@proxmox:~# pvenode config set --acme domains=example.invalid
root@proxmox:~# pvenode acme cert order
Loading ACME account details
Placing ACME order
...
Status is 'valid'!

All domains validated!
...
Downloading certificate
Setting pveproxy certificate and key
Restarting pveproxy
Task OK
```

Ejemplo: configurar la API de OVH para validar un dominio

NOTA: Los pasos de registro de la cuenta son los mismos independientemente de los plugins utilizados y no se repiten aquí.

NOTA: OVH_AK y OVH_AS deben obtenerse de OVH de acuerdo con la documentación de su API.

Primero debe obtener la información necesaria para que usted y Proxmox VE puedan acceder a la API:

```
root@proxmox:~# cat /path/to/api-token
OVH_AK=XXXXXXXXXXXXXXXX
OVH_AS=YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
root@proxmox:~# source /path/to/api-token
root@proxmox:~# curl -XPOST -H"X-Ovh-Application: $OVH_AK" -H "Content-type: application/json" \
https://eu.api.ovh.com/1.0/auth/credential -d '{
  "accessRules": [
    {"method": "GET","path": "/auth/time"},
    {"method": "GET","path": "/domain"},
    {"method": "GET","path": "/domain/zone/*"},
    {"method": "GET","path": "/domain/zone/*/record"},
    {"method": "POST","path": "/domain/zone/*/record"},
    {"method": "POST","path": "/domain/zone/*/refresh"},
    {"method": "PUT","path": "/domain/zone/*/record/"},
    {"method": "DELETE","path": "/domain/zone/*/record/*"}
  ]
}'
{"consumerKey":"ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ","state":"pendingValidation",
 "validationUrl":"https://eu.api.ovh.com/auth/?credentialToken=AAAAAAAA..."}
```

Abra la URL de validación y siga las instrucciones para vincular la Application Key con la cuenta/Consumer Key. Después:

```
root@proxmox:~# echo "OVH_CK=ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ" >> /path/to/api-token
```

Ya puede configurar el plugin ACME:

```
root@proxmox:~# pvenode acme plugin add dns example_plugin --api ovh --data /path/to/api_token
root@proxmox:~# pvenode acme plugin config example_plugin
key     value
api     ovh
data    OVH_AK=XXXXXXXXXXXXXXXX
        OVH_AS=YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
        OVH_CK=ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
digest  867fcf556363ca1bea866863093fcab83edf47a1
plugin  example_plugin
type    dns
```

Por último configure el dominio para el que desea obtener certificados y realice la solicitud:

```
root@proxmox:~# pvenode config set -acmedomain0 example.proxmox.com,plugin=example_plugin
root@proxmox:~# pvenode acme cert order
Loading ACME account details
Placing ACME order
Order URL: https://acme-staging-v02.api.letsencrypt.org/acme/order/11111111/22222222
Getting authorization details from 'https://acme-staging-v02.api.letsencrypt.org/acme/authz-v3/33333333'
The validation for example.proxmox.com is pending!
[Wed Apr 22 09:25:30 CEST 2020] Using OVH endpoint: ovh-eu
[Wed Apr 22 09:25:30 CEST 2020] Checking authentication
[Wed Apr 22 09:25:30 CEST 2020] Consumer key is ok.
[Wed Apr 22 09:25:31 CEST 2020] Adding record
[Wed Apr 22 09:25:32 CEST 2020] Added, sleep 10 seconds.
Add TXT record: _acme-challenge.example.proxmox.com
Triggering validation
Sleeping for 5 seconds
Status is 'valid'!
[Wed Apr 22 09:25:48 CEST 2020] Using OVH endpoint: ovh-eu
[Wed Apr 22 09:25:48 CEST 2020] Checking authentication
[Wed Apr 22 09:25:48 CEST 2020] Consumer key is ok.
Remove TXT record: _acme-challenge.example.proxmox.com

All domains validated!

Creating CSR
Checking order status
Order is ready, finalizing order
valid!

Downloading certificate
Setting pveproxy certificate and key
Restarting pveproxy
Task OK
```

Ejemplo: cambiar de staging al directorio ACME normal

Cambiar el directorio ACME de una cuenta existente no está soportado. Como Proxmox VE permite más de una cuenta, puede crear una nueva con el directorio ACME de producción -de confianza- como endpoint. También puede desactivar la cuenta de staging y volver a crearla.

Ejemplo para cambiar la cuenta ACME predeterminada de staging a producción mediante pvenode:

```
root@proxmox:~# pvenode acme account deactivate default
Renaming account file from '/etc/pve/priv/acme/default' to '/etc/pve/priv/acme/_deactivated_default_4'
Task OK

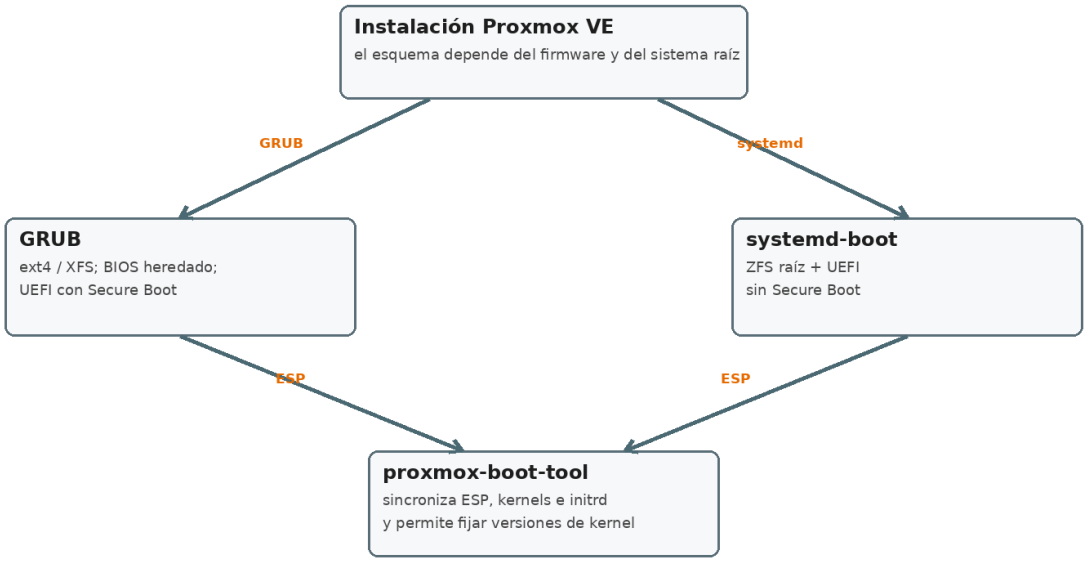
root@proxmox:~# pvenode acme account register default example@proxmox.com
Directory endpoints:
0) Let's Encrypt V2 (https://acme-v02.api.letsencrypt.org/directory)
1) Let's Encrypt V2 Staging (https://acme-staging-v02.api.letsencrypt.org/directory)
2) Custom
Enter selection: 0

Terms of Service: https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf
Do you agree to the above terms? [y|N]y
...
Task OK
```

3.13 Bootloader del host

Proxmox VE utiliza actualmente uno de dos bootloaders en función del esquema de disco seleccionado en el instalador. En sistemas EFI instalados con ZFS como sistema de archivos raíz se utiliza systemd-boot, salvo que Secure Boot esté habilitado. El resto de despliegues utiliza el bootloader GRUB estándar; esto suele aplicarse también a sistemas instalados sobre Debian.

Arranque del host: GRUB, systemd-boot y Secure Boot



Resumen del arranque: la combinación de firmware, sistema de archivos raíz y Secure Boot determina el bootloader; proxmox-boot-tool sincroniza las ESP cuando corresponde.

3.13.1 Esquema de particiones utilizado por el instalador

El instalador de Proxmox VE crea 3 particiones en todos los discos seleccionados para la instalación:

- Una BIOS Boot Partition de 1 MB -tipo EF02 en gdisk-.
- Una EFI System Partition (ESP) de 512 MB -tipo EF00 en gdisk-.
- Una tercera partición que ocupa el tamaño definido por el parámetro hdsize o el espacio restante, utilizada por el tipo de almacenamiento elegido.

Los sistemas con ZFS como sistema de archivos raíz arrancan con una imagen de kernel e initrd almacenada en la EFI System Partition de 512 MB. En sistemas BIOS heredados y sistemas EFI con Secure Boot se utiliza GRUB; en sistemas EFI sin Secure Boot se utiliza systemd-boot. Ambos se instalan y configuran para apuntar a las ESP.

En sistemas arrancados mediante GRUB, GRUB en modo BIOS -target i386-pc- se instala en la BIOS Boot Partition de todos los discos seleccionados.

NOTA: Esto se aplica a hosts x86. En arm64, Proxmox VE solo arranca mediante UEFI: no existe BIOS Boot Partition ni target i386-pc para GRUB. GRUB se instala en modo EFI utilizando el target arm64-efi.

3.13.2 Sincronizar el contenido de la ESP con proxmox-boot-tool

proxmox-boot-tool es una utilidad utilizada para mantener correctamente configurado y sincronizado el contenido de las EFI System Partitions. Copia determinadas versiones del kernel a todas las ESP y configura el bootloader correspondiente para arrancar desde las ESP formateadas como vfat. En un sistema con ZFS como raíz, esto permite utilizar todas las funciones opcionales del root pool en lugar de limitarse al subconjunto implementado también por ZFS en GRUB o de tener que crear un pequeño boot-pool independiente.

En configuraciones redundantes, el instalador particiona todos los discos con una ESP. Esto garantiza que el sistema pueda arrancar aunque falle el primer dispositivo de arranque o aunque la BIOS solo pueda arrancar desde un disco concreto.

Las ESP no permanecen montadas durante el funcionamiento normal. Esto ayuda a prevenir corrupción del sistema de archivos vfat en caso de caída del sistema y elimina la necesidad de adaptar manualmente /etc/fstab si falla el dispositivo de arranque primario.

proxmox-boot-tool se ocupa de las siguientes tareas:

- Formatear y preparar una partición nueva.
- Copiar y configurar nuevas imágenes de kernel e initrd en todas las ESP registradas.
- Sincronizar la configuración durante actualizaciones del kernel y otras tareas de mantenimiento.
- Gestionar la lista de versiones de kernel que se sincronizan.
- Configurar el bootloader para arrancar una versión concreta del kernel -pinning-.

Puede ver las ESP configuradas y su estado ejecutando:

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

Preparar una nueva partición como ESP sincronizada

Para formatear e inicializar una partición como ESP sincronizada -por ejemplo después de sustituir un vdev averiado en rpool o al convertir un sistema antiguo anterior al mecanismo de sincronización- puede utilizarse proxmox-boot-tool del paquete proxmox-kernel.

ADVERTENCIA: El comando format formateará <partition>. Asegúrese de indicar el dispositivo/partición correcto.

Para formatear /dev/sda2 como ESP:

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

Para incorporar una ESP existente y desmontada situada en /dev/sda2 al mecanismo de sincronización de kernels:

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

o, para forzar GRUB en lugar de systemd-boot -por ejemplo, para Secure Boot-:

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

Después, /etc/kernel/proxmox-boot-uuids debería contener una nueva línea con el UUID de la partición añadida. El comando init también ejecutará automáticamente un refresh de todas las ESP configuradas.

Referencia del manual original sobre ZFS raíz con GRUB: https://openzfs.github.io/openzfs-docs/Getting%20Started/Debian/Debian%20Bookworm%20Root%20on%20ZFS.html

Actualizar la configuración en todas las ESP

Para copiar y configurar todos los kernels arrancables y mantener sincronizadas todas las ESP indicadas en /etc/kernel/proxmox-boot-uuids:

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

Es el equivalente a ejecutar update-grub en sistemas con ext4 o xfs como sistema raíz. Es necesario tras modificar la línea de comandos del kernel o cuando se desea resincronizar todos los kernels e initrds.

NOTA: Tanto update-initramfs como apt ejecutarán automáticamente refresh cuando sea necesario.

Versiones de kernel consideradas por proxmox-boot-tool

De forma predeterminada se configuran las siguientes versiones:

- El kernel que se está ejecutando actualmente.
- La versión nueva que se está instalando mediante una actualización de paquetes.
- Los dos kernels instalados más recientes.
- La versión más reciente de la penúltima serie de kernels -por ejemplo 5.0, 5.3-, cuando corresponda.
- Cualquier kernel seleccionado manualmente.

Mantener manualmente un kernel arrancable

Para añadir una imagen de kernel e initrd concreta a la lista de kernels arrancables, utilice proxmox-boot-tool kernel add. Por ejemplo:

```
# proxmox-boot-tool kernel add 5.0.15-1-pve
```

proxmox-boot-tool kernel list muestra todas las versiones seleccionadas actualmente:

```
# proxmox-boot-tool kernel list
Manually selected kernels:
5.0.15-1-pve

Automatically selected kernels:
5.0.12-1-pve
4.15.18-18-pve
```

Para quitar un kernel de la lista de seleccionados manualmente:

```
# proxmox-boot-tool kernel remove 5.0.15-1-pve
```

NOTA: Después de añadir o eliminar manualmente un kernel debe ejecutar proxmox-boot-tool refresh para actualizar todas las EFI System Partitions.

3.13.3 Determinar qué bootloader se utiliza

La forma más sencilla y fiable de determinar qué bootloader se está utilizando es observar el proceso de arranque del nodo Proxmox VE: aparecerá el recuadro azul de GRUB o la interfaz sencilla, negra sobre blanco, de systemd-boot.

Determinar el bootloader desde un sistema ya arrancado puede no ser totalmente preciso. La forma más segura es ejecutar:

```
# efibootmgr -v
```

Si muestra un mensaje indicando que las variables EFI no están soportadas, se utiliza GRUB en modo BIOS/Legacy.

Si la salida contiene una línea similar a la siguiente, se utiliza GRUB en modo UEFI:

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

Si la entrada de arranque activa apunta a shimx64.efi, se utiliza GRUB en UEFI con Secure Boot habilitado:

```
Boot0005* proxmox               [...] File(\EFI\proxmox\shimx64.efi)
```

Si aparece una línea similar a la siguiente, se utiliza systemd-boot:

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

NOTA: Los nombres anteriores corresponden a hosts x86. En arm64, los binarios equivalentes se denominan grubaa64.efi, shimaa64.efi y systemd-bootaa64.efi.

Ejecutando proxmox-boot-tool status puede comprobar si proxmox-boot-tool está configurado, lo que constituye una buena indicación de cómo arranca el sistema.

3.13.4 GRUB

GRUB ha sido durante muchos años el estándar de facto para arrancar sistemas Linux y está ampliamente documentado.

Configuración

Los cambios de configuración de GRUB se realizan en /etc/default/grub o mediante fragmentos de configuración en /etc/default/grub.d. Después de modificar la configuración, regenérela ejecutando:

```
# update-grub
```

Referencia del manual original: GNU GRUB Manual - https://www.gnu.org/software/grub/manual/grub/grub.html

3.13.5 systemd-boot

systemd-boot es un bootloader EFI ligero. Lee directamente las imágenes del kernel e initrd desde la EFI System Partition en la que está instalado. La principal ventaja de cargar el kernel directamente desde la ESP es que no necesita volver a implementar controladores para acceder al almacenamiento. En Proxmox VE, proxmox-boot-tool mantiene sincronizada la configuración de las ESP.

Configuración

systemd-boot se configura mediante loader/loader.conf en el directorio raíz de una ESP. Consulte loader.conf(5) para obtener detalles. Cada entrada del bootloader se almacena en un archivo independiente dentro de loader/entries/.

Ejemplo de entry.conf, donde / representa la raíz de la 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
```

3.13.6 Editar la línea de comandos del kernel

Puede modificar la línea de comandos del kernel en los siguientes lugares, según el bootloader utilizado.

GRUB

La línea de comandos del kernel debe colocarse en la variable GRUB_CMDLINE_LINUX_DEFAULT del archivo /etc/default/grub. Al ejecutar update-grub, su contenido se añade a todas las entradas Linux de la configuración de GRUB.

systemd-boot

La línea de comandos del kernel debe escribirse en una única línea en /etc/kernel/cmdline. Para aplicar los cambios, ejecute proxmox-boot-tool refresh; esto la establecerá como línea options de todos los archivos loader/entries/proxmox-*.conf.

La lista completa de parámetros del kernel está disponible en https://www.kernel.org/doc/html/v<YOUR-KERNEL-VERSION>/admin-guide/kernel-parameters.html. Sustituya <YOUR-KERNEL-VERSION> por la versión major.minor, por ejemplo v6.5 para kernels basados en 6.5.

Puede consultar la versión del kernel en la interfaz web -Node -> Summary- o mediante:

```
# uname -r
```

Utilice los dos primeros números de la salida.

3.13.7 Sobrescribir la versión del kernel para el siguiente arranque

Para seleccionar un kernel que no sea actualmente el predeterminado puede:

- Utilizar el menú del bootloader que aparece al comienzo del proceso de arranque.
- Utilizar proxmox-boot-tool para fijar -pin- una versión del kernel una sola vez o de forma permanente hasta que se elimine el pin.

Esto puede ayudar a sortear incompatibilidades entre una versión nueva del kernel y el hardware.

NOTA: El pin debe eliminarse tan pronto como sea posible para que el sistema vuelva a recibir todos los parches de seguridad del kernel más reciente.

Para seleccionar permanentemente la versión 5.15.30-1-pve:

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

CONSEJO: La función de pinning funciona en todos los sistemas Proxmox VE, no solo en los que utilizan proxmox-boot-tool para sincronizar las ESP. Si su sistema no utiliza esa sincronización, puede omitir proxmox-boot-tool refresh al final.

También puede fijar una versión únicamente para el siguiente arranque, por ejemplo para comprobar si un kernel actualizado soluciona un problema:

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

Para eliminar cualquier configuración de pin utilice:

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

unpin también admite --next-boot, aunque solo sirve para borrar un pin establecido con --next-boot. Como ese pin se elimina automáticamente al arrancar, invocarlo manualmente suele tener poca utilidad.

Después de establecer o eliminar versiones fijadas, sincronice el contenido y la configuración de las ESP:

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

CONSEJO: Si ejecuta la herramienta de forma interactiva en sistemas gestionados por proxmox-boot-tool, se le ofrecerá realizar automáticamente el refresh.

3.13.8 Secure Boot

Desde Proxmox VE 8.1, Secure Boot está soportado de forma nativa mediante paquetes firmados e integración con proxmox-boot-tool.

Para que Secure Boot funcione se requieren los siguientes paquetes. Pueden instalarse todos a la vez mediante el metapaquete proxmox-secure-boot-support:

- shim-signed - bootloader shim firmado por Microsoft.
- shim-helpers-amd64-signed - bootloader de fallback y MOKManager, firmados por Proxmox.
- grub-efi-amd64-signed - bootloader GRUB EFI, firmado por Proxmox.
- proxmox-kernel-6.X.Y-Z-pve-signed - imagen del kernel firmada por Proxmox.

NOTA: Los paquetes shim-helpers-amd64-signed y grub-efi-amd64-signed corresponden a hosts x86. En arm64, los equivalentes son shim-helpers-arm64-signed y grub-efi-arm64-signed.

Solo GRUB está soportado de forma nativa como bootloader, ya que otros bootloaders no son actualmente aptos para el proceso de firma de código de Secure Boot. Cualquier instalación nueva de Proxmox VE incluirá automáticamente los paquetes anteriores.

La wiki de Proxmox contiene más detalles sobre el funcionamiento de Secure Boot y sobre cómo personalizar la configuración.

Cambiar una instalación existente a Secure Boot

ADVERTENCIA: Si no se realiza correctamente, este proceso puede dejar la instalación sin posibilidad de arranque. Reinstalar el host configura Secure Boot automáticamente cuando está disponible, sin pasos adicionales. Asegúrese de disponer de un backup funcional y bien probado del host Proxmox VE.

Una instalación UEFI existente puede migrarse a Secure Boot sin reinstalar Proxmox VE desde cero. Primero asegúrese de que todo el sistema está actualizado. Después instale proxmox-secure-boot-support. GRUB creará automáticamente la entrada EFI necesaria para arrancar mediante el shim predeterminado.

Caso systemd-boot

Si se utiliza systemd-boot como bootloader, se necesita configuración adicional. Esto solo ocurre cuando Proxmox VE se instaló con ZFS como sistema de archivos raíz. Para comprobarlo:

```
# findmnt /
```

Si el host utiliza ZFS como raíz, la columna FSTYPE debería contener zfs:

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

A continuación debe localizarse una ESP adecuada utilizando lsblk:

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

La salida puede ser similar a:

```
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  512M  0 part             vfat
`-sda3   8:3    0 31.5G  0 part             zfs_member
sdb      8:16   0   32G  0 disk
|-sdb1   8:17   0 1007K  0 part
|-sdb2   8:18   0  512M  0 part             vfat
`-sdb3   8:19   0 31.5G  0 part             zfs_member
```

En este ejemplo, sda2 y sdb2 son los destinos. Se identifican por su tamaño de 512M y porque su FSTYPE es vfat, en una instalación ZFS RAID-1.

Estas particiones deben prepararse correctamente para arrancar mediante GRUB utilizando proxmox-boot-tool. El comando siguiente debe ejecutarse por separado para cada ESP:

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

Después puede comprobar la configuración con:

```
# efibootmgr -v
```

La lista debería contener una entrada similar a:

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

NOTA: El antiguo bootloader systemd-boot se conserva, pero se dará preferencia a GRUB. De este modo, si el arranque mediante GRUB con Secure Boot no funciona por algún motivo, todavía podrá arrancarse mediante systemd-boot desactivando Secure Boot.

Ahora puede reiniciar el host y habilitar Secure Boot en el firmware UEFI. Tras reiniciar, debería aparecer una nueva entrada denominada proxmox en el menú de arranque UEFI, que utiliza el shim EFI prefirmado.

Si no aparece una entrada proxmox, puede intentar añadirla manualmente -si el firmware lo permite- utilizando el archivo \EFI\proxmox\shimx64.efi como entrada de arranque personalizada.

NOTA: Se sabe que algunos firmwares UEFI eliminan la opción de arranque proxmox después de reiniciar. Puede ocurrir si la entrada apunta a una instalación GRUB en un disco que no figura como opción de arranque. Si es posible, añada el disco como opción en el firmware UEFI y vuelva a ejecutar proxmox-boot-tool.

CONSEJO: Para registrar claves personalizadas, consulte la página de la wiki sobre Secure Boot.

Utilizar DKMS/módulos de terceros con Secure Boot

En sistemas con Secure Boot habilitado, el kernel se negará a cargar módulos que no estén firmados con una clave de confianza. El conjunto predeterminado de módulos incluido en los paquetes del kernel está firmado con una clave efímera embebida en la imagen del kernel, considerada de confianza por esa versión concreta.

Para cargar otros módulos, como los compilados mediante DKMS o manualmente, deben firmarse con una clave de confianza para la pila Secure Boot. La forma más sencilla es registrarla como Machine Owner Key (MOK) mediante mokutil.

dkms genera automáticamente un par de claves y un certificado en /var/lib/dkms/mok.key y /var/lib/dkms/mok.pub, y los utiliza para firmar los módulos que compila e instala.

Puede ver el contenido del certificado con:

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

y registrarlo con:

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

mokutil solicitará dos veces una contraseña temporal. Esa contraseña deberá introducirse una vez más en el siguiente paso. Al reiniciar, el sistema debería arrancar automáticamente MOKManager, que permite verificar la clave/certificado y confirmar su registro utilizando la contraseña elegida. Después, el kernel debería permitir cargar módulos construidos con DKMS y firmados con la MOK registrada. La MOK también puede utilizarse para firmar binarios EFI e imágenes de kernel personalizadas.

El mismo procedimiento puede utilizarse con módulos personalizados o de terceros que no estén gestionados por DKMS, aunque en ese caso la generación de clave/certificado y la firma deben realizarse manualmente.

3.14 Kernel Samepage Merging (KSM)

Kernel Samepage Merging (KSM) es una función opcional de deduplicación de memoria ofrecida por el kernel Linux y habilitada de forma predeterminada en Proxmox VE. KSM examina rangos de páginas de memoria física buscando contenidos idénticos e identifica las páginas virtuales que están mapeadas sobre ellas. Cuando encuentra páginas idénticas, vuelve a mapear las páginas virtuales para que todas apunten a una misma página física y libera las anteriores. Las páginas virtuales se marcan como copy-on-write, de modo que cualquier escritura se realiza en una nueva zona de memoria sin alterar la página física compartida.

3.14.1 Implicaciones de KSM

KSM puede optimizar el uso de memoria en entornos de virtualización, ya que varias VMs que ejecuten sistemas operativos o cargas de trabajo similares pueden compartir un número significativo de páginas de memoria comunes.

Sin embargo, aunque KSM puede reducir el uso de memoria, también conlleva riesgos de seguridad: puede exponer las VMs a ataques de canal lateral. Investigaciones han mostrado que es posible inferir información sobre una VM en ejecución desde una segunda VM situada en el mismo host explotando determinadas características de KSM.

Por ello, si utiliza Proxmox VE para ofrecer servicios de hosting, debería considerar deshabilitar KSM para proporcionar seguridad adicional a los usuarios. Además, debería comprobar la normativa de su país, ya que deshabilitar KSM puede ser un requisito legal.

3.14.2 Deshabilitar KSM

KSM puede deshabilitarse a nivel de nodo o individualmente para una VM.

Deshabilitar KSM en un nodo

Para comprobar si KSM está activo en un nodo:

```
# systemctl status ksmtuned
```

Si está activo, puede deshabilitarlo inmediatamente con:

```
# systemctl disable --now ksmtuned
```

Por último, para deshacer la fusión de todas las páginas actualmente combinadas:

```
# echo 2 > /sys/kernel/mm/ksm/run
```

Deshabilitar KSM para una VM concreta

La opción de configuración de VM allow-ksm controla si está permitida la fusión de páginas de memoria para una VM. Su valor predeterminado es true y puede deshabilitarse con:

```
# qm set <vmid> --allow-ksm 0
```

FIN DE LA PARTE 3B: Con esta entrega queda completado el Capítulo 3 del manual traducido. La siguiente parte comenzará en el Capítulo 4: Interfaz gráfica de usuario (Graphical User Interface).

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 - Traducción al castellano

PARTE 4 · Capítulo 4: Interfaz gráfica de usuario

Secciones 4.1 a 4.6 · Login, GUI, Datacenter, nodos, invitados, almacenamiento, pools, etiquetas y banner de consentimiento

Basado en “Proxmox VE Administration Guide”, Release 9.2.4
4 de agosto de 2026 · Proxmox Server Solutions GmbH

Aviso sobre esta parte

Esta Parte 4 continúa la traducción progresiva del manual oficial de Proxmox VE 9.2.4 y corresponde al Capítulo 4, “Graphical User Interface”, páginas 98 a 110 de la numeración interna del documento original.

Se conserva la organización y el nivel de detalle del texto fuente. Los nombres de comandos, rutas, privilegios y parámetros se mantienen literalmente cuando forman parte de la interfaz, la CLI o la configuración. Los esquemas de interfaz incluidos en esta edición son recreaciones didácticas en castellano basadas en la estructura descrita por el manual; no pretenden reproducir píxel a píxel la interfaz original.

La edición completa se publicará como versión modificada bajo la GNU Free Documentation License (GFDL), conservando los avisos de copyright y la licencia original en inglés conforme a sus condiciones.

Contenido de la Parte 4

- 4.1 Características
- 4.2 Inicio de sesión
- 4.3 Vista general de la GUI
- 4.4 Paneles de contenido: Datacenter, nodos, invitados, almacenamiento y pools
- 4.5 Etiquetas (tags), estilo y permisos
- 4.6 Banner de consentimiento

Capítulo 4 · Interfaz gráfica de usuario

Proxmox VE está diseñado para administrarse sin necesidad de instalar una herramienta de gestión separada: las tareas de administración pueden realizarse desde un navegador web moderno. El manual recomienda utilizar versiones actuales de Firefox o Google Chrome. Para acceder a la consola de los invitados se incluye una consola HTML5 y, como alternativa, puede utilizarse SPICE.

Gracias al sistema de archivos de clúster de Proxmox (pmxcfs), es posible conectarse a cualquier nodo y gestionar desde él todo el clúster. Cada nodo puede administrar el conjunto completo, por lo que no es necesario disponer de un nodo gestor dedicado.

Cuando Proxmox VE detecta una conexión desde un dispositivo móvil, redirige a una interfaz simplificada y adaptada al uso táctil. La interfaz web se alcanza mediante HTTPS en el puerto 8006:

```
https://youripaddress:8006
```

El usuario de inicio de sesión predeterminado es root y la contraseña es la especificada durante el proceso de instalación.

4.1 Características

- Integración y administración transparentes de clústeres Proxmox VE.
- Tecnologías AJAX para la actualización dinámica de recursos.
- Acceso seguro a máquinas virtuales y contenedores mediante cifrado SSL (HTTPS).
- Interfaz rápida basada en búsqueda, preparada para gestionar cientos y, probablemente, miles de máquinas virtuales.
- Consola HTML5 segura o SPICE.
- Gestión de permisos basada en roles para todos los objetos: VMs, almacenamientos, nodos, etc.
- Compatibilidad con múltiples fuentes de autenticación, por ejemplo autenticación local, Microsoft Active Directory (MS ADS) y LDAP.
- Autenticación de dos factores (OATH, YubiKey).
- Interfaz basada en el framework JavaScript ExtJS 7.x.

4.2 Inicio de sesión

Al conectarse al servidor aparece primero la ventana de inicio de sesión. Proxmox VE admite distintos backends de autenticación, denominados “Realm”, y permite seleccionar el idioma en esta misma pantalla. La interfaz gráfica está traducida a más de 20 idiomas.

NOTA: Puede guardar el nombre de usuario en el lado cliente marcando la casilla situada en la parte inferior. De este modo será necesario escribir menos datos en el siguiente inicio de sesión.

Inicio de sesión en la interfaz web



Recreación didáctica en castellano de la ventana de inicio de sesión.

4.3 Vista general de la GUI

La interfaz de usuario de Proxmox VE se organiza en cuatro regiones principales:

Región	Ubicación y función
Header (cabecera)	Parte superior. Muestra información de estado y contiene botones para las acciones más importantes.
Resource Tree (árbol de recursos)	Lateral izquierdo. Árbol de navegación para seleccionar objetos concretos.
Content Panel (panel de contenido)	Zona central. Muestra el estado y las opciones de configuración del objeto seleccionado.
Log Panel (panel de registro)	Parte inferior. Muestra entradas de las tareas recientes. Un doble clic permite ver detalles o abortar una tarea en ejecución.

NOTA: El árbol de recursos y el panel de registro pueden ampliarse o reducirse. El panel de registro también puede ocultarse por completo, algo útil en pantallas pequeñas para disponer de más espacio de trabajo.



Organización general de la interfaz web: cabecera, árbol de recursos, contenido y registro de tareas.

4.3.1 Cabecera

En la parte superior izquierda aparece el logotipo de Proxmox y, junto a él, la versión de Proxmox VE que se está ejecutando. La barra de búsqueda contigua permite localizar objetos concretos -máquinas virtuales, contenedores, nodos, etc.- y puede resultar más rápida que recorrer manualmente el árbol de recursos.

La parte derecha de la cabecera contiene cuatro botones principales:

Botón	Función
Documentation	Abre en una nueva ventana del navegador la documentación de referencia.
Create VM	Abre el asistente de creación de máquinas virtuales.
Create CT	Abre el asistente de creación de contenedores.
User Menu	Muestra la identidad del usuario conectado y abre un menú con opciones específicas de ese usuario.

Dentro del menú de usuario se encuentra el cuadro de diálogo My Settings, con ajustes locales de la interfaz. También hay accesos directos al autoservicio de TFA (autenticación de dos factores) y Password, además de las opciones para cambiar Language y Color Theme. En la parte inferior se encuentra Logout para cerrar la sesión.

4.3.2 My Settings

La ventana My Settings permite configurar opciones que se almacenan localmente en el navegador. Entre ellas se encuentran los Dashboard Storages, que permiten activar o desactivar almacenamientos concretos para que cuenten en la capacidad total que aparece en el resumen de Datacenter. Si no se marca ninguno, el total corresponde a la suma de todos los almacenamientos, equivalente a tenerlos todos activados.

Debajo de estos ajustes aparece el nombre de usuario almacenado, con un botón para borrarlo, y otro botón que permite restablecer todos los diseños de la GUI a sus valores predeterminados.

En el lado derecho se encuentran los ajustes de xterm.js:

Opción	Descripción
Font-Family	Fuente que se utilizará en xterm.js, por ejemplo Arial.
Font-Size	Tamaño de fuente preferido.
Letter Spacing	Aumenta o reduce el espacio entre caracteres.
Line Height	Especifica la altura absoluta de una línea.

“My Settings”: ajustes locales del navegador



Recreación del cuadro My Settings y de los ajustes locales de xterm.js.

4.3.3 Árbol de recursos

El árbol de recursos es el principal sistema de navegación. En su parte superior puede seleccionarse una de varias vistas predefinidas; al cambiar de vista también cambia la organización del árbol inferior. La vista predeterminada es Server View.

En Server View aparecen los siguientes tipos de objeto:

Objeto	Descripción
Datacenter	Contiene ajustes de ámbito de clúster, relevantes para todos los nodos.
Node	Representa los hosts que forman el clúster y en los que se ejecutan los invitados.
Guest	Máquinas virtuales, contenedores y plantillas.
Storage	Almacenamiento de datos.
Pool	Permite agrupar invitados para simplificar su administración.

Las vistas disponibles son:

- Server View: muestra todos los tipos de objeto agrupados por nodo.
- Folder View: muestra todos los tipos de objeto agrupados por tipo.
- Pool View: muestra máquinas virtuales y contenedores agrupados por pool.
- Tag View: muestra máquinas virtuales y contenedores agrupados por etiquetas.

4.3.4 Panel de registro

La finalidad principal del panel de registro es mostrar lo que está ocurriendo en ese momento en el clúster. Acciones como la creación de una nueva máquina virtual se ejecutan en segundo plano; el manual denomina "task" a este tipo de trabajo en background.

La salida de cada tarea se guarda en un archivo de registro independiente. Puede abrirse mediante doble clic sobre la entrada correspondiente y, si la tarea sigue ejecutándose, también es posible abortarla desde esa vista.

El panel muestra las tareas más recientes de todos los nodos del clúster, por lo que permite ver en tiempo real si otra persona está trabajando en otro nodo.

NOTA: Las tareas antiguas y ya finalizadas se eliminan del panel para mantener corta la lista. Siguen estando disponibles en el panel del nodo, dentro de Task History.

Algunas acciones de corta duración envían sus mensajes de registro a todos los miembros del clúster. Esos mensajes pueden consultarse en el panel Cluster log.

4.4 Paneles de contenido

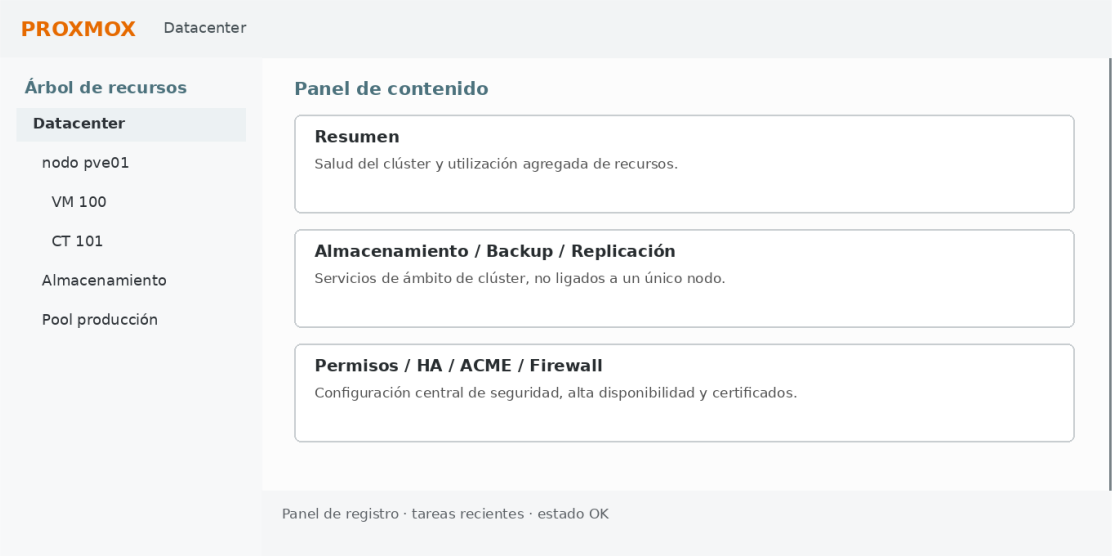
Al seleccionar un elemento del árbol de recursos, el objeto correspondiente muestra su información de configuración y estado en el panel de contenido. Las siguientes secciones ofrecen una visión general; los capítulos específicos del manual desarrollan cada función con mayor detalle.

4.4.1 Datacenter

En el nivel Datacenter se accede a la información y a los ajustes que afectan a todo el clúster:

- Search: búsqueda global en el clúster de nodos, VMs, contenedores, dispositivos de almacenamiento y pools.
- Summary: resumen de la salud del clúster y del uso de recursos.
- Cluster: funciones e información necesarias para crear un clúster o unirse a uno existente.
- Options: consulta y administración de los valores predeterminados de ámbito de clúster.
- Storage: interfaz para administrar el almacenamiento del clúster.
- Backup: programación de trabajos de copia de seguridad. Opera a nivel de clúster, con independencia del nodo en el que estén las VMs o contenedores.
- Replication: consulta y administración de trabajos de replicación.
- Permissions: administración de permisos de usuarios, grupos y tokens API, además de LDAP, MS-AD y autenticación de dos factores.
- HA: administración de la alta disponibilidad de Proxmox VE.
- ACME: configuración de certificados ACME (Let's Encrypt) para los nodos servidores.
- Firewall: configuración global del firewall de Proxmox y creación de plantillas.
- Metric Server: definición de servidores de métricas externos para Proxmox VE.
- Notifications: configuración del comportamiento y de los destinos de las notificaciones de Proxmox VE.
- Support: información sobre la suscripción de soporte.

Panel Datacenter



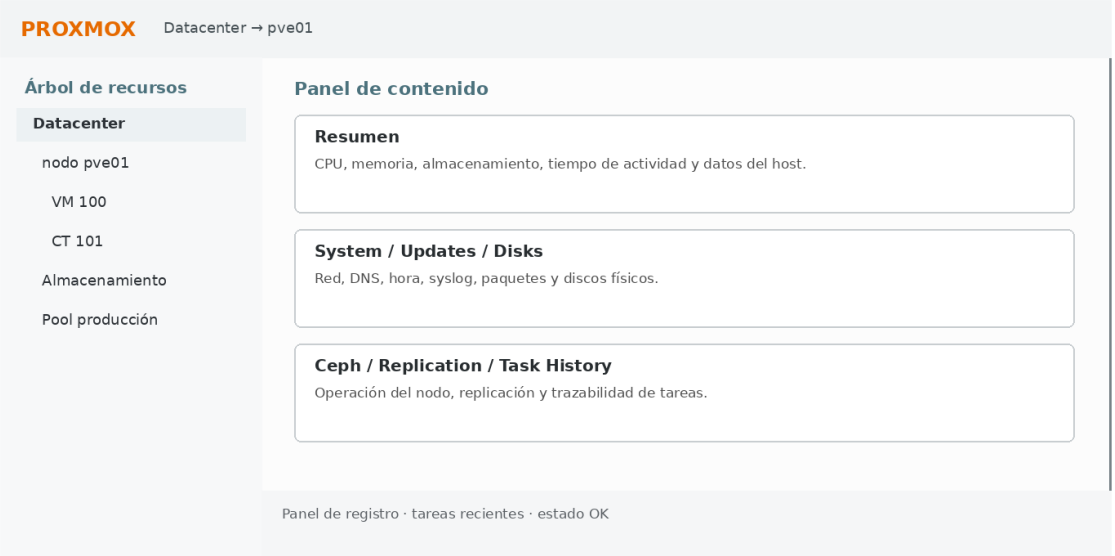
Datacenter concentra las funciones que afectan al conjunto del clúster.

4.4.2 Nodos

Los nodos del clúster pueden administrarse individualmente en este nivel. La cabecera superior dispone de botones como Reboot, Shutdown, Shell, Bulk Actions y Help. Shell ofrece las opciones noVNC, SPICE y xterm.js; Bulk Actions incluye Bulk Start, Bulk Shutdown y Bulk Migrate.

- Search: búsqueda dentro del nodo de VMs, contenedores, dispositivos de almacenamiento y pools.
- Summary: breve visión del uso de recursos del nodo.
- Notes: comentarios personalizados con sintaxis Markdown.
- Shell: acceso a una shell del nodo.
- System: configuración de red, DNS y hora, además de acceso al syslog.
- Updates: actualización del sistema y consulta de nuevos paquetes disponibles.
- Firewall: administración del firewall de Proxmox para un nodo concreto.
- Disks: visión general de los discos conectados y administración de su utilización.
- Ceph: disponible cuando se ha instalado un servidor Ceph en el host; permite administrar el clúster Ceph y consultar su estado.
- Replication: consulta y administración de los trabajos de replicación.
- Task History: lista de tareas anteriores.
- Subscription: carga de una clave de suscripción y generación de informes del sistema para casos de soporte.

Panel de nodo



Panel de nodo: configuración y operación del host seleccionado.

4.4.3 Invitados

Existen dos tipos de invitados: las máquinas virtuales basadas en KVM y los contenedores Linux LXC. Ambos pueden convertirse en plantillas. La navegación es muy parecida en los dos casos, aunque algunas opciones son específicas de cada tecnología.

Para acceder a las distintas interfaces de administración, seleccione una VM o un contenedor en el menú de la izquierda. La cabecera incluye acciones relacionadas con gestión de energía, migración, acceso y tipo de consola, clonación, HA y ayuda. Algunos botones incluyen menús desplegables: por ejemplo, Shutdown agrupa otras opciones de alimentación y Console permite elegir entre SPICE, noVNC y xterm.js.

El panel derecho muestra la interfaz asociada al elemento seleccionado en el menú izquierdo. Las interfaces disponibles son:

- Summary: visión general de la actividad de la VM y campo Notes para comentarios en Markdown.
- Console: acceso a una consola interactiva de la VM o del contenedor.
- (KVM) Hardware: definición del hardware disponible para la máquina virtual KVM.
- (LXC) Resources: definición de los recursos del sistema disponibles para el contenedor LXC.
- (LXC) Network: configuración de red del contenedor.
- (LXC) DNS: configuración DNS del contenedor.
- Options: administración de opciones del invitado.
- Task History: consulta de todas las tareas anteriores relacionadas con el invitado seleccionado.
- (KVM) Monitor: interfaz interactiva de comunicación con el proceso KVM.
- Backup: creación y restauración de copias de seguridad del sistema.
- Replication: consulta y administración de los trabajos de replicación del invitado.
- Snapshots: creación y restauración de snapshots de la VM.
- Firewall: configuración del firewall a nivel de VM.
- Permissions: administración de permisos del invitado seleccionado.



Panel de invitado: las opciones cambian ligeramente entre máquinas virtuales KVM y contenedores LXC.

4.4.4 Almacenamiento

Al igual que la interfaz de invitados, la vista de almacenamiento utiliza un menú a la izquierda y un panel a la derecha. Se trata de una vista dividida en dos zonas: las opciones del almacenamiento a la izquierda y, a la derecha, el contenido de la opción seleccionada.

- Summary: información importante del almacenamiento, como tipo, uso y clases de contenido que almacena.
- Content: una entrada de menú por cada tipo de contenido admitido por el almacenamiento; por ejemplo Backups, ISO Images y CT Templates.
- Permissions: administración de permisos sobre el almacenamiento.



Panel de almacenamiento: resumen, contenido y permisos.

4.4.5 Pools

La vista de pools vuelve a utilizar dos zonas: un menú en el lateral izquierdo y, en el lado derecho, la interfaz correspondiente a cada opción.

- Summary: muestra una descripción del pool.
- Members: muestra y administra los miembros del pool, tanto invitados como almacenamiento.
- Permissions: administra los permisos del pool.



Los pools permiten agrupar recursos para simplificar su organización y control de acceso.

4.5 Etiquetas (Tags)

Con fines organizativos pueden asignarse etiquetas a los invitados. En la versión descrita por el manual, estas etiquetas aportan principalmente valor informativo. Se muestran en dos lugares de la interfaz web: en el árbol de recursos y en la línea de estado cuando se selecciona un invitado.

Las etiquetas pueden añadirse, editarse y eliminarse desde la línea de estado del invitado haciendo clic en el icono del lápiz. Pueden añadirse varias mediante el botón + y eliminarse mediante el botón −. Los cambios se guardan con ✓ y se cancelan con ×.

También pueden configurarse desde la CLI. Varias etiquetas se separan con punto y coma:

```
# qm set ID --tags 'myfirsttag;mysecondtag'
```



Ejemplo didáctico de uso de tags para clasificar invitados.

4.5.1 Configuración de estilo

De forma predeterminada, los colores de las etiquetas se derivan de manera determinista a partir de su texto. Es posible personalizar el color, la forma utilizada en el árbol de recursos, la sensibilidad a mayúsculas y minúsculas y la forma en que se ordenan las etiquetas.

Desde la interfaz web se configura en Datacenter → Options → Tag Style Override. También puede hacerse mediante la CLI. Por ejemplo:

```
# pvesh set /cluster/options --tag-style color-map=example:000000:FFFFFF
```

Este ejemplo establece el color de fondo de la etiqueta example en negro (#000000) y el color del texto en blanco (#FFFFFF).

4.5.2 Permisos

De forma predeterminada, los usuarios que poseen el privilegio VM.Config.Options sobre un invitado (/vms/ID) pueden asignar cualquier etiqueta. Si se desea restringir este comportamiento, pueden establecerse los permisos apropiados en Datacenter → Options → User Tag Access:

Modo	Comportamiento
free	Los usuarios no tienen restricciones al establecer etiquetas. Es el valor predeterminado.
list	Los usuarios solo pueden establecer etiquetas incluidas en una lista predefinida.
existing	Como list, pero además pueden utilizar etiquetas que ya existan.
none	Se impide a los usuarios utilizar etiquetas.

La misma configuración también puede realizarse mediante la CLI.

NOTA: Un usuario con el privilegio Sys.Modify sobre / siempre puede establecer o eliminar cualquier etiqueta, independientemente de esta configuración.

Además, existe una lista configurable de registered tags que solo puede ser modificada por usuarios con el privilegio Sys.Modify sobre /. La lista puede editarse en Datacenter → Options → Registered Tags o mediante la CLI. El manual remite a Datacenter Configuration para consultar las opciones exactas y su invocación desde la línea de comandos.

4.6 Banner de consentimiento

Puede configurarse un banner de consentimiento personalizado que deba aceptarse antes de iniciar sesión. Se encuentra en Datacenter → Options → Consent Text. Si no existe contenido, el banner no se muestra.

El texto se almacena como una cadena base64 dentro del archivo de configuración:

```
/etc/pve/datacenter.cfg
```

Con esta sección queda completado el Capítulo 4 de la guía. La siguiente parte comienza en el Capítulo 5, "Cluster Manager", que aborda requisitos, creación y unión de nodos, eliminación de nodos, quorum, red de clúster, redundancia Corosync, QDevice, configuración de Corosync y migración de invitados.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 - Traducción al castellano

PARTE 5 · Capítulo 5: Gestor de clúster

Secciones 5.1 a 5.15 · creación, unión y retirada de nodos, quorum, Corosync, QDevice y migración

Basado en “Proxmox VE Administration Guide”, Release 9.2.4
4 de agosto de 2026 · Proxmox Server Solutions GmbH

Aviso sobre esta parte

Esta Parte 5 continúa la traducción progresiva del manual oficial de Proxmox VE 9.2.4 y corresponde al Capítulo 5, “Cluster Manager”, páginas 111 a 137 de la numeración interna del documento original.

Se conserva la organización, terminología técnica, comandos y nivel de detalle del texto fuente. Las rutas, parámetros y bloques de configuración permanecen sin traducir cuando forman parte literal de la CLI o de archivos de configuración. Los esquemas incluidos en esta edición son recreaciones didácticas en castellano para facilitar la comprensión.

La edición completa se publicará como versión modificada bajo la GNU Free Documentation License (GFDL), conservando los avisos de copyright y la licencia original en inglés conforme a sus condiciones.

Contenido de la Parte 5

- 5.1 Requisitos y 5.2 Preparación de nodos
- 5.3 Crear un clúster y 5.4 añadir nodos
- 5.5 eliminar un nodo y 5.6 separar un nodo sin reinstalar
- 5.7 Quorum y 5.8 red de clúster
- 5.9 redundancia de Corosync y 5.10 papel de SSH
- 5.11 QDevice y voto externo
- 5.12 configuración de Corosync
- 5.13 arranque en frío, 5.14 selección automática de VMID y 5.15 migración de invitados

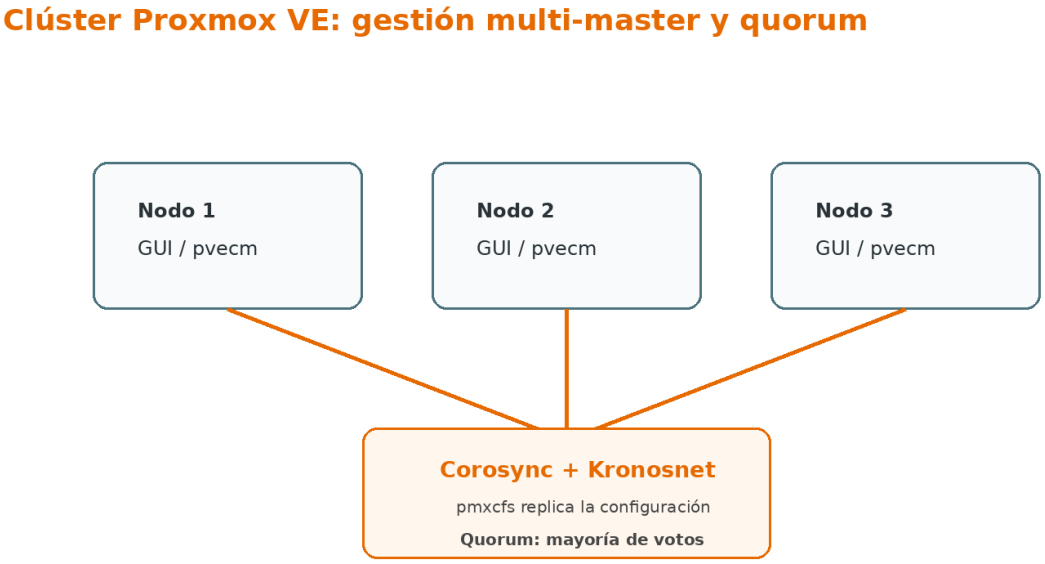
Capítulo 5 · Gestor de clúster

El gestor de clúster de Proxmox VE, pvecm, es una herramienta para crear un grupo de servidores físicos. A este grupo se le denomina clúster. Para la comunicación fiable del grupo se utiliza Corosync Cluster Engine. No existe un límite explícito al número de nodos de un clúster; en la práctica, el máximo alcanzable depende del rendimiento de los hosts y de la red. El manual señala que ya en 2021 existían clústeres de producción con más de 50 nodos sobre hardware empresarial de gama alta.

pvecm permite crear un clúster, unir nodos, abandonar el clúster, consultar su estado y realizar otras tareas relacionadas. Proxmox Cluster File System (pmxcfs) distribuye de forma transparente la configuración del clúster entre todos sus nodos.

Agrupar nodos en un clúster aporta, entre otras, las siguientes ventajas:

- Administración centralizada desde la web.
- Arquitectura multi-master: cualquier nodo puede realizar todas las tareas de administración.
- Uso de pmxcfs, un sistema de archivos basado en base de datos para guardar configuraciones y replicarlas en tiempo real mediante Corosync.
- Migración sencilla de máquinas virtuales y contenedores entre hosts físicos.
- Despliegue rápido.
- Servicios de ámbito de clúster, como firewall y alta disponibilidad (HA).



Esquema conceptual: cada nodo puede administrar todo el clúster; Corosync y pmxcfs mantienen el estado compartido.

5.1 Requisitos

- Todos los nodos deben poder comunicarse entre sí mediante los puertos UDP 5405-5412 para que Corosync funcione.
- La fecha y hora deben estar sincronizadas.
- Se requiere un túnel SSH por el puerto TCP 22 entre nodos.
- Para alta disponibilidad fiable se necesitan al menos tres nodos. Todos deberían ejecutar la misma versión.

NOTA: En clústeres pequeños de dos nodos puede utilizarse un QDevice para aportar un tercer voto.

Se recomienda una NIC física dedicada al tráfico de clúster. Corosync necesita una latencia baja y consistente, pero no requiere mucho ancho de banda; una NIC dedicada de 1 Gbit suele ser suficiente. Separar este tráfico evita que otros servicios consuman todo el ancho de banda y aumenten la latencia de los paquetes de Corosync.

Pueden configurarse enlaces adicionales de tráfico de clúster para aportar redundancia si falla la red dedicada. Corosync admite hasta ocho enlaces.

NOTA: Para que la redundancia sea fiable, al menos otro enlace debería discurrir por una red física diferente. Un único enlace respaldado por un bond puede ser problemático en determinados fallos; véase “Corosync sobre bonds”.

- Para añadir nodos se necesita la contraseña de root de un nodo ya perteneciente al clúster.
- La migración en línea de máquinas virtuales solo está soportada cuando los nodos tienen CPU del mismo fabricante. Puede funcionar en otros casos, pero no está garantizado.

5.2 Preparación de los nodos

Instale primero Proxmox VE en todos los nodos. Cada nodo debe quedar instalado con su nombre de host y su configuración IP definitivos. Después de crear el clúster no es posible cambiar el hostname ni la IP mediante el procedimiento normal de clúster.

Es habitual registrar los nombres de todos los nodos y sus IP en /etc/hosts, o resolverlos por otros medios, aunque no es estrictamente necesario para que el clúster funcione. Puede resultar útil para conectarse por SSH utilizando nombres fáciles de recordar. Aun así, Proxmox recomienda usar direcciones IP al referenciar nodos en la configuración del clúster.

5.3 Crear un clúster

Puede crear un clúster desde la consola -por ejemplo, iniciando sesión mediante SSH- o a través de la API utilizando la interfaz web de Proxmox VE en Datacenter → Cluster.

NOTA: Utilice un nombre único para el clúster. No puede cambiarse posteriormente y sigue las mismas reglas que los nombres de nodo.

5.3.1 Crear mediante la GUI web

En Datacenter → Cluster, pulse Create Cluster. Introduzca el nombre del clúster y seleccione en la lista desplegable la conexión de red que actuará como red principal del clúster (Link 0). Por defecto se utiliza la IP que resuelve el hostname del nodo.

Desde Proxmox VE 6.2 pueden añadirse hasta ocho enlaces de respaldo. Para agregar un enlace redundante, pulse Add y seleccione el número de enlace y la dirección IP correspondiente. En versiones anteriores a 6.2, el segundo enlace se configuraba mediante la casilla Advanced y Link 1.

NOTA: La red elegida para la comunicación del clúster no debería utilizarse para tráfico intenso, como almacenamiento de red o migración en vivo. Corosync genera poco tráfico, pero es muy sensible a la latencia.

5.3.2 Crear mediante la línea de comandos

Inicie sesión por SSH en el primer nodo de Proxmox VE y ejecute:

```
hp1# pvecm create CLUSTERNAME
```

Para comprobar el estado del nuevo clúster:

```
hp1# pvecm status
```

5.3.3 Varios clústeres en la misma red

Es posible crear varios clústeres en la misma red física o lógica. Cada clúster debe tener un nombre único para evitar colisiones en la pila de comunicación y reducir la confusión humana.

Aunque el consumo de ancho de banda de Corosync es relativamente bajo, la latencia de los paquetes y la tasa de paquetes por segundo (PPS) son factores limitantes. Varios clústeres en una misma red pueden competir por estos recursos; para clústeres grandes puede seguir siendo recomendable usar infraestructuras de red físicas separadas.

5.4 Añadir nodos al clúster

PRECAUCIÓN: Al unir un nodo a un clúster se sobrescribe toda la configuración existente en /etc/pve. El nodo que se incorpora no puede contener invitados, pues sus IDs podrían entrar en conflicto, y heredará la configuración de almacenamiento del clúster. Si el nodo contiene invitados, puede realizar copias con vzdump y restaurarlas con otros IDs después de la unión. Si el diseño de almacenamiento es distinto, deberá volver a añadir los almacenamientos del nodo y ajustar las restricciones por nodo.

5.4.1 Unir un nodo mediante la GUI

Inicie sesión en la interfaz web de un nodo que ya pertenezca al clúster. En Datacenter → Cluster, pulse Join Information y después Copy Information. También puede copiar manualmente la cadena del campo Information.

A continuación, inicie sesión en la interfaz web del nodo que desea añadir. Vaya a Datacenter → Cluster y pulse Join Cluster. Pegue en Information los datos obtenidos anteriormente. La mayoría de los parámetros se completarán automáticamente. Por razones de seguridad, la contraseña del clúster debe introducirse manualmente.

NOTA: Para introducir todos los datos manualmente, desactive la casilla Assisted Join.

Tras pulsar Join, el proceso de unión comienza de inmediato. Una vez incorporado el nodo, su certificado actual será sustituido por otro firmado por la autoridad certificadora (CA) del clúster. La sesión actual dejará de funcionar al cabo de unos segundos; puede ser necesario forzar la recarga de la interfaz web y volver a iniciar sesión con las credenciales del clúster.

El nuevo nodo debería aparecer entonces en Datacenter → Cluster.

5.4.2 Unir un nodo mediante la línea de comandos

Inicie sesión por SSH en el nodo que desea incorporar y ejecute:

```
# pvecm add IP-ADDRESS-CLUSTER
```

Utilice como IP-ADDRESS-CLUSTER la IP o el hostname de un nodo ya perteneciente al clúster. Se recomienda una dirección IP.

Por defecto, pvecm add se autentica contra la API del nodo existente, solicita su contraseña root@pam y verifica su huella SHA-256. Este flujo no requiere autenticación SSH por contraseña, por lo que puede mantenerse PasswordAuthentication no en sshd_config.

La opción --use_ssh permite volver al método histórico basado en SSH, que utiliza ssh-copy-id para fusionar la clave root del clúster existente con el nuevo nodo. Ese flujo sí requiere acceso SSH root mediante contraseña o clave.

Para comprobar el estado:

```
# pvecm status
```

Ejemplo de estado tras añadir cuatro nodos:

```
# pvecm status
Cluster information
~~~~~~~~~~~~~~~~~~~
Name:             prod-central
Config Version:   3
Transport:        knet
Secure auth:      on

Quorum information
~~~~~~~~~~~~~~~~~~
Date:             Tue Sep 14 11:06:47 2021
Quorum provider:  corosync_votequorum
Nodes:            4
Node ID:          0x00000001
Ring ID:          1.1a8
Quorate:          Yes

Votequorum information
~~~~~~~~~~~~~~~~~~~~~~
Expected votes:   4
Highest expected: 4
Total votes:      4
Quorum:           3
Flags:            Quorate

Membership information
~~~~~~~~~~~~~~~~~~~~~~
Nodeid      Votes Name
0x00000001      1 192.168.15.91
0x00000002      1 192.168.15.92 (local)
0x00000003      1 192.168.15.93
0x00000004      1 192.168.15.94
```

Si solo desea listar los nodos:

```
# pvecm nodes
Membership information
~~~~~~~~~~~~~~~~~~~~~~
Nodeid  Votes Name
1       1 hp1
2       1 hp2 (local)
3       1 hp3
4       1 hp4
```

5.4.3 Añadir nodos con una red de clúster separada

Al añadir un nodo a un clúster que utiliza una red de clúster separada, use el parámetro link0 para especificar la dirección local del nodo en esa red:

```
# pvecm add IP-ADDRESS-CLUSTER --link0 LOCAL-IP-ADDRESS-LINK0
```

Si desea usar la redundancia integrada de la capa de transporte Kronosnet, utilice también link1. En la GUI puede elegir la interfaz adecuada mediante los campos Link X del diálogo Cluster Join.

5.5 Eliminar un nodo del clúster

PRECAUCIÓN: Lea el procedimiento completo antes de continuar. Los pasos siguientes explican también el caso de un nodo que forma parte de un clúster Ceph. Si Ceph no está instalado, ignore las operaciones referidas a Ceph.

NOTA: Si la configuración incluye un QDevice, deberá retirarlo antes de eliminar el nodo del clúster.

5.5.1 Requisitos previos

- Mueva todas las máquinas virtuales y contenedores fuera del nodo.
- Realice una copia de seguridad de todos los datos locales del nodo que va a eliminar.
- Asegúrese de que el nodo ya no participa en ningún trabajo de replicación.

PRECAUCIÓN: Si no elimina los trabajos de replicación antes de retirar el nodo, estos pueden quedar imposibles de borrar. La replicación cambia automáticamente de dirección cuando se migra una VM replicada, por lo que migrar una VM desde el nodo que se va a eliminar puede volver a crear trabajos hacia ese nodo.

Si utiliza Ceph, compruebe que el clúster restante tiene espacio suficiente y que los OSD están "up" e "in". Destruir un OSD, especialmente el último de un nodo, provoca un reequilibrado de datos.

NOTA: Deben seguir existiendo suficientes nodos con OSD para satisfacer los parámetros size y min_size de los pools. Con menos nodos que size (por defecto 3) se degrada la redundancia; con menos que min_size (por defecto 2), la E/S de los pools queda bloqueada. Los invitados afectados pueden fallar si su E/S se bloquea.

Asegúrese también de conservar suficientes monitores, managers y, si se usa CephFS, servidores de metadatos.

5.5.2 Eliminar el nodo

Antes de retirar un nodo de Proxmox VE debe asegurarse de que ya no forma parte de Ceph y que no reside en él ningún recurso o servicio Ceph. El ejemplo original elimina node4 tanto del clúster Ceph como del clúster Corosync de Proxmox VE.

```
node4# pvecm nodes
Membership information
~~~~~~~~~~~~~~~~~~~~~~
Nodeid Votes Name
1 1 node1
2 1 node2
3 1 node3
4 1 node4 (local)
```

1. Retire primero el nodo del clúster Ceph:
   - Destruya uno por uno los OSD restantes del nodo que se va a eliminar.
   - Espere hasta que el estado de Ceph vuelva a HEALTH_OK.
   - Si existen, destruya los servidores de metadatos desde Ceph → CephFS o con el comando indicado a continuación.
   - Destruya el monitor Ceph restante.
   - Destruya el manager Ceph restante.
   - Finalmente elimine de la jerarquía CRUSH el bucket vacío correspondiente al nodo.

```
node4# pveceph mds destroy node4
node4# ceph osd crush remove node4
```

2. Apague node4 y asegúrese de que no vuelva a arrancar en esta red con su configuración actual.

IMPORTANTE: Es crítico apagar el nodo antes de eliminarlo y garantizar que no vuelva a encenderse en la red del clúster con su configuración actual. Si arranca de nuevo tal cual, el clúster podría quedar dañado y resultar difícil recuperarlo.

3. Inicie sesión en uno de los nodos restantes y elimine node4:

```
node1# pvecm delnode node4
```

NOTA: Puede aparecer el error "could not kill node (error = cs_err_not_exist)". No indica que la eliminación haya fallado; Corosync no puede "matar" un nodo que ya está offline, por lo que el mensaje puede ignorarse.

4. Verifique que el nodo desapareció del clúster:

```
node1# pvecm nodes
Membership information
~~~~~~~~~~~~~~~~~~~~~~
Nodeid Votes Name
1 1 node1 (local)
2 1 node2
3 1 node3
```

5.5.3 Limpieza después de eliminar el nodo

- Los archivos de configuración del nodo eliminado seguirán en /etc/pve/nodes/node4. Recupere cualquier configuración necesaria y elimine después el directorio.
- La huella SSH permanecerá en known_hosts de los demás nodos. Si al volver a añadir un nodo con la misma IP o hostname aparece un error SSH, ejecute pvecm updatecerts una vez en el nodo reincorporado para actualizar la huella en todo el clúster.
- Las reglas de alta disponibilidad existentes pueden seguir haciendo referencia al nodo eliminado; retírelo de esas reglas.
- La clave pública del nodo eliminado seguirá en /etc/pve/priv/authorized_keys y debería borrarse si ya no es necesaria.

Para volver a unir el mismo servidor al mismo clúster, Proxmox recomienda reinstalar Proxmox VE en el servidor y después volver a incorporarlo al clúster.

5.6 Separar un nodo sin reinstalar

PRECAUCIÓN: Este no es el método recomendado. Continúe solo si comprende bien sus implicaciones; en caso de duda utilice el procedimiento de eliminación y reinstalación anterior.

También es posible separar un nodo sin reinstalarlo desde cero. Sin embargo, después de retirarlo seguirá teniendo acceso a cualquier almacenamiento compartido. Esto debe resolverse antes de comenzar: dos clústeres distintos no pueden compartir exactamente el mismo almacenamiento, ya que los bloqueos de almacenamiento no funcionan a través de la frontera entre clústeres y pueden producirse conflictos de VMID.

Se recomienda crear un almacenamiento nuevo al que solo tenga acceso el nodo que se desea separar -por ejemplo un nuevo export NFS o un pool Ceph independiente-. Mueva allí todos los datos y VMs antes de separar el nodo.

ADVERTENCIA: Asegúrese de separar limpiamente todos los recursos compartidos. De lo contrario aparecerán conflictos y problemas.

Detenga Corosync y pve-cluster en el nodo:

```
systemctl stop pve-cluster
systemctl stop corosync
```

Inicie pmxcfs en modo local:

```
pmxcfs -l
```

Elimine la configuración de Corosync:

```
rm /etc/pve/corosync.conf
rm -r /etc/corosync/*
```

Reinicie el sistema de archivos como servicio normal:

```
killall pmxcfs
systemctl start pve-cluster
```

El nodo ya está separado. Desde cualquiera de los nodos restantes del clúster elimínelo con:

```
pvecm delnode oldnode
```

Si el comando falla por pérdida de quorum en el nodo restante, puede establecer temporalmente los votos esperados a 1 y repetir la eliminación:

```
pvecm expected 1
pvecm delnode oldnode
```

Vuelva al nodo separado y borre los archivos de clúster restantes:

```
rm /var/lib/corosync/*
```

Si lo desea, limpie también las configuraciones de los otros nodos que sigan presentes en /etc/pve/nodes/NODENAME, asegurándose absolutamente de usar el nombre correcto.

PRECAUCIÓN: Las claves SSH del nodo permanecerán en authorized_keys, por lo que los nodos aún podrán autenticarse entre sí por clave pública. Elimine las claves correspondientes de /etc/pve/priv/authorized_keys.

5.7 Quorum

Proxmox VE utiliza una técnica basada en quorum para mantener un estado consistente entre todos los nodos del clúster. En términos de sistemas distribuidos, el quorum es el número mínimo de votos que debe obtener una transacción distribuida para poder realizar una operación.

Si se produce una partición de red, los cambios de estado requieren que una mayoría de nodos continúe en línea. Si el clúster pierde el quorum, pasa a modo de solo lectura.

NOTA: Por defecto, Proxmox VE asigna un único voto a cada nodo.

5.8 Red de clúster

La red de clúster es el núcleo de un clúster. Todos los mensajes enviados por ella deben llegar de forma fiable a todos los nodos y en el orden correspondiente. Proxmox VE utiliza Corosync para esta tarea; Corosync sirve además al sistema de archivos de configuración descentralizado pmxcfs.

5.8.1 Requisitos de red

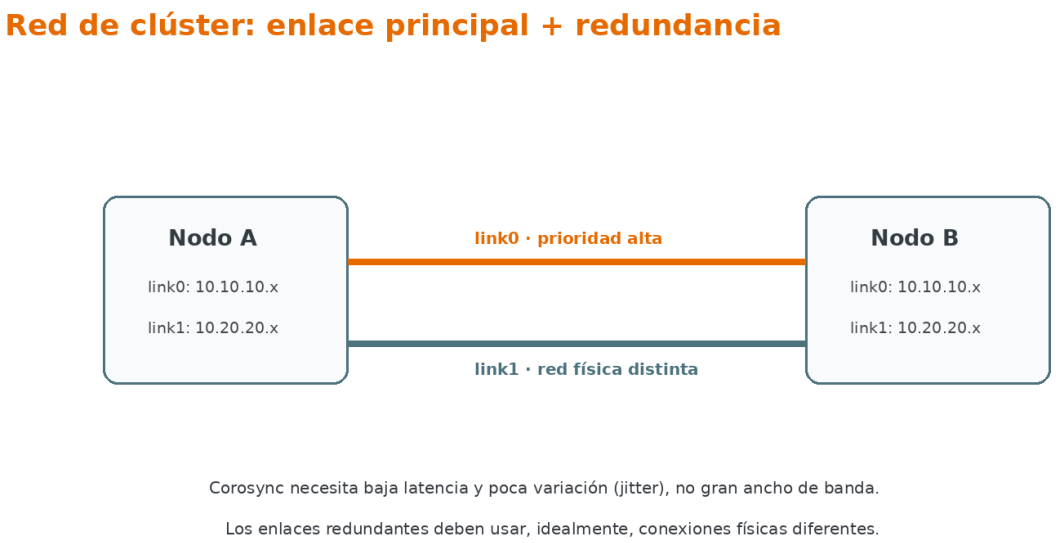
La pila de clúster de Proxmox VE requiere una red fiable con latencias inferiores a 5 ms -rendimiento de LAN- entre todos los nodos para funcionar de forma estable. En instalaciones pequeñas puede funcionar con más latencia, pero no está garantizado y se vuelve poco probable con más de tres nodos y latencias por encima de aproximadamente 10 ms.

La red no debería estar muy cargada por otros servicios. Corosync consume poco ancho de banda, pero es sensible al jitter de latencia; idealmente debería usar una red físicamente separada. En particular, no comparta la misma red para Corosync y almacenamiento, salvo como posible enlace de respaldo de baja prioridad en una configuración redundante.

Antes de crear el clúster es recomendable comprobar la red. Puede verificar la conectividad entre nodos con ping. Si el firewall de Proxmox VE está habilitado, se generan automáticamente reglas ACCEPT para Corosync.

NOTA: Corosync utilizaba multicast antes de la versión 3.0, incorporada en Proxmox VE 6.0. Las versiones modernas se apoyan en Kronosnet y, actualmente, emplean UDP unicast normal.

PRECAUCIÓN: Es posible activar multicast o el unicast heredado usando transport udp o udpu en corosync.conf, pero esto deshabilita el cifrado y el soporte de redundancia. No se recomienda.



Separar Corosync y disponer de un enlace redundante por otra red física mejora la tolerancia a fallos.

5.8.2 Corosync sobre bonds

Recomendaciones

Proxmox recomienda al menos una NIC física dedicada para el enlace principal de Corosync. Los bonds pueden utilizarse como enlaces adicionales, pero con varias precauciones:

- active-backup puede no proporcionar la redundancia esperada en determinados fallos.
- Se desaconsejan balance-rr, balance-xor, balance-tlb y balance-alb para tráfico de Corosync, porque pueden ser problemáticos en ciertos escenarios.
- Si se usa IEEE 802.3ad (LACP), se recomienda encarecidamente configurar bond-lacp-rate fast tanto en el nodo Proxmox VE como en el switch. El valor slow puede provocar tiempos de conmutación demasiado largos.

Motivo técnico

Un bond puede resultar problemático cuando una interfaz deja de transmitir pero conserva el estado de enlace “up” y no hay otros enlaces Corosync disponibles. Algunos modos de balanceo continúan enviando parte de los paquetes por la interfaz averiada, lo que puede crear conectividad asimétrica: distintos nodos solo pueden comunicarse con subconjuntos diferentes del clúster.

En esa situación, Corosync puede ser incapaz de formar un quorum estable. Si HA está habilitado, incluso nodos cuyo bond funciona correctamente pueden llegar a autoaislarse (fencing). En el peor caso, todo el clúster puede realizar fencing.

active-backup evita la conectividad asimétrica, pero el bond afectado puede no conmutar a tiempo y el nodo puede perder el clúster y realizar fencing. Los modos balance-rr, balance-xor, balance-tlb y balance-alb sí pueden producir conectividad asimétrica.

Con LACP, cada extremo puede dejar de usar una interfaz si no recibe tres LACPDU consecutivos. Con el valor predeterminado slow se envía un LACPDU cada 30 segundos, por lo que la conmutación puede tardar 90 segundos, demasiado para HA. Con bond-lacp-rate fast en ambos extremos, los LACPDU se envían cada segundo y el failover puede reducirse a unos 3 segundos.

5.8.3 Red de clúster separada

Si se crea un clúster sin parámetros, la red de Corosync suele compartirse con la interfaz web y la red de las VMs; según la instalación, incluso el almacenamiento puede usarla. Se recomienda separarla porque Corosync es una aplicación de tiempo crítico.

Configurar una red nueva

Configure primero una nueva interfaz de red sobre una red físicamente separada y compruebe que cumple los requisitos de la red de clúster.

Separar la red al crear el clúster

Use los parámetros linkX de pvecm create. Por ejemplo, si una NIC adicional tiene la dirección 10.10.10.1/25 y desea que toda la comunicación de clúster pase por ella:

```
pvecm create test --link0 10.10.10.1
systemctl status corosync
```

Después, añada los demás nodos indicando la red separada como se explicó anteriormente.

Separar la red después de crear el clúster

También puede cambiar la red de comunicación de un clúster ya creado sin reconstruirlo. El cambio puede provocar breves pérdidas de quorum mientras los nodos reinician Corosync y vuelven a incorporarse por la nueva red.

Edite corosync.conf siguiendo el procedimiento descrito en la sección 5.12.1. Un ejemplo de configuración original puede tener esta estructura:

```
logging {
  debug: off
  to_syslog: yes
}
nodelist {
  node { name: due  nodeid: 2  quorum_votes: 1  ring0_addr: due }
  node { name: tre  nodeid: 3  quorum_votes: 1  ring0_addr: tre }
  node { name: uno  nodeid: 1  quorum_votes: 1  ring0_addr: uno }
}
quorum { provider: corosync_votequorum }
totem {
  cluster_name: testcluster
  config_version: 3
  ip_version: ipv4-6
  secauth: on
  version: 2
  interface { linknumber: 0 }
}
```

NOTA: ringX_addr especifica realmente una dirección de enlace Corosync. El término “ring” se conserva por compatibilidad con versiones antiguas.

Asegúrese de que las propiedades name de cada nodo coinciden con su nombre real. Después sustituya los valores ring0_addr por las nuevas IP o por nombres resolubles desde todos los nodos. Se recomienda modificar un enlace cada vez para facilitar la recuperación si algo falla.

En el ejemplo, al cambiar al segmento 10.10.10.0/25, incremente también config_version:

```
nodelist {
  node { name: due  nodeid: 2 quorum_votes: 1 ring0_addr: 10.10.10.2 }
  node { name: tre  nodeid: 3 quorum_votes: 1 ring0_addr: 10.10.10.3 }
  node { name: uno  nodeid: 1 quorum_votes: 1 ring0_addr: 10.10.10.1 }
}
quorum { provider: corosync_votequorum }
totem {
  cluster_name: testcluster
  config_version: 4
  ip_version: ipv4-6
  secauth: on
  version: 2
  interface { linknumber: 0 }
}
```

Los cambios se aplican en vivo, por lo que reiniciar Corosync no es estrictamente necesario. Si se han modificado otros parámetros o aparecen avisos, puede reiniciarlo primero en un nodo y comprobar su estado:

```
systemctl restart corosync
systemctl status corosync
```

Si funciona, reinicie los demás nodos uno a uno; volverán a incorporarse al clúster utilizando la nueva red.

5.8.4 Direcciones de Corosync

Una dirección de enlace Corosync -ringX_addr por compatibilidad- puede especificarse de dos formas:

- Direcciones IPv4/IPv6 directamente. Es la opción recomendada porque son estáticas y no suelen modificarse accidentalmente.
- Hostnames resueltos mediante getaddrinfo. Por defecto se priorizará IPv6 si está disponible, lo que debe tenerse en cuenta al actualizar un clúster existente a IPv6.

PRECAUCIÓN: Los hostnames deben utilizarse con cuidado, ya que su resolución puede cambiar sin modificar Corosync ni el nodo. Si se prefieren nombres, se recomienda un hostname estático dedicado específicamente a Corosync y resoluble de forma coherente por todos los nodos.

Desde Proxmox VE 5.1 los nombres se resuelven al introducirlos y se guarda únicamente la IP resuelta. Los nodos incorporados con versiones anteriores pueden seguir mostrando hostnames sin resolver en corosync.conf; puede ser aconsejable sustituirlos por IPs o nombres dedicados.

5.9 Redundancia de Corosync

Corosync soporta redes redundantes mediante su capa Kronosnet integrada -no disponible con los transportes heredados udp/udpu-. Se habilita especificando varias direcciones de enlace: con parámetros --linkX de pvecm, mediante Link 1 en la GUI, o añadiendo varios ringX_addr en corosync.conf.

NOTA: Para que el failover sea realmente útil, cada enlace debería utilizar una conexión de red física distinta.

Los ejemplos del manual suponen que cada nodo tiene una dirección estática en 10.10.10.0/25 y otra en 10.20.20.0/25. Los enlaces se eligen según una prioridad configurable mediante knet_link_priority o, preferiblemente, con priority al crear el clúster:

```
# pvecm create CLUSTERNAME --link0 10.10.10.1,priority=15 --link1 10.20.20.1,priority=20
```

En este caso se utilizará primero link1 porque tiene la prioridad más alta. Si no se asignan prioridades, o son iguales, se utiliza primero el enlace con el número más bajo.

Incluso si todos los enlaces funcionan, solo el de mayor prioridad transporta tráfico de Corosync. Los enlaces con prioridades distintas no se mezclan. Esto permite utilizar como enlaces de baja prioridad redes destinadas normalmente a VMs o almacenamiento: en una emergencia, una red más congestionada puede ser preferible a no tener conectividad.

5.9.1 Añadir enlaces redundantes a un clúster existente

Edite corosync.conf siguiendo el procedimiento seguro de la sección 5.12.1. Añada un ringX_addr nuevo a cada nodo, usando el mismo X para todos y una dirección única por nodo. Después añada una sección interface nueva en totem.

```
nodelist {
  node { name: due nodeid: 2 quorum_votes: 1
    ring0_addr: 10.10.10.2
    ring1_addr: 10.20.20.2
  }
  node { name: tre nodeid: 3 quorum_votes: 1
    ring0_addr: 10.10.10.3
    ring1_addr: 10.20.20.3
  }
  node { name: uno nodeid: 1 quorum_votes: 1
    ring0_addr: 10.10.10.1
    ring1_addr: 10.20.20.1
  }
}
quorum { provider: corosync_votequorum }
totem {
  cluster_name: testcluster
  config_version: 4
  ip_version: ipv4-6
  secauth: on
  version: 2
  interface { linknumber: 0 }
  interface { linknumber: 1 }
}
```

El nuevo enlace queda activo al aplicar la nueva configuración y normalmente no es necesario reiniciar. Compruebe el registro y pruebe el enlace desconectando temporalmente el antiguo en un nodo:

```
journalctl -b -u corosync
pvecm status
```

Si el estado del clúster sigue siendo saludable, el nuevo enlace está funcionando.

5.10 Papel de SSH en los clústeres Proxmox VE

Proxmox VE utiliza túneles SSH en varias funcionalidades:

- Proxy de sesiones de consola/shell de nodos e invitados. Si se abre la shell del nodo B mientras se está conectado al nodo A, el terminal proxy de A se conecta mediante un túnel SSH no interactivo con la shell de inicio de B.
- Migración segura de memoria y almacenamiento local de VMs y CTs. Durante la migración se establecen uno o varios túneles SSH entre origen y destino para intercambiar información y transferir memoria y discos.
- Replicación de almacenamiento.

5.10.1 Configuración SSH

En Proxmox VE se realizan, entre otros, los siguientes ajustes:

- La configuración del cliente SSH de root se ajusta para preferir AES frente a ChaCha20.
- authorized_keys de root se enlaza con /etc/pve/priv/authorized_keys, fusionando las claves autorizadas del clúster.
- sshd se configura para permitir inicio de sesión root mediante contraseña.

NOTA: Sistemas antiguos pueden conservar /etc/ssh/ssh_known_hosts como enlace simbólico a /etc/pve/priv/known_hosts. Este mecanismo fue sustituido por fijación explícita de claves de host en pve-cluster. Si el enlace sigue existiendo, puede deshacerse con pvecm updatecerts --unmerge-known-hosts.

5.10.2 Riesgos de la ejecución automática de .bashrc

Si existe un .bashrc personalizado u otros archivos ejecutados al iniciar sesión, SSH puede ejecutarlos automáticamente al establecer una sesión. Esto puede causar comportamientos inesperados, porque se ejecutarían con privilegios root también durante operaciones internas como las descritas anteriormente.

Se recomienda comprobar al principio de /root/.bashrc si la sesión es interactiva y salir inmediatamente si no lo es:

```
# Early exit if not running interactively to avoid side-effects!
case $- in
  *i*) ;;
  *) return;;
esac
```

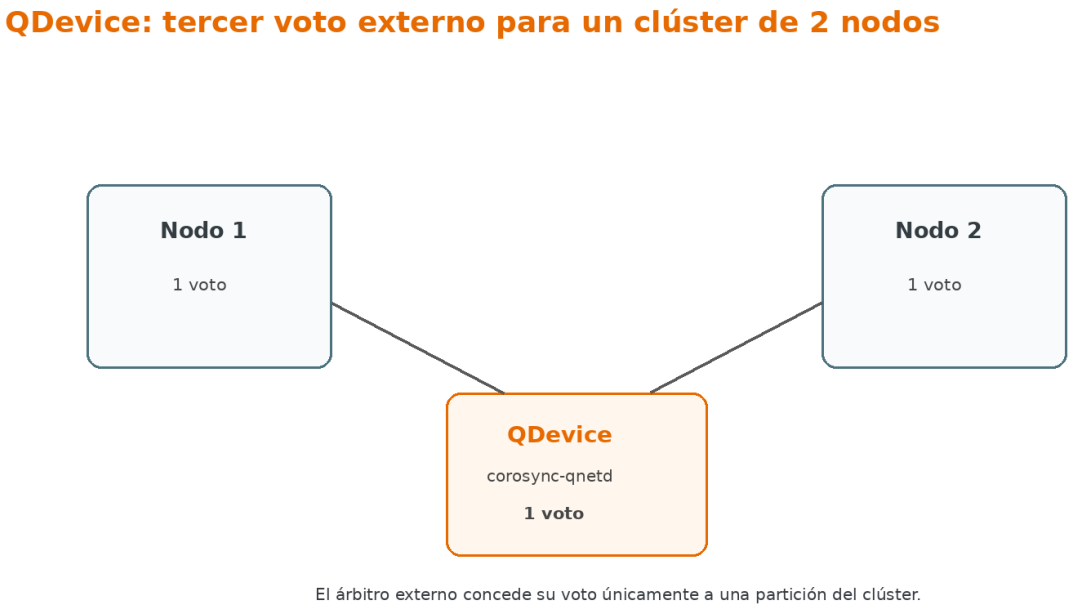
5.11 Soporte de voto externo de Corosync

Esta sección describe el despliegue de un votante externo en un clúster Proxmox VE. Cuando está configurado, el clúster puede soportar más fallos de nodos sin violar las propiedades de seguridad de la comunicación.

Intervienen dos servicios:

- Un demonio QDevice ejecutándose en cada nodo de Proxmox VE.
- Un demonio de voto externo ejecutándose en un servidor independiente.

Esto permite aumentar la disponibilidad, especialmente en configuraciones pequeñas, por ejemplo 2+1 nodos.



En un clúster de dos nodos, QDevice puede aportar el tercer voto necesario para resolver una partición.

5.11.1 Descripción técnica de QDevice

Corosync Quorum Device (QDevice) es un demonio que se ejecuta en cada nodo y proporciona al subsistema de quorum un número configurado de votos según la decisión de un árbitro externo. Su uso principal es permitir que el clúster tolere más fallos de los que permiten las reglas de quorum convencionales.

Actualmente solo se soporta QDevice Net como árbitro externo. El demonio entrega un voto a una partición si puede alcanzar a sus miembros por la red y nunca vota simultáneamente a dos particiones. Está diseñado para servir a varios clústeres y prácticamente no requiere configuración ni estado persistente.

El host externo solo necesita conectividad de red con el clúster y disponer del paquete corosync-qnetd. Proxmox proporciona paquete para sistemas Debian y otras distribuciones Linux suelen ofrecerlo en sus gestores de paquetes.

NOTA: A diferencia de Corosync, QDevice se conecta mediante TCP/IP. Puede ejecutarse fuera de la LAN del clúster y no está sujeto a los mismos requisitos estrictos de baja latencia.

5.11.2 Configuraciones soportadas

Proxmox soporta QDevice para clústeres con un número par de nodos y lo recomienda especialmente para clústeres de dos nodos que necesiten mayor disponibilidad. Para clústeres con número impar se desaconseja actualmente, debido a cómo cambia el número de votos aportados.

En clústeres pares, QDevice añade un único voto. Si QDevice falla, el clúster queda en la misma situación que si no lo tuviera.

En clústeres impares, QDevice aporta N-1 votos, siendo N el número de nodos. Esto evita split-brain, pero tiene inconvenientes:

- Si falla el propio QNet daemon, ya no puede fallar ningún nodo adicional sin perder inmediatamente el quorum. En un clúster de 15 nodos, sin QDevice podrían fallar 7; con QDevice configurado y QDevice caído, no podría fallar ni un nodo.
- Poder perder todos los nodos salvo uno y QDevice puede provocar una recuperación masiva de servicios HA sobre el único nodo restante. Además, Ceph deja de prestar servicio si quedan en línea ((N-1)/2) nodos o menos.

Si comprende estas implicaciones, puede decidir usar QDevice en un clúster impar bajo su propia evaluación.

5.11.3 Configuración de QDevice-Net

Se recomienda ejecutar el demonio que aporta votos como usuario sin privilegios. Los paquetes de Proxmox VE y Debian ya vienen preparados de esta forma. El tráfico entre el demonio y el clúster debe estar cifrado.

Instale corosync-qnetd en el servidor externo:

```
external# apt install corosync-qnetd
```

Instale corosync-qdevice en todos los nodos:

```
pve# apt install corosync-qdevice
```

Asegúrese de que todos los nodos estén online y ejecute desde uno de ellos:

```
pve# pvecm qdevice setup <QDEVICE-IP>
```

La clave SSH del clúster se copiará automáticamente al QDevice.

NOTA: Configure acceso por clave para root en el servidor externo o permita temporalmente el login root por contraseña durante la fase de configuración. Si aparece “Host key verification failed”, pvecm updatecerts puede resolverlo.

Al finalizar correctamente aparecerá “Done”. Verifique el estado:

```
pve# pvecm status
...
Votequorum information
~~~~~~~~~~~~~~~~~~~~~
Expected votes: 3
Highest expected: 3
Total votes: 3
Quorum: 2
Flags: Quorate Qdevice

Membership information
~~~~~~~~~~~~~~~~~~~~~~
Nodeid Votes Qdevice Name
0x00000001 1 A,V,NMW 192.168.22.180 (local)
0x00000002 1 A,V,NMW 192.168.22.181
0x00000000 1 Qdevice
```

Indicadores de estado de QDevice

Indicador	Significado
A / NA	Alive / Not Alive: indica si funciona la comunicación con corosync-qnetd.
V / NV	Indicasi QDevice emitirá un voto para ese nodo. En split-brain, solo una partición recibe el voto.
MW / NMW	Master wins / no masterwins. El valor predeterminado es NMW.
NR	QDevice no registrado.

NOTA: Si QDevice aparece como Not Alive (NA), compruebe que el puerto TCP 5403 -puerto predeterminado de qnetd- sea accesible en el servidor externo.

5.11.4 Preguntas frecuentes

Desempate

Si dos particiones del mismo tamaño no pueden verse entre sí pero ambas ven QDevice, QDevice elige la partición con el node id más bajo y le entrega el voto. Este comportamiento puede modificarse con tie_breaker y requiere reiniciar corosync-qdevice.service en todos los nodos.

Posibles implicaciones negativas

En clústeres con número par de nodos no hay implicaciones negativas adicionales: si QDevice deja de funcionar, la situación equivale a no tenerlo configurado.

Añadir o eliminar nodos después de configurar QDevice

Antes de añadir o eliminar nodos debe retirar QDevice. Después podrá modificar el clúster normalmente y, cuando vuelva a tener un número par de nodos, configurar QDevice de nuevo.

Eliminar QDevice

```
pve# pvecm qdevice remove
```

5.12 Configuración de Corosync

/etc/pve/corosync.conf desempeña un papel central en un clúster Proxmox VE: controla la pertenencia de nodos y la red de clúster. Para la pertenencia de nodos debe utilizarse siempre pvecm. Para otros cambios puede ser necesario editar manualmente la configuración.

5.12.1 Editar corosync.conf

En cada nodo existen dos copias: /etc/pve/corosync.conf y /etc/corosync/corosync.conf. Los cambios en la copia almacenada en el sistema de archivos de clúster se propagan a la copia local, pero no al revés.

La configuración se actualiza automáticamente cuando cambia el archivo. Para evitar aplicar cambios parciales mientras se edita, trabaje primero sobre una copia:

```
cp /etc/pve/corosync.conf /etc/pve/corosync.conf.new
```

Edite la copia con nano, vim.tiny u otro editor.

NOTA: Incremente siempre config_version después de modificar la configuración. Omitirlo puede causar problemas.

Antes de sustituir la configuración activa, cree una copia de seguridad:

```
cp /etc/pve/corosync.conf /etc/pve/corosync.conf.bak
mv /etc/pve/corosync.conf.new /etc/pve/corosync.conf
```

Compruebe si Corosync aplicó los cambios automáticamente:

```
systemctl status corosync
journalctl -b -u corosync
```

Si no se aplican automáticamente, reinicie el servicio:

```
systemctl restart corosync
```

5.12.2 Cambiar el coeficiente de token

El coeficiente de token se configura en la sección totem de /etc/pve/corosync.conf. Si no se especifica, el valor predeterminado es 650 ms. Desde Proxmox VE 9.2, los clústeres nuevos se crean con token_coefficient: 125 explícito para reducir el tiempo necesario para reconstruir la pertenencia del clúster después del fallo de un nodo.

Corosync utiliza este coeficiente para calcular distintos tiempos de espera en función del tamaño del clúster, especialmente token timeout y consensus timeout. El primero determina cuánto espera un nodo el token antes de declararlo perdido; el segundo cuánto esperan los nodos para alcanzar consenso sobre una nueva membresía. La suma de ambos define el tiempo mínimo para reconstruir el clúster después de que un nodo quede offline.

Mantener la suma por debajo de 30 segundos reduce el tiempo de recuperación. Con HA es especialmente importante mantenerla por debajo de 45 segundos, para que se forme una nueva membresía antes de que expire el watchdog de 60 segundos y se produzca fencing. Se recomienda dejar margen de seguridad.

Consulte los valores actuales en milisegundos con:

```
corosync-cmapctl | grep -Ew 'runtime.config.totem.token|runtime.config.totem.consensus'

runtime.config.totem.consensus (u32) = 5940
runtime.config.totem.token (u32) = 4950
```

En el ejemplo, la suma es 10,89 s. El manual recomienda: optimizar si supera 30 s, reducirlo si supera 40 s y reducirlo de forma especialmente prioritaria si supera 45 s.

Para reducirlo, compruebe primero que la red cumple los requisitos de latencia. Si todavía no existe token_coefficient: 125, añádalo en totem e incremente config_version. Si 125 ms no es suficiente, calcule un valor que mantenga la suma de los timeouts por debajo de 45 s. Corosync utiliza por defecto estas fórmulas:

```
token = 3000 + (number_of_nodes - 2) * token_coefficient
consensus = 1.2 * token
```

Después de ajustar el valor, las versiones recientes de Corosync lo adoptan automáticamente. En versiones anteriores a 3.1.10-pve1 es necesario reiniciar Corosync en todos los nodos.

5.12.3 Resolución de problemas

Error: quorum.expected_votes must be configured

Si Corosync falla al arrancar con un mensaje indicando que nodelist o quorum.expected_votes debe estar configurado, normalmente significa que el hostname indicado en algún ringX_addr no pudo resolverse.

```
corosync[1647]: [QUORUM] Quorum provider: corosync_votequorum failed to initialize.
corosync[1647]: [SERV ] Service engine 'corosync_quorum' failed to load for reason
'configuration error: nodelist or quorum.expected_votes must be configured!'
```

Escribir configuración sin quorum

Si necesita modificar /etc/pve/corosync.conf en un nodo sin quorum y comprende lo que está haciendo, puede establecer temporalmente los votos esperados a 1:

```
pvecm expected 1
```

Esto vuelve quorate el clúster y permite corregir la configuración o restaurar la copia anterior. Si Corosync ya no puede arrancar, edite la copia local /etc/corosync/corosync.conf para permitir su arranque. Asegúrese de que todos los nodos tengan el mismo contenido para evitar split-brain.

5.12.4 Glosario de configuración de Corosync

Parámetro	Descripción
ringX_addr	Nombra las distintas direcciones de enlace utilizadas por las conexiones Kronosnet entre nodos.

5.13 Arranque en frío del clúster

Cuando todos los nodos están apagados el clúster, lógicamente, no tiene quorum. Es una situación habitual después de un corte eléctrico.

NOTA: Es recomendable utilizar un sistema de alimentación ininterrumpida (UPS), especialmente cuando se desea alta disponibilidad.

Al arrancar un nodo, el servicio pve-guests se inicia y espera a que exista quorum. Cuando se alcanza, inicia todos los invitados que tengan activado el indicador onboot.

Tras recuperar la alimentación algunos nodos pueden arrancar antes que otros. El inicio automático de invitados queda retrasado hasta que el clúster alcanza quorum.

5.14 Selección automática del VMID de invitados

Al crear un invitado, la interfaz web solicita automáticamente al backend un VMID libre. El rango de búsqueda predeterminado va de 100 a 1000000, por debajo del máximo permitido por el esquema.

Puede resultar útil reservar rangos distintos -por ejemplo para separar VMs temporales- o usar VMIDs de longitud estable. Los límites inferior, superior o ambos pueden definirse en datacenter.cfg, editable también desde Datacenter → Options.

NOTA: Este rango solo se utiliza para la llamada API next-id; no constituye un límite rígido para los VMID.

5.15 Migración de invitados

Migrar invitados virtuales a otros nodos es una funcionalidad fundamental de un clúster. El comportamiento puede controlarse mediante datacenter.cfg o, para una migración concreta, mediante parámetros de API o línea de comandos.

El comportamiento varía según el invitado esté online u offline y según disponga de recursos locales, como discos locales. Los detalles específicos de migración de VMs se describen en el capítulo de migración QEMU/KVM y los de contenedores en el capítulo de migración de contenedores.

5.15.1 Tipo de migración

El tipo de migración determina si los datos se transmiten por un canal cifrado (secure) o sin cifrar (insecure). Con insecure, el contenido de la RAM del invitado también se transfiere sin cifrar, lo que puede exponer información crítica del interior d el invitado, como contraseñas o claves de cifrado.

Por ello, se recomienda encarecidamente el canal seguro siempre que no se tenga control completo sobre la red o no pueda garantizarse que nadie pueda escuchar el tráfico.

NOTA: La migración de almacenamiento no sigue este ajuste: actualmente el contenido del almacenamiento se transmite siempre por un canal seguro.

El cifrado consume capacidad de cálculo y por este motivo a veces se utiliza insecure para obtener más rendimiento. En sistemas modernos el impacto suele ser menor gracias al soporte de AES por hardware, aunque se hace más visible en redes rápidas de 10 Gbit/s o superiores.

5.15.2 Red de migración

Por defecto, Proxmox VE utiliza para las migraciones la misma red en la que se realiza la comunicación de clúster. No es lo ideal: puede interferir con el tráfico sensible de Corosync y, además, esa red puede no ser la de mayor ancho de banda.

El parámetro migration network permite dedicar una red a todo el tráfico de migración. Afecta tanto a la memoria como al tráfico de almacenamiento de migraciones offline.

La red se expresa en notación CIDR. Así no es necesario indicar una IP individual por nodo: Proxmox VE determina la dirección real del nodo destino a partir del segmento indicado. Para que funcione, cada nodo debe tener exactamente una IP en esa red.

El ejemplo del manual utiliza tres redes separadas: una pública, una de clúster y otra rápida dedicada a migración:

```
iface eno1 inet manual
# public network
auto vmbr0
iface vmbr0 inet static
    address 192.X.Y.57/24
    gateway 192.X.Y.1
    bridge-ports eno1
    bridge-stp off
    bridge-fd 0

# cluster network
auto eno2
iface eno2 inet static
    address 10.1.1.1/24

# fast network
auto eno3
iface eno3 inet static
    address 10.1.2.1/24
```

Para una migración puntual a través de 10.1.2.0/24:

```
# qm migrate 106 tre --online --migration_network 10.1.2.0/24
```

Para establecerla como red predeterminada para todas las migraciones del clúster, configure /etc/pve/datacenter.cfg:

```
# use dedicated migration network
migration: secure,network=10.1.2.0/24
```

NOTA: Cuando se define una red de migración en /etc/pve/datacenter.cfg, también debe establecerse siempre el tipo de migración.



Ejemplo conceptual de una red rápida separada para migración, distinta de la red pública y de la red de Corosync.

Con esta parte queda traducido el Capítulo 5 completo del manual: desde la creación del clúster y sus requisitos hasta la migración de invitados, pasando por quorum, redes de Corosync, QDevice y procedimientos de mantenimiento.

La siguiente parte corresponde al Capítulo 6, “Proxmox Cluster File System (pmxcfs)”.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 6

Capítulo 6 · Proxmox Cluster File System (pmxcfs)

Basado en Proxmox VE Administration Guide, Release 9.2.4

Proxmox Server Solutions GmbH · 4 de agosto de 2026

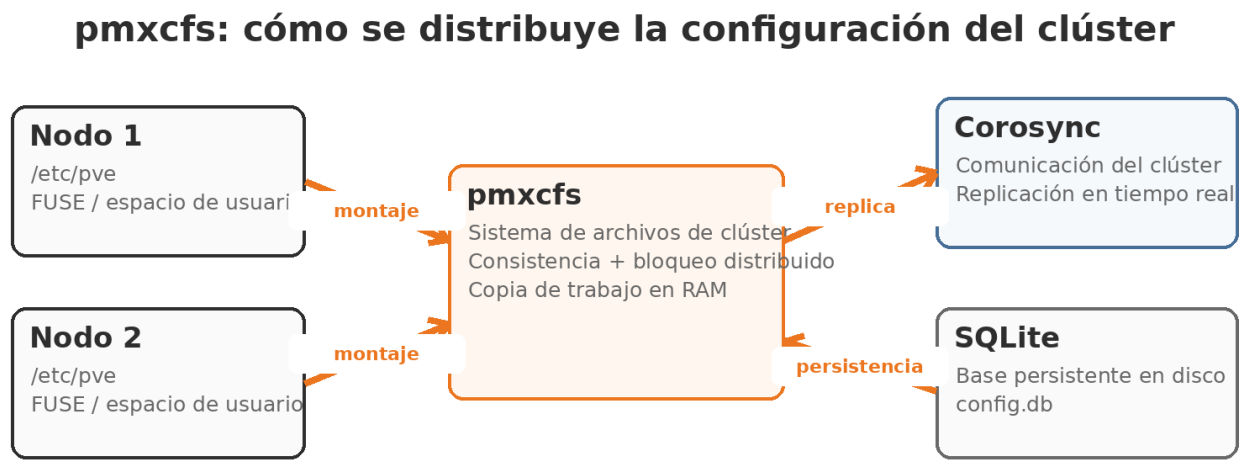
Traducción de trabajo al castellano · GNU Free Documentation License 1.3 o posterior

Capítulo 6 · Proxmox Cluster File System (pmxcfs)

El sistema de archivos de clúster de Proxmox, pmxcfs, es un sistema de archivos basado en una base de datos que almacena los archivos de configuración de Proxmox VE y los replica en tiempo real a todos los nodos del clúster mediante Corosync.

Aunque los datos se almacenan de forma persistente en disco, una copia reside en RAM. Esto limita actualmente el tamaño máximo a 128 MiB, capacidad suficiente para guardar la configuración de varios miles de máquinas virtuales.

- Replicación transparente de toda la configuración en tiempo real entre los nodos.
- Comprobaciones de consistencia estrictas para evitar identificadores de VM duplicados.
- Paso automático a modo de solo lectura cuando un nodo pierde el quorum.
- Actualización automática de la configuración de Corosync en todos los nodos.
- Mecanismo de bloqueo distribuido integrado.



Esquema conceptual del funcionamiento descrito en el capítulo.

6.1 Compatibilidad POSIX

El sistema de archivos se basa en FUSE, por lo que su comportamiento es similar a POSIX. No obstante, algunas funciones no están implementadas porque pmxcfs no las necesita para su finalidad.

- Se pueden crear archivos y directorios normales, pero no enlaces simbólicos arbitrarios.
- No se pueden renombrar directorios que no estén vacíos; esto facilita garantizar la unicidad de los VMID.
- No se pueden cambiar los permisos de archivo: los permisos se determinan en función de la ruta.
- Las creaciones con O_EXCL no eran atómicas, de forma similar a NFS antiguo.
- Las creaciones con O_TRUNC no son atómicas debido a una restricción de FUSE.

6.2 Derechos de acceso a archivos

Todos los archivos y directorios pertenecen al usuario root y al grupo www-data. Solo root dispone de permisos de escritura, mientras que el grupo www-data puede leer la mayoría de los archivos.

Las rutas siguientes solo son accesibles por root:

```
/etc/pve/priv/
/etc/pve/nodes/${NAME}/priv/
```

6.3 Tecnología

pmxcfs utiliza Corosync Cluster Engine para la comunicación del clúster, SQLite para el archivo de base de datos y FUSE para implementar el sistema de archivos en espacio de usuario.

6.4 Estructura del sistema de archivos

El sistema de archivos se monta en:

```
/etc/pve
```

6.4.1 Archivos

Ruta / archivo	Función
authkey.pub	Clave pública utilizada por el sistema de tickets.
ceph.conf	Archivo de configuración de Ceph. /etc/ceph/ceph.conf es un enlace simbólico a este archivo.
corosync.conf	Configuración del clúster Corosync. Antes de Proxmox VE 4.x se denominaba cluster.conf.
datacenter.cfg	Configuración global del Datacenter de Proxmox VE, como distribución de teclado o proxy.
domains.cfg	Dominios de autenticación de Proxmox VE.
firewall/cluster.fw	Configuración de firewall aplicada a todos los nodos.
firewall/<NAME>.fw	Configuración de firewall para nodos individuales.
firewall/<VMID>.fw	Configuración de firewall para máquinas virtuales y contenedores.
ha/crm_commands	Muestra las operaciones HA que está ejecutando en ese momento el CRM.
ha/manager_status	Información en formato JSON sobre los servicios HA del clúster.
ha/resources.cfg	Recursos administrados por alta disponibilidad y su estado actual.
ha/rules.cfg	Reglas que imponen restricciones a la planificación de recursos del gestor HA.
nodes/<NAME>/config	Configuración específica de un nodo.
nodes/<NAME>/lxc/<VMID>.conf	Datos de configuración de máquinas LXC.

Ruta / archivo	Función
nodes/<NAME>/openvz/	Utilizado antes de Proxmox VE 4.0 para datos de configuración de contenedores; obsoleto.
nodes/<NAME>/pve-ssl.key	Clave SSL privada para pve-ssl.pem.
nodes/<NAME>/pve-ssl.pem	Certificado SSL público del servidor web, firmado por la CA del clúster.
nodes/<NAME>/pveproxy-ssl.key	Clave SSL privada opcional para pveproxy-ssl.pem.
nodes/<NAME>/pveproxy-ssl.pem	Cadena de certificado SSL pública opcional para el servidor web; sustituye a pve-ssl.pem.
nodes/<NAME>/qemu-server/<VMID>.conf	Datos de configuración de las máquinas virtuales KVM.
priv/authkey.key	Clave privada utilizada por el sistema de tickets.
priv/authorized_keys	Claves SSH de los miembros del clúster para autenticación.
priv/ceph*	Claves de autenticación de Ceph y capacidades asociadas.
priv/known_hosts	Claves SSH de los miembros del clúster para verificación.
priv/lock/*	Archivos de bloqueo utilizados por servicios para operaciones seguras en todo el clúster.
priv/pve-root-ca.key	Clave privada de la CA del clúster.
priv/shadow.cfg	Archivo shadow de contraseñas para usuarios del realm PVE.
priv/storage/<STORAGE-ID>.pw	Contiene en texto plano la contraseña de un almacenamiento.
priv/tfa.cfg	Configuración de autenticación de dos factores codificada en Base64.
priv/token.cfg	Secretos de todos los tokens API.
pve-root-ca.pem	Certificado público de la CA del clúster.
pve-www.key	Clave privada utilizada para generar tokens CSRF.
sdn/*	Archivos de configuración compartidos de Software Defined Networking (SDN).
status.cfg	Configuración del servidor externo de métricas de Proxmox VE.
storage.cfg	Configuración de almacenamiento de Proxmox VE.
user.cfg	Configuración de control de acceso: usuarios, grupos, etc.
virtual-guest/cpu-models.conf	Almacena modelos de CPU personalizados.
vzdump.cron	Planificación de trabajos de backup vzdump para todo el clúster.

6.4.2 Enlaces

Algunos directorios de pmxcfs utilizan enlaces simbólicos para apuntar a los archivos de configuración propios del nodo local. Por ello, una misma ruta lógica puede referirse a archivos diferentes en cada nodo.

Enlace	Destino
local	nodes/<LOCAL_HOST_NAME>
lxc	nodes/<LOCAL_HOST_NAME>/lxc/
openvz	nodes/<LOCAL_HOST_NAME>/openvz/ (obsoleto)
qemu-server	nodes/<LOCAL_HOST_NAME>/qemu-server/

6.4.3 Archivos especiales de estado para depuración (JSON)

Ruta / archivo	Función
.version	Versiones de archivos, utilizadas para detectar modificaciones.
.members	Información sobre los miembros del clúster.
.vmlist	Lista de todas las máquinas virtuales.
.clusterlog	Registro del clúster; contiene las últimas 50 entradas.
.rrd	Datos RRD más recientes.

6.4.4 Activar o desactivar la depuración

Para activar mensajes detallados en syslog:

```
echo "1" >/etc/pve/.debug
```

Para desactivar de nuevo los mensajes detallados:

```
echo "0" >/etc/pve/.debug
```

6.5 Recuperación

Ante problemas graves en un host Proxmox VE, por ejemplo un fallo de hardware, puede resultar útil copiar la base de datos de pmxcfs:

```
/var/lib/pve-cluster/config.db
```

En el nuevo host, sin cargas en ejecución, se debe detener el servicio pve-cluster y sustituir config.db, manteniendo permisos 0600. Después hay que adaptar /etc/hostname y /etc/hosts de acuerdo con el host Proxmox VE perdido, reiniciar y comprobar el sistema. Los datos de las VM y CT deben estar disponibles por separado.

6.5.1 Eliminar la configuración del clúster

La forma recomendada de proceder después de eliminar un nodo de un clúster es reinstalarlo. Así se garantiza la destrucción de las claves secretas del clúster/SSH y de cualquier configuración compartida.

En determinados casos puede preferirse devolver el nodo a modo local sin reinstalarlo. El manual remite para ello al procedimiento “Separar un nodo sin reinstalar”.

6.5.2 Recuperar o mover invitados desde nodos averiados

Para los archivos de configuración de invitados ubicados en nodes/<NAME>/qemu-server/ (VM) y nodes/<NAME>/lxc/ (contenedores), Proxmox VE considera al nodo <NAME> como propietario lógico del invitado. Este diseño permite usar bloqueos locales en lugar de bloqueos distribuidos más costosos para impedir cambios concurrentes en la configuración.

Si el nodo propietario deja de estar disponible, una migración normal no es posible aunque todos los discos estén en almacenamiento compartido, porque no puede obtenerse el bloqueo local del nodo desconectado. En los invitados gestionados por HA, la pila de alta disponibilidad aporta los mecanismos de bloqueo y watchdog necesarios para una recuperación automática correcta tras el fencing.

Si un invitado que no está gestionado por HA utiliza únicamente discos compartidos y no depende de otros recursos locales del nodo averiado, se puede recuperar manualmente moviendo su archivo de configuración al directorio de un nodo en línea. Esto cambia su propietario lógico.

Ejemplo: recuperar la VM 100 desde node1, fuera de línea, hacia node2:

```
mv /etc/pve/nodes/node1/qemu-server/100.conf  /etc/pve/nodes/node2/qemu-server/
```

ADVERTENCIA: Antes de recuperar manualmente un invitado de esta forma, asegúrese absolutamente de que el nodo de origen averiado está realmente apagado o ha sido fenced. De lo contrario, el comando mv vulnera los principios de bloqueo de Proxmox VE y puede producir consecuencias inesperadas.

ADVERTENCIA: Los invitados con discos locales u otros recursos disponibles únicamente en el nodo fuera de línea no pueden recuperarse mediante este procedimiento. Debe esperarse a que el nodo vuelva al clúster o restaurar esos invitados desde copias de seguridad.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 7A · ALMACENAMIENTO EN PROXMOX VE

Capítulo 7 · secciones 7.1 a 7.9.5

Basado en “Proxmox VE Administration Guide”, Release 9.2.4
Proxmox Server Solutions GmbH · 4 de agosto de 2026

Traducción no oficial · Documento de trabajo

El documento original se distribuye bajo GNU Free Documentation License 1.3 o posterior.

Capítulo 7 · Almacenamiento en Proxmox VE

El modelo de almacenamiento de Proxmox VE es muy flexible. Las imágenes de las máquinas virtuales pueden guardarse en uno o varios almacenamientos locales o en almacenamiento compartido, como NFS o iSCSI (NAS/SAN). No existe un límite práctico impuesto por Proxmox VE al número de pools de almacenamiento que pueden configurarse, y pueden aprovecharse las tecnologías de almacenamiento disponibles en Debian GNU/Linux.

Una de las principales ventajas del almacenamiento compartido es la migración en vivo de máquinas virtuales sin tiempo de inactividad: todos los nodos del clúster acceden directamente a las imágenes de disco y no es necesario copiar sus datos durante la migración.

La biblioteca de almacenamiento, proporcionada por el paquete libpve-storage-perl, utiliza un sistema flexible de plugins que ofrece una interfaz común para los distintos tipos de almacenamiento y permite incorporar nuevos backends en el futuro.

7.1 Tipos de almacenamiento

Proxmox VE distingue, de forma general, dos grandes clases de almacenamiento:

`Almacenamiento a nivel de archivo` — ofrece acceso a un sistema de archivos POSIX completo. Suele ser más flexible que el almacenamiento a nivel de bloque y permite guardar distintos tipos de contenido. ZFS es uno de los sistemas más avanzados y admite instantáneas y clones.

`Almacenamiento a nivel de bloque` — permite almacenar grandes imágenes RAW. Normalmente no se utiliza para otros tipos de archivos, como ISO o copias de seguridad. Muchas implementaciones modernas admiten instantáneas y clones. Ceph RADOS, por ejemplo, replica datos entre nodos y expone RBD (RADOS Block Device).

Descripción	Plugin	Nivel	Compartido	Snapshots	Estable
ZFS (local)	zfspool	ambos[1]	no	sí	sí
Directory	dir	archivo	no	sí[2]	sí
BTRFS	btrfs	archivo	no	sí	TP[5]
NFS	nfs	archivo	sí	sí[2]	sí
CIFS	cifs	archivo	sí	sí[2]	sí
Proxmox Backup	pbs	ambos	sí	n/a	sí
CephFS	cephfs	archivo	sí	sí	sí
LVM	lvm	bloque	no[3]	sí[4]	sí
LVM-thin	lvmthin	bloque	no	sí	sí
iSCSI/kernel	iscsi	bloque	sí[3]	sí[4]	sí
iSCSI/libiscsi	iscsidirect	bloque	sí[3]	sí[4]	sí
FC/SAS nativo[6]	native	bloque	sí[3]	sí[4]	sí
Ceph/RBD	rbd	bloque	sí	sí	sí
ZFS sobre iSCSI	zfs	bloque	sí	sí	sí

[1] En ZFS, las imágenes de disco de VM se almacenan como datasets de volumen ZFS (zvol), que proporcionan funcionalidad de dispositivo de bloque.

[2] En almacenamientos basados en archivos, las instantáneas pueden realizarse con qcow2 mediante snapshots internos o mediante cadenas de volúmenes. Crear o eliminar snapshots internos qcow2 puede bloquear una VM en ejecución y resultar especialmente lento en almacenamientos de red como NFS.

[3] Cuando iSCSI, Fibre Channel (FC) o SAS se usan como almacenamiento de bloques compartido en un clúster, LVM se emplea para dividir el LUN en discos virtuales. En estos escenarios suele ser necesario configurar multipath.

[4] Desde Proxmox VE 9 están disponibles para VM las instantáneas como cadena de volúmenes, utilizando volúmenes separados para los datos de snapshot y superponiéndolos por capas.

[5] Tecnología en vista previa (Technology Preview), con soporte de mejor esfuerzo. [6] FC y SAS se gestionan directamente desde el host sin un plugin de almacenamiento dedicado.

7.1.1 Aprovisionamiento fino (Thin Provisioning)

Varios tipos de almacenamiento y el formato qcow2 de QEMU admiten thin provisioning. Cuando está activado, solo se escriben en el almacenamiento los bloques que el sistema invitado utiliza realmente.

Por ejemplo, una VM puede disponer de un disco virtual de 32 GB y utilizar únicamente 3 GB tras instalar el sistema operativo . En ese caso, solo esos 3 GB ocupan espacio físico, aunque el invitado siga viendo un disco de 32 GB. Esto permite crear discos lógicos mayores que la capacidad física actualmente disponible e ir ampliando el almacenamiento conforme crecen las necesidades.

Todos los tipos de almacenamiento que ofrecen la característica "Snapshots" admiten también thin provisioning.

PRECAUCIÓN: Si un almacenamiento se llena, los invitados que utilicen volúmenes de ese almacenamiento recibirán errores de E/S. Esto puede provocar inconsistencias en los sistemas de archivos y pérdida o corrupción de datos. Evite el sobreaprovisionamiento excesivo o supervise cuidadosamente el espacio libre.

7.2 Configuración del almacenamiento

Toda la configuración de almacenamiento relacionada con Proxmox VE se guarda en un único archivo de texto:

```
/etc/pve/storage.cfg
```

Al estar dentro de /etc/pve/, el archivo se distribuye automáticamente a todos los nodos del clúster. Por tanto, todos comparten la misma configuración de almacenamiento. Esto resulta natural para almacenamiento compartido, pero también es útil con almacenamiento local: la definición puede existir en todos los nodos aunque el contenido físico sea distinto en cada uno.

7.2.1 Pools de almacenamiento

Cada pool tiene un <type> y se identifica de forma única mediante un <STORAGE_ID>. La estructura general es:

```
<type>: <STORAGE_ID>
    <property> <value>
    <property> <value>
    <property>
```

La línea <type>: <STORAGE_ID> inicia la definición del pool. A continuación se enumeran las propiedades. La mayoría requieren un valor; algunas disponen de valores predeterminados razonables.

Configuración de almacenamiento típica tras una instalación:

```
dir: local
    path /var/lib/vz
    content iso,vztmpl,backup

# almacén de imágenes por defecto en una instalación basada en LVM
lvmthin: local-lvm
    thinpool data
    vgname pve
    content rootdir,images

# almacén de imágenes por defecto en una instalación basada en ZFS
zfspool: local-zfs
    pool rpool/data
    sparse
    content images,rootdir
```

PRECAUCIÓN: Es problemático definir varias configuraciones que apunten exactamente al mismo almacenamiento subyacente. Dos identificadores de volumen distintos podrían terminar apuntando a la misma imagen de disco, algo contrario a la expectativa de unicidad de Proxmox VE. Separar por tipos de contenido puede funcionar, pero no es recomendable.

7.2.2 Propiedades comunes de almacenamiento

nodes — lista de nodos del clúster en los que el almacenamiento es accesible. Permite restringirlo a un conjunto de nodos.

content — selecciona los tipos de contenido que puede almacenar el backend.

images — imágenes de máquinas virtuales QEMU/KVM.

rootdir — datos raíz de contenedores.

vztmpl — plantillas de contenedores.

backup — archivos de copia de seguridad de vzdump.

iso — imágenes ISO.

snippets — fragmentos, por ejemplo scripts hook de invitados.

import — OVA e imágenes de disco de VM que pueden importarse desde el almacenamiento.

shared — marca que se trata de un único almacenamiento con el mismo contenido visible en todos los nodos (o en los indicados por nodes). No convierte un almacenamiento local en compartido; solo declara como compartido uno que ya lo es.

disable — deshabilita completamente el almacenamiento.

prune-backups — opciones de retención de copias de seguridad.

format — formato de imagen predeterminado: raw, qcow2 o vmdk.

preallocation — modo de preasignación para imágenes raw y qcow2 en almacenamientos basados en archivos: off, metadata, falloc o full. El valor predeterminado es metadata; para raw se trata como off. En almacenamientos de red con imágenes qcow2 grandes, off puede ayudar a evitar timeouts.

ADVERTENCIA: No es aconsejable utilizar el mismo pool de almacenamiento desde clústeres Proxmox VE diferentes. Algunas operaciones requieren acceso exclusivo y bloqueo correcto; el bloqueo funciona dentro de un clúster, pero no entre clústeres distintos.

7.3 Volúmenes

Proxmox VE utiliza una notación especial para direccionar los datos de almacenamiento. Al asignar datos desde un pool se devuelve un identificador de volumen. Un <VOLUME_ID> se forma con el <STORAGE_ID>, seguido de dos puntos y un nombre dependiente del tipo de almacenamiento.

```
local:230/example-image.raw
local:iso/debian-501-amd64-netinst.iso
local:vztmpl/debian-5.0-joomla_1.5.9-1_i386.tar.gz
iscsi-storage:0.0.2.scsi-14f504e46494c4500494b5042546d2d646744372d31616d61
```

Para obtener la ruta del sistema de archivos correspondiente a un identificador de volumen:

```
pvesm path <VOLUME_ID>
```

7.3.1 Propiedad de los volúmenes

Existe una relación de propiedad para los volúmenes de tipo imagen. Cada volumen pertenece a una VM o a un contenedor. Por ejemplo, local:230/example-image.raw pertenece a la VM 230. La mayoría de los backends codifican esta información de propiedad en el propio nombre del volumen.

Cuando se elimina una VM o un contenedor, el sistema elimina también los volúmenes asociados que pertenecen a ese invitado.

7.4 Uso de la interfaz de línea de comandos

Aunque conviene comprender los pools y los identificadores de volumen, las operaciones habituales de asignación y eliminación suelen realizarlas las herramientas de gestión de VM y contenedores. Para tareas directas existe pvesm, el Proxmox VE Storage Manager.

7.4.1 Ejemplos

Añadir pools de almacenamiento:

```
pvesm add <TYPE> <STORAGE_ID> <OPTIONS>
pvesm add dir <STORAGE_ID> --path <PATH>
pvesm add nfs <STORAGE_ID> --path <PATH> --server <SERVER> --export <EXPORT>
pvesm add lvm <STORAGE_ID> --vgname <VGNAME>
pvesm add iscsi <STORAGE_ID> --portal <HOST[:PORT]> --target <TARGET>
```

Deshabilitar o habilitar un pool:

```
pvesm set <STORAGE_ID> --disable 1
pvesm set <STORAGE_ID> --disable 0
```

Cambiar opciones:

```
pvesm set <STORAGE_ID> <OPTIONS>
pvesm set <STORAGE_ID> --shared 1
pvesm set local --format qcow2
pvesm set <STORAGE_ID> --content iso
```

Eliminar una definición de almacenamiento (no elimina datos ni desmonta/desconecta el almacenamiento):

```
pvesm remove <STORAGE_ID>
```

Asignar volúmenes:

```
pvesm alloc <STORAGE_ID> <VMID> <name> <size> [--format <raw|qcow2>]
pvesm alloc local <VMID> '' 4G
```

Liberar un volumen:

```
pvesm free <VOLUME_ID>
```

ADVERTENCIA: El comando pvesm free destruye realmente todos los datos del volumen.

Consultar estado, contenido y rutas:

```
pvesm status
pvesm list <STORAGE_ID> [--vmid <VMID>]
pvesm list <STORAGE_ID> --vmid <VMID>
pvesm list <STORAGE_ID> --content iso
pvesm list <STORAGE_ID> --content vztmpl
pvesm path <VOLUME_ID>
```

Ejemplo de exportación de un volumen. El formato de flujo qcow2+size no es el mismo que un archivo qcow2 adjuntable directamente a una VM:

```
pvesm export local:103/vm-103-disk-0.qcow2 qcow2+size target --with-snapshots 1
```

7.5 Backend de directorio

Tipo de pool: dir

Proxmox VE puede utilizar directorios locales o recursos montados localmente. Es almacenamiento a nivel de archivo, por lo que admite contenidos como imágenes de VM, contenedores, plantillas, ISO o copias de seguridad.

NOTA: Puede montar almacenamiento adicional mediante /etc/fstab y definir después un almacenamiento de tipo directorio sobre ese punto de montaje. Así puede utilizar cualquier sistema de archivos soportado por Linux.

El backend presupone compatibilidad POSIX del directorio subyacente, pero no exige características adicionales. Por ello no ofrece snapshots a nivel de almacenamiento. Para imágenes de VM puede utilizarse qcow2, que dispone de snapshots internos.

CONSEJO: Algunos tipos de almacenamiento no admiten O_DIRECT y, por tanto, no pueden utilizar cache mode none. En esos casos use writeback.

Los backends a nivel de archivo utilizan una disposición de directorios predefinida:

Tipo de contenido	Subdirectorio
Imágenes de VM	`images/<VMID>/`
Imágenes ISO	`template/iso/`
Plantillas de contenedor	`template/cache/`
Copias de seguridad	`dump/`
Snippets	`snippets/`
Importación	`import/`

7.5.1 Configuración

Además de las propiedades comunes, este backend añade path y content-dirs. path debe ser una ruta absoluta. content-dirs permite modificar la disposición predeterminada mediante una lista separada por comas con formato vtype=path, donde vtype es un tipo de contenido permitido y path es relativo al punto de montaje.

```
dir: backup
    path /mnt/backup
    content backup
    prune-backups keep-last=7
    max-protected-backups 3
    content-dirs backup=custom/backup/dir
```

Este ejemplo define el pool backup. Puede conservar hasta 7 copias regulares y 3 copias protegidas por VM. Los archivos se guardarán bajo /mnt/backup/custom/backup/dir/...

7.5.2 Convenciones de nombres

```
vm-<VMID>-<NAME>.<FORMAT>
```

<VMID> — identifica la VM propietaria.

<NAME> — nombre ASCII sin espacios. El backend usa disk-[N] por defecto, incrementando N para garantizar unicidad.

<FORMAT> — raw, qcow2 o vmdk.

Al convertir una VM en plantilla, sus imágenes se renombran para indicar que son de solo lectura y que pueden usarse como base de clones:

```
base-<VMID>-<NAME>.<FORMAT>
```

NOTA: Las imágenes base deben permanecer inmutables. El backend cambia los permisos a 0444 y, si el almacenamiento lo soporta, establece el atributo inmutable con chattr +i.

7.5.3 Características

Como la mayoría de los sistemas de archivos no ofrecen snapshots nativos, el backend de directorio utiliza las capacidades internas de qcow2 para snapshots y clones.

7.5.4 Ejemplos

```
# pvesm alloc local 100 vm-100-disk10.raw 4G
Formatting '/var/lib/vz/images/100/vm-100-disk10.raw', fmt=raw size=4294967296
successfully created 'local:100/vm-100-disk10.raw'

# pvesm path local:100/vm-100-disk10.raw
/var/lib/vz/images/100/vm-100-disk10.raw

# pvesm free local:100/vm-100-disk10.raw
```

7.6 Backend NFS

Tipo de pool: nfs

El backend NFS se basa en el backend de directorio y comparte su disposición y convenciones de nombres. Su ventaja principal es que permite configurar directamente el servidor NFS y montar automáticamente el recurso, sin modificar /etc/fstab. También puede comprobar si el servidor está disponible y consultar sus exportaciones.

7.6.1 Configuración

server — IP o nombre DNS del servidor. Para evitar retrasos de resolución se suele preferir una IP, salvo que el DNS sea muy fiable o el servidor figure en /etc/hosts.

export — ruta NFS exportada, tal como la muestra pvesm scan nfs.

path — punto de montaje local; por defecto /mnt/pve/<STORAGE_ID>/.

content-dirs — sustituciones para la disposición de directorios predeterminada.

options — opciones de montaje NFS; consulte man nfs.

```
nfs: iso-templates
    path /mnt/pve/iso-templates
    server 10.0.0.10
    export /space/iso-templates
    options vers=3,soft
    content iso,vztmpl
```

CONSEJO: Tras un timeout, las peticiones NFS se reintentan indefinidamente por defecto. Para contenido de solo lectura puede ser útil considerar la opción soft, que limita el número de reintentos a tres.

7.6.2 Características

NFS no ofrece snapshots a nivel de almacenamiento, pero el backend utiliza las funciones de qcow2 para implementar snapshots y clonación.

7.6.3 Ejemplos

```
# pvesm scan nfs <server>
```

7.7 Backend CIFS

Tipo de pool: cifs

El backend CIFS amplía el backend de directorio para evitar el montaje manual de recursos CIFS. Puede añadirse directamente mediante la API o la interfaz web, con comprobación de disponibilidad y selección de recursos exportados.

7.7.1 Configuración

server — IP o nombre DNS del servidor. Obligatorio.

CONSEJO: Para evitar demoras de DNS, suele ser preferible utilizar la dirección IP, salvo que disponga de DNS fiable o una entrada en /etc/hosts.

share — recurso CIFS que se utilizará; puede consultarse con pvesm scan cifs <address> o desde la interfaz web. Obligatorio.

username — usuario del almacenamiento CIFS. Opcional; por defecto guest.

password — contraseña del usuario. Se guarda en /etc/pve/priv/storage/<STORAGE-ID>.pw con acceso exclusivo para root.

domain — dominio o grupo de trabajo. Opcional.

smbversion — versión del protocolo SMB. Opcional; por defecto 3. SMB1 no está soportado por motivos de seguridad.

path — punto de montaje local. Por defecto /mnt/pve/<STORAGE_ID>/.

content-dirs — sustituciones para la disposición de directorios. Opcional.

options — opciones adicionales de mount.cifs. Proxmox VE siempre fija soft y puede establecer automáticamente username, credentials, guest, domain y vers.

subdir — subdirectorio del recurso que se montará. Opcional; por defecto la raíz.

```
cifs: backup
    path /mnt/pve/backup
    server 10.0.0.11
    share VMData
    content backup
    options noserverino,echo_interval=30
    username anna
    smbversion 3
    subdir /data
```

7.7.2 Características

CIFS no soporta snapshots a nivel de almacenamiento. Si se necesitan snapshots y clonación pueden utilizarse archivos base qcow2.

7.7.3 Ejemplos

```
# pvesm scan cifs <server> [--username <username>] [--password]
# pvesm add cifs <storagename> --server <server> --share <share> [--username <username>] [--password]
```

7.8 Proxmox Backup Server

Tipo de pool: pbs

Este backend integra directamente Proxmox Backup Server en Proxmox VE como un almacenamiento más. Puede añadirse mediante API, CLI o interfaz web.

7.8.1 Configuración

server — IP o nombre DNS del servidor. Obligatorio.

port — puerto distinto del predeterminado 8007. Opcional.

username — usuario de Proxmox Backup Server. Obligatorio.

CONSEJO: No olvide incluir el realm en el nombre de usuario, por ejemplo root@pam o archiver@pbs.

password — contraseña del usuario. Se guarda en /etc/pve/priv/storage/<STORAGE-ID>.pw con acceso restringido a root. Obligatoria.

datastore — ID del datastore de Proxmox Backup Server. Obligatorio.

fingerprint — huella del certificado TLS de la API de PBS. Es obligatoria con certificados autofirmados o cuando la CA del servidor no es de confianza para el host.

encryption-key — clave para cifrar los datos de backup en el cliente. Se guarda en /etc/pve/priv/storage/<STORAGE-ID>.enc. El valor especial autogen genera una nueva clave mediante proxmox-backup-client key create --kdf none <path>. Opcional.

master-pubkey — clave pública RSA utilizada para cifrar la clave de cifrado del backup como parte de la tarea. Permite recuperación posterior. Se guarda en /etc/pve/priv/storage/<STORAGE-ID>.master.pem. Opcional; requiere encryption-key.

```
pbs: backup
    datastore main
    server enya.proxmox.com
    content backup
    fingerprint 09:54:ef:..snip..:88:af:47:fe:4c:3b:cf:8b:26:88:0b:4e:3c:b2
    prune-backups keep-all=1
    username archiver@pbs
    encryption-key a9:ee:c8:02:13:..snip..:2d:53:2c:98
    master-pubkey 1
```

7.8.2 Características

Proxmox Backup Server solo admite contenido de copia de seguridad. Proxmox VE utiliza backups a nivel de bloque para máquinas virtuales y a nivel de archivo para contenedores.

7.8.3 Cifrado

Opcionalmente puede configurarse cifrado del lado del cliente con AES-256 en modo GCM, ya sea desde la interfaz web o mediante la opción encryption-key en CLI. La clave se almacena en /etc/pve/priv/storage/<STORAGE-ID>.enc y solo es accesible por root.

ADVERTENCIA: Sin la clave, las copias cifradas serán inaccesibles. La clave debe conservarse separada del sistema que se está protegiendo. Si la única copia de la clave está dentro del sistema que falla, la restauración puede resultar imposible.

Se recomienda guardar la clave en un gestor de contraseñas y mantener además una copia en un dispositivo USB almacenado en un lugar seguro. Para un escenario extremo también puede conservarse una copia en papel. El subcomando paperkey genera una representación codificada en QR:

```
# proxmox-backup-client key paperkey /etc/pve/priv/storage/<STORAGE-ID>.enc --output-format text >
qrkey.txt
```

También puede utilizarse un único par de claves maestras RSA para recuperación. Los clientes cifran una copia de su clave AES con la clave pública maestra y la almacenan junto a cada backup. La clave privada maestra permite recuperar la clave AES y descifrar el backup aunque el sistema cliente ya no exista.

ADVERTENCIA: La clave privada maestra necesita las mismas medidas de custodia. Sin una copia de ella no será posible recuperar las claves AES protegidas con ese par.

Como el cifrado se gestiona en el cliente, un mismo datastore puede almacenar copias cifradas y sin cifrar, incluso con claves diferentes. Sin embargo, no existe deduplicación entre copias cifradas con claves distintas, por lo que a menudo conviene usar datastores separados.

NOTA: No utilice cifrado si no aporta un beneficio real. En un entorno local y de confianza, las copias sin cifrar son más sencillas de recuperar.

7.8.4 Ejemplo: añadir almacenamiento por CLI

```
# pvesm scan pbs <server> <username> [--password <string>] [--fingerprint <string>]
# pvesm add pbs <id> --server <server> --datastore <datastore> --username <username> --fingerprint
00:B4:... --password
```

7.9 Backend ZFS local

Tipo de pool: zfspool

Este backend permite acceder a pools ZFS locales o a sistemas de archivos ZFS contenidos dentro de ellos.

7.9.1 Configuración

pool — pool o sistema de archivos ZFS en el que se realizan las asignaciones.

blocksize — valor de blocksize de ZFS.

sparse — activa thin provisioning en ZFS. Un volumen sparse no reserva físicamente todo su tamaño lógico.

mountpoint — punto de montaje expuesto por el backend. Cambiarlo no modifica la propiedad mountpoint del dataset vista por zfs. Por defecto /<pool>.

```
zfspool: vmdata
    pool tank/vmdata
    content rootdir,images
    sparse
```

7.9.2 Convenciones de nombres

```
vm-<VMID>-<NAME>      # imagen normal de VM
base-<VMID>-<NAME>    # imagen de plantilla de VM (solo lectura)
subvol-<VMID>-<NAME>  # subvolumen ZFS para contenedores
```

<VMID> — identifica la VM propietaria.

<NAME> — nombre ASCII sin espacios. El backend usa disk[N] por defecto.

7.9.3 Características

ZFS es uno de los tipos de almacenamiento más avanzados para snapshots y clonación. El backend usa datasets ZFS tanto para imágenes de VM (formato raw) como para datos de contenedores (formato subvol). Las propiedades ZFS se heredan del dataset padre, por lo que pueden establecerse valores predeterminados en dicho padre.

7.9.4 Cifrado

ZFS admite cifrado de jerarquías de datasets. Cada dataset cifrado es una raíz de cifrado (encryption_root), que almacena las propiedades de cifrado, o un descendiente de dicha raíz.

El envío y recepción de datasets cifrados está sujeto a restricciones: algunos parámetros de cifrado, como los vectores de inicialización, deben transferirse con los datos o regenerarse en destino. En las migraciones y replicaciones de almacenamien to de datasets cifrados en Proxmox VE, los datos se envían sin las propiedades de cifrado y el estado final de cifrado lo determina el destino.

Si desea enviar y recibir datasets ZFS cifrados, debe asegurarse de que todos los almacenamientos implicados, en todos los nodos, estén cifrados.

7.9.5 Ejemplos

Se recomienda crear un sistema de archivos ZFS específico para las imágenes de VM:

```
# zfs create tank/vmdata
```

Para activar la compresión:

```
# zfs set compression=on tank/vmdata
```

Para listar los sistemas de archivos ZFS disponibles:

```
# pvesm scan zfs
```

Fin de la Parte 7A

La Parte 7B continuará con 7.10 LVM, LVM-thin, iSCSI, Ceph RBD/CephFS, BTRFS y ZFS sobre iSCSI.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 7B · ALMACENAMIENTO EN PROXMOX VE

Capítulo 7 · secciones 7.10 a 7.17.2

Basado en "Proxmox VE Administration Guide", Release 9.2.4

Proxmox Server Solutions GmbH · 4 de agosto de 2026

Traducción no oficial · Documento de trabajo

El documento original se distribuye bajo GNU Free Documentation License 1.3 o posterior.

Capítulo 7 · Almacenamiento en Proxmox VE

Esta segunda parte continúa el capítulo de almacenamiento desde el backend LVM hasta ZFS sobre iSCSI. El contenido se centra en almacenamiento de bloques, thin provisioning, iSCSI, Ceph RBD/CephFS y BTRFS, manteniendo la estructura y terminología del manual original.

Fuente: traducción no oficial del capítulo 7 de "Proxmox VE Administration Guide", Release 9.2.4 (4 de agosto de 2026). Se conservan nombres de comandos, parámetros, rutas y formatos técnicos del original.

7.10 Backend LVM

Tipo de pool de almacenamiento: **lvm**

LVM es una capa de software ligera situada sobre discos duros y particiones. Permite dividir el espacio disponible en volúmenes lógicos más pequeños.

Otro caso de uso consiste en colocar LVM sobre una LUN iSCSI de gran tamaño o sobre una SAN conectada mediante Fibre Channel. Así se puede gestionar con facilidad el espacio de una LUN iSCSI, algo que de otro modo resultaría difícil porque la especificación iSCSI no define una interfaz de administración para la asignación de espacio.

7.10.1 Configuración

El backend LVM admite las propiedades comunes `content, nodes, disable` y las siguientes propiedades específicas de LVM:

`vgname` Nombre del grupo de volúmenes LVM. Debe apuntar a un grupo de volúmenes existente.

`base` Volumen base. Este volumen se activa automáticamente antes de acceder al almacenamiento. Resulta especialmente útil cuando el grupo de volúmenes LVM reside en un servidor iSCSI remoto.

`saferemove` En la interfaz web se denomina "Wipe Removed Volumes". Pone a cero los datos al eliminar volúmenes lógicos. Así se evita que datos anteriores puedan ser recuperados por otros LV creados posteriormente que reciban los mismos extents físicos. Es una operación costosa, pero puede ser necesaria como medida de seguridad.

Los dispositivos de almacenamiento que admiten la operación "write zeroes" utilizan `blkdiscard` para poner a cero los bloques. En caso contrario, se recurre a `cstream`.

`saferemove-stepsize` Tamaño de paso de borrado en MiB (valor del parámetro blkdiscard -p), limitado al tamaño máximo que admita el almacenamiento subyacente. El valor máximo predeterminado es 32 MiB.

`saferemove_throughput` Rendimiento del borrado (valor del parámetro cstream -t), con un máximo predeterminado de 10 MiB/s.

`snapshot-as-volume-chain` Activa el soporte de snapshots para máquinas virtuales en LVM mediante una cadena de volúmenes de respaldo.

Con esta opción, al crear un snapshot se conserva el estado actual con el nombre del snapshot y se inicia un nuevo volumen respaldado por ese snapshot. Cada volumen basado en un snapshot referencia al volumen snapshot padre como volumen de respaldo y registra únicamente las diferencias.

Los volúmenes snapshot son, actualmente, volúmenes lógicos LVM con aprovisionamiento completo. El diseño evita problemas de los snapshots LVM nativos, como penalizaciones importantes de E/S y comportamientos peligrosos cuando se agota el espacio preasignado.

Los snapshots como cadenas de volúmenes proporcionan soporte independiente del fabricante para cualquier sistema que admita almacenamiento de bloques, incluidos SAN conectadas por iSCSI y Fibre Channel. Aunque esta función se apoya en qcow2, solo utiliza su capacidad para encadenar volúmenes; la gestión del snapshot la realiza el subsistema de almacenamiento de Proxmox VE.

Activar o desactivar este indicador solo afecta a los nuevos discos virtuales. Para un funcionamiento eficiente, el almacenamiento inferior debe admitir thin provisioning y discard. Cada snapshot parecerá ocupar el tamaño completo desde Proxmox VE, aunque el uso real de espacio será menor si se cumplen esas condiciones.

ADVERTENCIA Los snapshots como cadenas de volúmenes son actualmente una función en fase de vista previa tecnológica en Proxmox VE.

```
lvm: myspace
    vgname myspace
    content rootdir,images
```

7.10.2 Convenciones de nombres de archivo

El backend utiliza básicamente las mismas convenciones de nombres que el backend de pool ZFS:

```
vm-<VMID>-<NAME>     # imágenes normales de VM
```

7.10.3 Funciones de almacenamiento

LVM es un sistema típico de almacenamiento por bloques. Los snapshots LVM convencionales son ineficientes porque interfieren con todas las operaciones de escritura del grupo de volúmenes mientras el snapshot permanece activo, provocando una degradación importante de E/S. Por ello LVM no admite clones enlazados, y Proxmox VE incorporó el soporte de snapshots como cadenas de volúmenes.

Una ventaja de LVM es que puede utilizarse con almacenamiento compartido, por ejemplo una LUN iSCSI. Si el almacenamiento está marcado como compartido, el backend implementa el bloqueo adecuado a nivel de todo el clúster.

CONSEJO Para almacenamiento local no compartido puede utilizarse el backend LVM-thin, que admite snapshots y clones enlazados.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones completos	Clones enlazados
images, rootdir	raw, qcow2[1]	posible	sí[1]	sí	no

[1] *Desde Proxmox VE 9 están disponibles los snapshots como cadena de volúmenes para VM; consulte la configuración LVM anterior.*

7.10.4 Ejemplos

Para obtener la lista de grupos de volúmenes LVM disponibles:

```
# pvesm scan lvm
```

7.11 Backend LVM-thin

Tipo de pool de almacenamiento: **lvmthin**

LVM normalmente asigna bloques al crear un volumen. Los thin pools de LVM, en cambio, asignan los bloques cuando se escriben. Este comportamiento se denomina thin provisioning, ya que los volúmenes pueden ser mucho mayores que el espacio físico disponible.

Puede utilizar las herramientas habituales de línea de comandos de LVM para administrar y crear thin pools. Si ya existe un grupo de volúmenes llamado `pve`, estos comandos crean un thin pool de 100 G llamado `data`:

```
lvcreate -L 100G -n data pve
lvconvert --type thin-pool pve/data
```

7.11.1 Configuración

El backend LVM-thin admite las propiedades comunes `content, nodes, disable` y las siguientes propiedades específicas:

`vgname` Nombre del grupo de volúmenes LVM. Debe apuntar a un grupo de volúmenes existente.

`thinpool` Nombre del thin pool LVM.

```
lvmthin: local-lvm
    thinpool data
    vgname pve
    content rootdir,images
```

7.11.2 Convenciones de nombres de archivo

El backend utiliza básicamente las mismas convenciones de nombres que el backend de pool ZFS:

```
vm-<VMID>-<NAME>     # imágenes normales de VM
```

7.11.3 Funciones de almacenamiento

LVM-thin es almacenamiento por bloques, pero admite de forma eficiente snapshots y clones. Los nuevos volúmenes se inicializan automáticamente a cero.

NOTA Los thin pools LVM no pueden compartirse entre varios nodos; por tanto, solo pueden utilizarse como almacenamiento local.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones
images, rootdir	raw	no	sí	sí

7.11.4 Ejemplos

Para listar los thin pools LVM disponibles dentro del grupo de volúmenes `pve`:

```
# pvesm scan lvmthin pve
```

7.12 Iniciador Open-iSCSI

Tipo de pool de almacenamiento: **iscsi**

iSCSI es una tecnología ampliamente utilizada para conectarse a servidores de almacenamiento. Prácticamente todos los fabricantes de almacenamiento ofrecen soporte iSCSI. También existen soluciones open source de destino iSCSI, por ejemplo OpenMediaVault, basado en Debian.

Para utilizar este backend debe instalarse el paquete Open-iSCSI (`open-iscsi`). Es un paquete estándar de Debian, pero no se instala de forma predeterminada para ahorrar recursos.

```
# apt-get install open-iscsi
```

Las tareas de administración iSCSI de bajo nivel pueden realizarse con la herramienta `iscsiadm`.

7.12.1 Configuración

El backend admite las propiedades comunes `content, nodes, disable` y las siguientes propiedades específicas de iSCSI:

`portal` Portal iSCSI (IP o nombre DNS con puerto opcional).

`target` Destino iSCSI.

```
iscsi: mynas
    portal 10.10.10.1
    target iqn.2006-01.openfiler.com:tsn.dcb5aaaddd
    content none
```

CONSEJO Si desea utilizar LVM sobre iSCSI, tiene sentido establecer “content none”. De este modo no será posible crear VM utilizando directamente las LUN iSCSI.

7.12.2 Convenciones de nombres de archivo

El protocolo iSCSI no define una interfaz para asignar o eliminar datos. Esa tarea debe realizarse en el destino y depende del fabricante. El destino simplemente exporta LUN numeradas. Por ello, los nombres de volúmenes iSCSI de Proxmox VE codifican información sobre la LUN tal y como la ve el kernel de Linux.

7.12.3 Funciones de almacenamiento

iSCSI es almacenamiento a nivel de bloque y no proporciona una interfaz de gestión del espacio. Por ello suele ser preferible exportar una LUN grande y configurar LVM encima. Después puede utilizarse el plugin LVM para gestionar el espacio de esa LUN iSCSI.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones
images / none	raw	sí	no	no

7.12.4 Ejemplos

Para explorar un portal iSCSI remoto y obtener la lista de destinos posibles:

```
pvesm scan iscsi <HOST[:PORT]>
```

7.13 Backend iSCSI en modo usuario

Tipo de pool de almacenamiento: **iscsidirect**

Este backend proporciona esencialmente la misma funcionalidad que Open-iSCSI, pero utiliza una biblioteca en espacio de usuario para implementarla. Para utilizarlo debe instalarse el paquete `libiscsi-bin`.

No intervienen controladores del kernel, por lo que puede considerarse una optimización de rendimiento. La contrapartida es que no puede utilizarse LVM sobre estas LUN iSCSI; toda la asignación de espacio debe gestionarse en el servidor de almacenamiento.

7.13.1 Configuración

El backend iSCSI en modo usuario utiliza las mismas opciones de configuración que el backend Open-iSCSI.

```
iscsidirect: faststore
    portal 10.10.10.1
    target iqn.2006-01.openfiler.com:tsn.dcb5aaaddd
```

7.13.2 Funciones de almacenamiento

NOTA Este backend funciona únicamente con máquinas virtuales. Los contenedores no pueden utilizar este controlador.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones
images	raw	sí	no	no

7.14 Dispositivos de bloques Ceph RADOS (RBD)

Tipo de pool de almacenamiento: **rbd**

Ceph es un almacén de objetos distribuido y un sistema de archivos diseñado para ofrecer un rendimiento, fiabilidad y escalabilidad elevados. Los dispositivos de bloques RADOS implementan un almacenamiento a nivel de bloque muy completo y aportan las siguientes ventajas:

- thin provisioning
- volúmenes redimensionables
- distribución y redundancia, repartiendo los datos entre varios OSD
- funciones completas de snapshot y clonación
- autorrecuperación
- sin punto único de fallo
- escalabilidad hasta nivel exabyte
- implementaciones disponibles tanto en kernel como en espacio de usuario

NOTA En despliegues pequeños también es posible ejecutar servicios Ceph directamente en los nodos Proxmox VE. El hardware reciente suele disponer de CPU y RAM suficientes para ejecutar en el mismo nodo servicios de almacenamiento y VM.

7.14.1 Configuración

Este backend admite las propiedades comunes `nodes, disable, content` y las siguientes propiedades específicas de RBD:

`monhost` Lista de direcciones IP de los monitores. Opcional; solo se necesita cuando Ceph no se ejecuta en el clúster Proxmox VE.

`pool` Nombre del pool de Ceph.

`username` ID de usuario RBD. Opcional; solo se necesita cuando Ceph no se ejecuta en el clúster Proxmox VE. Debe indicarse solo el ID de usuario, sin el prefijo “client.”.

`krbd` Fuerza el acceso a los dispositivos de bloques RADOS mediante el módulo de kernel krbd. Opcional.

NOTA Los contenedores utilizarán krbd independientemente del valor configurado en esta opción.

```
rbd: ceph-external
    monhost 10.1.1.20 10.1.1.21 10.1.1.22
    pool ceph-external
    content images
    username admin
```

CONSEJO Puede utilizar la utilidad rbd para tareas de administración de bajo nivel.

7.14.2 Autenticación

NOTA Si Ceph está instalado localmente en el clúster Proxmox VE, estos pasos se realizan automáticamente al añadir el almacenamiento.

Si utiliza autenticación `cephx`, habilitada de forma predeterminada, debe proporcionar el keyring del clúster Ceph externo.

Para configurar el almacenamiento mediante CLI, primero haga accesible el archivo que contiene el keyring. Por ejemplo:

```
# scp <external cephserver>:/etc/ceph/ceph.client.admin.keyring /root/rbd.keyring
```

Después utilice `pvesm` con el parámetro `--keyring`:

```
# pvesm add rbd <name> --monhost "10.1.1.20 10.1.1.21 10.1.1.22" --content images --keyring /root/rbd.keyring
```

Si se configura desde la interfaz web, puede copiar y pegar el keyring en el campo correspondiente. El keyring se almacenará en:

```
/etc/pve/priv/ceph/<STORAGE_ID>.keyring
```

CONSEJO Al conectar con un clúster externo se recomienda crear un keyring que incluya únicamente las capacidades necesarias.

7.14.3 Configuración del cliente Ceph (opcional)

Al conectarse a un almacenamiento Ceph externo no siempre es posible establecer opciones específicas del cliente en la base de datos de configuración del clúster externo. Puede añadir un archivo `ceph.conf` junto al keyring para modificar la configuración del cliente Ceph de ese almacenamiento.

El archivo debe tener el mismo nombre que el almacenamiento:

```
/etc/pve/priv/ceph/<STORAGE_ID>.conf
```

NOTA No cambie estos parámetros a la ligera. Proxmox VE combina el archivo <STORAGE_ID>.conf con la configuración del almacenamiento.

7.14.4 Funciones de almacenamiento

El backend RBD es almacenamiento a nivel de bloque e implementa funciones completas de snapshots y clonación.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones
images, rootdir	raw	sí	sí	sí

7.15 Sistema de archivos Ceph (CephFS)

Tipo de pool de almacenamiento: **cephfs**

CephFS implementa un sistema de archivos compatible con POSIX utilizando un clúster Ceph para almacenar los datos. Al construirse sobre Ceph, comparte muchas de sus propiedades: redundancia, escalabilidad, autorrecuperación y alta disponibilidad.

CONSEJO Proxmox VE puede gestionar despliegues Ceph, lo que simplifica la configuración de CephFS. En hardware moderno es posible ejecutar servicios de almacenamiento y VM en el mismo nodo.

Para utilizar el plugin CephFS debe sustituir el cliente Ceph estándar de Debian añadiendo el repositorio Ceph de Proxmox. Después ejecute `apt update` seguido de `apt dist-upgrade` para obtener los paquetes más recientes.

ADVERTENCIA Asegúrese de que no existan otros repositorios Ceph configurados. De lo contrario, la instalación puede fallar o mezclarse versiones de paquetes, provocando un comportamiento inesperado.

7.15.1 Configuración

Este backend admite las propiedades comunes `nodes`, `disable`, `content` y las siguientes propiedades específicas de CephFS:

`fs-name` Nombre del sistema de archivos CephFS.

`monhost` Lista de direcciones de monitores. Opcional; solo necesaria si Ceph no se ejecuta en el clúster Proxmox VE.

`path` Punto de montaje local. Opcional; por defecto /mnt/pve/<STORAGE_ID>/.

`username` ID de usuario Ceph. Opcional; solo necesario si Ceph no se ejecuta en el clúster Proxmox VE, donde el valor predeterminado es admin.

`subdir` Subdirectorio de CephFS que se va a montar. Opcional; por defecto /.

`fuse` Acceso a CephFS mediante FUSE en lugar del cliente de kernel. Opcional; por defecto 0.

```
cephfs: cephfs-external
    monhost 10.1.1.20 10.1.1.21 10.1.1.22
    path /mnt/pve/cephfs-external
    content backup
    username admin
    fs-name cephfs
```

NOTA No olvide configurar el archivo con la clave secreta del cliente si cephx no está deshabilitado.

7.15.2 Autenticación

NOTA Si Ceph está instalado localmente en el clúster Proxmox VE, estos pasos se realizan automáticamente al añadir el almacenamiento.

Si utiliza autenticación `cephx`, habilitada por defecto, debe proporcionar el secreto del clúster Ceph externo.

Una forma es copiar el archivo secreto al nodo Proxmox VE:

```
# scp <external cephserver>:/etc/ceph/cephfs.secret /root/cephfs.secret
```

Después configure el almacenamiento CephFS externo con `pvesm` y el parámetro `--keyring`:

```
# pvesm add cephfs <name> --monhost "10.1.1.20 10.1.1.21 10.1.1.22" --content backup --keyring /root/cephfs.secret
```

En la interfaz web, el secreto puede copiarse y pegarse en el campo correspondiente. A diferencia del backend RBD, el secreto contiene únicamente la clave, no una sección `[client.userid]`. Se almacenará en:

```
/etc/pve/priv/ceph/<STORAGE_ID>.secret
```

Como administrador de Ceph puede obtenerse un secreto mediante:

```
# ceph auth get-key client.userid > cephfs.secret
```

7.15.3 Funciones de almacenamiento

El backend CephFS es un sistema de archivos compatible con POSIX sobre un clúster Ceph.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones
vztmpl, iso, backup, snippets	none	sí	sí[1]	no

[1] *Aunque no se conocen fallos, los snapshots todavía no se garantizan como estables porque no han recibido pruebas suficientes.*

7.16 Backend BTRFS

Tipo de pool de almacenamiento: **btrfs**

En apariencia, este tipo de almacenamiento es muy similar al almacenamiento de directorio. La diferencia principal es que, en BTRFS, los discos con formato raw se colocan dentro de un subvolumen para permitir snapshots y migraciones de almacenamiento offline conservando los snapshots.

NOTA BTRFS respeta el indicador O_DIRECT al abrir archivos. Por tanto, las VM no deberían utilizar el modo de caché “none”; de lo contrario pueden producirse errores de checksum.

7.16.1 Configuración

Este backend se configura de forma similar al almacenamiento de directorio. Cuando se añade como almacenamiento BTRFS un directorio que no es el propio punto de montaje, se recomienda especificar el punto de montaje real mediante la opción `is_mountpoint`.

Por ejemplo, si un sistema de archivos BTRFS está montado en `/mnt/data2` y se desea añadir su subdirectorio `pve-storage/` como pool llamado `data2`:

```
btrfs: data2
    path /mnt/data2/pve-storage
    content rootdir,images
    is_mountpoint /mnt/data2
```

7.16.2 Snapshots

Al crear un snapshot de un subvolumen o de un archivo raw, el snapshot se crea como subvolumen de solo lectura con la misma ruta, seguida de `@` y el nombre del snapshot.

7.17 Backend ZFS sobre iSCSI

Tipo de pool de almacenamiento: **zfs**

Este backend accede, mediante SSH, a una máquina remota que dispone de un pool ZFS y de una implementación de destino iSCSI. Para cada disco invitado crea un ZVOL y lo exporta como una LUN iSCSI. Proxmox VE utiliza esa LUN como disco del invitado.

Se admiten las siguientes implementaciones de destino iSCSI:

- LIO (Linux)
- IET (Linux)
- ISTGT (FreeBSD)
- Comstar (Solaris)

NOTA Este plugin necesita un appliance remoto con soporte ZFS. No puede utilizarse para crear un pool ZFS en un appliance de almacenamiento/SAN convencional que no disponga de ZFS.

7.17.1 Configuración

Para utilizar el plugin ZFS sobre iSCSI debe configurarse la máquina remota (destino) para aceptar conexiones SSH desde el nodo Proxmox VE. Proxmox VE se conecta al destino para crear los ZVOL y exportarlos mediante iSCSI. La autenticación se realiza mediante una clave SSH sin contraseña guardada en:

```
/etc/pve/priv/zfs/<target_ip>_id_rsa
```

Los siguientes pasos crean una clave SSH y la distribuyen a la máquina de almacenamiento con IP `192.0.2.1`:

```
mkdir /etc/pve/priv/zfs
ssh-keygen -f /etc/pve/priv/zfs/192.0.2.1_id_rsa
ssh-copy-id -i /etc/pve/priv/zfs/192.0.2.1_id_rsa.pub root@192.0.2.1
ssh -i /etc/pve/priv/zfs/192.0.2.1_id_rsa root@192.0.2.1
```

El backend admite las propiedades comunes `content, nodes, disable` y las siguientes propiedades específicas de ZFS sobre iSCSI:

`pool` Pool/sistema de archivos ZFS del destino iSCSI. Todas las asignaciones se realizan dentro de ese pool.

`portal` Portal iSCSI (IP o nombre DNS con puerto opcional).

`target` Destino iSCSI.

`iscsiprovider` Implementación de destino iSCSI utilizada en la máquina remota.

`comstar_tg` Grupo de destino para vistas Comstar.

`comstar_hg` Grupo de hosts para vistas Comstar.

`lio_tpg` Grupo de portal de destino para destinos Linux LIO.

`nowritecache` Deshabilita la caché de escritura en el destino.

`blocksize` Establece el parámetro de tamaño de bloque ZFS.

`sparse` Utiliza thin provisioning ZFS. Un volumen sparse tiene una reserva distinta del tamaño nominal del volumen.

Ejemplo LIO

```
zfs: lio
    blocksize 4k
    iscsiprovider LIO
    pool tank
    portal 192.0.2.111
    target iqn.2003-01.org.linux-iscsi.lio.x8664:sn.xxxxxxxxxxxx
    content images
    lio_tpg tpg1
    sparse 1
```

Ejemplo Solaris / Comstar

```
zfs: solaris
    blocksize 4k
    target iqn.2010-08.org.illumos:02:xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx:tank1
    pool tank
    iscsiprovider comstar
    portal 192.0.2.112
    content images
```

Ejemplo FreeBSD / ISTGT

```
zfs: freebsd
    blocksize 4k
    target iqn.2007-09.jp.ne.peach.istgt:tank1
    pool tank
    iscsiprovider istgt
    portal 192.0.2.113
    content images
```

Ejemplo IET

```
zfs: iet
    blocksize 4k
    target iqn.2001-04.com.example:tank1
    pool tank
    iscsiprovider iet
    portal 192.0.2.114
    content images
```

7.17.2 Funciones de almacenamiento

El plugin ZFS sobre iSCSI proporciona almacenamiento compartido con capacidad de snapshots. Debe asegurarse de que el appliance ZFS no se convierta en un punto único de fallo de su despliegue.

Tipos de contenido	Formatos de imagen	Compartido	Snapshots	Clones
images	raw	sí	sí	no

Fin de la Parte 7B

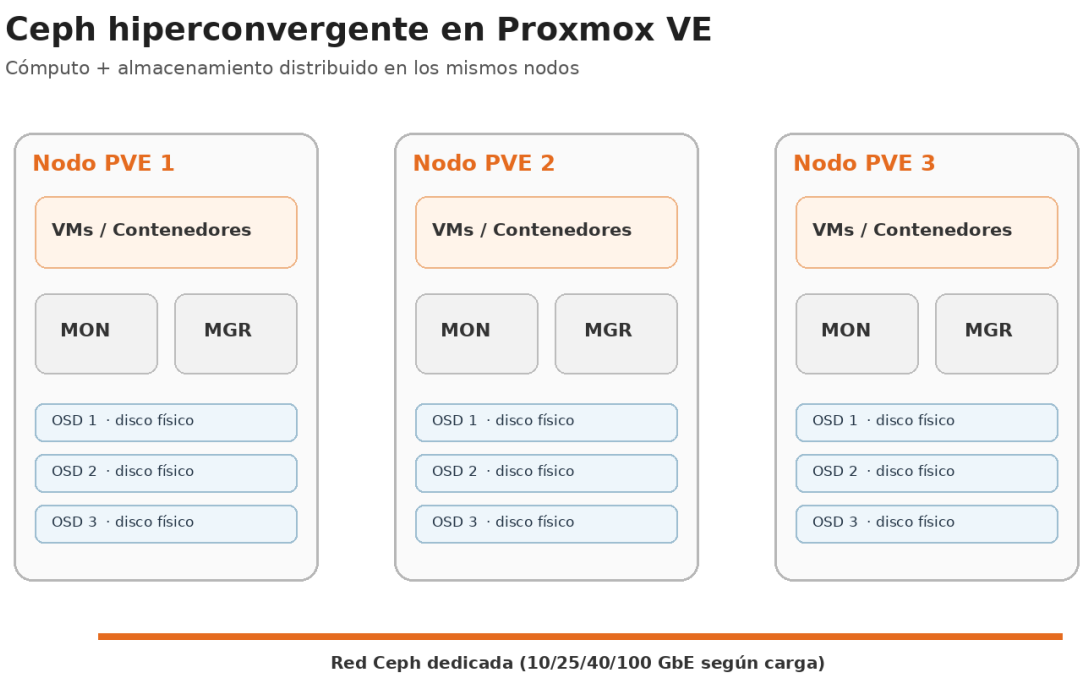
Con esta parte queda completado el Capítulo 7 del manual: “Proxmox VE Storage”.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 8 · Despliegue de un clúster Ceph hiperconvergente



Visión general: Proxmox VE y Ceph compartiendo los mismos nodos.

Basado en "Proxmox VE Administration Guide", Release 9.2.4. Traducción de trabajo al castellano; comandos, rutas y nombres propios se mantienen sin traducir.

NOTA: Esta parte reproduce el contenido técnico del Capítulo 8 del manual fuente. Las referencias de Ceph y los comandos se conservan según el original.

Capítulo 8 · Despliegue de un clúster Ceph hiperconvergente

8.1 Introducción

Proxmox VE unifica los sistemas de cómputo y almacenamiento: los mismos nodos físicos del clúster pueden emplearse tanto para ejecutar máquinas virtuales y contenedores como para proporcionar almacenamiento replicado. Los silos tradicionales de cómputo y almacenamiento pueden integrarse en un único sistema hiperconvergente. Las redes SAN separadas y las conexiones mediante almacenamiento NAS dejan de ser imprescindibles. Gracias a la integración de Ceph, una plataforma de almacenamiento definido por software y de código abierto, Proxmox VE puede ejecutar y administrar el almacenamiento Ceph directamente sobre los nodos del hipervisor.

Ceph es un almacén de objetos distribuido y un sistema de archivos diseñado para ofrecer un alto rendimiento, fiabilidad y escalabilidad.

Algunas ventajas de Ceph en Proxmox VE:

- Configuración y administración sencillas mediante CLI y GUI.
- Thin provisioning.
- Compatibilidad con snapshots.
- Autorreparación (self-healing).
- Escalabilidad hasta el nivel de exabytes.
- Proporciona almacenamiento en bloque, sistema de archivos y objetos.
- Permite crear pools con diferentes características de rendimiento y redundancia.
- Los datos se replican, lo que aporta tolerancia a fallos.
- Puede ejecutarse sobre hardware convencional.
- No necesita controladoras RAID por hardware.
- Es software de código abierto.

En despliegues pequeños y medianos es posible instalar Ceph directamente en los nodos del clúster Proxmox VE y utilizar RADOS Block Devices (RBD) o CephFS. El hardware actual dispone de suficiente CPU y memoria para que los servicios de almacenamiento y los invitados virtuales compartan nodo.

Para simplificar la administración, Proxmox VE integra de forma nativa la instalación y gestión de servicios Ceph tanto desde la interfaz web como mediante la herramienta de línea de comandos pveceph.

8.2 Terminología

Para utilizar Ceph como almacenamiento RBD intervienen principalmente los siguientes demonios:

- Ceph Monitor (ceph-mon o MON).
- Ceph Manager (ceph-mgr o MGR).
- Ceph Metadata Service (ceph-mds o MDS).
- Ceph Object Storage Daemon (ceph-osd u OSD).

CONSEJO: Es muy recomendable familiarizarse con Ceph, su arquitectura y su vocabulario antes de diseñar un entorno de producción.

Servicio	Función principal
MON	Mantiene el mapa maestro del clúster y participa en el quorum.
MGR	Expone supervisión y funciones de gestión; PG autoscaling, telemetría y salud de dispositivos.
MDS	Gestiona metadatos de CephFS.
OSD	Almacena objetos en discos y participa en replicación, recuperación y backfill.

8.3 Recomendaciones para un clúster Ceph saludable

Para construir un clúster hiperconvergente Proxmox + Ceph deben utilizarse al menos tres servidores y, preferiblemente, servidores idénticos.

NOTA: Las recomendaciones siguientes son una orientación aproximada para escoger hardware. Deben adaptarse a cada carga real, probarse y supervisarse de forma continua.

CPU

Los servicios de Ceph pueden agruparse en dos categorías. Los que hacen un uso intensivo de CPU y se benefician de frecuencias base altas y múltiples núcleos son los OSD y los MDS usados por CephFS. Los servicios MON y MGR tienen un consumo de CPU más moderado y no necesitan varios núcleos dedicados en la misma medida.

Como regla sencilla, reserve al menos un núcleo o hilo de CPU por cada servicio Ceph para disponer de los recursos mínimos de un funcionamiento estable. Por ejemplo, un nodo con un monitor, un manager y seis OSD debería reservar aproximadamente ocho núcleos exclusivamente para Ceph si se busca un rendimiento básico y estable.

El consumo de CPU de un OSD depende en gran medida del rendimiento del disco. Cuantos más IOPS pueda sostener el dispositivo, más CPU podrá utilizar el OSD. En SSD empresariales y NVMe capaces de mantener más de 100.000 IOPS con latencia inferior al milisegundo, un único OSD puede llegar a utilizar varios hilos; para discos de muy alto rendimiento son habituales cuatro a seis hilos por OSD.

Memoria

En una arquitectura hiperconvergente hay que planificar y vigilar cuidadosamente el consumo de memoria. Además de la RAM prevista para máquinas virtuales y contenedores, Ceph necesita margen suficiente para mantener un rendimiento estable, especialmente durante recuperación, rebalanceo o backfill.

La recomendación actual del manual es configurar los OSD con al menos 8 GiB de memoria para obtener buen rendimiento. El demonio OSD requiere 4 GiB de forma predeterminada. El consumo total viene determinado sobre todo por el número de OSD, aunque durante operaciones críticas todos los servicios Ceph pueden requerir memoria adicional.

Red

Se recomienda una red de al menos 10 Gbps dedicada exclusivamente a Ceph. En clústeres de tres a cinco nodos también puede utilizarse una topología de red completamente mallada cuando no se disponga de switches de 10 Gbps o superiores.

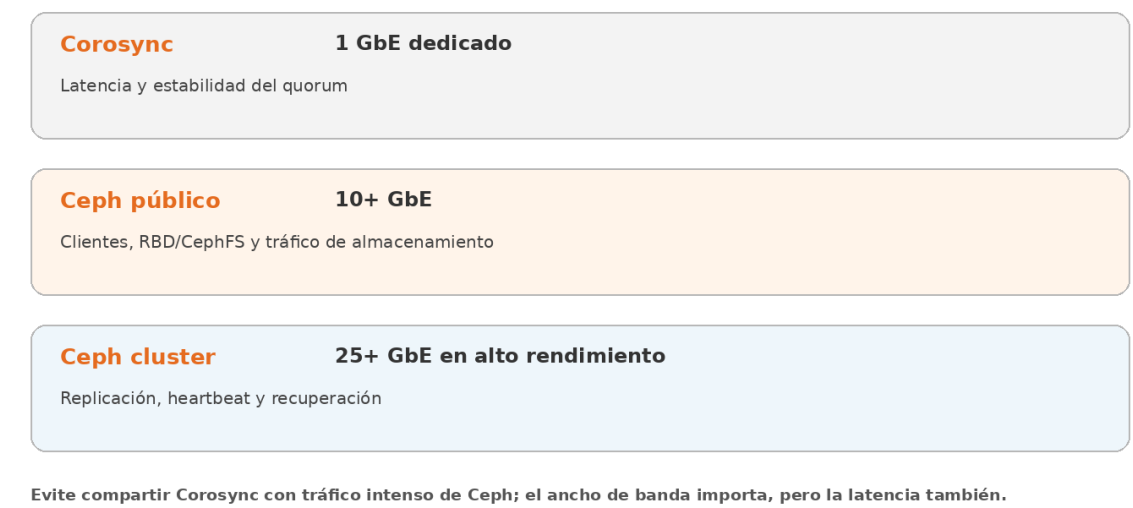
IMPORTANTE: El volumen de tráfico de Ceph, especialmente durante una recuperación, puede interferir con otros servicios. Corosync es sensible a la latencia y podría incluso perder quorum. Separar físicamente las redes de Ceph evita interferencias con Corosync y con el tráfico de los invitados.

Las necesidades de ancho de banda dependen también del rendimiento de los discos. Un único HDD puede no saturar 1 GbE, pero varios OSD basados en HDD por nodo sí pueden saturar 10 GbE. Un NVMe moderno puede saturar por sí solo 10 Gbps o más. Para estos entornos se recomienda al menos 25 Gbps y, si se pretende aprovechar por completo el potencial de los discos, pueden ser necesarios 40 o 100+ Gbps.

En configuraciones de alto rendimiento, si existe duda, el manual recomienda tres redes físicas separadas:

- Una red de muy alto ancho de banda (25+ Gbps) para tráfico interno del clúster Ceph.
- Una red de alto ancho de banda (10+ Gbps) para el tráfico público Ceph entre servidores y clientes. Según las necesidades, también puede alojar tráfico de invitados y migración en vivo.
- Una red de ancho de banda medio (1 Gbps) dedicada en exclusiva a la comunicación de clúster Corosync, sensible a la latencia.

Si hay firewall, deben abrirse los puertos que utiliza Ceph; una forma sencilla es usar una macro de firewall.



Separación de redes recomendada por el manual para entornos de alto rendimiento.

Discos

Al dimensionar el clúster hay que considerar el tiempo de recuperación. En clústeres pequeños, una recuperación larga aumenta la ventana de riesgo ante un segundo fallo; por ello se recomiendan SSD frente a HDD cuando sea viable. Los SSD también ofrecen más IOPS y pueden justificar pools separados por clases de dispositivo.

Otra forma de acelerar OSD es colocar el journal o los dispositivos DB/WAL en almacenamiento más rápido. Si un dispositivo rápido da servicio a varios OSD, debe dimensionarse correctamente para no convertirse en el cuello de botella de todos ellos.

Ceph funciona mejor cuando los discos tienen tamaños homogéneos y están repartidos de forma uniforme entre nodos. Por ejemplo, cuatro discos de 500 GB por nodo son preferibles a una mezcla de un disco de 1 TB y tres de 250 GB. También debe equilibrarse el número de OSD y la capacidad individual: discos mayores aumentan la densidad, pero un fallo obliga a recuperar una cantidad mayor de datos de una sola vez.

Evitar RAID

Ceph ya gestiona la redundancia de objetos y las escrituras paralelas a los OSD. Por ello, una controladora RAID normalmente no mejora la disponibilidad ni el rendimiento. Ceph está pensado para controlar directamente los discos, sin capas de abstracción intermedias; los algoritmos de escritura y caché de una controladora RAID pueden incluso interferir con los de Ceph.

ADVERTENCIA: Evite controladoras RAID. Utilice preferentemente un HBA (Host Bus Adapter).

8.4 Instalación y configuración inicial de Ceph

8.4.1 Uso del asistente web

Proxmox VE incorpora un asistente de instalación de Ceph. Seleccione uno de los nodos del clúster y abra la sección Ceph del árbol de menús. Si Ceph todavía no está instalado, la interfaz ofrecerá instalarlo.

El asistente se divide en varias fases y cada una debe completarse correctamente. Primero se elige la versión de Ceph. En un clúster existente conviene usar la misma versión que el resto de nodos; si es el primer nodo, se recomienda la versión más reciente ofrecida por el entorno.

Al iniciar la instalación, el asistente descarga e instala todos los paquetes necesarios desde el repositorio Ceph de Proxmox VE. Después se crea la configuración inicial. Este paso solo es necesario una vez por clúster, ya que pmxcfs distribuye automáticamente la configuración a todos los nodos.

La configuración inicial incluye:

- Public Network: red utilizada para la comunicación pública de almacenamiento -por ejemplo, VMs con discos RBD o montajes CephFS- y para la comunicación entre servicios Ceph. Es obligatoria.
- Cluster Network: red opcional separada para replicación y heartbeat de OSD. Una red físicamente separada descarga la red pública y la de invitados y puede mejorar significativamente el rendimiento de Ceph.

NOTA: La red de clúster Ceph puede configurarse o trasladarse posteriormente a otra red físicamente separada.

Existen dos opciones avanzadas que solo deberían modificarse si se conocen sus implicaciones:

- Number of replicas: número de copias de cada objeto.
- Minimum replicas: número mínimo de réplicas necesarias para considerar válida una operación de E/S.

También debe seleccionarse el primer nodo monitor. Al finalizar, el asistente muestra una pantalla de éxito con los siguientes pasos. Para comenzar a utilizar Ceph todavía habrá que crear monitores adicionales, OSD y al menos un pool.

8.4.2 Instalación de paquetes Ceph mediante CLI

Como alternativa al asistente web, ejecute en cada nodo:

```
pveceph install
```

El comando configura un repositorio APT en /etc/apt/sources.list.d/ceph.sources e instala el software necesario.

8.4.3 Configuración inicial mediante CLI

En un nodo, use el asistente web o ejecute, por ejemplo:

```
pveceph init --network 10.10.10.0/24
```

Esto crea /etc/pve/ceph.conf con una red dedicada a Ceph. pmxcfs distribuye el archivo automáticamente a todos los nodos. También se crea el enlace simbólico /etc/ceph/ceph.conf, de forma que los comandos Ceph pueden utilizarse sin indicar explícitamente el archivo de configuración.

8.5 Ceph Monitor

El Ceph Monitor (MON) mantiene una copia maestra del mapa del clúster. Para alta disponibilidad se necesitan al menos tres monitores. Si se utilizó el asistente de instalación ya existirá uno. En clústeres pequeños y medianos normalmente no se necesitan más de tres.

8.5.1 Crear monitores

En cada nodo donde quiera ubicar un monitor, use Ceph → Monitor en la GUI o ejecute:

```
pveceph mon create
```

8.5.2 Destruir monitores

En la GUI, seleccione el nodo, abra Ceph → Monitor, seleccione el MON y pulse Destroy. En CLI, conéctese al nodo donde se ejecuta el monitor y ejecute:

```
pveceph mon destroy
```

NOTA: Se necesitan al menos tres monitores para disponer de quorum.

8.6 Ceph Manager

El demonio Manager se ejecuta junto a los monitores y proporciona una interfaz de supervisión del clúster. Desde Ceph Luminous es necesario al menos un ceph-mgr.

8.6.1 Crear Manager

Pueden instalarse varios Managers, aunque solo uno permanece activo en cada momento:

```
pveceph mgr create
```

NOTA: Se recomienda instalar Ceph Manager en los nodos que ejecutan monitores y disponer de más de uno para alta disponibilidad.

8.6.2 Destruir Manager

En la GUI, abra Ceph → Monitor, seleccione el Manager y pulse Destroy. En CLI, desde el nodo donde se ejecuta:

```
pveceph mgr destroy
```

NOTA: Aunque MGR no es una dependencia dura para que Ceph almacene datos, es crucial porque gestiona funciones como PG autoscaling, monitorización de salud de dispositivos, telemetría y otras.

8.7 Ceph OSD

Los Ceph Object Storage Daemons almacenan objetos de Ceph a través de la red. Se recomienda utilizar un OSD por disco físico.

8.7.1 Crear OSD

Puede crearse un OSD desde la interfaz web o mediante pveceph:

```
pveceph osd create /dev/sd[X]
```

CONSEJO: El manual recomienda al menos tres nodos y, como referencia, un mínimo de 12 OSD repartidos uniformemente entre ellos.

Si el disco se utilizó anteriormente -por ejemplo con ZFS o como OSD- hay que eliminar las huellas previas. Para borrar tabla de particiones, sector de arranque y restos de OSD:

```
ceph-volume lvm zap /dev/sd[X] --destroy
```

ADVERTENCIA: El comando anterior destruye todos los datos del disco.

Ceph BlueStore

BlueStore se introdujo a partir de Ceph Kraken y es el tipo de almacenamiento OSD predeterminado desde Ceph Luminous. La creación normal sigue siendo:

```
pveceph osd create /dev/sd[X]
```

block.db y block.wal

Si se desea usar un dispositivo DB/WAL separado, pueden indicarse -db_dev y -wal_dev. Si WAL no se especifica por separado, se ubica junto a DB:

```
pveceph osd create /dev/sd[X] -db_dev /dev/sd[Y] -wal_dev /dev/sd[Z]
```

Los tamaños pueden fijarse con -db_size y -wal_size. Si no se especifican, se toman, por este orden, los valores bluestore_block_{db,wal}_size de la configuración Ceph -secciones osd o global, en la base de configuración o archivo- y, en último término, el 10 % del tamaño del OSD para DB y el 1 % para WAL.

NOTA: DB almacena metadatos internos de BlueStore y WAL es su journal o write-ahead log. Para mejorar el rendimiento se recomienda un SSD rápido o NVRAM.

Ceph FileStore

Antes de Ceph Luminous, FileStore era el tipo predeterminado. Desde Ceph Nautilus Proxmox VE ya no permite crear este tipo de OSD mediante pveceph. Si todavía fuera necesario, debe utilizarse ceph-volume directamente:

```
ceph-volume lvm create --filestore --data /dev/sd[X] --journal /dev/sd[Y]
```

8.7.2 Destruir OSD

Si un OSD o su disco presenta problemas, primero debe diagnosticarse para determinar si realmente necesita sustitución. Para destruirlo desde <Nodo> → Ceph → OSD o mediante CLI, siga este proceso:

1. Asegúrese de que el clúster dispone de espacio suficiente. Si el OSD sigue en estado up e in, compruebe que el valor Used (%) de todos los OSD se mantiene claramente por debajo de nearfull_ratio (85 % de forma predeterminada). En CLI puede usar ceph osd df tree.

2. Si el OSD todavía no está marcado como out, márquelo como Out para excluirlo de la distribución de datos e iniciar el rebalanceo. En CLI: ceph osd out <id>.

3. Si es posible, espere a que Ceph termine el rebalanceo y el OSD quede vacío; cuando lo esté, mostrará 0 PG.

4. Detenga el OSD. Si Proxmox VE avisa de que todavía no es seguro hacerlo, cancele y vuelva a intentarlo unos minutos después. Puede comprobarlo con ceph osd ok-to-stop <id> y detenerlo con pveceph stop --service osd.<id>.

5. Finalmente, elimine el OSD de Ceph y borre sus datos. En la GUI use More → Destroy y, si desea reutilizar inmediatamente el disco, active la opción de limpieza. En CLI: pveceph osd destroy <id> [--cleanup].

```
ceph osd df tree
ceph osd out <id>
ceph osd ok-to-stop <id>
pveceph stop --service osd.<id>
pveceph osd destroy <id> [--cleanup]
```

8.8 Configuración de Ceph

Los demonios y clientes Ceph obtienen su configuración de una o varias fuentes. En Proxmox VE, ceph.conf se mantiene deliberadamente mínimo y contiene principalmente parámetros de arranque. La mayor parte de la configuración reside en la base de configuración central mantenida por los servicios MON.

Para cambiar valores de forma persistente utilice:

Comando	Función
ceph config dump	Muestra todas las opciones de configuración.
ceph config get <who> <name>	Muestra una opción para una entidad.
ceph config set <who> <name> <value>	Establece una opción para una o varias entidades.
ceph config rm <who> <name>	Elimina una opción para una o varias entidades.

Ejemplo para aumentar osd_memory_target de todos los OSD:

```
ceph config set osd osd_memory_target 8G
```

NOTA: Si una opción existe en un archivo de configuración local, prevalece sobre el valor de la base de datos MON, ya que la configuración MON tiene menor precedencia.

8.9 Pools de Ceph

Un pool es un grupo lógico para almacenar objetos. Contiene una colección de objetos organizada en Placement Groups (PG, pg_num).

8.9.1 Crear y editar pools

Los pools pueden crearse y editarse desde la línea de comandos o desde Ceph → Pools en cualquier nodo Proxmox VE. Si no se especifican opciones, Proxmox VE utiliza por defecto 128 PG, size=3 y min_size=2 para reducir el riesgo de pérdida de datos ante el fallo de un OSD.

ADVERTENCIA: No configure min_size=1. Un pool replicado con min_size 1 permite E/S cuando solo existe una réplica, lo que puede provocar pérdida de datos, PG incompletos u objetos no localizados.

Se aconseja activar PG Autoscaler o calcular el número de PG según el entorno. Desde Ceph Nautilus el número de PG puede modificarse tras la creación. El autoscaler puede ajustar automáticamente el número de PG en segundo plano. Los parámetros avanzados Target Size y Target Ratio ayudan al autoscaler a tomar mejores decisiones.

```
pveceph pool create <pool-name> --add_storages
```

CONSEJO: Para crear automáticamente una entrada de almacenamiento asociada, mantenga marcada "Add as Storage" en la GUI o use --add_storages al crear el pool.

Opciones de pool:

Opción	Descripción
Name	Nombre único del pool; no puede cambiarse después.
Size	Número de réplicas por objeto. Valor predeterminado: 3.
PG Autoscale Mode	Modo del autoescalado de PG. warn avisa si el número de PG no es óptimo. Predeterminado: warn.
Add as Storage	Crea almacenamiento VM/CT basado en el pool. Predeterminado: true al crear.
Min. Size	Número mínimo de réplicas necesario para aceptar E/S. Predeterminado: 2.
Crush Rule	Regla de colocación de objetos en el clúster.
# of PGs	Número inicial de Placement Groups. Predeterminado: 128.
Target Ratio	Proporción de datos esperada en relación con otros pools.
Target Size	Cantidad estimada de datos esperada en el pool.
Min. # of PGs	Límite inferior para que el autoscaler no fusione PG por debajo de ese valor.

8.9.2 Pools Erasure Coded

Erasure Coding (EC) es una forma de corrección anticipada de errores que permite recuperarse de una determinada pérdida de datos. Puede ofrecer más capacidad útil que un pool replicado, a cambio de menor rendimiento.

En un pool replicado se guardan varias réplicas completas. En un pool EC, los datos se dividen en k fragmentos de datos y se añaden m fragmentos de codificación. Estos fragmentos permiten reconstruir información ausente. El parámetro m define cuántos OSD pueden perderse sin perder datos y el total de fragmentos es k + m.

El min_size predeterminado de un pool EC depende de m. Si m=1, min_size será k. Si m>1, min_size será k+1. La documentación Ceph recomienda un valor conservador k+2. Si hay menos OSD disponibles que min_size, se bloquea la E/S del pool hasta recuperar suficientes OSD.

NOTA: Al diseñar un pool EC, vigile min_size: determina cuántos OSD deben permanecer disponibles para que continúe la E/S.

Ejemplo: con k=2 y m=1 se obtiene size=3 y min_size=2, por lo que el pool sigue operativo con un OSD averiado. Con k=2 y m=2, size=4 y min_size=3, y sigue operativo con la pérdida de un OSD.

```
pveceph pool create <pool-name> --erasure-coding k=2,m=1
```

Los parámetros opcionales son failure-domain y device-class. Si se necesita cambiar el perfil EC utilizado por el pool, deberá crearse un pool nuevo con un perfil nuevo. Proxmox VE creará el pool EC de datos y un pool replicado para omap y metadatos: <pool-name>-data y <pool-name>-metadata. De forma predeterminada también crea la configuración de almacenamiento; puede desactivarse con --add_storages 0. En una configuración manual debe indicarse data-pool para que el pool EC almacene los objetos de datos.

NOTA: Los parámetros --size, --min_size y --crush_rule afectan al pool replicado de metadatos, no al pool EC de datos. min_size del pool de datos puede modificarse después; size y crush_rule no pueden cambiarse en un pool EC.

Si necesita personalizar más el perfil EC, créelo directamente con las herramientas Ceph y páselo mediante profile:

```
pveceph pool create <pool-name> --erasure-coding profile=<profile-name>
```

FastEC (allow_ec_optimizations)

Ceph Tentacle incorpora una nueva ruta de E/S para erasure coding llamada FastEC. Se activa por pool con allow_ec_optimizations y añade lecturas y escrituras parciales en pools EC. Esto mejora significativamente el rendimiento de E/S aleatoria pequeña, patrón muy habitual en discos de máquinas virtuales. Upstream lo recomienda para almacenamiento basado en RBD y CephFS.

El monitor de Ceph comprueba cuatro requisitos antes de activarlo:

- El pool debe ser erasure coded.
- El clúster debe estar en require_osd_release tentacle o posterior.
- El perfil de erasure code debe usar una combinación de plugin y técnica compatible con la ruta optimizada.
- stripe_unit debe ser múltiplo de 4096 bytes. El valor predeterminado ya cumple este requisito.

Plugin	Técnica	FastEC
isa	reed_sol_van	sí
isa	cauchy	no
jerasure	reed_sol_van	sí
jerasure	resto	no
lrc	cualquiera	no
shec	cualquiera	no
clay	cualquiera	no

pveceph no fija explícitamente plugin o technique al crear el perfil EC; se heredan de osd_pool_default_erasure_code_profile del monitor, cuyo valor predeterminado en Squid y Tentacle es reed_sol_van. Puede comprobarlo con:

```
ceph config get mon osd_pool_default_erasure_code_profile
```

Compruebe la versión mínima del clúster con:

```
ceph osd dump | grep require_osd_release
```

Si informa de una versión anterior a tentacle, después de actualizar todos los OSD puede ejecutar:

```
ceph osd require-osd-release tentacle
```

IMPORTANTE: allow_ec_optimizations es un cambio de una sola dirección. Una vez activado en un pool, el monitor de Ceph no permite desactivarlo. Volver atrás exige vaciar y recrear el pool. Valídelo primero con una carga de prueba.

Para activar FastEC sobre el pool de datos:

```
ceph osd pool set <pool-name>-data allow_ec_optimizations 1
ceph osd pool get <pool-name>-data allow_ec_optimizations
```

Al activarlo en un pool con datos existentes no se migran ni se recodifican datos; solo cambia metadato del pool. Sin embargo , puede haber reelección de primarios y re-peering de PG, con una breve pausa de E/S por PG afectado. Además, cualquier scrub en curso sobre esos PG se cancela y deberá repetirse después.

Añadir pools EC como almacenamiento

Un pool EC ya existente puede añadirse a Proxmox VE igual que un pool RBD, pero es necesario indicar data-pool:

```
pvesm add rbd <storage-name> --pool <replicated-pool> --data-pool <ec-pool>
```

CONSEJO: En clústeres Ceph externos no gestionados por el Proxmox VE local, recuerde añadir keyring y monhost.

8.9.3 Destruir pools

En la GUI, seleccione un nodo, abra Ceph → Pools, elija el pool y pulse Destroy; para confirmar debe escribir el nombre del p ool. En CLI:

```
pveceph pool destroy <name>
```

Puede especificarse -remove_storages para eliminar también el almacenamiento asociado. La eliminación se realiza en segundo plano y puede tardar; durante el proceso irá reduciéndose el uso de datos del clúster.

8.9.4 PG Autoscaler

PG Autoscaler permite que el clúster tenga en cuenta la cantidad real o esperada de datos de cada pool y elija automáticament e los valores pg_num adecuados. Está disponible desde Ceph Nautilus.

Si fuera necesario, active el módulo:

```
ceph mgr module enable pg_autoscaler
```

El modo se configura por pool:

Modo	Comportamiento
warn	Genera una advertencia si pg_num se aleja demasiado del valor sugerido.
on	Ajusta pg_num automáticamente sin intervención manual.
off	No ajusta pg_num ni genera advertencias por un número no óptimo.

El factor de escalado puede afinarse con target_size, target_size_ratio y pg_num_min.

ADVERTENCIA: De forma predeterminada, el autoscaler considera modificar el número de PG si difiere por un factor de 3. Esto puede provocar un movimiento considerable de datos y una carga elevada sobre el clúster.

8.10 Ceph CRUSH y clases de dispositivo

El algoritmo CRUSH (Controlled Replication Under Scalable Hashing) es uno de los fundamentos de Ceph. Calcula dónde almacenar y de dónde recuperar los datos sin necesidad de un servicio central de indexación. Trabaja con un mapa de OSD, buckets -ubicaciones de dispositivos- y rulesets que definen la replicación de cada pool.

El mapa puede modificarse para reflejar jerarquías distintas de replicación y separar las réplicas por dominios de fallo. Es habitual usar clases distintas de discos para pools distintos. Ceph introdujo las device classes para facilitar la generación de ru lesets en estos casos.

Puede ver las clases en ceph osd tree y sus raíces shadow con:

```
ceph osd crush tree --show-shadow
```

Para restringir un pool a una clase de dispositivo, primero cree una regla:

```
ceph osd crush rule create-replicated <rule-name> <root> <failure-domain> <class>
```

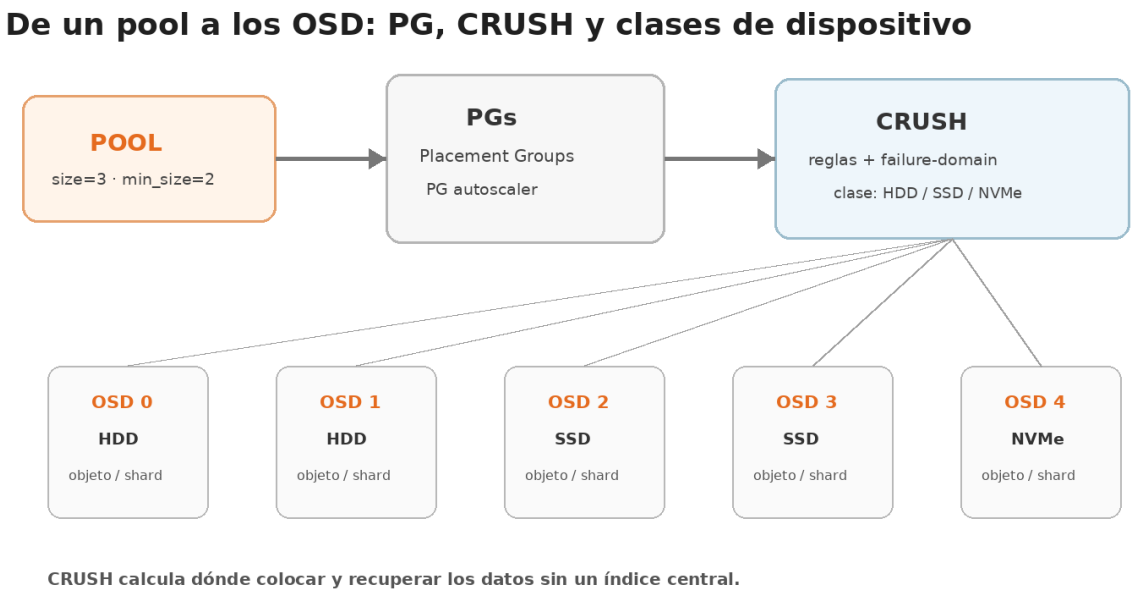
Parámetro	Significado
<rule-name>	Nombre de la regla, que se asociará posteriormente a un pool.
<root>	Raíz CRUSH; normalmente default.
<failure-domain>	Nivel al que deben distribuirse los objetos; normalmente host.
<class>	Clase de almacenamiento: nvme, ssd, hdd, etc.

Después, asigne la regla al pool:

```
ceph osd pool set <pool-name> crush_rule <rule-name>
```

CONSEJO: Si el pool ya contiene objetos, deberán recolocarse. Esto puede tener un impacto importante de rendimiento. Como alternativa, puede crearse un pool nuevo y migrar los discos de forma gradual.

ADVERTENCIA: Con PG Autoscaler, todos los pools deben usar reglas CRUSH del mismo tipo: o todos especifican una device class o ninguno. De lo contrario, el autoscaler omitirá los pools afectados.



Flujo conceptual: pool → PG → reglas CRUSH → OSD y clases de dispositivo.

8.11 Cliente Ceph

Tras completar la configuración anterior, Proxmox VE puede usar los pools para almacenar imágenes de VM y contenedores. Basta con añadir un nuevo almacenamiento RBD desde la GUI.

En un clúster Ceph externo hay que copiar el keyring a una ubicación predefinida. Cuando Ceph está instalado en los propios nodos Proxmox VE, este paso se realiza automáticamente.

NOTA: El nombre del archivo debe ser <storage_id>.keyring, donde <storage_id> es el identificador que aparece después de rbd: en /etc/pve/storage.cfg.

```
mkdir /etc/pve/priv/ceph
cp /etc/ceph/ceph.client.admin.keyring /etc/pve/priv/ceph/my-ceph-storage.keyring
```

8.12 CephFS

Ceph ofrece también un sistema de archivos construido sobre el mismo almacenamiento de objetos usado por RADOS Block Devices. Un Metadata Server (MDS) mapea los objetos RADOS a archivos y directorios, permitiendo ofrecer un sistema de archivo s POSIX, replicado, compartido y altamente disponible. Los MDS distribuyen metadatos de forma que una carga elevada no recaiga en un único host, problema frecuente en soluciones compartidas tradicionales como NFS.

Proxmox VE permite crear un CephFS hiperconvergente y también utilizar un CephFS existente como almacenamiento para backups, ISO y plantillas de contenedores.

8.12.1 Metadata Server (MDS)

CephFS necesita al menos un MDS en ejecución. Puede crearlo desde Node → CephFS o con:

```
pveceph mds create
```

Pueden crearse varios MDS, aunque con la configuración predeterminada solo uno está activo. Si el activo o su nodo falla, un MDS standby será promovido. Para acelerar la transición puede usar la opción hotstandby al crearlo o añadir en la sección MDS correspondiente de /etc/pve/ceph.conf:

```
mds standby replay = true
```

NOTA: El sondeo activo de standby replay mantiene el MDS de reserva más caliente, pero añade carga al sistema y al MDS activo.

Desde Ceph Luminous es posible ejecutar varios MDS activos simultáneamente. Normalmente solo resulta útil cuando hay muchos clientes en paralelo, ya que el MDS rara vez es el cuello de botella de un entorno convencional.

8.12.2 Crear CephFS

Requisitos para crear un CephFS correctamente:

- Tener instalados los paquetes Ceph. Si la instalación es antigua, conviene repetir la instalación sobre un sistema actualizad o para asegurar la presencia de paquetes relacionados con CephFS.
- Tener configurados los monitores.
- Tener configurados los OSD.
- Tener al menos un MDS.

Después puede crearse desde Node → CephFS o con:

```
pveceph fs create --pg_num 128 --add-storage
```

Este ejemplo crea un CephFS llamado cephfs. Usa un pool de datos cephfs_data con 128 PG y un pool de metadatos cephfs_metadata con una cuarta parte de los PG del pool de datos (32). --add-storage añade automáticamente CephFS a la configuración de almacenamiento de Proxmox VE.

8.12.3 Destruir CephFS

ADVERTENCIA: Destruir un CephFS deja inutilizables todos sus datos. No puede deshacerse.

Para eliminar un CephFS de forma completa y ordenada:

1. Desconecte todos los clientes no Proxmox VE, por ejemplo desmontando CephFS en los invitados.

2. Deshabilite las entradas de almacenamiento Proxmox VE relacionadas para evitar que se monten automáticamente.

3. Elimine de los invitados los recursos que estén alojados en ese CephFS, por ejemplo ISO.

4. Desmonte manualmente los almacenamientos CephFS en todos los nodos.

5. Detenga o destruya todos los MDS de ese CephFS. Recuerde que los standby pueden promocionarse automáticamente.

6. Destruya CephFS con pveceph fs destroy, opcionalmente eliminando storages y pools asociados.

```
umount /mnt/pve/<STORAGE-NAME>
pveceph stop --service mds.NAME
pveceph mds destroy NAME
pveceph fs destroy NAME --remove-storages --remove-pools
```

8.13 Mantenimiento de Ceph

8.13.1 Sustituir OSD

Para sustituir el disco de un OSD, una de las tareas de mantenimiento más habituales:

1. Si el disco ha fallado, obtenga un reemplazo recomendado del mismo tipo y tamaño.

2. Destruya el OSD afectado.

3. Desconecte el disco antiguo del servidor y conecte el nuevo.

4. Cree de nuevo el OSD.

5. Tras el rebalanceo automático, el estado del clúster debería volver a HEALTH_OK. Los crashes que aún figuren pueden archivarse con ceph crash archive-all.

```
ceph crash archive-all
```

8.13.2 Trim / Discard

Es una buena práctica ejecutar fstrim (discard) de forma periódica en máquinas virtuales y contenedores. Esto libera bloques que el sistema de archivos ya no utiliza y reduce el consumo de datos y la carga de recursos. Muchos sistemas operativos modernos ya envían comandos discard regularmente. En las VM debe habilitarse la opción discard del disco virtual.

8.13.3 Scrub y Deep Scrub

Ceph mantiene la integridad de los datos mediante scrubbing de los Placement Groups. Comprueba los objetos de cada PG. Existen dos niveles: comprobaciones diarias de metadatos, más ligeras, y deep scrub semanal, que lee los objetos y verifica checksums. Si un scrub afecta al rendimiento en horario productivo, puede ajustarse la ventana de ejecución.

8.13.4 Apagar un clúster HCI Proxmox VE + Ceph

Para apagar todo el clúster, primero detenga todos los clientes Ceph, principalmente máquinas virtuales y contenedores. Deten ga también otros clientes que puedan usar CephFS o un RADOS Gateway. Los invitados HA pasarán al estado stopped cuando se apaguen mediante las herramientas de Proxmox VE.

Cuando ningún cliente acceda a Ceph, compruebe que el clúster está saludable:

```
ceph -s
```

Para evitar procesos de recuperación durante el apagado y encendido, active la bandera OSD noout desde Ceph → OSD → Manage Global Flags o con:

```
ceph osd set noout
```

Apague primero los nodos que no ejecuten MON. Después apague los nodos con monitores. En el encendido haga lo contrario: arranque primero los nodos con MON y, cuando todos los nodos y servicios estén activos, retire noout:

```
ceph osd unset noout
```

Después pueden iniciarse los invitados. Los invitados HA cambiarán a started cuando arranquen.

8.13.5 Cambios de red

Es posible cambiar las redes usadas por Ceph en un entorno HCI sin interrupción siempre que la red antigua y la nueva puedan coexistir durante la transición. El procedimiento cambia según se trate de la red pública o la red de clúster Ceph.

IMPORTANTE: El cambio de redes Ceph debe realizarse con mucho cuidado. Un error puede dejar el clúster inoperativo y requerir una intervención con parada. El manual recomienda ensayar el procedimiento antes en un clúster de pruebas.

Una vez configurada la nueva red en todos los hosts, verifique la conectividad antes de continuar. Si utiliza MTU grande, compruébela también. Para MTU 9000, el tamaño del payload de ping recomendado es 8972 bytes en IPv4 y 8952 bytes en IPv6, teniendo en cuenta las cabeceras IP e ICMP.

```
ping -M do -s 8972 {target IP}
```

Al cambiar entre IPv4 e IPv6, revise que ceph.conf contenga los valores correctos:

```
ms_bind_ipv4 = true
ms_bind_ipv6 = false
```

Cambiar la red pública de Ceph

La red pública es el canal principal entre servicios Ceph y clientes. Cambiarla exige más pasos que la red de clúster porque los MON conservan internamente un monmap con las direcciones de los monitores.

Procedimiento general:

1. Cambie public_network en /etc/pve/ceph.conf sin modificar otros valores.
2. Reinicie en un host los servicios que no son MON: OSD, MGR y MDS.
3. Espere a que Ceph vuelva a HEALTH_OK.
4. Compruebe que los servicios escuchan en la nueva red.
5. Repita el reinicio de servicios host por host.
6. Destruya un MON.
7. Créelo de nuevo.
8. Espere de nuevo HEALTH_OK.
9. Repita el proceso de destrucción y recreación con el resto de MON.

```
public_network = 10.9.9.30/24
```

ADVERTENCIA: No modifique manualmente mon_host ni las secciones [mon.HOSTNAME]. Se actualizarán automáticamente al destruir y recrear los MON.

Reinicie los servicios no MON, un nodo cada vez:

```
systemctl restart ceph-osd.target
systemctl restart ceph-mgr.target
systemctl restart ceph-mds.target
```

NOTA: Solo habrá demonios MDS si se usa CephFS. Tras reiniciar el primer OSD, la GUI puede avisar temporalmente de que el OSD no es alcanzable hasta que MGR se reinicie; las VM pueden seguir accediendo.

ADVERTENCIA: No reinicie OSD de varios hosts a la vez. Podrían quedar por debajo de min_size algunas PG y detenerse la E/S hasta recuperar suficientes réplicas.

Compruebe las escuchas con:

```
ss -tulpn | grep ceph
```

Una vez todos los servicios no MON estén en la nueva red y el clúster esté HEALTH_OK, destruya y recree cada MON de uno en uno. Verifique al final que mon_host y las secciones [mon.HOSTNAME] ya apuntan a la nueva red y que la red antigua ha dejado de utilizarse.

Cambiar la red de clúster Ceph

La red de clúster se usa para la replicación entre OSD. El proceso es más sencillo:

1. Cambie cluster_network en /etc/pve/ceph.conf.
2. Reinicie los OSD de un único host.
3. Espere a HEALTH_OK.
4. Verifique que los OSD utilizan la nueva red.
5. Repita host por host.

```
cluster_network = 10.9.9.30/24
systemctl restart ceph-osd.target
ss -tulpn | grep ceph-osd
```

ADVERTENCIA: No reinicie OSD en varios hosts simultáneamente. Se aplican las mismas consideraciones de min_size y disponibilidad de PG.

8.14 Supervisión y resolución de problemas de Ceph

Es importante supervisar la salud del despliegue desde el principio, utilizando las herramientas Ceph o consultando el estado a través de la API de Proxmox VE. Los estados habituales son HEALTH_OK, HEALTH_WARN y HEALTH_ERR. Cuando el clúster no está sano, los comandos de estado ayudan a conocer los eventos actuales y las acciones sugeridas.

Para supervisar continuamente el estado:

```
watch ceph --status
```

Para imprimir el estado una vez y seguir añadiendo eventos:

```
ceph --watch
```

8.14.1 Resolución de problemas

Fuentes de información relevantes en el nodo afectado:

- Supervisión de salud de discos.
- System → System Log o journalctl desde la CLI.
- Registros IPMI y, si existe, de la controladora RAID.
- Archivos de log específicos de servicios Ceph en /var/log/ceph/.

```
journalctl --since "2 days ago"
ceph crash ls
ceph crash info <crash_id>
ceph crash archive-all
```

Los crashes marcados como nuevos pueden archivarse con ceph crash archive-all. Si hace falta más detalle, puede aumentarse el nivel de log de Ceph.

Causas comunes de problemas Ceph:

- Problemas de red: congestión, switch defectuoso, interfaz apagada o firewall bloqueando tráfico. Debe comprobarse la conectividad fiable entre nodos en Corosync y en las redes pública y cluster de Ceph.
- Discos o conexiones defectuosos, mal asentados o con rendimiento de E/S insuficiente bajo carga. Esto es más frecuente con HDD, hardware de consumo o controladoras RAID desaconsejadas.
- No cumplir las recomendaciones de diseño de un clúster Ceph saludable.

Problema habitual: OSD down/crashed. Un OSD defectuoso se reportará como down y, normalmente, pasará automáticamente a out aproximadamente diez minutos después. Según la causa, también puede recuperar automáticamente el estado up/in. Para una activación manual desde la GUI use Any node → Ceph → OSD, seleccione el OSD y pulse Start, In y Reload. En CLI:

```
ceph-volume lvm activate --all
```

Para recuperar un OSD fallido puede ser necesario reiniciar de forma segura el nodo correspondiente o, como último recurso, recrear o sustituir el OSD.

Referencias y licencia de esta traducción

Esta traducción corresponde al Capítulo 8 del “Proxmox VE Administration Guide” 9.2.4 de Proxmox Server Solutions GmbH. El manual original se distribuye bajo GNU Free Documentation License 1.3 o posterior. Los nombres de productos, comandos, rutas, valores de ejemplo y términos técnicos se han mantenido para preservar su utilidad operativa.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 9

Capítulo 9 · Replicación de almacenamiento

Basado en "Proxmox VE Administration Guide", Release 9.2.4

Traducción técnica al castellano · Comandos y rutas conservados

Capítulo 9 · Replicación de almacenamiento

La herramienta de línea de comandos pvesr administra el sistema de replicación de almacenamiento de Proxmox VE. La replicación de almacenamiento aporta redundancia a los invitados que utilizan almacenamiento local y reduce el tiempo necesario para realizar migraciones.

Los volúmenes de un invitado se replican a otro nodo para que los datos estén disponibles sin necesidad de utilizar almacenamiento compartido. La replicación utiliza snapshots para reducir el tráfico de red: después de la primera sincronización completa, solo se envían de forma incremental los datos nuevos o modificados. Si un nodo falla, los datos del invitado continúan disponibles en el nodo que contiene la réplica.

La replicación se ejecuta automáticamente a intervalos configurables. El intervalo mínimo es de un minuto y el máximo de una vez por semana. El formato empleado para definir estos intervalos es un subconjunto de los eventos de calendario de systemd; consulte la sección "Formato de programación".

Es posible replicar un invitado hacia varios nodos de destino, pero no crear dos trabajos de replicación hacia el mismo nodo de destino.

También es posible limitar el ancho de banda de cada replicación para evitar sobrecargar el almacenamiento o el servidor.

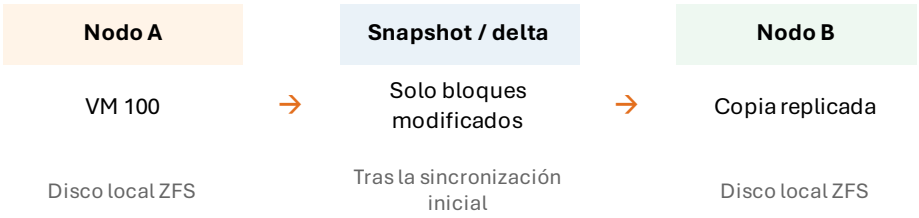
Si el invitado se migra a un nodo que ya dispone de una réplica, solo es necesario transferir los cambios producidos desde la última replicación —los denominados deltas—. Esto reduce significativamente el tiempo de migración. Cuando se migra un invitado al nodo de destino de su réplica, la dirección de la replicación se invierte automáticamente.

EJEMPLO: La VM 100 se ejecuta inicialmente en nodeA y se replica en nodeB. Si se migra a nodeB, a partir de ese momento se replica automáticamente desde nodeB hacia nodeA.

Si se migra a un nodo en el que el invitado todavía no está replicado, es necesario transferir todos los datos de sus discos. Una vez finalizada la migración, el trabajo de replicación continúa replicando el invitado hacia los nodos configurados.

IMPORTANTE: La alta disponibilidad (HA) puede utilizarse conjuntamente con la replicación de almacenamiento, pero puede producirse pérdida de los cambios realizados entre la última sincronización completada y el instante en que falla el nodo.

Cómo funciona la replicación incremental



La migración hacia un nodo que ya contiene una réplica solo necesita transferir los cambios pendientes, reduciendo de forma significativa el tiempo de migración.

9.1 Tipos de almacenamiento compatibles

Descripción	Tipo de plugin	Snapshots	Estable
ZFS (local)	zfspool	sí	sí

9.2 Formato de programación

La replicación utiliza eventos de calendario para configurar la programación de los trabajos.

9.3 Tratamiento de errores

Si un trabajo de replicación encuentra un problema, pasa al estado de error. Mientras permanece en este estado, los intervalos de replicación configurados se suspenden temporalmente. La replicación fallida vuelve a intentarse cada 30 minutos. Cuando uno de estos intentos finaliza correctamente, se reactiva la programación original.

9.3.1 Posibles problemas

Algunos de los problemas más habituales se indican a continuación. Dependiendo de la configuración, la causa también puede ser diferente.

- La red no funciona correctamente.
- No queda espacio libre en el almacenamiento de destino de la replicación.
- En el nodo de destino no está disponible un almacenamiento con el mismo identificador de almacenamiento (Storage ID).

NOTA: El registro del trabajo de replicación permite determinar la causa concreta del problema.

9.3.2 Migrar un invitado cuando existe un error

En caso de un error grave, un invitado virtual puede quedar asociado a un nodo averiado. En ese caso es necesario trasladarlo manualmente a un nodo operativo.

9.3.3 Ejemplo

Supongamos que existen dos invitados —VM 100 y CT 200— ejecutándose en el nodo A y replicados hacia el nodo B. El nodo A ha fallado y no puede volver a ponerse en línea. Los invitados deberán trasladarse manualmente al nodo B.

- Conéctese al nodo B mediante SSH o abra su shell desde la interfaz web.
- Compruebe que el clúster dispone de quorum.

```
# pvecm status
```

Si no hay quorum, se recomienda encarecidamente resolver primero este problema y volver a dejar operativo el clúster. Solo cuando esto no sea posible temporalmente puede forzarse el quorum del nodo actual mediante:

```
# pvecm expected 1
```

ADVERTENCIA: Evite por completo cualquier cambio que afecte al clúster mientras el número de votos esperados esté modificado (por ejemplo, añadir o eliminar nodos, almacenamientos o invitados virtuales). Utilice esta medida únicamente para volver a poner en marcha invitados esenciales o para resolver el propio problema de quorum.

Mueva los archivos de configuración de ambos invitados desde el nodo A al nodo B:

```
# mv /etc/pve/nodes/A/qemu-server/100.conf /etc/pve/nodes/B/qemu-server/100.conf
# mv /etc/pve/nodes/A/lxc/200.conf /etc/pve/nodes/B/lxc/200.conf
```

A continuación, inicie de nuevo los invitados:

```
# qm start 100
# pct start 200
```

Sustituya los VMID y los nombres de los nodos por los valores correspondientes a su entorno.

9.4 Gestión de trabajos

La interfaz web permite crear, modificar y eliminar fácilmente trabajos de replicación. También puede utilizarse la herramienta de línea de comandos pvesr.

El panel de replicación está disponible en los niveles Datacenter, nodo e invitado virtual. La diferencia entre ellos es el conjunto de trabajos que muestran: todos los trabajos, los específicos de un nodo o los específicos de un invitado.

Al crear un nuevo trabajo debe indicarse el invitado —si no estaba ya seleccionado— y el nodo de destino. Si no se desea utilizar el valor predeterminado de cada 15 minutos, puede definirse una programación específica. También puede establecerse un límite de transferencia para el trabajo, lo que ayuda a mantener la carga del almacenamiento dentro de valores aceptables.

Cada trabajo de replicación se identifica mediante un ID único en todo el clúster. Este identificador está compuesto por el VMID seguido de un número de trabajo. Solo es necesario especificarlo manualmente cuando se utiliza la herramienta CLI.

9.5 Red

De forma predeterminada, el tráfico de replicación utiliza la misma red que el tráfico de migración de invitados. Si no se ha configurado una red específica de migración, se utilizará a su vez la red de administración.

Para emplear una red diferente para la replicación, configure “Replication Network” en la interfaz web, en Datacenter → Opciones → Replication Settings, o defina la propiedad replication en el archivo datacenter.cfg:

```
# use dedicated replication network
replication: secure,network=10.1.2.0/24
```

9.6 Ejemplos de la interfaz de línea de comandos

Crear un trabajo de replicación para el invitado con ID 100 que se ejecute cada 5 minutos, con un límite de ancho de banda de 10 MB/s:

```
# pvesr create-local-job 100-0 pve1 --schedule "*/5" --rate 10
```

Desactivar el trabajo activo con ID 100-0:

```
# pvesr disable 100-0
```

Activar de nuevo el trabajo con ID 100-0:

```
# pvesr enable 100-0
```

Modificar el intervalo de programación del trabajo 100-0 para que se ejecute una vez por hora:

```
# pvesr update 100-0 --schedule '*/00'
```

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 10A

Capítulo 10 · Máquinas virtuales QEMU/KVM
Introducción y configuración de la máquina virtual (10.1–10.2.20)

Traducción de trabajo basada en "Proxmox VE Administration Guide", versión 9.2.4
Se conservan comandos, rutas, nombres de opciones y terminología técnica del original.

Capítulo 10 · Máquinas virtuales QEMU/KVM

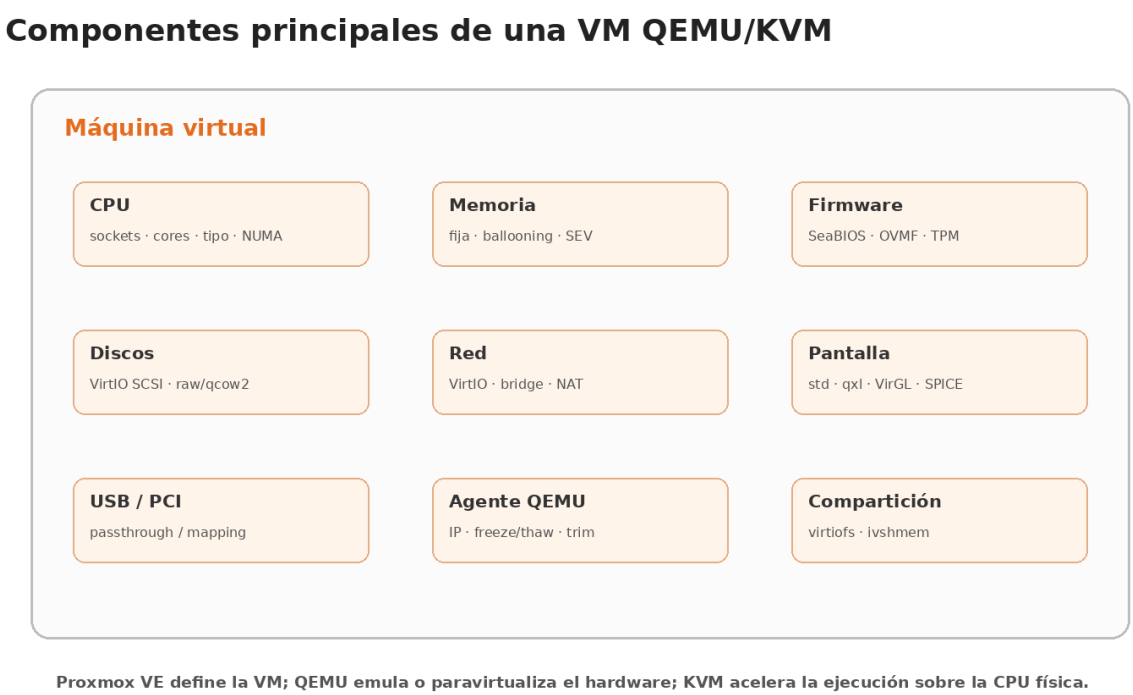
QEMU (abreviatura de Quick Emulator) es un hipervisor de código abierto que emula un ordenador físico. Desde el punto de vista del sistema anfitrión, QEMU es un programa de usuario con acceso a recursos locales -particiones, archivos, tarjetas de red y otros dispositivos- que se presentan después al ordenador emulado como si fueran dispositivos reales.

Un sistema operativo invitado que se ejecuta dentro de ese ordenador emulado accede a dichos dispositivos y se comporta como si funcionara sobre hardware real. Por ejemplo, es posible pasar una imagen ISO como parámetro a QEMU y el sistema operativo invitado verá un CD-ROM real insertado en una unidad óptica.

QEMU puede emular una gran variedad de arquitecturas, desde ARM hasta SPARC. En anfitriones x86, Proxmox VE se centra en la emulación de PCs de 32 y 64 bits, que representan la inmensa mayoría del hardware de servidor. En anfitriones arm64, Proxmox VE emula máquinas ARM de 64 bits (aarch64). Emular la misma arquitectura que la del anfitrión es además una de las opciones más rápidas, porque permite utilizar extensiones del procesador que aceleran notablemente QEMU cuando la arquitectura emulada coincide con la física.

NOTA: A veces se utiliza el término KVM (Kernel-based Virtual Machine). Significa que QEMU se ejecuta con ayuda de las extensiones de virtualización del procesador a través del módulo KVM de Linux. En el contexto de Proxmox VE, QEMU y KVM pueden usarse prácticamente como sinónimos, porque QEMU intentará cargar siempre el módulo KVM.

Dentro de Proxmox VE, QEMU se ejecuta como proceso root, ya que necesita acceso a dispositivos de bloques y PCI.



Vista resumida de los componentes de una VM QEMU/KVM según las opciones tratadas en este capítulo.

10.1 Dispositivos emulados y dispositivos paravirtualizados

El hardware de PC emulado por QEMU incluye placa base, controladores de red, controladores SCSI, IDE y SATA, puertos serie y otros dispositivos. Todos ellos son equivalentes software de hardware físico existente; si el sistema operativo invitado disp one del controlador adecuado, los utilizará como si estuviera ejecutándose sobre el dispositivo real. Esto permite ejecutar sistemas operativos sin modificar.

La contrapartida es el coste de rendimiento: reproducir por software aquello que originalmente se diseñó para ejecutarse en hardware implica trabajo adicional para la CPU del anfitrión. Para reducir esa sobrecarga, QEMU puede presentar dispositivos paravirtualizados, de modo que el sistema invitado reconoce que se ejecuta dentro de QEMU y coopera con el hipervisor.

QEMU se apoya en el estándar VirtIO y puede presentar dispositivos VirtIO paravirtualizados: controlador genérico de disco, t arjeta de red, puerto serie, controlador SCSI y otros.

CONSEJO: Utilice dispositivos VirtIO siempre que sea posible. Proporcionan una mejora importante de rendimiento y suelen estar mejor mantenidos. El manual cita como referencia que VirtIO para disco puede duplicar el rendimiento secuencial de escritura frente a IDE emulado y que una NIC VirtIO puede alcanzar hasta tres veces el rendimiento de una Intel E1000 emulada, según las pruebas referenciadas por KVM.



La emulación maximiza compatibilidad; VirtIO reduce la sobrecarga al cooperar el invitado con QEMU/KVM.

10.2 Configuración de máquinas virtuales

En general, Proxmox VE intenta elegir valores predeterminados sensatos para las máquinas virtuales. Conviene comprender el significado de cualquier ajuste que se modifique, ya que una configuración inapropiada puede reducir el rendimiento o incluso poner en riesgo los datos.

10.2.1 Ajustes generales

Los ajustes generales de una VM incluyen:

- Node: el servidor físico en el que se ejecutará la VM.
- VM ID: número único dentro de la instalación de Proxmox VE que identifica la máquina virtual.
- Name: texto libre que permite describir la VM.
- Resource Pool: grupo lógico de máquinas virtuales.

10.2.2 Ajustes del sistema operativo

Al crear una VM, seleccionar correctamente el sistema operativo permite que Proxmox VE optimice ciertos parámetros de bajo nivel. Por ejemplo, los sistemas Windows esperan normalmente que el reloj de BIOS utilice la hora local, mientras que los sistemas basados en Unix suelen esperar que la BIOS mantenga la hora UTC.

10.2.3 Ajustes del sistema

Durante la creación de la VM pueden modificarse algunos componentes básicos del sistema, incluido el tipo de pantalla y el controlador SCSI.

Si se prevé instalar QEMU Guest Agent, o la imagen ISO elegida ya lo incluye y lo instala automáticamente, puede marcarse la opción QEMU Agent. De este modo Proxmox VE sabe que puede usar sus funciones para mostrar información adicional y ejecutar determinadas operaciones -por ejemplo apagados o instantáneas- de forma más inteligente.

Proxmox VE permite arrancar VMs con distintos firmwares y tipos de máquina. En x86 están disponibles SeaBIOS y OVMF; normalmente se mantiene SeaBIOS salvo que haya un motivo para usar OVMF, como ciertos escenarios de PCIe passthrough. En arm64, OVMF es la opción predeterminada y la única disponible, porque SeaBIOS es exclusivo de x86.

Tipo de máquina

El tipo de máquina define la disposición del hardware de la placa base virtual. En x86 puede elegirse entre el chipset Intel 440FX predeterminado o Q35. Q35 añade un bus PCIe virtual y puede ser preferible cuando se desea pasar hardware PCIe directamente a la VM. También puede seleccionarse una implementación vIOMMU.

NOTA: Los chipsets Intel 440FX y Q35 son específicos de invitados x86. En anfitriones arm64 las VMs utilizan el tipo de máquina virt, que modela una máquina ARM genérica basada en UEFI.

Versión de máquina

Cada tipo de máquina está versionado en QEMU y un mismo binario de QEMU admite múltiples versiones. Las versiones nuevas pueden añadir funciones, correcciones o mejoras generales, pero también modifican propiedades del hardware virtual. Para evitar cambios repentinos desde la perspectiva del invitado y conservar la compatibilidad del estado de la VM, las migraciones en vivo y las instantáneas con RAM siguen utilizando la misma versión de máquina en la nueva instancia de QEMU.

En invitados Windows, la versión de máquina queda fijada durante la creación, porque Windows es sensible a cambios del hardware virtual incluso entre arranques en frío. Por ejemplo, la enumeración de dispositivos de red puede variar entre versiones. Otros sistemas, como Linux, suelen tolerar bien esos cambios y por defecto utilizan Latest; tras un arranque nuevo se emplea la versión más reciente soportada por el binario QEMU instalado.

La versión de máquina también se usa como mecanismo de seguridad para introducir funciones o correcciones que modifican la disposición del hardware manteniendo compatibilidad hacia atrás. En operaciones sobre una VM en ejecución -como una migración en vivo- se conserva la versión activa para poder recuperar exactamente el mismo entorno, tanto desde el punto de vista de QEMU como de la forma en que Proxmox VE construye la instancia.

Revisión de máquina PVE

En ocasiones Proxmox VE necesita modificar la disposición del hardware o ciertos valores sin esperar a una nueva versión de QEMU. Para ello añade una revisión downstream con el formato +pveX. En cada nueva versión de máquina QEMU, X comienza en 0 y puede omitirse; por ejemplo, pc-q35-9.2 equivale a pc-q35-9.2+pve0. Si Proxmox VE necesita cambiar la disposición del hardware o un valor predeterminado, incrementa esta revisión y la utiliza para invitados nuevos o, en VMs configuradas para usar siempre la versión más reciente, en el siguiente reinicio.

Retirada de versiones antiguas de máquina QEMU

A partir de QEMU 10.1, upstream elimina versiones de máquina cuando tienen más de seis años. Las versiones principales de Proxmox VE aparecen aproximadamente cada dos años, por lo que una versión principal de Proxmox VE admite en torno a dos generaciones principales anteriores de versiones de máquina.

Antes de actualizar a una nueva versión principal de Proxmox VE conviene actualizar las configuraciones de VM para evitar versiones de máquina que se vayan a retirar durante el siguiente ciclo. El manual indica que el último binario QEMU esperado en Proxmox VE 9 es QEMU 11.2 y que ese binario eliminará versiones anteriores a 6.0; por tanto, 6.0 actúa como línea base durante el ciclo de Proxmox VE 9. Se espera que dicha línea base aumente aproximadamente dos versiones principales en cada gran lanzamiento de Proxmox VE.

Actualizar a una versión de máquina más nueva

Si aparece un aviso de obsolescencia, debe cambiarse la versión de máquina por una más reciente. Antes es imprescindible disponer de una copia de seguridad funcional y estar preparado para cambios en la forma en que el invitado detecta el hardware; en algunos escenarios puede ser necesario reinstalar controladores. También deben revisarse las instantáneas con RAM creadas con esas versiones -entrada runningmachine-. No existe un modo de cambiar la versión de máquina de una instantánea, por lo que habría que cargarla para recuperar cualquier dato necesario.

10.2.4 Disco duro

Bus / controlador

QEMU puede emular varios controladores de almacenamiento.

CONSEJO: Por rendimiento y mantenimiento, el manual recomienda VirtIO SCSI o VirtIO Block siempre que sea posible.

- IDE: diseño heredado del controlador de disco del PC/AT de 1984. Su gran ventaja es la compatibilidad: prácticamente cualquier sistema operativo dispone de controlador. Es útil para sistemas anteriores a 2003. Permite hasta 4 dispositivos.
- SATA (Serial ATA): diseño más moderno, de 2003, con mayor rendimiento y más dispositivos. Permite hasta 6 dispositivos.
- SCSI: tecnología habitual en hardware de servidor. Puede conectar hasta 14 dispositivos de almacenamiento. Proxmox VE emula de forma predeterminada un controlador LSI 53C895A.
- VirtIO SCSI single: recomendado para rendimiento junto con IO Thread en los discos conectados. Cada disco dispone de su propio controlador VirtIO SCSI y QEMU procesa su E/S en un hilo dedicado. Es el valor predeterminado para nuevas VMs Linux desde Proxmox VE 7.3. Linux dispone de soporte desde 2012 y FreeBSD desde 2014; en Windows es necesario aportar durante la instalación una ISO adicional con los controladores.
- VirtIO Block (virtio-blk): controlador paravirtualizado más antiguo. Ha sido superado en funciones por VirtIO SCSI.

Formato de imagen

A cada controlador pueden conectarse discos emulados respaldados por un archivo o un dispositivo de bloques situado en el almacenamiento configurado. El tipo de almacenamiento condiciona el formato: LVM, ZFS y Ceph, que exponen dispositivos de bloques, requieren normalmente raw; los almacenamientos basados en archivos, como Ext4, XFS, NFS o CIFS, permiten elegir raw o el formato de imagen de QEMU.

- QEMU image (qcow2): formato copy-on-write que permite instantáneas y thin provisioning.
- raw: imagen bit a bit del disco, similar a la obtenida con dd. Por sí sola no ofrece thin provisioning ni snapshots, por lo que necesita soporte de la capa de almacenamiento. El manual indica que puede llegar a ser hasta un 10 % más rápida que qcow2 en determinadas pruebas.
- VMware image: tiene sentido principalmente para importar o exportar la imagen hacia otros hipervisores.

Modo de caché

El modo Cache determina cuándo el anfitrión notifica al invitado que una escritura de bloques ha finalizado. El valor predeterminado No cache considera completada la escritura cuando el bloque alcanza la cola de escritura del almacenamiento físico, ignorando la page cache del anfitrión. Proporciona un equilibrio razonable entre seguridad y velocidad.

La opción No backup permite excluir un disco de las copias de seguridad de la VM. La opción Skip replication permite excluirlo de los trabajos de replicación. El manual recuerda que la replicación de Proxmox VE requiere que los discos replicados residan en almacenamiento zfspool, por lo que un disco situado en otro tipo de almacenamiento debe marcarse para omitir la replicación cuando la VM tenga trabajos de replicación configurados.

TRIM / Discard

Si el almacenamiento admite thin provisioning puede activarse Discard. Con un sistema invitado que soporte TRIM, cuando el sistema de archivos marque bloques como no usados tras eliminar datos, el controlador puede transmitir esa información al almacenamiento para reducir el espacio realmente ocupado por la imagen. Para que el invitado pueda emitir TRIM, Discard debe estar habilitado; algunos sistemas operativos también pueden necesitar SSD Emulation. En unidades VirtIO Block, Discard requiere un invitado con kernel Linux 5.0 o posterior.

SSD Emulation presenta la unidad al invitado como SSD aunque el almacenamiento subyacente no tenga por qué ser SSD. No está disponible para unidades VirtIO Block.

IO Thread

IO Thread puede utilizarse con discos VirtIO o con SCSI cuando el controlador emulado es VirtIO SCSI single. Al habilitarlo, QEMU crea un hilo de E/S por controlador de almacenamiento, en lugar de gestionar toda la E/S en el bucle principal o en los hilos vCPU. Esto mejora la distribución del trabajo y el aprovechamiento del almacenamiento, y puede reducir la latencia o los bloqueos del invitado en cargas de E/S muy intensas, al evitar que un hilo principal o una vCPU queden bloqueados por operaciones de disco.

10.2.5 CPU

Un socket de CPU es el zócalo físico de una placa base donde se instala un procesador. Ese procesador puede contener uno o varios núcleos independientes. Desde el punto de vista del rendimiento, disponer de un socket con cuatro núcleos o dos sockets con dos núcleos suele ser poco relevante, aunque algunas licencias de software dependen del número de sockets; en ese caso tiene sentido ajustar la topología virtual a lo permitido por la licencia.

Aumentar el número de CPUs virtuales -núcleos y sockets- suele mejorar el rendimiento, aunque el resultado depende mucho de la carga. Las aplicaciones multihilo se benefician de más vCPU, ya que por cada vCPU QEMU crea un nuevo hilo de ejecución en el anfitrión. Si se desconoce la carga, el manual considera razonable empezar con Total cores = 2.

NOTA: Es seguro que la suma de núcleos asignados a todas las VMs sea mayor que el número de núcleos físicos del servidor. El anfitrión repartirá los hilos QEMU entre los núcleos, como con cualquier aplicación multihilo. Sin embargo, Proxmox VE impide iniciar una VM con más núcleos virtuales que los núcleos físicos disponibles, porque el coste de los cambios de contexto reduciría el rendimiento.

Límites de recursos

cpulimit: limita el tiempo total de CPU del anfitrión disponible para la VM. Es un número en coma flotante expresado como múltiplo de 100 %: 1.0 equivale a 100 %, 2.5 a 250 %, etc. Una VM de cuatro núcleos utilizando plenamente sus vCPU podría consumir teóricamente 400 %, e incluso algo más por los hilos adicionales de periféricos de QEMU.

El ajuste resulta útil cuando una VM necesita varias vCPU para ejecutar tareas en paralelo, pero no debe poder saturarlas todas simultáneamente. Por ejemplo, una VM con 8 vCPU y cpulimit=4.0 queda limitada a 400 % en total. Si usa las 8 vCPU a la vez, cada una dispondrá como máximo de aproximadamente el 50 % de un núcleo físico; si solo utiliza cuatro, esas cuatro podrían alcanzar el 100 % hasta sumar el mismo 400 %.

NOTA: Una VM puede usar hilos adicionales para red, E/S o migración en vivo. Si se quiere garantizar que nunca consuma más tiempo de CPU que el asignable a sus vCPU, establezca cpulimit al mismo valor que el número total de núcleos virtuales.

cpuunits: también denominado CPU shares o CPU weight, controla cuánto tiempo de CPU recibe una VM en relación con otras. El valor relativo predeterminado es 100 -o 1024 en hosts con cgroup v1 heredado-. Si una VM tiene 200 y otra 100, la primera puede recibir aproximadamente el doble de ancho de banda de CPU cuando ambas compiten por recursos.

En systemd.resource-control, CPUQuota corresponde conceptualmente a cpulimit y CPUWeight a cpuunits.

affinity: permite indicar los núcleos físicos concretos en los que pueden ejecutarse las vCPU de una VM. Los procesos periféricos de la VM, como la E/S, no se ven afectados. La afinidad no es una función de seguridad y aumenta la complejidad de mantenimiento. Puede producir desequilibrios si algunos núcleos quedan saturados mientras otros permanecen ociosos.

La afinidad se expresa mediante el formato de lista aceptado por taskset/cpuset. Por ejemplo, 0-1,8-11 equivale a 0,1,8,9,10,11 y restringiría las vCPU a esos seis núcleos del anfitrión.

Tipo de CPU

QEMU puede emular numerosos tipos de CPU, desde 486 hasta generaciones recientes de Xeon. Cada generación incorpora capacidades nuevas -renderizado asistido por hardware, generación aleatoria, protección de memoria, etc.- y una misma generación puede recibir correcciones mediante actualizaciones de microcódigo.

Normalmente conviene seleccionar un tipo de procesador que se aproxime a la CPU física, para que el invitado pueda utilizar más características del anfitrión. El tipo host ofrece una correspondencia prácticamente exacta de flags de CPU.

El inconveniente aparece en migraciones en vivo entre anfitriones con CPUs o microcódigos diferentes. Si el invitado utiliza flags que faltan en el destino, QEMU puede detenerse. Para evitarlo, Proxmox VE utiliza por defecto modelos virtuales de CPU definidos por QEMU.

El backend utiliza kvm64 como valor base ampliamente compatible en x86_64, mientras que la interfaz de creación de VM propone x86-64-v2-AES, que requiere al menos Intel Westmere o AMD Opteron de cuarta generación.

NOTA: Los modelos kvm64 y x86-64-v* son exclusivos de x86. En arm64 se utilizan modelos ARM: host para exponer la CPU física -recomendado en el hardware soportado- o un modelo genérico como max junto al tipo de máquina virt.

Resumen de elección sugerido por el manual:

- Si la migración en vivo no importa, o todos los nodos son homogéneos y tienen la misma CPU y microcódigo, use host para maximizar las prestaciones disponibles.
- Si se necesita migración en vivo y seguridad, y todo el clúster es Intel o todo AMD, elija el modelo correspondiente a la generación más antigua presente en el clúster.
- Si se necesita migración en vivo sin priorizar esa exposición de características, o el clúster mezcla Intel y AMD, elija el tipo virtual QEMU compatible más bajo.

NOTA: No existe garantía de que las migraciones en vivo entre CPUs anfitrionas Intel y AMD funcionen correctamente.

Tipos de CPU QEMU

QEMU ofrece tipos virtuales compatibles con CPUs anfitrionas Intel y AMD. Históricamente Proxmox VE utilizaba kvm64, con flags equivalentes aproximadamente a Pentium 4, lo que podía limitar el rendimiento de algunas cargas. En 2020 AMD, Intel, Red Hat y SUSE definieron niveles de microarquitectura x86-64 con flags modernos sobre la base x86-64.

NOTA: Algunas distribuciones recientes, como CentOS 9, se compilan exigiendo como mínimo x86-64-v2.

Modelo	Compatibilidad orientativa	Flags añadidos principales
kvm64 (x86-64-v1)	Intel >= Pentium 4 · AMD >= Phenom	Base x86-64-v1
x86-64-v2	Intel >= Nehalem · AMD >= Opteron_G3	cx16, lahf-lm, popcnt, pni, sse4.1, sse4.2, ssse3
x86-64-v2-AES	Intel >= Westmere · AMD >= Opteron_G4	aes
x86-64-v3	Intel >= Haswell · AMD >= EPYC	avx, avx2, bmi1/2, f16c, fma, movbe, xsave
x86-64-v4	Intel >= Skylake · AMD >= EPYC v4 Genoa	avx512f/bw/cd/dq/vl

Tipos de CPU personalizados

Es posible definir tipos de CPU personalizados con modelo base, fabricante, bits de dirección física y activación o desactivación de flags. Después pueden asignarse a una VM como cualquier modelo integrado, usando custom-<name>.

Se gestionan en Datacenter -> Custom CPU Models. El diálogo incorpora una cuadrícula de flags con búsqueda por nombre y descripción, selector de acelerador kvm/tcg y selección de nodos que resalta qué flags declara como compatibles cada nodo. La misma cuadrícula se reutiliza en el editor Processor de cada VM, limitada allí a los flags que pueden definirse por VM. Los flags desconocidos para la API o no admitidos por el acelerador seleccionado aparecen con un aviso.

Los modelos siguen respaldados por /etc/pve/virtual-guest/cpu-models.conf y pueden administrarse también por CLI; el formato se describe en man cpu-models.conf.

El acceso se controla por modelo mediante la ruta ACL /mapping/cpu/<name>. Mapping.Audit permite ver el modelo en listados; Mapping.Modify controla crear, actualizar y eliminar; Mapping.Use es necesario para asignarlo a una VM. Este último permiso se comprueba al crear, actualizar o clonar una VM, por lo que incluso un usuario con permiso para clonar una VM que usa un modelo personalizado necesita Mapping.Use sobre la ruta ACL de dicho modelo.

Flags relacionados con Meltdown y Spectre

Existen varios flags de CPU vinculados a mitigaciones de Meltdown y Spectre que deben activarse manualmente salvo que el modelo de CPU seleccionado ya los incluya.

- La CPU física debe soportar la función y propagarla a las vCPU del invitado.
- El sistema operativo invitado debe estar actualizado a una versión que implemente la mitigación y pueda utilizar el flag.

En caso contrario, el flag deseado debe añadirse desde las opciones de CPU de la interfaz web o mediante la propiedad flags de cpu en la configuración de la VM. Para mitigaciones Spectre v1, v2 y v4, el fabricante de CPU o sistema también puede necesitar proporcionar una actualización de microcódigo; no todas las CPUs afectadas pueden actualizarse para soportar spec-ctrl.

Para comprobar las vulnerabilidades visibles en el anfitrión:

```
for f in /sys/devices/system/cpu/vulnerabilities/*; do echo "${f##*/} -" $(cat "$f"); done
```

Procesadores Intel

- pcid: reduce el impacto de rendimiento de la mitigación de Meltdown (CVE-2017-5754) denominada Kernel Page-Table Isolation (KPTI). Sin PCID, KPTI puede resultar costoso. Para comprobar soporte: grep ' pcid ' /proc/cpuinfo.
- spec-ctrl: necesario para mitigaciones de Spectre v1/v2 cuando retpolines no es suficiente. Se incluye por defecto en modelos Intel con sufijo -IBRS; en el resto debe activarse explícitamente. Requiere microcódigo actualizado (intel-microcode >= 20180425).
- ssbd: necesario para Spectre v4 (CVE-2018-3639). No se incluye por defecto en ningún modelo Intel; debe activarse explícitamente y requiere microcódigo actualizado (intel-microcode >= 20180703).

Procesadores AMD

- ibpb: necesario para Spectre v1/v2 cuando retpolines no es suficiente. Se incluye por defecto en modelos AMD con sufijo -IBPB; en los demás debe activarse y requiere soporte del microcódigo del anfitrión.
- virt-ssbd: necesario para Spectre v4. Debe activarse explícitamente, incluso con modelo host, porque es una característica virtual que no existe como tal en la CPU física. Se recomienda exponerla para maximizar compatibilidad.
- amd-ssbd: mitigación de Spectre v4 con mejor rendimiento que virt-ssbd cuando está soportada. Se recomienda exponer ambas si es posible, ya que algunos kernels solo reconocen virt-ssbd.
- amd-no-ssb: indica que el anfitrión no es vulnerable a Spectre v4, evitando que el invitado active mitigaciones innecesarias. Es mutuamente excluyente con virt-ssbd y amd-ssbd.

Ajustes de CPU relacionados con Windows

- cet-ibt y cet-ss (Control-flow Enforcement Technology: indirect branch tracking y shadow stack) se deshabilitan de forma predeterminada para tipos de máquina Windows 11 porque actualmente pueden impedir el arranque de invitados con seguridad basada en virtualización. Pueden reactivarse por VM desde el editor de flags de Processor si una carga los necesita.
- La propiedad QEMU level está expuesta por VM y en modelos personalizados. level=30 es una solución conocida para ciertos fallos de arranque de Hyper-V en invitados Windows sobre hosts Intel recientes cuando se usa CPU host o max. Solo se aplica a vCPU x86_64 y se ignora silenciosamente en otras arquitecturas.

NUMA

Opcionalmente puede emularse una arquitectura NUMA. En lugar de disponer de un único pool global de memoria para todos los núcleos, la RAM se reparte en bancos locales próximos a cada socket. Esto puede mejorar el rendimiento al reducir cuellos de botella del bus de memoria. Si el anfitrión es NUMA, el manual recomienda activar NUMA para distribuir correctamente los recursos de la VM. También es necesario para hot-plug de CPU o RAM.

Cuando se usa NUMA, se recomienda configurar el número de sockets de la VM igual al número de nodos NUMA del anfitrión. Un anfitrión presenta más de un nodo NUMA si numactl --hardware | grep available devuelve más de uno.

Hot-plug de vCPU

Los sistemas modernos pueden añadir y, hasta cierto punto, retirar CPUs en caliente. En virtualización se evitan muchos problemas físicos, pero sigue siendo una función relativamente nueva y compleja, por lo que el manual recomienda reservarla para casos en los que resulte realmente necesaria.

En Proxmox VE el máximo de CPUs conectadas es cores * sockets. Para iniciar una VM con menos vCPU que ese máximo puede usarse vcpus, que indica cuántas deben estar conectadas al arrancar.

La función se admite actualmente en Linux con kernel posterior a 3.10, y se recomienda uno posterior a 4.7. Para poner automáticamente en línea nuevas CPUs dentro del invitado puede usarse una regla udev:

```
SUBSYSTEM=="cpu", ACTION=="add", TEST=="online", ATTR{online}=="0", ATTR{online}="1"
```

Guarde la regla en /etc/udev/rules.d/ con un nombre terminado en .rules. La retirada en caliente depende del tipo de máquina y requiere cooperación del sistema invitado. La orden de eliminación es en realidad una solicitud al invitado -por ejemplo mediante ACPI en x86/amd64- y no garantiza que la CPU llegue a retirarse.

10.2.6 Memoria

Cada VM puede utilizar una cantidad fija de memoria o permitir que Proxmox VE la ajuste dinámicamente en función del uso de RAM del anfitrión.

Asignación fija

Cuando memory y minimum memory tienen el mismo valor, Proxmox VE asigna a la VM exactamente la cantidad indicada.

Incluso con tamaño fijo se añade el dispositivo de ballooning, porque proporciona información útil sobre el uso real de memoria del invitado. En general se recomienda mantenerlo habilitado. Para desactivarlo, por ejemplo durante una depuración, desmarque Ballooning Device o configure:

```
balloon: 0
```

Asignación automática

Cuando minimum memory es inferior a memory, Proxmox VE garantiza siempre la cantidad mínima y, mientras el uso de RAM del anfitrión esté por debajo de un porcentaje objetivo, puede aumentar dinámicamente la memoria del invitado hasta el máximo configurado. El objetivo predeterminado es 80 % y se puede cambiar en las opciones del nodo.

Cuando el anfitrión se queda corto de RAM, la VM puede devolver páginas de memoria; el invitado puede terminar usando swap y, como último recurso, su OOM killer. El intercambio de memoria entre anfitrión e invitado lo realiza un controlador balloon dentro del invitado.

Si varias VMs usan asignación automática, Shares define la proporción relativa de memoria libre que recibe cada una. El ejemplo del manual usa cuatro VMs: tres servidores HTTP con Shares=1000 y una base de datos con Shares=3000. En un host de 32 GB configurado al 80 % y que ya usa 16 GB quedan 9 GB adicionales para repartir sobre los mínimos. La VM de base de datos obtiene 4,5 GB y cada servidor HTTP 1,5 GB.

Las distribuciones Linux posteriores a 2010 suelen incluir el controlador balloon. En Windows debe instalarse manualmente y puede introducir cierta ralentización, por lo que el manual no recomienda usarlo en sistemas críticos.

CONSEJO: Como regla general, al asignar RAM a las VMs conviene dejar siempre alrededor de 1 GB disponible para el anfitrión.



La asignación automática mantiene un mínimo y puede redistribuir memoria libre mediante ballooning.

10.2.7 Cifrado de memoria

AMD SEV

SEV (Secure Encrypted Virtualization) permite cifrar la memoria de cada VM con AES-128 mediante el AMD Secure Processor. SEV-ES (Encrypted State) añade el cifrado del contenido de los registros de CPU para reducir la exposición de información al hipervisor. SEV-SNP (Secure Nested Paging) intenta además proteger frente a ataques de integridad basados en software.

Requisitos del anfitrión

- CPU AMD EPYC.
- SEV-ES: AMD EPYC serie 7002 o posterior.
- SEV-SNP: AMD EPYC serie 7003 o posterior.
- SEV-SNP requiere kernel del anfitrión 6.11 o superior.
- Activar el cifrado de memoria AMD en la BIOS del anfitrión.
- Añadir kvm_amd.sev=1 a los parámetros del kernel si no está habilitado por defecto.
- Añadir mem_encrypt=on si se quiere cifrar también la memoria del anfitrión mediante SME.
- Puede ser necesario aumentar SWIOTLB según el entorno.

Para comprobar si SEV está habilitado:

```
# dmesg | grep -i sev
[...] ccp 0000:45:00.1: sev enabled
[...] SEV supported: <number> ASIDs
[...] SEV-ES supported: <number> ASIDs
# cat /sys/module/kvm_amd/parameters/sev
Y
```

Requisitos del invitado

- edk2-OVMF.
- Se recomienda Q35.
- El sistema operativo invitado debe incluir soporte SEV.

Limitaciones

- Al estar la memoria cifrada, la medición de uso de memoria del anfitrión no es correcta.
- Las operaciones que guardan o restauran memoria, como instantáneas y migración en vivo, todavía no funcionan o pueden ser atacables.
- PCI passthrough no está soportado.
- SEV-ES y SEV-SNP son funciones muy experimentales.
- Los discos EFI no están soportados con SEV-SNP.
- Con SEV-SNP, la orden reboot dentro de la VM simplemente apaga la máquina.

Ejemplo SEV

```
# qm set <vmid> -amd-sev type=std,no-debug=1,no-key-sharing=1,kernel-hashes=1
```

type define la tecnología: std, es o snp. La política QEMU se calcula a partir de no-debug y no-key-sharing, que corresponden a los bits 0 y 1. Con type=es se activa además el bit 2 para SEV-ES. El bit 3 (nosend) se establece siempre a 1 para evitar ataques relacionados con migración.

kernel-hashes está desactivado por defecto por compatibilidad con imágenes OVMF antiguas e invitados que no miden kernel/initrd.

Comprobación dentro de la VM mediante dmesg:

```
# dmesg | grep -i sev
AMD Memory Encryption Features active: SEV
```

Comprobación mediante MSR 0xc0010131:

```
# apt install msr-tools
# modprobe msr
# rdmsr -a 0xc0010131
1
```

Ejemplo SEV-SNP

```
# qm set <vmid> -amd-sev type=snp,allow-smt=1,no-debug=1,kernel-hashes=1
```

El bit de política allow-smt está activado por defecto. Si se establece allow-smt=0, SMT debe estar deshabilitado en el anfitrión para que la VM pueda arrancar.

```
# dmesg | grep -i snp
Memory Encryption Features active: AMD SEV SEV-ES SEV-SNP
SEV: Using SNP CPUID table, 29 entries present.
SEV: SNP guest platform device initialized.
```

Referencias indicadas por el manual: https://developer.amd.com/sev/ · https://github.com/AMDESE/AMDSEV · documentación QEMU sobre AMD memory encryption · especificación AMD SEV API y documentación de SUSE sobre SEV.

10.2.8 Dispositivo de red

Una VM puede tener múltiples controladores de interfaz de red de cuatro tipos principales:

- Intel E1000: valor tradicional predeterminado; emula una tarjeta Gigabit Intel.
- VirtIO: NIC paravirtualizada recomendada cuando se busca máximo rendimiento. El invitado debe tener instalado el controlador VirtIO.
- Realtek 8139: emula una NIC antigua de 100 MB/s y debería reservarse para sistemas anteriores aproximadamente a 2002.
- vmxnet3: otro dispositivo paravirtualizado, pensado principalmente para VMs importadas desde otros hipervisores.

Proxmox VE genera para cada NIC una dirección MAC aleatoria, de forma que la VM sea direccionable en redes Ethernet.

Cada NIC puede funcionar en dos modelos de conectividad:

- Bridged, predeterminado: cada NIC virtual se respalda en el anfitrión mediante un dispositivo tap que se añade a un bridge, normalmente vmbr0. La VM accede directamente a la LAN Ethernet en la que está el host.
- NAT: la NIC se comunica únicamente con la pila de red de usuario de QEMU, que incorpora router y DHCP. El DHCP integrado entrega direcciones privadas 10.0.2.0/24. Es mucho más lento que bridge y se recomienda solo para pruebas. Solo está disponible mediante CLI o API, no desde la interfaz web.

Durante la creación puede elegirse No network device para no añadir NIC. La MTU puede sobrescribirse por dispositivo; un campo vacío o mtu=1 hereda la MTU del bridge subyacente. Esta opción solo está disponible con VirtIO.

Multiqueue

Con VirtIO puede activarse Multiqueue para que el invitado procese paquetes con varias vCPU y aumente el número total de paquetes transferidos. Cada cola se entrega al kernel del anfitrión, donde la procesa un hilo creado por vhost; Multiqueue permite proporcionar varias colas por NIC.

Se recomienda configurar un número de colas igual al número de vCPU del invitado. Dentro de una VM Linux debe configurarse también el número de canales combinados de cada NIC VirtIO, por ejemplo:

```
ethtool -L ens1 combined X
```

X es el número de vCPU de la VM. En Windows se instalan los controladores Red Hat VirtIO Ethernet Adapter y, en las propiedades avanzadas de la NIC, se habilita Receive Side Scaling y se ajusta Maximum number of RSS Queues al número de vCPU.

NOTA: Multiqueue aumenta el consumo de CPU del host y del invitado a medida que crece el tráfico. El manual recomienda activarlo cuando la VM procese muchas conexiones entrantes, por ejemplo en routers, proxies inversos o servidores HTTP muy cargados.

10.2.9 Pantalla

QEMU puede virtualizar distintos adaptadores VGA:

- std: predeterminado; emula una tarjeta con extensiones Bochs VBE.
- cirrus: antiguo valor predeterminado. Emula hardware muy viejo y solo debería usarse cuando sea realmente necesario, por ejemplo con Windows XP o anterior.
- vmware: adaptador compatible con VMware SVGA-II.
- qxl: tarjeta gráfica QXL paravirtualizada. Al seleccionarla también se habilita SPICE.
- virtio-gl / VirGL: GPU 3D virtual que puede descargar trabajo en la GPU del anfitrión sin requerir modelos caros ni dedicar por completo la GPU física, permitiendo reutilizarla entre varios invitados y/o el host.

NOTA: VirGL necesita bibliotecas adicionales que no se instalan por defecto. En muchos entornos basta con: apt install libgl1 libegl1.

```
apt install libgl1 libegl1
```

La opción memory controla cuánta memoria recibe la GPU virtual y puede permitir resoluciones superiores, especialmente con SPICE/QXL.

En SPICE Multi-Monitor, la memoria se reserva por dispositivo. Windows necesita un dispositivo por monitor, por lo que Proxmox VE añade dispositivos adicionales y cada uno recibe la memoria especificada. En Linux pueden habilitarse más monitores virtuales, pero elegir un modo multi-monitor multiplica la memoria entregada al dispositivo por el número de monitores.

Seleccionar serialX como tipo de pantalla deshabilita la salida VGA y redirige la consola web al puerto serie seleccionado. En ese caso se ignora la memoria de pantalla configurada.

Portapapeles VNC

El portapapeles VNC puede habilitarse estableciendo clipboard=vnc:

```
# qm set <vmid> -vga <displaytype>,clipboard=vnc
```

Para usarlo deben instalarse primero las herramientas de invitado SPICE. En Debian y derivados puede instalarse spice-vdagent. Una vez instalado, el portapapeles VNC puede utilizarse en noVNC. Si se usa SPICE, virtio o virgl, hay que elegir qué portapapeles utilizar, ya que el portapapeles SPICE predeterminado se sustituye por el de VNC cuando clipboard=vnc.

NOTA: Para habilitar el portapapeles VNC, QEMU utiliza el dispositivo qemu-vdagent. El soporte de migración en vivo de este dispositivo se añadió en QEMU 10.1. Una VM ejecutándose con QEMU o versión de máquina anterior a 10.1 y con portapapeles VNC habilitado no puede migrarse en vivo.

10.2.10 USB Passthrough

Existen dos formas distintas de pasar dispositivos USB: Host USB passthrough y SPICE USB passthrough.

Host USB passthrough entrega a la VM un dispositivo USB conectado físicamente al anfitrión. Puede identificarse por vendor/product ID o por bus/puerto.

- Vendor/product ID: formato 0123:abcd, donde 0123 identifica al fabricante y abcd al producto. Dos unidades del mismo modelo comparten estos IDs.
- Bus/port: por ejemplo 1-2.3.4, donde 1 es el bus y 2.3.4 la ruta física del puerto en el anfitrión.

Si el dispositivo está configurado pero no presente al arrancar la VM, esta puede iniciar con normalidad. Cuando el dispositivo o puerto aparece en el anfitrión, se pasa a la VM.

ADVERTENCIA: Con Host USB passthrough la VM no puede migrarse en vivo a otro host, ya que el hardware solo está disponible en el nodo físico actual.

SPICE USB passthrough permite añadir uno o más puertos USB SPICE a la VM y redirigir dinámicamente desde el cliente SPICE un dispositivo USB local. Es útil para periféricos de entrada o dongles temporales.

También es posible mapear dispositivos a nivel de clúster para utilizarlos correctamente con HA, detectar cambios de hardware y permitir que usuarios no root los configuren. Consulte Resource Mapping.

10.2.11 BIOS y UEFI

QEMU necesita firmware para emular correctamente un ordenador. La BIOS o UEFI se ejecuta al principio del arranque, realiza la inicialización básica del hardware y proporciona al sistema operativo una interfaz con el firmware y los dispositivos. De forma predeterminada, QEMU utiliza SeaBIOS, una implementación BIOS x86 de código abierto adecuada para la mayoría de entornos estándar.

NOTA: SeaBIOS -y bios=seabios- solo está disponible para invitados x86. En arm64 las VMs arrancan siempre mediante UEFI usando la compilación ARM de OVMF, AAVMF; bios=ovmf es la única opción soportada.

Algunos sistemas, como Windows 11, pueden requerir una implementación UEFI compatible. En esos casos debe usarse OVMF, una implementación UEFI de código abierto. OVMF también puede ser preferible en otros escenarios, como ciertos passthrough de VGA.

Para que OVMF pueda guardar parámetros como el orden de arranque se necesita un EFI Disk. Ese disco se incluye en copias de seguridad e instantáneas y solo puede existir uno por VM.

```
# qm set <vmid> -efidisk0 <storage>:1,format=<format>,efitype=4m,pre-enrolled-keys=1
```

storage es el almacenamiento del disco y format un formato soportado por dicho almacenamiento. También puede añadirse desde Hardware -> Add -> EFI Disk.

efitype selecciona la versión de firmware OVMF. Para nuevas VMs debe utilizarse 4m, que admite Secure Boot y reserva más espacio para evoluciones futuras; es el valor predeterminado en la GUI.

pre-enrolled-keys indica si el EFI Disk debe incluir claves Secure Boot específicas de distribución y las claves estándar de Microsoft. También activa Secure Boot por defecto, aunque puede desactivarse desde el menú OVMF de la VM.

NOTA: Para habilitar Secure Boot en una VM existente con EFI Disk 2m hay que recrearlo: eliminar el antiguo con qm set <vmid> -delete efidisk0 y añadir uno nuevo. Esto borra cualquier configuración personalizada realizada en el menú OVMF.

Con OVMF y pantalla virtual, sin passthrough VGA, debe configurarse la resolución desde el menú OVMF -accesible con ESC durante el arranque- o utilizar SPICE. Para PXE con OVMF es necesario añadir un dispositivo RNG; por motivos de seguridad el firmware deshabilita PXE si el invitado no dispone de generador aleatorio.

Caducidad de certificados Secure Boot

El conjunto original de certificados Microsoft de 2011, usado por Windows y distribuciones Linux comunes para Secure Boot, caduca en junio de 2026. Microsoft creó certificados de reemplazo en 2023.

El firmware solo permite cargadores firmados con certificados presentes en el EFI Disk. Un disco que contenga únicamente certificados de 2011 rechazará cargadores firmados con los de 2023. El marcador ms-cert=2023k en la configuración del EFI Disk indica que los nuevos certificados están inscritos.

Con pve-edk2-firmware >= 4.2025.05-1, los nuevos EFI Disk incluyen certificados 2011 y 2023 y llevan el marcador ms-cert=2023k. Para discos anteriores hay que inscribirlos manualmente.

En Windows con BitLocker, antes de continuar debe deshabilitarse temporalmente el protector en cada unidad cifrada, por ejemplo:

```
manage-bde -protectors -disable C:
```

IMPORTANTE: Si no se hace para cada unidad protegida con BitLocker, el siguiente arranque puede solicitar la clave de recuperación de BitLocker.

En la interfaz, seleccione el EFI Disk y use Disk Action -> Enroll Updated Certificates. También puede usarse el endpoint API /nodes/{node}/qemu/{vmid}/config. La inscripción se aplica en el siguiente arranque. Desde CLI, con la VM apagada, el equivalente es:

```
qm enroll-efi-keys <vmid>
```

Los marcadores ms-cert=2023 y ms-cert=2023w pueden indicar una inscripción parcial; la tarea de arranque avisará de ello y también debería aplicarse el procedimiento de inscripción.

10.2.12 Trusted Platform Module (TPM)

Un TPM almacena de forma segura datos secretos -por ejemplo claves de cifrado- y proporciona funciones resistentes a manipulaciones para validar el arranque del sistema. Algunos sistemas operativos, entre ellos Windows 11, requieren que la máquina disponga de TPM físico o virtual.

En Proxmox VE se añade mediante un volumen tpmstate. Se comporta de forma parecida a un EFI Disk en el sentido de que, una vez creado, no puede modificarse: solo retirarse. Ejemplo:

```
# qm set <vmid> -tpmstate0 <storage>:1,version=<version>
```

version puede ser v1.2 o v2.0. También puede añadirse en Hardware -> Add -> TPM State. TPM 2.0 es más reciente y está mejor soportado, por lo que debería preferirse salvo necesidad concreta de TPM 1.2.

NOTA: Un TPM emulado no proporciona los mismos beneficios de seguridad que un TPM físico. Sus datos se almacenan en un volumen ordinario y, por tanto, alguien con acceso suficiente podría modificarlos fuera de los mecanismos definidos por la especificación TPM.

10.2.13 Memoria compartida entre VMs

Puede añadirse un dispositivo de memoria compartida Inter-VM (ivshmem) para compartir memoria entre anfitrión e invitado o entre varios invitados.

```
# qm set <vmid> -ivshmem size=32,name=foo
```

size se expresa en MiB. El archivo se sitúa en /dev/shm/pve-shm-$name; si no se especifica nombre se utiliza el VMID.

NOTA: Actualmente el dispositivo se elimina en cuanto se apaga o detiene cualquiera de las VMs que lo utilicen. Las conexiones ya abiertas siguen existiendo, pero no pueden crearse nuevas conexiones al mismo dispositivo.

Un caso de uso citado por el manual es Looking Glass, que permite duplicación de pantalla de alto rendimiento y baja latencia entre host e invitado: https://looking-glass.io/.

10.2.14 Dispositivo de audio

Para añadir un dispositivo de audio:

```
qm set <vmid> -audio0 device=<device>
```

Dispositivos soportados:

- ich9-intel-hda: Intel HD Audio Controller, emula ICH9.
- intel-hda: Intel HD Audio Controller, emula ICH6.
- AC97: Audio Codec '97, útil para sistemas antiguos como Windows XP.

Backends disponibles: spice y none. spice se utiliza junto a SPICE; none puede ser útil cuando algún software necesita detectar un dispositivo de audio aunque no se requiera salida. Para utilizar el dispositivo de audio físico del host debe recurrirse a passthrough PCI o USB. Protocolos remotos como RDP también disponen de mecanismos para reproducir sonido.

10.2.15 VirtIO RNG

Un RNG (Random Number Generator) aporta entropía al sistema. Un RNG virtual puede proporcionar al invitado entropía obtenida del anfitrión y ayudar a evitar problemas de falta de entropía, especialmente durante el arranque.

```
qm set <vmid> -rng0 source=<source>[,max_bytes=X,period=Y]
```

source puede ser:

- /dev/urandom: pool de entropía del kernel no bloqueante; es la opción preferida.
- /dev/random: pool bloqueante; no recomendado porque puede provocar falta de entropía en el propio host.
- /dev/hwrng: pasa un RNG hardware conectado al anfitrión. Si hay varios, se usa el seleccionado en /sys/devices/virtual/misc/hw_random/rng_current.

max_bytes y period establecen un límite de bytes por periodo en milisegundos. No es una tasa lineal: 1024 B/1000 ms significa que hasta 1 KiB queda disponible en cada temporizador de un segundo, no que se transmita de forma continua durante el segundo. Reducir period permite inyectar entropía con más frecuencia.

El límite predeterminado es 1024 bytes cada 1000 ms -1 KiB/s-. El manual recomienda mantener siempre un limitador para evitar que un invitado consuma demasiados recursos del host. max_bytes=0 deshabilita los límites.

10.2.16 Virtiofs

Virtiofs es un sistema de archivos compartido diseñado para entornos virtuales. Permite compartir con la VM un árbol de directorios disponible en el anfitrión y montarlo dentro del invitado. No utiliza la pila de red y busca ofrecer rendimiento y semántica similares al sistema de archivos de origen.

Para utilizarlo debe ejecutarse virtiofsd en segundo plano; Proxmox VE lo inicia automáticamente al arrancar una VM que tenga un montaje virtiofs. Las VMs Linux con kernel >= 5.4 lo soportan de forma predeterminada, aunque algunas funciones necesitan kernels posteriores.

En el anfitrión debe estar instalado virtiofsd:

```
apt install virtiofsd
```

Limitaciones conocidas

- Si virtiofsd se bloquea, el punto de montaje puede quedar colgado dentro de la VM hasta que esta se detenga completamente.
- Si virtiofsd deja de responder, puede producirse un montaje bloqueado similar al de un NFS inaccesible.
- El hot-plug de memoria no funciona junto con virtiofs y también puede provocar accesos bloqueados.
- Las funciones relacionadas con guardar memoria -migración en vivo, snapshots e hibernación- no están disponibles con dispositivos virtiofs.
- Windows no interpreta ACL en el contexto de virtiofs. No deben exponerse ACL a VMs Windows, o el dispositivo virtiofs puede no aparecer dentro de la VM.

Crear un mapping para directorios compartidos

El mapeo puede definirse a través de la API usando pvesh, del mismo modo que en Resource Mapping:

```
pvesh create /cluster/mapping/dir --id dir1 \
--map node=node1,path=/path/to/share1 \
--map node=node2,path=/path/to/share2
```

Añadir virtiofs a una VM

Para compartir un directorio, añada virtiofs<N> -N entre 0 y 9- a la configuración y utilice un dirid previamente definido en Resource Mapping. cache puede ser always, never, metadata o auto; el valor predeterminado es auto.

virtiofsd puede exponer ACL y atributos extendidos mediante expose-acl y expose-xattr, siempre que el sistema de archivos del host los soporte y sean compatibles con el invitado. expose-acl implica automáticamente expose-xattr.

direct-io=1 hace que virtiofs respete O_DIRECT. Puede reducir el rendimiento, pero es útil cuando las aplicaciones realizan su propia caché.

```
qm set <vmid> -virtiofs0 dirid=<dirid>,cache=always,direct-io=1
qm set <vmid> -virtiofs1 <dirid>,cache=never,expose-xattr=1
qm set <vmid> -virtiofs2 <dirid>,expose-acl=1
```

Montaje temporal en un invitado Linux:

```
mount -t virtiofs <dirid> <mount point>
```

Para un montaje persistente, añada una entrada fstab:

```
<dirid> <mount point> virtiofs rw,relatime 0 0
```

El dirid asociado a la ruta del nodo actual se utiliza también como mount tag, es decir, como nombre con el que el dispositivo se monta dentro del invitado.

10.2.17 Orden de arranque de dispositivos

QEMU puede indicar al invitado desde qué dispositivos debe intentar arrancar y en qué orden. Ejemplo:

```
boot: order=scsi0;net0;hostpci0
```

El invitado intentaría primero scsi0, después arranque por red mediante net0 y, si también falla, el dispositivo PCIe pasado -como disco en el caso de NVMe o intentando ejecutar una option ROM en otros casos-.

En la GUI hay un editor drag-and-drop para ordenar los dispositivos y casillas para habilitar o deshabilitar cada uno como arrancable.

NOTA: Si el sistema invitado utiliza varios discos para arrancar o cargar el bootloader, todos deben estar marcados como bootables. Versiones recientes de SeaBIOS y OVMF solo inicializan discos marcados como arrancables. Los dispositivos no incluidos en el orden siguen estando disponibles una vez que el sistema operativo ha arrancado; la marca solo afecta a BIOS/firmware y bootloader.

10.2.18 Inicio y apagado automáticos de máquinas virtuales

Para que una VM se inicie automáticamente cuando arranca el host, active Start at boot en Options o use:

```
# qm set <vmid> -onboot 1
```

Orden de inicio y apagado

Puede ser necesario afinar el orden, por ejemplo si una VM proporciona firewall o DHCP a otras. Se utilizan estos parámetros:

- Start/Shutdown order: prioridad de inicio. Un valor 1 hace que la VM arranque antes. Para apagar se utiliza el orden inverso, por lo que una VM con orden 1 será de las últimas en detenerse. Si varias VMs comparten el mismo valor en un host, se ordenan además por VMID ascendente.
- Startup delay: intervalo en segundos entre el inicio de esta VM y el de las siguientes. Un valor 240 espera cuatro minutos antes de arrancar las posteriores.
- Shutdown timeout: segundos que Proxmox VE espera a que la VM se apague después de enviar la orden. El valor predeterminado es 180; si pasado ese tiempo sigue encendida, se detiene por la fuerza.

NOTA: Las VMs administradas por la pila HA no siguen actualmente Start at boot ni el orden de arranque. El gestor HA se encarga de iniciarlas y detenerlas.

Las máquinas sin Start/Shutdown order arrancan después de las que sí lo tienen. Este orden solo puede imponerse entre VMs que se ejecutan en el mismo host, no de forma global en todo el clúster. Para retrasar el arranque de la primera VM respecto al arranque del host, consulte Proxmox VE Node Management.

10.2.19 QEMU Guest Agent

QEMU Guest Agent es un servicio que se ejecuta dentro de la VM y proporciona un canal de comunicación entre anfitrión e invitado. Permite intercambiar información y ejecutar órdenes desde el host. Por ejemplo, las direcciones IP mostradas en el resumen de la VM se obtienen a través del agente.

También interviene en copias de seguridad, donde el invitado puede sincronizar escrituras pendientes mediante guest-fsfreeze-freeze y guest-fsfreeze-thaw para mejorar la consistencia.

Para que funcione correctamente:

- Instale el agente dentro del invitado y asegúrese de que está en ejecución.
- Habilite la comunicación con el agente en Proxmox VE.

Instalación del agente

En la mayoría de distribuciones Linux el paquete se llama qemu-guest-agent. En Windows puede instalarse desde la ISO de controladores VirtIO de Fedora.

Habilitar la comunicación

Se activa desde Options de la VM. Es necesario un arranque nuevo de la máquina para que el cambio tenga efecto.

TRIM automático mediante QGA

Puede habilitarse Run guest-trim. Proxmox VE enviará una orden trim al invitado después de operaciones que pueden escribir ceros en el almacenamiento, especialmente:

- Mover un disco a otro almacenamiento.
- Migrar en vivo una VM a otro nodo cuando utiliza almacenamiento local.

En almacenamiento thin-provisioned esto ayuda a liberar espacio no utilizado.

NOTA: Ext4 en Linux utiliza una optimización en memoria para evitar solicitudes TRIM duplicadas. Como el invitado no conoce el cambio de almacenamiento subyacente, solo el primer guest-trim se ejecuta completamente; los siguientes, hasta reiniciar, solo consideran zonas del sistema de archivos modificadas desde entonces.

Freeze y thaw del sistema de archivos

Por defecto, si QEMU Guest Agent está habilitado en la configuración y disponible dentro del invitado, Proxmox VE sincroniza los sistemas de archivos mediante guest-fsfreeze-freeze antes de ciertas operaciones para mejorar la consistencia.

Se realiza freeze en estas situaciones:

- Copia de seguridad en modo snapshot.
- Clonado de una VM mientras está en ejecución.
- Replicación de una VM mientras está en ejecución.
- Creación de un snapshot sin RAM de una VM en ejecución.
- Importación de una imagen de disco desde un invitado en ejecución.

En Windows, algunas aplicaciones administran su propia consistencia mediante VSS. En esos casos el freeze puede interferir. El manual cita como ejemplo que algunos SQL Server, al recibir guest-fsfreeze-freeze, pueden activar SQL Writer VSS de un modo que rompe la cadena de copias diferenciales.

Hay dos formas de manejarlo:

19. Configurar QEMU Guest Agent para utilizar una variante de VSS distinta que no interfiera con otros usuarios de VSS. La wiki de Proxmox VE contiene detalles.
20. Como alternativa, impedir que Proxmox VE ejecute el ciclo freeze/thaw deshabilitando la opción freeze-fs del agente. En la GUI: Options -> QEMU Guest Agent -> deshabilitar Freeze/thaw guest filesystems during certain operations for consistency.

IMPORTANTE: Deshabilitar freeze-fs puede producir copias de seguridad con sistemas de archivos inconsistentes. La opción preferida es adaptar la configuración del QEMU Guest Agent dentro del invitado.

Resolución de problemas: la VM no se apaga

Compruebe que el agente está instalado y ejecutándose. Cuando QEMU Guest Agent está habilitado, Proxmox VE envía órdenes de energía, como shutdown, a través del agente. Si el servicio no está funcionando, la orden no puede ejecutarse correctamente y terminará agotando el tiempo de espera.

10.2.20 Mejoras SPICE

SPICE Enhancements son funciones opcionales que pueden mejorar la experiencia con el visor remoto. Se habilitan desde Options o con:

```
qm set <vmid> -spice_enhancements foldersharing=1,videostreaming=all
```

NOTA: Para utilizar estas funciones, Display debe estar configurado como SPICE (qxl).

Compartición de carpetas

Permite compartir con el invitado una carpeta local del cliente. Dentro del invitado debe estar instalado spice-webdavd, que expone la carpeta mediante un servidor WebDAV local en http://localhost:9843. En Windows, el instalador de Spice WebDAV puede descargarse desde el sitio oficial de SPICE; muchas distribuciones Linux ofrecen el paquete spice-webdavd.

En Virt-Viewer (Remote Viewer), abra File -> Preferences, elija la carpeta que se desea compartir y active la casilla correspondiente.

NOTA: La compartición de carpetas funciona actualmente solo en la versión Linux de Virt-Viewer.

PRECAUCIÓN: Función experimental: el propio manual indica que actualmente no funciona de forma totalmente fiable.

Video streaming

Las zonas que se actualizan rápidamente pueden codificarse como flujo de vídeo. Hay dos opciones:

- all: cualquier área con refresco rápido se codifica como vídeo.
- filter: se aplican filtros adicionales para decidir cuándo utilizar vídeo; actualmente se omiten superficies de ventana pequeñas.

El manual no establece una recomendación universal sobre si debe activarse ni qué modo elegir; el resultado depende del escenario concreto.

Resolución de problemas: no aparece la carpeta compartida

Compruebe que el servicio WebDAV está habilitado y en ejecución dentro del invitado. En Windows se denomina Spice webdav proxy; en Linux suele llamarse spice-webdavd, aunque puede variar según la distribución.

Si el servicio está activo, abra http://localhost:9843 desde un navegador dentro de la VM para comprobar el servidor WebDAV. Reiniciar la sesión SPICE también puede ayudar.

NOTA: La Parte 10A termina aquí, justo antes de la sección 10.3 Migración. La continuación del capítulo se preparará en la siguiente entrega.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 10B

Capítulo 10 · Máquinas virtuales QEMU/KVM
Migración, clonación, importación y Cloud-Init (10.3–10.8.7)

Traducción de trabajo basada en "Proxmox VE Administration Guide", versión 9.2.4
Se conservan comandos, rutas, nombres de opciones y terminología técnica del original.

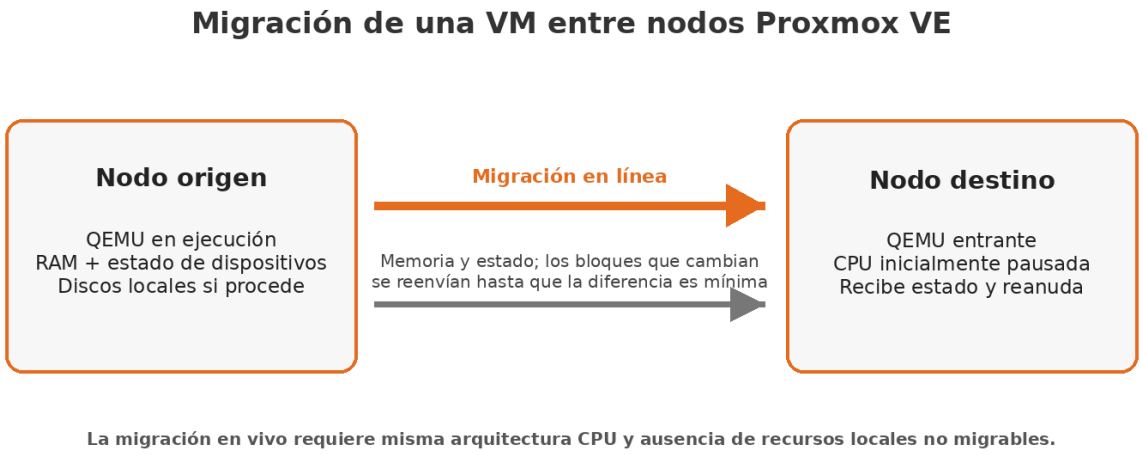
10.3 Migración

Si dispone de un clúster, puede migrar una máquina virtual a otro host con:

```
# qm migrate <vmid> <target>
```

Existen, en términos generales, dos mecanismos:

- Migración en línea (Online Migration o Live Migration).
- Migración fuera de línea (Offline Migration).



Esquema conceptual de la migración de una VM entre nodos Proxmox VE.

10.3.1 Migración en línea

Si la VM está en ejecución y no tiene configurados recursos vinculados localmente -por ejemplo, dispositivos pasados directamente al invitado-, puede iniciar una migración en vivo utilizando el indicador --online en el comando qm migrate. Cuando la VM está encendida, la interfaz web utiliza por defecto la migración en vivo.

NOTA: La migración en vivo solo es posible entre hosts con la misma arquitectura de CPU. El estado de una CPU en ejecución no puede transferirse entre un host x86 y uno arm64.

Cómo funciona

La migración en línea inicia primero un nuevo proceso QEMU en el host de destino con el indicador incoming. Ese proceso realiza únicamente la inicialización básica, mantiene las vCPU del invitado en pausa y espera los flujos de memoria y del estado de los dispositivos procedentes de la máquina virtual de origen.

Los demás recursos, como los discos, están compartidos o ya se han transferido antes de comenzar la migración del estado de ejecución. Por tanto, en ese momento quedan por enviar principalmente la memoria y el estado de los dispositivos.

Una vez establecida la conexión, el origen comienza a enviar de forma asíncrona el contenido de la memoria al destino. Si la memoria del invitado cambia durante el proceso, esas zonas se marcan como modificadas (dirty) y se realiza otra pasada para reenviarlas. El ciclo se repite hasta que la diferencia entre la VM que sigue ejecutándose en el origen y la VM entrante en el destino es suficientemente pequeña como para transmitirse en unos pocos milisegundos. Entonces la VM de origen se pausa completamente durante un instante, se envían los datos restantes y se reanudan las CPU de la VM del destino.

Requisitos

- La VM no debe tener recursos locales que no puedan migrarse. Por ejemplo, los dispositivos PCI o USB pasados directamente al invitado bloquean actualmente la migración en vivo. Los discos locales sí pueden migrarse enviándolos al destino.
- Los hosts deben pertenecer al mismo clúster de Proxmox VE.
- Debe existir una conexión de red funcional y fiable entre los hosts.
- El host de destino debe tener la misma versión o una versión superior de los paquetes de Proxmox VE. Aunque a veces puede funcionar en sentido contrario, no está garantizado.
- Los hosts deben utilizar CPU del mismo fabricante y con capacidades similares. Mezclar fabricantes puede funcionar dependiendo de los modelos reales y del tipo de CPU configurado para la VM, pero no está garantizado; debe probarse antes de usarlo en producción.

Migración del estado conntrack

> **NOTA:** La migración del estado conntrack se considera solo de mejor esfuerzo (best-effort) y puede no funcionar, ya que depende en gran medida del diseño de red. Por ejemplo, las configuraciones que usan SNAT/masquerading de origen en el host probablemente no funcionen.

Conntrack es un mecanismo del kernel de Linux que permite un firewall con estado mediante el seguimiento de conexiones individuales. Durante una migración en vivo, las conexiones activas de entrada o salida podrían interrumpirse cuando la VM empiece a ejecutarse en el destino, porque el nuevo nodo no dispone de las mismas entradas conntrack y el firewall podría descartar paquetes.

La migración del estado conntrack copia al nodo de destino las entradas conntrack del host asociadas con la VM migrada y, posteriormente, elimina del origen las entradas ya transferidas.

10.3.2 Migración fuera de línea

Si la VM utiliza recursos locales, todavía puede migrarse fuera de línea siempre que todos los discos se encuentren en un almacenamiento definido en ambos hosts. La migración copiará los discos al host de destino a través de la red, de forma similar a la migración en línea.

> **NOTA:** Cualquier configuración de passthrough de hardware puede necesitar adaptarse a la ubicación real del dispositivo en el host de destino.

10.4 Copias y clones

La instalación de una VM suele realizarse desde un medio de instalación del proveedor del sistema operativo. Dependiendo del sistema, puede ser un proceso lento que interese evitar al desplegar muchas máquinas del mismo tipo.

Una forma sencilla de desplegar varias VMs similares es copiar una máquina virtual existente. Proxmox VE denomina clone a estas copias y distingue entre clones completos y clones enlazados.

Clon completo

El resultado es una VM independiente. La nueva máquina no comparte recursos de almacenamiento con la original. Puede seleccionarse un almacenamiento de destino diferente, por lo que esta operación también puede utilizarse para mover una VM a otro almacenamiento. También puede cambiarse el formato de la imagen de disco si el controlador de almacenamiento admite varios formatos.

> **NOTA:** Un clon completo necesita leer y copiar todos los datos de la imagen de la VM. Normalmente es mucho más lento que crear un clon enlazado.

Algunos tipos de almacenamiento permiten copiar una instantánea concreta; por defecto se copia el estado actual de la VM. La copia final no incluye las instantáneas adicionales de la VM original.

Clon enlazado

Los controladores de almacenamiento modernos pueden generar clones enlazados de forma muy rápida. Se trata de una copia escribible cuyo contenido inicial coincide con el original. Su creación es casi instantánea y, inicialmente, no consume espacio adicional.

Se denominan enlazados porque la nueva imagen sigue haciendo referencia a la original. Los bloques no modificados se leen de la imagen original, mientras que los cambios se escriben -y después se leen- desde una nueva ubicación. Esta técnica se denomina Copy-on-write.

Para ello, el volumen original debe ser de solo lectura. Proxmox VE permite convertir una VM en una plantilla de solo lectura, que puede utilizarse después como base para clones enlazados eficientes.

> **NOTA:** No puede eliminarse una plantilla original mientras existan clones enlazados. Tampoco puede cambiarse el almacenamiento de destino de un clon enlazado, ya que se trata de una función interna del almacenamiento.

La opción Target node permite crear la nueva VM en un nodo diferente. La restricción es que la VM debe encontrarse en almacenamiento compartido y dicho almacenamiento debe estar disponible también en el nodo de destino.

Para evitar conflictos de recursos, se generan aleatoriamente las direcciones MAC de todas las interfaces de red y se crea un nuevo UUID para la configuración BIOS de la VM (smbios1).



Diferencia conceptual entre un clon completo y un clon enlazado.

10.5 Plantillas de máquinas virtuales

Una VM puede convertirse en una plantilla. Las plantillas son de solo lectura y pueden utilizarse para crear clones enlazados.

NOTA: No es posible iniciar una plantilla, porque hacerlo modificaría sus imágenes de disco. Si necesita cambiarla, cree un clon enlazado y modifíquelo.

10.6 VM Generation ID

Proxmox VE admite Virtual Machine Generation ID (vmgenid) para máquinas virtuales. El sistema operativo invitado puede usarlo para detectar eventos que impliquen un cambio de generación o un salto temporal, por ejemplo al restaurar una copia de seguridad o revertir una instantánea.

Al crear nuevas VMs se genera automáticamente un vmgenid y se guarda en su archivo de configuración. Para añadirlo a una VM existente puede usar el valor especial 1, que hace que Proxmox VE genere uno automáticamente, o especificar manualmente un UUID:

```
# qm set VMID -vmgenid 1
# qm set VMID -vmgenid 00000000-0000-0000-0000-000000000000
```

NOTA: La primera incorporación de un dispositivo vmgenid a una VM existente puede ser interpretada por el invitado como un cambio de generación, con efectos similares a los de una reversión de snapshot, una restauración de backup u operaciones equivalentes.

En el caso poco habitual de que no se desee vmgenid, puede usarse 0 al crear la VM o eliminar después la propiedad de la configuración:

```
# qm set VMID -delete vmgenid
```

Uno de los casos de uso más destacados son las versiones modernas de Microsoft Windows, que utilizan vmgenid para evitar problemas en servicios sensibles al tiempo o replicados -por ejemplo, bases de datos o controladores de dominio- cuando se revierten snapshots, se restauran copias de seguridad o se clona una VM completa.

10.7 Importación de máquinas virtuales

Las máquinas virtuales existentes en otros hipervisores o en otros clústeres de Proxmox VE pueden importarse mediante varios métodos. Los más habituales son:

- Utilizar el asistente nativo de importación, que aprovecha el tipo de contenido import proporcionado, por ejemplo, por el almacenamiento especial para ESXi.
- Realizar una copia de seguridad en el origen y restaurarla en el destino. Este método funciona especialmente bien al migrar desde otra instancia de Proxmox VE.
- Utilizar el comando de importación específico para OVF del comando qm.

Si importa VMs desde otros hipervisores, se recomienda familiarizarse previamente con los conceptos de Proxmox VE.

10.7.1 Asistente de importación

Proxmox VE incluye un importador de máquinas virtuales integrado en el sistema de plugins de almacenamiento. De este modo, la funcionalidad se integra de forma nativa en la API y en la interfaz web. Puede importar una VM completa, traduciendo gran parte de su configuración al modelo de configuración de Proxmox VE y reduciendo el tiempo de parada.

> **NOTA:** El asistente de importación se añadió durante el ciclo de desarrollo de Proxmox VE 8.2 y se considera una función en estado de vista previa tecnológica (tech preview). Aunque ya es prometedor y funciona de forma estable, continúa en desarrollo activo.

Para utilizarlo, primero debe configurarse un nuevo almacenamiento que actúe como fuente de importación desde Datacenter → Storage → Add. Después puede seleccionarse ese almacenamiento en el árbol de recursos y utilizar la pestaña Virtual Guests para ver los invitados disponibles para importar.

Seleccione un invitado y pulse Import -o haga doble clic- para abrir el asistente. En él puede modificar un subconjunto de las opciones disponibles y comenzar la importación. Los ajustes más avanzados pueden modificarse después de finalizar el proceso.

> **CONSEJO:** El asistente de importación de ESXi se ha probado con versiones ESXi 6.5 a 8.0. Los invitados almacenados en vSAN no pueden importarse directamente: sus discos deben moverse primero a otro almacenamiento. Es posible utilizar vCenter como origen, pero el rendimiento se degrada notablemente (entre 5 y 10 veces más lento).

Importación OVA/OVF

Para importar archivos OVA/OVF necesita primero un almacenamiento basado en archivos que admita el tipo de contenido import. En ese almacenamiento existirá una carpeta import donde puede colocar archivos OVA o archivos OVF junto con sus imágenes correspondientes, en una estructura plana. Como alternativa, la interfaz web permite cargar o descargar directamente archivos OVA.

Para los archivos OVA se necesita espacio adicional para extraer temporalmente la imagen. Este espacio debe estar en un almacenamiento basado en archivos con el tipo de contenido images habilitado. Por defecto se utiliza el almacenamiento de origen, aunque puede especificarse un Import Working Storage para extraer allí las imágenes antes de importarlas al almacenamiento de destino definitivo.

> **NOTA:** La estructura y el contenido de archivos OVA/OVF no siempre están bien definidos o mantenidos. Puede ser necesario adaptar manualmente algunos parámetros del invitado. Por ejemplo, el tipo de controlador SCSI rara vez está definido en OVA/OVF, y el valor por defecto puede no arrancar con OVMF (UEFI); en esos casos debe seleccionarse VirtIO SCSI o VMware PVSCSI.

10.7.2 Importación OVF/OVA desde CLI

Una exportación de VM desde un hipervisor externo suele consistir en una o varias imágenes de disco acompañadas por un archivo de configuración que describe parámetros como RAM o número de núcleos.

Las imágenes pueden estar en formato vmdk cuando proceden de VMware o VirtualBox, o en qcow2 si proceden de un hipervisor KVM. OVF es uno de los formatos de configuración más utilizados, pero en la práctica la interoperabilidad es limitada porque muchos parámetros no forman parte del estándar y los hipervisores incorporan información adicional mediante extensiones no estándar.

Además del formato, la importación puede fallar si el hardware emulado cambia demasiado entre un hipervisor y otro. Las VMs Windows son especialmente sensibles a esos cambios. El manual señala que este problema puede mitigarse instalando antes de la exportación la utilidad MergeIDE.zip y seleccionando un disco IDE para el primer arranque de la VM importada.

También deben tenerse en cuenta los controladores paravirtualizados. GNU/Linux y otros Unix libres suelen incluirlos de serie, por lo que normalmente pueden cambiarse a controladores paravirtualizados tras la importación. En Windows deben instalarse los controladores paravirtualizados correspondientes.

GNU/Linux y otros Unix libres suelen poder importarse sin grandes dificultades. El documento original advierte que no puede garantizarse el éxito de la importación/exportación de VMs Windows en todos los casos debido a los factores anteriores.

Ejemplo paso a paso de importación de un OVF de Windows

Tras descargar una VM de evaluación de Windows para plataforma VMware y descomprimir el ZIP, copie por SSH/SCP al host Proxmox VE los archivos OVF y VMDK. El siguiente comando crea una nueva VM leyendo del manifiesto OVF el número de núcleos, la memoria y el nombre, e importa sus discos al almacenamiento local-lvm. La red debe configurarse manualmente:

```
# qm importovf 999 WinDev1709Eval.ovf local-lvm
```

La VM queda preparada para arrancar.

Añadir una imagen de disco externa a una máquina virtual

También puede añadir una imagen de disco existente a una VM, tanto si procede de otro hipervisor como si la ha creado usted mismo. El manual muestra como ejemplo una imagen Debian/Ubuntu creada con vmdebootstrap:

```
vmdebootstrap --verbose \
--size 10GiB --serial-console \
--grub --no-extlinux \
--package openssh-server \
--package avahi-daemon \
--package qemu-guest-agent \
--hostname vm600 --enable-dhcp \
--customize=./copy_pub_ssh.sh \
--sparse --image vm600.raw
```

A continuación puede crearse la VM de destino, importar la imagen al almacenamiento pvedir y conectarla a su controlador SCSI:

```
# qm create 600 --net0 virtio,bridge=vmbr0 --name vm600 --serial0 socket \
--boot order=scsi0 --scsihw virtio-scsi-pci --ostype l26 \
--scsi0 pvedir:0,import-from=/path/to/dir/vm600.raw
```

La VM queda preparada para arrancar.

10.8 Compatibilidad con Cloud-Init

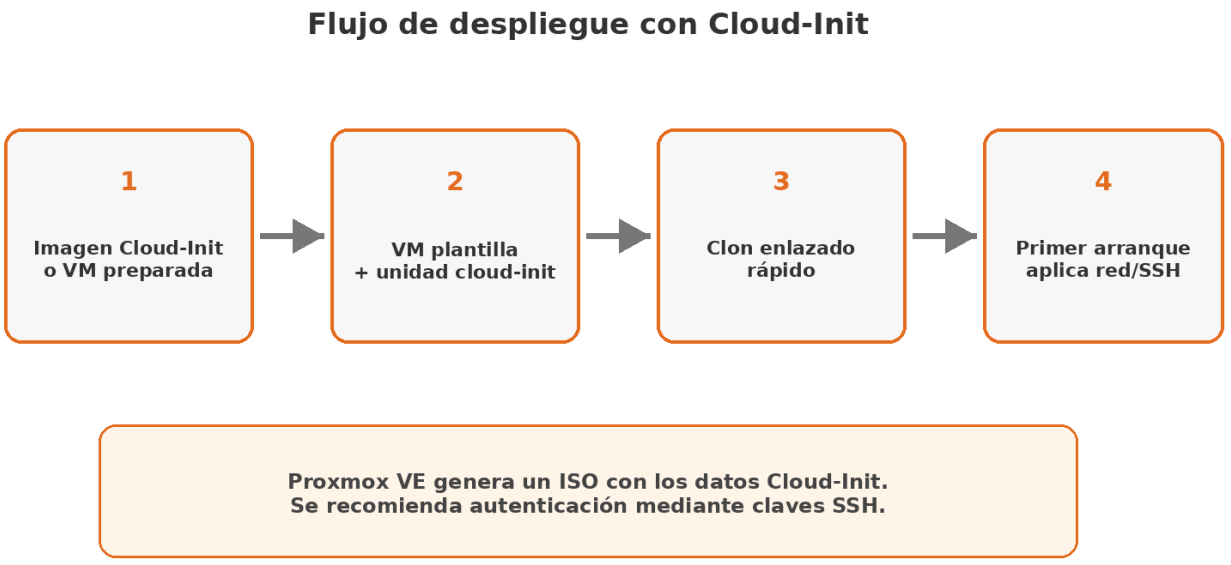
Cloud-Init es el paquete de facto, compatible con múltiples distribuciones, para gestionar la inicialización temprana de una instancia de máquina virtual. Permite configurar desde el hipervisor parámetros como dispositivos de red y claves SSH. En el primer arranque, el software Cloud-Init instalado dentro de la VM aplica esos ajustes.

Muchas distribuciones Linux publican imágenes Cloud-Init listas para usar, diseñadas principalmente para OpenStack. También funcionan con Proxmox VE. Aunque resulte cómodo utilizarlas, el manual recomienda normalmente preparar las imágenes por cuenta propia, ya que así se conoce exactamente qué software contienen y resulta más sencillo personalizarlas posteriormente.

Una vez creada una imagen Cloud-Init, se recomienda convertirla en una plantilla de VM. Desde una plantilla pueden crearse rápidamente clones enlazados; basta con configurar la red y, en su caso, las claves SSH antes del primer arranque.

Se recomienda la autenticación mediante claves SSH para acceder a las VMs aprovisionadas con Cloud-Init. También puede establecerse una contraseña, pero es menos seguro porque Proxmox VE debe almacenar una versión cifrada de esa contraseña dentro de los datos de Cloud-Init.

Proxmox VE genera una imagen ISO para transmitir los datos de Cloud-Init a la VM. Por ello, todas las VMs Cloud-Init necesitan una unidad de CD-ROM asignada. Normalmente también debe añadirse una consola serie y utilizarla como pantalla, ya que muchas imágenes Cloud-Init dependen de ella y OpenStack la requiere. Si una imagen concreta tiene problemas con esta configuración, debe volver a utilizarse la pantalla por defecto.



Flujo básico recomendado para preparar y desplegar una plantilla Cloud-Init.

10.8.1 Preparación de plantillas Cloud-Init

El primer paso es preparar la VM. En principio puede utilizarse cualquier máquina virtual; basta con instalar dentro de ella los paquetes Cloud-Init. En Debian/Ubuntu:

```
apt-get install cloud-init
```

ADVERTENCIA: Este comando no debe ejecutarse en el host Proxmox VE, sino únicamente dentro de la máquina virtual que se está preparando.

Muchas distribuciones ofrecen imágenes Cloud-Init listas para usar en formato .qcow2. El siguiente ejemplo utiliza una imagen de nube de Ubuntu:

```
# descargar la imagen
wget https://cloud-images.ubuntu.com/bionic/current/bionic-server-cloudimg-amd64.img

# crear una VM nueva con controlador VirtIO SCSI
qm create 9000 --memory 2048 --net0 virtio,bridge=vmbr0 --scsihw virtio-scsi-pci

# importar el disco al almacenamiento local-lvm y conectarlo como unidad SCSI
qm set 9000 --scsi0 local-lvm:0,import-from=/path/to/bionic-server-cloudimg-amd64.img
```

NOTA: Las imágenes Cloud-Init de Ubuntu requieren el controlador virtio-scsi-pci para las unidades SCSI.

Añadir la unidad CD-ROM de Cloud-Init

Configure una unidad CD-ROM que servirá para pasar los datos Cloud-Init a la VM:

```
qm set 9000 --ide2 local-lvm:cloudinit
```

Para arrancar directamente desde la imagen Cloud-Init, limite el orden de arranque a scsi0. Esto acelera el inicio porque la BIOS de la VM evita buscar un CD-ROM arrancable:

```
qm set 9000 --boot order=scsi0
```

Muchas imágenes Cloud-Init requieren una consola serie como pantalla. Si una imagen concreta no funciona con esta configuración, vuelva a la pantalla predeterminada:

```
qm set 9000 --serial0 socket --vga serial0
```

Por último, resulta práctico convertir la VM en plantilla. Desde ella pueden crearse clones enlazados mucho más rápidamente que mediante una copia completa:

```
qm template 9000
```

10.8.2 Despliegue de plantillas Cloud-Init

Una plantilla puede desplegarse fácilmente mediante clonación:

```
qm clone 9000 123 --name ubuntu2
```

Después configure la clave pública SSH para autenticación y los parámetros IP:

```
qm set 123 --sshkey ~/.ssh/id_rsa.pub
qm set 123 --ipconfig0 ip=10.0.10.123/24,gw=10.0.10.1
```

También es posible configurar todas las opciones Cloud-Init en un único comando. El ejemplo las separa únicamente para mantener las líneas legibles. La configuración IP debe adaptarse al entorno real.

10.8.3 Configuración Cloud-Init personalizada

La integración de Cloud-Init permite utilizar archivos de configuración personalizados en lugar de los generados automáticamente. Se hace mediante la opción cicustom:

```
qm set 9000 --cicustom "user=<volume>,network=<volume>,meta=<volume>"
```

Los archivos personalizados deben encontrarse en un almacenamiento compatible con snippets y estar disponibles en todos los nodos a los que pueda migrarse la VM. De lo contrario, la VM no podrá arrancar. Ejemplo:

```
qm set 9000 --cicustom "user=local:snippets/userconfig.yaml"
```

Cloud-Init maneja tres clases principales de configuración: user, network y meta. Pueden especificarse conjuntamente o combinarse según las necesidades. Para cualquier tipo no sustituido por un archivo personalizado se seguirá utilizando la configuración generada automáticamente.

La configuración generada puede volcarse para usarla como base de un archivo personalizado:

```
qm cloudinit dump 9000 user
```

El mismo comando está disponible para network y meta.

10.8.4 Cloud-Init en Windows

Para Windows existe una reimplementación de Cloud-Init denominada Cloudbase-Init. No todas las funciones de Cloud-Init están disponibles en Cloudbase-Init y algunas se comportan de forma distinta.

Cloudbase-Init requiere que ostype esté configurado con alguna versión de Windows y que citype sea configdrive2, que es el valor predeterminado para cualquier ostype de Windows.

No existen imágenes cloud de Windows listas para usar y disponibles gratuitamente. Por ello, Cloudbase-Init requiere instalar y configurar manualmente un invitado Windows.

10.8.5 Preparación de plantillas Cloudbase-Init

El primer paso consiste en instalar Windows en una VM y, dentro del invitado, instalar Cloudbase-Init. Puede ser necesario utilizar la versión Beta. No ejecute Sysprep al finalizar la instalación; configure primero Cloudbase-Init.

Algunas opciones habituales son:

Opción	Uso
username	Define el nombre del usuario administrador.
groups	Permite añadir el usuario al grupo Administrators.
inject_user_password	Debe establecerse en true para permitir definir la contraseña desde la configuración de la VM.
first_logon_behaviour	Puede establecerse en no para no exigir una nueva contraseña al iniciar sesión.
rename_admin_user	Permite renombrar el usuario Administrator predeterminado con el valor de username.
metadata_services	Puede apuntar a cloudbaseinit.metadata.services.configdrive.ConfigDriveService para que Cloudbase-Init compruebe primero este servicio.
allow_reboot	Puede establecerse en false para impedir reinicios automáticos realizados por Cloudbase-Init.

Algunos plugins, como SetHostnamePlugin, necesitan reinicios y los realizarán automáticamente salvo que se desactive con allow_reboot=false.

Puede ser útil crear una instantánea después de configurar Cloudbase-Init, por si más adelante fuese necesario ajustar algún parámetro. Después, apague el invitado Windows, añada un disco Cloud-Init y conviértalo en plantilla:

```
qm set 9000 --ide2 local-lvm:cloudinit
qm template 9000
```

Clone la plantilla:

```
qm clone 9000 123 --name windows123
```

Después defina contraseña, red y clave SSH:

```
qm set 123 --cipassword <password>
qm set 123 --ipconfig0 ip=10.0.10.123/24,gw=10.0.10.1
qm set 123 --sshkey ~/.ssh/id_rsa.pub
```

IMPORTANTE: Compruebe que ostype está configurado con alguna versión de Windows antes de establecer la contraseña. En caso contrario, la contraseña se cifrará y Cloudbase-Init utilizará como contraseña en texto plano la cadena cifrada.

Una vez completada la configuración, inicie el clon. En el primer arranque el inicio de sesión no funcionará y el sistema se reiniciará automáticamente debido al cambio de hostname. Tras ese reinicio, la nueva contraseña debería quedar aplicada y el acceso funcionar.

10.8.6 Cloudbase-Init y Sysprep

Sysprep permite restablecer la configuración de Windows y preparar un sistema nuevo. Puede utilizarse junto con Cloudbase-Init para crear una plantilla limpia.

Con Sysprep deben adaptarse dos archivos de configuración: el archivo principal y el que termina en -unattend.conf. Cloudbase-Init se ejecuta en dos fases: primero la fase de Sysprep utilizando -unattend.conf y después la fase normal usando el archivo de configuración principal.

En Windows Server, ejecutar Sysprep con el archivo Unattend.xml proporcionado debería funcionar directamente. Las ediciones normales de Windows requieren pasos adicionales:

21. Abra una instancia de PowerShell.
22. Habilite el usuario Administrator con: net user Administrator /active:yes
23. Instale Cloudbase-Init utilizando el usuario Administrator.
24. Modifique Unattend.xml para incluir un comando que habilite el usuario Administrator durante el primer arranque después de Sysprep. Asegúrese de que el valor <Order> no entre en conflicto con otros comandos y aumente el orden del comando Cloudbase-Init para que se ejecute después.
25. Solo en Windows 11: elimine el paquete conflictivo Microsoft.OneDriveSync.
26. Acceda al directorio de configuración de Cloudbase-Init: C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf
27. Opcionalmente, cree una instantánea de la VM antes de ejecutar Sysprep por si existiese alguna configuración incorrecta.
28. Ejecute Sysprep con /generalize /oobe y el archivo Unattend.xml.

```
<RunSynchronousCommand wcm:action="add">
    <Path>net user administrator /active:yes</Path>
    <Order>1</Order>
    <Description>Enable Administrator User</Description>
</RunSynchronousCommand>
```

```
Get-AppxPackage -AllUsers Microsoft.OneDriveSync | Remove-AppxPackage -AllUsers
```

```
cd 'C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf'
```

```
C:\Windows\System32\Sysprep\sysprep.exe /generalize /oobe /unattend:Unattend.xml
```

Después de estos pasos, la VM debería quedar apagada como consecuencia de Sysprep. A partir de ahí puede convertirse en plantilla, clonarse y configurarse según las necesidades.

10.8.7 Opciones específicas de Cloud-Init

Las siguientes opciones aparecen en la configuración de Proxmox VE para Cloud-Init:

Opción	Descripción
cicustom: [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>]	Permite sustituir los archivos generados automáticamente al arrancar por archivos personalizados.
meta=<volume>	Archivo personalizado con los metadatos enviados a la VM. Es específico del proveedor: configdrive2 y nocloud utilizan formatos distintos.
network=<volume>	Archivo personalizado con los datos de red que se entregan a la VM mediante Cloud-Init.
user=<volume>	Archivo personalizado con los datos de usuario.
vendor=<volume>	Archivo personalizado con los datos del proveedor (vendor data).
cipassword: <string>	Contraseña que se asignará al usuario. En general no se recomienda; es preferible utilizar claves SSH. Versiones antiguas de cloud-init no admiten contraseñas hash.
citype: <configdrive2 | nocloud | opennebula>	Formato de configuración Cloud-Init. El valor por defecto depende de ostype: nocloud para Linux y configdrive2 para Windows.
ciupgrade: <boolean> (default = 1)	Realiza una actualización automática de paquetes después del primer arranque.
ciuser: <string>	Nombre de usuario al que aplicar claves SSH y contraseña en lugar del usuario predeterminado de la imagen.
ipconfig[n]	Define direcciones IP y gateways de la interfaz correspondiente. Las IP usan notación CIDR; los gateways son opcionales, pero deben ser del mismo tipo de IP.

NOTA: Esta Parte 10B finaliza en el apartado 10.8.7. La siguiente parte continuará con 10.9 PCI(e) Passthrough y el resto del Capítulo 10.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 10C

PCI(e) Passthrough · SR-IOV · vGPU · vIOMMU · Hookscripts · Hibernación · Resource Mapping

Traducción no oficial del manual Proxmox VE Administration Guide 9.2.4 · GFDL 1.3 o posterior

Capítulo 10 · Máquinas virtuales QEMU/KVM

Esta parte continúa exactamente donde terminó la Parte 10B y completa el final de las opciones específicas de Cloud-Init antes de entrar en PCI(e) Passthrough. El bloque termina en 10.12 Resource Mapping, justo antes de 10.13 Managing Virtual Machines with qm.

10.8.7 Opciones específicas de Cloud-Init — continuación

Para las direcciones IP pueden usarse valores especiales. La cadena dhcp indica que debe utilizarse DHCP; en ese caso no debe indicarse una puerta de enlace explícita. Para IPv6 puede utilizarse auto para autoconfiguración sin estado. Esta última opción requiere cloud-init 19.4 o posterior.

Si Cloud-Init está habilitado y no se especifica ninguna dirección IPv4 ni IPv6, por defecto se utiliza DHCP para IPv4.

gw=<GatewayIPv4> — Puerta de enlace predeterminada para el tráfico IPv4. Requiere que se haya definido la opción ip.

gw6=<GatewayIPv6> — Puerta de enlace predeterminada para el tráfico IPv6. Requiere que se haya definido la opción ip6.

ip=<IPv4Format/CIDR> (predeterminado = dhcp) — Dirección IPv4 en notación CIDR.

ip6=<IPv6Format/CIDR> (predeterminado = dhcp) — Dirección IPv6 en notación CIDR.

nameserver: <string> — Establece la dirección IP del servidor DNS. Si no se define ni searchdomain ni nameserver, al crear la instancia se usará automáticamente la configuración del host.

searchdomain: <string> — Establece los dominios de búsqueda DNS. Si no se define ni searchdomain ni nameserver, al crear la instancia se usará automáticamente la configuración del host.

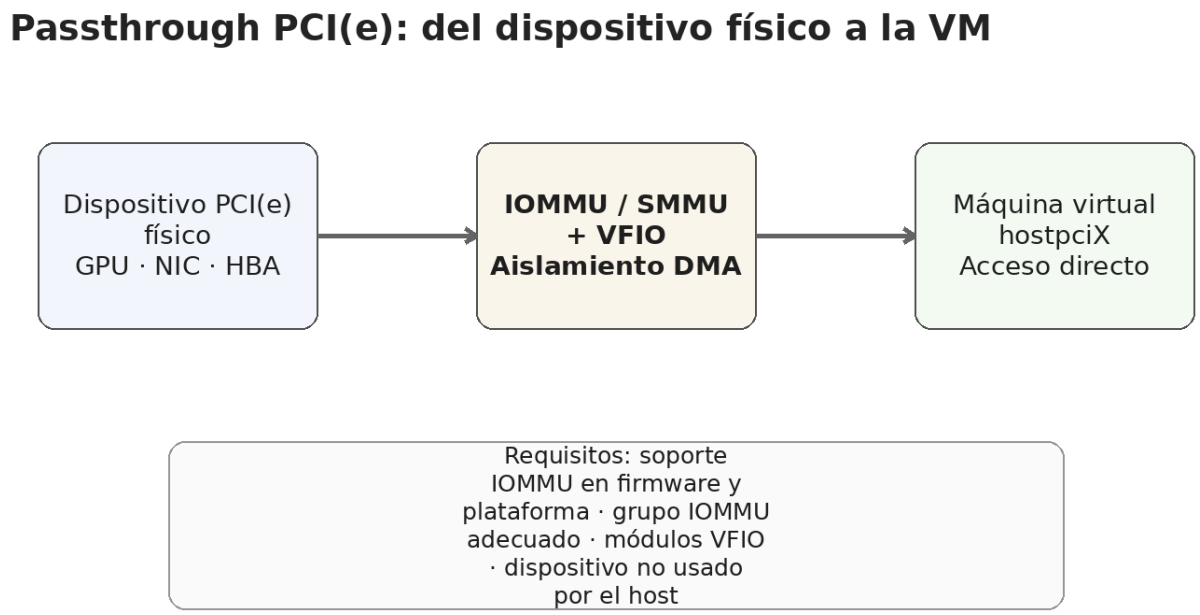
sshkeys: <string> — Configura claves públicas SSH, una por línea, en formato OpenSSH.

10.9 PCI(e) Passthrough

PCI(e) passthrough es un mecanismo que permite entregar a una máquina virtual el control directo de un dispositivo PCI del host. Frente al hardware virtualizado puede ofrecer menor latencia, mayor rendimiento o acceso a funciones específicas del dispositivo, como ciertas capacidades de offloading.

Cuando un dispositivo se asigna mediante passthrough a una VM, deja de estar disponible para el host y para las demás máquinas virtuales.

NOTA: El passthrough PCI está disponible con máquinas i440fx y q35. El passthrough PCIe solo está disponible con q35. Presentar un dispositivo PCIe al invitado como PCI no lo limita a "velocidad PCI"; principalmente cambia cómo el invitado identifica el dispositivo.



10.9.1 Requisitos generales

Hardware

El hardware debe admitir remapeo de interrupciones mediante IOMMU (I/O Memory Management Unit). Esto implica compatibilidad tanto en la CPU como en la placa base.

En sistemas Intel esta función suele asociarse a VT-d y en AMD a AMD-Vi. En arm64, el equivalente es SMMU (System Memory Management Unit), proporcionado por las plataformas compatibles. Aun así, la existencia de la función no garantiza que todo funcione sin ajustes, ya que influyen la implementación del hardware y la calidad de los controladores.

El hardware de servidor suele ofrecer mejor soporte que el de consumo. Debe consultarse al fabricante para confirmar la compatibilidad con Linux de la plataforma concreta.

Determinar la dirección de la tarjeta PCI

La forma más sencilla es utilizar la interfaz web y añadir un dispositivo de tipo Host PCI en la pestaña Hardware de la VM. También puede localizarse desde la línea de comandos:

```
lspci
```

Configuración de IOMMU

IOMMU debe estar habilitado en la BIOS/UEFI. La opción suele denominarse IOMMU o VT-d, aunque el nombre exacto depende del fabricante.

En CPUs AMD IOMMU está habilitado de forma predeterminada. Con kernels recientes (6.8 o posteriores), también ocurre así en Intel. En kernels Intel más antiguos es necesario añadir a la línea de comandos del kernel:

```
intel_iommu=on
```

NOTA: intel_iommu=on se aplica a hosts x86. En arm64, SMMU se habilita mediante firmware y ACPI, por lo que no se necesita este parámetro de kernel.

Modo IOMMU passthrough

Si el hardware admite el modo passthrough de IOMMU, activarlo puede mejorar el rendimiento. Las VMs pueden evitar parte de la traducción DMA predeterminada del hipervisor y enviar las peticiones DMA directamente al IOMMU de hardware.

```
iommu=pt
```

Módulos del kernel

Para passthrough convencional, deben cargarse los módulos VFIO siguientes. Pueden añadirse a /etc/modules:

```
vfio
vfio_iommu_type1
vfio_pci
```

NOTA: Para mediated devices, como vGPU, esta lista no es necesaria de la misma forma: el dispositivo permanece controlado directamente por el driver apropiado del host.

Después de modificar la configuración de módulos, debe regenerarse initramfs:

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

Para comprobar que VFIO se ha cargado:

```
# lsmod | grep vfio
```

Finalizar y verificar la configuración

Reinicie el host y compruebe que IOMMU, Directed I/O o Interrupt Remapping aparecen como habilitados:

```
# dmesg | grep -e DMAR -e IOMMU -e AMD-Vi
```

También es importante que el dispositivo que se pretende asignar esté en un grupo IOMMU adecuado. Puede consultarse desde la API de Proxmox VE:

```
# pvesh get /nodes/{nodename}/hardware/pci --pci-class-blacklist ""
```

Es aceptable que el dispositivo comparta grupo IOMMU con otras funciones del mismo dispositivo —por ejemplo, una GPU y su audio HDMI— o con su root port / puente PCI(e).

Slots PCI(e) y unsafe interrupts

Algunas plataformas agrupan los slots físicos de manera diferente. Si no se obtiene la separación IOMMU deseada, puede ser útil mover la tarjeta a otro slot PCI(e).

En determinadas plataformas puede ser necesario permitir interrupciones no seguras. Para ello se añade a un archivo .conf en /etc/modprobe.d/:

```
options vfio_iommu_type1 allow_unsafe_interrupts=1
```

ADVERTENCIA: Esta opción puede hacer que el sistema sea inestable.

Notas sobre GPU Passthrough

El framebuffer de una GPU pasada directamente a la VM no puede mostrarse mediante NoVNC o SPICE en la interfaz web de Proxmox VE. Si se necesita salida gráfica, debe conectarse físicamente un monitor a la tarjeta o configurarse software de escritorio remoto dentro del invitado, como VNC o RDP.

Si la GPU se utiliza solo como acelerador de cómputo —por ejemplo, OpenCL o CUDA— no es necesaria una salida gráfica.

10.9.2 Passthrough de dispositivo del host

La variante más habitual consiste en asignar una tarjeta PCI(e) completa, como una GPU o una tarjeta de red.

Configuración del host

Proxmox VE intenta dejar automáticamente el dispositivo fuera del control normal del host. Si esto no funciona, existen dos alternativas principales.

- Pasar los identificadores del dispositivo al módulo vfio-pci mediante una línea como:

```
options vfio-pci ids=1234:5678,4321:8765
```

Los IDs de fabricante y dispositivo pueden obtenerse con:

```
# lspci -nn
```

- Bloquear completamente el driver del host para que el dispositivo quede libre para VFIO:

```
blacklist DRIVERNAME
```

El nombre del driver puede consultarse con:

```
# lspci -k
# lspci -k | grep -A 3 "VGA"
```

Un resultado típico muestra el controlador, el subsistema, el driver en uso y los módulos disponibles. El driver puede añadirse a una blacklist, por ejemplo:

```
echo "blacklist <some-module>" >> /etc/modprobe.d/blacklist.conf
```

En ambos métodos hay que actualizar initramfs y reiniciar. Si aun así el dispositivo se enlaza primero al driver gráfico, puede definirse una dependencia blanda para cargar vfio-pci antes:

```
# echo "softdep <some-module> pre: vfio-pci" >> /etc/modprobe.d/<some-module>.conf
```

Verificar la configuración

Use lspci -nnk y revise la entrada del dispositivo. Si aparece Kernel driver in use: vfio-pci —o no aparece ninguna línea de driver en uso— el dispositivo está preparado para passthrough.

NOTA: En mediated devices la situación es distinta: el dispositivo puede permanecer asociado directamente al driver del host en lugar de vfio-pci.

Configuración de la VM

Para GPU passthrough, la mejor compatibilidad se obtiene normalmente con máquina q35, OVMF (UEFI) en lugar de SeaBIOS y PCIe en lugar de PCI. Con OVMF, la GPU debe disponer de una ROM compatible con UEFI; de lo contrario debe utilizarse SeaBIOS.

Con OVMF puede ser posible deshabilitar la arbitraje VGA para reducir código heredado durante el arranque. Ejemplo:

```
echo "options vfio-pci ids=<vendor-id>,<device-id> disable_vga=1" > /etc/modprobe.d/vfio.conf
```

Los dispositivos PCI pueden añadirse desde Hardware en la interfaz web o mediante hostpciX:

```
# qm set VMID -hostpci0 00:02.0
# equivalente en el archivo de configuración:
hostpci0: 00:02.0
```

Si el dispositivo tiene varias funciones, por ejemplo 00:02.0 y 00:02.1, puede utilizarse la forma abreviada 00:02 para asignarlas conjuntamente. Equivale a marcar All Functions en la GUI.

`x-vga=on|off` — Marca el dispositivo como GPU primaria de la VM. Si se activa, se ignora la opción vga.

`pcie=on|off` — Indica a Proxmox VE que utilice un puerto PCIe o PCI. PCIe solo está disponible con máquinas q35.

`rombar=on|off` — Hace visible al invitado la ROM de firmware. El valor predeterminado es on; algunos dispositivos requieren off.

`romfile=<path>` — Ruta opcional a una ROM que se utilizará para el dispositivo. La ruta es relativa a /usr/share/kvm/.

Ejemplo de GPU PCIe configurada como primaria:

```
# qm set VMID -hostpci0 02:00,pcie=on,x-vga=on
```

Sobrescritura de IDs PCI

Es posible sobrescribir los IDs de fabricante, dispositivo y subsistema que verá el invitado. Esto puede ser útil cuando una variante concreta comparte chipset con un modelo soportado por el driver del invitado, pero usa identificadores distintos.

Las opciones disponibles son vendor-id, device-id, sub-vendor-id y sub-device-id. Puede definirse una o varias:

```
# qm set VMID -hostpci0 02:00,device-id=0x10f6,sub-vendor-id=0x0000
```

10.9.3 SR-IOV

SR-IOV (Single-Root Input/Output Virtualization) permite que un único dispositivo físico exponga varias funciones virtuales o VF (Virtual Functions). Cada VF puede asignarse a una VM distinta, ofreciendo funciones de hardware, mejor rendimiento y menor latencia que una emulación puramente software.

El caso de uso más común son las tarjetas de red compatibles con SR-IOV. Una sola interfaz física puede ofrecer varias VF por puerto y permitir dentro de las VMs funciones como checksum offloading, reduciendo carga de CPU del host.

Habilitar SR-IOV

El soporte de la plataforma es especialmente importante. Puede ser necesario habilitar SR-IOV en BIOS/UEFI o utilizar un slot PCI(e) concreto. Debe consultarse la documentación del fabricante.

Configuración del host

Hay dos métodos habituales para crear funciones virtuales:

- Algunos drivers aceptan un parámetro de módulo, por ejemplo max_vfs=4, que puede guardarse en un archivo .conf bajo /etc/modprobe.d/. Después debe regenerarse initramfs.
- Un método más genérico usa sysfs. Para crear 4 VF sobre el dispositivo 0000:01:00.0:

```
# echo 4 > /sys/bus/pci/devices/0000:01:00.0/sriov_numvfs
```

Para hacer persistente este ajuste puede utilizarse el paquete Debian sysfsutils y configurarlo mediante /etc/sysfs.conf o un archivo .conf bajo /etc/sysfs.d/.

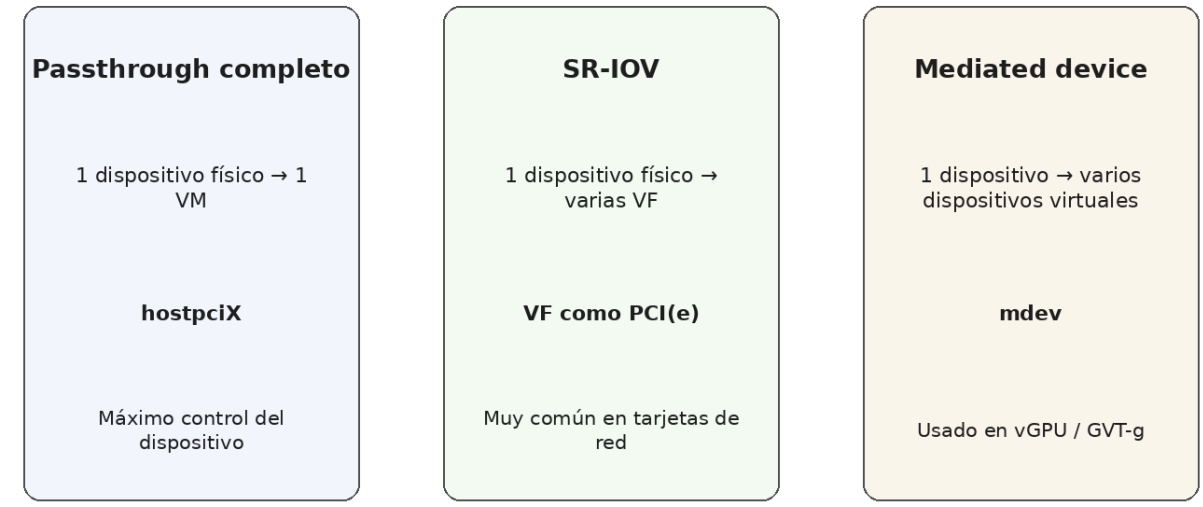
Configuración de la VM

Una vez creadas, las VF aparecen como dispositivos PCI(e) independientes en lspci. Obtenga su identificador y páselas a la VM como cualquier otro dispositivo PCI(e).

10.9.4 Mediated Devices (vGPU, GVT-g)

Los mediated devices permiten reutilizar funciones y rendimiento del hardware físico creando dispositivos virtuales gestionados por el driver del host. Son habituales en escenarios de vGPU, como Intel GVT-g y determinadas tecnologías NVIDIA GRID.

A diferencia de SR-IOV, los dispositivos mediados no aparecen necesariamente como dispositivos PCI(e) independientes del host y están orientados a su uso dentro de máquinas virtuales.



Configuración del host

El driver de la tarjeta debe soportar mediated devices. Debe consultarse al fabricante para conocer controladores y procedimiento de configuración compatibles.

En el caso de Intel GVT-g, el manual indica que los drivers están integrados en el kernel y pueden utilizarse con procesadores Intel Core de 5.ª, 6.ª y 7.ª generación, así como Xeon E3 v4, E3 v5 y E3 v6.

Para Intel Graphics debe cargarse el módulo kvmgt —por ejemplo desde /etc/modules— y añadir al kernel:

```
i915.enable_gvt=1
```

Después debe actualizarse initramfs y reiniciarse el host.

Configuración de la VM

Para usar un dispositivo mediado se especifica la propiedad mdev en una opción hostpciX. Los tipos soportados pueden consultarse en sysfs. Por ejemplo:

```
# ls /sys/bus/pci/devices/0000:00:02.0/mdev_supported_types
```

Cada directorio de tipo incluye, entre otros, available_instances —número de instancias aún disponibles—, description —descripción de capacidades— y create —endpoint utilizado para crear el dispositivo. Proxmox VE realiza la creación automáticamente cuando la VM tiene configurado mdev.

Ejemplo con una vGPU Intel GVT-g:

```
# qm set VMID -hostpci0 00:02.0,mdev=i915-GVTg_V5_4
```

Con esta configuración, Proxmox VE crea el dispositivo al iniciar la VM y lo elimina de nuevo cuando la VM se detiene.

10.9.5 Uso en clústeres

También es posible mapear dispositivos a nivel de clúster. Esto permite utilizarlos de forma más segura con HA, detectar cambios de hardware y permitir que usuarios no root los configuren. Esta función se desarrolla en 10.12 Resource Mapping.

10.9.6 vIOMMU (IOMMU emulada)

vIOMMU emula una IOMMU de hardware dentro de la propia máquina virtual. Proporciona mayor control de acceso a memoria y seguridad para dispositivos de E/S virtualizados. También permite, junto con virtualización anidada, pasar dispositivos PCI(e) a VMs de nivel 2 que se ejecutan dentro de una VM de nivel 1.

Actualmente se describen dos implementaciones: Intel y VirtIO.

Intel vIOMMU

- Es necesario definir intel_iommu=on en los parámetros del kernel de la VM, incluso si el host utiliza una CPU AMD.
- La VM debe utilizar q35 como tipo de máquina.

```
# qm set VMID -machine q35,viommu=intel
```

VirtIO vIOMMU

La implementación VirtIO es más reciente y tiene menos limitaciones que Intel vIOMMU, aunque el manual señala que está menos extendida en producción y menos documentada.

No necesita parámetros de kernel específicos. Tampoco exige q35, aunque se aconseja si se quiere utilizar PCIe:

```
# qm set VMID -machine q35,viommu=virtio
```

10.10 Hookscripts

Puede asociarse un script de hooks a una VM mediante la propiedad hookscript. El script será llamado en distintas fases del ciclo de vida del invitado.

```
# qm set 100 --hookscript local:snippets/hookscript.pl
```

Proxmox VE incluye un ejemplo y documentación en:

```
/usr/share/pve-docs/examples/guest-example-hookscript.pl
```

10.11 Hibernación

Una VM puede suspenderse a disco mediante la opción Hibernate de la GUI o con:

```
# qm suspend ID --todisk
```

El contenido actual de la memoria se guarda en disco y la VM se detiene. En el siguiente arranque, la memoria se restaura y la VM puede continuar desde el punto en que quedó.

Selección del almacenamiento del estado

Si no se especifica dónde guardar la memoria, Proxmox VE elige automáticamente el primer almacenamiento que cumpla esta prioridad:

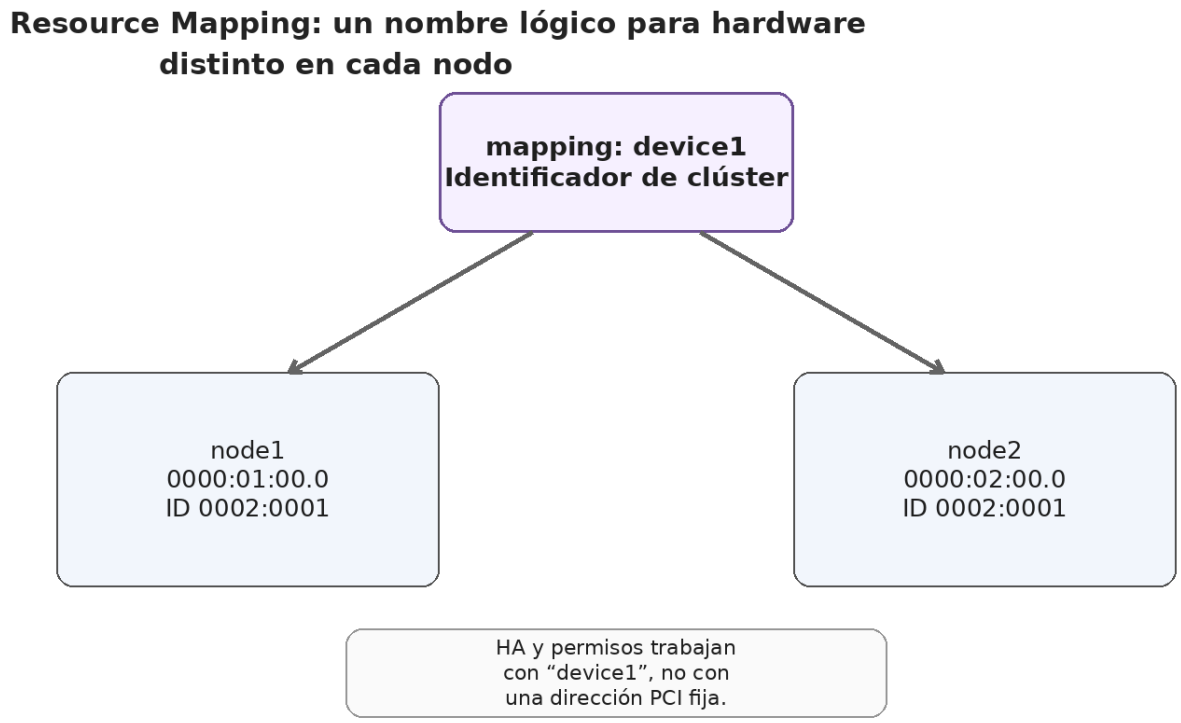
29. El almacenamiento vmstatestorage definido en la configuración de la VM.
30. El primer almacenamiento compartido de cualquiera de los discos de la VM.
31. El primer almacenamiento no compartido de cualquiera de los discos de la VM.
32. El almacenamiento local como último recurso.

10.12 Resource Mapping

Cuando se utilizan recursos locales —por ejemplo la dirección física de un dispositivo PCI—, referirse directamente a una dirección o ID puede ser problemático.

- Con HA, un nodo de destino podría contener otro dispositivo con el mismo ID o una ruta distinta; una asignación poco cuidadosa podría hacer que la VM utilice hardware equivocado.
- Un cambio de hardware puede modificar IDs y rutas, obligando a revisar manualmente todas las VMs que tengan dispositivos asignados.

Para evitarlo, Proxmox VE permite definir mapeos de recursos a nivel de clúster. Un recurso recibe un identificador único elegido por el administrador y ese nombre lógico puede corresponder a dispositivos físicos distintos en cada nodo. Así, HA puede evitar iniciar una VM con un dispositivo incorrecto y es posible detectar cambios de hardware.



Los mapeos se crean desde Datacenter, en la categoría Resource Mappings, o mediante la CLI:

```
# pvesh create /cluster/mapping/<type> <options>
```

El parámetro <type> identifica el tipo de hardware; actualmente el manual enumera pci, usb y dir. Las opciones deben incluir una propiedad map con los datos que permitan identificar de forma inequívoca el hardware y verificar que no ha cambiado.

Ejemplo: mapear como device1 un dispositivo PCI que en node1 está en 0000:01:00.0 y en node2 en 0000:02:00.0, ambos con ID de dispositivo 0001 y fabricante 0002:

```
# pvesh create /cluster/mapping/pci --id device1 \
  --map node=node1,path=0000:01:00.0,id=0002:0001 \
  --map node=node2,path=0000:02:00.0,id=0002:0001
```

La propiedad map debe repetirse para cada nodo en el que exista una correspondencia. En USB, el manual indica que actualmente solo puede mapearse un dispositivo por nodo y por mapping.

La GUI simplifica el proceso porque detecta automáticamente las propiedades correctas y las envía a la API.

En PCI es posible definir varios dispositivos por nodo con varias propiedades map. Cuando el mapping se asigna a una VM, se utiliza el primer dispositivo libre. El orden de las rutas determina el orden de intento, lo que permite construir políticas sencillas de asignación.

Esto es especialmente útil con SR-IOV, donde a menudo no importa qué VF concreta se asigne a la VM.

Un mapping puede asignarse a una VM mediante la GUI o con:

```
# qm set ID -hostpci0 <name>
# para USB:
# qm set <vmid> -usb0 <name>
```

Donde <vmid> es el identificador del invitado y <name> el nombre del mapping. Siguen estando disponibles las opciones habituales de passthrough, incluida mdev.

Permisos

Para crear o modificar mappings se necesita Mapping.Modify sobre /mapping/<type>/<name>. Para utilizarlos se requiere Mapping.Use sobre la misma ruta, además de los privilegios normales necesarios para modificar la configuración del invitado.

Opciones adicionales

- mdev (PCI): marca el dispositivo PCI como capaz de proporcionar mediated devices. Si hay varios dispositivos en el mapping, el dispositivo mediado se crea en el primero que tenga una instancia disponible del tipo seleccionado.
- live-migration-capable (PCI): marca el dispositivo como apto para migración en vivo entre nodos. Requiere soporte del hardware y del driver. El manual indica que se conoce soporte en GPUs NVIDIA con kernels recientes y advierte que esta función sigue siendo experimental.

ADVERTENCIA: La migración en vivo de dispositivos pasados directamente es una función experimental y puede no funcionar o provocar incidencias.

Fin de la Parte 10C · La siguiente parte comienza en 10.13 Managing Virtual Machines with qm.

Traducción no oficial del manual Proxmox VE Administration Guide 9.2.4 · GFDL 1.3 o posterior

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 10D

qm · Archivos de configuración · Snapshots · Opciones avanzadas · Locks

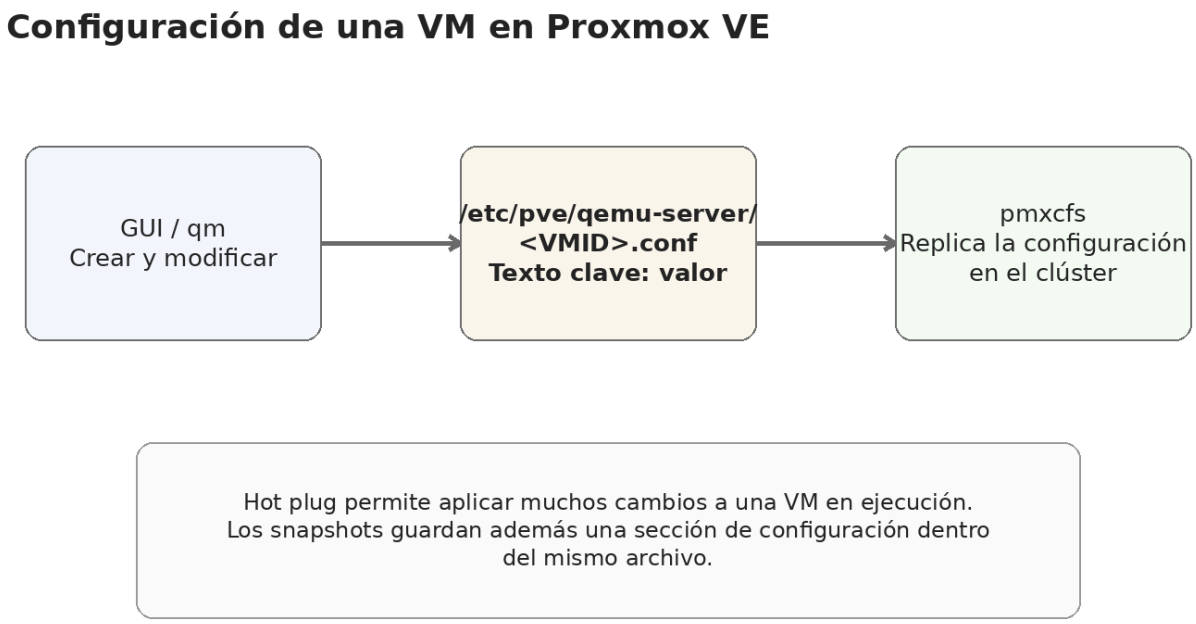
Traducción no oficial del manual Proxmox VE Administration Guide 9.2.4 · GFDL 1.3 o posterior

Capítulo 10 · Máquinas virtuales QEMU/KVM

Esta parte completa el capítulo 10. Comienza en 10.13, con la gestión de máquinas virtuales mediante qm, continúa con el formato de los archivos de configuración y el catálogo completo de opciones, y finaliza con 10.15 Locks.

10.13 Gestión de máquinas virtuales con qm

qm es la herramienta para administrar máquinas virtuales QEMU/KVM en Proxmox VE. Permite crear y destruir máquinas virtuales, controlar su ejecución (start/stop/suspend/resume), modificar los parámetros del archivo de configuración asociado y crear o eliminar discos virtuales.



10.13.1 Ejemplos de uso de la CLI

Usando un archivo ISO subido al almacenamiento local, crear una VM con un disco IDE de 4 GB en local-lvm:

```
# qm create 300 -ide0 local-lvm:4 -net0 e1000 -cdrom local:iso/proxmox-mailgateway_2.1.iso
```

Iniciar la nueva VM:

```
# qm start 300
```

Enviar una solicitud de apagado y esperar hasta que la VM se detenga:

```
# qm shutdown 300 && qm wait 300
```

Igual que el caso anterior, pero esperando solo 40 segundos:

```
# qm shutdown 300 && qm wait 300 -timeout 40
```

Si la VM no se apaga, detenerla a la fuerza y anular cualquier tarea de apagado que siga en ejecución. Detener una VM puede provocar pérdida de datos, por lo que debe hacerse con precaución:

```
# qm stop 300 -overrule-shutdown 1
```

Destruir una VM siempre la elimina de las listas de control de acceso y borra su configuración de firewall. Debe activarse --purge si además se desea retirar la VM de trabajos de replicación, trabajos de copia de seguridad y configuraciones de recursos HA.

NOTA: Activar purge también elimina el recurso HA de las reglas de afinidad que lo referencien y elimina las reglas que solo conserven ese único recurso.

```
# qm destroy 300 --purge
```

Mover una imagen de disco a otro almacenamiento:

```
# qm move-disk 300 scsi0 other-storage
```

Reasignar una imagen de disco a otra VM. El ejemplo elimina scsi1 de la VM de origen y lo adjunta como scsi3 a la VM de destino. En segundo plano, la imagen se renombra para que su nombre coincida con el nuevo propietario:

```
# qm move-disk 300 scsi1 --target-vmid 400 --target-disk scsi3
```

10.14 Configuración

Los archivos de configuración de las VMs se almacenan dentro del sistema de archivos de clúster de Proxmox y pueden consultarse en /etc/pve/qemu-server/<VMID>.conf. Como los demás archivos guardados en /etc/pve/, se replican automáticamente al resto de nodos del clúster.

NOTA: Los VMID inferiores a 100 están reservados para usos internos y los VMID deben ser únicos en todo el clúster.

Ejemplo de configuración de una VM

```
boot: order=virtio0;net0
cores: 1
sockets: 1
memory: 512
name: webmail
ostype: l26
net0: e1000=EE:D2:28:5F:B6:3E,bridge=vmbr0
virtio0: local:vm-100-disk-1,size=32G
```

Los archivos de configuración son archivos de texto sencillos y pueden editarse con un editor normal, como vi o nano. Esto puede ser útil para pequeñas correcciones, pero debe tenerse en cuenta que es necesario reiniciar la VM para aplicar determinados cambios realizados de esta forma.

Por este motivo, normalmente es preferible utilizar qm para generar y modificar esos archivos, o realizar la operación desde la interfaz gráfica. Las herramientas de Proxmox VE pueden aplicar inmediatamente muchos cambios a una VM en ejecución. Esta función se denomina hot plug y, en esos casos, no es necesario reiniciar la máquina virtual.

10.14.1 Formato del archivo

Los archivos de configuración de VM usan un formato sencillo clave/valor separado por dos puntos. Cada línea tiene el formato siguiente:

```
# esto es un comentario
OPTION: value
```

Las líneas en blanco se ignoran. Las líneas que comienzan por el carácter # se consideran comentarios y también se ignoran.

10.14.2 Snapshots

Al crear un snapshot, qm guarda la configuración existente en ese momento en una sección de snapshot independiente dentro del mismo archivo de configuración. Por ejemplo, después de crear un snapshot llamado testsnapshot, el archivo tendrá una estructura similar a esta:

```
memory: 512
swap: 512
parent: testsnaphot
...

[testsnaphot]
memory: 512
swap: 512
snaptime: 1457170803
...
```

Existen varias propiedades específicas de snapshots, como parent y snaptime. parent almacena la relación padre/hijo entre snapshots. snaptime contiene la marca temporal de creación del snapshot en formato Unix epoch.

Opcionalmente puede guardarse la memoria de una VM en ejecución mediante la opción vmstate. Para conocer cómo se selecciona el almacenamiento de destino para el estado de la VM, consulte la selección de almacenamiento de estado explicada en el apartado de hibernación.

10.14.3 Opciones

acpi: <boolean> (default = 1)
Habilita o deshabilita ACPI.

affinity: <string>
Lista de núcleos del host utilizados para ejecutar los procesos del invitado, por ejemplo: 0,5,8-11.

agent: [enabled=]<1|0> [,freeze-fs=<1|0>] [,fstrim_cloned_disks=<1|0>] [,type=<virtio|isa>]
Habilita o deshabilita la comunicación con QEMU Guest Agent y define sus propiedades.

enabled=<boolean> (default = 0) Habilita o deshabilita la comunicación con QEMU Guest Agent (QGA) que se ejecuta dentro de la VM.

freeze-fs=<boolean> (default = 1) Indica si deben emitirse los comandos guest-fsfreeze-freeze y guest-fsfreeze-thaw. Los backups en modo snapshot, clones, snapshots sin RAM, importaciones de discos desde un invitado en ejecución y replicaciones normalmente congelan y descongelan el sistema de archivos cuando QGA está habilitado y en ejecución. La opción obsoleta freeze-fs-on-backup se trata como alias de esta opción.

fstrim_cloned_disks=<boolean> (default = 0) Ejecuta fstrim después de mover un disco o migrar la VM.

type=<isa | virtio> (default = virtio) Selecciona el tipo de agente.

allow-ksm: <boolean> (default = 1)
Permite que las páginas de memoria de este invitado se combinen mediante KSM (Kernel Samepage Merging).

amd-sev: [type=]<sev-type> [,allow-smt=<1|0>] [,kernel-hashes=<1|0>] [,no-debug=<1|0>] [,no-key-sharing=<1|0>]
Funciones Secure Encrypted Virtualization (SEV) de las CPU AMD.

allow-smt=<boolean> (default = 1) Establece el bit de política que permite Simultaneous Multi Threading (SMT). Se ignora salvo en SEV-SNP.

kernel-hashes=<boolean> (default = 0) Añade hashes del kernel al firmware del invitado para el arranque medido del kernel Linux.

no-debug=<boolean> (default = 0) Establece el bit de política para impedir la depuración del invitado.

no-key-sharing=<boolean> (default = 0) Impide compartir claves con otros invitados. Se ignora con SEV-SNP.

type=<sev-type> Activa SEV estándar con type=std, SEV-ES experimental con es o SEV-SNP experimental con snp.

arch: <aarch64 | x86_64>
Arquitectura del procesador virtual. Por defecto se utiliza la arquitectura del host.

args: <string>
Argumentos arbitrarios pasados a kvm. Ejemplo:

```
args: -no-reboot -smbios type=0,vendor=FOO
```

NOTA: Esta opción está destinada únicamente a usuarios expertos.

audio0: device=<ich9-intel-hda|intel-hda|AC97> [,driver=<spice|none>]
Configura un dispositivo de audio, útil junto con QXL/SPICE.

device=<AC97 | ich9-intel-hda | intel-hda> Selecciona el dispositivo de audio.

driver=<none | spice> (default = spice) Backend del controlador de audio.

autostart: <boolean> (default = 0)
Reinicio automático después de una caída. Actualmente se ignora.

balloon: <integer> (0 - N)
Cantidad de RAM objetivo para la VM en MiB. El controlador balloon está habilitado de forma predeterminada salvo que se desactive explícitamente estableciendo el valor a cero.

bios: <ovmf | seabios> (default = seabios)
Selecciona la implementación de BIOS.

boot: [[legacy=]<[acdn]{1,4}>] [,order=<device[;device...]>]
Especifica el orden de arranque del invitado. Se recomienda la subpropiedad order=; el uso sin clave o con legacy= está obsoleto.

legacy=<[acdn]{1,4}> (default = cdn) Arranca desde disquete (a), disco duro (c), CD-ROM (d) o red (n). Está obsoleto; utilice order=.

order=<device[;device...]> El invitado intenta arrancar desde los dispositivos en el orden indicado. Discos, unidades ópticas y dispositivos USB de almacenamiento passthrough arrancan directamente; las NIC cargan PXE y los dispositivos PCIe se comportan como discos o cargan una option ROM. Solo los dispositivos de esta lista se marcan como arrancables. Para software RAID deben incluirse todos los discos necesarios. Si se especifica, sustituye al valor legacy=[acdn]*.

bootdisk: (ide|sata|scsi|virtio)\d+

Habilita el arranque desde el disco indicado. Obsoleto: utilice boot: order=foo;bar.

cdrom: <volume>

Alias de la opción -ide2.

cicustom: [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>]

Cloud-Init: permite sustituir los archivos generados automáticamente al inicio por archivos personalizados.

meta=<volume> Archivo personalizado con los metadatos que se pasan a la VM mediante cloud-init. Depende del proveedor: configdrive2 y nocloud difieren.

network=<volume> Archivo personalizado con los datos de red que se pasan a la VM.

user=<volume> Archivo personalizado con los datos de usuario que se pasan a la VM.

vendor=<volume> Archivo personalizado con los datos del proveedor que se pasan a la VM.

cipassword: <string>

Cloud-Init: contraseña que se asignará al usuario. En general no se recomienda; es preferible utilizar claves SSH. Las versiones antiguas de cloud-init no admiten contraseñas con hash.

citype: <configdrive2 | nocloud | opennebula>

Especifica el formato de configuración de Cloud-Init. El valor predeterminado depende del sistema operativo configurado con ostype: nocloud para Linux y configdrive2 para Windows.

ciupgrade: <boolean> (default = 1)

Cloud-Init: realiza una actualización automática de paquetes después del primer arranque.

ciuser: <string>

Cloud-Init: nombre de usuario al que se aplicarán las claves SSH y la contraseña, en lugar del usuario predeterminado definido en la imagen.

cores: <integer> (1 - N) (default = 1)

Número de núcleos por socket.

cpu: [[cputype=]<string>] [,flags=<+FLAG[;-FLAG...]>] [,guest-phys-bits=<integer>] [,hidden=<1|0>] [,hv-vendor-id=<vendor-id>] [,level=<integer>] [,phys-bits=<8-64|host>] [,reported-model=<enum>]

Tipo de CPU emulada.

cputype=<string> (default = kvm64) Tipo de CPU emulada. Puede ser un nombre predeterminado o personalizado; los modelos personalizados deben comenzar por custom-.

flags=<+FLAG[;-FLAG...]> Lista de flags CPU adicionales separadas por ;. Use +FLAG para habilitar y -FLAG para deshabilitar. nested-virt controla la virtualización anidada: svm en AMD y vmx en Intel. Los modelos personalizados pueden utilizar cualquier flag admitida por QEMU/KVM; las flags específicas de una VM se restringen por seguridad al conjunto documentado por Proxmox VE.

guest-phys-bits=<integer> (32 - 64) Número de bits de dirección física disponibles para el invitado.

hidden=<boolean> (default = 0) Evita identificar la máquina como una VM KVM. Solo afecta a vCPU con arquitectura x86-64.

hv-vendor-id=<vendor-id> ID de proveedor Hyper-V. Algunos controladores o programas de invitados Windows necesitan un ID concreto.

level=<integer> (0 - 4294967295) Máximo valor de entrada para las hojas CPUID básicas que puede consultar el invitado. Las hojas con número superior se ocultan. El valor 30 es una solución habitual para fallos de arranque de Hyper-V en invitados Windows sobre hosts Intel recientes. Solo aplica a vCPU x86_64.

phys-bits=<8-64|host> Bits de dirección física de memoria que se informan al sistema operativo invitado. Deben ser menores o iguales a los del host. host utiliza el valor de la CPU del host, pero rompe la migración en vivo hacia CPUs con un valor diferente.

reported-model=<enum> (default = kvm64) Modelo de CPU y fabricante que se informa al invitado. Debe ser un modelo compatible con QEMU/KVM. Solo es válido para definiciones de modelos CPU personalizados; los modelos predeterminados siempre se identifican como ellos mismos.

```
Valores documentados para reported-model:
486 | Broadwell | Broadwell-IBRS | Broadwell-noTSX | Broadwell-noTSX-IBRS | Cascadelake-Server |
Cascadelake-Server-noTSX | Cascadelake-Server-v2 | Cascadelake-Server-v4 | Cascadelake-Server-v5 |
ClearwaterForest | ClearwaterForest-v2 | ClearwaterForest-v3 | Conroe | Cooperlake | Cooperlake-v2 |
DiamondRapids | EPYC | EPYC-Genoa | EPYC-Genoa-v2 | EPYC-IBPB | EPYC-Milan | EPYC-Milan-v2 | EPYC-Milan-
v3 | EPYC-Rome | EPYC-Rome-v2 | EPYC-Rome-v3 | EPYC-Rome-v4 | EPYC-Rome-v5 | EPYC-Turin | EPYC-v3 |
EPYC-v4 | EPYC-v5 | GraniteRapids | GraniteRapids-v2 | GraniteRapids-v3 | GraniteRapids-v4 |
GraniteRapids-v5 | Haswell | Haswell-IBRS | Haswell-noTSX | Haswell-noTSX-IBRS | Icelake-Client |
Icelake-Client-noTSX | Icelake-Server | Icelake-Server-noTSX | Icelake-Server-v3 | Icelake-Server-v4 |
Icelake-Server-v5 | Icelake-Server-v6 | Icelake-Server-v7 | IvyBridge | IvyBridge-IBRS | KnightsMill |
Nehalem | Nehalem-IBRS | Opteron_G1 | Opteron_G2 | Opteron_G3 | Opteron_G4 | Opteron_G5 | Penryn |
SandyBridge | SandyBridge-IBRS | SapphireRapids | SapphireRapids-v2 | SapphireRapids-v3 |
SapphireRapids-v4 | SapphireRapids-v5 | SapphireRapids-v6 | SierraForest | SierraForest-v2 |
SierraForest-v3 | SierraForest-v4 | SierraForest-v5 | Skylake-Client | Skylake-Client-IBRS | Skylake-
Client-noTSX-IBRS | Skylake-Client-v4 | Skylake-Server | Skylake-Server-IBRS | Skylake-Server-noTSX-IBRS
| Skylake-Server-v4 | Skylake-Server-v5 | Westmere | Westmere-IBRS | a64fx | athlon | core2duo | coreduo
| cortex-a35 | cortex-a53 | cortex-a55 | cortex-a57 | cortex-a710 | cortex-a72 | cortex-a76 | cortex-
a78ae | host | kvm32 | kvm64 | max | neoverse-n1 | neoverse-n2 | neoverse-v1 | pentium | pentium2 |
pentium3 | phenom | qemu32 | qemu64
```

cpulimit: <number> (0 - 128) (default = 0)

Límite de uso de CPU. Si el equipo tiene 2 CPUs dispone de un total de 2 unidades de tiempo de CPU. El valor 0 indica que no existe límite.

cpuunits: <integer> (1 - 262144) (default = cgroup v1: 1024, cgroup v2: 100)

Peso de CPU de la VM utilizado por el planificador justo del kernel. Cuanto mayor sea el número, más tiempo de CPU recibe la VM. El valor es relativo al peso del resto de VMs en ejecución.

description: <string>

Descripción de la VM. Se muestra en el resumen de la VM en la interfaz web y se guarda como comentario dentro del archivo de configuración.

efidisk0: [file=]<volume> [,efitype=<2m|4m>] [,format=<enum>] [,ms-cert=<enum>] [,pre-enrolled-keys=<1|0>] [,size=<DiskSize>]

Configura un disco para almacenar las variables EFI.

efitype=<2m | 4m> (default = 2m) Tamaño y tipo de las variables EFI de OVMF. 4m es más reciente, recomendado y obligatorio para Secure Boot. Por compatibilidad, si no se especifica se usa 2m. Se ignora en VMs arch=aarch64.

file=<volume> Volumen subyacente de la unidad.

format=<cloop | qcow | qcow2 | qed | raw | vmdk> Formato de datos del archivo subyacente.

ms-cert=<2011 | 2023 | 2023k | 2023w> (default = 2011) Marcador informativo que indica la versión más reciente de certificados UEFI de Microsoft inscritos por Proxmox VE. 2023k incluye Microsoft UEFI CA 2023, Windows UEFI CA 2023 y Microsoft Corporation KEK 2K CA 2023. 2023 y 2023w están obsoletos y se mantienen por compatibilidad.

pre-enrolled-keys=<boolean> (default = 0) Usa una plantilla de variables EFI con claves estándar de Microsoft y específicas de distribución ya inscritas, si se usa efitype=4m. Esto activa Secure Boot por defecto, aunque puede deshabilitarse desde la VM.

size=<DiskSize> Tamaño del disco. Es puramente informativo y no tiene efecto.

freeze: <boolean>

Congela la CPU al arrancar. Utilice el comando c del monitor para iniciar la ejecución.

hookscript: <string>

Script que se ejecutará durante distintas fases del ciclo de vida de la VM.

hostpci[n]: [[host=]<HOSTPCIID[;HOSTPCIID2...]>] [,device-id=<hex id>] [,driver=<vfio|keep>] [,legacy-igd=<1|0>] [,mapping=<mapping-id>] [,mdev=<string>] [,pcie=<1|0>] [,rombar=<1|0>] [,romfile=<string>] [,sub-device-id=<hex id>] [,sub-vendor-id=<hex id>] [,vendor-id=<hex id>] [,x-vga=<1|0>]

Asigna dispositivos PCI del host al invitado.

NOTA: Esta opción permite acceso directo al hardware del host. Por tanto, deja de ser posible migrar esas máquinas; utilícela con especial precaución.

PRECAUCIÓN: Función experimental. Los usuarios han comunicado problemas con esta opción.

device-id=<hex id> Sobrescribe el ID de dispositivo PCI visible para el invitado.

driver=<keep | vfio> (default = vfio) Con keep el dispositivo no se restablece ni se enlaza al controlador vfio-pci. Es útil para dispositivos que ya tienen cargado el controlador correcto.

host=<HOSTPCIID[;HOSTPCIID2...]> Passthrough de dispositivo PCI del host. Puede ser un ID PCI o una lista de funciones virtuales PCI. Sintaxis HOSTPCIID: bus:dev.func en hexadecimal. Puede usar lspci para listar dispositivos. Debe definirse host o mapping.

legacy-igd=<boolean> (default = 0) Pasa el dispositivo en modo IGD heredado, convirtiéndolo en el dispositivo gráfico primario y exclusivo de la VM. Requiere máquina pc-i440fx y VGA en none.

mapping=<mapping-id> ID de un mapeo de recursos a nivel de clúster. Debe especificarse esta clave o host.

mdev=<string> Tipo de mediated device que se usará. Se crea una instancia al iniciar la VM y se elimina cuando la VM se detiene.

pcie=<boolean> (default = 0) Utiliza el bus PCI Express. Requiere el modelo de máquina q35.

rombar=<boolean> (default = 1) Indica si la ROM del dispositivo será visible en el mapa de memoria del invitado.

romfile=<string> Nombre de archivo de ROM PCI personalizado, situado en /usr/share/kvm/.

sub-device-id=<hex id> Sobrescribe el ID de dispositivo del subsistema PCI visible para el invitado.

sub-vendor-id=<hex id> Sobrescribe el ID del fabricante del subsistema PCI visible para el invitado.

vendor-id=<hex id> Sobrescribe el ID del fabricante PCI visible para el invitado.

x-vga=<boolean> (default = 0) Habilita soporte vfio-vga.

hotplug: <string> (default = network,disk,usb)

Habilita selectivamente funciones hotplug. La lista se separa por comas: network, disk, cpu, memory, usb y cloudinit. Use 0 para deshabilitar hotplug por completo. El valor 1 es un alias de network,disk,usb. El hotplug USB es posible con machine version >= 7.1 y ostype l26 o Windows > 7.

hugepages: <1024 | 2 | any>

Habilita memoria hugepages. Define el tamaño en MiB. Si se usa any, se intentan páginas de 1 GiB y, si no es posible, se usan páginas de 2 MiB.

ide[n]: [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] ... [,wwn=<wwn>]

Utiliza el volumen como disco duro IDE o CD-ROM (n de 0 a 3).

aio=<io_uring | native | threads> Tipo de AIO que se utilizará.

backup=<boolean> Indica si la unidad debe incluirse al realizar copias de seguridad.

bps=<bps> Velocidad máxima de lectura/escritura en bytes por segundo.

bps_max_length=<seconds> Duración máxima de ráfagas de E/S en segundos.

bps_rd=<bps> Velocidad máxima de lectura en bytes por segundo.

bps_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

bps_wr=<bps> Velocidad máxima de escritura en bytes por segundo.

bps_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

cache=<directsync | none | unsafe | writeback | writethrough> Modo de caché de la unidad.

detect_zeroes=<boolean> Controla si se detectan e intentan optimizar las escrituras de ceros.

discard=<ignore | on> Controla si las peticiones discard/trim se pasan al almacenamiento subyacente.

file=<volume> Volumen subyacente de la unidad.

format=<cloop | qcow | qcow2 | qed | raw | vmdk> Formato de datos del archivo subyacente.

iops=<iops> Máximo de E/S de lectura/escritura en operaciones por segundo.

iops_max=<iops> Reserva máxima no limitada de E/S de lectura/escritura en operaciones por segundo.

iops_max_length=<seconds> Duración máxima de las ráfagas de E/S en segundos.

iops_rd=<iops> Máximo de E/S de lectura en operaciones por segundo.

iops_rd_max=<iops> Reserva máxima no limitada de E/S de lectura en operaciones por segundo.

iops_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

iops_wr=<iops> Máximo de E/S de escritura en operaciones por segundo.

iops_wr_max=<iops> Reserva máxima no limitada de E/S de escritura en operaciones por segundo.

iops_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

mbps=<mbps> Velocidad máxima de lectura/escritura en megabytes por segundo.

mbps_max=<mbps> Reserva máxima no limitada de lectura/escritura en megabytes por segundo.

mbps_rd=<mbps> Velocidad máxima de lectura en megabytes por segundo.

mbps_rd_max=<mbps> Reserva máxima no limitada de lectura en megabytes por segundo.

mbps_wr=<mbps> Velocidad máxima de escritura en megabytes por segundo.

mbps_wr_max=<mbps> Reserva máxima no limitada de escritura en megabytes por segundo.

media=<cdrom | disk> (default = disk) Tipo de medio de la unidad.

replicate=<boolean> (default = 1) Indica si la unidad debe tenerse en cuenta en los trabajos de replicación.

rerror=<ignore | report | stop> Acción ante errores de lectura.

serial=<serial> Número de serie informado por la unidad, codificado para URL, con una longitud máxima de 20 bytes.

shared=<boolean> (default = 0) Marca este volumen gestionado localmente como disponible en todos los nodos. No comparte el volumen automáticamente; presupone que ya es compartido.

size=<DiskSize> Tamaño del disco. Es puramente informativo y no tiene efecto.

snapshot=<boolean> Controla el modo snapshot de QEMU. Si está activo, los cambios realizados en el disco son temporales y se descartan al apagar la VM.

ssd=<boolean> Expone la unidad como SSD en lugar de disco rotacional.

werror=<enospc | ignore | report | stop> Acción ante errores de escritura.

wwn=<wwn> World Wide Name de la unidad, codificado como 16 bytes hexadecimales con prefijo 0x.

intel-tdx: [type=]<tdx-type> ,attestation=<1|0> [,vsock-cid=<integer>] [,vsock-port=<integer>]

Funciones Trusted Domain Extension (TDX) de las CPU Intel.

attestation=<boolean> (default = 1) Habilita la atestación TDX incluyendo quote-generation-socket.

type=<tdx-type> Habilita TDX.

vsock-cid=<integer> (2 - N) (default = 2) CID de vsock del Quote Generation Service.

vsock-port=<integer> (0 - N) (default = 4050) Puerto vsock del Quote Generation Service.

ipconfig[n]: [gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>]

Cloud-Init: especifica direcciones IP y puertas de enlace para la interfaz correspondiente. Las direcciones usan notación CIDR. dhcp puede utilizarse como valor especial para IPv4 y auto para autoconfiguración sin estado en IPv6, que requiere cloud -init 19.4 o posterior. Si no se define IPv4 ni IPv6, se usa DHCP en IPv4.

gw=<GatewayIPv4> Puerta de enlace predeterminada IPv4. Requiere ip.

gw6=<GatewayIPv6> Puerta de enlace predeterminada IPv6. Requiere ip6.

ip=<IPv4Format/CIDR> (default = dhcp) Dirección IPv4 en notación CIDR.

ip6=<IPv6Format/CIDR> (default = dhcp) Dirección IPv6 en notación CIDR.

ivshmem: size=<integer> [,name=<string>]

Memoria compartida entre VMs. Resulta útil para comunicación directa entre VMs o con el host.

name=<string> Nombre del archivo. Se antepone pve-shm-. Por defecto se usa el VMID. El archivo se elimina al detener la VM.

size=<integer> (1 - N) Tamaño del archivo en MB.

keephugepages: <boolean> (default = 0)

Se utiliza junto con hugepages. Si está habilitado, las hugepages no se eliminan tras apagar la VM y pueden reutilizarse en arranques posteriores.

keyboard: <da | de | de-ch | en-gb | en-us | es | fi | fr | fr-be | fr-ca | fr-ch | hu | is | it | ja | lt | mk | nl | no | pl | pt | pt-br | sl | sv | tr>

Distribución de teclado para el servidor VNC. En general no es necesaria y suele ser preferible configurarla dentro del sistema operativo invitado.

kvm: <boolean> (default = 1)

Habilita o deshabilita la virtualización por hardware KVM.

localtime: <boolean>

Configura el reloj en tiempo real (RTC) en hora local. Se habilita por defecto cuando ostype indica un sistema Microsoft Windows.

lock: <backup | clone | create | migrate | rollback | snapshot | snapshot-delete | suspended | suspending>

Bloquea o desbloquea la VM.

machine: [[type=]<machine type>] [,aw-bits=<number>] [,enable-s3=<1|0>] [,enable-s4=<1|0>] [,viommu=<intel|virtio>]

Especifica la máquina QEMU.

aw-bits=<number> (32 - 64) Anchura en bits del espacio de direcciones vIOMMU. Intel vIOMMU admite 39 o 48 bits; VirtIO vIOMMU admite cualquier valor entre 32 y 64.

enable-s3=<boolean> Habilita el estado de energía S3. De forma predeterminada es false a partir de machine types 9.2+pve1 y true en versiones anteriores.

enable-s4=<boolean> Habilita el estado de energía S4. De forma predeterminada es false a partir de machine types 9.2+pve1 y true en versiones anteriores.

type=<machine type> Especifica el tipo de máquina QEMU.

viommu=<intel | virtio> Habilita y selecciona la variante vIOMMU del invitado. Intel vIOMMU requiere q35.

memory: [current=]<integer>

Propiedades de memoria.

current=<integer> (16 - N) (default = 512) Cantidad actual de RAM en línea para la VM en MiB. Es la memoria máxima disponible cuando se utiliza el dispositivo balloon.

migrate_downtime: <number> (0 - N) (default = 0.1)

Tiempo máximo de parada tolerado en migraciones, en segundos. Si la migración no logra converger al final por la cantidad de RAM modificada, el límite aumenta automáticamente por pasos hasta que pueda converger, con un máximo de 2000 segundos en QEMU.

migrate_speed: <integer> (0 - N) (default = 0)

Velocidad máxima de migración en MB/s. 0 significa sin límite.

name: <string>

Nombre de la VM. Solo se utiliza en la interfaz web de configuración.

nameserver: <string>

Cloud-Init: dirección IP del servidor DNS. Si no se especifica ni searchdomain ni nameserver, se usará automáticamente la configuración del host.

net[n]: [model=]<enum> [,bridge=<bridge>] [,firewall=<1|0>] [,host-tunnel=<1|0>] [,link_down=<1|0>] [,macaddr=<XX:XX:XX:XX:XX:XX>] [,mtu=<integer>] [,queues=<integer>] [,rate=<number>] [,tag=<integer>] [,trunks=<vlanid[;vlanid...]>] [,<model>=<macaddr>]

Especifica dispositivos de red.

bridge=<bridge> Bridge al que se conecta el dispositivo. El bridge estándar de Proxmox VE se llama vmbr0. Si no se especifica bridge, se crea una red de usuario KVM con NAT, DHCP y DNS. Usa 10.0.2.2 como gateway, 10.0.2.3 como DNS, 10.0.2.4 como servidor SMB y asigna DHCP a partir de 10.0.2.15.

firewall=<boolean> Indica si esta interfaz debe estar protegida por el firewall.

host-tunnel=<boolean> Habilita host GSO over UDP tunnel offload (solo VirtIO). Requiere QEMU > 10.2 y compatibilidad del kernel del host y del invitado. Está deshabilitado por defecto a partir de machine version 11.0+pve1 para evitar un problema del controlador virtio-net; con versiones 10.2 y 11.0 está habilitado por defecto.

link_down=<boolean> Indica si la interfaz debe permanecer desconectada, como si se retirara el cable.

macaddr=<XX:XX:XX:XX:XX:XX> Dirección MAC común con el bit I/G (Individual/Group) sin establecer.

model=<...> Modelo de tarjeta de red. El modelo virtio ofrece el mejor rendimiento con muy poca sobrecarga de CPU. Si el invitado no admite ese controlador, normalmente es preferible e1000.

```
Modelos documentados: e1000 | e1000-82540em | e1000-82544gc | e1000-82545em | e1000e | i82551 | i82557b | i82559er | ne2k_isa | ne2k_pci | pcnet | rtl8139 | virtio | vmxnet3
```

mtu=<integer> (1 - 65520) Fuerza la MTU del dispositivo de red, solo VirtIO. El valor 1 o vacío utiliza la MTU del bridge.

queues=<integer> (0 - 64) Número de colas de paquetes utilizadas por el dispositivo.

rate=<number> (0 - N) Límite de velocidad en MB/s como número de coma flotante.

tag=<integer> (1 - 4094) Etiqueta VLAN que se aplicará a los paquetes de esta interfaz.

trunks=<vlanid[;vlanid...]> Troncales VLAN que se pasarán a través de esta interfaz.

numa: <boolean> (default = 0)

Habilita o deshabilita NUMA.

numa[n]: cpus=<id[-id];...> [,hostnodes=<id[-id];...>] [,memory=<number>] [,policy=<preferred|bind|interleave>]

Define la topología NUMA.

cpus=<id[-id];...> CPUs que acceden a este nodo NUMA.

hostnodes=<id[-id];...> Nodos NUMA del host que se utilizarán.

memory=<number> Cantidad de memoria proporcionada por este nodo NUMA.

policy=<bind | interleave | preferred> Política de asignación NUMA.

onboot: <boolean> (default = 0)

Indica si la VM se iniciará durante el arranque del sistema.

ostype: <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> (default = other)

Sistema operativo invitado. Se utiliza para habilitar optimizaciones y funciones específicas.

```
other = SO no especificado
wxp = Microsoft Windows XP
w2k = Microsoft Windows 2000
w2k3 = Microsoft Windows 2003
w2k8 = Microsoft Windows 2008
wvista = Microsoft Windows Vista
win7 = Microsoft Windows 7
win8 = Microsoft Windows 8/2012/2012r2
win10 = Microsoft Windows 10/2016/2019
win11 = Microsoft Windows 11/2022/2025
l24 = kernel Linux 2.4
l26 = kernel Linux 2.6 - 7.X
solaris = kernel Solaris/OpenSolaris/OpenIndiania
```

parallel[n]: /dev/parport\d+|/dev/usb/lp\d+

Mapea dispositivos paralelos del host (n de 0 a 2).

NOTA: Esta opción proporciona acceso directo al hardware del host, por lo que deja de ser posible migrar esas máquinas. Utilícela con especial precaución.

PRECAUCIÓN: Función experimental. Los usuarios han comunicado problemas con esta opción.

protection: <boolean> (default = 0)

Activa la protección de la VM. Deshabilita las operaciones de eliminar VM y eliminar disco.

reboot: <boolean> (default = 1)

Permite reiniciar. Si se establece a 0, la VM se cierra cuando intenta reiniciarse.

rng0: [source=]</dev/urandom|/dev/random|/dev/hwrng> [,max_bytes=<integer>] [,period=<integer>]

Configura un generador de números aleatorios basado en VirtIO.

max_bytes=<integer> (default = 1024) Máximo de bytes de entropía que pueden inyectarse al invitado cada period milisegundos. Use 0 para deshabilitar el límite; puede ser peligroso.

period=<integer> (default = 1000) Cada period milisegundos se reinicia la cuota de inyección, permitiendo otros max_bytes de entropía.

source=</dev/hwrng | /dev/random | /dev/urandom> Archivo del host del que se obtiene entropía. Usar urandom no reduce la seguridad de forma significativa, ya que sigue estando sembrado con entropía real. /dev/hwrng permite pasar al invitado un RNG de hardware del host.

sata[n]: [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] ... [,wwn=<wwn>]

Utiliza el volumen como disco duro SATA o CD-ROM (n de 0 a 5).

aio=<io_uring | native | threads> Tipo de AIO que se utilizará.

backup=<boolean> Indica si la unidad debe incluirse al realizar copias de seguridad.

bps=<bps> Velocidad máxima de lectura/escritura en bytes por segundo.

bps_max_length=<seconds> Duración máxima de ráfagas de E/S en segundos.

bps_rd=<bps> Velocidad máxima de lectura en bytes por segundo.

bps_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

bps_wr=<bps> Velocidad máxima de escritura en bytes por segundo.

bps_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

cache=<directsync | none | unsafe | writeback | writethrough> Modo de caché de la unidad.

detect_zeroes=<boolean> Controla si se detectan e intentan optimizar las escrituras de ceros.

discard=<ignore | on> Controla si las peticiones discard/trim se pasan al almacenamiento subyacente.

file=<volume> Volumen subyacente de la unidad.

format=<cloop | qcow | qcow2 | qed | raw | vmdk> Formato de datos del archivo subyacente.

iops=<iops> Máximo de E/S de lectura/escritura en operaciones por segundo.

iops_max=<iops> Reserva máxima no limitada de E/S de lectura/escritura en operaciones por segundo.

iops_max_length=<seconds> Duración máxima de las ráfagas de E/S en segundos.

iops_rd=<iops> Máximo de E/S de lectura en operaciones por segundo.

iops_rd_max=<iops> Reserva máxima no limitada de E/S de lectura en operaciones por segundo.

iops_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

iops_wr=<iops> Máximo de E/S de escritura en operaciones por segundo.

iops_wr_max=<iops> Reserva máxima no limitada de E/S de escritura en operaciones por segundo.

iops_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

mbps=<mbps> Velocidad máxima de lectura/escritura en megabytes por segundo.

mbps_max=<mbps> Reserva máxima no limitada de lectura/escritura en megabytes por segundo.

mbps_rd=<mbps> Velocidad máxima de lectura en megabytes por segundo.

mbps_rd_max=<mbps> Reserva máxima no limitada de lectura en megabytes por segundo.

mbps_wr=<mbps> Velocidad máxima de escritura en megabytes por segundo.

mbps_wr_max=<mbps> Reserva máxima no limitada de escritura en megabytes por segundo.

media=<cdrom | disk> (default = disk) Tipo de medio de la unidad.

replicate=<boolean> (default = 1) Indica si la unidad debe tenerse en cuenta en los trabajos de replicación.

rerror=<ignore | report | stop> Acción ante errores de lectura.

serial=<serial> Número de serie informado por la unidad, codificado para URL, con una longitud máxima de 20 bytes.

shared=<boolean> (default = 0) Marca este volumen gestionado localmente como disponible en todos los nodos. No comparte el volumen automáticamente; presupone que ya es compartido.

size=<DiskSize> Tamaño del disco. Es puramente informativo y no tiene efecto.

snapshot=<boolean> Controla el modo snapshot de QEMU. Si está activo, los cambios realizados en el disco son temporales y se descartan al apagar la VM.

ssd=<boolean> Expone la unidad como SSD en lugar de disco rotacional.

werror=<enospc | ignore | report | stop> Acción ante errores de escritura.

wwn=<wwn> World Wide Name de la unidad, codificado como 16 bytes hexadecimales con prefijo 0x.

`scsi[n]: [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] ... [,wwn=<wwn>]`

Utiliza el volumen como disco duro SCSI o CD-ROM (n de 0 a 30).

aio=<io_uring | native | threads> Tipo de AIO que se utilizará.

backup=<boolean> Indica si la unidad debe incluirse al realizar copias de seguridad.

bps=<bps> Velocidad máxima de lectura/escritura en bytes por segundo.

bps_max_length=<seconds> Duración máxima de ráfagas de E/S en segundos.

bps_rd=<bps> Velocidad máxima de lectura en bytes por segundo.

bps_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

bps_wr=<bps> Velocidad máxima de escritura en bytes por segundo.

bps_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

cache=<directsync | none | unsafe | writeback | writethrough> Modo de caché de la unidad.

detect_zeroes=<boolean> Controla si se detectan e intentan optimizar las escrituras de ceros.

discard=<ignore | on> Controla si las peticiones discard/trim se pasan al almacenamiento subyacente.

file=<volume> Volumen subyacente de la unidad.

format=<cloop | qcow | qcow2 | qed | raw | vmdk> Formato de datos del archivo subyacente.

iops=<iops> Máximo de E/S de lectura/escritura en operaciones por segundo.

iops_max=<iops> Reserva máxima no limitada de E/S de lectura/escritura en operaciones por segundo.

iops_max_length=<seconds> Duración máxima de las ráfagas de E/S en segundos.

iops_rd=<iops> Máximo de E/S de lectura en operaciones por segundo.

iops_rd_max=<iops> Reserva máxima no limitada de E/S de lectura en operaciones por segundo.

iops_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

iops_wr=<iops> Máximo de E/S de escritura en operaciones por segundo.

iops_wr_max=<iops> Reserva máxima no limitada de E/S de escritura en operaciones por segundo.

iops_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

iothread=<boolean> Indica si debe utilizarse iothreads para esta unidad.

product=<product> Nombre de producto de la unidad, hasta 16 bytes.

queues=<integer> (2 - N) Número de colas.

ro=<boolean> Indica si la unidad es de solo lectura.

scsiblock=<boolean> (default = 0) Utiliza scsi-block para passthrough completo de un dispositivo de bloques del host. Puede provocar errores de E/S si el host tiene poca memoria o una fragmentación de memoria elevada.

vendor=<vendor> Nombre de fabricante de la unidad, hasta 8 bytes.

mbps=<mbps> Velocidad máxima de lectura/escritura en megabytes por segundo.

mbps_max=<mbps> Reserva máxima no limitada de lectura/escritura en megabytes por segundo.

mbps_rd=<mbps> Velocidad máxima de lectura en megabytes por segundo.

mbps_rd_max=<mbps> Reserva máxima no limitada de lectura en megabytes por segundo.

mbps_wr=<mbps> Velocidad máxima de escritura en megabytes por segundo.

mbps_wr_max=<mbps> Reserva máxima no limitada de escritura en megabytes por segundo.

media=<cdrom | disk> (default = disk) Tipo de medio de la unidad.

replicate=<boolean> (default = 1) Indica si la unidad debe tenerse en cuenta en los trabajos de replicación.

rerror=<ignore | report | stop> Acción ante errores de lectura.

serial=<serial> Número de serie informado por la unidad, codificado para URL, con una longitud máxima de 20 bytes.

shared=<boolean> (default = 0) Marca este volumen gestionado localmente como disponible en todos los nodos. No comparte el volumen automáticamente; presupone que ya es compartido.

size=<DiskSize> Tamaño del disco. Es puramente informativo y no tiene efecto.

snapshot=<boolean> Controla el modo snapshot de QEMU. Si está activo, los cambios realizados en el disco son temporales y se descartan al apagar la VM.

ssd=<boolean> Expone la unidad como SSD en lugar de disco rotacional.

werror=<enospc | ignore | report | stop> Acción ante errores de escritura.

wwn=<wwn> World Wide Name de la unidad, codificado como 16 bytes hexadecimales con prefijo 0x.

scsihw: <lsi | lsi53c810 | megasas | pvscsi | virtio-scsi-pci | virtio-scsi-single> (default = lsi)

Modelo de controlador SCSI.

searchdomain: <string>

Cloud-Init: establece dominios de búsqueda DNS. Si no se especifica ni searchdomain ni nameserver, se usará automáticamente la configuración del host.

serial[n]: (/dev/[^,]+|socket)

Crea un dispositivo serie dentro de la VM (n de 0 a 3) y pasa un dispositivo serie del host, como /dev/ttyS0, o crea un socket Unix en el host. Use qm terminal para abrir una conexión de terminal.

NOTA: Si se pasa un dispositivo serie físico del host, deja de ser posible migrar la máquina. Utilícelo con especial precaución.

PRECAUCIÓN: Función experimental. Los usuarios han comunicado problemas con esta opción.

shares: <integer> (0 - 50000) (default = 1000)

Cantidad de memory shares para auto-ballooning. Cuanto mayor sea el número, más memoria obtiene la VM en relación con el resto. El valor 0 deshabilita auto-ballooning. pvestatd realiza el auto-ballooning.

smbios1: [base64=<1|0>] [,family=<Base64 encoded string>] [,manufacturer=<Base64 encoded string>] [,product=<Base64 encoded string>] [,serial=<Base64 encoded string>] [,sku=<Base64 encoded string>] [,uuid=<UUID>] [,version=<Base64 encoded string>]

Especifica campos SMBIOS tipo 1.

base64=<boolean> Indica que los valores SMBIOS están codificados en base64.

family=<Base64 encoded string> Establece la cadena family de SMBIOS1.

manufacturer=<Base64 encoded string> Establece el fabricante SMBIOS1.

product=<Base64 encoded string> Establece el ID de producto SMBIOS1.

serial=<Base64 encoded string> Establece el número de serie SMBIOS1.

sku=<Base64 encoded string> Establece la cadena SKU SMBIOS1.

uuid=<UUID> Establece el UUID SMBIOS1.

version=<Base64 encoded string> Establece la versión SMBIOS1.

smp: <integer> (1 - N) (default = 1)

Número de CPUs. Se recomienda utilizar la opción -sockets en su lugar.

sockets: <integer> (1 - N) (default = 1)

Número de sockets de CPU.

spice_enhancements: [foldersharing=<1|0>] [,videostreaming=<off|all|filter>]

Configura mejoras adicionales de SPICE.

foldersharing=<boolean> (default = 0) Habilita el uso compartido de carpetas mediante SPICE. Requiere Spice-WebDAV dentro de la VM.

videostreaming=<all | filter | off> (default = off) Habilita streaming de vídeo, utilizando compresión en los flujos detectados.

sshkeys: <string>

Cloud-Init: configura claves públicas SSH, una por línea, en formato OpenSSH.

startdate: (now | YYYY-MM-DD | YYYY-MM-DDTHH:MM:SS) (default = now)

Fecha inicial del reloj de tiempo real. Formatos válidos: now, 2006-06-17T16:01:21 o 2006-06-17.

startup: [[order=]\d+] [,up=\d+] [,down=\d+]

Comportamiento de inicio y apagado. order es un número no negativo que establece el orden general de arranque. El apagado se realiza en orden inverso. up y down establecen retardos en segundos antes de iniciar o detener la siguiente VM.

tablet: <boolean> (default = 1)

Habilita o deshabilita el dispositivo USB tablet, normalmente necesario para posicionamiento absoluto del ratón con VNC. En hosts con muchos invitados sin consola puede deshabilitarse para ahorrar cambios de contexto. Se desactiva por defecto si se utiliza SPICE (qm set <vmid> --vga qxl).

tags: <string>

Etiquetas de la VM. Son solo metainformación.

tdf: <boolean> (default = 0)

Habilita o deshabilita la corrección de deriva temporal.

template: <boolean> (default = 0)
Habilita o deshabilita el estado de plantilla.

tpmstate0: [file=]<volume> [,format=<raw|qcow2|vmdk>] [,size=<DiskSize>] [,version=<v1.2|v2.0>]
Configura un disco para almacenar el estado TPM. El formato real está fijado a raw.

file=<volume> Volumen subyacente de la unidad.

format=<qcow2 | raw | vmdk> Formato de la imagen.

size=<DiskSize> Tamaño del disco. Es puramente informativo y no tiene efecto.

version=<v1.2 | v2.0> (default = v1.2) Versión de la interfaz TPM. v2.0 es más reciente y debería preferirse. No puede cambiarse posteriormente.

unused[n]: [file=]<volume>
Referencia a volúmenes no utilizados. Se emplea internamente y no debería modificarse manualmente.

file=<volume> Volumen subyacente de la unidad.

usb[n]: [[host=]<HOSTUSBDEVICE|spice>] [,mapping=<mapping-id>] [,usb3=<1|0>]
Configura un dispositivo USB. n va de 0 a 4; con machine version >= 7.1 y ostype l26 o Windows > 7 puede llegar hasta 14.

host=<HOSTUSBDEVICE|spice> Dispositivo o puerto USB del host, o el valor spice. Sintaxis: bus-port(.port)* en decimal, vendor_id:product_id en hexadecimal o spice. Puede utilizar lsusb -t. El valor spice añade un dispositivo de redirección USB para SPICE. Debe definirse host o mapping.

NOTA: Esta opción permite acceso directo al hardware del host y, por tanto, impide migrar esas máquinas. Utilícela con especial precaución.

mapping=<mapping-id> ID de un mapeo de recursos de todo el clúster. Debe establecerse esta clave o host.

usb3=<boolean> (default = 0) Indica que el host o puerto dado es USB3. En invitados modernos (machine version >= 7.1 y ostype l26 o Windows > 7), esta flag es irrelevante porque los dispositivos se conectan a un controlador xHCI.

vcpus: <integer> (1 - N) (default = 0)
Número de vCPU conectadas mediante hotplug.

vga: [[type=]<enum>] [,clipboard=<vnc>] [,memory=<integer>]
Configura el hardware VGA. Para resoluciones altas puede ser necesario aumentar la memoria VGA. Desde QEMU 2.9 el tipo predeterminado es std para la mayoría de sistemas, salvo algunas versiones antiguas de Windows que usan cirrus. qxl habilita el servidor SPICE. También es posible trabajar sin tarjeta gráfica usando un dispositivo serie como terminal.

clipboard=<vnc> Habilita un portapapeles específico. Si no se establece, puede añadirse el de SPICE según el tipo de pantalla. La migración en vivo con portapapeles VNC no es posible con QEMU machine version < 10.1.

memory=<integer> (4 - 512) Memoria VGA en MiB. No tiene efecto con pantalla serie.

type=<cirrus | none | qxl | qxl2 | qxl3 | qxl4 | serial0 | serial1 | serial2 | serial3 | std | virtio | virtio-gl | vmware> (default = std) Tipo de VGA. No se recomienda cirrus.

virtio[n]: [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] ... [,werror=<enum>]
Utiliza el volumen como disco duro VIRTIO (n de 0 a 15).

aio=<io_uring | native | threads> Tipo de AIO que se utilizará.

backup=<boolean> Indica si la unidad debe incluirse al realizar copias de seguridad.

bps=<bps> Velocidad máxima de lectura/escritura en bytes por segundo.

bps_max_length=<seconds> Duración máxima de ráfagas de E/S en segundos.

bps_rd=<bps> Velocidad máxima de lectura en bytes por segundo.

bps_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

bps_wr=<bps> Velocidad máxima de escritura en bytes por segundo.

bps_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

cache=<directsync | none | unsafe | writeback | writethrough> Modo de caché de la unidad.

detect_zeroes=<boolean> Controla si se detectan e intentan optimizar las escrituras de ceros.

discard=<ignore | on> Controla si las peticiones discard/trim se pasan al almacenamiento subyacente.

file=<volume> Volumen subyacente de la unidad.

format=<cloop | qcow | qcow2 | qed | raw | vmdk> Formato de datos del archivo subyacente.

iops=<iops> Máximo de E/S de lectura/escritura en operaciones por segundo.

iops_max=<iops> Reserva máxima no limitada de E/S de lectura/escritura en operaciones por segundo.

iops_max_length=<seconds> Duración máxima de las ráfagas de E/S en segundos.

iops_rd=<iops> Máximo de E/S de lectura en operaciones por segundo.

iops_rd_max=<iops> Reserva máxima no limitada de E/S de lectura en operaciones por segundo.

iops_rd_max_length=<seconds> Duración máxima de ráfagas de lectura en segundos.

iops_wr=<iops> Máximo de E/S de escritura en operaciones por segundo.

iops_wr_max=<iops> Reserva máxima no limitada de E/S de escritura en operaciones por segundo.

iops_wr_max_length=<seconds> Duración máxima de ráfagas de escritura en segundos.

iothread=<boolean> Indica si debe utilizarse iothreads para esta unidad.

ro=<boolean> Indica si la unidad es de solo lectura.

mbps=<mbps> Velocidad máxima de lectura/escritura en megabytes por segundo.

mbps_max=<mbps> Reserva máxima no limitada de lectura/escritura en megabytes por segundo.

mbps_rd=<mbps> Velocidad máxima de lectura en megabytes por segundo.

mbps_rd_max=<mbps> Reserva máxima no limitada de lectura en megabytes por segundo.

mbps_wr=<mbps> Velocidad máxima de escritura en megabytes por segundo.

mbps_wr_max=<mbps> Reserva máxima no limitada de escritura en megabytes por segundo.

media=<cdrom | disk> (default = disk) Tipo de medio de la unidad.

replicate=<boolean> (default = 1) Indica si la unidad debe tenerse en cuenta en los trabajos de replicación.

rerror=<ignore | report | stop> Acción ante errores de lectura.

serial=<serial> Número de serie informado por la unidad, codificado para URL, con una longitud máxima de 20 bytes.

shared=<boolean> (default = 0) Marca este volumen gestionado localmente como disponible en todos los nodos. No comparte el volumen automáticamente; presupone que ya es compartido.

size=<DiskSize> Tamaño del disco. Es puramente informativo y no tiene efecto.

snapshot=<boolean> Controla el modo snapshot de QEMU. Si está activo, los cambios realizados en el disco son temporales y se descartan al apagar la VM.

werror=<enospc | ignore | report | stop> Acción ante errores de escritura.

virtiofs[n]: [dirid=]<mapping-id> [,cache=<enum>] [,direct-io=<1|0>] [,expose-acl=<1|0>] [,expose-xattr=<1|0>]

Configura el uso compartido de un directorio entre host e invitado mediante Virtio-fs.

cache=<always | auto | metadata | never> (default = auto) Política de caché del sistema de archivos: auto, always, metadata o never.

direct-io=<boolean> (default = 0) Respeta la flag O_DIRECT transmitida por las aplicaciones del invitado.

dirid=<mapping-id> Identificador del mapeo de directorio compartido con el invitado. También se utiliza como etiqueta de montaje dentro de la VM.

expose-acl=<boolean> (default = 0) Habilita ACL POSIX para este montaje. Habilitar ACL implica xattr.

expose-xattr=<boolean> (default = 0) Habilita atributos extendidos para este montaje.

vmgenid: <UUID> (default = 1 (autogenerated))

El dispositivo VM Generation ID expone al sistema operativo invitado un identificador entero de 128 bits. Permite avisar al invitado cuando la VM se ejecuta con una configuración diferente, por ejemplo tras restaurar un snapshot o crearla desde una plantilla. El SO invitado puede reaccionar marcando copias de bases de datos distribuidas como no fiables, reinicializando su generador aleatorio, etc. La creación automática solo funciona mediante métodos create/update de API/CLI, no al editar manualmente el archivo.

vmstatestorage: <storage ID>

Almacenamiento predeterminado para volúmenes o archivos que contienen el estado de la VM.

watchdog: [[model=]<i6300esb|ib700>] [,action=<enum>]

Crea un watchdog virtual de hardware. Una vez activado por el invitado, debe sondearse periódicamente mediante un agente dentro de la VM; en caso contrario, el watchdog reinicia el invitado o ejecuta la acción configurada.

action=<debug | none | pause | poweroff | reset | shutdown> Acción que se ejecuta si, tras activarse, el invitado deja de sondear el watchdog a tiempo.

model=<i6300esb | ib700> (default = i6300esb) Tipo de watchdog que se emulará.

10.15 Locks

Las migraciones en línea, los snapshots y las copias de seguridad realizadas con vzdump establecen un lock para impedir accio nes incompatibles concurrentes sobre las VMs afectadas. En algunas situaciones, por ejemplo después de un corte de alimentación, puede ser necesario retirar manualmente un lock.

```
# qm unlock <vmid>
```

PRECAUCIÓN: Utilice este comando únicamente si está seguro de que la acción que estableció el lock ya no sigue ejecutándose.

Fin del Capítulo 10 · Máquinas virtuales QEMU/KVM.

Traducción no oficial del manual Proxmox VE Administration Guide 9.2.4 · GFDL 1.3 o posterior

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 11A

Capítulo 11 · Proxmox Container Toolkit

De 11.1 a 11.7.1 · LXC, imágenes, recursos, seguridad y almacenamiento

LICENCIA: Traducción basada en Proxmox VE Administration Guide 9.2.4, distribuida bajo GNU Free Documentation License 1.3 o posterior. Los comandos, rutas, nombres de paquetes, parámetros y siglas se conservan sin traducir.

Fuente: Proxmox Server Solutions GmbH · Edición original de agosto de 2026

Capítulo 11 · Proxmox Container Toolkit

Los contenedores son una alternativa ligera a las máquinas completamente virtualizadas (VM). En lugar de emular un sistema operativo completo, utilizan el kernel del sistema anfitrión en el que se ejecutan. Esto permite que accedan de forma directa y eficiente a determinados recursos del host.

El coste de ejecución de los contenedores es bajo y, normalmente, despreciable. Sin embargo, hay limitaciones importantes que deben tenerse en cuenta:

- En Proxmox Containers solo pueden ejecutarse distribuciones Linux. No es posible ejecutar sistemas como FreeBSD o Microsoft Windows dentro de un contenedor.
- Por motivos de seguridad, el acceso a los recursos del host debe restringirse. Los contenedores se ejecutan en namespaces separados y determinadas syscalls (solicitudes de espacio de usuario al kernel Linux) no están permitidas dentro del contenedor.

Proxmox VE utiliza Linux Containers (LXC) como tecnología subyacente. El «Proxmox Container Toolkit» (pct) simplifica el uso y la gestión de LXC mediante una interfaz que abstrae las tareas complejas.

Los contenedores están estrechamente integrados con Proxmox VE: conocen la configuración del clúster, utilizan la misma infraestructura de red y almacenamiento que las máquinas virtuales, pueden beneficiarse del firewall de Proxmox VE y también integrarse en el marco de alta disponibilidad (HA).

Tradicionalmente, el objetivo principal ha sido ofrecer un entorno con muchas de las ventajas de una VM, pero sin su sobrecarga adicional. Por ello, Proxmox Containers se han considerado principalmente «contenedores de sistema».

Con la incorporación del soporte para imágenes OCI (Open Container Initiative), Proxmox VE integra también «contenedores de aplicación» como tecnología preliminar. Al crear un contenedor desde una imagen OCI, esta se convierte automáticamente al stack LXC utilizado por Proxmox VE.

Este enfoque permite aprovechar el amplio ecosistema de aplicaciones preempaquetadas manteniendo las capacidades de gestión de Proxmox VE. Aun así, para casos que requieran el máximo aislamiento o migración en vivo, sigue siendo recomendable ejecutar los contenedores dentro de una VM QEMU de Proxmox VE.

ESQUEMA · Dónde encaja LXC en Proxmox VE

Contenedor 101	**Contenedor 102**	**Contenedor OCI**
Namespaces + cgroups	AppArmor + seccomp	LXC / pct
Kernel Linux del host	**Kernel Linux del host**	**Kernel Linux del host**
Almacenamiento Proxmox	**Red / Firewall / HA**	**Nodo Proxmox VE**

11.1 Visión general de la tecnología

- LXC (Linux Containers).
- Integración en la interfaz gráfica web de Proxmox VE (GUI).
- Herramienta de línea de comandos pct, sencilla de utilizar.
- Acceso a través de la API REST de Proxmox VE.
- lxcfs para proporcionar un sistema /proc adaptado al contenedor.
- Control groups (cgroups) para aislar y limitar recursos.
- AppArmor y seccomp para reforzar la seguridad.
- Kernels Linux modernos.
- Despliegue basado en imágenes (plantillas).
- Uso de la biblioteca de almacenamiento de Proxmox VE.
- Configuración del contenedor desde el host: red, DNS, almacenamiento, etc.

11.2 Distribuciones compatibles

A continuación se enumeran las distribuciones soportadas oficialmente. Las plantillas de estas distribuciones están disponibles a través de los repositorios de Proxmox VE y pueden descargarse con la herramienta pveam o desde la interfaz gráfica.

11.2.1 Alpine Linux

Distribución Linux ligera y orientada a la seguridad, basada en musl libc y BusyBox. El proyecto mantiene su propia relación de versiones soportadas.

11.2.2 Arch Linux

Distribución Linux ligera y flexible que sigue el principio «Keep It Simple». Utiliza un modelo rolling release.

11.2.3 CentOS / AlmaLinux / Rocky Linux

CentOS/CentOS Stream deriva del ecosistema RHEL. AlmaLinux es una distribución empresarial comunitaria compatible binariamente con RHEL y CentOS pre-Stream. Rocky Linux se orienta igualmente a la compatibilidad con el ecosistema empresarial de Red Hat.

11.2.4 Debian

Sistema operativo libre desarrollado y mantenido por el proyecto Debian, con un amplio repositorio de aplicaciones.

11.2.5 Devuan

Fork de Debian sin systemd, orientado a preservar alternativas de init.

11.2.6 Fedora

Plataforma Linux libre y abierta orientada a hardware, nube y contenedores.

11.2.7 Gentoo

Distribución altamente flexible basada en compilación desde código fuente y con modelo rolling release.

11.2.8 openSUSE

Distribución orientada a administradores, desarrolladores y usuarios de escritorio.

11.2.9 Ubuntu

Sistema Linux abierto para servidor, escritorio, nube e IoT.

NOTA: Las versiones concretas soportadas cambian con el tiempo. El manual original remite a los sitios oficiales de cada distribución para consultar las versiones vigentes.

11.3 Imágenes de contenedor

Las imágenes de contenedor, también denominadas «plantillas» o «appliances», son archivos tar que contienen todo lo necesario para ejecutar un contenedor. Proxmox VE puede utilizar dos grandes tipos de imágenes: plantillas de contenedor de sistema, destinadas a entornos Linux completos, e imágenes de contenedor de aplicación basadas en OCI para ejecutar aplicaciones concretas.

11.3.1 Plantillas de contenedor de sistema

Proxmox VE proporciona plantillas básicas para las distribuciones Linux más habituales. Se pueden descargar desde la GUI o mediante pveam (Proxmox VE Appliance Manager). También están disponibles plantillas de TurnKey Linux.

La lista de plantillas disponibles se actualiza diariamente mediante el temporizador pve-daily-update. También puede forzarse la actualización manualmente:

```
# pveam update
```

Para consultar todas las imágenes disponibles:

```
# pveam available
```

La lista puede filtrarse por sección. Por ejemplo, para mostrar solo imágenes básicas de sistema:

```
# pveam available --section system

system alpine-3.12-default_20200823_amd64.tar.xz
system archlinux-base_20210420-1_amd64.tar.gz
system centos-8-default_20201210_amd64.tar.xz
system debian-10-standard_10.7-1_amd64.tar.gz
system ubuntu-20.04-standard_20.04-1_amd64.tar.gz
...
```

Antes de utilizar una plantilla debe descargarse a uno de los almacenamientos. Si no se necesita una ubicación específica, pu ede usarse el almacenamiento local. En instalaciones en clúster es preferible un almacenamiento compartido para que todos los nodos accedan a las mismas imágenes.

```
# pveam download local debian-10.0-standard_10.0-1_amd64.tar.gz
```

Para listar las imágenes descargadas en local:

```
# pveam list local
local:vztmpl/debian-10.0-standard_10.0-1_amd64.tar.gz 219.95MB
```

pct puede utilizar directamente ese identificador de volumen para crear un contenedor:

```
# pct create 999 local:vztmpl/debian-10.0-standard_10.0-1_amd64.tar.gz
```

Y la plantilla puede eliminarse posteriormente con:

```
# pveam remove local:vztmpl/debian-10.0-standard_10.0-1_amd64.tar.gz
```

11.3.2 Imágenes Open Container Initiative (OCI)

Proxmox VE también puede utilizar imágenes OCI para crear contenedores, tanto de sistema como de aplicación. La ejecución de contenedores de aplicación en Proxmox VE se considera actualmente una tecnología preliminar. Un contenedor creado desde una imagen OCI sigue utilizando el framework LXC existente.

11.3.3 Obtención de imágenes OCI

En la interfaz web puede cargarse manualmente una imagen OCI o extraerse desde un registro mediante el botón «Pull from OCI registry» de la vista de plantillas del almacenamiento. Una vez almacenada la imagen, el contenedor puede crearse con pct create o mediante el asistente de la interfaz web.

11.4 Configuración de contenedores

11.4.1 Ajustes generales

Los ajustes generales de un contenedor incluyen:

- Node: servidor físico en el que se ejecutará el contenedor.
- CT ID: número único dentro de la instalación de Proxmox VE que identifica al contenedor.
- Hostname: nombre de host del contenedor.
- Resource Pool: grupo lógico de contenedores y máquinas virtuales.
- Password: contraseña de root del contenedor.
- SSH Public Key: clave pública para conectarse a la cuenta root mediante SSH.
- Unprivileged container: permite elegir en el momento de la creación si el contenedor será privilegiado o no privilegiado.
- Nesting: expone procfs y sysfs para permitir contenedores anidados. systemd también utiliza esta capacidad para aislar servicios.

Contenedores no privilegiados

Los contenedores no privilegiados utilizan user namespaces. El UID 0 (root) del interior se mapea a un usuario sin privilegios en el host. De este modo, muchos fallos de seguridad o abusos de recursos afectarían a un usuario genérico sin privilegios en vez de conceder acceso root al host. Esta es la opción predeterminada al crear un contenedor nuevo.

> **NOTA:** Si el contenedor utiliza systemd como sistema init, la versión de systemd del interior debería ser 220 o posterior.

Contenedores privilegiados

La seguridad se basa en AppArmor, filtros seccomp y namespaces del kernel. El equipo LXC considera esta modalidad menos segura; por ello, los contenedores privilegiados deberían utilizarse únicamente en entornos de confianza.

11.4.2 CPU

El número de CPU visibles dentro del contenedor puede limitarse mediante la opción cores. Se implementa usando el cgroup cpuset de Linux. pvestatd intenta distribuir periódicamente los contenedores en ejecución entre las CPU disponibles.

Para consultar las CPU asignadas:

```
# pct cpusets
---------------------
102: 6 7
105: 2 3 4 5
108: 0 1
---------------------
```

Los contenedores utilizan directamente el kernel del host. Todas las tareas del interior son gestionadas por el planificador de CPU del host. Proxmox VE utiliza de forma predeterminada CFS (Completely Fair Scheduler), que permite además controlar el ancho de banda de CPU.

cpulimit	Limita adicionalmente el tiempo total de CPU. Es un número de coma flotante: se pueden asignar dos cores visibles pero limitar el consumo global, por ejemplo, a medio core.
cores	Número de CPU visibles para el contenedor.
cpuunits	Peso relativo que recibe el contenedor en el planificador del kernel. Cuanto mayor sea el valor, mayor prioridad relativa de CPU. El valor predeterminado es 100, o 1024 en hosts con cgroup v1 heredado.

```
cores: 2
cpulimit: 0.5
```

11.4.3 Memoria

La memoria de los contenedores se controla mediante el controlador de memoria de cgroups.

memory	Limita el uso total de memoria. En cgroup v1 correspondía a memory.limit_in_bytes.
swap	Permite utilizar memoria swap adicional del host. En el modelo heredado correspondía a memory.memsw.limit_in_bytes y se calculaba como memory + swap.

11.4.4 Puntos de montaje

El punto de montaje raíz se configura mediante la propiedad rootfs. Pueden configurarse hasta 256 puntos de montaje adicionales, denominados mp0 a mp255.

```
rootfs: [volume=]<volume> [,acl=<1|0>]
        [,idmap=<type:container:disk:range-size[;...] >]
        [,mountoptions=<opt[;opt...]>] [,quota=<1|0>]
        [,replicate=<1|0>] [,ro=<1|0>] [,shared=<1|0>]
        [,size=<DiskSize>]

mp[n]: [volume=]<volume> ,mp=<Path> [,acl=<1|0>] [,backup=<1|0>]
       [,idmap=<type:container:disk:range-size[;...] >]
       [,keepattrs=<1|0>] [,mountoptions=<opt[;opt...]>]
       [,quota=<1|0>] [,replicate=<1|0>] [,ro=<1|0>]
       [,shared=<1|0>] [,size=<DiskSize>]
```

La sintaxis especial STORAGE_ID:SIZE_IN_GiB permite asignar automáticamente un volumen nuevo.

acl	Activa o desactiva explícitamente el soporte ACL.
backup	Incluye el punto de montaje en las copias de seguridad. Solo se utiliza en puntos de montaje basados en volúmenes.
idmap	Personaliza el mapeo UID/GID del punto de montaje. Acepta entradas type:container:disk:range-size separadas por punto y coma.
keepattrs	Hereda UID, GID y modo de acceso del directorio de montaje si ya existe.
mountoptions	Opciones adicionales de montaje para rootfs o mp[n].
mp	Ruta del punto de montaje vista desde el interior del contenedor. No debe contener enlaces simbólicos por motivos de seguridad.

quota	Activa cuotas de usuario dentro del contenedor; no está soportado con subvolúmenes ZFS.
replicate	Incluye el volumen en trabajos de replicación de almacenamiento.
ro	Punto de montaje de solo lectura.
shared	Marca un punto de montaje no basado en volumen como disponible en todos los nodos. No lo comparte automáticamente.
size	Tamaño del volumen; valor de solo lectura.
volume	Volumen, dispositivo o directorio que se montará dentro del contenedor.

Ejemplos de idmap:

- u:123:456:1 mapea el UID 123 del contenedor al UID 456 en el almacenamiento.
- g:100:50:5 mapea cinco GID consecutivos: 100-104 del contenedor se corresponden con 50-54 en disco.
- passthrough realiza un mapeo de identidad de todos los UID y GID.

Configuración típica de rootfs:

```
rootfs: thin1:base-100-disk-1,size=8G
```

Puntos de montaje respaldados por almacenamiento

Son administrados por el subsistema de almacenamiento de Proxmox VE y pueden ser de tres tipos:

- Basados en imagen: imágenes raw que contienen un único sistema de archivos ext4.
- Subvolúmenes ZFS: técnicamente son bind mounts gestionados, por lo que permiten redimensionado y snapshots.
- Directorios: size=0 activa un caso especial en el que se crea un directorio en lugar de una imagen raw.

NOTA: La sintaxis STORAGE_ID:SIZE_IN_GB asigna automáticamente un volumen. Por ejemplo, pct set 100 -mp0 thin1:10,mp=/path/in/container crea un volumen de 10 GB en thin1 y lo monta en /path/in/container.

Bind mounts

Permiten acceder desde el contenedor a directorios arbitrarios del host Proxmox VE. Entre sus usos habituales están acceder a un directorio del host, exponer un directorio de un dispositivo USB o hacer visible un montaje NFS del host.

No están gestionados por el subsistema de almacenamiento, por lo que no admiten snapshots ni cuotas desde el contenedor. En contenedores no privilegiados pueden aparecer problemas de permisos por el mapeo de usuarios y no pueden utilizarse ACL de la misma manera.

NOTA: El contenido de los bind mounts no se incluye en las copias realizadas con vzdump.

ADVERTENCIA: Por seguridad, utilice directorios reservados específicamente para bind mounts, por ejemplo bajo /mnt/bindmounts. Nunca monte directorios del sistema como /, /var o /etc dentro de un contenedor.

NOTA: La ruta origen del bind mount no debe contener enlaces simbólicos.

Ejemplo: hacer accesible /mnt/bindmounts/shared dentro del contenedor 100 como /shared:

```
mp0: /mnt/bindmounts/shared,mp=/shared

pct set 100 -mp0 /mnt/bindmounts/shared,mp=/shared
```

Puntos de montaje de dispositivo

Permiten montar directamente dispositivos de bloques del host dentro del contenedor. Igual que los bind mounts, no los gestiona el subsistema de almacenamiento de Proxmox VE, aunque se respetan las opciones quota y acl.

NOTA: Deben reservarse para casos especiales. En la mayoría de situaciones, un punto de montaje respaldado por almacenamiento ofrece el mismo rendimiento y muchas más funciones.

NOTA: El contenido de los puntos de montaje de dispositivo no se incluye en las copias de seguridad realizadas con vzdump.

11.4.5 Red

Puede configurarse un máximo de 10 interfaces de red por contenedor, denominadas net0 a net9.

```
net[n]: name=<string> [,bridge=<bridge>] [,firewall=<1|0>]
        [,gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,host-managed=<1|0>]
        [,hwaddr=<XX:XX:XX:XX:XX:XX>] [,ip=<(IPv4/CIDR|dhcp|manual)>]
        [,ip6=<(IPv6/CIDR|auto|dhcp|manual)>] [,link_down=<1|0>]
        [,mtu=<integer>] [,rate=<mbps>] [,tag=<integer>]
        [,trunks=<vlanid[;vlanid...]>] [,type=<veth>]
```

bridge	Bridge al que se conecta la interfaz.
firewall	Indica si deben aplicarse las reglas del firewall a la interfaz.
gw / gw6	Puerta de enlace predeterminada IPv4 / IPv6.
host-managed	Hace responsable al host de la configuración IP. Es útil en contenedores de aplicación que no disponen de un stack propio de gestión de red.
hwaddr	Dirección MAC de la interfaz.
ip / ip6	Dirección IPv4 o IPv6 en CIDR; también admite dhcp, auto o manual según el caso.
link_down	Simula la desconexión física del enlace.
mtu	MTU de la interfaz, entre 64 y 65535.
name	Nombre de la interfaz visto desde el interior del contenedor.
rate	Límite de velocidad aplicado a la interfaz.
tag	Etiqueta VLAN de 1 a 4094.
trunks	VLAN que pueden atravesar la interfaz.
type	Tipo de interfaz; actualmente veth.

11.4.6 Inicio y apagado automáticos de contenedores

Para iniciar automáticamente un contenedor cuando arranca el host, active «Start at boot» en el panel Options o ejecute:

```
# pct set CTID -onboot 1
```

El orden de arranque y apagado puede ajustarse con los siguientes parámetros:

- Start/Shutdown order: prioridad de inicio. El orden se invierte durante el apagado: un contenedor con orden 1 será de los últimos en detenerse.
- Startup delay: intervalo, en segundos, entre el arranque de este contenedor y el siguiente.
- Shutdown timeout: tiempo máximo que Proxmox VE espera tras solicitar el apagado antes de considerar fallida la operación. El valor predeterminado es 60 segundos.

Los contenedores sin un valor Start/Shutdown order arrancan después de los que sí lo tienen. Esta prioridad solo ordena las máquinas del mismo host y no es un orden global del clúster.

11.4.7 Hookscripts

Puede asociarse un script de hook a un contenedor mediante la propiedad hookscript:

```
# pct set 100 -hookscript local:snippets/hookscript.pl
```

El script se invocará durante distintas fases del ciclo de vida del invitado. Proxmox VE incluye un ejemplo en:

```
/usr/share/pve-docs/examples/guest-example-hookscript.pl
```

11.5 Consideraciones de seguridad

Los contenedores utilizan el kernel del host, lo que expone una superficie de ataque distinta a la de una VM completa. En gen eral, las máquinas virtuales proporcionan un aislamiento más fuerte; esto debe tenerse en cuenta si los contenedores se ofrecen a usuarios desconocidos o no confiables.

Para reducir la superficie de ataque, LXC emplea varias funciones de seguridad del kernel, entre ellas AppArmor, cgroups y namespaces.

11.5.1 AppArmor

Los perfiles AppArmor restringen operaciones potencialmente peligrosas. Algunas llamadas al sistema, como determinados montajes, pueden quedar bloqueadas.

Para revisar la actividad de AppArmor:

```
# dmesg | grep apparmor
```

Aunque no se recomienda, AppArmor puede desactivarse para un contenedor. Esto aumenta el riesgo de seguridad porque algunas syscalls podrían facilitar escaladas de privilegios si existe una mala configuración o una vulnerabilidad en LXC o en el kern el Linux.

Para desactivar AppArmor en un contenedor, añada a /etc/pve/lxc/CTID.conf:

```
lxc.apparmor.profile = unconfined
```

ADVERTENCIA: No se recomienda utilizar esta configuración en producción.

11.5.2 Control Groups (cgroup)

cgroup es un mecanismo del kernel para organizar jerárquicamente procesos y distribuir recursos del sistema. Controla principalmente tiempo de CPU, límites de memoria y swap, y acceso a nodos de dispositivo. También se utiliza para «congelar» un contenedor antes de realizar snapshots.

La versión actual es cgroupv2. cgroup v1 se marcó como obsoleto en Proxmox VE 7.0 y fue eliminado por completo en Proxmox VE 9.0. Antes de Proxmox VE 7.0, el modo híbrido era el predeterminado.

Compatibilidad con cgroupv2

Con cgroupv2, memoria y swap se controlan de forma independiente. También cambia por completo el controlador de dispositivos. El manual indica que las cuotas de sistema de archivos no están soportadas en un entorno cgroupv2 puro.

El sistema operativo del contenedor necesita soporte para cgroupv2. Los contenedores con systemd 231 o posterior lo soportan, al igual que los contenedores que no utilizan systemd como init.

NOTA: CentOS 7 y Ubuntu 16.10 son ejemplos de distribuciones con systemd demasiado antiguo para un entorno cgroupv2 puro. Las alternativas son actualizar la distribución, actualizar systemd si existe un backport adecuado, o trasladar el servicio a una VM.

Cambio de versión de cgroup

Antes de Proxmox VE 9.0 era posible volver a la versión anterior mediante el parámetro de línea de comandos del kernel:

```
systemd.unified_cgroup_hierarchy=0
```

Esta opción se documenta como histórica, ya que cgroup v1 fue retirado en Proxmox VE 9.0.

11.6 Configuración del sistema operativo invitado

Proxmox VE intenta detectar la distribución Linux que se ejecuta dentro del contenedor y modifica determinados archivos durante el arranque.

/etc/hostname	Establece el nombre del contenedor.
/etc/hosts	Permite resolver localmente el hostname.
Red	Pasa al contenedor la configuración de red completa.
DNS	Entrega al contenedor la información de servidores DNS.
Sistema init	Adapta parámetros como el número de procesos getty.
Contraseña root	Se establece durante la creación de un contenedor nuevo.

ssh_host_keys	Se regeneran para que cada contenedor tenga claves únicas.
crontab	Se aleatoriza para evitar que todos los contenedores ejecuten cron al mismo tiempo.

Los cambios realizados por Proxmox VE se delimitan con marcadores:

```
# --- BEGIN PVE ---
<data>
# --- END PVE ---
```

Si la sección ya existe, se actualiza en su ubicación actual. Puede impedirse la modificación de un archivo creando un archivo .pve-ignore correspondiente. Por ejemplo, para impedir que Proxmox VE modifique /etc/hosts:

```
# touch /etc/.pve-ignore.hosts
```

La mayoría de modificaciones dependen del sistema operativo. Es posible desactivarlas completamente configurando manualmente ostype como unmanaged.

La detección del sistema operativo comienza examinando /etc/os-release. Si no existe o no identifica claramente la distribución, Proxmox VE comprueba archivos específicos:

Ubuntu	/etc/lsb-release con DISTRIB_ID=Ubuntu
Debian	/etc/debian_version
Fedora	/etc/fedora-release
Red Hat / CentOS	/etc/redhat-release
Arch Linux	/etc/arch-release
Alpine	/etc/alpine-release
Gentoo	/etc/gentoo-release

NOTA: El arranque del contenedor falla si el valor ostype configurado no coincide con el tipo detectado automáticamente.

11.7 Almacenamiento de contenedores

El modelo de almacenamiento LXC de Proxmox VE es más flexible que los modelos tradicionales de contenedores, ya que un contenedor puede disponer de múltiples puntos de montaje. Esto permite elegir el almacenamiento más adecuado para cada carga.

Por ejemplo, el sistema de archivos raíz puede estar en un almacenamiento lento y económico mientras que una base de datos utiliza un segundo punto de montaje situado sobre un almacenamiento rápido y distribuido.

Puede utilizarse cualquier tipo de almacenamiento soportado por la biblioteca de Proxmox VE: almacenamiento local como LVM, ZFS o directorios; almacenamiento compartido externo como iSCSI o NFS; y sistemas distribuidos como Ceph. Las funciones avanzadas, como snapshots o clones, están disponibles cuando el backend subyacente las soporta. vzdump puede utilizar snapshots para obtener copias consistentes.

También pueden montarse directamente dispositivos o directorios locales mediante bind mounts, con una sobrecarga prácticamente nula. Este mecanismo puede utilizarse para compartir datos entre contenedores.

11.7.1 Montajes FUSE

ADVERTENCIA: Debido a problemas existentes en el subsistema freezer del kernel Linux, el manual desaconseja encarecidamente utilizar montajes FUSE dentro de un contenedor. Los contenedores necesitan congelarse para copias en modo suspend o snapshot.

Si un montaje FUSE no puede sustituirse por otro mecanismo o tecnología de almacenamiento, una alternativa es realizar el montaje FUSE en el propio host Proxmox VE y exponerlo al contenedor mediante un bind mount.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 11B

Capítulo 11 · Proxmox Container Toolkit

De 11.7.2 a 11.12 · Cuotas, backup, pct, migración, configuración y locks

LICENCIA: Traducción basada en Proxmox VE Administration Guide 9.2.4, distribuida bajo GNU Free Documentation License 1.3 o posterior. Los comandos, rutas, nombres de paquetes, parámetros y siglas se conservan sin traducir.

Fuente: Proxmox Server Solutions GmbH · Edición original de agosto de 2026

11.7.2 Uso de cuotas dentro de contenedores

Las cuotas permiten establecer, desde el interior de un contenedor, límites sobre la cantidad de espacio en disco que puede utilizar cada usuario.

NOTA: En esta versión, esta función requiere el uso de cgroups heredados (legacy cgroups).

NOTA: Solo funciona con almacenamiento basado en imágenes ext4 y, actualmente, únicamente con contenedores privilegiados.

Al activar la opción quota, Proxmox VE utiliza en el punto de montaje las siguientes opciones:

```
usrjquota=aquota.user,grpjquota=aquota.group,jqfmt=vfsv0
```

A partir de ahí, las cuotas pueden gestionarse como en cualquier otro sistema Linux. Los archivos /aquota.user y /aquota.grou p pueden inicializarse con:

```
# quotacheck -cmug /
# quotaon /
```

Después, los límites pueden editarse mediante el comando edquota. El manual remite a la documentación de la distribución ejecutada dentro del contenedor para los detalles de uso.

NOTA: Los comandos anteriores deben ejecutarse para cada punto de montaje, sustituyendo / por la ruta del punto de montaje correspondiente.

11.7.3 Uso de ACL dentro de contenedores

Las listas de control de acceso POSIX (ACL) estándar también están disponibles dentro de los contenedores. Las ACL permiten definir permisos y propiedad de archivos con mayor detalle que el modelo tradicional de usuario/grupo/otros.

11.7.4 Copia de seguridad de puntos de montaje del contenedor

Para incluir un punto de montaje en las copias de seguridad, active la opción backup en la configuración del contenedor.

Por ejemplo, para un punto de montaje mp0 existente:

```
mp0: guests:subvol-100-disk-1,mp=/root/files,size=8G
```

añada backup=1:

```
mp0: guests:subvol-100-disk-1,mp=/root/files,size=8G,backup=1
```

NOTA: Al crear un nuevo punto de montaje desde la interfaz gráfica, esta opción está activada de forma predeterminada.

Para excluir un punto de montaje de las copias, utilice backup=0 o desmarque la casilla Backup en la interfaz gráfica.

11.7.5 Replicación de puntos de montaje de contenedores

De forma predeterminada, los puntos de montaje adicionales se replican cuando se replica el disco raíz. Si desea que el mecanismo de replicación de almacenamiento de Proxmox VE omita un punto de montaje, active la opción Skip replication para ese punto.

El manual indica que la replicación requiere almacenamiento de tipo zfspool. Si un contenedor tiene replicación configurada y se añade un punto de montaje sobre otro tipo de almacenamiento, ese punto debe tener activada la opción Skip replication.

11.8 Copia de seguridad y restauración

11.8.1 Copia de seguridad de contenedores

Los contenedores pueden copiarse con la herramienta vzdump. El manual original remite a la página de manual de vzdump para las opciones completas.

11.8.2 Restauración de copias de contenedores

Las copias creadas con vzdump pueden restaurarse mediante pct restore. De forma predeterminada, pct restore intenta recuperar la mayor parte posible de la configuración almacenada en el backup.

La configuración incluida en la copia puede sobrescribirse especificando manualmente opciones del contenedor en la línea de comandos.

NOTA: pvesm extractconfig permite consultar la configuración almacenada dentro de un archivo de backup vzdump.

Modo de restauración «Simple»

Si no se especifican de forma explícita ni rootfs ni ninguno de los parámetros mpX opcionales, la configuración de puntos de montaje incluida en la copia se restaura siguiendo estos pasos:

- Extraer del backup los puntos de montaje y sus opciones.
- Crear los volúmenes correspondientes a los puntos respaldados por almacenamiento en el almacenamiento indicado mediante storage (por defecto, local).
- Extraer los archivos del archivo de backup.
- Añadir a la configuración restaurada los bind mounts y puntos de montaje de dispositivo (operación limitada al usuario root).

NOTA: Los bind mounts y device mount points nunca se incluyen en el backup. En el último paso solo se recupera su configuración, no su contenido. Se presupone que esos datos se protegen por otro mecanismo (por ejemplo, NFS montado en varios contenedores) o que no necesitan copia.

Este modo simple es también el utilizado por las operaciones de restauración de contenedores de la interfaz web.

Modo de restauración «Avanzado»

Al establecer el parámetro rootfs y, opcionalmente, cualquier combinación de parámetros mpX, pct restore cambia automáticamente al modo avanzado. En este modo se ignoran por completo las opciones rootfs y mpX incluidas en la copia y se utilizan únicamente las indicadas explícitamente como parámetros.

Esto permite, entre otras cosas:

- Definir individualmente almacenamiento de destino, tamaño de volumen y otras opciones para cada punto de montaje.
- Redistribuir los archivos de la copia según un nuevo esquema de puntos de montaje.
- Restaurar directamente sobre puntos de montaje de dispositivo y/o bind mounts (limitado al usuario root).

ESQUEMA · Backup y restauración de un contenedor

CT + configuración	vzdump	Archivo de backup	pct restore	CT restaurado

11.9 Gestión de contenedores con pct

Proxmox Container Toolkit (pct) es la herramienta de línea de comandos para administrar contenedores en Proxmox VE. Permite crearlos y destruirlos, controlar su ejecución (start, stop, reboot, migrate, etc.) y modificar parámetros del archivo de configuración, como red o límites de memoria.

11.9.1 Ejemplos de uso de la CLI

Crear el contenedor 100 a partir de una plantilla Debian ya descargada:

```
# pct create 100 /var/lib/vz/template/cache/debian-10.0-standard_10.0-1_amd64.tar.gz
```

Iniciar el contenedor:

```
# pct start 100
```

Abrir una sesión de login mediante getty:

```
# pct console 100
```

Entrar en el namespace LXC y ejecutar una shell como root:

```
# pct enter 100
```

Mostrar la configuración:

```
# pct config 100
```

Añadir en caliente una interfaz eth0 conectada al bridge vmbr0, con dirección y gateway:

```
# pct set 100 -net0 name=eth0,bridge=vmbr0,ip=192.168.15.147/24,gw=192.168.15.1
```

Reducir la memoria del contenedor a 512 MB:

```
# pct set 100 -memory 512
```

Al destruir un contenedor siempre se elimina de las ACL y se borra su configuración de firewall. Utilice --purge para eliminarlo además de trabajos de replicación, trabajos de backup y recursos HA:

```
# pct destroy 100 --purge
```

NOTA: Activar purge también elimina el recurso HA de las reglas de afinidad que lo referencien y elimina aquellas reglas que, tras la operación, solo contengan ese recurso.

Mover el volumen de un punto de montaje a otro almacenamiento:

```
# pct move-volume 100 mp0 other-storage
```

Reasignar un volumen a otro contenedor: se elimina mp0 del CT origen y se conecta como mp1 al CT destino; internamente el volumen se renombra para reflejar el nuevo propietario:

```
# pct move-volume 100 mp0 --target-vmid 200 --target-volume mp1
```

11.9.2 Obtención de registros de depuración

Si pct start no puede arrancar un contenedor, puede recopilar información adicional con --debug:

```
# pct start CTID --debug
```

Como alternativa, lxc-start puede guardar el registro de depuración en un archivo:

```
# lxc-start -n CTID -F -l DEBUG -o /tmp/lxc-CTID.log
```

Este comando intenta iniciar el contenedor en primer plano. Para detenerlo, ejecute pct shutdown CTID o pct stop CTID desde o tra terminal. El log se guarda en /tmp/lxc-CTID.log.

NOTA: Si la configuración del contenedor cambió desde el último intento con pct start, ejecute pct start al menos una vez para actualizar también la configuración utilizada por lxc-start.

11.10 Migración

En un clúster, los contenedores pueden migrarse a otro nodo con:

```
# pct migrate <ctid> <target>
```

La migración estándar funciona mientras el contenedor esté detenido. Si tiene volúmenes locales o puntos de montaje definidos, el contenido se copia por red al host destino siempre que allí exista el mismo almacenamiento.

Los contenedores en ejecución no pueden migrarse en vivo por limitaciones técnicas. Proxmox VE ofrece una «restart migration»: apaga el contenedor, lo mueve y vuelve a iniciarlo en el nodo destino. Debido a que los contenedores son ligeros, el tiempo de interrupción suele ser de unos cientos de milisegundos.

Puede realizarse desde la interfaz web o con el parámetro --restart de pct migrate. La operación espera al apagado durante el timeout configurado (180 segundos por defecto), mata el contenedor si excede ese plazo, ejecuta la migración como si estuviera offline y, al terminar, lo arranca en el nodo destino.

ESQUEMA · Restart migration

CT activo	Shutdown	Copia / migración	Inicio en destino	CT activo

11.11 Configuración

La configuración de cada contenedor se almacena en /etc/pve/lxc/<CTID>.conf, donde <CTID> es su identificador numérico. Como el resto de archivos bajo /etc/pve/, se replica automáticamente a los demás nodos del clúster.

NOTA: Los CTID inferiores a 100 están reservados para usos internos y los identificadores deben ser únicos en todo el clúster.

Ejemplo de configuración:

```
ostype: debian
arch: amd64
hostname: www
memory: 512
swap: 512
net0: bridge=vmbr0,hwaddr=66:64:66:64:64:36,ip=dhcp,name=eth0,type=veth
rootfs: local:107/vm-107-disk-1.raw,size=7G
```

Son archivos de texto simples y pueden editarse con herramientas como vi o nano. No obstante, cuando el cambio no puede aplicarse en caliente, será necesario reiniciar el contenedor.

Por ello suele ser preferible modificar la configuración mediante pct o la GUI. Muchas modificaciones pueden aplicarse instantáneamente mediante hot plug. Las que no pueden aplicarse así quedan registradas como cambios pendientes (en rojo en la interfaz) y se aplican tras reiniciar el contenedor.

11.11.1 Formato del archivo

El archivo usa un formato clave/valor separado por dos puntos:

```
# this is a comment
OPTION: value
```

Las líneas en blanco se ignoran y las que comienzan por # se consideran comentarios. También es posible añadir configuración LXC de bajo nivel directamente:

```
lxc.init_cmd: /sbin/my_own_init
# o bien
lxc.init_cmd = /sbin/my_own_init
```

Estos ajustes se transfieren directamente a las herramientas LXC de bajo nivel.

11.11.2 Snapshots

Al crear un snapshot, pct guarda la configuración de ese instante en una sección independiente dentro del mismo archivo. Por ejemplo:

```
memory: 512
swap: 512
parent: testsnaphot
...

[testsnaphot]
memory: 512
swap: 512
snaptime: 1457170803
...
```

parent conserva la relación padre/hijo entre snapshots y snaptime almacena la fecha de creación en formato Unix epoch.

11.11.3 Opciones

Las siguientes opciones corresponden a la referencia incluida en el manual original. Se mantienen nombres, valores y sintaxis porque forman parte de la interfaz de configuración de Proxmox VE.

arch

```
arch: <amd64 | arm64 | armhf | i386 | riscv32 | riscv64> (default = amd64)
```

Arquitectura del sistema operativo del contenedor.

cmode

```
cmode: <console | shell | tty> (default = tty)
```

Modo de consola. tty intenta conectarse a uno de los dispositivos tty disponibles; console utiliza /dev/console; shell abre directamente una shell dentro del contenedor sin proceso de login.

console

```
console: <boolean> (default = 1)
```

Conecta un dispositivo de consola (/dev/console) al contenedor.

cores

```
cores: <integer> (1 - 8192)
```

Número de cores asignados. Por defecto, un contenedor puede utilizar todos los cores disponibles.

cpulimit

```
cpulimit: <number> (0 - 8192) (default = 0)
```

Límite total de uso de CPU. El valor 0 significa sin límite.

NOTA: Si el host dispone de 2 CPU, dispone de un total de 2 unidades de tiempo de CPU.

cpuunits

```
cpuunits: <integer> (0 - 500000) (default = cgroup v1: 1024, cgroup v2: 100)
```

Peso de CPU relativo utilizado por el planificador justo del kernel. Cuanto mayor sea el número, más tiempo de CPU recibe el contenedor en relación con los demás invitados.

debug

```
debug: <boolean> (default = 0)
```

Intenta producir información más detallada. Actualmente activa el nivel de log de depuración durante el arranque.

description

```
description: <string>
```

Descripción del contenedor mostrada en el resumen de la interfaz web. Se guarda como comentario dentro del archivo de configuración.

dev[n]

```
dev[n]: [[path=]<Path>] [,deny-write=<1|0>] [,gid=<integer>]
        [,mode=<Octal access mode>] [,uid=<integer>]
```

Dispositivo del host que se pasa al contenedor.

deny-write: Impide que el contenedor escriba en el dispositivo.

gid: GID que se asignará al nodo de dispositivo.

mode: Modo de acceso octal que se establecerá sobre el nodo de dispositivo.

path: Ruta del dispositivo que se expondrá al contenedor.

uid: UID que se asignará al nodo de dispositivo.

entrypoint

```
entrypoint: (?^:[^\x00-\x08\x0a-\x1F\x7F]+) (default = /sbin/init)
```

Comando que se ejecutará como init, opcionalmente con argumentos. Puede comenzar con ruta absoluta, ruta relativa o corresponder a un binario disponible en $PATH.

env

```
env: (?^:(?:\w+=[^\x00-\x08\x0a-\x1F\x7F]*)(?:\0\w+=[^\x00-\x08\x0a-\x1F\x7F]*)*)
```

Entorno de ejecución del contenedor expresado como lista separada por NUL. Sustituye las entradas lxc.environment.runtime del archivo de configuración.

features

```
features: [force_rw_sys=<1|0>] [,fuse=<1|0>] [,keyctl=<1|0>]
          [,mknod=<1|0>] [,mount=<fstype;fstype;...>] [,nesting=<1|0>]
```

Permite que los contenedores utilicen funciones avanzadas.

force_rw_sys: Monta /sys como lectura/escritura en contenedores no privilegiados en lugar del modo mixto. Puede romper la red con systemd-networkd moderno (>= v245).

fuse: Permite sistemas de archivos FUSE. La interacción entre FUSE y el freezer cgroup puede causar interbloqueos de E/S.

keyctl: Solo para contenedores no privilegiados. Permite keyctl(); es necesario para ejecutar Docker dentro del contenedor. Es, en buena medida, un workaround para systemd-networkd.

mknod: Permite a contenedores no privilegiados usar mknod() para determinados nodos de dispositivo. Requiere kernel >= 5.3 con soporte seccomp trap a espacio de usuario. Es experimental.

mount: Lista de tipos de sistema de archivos que el contenedor puede montar. Puede tener implicaciones graves de seguridad e incluso afectar a la E/S del host.

nesting: Permite anidamiento. Se recomienda con contenedores no privilegiados y mapeo ID adicional. Expone contenidos procfs y sysfs del host y también lo requiere systemd para aislar servicios.

hookscript

```
hookscript: <string>
```

Script ejecutado durante distintas fases del ciclo de vida del contenedor.

hostname

```
hostname: <string>
```

Nombre de host del contenedor.

lock

```
lock: <backup | create | destroyed | disk | fstrim | migrate | mounted | rollback | snapshot | snapshot-delete>
```

Estado de bloqueo/desbloqueo del contenedor.

memory

```
memory: <integer> (16 - N) (default = 512)
```

Cantidad de RAM asignada al contenedor, en MB.

mp[n]

```
mp[n]: [volume=]<volume> ,mp=<Path> [,acl=<1|0>] [,backup=<1|0>]
       [,idmap=<type:container:disk:range-size[;type:container:disk:range-size;...]>]
       [,keepattrs=<1|0>] [,mountoptions=<opt[;opt...]>] [,quota=<1|0>]
       [,replicate=<1|0>] [,ro=<1|0>] [,shared=<1|0>] [,size=<DiskSize>]
```

Utiliza un volumen como punto de montaje del contenedor. La sintaxis especial STORAGE_ID:SIZE_IN_GiB crea un volumen nuevo.

acl: Activa o desactiva explícitamente ACL.

backup: Incluye el punto de montaje en backups; solo se aplica a puntos respaldados por volúmenes.

idmap: Personaliza mapeos UID/GID para este punto. Formato type:container:disk:range-size separado por punto y coma; type es u o g, container es el primer ID dentro del contenedor, disk el ID correspondiente en el almacenamiento y range-size el número de IDs consecutivos. IDs no mapeados usan lxc.idmap.

keepattrs: Hereda UID, GID y modo de acceso del directorio de montaje si ya existe.

mountoptions: Opciones adicionales de montaje para rootfs/mp.

mp: Ruta del punto de montaje vista desde el contenedor. Por seguridad no debe contener enlaces simbólicos.

quota: Activa cuotas de usuario; no está soportado con subvolúmenes ZFS.

replicate: Incluye el volumen en trabajos de replicación.

ro: Monta como solo lectura.

shared: Marca el punto no basado en volumen como disponible en todos los nodos. No lo comparte automáticamente.

size: Tamaño del volumen, valor de solo lectura.

volume: Volumen, dispositivo o directorio que se montará dentro del contenedor.

ADVERTENCIA: La opción shared no comparte el punto automáticamente; presupone que ya es accesible desde todos los nodos.

Ejemplos de idmap para mp[n]:

- u:123:456:1 mapea UID 123 del contenedor a UID 456 en disco.
- g:100:50:5 mapea los GID 100-104 del contenedor a 50-54 en disco.
- passthrough realiza un identity map de todos los UID y GID.

nameserver

```
nameserver: <string>
```

Dirección IP del servidor DNS del contenedor. Durante la creación se hereda del host si no se establecen ni searchdomain ni nameserver.

net[n]

```
net[n]: name=<string> [,bridge=<bridge>] [,firewall=<1|0>]
        [,gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,host-managed=<1|0>]
        [,hwaddr=<XX:XX:XX:XX:XX:XX>] [,ip=<(IPv4/CIDR|dhcp|manual)>]
        [,ip6=<(IPv6/CIDR|auto|dhcp|manual)>] [,link_down=<1|0>]
        [,mtu=<integer>] [,rate=<mbps>] [,tag=<integer>]
        [,trunks=<vlanid[;vlanid...]>] [,type=<veth>]
```

Define una interfaz de red del contenedor.

bridge: Bridge al que se conecta la interfaz.

firewall: Indica si se utilizan las reglas de firewall de esta interfaz.

gw / gw6: Puerta de enlace predeterminada IPv4 / IPv6.

host-managed: El host gestiona la configuración IP en vez del propio contenedor. Es útil para contenedores de aplicación sin gestor de red interno.

hwaddr: Dirección MAC.

ip: Dirección IPv4 en CIDR o los valores dhcp/manual.
ip6: Dirección IPv6 en CIDR o auto/dhcp/manual.
link_down: Desconecta lógicamente la interfaz.
mtu: MTU entre 64 y 65535.
name: Nombre de la interfaz visto desde el contenedor.
rate: Límite de velocidad.
tag: Etiqueta VLAN entre 1 y 4094.
trunks: VLAN que pueden atravesar la interfaz.
type: Tipo de interfaz; veth.

onboot

```
onboot: <boolean> (default = 0)
```

Indica si el contenedor arrancará durante el inicio del sistema.

ostype

```
ostype: <alpine | archlinux | centos | debian | devuan | fedora | gentoo | nixos | opensuse | ubuntu |
unmanaged>
```

Tipo de sistema operativo. Se utiliza para aplicar configuración dentro del contenedor y corresponde a scripts LXC bajo /usr/share/lxc/config/<ostype>.common.conf. unmanaged omite la configuración específica del sistema operativo.

protection

```
protection: <boolean> (default = 0)
```

Activa el flag de protección. Impide eliminar el contenedor o eliminar/actualizar sus discos.

rootfs

```
rootfs: [volume=]<volume> [,acl=<1|0>]
        [,idmap=<type:container:disk:range-size[;type:container:disk:range-size;...]>]
        [,mountoptions=<opt[;opt...]>] [,quota=<1|0>] [,replicate=<1|0>]
        [,ro=<1|0>] [,shared=<1|0>] [,size=<DiskSize>]
```

Utiliza un volumen como sistema de archivos raíz del contenedor.
acl: Activa o desactiva explícitamente ACL.
idmap: Personaliza mapeos UID/GID con la misma sintaxis descrita para mp[n].
mountoptions: Opciones adicionales de montaje.
quota: Activa cuotas de usuario; no compatible con subvolúmenes ZFS.
replicate: Incluye el volumen en trabajos de replicación.
ro: Monta el rootfs como solo lectura.
shared: Marca el volumen administrado localmente como disponible en todos los nodos; no lo comparte automáticamente.
size: Tamaño del volumen, valor de solo lectura.
volume: Volumen, dispositivo o directorio que se monta como raíz.

ADVERTENCIA: shared presupone que el volumen ya está compartido; la opción no realiza la compartición por sí misma.

searchdomain

```
searchdomain: <string>
```

Dominios de búsqueda DNS. Durante la creación se heredan del host si no se establecen ni searchdomain ni nameserver.

startup

```
startup: [[order=]\d+] [,up=\d+] [,down=\d+]
```

Comportamiento de inicio y apagado. order define el orden general; el apagado utiliza el orden inverso. up y down establecen retardos en segundos antes de continuar con el siguiente invitado.

swap

```
swap: <integer> (0 - N) (default = 512)
```

Cantidad de SWAP del contenedor, en MB.

tags

```
tags: <string>
```

Etiquetas del contenedor. Solo son metadatos.

template

```
template: <boolean> (default = 0)
```

Activa o desactiva el estado de plantilla.

timezone

```
timezone: <string>
```

Zona horaria del contenedor. Si no se establece, no se modifica. Puede utilizar host para coincidir con el host o cualquier zona válida de /usr/share/zoneinfo/zone.tab.

tty

```
tty: <integer> (0 - 6) (default = 2)
```

Número de tty disponibles dentro del contenedor.

unprivileged

```
unprivileged: <boolean> (default = 0)
```

Hace que el contenedor se ejecute como usuario no privilegiado. Durante creación el valor predeterminado es 1; durante restauración se usa el valor guardado en el backup. No debería modificarse manualmente.

unused[n]

```
unused[n]: [volume=]<volume>
```

Referencia a volúmenes sin utilizar. Se usa internamente y no debe modificarse a mano.

volume: Volumen que actualmente no está en uso.

11.12 Locks

Las migraciones, snapshots y copias de seguridad (vzdump) establecen locks para impedir acciones simultáneas incompatibles sobre el contenedor. En determinadas situaciones, por ejemplo después de un fallo eléctrico, puede ser necesario retirar manualmente uno de esos bloqueos:

```
# pct unlock <CTID>
```

PRECAUCIÓN: Utilice pct unlock únicamente si está seguro de que la operación que creó el lock ya no se está ejecutando.

CAPÍTULO COMPLETADO: Con esta Parte 11B queda traducido el Capítulo 11 completo. La siguiente parte comienza en el Capítulo 12: Software-Defined Network (SDN).

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 12A

Capítulo 12 · Software-Defined Network (SDN)

Secciones 12.1 a 12.10.6 · Introducción, zonas, VNets, subredes, controladores y fabrics

Basado en "Proxmox VE Administration Guide", Release 9.2.4
Proxmox Server Solutions GmbH · Traducción de trabajo al castellano
GNU Free Documentation License 1.3 o posterior

12 · Software-Defined Network

La función Software-Defined Network (SDN) de Proxmox VE permite crear zonas virtuales y redes virtuales (VNets). Esto simplifica configuraciones avanzadas de red y escenarios multitenant, manteniendo la definición de red bajo control del clúster.

12.1 Introducción

El SDN de Proxmox VE permite separar y controlar con gran precisión las redes de los invitados mediante configuraciones flexibles controladas por software.

La separación se estructura en tres conceptos: zonas, redes virtuales (VNets) y subredes. Una zona es un área de red virtualmente separada; un VNet es una red virtual que pertenece a una zona; y una subred representa un rango IP dentro de un VNet.

El comportamiento concreto de la red depende del tipo de zona elegido. Cada tecnología aporta funciones, ventajas y limitaciones distintas. Los casos de uso van desde una red privada aislada en cada nodo hasta overlays complejos que atraviesan varios clústeres Proxmox VE y ubicaciones diferentes.

Una vez configurado un VNet desde la administración SDN del Datacenter, el VNet aparece localmente en cada nodo como un bridge Linux común y puede asignarse a máquinas virtuales y contenedores.

Concepto	Función
Zona	Área lógica de red virtualmente separada.
VNet	Red virtual asociada a una zona; se presenta como un bridge local en los nodos.
Subred	Rango IP definido dentro de un VNet; puede aportar gateway, SNAT, IPAM y DNS.

12.2 Estado de soporte

12.2.1 Historia

La pila SDN de Proxmox VE está disponible desde 2019 y ha evolucionado de forma continua. Su integración en la interfaz web de Proxmox VE 6.2 fue un hito importante. Durante el ciclo de Proxmox VE 7 se añadieron numerosas mejoras y, a partir de Proxmox VE 8, la gestión de redes e interfaces pasó a formar parte de la pila central de control de acceso. En Proxmox VE 8.1 se incorporó la integración DHCP con IPAM y el SDN pasó a instalarse de forma predeterminada.

12.2.2 Estado actual

- El núcleo SDN, incluida la gestión de VNets y su integración con Proxmox VE, está plenamente soportado.
- IPAM, incluida la gestión DHCP para invitados virtuales, se considera “technology preview”.
- El enrutamiento complejo mediante FRRouting y la integración de controladores también se considera “technology preview”.

12.3 Instalación

12.3.1 Núcleo SDN

Desde Proxmox VE 8.1, los paquetes principales de SDN se instalan por defecto. Si el sistema se actualizó desde una versión anterior, instale libpve-network-perl en cada nodo:

```
apt update
apt install libpve-network-perl
```

NOTA: Proxmox VE 7.0 y posteriores instalan ifupdown2 por defecto. Si la instalación original procede de una versión más antigua, puede ser necesario instalar ifupdown2 explícitamente.

Compruebe además que al final de /etc/network/interfaces, en todos los nodos, esté presente la inclusión de los fragmentos de configuración:

```
source /etc/network/interfaces.d/*
```

12.3.2 DHCP IPAM

La integración DHCP de la pila IPAM incorporada usa dnsmasq para asignar leases DHCP. La función es opcional y debe activarse expresamente.

```
apt update
apt install dnsmasq
# desactivar la instancia predeterminada
systemctl disable --now dnsmasq
```

12.3.3 FRRouting

Las configuraciones SDN avanzadas usan el proyecto FRRouting (FRR). También es una función opcional. Para habilitar la integración de enrutamiento SDN, instale frr-pythontools en todos los nodos y active el servicio FRR:

```
apt update
apt install frr-pythontools
systemctl enable frr.service
```

12.4 Visión general de la configuración

La configuración se realiza a nivel Datacenter en la interfaz web y se divide en varias áreas:

Área	Finalidad
SDN	Vista del estado SDN activo y punto desde el que se aplican todos los cambios pendientes al clúster.
Zones	Creación y gestión de zonas de red virtualmente separadas.
VNets	Creación de bridges virtuales y gestión de las subredes asociadas.
Controllers	Control del enrutamiento de capa 3 en diseños complejos.
DHCP	Servidor DHCP para una zona; asigna IP mediante IPAM y las entrega al invitado por DHCP.
IPAM	Integración con gestión de direcciones IP para invitados.
DNS	Integración con un servidor DNS para registrar hostnames y direcciones IP de invitados.

12.5 Tecnología y configuración

La implementación SDN de Proxmox VE utiliza las capacidades de red estándar de Linux siempre que es posible. Así se reducen dependencias externas y el número de componentes que podrían fallar.

Las configuraciones SDN se guardan en /etc/pve/sdn y se comparten con todos los nodos mediante el sistema de archivos de configuración del clúster. Proxmox VE traduce esas definiciones a los formatos de las herramientas que gestionan la red subyacente, como ifupdown2 o FRR.

Los cambios nuevos no se aplican inmediatamente: primero se registran como pendientes. Desde la vista principal de SDN se pueden aplicar conjuntamente varios cambios como una operación coherente. El estado desplegado se registra en .running-config y .version dentro de /etc/pve/sdn.

12.6 Zonas

Una zona define una red virtualmente separada. Puede limitarse a nodos concretos y asociarse a permisos, de manera que los usuarios solo accedan a una determinada zona y a sus VNets.

Tipo	Descripción
Simple	Bridge aislado. Red de capa 3 sencilla para configuraciones enrutadas o con NAT.
VLAN	Segmentación clásica mediante etiquetas VLAN sobre un bridge Linux u OVS existente.
QinQ	Apilamiento de VLAN, formalmente IEEE 802.1ad: VLAN de servicio exterior y VLAN interior por VNet.
VXLAN	Overlay de capa 2 encapsulado en UDP sobre una red underlay existente.
EVPN	VXLAN con BGP EVPN para incorporar enrutamiento de capa 3 y despliegues entre clústeres.

12.6.1 Opciones comunes

Opción	Descripción
Nodes	Nodos en los que se desplegarán la zona y sus VNets.
IPAM	Herramienta de gestión de direcciones IP. Opcional; por defecto pve.
DNS	Servidor API DNS. Opcional.
ReverseDNS	Servidor API para DNS inverso. Opcional.
DNSZone	Dominio utilizado para registrar nombres como <hostname>.<domain>. La zona debe existir previamente en el servidor DNS.

12.6.2 Zonas Simple

Es el plugin más sencillo. Crea un bridge VNet aislado que no está conectado a ninguna interfaz física. El tráfico de las VMs permanece local a cada nodo. Puede utilizarse tanto en configuraciones NAT como enrutadas.

12.6.3 Zonas VLAN

El plugin VLAN utiliza un bridge Linux u OVS existente para enlazar con la interfaz física del nodo. Las etiquetas VLAN se definen en cada VNet y aíslan los distintos segmentos, permitiendo conectividad entre VMs situadas en nodos diferentes.

Opción	Descripción
Bridge	Bridge Linux u OVS local, ya configurado en cada nodo y con conectividad nodo-a-nodo.

12.6.4 Zonas QinQ

QinQ, o VLAN stacking, emplea varias capas de etiquetas VLAN. La zona QinQ define la VLAN exterior o Service VLAN; la etiqueta interior se define en el VNet.

NOTA: Los switches físicos deben soportar VLAN apiladas para poder utilizar esta configuración.

Opción	Descripción
Bridge	Bridge local VLAN-aware ya configurado en cada nodo.
Service VLAN	Etiqueta VLAN principal de la zona.
Service VLAN Protocol	Permite elegir entre 802.1q (predeterminado) y 802.1ad.
MTU	El apilamiento añade 4 bytes. Si la interfaz física usa MTU 1500, reduzca la MTU a 1496.

12.6.5 Zonas VXLAN

VXLAN crea un túnel (overlay) sobre una red existente (underlay). Encapsula tramas Ethernet de capa 2 dentro de datagramas UDP de capa 4, usando por defecto el puerto de destino 4789. La red underlay debe proporcionar conectividad UDP entre todos los peers.

Esto permite, por ejemplo, construir un overlay VXLAN sobre Internet de forma que las VMs perciban una misma red local de capa 2 aunque estén en ubicaciones diferentes.

ADVERTENCIA: VXLAN por sí solo no cifra el tráfico. Si une distintas ubicaciones mediante VXLAN, establezca una conexión segura entre ellas, por ejemplo una VPN site-to-site.

Opción	Descripción
Peers Address List	Lista de las direcciones IP de todos los nodos o peers de la zona VXLAN; pueden ser nodos externos alcanzables.
SDN Fabric	Puede usarse un Fabric para generar automáticamente la lista de peers.
MTU	La encapsulación VXLAN consume 50 bytes; la MTU debe ser 50 bytes menor que la de la interfaz física de salida.

12.6.6 Zonas EVPN

Una zona EVPN crea una red enrutable de capa 3 capaz de abarcar varios clústeres. Utiliza VXLAN y BGP para construir el plano de datos y el plano de control. Los VNets pueden usar una dirección IP y/o MAC anycast: el gateway del bridge es idéntico en cada nodo.

El enrutamiento entre VNets de zonas diferentes puede realizarse mediante una interfaz VRF (Virtual Routing and Forwarding).

Opción EVPN	Descripción
VRF VXLAN ID	VXLAN-ID dedicado al interconectado de enrutamiento entre VNets. Debe ser distinto de los VXLAN-ID asignados a los propios VNets.
Primary Controller	Controlador EVPN principal. Se usa para derivar automáticamente las IP VTEP a partir de la definición de peers.
Additional Controllers	Controladores EVPN adicionales que anunciarán las rutas L2VPN EVPN de la zona.
VNet MAC Address	MAC anycast asignada a todos los VNets de la zona. Se genera automáticamente si no se especifica.
Exit Nodes	Nodos que actúan como gateways de salida desde EVPN hacia la red real y anuncian una ruta por defecto.
Primary Exit Node	Fuerza el tráfico por un nodo de salida principal en vez de balancearlo. Necesario para SNAT o cuando el router upstream no soporta ECMP.
Exit Nodes Local Routing	Permite alcanzar servicios de VM/CT desde un nodo de salida; normalmente el exit node solo reenvía entre red real y EVPN.
Advertise Subnets	Anuncia la subred completa en EVPN; útil para invitados silenciosos o con varias IP que no generan tráfico hacia el gateway anycast.
Disable ARP ND Suppression	Desactiva la supresión de ARP/Neighbor Discovery. Es necesario si se usan IP/MAC flotantes que se mueven entre sistemas.
Route-target Import	Importa route targets EVPN externos para interconectar datacenters o redes EVPN diferentes.
MTU	VXLAN consume 50 bytes. El valor predeterminado es 1450.

12.7 VNets

Tras crear un VNet mediante la GUI SDN, aparece en cada nodo una interfaz de red local con el mismo nombre. Para conectar un invitado basta con asignarle esa interfaz y configurar su dirección IP.

ADVERTENCIA: En el estado actual, algunas opciones pueden no tener efecto o no funcionar en determinados tipos de zona.

Opción	Descripción
ID	Identificador del VNet de hasta 8 caracteres.
Alias	Identificador descriptivo adicional; se asigna como alias de la interfaz.
Zone	Zona a la que pertenece el VNet.
Tag	ID VLAN o VXLAN único.
VLAN Aware	Activa el modo VLAN-aware en la interfaz para permitir configuración de VLAN dentro del invitado.
Isolate Ports	Marca como aislados todos los puertos de invitados, pero no el bridge. Los invitados solo pueden hablar con puertos no aislados, es decir, con el propio bridge.

NOTA: El aislamiento de puertos es local a cada host. Para aislar tráfico del VNet entre nodos utilice el firewall del VNet, por ejemplo con política DROP y reglas explícitas hacia/desde el gateway.

12.8 Subredes

Una subred define un rango IP mediante una dirección de red CIDR. Cada VNet puede contener una o varias subredes.

- Restringir las direcciones IP que pueden definirse en un VNet.
- Asignar rutas y gateways en zonas de capa 3.
- Habilitar SNAT en zonas de capa 3.
- Asignar automáticamente IP a VM o CT mediante plugins IPAM.
- Registrar DNS mediante plugins DNS.

Si la zona tiene asociado un servidor IPAM, el prefijo de la subred se registra automáticamente en el IPAM.

Opción	Descripción
ID	Dirección de red CIDR, por ejemplo 10.0.0.0/8.
Gateway	Dirección del gateway predeterminado. En zonas de capa 3 (Simple/EVPN) se desplegará sobre el VNet.
SNAT	Activa Source NAT para permitir que las VMs del VNet salgan a redes externas a través de la interfaz de salida del nodo; en EVPN, el reenvío lo realizan los exit nodes.
DNS Zone Prefix	Prefijo opcional para registrar nombres como <hostname>.prefijo.<dominio>.

12.9 Controladores

Algunas zonas separan el plano de control del plano de datos y necesitan un controlador externo para gestionar el plano de control del VNet. En la implementación actual, la zona que requiere un controlador externo es EVPN.

12.9.1 Controlador EVPN

El controlador EVPN configura FRRouting (FRR) para gestionar el plano de control. Antes de utilizarlo, FRR debe estar habilitado en todos los nodos participantes.

Opción	Descripción
ASN #	Número ASN BGP único. Se recomienda usar ASN privados: 64512-65534 o 4200000000-4294967294. Todos los controladores EVPN configurados en un mismo nodo deben usar el mismo ASN.
SDN Fabric	Fabric que contiene todos los nodos de la zona EVPN y que se usa como red underlay.
Peers	Lista de direcciones IP de nodos o route reflectors que participan en la zona EVPN.
Nodes	Nodos concretos en los que este controlador estará activo.
Peer Group Name	EVPN usa VTEP por defecto. Si hay varios controladores, cada adicional necesita un nombre de peer group distinto.
BGP Mode	Permite fijar explícitamente el tipo de sesión BGP. El modo auto conserva el comportamiento histórico del stack SDN.
ebgp-multihop	Aumenta el número de saltos admitidos para peers no conectados directamente o que utilizan loopback; solo en modo BGP external.

ASN y coexistencia de sesiones BGP

Cada router solo puede pertenecer a un Autonomous System dentro de una misma instancia BGP. Por ello, todas las sesiones iBGP de un nodo deben usar el mismo ASN. Proxmox VE rechaza configuraciones con dos sesiones iBGP y ASN diferentes en la misma instancia.

El ASN local de FRR se deriva de la configuración de los controladores. Si solo existe un controlador BGP o EVPN se usa su ASN. Por compatibilidad, si hay controladores EVPN en modo auto y existe un controlador BGP, prevalece el ASN del controlador BGP. En configuraciones con ASN distintos y modo auto deshabilitado, el stack SDN utiliza la directiva local-as con no-prepend replace-as en FRR.

Generación automática de IP VTEP

- Si se usa un controlador BGP o IS-IS con interfaz loopback, se usa la IP del loopback como VTEP.
- Si la lista de peers contiene una IP configurada localmente en el nodo, se usa esa IP como VTEP.
- En otro caso, el controlador EVPN consulta la ruta de la tabla principal hacia el primer peer y usa la dirección origen indicada por el kernel. Esto puede fallar si esa ruta aparece dinámicamente, por ejemplo a través de un fabric; la ruta debe existir al aplicar la configuración.

12.9.2 Controlador BGP

El controlador BGP no se asocia directamente a una zona. Sirve para configurar FRR y gestionar peers BGP. En BGP-EVPN puede definir un ASN diferente por nodo para trabajar con eBGP, o exportar rutas EVPN hacia un peer BGP externo.

NOTA: Para un EVPN full-mesh sencillo no es necesario definir un controlador BGP.

Opción	Descripción
Node	Nodo al que pertenece el controlador.
ASN #	ASN BGP privado recomendado en los rangos 64512-65534 o 4200000000-4294967294.
Peer	Lista de direcciones IP de los peers BGP.
EBGP	Actívelo cuando el remote-AS del peer sea diferente.
Loopback Interface	Usa una interfaz loopback o dummy como origen de la red EVPN para multipath.
ebgp-multihop	Aumenta los saltos permitidos para peers indirectos o loopback.
bgp-multipath-as-path-relax	Permite ECMP incluso cuando los peers tienen ASN diferentes.

12.9.3 Controlador IS-IS

El controlador IS-IS tampoco se asocia directamente a una zona. Se utiliza para configurar FRR y exportar rutas EVPN a un dominio IS-IS.

Opción	Descripción
Node	Nodo en el que se ejecuta el controlador IS-IS.
Domain	Dominio IS-IS único.
Network Entity Title	Dirección de red IS-IS única que identifica al nodo.
Interfaces	Interfaces físicas usadas por IS-IS.
Loopback	Interfaz loopback o dummy usada como origen de EVPN para multipath.

12.10 Fabrics

Los Fabrics de Proxmox VE SDN proporcionan enrutamiento automatizado entre los nodos del clúster. Simplifican la configuración de las redes underlay entre nodos y crean una base resiliente y auto-configurable que se adapta a cambios de topología.

Estos fabrics pueden utilizarse como red full-mesh para Ceph o como red underlay para controladores EVPN y zonas VXLAN.

12.10.1 Instalación

Las implementaciones de OpenFabric y OSPF se apoyan en FRR. Instale los paquetes necesarios:

```
apt update
apt install frr frr-pythontools
```

12.10.2 Permisos

- Para ver la configuración de un fabric: SDN.Audit o SDN.Allocate.
- Para crear o modificar un fabric: SDN.Allocate.
- Para ver la configuración de un nodo: Sys.Audit o Sys.Modify.
- Para añadir o actualizar nodos dentro de un fabric: Sys.Modify sobre el nodo concreto, porque se escribe en /etc/network/interfaces.

12.10.3 Configuración

Para crear un Fabric, vaya a Datacenter → SDN → Fabrics y pulse "Add Fabric". Seleccione el protocolo y, con el botón "+", añ ada los nodos y las interfaces que se usarán para comunicarse con los demás nodos.

Elemento	Función
Loopback Prefix	Rango CIDR opcional, por ejemplo 192.0.2.0/24. Proxmox VE verifica que todos los router-ID pertenezcan a ese prefijo.
Router-ID	Cada nodo necesita un router-ID único. En OpenFabric también puede ser IPv6. Proxmox VE crea automáticamente una interfaz dummy que actúa como loopback y que es pasiva por defecto.
RouteMaps	Para cada fabric se generan automáticamente una access-list y un route-map. Se reescribe la dirección origen del tráfico saliente para que use la IP del loopback/dummy, proporcionando un comportamiento de enrutamiento consistente.

Notas sobre IPv6

IPv6 se admite actualmente en fabrics OpenFabric. Estos fabrics necesitan IPv6 forwarding global habilitado en todos los nodos; de lo contrario, en topologías que no sean full-mesh los nodos de tránsito no reenviarán paquetes hacia los extremos.

Puede habilitarse añadiendo la siguiente línea a una interfaz del fabric en /etc/network/interfaces:

```
post-up sysctl -w net.ipv6.conf.all.forwarding=1
```

IMPORTANTE: El forwarding IPv6 global cambia el comportamiento de SLAAC, Neighbor Advertisements, Router Solicitations y Router Advertisements. Revise estos efectos antes de aplicarlo en producción.

12.10.4 OpenFabric

OpenFabric es un protocolo de enrutamiento orientado específicamente a fabrics de centros de datos. Está basado en IS-IS y optimizado para topologías spine-leaf.

Opciones del Fabric

Opción	Descripción
Name	Nombre del fabric OpenFabric, máximo 8 caracteres.
IPv4 Prefix	Rango CIDR IPv4 usado para validar que todos los router-ID pertenezcan al prefijo.
IPv6 Prefix	Rango CIDR IPv6 usado para validar router-ID IPv6.

Hello Interval	Frecuencia, en segundos, de los paquetes hello. Menor valor detecta fallos antes, pero genera más tráfico. Predeterminado: 3 s.
CSNP Interval	Frecuencia de sincronización de la base de datos de enrutamiento con vecinos. Predeterminado: 10 s.

ADVERTENCIA: Para fabrics IPv6, el forwarding IPv6 global debe estar habilitado en todos los nodos.

Opciones del nodo

Opción	Descripción
Node	Nodo del clúster que se añade al fabric.
IPv4	Dirección IPv4 única usada para generar el Network Entity Title (NET). Cada nodo del mismo fabric necesita un router-ID distinto; un mismo nodo debe usar el mismo NET en todos sus fabrics.
IPv6	Dirección IPv6 única para generar el NET. Si se configuran IPv4 e IPv6, se utiliza IPv4 para derivar el NET.
Interfaces	Interfaces empleadas para establecer peering OpenFabric. Se recomienda seleccionar interfaces sin IP preasignada y definir las direcciones en las columnas IPv4/IPv6.

ADVERTENCIA: Con direcciones IPv6, los tres últimos segmentos se utilizan para generar el NET. Asegúrese de que difieren entre nodos.

Opciones de interfaz

Opción	Descripción
IP	IPv4 que se configurará automáticamente en la interfaz, con máscara, por ejemplo /31.
IPv6	IPv6 que se configurará automáticamente en la interfaz, con máscara, por ejemplo /127.
Hello Multiplier	Número de paquetes hello perdidos que se consideran un fallo. Valores altos toleran pérdidas, pero retrasan la detección. Predeterminado: 10.

ADVERTENCIA: Si elimina una interfaz cuya entrada en /etc/network/interfaces contiene “manual”, la IP no se elimina automáticamente al aplicar la configuración SDN.

12.10.5 OSPF

OSPF (Open Shortest Path First) es un protocolo de estado de enlace ampliamente utilizado que calcula eficientemente la ruta más corta en redes IP.

Opciones del Fabric

Opción	Descripción
IPv4 Prefix	Rango CIDR IPv4 usado para validar los router-ID.
Area	Identificador del área OSPF, como entero de 32 bits o dirección IP. El área 0/0.0.0.0 es el backbone.
Redistribute	Lista opcional de protocolos de origen cuyas rutas se importan a OSPF: bgp, connected, kernel o static. Puede asociarse un route map para filtrar o transformar las rutas.

Opciones del nodo y la interfaz

Opción	Descripción
Node	Nodo que se añade al fabric.
IPv4	Router-ID IPv4 único dentro de la red OSPF.
Interfaces	Interfaces usadas para peering OSPF. Preferiblemente sin IP preasignada; la dirección puede definirse en la columna IPv4.
IP de interfaz	IPv4 con máscara, por ejemplo /31, configurada automáticamente sobre la interfaz.

NOTA: La interfaz dummy se configura automáticamente como pasiva. Toda interfaz sin dirección IP configurada se trata como enlace punto a punto.

ADVERTENCIA: Al igual que en OpenFabric, una entrada "manual" en /etc/network/interfaces puede impedir que la IP se elimine automáticamente al retirar la interfaz del fabric.

12.10.6 WireGuard

WireGuard puede establecer una VPN entre nodos Proxmox VE y/o nodos externos. No proporciona enrutamiento dinámico por sí mismo, pero puede combinarse con protocolos de capa 3 o superior, como OSPF o BGP, para obtener un transporte cifrado con enrutamiento dinámico. Es especialmente útil para transportar EVPN o VXLAN entre ubicaciones.

NOTA: Para usar WireGuard debe estar instalado el paquete wireguard-tools.

Opciones del Fabric

Opción	Descripción
Name	Nombre del fabric WireGuard, máximo 8 caracteres.
Persistent Keepalive	Si se configura, WireGuard envía un paquete autenticado vacío cada N segundos a cada peer. Ayuda a mantener conexiones detrás de NAT o firewalls stateful.

Nodos internos

Un nodo interno es un nodo Proxmox VE del clúster. Su definición puede reutilizarse como peer en distintas interfaces del fabric.

Opción	Descripción
Endpoint	IP o hostname que los demás nodos usarán para conectarse. Se combina con el Listen Port de la interfaz; por ejemplo 192.0.2.1 + 51820 produce 192.0.2.1:51820.
Allowed IPs	Lista de CIDR separada por comas. Se utiliza como AllowedIPs de WireGuard: filtra el origen del tráfico entrante y define destinos que se enrutan hacia ese peer.

Nodos externos

Un nodo externo representa cualquier peer que no sea un nodo Proxmox VE del clúster. Estas definiciones también pueden reutilizarse en todo el clúster.

Opción	Descripción
Name	Nombre del nodo externo.
Public Key	Clave pública del peer externo.
Endpoint	IP o hostname con puerto, por ejemplo 192.0.2.1:51820. Para IPv6, la dirección debe ir entre corchetes, por ejemplo [2001:db8::1]:51820.
Allowed IPs	CIDR que se aceptan como origen del peer y se enrutan hacia él como destino.

Opciones de interfaz WireGuard

Opción	Descripción
Name	Nombre de la interfaz Linux. Entre 1 y 8 caracteres; debe comenzar por un carácter alfanumérico y el resto puede incluir guiones.
IP	Dirección IPv4 en notación CIDR, por ejemplo 198.51.100.1/24.
IPv6	Dirección IPv6 en notación CIDR, por ejemplo 2001:db8::1/64.
Listen Port	Puerto en el que escucha la interfaz.
Peers	Peers que se configurarán en la interfaz. Pueden elegirse todos los nodos del fabric y la definición se genera a partir de la configuración de cada nodo.

Al guardar una interfaz, Proxmox VE genera automáticamente una clave privada y la almacena en /etc/pve/priv/wg-keys.cfg. La clave pública correspondiente se guarda junto con la configuración del fabric para mostrarla en la interfaz web. Eliminar una interfaz también elimina de wg-keys.cfg su clave privada asociada.

El fabric crea rutas automáticamente a través de la interfaz WireGuard para los Allowed IPs de cada peer. Por ejemplo, si wg0 tiene peers con 198.51.100.0/24 y 203.0.113.0/24, se crean rutas hacia ambas subredes. Si el peer corresponde a una interfaz de otro nodo Proxmox VE, se añade además una ruta host /32 (o su equivalente IPv6) hacia la dirección concreta de esa interfaz.

Resumen de la Parte 12ª

Esta primera parte del capítulo SDN cubre la arquitectura fundamental y los bloques necesarios para diseñar la red: zonas, VN ets, subredes, controladores y fabrics. La Parte 12B continuará con Prefix Lists y Route Maps, IPAM, DNS, DHCP, integración con firewall, ejemplos de configuración y notas avanzadas.

Tecnología	Idea principal
Simple	Aislamiento local y escenarios NAT/enrutados sencillos.
VLAN	Segmentación tradicional y conectividad entre nodos mediante VLAN.
QinQ	Apilamiento de VLAN cuando se necesita una jerarquía adicional.
VXLAN	Overlay L2 sobre una underlay IP. No cifra por sí mismo.
EVPN	Overlay VXLAN con BGP para control de rutas L2/L3 y entornos multiclúster.
OpenFabric/OSPF	Construyen automáticamente la underlay enrutada entre nodos.
WireGuard	Aporta transporte cifrado; puede transportar BGP/OSPF/EVPN/VXLAN sobre una VPN.

IMPORTANTE: Antes de aplicar cambios SDN en producción, valide MTU, rutas, forwarding y conectividad entre nodos. En tecnologías overlay, una configuración incorrecta de underlay o del tamaño MTU puede provocar fallos difíciles de diagnosticar.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 12B

Software-Defined Network (SDN)

Prefix Lists y Route Maps · BGP · IPAM · DNS · DHCP · Firewall · Ejemplos · Notas avanzadas

Política de rutas	Servicios SDN	Seguridad	Ejemplos
Prefix Lists + Route Maps + BGP	IPAM + DNS + DHCP	IPSets + Firewall	Simple · SNAT · VLAN · QinQ · VXLAN · EVPN · WireGuard

Basado en Proxmox VE Administration Guide, Release 9.2.4

Traducción progresiva al castellano · Se conservan comandos, rutas, identificadores y sintaxis técnica.

12.10.6 WireGuard — continuación

Opciones del peer

Omitir generación de rutas (Skip Route Generation) — De forma predeterminada, el Fabric genera automáticamente rutas en la tabla de encaminamiento del kernel para todas las Allowed IPs de un peer. Si se activa esta opción, no se insertarán dichas rutas en la tabla del kernel.

12.11 Prefix Lists y Route Maps

Las listas de prefijos y los mapas de rutas permiten controlar qué rutas se intercambian entre los nodos Proxmox VE y sus peers BGP, EVPN, OSPF u OpenFabric, así como modificar atributos de esas rutas al entrar o salir.

Ambos mecanismos se complementan: una Prefix List clasifica rutas según su CIDR de destino, mientras que un Route Map encadena condiciones de coincidencia y acciones en forma de política ordenada.

Si ya ha utilizado ip prefix-list y route-map en FRR, Cisco IOS o Juniper, las entidades de Proxmox VE siguen la misma lógica. Proxmox VE administra la configuración como entidades de ámbito de clúster bajo /etc/pve/sdn/ y las traduce a bloques de configuración FRR cuando se aplican los cambios SDN. La configuración FRR combinada termina en /etc/frr/frr.conf; las adiciones locales del nodo en /etc/frr/frr.conf.local se anexan posteriormente.

NOTA: Como en el resto de SDN, los cambios quedan primero en estado pendiente y solo se activan al aplicar la configuración desde el panel principal de SDN.

12.11.1 Prefix Lists

Una Prefix List es una lista ordenada y con nombre. Cada entrada compara un único CIDR de destino —o un rango de longitudes de prefijo dentro de ese CIDR— y lo permite o lo deniega. FRR evalúa las entradas según su número de secuencia y se detiene en la primera coincidencia. Si ninguna entrada coincide, existe una denegación implícita.

Las Prefix Lists se administran en Datacenter -> SDN -> Prefix Lists. Cada lista tiene un nombre y una o más entradas:

Número de secuencia (Sequence Nr.) — Orden de evaluación de la entrada. Debe ser único dentro de la lista y como mínimo 1. Si se deja vacío al crearla, Proxmox VE asigna automáticamente el siguiente número libre siguiendo max + 5, dejando huecos para futuras inserciones.

Acción (Action) — permit o deny. Al final de cada lista existe un deny implícito.

Prefijo (Prefix) — Red de destino en notación CIDR, por ejemplo 192.0.2.0/24 o 2001:db8::/32. 0.0.0.0/0 y ::/0 son válidos y representan la ruta por defecto.

Prefix >= (ge) — Longitud mínima opcional. La entrada coincide con cualquier subred del prefijo cuya longitud sea al menos ge.

Prefix <= (le) — Longitud máxima opcional. La entrada coincide con cualquier subred del prefijo cuya longitud sea como máximo le.

Sin le ni ge, la coincidencia es exacta. Con ambos, se admite cualquier subred cuya longitud esté dentro del intervalo indicado. Por ejemplo, 10.0.0.0/8 ge 16 le 24 coincide con todas las subredes /16 a /24 dentro de 10.0.0.0/8. Proxmox VE valida que ge no sea mayor que le, que ninguno sea menor que la longitud del propio prefijo y que ambos respeten el límite de la familia de direcciones (/32 en IPv4 y /128 en IPv6).

NOTA: Los nombres only_default, only_default_v6 y loopbacks_ips están reservados. Los controladores BGP y EVPN los generan automáticamente y no pueden utilizarse como nombres de Prefix Lists definidas por el usuario.

12.11.2 Route Maps

Un Route Map es una lista ordenada de entradas que compara rutas entrantes o salientes con determinados criterios, puede modificar atributos —por ejemplo métrica o preferencia local— y decide si la ruta se permite o se deniega. FRR procesa las entradas por orden; la primera coincidencia gobierna la acción. Al final existe un deny implícito para cualquier ruta no coincidente.

Se gestionan en Datacenter -> SDN -> Route Maps. Cada entrada dispone de:

Order — Orden de evaluación, entre 0 y 65535. Es obligatorio y único dentro del Route Map. Es recomendable dejar huecos, por ejemplo 10, 20, 30, para poder insertar reglas después.

Action — permit deja pasar la ruta —después de aplicar las acciones Set— y deny la descarta.

Match — Una o más condiciones. Claves disponibles: route-type, vni, ip-address-prefix-list, ip6-address-prefix-list, ip-next-hop-prefix-list, ip6-next-hop-prefix-list, ip-next-hop-address, ip6-next-hop-address, metric, local-preference, peer y tag. Las claves *-prefix-list hacen referencia a una Prefix List por nombre. tag compara un atributo de etiqueta de ruta de 32 bits. Todas las condiciones de una entrada deben cumplirse.

Set — Atributos que se modifican cuando la entrada coincide. Claves: ip-next-hop, ip-next-hop-peer-address, ip-next-hop-unchanged, ip6-next-hop, ip6-next-hop-peer-address, ip6-next-hop-prefer-global, local-preference, tag, weight, metric y src. metric admite N, +N, -N, rtt, +rtt, -rtt, igp y aigp. tag admite un entero o el literal untagged.

Exit Action — Controla cómo continúa la evaluación tras una coincidencia. Vacío termina el procesamiento. on-match-next continúa con la entrada siguiente; on-match-goto <order> salta a la primera entrada cuyo orden sea mayor o igual; continue <order> salta exactamente al orden indicado.

Call — Llama opcionalmente a otro Route Map al final del procesamiento de la entrada, equivalente a la directiva call de FRR.

NOTA: No puede eliminarse un Route Map utilizado por un controlador. Elimine antes la referencia. Los nombres MAP_VTEP_IN, MAP_VTEP_OUT, correct_src y cualquier nombre que empiece por pve_ están reservados.

12.11.3 Dónde se utilizan Prefix Lists y Route Maps

Los controladores BGP y EVPN admiten un Route Map de entrada y otro de salida. El de entrada se aplica a las rutas recibidas antes de instalarlas; el de salida, antes de anunciarlas. Ambos son opcionales.

Los Fabrics OSPF y OpenFabric admiten una opción Route Filter que referencia una Prefix List. Si se define, solo se instalan en la tabla de rutas del kernel los destinos que superan ese filtro. Sin un filtro explícito, el Fabric genera uno a partir de sus prefijos IPv4 e IPv6. Un filtro personalizado resulta útil, por ejemplo, para permitir una ruta por defecto aprendida desde un nodo de borde.

12.11.4 Ejemplos

Permitir únicamente la ruta IPv4 por defecto:

```
# /etc/pve/sdn/prefix-lists.cfg
prefix-list: default-only
entries action=permit,prefix=0.0.0.0/0,seq=10
```

Permitir todas las subredes /16 a /24 dentro de 10.0.0.0/8 y denegar el resto del bloque:

```
prefix-list: internal-subnets
entries action=permit,prefix=10.0.0.0/8,ge=16,le=24,seq=10
entries action=deny,prefix=10.0.0.0/8,le=32,seq=20
```

Política BGP de salida que reduce la preferencia local a 50 en las rutas coincidentes con internal-subnets y deja pasar el resto sin cambios:

```
# /etc/pve/sdn/route-maps.cfg
route-map-entry: bgp-out_10
action permit
match key=ip-address-prefix-list,value=internal-subnets
set key=local-preference,value=50

route-map-entry: bgp-out_20
action permit
```

NOTA: La entrada permit final, sin condiciones Match, implementa el patrón FRR «permitir todo lo demás». Si se omite, el deny implícito del final descartaría las rutas no coincidentes.

12.11.5 BGP

BGP puede utilizarse como un Fabric eBGP unnumbered. Cada nodo dispone de su propio ASN y establece vecindad con los nodos adyacentes a través de interfaces físicas sin requerir direcciones IP en los enlaces del Fabric.

Opciones del Fabric

IPv4 Prefix — Rango CIDR IPv4 utilizado para verificar que las direcciones IPv4 de los nodos —router IDs BGP— pertenecen al prefijo.

IPv6 Prefix — Rango CIDR IPv6 utilizado para verificar que las direcciones IPv6 de los nodos pertenecen al Fabric.

BFD — Activa Bidirectional Forwarding Detection en las sesiones de peering para detectar fallos de enlace con rapidez.

Route Filter — Prefix List aplicada en entrada al peer-group del Fabric. Sustituye el filtro implícito que obliga a permanecer dentro del prefijo del Fabric. Permite ampliar o expresar de forma más precisa el conjunto aceptado.

Incoming Route Map — Route Map encadenado tras el filtro de entrada implícito mediante la acción call de FRR. Puede transformar o restringir más las rutas aceptadas, pero no ampliar el conjunto previamente permitido.

Outgoing Route Map — Route Map aplicado a las rutas anunciadas a los vecinos del Fabric.

Redistribute — Protocolos cuyas rutas se redistribuyen hacia BGP. Fuentes admitidas: connected, kernel, static y ospf. Puede asociarse un Route Map a cada fuente.

NOTA: Debe configurarse al menos uno de los prefijos IPv4 Prefix o IPv6 Prefix.

Opciones del nodo

Node — Nodo que se incorpora al Fabric.

ASN — Número de Sistema Autónomo BGP para el nodo. Se recomienda utilizar ASNs privados: 64512-65534 (16 bits) o 4200000000-4294967294 (32 bits).

IPv4 — Dirección IPv4 única del nodo, utilizada también como router-id BGP. Es obligatoria si el Fabric define prefijo IPv4.

IPv6 — Dirección IPv6 del nodo. Si solo existe IPv6, el router-id BGP se deriva mediante un hash FNV-1a.

Interfaces — Interfaces utilizadas para el peering BGP unnumbered. No necesitan direcciones IP; el peering se establece automáticamente mediante direcciones link-local IPv6.

NOTA: En un Fabric eBGP unnumbered, los nodos conectados directamente deben utilizar ASNs diferentes.

Uso de BGP Fabrics con EVPN

Al configurar un controlador EVPN, puede seleccionarse un BGP Fabric como underlay en lugar de definir manualmente las direcciones de los peers. Por defecto, el overlay EVPN —sesiones VTEP— funciona como iBGP usando el ASN del controlador EVPN, mientras que el ASN por nodo del Fabric se aplica mediante local-as al grupo de vecinos del underlay.

Por ello, el ASN del controlador EVPN y los ASNs por nodo del Fabric deberían ser distintos. Ejemplo: si los tres nodos del underlay usan 65001, 65002 y 65003, el controlador EVPN puede utilizar 65000 para el overlay.

Si BGP Mode se establece en external en el controlador EVPN, las sesiones VTEP funcionan como eBGP. El ASN del Fabric de cada nodo se utiliza también como ASN local del peer-group VTEP. El ASN del controlador EVPN sigue empleándose para generar automáticamente los route-targets.

NOTA: Usar un BGP Fabric como underlay de EVPN exige que cada nodo tenga una dirección IPv4, ya que EVPN la utiliza como dirección VTEP.

12.12 IPAM

Las herramientas IP Address Management (IPAM) administran las direcciones IP de los clientes de la red. En SDN, Proxmox VE utiliza IPAM, por ejemplo, para localizar direcciones libres al crear nuevos invitados. Una misma instancia IPAM puede asociarse a una o varias zonas.

12.12.1 Plugin PVE IPAM

Es el IPAM integrado y predeterminado del clúster Proxmox VE. Su estado se consulta desde el panel IPAM de la sección SDN del Datacenter. Desde la interfaz es posible crear, actualizar y eliminar asignaciones IP, especialmente útil en combinación con DHCP.

Si se utiliza DHCP, el panel IPAM permite crear o editar leases para máquinas virtuales concretas. Después de cambiar la IP de una VM que usa DHCP, debe forzarse al invitado a solicitar un nuevo lease, normalmente recargando su pila de red o reiniciándolo.

12.12.2 Plugin NetBox IPAM

NetBox es una herramienta open source de IPAM y DCIM. Para integrarla con Proxmox VE SDN debe crearse un token de API en NetBox.

URL — Endpoint de la API REST de NetBox, por ejemplo http://yournetbox.domain.com/api

Token — Token de acceso a la API.

Fingerprint — Huella SHA-256 del certificado de la API de NetBox. Con el certificado predeterminado puede obtenerse con:

```
openssl x509 -in /etc/ssl/certs/netbox.crt -noout -fingerprint -sha256
```

12.12.3 Plugin phpIPAM

En phpIPAM debe crearse una «application» y añadirle un token de API con privilegios de administrador.

URL — Endpoint REST, por ejemplo http://phpipam.domain.com/api/<appname>/

Token — Token de acceso a la API.

Section — ID entero de sección. Las secciones agrupan subredes en phpIPAM. En instalaciones predeterminadas suele utilizarse sectionid=1.

12.13 DNS

El plugin DNS de Proxmox VE SDN define un servidor con API DNS para registrar nombres de host y direcciones IP. Una configuración DNS puede asociarse a una o varias zonas para proporcionar registro DNS a todas las IP de subred configuradas en esas zonas.

12.13.1 Plugin PowerDNS

En PowerDNS es necesario habilitar el servidor web y la API:

```
api=yes
api-key=arandomgeneratedstring
webserver=yes
webserver-port=8081
```

url — Endpoint REST, por ejemplo http://yourpowerdnserver.domain.com:8081/api/v1/servers/localhost

key — Clave de acceso a la API.

ttl — TTL predeterminado de los registros.

12.14 DHCP

El plugin DHCP de Proxmox VE SDN permite desplegar automáticamente un servidor DHCP para una zona. Proporciona DHCP a todas las subredes de la zona que tengan configurado un rango DHCP. Actualmente el único backend disponible es dnsmasq.

Cuando se añade una nueva interfaz de red a una VM o CT, el plugin DHCP reserva una IP en el IPAM configurado para la zona. Al arrancar el invitado se crea la asociación entre MAC e IP. Si se elimina la interfaz o se destruye la VM/CT, también se eliminan las entradas correspondientes del IPAM y del servidor DHCP.

NOTA: Algunas operaciones —crear, editar o eliminar mapeos IP— solo están disponibles actualmente con el plugin PVE IPAM.

12.14.1 Configuración

El DHCP automático puede habilitarse desde la interfaz web en el panel Zones, activando DHCP dentro de las opciones avanzadas de la zona.

NOTA: Actualmente solo las Simple Zones admiten DHCP automático.

Después de habilitarlo, deben configurarse rangos DHCP en las subredes de la zona. En la GUI, vaya a VNets, seleccione la Subnet y defina los rangos en la pestaña correspondiente. También puede utilizarse la CLI:

```
pvesh set /cluster/sdn/vnets/<vnet>/subnets/<subnet> \
-dhcp-range start-address=10.0.1.100,end-address=10.0.1.200 \
-dhcp-range start-address=10.0.2.100,end-address=10.0.2.200
```

La subred debe tener configurado un gateway; de lo contrario, DHCP automático no funcionará. El plugin solo asignará direcciones IP dentro de los rangos definidos. También deben completarse los pasos de instalación del plugin dnsmasq.

12.14.2 Plugins

Plugin dnsmasq

Es actualmente el único plugin DHCP, por lo que es el que se utiliza al habilitar DHCP en una zona.

El plugin crea un servicio systemd por cada zona: dnsmasq@<zone>. El ciclo de vida del servicio queda administrado por el propio plugin DHCP.

También genera automáticamente los siguientes archivos bajo /etc/dnsmasq.d/<zone>:

00-default.conf — Configuración global predeterminada de la instancia dnsmasq.

10-<zone>-<subnet_cidr>.conf — Opciones específicas de una subred, como el servidor DNS entregado por DHCP.

10-<zone>-<subnet_cidr>.ranges.conf — Rangos DHCP de la instancia.

ethers — Asociaciones MAC-IP procedentes del plugin IPAM. No debe editarse manualmente, ya que será sobrescrito.

IMPORTANTE: No edite los archivos anteriores. Si necesita personalizar dnsmasq, cree archivos adicionales —por ejemplo 90-custom.conf— dentro del mismo directorio. Se leen en orden alfabético.

Los leases DHCP se guardan en /var/lib/misc/dnsmasq.<zone>.leases. Con PVE IPAM pueden actualizarse, crearse y borrarse leases desde la gestión IPAM; esta capacidad no está disponible actualmente para los demás plugins IPAM.

12.15 Integración con el firewall

SDN se integra con el firewall de Proxmox VE generando automáticamente IPSets que pueden utilizarse en los campos de origen y destino de las reglas. Esto se realiza automáticamente tanto para VNets como para entradas IPAM.

12.15.1 VNets y Subnets

Para cada VNet, el firewall genera en el ámbito SDN los siguientes IPSets:

vnet-all — CIDRs de todas las subredes de la VNet.

vnet-gateway — Direcciones IP de los gateways de todas las subredes.

vnet-no-gateway — CIDRs de las subredes excluyendo las direcciones de gateway.

vnet-dhcp — Todos los rangos DHCP configurados en las subredes.

Estos IPSets se actualizan automáticamente cuando cambia la configuración, por lo que las reglas de firewall no tienen que modificarse manualmente.

Ejemplo de Simple Zone

```
# /etc/pve/sdn/vnets.cfg
vnet: vnet0
zone simple

# /etc/pve/sdn/subnets.cfg
subnet: simple-192.0.2.0-24
vnet vnet0
dhcp-range start-address=192.0.2.100,end-address=192.0.2.199
gateway 192.0.2.1

subnet: simple-2001:db8::-64
vnet vnet0
dhcp-range start-address=2001:db8::1000,end-address=2001:db8::1999
gateway 2001:db8::1
```

Con esta configuración, los IPSets resultantes serían:

```
vnet0-all
  192.0.2.0/24
  2001:db8::/64

vnet0-gateway
  192.0.2.1
  2001:db8::1

vnet0-no-gateway
  192.0.2.0/24
  2001:db8::/64
  !192.0.2.1
  !2001:db8::1

vnet0-dhcp
  192.0.2.100 - 192.0.2.199
  2001:db8::1000 - 2001:db8::1999
```

12.15.2 IPAM

Con PVE IPAM, el firewall genera automáticamente un IPSet para cada invitado que tenga entradas IPAM. Para un invitado con ID 100 se denomina guest-ipam-100 e incluye todas sus direcciones IP, aunque pertenezcan a varias VNets.

ADVERTENCIA: Si se eliminan todas las entradas de un invitado pero siguen existiendo reglas que referencian el IPSet generado, el firewall no podrá actualizar el ruleset porque el IPSet habrá dejado de existir.

12.16 Ejemplos

Esta sección reúne configuraciones habituales de SDN para mostrar de forma práctica cómo se combinan zonas, VNets, subredes, encapsulación y routing.

12.16.1 Ejemplo de Simple Zone

Una Simple Zone crea una red aislada para que los invitados de un mismo host se comuniquen entre sí.

CONSEJO: La comunicación es posible entre invitados que residen en el mismo host; no se extiende automáticamente a otros nodos.

- Crear una Simple Zone llamada simple.
- Añadir una VNet llamada vnet1.
- Crear una Subnet con gateway y habilitar SNAT si se desea salida.
- Esto crea el bridge vnet1 en el nodo. Asigne ese bridge a los invitados y configure sus IP.

Ejemplo de configuración de dos invitados en 10.0.1.0/24:

```
allow-hotplug ens19
iface ens19 inet static
address 10.0.1.14/24

allow-hotplug ens19
iface ens19 inet static
address 10.0.1.15/24
```

12.16.2 Ejemplo de Source NAT

Para permitir conexiones salientes desde una Simple Zone puede utilizarse la opción Source NAT (SNAT). Partiendo de la configuración anterior, añada una Subnet a vnet1, establezca el gateway y active SNAT:

```
Subnet: 172.16.0.0/24
Gateway: 172.16.0.1
SNAT: checked
```

Configure en los invitados direcciones estáticas dentro del rango. El nodo se unirá a la red con 172.16.0.1 y actuará como gateway NAT.

12.16.3 Ejemplo de VLAN

Cuando VMs alojadas en distintos nodos necesitan comunicarse a través de una red aislada, una VLAN Zone proporciona separación mediante tags VLAN.

```
ID: myvlanzone
Bridge: vmbr0
```

Crear una VNet con VLAN tag 10:

```
ID: myvnet1
Zone: myvlanzone
Tag: 10
```

Aplique la configuración desde el panel principal de SDN. Cree vm1 en node1 y vm2 en node2, ambas conectadas a myvnet1:

```
# vm1
auto eth0
iface eth0 inet static
address 10.0.3.100/24

# vm2
auto eth0
iface eth0 inet static
address 10.0.3.101/24
```

Tras ello debería existir conectividad IP entre ambas VMs.

12.16.4 Ejemplo de QinQ

QinQ añade una segunda capa de etiquetas VLAN. Un caso típico es un proveedor que ofrece redes aisladas a diferentes clientes aunque estos reutilicen la misma VLAN interna.

```
# Zona 1
ID: qinqzone1
Bridge: vmbr0
Service VLAN: 20

# Zona 2
ID: qinqzone2
Bridge: vmbr0
Service VLAN: 30

# VNet sobre zona 1
ID: qinqvnet1
Zone: qinqzone1
Tag: 100

# VNet sobre zona 2
ID: qinqvnet2
Zone: qinqzone2
Tag: 100
```

Cree cuatro VMs: vm1 y vm2 en qinqvnet1; vm3 y vm4 en qinqvnet2. Puede utilizar, por ejemplo, direcciones 10.0.3.101/24 a 10.0.3.104/24. vm1 y vm2 podrán comunicarse entre sí, al igual que vm3 y vm4, pero no entre ambos grupos porque utilizan Service VLAN diferentes.

12.16.5 Ejemplo de VXLAN

Se asume un clúster de tres nodos: 192.168.0.1, 192.168.0.2 y 192.168.0.3.

```
ID: myvxlanzone
Peers Address List: 192.168.0.1,192.168.0.2,192.168.0.3

ID: vxvnet1
Zone: myvxlanzone
Tag: 100000
```

Tras aplicar SDN, conecte una VM de node1 y otra de node3 a vxvnet1. Debido a la encapsulación, el ejemplo utiliza MTU 1450:

```
# vm1
auto eth0
iface eth0 inet static
address 10.0.3.100/24
mtu 1450

# vm2
auto eth0
iface eth0 inet static
address 10.0.3.101/24
mtu 1450
```

Las dos máquinas deberían poder comunicarse a través del overlay VXLAN.

12.16.6 Ejemplo de EVPN

Se asume un clúster de tres nodos con IP 192.168.0.1, 192.168.0.2 y 192.168.0.3.

```
# Controlador EVPN
ID: myevpnctl
ASN#: 65000
Peers: 192.168.0.1,192.168.0.2,192.168.0.3

# Zona EVPN
ID: myevpnzone
VRF VXLAN Tag: 10000
Controller: myevpnctl
MTU: 1450
VNet MAC Address: 32:F4:05:FE:6C:0A
Exit Nodes: node1,node2

# VNet 1
ID: myvnet1
Zone: myevpnzone
Tag: 11000
Subnet: 10.0.1.0/24
Gateway: 10.0.1.1

# VNet 2
ID: myvnet2
Zone: myevpnzone
Tag: 12000
Subnet: 10.0.2.0/24
Gateway: 10.0.2.1
```

Aplique la configuración para crear las VNets y generar la configuración FRR. Ejemplo de vm1 en myvnet1:

```
auto eth0
iface eth0 inet static
address 10.0.1.100/24
gateway 10.0.1.1
mtu 1450
```

Ejemplo de vm2 en myvnet2:

```
auto eth0
iface eth0 inet static
address 10.0.2.100/24
gateway 10.0.2.1
mtu 1450
```

Las VMs deberían poder comunicarse entre VNets a través del routing EVPN. Si vm2 se encuentra en un nodo que no es de salida, el tráfico hacia una IP externa se dirige al gateway de myvnet2, luego a uno de los exit nodes y finalmente sale por el gateway real de node1 o node2.

NOTA: En el gateway externo deben añadirse rutas de retorno para 10.0.1.0/24 y 10.0.2.0/24 hacia node1/node2. Si existe un router BGP externo, las rutas BGP-EVPN pueden anunciarse dinámicamente.

12.16.7 Ejemplo de WireGuard

El ejemplo utiliza tres nodos Proxmox —sdn1, sdn2 y sdn3— con 192.0.2.1, 192.0.2.2 y 192.0.2.3, más un peer externo sdn-router con 192.0.2.10. Los nodos y el peer externo formarán un full mesh WireGuard sobre 198.51.100.0/24. Además, 203.0.113.0/24 se encaminará a través del peer externo 198.51.100.10.

Crear el WireGuard Fabric

En Datacenter -> SDN -> Fabrics, cree un Fabric WireGuard. En el ejemplo se utiliza el nombre example.

Configurar interfaces WireGuard

Cada nodo participante necesita una interfaz WireGuard. Añada un nodo de tipo Internal al Fabric, seleccione el nodo correspondiente y cree una interfaz WireGuard.

En el nodo se configuran Endpoint y Allowed IPs. Para sdn1, el Endpoint es 192.0.2.1. Como no se enrutan subredes adicionales por ese nodo, Allowed IPs queda vacío. Cree wg0 con 198.51.100.1/24. El puerto predeterminado es 51820. Proxmox VE genera automáticamente un par de claves pública/privada para cada interfaz.

Configurar el nodo externo

Añada otro nodo de tipo External. Introduzca su clave pública, el endpoint 192.0.2.10:51820 y las Allowed IPs 198.51.100.10/32 y 203.0.113.0/24.

Configurar peerings WireGuard

Una vez definidas las interfaces y peers externos, pueden seleccionarse como peers de cada interfaz. Para wg0 en sdn1, añada sdn2, sdn3 y sdn-router. El resultado genera un fichero como /etc/wireguard/proxmox/wg0.conf:

```
[Interface]
PrivateKey = <clave-privada-generada>
ListenPort = 51820

[Peer]
PublicKey = <clave-publica-sdn-router>
AllowedIPs = 198.51.100.10/32, 203.0.113.0/24
Endpoint = 192.0.2.10:51820

[Peer]
PublicKey = <clave-publica-sdn2>
AllowedIPs = 198.51.100.2/32
Endpoint = 192.0.2.2:51820

[Peer]
PublicKey = <clave-publica-sdn3>
AllowedIPs = 198.51.100.3/32
Endpoint = 192.0.2.3:51820
```

NOTA: El manual original muestra claves WireGuard de ejemplo. En esta traducción se representan como marcadores para evitar que parezcan material que deba reutilizarse. Proxmox VE genera las claves reales automáticamente.

Verificación

Compruebe conectividad con los participantes del Fabric mediante ping:

```
ping -c1 198.51.100.2
ping -c1 198.51.100.3
ping -c1 198.51.100.10
```

12.17 Notas

12.17.1 Múltiples nodos de salida EVPN

Si existen varios gateway nodes, se recomienda deshabilitar rp_filter (Strict Reverse Path Filter), ya que un paquete puede entrar por un nodo y salir por otro.

Cree /etc/sysctl.d/99-rp-filter.conf:

```
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.all.rp_filter=0
```

12.17.2 Cifrado IPsec sobre VXLAN

Para añadir cifrado IPsec sobre VXLAN, el manual muestra un ejemplo con strongSwan. Debe reducirse adicionalmente la MTU: 60 bytes para IPv4 u 80 bytes para IPv6. Con una MTU física de 1500, el ejemplo utiliza 1370 para contemplar 80 bytes de IPsec más 50 bytes de VXLAN.

Instale strongSwan:

```
apt install strongswan
```

Añada configuración a /etc/ipsec.conf. Solo se cifra el tráfico UDP del puerto VXLAN 4789:

```
conn %default
    ike=aes256-sha1-modp1024!
    esp=aes256-sha1!
    leftfirewall=yes

conn output
    rightsubnet=%dynamic[udp/4789]
    right=%any
    type=transport
    authby=psk
    auto=route

conn input
    leftsubnet=%dynamic[udp/4789]
    type=transport
    authby=psk
    auto=route
```

Genere una clave precompartida:

```
openssl rand -base64 128
```

Añádala a /etc/ipsec.secrets:

```
: PSK <generatedbase64key>
```

Copie la PSK y la configuración a todos los nodos que participen en la red VXLAN.

Fin del Capítulo 12 · Software-Defined Network

Con esta Parte 12B queda completada la traducción del Capítulo 12. La siguiente entrega comienza el Capítulo 13 · Proxmox VE Firewall.

CRITERIO DE TRADUCCIÓN: Se han conservado sin traducir los nombres de comandos, rutas, claves de configuración, campos de API, identificadores y valores que deben introducirse literalmente. Los nombres visibles de opciones de la interfaz se traducen cuando mejora la comprensión, conservando el término original cuando es útil.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 13A

Capítulo 13 · Proxmox VE Firewall

Zonas y direcciones · Archivos de configuración · Reglas · Security Groups · Aliases · IP Sets · Servicios · Reglas predeterminadas · Logging

Zona	IN	OUT	FORWARD
Host	Sí	Sí	Sí*
VM/CT	Sí	Sí	No
VNet	No	No	Sí*

** FORWARD y VNet requieren el nuevo firewall basado en nftables.*

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, rutas, nombres de parámetros y sintaxis de configuración.

Capítulo 13 · Proxmox VE Firewall

Proxmox VE Firewall proporciona una forma sencilla de proteger la infraestructura TI. Puede definir reglas para todos los hosts de un clúster o reglas específicas para máquinas virtuales y contenedores. Funciones como macros de firewall, grupos de seguridad, conjuntos de IP y aliases facilitan la administración de políticas complejas.

Aunque toda la configuración se almacena en el sistema de archivos del clúster, el servicio de firewall basado en iptables se ejecuta en cada nodo. Esto proporciona aislamiento completo entre máquinas virtuales y, por su naturaleza distribuida, permite un ancho de banda muy superior al de una solución de firewall centralizada.

El firewall ofrece soporte completo para IPv4 e IPv6. El soporte IPv6 es transparente: de forma predeterminada se filtra el tráfico de ambos protocolos, por lo que no es necesario mantener un conjunto independiente de reglas para IPv6.

13.1 Direcciones y zonas

El firewall de Proxmox VE divide la red en varias zonas lógicas. Las reglas se definen de forma independiente para cada zona. Según la zona, pueden crearse reglas para tráfico entrante, saliente o reenviado.

13.1.1 Direcciones

Elemento	Descripción
IN	Tráfico que llega a una zona.
OUT	Tráfico que abandona una zona.
FORWARD	Tráfico que atraviesa una zona. En la zona Host puede ser tráfico enrutado cuando el host actúa como gateway o realiza NAT. A nivel de VNet afecta a todo el tráfico que atraviesa la VNet, incluido el tráfico desde/hacia interfaces de red puenteadas.

IMPORTANTE: La creación de reglas para tráfico FORWARD solo está disponible actualmente con el nuevo proxmox-firewall basado en nftables. Las reglas FORWARD son ignoradas por el pve-firewall clásico y no tienen efecto.

13.1.2 Zonas

Host — Tráfico desde/hacia un host o reenviado por un host. Las reglas pueden definirse a nivel Datacenter o a nivel de host. Las reglas del host tienen prioridad sobre las reglas del Datacenter.

VM — Tráfico desde/hacia una VM o CT. Solo se pueden definir reglas de entrada y salida; no reglas FORWARD.

VNet — Tráfico que atraviesa una VNet SDN, ya sea entre invitados o entre host e invitado. Como siempre es tráfico reenviado, únicamente admite la dirección FORWARD.

IMPORTANTE: Las reglas a nivel VNet solo funcionan actualmente con el nuevo proxmox-firewall basado en nftables. El pve-firewall clásico las ignora.

13.2 Archivos de configuración

Toda la configuración relacionada con el firewall se almacena en el sistema de archivos de clúster de Proxmox. Por ello, los archivos se distribuyen automáticamente a todos los nodos y el servicio pve-firewall actualiza las reglas iptables subyacentes cuando se detectan cambios.

Puede realizar toda la configuración desde la GUI —por ejemplo, Datacenter → Firewall o Node → Firewall— o editar directamente los archivos con su editor preferido.

Los archivos contienen secciones de pares clave/valor. Las líneas que comienzan por # y las líneas vacías se consideran comentarios. Las secciones comienzan con una cabecera cuyo nombre aparece entre [y].

13.2.1 Configuración global del clúster

La configuración global del firewall se almacena en:

```
/etc/pve/firewall/cluster.fw
```

Puede contener las siguientes secciones:

[OPTIONS] — Opciones globales del firewall para todo el clúster.

[RULES] — Reglas de firewall de ámbito de clúster para todos los nodos.

[IPSET <name>] — Definiciones de conjuntos IP de ámbito de clúster.

[GROUP <name>] — Definiciones de Security Groups de ámbito de clúster.

[ALIASES] — Definiciones de aliases de IP de ámbito de clúster.

Opciones de [OPTIONS]

ebtables: <boolean> (default = 1) — Activa las reglas ebtables en todo el clúster.

enable: <integer> (0 - N) (default = 0) — Activa o desactiva globalmente el firewall.

log_ratelimit: [enable=]<1|0> [,burst=<integer>] [,rate=<rate>] — Configuración del límite de frecuencia para el logging.

burst=<integer> (0 - N) (default = 5) — Ráfaga inicial de paquetes que siempre se registrarán antes de aplicar el límite.

enable=<boolean> (default = 1) — Activa o desactiva la limitación de frecuencia del logging.

rate=<rate> (default = 1/second) — Frecuencia de recarga del bucket de ráfaga.

policy_forward: <ACCEPT | DROP> — Política para tráfico reenviado.

policy_in: <ACCEPT | DROP | REJECT> — Política de entrada.

policy_out: <ACCEPT | DROP | REJECT> — Política de salida.

Activar el firewall

El firewall está completamente desactivado de forma predeterminada. Para activarlo en todo el clúster, establezca enable en 1:

```
[OPTIONS]
# enable firewall (cluster-wide setting, default is disabled)
enable: 1
```

IMPORTANTE: Al habilitar el firewall, el tráfico hacia todos los hosts queda bloqueado por defecto, salvo determinadas excepciones para tráfico procedente de la red local hacia la WebUI, SSH y otros servicios esenciales. Los servicios adicionales que necesiten red deben permitirse de forma explícita.

Si administra los hosts Proxmox VE desde una red remota, debe crear reglas que permitan el acceso a la GUI web —puerto 8006— desde las IP remotas. Es habitual permitir también SSH —puerto 22— y, si se utiliza, SPICE —puerto 3128—.

CONSEJO: Antes de activar el firewall, mantenga abierta una sesión SSH con uno de los hosts Proxmox VE. Así conservará acceso al sistema si una regla impide el acceso a la interfaz web.

Como alternativa, puede crear un IPSet denominado management y añadir allí todas las IP remotas de administración. Esto genera las reglas necesarias para el acceso remoto a la GUI.

13.2.2 Configuración específica de host

La configuración específica de cada host se lee desde:

```
/etc/pve/nodes/<nodename>/host.fw
```

Es útil para sobrescribir reglas procedentes de cluster.fw, aumentar la verbosidad del registro o ajustar opciones relacionadas con netfilter. Puede contener las siguientes secciones:

[OPTIONS] — Opciones de firewall específicas del host.

[RULES] — Reglas de firewall específicas del host.

Opciones de host

enable: <boolean> (default = 1) — Activa las reglas de firewall del host.

log_level_forward — Nivel de log para tráfico reenviado: alert, crit, debug, emerg, err, info, nolog, notice o warning.

log_level_in — Nivel de log para tráfico entrante.

log_level_out — Nivel de log para tráfico saliente.

log_nf_conntrack: <boolean> (default = 0) — Activa el registro de información de conntrack.

ndp: <boolean> (default = 1) — Activa NDP (Neighbor Discovery Protocol).

nf_conntrack_allow_invalid: <boolean> (default = 0) — Permite paquetes marcados como inválidos por connection tracking.

nf_conntrack_helpers: <string> (default = ``) — Activa helpers conntrack para protocolos concretos. Soportados: amanda, ftp, irc, netbios-ns, pptp, sane, sip, snmp y tftp.

nf_conntrack_max: <integer> (32768 - N) (default = 262144) — Número máximo de conexiones seguidas por conntrack.

nf_conntrack_tcp_timeout_established: <integer> (7875 - N) (default = 432000) — Timeout de conntrack para conexiones TCP establecidas.

nf_conntrack_tcp_timeout_syn_recv: <integer> (30 - 60) (default = 60) — Timeout de conntrack para estado SYN_RECV.

nftables: <boolean> (default = 0) — Activa el firewall basado en nftables (tech preview).

nosmurfs: <boolean> — Activa el filtro SMURFS.

protection_synflood: <boolean> (default = 0) — Activa protección frente a SYN flood.

protection_synflood_burst: <integer> (default = 1000) — Tamaño de ráfaga de protección SYN flood por IP de origen.

protection_synflood_rate: <integer> (default = 200) — Límite de SYN por segundo y por IP de origen.

smurf_log_level — Nivel de log del filtro SMURFS.

tcp_flags_log_level — Nivel de log del filtro de combinaciones ilegales de flags TCP.

tcpflags: <boolean> (default = 0) — Filtra combinaciones ilegales de flags TCP.

13.2.3 Configuración de VM/Container

La configuración de firewall de una VM o contenedor se obtiene de:

```
/etc/pve/firewall/<VMID>.fw
```

Puede contener [OPTIONS], [RULES], [IPSET <name>] y [ALIASES]. Sus principales opciones son:

dhcp: <boolean> (default = 0) — Activa DHCP.

enable: <boolean> (default = 0) — Activa o desactiva las reglas de firewall del invitado.

ipfilter: <boolean> — Activa filtros IP predeterminados. Equivale a añadir un IPSet vacío ipfilter-net<id> por interfaz. Estos conjuntos incluyen restricciones razonables, como limitar direcciones IPv6 link-local a la derivada de la MAC. En contenedores se añaden implícitamente las IP configuradas.

log_level_in — Nivel de log para tráfico entrante.

log_level_out — Nivel de log para tráfico saliente.

macfilter: <boolean> (default = 1) — Activa o desactiva el filtro de direcciones MAC.

ndp: <boolean> (default = 1) — Activa NDP.

policy_in: <ACCEPT | DROP | REJECT> — Política de entrada.

policy_out: <ACCEPT | DROP | REJECT> — Política de salida.

radv: <boolean> — Permite el envío de Router Advertisements.

IMPORTANTE: Cada dispositivo de red virtual tiene su propia casilla de activación del firewall. Debe activarse en la interfaz concreta además de activar el firewall general de la VM/CT.

13.2.4 Configuración de VNet

La configuración relacionada con una VNet se lee desde:

```
/etc/pve/sdn/firewall/<vnet_name>.fw
```

Permite aplicar reglas globalmente a nivel de VNet sin definirlas individualmente para cada VM incluida en ella. Solo puede contener reglas en dirección FORWARD, ya que en una VNet no existe la noción de tráfico IN u OUT. Afecta a todo el tráfico que viaja de un puerto del bridge a otro, incluida la interfaz del host.

ADVERTENCIA: Esta función solo está disponible actualmente con el nuevo proxmox-firewall basado en nftables.

Como el tráfico que atraviesa FORWARD es bidireccional, deben crearse reglas para ambos sentidos si se desea que una conversación pueda circular en las dos direcciones. Por ejemplo, para permitir HTTP hacia un host concreto:

```
FORWARD ACCEPT -dest 10.0.0.1 -dport 80
FORWARD ACCEPT -source 10.0.0.1 -sport 80
```

enable: <boolean> (default = 0) — Activa o desactiva las reglas de la VNet.

log_level_forward — Nivel de log para tráfico reenviado.

policy_forward: <ACCEPT | DROP> — Política de reenvío.

[RULES] — Contiene las reglas específicas de la VNet.

13.3 Reglas de firewall

Una regla de firewall consta de una dirección —IN, OUT o FORWARD— y una acción —ACCEPT, DENY o REJECT—. También puede utilizarse un nombre de macro. Las macros representan conjuntos predefinidos de reglas y opciones. Una regla puede deshabilitarse anteponiendo el carácter |.

```
[RULES]
DIRECTION ACTION [OPTIONS]
|DIRECTION ACTION [OPTIONS] # disabled rule
DIRECTION MACRO(ACTION) [OPTIONS] # use predefined macro
```

Las siguientes opciones permiten refinar la coincidencia de una regla:

--dest <string> — Restringe la dirección de destino. Puede ser una IP, un IPSet (+ipsetname), un alias IP, un rango como 20.34.101.207-201.3.9.99 o una lista separada por comas. No mezcle IPv4 e IPv6 en una misma lista.

--dport <string> — Restringe el puerto TCP/UDP de destino. Admite nombres de servicio o números 0-65535, rangos como 80:85 y listas separadas por comas.

--icmp-type <string> — Especifica el tipo ICMP. Solo válido si proto es icmp, icmpv6 o ipv6-icmp.

--iface <string> — Nombre de interfaz. Para VMs/CT deben usarse las claves net\d+; las reglas del host admiten cadenas arbitrarias.

--log <alert | crit | debug | emerg | err | info | nolog | notice | warning> — Nivel de log de la regla.

--proto <string> — Protocolo IP. Puede utilizar nombres como tcp/udp o números definidos en /etc/protocols.

--source <string> — Restringe la dirección de origen. Admite IP, IPSet, alias, rango o lista; no mezcle IPv4 e IPv6 en una misma lista.

--sport <string> — Restringe el puerto TCP/UDP de origen. Admite nombres, números, rangos y listas.

Ejemplos:

```
[RULES]
IN SSH(ACCEPT) -i net0
IN SSH(ACCEPT) -i net0 # a comment
IN SSH(ACCEPT) -i net0 -source 192.168.2.192 # only allow SSH from 192.168.2.192
IN SSH(ACCEPT) -i net0 -source 10.0.0.1-10.0.0.10 # accept SSH for IP range
IN SSH(ACCEPT) -i net0 -source 10.0.0.1,10.0.0.2,10.0.0.3 # accept ssh for IP list
IN SSH(ACCEPT) -i net0 -source +mynetgroup # accept ssh for ipset mynetgroup
IN SSH(ACCEPT) -i net0 -source myserveralias # accept ssh for alias myserveralias
|IN SSH(ACCEPT) -i net0 # disabled rule
IN DROP # drop all incoming packages
OUT ACCEPT # accept all outgoing packages
```

13.4 Security Groups

Un Security Group es una colección de reglas definida a nivel de clúster y reutilizable en las reglas de cualquier VM. Por ejemplo, puede crear un grupo webserver que abra HTTP y HTTPS:

```
# /etc/pve/firewall/cluster.fw
[group webserver]
IN ACCEPT -p tcp -dport 80
IN ACCEPT -p tcp -dport 443
```

Después puede añadir el grupo al firewall de una VM:

```
# /etc/pve/firewall/<VMID>.fw
[RULES]
GROUP webserver
```

13.5 IP Aliases

Los IP Aliases permiten asociar un nombre a direcciones IP o redes. Ese nombre puede utilizarse dentro de definiciones de IPS et y en las propiedades source y dest de reglas de firewall.

13.5.1 Alias estándar local_network

El alias local_network se define automáticamente. Para consultar el valor detectado, utilice:

```
# pve-firewall localnet
local hostname: example
local IP address: 192.168.2.100
network auto detect: 192.168.0.0/20
using detected local_network: 192.168.0.0/20
```

El firewall crea automáticamente las reglas necesarias para la comunicación del clúster —corosync, API y SSH— utilizando este alias.

El usuario puede sobrescribir el valor en la sección [ALIASES] de cluster.fw. En un host único conectado a una red pública es preferible asignar explícitamente la IP local:

```
# /etc/pve/firewall/cluster.fw
[ALIASES]
local_network 1.2.3.4 # use the single IP address
```

13.6 IP Sets

Los IP Sets permiten definir grupos de redes y hosts. Se referencian como +name en las propiedades source y dest de las reglas. Ejemplo: permitir HTTP desde el IPSet management:

```
IN HTTP(ACCEPT) -source +management
```

13.6.1 IPSet estándar management

Este IPSet solo se aplica a los firewalls de host, no a los de las VMs. Sus direcciones tienen permiso para realizar tareas normales de administración: GUI de Proxmox VE, VNC, SPICE y SSH.

La red local del clúster se añade automáticamente a este conjunto mediante el alias cluster_network para permitir la comunicación entre hosts del clúster.

```
# /etc/pve/firewall/cluster.fw
[IPSET management]
192.168.2.10
192.168.2.10/24
```

13.6.2 IPSet estándar blacklist

El tráfico procedente de estas direcciones se descarta en los firewalls de todos los hosts y VMs.

```
# /etc/pve/firewall/cluster.fw
[IPSET blacklist]
77.240.159.182
213.87.123.0/24
```

13.6.3 IPSet estándar ipfilter-net*

Estos filtros pertenecen a una interfaz de red de una VM y se utilizan principalmente para impedir IP spoofing. Si existe un conjunto de este tipo para una interfaz, cualquier tráfico saliente cuya IP de origen no coincida con el IPSet correspondiente será descartado.

En contenedores que ya tienen direcciones IP configuradas, dichos conjuntos —si existen o si se activan mediante la opción general IP Filter— contienen implícitamente las IP asociadas. Tanto en VMs como en contenedores también incluyen de forma implícita la dirección IPv6 link-local estándar derivada de la MAC para permitir Neighbor Discovery.

```
# /etc/pve/firewall/<VMID>.fw
[IPSET ipfilter-net0] # only allow specified IPs on net0
192.168.2.10
```

13.7 Servicios y comandos

En cada nodo se ejecutan dos demonios relacionados con el firewall:

- **pvefw-logger** — demonio NFLOG, reemplazo de ulogd.
- **pve-firewall** — actualiza las reglas iptables.

También existe el comando pve-firewall para iniciar y detener el servicio:

```
# pve-firewall start
# pve-firewall stop
```

Para consultar el estado:

```
# pve-firewall status
```

Este comando lee y compila todas las reglas, por lo que también mostrará avisos si la configuración contiene errores.

Para ver las reglas iptables generadas:

```
# iptables-save
```

13.8 Reglas predeterminadas del firewall

La configuración predeterminada aplica filtrado adicional incluso antes de considerar las reglas definidas por el usuario.

13.8.1 Datacenter: política IN/OUT DROP o REJECT

Si la política de entrada o salida se establece en DROP o REJECT, el siguiente tráfico continúa estando permitido para todos los hosts Proxmox VE del clúster:

- el tráfico sobre la interfaz loopback;
- conexiones ya establecidas;
- el protocolo IGMP;
- TCP desde hosts de administración al puerto 8006 para la interfaz web;
- TCP desde hosts de administración a 5900-5999 para la consola VNC web;
- TCP desde hosts de administración al puerto 3128 para el proxy SPICE;
- TCP desde hosts de administración al puerto 22 para SSH;
- TCP desde hosts de administración a 60000-60050 para migraciones;
- UDP dentro de la red del clúster a los puertos 5405-5412 para corosync;
- UDP multicast dentro de la red del clúster;
- ICMP tipos 3 (Destination Unreachable), 4 (congestion control) y 11 (Time Exceeded).

El siguiente tráfico se descarta, pero no se registra aunque el logging esté activado:

- conexiones TCP con estado de connection tracking inválido;
- Broadcast, multicast y anycast no relacionado con corosync, es decir, que no llegue por 5405-5412;
- TCP al puerto 43;
- UDP a los puertos 135 y 445;
- UDP a 137-139;
- UDP con puerto de origen 137 hacia 1024-65535;
- UDP al puerto 1900;
- TCP a los puertos 135, 139 y 445;
- UDP cuyo puerto de origen sea 53.

El resto del tráfico se descarta o rechaza según la política, y se registra. El resultado puede variar según opciones adicionales activadas en Firewall → Options, como NDP, SMURFS o el filtrado de flags TCP.

Para inspeccionar las cadenas y reglas realmente activas en el sistema utilice:

```
# iptables-save
```

Esta salida también se incluye en el System Report, accesible desde la pestaña Subscription de un nodo en la GUI o mediante el comando pvereport.

13.8.2 VM/CT: política IN/OUT DROP o REJECT

Una política DROP o REJECT descarta o rechaza todo el tráfico hacia las VMs/CT salvo determinadas excepciones para DHCP, NDP, Router Advertisement y filtros MAC/IP, en función de la configuración aplicada. Las reglas de descarte/rechazo se heredan del Datacenter, pero las excepciones de tráfico aceptado propias del host no se aplican a las VMs/CT.

De nuevo, puede utilizar iptables-save para inspeccionar todas las reglas y cadenas aplicadas.

13.9 Logging de reglas de firewall

De forma predeterminada, el registro del tráfico filtrado por el firewall está desactivado. Para habilitarlo debe establecers e un log level para tráfico entrante y/o saliente en Firewall → Options. Puede configurarse tanto para el host como de manera indep endiente para cada VM o CT.

Al hacerlo se activa el logging de las reglas estándar de Proxmox VE y la salida puede consultarse en Firewall → Log. En las reglas estándar solo se registran determinados paquetes descartados o rechazados, según lo descrito en la sección de reglas predeterminadas.

CONTINUACIÓN: La Parte 13B continuará con 13.9.1 Logging de reglas definidas por el usuario, 13.10 Tips and Tricks, Suricata IPS, notas IPv6, puertos utilizados por Proxmox VE y 13.13 nftables.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 13B

Proxmox VE Firewall

Logging avanzado · FTP · Suricata IPS · IPv6 · Puertos · nftables

Registro	Integraciones	Red	Backend moderno
Reglas estándar + reglas de usuario	FTP + Suricata IPS	IPv6 + puertos Proxmox	proxmox-firewall + nftables

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, rutas, puertos, nombres de servicios y sintaxis técnica.

13.9 Registro de reglas de firewall — continuación

El nivel de registro (loglevel) no determina cuánta cantidad de tráfico filtrado se registra. Lo que modifica es el LOGID que se añade como prefijo a la salida de log, facilitando el filtrado y el posprocesado.

Niveles de log y LOGID

loglevel	LOGID
nolog	—
emerg	0
alert	1
crit	2
err	3
warning	4
notice	5
info	6
debug	7

Formato típico de una entrada de registro:

```
VMID LOGID CHAIN TIMESTAMP POLICY: PACKET_DETAILS
```

En el firewall del host, VMID es 0.

13.9.1 Registro de reglas definidas por el usuario

Para registrar paquetes filtrados por reglas creadas por el usuario, puede definirse un parámetro log-level en cada regla de forma individual. Esto permite un registro granular e independiente del nivel configurado para las reglas estándar en Firewall -> Options.

El nivel puede configurarse desde la interfaz web, mediante las llamadas API correspondientes con pvesh o directamente en el archivo de configuración añadiendo -log <loglevel>.

Estas dos reglas son equivalentes porque no generan log:

```
IN REJECT -p icmp -log nolog
IN REJECT -p icmp
```

En cambio, la siguiente genera una entrada marcada con nivel debug:

```
IN REJECT -p icmp -log debug
```

13.10 Consejos y trucos

13.10.1 Cómo permitir FTP

FTP utiliza el puerto 21 y, además, varios puertos dinámicos. Por ello no basta con permitir únicamente el puerto 21: también debe cargarse el módulo de seguimiento de conexiones FTP.

```
modprobe ip_conntrack_ftp
```

Añada también ip_conntrack_ftp a /etc/modules para que el módulo se cargue automáticamente después de reiniciar.

13.10.2 Integración con Suricata IPS

Es posible integrar Suricata como IPS (Intrusion Prevention System). Los paquetes se envían al IPS únicamente después de que el firewall los haya aceptado. Los paquetes rechazados o descartados por el firewall no llegan a Suricata.

Instale Suricata en el host Proxmox y cargue nfnetlink_queue:

```
apt-get install suricata
modprobe nfnetlink_queue
```

Añada nfnetlink_queue a /etc/modules para conservar la configuración tras un reinicio.

Para habilitar IPS en una VM concreta, añada:

```
# /etc/pve/firewall/<VMID>.fw
[OPTIONS]
ips: 1
ips_queues: 0
```

ips_queues vincula la VM a una cola de CPU concreta. Las colas disponibles se definen en:

```
# /etc/default/suricata
NFQUEUE=0
```

13.11 Notas sobre IPv6

El firewall incluye varias opciones específicas para IPv6. A diferencia de IPv4, IPv6 no utiliza ARP; emplea NDP (Neighbor Di scovery Protocol), que funciona a nivel IP y necesita direcciones IP para operar.

Para ello se utilizan direcciones link-local derivadas de la dirección MAC de la interfaz. Por defecto, la opción NDP está habilitada tanto a nivel de host como de VM, permitiendo enviar y recibir paquetes de descubrimiento de vecinos.

NDP también se utiliza para otras funciones, como autoconfiguración y anuncio de routers.

De forma predeterminada, las VMs pueden enviar mensajes Router Solicitation para localizar routers y recibir Router Advertisement, lo que permite utilizar autoconfiguración sin estado. Sin embargo, no pueden anunciarse ellas mismas como routers salvo que se active la opción Allow Router Advertisement:

```
radv: 1
```

También existe la opción IP Filter:

```
ipfilter: 1
```

Su efecto equivale a crear un IPSet ipfilter-net* para cada interfaz de red de la VM, incluyendo las direcciones link-local correspondientes. Esto se relaciona con los filtros ipfilter-net* explicados en la Parte 13A.

13.12 Puertos utilizados por Proxmox VE

El manual enumera los siguientes puertos y protocolos relevantes:

Servicio	Puerto	Protocolo / uso
Interfaz web	8006/TCP	HTTP/1.1 sobre TLS
Consola web VNC	5900-5999/TCP	WebSocket
Proxy SPICE	3128/TCP	SPICE proxy
sshd	22/TCP	Acciones de clúster y administración
rpcbind	111/UDP	RPC
sendmail	25/TCP	Tráfico saliente
Corosync	5405-5412/UDP	Tráfico de clúster
Migración en vivo	60000-60050/TCP	Memoria de VM y datos de discos locales

13.13 nftables

Como alternativa a pve-firewall, Proxmox VE ofrece proxmox-firewall, una implementación basada en nftables en lugar de iptables.

ADVERTENCIA: proxmox-firewall se encuentra en tech preview en esta versión del manual. Puede contener errores o incompatibilidades respecto al firewall original y no se considera adecuado para uso en producción.

La implementación utiliza los mismos archivos y el mismo formato de configuración, por lo que una configuración existente puede reutilizarse al cambiar de backend.

Según el manual, existen algunas diferencias respecto a pve-firewall:

- Con Linux bridges no se crean bridges de firewall adicionales fwbrX. Las interfaces de invitados que utilicen OVS bridges siguen utilizando bridges de firewall.
- REJECT no está disponible actualmente para tráfico de invitados; en su lugar el tráfico se descarta.
- Las opciones NDP, Router Advertisement y DHCP siempre generan reglas de firewall, independientemente de la política predeterminada.
- Las reglas de invitados se evalúan incluso para conexiones que ya poseen entradas en la tabla conntrack.

13.13.1 Instalación y uso

Instale el paquete:

```
apt install proxmox-firewall
```

Active el backend nftables desde la interfaz web en Host -> Firewall -> Options -> nftables, o en el archivo del host:

```
# /etc/pve/nodes/<node_name>/host.fw
[OPTIONS]
nftables: 1
```

NOTA: Después de activar o desactivar proxmox-firewall, todas las VMs y contenedores en ejecución deben reiniciarse para que el firewall antiguo o el nuevo funcionen correctamente.

Después de establecer nftables: 1, el servicio proxmox-firewall toma el control. Compruebe su estado con:

```
systemctl status proxmox-firewall
```

También conviene verificar que pve-firewall ya no esté generando reglas iptables. Para volver al firewall anterior basta con devolver el valor de nftables a 0 / No.

13.13.2 Uso

proxmox-firewall crea dos tablas administradas por el servicio:

- proxmox-firewall
- proxmox-firewall-guests

Si necesita reglas personalizadas fuera de la configuración de firewall de Proxmox VE, puede crear sus propias tablas. proxmox-firewall solo modifica las tablas que genera, lo que permite extender su comportamiento sin alterar las tablas gestionadas.

En lugar del comando pve-firewall, el backend nftables utiliza el servicio systemd proxmox-firewall:

```
systemctl start proxmox-firewall
systemctl stop proxmox-firewall
systemctl status proxmox-firewall
```

Al detener el servicio se eliminan todas las reglas generadas.

13.13.3 Comandos útiles

Para consultar el ruleset generado:

```
nft list ruleset
```

Para depurar proxmox-firewall puede volcar los comandos generados mediante el subcomando compile. La variable de entorno PVE_LOG permite escribir mensajes de depuración en STDERR:

```
PVE_LOG=trace /usr/libexec/proxmox/proxmox-firewall compile > firewall.json
```

El ruleset de nftables está formado por un esqueleto incluido en el binario de proxmox-firewall y por las reglas generadas a partir de la configuración. Puede obtenerse el ruleset base con:

```
/usr/libexec/proxmox/proxmox-firewall skeleton
```

Ambas salidas pueden enviarse directamente a nft. Para recrear el ruleset completo desde cero:

```
/usr/libexec/proxmox/proxmox-firewall skeleton | nft -f -
/usr/libexec/proxmox/proxmox-firewall compile | nft -j -f -
```

Registro detallado del servicio

Para aumentar el detalle del registro mientras el daemon está en ejecución:

```
systemctl edit proxmox-firewall
```

Añada el siguiente override:

```
[Service]
Environment="PVE_LOG=trace"
```

ADVERTENCIA: El nivel trace puede generar una gran cantidad de registros en muy poco tiempo. Utilícelo únicamente para depuración. También existen los niveles menos verbosos info y debug.

Trazado de paquetes con nftrace

Para depurar reglas puede ser útil seguir el recorrido de paquetes a través de las distintas cadenas. Esto se consigue activando nftrace únicamente para el tráfico que se desea analizar. El ejemplo del manual limita el trazado a ICMP:

```
#!/usr/sbin/nft -f
table bridge tracebridge
delete table bridge tracebridge
table bridge tracebridge {
    chain trace {
        meta l4proto icmp meta nftrace set 1
    }
    chain prerouting {
        type filter hook prerouting priority -350; policy accept;
        jump trace
    }
    chain postrouting {
        type filter hook postrouting priority -350; policy accept;
        jump trace
    }
}
```

Guarde el archivo, hágalo ejecutable y ejecútelo una vez para crear las cadenas de trazado. Después puede consultar el resultado en la interfaz web de Proxmox VE, Firewall -> Log, o mediante:

```
nft monitor trace
```

El ejemplo anterior traza tráfico sobre todos los bridges, que normalmente es donde circula el tráfico de invitados. Para examinar tráfico del host, cree las cadenas en la tabla inet en lugar de bridge.

NOTA: El trazado puede generar una gran cantidad de logs y degradar de forma significativa el rendimiento de la pila de red.

Para eliminar las reglas de tracing:

```
nft delete table bridge tracebridge
```

Fin del Capítulo 13 · Proxmox VE Firewall

Con esta Parte 13B queda completada la traducción del Capítulo 13 del manual. La siguiente entrega comienza el Capítulo 14 · Gestión de usuarios.

CRITERIO DE TRADUCCIÓN: Se han mantenido literalmente rutas, comandos, nombres de servicios, puertos, opciones y sintaxis que deben utilizarse tal como aparecen en Proxmox VE. Los textos descriptivos se han traducido al castellano preservando la organización y el alcance del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 14A

Gestión de usuarios

Usuarios · grupos · API Tokens · Resource Pools · PAM · LDAP · Active Directory · OpenID Connect · 2FA

Identidad	Automatización	Directorios	Segundo factor
Usuarios + grupos	API Tokens	LDAP + AD + OIDC	TOTP + YubiKey + WebAuthn

Basado en Proxmox VE Administration Guide, Release 9.2.4

Traducción progresiva al castellano. Se conservan comandos, rutas, nombres de opciones y sintaxis técnica.

Capítulo 14 · Gestión de usuarios

Proxmox VE admite múltiples fuentes de autenticación, entre ellas Linux PAM, el servidor de autenticación integrado de Proxmox VE, LDAP, Microsoft Active Directory y OpenID Connect.

Mediante la gestión de usuarios y permisos basada en roles para todos los objetos - máquinas virtuales, almacenamiento, nodos y otros recursos - es posible definir un control de acceso granular.

14.1 Usuarios

Proxmox VE guarda los atributos de usuario en /etc/pve/user.cfg. Las contraseñas no se almacenan en ese archivo; cada usuario se asocia a uno de los realms de autenticación descritos más adelante. Por este motivo, internamente un usuario suele identificarse mediante su nombre y realm con el formato <userid>@<realm>.

Cada entrada de usuario puede contener la siguiente información:

- Nombre.
- Apellidos.
- Dirección de correo electrónico.
- Pertenencia a grupos.
- Una fecha de expiración opcional.
- Un comentario o nota sobre el usuario.
- Si el usuario está habilitado o deshabilitado.
- Claves opcionales de autenticación de dos factores.

PRECAUCIÓN: Si deshabilita o elimina un usuario, o si su fecha de expiración ya ha pasado, ese usuario no podrá iniciar nuevas sesiones ni lanzar nuevas tareas. Las tareas que ya hubiera iniciado - por ejemplo, sesiones de terminal - no se finalizarán automáticamente.

14.1.1 Administrador del sistema

El usuario root del sistema siempre puede iniciar sesión mediante el realm Linux PAM y actúa como administrador sin restricciones. No puede eliminarse, aunque sí pueden modificarse sus atributos. Los correos del sistema se enviarán a la dirección de correo asociada a este usuario.

14.2 Grupos

Cada usuario puede pertenecer a varios grupos. Los grupos son la forma recomendada de organizar los permisos de acceso. Siempre que sea posible, conviene conceder permisos a grupos en lugar de a usuarios individuales, ya que así se obtiene una lista de control de acceso mucho más fácil de mantener.

14.3 API Tokens

Los API Tokens permiten acceso sin estado a la mayor parte de la API REST desde otro sistema, software o cliente API. Existen excepciones: los endpoints que proporcionan acceso a la consola de una máquina virtual o del sistema requieren privilegios de usuario y no pueden utilizarse mediante un API Token.

Los tokens pueden generarse para usuarios individuales y recibir permisos y fechas de expiración independientes, limitando así tanto el alcance como la duración del acceso. Si un token queda comprometido, puede revocarse sin necesidad de deshabilitar al usuario.

Existen dos tipos básicos de API Tokens:

- Privilegios separados: el token debe recibir permisos explícitos mediante ACL. Sus permisos efectivos se calculan como la intersección entre los permisos del usuario y los del token. Es el modo predeterminado para tokens nuevos.
- Privilegios completos: el token hereda exactamente los permisos del usuario asociado. Solo debe desactivarse Privilege Separation cuando la integración necesite realmente todos los permisos del usuario.

Al crear un token puede definirse una fecha de expiración en forma de timestamp Unix, evitando que automatizaciones abandonadas sigan funcionando indefinidamente. Una vez superado ese momento, el código de autenticación rechazará las peticiones.

PRECAUCIÓN: El valor secreto del token solo se muestra una vez, en el momento de generarlo. No puede recuperarse posteriormente mediante la API.

Para usar un API Token, establezca la cabecera HTTP Authorization con el valor mostrado en el formato:

```
PVEAPIToken=USER@REALM!TOKENID=UUID
```

También puede seguir la documentación del cliente API que utilice.

14.4 Resource Pools

Un Resource Pool es un conjunto de máquinas virtuales, contenedores y dispositivos de almacenamiento. Resulta útil cuando determinados usuarios deben tener acceso controlado a un conjunto concreto de recursos, ya que permite aplicar un único permiso a varios elementos en lugar de administrarlo recurso por recurso.

Los Resource Pools se utilizan a menudo junto con grupos, de forma que los miembros de un grupo puedan disponer de permisos sobre un conjunto específico de máquinas y almacenamiento.

14.5 Realms de autenticación

Los usuarios de Proxmox VE actúan como contrapartes de usuarios existentes en algún realm externo. Los realms se configuran en /etc/pve/domains.cfg. Están disponibles los siguientes métodos de autenticación:

Linux PAM Standard Authentication - Framework de autenticación de usuarios del sistema. Los usuarios se crean en el host mediante herramientas como adduser. Si existen como usuarios PAM en el host Proxmox VE, pueden añadirse las entradas correspondientes en Proxmox VE para permitir el acceso con su usuario y contraseña del sistema.

Proxmox VE Authentication Server - Almacén de contraseñas de estilo Unix. Guarda hashes en /etc/pve/priv/shadow.cfg utilizando SHA-256. Es especialmente cómodo en instalaciones pequeñas o medianas donde los usuarios no necesitan acceso fuera de Proxmox VE. Los usuarios se gestionan íntegramente desde Proxmox VE y pueden cambiar su propia contraseña desde la GUI.

LDAP - Lightweight Directory Access Protocol es un protocolo abierto y multiplataforma para autenticación mediante servicios de directorio. OpenLDAP es una implementación open source popular.

Microsoft Active Directory (AD) - Servicio de directorio para redes de dominio Windows, admitido como realm de autenticación en Proxmox VE. Utiliza LDAP como uno de sus protocolos de autenticación.

OpenID Connect - Capa de identidad construida sobre OAuth 2.0. Permite a los clientes verificar la identidad del usuario basándose en una autenticación realizada por un servidor de autorización externo.

14.5.1 Autenticación estándar Linux PAM

Como Linux PAM corresponde a usuarios del sistema host, debe existir un usuario del sistema en cada nodo en el que se le permita iniciar sesión. El usuario se autentica con su contraseña habitual del sistema. Este realm se añade de forma predeterminada y no puede eliminarse.

Los cambios de contraseña realizados desde la GUI o mediante el endpoint /access/password solo afectan al nodo local, no a todo el clúster. Aunque Proxmox VE utiliza un diseño multi-master, mantener contraseñas distintas entre nodos puede aportar una ventaja de seguridad.

Un administrador puede exigir autenticación de dos factores para los inicios de sesión de este realm y también establecerlo como realm de autenticación predeterminado.

14.5.2 Servidor de autenticación de Proxmox VE

El realm de autenticación propio de Proxmox VE es un almacén sencillo de contraseñas de estilo Unix. Se crea de forma predeterminada y, como ocurre con Linux PAM, sus principales opciones son exigir autenticación de dos factores y marcarlo como realm predeterminado.

A diferencia de otros realms, los usuarios se crean y autentican completamente dentro de Proxmox VE. Por ello es obligatorio establecer una contraseña al crear este tipo de usuario.

14.5.3 LDAP

También puede utilizarse un servidor LDAP externo, por ejemplo OpenLDAP. Los usuarios se buscan bajo un Base Domain Name (base_dn), utilizando como identificador el atributo especificado en User Attribute Name (user_attr).

Puede configurarse un servidor principal y, opcionalmente, uno de respaldo. La conexión puede cifrarse mediante SSL. También pueden aplicarse filtros a directorios y grupos para limitar el alcance del realm.

Ejemplo de entrada LDIF para un usuario:

```
# user1 of People at ldap-test.com
dn: uid=user1,ou=People,dc=ldap-test,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: user1
cn: Test User 1
sn: Testers
description: This is the first test user.
```

En este caso, el Base Domain Name sería:

```
ou=People,dc=ldap-test,dc=com
```

y el atributo de usuario sería:

```
uid
```

Si Proxmox VE necesita autenticarse - realizar bind - frente al servidor LDAP antes de poder consultar y autenticar usuarios, puede configurarse un Bind DN mediante bind_dn en /etc/pve/domains.cfg.

La contraseña correspondiente debe almacenarse en:

```
/etc/pve/priv/realm/<realmname>.pw
```

Por ejemplo:

```
/etc/pve/priv/realm/my-ldap.pw
```

El archivo debe contener una única línea con la contraseña sin procesar.

Para verificar certificados debe configurarse capath. Puede apuntar directamente al certificado de la CA del servidor LDAP o al directorio del sistema que contiene las CA de confianza:

```
/etc/ssl/certs
```

Además debe activarse la opción verify, lo que también puede realizarse desde la interfaz web.

Opciones principales de un realm LDAP:

- Realm (realm): identificador del realm para los usuarios de Proxmox VE.
- Base Domain Name (base_dn): directorio bajo el que se buscan usuarios.
- User Attribute Name (user_attr): atributo LDAP que contiene el nombre con el que el usuario inicia sesión.
- Server (server1): servidor que aloja el directorio LDAP.
- Fallback Server (server2): servidor alternativo opcional si el principal no está disponible.
- Port (port): puerto en el que escucha el servidor LDAP.

NOTA: Para que un usuario concreto pueda autenticarse mediante LDAP, también debe existir como usuario de ese realm en Proxmox VE. Esta creación puede automatizarse mediante la sincronización.

14.5.4 Microsoft Active Directory (AD)

Para configurar Microsoft AD como realm deben especificarse una dirección de servidor y un dominio de autenticación. Active Directory admite la mayoría de las propiedades disponibles en LDAP, como servidor alternativo, puerto y cifrado SSL.

Tras configurar el realm, los usuarios pueden añadirse automáticamente a Proxmox VE mediante operaciones de sincronización.

Como en LDAP, si Proxmox VE necesita autenticarse antes de hacer bind con el servidor AD, debe configurarse la propiedad Bind User (bind_dn). Esta propiedad suele ser necesaria por defecto con Microsoft AD.

Parámetros principales:

- Realm (realm): identificador del realm para los usuarios de Proxmox VE.
- Domain (domain): dominio de Active Directory.
- Server (server1): FQDN o dirección IP del servidor.
- Fallback Server (server2): servidor alternativo opcional.
- Port (port): puerto en el que escucha Active Directory.

NOTA: Microsoft AD suele tratar los nombres de usuario sin distinguir mayúsculas y minúsculas. Para que Proxmox VE se comporte igual, puede desactivar la opción case-sensitive desde la GUI o con: pveum realm modify ID --case-sensitive 0

14.5.5 Sincronización de realms basados en LDAP

Es posible sincronizar automáticamente usuarios y grupos de realms LDAP y Microsoft Active Directory, evitando tener que añadirlos manualmente a Proxmox VE.

Las opciones de sincronización están disponibles en la ventana Add/Edit del panel Authentication o mediante pveum realm add/modify. La sincronización puede ejecutarse desde la GUI o con:

```
pveum realm sync <realm>
```

Los usuarios y grupos sincronizados se guardan en el archivo de configuración de ámbito de clúster:

```
/etc/pve/user.cfg
```

Atributos a propiedades

Si la respuesta de sincronización incluye atributos del usuario, estos se copian a la propiedad correspondiente de user.cfg, por ejemplo firstname o lastname.

Si los nombres de los atributos no coinciden con las propiedades de Proxmox VE, puede definirse un mapeo personalizado mediante sync_attributes.

El tratamiento de propiedades que desaparecen puede controlarse mediante las opciones de sincronización descritas a continuación.

Configuración de sincronización

Las opciones se encuentran en la pestaña Sync Options de la ventana Add/Edit.

- Bind User (bind_dn): cuenta LDAP usada para consultar usuarios y grupos. Debe tener acceso a todas las entradas deseadas. Si se configura, las búsquedas se realizan autenticándose; de lo contrario se hacen de forma anónima. Debe indicarse como DN LDAP completo, por ejemplo cn=admin,dc=example,dc=com.
- Groupname attr. (group_name_attr): atributo que representa los grupos del usuario. Solo se sincronizan entradas que cumplan las restricciones habituales de user.cfg. Los grupos se guardan con -$realm añadido al nombre para evitar colisiones.
- User classes (user_classes): objectClasses asociados a usuarios.
- Group classes (group_classes): objectClasses asociados a grupos.
- E-Mail attribute: atributo del servidor LDAP que contiene la dirección de correo; desde CLI puede configurarse mediante --sync_attributes.
- User Filter (filter): filtro adicional para limitar los usuarios objetivo.
- Group Filter (group_filter): filtro adicional para limitar los grupos objetivo.

NOTA: Los filtros permiten añadir criterios de coincidencia para reducir el alcance de la sincronización. El manual remite a ldap.com para la sintaxis y tipos de filtros LDAP disponibles.

Opciones de la operación de sincronización

Además de la configuración anterior, pueden establecerse opciones que describen el comportamiento de cada sincronización. Se pasan como parámetros al ejecutar el sync o se guardan como valores predeterminados mediante sync-defaults-options.

- Scope (scope): qué sincronizar: users, groups o ambos.
- Enable new (enable-new): si se activa, los usuarios recién sincronizados quedan habilitados y pueden iniciar sesión. El valor predeterminado es true.
- Remove Vanished (remove-vanished): controla qué se elimina cuando ya no aparece en la respuesta de sincronización.
- Preview (dry-run): no escribe datos en la configuración y permite comprobar qué usuarios y grupos se sincronizarían.

Las opciones de Remove Vanished son:

- ACL (acl): elimina ACL de usuarios y grupos que ya no aparecen en la respuesta. Normalmente se usa junto con Entry.
- Entry (entry): elimina las entradas de usuario o grupo ausentes de la respuesta.
- Properties (properties): elimina propiedades de entradas cuando el usuario sincronizado ya no contiene esos atributos. Se aplica incluso a propiedades que nunca fueron creadas por una sincronización; los tokens y el flag enable se conservan.

Caracteres reservados

Algunos caracteres están reservados por LDAP/RFC2253 y deben escaparse correctamente cuando forman parte de valores usados dentro de DN.

Los siguientes caracteres necesitan tratamiento especial:

- Espacio al principio o al final.
- Signo # al principio.
- Coma (,).
- Signo +.
- Comillas dobles (").
- Barra inclinada (/).
- Signos menor y mayor (< >).
- Punto y coma (;).
- Signo igual (=).

Para utilizarlos en DN, el valor del atributo puede rodearse con comillas dobles. Por ejemplo, para un CN llamado Example, User:

```
CN="Example, User",OU=people,DC=example,DC=com
```

Esto se aplica a base_dn, bind_dn y group_dn.

NOTA: Los usuarios cuyos nombres contengan dos puntos (:) o barras inclinadas (/) no pueden sincronizarse porque esos caracteres están reservados en los nombres de usuario.

14.5.6 OpenID Connect

Opciones principales de configuración OpenID Connect:

- Issuer URL (issuer-url): URL del servidor de autorización. Proxmox VE utiliza OpenID Connect Discovery para obtener automáticamente el resto de detalles. Aunque puede usarse http://, el manual recomienda conexiones cifradas https://.
- Realm (realm): identificador del realm para usuarios de Proxmox VE.
- Client ID (client-id): identificador del cliente OpenID.
- Client Key (client-key): clave opcional del cliente OpenID.
- Autocreate Users (autocreate): crea automáticamente usuarios que aún no existan. Aunque la autenticación se realiza en el servidor OpenID, todos los usuarios necesitan una entrada en la configuración de Proxmox VE.
- Username Claim (username-claim): claim OpenID usado para generar el nombre de usuario: subject, username o email.
- Autocreate Groups (groups-autocreate): crea los grupos presentes en el claim en lugar de utilizar únicamente grupos ya existentes en PVE.
- Groups Claim (groups-claim): claim OpenID del que se obtiene la lista de grupos desde el ID token o el endpoint userinfo.
- Overwrite Groups (groups-overwrite): reemplaza todos los grupos asignados al usuario en lugar de añadir los recibidos a los ya existentes.

Mapeo de nombres de usuario

La especificación OpenID Connect define un atributo único llamado subject. Por defecto, Proxmox VE utiliza su valor y le añade @<realm> para generar el nombre del usuario:

```
${subject}@${realm}
```

Muchos servidores OpenID usan cadenas aleatorias como subject, por ejemplo DGH76OKH34BNG3245SB, lo que produce nombres difíciles de reconocer.

La opción username-claim permite utilizar otros atributos. Si el servidor garantiza unicidad, username suele ser más legible. También puede usarse email cuando el proveedor garantiza que ese valor es único.

Mapeo de grupos

La opción groups-claim habilita el mapeo de grupos. El claim debe contener una lista de cadenas correspondientes a grupos de Proxmox VE.

Para evitar colisiones, los nombres procedentes de OpenID se guardan con el sufijo -<realm>. Por ejemplo, el grupo my-openid-group del realm oidc se convierte en my-openid-group-oidc.

Por defecto, los grupos que no existen en Proxmox VE se ignoran. Con groups-autocreate pueden crearse automáticamente durante el inicio de sesión.

También por defecto, los grupos recibidos se añaden a los grupos que ya tiene el usuario. Si se habilita groups-overwrite, los grupos existentes se eliminan antes de aplicar los recibidos desde OpenID.

Si el proveedor envía nombres de grupos con caracteres no válidos para los IDs de grupo de Proxmox VE, dichos grupos se omiten y se registra una advertencia.

Opciones avanzadas

Query userinfo endpoint (query-userinfo) - Cuando se activa, el autenticador OpenID Connect consulta el endpoint "userinfo" para obtener claims. Desactivarlo puede ser útil con proveedores que no implementan ese endpoint, por ejemplo ADFS.

Ejemplos

Ejemplo de creación de un realm OpenID utilizando Google. Deben sustituirse client-id y client-key por los valores de la configuración OpenID de Google:

```
pveum realm add myrealm1 --type openid \
--issuer-url https://accounts.google.com \
--client-id XXXX --client-key YYYY \
--username-claim email
```

Con --username-claim email, el nombre de usuario en Proxmox VE tendrá un formato similar a:

```
example.user@google.com@myrealm1
```

Keycloak es una plataforma open source de Identity and Access Management compatible con OpenID Connect. Ejemplo:

```
pveum realm add myrealm2 --type openid \
--issuer-url https://your.server:8080/realms/your-realm \
--client-id XXX --username-claim username
```

Con --username-claim username, el usuario en Proxmox VE puede verse como:

```
example.user@myrealm2
```

ADVERTENCIA: Debe asegurarse de que el propio usuario no pueda modificar en el servidor Keycloak el atributo utilizado como username, ya que Proxmox VE confía en él para la identificación.

14.6 Autenticación de dos factores

Existen dos formas de utilizar autenticación de dos factores en Proxmox VE.

La primera consiste en exigirla desde el realm de autenticación, mediante TOTP (Time-based One-Time Password) o YubiKey OTP. En este caso, un usuario recién creado necesita tener su segundo factor configurado inmediatamente, ya que no podrá iniciar sesión sin él. Con TOTP, el usuario puede cambiar posteriormente su configuración siempre que consiga iniciar sesión.

La segunda posibilidad es que los propios usuarios activen voluntariamente 2FA más adelante, incluso aunque el realm no lo exija.

14.6.1 Factores secundarios disponibles

Puede configurarse más de un segundo factor para evitar que la pérdida del teléfono o de una llave de seguridad bloquee permanentemente la cuenta.

Además de TOTP y YubiKey OTP exigidos por realm, están disponibles:

- TOTP configurado por el usuario: código corto derivado de un secreto compartido y de la hora actual; cambia cada 30 segundos.
- WebAuthn: estándar general de autenticación implementado por dispositivos como llaves hardware o TPM de un ordenador o teléfono.
- Recovery Keys de un solo uso: lista de claves que conviene imprimir y guardar en un lugar seguro o almacenar en una bóveda electrónica. Cada clave solo puede utilizarse una vez.

Antes de que Proxmox VE admitiera WebAuthn, los usuarios podían configurar U2F. Los factores U2F existentes siguen funcionando, pero el manual recomienda migrar a WebAuthn una vez configurado en el servidor.

14.6.2 2FA exigido por el realm

Se configura seleccionando uno de los métodos disponibles en el desplegable TFA al añadir o editar un realm. Cuando TFA está habilitado en un realm, se convierte en requisito y solo los usuarios con un segundo factor configurado podrán iniciar sesión .

Time-based OATH (TOTP)

Utiliza el algoritmo estándar HMAC-SHA1, combinando la hora actual con la clave configurada del usuario. El intervalo temporal y la longitud del código son configurables.

Un usuario puede tener varias claves configuradas, separadas por espacios. Las claves pueden expresarse en Base32 (RFC3548) o en hexadecimal.

Proxmox VE proporciona la herramienta oathkeygen, que genera una clave aleatoria en Base32 compatible con herramientas OTP como oathtool y con aplicaciones como Google Authenticator, FreeOTP, andOTP u otras equivalentes.

YubiKey OTP

Para autenticación mediante YubiKey deben configurarse un Yubico API ID, una API KEY y la URL del servidor de validación, y el usuario debe disponer de una YubiKey.

Para obtener el Key ID de una YubiKey, puede activarse una vez tras conectarla por USB y copiar los primeros 12 caracteres de la contraseña escrita en el campo Key IDs del usuario.

El manual remite a la documentación de YubiKey OTP para utilizar YubiCloud o desplegar un servidor de verificación propio.

14.6.3 Límites y bloqueo de autenticación de dos factores

El segundo factor está pensado para proteger la cuenta cuando la contraseña se ha filtrado o ha sido adivinada. Aun así, algu nos factores pueden ser vulnerables a fuerza bruta, por lo que Proxmox VE bloquea al usuario después de demasiados intentos fallidos.

En TOTP, 8 intentos fallidos deshabilitan los factores TOTP del usuario. Se desbloquean al iniciar sesión con una Recovery Key. Si TOTP era el único factor disponible, será necesaria la intervención de un administrador y el manual recomienda exigir al usuario un cambio inmediato de contraseña.

FIDO2/WebAuthn y las Recovery Keys son menos vulnerables a fuerza bruta, por lo que el límite es mayor: 100 intentos. Al superarlo, todos los segundos factores quedan bloqueados durante una hora.

Un administrador puede desbloquear en cualquier momento la autenticación de dos factores de un usuario desde la interfaz o mediante:

```
pveum user tfa unlock joe@pve
```

14.6.4 TOTP configurado por el usuario

Los usuarios pueden optar por activar TOTP o WebAuthn como segundo factor mediante el botón TFA de la lista de usuarios, salvo que el realm imponga YubiKey OTP.

Los usuarios siempre pueden añadir y utilizar Recovery Keys de un solo uso.

Al abrir la ventana TFA, aparece un diálogo para configurar TOTP. El campo Secret contiene la clave y puede generarse aleatoriamente mediante Randomize. También puede añadirse un Issuer Name opcional para que la aplicación TOTP muestre de forma más clara a qué servicio pertenece la clave. El nombre de usuario se incluye igualmente en el código QR.

Tras generar la clave se muestra un código QR compatible con la mayoría de aplicaciones OTP, como FreeOTP.

Para completar la configuración, el usuario debe verificar su contraseña actual - salvo que haya iniciado sesión como root - y demostrar que puede utilizar correctamente la clave TOTP introduciendo el valor OTP actual en Verification Code y pulsando Apply.

Fin de la Parte 14A

Esta entrega cubre desde el inicio del Capítulo 14 hasta 14.6.4. La siguiente parte continuará con TOTP, WebAuthn, Recovery Keys, configuración de U2F/WebAuthn en el servidor, gestión de permisos, roles, privilegios, objetos, pools, herramienta de línea de comandos y ejemplos reales.

CRITERIO DE TRADUCCIÓN: Se han preservado la organización, nombres de opciones, comandos, rutas y ejemplos del manual original. Los textos explicativos se han traducido al castellano sin sustituir ni ampliar silenciosamente el contenido de la fuente.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 14B

Gestión de usuarios

WebAuthn · Recovery Keys · U2F · Roles · Privilegios · ACL · pveum · Ejemplos reales

Segundo factor	Modelo RBAC	Árbol ACL	Administración
TOTP · WebAuthn · U2F	Roles + privilegios	Objetos + paths + herencia	pveum + ejemplos

Basado en Proxmox VE Administration Guide, Release 9.2.4

Continuación del Capítulo 14. Se conservan comandos, rutas, nombres de roles y sintaxis técnica.

14.6.5 TOTP

No se necesita ninguna configuración en el servidor. Basta con instalar una aplicación TOTP en el smartphone - por ejemplo, FreeOTP - y utilizar la interfaz web de Proxmox VE para añadir un factor TOTP.

14.6.6 WebAuthn

Para que WebAuthn funcione se necesitan dos elementos:

- Un certificado HTTPS de confianza, por ejemplo mediante Let's Encrypt. Aunque puede funcionar con un certificado no confiable, algunos navegadores pueden mostrar advertencias o rechazar las operaciones WebAuthn.
- Configurar WebAuthn en Datacenter -> Options -> WebAuthn Settings dentro de la interfaz web de Proxmox VE. En la mayoría de instalaciones los campos pueden autocompletarse.

Una vez cumplidos ambos requisitos, puede añadirse una configuración WebAuthn desde el panel Two Factor en Datacenter -> Permissions -> Two Factor.

14.6.7 Recovery Keys

Las Recovery Keys no requieren preparación previa. Puede generarse directamente un conjunto de claves de recuperación en Datacenter -> Permissions -> Two Factor.

NOTA: Solo puede existir un conjunto de Recovery Keys de un solo uso por usuario en cada momento.

14.6.8 Configuración WebAuthn en el servidor

Para permitir WebAuthn es necesario utilizar un dominio válido con un certificado SSL válido. De lo contrario algunos navegad ores pueden advertir al usuario o rechazar por completo la autenticación.

NOTA: Cambiar la configuración WebAuthn puede hacer que todos los registros WebAuthn existentes dejen de ser utilizables.

La configuración se guarda en /etc/pve/datacenter.cfg. Por ejemplo:

```
webauthn: rp=mypve.example.com,origin=https://mypve.example.com:8006,id=mypve.example.com
```

14.6.9 Configuración U2F en el servidor

NOTA: El manual recomienda utilizar WebAuthn en lugar de U2F.

Para permitir U2F puede ser necesario utilizar un dominio válido con un certificado SSL válido. En caso contrario algunos navegadores pueden mostrar una advertencia o rechazar U2F. Primero debe configurarse un AppId.

NOTA: Cambiar el AppId hará inutilizables todos los registros U2F existentes.

La configuración también se realiza en /etc/pve/datacenter.cfg:

```
u2f: appid=https://mypve.example.com:8006
```

En un único nodo, el AppId puede ser simplemente la dirección de la interfaz web tal y como se usa en el navegador, incluyend o https:// y el puerto. Algunos navegadores son más estrictos que otros al comparar AppIds.

En instalaciones con varios nodos, el manual recomienda disponer de un servidor HTTPS independiente que entregue un archivo appid.json. Si todos los nodos utilizan subdominios del mismo dominio de nivel superior, puede ser suficiente utilizar ese do minio como AppId, aunque algunos navegadores pueden no aceptarlo.

NOTA: Un AppId incorrecto suele producir un error, pero el manual indica que se han observado casos en que no ocurre. Por ello recomienda probar la configuración con varios navegadores antes de desplegarla, ya que cambiar el AppId posteriormente invalida los registros U2F existentes.

14.6.10 Activar U2F como usuario

Para habilitar U2F, abra la pestaña U2F de la ventana TFA, introduzca la contraseña actual - salvo que haya iniciado sesión como root - y pulse Register.

Si el servidor está correctamente configurado y el navegador acepta el AppId proporcionado, aparecerá un mensaje solicitando que se pulse el botón del dispositivo U2F. En una YubiKey, la luz del botón debería parpadear de forma regular.

Los usuarios de Firefox pueden necesitar habilitar security.webauth.u2f desde about:config antes de utilizar un token U2F.

14.7 Gestión de permisos

Para que un usuario pueda realizar una acción - por ejemplo listar, modificar o eliminar parte de la configuración de una VM - necesita disponer de los permisos adecuados.

Proxmox VE utiliza un sistema de permisos basado en roles y rutas. Una entrada de la tabla de permisos permite que un usuario , grupo o token adopte un rol concreto al acceder a un objeto o una ruta.

Una regla de acceso puede representarse como una de estas triples:

```
(path, user, role)
(path, group, role)
(path, token, role)
```

El rol contiene el conjunto de acciones permitidas y la ruta identifica el objetivo sobre el que se aplican.

14.7.1 Roles

Un rol es una lista de privilegios. Proxmox VE incluye varios roles predefinidos que cubren la mayoría de necesidades:

Administrator - Privilegios completos.

NoAccess - Sin privilegios; se utiliza para prohibir acceso.

PVEAdmin - Puede realizar la mayoría de tareas, pero no modificar ciertos ajustes del sistema ni permisos.

PVEAuditor - Acceso de solo lectura.

PVEDatastoreAdmin - Crear y asignar espacio de backup y plantillas.

PVEDatastoreUser - Asignar espacio de backup y consultar almacenamiento.

PVEMappingAdmin - Administrar Resource Mappings.

PVEMappingUser - Consultar y utilizar Resource Mappings.

PVEPoolAdmin - Asignar pools.

PVEPoolUser - Consultar pools.

PVESDNAdmin - Administrar configuración SDN.

PVESDNUser - Acceder a bridges y VNets.

PVESysAdmin - Auditoría, consola del sistema y logs del sistema.

PVETemplateUser - Ver y clonar plantillas.

PVEUserAdmin - Administrar usuarios.

PVEVMAdmin - Administración completa de VMs.

PVEVMUser - Ver, realizar backups, configurar CD-ROM, usar consola y controlar encendido/apagado de VMs.

La lista completa de roles predefinidos puede consultarse en la GUI. También pueden añadirse roles personalizados desde Datacenter -> Permissions -> Roles o mediante pveum.

```
pveum role add VM_Power-only --privs "VM.PowerMgmt VM.Console"
pveum role add Sys_Power-only --privs "Sys.PowerMgmt Sys.Console"
```

NOTA: Los roles que comienzan por PVE están reservados a roles integrados. Los roles personalizados no pueden utilizar ese prefijo.

14.7.2 Privilegios

Un privilegio representa el derecho a realizar una acción específica. Para simplificar la administración, los privilegios se agrupan en roles. No pueden asignarse directamente a usuarios y rutas sin formar parte de un rol.

Privilegios relacionados con nodos y sistema

- Group.Allocate: crear, modificar o eliminar grupos.
- Mapping.Audit: consultar Resource Mappings.
- Mapping.Modify: administrar Resource Mappings.
- Mapping.Use: utilizar Resource Mappings.
- Permissions.Modify: modificar permisos de acceso.
- Pool.Allocate: crear, modificar o eliminar un pool.
- Pool.Audit: consultar un pool.
- Realm.AllocateUser: asignar un usuario a un realm.
- Realm.Allocate: crear, modificar o eliminar realms de autenticación.
- SDN.Allocate: administrar configuración SDN.
- SDN.Audit: consultar configuración SDN.

- Sys.Audit: consultar estado/configuración del nodo, configuración de Corosync y configuración HA.
- Sys.Console: acceso a la consola del nodo.
- Sys.Incoming: permitir flujos de datos entrantes desde otros clústeres (experimental).
- Sys.Modify: crear, modificar o eliminar parámetros de red del nodo.
- Sys.PowerMgmt: administración de energía del nodo (start, stop, reset, shutdown, etc.).
- Sys.Syslog: consultar syslog.
- User.Modify: crear, modificar o eliminar accesos y detalles de usuarios.

Privilegios relacionados con máquinas virtuales

- SDN.Use: acceso a VNets SDN y bridges locales.
- VM.Allocate: crear o eliminar una VM en un servidor.
- VM.Audit: consultar la configuración de una VM.
- VM.Backup: realizar backup y restauración de VMs.
- VM.Clone: clonar o copiar una VM.
- VM.Config.CDROM: expulsar o cambiar el CD-ROM.
- VM.Config.CPU: modificar ajustes de CPU.
- VM.Config.Cloudinit: modificar parámetros Cloud-Init.
- VM.Config.Disk: añadir, modificar o eliminar discos.
- VM.Config.HWType: modificar tipos de hardware emulado.
- VM.Config.Memory: modificar memoria.
- VM.Config.Network: añadir, modificar o eliminar dispositivos de red.
- VM.Config.Options: modificar otras opciones de configuración de la VM.
- VM.Console: acceso a la consola de la VM.
- VM.GuestAgent.Audit: ejecutar comandos informativos del QEMU Guest Agent.
- VM.GuestAgent.FileRead: leer archivos del invitado mediante QEMU Guest Agent.
- VM.GuestAgent.FileSystemMgmt: freeze, thaw o trim de sistemas de archivos mediante QEMU Guest Agent.
- VM.GuestAgent.FileWrite: escribir archivos en el invitado mediante QEMU Guest Agent.
- VM.GuestAgent.Unrestricted: ejecutar comandos arbitrarios del QEMU Guest Agent.
- VM.Migrate: migrar la VM a otro servidor del clúster.
- VM.PowerMgmt: administración de energía de la VM.
- VM.Replicate: configurar y ejecutar replicación del invitado.
- VM.Snapshot.Rollback: revertir la VM a uno de sus snapshots.
- VM.Snapshot: crear o eliminar snapshots de la VM.

Privilegios relacionados con almacenamiento

- Datastore.Allocate: crear, modificar o eliminar un datastore y borrar volúmenes.
- Datastore.AllocateSpace: asignar espacio en un datastore.
- Datastore.AllocateTemplate: asignar o subir plantillas e imágenes ISO.
- Datastore.Audit: consultar o explorar un datastore.

ADVERTENCIA: Permissions.Modify y Sys.Modify deben utilizarse con cuidado, ya que permiten modificar partes sensibles o potencialmente peligrosas de la configuración del sistema.

ADVERTENCIA: Es importante comprender las reglas de herencia de ACL antes de asignar roles, ya que los permisos pueden propagarse a lo largo del árbol de objetos.

14.7.3 Objetos y rutas

Los permisos de acceso se asignan a objetos como máquinas virtuales, almacenamientos o Resource Pools. Proxmox VE utiliza rutas similares a las de un sistema de archivos para direccionar estos objetos. Las rutas forman un árbol natural y los permisos de niveles superiores pueden propagarse opcionalmente hacia niveles inferiores.

Las rutas pueden ser plantillas. Cuando una llamada API requiere permisos sobre una ruta parametrizada, esta puede incluir referencias a parámetros de la llamada entre llaves.

Por ejemplo, el permiso /nodes/{node} al llamar a /nodes/mynode/status exige permisos sobre /nodes/mynode. De forma similar, la ruta {path} en una petición PUT a /access/acl se refiere al parámetro path del método.

Ejemplos de rutas:

- /nodes/{node}: acceso a nodos Proxmox VE.
- /vms: cubre todas las VMs.
- /vms/{vmid}: acceso a una VM concreta.
- /storage/{storeid}: acceso a un almacenamiento concreto.
- /pool/{poolname}: acceso a los recursos contenidos en un pool.
- /access/groups: administración de grupos.
- /access/realms/{realmid}: acceso administrativo a realms.

Herencia

Las rutas de objetos forman un árbol y, de forma predeterminada, los permisos pueden heredarse hacia abajo mediante el flag propagate.

- Los permisos asignados directamente a un usuario siempre sustituyen a los permisos procedentes de grupos.
- Los permisos de grupo se aplican cuando el usuario pertenece a dicho grupo.
- Los permisos definidos en niveles más profundos sustituyen a los heredados de un nivel superior.
- NoAccess cancela los demás roles en la misma ruta.

Además, un token con privilegios separados nunca puede obtener en una ruta permisos que el usuario asociado no posea.

14.7.4 Pools

Los pools permiten agrupar máquinas virtuales y datastores. Los permisos pueden asignarse directamente sobre /pool/{poolid} y serán heredados por todos los miembros del pool. Esto simplifica de forma considerable el control de acceso.

14.7.5 ¿Qué permisos necesito?

Los permisos requeridos por cada método de la API están documentados individualmente en el API Viewer de Proxmox VE. En el manual aparecen como listas que pueden interpretarse como un árbol lógico de funciones de comprobación de acceso.

```
["and", <subtests>...]
["or", <subtests>...]
```

En una expresión and deben ser verdaderos todos los elementos; en una expresión or basta con que sea verdadero cualquiera de ellos.

```
["perm", <path>, [ <privileges>... ], <options>...]
```

path es una ruta parametrizada. Por defecto deben concederse todos los privilegios indicados; si se utiliza la opción any, basta cualquiera de ellos. Si aparece require-param, ese parámetro pasa a ser obligatorio aunque el esquema de la API lo marque como opcional.

```
["userid-group", [ <privileges>... ], <options>...]
```

El llamante debe poseer alguno de los privilegios listados en /access/groups. Además:

- Si groups_param está definido, la llamada API contiene un parámetro groups obligatorio y el llamante debe tener alguno de los privilegios indicados sobre todos los grupos enumerados.
- Si groups_param no está definido, el usuario indicado por userid debe existir y pertenecer a un grupo sobre el que el llamante tenga alguno de los privilegios indicados mediante /access/groups/<group>.

```
["userid-param", "self"]
```

El valor del parámetro userid debe corresponder al propio usuario que ejecuta la acción. Suele utilizarse junto con or para permitir que un usuario actúe sobre sí mismo aunque no tenga privilegios elevados.

```
["userid-param", "Realm.AllocateUser"]
```

El usuario necesita Realm.AllocateUser sobre /access/realm/<realm>, donde <realm> es el realm del usuario indicado por userid. El usuario no necesita existir todavía, ya que los IDs se expresan como <username>@<realm>.

```
["perm-modify", <path>]
```

El usuario necesita Permissions.Modify en la ruta indicada o, dependiendo de la ruta, uno de estos privilegios alternativos:

- /storage/...: Datastore.Allocate.
- /vms/...: VM.Allocate.
- /pool/...: Pool.Allocate.

Si la ruta está vacía, se requiere Permissions.Modify sobre /access.

Si el usuario no posee Permissions.Modify, solo puede delegar subconjuntos de sus propios privilegios en esa ruta. Por ejemplo, un usuario con PVEVMAdmin podría asignar PVEVMUser, pero no PVEAdmin.

14.8 Herramienta de línea de comandos

La mayoría de usuarios administrará cuentas desde la GUI, pero Proxmox VE también dispone de la herramienta pveum (Proxmox VE User Manager). Las herramientas CLI de Proxmox VE son wrappers de la API, por lo que estas funciones también están disponibles a través de la REST API.

Ayuda general:

```
pveum
```

Ayuda detallada de un comando:

```
pveum help user add
```

Crear un usuario:

```
pveum user add testuser@pve -comment "Just a test"
```

Establecer o cambiar la contraseña - no todos los realms lo admiten:

```
pveum passwd testuser@pve
```

Deshabilitar un usuario:

```
pveum user modify testuser@pve -enable 0
```

Crear un grupo:

```
pveum group add testgroup
```

Crear un rol:

```
pveum role add PVE_Power-only -privs "VM.PowerMgmt VM.Console"
```

14.9 Ejemplos reales

14.9.1 Grupo de administradores

Un administrador puede querer crear un grupo de usuarios con privilegios completos sin utilizar directamente la cuenta root.

Primero se crea el grupo:

```
pveum group add admin -comment "System Administrators"
```

Después se asigna el rol Administrator:

```
pveum acl modify / -group admin -role Administrator
```

Por último se incorporan usuarios al grupo:

```
pveum user modify testuser@pve -group admin
```

14.9.2 Auditores

Puede concederse acceso de solo lectura asignando el rol PVEAuditor a usuarios o grupos.

Ejemplo 1: permitir que joe@pve pueda verlo todo:

```
pveum acl modify / -user joe@pve -role PVEAuditor
```

Ejemplo 2: permitir que joe@pve vea todas las máquinas virtuales:

```
pveum acl modify /vms -user joe@pve -role PVEAuditor
```

14.9.3 Delegar la gestión de usuarios

Para delegar la administración de usuarios a joe@pve:

```
pveum acl modify /access -user joe@pve -role PVEUserAdmin
```

joe@pve podrá añadir y eliminar usuarios y modificar otros atributos, como contraseñas.

Es un rol muy potente, por lo que normalmente conviene limitarlo a determinados realms y grupos. El siguiente ejemplo permite a joe@pve modificar usuarios del realm pve siempre que pertenezcan al grupo customers:

```
pveum acl modify /access/realm/pve -user joe@pve -role PVEUserAdmin
pveum acl modify /access/groups/customers -user joe@pve -role PVEUserAdmin
```

NOTA: El usuario puede añadir otros usuarios, pero únicamente si pertenecen al grupo customers y al realm pve.

14.9.4 API Token limitado para monitorización

Los permisos de un API Token siempre son un subconjunto de los permisos de su usuario asociado. Por tanto, un token no puede ejecutar una tarea que el usuario base no tenga permitida.

Primero se concede a joe@pve el rol PVEVMAdmin sobre todas las VMs:

```
pveum acl modify /vms -user joe@pve -role PVEVMAdmin
```

A continuación se crea un token con privilegios separados y acceso únicamente de lectura para monitorización:

```
pveum user token add joe@pve monitoring -privsep 1
pveum acl modify /vms -token 'joe@pve!monitoring' -role PVEAuditor
```

Para comprobar los permisos efectivos:

```
pveum user permissions joe@pve
pveum user token permissions joe@pve monitoring
```

14.9.5 Resource Pools

En una empresa suele ser habitual dividir recursos y responsabilidades por departamentos. El ejemplo del manual crea un pool para un departamento de desarrollo de software.

Primero se crea el grupo:

```
pveum group add developers -comment "Our software developers"
```

Después se crea un usuario que pertenece al grupo:

```
pveum user add developer1@pve -group developers -password
```

NOTA: El parámetro -password solicitará la contraseña de forma interactiva.

Se crea un Resource Pool para el departamento:

```
pveum pool add dev-pool --comment "IT development pool"
```

Finalmente se asignan permisos sobre el pool:

```
pveum acl modify /pool/dev-pool/ -group developers -role PVEAdmin
```

A partir de ese momento, los desarrolladores del grupo pueden administrar los recursos asignados a dev-pool.

Fin del Capítulo 14 · Gestión de usuarios

Con esta Parte 14B queda completada la traducción del Capítulo 14. La siguiente entrega comienza el Capítulo 15 · Alta disponibilidad (High Availability).

CRITERIO DE TRADUCCIÓN: Se han conservado nombres de roles, privilegios, comandos, rutas y sintaxis exactamente como deben utilizarse. El contenido explicativo se ha traducido al castellano manteniendo la organización y el alcance del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 15A

Alta disponibilidad (High Availability)

Requisitos · Recursos · HA Manager · estados · LRM/CRM · simulador · configuración · grupos

Disponibilidad	Orquestación	Recuperación	Placement
Uptime + redundancia	ha-manager	LRM + CRM + fencing	recursos + grupos

Basado en Proxmox VE Administration Guide, Release 9.2.4

Traducción progresiva al castellano. Se conservan comandos, rutas, estados y nombres técnicos.

Capítulo 15 · Alta disponibilidad

La sociedad actual depende en gran medida de la información ofrecida por sistemas informáticos a través de la red. Los dispositivos móviles han aumentado esa dependencia al permitir el acceso desde prácticamente cualquier lugar y en cualquier momento. Cuando se prestan este tipo de servicios, resulta muy importante que estén disponibles la mayor parte del tiempo.

La disponibilidad puede definirse matemáticamente como la relación entre el tiempo total durante el que un servicio puede utilizarse dentro de un intervalo y la duración total de dicho intervalo. Normalmente se expresa como porcentaje de uptime anual.

Disponibilidad y tiempo de inactividad anual

Disponibilidad	Tiempo de inactividad por año
99%	3,65 días
99,9%	8,76 horas
99,99%	52,56 minutos
99,999%	5,26 minutos
99,9999%	31,5 segundos
99,99999%	3,15 segundos

Existen distintas maneras de aumentar la disponibilidad. La solución más elegante consiste en diseñar el software para ejecutarse simultáneamente en varios hosts, de forma que sea el propio software quien detecte fallos y realice failover. Esto puede ser sencillo para ciertos servicios de solo lectura, pero en general es complejo o incluso imposible si no se controla el software.

El manual propone varias medidas que no requieren modificar las aplicaciones:

- Utilizar componentes de servidor fiables.
- Eliminar puntos únicos de fallo mediante componentes redundantes.
- Reducir el tiempo de indisponibilidad mediante detección automática de errores y failover automático.

NOTA: Los componentes con la misma función pueden ofrecer niveles de fiabilidad diferentes según su calidad. Los fabricantes suelen comercializar como componentes “server” aquellos orientados a mayor fiabilidad, normalmente a un precio superior.

Entre las medidas de redundancia se incluyen:

- UPS (sistema de alimentación ininterrumpida).
- Fuentes de alimentación redundantes.
- Memoria ECC.
- Hardware de red redundante.
- RAID para almacenamiento local.
- Almacenamiento distribuido y redundante para los datos de las VMs.

Para reducir el tiempo de caída se recomienda disponer de administradores accesibles rápidamente, piezas o nodos de repuesto, detección automática de errores mediante ha-manager y failover automático también gestionado por ha-manager.

Los entornos de virtualización como Proxmox VE facilitan la alta disponibilidad porque desacoplan los servicios del hardware concreto. También permiten utilizar almacenamiento y redes redundantes; si un host falla, los servicios pueden iniciarse en otro nodo del clúster.

Proxmox VE incorpora la pila ha-manager, que actúa de forma similar a un administrador automatizado: se definen los recursos que debe gestionar, supervisa su funcionamiento y realiza failover a otro nodo si detecta errores. También procesa solicitudes normales del usuario para iniciar, detener, reubicar o migrar servicios.

CONSEJO: Aumentar la disponibilidad del 99% al 99,9% es relativamente sencillo. En cambio, pasar de 99,9999% a 99,99999% es muy difícil y costoso. El manual indica tiempos típicos de detección de error y failover de aproximadamente 2 minutos para ha-manager, por lo que no debe interpretarse como una solución capaz de alcanzar cualquier cifra de disponibilidad.

15.1 Requisitos

Antes de empezar con HA deben cumplirse los siguientes requisitos:

- Al menos tres nodos de clúster, para disponer de quorum fiable.
- Almacenamiento compartido para máquinas virtuales y contenedores.
- Redundancia de hardware en todos los puntos críticos.
- Uso de componentes de servidor fiables.

De forma opcional puede utilizarse un watchdog de hardware. Si no está disponible, Proxmox VE recurre al watchdog software del kernel de Linux (softdog).

15.2 Recursos

La unidad principal de gestión del HA Manager se denomina recurso HA o servicio HA. Actualmente, un recurso HA puede ser una máquina virtual (VM) o un contenedor Linux (CT).

Cada recurso HA se identifica de manera única mediante un Service ID (sid), formado por el tipo de recurso y un identificador específico. Para una VM con VMID 100, por ejemplo:

```
vm:100
```

Un recurso HA solo puede ejecutarse en nodos donde estén disponibles todas sus dependencias. De forma predeterminada, HA Manager considera elegibles todos los nodos del clúster.

Para mantener ese comportamiento predeterminado, una VM o CT debería:

- Usar opciones compatibles con todos los nodos del clúster.
- Usar únicamente almacenamiento compartido accesible desde todos los nodos.
- Usar interfaces de red que existan en todos los nodos.
- Usar dispositivos passthrough presentes en todos los nodos.

Si alguna dependencia solo existe en un subconjunto de nodos, el recurso HA debe restringirse a esos nodos mediante una regla de afinidad estricta. HA Manager limitará entonces el placement, la recuperación y la migración a los nodos incluidos en la regla.

15.3 Tareas de administración

El primer paso consiste en habilitar HA para un recurso, añadiéndolo a la configuración de recursos HA. Puede hacerse desde la GUI o con ha-manager:

```
ha-manager add vm:100
```

A partir de ese momento la pila HA intenta iniciar el recurso y mantenerlo en funcionamiento. También puede definirse el estado solicitado. Para detenerlo:

```
ha-manager set vm:100 --state stopped
```

Y para volver a iniciarlo:

```
ha-manager set vm:100 --state started
```

Los comandos habituales de administración de VMs y contenedores se integran con HA. Por ejemplo:

```
qm start 100
```

equivale a solicitar el estado started. Del mismo modo, qm stop solicita el estado stopped.

NOTA: La pila HA funciona de forma completamente asíncrona y necesita comunicarse con otros miembros del clúster. Por ello pueden transcurrir algunos segundos hasta observar el resultado de una acción.

Para consultar la configuración HA actual:

```
ha-manager config
vm:100
    state stopped
```

Para consultar el estado real del HA Manager y de sus recursos:

```
ha-manager status
quorum OK
master node1 (active, Wed Nov 23 11:07:23 2016)
lrm elsa (active, Wed Nov 23 11:07:19 2016)
service vm:100 (node1, started)
```

Puede iniciarse una migración online hacia otro nodo:

```
ha-manager migrate vm:100 node2
```

La migración online intenta mantener la VM en ejecución. Como debe transferir por red toda la memoria utilizada, en determina dos casos puede ser más rápido detener la VM y arrancarla de nuevo en el destino mediante relocate:

```
ha-manager relocate vm:100 node2
```

Finalmente puede retirarse el recurso de la configuración HA:

```
ha-manager remove vm:100
```

De forma predeterminada, esto también elimina el recurso de cualquier regla que lo referencie y elimina las reglas que solo contengan ese recurso. Puede cambiarse ese comportamiento con --purge 0.

NOTA: Eliminar un recurso de la configuración HA no lo inicia ni lo detiene.

Todas estas tareas también pueden realizarse desde la interfaz gráfica.

15.4 Cómo funciona

Para proporcionar alta disponibilidad se ejecutan dos daemons en cada nodo:

pve-ha-lrm - Local Resource Manager (LRM). Controla los servicios que se ejecutan en el nodo local, lee los estados solicitados en el estado del manager y ejecuta los comandos correspondientes.

pve-ha-crm - Cluster Resource Manager (CRM). Toma las decisiones globales del clúster, envía comandos a los LRM, procesa resultados, mueve recursos a otros nodos cuando se produce un fallo y gestiona el fencing de nodos.

NOTA: Los locks son proporcionados por pmxcfs. Garantizan que cada LRM esté activo una sola vez. Como el LRM solo ejecuta acciones mientras conserva su lock, un nodo fallido puede considerarse fenced cuando el CRM logra adquirir su lock; entonces los servicios HA pueden recuperarse de forma segura en otro nodo.

15.4.1 Estados de servicio

El CRM utiliza una enumeración de estados para registrar el estado actual de cada servicio. Se muestra en la GUI y puede consultarse con ha-manager status.

```
ha-manager status
quorum OK
master elsa (active, Mon Nov 21 07:23:29 2016)
lrm elsa (active, Mon Nov 21 07:23:22 2016)
service ct:100 (elsa, stopped)
service ct:102 (elsa, started)
service vm:501 (elsa, started)
```

stopped - El servicio está detenido y el LRM lo ha confirmado. Si el LRM detecta que sigue ejecutándose, vuelve a detenerlo.

request_stop - El servicio debe detenerse; el CRM espera confirmación del LRM.

stopping - Existe una solicitud de parada pendiente, pero el CRM todavía no ha recibido la confirmación correspondiente.

started - El servicio debe estar activo. Si no está ejecutándose, el LRM debe iniciarlo lo antes posible. Si falla, el LRM intenta reiniciarlo según la Start Failure Policy.

starting - Solicitud de inicio pendiente; el CRM todavía no tiene confirmación de que el servicio esté funcionando.

fence - Espera al fencing del nodo porque el nodo que alojaba el servicio no pertenece a la partición del clúster con quorum. Tras un fencing correcto, el servicio pasa a recovery.

recovery - Espera la recuperación del servicio. HA Manager busca un nodo nuevo válido según nodos online, quorum y reglas de afinidad. Cuando encuentra uno, mueve el servicio y lo sitúa inicialmente en stopped; si está configurado para ejecutarse, el nuevo nodo lo inicia.

freeze - No modificar el estado del servicio. Se utiliza al reiniciar un nodo o al reiniciar el daemon LRM.

ignored - Actuar como si el servicio no estuviera gestionado por HA. Es útil para obtener control temporal completo sin eliminarlo de la configuración HA. No se reubica ante fallos de nodo.

migrate - Migrar el servicio en vivo a otro nodo.

error - El servicio queda deshabilitado por errores del LRM y necesita intervención manual.

queued - El servicio acaba de añadirse y el CRM todavía no lo ha procesado.

disabled - El servicio está detenido y marcado como deshabilitado.

15.4.2 Local Resource Manager (LRM)

El Local Resource Manager (pve-ha-lrm) arranca como daemon durante el boot y espera hasta que el clúster HA dispone de quorum y, por tanto, funcionan los locks globales del clúster.

Puede encontrarse en tres estados:

wait for agent lock - El LRM espera su lock exclusivo. También se utiliza como estado de reposo cuando no hay servicios configurados.

active - El LRM mantiene su lock exclusivo y tiene servicios configurados.

lost agent lock - El LRM ha perdido su lock; se ha producido un fallo y se ha perdido el quorum.

Cuando entra en estado active, el LRM lee /etc/pve/ha/manager_status y determina qué comandos debe ejecutar para los servicio s que posee. Cada comando se ejecuta mediante un worker independiente. Los workers se ejecutan en paralelo, con un máximo predeterminado de 4. Este valor puede modificarse mediante la clave max_worker de la configuración del Datacenter.

NOTA: El valor predeterminado de cuatro workers concurrentes puede no ser apropiado en todos los entornos. Cuatro migraciones en vivo simultáneas pueden saturar redes lentas o servicios con mucha memoria. Debe elegirse un valor que mantenga la congestión bajo control incluso en el peor caso. En entornos de alto rendimiento también puede aumentarse.

Cada comando solicitado por el CRM se identifica de forma única mediante un UID. Cuando un worker termina, su resultado se procesa y se escribe en:

```
/etc/pve/nodes/<nodename>/lrm_status
```

El CRM recoge el resultado desde ese archivo y su máquina de estados actúa en función del resultado.

Normalmente las acciones entre CRM y LRM están sincronizadas mediante el UID: el CRM solicita un estado, el LRM ejecuta la acción una sola vez y devuelve el resultado con el mismo UID. Esto evita ejecutar comandos obsoletos. Las excepciones son sto p y error, que no dependen de un resultado previo del mismo modo.

NOTA: La pila HA registra todas las acciones. Para comprender qué ha ocurrido y por qué, conviene revisar tanto el LRM como el CRM: journalctl -u pve-ha-lrm en los nodos donde se encuentra el servicio y journalctl -u pve-ha-crm en el nodo que sea master del CRM.

15.4.3 Cluster Resource Manager (CRM)

El Cluster Resource Manager (pve-ha-crm) se ejecuta en cada nodo y espera el manager lock, que solo puede estar en manos de un nodo al mismo tiempo. El nodo que logra adquirirlo pasa a ser el CRM master.

Sus tres estados son equivalentes a los del LRM: wait for agent lock, active y lost agent lock.

Su función principal es gestionar los servicios marcados como altamente disponibles e intentar imponer siempre el estado solicitado. Por ejemplo, un servicio cuyo estado solicitado sea started se inicia si no está ejecutándose; si falla, intenta arrancarse de nuevo. De esta forma, el CRM dicta las acciones que debe ejecutar el LRM.

Cuando un nodo abandona el quorum del clúster, su estado pasa a unknown. Si el CRM activo consigue adquirir el lock del nodo fallido, puede tomar sus servicios y reiniciarlos en otro nodo.

Si un miembro del clúster detecta que ya no forma parte del quorum, su LRM espera a que se forme un nuevo quorum. Mientras no exista quorum, el nodo no puede resetear el watchdog. Si existen servicios activos, o si los procesos LRM o CRM no se ejecutan correctamente, el watchdog provocará un reinicio al expirar; el manual indica un timeout de 60 segundos.

Si un nodo tiene un CRM activo pero el LRM está idle, una pérdida de quorum no provoca un self-fence reset. Los archivos de estado y configuración que usa el CRM están respaldados por pmxcfs, que pasa a modo solo lectura al perder quorum. El watchdog abierto protege frente a situaciones en las que otro CRM pudiera asumir el control sin conocer el estado anterior.

Si no existe ningún servicio configurado durante más de 15 minutos, el CRM vuelve automáticamente al estado idle y cierra por completo el watchdog.

15.5 Simulador HA

El simulador HA permite probar y aprender las funciones de alta disponibilidad sin configurar un clúster real. Por defecto si mula un clúster de tres nodos con seis VMs y permite añadir o eliminar VMs o contenedores.

Instalación:

```
apt install pve-ha-simulator
```

El paquete también puede instalarse en un sistema basado en Debian sin el resto de paquetes Proxmox VE, descargándolo y copiándolo al sistema. Si se instala mediante apt desde un archivo local, apt resolverá las dependencias necesarias.

Para iniciar el simulador en una máquina remota se necesita redirección X11. En Linux puede utilizarse:

```
ssh root@<IPofPVE> -Y
```

En Windows, el manual cita MobaXterm como opción.

Cree un directorio de trabajo donde el simulador guarde su estado y la configuración predeterminada:

```
mkdir working
```

Después inicie el simulador pasando ese directorio:

```
pve-ha-simulator working/
```

Desde el simulador puede iniciar, detener y migrar servicios HA simulados y observar qué ocurre cuando falla un nodo.

15.6 Configuración

La pila HA está integrada en la API de Proxmox VE. Puede administrarse con ha-manager, con la interfaz web o directamente mediante la API para herramientas de automatización.

Todos los archivos de configuración HA se encuentran bajo:

```
/etc/pve/ha/
```

Al estar dentro del sistema de archivos de configuración del clúster, se distribuyen automáticamente a todos los nodos y todo s ellos comparten la misma configuración HA.

15.6.1 Recursos

El archivo /etc/pve/ha/resources.cfg contiene la lista de recursos gestionados por ha-manager. Una entrada tiene el formato:

```
<type>: <name>
<property> <value>
...
```

El tipo y el nombre específico del recurso, separados por dos puntos, forman el HA resource ID utilizado por ha-manager, por ejemplo vm:100 o ct:101.

auto-rebalance - Valor booleano, predeterminado 1. Permite migrar el recurso durante el rebalanceo automático.

comment - Descripción del recurso.

failback - Valor booleano, predeterminado 1. Migra automáticamente el recurso hacia el nodo con mayor prioridad según sus reglas de afinidad cuando vuelve a estar disponible un nodo con prioridad superior a la actual.

group - Identificador del HA group.

max_relocate - Número máximo de intentos de relocate cuando un recurso no consigue arrancar. Predeterminado: 1.

max_restart - Número máximo de intentos de reinicio en el mismo nodo después de un fallo de inicio. Predeterminado: 1. Al alcanzar el límite, HA Manager intentará reubicar el recurso en un nodo elegible.

state - Estado solicitado: disabled, enabled, ignored, started o stopped. El valor predeterminado es started. enabled es un alias de started.

Semántica de los principales estados configurables:

started - El CRM intenta iniciar el recurso y mantenerlo funcionando. Ante fallos de nodo o de inicio intenta recuperarlo; si todo falla, el servicio termina en estado error.

stopped - El CRM intenta mantener el recurso detenido, pero ante fallos de nodo continúa intentando reubicarlo.

disabled - El CRM intenta mantenerlo detenido y no intenta reubicarlo ante fallos de nodo. Este estado se utiliza especialmente para recuperación de errores, ya que permite sacar un recurso del estado error.

ignored - El recurso se elimina del estado del manager; CRM y LRM dejan de tocarlo. Las llamadas API actúan directamente, evitando la pila HA. No se reubica ante fallos de nodo.

Ejemplo real con una VM y un contenedor:

```
# /etc/pve/ha/resources.cfg
vm: 501
    state started
    max_relocate 2

ct: 102
    # utilizar valores predeterminados
```

Configuración equivalente mediante ha-manager:

```
ha-manager add vm:501 --state started --max_relocate 2
ha-manager add ct:102
```

15.6.2 Grupos

NOTA: Los HA Groups están obsoletos y, desde Proxmox VE 9.0, se migran a reglas HA Node Affinity. Esta sección se conserva porque forma parte del manual 9.2.4 y explica configuraciones existentes.

El archivo /etc/pve/ha/groups.cfg define grupos de nodos del clúster. Un recurso puede restringirse para ejecutarse únicamente en los miembros de un grupo.

```
group: <group>
    nodes <node_list>
    <property> <value>
    ...
```

comment - Descripción del grupo.

nodes - Lista de nodos miembros. Puede asignarse una prioridad a cada nodo con <node>:<pri>. El recurso se ejecuta en los nodos disponibles con mayor prioridad; si hay varios con la misma prioridad, los recursos se distribuyen entre ellos. Las prioridades son relativas: cuanto mayor el número, mayor la prioridad.

nofailback - Booleano, predeterminado 0. Normalmente el CRM migra un servicio al nodo de mayor prioridad cuando este vuelve a estar online. Activar nofailback evita ese comportamiento.

restricted - Booleano, predeterminado 0. Los recursos ligados a un grupo restricted solo pueden ejecutarse en los nodos definidos. Si ninguno está online, el recurso queda en stopped. En grupos no restringidos, el recurso puede ejecutarse en cualquier nodo si todos los miembros del grupo están offline, y volverá a migrar cuando alguno reaparezca.

Un requisito habitual es preferir un nodo concreto sin impedir que el recurso se ejecute en otros. Puede utilizarse un grupo no restringido con un único miembro:

```
ha-manager groupadd prefer_node1 --nodes node1
```

En clústeres mayores puede definirse un orden de failover más detallado. Por ejemplo, preferir node1; si no está disponible, repartir entre node2 y node3; y utilizar node4 como última alternativa:

```
ha-manager groupadd mygroup1 -nodes "node1:2,node2:1,node3:1,node4"
```

Si un recurso depende de elementos solo presentes en node1 y node2, debe crearse un grupo restricted:

```
ha-manager groupadd mygroup2 -nodes "node1,node2" -restricted
```

El resultado en groups.cfg sería:

```
group: prefer_node1
    nodes node1

group: mygroup1
    nodes node2:1,node4,node1:2,node3:1

group: mygroup2
    nodes node2,node1
    restricted 1
```

La opción nofailback resulta útil para evitar movimientos no deseados durante tareas administrativas. Por ejemplo, si se migra un servicio a un nodo que no tiene la prioridad más alta, nofailback evita que HA Manager lo devuelva inmediatamente.

También es útil cuando un servicio se recuperó en otro nodo después de que su nodo original fuera fenced. Mientras el administrador repara y comprueba la estabilidad del nodo recuperado, nofailback evita que los servicios vuelvan automáticamente a él.

Fin de la Parte 15A

Esta entrega cubre desde el inicio del Capítulo 15 hasta 15.6.2 Grupos. La siguiente parte continuará con 15.6.3 Rules, conflictos de reglas, fencing y watchdog, recuperación de errores, mantenimiento de nodos y Cluster Resource Scheduling (CRS).

CRITERIO DE TRADUCCIÓN: Se ha mantenido la estructura del manual y se han conservado literalmente nombres de estados, servicios, comandos, rutas y claves de configuración cuando deben utilizarse tal como aparecen en Proxmox VE.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 15B

Alta disponibilidad (High Availability)

HA Rules · Fencing · Watchdog · Recuperación · Mantenimiento · CRS · Balanceo de carga

Reglas HA	Seguridad	Mantenimiento	Planificación
Node/Resource Affinity	Fencing + watchdog	Maintenance + disarm	CRS + load balancer

Basado en Proxmox VE Administration Guide, Release 9.2.4

Continuación del Capítulo 15. Se conservan comandos, rutas, estados y sintaxis técnica.

15.6.3 Reglas

Las reglas HA permiten imponer restricciones sobre los recursos administrados por HA. Se definen en el archivo /etc/pve/ha/rules.cfg.

```
<type>: <rule>
resources <resources_list>
<property> <value>
...
```

comment: <string> - Descripción de la regla HA.

disable: <boolean> - Indica si la regla HA está deshabilitada.

resources: <type>:<name>{,<type>:<name>}* - Lista de IDs de recursos HA. Cada ID combina tipo y nombre específico separados por dos puntos, por ejemplo vm:100 o ct:101.

Tipos de reglas HA disponibles:

Tipo	Descripción
node-affinity	Aplica afinidad entre uno o varios recursos HA y uno o varios nodos. Una afinidad positive indica que los recursos deberían o deben permanecer en los nodos especificados; negative indica que deberían o deben evitarlos. strict define si la restricción es blanda (should) o dura (must).
resource-affinity	Aplica afinidad entre dos o más recursos HA. positive obliga a mantenerlos en el mismo nodo; negative obliga a mantenerlos en nodos distintos.

Reglas de afinidad de nodo

Por defecto, un recurso HA puede ejecutarse en cualquier nodo del clúster. Un requisito frecuente es que un recurso prefiera un nodo concreto. El siguiente ejemplo hace que vm:100 prefiera node1:

```
ha-manager rules add node-affinity ha-rule-vm100 --resources vm:100 --nodes node1
```

Las reglas de afinidad de nodo no son estrictas por defecto. Si ninguno de los nodos indicados está disponible, el recurso pu ede moverse a otros nodos. Para restringir vm:100 exclusivamente a node1:

```
ha-manager rules set node-affinity ha-rule-vm100 --strict 1
```

La afinidad es positive por defecto. En clústeres grandes puede resultar más práctico utilizar un enfoque de exclusión con af inidad negative. El siguiente ejemplo hace que ct:200 y vm:300 eviten node3:

```
ha-manager rules add node-affinity ha-rule-negative \
--affinity negative --resources ct:200,vm:300 --nodes node3
```

También pueden asignarse prioridades a los nodos para definir una cascada de failover. En el siguiente caso, vm:400 y ct:500 prefieren node1; si no está disponible, se reparten entre node2 y node3; y node4 queda como última opción con prioridad 0 implícita:

```
ha-manager rules add node-affinity priority-cascade \
--resources vm:400,ct:500 --nodes "node1:2,node2:1,node3:1,node4"
```

Configuración resultante de ejemplo:

```
node-affinity: ha-rule-vm100
nodes node1
resources vm:100
strict 1

node-affinity: ha-rule-negative
nodes node3
resources ct:200,vm:300
affinity negative

node-affinity: priority-cascade
nodes node1:2,node2:1,node3:1,node4
resources vm:400,ct:500
```

Propiedades de una regla Node Affinity:

affinity: <negative | positive> (default = positive) - positive coloca los recursos en los nodos indicados; negative intenta colocarlos en cualquier nodo excepto los indicados.

nodes: <node>[:<pri>]{,<node>[:<pri>]}* - Lista de nodos con prioridad opcional. Los recursos se ejecutan en los nodos disponibles de mayor prioridad; si hay varios en la misma clase, se distribuyen entre ellos. Las prioridades son relativas.

resources: <type>:<name>{,<type>:<name>}* - Lista de IDs de recursos HA afectados.

strict: <boolean> (default = 0) - En modo no estricto los nodos definidos son preferidos; si no hay ninguno disponible se permite otro nodo. En modo estricto los recursos quedan restringidos a los nodos definidos y se detienen si ninguno está disponible.

Reglas de afinidad de recursos

Otro requisito habitual es mantener varios recursos HA en el mismo nodo o, por el contrario, obligarlos a ejecutarse en nodos separados. Estas restricciones se conocen también como Affinity/Anti-Affinity.

Si vm:100 y vm:200 intercambian mucho tráfico - por ejemplo un servidor web y una base de datos - una afinidad positiva puede reducir latencia y tráfico de red al mantenerlos juntos:

```
ha-manager rules add resource-affinity keep-together \
--affinity positive --resources vm:100,vm:200
```

NOTA: Cuando dos o más reglas positivas comparten recursos HA, se tratan como una única regla positiva. Por ejemplo, una regla vm:100+vm:101 y otra vm:101+vm:102 equivalen a mantener vm:100, vm:101 y vm:102 juntos.

NOTA: Si los recursos de una regla positiva están ejecutándose en nodos distintos, CRS los mueve al nodo donde ya se encuentre la mayoría. En caso de empate se selecciona el nodo cuyo nombre aparezca primero alfabéticamente.

Para cargas intensivas o distribuidas, una afinidad negativa puede evitar que varios recursos compitan por el mismo hardware. Ejemplo:

```
ha-manager rules add resource-affinity keep-separate \
--affinity negative --resources vm:200,ct:300
```

A diferencia de las reglas de afinidad de nodo, las reglas de afinidad de recursos son estrictas por defecto. Si sus restricciones no pueden cumplirse, el HA Manager coloca el recurso en estado recovery durante un failover o en error en otros casos.

Configuración resultante de ejemplo:

```
resource-affinity: keep-together
affinity positive
resources vm:100,vm:200

resource-affinity: keep-separate
affinity negative
resources vm:200,ct:300
```

Interacción entre reglas positivas y negativas de recursos

Si un recurso incluido en una regla positiva también forma parte de una regla negativa, la afinidad negativa se extiende al resto de recursos de la regla positiva. Por ejemplo, si vm:100, vm:101 y vm:102 deben permanecer juntos y vm:100 debe separarse de ct:200, entonces vm:101 y vm:102 también quedan en afinidad negativa respecto de ct:200.

Si dos o más recursos aparecen simultáneamente en una regla positiva y en una negativa, las reglas se deshabilitan por conflicto: no es posible exigir que los mismos recursos estén juntos y separados al mismo tiempo.

Interacción entre afinidad de nodo y afinidad positiva de recursos

Si recursos de una regla de afinidad positiva también están sujetos a una regla de afinidad de nodo, los demás recursos de la regla positiva heredan esa afinidad de nodo. Por ejemplo, si vm:100, vm:101 y vm:102 deben ir juntos y vm:102 está restringida a node3, entonces vm:100 y vm:101 también quedan restringidas a node3.

Si varios recursos de una regla positiva pertenecen a reglas de afinidad de nodo diferentes, se produce un error y dichas reglas se deshabilitan, ya que esta combinación no está soportada actualmente.

affinity: <negative | positive> - positive mantiene los recursos en el mismo nodo; negative los mantiene en nodos distintos.

resources: <type>:<name>{,<type>:<name>}* - Lista de IDs de recursos HA afectados.

15.6.4 Conflictos y errores de reglas

Las reglas HA pueden producir restricciones complejas. Antes de aplicar una regla nueva o modificada, Proxmox VE comprueba su viabilidad para evitar ambigüedad en el planificador CRS. Las reglas que fallan estas pruebas quedan deshabilitadas hasta res olver el conflicto.

Comprobaciones de viabilidad indicadas por el manual:

- Un recurso HA solo puede formar parte de una única regla HA de afinidad de nodo.
- Las reglas negativas de afinidad de nodo no pueden especificar prioridades de nodos.
- Una regla negativa de afinidad de nodo no puede especificar todos los nodos del clúster.
- Una regla de afinidad de recursos debe contener al menos dos recursos HA.
- Una regla negativa de afinidad de recursos no puede incluir más recursos HA que nodos existan en el clúster, porque no habría nodos suficientes para separarlos.
- Una regla positiva y una negativa de afinidad de recursos no pueden contener los mismos dos o más recursos, ya que exigirían mantenerlos juntos y separados a la vez.
- Un recurso HA solo puede estar simultáneamente en una regla de afinidad de nodo y una de afinidad de recursos si la regla de nodos tiene una sola clase de prioridad.
- Los recursos de una regla positiva de afinidad de recursos pueden pertenecer como máximo a una única regla de afinidad de nodo.
- Los recursos de una regla negativa de afinidad no pueden quedar restringidos por reglas de nodo a menos nodos que el número de recursos que deben mantenerse separados.

15.7 Fencing

Ante un fallo de nodo, el fencing garantiza que el nodo erróneo quede realmente fuera de servicio. Esto es imprescindible par a evitar que un recurso se ejecute dos veces cuando se recupera en otro nodo.

Sin fencing, el nodo fallido quedaría en un estado desconocido y podría seguir accediendo al almacenamiento compartido. Un caso especialmente peligroso sería perder todas las redes salvo la de almacenamiento: la VM podría seguir ejecutándose y escribiendo en el almacenamiento aunque ya no fuera accesible desde la red pública. Si se inicia esa misma VM en otro nodo, ambos podrían escribir simultáneamente, provocando corrupción de datos o incluso impidiendo el montaje del almacenamiento.

15.7.1 Cómo realiza fencing Proxmox VE

Existen varios métodos posibles para aislar un nodo:

- Interruptores de alimentación externos.
- Aislar el nodo deshabilitando completamente su tráfico de red desde el switch.
- Self-fencing mediante temporizadores watchdog.

Proxmox VE integra una solución que no necesita hardware de fencing externo utilizando watchdog timers. Durante el funcionamiento normal, ha-manager reinicia periódicamente el temporizador. Si un fallo de hardware o software impide hacerlo, el temporizador expira y fuerza el reinicio completo del servidor.

Muchas placas base de servidor modernas incorporan watchdog hardware. Si no existe o no está configurado, Proxmox VE utiliza el watchdog software softdog del kernel Linux. Aunque es útil, no es independiente del hardware del servidor y por tanto ofrece menor fiabilidad que un watchdog físico.

15.7.2 Configurar un hardware watchdog

Por seguridad, los módulos de watchdog hardware están bloqueados por defecto. Para habilitar uno debe especificarse el módulo que debe cargarse en /etc/default/pve-ha-manager. Ejemplo:

```
# select watchdog module (default is softdog)
WATCHDOG_MODULE=iTCO_wdt
```

Esta configuración es leída por watchdog-mux, que carga el módulo seleccionado durante el arranque.

15.7.3 Recuperar servicios tras fencing

Después de que un nodo falle y el fencing se complete correctamente, el CRM intenta mover sus servicios a nodos que sigan online.

La elección del nodo de recuperación depende de los nodos activos, de su carga según el scheduler configurado y de las reglas de afinidad asociadas al servicio.

1. El CRM construye el conjunto de nodos disponibles para el servicio.
2. Si existe una regla de afinidad de nodo, el conjunto se reduce a los nodos de mayor prioridad permitidos por esa regla.
3. Si existe una regla de afinidad de recursos, el conjunto se reduce de nuevo para cumplir sus restricciones de agrupación o separación.
4. Finalmente, el CRM elige el nodo con menor carga según el scheduler activo para reducir el riesgo de sobrecarga.

PRECAUCIÓN: Tras la caída de un nodo, sus servicios se redistribuyen entre los nodos restantes. Esto incrementa la carga de esos nodos, especialmente en clústeres pequeños. El clúster debe dimensionarse para soportar ese escenario de peor caso.

15.7.4 Estado de fencing y watchdog

La salida de ha-manager status incluye una entrada fencing que refleja el estado del watchdog del CRM. Cada entrada LRM muestra además su propio estado.

armed - El CRM administra servicios activamente y mantiene abierto su watchdog. Cada LRM también mantiene uno mientras conserva su agent lock. Una pérdida de quorum o un fallo del daemon provoca un reset del nodo para asegurar un failover seguro.

standby - El stack HA está preparado, pero ningún CRM actúa todavía como master, por ejemplo cuando aún no hay recursos HA o el clúster acaba de arrancar. El watchdog del CRM no está abierto. Cuando un CRM toma el rol de master, el fencing pasa automáticamente a armed.

disarming - Se ha emitido disarm-ha. El CRM congela o suspende el seguimiento de servicios y espera a que todos los LRM liberen sus watchdogs. El watchdog del CRM permanece activo durante esta fase. Los LRM cambian a released según van confirmando el desarme.

disarmed - Todos los watchdogs del clúster han sido liberados. No se realiza fencing, failover ni recuperación automática. Consulte 15.12.

NOTA: watchdog-mux mantiene abierto /dev/watchdog durante toda su vida, incluso cuando ningún cliente HA está conectado. Esto evita que otros procesos reclamen el dispositivo y garantiza que HA pueda recuperarlo. Algunos drivers no soportan magic close y cerrar el dispositivo podría provocar un reset no deseado.

15.8 Política ante fallos de arranque

La política de fallo de arranque entra en juego cuando un servicio no consigue iniciarse una o varias veces. Permite controlar cuántos reintentos se realizan en el mismo nodo y cuántas veces se intenta reubicar el servicio en otro nodo.

El objetivo es superar indisponibilidades temporales de recursos compartidos en un nodo. Por ejemplo, si un almacenamiento compartido deja de estar disponible en un nodo que aún conserva quorum pero sigue siendo accesible desde otros nodos, la política de relocate puede permitir arrancar el servicio en otro lugar.

max_restart - Número máximo de intentos para reiniciar el servicio en el nodo actual. Valor predeterminado: 1.

max_relocate - Número máximo de intentos para reubicar el servicio en otro nodo. Solo se intenta después de superar max_restart en el nodo actual. Valor predeterminado: 1.

NOTA: El contador de relocate solo vuelve a cero después de que el servicio haya arrancado correctamente al menos una vez. Si se repite el arranque sin corregir el problema, solo vuelve a ejecutarse la política de restart.

15.9 Recuperación desde estado error

Si se agotan todos los intentos de recuperación, el servicio pasa al estado error. En ese estado, el stack HA deja de actuar sobre él. La única forma de salir es deshabilitar el servicio:

```
ha-manager set vm:100 --state disabled
```

También puede realizarse desde la interfaz web. El procedimiento de recuperación indicado por el manual es:

1. Devolver el recurso a un estado seguro y coherente; por ejemplo, finalizar su proceso si no pudo detenerse correctamente.
2. Deshabilitar el recurso para eliminar el indicador de error.
3. Corregir la causa que provocó los fallos.
4. Una vez resueltos todos los errores, solicitar de nuevo el arranque del servicio.

15.10 Actualizaciones de paquetes

Al actualizar ha-manager, los nodos deben actualizarse uno a uno y no todos simultáneamente.

Además de reducir el riesgo ante un posible problema específico de la instalación, el stack HA utiliza un protocolo de solicitud y confirmación entre el clúster y el Local Resource Manager. Al reiniciarse, el LRM solicita al CRM que congele todos sus servicios, evitando que el clúster los modifique durante el breve periodo en que el LRM se reinicia.

Después de esa confirmación, el LRM puede cerrar el watchdog de forma segura durante el reinicio. Este comportamiento se usa normalmente durante una actualización de paquetes y necesita que exista un CRM master activo. Si no lo hay, la actualización puede prolongarse demasiado y, en el peor caso, el watchdog podría provocar un reset.

15.11 Mantenimiento de nodos

En ocasiones es necesario realizar mantenimiento sobre un nodo, por ejemplo sustituir hardware o instalar un nuevo kernel. El stack HA admite principalmente dos enfoques:

- Para apagados o reinicios generales, puede configurarse la política de shutdown.
- Para mantenimiento sin apagado/reinicio, o que deba mantenerse activo más allá de un único reinicio, puede utilizarse el modo de mantenimiento manual.

15.11.1 Maintenance Mode

El modo de mantenimiento manual marca el nodo como no disponible para HA y provoca que todos los servicios gestionados por HA migren a otros nodos.

Los nodos destino se seleccionan entre los demás nodos disponibles según las reglas HA y el modo del Cluster Resource Scheduler (CRS). Durante cada migración se registra el nodo original para que el servicio pueda regresar automáticamente cuando se desactive el modo de mantenimiento y el nodo vuelva a estar online.

Actualmente se activa o desactiva mediante ha-manager.

Activar mantenimiento para un nodo:

```
ha-manager crm-command node-maintenance enable NODENAME
```

El comando se encola para que lo procese el CRM y puede ejecutarse desde cualquier nodo del clúster, no necesariamente desde el nodo que entra en mantenimiento.

Cuando el LRM del nodo recoge la orden, se marca como no disponible pero continúa procesando las migraciones. El watchdog de self-fencing permanece activo hasta que todos los servicios activos se hayan movido y finalicen todos los workers.

El estado del LRM mostrará maintenance mode desde que recoge la petición, no solo cuando termine de mover todos los servicios. Para comprobar que la transición ha finalizado puede verificarse que no queden servicios HA activos o buscar una línea de log similar a:

```
pve-ha-lrm[PID]: watchdog closed (disabled)
```

NOTA: El modo de mantenimiento manual no se elimina automáticamente al reiniciar el nodo. Solo se desactiva manualmente con ha-manager o si se limpia manualmente manager-status.

Desactivar mantenimiento:

```
ha-manager crm-command node-maintenance disable NODENAME
```

Cuando se procesa esta orden, el LRM vuelve a considerarse disponible. Los servicios que estaban en el nodo cuando se activó mantenimiento se moverán de vuelta.

15.11.2 Política de apagado (Shutdown Policy)

La política de shutdown determina cómo debe actuar HA cuando un nodo se apaga. En esta versión, Conditional es el valor predeterminado por compatibilidad histórica, aunque algunos usuarios pueden considerar que Migrate refleja mejor el comportamiento esperado.

Puede configurarse desde Datacenter -> Options -> HA Settings o directamente en datacenter.cfg:

```
ha: shutdown_policy=<value>
```

Migrate

Cuando el LRM recibe una solicitud de apagado y esta política está activa, se marca como no disponible para el HA Manager. Esto desencadena la migración de todos los servicios HA alojados en el nodo. El LRM intenta retrasar el apagado hasta que hayan sido movidos.

Este comportamiento presupone que los servicios pueden migrarse. Si están ligados al nodo, por ejemplo por hardware passthrough o por una regla estricta de afinidad que no deje destinos disponibles, el apagado puede quedar bloqueado hasta que intervenga un administrador.

Cuando el nodo vuelve a estar online, los servicios desplazados se trasladan de vuelta, salvo que hayan sido migrados manualmente mientras tanto.

NOTA: El watchdog sigue activo durante las migraciones de un apagado. Si el nodo pierde quorum será fenced y los servicios se recuperarán.

Si se inicia un servicio previamente detenido en un nodo que está en mantenimiento, el nodo debe ser fenced para garantizar que el servicio pueda moverse y arrancarse en otro nodo disponible.

Failover

Garantiza que todos los servicios se detengan, pero permite que sean recuperados si el nodo actual no vuelve online pronto. Puede ser útil en mantenimientos a escala de clúster, cuando migrar en vivo muchas VMs no es viable pero se desea que los servicios HA vuelvan a arrancar tan pronto como sea posible.

Freeze

Detiene y congela todos los servicios, evitando que sean recuperados hasta que el nodo vuelva a estar online.

Conditional

Conditional detecta automáticamente si se ha solicitado un apagado o un reinicio y modifica el comportamiento en consecuencia.

Shutdown - Un apagado suele indicar que el nodo permanecerá fuera de servicio un tiempo. El LRM detiene todos los servicios gestionados y otros nodos los recuperan posteriormente.

NOTA: El manual indica que los servidores modernos pueden disponer de mucha RAM. Para evitar migrar en vivo todo ese contenido, el comportamiento predeterminado detiene los recursos y los reinicia. Si se desea online migration debe ejecutarse manualmente antes del shutdown.

Reboot - El LRM informa al CRM de que desea reiniciarse y espera a que el CRM ponga todos los recursos en estado freeze. Tras el reinicio, esos recursos se arrancan de nuevo en el mismo nodo.

Movimiento manual - También es posible mover manualmente los recursos antes de apagar o reiniciar un nodo, lo que permite decidir explícitamente si se utiliza migración online o no.

NOTA: No deben finalizarse manualmente servicios como pve-ha-crm, pve-ha-lrm o watchdog-mux. Administran y utilizan el watchdog, por lo que hacerlo puede provocar un reinicio o reset inmediato del nodo.

15.12 Desarmar HA para mantenimiento del clúster

Algunas tareas de mantenimiento del clúster, como reconfigurar la red o el stack de comunicación de Corosync, pueden provocar pérdidas temporales de quorum o particiones de red. En condiciones normales, HA podría interpretarlas como fallos de nodo y ejecutar self-fencing.

El mecanismo disarm libera todos los watchdogs CRM y LRM de todo el clúster, permitiendo realizar ese mantenimiento sin que los nodos sean fenced.

IMPORTANTE: Mientras HA está disarmed, los servicios no están protegidos. Los fallos que se produzcan durante ese periodo no se recuperan automáticamente. Mantenga la ventana de desarme lo más corta posible.

Modos de recursos durante disarm

freeze - No se aplican nuevos comandos ni cambios de estado. Los servicios mantienen su estado actual, pero HA no reacciona a fallos ni procesa nuevas solicitudes. Es la opción más segura si se espera que todos los nodos sigan ejecutándose.

ignore - Los recursos quedan suspendidos del seguimiento HA y pueden administrarse como si no fueran HA. Permite iniciar, detener o migrar servicios manualmente. Al rearmar, el CRM vuelve a comprobar las ubicaciones y recoge las migraciones realizadas manualmente.

Desarmar y rearmar

```
ha-manager crm-command disarm-ha freeze
ha-manager crm-command disarm-ha ignore
ha-manager crm-command arm-ha
```

El estado puede monitorizarse con:

```
ha-manager status
```

La línea fencing muestra el estado actual del mecanismo y los estados de watchdog del CRM y los LRM.

Proceso de disarm

1. El CRM congela los servicios o suspende su seguimiento según el resource mode seleccionado.
2. Cada LRM termina sus workers activos y libera su agent lock y su watchdog.
3. Cuando todos los LRM online quedan inactivos, el CRM libera también su watchdog.

El CRM mantiene el manager lock durante todo el proceso, por lo que puede aceptar y procesar arm-ha para revertir el estado.

Si hay servicios en proceso de fencing, recovery o migration, el disarm se pospone hasta que dichas operaciones terminen para evitar estados incoherentes.

Nodos offline durante disarm

Si un nodo está offline al desarmar HA, su LRM no puede procesar la petición, pero no bloquea el paso global a disarmed. Si vuelve online mientras HA sigue desarmado, su LRM recoge ese estado y libera su watchdog sin intentar recuperar servicios.

Al rearmar HA, los servicios que estaban en el nodo offline se tratan según las reglas normales de recuperación: se aplica fencing y recuperación si el nodo sigue inaccesible, o se reinician allí si ya está online.

Interacción con Maintenance Mode

Si un nodo ya está en mantenimiento cuando se solicita disarm, las migraciones de mantenimiento continúan hasta que se hayan movido los servicios. Cuando no quedan servicios activos ni workers, el LRM libera su lock y watchdog como parte del desarme.

Al rearmar HA, el estado de mantenimiento se conserva. El nodo continúa en maintenance mode y los servicios no regresan hasta que se desactive explícitamente.

PRECAUCIÓN: Con HA disarmed no hay recuperación automática, failover ni fencing. Un fallo de nodo durante esta ventana no será detectado ni gestionado por HA. Rearme únicamente cuando el clúster esté de nuevo en estado saludable.

15.13 Cluster Resource Scheduling

El Cluster Resource Scheduler (CRS) selecciona nuevas ubicaciones de nodo para los recursos HA y se asegura de mantenerlos dentro de los nodos permitidos.

En cada ronda del HA Manager, el scheduler revisa todos los recursos HA para determinar si alguno necesita una nueva ubicación. Esto puede deberse, por ejemplo, a recuperar recursos de un nodo fenced o a la creación/modificación de reglas de afinidad.

15.13.1 Modo de planificación CRS

El modo CRS controla cómo se calcula la información de uso del clúster que servirá de base para las decisiones de scheduling. Puede cambiarse en Datacenter -> Options -> Cluster Resource Scheduling. El modo predeterminado es basic. El cambio se aplica en la siguiente ronda del HA Manager, normalmente en no más de 10 segundos.

Basic Scheduler - Utiliza el número de guests activos en cada nodo para elegir el nodo más adecuado.

Static-Load Scheduler - Utiliza información estática de uso de los guests activos, incluidas las cuotas de CPU y memoria configuradas.

Dynamic-Load Scheduler - Utiliza información dinámica de uso: consumo medio de CPU y memoria, además de las cuotas configuradas.

15.13.2 Puntos de scheduling CRS

En cada ronda, CRS comprueba si algún recurso HA necesita una nueva ubicación. Los puntos de scheduling descritos por el manual son:

Recuperación de recurso HA (siempre activa) - Cuando falla un nodo con recursos HA activos, CRS distribuye la recuperación entre los nodos restantes.

Cambios en reglas HA (siempre activa) - Cuando una regla cambia las restricciones, CRS busca nuevas ubicaciones según el tipo de regla.

- Node affinity: al crear una regla o añadir recursos/nodos, CRS garantiza que los recursos respeten nodos y prioridades.
- Positive resource affinity: al crear una regla o añadir recursos, CRS mueve los recursos afectados a un nodo común.
- Negative resource affinity: al crear una regla o añadir recursos, CRS mueve los recursos afectados a nodos separados.

Rebalance al arrancar recurso HA (opt-in) - Si un recurso tiene habilitado rebalance on start, CRS selecciona el nodo considerado más adecuado y migra el recurso si no coincide con el nodo actual.

En modo basic, se elige el nodo con menos guests activos. En modos static-load y dynamic-load, cada nodo se evalúa como si el recurso ya estuviera ejecutándose en él, utilizando CPU y memoria de la configuración del guest. Se consideran el uso máximo y medio de todos los nodos, dando mucho más peso a la memoria porque es un recurso realmente limitado.

La opción puede habilitarse mediante ha-rebalance-on-start en Datacenter -> Options -> Cluster Resource Scheduling.

IMPORTANTE: En static-load y dynamic-load, rebalance on start está todavía en technology preview. Cuantos más recursos HA haya, más combinaciones deben evaluarse; el manual no recomienda esta función con miles de recursos HA.

15.13.3 Balanceador de carga CRS

La utilización de los nodos puede desequilibrarse con el tiempo: un nodo puede sufrir contención mientras otro permanece poco utilizado. El balanceador automático del HA Manager puede reducir ese desequilibrio emitiendo migraciones de rebalanceo.

Actualmente estas migraciones solo se emiten para recursos HA, se ejecutan secuencialmente y siempre respetan las reglas de afinidad vigentes.

Deshabilitar auto-rebalance en un recurso HA evita que dicho recurso participe en migraciones de rebalanceo automáticas.

NOTA: Cualquier recurso HA que tenga afinidad positiva con otro recurso cuyo auto-rebalance esté deshabilitado tampoco participará en migraciones de rebalanceo.

El balanceo automático se configura en Datacenter -> Options -> Cluster Resource Scheduling y requiere utilizar static-load o dynamic-load.

En cada ronda del HA Manager - aproximadamente cada 10 segundos - se comprueba si el desequilibrio del clúster supera el imbalance threshold. Si permanece por encima durante al menos tantas rondas consecutivas como indique hold duration, el balanceador selecciona la migración de un recurso HA que reduzca más el desequilibrio global.

El algoritmo de selección se controla mediante rebalancing method. La migración solo se ejecuta si mejora el desequilibrio al menos en el porcentaje definido por imbalance improvement.

El desequilibrio se deriva de la desviación estándar y la media de las cargas de los nodos y se normaliza a un rango 0-100 %. imbalance threshold e imbalance improvement se expresan en porcentaje.

- imbalance threshold: valor predeterminado 30 %. Controla la sensibilidad. Con 0, se intentará seleccionar una migración cada vez que se supere hold duration.
- imbalance improvement: valor predeterminado 10 %. Define qué mejora mínima se considera aceptable. Con 0, se acepta una migración incluso si el desequilibrio esperado no cambia. Nunca puede ser negativo, por lo que el balanceador no ejecutará una migración que empeore el desequilibrio previsto.

NOTA: Está previsto que en el futuro el balanceo de carga incluya también recursos no HA.

Fin del Capítulo 15 · Alta disponibilidad

Con esta Parte 15B queda completada la traducción del Capítulo 15. La siguiente entrega comienza el Capítulo 16 · Backup and Restore.

CRITERIO DE TRADUCCIÓN: Se han preservado nombres de estados, propiedades, comandos, rutas y opciones que deben utilizarse literalmente. El contenido explicativo se ha traducido al castellano manteniendo la organización, advertencias y nivel de detalle del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 16

Backup y restauración

Modos · fleecing · PBS · retención · protección · restore · live-restore · configuración · hooks · exclusiones

Consistencia	Rendimiento	Retención	Recuperación
stop · suspend · snapshot	fleecing · bwlimit · workers	prune-backups · protected	restore · live-restore · archivos

Basado en Proxmox VE Administration Guide, Release 9.2.4

Capítulo 16 completo · Traducción progresiva al castellano

Capítulo 16 · Backup y restauración

Las copias de seguridad son un requisito para cualquier despliegue IT sensato. Proxmox VE proporciona una solución integrada que aprovecha las capacidades de cada almacenamiento y de cada tipo de sistema invitado. Mediante la opción mode, el administrador puede ajustar el equilibrio entre consistencia de la copia y tiempo de indisponibilidad del invitado.

Las copias de Proxmox VE son siempre copias completas: contienen tanto la configuración de la VM o CT como todos sus datos. Pueden iniciarse desde la interfaz gráfica o mediante la herramienta de línea de comandos vzdump.

Almacenamiento de backup

Antes de ejecutar una copia debe existir un almacenamiento configurado para backups. Puede ser un almacenamiento Proxmox Backup Server, donde los backups se guardan como chunks deduplicados y metadatos, o un almacenamiento a nivel de archivos, donde se guardan como ficheros normales.

El manual recomienda utilizar Proxmox Backup Server en un host dedicado por sus funciones avanzadas. Un servidor NFS es una alternativa válida. En ambos casos, posteriormente pueden archivarse las copias en cinta para conservarlas fuera del sitio.

Backup programado

Los trabajos de backup pueden programarse para ejecutarse automáticamente en determinados días y horas, sobre nodos e invitados seleccionados. La sección 16.5 detalla su configuración.

16.1 Modos de backup

La opción mode determina cómo se busca consistencia durante la copia. El comportamiento depende de si el invitado es una máquina virtual o un contenedor.

Modos de backup para máquinas virtuales

stop mode - Proporciona la máxima consistencia a costa de una breve interrupción. Proxmox VE apaga ordenadamente la VM y después ejecuta un proceso QEMU en segundo plano que copia los datos. Una vez iniciado el backup, la VM vuelve a operar con normalidad si estaba encendida. La consistencia se garantiza usando la función de live backup.

suspend mode - Se mantiene por compatibilidad. Suspende la VM antes de invocar snapshot mode. Como provoca más tiempo de parada y no garantiza una mejora de consistencia, el manual recomienda utilizar snapshot mode.

snapshot mode - Reduce al mínimo la interrupción. Proxmox VE realiza un live backup copiando bloques mientras la VM sigue funcionando. Si QEMU Guest Agent está habilitado (agent: 1) y en ejecución, se invocan guest-fsfreeze-freeze y guest-fsfreeze-thaw para mejorar la consistencia.

NOTA: En invitados Windows, si se utiliza otro software de backup dentro del propio sistema invitado, es necesario configurar QEMU Guest Agent. El manual remite al apartado Freeze & Thaw del Guest Agent.

NOTA: El live backup de Proxmox VE ofrece semántica similar a snapshot sobre cualquier tipo de almacenamiento; no necesita que el backend soporte snapshots. Una VM apagada puede aparecer como running durante un breve periodo mientras QEMU lee sus discos, pero la VM no se arranca.

Modos de backup para contenedores

stop mode - Detiene el contenedor durante toda la copia. Puede producir una interrupción prolongada.

suspend mode - Usa rsync para copiar los datos del contenedor a una ubicación temporal (--tmpdir). Después suspende el contenedor y ejecuta un segundo rsync con los cambios. Finalmente lo reanuda. Reduce la interrupción, pero necesita espacio adicional para la copia temporal.

Si el contenedor reside en un sistema de archivos local y el destino del backup es NFS/CIFS, el manual recomienda que --tmpdir también esté en un sistema de archivos local para obtener una mejora de rendimiento muy notable. Además, un tmpdir local es necesario para respaldar con ACL un contenedor local en modo suspend cuando el almacenamiento de backup es NFS.

snapshot mode - Utiliza las capacidades de snapshot del almacenamiento subyacente. Primero suspende el contenedor para asegurar consistencia, crea un snapshot temporal de sus volúmenes, archiva su contenido en un fichero tar y elimina después el snapshot temporal.

NOTA: snapshot mode requiere que todos los volúmenes incluidos estén sobre almacenamientos con soporte de snapshots. Puede excluirse un volumen mediante backup=no.

NOTA: Por defecto, los mount points adicionales al Root Disk no se incluyen. En volume mount points puede activarse Backup para incluirlos. Device mounts y bind mounts nunca se respaldan porque su contenido se administra fuera de la biblioteca de almacenamiento de Proxmox VE.

16.1.1 VM Backup Fleecing

Al iniciar el backup de una VM, QEMU instala un filtro copy-before-write en su capa de bloques. Cuando el invitado escribe sobre un sector todavía necesario para el backup, los datos antiguos se envían primero al destino. La escritura del invitado queda bloqueada hasta terminar esa operación, de modo que el I/O hacia sectores aún no respaldados queda limitado por la velocidad del destino de backup.

Con backup fleecing, esos datos antiguos se almacenan temporalmente en una imagen de fleecing en lugar de enviarse directamente al destino. Esto puede mejorar el rendimiento de I/O del invitado e incluso evitar bloqueos en determinados escenarios, a costa de necesitar espacio de almacenamiento adicional.

Ejemplo para VM 123 utilizando local-lvm como almacenamiento de fleecing:

```
vzdump 123 --fleecing enabled=1,storage=local-lvm
```

La opción puede definirse en un job de backup o como valor de respaldo a nivel de nodo. En la GUI se encuentra en la pestaña Advanced del trabajo de backup.

El almacenamiento de fleecing debería ser local y rápido, con thin provisioning y soporte discard. El manual cita LVM-thin, RBD, ZFS con sparse 1 y muchos almacenamientos basados en archivos. Idealmente debería ser un almacenamiento dedicado, para que agotarlo no afecte a otros invitados y provoque únicamente el fallo del backup.

Las partes de la imagen de fleecing que ya han sido respaldadas se descartan para reducir el uso de espacio. En almacenamientos basados en archivos que no soportan discard - por ejemplo NFS anterior a 4.2 - se recomienda preallocation off. Combinado con qcow2, permite reutilizar zonas ya asignadas de la imagen y ahorrar espacio.

ADVERTENCIA: En un almacenamiento no thin-provisioned, como LVM o ZFS sin sparse, debe reservarse por adelantado el tamaño completo del disco original para la imagen de fleecing. En un almacenamiento thin, la imagen solo puede crecer hasta el tamaño del disco original si el invitado reescribe todo un disco mientras el backup está ocupado con otro.

16.1.2 Modo de detección de cambios en CT

El modo de detección de cambios determina el formato de codificación de los archivos pxar y cómo se tratan los ficheros modificados o sin cambios en backups de contenedores cuyo destino es Proxmox Backup Server.

Puede configurarse por job en la pestaña Advanced o como fallback a nivel de nodo. Hay tres modos:

Modo	Descripción
Default	Lee y codifica todos los archivos en un único archivo usando pxar versión 1.
Data	Lee y codifica todos los archivos, separando datos y metadatos en streams distintos con pxar versión 2.
Metadata	Usa streams separados y pxar v2 como Data, pero toma el archivo de metadatos del snapshot anterior, si existe, para detectar archivos sin cambios y reutilizar sus chunks sin leer sus contenidos del disco cuando sea posible.

Ejemplo usando metadata:

```
vzdump 123 --storage pbs-storage --pbs-change-detection-mode metadata
```

NOTA: Los backups de máquinas virtuales y los backups enviados a backends distintos de Proxmox Backup Server no se ven afectados por esta opción.

16.2 Nombres de los archivos de backup

Las versiones recientes de vzdump codifican en el nombre del archivo el tipo de invitado y la hora del backup. Por ejemplo:

```
vzdump-lxc-105-2009_10_09-11_04_43.tar
```

Esto permite almacenar varias copias en un mismo directorio. El número de copias conservadas puede limitarse mediante las opciones de retención descritas más adelante.

16.3 Compresión de los backups

Los archivos de backup pueden comprimirse con lzo, gzip o zstd. El manual indica que Zstandard (zstd) es actualmente el más rápido de los tres y además ofrece multihilo. lzo y gzip están más extendidos y suelen estar instalados por defecto.

Puede instalarse pigz como reemplazo compatible de gzip para mejorar rendimiento mediante multihilo. Tanto con pigz como con zstd puede configurarse el número de hilos o núcleos utilizados.

Extensión	Compresión
.zst	Zstandard (zstd)
.gz o .tgz	gzip
.lzo	lzo

Si el nombre no termina con ninguna de esas extensiones, vzdump no comprimió el backup.

16.4 Cifrado de backups

En almacenamientos Proxmox Backup Server puede configurarse de forma opcional cifrado del lado del cliente. El manual remite a la sección correspondiente de PBS para los detalles.

16.5 Trabajos de backup

Además de los backups manuales, pueden crearse trabajos periódicos que respalden todos los invitados o una selección hacia un almacenamiento concreto. Se administran en Datacenter -> Backup o mediante el endpoint /cluster/backup de la API.

Ambos métodos generan entradas en /etc/pve/jobs.cfg, que son interpretadas y ejecutadas por el daemon pvescheduler.

Un job puede aplicarse a todos los nodos del clúster o a un nodo específico y se ejecuta según una planificación. El formato de Schedule es muy similar a los eventos de calendario de systemd. En la GUI el campo es editable y su lista desplegable ofrece varios ejemplos como punto de partida.

Los jobs pueden definir opciones de retención propias, que sustituyen las configuradas en el almacenamiento o en el nodo, así como una plantilla de notas para guardar información adicional con el backup.

Si un backup programado se pierde porque el host estaba apagado o pvescheduler estaba deshabilitado en la hora prevista, puede habilitarse Repeat missed - repeat-missed en la configuración - para ejecutar el job perdido tan pronto como sea posible.

Existen varias opciones de rendimiento. La más destacada es bwlimit, que limita el ancho de banda de I/O. El número de hilos del compresor se controla con pigz o zstd. También existen ionice, max-workers para backups de VM y pbs-entries-max para backups de contenedores.

16.6 Retención de backups

La opción prune-backups permite definir de manera flexible qué copias se conservan.

keep-all <boolean> - Conserva todos los backups. Si es true no pueden definirse las demás opciones.

keep-last <N> - Conserva los últimos N backups.

keep-hourly <N> - Conserva backups de las últimas N horas. Si hay más de uno en una hora, solo se mantiene el más reciente.

keep-daily <N> - Conserva backups de los últimos N días, manteniendo el más reciente de cada día.

keep-weekly <N> - Conserva backups de las últimas N semanas, manteniendo el más reciente de cada semana.

keep-monthly <N> - Conserva backups de los últimos N meses, manteniendo el más reciente de cada mes.

keep-yearly <N> - Conserva backups de los últimos N años, manteniendo el más reciente de cada año.

NOTA: Las semanas comienzan el lunes y terminan el domingo. Proxmox VE utiliza el sistema ISO week date y gestiona correctamente las semanas al final del año.

Las opciones se procesan en el orden anterior. Cada opción solo considera los backups dentro de su periodo; las opciones siguientes no vuelven a tratar los backups ya cubiertos.

Se especifican como una lista separada por comas:

```
vzdump 777 --prune-backups keep-last=3,keep-daily=13,keep-yearly=9
```

Aunque prune-backups puede pasarse directamente a vzdump, normalmente resulta más práctico definirlo en el almacenamiento desde la interfaz web.

16.6.1 Prune Simulator

La documentación de Proxmox Backup Server incluye un simulador de pruning que permite explorar el efecto de distintas opciones de retención combinadas con diferentes calendarios de backup.

16.6.2 Ejemplo de retención

La frecuencia y la conservación dependen de cuánto cambian los datos y de la importancia de poder volver a estados antiguos. Si los backups actúan como archivo documental de empresa, pueden existir además requisitos legales de conservación.

El ejemplo del manual supone backups diarios, una retención total de diez años y un espaciado creciente entre copias antiguas:

- keep-last=3: conserva las tres últimas copias, incluyendo posibles backups manuales alrededor de una actualización importante.
- keep-hourly no se define porque con backups diarios no aporta valor; los backups manuales extra ya quedan cubiertos por keep-last.
- keep-daily=13: junto con keep-last garantiza al menos dos semanas de copias.
- keep-weekly=8: garantiza al menos dos meses completos de backups semanales.
- keep-monthly=11: junto con los anteriores garantiza al menos un año de backups mensuales.
- keep-yearly=9: aporta el archivo a largo plazo. El año actual ya queda cubierto por las opciones anteriores, por lo que 9 años adicionales proporcionan al menos 10 años en total.

RECOMENDACIÓN DEL MANUAL: Conviene usar un periodo de retención superior al mínimo exigido. Siempre puede reducirse después, pero una copia eliminada no puede recrearse.

16.7 Protección de backups

Un backup puede marcarse como protected para impedir que sea eliminado desde la interfaz, la CLI o la API de Proxmox VE. Esta protección la aplica Proxmox VE, no el sistema de archivos: quien tenga acceso de escritura al almacenamiento subyacente todavía puede borrar manualmente el fichero.

NOTA: Los backups protegidos se ignoran durante el pruning y no cuentan para los límites de retención.

En almacenamientos basados en sistema de archivos la protección se implementa mediante un archivo centinela <backup-name>.protected. En Proxmox Backup Server se gestiona en el servidor, disponible desde PBS 2.1.

La opción max-protected-backups controla cuántos backups protegidos por invitado se permiten en el almacenamiento. El valor -1 significa ilimitados. El valor por defecto es ilimitado para usuarios con privilegio Datastore.Allocate y 5 para el resto.

16.8 Notas de backup

Las copias pueden recibir notas mediante Edit Notes en la GUI o utilizando la API de contenido del almacenamiento.

También puede definirse una plantilla dinámica para trabajos programados y copias manuales. Las variables soportadas en esta versión son:

- {{cluster}}: nombre del clúster, si existe.
- {{guestname}}: nombre asignado al invitado.
- {{node}}: nombre del host donde se crea la copia.
- {{vmid}}: VMID numérico del invitado.

Cuando se especifica desde API o CLI debe escribirse en una sola línea; los saltos de línea y las barras invertidas se escapan como \n y \\ respectivamente.

16.9 Restauración

Un archivo de backup puede restaurarse desde la GUI de Proxmox VE o mediante estas herramientas:

pct restore - Utilidad de restauración de contenedores.

qmrestore - Utilidad de restauración de máquinas virtuales.

16.9.1 Límite de ancho de banda

Restaurar backups grandes puede consumir gran cantidad de recursos, especialmente ancho de banda de almacenamiento tanto al leer el backup como al escribir en el destino. Esto puede perjudicar a otros invitados si el almacenamiento se congestiona.

Proxmox VE implementa dos clases de límites:

- Límite por restauración: ancho de banda máximo para leer un archivo de backup.
- Límite de escritura por almacenamiento: ancho de banda máximo utilizado al escribir sobre un almacenamiento concreto.

El límite de lectura influye indirectamente en el de escritura. Un límite por trabajo menor sustituye a un límite por almacen amiento mayor. Un límite por trabajo mayor solo puede superar el límite por almacenamiento si el usuario dispone de permisos Data.Allocate en el almacenamiento afectado.

En las herramientas CLI de restore puede utilizarse --bwlimit <integer>. La unidad es KiB/s; por ejemplo 10240 limita la lectura del backup a 10 MiB/s.

NOTA: El valor 0 para bwlimit desactiva los límites en una restauración concreta. Esto puede ser útil para recuperar lo antes posible un invitado crítico, pero requiere permisos Data.Allocate sobre el almacenamiento.

Puede configurarse un límite predeterminado por almacenamiento:

```
pvesm set STORAGEID --bwlimit restore=KIBs
```

16.9.2 Live-Restore

Restaurar una copia grande puede tardar mucho tiempo durante el cual la VM seguiría indisponible. Para backups de VMs almacenados en Proxmox Backup Server, la opción live-restore permite reducir esa espera.

Al activarla desde la GUI o mediante --live-restore en qmrestore, la VM se inicia en cuanto comienza la restauración. Los datos se copian en segundo plano priorizando los chunks a los que la VM accede activamente.

ADVERTENCIA: Durante live-restore la velocidad de lectura de disco de la VM estará limitada porque los datos deben cargarse desde el servidor de backup. Una vez cargados quedan disponibles en el almacenamiento de destino. Las escrituras apenas se ven afectadas.

ADVERTENCIA: Si live-restore falla, la VM queda en un estado indefinido: es posible que no se haya copiado todo el contenido y probablemente no pueda conservarse lo escrito durante la restauración fallida.

Este modo resulta especialmente útil en VMs grandes que solo necesitan una pequeña parte de sus datos para volver a prestar servicio, por ejemplo servidores web: el sistema operativo y los servicios pueden iniciar mientras la tarea en segundo plano continúa copiando datos poco utilizados.

16.9.3 Restauración de archivos individuales

El botón File Restore de la pestaña Backups del almacenamiento abre un explorador directamente sobre los datos incluidos en u na copia. Esta función solo está disponible para backups almacenados en Proxmox Backup Server.

En contenedores, el primer nivel del árbol muestra los archivos pxar incluidos. En VMs, muestra las imágenes de disco, que pu eden abrirse para ver las tecnologías de almacenamiento detectadas. En el caso más simple aparece part, que representa la tabla de particiones, seguida por sus particiones.

No todo el contenido de una VM tiene por qué ser accesible: algunos sistemas de archivos o tecnologías de almacenamiento del invitado pueden no estar soportados.

Los archivos y directorios pueden descargarse con Download. Los directorios se comprimen al vuelo en un archivo ZIP.

Para acceder de forma segura a imágenes de VM potencialmente no confiables, Proxmox VE inicia una VM temporal, no visible como invitado. Esta VM se detiene automáticamente después de un timeout. Esto evita exponer directamente el hipervisor, aunqu e no significa que los datos descargados del backup sean necesariamente seguros.

NOTA: Cada VM temporal de file restore genera un log en /var/log/proxmox-backup/file-restore/. Puede contener información adicional si falla el acceso a sistemas de archivos o la restauración de ficheros individuales.

16.10 Configuración

La configuración global se guarda en /etc/vzdump.conf. Utiliza un formato sencillo clave/valor separado por dos puntos:

```
OPTION: value
```

Las líneas vacías se ignoran y las que comienzan por # son comentarios. Los valores del archivo actúan como predeterminados y pueden sobrescribirse desde la línea de comandos.

bwlimit: <integer> (0 - N) (default = 0) - Limita el ancho de banda de I/O en KiB/s.

compress: <0 | 1 | gzip | lzo | zstd> (default = 0) - Comprime el archivo de backup.

dumpdir: <string> - Guarda los archivos resultantes en el directorio indicado.

exclude-path: <array> - Excluye archivos o directorios mediante glob de shell. Las rutas que comienzan por / se anclan a la raíz del contenedor; el resto se compara de forma relativa en cada subdirectorio.

fleecing: [[enabled=]<1|0>] [,storage=<storage ID>] - Opciones de fleecing para backups de VM.

ionice: <integer> (0 - 8) (default = 7) - Prioridad de I/O cuando se usa el scheduler BFQ. En snapshot y suspend de VMs afecta únicamente al compresor. 8 usa prioridad idle; otros valores usan best-effort con la prioridad indicada.

lockwait: <integer> (0 - N) (default = 180) - Tiempo máximo de espera del lock global, en minutos.

mailnotification: <always | failure> (default = always) - Obsoleto; el manual recomienda notification targets/matchers.

mailto: <string> - Obsoleto; lista separada por comas de direcciones o usuarios que recibirían notificaciones por correo.

mode: <snapshot | stop | suspend> (default = snapshot) - Modo de backup.

notes-template: <string> - Plantilla para notas. Admite {{cluster}}, {{guestname}}, {{node}} y {{vmid}}. Debe ser una sola línea; se escapan newline y backslash.

notification-mode: <auto | legacy-sendmail | notification-system> (default = auto) - Selecciona el sistema de notificación. legacy-sendmail usa mailto/mailnotification; notification-system usa el sistema de notificaciones de PVE; auto usa correo si mailto está configurado y el nuevo sistema en caso contrario.

pbs-change-detection-mode: <data | legacy | metadata> - Modo PBS de detección de cambios y formato de codificación para backups de contenedores.

performance: [max-workers=<integer>] [,pbs-entries-max=<integer>] - Ajustes de rendimiento adicionales.

pigz: <integer> (default = 0) - Usa pigz en lugar de gzip cuando N>0. N=1 usa la mitad de los cores; N>1 usa exactamente N hilos.

pool: <string> - Respalda todos los invitados conocidos del pool indicado.

protected: <boolean> - Marca las copias como protegidas.

prune-backups: ... (default = keep-all=1) - Sustituye las opciones de retención del almacenamiento por las definidas aquí.

remove: <boolean> (default = 1) - Elimina backups antiguos según prune-backups.

script: <string> - Ejecuta el hook script indicado.

stdexcludes: <boolean> (default = 1) - Excluye archivos temporales y logs estándar.

stopwait: <integer> (0 - N) (default = 10) - Tiempo máximo de espera hasta que se detenga un invitado, en minutos.

storage: <storage ID> - Almacenamiento donde se guarda el resultado.

tmpdir: <string> - Directorio para archivos temporales.

zstd: <integer> (default = 1) - Número de hilos zstd. N=0 usa la mitad de los cores disponibles; N>0 usa exactamente N hilos.

Opciones internas de fleecing

enabled=<boolean> (default = 0) - Activa backup fleecing. Los datos antiguos necesarios para el backup se almacenan en el almacenamiento indicado en lugar de copiarse directamente al destino.

storage=<storage ID> - Almacenamiento de las imágenes de fleecing. Para ahorrar espacio, el manual recomienda un almacenamiento local con discard y thin provisioning o archivos sparse.

Opciones de performance

max-workers=<integer> (1 - 256) (default = 16) - Solo para VMs. Permite hasta ese número de workers de I/O simultáneos.

pbs-entries-max=<integer> (1 - N) (default = 1048576) - Solo para backups de contenedores hacia PBS. Limita el número de entradas mantenidas en memoria para evitar OOM; puede aumentarse para contenedores con gran cantidad de archivos.

Ejemplo de /etc/vzdump.conf

```
tmpdir: /mnt/fast_local_disk
storage: my_backup_storage
mode: snapshot
bwlimit: 10000
```

16.11 Hook Scripts

Puede indicarse un script hook mediante --script. Se invoca en distintas fases del proceso de backup con los parámetros correspondientes. El manual señala un ejemplo en el directorio de documentación:

```
vzdump-hook-script.pl
```

16.12 Exclusiones de archivos

NOTA: Esta opción solo está disponible para backups de contenedores.

vzdump omite por defecto estos archivos y patrones; puede desactivarse con --stdexcludes 0:

```
/tmp/?*
/var/tmp/?*
/var/run/?*pid
```

También pueden añadirse exclusiones manuales. Por ejemplo:

```
vzdump 777 --exclude-path /tmp/ --exclude-path '/var/foo*'
```

El ejemplo excluye /tmp/ y cualquier archivo o directorio cuyo nombre empiece por /var/foo, como /var/foo o /var/foobar.

ADVERTENCIA: Para backups hacia PBS y backups en modo suspend, los patrones terminados en / coinciden con directorios, pero no con archivos. En cambio, en snapshot y stop no-PBS, los patrones terminados en / no coinciden actualmente porque tar no lo soporta.

Las rutas que no comienzan por / no se anclan a la raíz del contenedor y se comparan relativamente desde cualquier subdirecto rio. Ejemplo:

```
vzdump 777 --exclude-path bar
```

Esto excluye /bar, /var/bar, /var/foo/bar y equivalentes, pero no /bar2.

Los archivos de configuración también se almacenan dentro del backup, en ./etc/vzdump/, y se restauran correctamente.

16.13 Ejemplos

Backup simple del invitado 777, sin snapshot, archivando el área privada y la configuración en el directorio de dump predeterminado:

```
vzdump 777
```

Usar rsync con suspend/resume para minimizar la interrupción:

```
vzdump 777 --mode suspend
```

Respaldar todos los invitados y enviar correos a root y admin. Como mailto está definido y notification-mode es auto, se utiliza sendmail:

```
vzdump --all --mode suspend --mailto root --mailto admin
```

Snapshot mode sin parada y con directorio de dump no predeterminado:

```
vzdump 777 --dumpdir /mnt/backup --mode snapshot
```

Respaldar varios invitados seleccionados:

```
vzdump 101 102 103 --mailto root
```

Respaldar todos los invitados excepto 101 y 102:

```
vzdump --mode suspend --exclude 101,102
```

Restaurar un contenedor como nuevo CT 600:

```
pct restore 600 /mnt/backup/vzdump-lxc-777.tar
```

Restaurar una VM QemuServer como VM 601:

```
qmrestore /mnt/backup/vzdump-qemu-888.vma 601
```

Clonar el contenedor 101 a un contenedor 300 con rootfs de 4 GB utilizando pipes:

```
vzdump 101 --stdout | pct restore --rootfs 4 300 -
```

Fin del Capítulo 16 · Backup y restauración

Con esta Parte 16 queda completada la traducción del Capítulo 16. La siguiente entrega comienza el Capítulo 17 · Notificacion es.

CRITERIO DE TRADUCCIÓN: Se han mantenido literalmente los comandos, rutas, opciones, nombres de herramientas y valores de configuración que deben utilizarse tal como aparecen en Proxmox VE. El contenido explicativo se ha traducido al castellano preservando la organización y nivel de detalle del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 17

Notificaciones

Sendmail · SMTP · Gotify · Webhook · Matchers · Eventos · Permisos · Plantillas

Destinos	Enrutado	Eventos	Personalización
Sendmail · SMTP · Gotify · Webhook	Notification Matchers	Severidad + metadatos	Plantillas Handlebars

Basado en Proxmox VE Administration Guide, Release 9.2.4

Traducción progresiva al castellano. Se conservan comandos, rutas, nombres de opciones y sintaxis técnica.

Capítulo 17 · Notificaciones

17.1 Visión general

Proxmox VE genera eventos de notificación ante fallos de replicación de almacenamiento, fencing de nodos, finalización o fallo de backups y otros sucesos. Estos eventos se procesan según la configuración global de notificaciones.

Cada evento incluye metadatos como marca de tiempo, nivel de severidad, tipo y otros campos específicos.

- Notification Events: representan los sucesos que genera Proxmox VE.
- Notification Matchers: encaminan un evento hacia uno o varios destinos y pueden aplicar reglas según sus metadatos.
- Notification Targets: son los destinos finales. El manual incluye Sendmail, SMTP, Gotify y Webhook.

La configuración global se administra desde Datacenter -> Notifications.

Los archivos de configuración son:

```
/etc/pve/notifications.cfg
/etc/pve/priv/notifications.cfg
```

El segundo archivo contiene opciones sensibles, como contraseñas y tokens de autenticación, y solo puede ser leído por root.

17.2 Destinos de notificación

17.2.1 Sendmail

El destino Sendmail utiliza el binario sendmail para enviar correo a una lista de usuarios o direcciones configuradas. Si se selecciona un usuario, se usa la dirección definida en sus ajustes. Para root@pam se utiliza la dirección introducida durante la instalación.

NOTA: En instalaciones estándar, sendmail lo proporciona Postfix. Puede ser necesario configurar Postfix, por ejemplo con un relay externo o smart host. Si falla la entrega, consulte los logs de Postfix.

- mailto: dirección de correo de destino; puede repetirse.
- mailto-user: usuario destinatario; su correo se consulta en users.cfg; puede repetirse.
- author: autor del mensaje; por defecto Proxmox VE.
- from-address: dirección From. Si no se define, se usa email_from de datacenter.cfg; si tampoco existe, root@$hostname.
- comment: comentario asociado al destino.

Ejemplo en /etc/pve/notifications.cfg:

```
sendmail: example
mailto-user root@pam
mailto-user admin@pve
mailto max@example.com
from-address pve1@example.com
comment Send to multiple users/addresses
```

17.2.2 SMTP

Los destinos SMTP envían correo directamente a un relay SMTP y no utilizan el MTA del sistema. Si se selecciona un usuario, se utiliza su dirección configurada.

NOTA: A diferencia de Sendmail, los destinos SMTP no disponen de un mecanismo de cola o reintento si falla la entrega.

- mailto: dirección de destino; puede repetirse.
- mailto-user: usuario destinatario; puede repetirse.
- author: autor del correo; por defecto Proxmox VE.
- from-address: dirección From; algunos relays exigen que pertenezca al usuario autenticado para evitar spoofing.
- username: usuario de autenticación. Si se omite, no se autentica. Se admiten PLAIN y LOGIN.
- password: contraseña para autenticación.
- mode: cifrado insecure, starttls o tls; por defecto tls.
- server: dirección o IP del relay SMTP.
- port: 25 para insecure, 465 para tls o 587 para starttls si no se indica otro.
- comment: comentario del destino.

Ejemplo público en /etc/pve/notifications.cfg:

```
smtp: example
mailto-user root@pam
mailto-user admin@pve
mailto max@example.com
from-address pve1@example.com
username pve1
server mail.example.com
mode starttls
```

La contraseña se almacena en /etc/pve/priv/notifications.cfg:

```
smtp: example
password somepassword
```

17.2.3 Gotify

Gotify es un servidor open source y autoalojado de notificaciones push. Ofrece una API y una interfaz web para integrar distintos dispositivos y servicios.

- server: URL base del servidor Gotify, por ejemplo http://<ip>:8888.
- token: token de autenticación generado en la interfaz de Gotify.
- comment: comentario del destino.

NOTA: El plugin Gotify respeta la configuración de proxy HTTP definida a nivel de Datacenter.

```
gotify: example
server http://gotify.example.com:8888
comment Send to multiple users/addresses
```

El token secreto se guarda en /etc/pve/priv/notifications.cfg:

```
gotify: example
token somesecrettoken
```

17.2.4 Webhook

Los destinos Webhook realizan peticiones HTTP a una URL configurable.

- url: URL de destino; admite plantillas con contenido, metadatos y secretos.
- method: POST, PUT o GET.
- header: cabeceras HTTP; admiten plantillas.
- body: cuerpo HTTP; admite plantillas.
- secret: pares clave/valor secretos almacenados en el archivo protegido y accesibles mediante el namespace secrets.
- comment: comentario del destino.

Las plantillas usan Handlebars. Propiedades disponibles:

- {{ title }}: título renderizado.
- {{ message }}: cuerpo renderizado.
- {{ severity }}: info, notice, warning, error o unknown.
- {{ timestamp }}: UNIX epoch en segundos.
- {{ fields.<name> }}: metadatos del evento, por ejemplo fields.type.

- {{ secrets.<name> }}: secretos, por ejemplo secrets.token.

Helpers disponibles:

- {{ url-encode <value/property> }}: codificación URL.
- {{ escape <value/property> }}: escape seguro de caracteres de control para JSON.
- {{ json <value/property> }}: renderiza un valor o namespace como JSON.

Ejemplos de Webhook

ntfy.sh

```
Method: POST
URL: https://ntfy.sh/{{ secrets.channel }}
Header:
  Markdown: Yes
Body:
  {{ message }}
Secret:
  channel: <your ntfy.sh channel>
```

Discord

```
Method: POST
URL: https://discord.com/api/webhooks/{{ secrets.token }}
Header:
  Content-Type: application/json
Body:
{
  "content": "``` {{ escape message }}```"
}
Secret:
  token: <token>
```

Slack

```
Method: POST
URL: https://hooks.slack.com/services/{{ secrets.token }}
Header:
  Content-Type: application/json
Body:
{
  "text": "``` {{escape message}}```",
  "type": "mrkdwn"
}
Secret:
  token: <token>
```

17.3 Notification Matchers

Los Notification Matchers encaminan notificaciones hacia targets según reglas de coincidencia. Pueden evaluar la marca de tiempo mediante match-calendar, la severidad con match-severity o campos de metadatos mediante match-field.

Puede crearse cualquier número de matchers. Cada target se notifica como máximo una vez por evento, aunque aparezca en varios matchers. Un matcher sin reglas siempre coincide:

```
matcher: always-matches
target admin
comment This matcher always matches
```

17.3.1 Opciones de los Matchers

- target: target notificado si el matcher coincide; puede repetirse.
- invert-match: invierte el resultado final.
- mode: all exige que coincidan todas las reglas; any exige al menos una. Por defecto all.
- match-calendar: compara el timestamp con un calendario.
- match-field: compara metadatos.
- match-severity: compara severidad.
- comment: comentario del matcher.

17.3.2 Reglas de calendario

```
match-calendar 8-12
match-calendar 8:00-15:30
match-calendar mon..fri 9:00-17:00
match-calendar sun,tue..wed,fri 9-17
```

17.3.3 Reglas por campos

En modo exact puede utilizarse una coma como separador. La regla coincide si el campo contiene cualquiera de los valores indicados.

```
match-field exact:type=vzdump
match-field exact:type=replication,fencing
match-field regex:hostname=^.+\.example\.com$
```

Si el campo de metadatos indicado no existe, la notificación no coincide.

17.3.4 Reglas por severidad

```
match-severity error
match-severity warning,error
```

Las severidades utilizadas son info, notice, warning, error y unknown.

17.3.5 Ejemplos

```
matcher: workday
match-calendar mon..fri 9-17
target admin
comment Notify admins during working hours

matcher: night-and-weekend
match-calendar mon..fri 9-17
invert-match true
target on-call-admins
comment Separate target for non-working hours

matcher: backup-failures
match-field exact:type=vzdump
match-severity error
target backup-admins
comment Send notifications about backup failures to one group of admins

matcher: cluster-failures
match-field exact:type=replication,fencing
target cluster-admins
comment Send cluster-related notifications to other group of admins
```

17.4 Eventos de notificación

Eventos, tipos, severidades y metadatos enumerados por el manual:

Evento	type	Severidad	Metadatos adicionales
Actualizaciones disponibles	package-updates	info	hostname
Nodo fenced	fencing	error	hostname
Fallo de replicación	replication	error	hostname, job-id
Backup completado	vzdump	info	hostname, job-id (solo jobs)
Backup fallido	vzdump	error	hostname, job-id (solo jobs)
Correo para root	system-mail	unknown	hostname

Campos de metadatos

Campo	Descripción
type	Tipo de la notificación.
hostname	Hostname sin dominio, por ejemplo pve1.
job-id	ID del trabajo.

NOTA: Las notificaciones de jobs de backup solo incluyen job-id cuando el trabajo se ejecuta automáticamente según su programación, no cuando se lanza manualmente mediante Run now.

17.5 Reenvío del correo del sistema

Algunos daemons locales, como smartd, generan correos dirigidos inicialmente al usuario root local. Estos correos se convierten en en eventos de tipo system-mail y severidad unknown y se procesan mediante la configuración global de notificaciones.

Cuando se reenvían a un target Sendmail, contenido y cabeceras se conservan. Para otros targets, el sistema intenta extraer asunto y cuerpo principal; si el mensaje solo contiene HTML, se transforma a texto plano.

17.6 Permisos

Para modificar o consultar la configuración de Notification Targets se requieren Mapping.Modify o Mapping.Audit sobre:

```
/mapping/notifications
```

Para probar un target se necesita Mapping.Use, Mapping.Audit o Mapping.Modify sobre la misma ruta.

17.7 Notification Mode

Los jobs de backup permiten elegir entre dos modos mediante notification-mode:

- notification-system: usa la configuración global de notificaciones.
- legacy-sendmail: envía correo mediante sendmail a la dirección configurada en el job y permite elegir si se envía siempre o solo en caso de fallo. Los targets y matchers globales se ignoran.

legacy-sendmail reproduce el comportamiento anterior a Proxmox VE 8.1 y el manual indica que podría eliminarse en una versión futura.

17.8 Sustitución de plantillas de notificación

Proxmox VE utiliza plantillas Handlebars. Las plantillas originales están en:

```
/usr/share/pve-manager/templates/default/
```

Pueden sustituirse mediante archivos personalizados en:

```
/etc/pve/notification-templates/default/
```

Al renderizar, Proxmox VE intenta primero el override; si no existe o falla, usa la plantilla original.

Convención de nombres:

```
<type>-<body|subject>.<html|txt>.hbs
```

- vzdump-body.html.hbs: plantilla HTML del cuerpo de notificaciones de backup.
- package-updates-subject.txt.hbs: asunto de notificaciones sobre actualizaciones disponibles.

Sendmail y SMTP envían mensajes multipart HTML + texto plano, por lo que usan tanto <type>-body.html.hbs como <type>-body.txt.hbs. Los demás targets solo usan <type>-body.txt.hbs.

Fin del Capítulo 17 · Notificaciones

Con esta Parte 17 queda completada la traducción del Capítulo 17. La siguiente entrega comienza el Capítulo 18 · Daemons de servicio importantes.

CRITERIO DE TRADUCCIÓN: Se han conservado literalmente rutas, comandos, nombres de campos, valores, opciones de configuración y ejemplos de sintaxis. El contenido explicativo se ha traducido al castellano manteniendo la estructura y el alcance del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 18

Daemons de servicio importantes

pvedaemon · pveproxy · pvestatd · spiceproxy · pvescheduler

API local	Proxy HTTPS	Estado	SPICE	Scheduler
pvedaemon	pveproxy	pvestatd	spiceproxy	pvescheduler

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente servicios, puertos, rutas, variables y comandos.

Capítulo 18 · Daemons de servicio importantes

18.1 pvedaemon - Proxmox VE API Daemon

Este daemon expone toda la API de Proxmox VE en 127.0.0.1:85. Se ejecuta como root y, por tanto, dispone de permisos para realizar todas las operaciones privilegiadas.

NOTA: El daemon escucha únicamente en una dirección local, por lo que no puede accederse a él desde el exterior. El daemon pveproxy es el encargado de exponer la API al mundo exterior.

18.1.1 Número de workers

pvedaemon delega el tratamiento de las peticiones entrantes en procesos worker. De forma predeterminada inicia 3 workers, cantidad suficiente para la mayoría de cargas de trabajo.

En entornos con mucha automatización, donde se envía un volumen muy alto de solicitudes API y se observan tiempos de respuesta lentos o timeouts, puede incrementarse el número de workers mediante MAX_WORKERS en /etc/default/pvedaemon.

```
MAX_WORKERS=5
```

Un número mayor de workers puede aumentar el consumo de CPU. El valor debe ser superior a 0 e inferior a 128.

NOTA: La misma opción MAX_WORKERS existe también para pveproxy.

18.2 pveproxy - Proxmox VE API Proxy Daemon

Este daemon expone toda la API de Proxmox VE mediante HTTPS en el puerto TCP 8006. Se ejecuta como usuario www-data y dispone de permisos muy limitados.

Las operaciones que requieren más privilegios se reenvían al pvedaemon local. Las peticiones dirigidas a otros nodos se reenvían automáticamente a esos nodos, por lo que es posible administrar todo el clúster conectándose a un único nodo Proxmox VE.

18.2.1 Control de acceso basado en host

Puede configurarse una lista de control de acceso similar a la de Apache. Los valores se leen desde /etc/default/pveproxy.

```
ALLOW_FROM="10.0.0.1-10.0.0.5,192.168.0.0/22"
DENY_FROM="all"
POLICY="allow"
```

Las direcciones IP pueden expresarse utilizando cualquier sintaxis comprendida por Net::IP. El nombre all es un alias de 0/0 y ::/0, es decir, todas las direcciones IPv4 e IPv6.

La política predeterminada es allow.

Coincidencia	POLICY=deny	POLICY=allow
Solo Allow	allow	allow
Solo Deny	deny	deny
Sin coincidencia	deny	allow
Coinciden Allow y Deny	deny	allow

18.2.2 Dirección IP de escucha

Por defecto, pveproxy y spiceproxy escuchan en la dirección wildcard y aceptan conexiones tanto de clientes IPv4 como IPv6.

La variable LISTEN_IP de /etc/default/pveproxy permite controlar la dirección IP a la que se enlazan pveproxy y spiceproxy. La dirección debe estar configurada en el sistema.

Si el sysctl net.ipv6.bindv6only se establece al valor no predeterminado 1, los daemons solo aceptarán conexiones IPv6 y, además, pueden aparecer otros problemas. El manual recomienda eliminar ese ajuste o usar LISTEN_IP=0.0.0.0 si se desea aceptar únicamente IPv4.

LISTEN_IP puede emplearse para limitar el socket a una interfaz interna y reducir la exposición a Internet. Ejemplo IPv4:

```
LISTEN_IP="192.0.2.1"
```

Ejemplo IPv6:

```
LISTEN_IP="2001:db8:85a3::1"
```

Para una dirección IPv6 link-local debe añadirse el nombre de la interfaz:

```
LISTEN_IP="fe80::c463:8cff:feb9:6a4e%vmbr0"
```

ADVERTENCIA: Los nodos de un clúster necesitan acceder a pveproxy para comunicarse, posiblemente desde subredes distintas. El manual no recomienda definir LISTEN_IP en sistemas agrupados en clúster.

Para aplicar el cambio debe reiniciarse el nodo o reiniciar por completo los servicios pveproxy y spiceproxy:

```
systemctl restart pveproxy.service spiceproxy.service
```

NOTA: A diferencia de reload, un restart de pveproxy puede interrumpir workers de larga duración, por ejemplo una consola o shell abierta de un invitado. Aplique este cambio dentro de una ventana de mantenimiento.

18.2.3 Suite de cifrado SSL/TLS

La lista de cifrados puede definirse en /etc/default/pveproxy mediante CIPHERS para TLS <= 1.2 y CIPHERSUITES para TLS >= 1.3.

```
CIPHERS="ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:\
ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:\
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:\
ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA384:\
ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256"

CIPHERSUITES="TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:\
TLS_AES_128_GCM_SHA256"
```

Los valores anteriores son los predeterminados del manual. Para ver todas las opciones disponibles puede consultarse la página de manual ciphers(1) del paquete OpenSSL.

También puede permitirse que el cliente elija el cifrado mediante:

```
HONOR_CIPHER_ORDER=0
```

El valor predeterminado es utilizar el primer cifrado de la lista compatible tanto con el cliente como con pveproxy.

18.2.4 Versiones TLS soportadas

Las versiones inseguras SSL 2 y SSL 3 están deshabilitadas de forma incondicional en pveproxy. Las versiones TLS anteriores a 1.1 también están deshabilitadas por defecto en versiones recientes de OpenSSL, y pveproxy respeta esa configuración definida en /etc/ssl/openssl.cnf.

Para deshabilitar TLS 1.2 o TLS 1.3 pueden establecerse, respectivamente:

```
DISABLE_TLS_1_2=1
DISABLE_TLS_1_3=1
```

NOTA: Salvo que exista un motivo concreto, no se recomienda modificar manualmente las versiones TLS admitidas.

18.2.5 Parámetros Diffie-Hellman

Los parámetros Diffie-Hellman utilizados pueden definirse en /etc/default/pveproxy mediante DHPARAMS, apuntando a un archivo PEM:

```
DHPARAMS="/path/to/dhparams.pem"
```

Si la opción no se establece, se utilizan los parámetros integrados skip2048.

NOTA: Los parámetros DH solo se utilizan cuando se negocia una suite criptográfica que emplea el algoritmo de intercambio de claves DH.

18.2.6 Certificado HTTPS alternativo

Puede cambiarse el certificado por uno externo o por uno obtenido mediante ACME.

Si existen, pveproxy utiliza primero:

```
/etc/pve/local/pveproxy-ssl.pem
/etc/pve/local/pveproxy-ssl.key
```

Si no están presentes, utiliza como fallback:

```
/etc/pve/local/pve-ssl.pem
/etc/pve/local/pve-ssl.key
```

La clave privada no puede utilizar una passphrase.

La ubicación de la clave privada /etc/pve/local/pveproxy-ssl.key puede sobrescribirse mediante TLS_KEY_FILE en /etc/default/pveproxy:

```
TLS_KEY_FILE="/secrets/pveproxy.key"
```

NOTA: La integración ACME incluida no respeta este ajuste. El manual remite al capítulo de administración del sistema host para más detalles.

18.2.7 Compresión de respuestas

Por defecto, pveproxy utiliza compresión HTTP gzip para contenido comprimible cuando el cliente la soporta. Puede deshabilitarse en /etc/default/pveproxy:

```
COMPRESSION=0
```

NOTA: Deshabilitar la compresión mitiga ataques de canal lateral de la clase BREACH contra cookies de sesión autenticadas, a costa de usar algo más de ancho de banda en la interfaz web. La mayoría de despliegues no necesitan cambiar el valor predeterminado; el manual sugiere valorarlo cuando la interfaz web está expuesta a redes que no se controlan completamente.

18.2.8 Registro de la IP real del cliente

Por defecto, pveproxy registra la dirección IP del cliente que envió la solicitud. Si existe un proxy por delante, puede ser preferible registrar la IP del cliente original en lugar de la del proxy.

Para procesar una cabecera HTTP proporcionada por el proxy con fines de logging, defina PROXY_REAL_IP_HEADER:

```
PROXY_REAL_IP_HEADER="X-Forwarded-For"
```

Los valores no válidos presentes en la cabecera se ignoran.

Por defecto, el valor de esa cabecera se registra para todas las peticiones entrantes. Puede limitarse qué proxies son de confianza mediante PROXY_REAL_IP_ALLOW_FROM:

```
PROXY_REAL_IP_ALLOW_FROM="192.168.0.2"
```

Esta opción admite valores similares a ALLOW_FROM y DENY_FROM. Las direcciones pueden expresarse con la sintaxis reconocida por Net::IP, y all equivale a 0/0 y ::/0.

18.2.9 Número de workers

pveproxy también delega las peticiones entrantes en procesos worker. Inicia 3 por defecto. En cargas muy intensivas de automatización, si aparecen lentitud o timeouts, puede aumentarse MAX_WORKERS en /etc/default/pveproxy:

```
MAX_WORKERS=5
```

Un número mayor puede incrementar el consumo de CPU. El valor debe ser mayor que 0 y menor que 128. La misma opción existe en pvedaemon.

18.3 pvestatd - Proxmox VE Status Daemon

Este daemon consulta periódicamente el estado de las máquinas virtuales, los almacenamientos y los contenedores. El resultado se distribuye a todos los nodos del clúster.

18.4 spiceproxy - SPICE Proxy Service

SPICE (Simple Protocol for Independent Computing Environments) es una solución abierta de computación remota que permite acceso de cliente a pantallas y dispositivos remotos, como teclado, ratón o audio.

Su principal caso de uso es proporcionar acceso remoto a máquinas virtuales y contenedores.

spiceproxy escucha en el puerto TCP 3128 e implementa un proxy HTTP que reenvía solicitudes CONNECT desde el cliente SPICE hacia la VM correcta de Proxmox VE. Se ejecuta como usuario www-data y dispone de permisos muy limitados.

18.4.1 Control de acceso basado en host

Puede configurarse una lista de control de acceso similar a Apache. Los valores se leen desde /etc/default/pveproxy. El manual remite a la documentación de pveproxy para los detalles de ALLOW_FROM, DENY_FROM y POLICY.

18.5 pvescheduler - Proxmox VE Scheduler Daemon

Este daemon es responsable de iniciar trabajos de acuerdo con su programación, por ejemplo tareas de replicación y trabajos vzdump.

Para los trabajos vzdump, la configuración se obtiene del archivo:

```
/etc/pve/jobs.cfg
```

Resumen operativo de los daemons del capítulo

Daemon	Función principal	Escucha / ámbito	Identidad / notas
pvedaemon	API privilegiada local	127.0.0.1:85	root
pveproxy	API HTTPS y proxy entre nodos	TCP 8006	www-data; reenvía operaciones privilegiadas
pvestatd	Recopila estado de VMs, CT y storages	Todos los nodos	Distribuye estado al clúster
spiceproxy	Proxy HTTP para SPICE	TCP 3128	www-data
pvescheduler	Lanza trabajos programados	jobs del clúster	Replicación y vzdump; usa /etc/pve/jobs.cfg

IDEA CLAVE: pveproxy es la puerta HTTPS externa; pvedaemon es el backend local privilegiado. pvestatd aporta el estado distribuido, spiceproxy canaliza SPICE y pvescheduler ejecuta trabajos programados.

Fin del Capítulo 18 · Daemons de servicio importantes

Con esta Parte 18 queda completada la traducción del Capítulo 18. La siguiente entrega comienza el Capítulo 19 · Herramientas útiles de línea de comandos.

CRITERIO DE TRADUCCIÓN: Se han conservado literalmente los nombres de servicios, rutas, puertos, variables de entorno/configuración, comandos y valores que deben utilizarse tal como aparecen en Proxmox VE. El contenido descriptivo se ha traducido al castellano respetando la estructura del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 19

Herramientas útiles de línea de comandos

pvesubscription · pveperf · pvesh

Herramienta	Finalidad	Ámbito
pvesubscription	Gestionar suscripciones	Suscripción Proxmox VE
pveperf	Benchmark básico	CPU · disco · DNS
pvesh	Invocar la API localmente	API de Proxmox VE

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente nombres de comandos, parámetros y rutas.

Capítulo 19 · Herramientas útiles de línea de comandos

Este capítulo reúne tres utilidades de línea de comandos incluidas en Proxmox VE: gestión de suscripciones, una prueba básica de rendimiento y una interfaz shell para invocar directamente la API.

19.1 pvesubscription - Gestión de suscripciones

La herramienta pvesubscription se utiliza para gestionar las suscripciones de Proxmox VE.

19.2 pveperf - Script de benchmark de Proxmox VE

pveperf intenta recopilar algunos datos de rendimiento de CPU y disco sobre el sistema de archivos montado en PATH. Si no se especifica una ruta, utiliza / de forma predeterminada.

El resultado incluye las siguientes métricas:

Métrica	Qué mide	Referencia indicada en el manual
CPU BOGOMIPS	Suma de bogomips de todas las CPU.	-
REGEX/SECOND	Expresiones regulares por segundo; prueba de rendimiento de Perl.	Debería superar 300000.
HD SIZE	Tamaño del disco duro.	-
BUFFERED READS	Prueba simple de lectura del disco.	Los discos modernos deberían alcanzar al menos 40 MB/s.
AVERAGE SEEK TIME	Tiempo medio de búsqueda.	SCSI rápidos: < 8 ms; IDE/SATA comunes: 15-20 ms.
FSYNCS/SECOND	Número de fsync por segundo.	Debería ser > 200; el manual indica habilitar write-back cache en el controlador RAID si dispone de caché protegida por batería (BBWC).
DNS EXT	Tiempo medio para resolver un nombre DNS externo.	-
DNS INT	Tiempo medio para resolver un nombre DNS local.	-

19.3 Interfaz shell para la API de Proxmox VE

La herramienta de administración pvesh permite invocar directamente funciones de la API de Proxmox VE sin utilizar el servidor REST/HTTPS.

NOTA: Solo root puede utilizar esta interfaz directamente.

19.3.1 Ejemplos

Obtener la lista de nodos del clúster:

```
# pvesh get /nodes
```

Mostrar las opciones disponibles para el Datacenter:

```
# pvesh usage cluster/options -v
```

Establecer la consola HTML5 NoVNC como consola predeterminada del Datacenter:

```
# pvesh set cluster/options -console html5
```

Fin del Capítulo 19 · Herramientas útiles de línea de comandos

Con esta Parte 19 queda completada la traducción del Capítulo 19. La siguiente entrega comienza el Capítulo 20 · Preguntas frecuentes (FAQ).

NOTA: Criterio de traducción: se ha mantenido la organización del capítulo y se han preservado literalmente nombres de herramientas, comandos, parámetros y valores técnicos del manual original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 20

Preguntas frecuentes

Arquitectura · licencia · CPU · virtualización · soporte · actualizaciones · contenedores

Base	Virtualización	Ciclo de vida	Contenedores
Debian GNU/Linux	KVM · QEMU · LXC	Updates + upgrades	LXC · LXD · Docker

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se mantiene el contenido y el orden de las preguntas del manual original.

Capítulo 20 · Preguntas frecuentes

NOTA: Las nuevas preguntas frecuentes se añaden al final de esta sección.

1. ¿En qué distribución se basa Proxmox VE?

Proxmox VE se basa en Debian GNU/Linux.

2. ¿Qué licencia utiliza el proyecto Proxmox VE?

El código de Proxmox VE está licenciado bajo GNU Affero General Public License, versión 3 (AGPLv3).

3. ¿Proxmox VE puede ejecutarse en un procesador de 32 bits?

No. Proxmox VE funciona únicamente sobre CPU de 64 bits. Las arquitecturas compatibles indicadas en esta versión del manual son x86_64 (AMD64 o Intel 64) y aarch64 en las plataformas NVIDIA Grace Hopper y NVIDIA Vera. No existe plan para admitir hosts de 32 bits.

NOTA: Las máquinas virtuales y los contenedores sí pueden ser tanto de 32 como de 64 bits.

4. ¿Mi CPU admite virtualización?

En hosts x86 puede comprobarse buscando las extensiones vmx o svm en la salida de /proc/cpuinfo:

```
egrep '(vmx|svm)' /proc/cpuinfo
```

NOTA: Esta comprobación se aplica únicamente a hosts x86. En las plataformas arm64 compatibles, la virtualización asistida por hardware KVM está disponible y no existe un flag de CPU equivalente.

5. CPU Intel compatibles

Procesadores de 64 bits con soporte para Intel Virtualization Technology (Intel VT-x).

6. CPU AMD compatibles

Procesadores de 64 bits con soporte para AMD Virtualization Technology (AMD-V).

7. Plataformas arm64 compatibles

Las plataformas NVIDIA Grace Hopper y NVIDIA Vera. El manual remite a la sección de soporte ARM de 64 bits para más detalles.

8. ¿Qué es un contenedor / virtual environment (VE) / virtual private server (VPS)?

En el contexto de los contenedores, estos términos se refieren al concepto de virtualización a nivel de sistema operativo. El kernel permite ejecutar varias instancias aisladas que comparten el mismo kernel. Cuando se habla de LXC, Proxmox VE denomina contenedores a esas instancias. Al utilizar el kernel del host en vez de emular un sistema operativo completo, el overhead es menor, pero se limita a invitados Linux.

9. ¿Qué es un invitado QEMU/KVM o máquina virtual?

Es un sistema invitado virtualizado bajo Proxmox VE utilizando QEMU y el módulo KVM del kernel de Linux.

10. ¿Qué es QEMU?

QEMU es un emulador y virtualizador de máquinas genérico y open source. Utiliza el módulo KVM del kernel de Linux para conseguir un rendimiento cercano al nativo ejecutando directamente el código del invitado sobre la CPU del host. No está limitado a invitados Linux y permite ejecutar otros sistemas operativos.

11. ¿Durante cuánto tiempo tendrá soporte mi versión de Proxmox VE?

Las versiones de Proxmox VE reciben soporte al menos durante el mismo periodo en que la versión correspondiente de Debian está soportada por el Debian Security Team, aproximadamente tres años después de su primera publicación. Proxmox VE sigue un modelo rolling release y el manual recomienda utilizar la última versión estable.

Versión Proxmox VE	Versión Debian	Primera publicación	EOL Debian	EOL Proxmox
Proxmox VE 9	Debian 13 (Trixie)	2025-08	tba	tba
Proxmox VE 8	Debian 12 (Bookworm)	2023-06	2026-07	2026-08
Proxmox VE 7	Debian 11 (Bullseye)	2021-07	2024-07	2024-07
Proxmox VE 6	Debian 10 (Buster)	2019-07	2022-09	2022-09
Proxmox VE 5	Debian 9 (Stretch)	2017-07	2020-07	2020-07
Proxmox VE 4	Debian 8 (Jessie)	2015-10	2018-06	2018-06
Proxmox VE 3	Debian 7 (Wheezy)	2013-05	2016-04	2017-02
Proxmox VE 2	Debian 6 (Squeeze)	2012-04	2014-05	2014-05
Proxmox VE 1	Debian 5 (Lenny)	2008-10	2012-03	2013-01

12. ¿Cómo puedo actualizar Proxmox VE a la siguiente point release?

Las actualizaciones menores, por ejemplo de Proxmox VE 8.1 a 8.2 o 8.3, pueden realizarse como cualquier actualización normal. Aun así, el manual recomienda revisar primero las release notes por si existe algún cambio relevante o incompatible.

La actualización puede hacerse desde Node -> Updates en la interfaz web o desde la CLI:

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

NOTA: Compruebe siempre que los repositorios estén configurados correctamente y continúe con la actualización únicamente si apt update finaliza sin errores.

13. ¿Cómo puedo actualizar Proxmox VE a la siguiente major release?

Las actualizaciones de versión principal, por ejemplo de Proxmox VE 8.4 a 9.0, también están soportadas. Deben planificarse y probarse cuidadosamente y nunca deben iniciarse sin disponer de una copia de seguridad actual.

Los pasos concretos dependen de cada instalación. El manual proporciona guías generales para los siguientes saltos de versión :

- Proxmox VE 8 -> 9
- Proxmox VE 7 -> 8
- Proxmox VE 6 -> 7
- Proxmox VE 5 -> 6
- Proxmox VE 4 -> 5
- Proxmox VE 3 -> 4

14. LXC vs LXD vs Proxmox Containers vs Docker

LXC es una interfaz de espacio de usuario para las funciones de aislamiento del kernel Linux. Mediante una API y herramientas sencillas permite crear y administrar contenedores de sistema. Al igual que el antiguo OpenVZ, LXC está orientado a virtualización de sistemas: permite ejecutar un sistema operativo completo dentro del contenedor, iniciar sesión por SSH, crear usuarios, ejecutar Apache, etc.

LXD se construye sobre LXC para ofrecer una experiencia de usuario diferente. Internamente usa LXC mediante liblxc y sus bindings de Go para crear y administrar contenedores. En esencia, ofrece herramientas y un sistema de plantillas alternativos, además de administración a través de red.

Proxmox Containers es el nombre que el manual utiliza para los contenedores creados y gestionados mediante Proxmox Container Toolkit (pct). También están orientados a virtualización de sistemas y utilizan LXC como base. Con el soporte para imágenes OCI - indicado como technology preview en esta versión - Proxmox VE también puede crear contenedores a partir de imágenes OCI, incluidos contenedores de sistema completos y contenedores de aplicación, estos últimos también en tech preview.

Proxmox Container Toolkit está estrechamente integrado con Proxmox VE: conoce la configuración de clúster, puede utilizar las mismas redes y almacenamientos que las máquinas virtuales QEMU, permite aplicar el firewall de Proxmox VE, crear y restaurar backups, gestionar contenedores mediante HA y controlar todo a través de la API de Proxmox VE.

Docker está orientado a ejecutar una aplicación individual dentro de un entorno aislado y autocontenido. Estos entornos suelen denominarse Application Containers, frente a los System Containers de LXC. Una instancia Docker se administra desde el host mediante la CLI de Docker Engine.

NOTA: El manual no recomienda ejecutar Docker directamente sobre el host Proxmox VE. Aunque ejecutar Application Containers directamente como Proxmox Containers puede resultar cómodo, este uso continúa indicado como technology preview. Para casos que requieran orquestación de contenedores o migración en vivo, el manual sigue recomendando ejecutarlos dentro de una máquina virtual QEMU de Proxmox VE.

Fin del Capítulo 20 · Preguntas frecuentes

Con esta Parte 20 queda completada la traducción del Capítulo 20. La siguiente entrega corresponde al Capítulo 21 · Bibliografía.

CRITERIO DE TRADUCCIÓN: Se han mantenido el orden, el alcance y las afirmaciones de la FAQ original. No se han actualizado fechas, versiones o recomendaciones con información externa; los datos reflejan el manual fuente.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE 21

Bibliografía

Proxmox VE · tecnologías relacionadas · temas relacionados

Proxmox VE	Tecnología relacionada	Temas relacionados
5 referencias	10 referencias	1 referencia

Basado en Proxmox VE Administration Guide, Release 9.2.4

Las referencias bibliográficas se conservan en su idioma y forma originales.

Capítulo 21 · Bibliografía

NOTA: En una bibliografía no se traducen los títulos de las obras citadas. Se mantienen autores, títulos, editoriales, años, ISBN/ASIN y claves bibliográficas tal como aparecen en el manual fuente.

21.1 Libros sobre Proxmox VE

[1] [Ahmed16] Wasim Ahmed. Mastering Proxmox - Third Edition. Packt Publishing, 2017. ISBN 978-1788397605

[2] [Ahmed15] Wasim Ahmed. Proxmox Cookbook. Packt Publishing, 2015. ISBN 978-1783980901

[3] [Cheng14] Simon M.C. Cheng. Proxmox High Availability. Packt Publishing, 2014. ISBN 978-1783980888

[4] [Goldman16] Rik Goldman. Learning Proxmox VE. Packt Publishing, 2016. ISBN 978-1783981786

[5] [Surber16]] Lee R. Surber. Virtualization Complete: Business Basic Edition. Linux Solutions (LRS-TEK), 2016. ASIN B01BBVQZT6

21.2 Libros sobre tecnología relacionada

[6] [Hertzog13] Raphaël Hertzog, Roland Mas., Freexian SARL The Debian Administrator's Handbook: Debian Bullseye from Discovery to Mastery, Freexian, 2021. ISBN 979-10-91414-20-3

[7] [Bir96] Kenneth P. Birman. Building Secure and Reliable Network Applications. Manning Publications Co, 1996. ISBN 978-1884777295

[8] [Walsh10] Norman Walsh. DocBook 5: The Definitive Guide. O'Reilly & Associates, 2010. ISBN 978-0596805029

[9] [Richardson07] Leonard Richardson & Sam Ruby. RESTful Web Services. O'Reilly Media, 2007. ISBN 978-0596529260

[10] [Singh15] Karan Singh. Learning Ceph. Packt Publishing, 2015. ISBN 978-1783985623

[11] [Singh16] Karan Signh. Ceph Cookbook Packt Publishing, 2016. ISBN 978-1784393502

[12] [Mauerer08] Wolfgang Mauerer. Professional Linux Kernel Architecture. John Wiley & Sons, 2008. ISBN 978-0470343432

[13] [Loshin03] Pete Loshin, IPv6: Theory, Protocol, and Practice, 2nd Edition. Morgan Kaufmann, 2003. ISBN 978-1558608108

[14] [Loeliger12] Jon Loeliger & Matthew McCullough. Version Control with Git: Powerful tools and techniques for collaborative software development. O'Reilly and Associates, 2012. ISBN 978-1449316389

[15] [Kreibich10] Jay A. Kreibich. Using SQLite, O'Reilly and Associates, 2010. ISBN 978-0596521189

21.3 Libros sobre temas relacionados

[16] [Bessen09] James Bessen & Michael J. Meurer, Patent Failure: How Judges, Bureaucrats, and Lawyers Put Innovators at Risk. Princeton Univ Press, 2009. ISBN 978-0691143217

Fin del Capítulo 21 · Bibliografía

Con esta Parte 21 queda completada la traducción de los capítulos principales del manual. La siguiente entrega comienza el Apéndice A · Interfaz de línea de comandos.

CRITERIO DE TRADUCCIÓN: Se han traducido únicamente los encabezados y textos editoriales del capítulo. Las referencias bibliográficas permanecen sin traducir para conservar la identificación exacta de cada obra.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A1

Interfaz de línea de comandos

A.1 General · A.2 Formato de salida · A.3 pvesm - Proxmox VE Storage Manager

Bloque	Herramienta	Contenido
A.1-A.2	pvesm	Formato CLI · almacenamiento · scan · alloc · prune · import/export

Basado en Proxmox VE Administration Guide, Release 9.2.4

Los comandos, opciones, valores literales, rutas y expresiones regulares se conservan sin traducir.

Apéndice A · Interfaz de línea de comandos

A.1 General

Respecto al uso históricamente no uniforme de mayúsculas y minúsculas en los nombres de opciones, consulte la sección relacionada con los archivos de configuración.

A.2 Opciones de formato de salida [FORMAT_OPTIONS]

Es posible especificar el formato de salida mediante el parámetro --output-format. El formato predeterminado, text, utiliza arte ASCII para dibujar bordes alrededor de las tablas y transforma ciertos valores a una representación más legible:

- Los valores Unix epoch se muestran como fechas ISO 8601.
- Las duraciones se muestran como semanas/días/horas/minutos/segundos, por ejemplo 1d 5h.
- Los tamaños en bytes incluyen unidades: B, KiB, MiB, GiB, TiB y PiB.
- Las fracciones se muestran como porcentajes; por ejemplo, 1.0 se representa como 100%.

También puede suprimirse completamente la salida mediante --quiet.

--human-readable <boolean> (default = 1) — Ejecuta las funciones de renderizado de salida para producir texto legible.

--noborder <boolean> (default = 0) — No dibuja bordes cuando el formato es text.

--noheader <boolean> (default = 0) — No muestra las cabeceras de columna en formato text.

--output-format <json | json-pretty | text | yaml> (default = text) — Formato de salida.

--quiet <boolean> — Suprime la impresión de resultados.

A.3 pvesm - Proxmox VE Storage Manager

```
pvesm <COMMAND> [ARGS] [OPTIONS]
```

pvesm add <type> <storage> [OPTIONS]

Crea una nueva definición de almacenamiento.

<type>: <btrfs | cephfs | cifs | dir | esxi | iscsi | iscsidirect | lvm | lvmthin | nfs | pbs | rbd | zfs | zfspool> — Tipo de almacenamiento.

<storage>: <storage ID> — Identificador del almacenamiento.

--authsupported <string> — Authsupported.

--base <string> — Volumen base. Este volumen se activa automáticamente.

--blocksize <a power of 2 with optional k or m suffix> — Tamaño de bloque ZFS.

--bwlimit [clone=<LIMIT>] [,default=<LIMIT>] [,migration=<LIMIT>] [,move=<LIMIT>] [,restore=<LIMIT>] — Establece límites de ancho de banda de E/S para distintas operaciones, en KiB/s.

--comstar_hg <string> — Grupo de hosts para vistas COMSTAR.

--comstar_tg <string> — Grupo de targets para vistas COMSTAR.

--content <string> — Tipos de contenido permitidos.

--content-dirs <string> — Sobrescribe los directorios predeterminados para tipos de contenido.

--create-base-path <boolean> (default = yes) — Crea el directorio base si no existe.

--create-subdirs <boolean> (default = yes) — Puebla el directorio con la estructura predeterminada.

--data-pool <string> — Data Pool; solo para erasure coding.

--datastore <string> — Nombre del datastore de Proxmox Backup Server.

--disable <boolean> — Marca para deshabilitar el almacenamiento.

--domain <string> — Dominio CIFS.

--encryption-key a file containing an encryption key, or the special value "autogen" — Clave de cifrado. Use autogen para generar una automáticamente sin passphrase.

--export <string> — Ruta de exportación NFS.

--fingerprint ([A-Fa-f0-9]{2}:){31}[A-Fa-f0-9]{2} — Huella SHA-256 del certificado.

--format <qcow2 | raw | subvol | vmdk> — Formato de imagen predeterminado.

--fs-name <string> — Nombre del sistema de archivos Ceph.

--fuse <boolean> — Monta CephFS mediante FUSE.

--is_mountpoint <string> (default = no) — Considera la ruta como un punto de montaje gestionado externamente y marca el almacenamiento como offline si no está montado. Un valor booleano yes/no equivale a indicar la ruta objetivo.

--iscsiprovider <string> — Proveedor iSCSI.

--keyring file containing the keyring to authenticate in the Ceph cluster — Contenido del keyring de cliente para autenticarse en clústeres Ceph externos.

--krbd <boolean> (default = 0) — Accede siempre a RBD mediante el módulo kernel krbd.

--lio_tpg <string> — Target portal group para targets Linux LIO.

--master-pubkey a file containing a PEM-formatted master public key — Clave pública RSA en formato PEM codificada en Base64. Se utiliza para cifrar una copia de encryption-key que se añade a cada backup cifrado.

--max-protected-backups <integer> (-1 - N) (default = Unlimited for users with Datastore.Allocate privilege, 5 for other users) — Número máximo de backups protegidos por invitado. Use -1 para ilimitado.

--mkdir <boolean> (default = yes) — Crea el directorio si no existe y lo puebla con subdirectorios predeterminados. Obsoleto: use create-base-path y create-subdirs.

--monhost <string> — Direcciones IP de monitores para clústeres externos.

--mountpoint <string> — Punto de montaje.

--namespace <string> — Namespace.

--nocow <boolean> (default = 0) — Establece NOCOW en los archivos. Deshabilita checksumming de datos y hace que los errores de datos no sean recuperables, aunque permite E/S directa. Úselo solo cuando los datos no necesiten más protección qu e en un único disco ext4 sin RAID subyacente.

--nodes <string> — Lista de nodos a los que se aplica la configuración del almacenamiento.

--nowritecache <boolean> — Deshabilita la caché de escritura en el target.

--options <string> — Opciones de montaje NFS/CIFS; consulte man nfs o man mount.cifs.

--password <password> — Contraseña para acceder al recurso compartido o datastore.

--path <string> — Ruta del sistema de archivos.

--pool <string> — Pool.

--port <integer> (1 - 65535) — Puerto para conectarse al almacenamiento en lugar del predeterminado, por ejemplo con PBS o ESXi. Para NFS/CIFS, configure el puerto dentro de options.

--portal <string> — Portal iSCSI, como IP o nombre DNS con puerto opcional.

--preallocation <falloc | full | metadata | off> (default = metadata) — Modo de preasignación para imágenes raw y qcow2. metadata sobre raw equivale a preallocation=off.

--prune-backups [keep-all=<1|0>] [,keep-daily=<N>] [,keep-hourly=<N>] [,keep-last=<N>] [,keep-monthly=<N>] [,keep-weekly=<N>] [,keep-yearly=<N>] — Opciones de retención. Los intervalos cortos se procesan primero, empezando por keep-last; las copias ya cubiertas por una regla no vuelven a evaluarse por las siguientes.

--saferemove <boolean> — Sobrescribe con ceros los datos al eliminar LVs.

--saferemove-stepsize <1 | 16 | 2 | 32 | 4 | 8> (default = 32) — Tamaño de paso de borrado en MiB, limitado al máximo admitido por el almacenamiento.

--saferemove_throughput <string> — Rendimiento del borrado, valor del parámetro cstream -t.

--server <string> — IP o nombre DNS del servidor.

--share <string> — Recurso compartido CIFS.

--shared <boolean> — Indica que el almacenamiento tiene el mismo contenido en todos los nodos, o en los incluidos en nodes. No convierte un almacenamiento local en compartido; solo marca como compartido uno que ya lo sea.

--skip-cert-verification <boolean> (default = false) — Deshabilita la verificación del certificado TLS; actívelo solo en redes completamente confiables.

--smbversion <2.0 | 2.1 | 3 | 3.0 | 3.11 | default> (default = default) — Versión del protocolo SMB. default negocia la versión SMB2+ más alta admitida por cliente y servidor.

--snapshot-as-volume-chain <boolean> (default = 0) — Habilita snapshots independientes del fabricante del almacenamiento mediante cadenas de backing volumes.

--sparse <boolean> — Utiliza volúmenes sparse.

--subdir <string> — Subdirectorio que se montará.

--tagged_only <boolean> — Lista únicamente volúmenes lógicos etiquetados con pve-vm-ID.

--target <string> — Target iSCSI.

--thinpool <string> — Nombre del LV del thin pool LVM.

--username <string> — ID de RBD.

--vgname <string> — Nombre del Volume Group.

--zfs-base-path <string> — Ruta base donde buscar los dispositivos de bloques ZFS creados. Se establece automáticamente si se omite; normalmente /dev/zvol.

NOTA: El valor rootdir se utiliza para contenedores y images para máquinas virtuales.

pvesm alloc <storage> <vmid> <filename> <size> [OPTIONS]

Asigna una imagen de disco.

<storage>: <storage ID> — Identificador del almacenamiento.

<vmid>: <integer> (100 - 999999999) — VM propietaria del volumen.

<filename>: <string> — Nombre del archivo que se creará.

<size>: \d+[MG]? — Tamaño en KiB. Admite los sufijos M y G.

--format <qcow2 | raw | subvol | vmdk> — Formato de la imagen.

NOTA: La opción format requiere que se especifique size.

pvesm apiinfo

Devuelve APIVER y APIAGE.

pvesm cifsscan

Alias de pvesm scan cifs.

pvesm export <volume> <format> <filename> [OPTIONS]

Se utiliza internamente para exportar un volumen.

<volume>: <string> — Identificador del volumen.

<format>: <btrfs | qcow2+size | raw+size | tar+size | vmdk+size | zfs> — Formato del stream de exportación.

<filename>: <string> — Nombre del archivo de destino.

--base (?^i:[a-z0-9_\-]{1,40}) — Snapshot desde el que se inicia un stream incremental.

--snapshot (?^i:[a-z0-9_\-]{1,40}) — Snapshot que se exportará.

--snapshot-list <string> — Lista ordenada de snapshots que se transferirán.

--with-snapshots <boolean> (default = 0) — Incluye o no snapshots intermedios en el stream.

pvesm extractconfig <volume>

Extrae la configuración desde un archivo de backup vzdump.

<volume>: <string> — Identificador del volumen.

pvesm free <volume> [OPTIONS]

Elimina un volumen.

<volume>: <string> — Identificador del volumen.

--delay <integer> (1 - 30) — Tiempo de espera hasta que finalice la tarea. Devuelve null si termina dentro de ese intervalo.

--storage <storage ID> — Identificador del almacenamiento.

pvesm help [OPTIONS]

Muestra ayuda sobre un comando.

--extra-args <array> — Muestra ayuda de un comando específico.

--verbose <boolean> — Formato de salida detallado.

pvesm import <volume> <format> <filename> [OPTIONS]

Se utiliza internamente para importar un volumen.

<volume>: <string> — Identificador del volumen.

<format>: <btrfs | qcow2+size | raw+size | tar+size | vmdk+size | zfs> — Formato del stream de importación.

<filename>: <string> — Nombre del archivo origen. Con - se usa stdin; tcp://<IP-or-CIDR> permite una conexión TCP y unix://PATH-TO-SOCKET un socket UNIX. Cualquier otro valor se trata como archivo normal.

--allow-rename <boolean> (default = 0) — Elige un nuevo ID de volumen si el solicitado ya existe, en vez de devolver un error.

--base (?^i:[a-z0-9_\-]{1,40}) — Snapshot base de un stream incremental.

--delete-snapshot (?^i:[a-z0-9_\-]{1,80}) — Snapshot que se eliminará si la importación termina correctamente.

--snapshot (?^i:[a-z0-9_\-]{1,40}) — Snapshot del estado actual cuando el stream contiene snapshots.

--with-snapshots <boolean> (default = 0) — Indica si el stream incluye snapshots intermedios.

pvesm iscsiscan

Alias de pvesm scan iscsi.

pvesm list <storage> [OPTIONS]

Lista el contenido del almacenamiento.

<storage>: <storage ID> — Identificador del almacenamiento.

--content <string> — Lista únicamente contenido de este tipo.

--vmid <integer> (100 - 999999999) — Lista únicamente imágenes de esta VM.

pvesm lvmscan

Alias de pvesm scan lvm.

pvesm lvmthinscan

Alias de pvesm scan lvmthin.

pvesm nfsscan

Alias de pvesm scan nfs.

pvesm path <volume>

Obtiene la ruta del sistema de archivos para un volumen.

<volume>: <string> — Identificador del volumen.

pvesm prune-backups <storage> [OPTIONS]

Elimina backups según las reglas de retención. Solo se consideran los que usan el esquema de nombres estándar. Si no se indican opciones keep, se usan las de la configuración del almacenamiento.

<storage>: <storage ID> — Identificador del almacenamiento.

--dry-run <boolean> — Muestra qué se eliminaría sin borrar nada.

--keep-all <boolean> — Conserva todos los backups; entra en conflicto con el resto de opciones keep.

--keep-daily <N> — Conserva backups de los últimos N días distintos; si hay varios en un día, conserva el más reciente.

--keep-hourly <N> — Conserva backups de las últimas N horas distintas; si hay varios en una hora, conserva el más reciente.

--keep-last <N> — Conserva los últimos N backups.

--keep-monthly <N> — Conserva backups de los últimos N meses distintos; si hay varios en un mes, conserva el más reciente.

--keep-weekly <N> — Conserva backups de las últimas N semanas distintas; si hay varios en una semana, conserva el más reciente.

--keep-yearly <N> — Conserva backups de los últimos N años distintos; si hay varios en un año, conserva el más reciente.

--type <lxc | qemu> — Considera solo backups del tipo de invitado indicado.

--vmid <integer> (100 - 999999999) — Considera solo backups de este invitado.

pvesm remove <storage>

Elimina la configuración de un almacenamiento.

<storage>: <storage ID> — Identificador del almacenamiento.

pvesm scan cifs <server> [OPTIONS]

Explora un servidor CIFS remoto.

<server>: <string> — Dirección del servidor, por nombre o IP.

--domain <string> — Dominio SMB o Workgroup.

--password <password> — Contraseña del usuario.

--username <string> — Nombre de usuario.

pvesm scan iscsi <portal>

Explora un servidor iSCSI remoto.

<portal>: <string> — Portal iSCSI, por IP o nombre DNS con puerto opcional.

pvesm scan lvm

Lista los Volume Groups LVM locales.

pvesm scan lvmthin <vg>

Lista los LVM Thin Pools locales.

<vg>: [a-zA-Z0-9\.\+_][a-zA-Z0-9\.\+_\-]+ — Sin descripción disponible en la fuente.

pvesm scan nfs <server>

Explora un servidor NFS remoto.

<server>: <string> — Dirección del servidor, por nombre o IP.

pvesm scan pbs <server> <username> --password <string> [OPTIONS] [FORMAT_OPTIONS]

Explora un Proxmox Backup Server remoto.

<server>: <string> — Dirección del servidor, por nombre o IP.

<username>: <string> — Nombre de usuario o ID de API Token.

--fingerprint ([A-Fa-f0-9]{2}:){31}[A-Fa-f0-9]{2} — Huella SHA-256 del certificado.

--password <string> — Contraseña del usuario o secreto del API Token.

--port <integer> (1 - 65535) (default = 8007) — Puerto opcional.

pvesm scan zfs

Explora la lista de pools ZFS del nodo local.

pvesm set <storage> [OPTIONS]

Actualiza la configuración de un almacenamiento.

<storage>: <storage ID> — Identificador del almacenamiento.

--blocksize <a power of 2 with optional k or m suffix> — Tamaño de bloque ZFS.

--bwlimit [clone=<LIMIT>] [,default=<LIMIT>] [,migration=<LIMIT>] [,move=<LIMIT>] [,restore=<LIMIT>] — Establece límites de ancho de banda de E/S para distintas operaciones, en KiB/s.

--comstar_hg <string> — Grupo de hosts para vistas COMSTAR.

--comstar_tg <string> — Grupo de targets para vistas COMSTAR.

--content <string> — Tipos de contenido permitidos.

--content-dirs <string> — Sobrescribe los directorios predeterminados de los tipos de contenido.

--create-base-path <boolean> (default = yes) — Crea el directorio base si no existe.

--create-subdirs <boolean> (default = yes) — Puebla el directorio con la estructura predeterminada.

--data-pool <string> — Data Pool, solo para erasure coding.

--delete <string> — Lista de ajustes que se desean eliminar.

--digest <string> — Impide cambios si el archivo de configuración actual tiene un digest distinto; permite evitar modificaciones concurrentes.

--disable <boolean> — Marca para deshabilitar el almacenamiento.

--domain <string> — Dominio CIFS.

--encryption-key a file containing an encryption key, or the special value "autogen" — Clave de cifrado. autogen genera una automáticamente sin passphrase.

--fingerprint ([A-Fa-f0-9]{2}:){31}[A-Fa-f0-9]{2} — Huella SHA-256 del certificado.

--format <qcow2 | raw | subvol | vmdk> — Formato de imagen predeterminado.

--fs-name <string> — Nombre del sistema de archivos Ceph.

--fuse <boolean> — Monta CephFS mediante FUSE.

--is_mountpoint <string> (default = no) — Trata la ruta como un punto de montaje gestionado externamente y marca el almacenamiento como offline si no está montado.

--keyring file containing the keyring to authenticate in the Ceph cluster — Contenido del keyring de cliente para clústeres Ceph externos.

--krbd <boolean> (default = 0) — Accede siempre a RBD mediante el módulo kernel krbd.

--lio_tpg <string> — Target portal group para Linux LIO.

--master-pubkey a file containing a PEM-formatted master public key — Clave pública RSA PEM codificada en Base64 usada para cifrar una copia de encryption-key dentro de cada backup cifrado.

--max-protected-backups <integer> (-1 - N) (default = Unlimited for users with Datastore.Allocate privilege, 5 for other users) — Número máximo de backups protegidos por invitado; -1 significa ilimitado.

--mkdir <boolean> (default = yes) — Obsoleto. Crea el directorio y subdirectorios predeterminados; use create-base-path y create-subdirs.

--monhost <string> — Direcciones IP de monitores para clústeres externos.

--mountpoint <string> — Punto de montaje.

--namespace <string> — Namespace.

--nocow <boolean> (default = 0) — Activa NOCOW. Deshabilita checksumming de datos y hace que los errores no sean recuperables, permitiendo E/S directa. Úselo solo cuando resulte aceptable un nivel de protección equivalente a un único disco ext4 sin RAID.

--nodes <string> — Lista de nodos a los que se aplica la configuración.

--nowritecache <boolean> — Deshabilita la caché de escritura en el target.

--options <string> — Opciones de montaje NFS/CIFS.

--password <password> — Contraseña para acceder al recurso compartido o datastore.

--pool <string> — Pool.

--port <integer> (1 - 65535) — Puerto alternativo para el almacenamiento; en NFS/CIFS el puerto se configura dentro de options.

--preallocation <falloc | full | metadata | off> (default = metadata) — Modo de preasignación para raw/qcow2. metadata sobre raw equivale a off.

--prune-backups [keep-all=<1|0>] [,keep-daily=<N>] [,keep-hourly=<N>] [,keep-last=<N>] [,keep-monthly=<N>] [,keep-weekly=<N>] [,keep-yearly=<N>] — Reglas de retención; se procesan primero los intervalos cortos y las copias ya cubiertas no se vuelven a considerar.

--saferemove <boolean> — Sobrescribe con ceros los datos al eliminar LVs.

--saferemove-stepsize <1 | 16 | 2 | 32 | 4 | 8> (default = 32) — Tamaño del paso de borrado en MiB.

--saferemove_throughput <string> — Rendimiento del borrado, parámetro cstream -t.

--server <string> — IP o nombre DNS del servidor.

--shared <boolean> — Marca un almacenamiento ya compartido como accesible con el mismo contenido desde todos los nodos indicados. No hace compartido un almacenamiento local.

--skip-cert-verification <boolean> (default = false) — Deshabilita la verificación TLS; solo para redes totalmente confiables.

--smbversion <2.0 | 2.1 | 3 | 3.0 | 3.11 | default> (default = default) — Versión SMB; default negocia la versión SMB2+ más alta soportada.

--snapshot-as-volume-chain <boolean> (default = 0) — Habilita snapshots agnósticos respecto al proveedor mediante cadenas de volúmenes backing.

--sparse <boolean> — Utiliza volúmenes sparse.

--subdir <string> — Subdirectorio que se montará.

--tagged_only <boolean> — Lista únicamente volúmenes lógicos etiquetados con pve-vm-ID.

--username <string> — ID de RBD.

--zfs-base-path <string> — Ruta base de los dispositivos de bloques ZFS. Normalmente /dev/zvol.

NOTA: Para content, rootdir corresponde a contenedores e images a máquinas virtuales.

pvesm status [OPTIONS]

Obtiene el estado de todos los datastores.

--content <string> — Lista solo almacenamientos que admiten este tipo de contenido.

--enabled <boolean> (default = 0) — Lista únicamente almacenamientos habilitados.

--format <boolean> (default = 0) — Incluye información sobre formatos.

--storage <storage ID> — Lista únicamente el estado del almacenamiento indicado.

--target <string> — Si el target es distinto del nodo actual, lista únicamente almacenamientos compartidos cuyo contenido sea accesible desde el nodo actual y desde el nodo target.

pvesm zfsscan

Alias de pvesm scan zfs.

Fin del Apéndice A1

Esta entrega cubre A.1 General, A.2 Opciones de formato de salida y A.3 pvesm - Proxmox VE Storage Manager. La siguiente parte comienza en A.4 pvesubscription.

CRITERIO DE TRADUCCIÓN: Se han conservado literalmente comandos, argumentos, expresiones regulares, rutas, nombres de opciones y valores enumerados. Solo se han traducido las descripciones y textos explicativos.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A2

Interfaz de línea de comandos

A.4 pvesubscription · A.5 pveperf · A.6 pveceph

Suscripción	Rendimiento	Ceph
pvesubscription	pveperf	pveceph

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, nombres de opciones, expresiones regulares, rutas y valores que deben introducirse tal como aparecen.

A.4 pvesubscription - Gestor de suscripciones de Proxmox VE

```
pvesubscription <COMMAND> [ARGS] [OPTIONS]
```

Herramienta para gestionar la suscripción del nodo Proxmox VE.

pvesubscription delete

```
pvesubscription delete
```

Elimina la clave de suscripción de este nodo.

pvesubscription get

```
pvesubscription get
```

Lee la información de suscripción.

pvesubscription help

```
pvesubscription help [OPTIONS]
```

Muestra ayuda sobre el comando especificado.

`--extra-args <array>` — Muestra ayuda para un comando concreto.

`--verbose <boolean>` — Formato de salida detallado.

pvesubscription set

```
pvesubscription set <key>
```

Establece la clave de suscripción.

`<key>: \s*pve([1248])([cbsp])-(arm-)?[0-9a-f]{10}\s*` — Clave de suscripción de Proxmox VE.

pvesubscription set-offline-key

```
pvesubscription set-offline-key <data>
```

Solo para uso interno. Para establecer una clave offline debe utilizarse el paquete proxmox-offline-mirror-helper.

`<data>: <string>` — Blob firmado con información de suscripción.

pvesubscription update

```
pvesubscription update [OPTIONS]
```

Actualiza la información de suscripción.

`--force <boolean> (default = 0)` — Conecta siempre con el servidor aunque la caché local siga siendo válida.

A.5 pveperf - Script de benchmark de Proxmox VE

```
pveperf [PATH]
```

Ejecuta el benchmark de Proxmox VE sobre la ruta indicada. Si se omite PATH, se utiliza la ruta predeterminada correspondiente al sistema.

NOTA: En el Capítulo 19 se explican las métricas que muestra pveperf, como CPU BOGOMIPS, REGEX/SECOND, lecturas, latencia de seek, FSYNCS/SECOND y tiempos DNS. En este apéndice, la referencia CLI original solo documenta la sintaxis pveperf [PATH].

A.6 pveceph - Gestión de servicios Ceph en nodos Proxmox VE

```
pveceph <COMMAND> [ARGS] [OPTIONS]
```

Herramienta de línea de comandos para instalar, inicializar y administrar componentes Ceph integrados en Proxmox VE.

Alias de compatibilidad

```
pveceph createmgr
```

Alias de pveceph mgr create.

```
pveceph createmon
```

Alias de pveceph mon create.

```
pveceph createosd
```

Alias de pveceph osd create.

```
pveceph createpool
```

Alias de pveceph pool create.

```
pveceph destroymgr
```

Alias de pveceph mgr destroy.

```
pveceph destroymon
```

Alias de pveceph mon destroy.

```
pveceph destroyosd
```

Alias de pveceph osd destroy.

```
pveceph destroypool
```

Alias de pveceph pool destroy.

Sistema de archivos CephFS

pveceph fs create

```
pveceph fs create [OPTIONS]
```

Crea un sistema de archivos Ceph.

--add-storage <boolean> (default = 0) — Configura el CephFS creado como almacenamiento del clúster.

--name (?:^[^:/\s]+$) (default = cephfs) — Nombre del sistema de archivos Ceph.

--pg_num <integer> (8 - 32768) (default = 128) — Número de placement groups para el pool de datos subyacente. El pool de metadatos utilizará una cuarta parte de este valor.

pveceph fs destroy

```
pveceph fs destroy <name> [OPTIONS]
```

Destruye un sistema de archivos Ceph. Rechaza la operación si alguna entrada de almacenamiento PVE de tipo cephfs todavía referencia el sistema y no está deshabilitada. Opcionalmente puede eliminar también las entradas de almacenamiento y/o los pools de metadatos y datos.

<name>: <string> — Nombre del sistema de archivos Ceph.

--remove-pools <boolean> (default = 0) — Elimina los pools de metadatos y datos utilizados por este sistema de archivos.

--remove-storages <boolean> (default = 0) — Elimina los almacenamientos administrados por pveceph configurados para este sistema de archivos.

pveceph help

```
pveceph help [OPTIONS]
```

Muestra ayuda sobre el comando especificado.

--extra-args <array> — Muestra ayuda para un comando concreto.

--verbose <boolean> — Formato de salida detallado.

Inicialización e instalación

pveceph init

```
pveceph init [OPTIONS]
```

Crea la configuración inicial predeterminada de Ceph y prepara los enlaces simbólicos. Es idempotente: si ya existe una sección [global] en ceph.conf, se conservan fsid, auth y los valores predeterminados de pools, y la mayoría de parámetros se ignoran silenciosamente.

--cluster-network <string> — Declara una red de clúster separada; los OSD enrutarán por ella heartbeat, replicación de objetos y tráfico de recuperación.

NOTA: La opción --cluster-network requiere que también se especifique network.

--disable_cephx <boolean> (default = 0) — Deshabilita la autenticación cephx.

ADVERTENCIA: cephx protege frente a ataques man-in-the-middle. Solo considere deshabilitarlo si la red es privada.

--min_size <integer> (1 - 7) (default = 2) — Número mínimo de réplicas disponibles por objeto para permitir E/S.

--network <string> — Utiliza una red concreta para todo el tráfico relacionado con Ceph.

--pg_bits <integer> (6 - 14) (default = 6) — Bits de placement groups usados para definir el número predeterminado de PG. Esta opción está obsoleta en versiones recientes de Ceph.

--size <integer> (1 - 7) (default = 3) — Número objetivo de réplicas por objeto.

pveceph install

```
pveceph install [OPTIONS]
```

Instala los paquetes relacionados con Ceph.

--allow-experimental <boolean> (default = 0) — Permite versiones experimentales. Utilícelo con precaución.

--repository <enterprise | manual | no-subscription | test> (default = enterprise) — Repositorio Ceph a utilizar. manual no configura ningún repositorio; se usa cuando el host no puede acceder a repositorios públicos, por ejemplo con Proxmox Offline Mirror. Debe existir un repositorio configurado manualmente que contenga los paquetes de la versión antes de comenzar.

--version <squid | tentacle> (default = tentacle) — Versión de Ceph que se instalará.

```
pveceph lspools
```

Alias de pveceph pool ls.

Metadata Server (MDS)

pveceph mds create

```
pveceph mds create [OPTIONS]
```

Crea un Ceph Metadata Server (MDS).

--hotstandby <boolean> (default = 0) — Hace que un daemon ceph-mds sondee y reproduzca el log de un MDS activo. Acelera el cambio ante fallo del MDS, pero consume más recursos inactivos.

--name [a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])? (default = nodename) — ID del MDS. Si se omite, utiliza el nombre del nodo.

pveceph mds destroy

```
pveceph mds destroy <name>
```

Destruye un Ceph Metadata Server.

<name>: [a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])? — Nombre (ID) del MDS.

Ceph Manager (MGR)

pveceph mgr create

```
pveceph mgr create [OPTIONS]
```

Crea un Ceph Manager.

--id [a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])? (default = nodename) — ID del manager. Si se omite, se utiliza el nombre del nodo.

pveceph mgr destroy

```
pveceph mgr destroy <id>
```

Destruye un Ceph Manager.

<id>: [a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])? — ID del manager.

Ceph Monitor (MON)

pveceph mon create

```
pveceph mon create [OPTIONS]
```

Crea un Ceph Monitor. Al crear el primer monitor también crea automáticamente un Manager.

--mon-address <string> — Sobrescribe la dirección o direcciones IP de monitor detectadas automáticamente. Deben pertenecer a la red pública de Ceph.

--monid [a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])? (default = nodename) — ID del monitor. Si se omite, usa el nombre del nodo.

pveceph mon destroy

```
pveceph mon destroy <monid>
```

Destruye un Ceph Monitor. Rechaza eliminar el último monitor del clúster. No destruye ningún Manager del mismo nodo; este debe eliminarse por separado.

<monid>: [a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])? — ID del monitor.

OSD

pveceph osd create

```
pveceph osd create <dev> [OPTIONS]
```

Crea un OSD.

<dev>: <string> — Nombre del dispositivo de bloques.

--crush-device-class <string> — Establece la clase de dispositivo del OSD en CRUSH.

--db_dev <string> — Dispositivo de bloques para block.db.

--db_dev_size <number> (1 - N) — Tamaño en GiB para block.db. Requiere db_dev.

--encrypted <boolean> (default = 0) — Activa el cifrado del OSD.

--osds-per-device <integer> (1 - N) — Número de servicios OSD por dispositivo físico. Solo resulta útil en dispositivos NVMe rápidos para aprovechar mejor su rendimiento. Es mutuamente excluyente con db_dev y wal_dev.

--wal_dev <string> — Dispositivo de bloques para block.wal.

--wal_dev_size <number> (0.5 - N) — Tamaño en GiB para block.wal. Requiere wal_dev.

pveceph osd destroy

```
pveceph osd destroy <osdid> [OPTIONS]
```

Destruye un OSD.

<osdid>: <integer> — ID del OSD.

--cleanup <boolean> (default = 0) — Si se activa, también destruye los volúmenes lógicos subyacentes mediante ceph-volume lvm zap --destroy, elimina el volumen físico del volume group con pvremove y borra particiones journal/block.db/block.wal que pudieran quedar de OSDs filestore. Sin esta opción, LVs y particiones permanecen intactos para inspección.

pveceph osd details

```
pveceph osd details <osdid> [OPTIONS] [FORMAT_OPTIONS]
```

Obtiene detalles del OSD.

<osdid>: <string> — ID del OSD.

--verbose <boolean> (default = 0) — Muestra información detallada, equivalente al formato de salida json-pretty.

Pools Ceph

pveceph pool create

```
pveceph pool create <name> [OPTIONS]
```

Crea un pool Ceph.

<name>: (?:^[^:/\s]+$) — Nombre del pool. Debe ser único.

--add_storages <boolean> (default = 0) — Configura almacenamiento para VMs y CT utilizando el nuevo pool. El valor predeterminado es false para pools replicados y true para pools Erasure Coded, ya que estos suelen ser útiles cuando se integran como almacenamiento.

--application <cephfs | rbd | rgw> (default = rbd) — Aplicación del pool.

--crush_rule <string> — Regla utilizada para mapear la ubicación de objetos en el clúster.

--erasure-coding k=<integer>,m=<integer>[,device-class=<class>][,failure-domain=<domain>][,profile=<profile>] — Crea un pool Erasure Coded para RBD acompañado de un pool replicado para metadatos. En EC, size, min_size y crush_rule se aplican al pool de metadatos.

--min_size <integer> (1 - 7) (default = 2) — Número mínimo de réplicas por objeto.

--pg_autoscale_mode <off | on | warn> (default = warn) — Modo de escalado automático de PG del pool.

--pg_num <integer> (1 - 32768) (default = 128) — Número de placement groups.

--pg_num_min <integer> (-N - 32768) — Número mínimo de placement groups.

--size <integer> (1 - 7) (default = 3) — Número de réplicas por objeto.

--target_size ^(\d+(\.\d+)?)([KMGT])?$ — Tamaño objetivo estimado del pool para el PG autoscaler.

--target_size_ratio <number> — Proporción objetivo estimada del pool para el PG autoscaler.

pveceph pool destroy

```
pveceph pool destroy <name> [OPTIONS]
```

Destruye un pool.

<name>: <string> — Nombre del pool. Debe ser único.

--force <boolean> (default = 0) — Si es true, destruye el pool incluso si está en uso.

--remove_ecprofile <boolean> (default = 1) — Elimina el perfil Erasure Code. De forma predeterminada es true cuando aplica.

--remove_storages <boolean> (default = 0) — Elimina todos los almacenamientos administrados por pveceph configurados para este pool.

pveceph pool get

```
pveceph pool get <name> [OPTIONS] [FORMAT_OPTIONS]
```

Muestra el estado actual del pool.

<name>: <string> — Nombre del pool. Debe ser único.

--verbose <boolean> (default = 0) — Muestra datos adicionales, como estadísticas.

pveceph pool ls

```
pveceph pool ls [FORMAT_OPTIONS]
```

Lista todos los pools y sus ajustes configurables mediante endpoints POST/PUT.

pveceph pool set

```
pveceph pool set <name> [OPTIONS]
```

Modifica los ajustes de un pool.

<name>: (?:^[^:/\s]+$) — Nombre del pool. Debe ser único.

--application <cephfs | rbd | rgw> — Aplicación del pool.

--crush_rule <string> — Regla de ubicación de objetos en el clúster.

--min_size <integer> (1 - 7) — Número mínimo de réplicas por objeto.

--pg_autoscale_mode <off | on | warn> — Modo de escalado automático de PG.

--pg_num <integer> (1 - 32768) — Número de placement groups.

--pg_num_min <integer> (-N - 32768) — Número mínimo de placement groups.

--size <integer> (1 - 7) — Número de réplicas por objeto.

--target_size ^(\d+(\.\d+)?)([KMGT])?$ — Tamaño objetivo estimado para el PG autoscaler.

--target_size_ratio <number> — Proporción objetivo estimada para el PG autoscaler.

Administración global del servicio

pveceph purge

```
pveceph purge [OPTIONS]
```

Destruye los datos y archivos de configuración relacionados con Ceph.

--crash <boolean> — Elimina también los logs de fallos de Ceph en /var/lib/ceph/crash.

--logs <boolean> — Elimina también los logs de Ceph en /var/log/ceph.

pveceph start

```
pveceph start [OPTIONS]
```

Inicia servicios Ceph.

--service (ceph|mon|mds|osd|mgr)(\.[a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])?)? (default = ceph.target) — Nombre del servicio Ceph.

pveceph status

```
pveceph status
```

Obtiene el estado de Ceph.

pveceph stop

```
pveceph stop [OPTIONS]
```

Detiene servicios Ceph.

--service (ceph|mon|mds|osd|mgr)(\.[a-zA-Z0-9]([a-zA-Z0-9\-]*[a-zA-Z0-9])?)? (default = ceph.target) — Nombre del servicio Ceph.

Fin de la Parte A2

Esta entrega cubre íntegramente A.4 pvesubscription, A.5 pveperf y A.6 pveceph. La siguiente parte comenzará en A.7 pvenode - Proxmox VE Node Management.

CRITERIO DE TRADUCCIÓN: Los nombres de comandos, argumentos, opciones, expresiones regulares, rutas y valores válidos se mantienen sin traducir para que puedan copiarse y utilizarse directamente. Las descripciones se traducen al castellano respetando el contenido de la referencia original.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A3

A.7 pvenode - Gestión de nodos de Proxmox VE

ACME · certificados · configuración del nodo · migración · arranque/parada · tareas · Wake-on-LAN

Certificados	Nodo	Operaciones	Seguimiento
ACME + certificados	config get/set	migrateall + startall + stopall	task list/log/status + WoL

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, parámetros, rutas, expresiones regulares e identificadores.

A.7 pvenode - Gestión de nodos de Proxmox VE

```
pvenode <COMMAND> [ARGS] [OPTIONS]
```

La herramienta pvenode agrupa operaciones de administración específicas de un nodo Proxmox VE. Esta sección reproduce la referencia de comandos del manual original.

Cuentas ACME

`pvenode acme account deactivate [<name>]`

Desactiva una cuenta ACME existente en la autoridad certificadora (CA).

`<name>: <name> (default = default)`
Nombre del archivo de configuración de la cuenta ACME.

`pvenode acme account info [<name>] [FORMAT_OPTIONS]`

Devuelve la información de una cuenta ACME existente.

`<name>: <name> (default = default)`
Nombre del archivo de configuración de la cuenta ACME.

`pvenode acme account list`

Muestra el índice de cuentas ACME.

`pvenode acme account register [<name>] {<contact>} [OPTIONS]`

Registra una nueva cuenta ACME en una CA compatible.

`<name>: <name> (default = default)`
Nombre del archivo de configuración de la cuenta ACME.

`<contact>: <string>`
Direcciones de correo electrónico de contacto.

`--directory ^https?://.*`
URL del endpoint de directorio de la autoridad certificadora ACME.

`pvenode acme account update [<name>] [OPTIONS]`

Actualiza en la CA la información de una cuenta ACME existente. Si no se especifica información nueva, se fuerza una actualización/refresco.

`<name>: <name> (default = default)`
Nombre del archivo de configuración de la cuenta ACME.

`--contact <string>`
Direcciones de correo electrónico de contacto.

Certificados ACME

`pvenode acme cert order [OPTIONS]`

Solicita un nuevo certificado a una CA compatible con ACME.

`--force <boolean> (default = 0)`
Sobrescribe un certificado personalizado existente.

`pvenode acme cert renew [OPTIONS]`

Renueva el certificado existente mediante la CA.

`--force <boolean> (default = 0)`
Fuerza la renovación aunque falten más de 30 días para la expiración.

`pvenode acme cert revoke`

Revoca el certificado existente en la CA.

Plugins ACME

`pvenode acme plugin add <type> <id> [OPTIONS]`

Añade una configuración de plugin ACME.

`<type>: <dns | standalone>`
Tipo de challenge ACME.

`<id>: <string>`
Nombre/ID del plugin ACME.

Valores admitidos para --api:

```
--api <1984hosting | acmedns | acmeproxy | active24 | ad | ali | alviy | anx | artfiles | arvan | aurora | autodns | aws |
azion | azure | beget | bookmyname | bunny | cf | clouddns | cloudns | cn | conoha | constellix | cpanel | curanet | cyon | da
```

```
| ddnss | desec | df | dgon | dnsexit | dnshome | dnsimple | dnsservices | doapi | domeneshop | dp | dpi | dreamhost | duckdns
| durabledns | dyn | dynu | dynv6 | easydns | edgecenter | edgedns | euserv | exoscale | fornex | freedns | freemyip |
gandi_livedns | gcloud | gcore | gd | geoscaling | googledomains | he | he_ddns | hetzner | hetznercloud | hexonet | hostingde
| huaweicloud | infoblox | infomaniak | internetbs | inwx | ionos | ionos_cloud | ipv64 | ispconfig | jd | joker | kappernet |
kas | kinghost | knot | la | leaseweb | lexicon | limacity | linode | linode_v4 | loopia | lua | maradns | me | miab | mijnhost
| misaka | myapi | mydevil | mydnsjp | mythic_beasts | namecheap | namecom | namesilo | nanelo | nederhost | neodigit | netcup
| netlify | nic | njalla | nm | nsd | nsone | nsupdate | nw | oci | omglol | one | online | openprovider | openprovider_rest |
openstack | opnsense | ovh | pdns | pleskxml | pointhq | porkbun | rackcorp | rackspace | rage4 | rcode0 | regru | scaleway |
schlundtech | selectel | selfhost | servercow | simply | spaceship | technitium | tele3 | tencent | timeweb | transip | udr |
ultra | unoeuro | variomedia | veesp | vercel | vscale | vultr | websupport | west_cn | world4you | yandex360 | yc | zilore |
zone | zoneedit | zonomi>
```

Nombre del plugin de API.

--data File with one key-value pair per line, will be base64url encode for storage in plugin config.
Datos del plugin DNS; se almacenan codificados en Base64.

--disable <boolean>
Marca para deshabilitar la configuración.

--nodes <string>
Lista de nombres de nodos del clúster.

--validation-delay <integer> (0 - 172800) (default = 30)
Retardo adicional, en segundos, antes de solicitar la validación. Permite adaptarse a TTL DNS largos.

pvenode acme plugin config <id> [FORMAT_OPTIONS]

Obtiene la configuración de un plugin ACME.

<id>: <string>
Identificador único de la instancia del plugin ACME.

pvenode acme plugin list [OPTIONS] [FORMAT_OPTIONS]

Muestra el índice de plugins ACME.

--type <dns | standalone>
Limita la lista a plugins ACME de un tipo concreto.

pvenode acme plugin remove <id>

Elimina la configuración del plugin ACME.

<id>: <string>
Identificador único de la instancia del plugin ACME.

pvenode acme plugin set <id> [OPTIONS]

Actualiza la configuración de un plugin ACME.

<id>: <string>
Nombre/ID del plugin ACME.

Valores admitidos para --api:

```
--api <1984hosting | acmedns | acmeproxy | active24 | ad | ali | alviy | anx | artfiles | arvan | aurora | autodns | aws |
azion | azure | beget | bookmyname | bunny | cf | clouddns | cloudns | cn | conoha | constellix | cpanel | curanet | cyon | da
| ddnss | desec | df | dgon | dnsexit | dnshome | dnsimple | dnsservices | doapi | domeneshop | dp | dpi | dreamhost | duckdns
| durabledns | dyn | dynu | dynv6 | easydns | edgecenter | edgedns | euserv | exoscale | fornex | freedns | freemyip |
gandi_livedns | gcloud | gcore | gd | geoscaling | googledomains | he | he_ddns | hetzner | hetznercloud | hexonet | hostingde
| huaweicloud | infoblox | infomaniak | internetbs | inwx | ionos | ionos_cloud | ipv64 | ispconfig | jd | joker | kappernet |
kas | kinghost | knot | la | leaseweb | lexicon | limacity | linode | linode_v4 | loopia | lua | maradns | me | miab | mijnhost
| misaka | myapi | mydevil | mydnsjp | mythic_beasts | namecheap | namecom | namesilo | nanelo | nederhost | neodigit | netcup
| netlify | nic | njalla | nm | nsd | nsone | nsupdate | nw | oci | omglol | one | online | openprovider | openprovider_rest |
openstack | opnsense | ovh | pdns | pleskxml | pointhq | porkbun | rackcorp | rackspace | rage4 | rcode0 | regru | scaleway |
schlundtech | selectel | selfhost | servercow | simply | spaceship | technitium | tele3 | tencent | timeweb | transip | udr |
ultra | unoeuro | variomedia | veesp | vercel | vscale | vultr | websupport | west_cn | world4you | yandex360 | yc | zilore |
zone | zoneedit | zonomi>
```

Nombre del plugin de API.

--data File with one key-value pair per line, will be base64url encode for storage in plugin config.
Datos del plugin DNS; se almacenan codificados en Base64.

--delete <string>
Lista de ajustes que se desean eliminar.

--digest <string>
Impide cambios si el archivo de configuración actual tiene un digest diferente. Puede utilizarse para evitar modificaciones concurrentes.

--disable <boolean>
Marca para deshabilitar la configuración.

--nodes <string>
Lista de nombres de nodos del clúster.
--validation-delay <integer> (0 - 172800) (default = 30)
Retardo adicional, en segundos, antes de solicitar la validación.

Certificados personalizados del nodo

pvenode cert delete [<restart>]

Elimina la cadena de certificado personalizada y su clave.
<restart>: <boolean> (default = 0)
Reinicia pveproxy.

pvenode cert info [FORMAT_OPTIONS]

Obtiene información sobre los certificados del nodo.

pvenode cert set <certificates> [<key>] [OPTIONS] [FORMAT_OPTIONS]

Carga o actualiza una cadena de certificados personalizada y su clave.
<certificates>: <string>
Certificado o cadena de certificados codificados en PEM.
<key>: <string>
Clave privada codificada en PEM.
--force <boolean> (default = 0)
Sobrescribe archivos de certificado personalizados o ACME existentes.
--restart <boolean> (default = 0)
Reinicia pveproxy.

Configuración del nodo

pvenode config get [OPTIONS]

Obtiene opciones de configuración del nodo.
--property <acme | acmedomain0 | acmedomain1 | acmedomain2 | acmedomain3 | acmedomain4 | acmedomain5 | ballooning-target | description | location | startall-onboot-delay | wakeonlan> (default = all)
Devuelve únicamente una propiedad concreta de la configuración del nodo.

pvenode config set [OPTIONS]

Establece opciones de configuración del nodo.
--acme [account=<name>] [,domains=<domain[;domain;...]>]
Ajustes ACME específicos del nodo.
--acmedomain[n] [domain=]<domain> [,alias=<domain>] [,plugin=<name of the plugin configuration>]
Dominio ACME y plugin de validación.
--ballooning-target <integer> (0 - 100) (default = 80)
Objetivo de uso de RAM para ballooning, expresado como porcentaje de la memoria total.
--delete <string>
Lista de ajustes que se desean eliminar.
--description <string>
Descripción del nodo. Se muestra en el panel de notas del nodo en la interfaz web y se guarda como comentario en el archivo de configuración.
--digest <string>
Impide cambios si el archivo de configuración actual tiene un digest SHA1 diferente. Puede utilizarse para evitar modificaciones concurrentes.
--location latitude=<number> ,longitude=<number> [,name=<name>]
Ubicación del nodo. Sustituye el valor predeterminado de la configuración del Datacenter.
--startall-onboot-delay <integer> (0 - 300) (default = 0)
Retardo inicial, en segundos, antes de iniciar todos los invitados virtuales configurados con on-boot.
--wakeonlan [mac=]<MAC address> [,bind-interface=<bind interface>] [,broadcast-address=<IPv4 broadcast address>]
Configuración Wake-on-LAN específica del nodo.

Ayuda y operaciones globales del nodo

pvenode help [OPTIONS]

Muestra ayuda sobre un comando determinado.

--extra-args <array>
Muestra ayuda para un comando específico.
--verbose <boolean>
Formato de salida detallado.

pvenode migrateall <target> [OPTIONS]

Migra todas las máquinas virtuales y contenedores.
<target>: <string>
Nodo de destino.
--max-workers <integer> (1 - 64)
Número máximo de trabajos de migración en paralelo. Si no se define, utiliza max_workers de datacenter.cfg. Uno de los dos debe estar establecido.
--maxworkers <integer> (1 - 64)
Opción obsoleta. Equivale a max-workers; debe utilizarse --max-workers.
--vms <string>
Considera únicamente los invitados con estos IDs.
--with-local-disks <boolean>
Habilita la migración en vivo del almacenamiento para discos locales.

pvenode startall [OPTIONS]

Inicia todas las máquinas virtuales y contenedores ubicados en este nodo; de forma predeterminada, únicamente los que tienen onboot=1.
--force <boolean> (default = off)
Emite la orden de inicio aunque el invitado no tenga onboot configurado o esté desactivado.
--max-workers <integer> (1 - 64)
Número máximo de tareas concurrentes. Si no se define, usa max_workers de datacenter.cfg; si tampoco existe, utiliza los hilos de CPU disponibles con un máximo de 8.
--vms <string>
Considera únicamente los invitados de esta lista de VMIDs separados por comas.

pvenode stopall [OPTIONS]

Detiene todas las máquinas virtuales y contenedores.
--force-stop <boolean> (default = 1)
Fuerza una parada dura después de superar el timeout.
--max-workers <integer> (1 - 64)
Número máximo de tareas concurrentes. Si no se define, usa max_workers de datacenter.cfg; si tampoco existe, utiliza los hilos de CPU disponibles con un máximo de 8.
--timeout <integer> (0 - 7200) (default = 180)
Tiempo de espera para cada tarea de apagado del invitado. Según force-stop, al agotarse el tiempo el apagado se aborta o se fuerza una parada dura.
--vms <string>
Considera únicamente los invitados con estos IDs.

Tareas del nodo

pvenode task list [OPTIONS] [FORMAT_OPTIONS]

Lee la lista de tareas de un nodo, incluidas las finalizadas.
--errors <boolean> (default = 0)
Muestra únicamente tareas con estado ERROR.
--limit <integer> (0 - N) (default = 50)
Limita el número de tareas devueltas.
--since <integer>
Muestra únicamente tareas posteriores a este UNIX epoch.
--source <active | all | archive> (default = archive)
Lista tareas archivadas, activas o todas.
--start <integer> (0 - N) (default = 0)
Empieza a listar tareas desde este desplazamiento.
--statusfilter <string>
Lista de estados de tarea que deben devolverse.
--typefilter <string>
Muestra únicamente tareas de este tipo, por ejemplo vzstart o vzdump.

--until <integer>
Muestra únicamente tareas anteriores a este UNIX epoch.
--userfilter <string>
Muestra únicamente tareas pertenecientes a este usuario.
--vmid <integer> (100 - 999999999)
Muestra únicamente tareas de esta VM.

pvenode task log <upid> [OPTIONS]

Lee el registro de una tarea.
<upid>: <string>
ID único de la tarea.
--download <boolean>
Indica si debe descargarse el archivo de log. No puede utilizarse junto con otros parámetros.
--start <integer> (0 - N) (default = 0)
Línea desde la que se comienza a leer el registro de la tarea.

pvenode task status <upid> [FORMAT_OPTIONS]

Lee el estado de una tarea.
<upid>: <string>
ID único de la tarea.

Wake-on-LAN

pvenode wakeonlan <node>

Intenta despertar un nodo mediante un paquete Wake-on-LAN.
<node>: <string>
Nodo de destino del paquete Wake-on-LAN.

CRITERIO DE TRADUCCIÓN: Los comandos, nombres de opciones, expresiones regulares, nombres de archivo y valores válidos se han mantenido tal como aparecen en la referencia original. Las descripciones se han traducido al castellano sin añadir comportamiento no documentado.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A · PARTE A4

Interfaz de línea de comandos

A.8 pvesh · A.9 qm (inicio y qm create)

API	Clonado	Cloud-Init	Creación de VM
pvesh	qm clone	dump · pending · update	qm create

Basado en Proxmox VE Administration Guide, Release 9.2.4

Los comandos, parámetros, valores y expresiones se conservan literalmente.

A.8 pvesh - Interfaz shell para la API de Proxmox VE

```
pvesh <COMMAND> [ARGS] [OPTIONS]
```

pvesh create

```
pvesh create <api_path> [OPTIONS] [FORMAT_OPTIONS]
```

Ejecuta una llamada API POST sobre <api_path>.

<api_path>: <string> - Ruta de la API.

--noproxy <boolean> - Desactiva el proxy automático.

pvesh delete

```
pvesh delete <api_path> [OPTIONS] [FORMAT_OPTIONS]
```

Ejecuta una llamada API DELETE sobre <api_path>.

<api_path>: <string> - Ruta de la API.

--noproxy <boolean> - Desactiva el proxy automático.

pvesh get

```
pvesh get <api_path> [OPTIONS] [FORMAT_OPTIONS]
```

Ejecuta una llamada API GET sobre <api_path>.

<api_path>: <string> - Ruta de la API.

--noproxy <boolean> - Desactiva el proxy automático.

pvesh help

```
pvesh help [OPTIONS]
```

Obtiene ayuda sobre un comando concreto.

--extra-args <array> - Muestra ayuda para un comando específico.

--verbose <boolean> - Formato de salida detallado.

pvesh ls

```
pvesh ls <api_path> [OPTIONS] [FORMAT_OPTIONS]
```

Lista objetos hijos en <api_path>.

<api_path>: <string> - Ruta de la API.

--noproxy <boolean> - Desactiva el proxy automático.

pvesh set

```
pvesh set <api_path> [OPTIONS] [FORMAT_OPTIONS]
```

Ejecuta una llamada API PUT sobre <api_path>.

<api_path>: <string> - Ruta de la API.

--noproxy <boolean> - Desactiva el proxy automático.

pvesh usage

```
pvesh usage <api_path> [OPTIONS]
```

Muestra información de uso de la API para <api_path>.

<api_path>: <string> - Ruta de la API.

--command <create | delete | get | set> - Comando API.

--returns <boolean> - Incluye el esquema de los datos devueltos.

--verbose <boolean> - Formato de salida detallado.

A.9 qm - Gestor de máquinas virtuales QEMU/KVM

```
qm <COMMAND> [ARGS] [OPTIONS]
```

qm agent

Alias de qm guest cmd.

qm cleanup

```
qm cleanup <vmid> <clean-shutdown> <guest-requested>
```

Limpia recursos como dispositivos tap, vGPU y otros elementos después de que una VM se apague, se bloquee o termine.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

<clean-shutdown>: <boolean> - Indica si QEMU se apagó de forma limpia.

<guest-requested>: <boolean> - Indica si el apagado fue solicitado por el invitado o mediante QMP.

qm clone

```
qm clone <vmid> <newid> [OPTIONS]
```

Crea una copia de una máquina virtual o plantilla.

<vmid>: <integer> (100 - 999999999) - ID único de la VM origen.

<newid>: <integer> (100 - 999999999) - VMID para el clon.

--bwlimit <integer> (0 - N) - Sobrescribe el límite de ancho de banda de E/S en KiB/s; por defecto usa el límite de clonación del Datacenter o del almacenamiento.

--description <string> - Descripción de la nueva VM.

--format <qcow2 | raw | vmdk> - Formato de disco de destino. Solo válido para un clon completo; si el almacenamiento no lo admite, se usa su formato predeterminado.

--full <boolean> - Crea una copia completa de todos los discos. Siempre se hace al clonar una VM normal; para plantillas se intenta crear un clon enlazado de forma predeterminada.

--name <string> - Nombre de la nueva VM.

--pool <string> - Añade la nueva VM al pool indicado.

--snapname <string> - Nombre del snapshot.

--storage <storage ID> - Almacenamiento de destino para clon completo.

--target <string> - Nodo de destino. Solo permitido si la VM original está en almacenamiento compartido.

Cloud-Init

```
qm cloudinit dump <vmid> <type>
```

Obtiene la configuración Cloud-Init generada automáticamente.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

<type>: <meta | network | user> - Tipo de configuración.

```
qm cloudinit pending <vmid>
```

Obtiene la configuración Cloud-Init mostrando tanto los valores actuales como los pendientes.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

```
qm cloudinit update <vmid>
```

Regenera y modifica la unidad de configuración Cloud-Init.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

qm config

```
qm config <vmid> [OPTIONS]
```

Obtiene la configuración de la VM aplicando los cambios pendientes. Use current para consultar la configuración actualmente activa.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

--current <boolean> (default = 0) - Obtiene los valores actuales en lugar de los pendientes.

--snapshot <string> - Obtiene valores de configuración desde el snapshot indicado.

qm create

```
qm create <vmid> [OPTIONS]
```

Crea o restaura una máquina virtual.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

--acpi <boolean> (default = 1) - Activa o desactiva ACPI.

--affinity <string> - Lista de cores del host utilizados para ejecutar procesos del invitado, por ejemplo: 0,5,8-11.

--agent [enabled=]<1|0> [,freeze-fs=<1|0>] [,fstrim_cloned_disks=<1|0>] [,type=<virtio|isa>] - Activa o desactiva la comunicación con QEMU Guest Agent y sus propiedades.

--allow-ksm <boolean> (default = 1) - Permite que las páginas de memoria del invitado se combinen mediante KSM (Kernel Samepage Merging).

--amd-sev [type=]<sev-type> [,allow-smt=<1|0>] [,kernel-hashes=<1|0>] [,no-debug=<1|0>] [,no-key-sharing=<1|0>] - Características Secure Encrypted Virtualization (SEV) de CPUs AMD.

--arch <aarch64 | x86_64> - Arquitectura del procesador virtual. Por defecto se usa la arquitectura del host.

--archive <string> - Archivo de backup. Puede ser una ruta de sistema a un .tar o .vma (use - para leer de stdin) o un identificador de volumen de backup de un almacenamiento Proxmox.

--args <string> - Argumentos arbitrarios pasados a kvm.

--audio0 device=<ich9-intel-hda|intel-hda|AC97> [,driver=<spice|none>] - Configura un dispositivo de audio, útil junto con QXL/Spice.

--autostart <boolean> (default = 0) - Reinicio automático después de un fallo; actualmente se ignora.

--balloon <integer> (0 - N) - Cantidad de RAM objetivo para la VM en MiB. El driver balloon se activa por defecto salvo que se establezca a cero.

--bios <ovmf | seabios> (default = seabios) - Selecciona la implementación BIOS.

--boot [[legacy=]<[acdn]{1,4}>] [,order=<device[;device...]>] - Especifica el orden de arranque. Use la subpropiedad order=; el uso sin clave o con legacy= está obsoleto.

--bootdisk (ide|sata|scsi|virtio)\d+ - Activa el arranque desde el disco indicado. Obsoleto: use boot: order=foo;bar.

--bwlimit <integer> (0 - N) - Sobrescribe el límite de ancho de banda de E/S en KiB/s; por defecto utiliza el límite de restauración del Datacenter o del almacenamiento.

--cdrom <volume> - Alias de la opción -ide2.

--cicustom [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>] - Cloud-Init: especifica archivos personalizados que sustituyen a los generados automáticamente al iniciar.

--cipassword <password> - Cloud-Init: contraseña a asignar al usuario. No se recomienda; use claves SSH. Versiones antiguas de Cloud-Init no admiten contraseñas con hash.

--citype <configdrive2 | nocloud | opennebula> - Formato de configuración Cloud-Init. El valor por defecto depende de ostype: nocloud para Linux y configdrive2 para Windows.

--ciupgrade <boolean> (default = 1) - Cloud-Init: realiza una actualización automática de paquetes después del primer arranque.

--ciuser <string> - Cloud-Init: usuario al que se aplican claves SSH y contraseña, en lugar del usuario predeterminado de la imagen.

--cores <integer> (1 - N) (default = 1) - Número de cores por socket.

--cpu [[cputype=]<string>] [,flags=<+FLAG[;-FLAG...]>] [,guest-phys-bits=<integer>] [,hidden=<1|0>] [,hv-vendor-id=<vendor-id>] [,level=<integer>] [,phys-bits=<8-64|host>] [,reported-model=<enum>] - Tipo de CPU emulada.

--cpulimit <number> (0 - 128) (default = 0) - Límite de uso de CPU.

--cpuunits <integer> (1 - 262144) (default = cgroup v1: 1024, cgroup v2: 100) - Peso de CPU de una VM; con cgroup v2 se limita al intervalo [1, 10000].

--description <string> - Descripción de la VM mostrada en el resumen web. Se guarda como comentario en el archivo de configuración.

--efidisk0 [file=]<volume> [,efitype=<2m|4m>] [,format=<enum>] [,import-from=<source volume>] [,ms-cert=<enum>] [,pre-enrolled-keys=<1|0>] [,size=<DiskSize>] - Configura el disco que almacena las variables EFI. Para asignar un volumen nuevo use STORAGE_ID:SIZE_IN_GiB; aquí SIZE_IN_GiB se ignora y se copian las variables EFI predeterminadas. Para importar use STORAGE_ID:0 y import-from.

--force <boolean> - Permite sobrescribir una VM existente. Requiere archive.

--freeze <boolean> - Congela la CPU al iniciar; use el comando c del monitor para comenzar la ejecución.

--ha-managed <boolean> (default = 0) - Añade la VM como recurso HA después de crearla.

--hookscript <string> - Script ejecutado durante distintas fases del ciclo de vida de la VM.

--hostpci[n] [[host=]<HOSTPCIID[;HOSTPCIID2...]>] [,device-id=<hex id>] [,driver=<vfio|keep>] [,legacy-igd=<1|0>] [,mapping=<mapping-id>] [,mdev=<string>] [,pcie=<1|0>] [,rombar=<1|0>] [,romfile=<string>] [,sub-device-id=<hex id>] [,sub-vendor-id=<hex id>] [,vendor-id=<hex id>] [,x-vga=<1|0>] - Mapea dispositivos PCI del host dentro del invitado.

--hotplug <string> (default = network,disk,usb) - Activa selectivamente hotplug para network, disk, cpu, memory, usb y cloudinit. Use 0 para desactivarlo. El valor 1 equivale a network,disk,usb.

--hugepages <1024 | 2 | any> - Activa memoria hugepages. El valor indica MiB; any usa páginas de 1 GiB si es posible y cae a 2 MiB en caso contrario.

--ide[n] [file=]<volume> [...] - Usa un volumen como disco IDE o CD-ROM (n de 0 a 3). Use STORAGE_ID:SIZE_IN_GiB para crear volumen nuevo, o STORAGE_ID:0 con import-from para importar uno existente.

--import-working-storage <storage ID> - Almacenamiento basado en archivos con content-type images usado como almacenamiento intermedio durante una importación. Por defecto se usa el almacenamiento origen.

--intel-tdx [type=]<tdx-type> ,attestation=<1|0> [,vsock-cid=<integer>] [,vsock-port=<integer>] - Características Trusted Domain Extension (TDX) de CPUs Intel.

--ipconfig[n] [gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>] - Cloud-Init: especifica IP y gateways de la interfaz. Las IP usan CIDR. dhcp activa DHCP; para IPv6 auto activa autoconfiguración si n estado. Si no se especifica IP y Cloud-Init está activo, IPv4 usa dhcp por defecto.

--ivshmem size=<integer> [,name=<string>] - Memoria compartida entre VMs; útil para comunicación directa entre VMs o con el host.

--keephugepages <boolean> (default = 0) - Junto con hugepages, evita liberar las hugepages al apagar la VM, permitiendo reutilizarlas en arranques posteriores.

--keyboard <da | de | de-ch | en-gb | en-us | es | fi | fr | fr-be | fr-ca | fr-ch | hu | is | it | ja | lt | mk | nl | no | pl | pt | pt-br | sl | sv | tr> - Distribución de teclado para VNC. Normalmente es mejor configurarla dentro del sistema invitado.

--kvm <boolean> (default = 1) - Activa o desactiva la virtualización hardware KVM.

--live-restore <boolean> - Inicia la VM inmediatamente mientras la importación o restauración continúa en segundo plano.

--localtime <boolean> - Configura el RTC en hora local. Se activa por defecto si ostype identifica un Windows.

--lock <backup | clone | create | migrate | rollback | snapshot | snapshot-delete | suspended | suspending> - Bloquea o desbloquea la VM.

--machine [[type=]<machine type>] [,aw-bits=<number>] [,enable-s3=<1|0>] [,enable-s4=<1|0>] [,viommu=<intel|virtio>] - Especifica la máquina QEMU.

--memory [current=]<integer> - Propiedades de memoria.

--migrate_downtime <number> (0 - N) (default = 0.1) - Tiempo máximo tolerado de parada, en segundos, durante migraciones. Si no converge, el límite se aumenta progresivamente hasta un máximo de 2000 s.

--migrate_speed <integer> (0 - N) (default = 0) - Velocidad máxima de migración en MB/s. 0 significa sin límite.

--name <string> - Nombre de la VM, usado en la interfaz web.

--nameserver <string> - Cloud-Init: servidor DNS. Si no se define ni searchdomain ni nameserver, create usa automáticamente la configuración del host.

--net[n] [model=]<enum> [,bridge=<bridge>] [,firewall=<1|0>] [,host-tunnel=<1|0>] [,link_down=<1|0>] [,macaddr=<XX:XX:XX:XX:XX:XX>] [,mtu=<integer>] [,queues=<integer>] [,rate=<number>] [,tag=<integer>] [,trunks=<vlanid[;vlanid...]>] [,<model>=<macaddr>] - Especifica dispositivos de red.

--numa <boolean> (default = 0) - Activa o desactiva NUMA.

--numa[n] cpus=<id[-id];...> [,hostnodes=<id[-id];...>] [,memory=<number>] [,policy=<preferred|bind|interleave>] - Topología NUMA.

--onboot <boolean> (default = 0) - Indica si la VM debe arrancar durante el inicio del sistema.

--ostype <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> (default = other) - Sistema operativo invitado.

--parallel[n] /dev/parport\d+|/dev/usb/lp\d+ - Mapea dispositivos paralelos del host, n de 0 a 2.

--pool <string> - Añade la VM al pool indicado.

--protection <boolean> (default = 0) - Activa el flag de protección de la VM, deshabilitando las operaciones de eliminar VM y eliminar disco.

--reboot <boolean> (default = 1) - Permite el reinicio. Si vale 0, la VM finaliza al reiniciarse.

--rng0 [source=]</dev/urandom|/dev/random|/dev/hwrng> [,max_bytes=<integer>] [,period=<integer>] - Configura un generador de números aleatorios basado en VirtIO.

--sata[n] [file=]<volume> [...] - Usa un volumen como disco SATA o CD-ROM (n de 0 a 5). Use STORAGE_ID:SIZE_IN_GiB para crear uno nuevo o STORAGE_ID:0 con import-from para importar.

--scsi[n] [file=]<volume> [...] - Usa un volumen como disco SCSI o CD-ROM (n de 0 a 30). Use STORAGE_ID:SIZE_IN_GiB para crear uno nuevo o STORAGE_ID:0 con import-from para importar.

--scsihw <lsi | lsi53c810 | megasas | pvscsi | virtio-scsi-pci | virtio-scsi-single> (default = lsi) - Modelo de controlador SCSI.

--searchdomain <string> - Cloud-Init: dominios de búsqueda DNS. Si no se define ni searchdomain ni nameserver, create utiliza automáticamente la configuración del host.

--serial[n] (/dev/[^,]+|socket) - Crea un dispositivo serie dentro de la VM, n de 0 a 3.

--shares <integer> (0 - 50000) (default = 1000) - Peso de memoria para auto-ballooning. Cuanto mayor, más memoria recibe la VM respecto a otras. 0 desactiva auto-ballooning, gestionado por pvestatd.

--smbios1 [base64=<1|0>] [,family=<Base64 encoded string>] [,manufacturer=<Base64 encoded string>] [,product=<Base64 encoded string>] [,serial=<Base64 encoded string>] [,sku=<Base64 encoded string>] [,uuid=<UUID>] [,version=<Base64 encoded string>] - Especifica campos SMBIOS type 1.

--smp <integer> (1 - N) (default = 1) - Número de CPUs. El manual recomienda usar -sockets en su lugar.

--sockets <integer> (1 - N) (default = 1) - Número de sockets de CPU.

--spice_enhancements [foldersharing=<1|0>] [,videostreaming=<off|all|filter>] - Configura mejoras adicionales para SPICE.

--sshkeys <filepath> - Cloud-Init: configura claves SSH públicas, una por línea en formato OpenSSH.

--start <boolean> (default = 0) - Inicia la VM después de crearla correctamente.

--startdate (now | YYYY-MM-DD | YYYY-MM-DDTHH:MM:SS) (default = now) - Fecha inicial del RTC. Formatos válidos: now, 2006-06-17T16:01:21 o 2006-06-17.

--startup [[order=]\d+] [,up=\d+] [,down=\d+] - Comportamiento de arranque y apagado. order define el orden general; el apagado se realiza en orden inverso. up/down añaden retardos en segundos.

--storage <storage ID> - Almacenamiento predeterminado.

--tablet <boolean> (default = 1) - Activa o desactiva el dispositivo USB tablet.

--tags <string> - Etiquetas de la VM. Son solo metadatos.

--tdf <boolean> (default = 0) - Activa o desactiva la corrección de deriva temporal.

--template <boolean> (default = 0) - Activa o desactiva el modo plantilla.

--tpmstate0 [file=]<volume> [,format=<raw|qcow2|vmdk>] [,import-from=<source volume>] [,size=<DiskSize>] [,version=<v1.2|v2.0>] - Configura un disco para el estado TPM. El formato queda fijado a raw. STORAGE_ID:SIZE_IN_GiB crea volumen nuevo, aunque el tamaño se ignora y se usan 4 MiB. Use STORAGE_ID:0 con import-from para importar.

--unique <boolean> - Asigna una dirección Ethernet aleatoria y única. Requiere archive.

--unused[n] [file=]<volume> - Referencia a volúmenes no usados. Se utiliza internamente y no debe modificarse manualmente.

--usb[n] [[host=]<HOSTUSBDEVICE|spice>] [,mapping=<mapping-id>] [,usb3=<1|0>] - Configura un dispositivo USB. n va de 0 a 4; con machine >= 7.1 y ostype l26 o Windows > 7 puede llegar a 14.

--vcpus <integer> (1 - N) (default = 0) - Número de vCPU conectadas mediante hotplug.

--vga [[type=]<enum>] [,clipboard=<vnc>] [,memory=<integer>] - Configura el hardware VGA.

--virtio[n] [file=]<volume> [...] - Usa un volumen como disco VIRTIO (n de 0 a 15). Use STORAGE_ID:SIZE_IN_GiB para crear uno nuevo o STORAGE_ID:0 con import-from para importar.

--virtiofs[n] [dirid=]<mapping-id> [,cache=<enum>] [,direct-io=<1|0>] [,expose-acl=<1|0>] [,expose-xattr=<1|0>] - Configura el uso compartido de un directorio entre host e invitado mediante Virtio-fs.

--vmgenid <UUID> (default = 1 (autogenerated)) - Configura VM Generation ID. Use 1 para generarlo automáticamente al crear o actualizar; use 0 para desactivarlo explícitamente.

--vmstatestorage <storage ID> - Almacenamiento predeterminado para volúmenes o archivos de estado de la VM.

--watchdog [[model=]<i6300esb|ib700>] [,action=<enum>] - Crea un dispositivo watchdog virtual de hardware.

NOTA: Las opciones abreviadas con [...] en esta traducción conservan el encabezado técnico de la opción y su explicación original, pero evitan repetir listas de subparámetros idénticas ya visibles en el manual para IDE/SATA/SCSI/VIRTIO. No se alt era el significado funcional.

Fin de la Parte A4

Esta entrega cubre A.8 pvesh completo y el inicio de A.9 qm hasta el final de qm create. La siguiente parte continuará con qm delsnapshot, qm destroy, importación de discos y el resto de comandos de gestión de máquinas virtuales.

CRITERIO DE TRADUCCIÓN: Se han preservado literalmente los nombres de comandos, parámetros, valores y rutas. Las descripciones se han traducido al castellano siguiendo la organización del manual fuente.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A5

Interfaz de línea de comandos · qm (continuación)

Snapshots · destrucción · discos · EFI · QEMU Guest Agent

Snapshots	Discos	EFI	Guest Agent
delsnapshot · destroy	import · move · resize · unlink	enroll-efi-keys	cmd · exec · passwd

Basado en Proxmox VE Administration Guide, Release 9.2.4

Continuación exacta de A.9 qm. Los comandos y valores se conservan literalmente.

A.9 qm - Gestor de máquinas virtuales QEMU/KVM (continuación)

qm delsnapshot <vmid> <snapname> [OPTIONS]

Elimina un snapshot de una máquina virtual.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<snapname>: <string>
```

Nombre del snapshot.

```
--force <boolean>
```

Fuerza la eliminación de la referencia del archivo de configuración incluso si falla la eliminación de los snapshots de disco .

qm destroy <vmid> [OPTIONS]

Destruye la VM y todos los volúmenes utilizados o propiedad de ella. También elimina los permisos y reglas de firewall específicos de la VM.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
--destroy-unreferenced-disks <boolean> (default = 0)
```

Si se activa, destruye además todos los discos no referenciados por la configuración pero cuyo VMID coincida, buscando en todos los almacenamientos habilitados.

```
--purge <boolean>
```

Elimina el VMID de otras configuraciones, como trabajos de backup, replicación y HA.

```
--skiplock <boolean>
```

Ignora los locks. Solo root puede utilizar esta opción.

qm disk import <vmid> <source> <storage> [OPTIONS]

Importa una imagen de disco externa como disco no utilizado de una VM. El formato de la imagen debe ser compatible con qemu-img(1).

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<source>: <string>
```

Ruta de la imagen de disco que se desea importar.

```
<storage>: <storage ID>
```

ID del almacenamiento de destino.

```
--format <qcow2 | raw | vmdk>
```

Formato de destino.

```
--target-disk <efidisk0 | ide0 | ide1 | ide2 | ide3 | sata0 | sata1
| sata2 | sata3 | sata4 | sata5 | scsi0 | scsi1 | scsi10 | scsi11 |
scsi12 | scsi13 | scsi14 | scsi15 | scsi16 | scsi17 | scsi18 |
scsi19 | scsi2 | scsi20 | scsi21 | scsi22 | scsi23 | scsi24 |
scsi25 | scsi26 | scsi27 | scsi28 | scsi29 | scsi3 | scsi30 | scsi4
| scsi5 | scsi6 | scsi7 | scsi8 | scsi9 | tpmstate0 | unused0 |
unused1 | unused10 | unused100 | unused101 | unused102 | unused103 |
unused104 | unused105 | unused106 | unused107 | unused108 |
unused109 | unused11 | unused110 | unused111 | unused112 |
unused113 | unused114 | unused115 | unused116 | unused117 |
unused118 | unused119 | unused12 | unused120 | unused121 |
unused122 | unused123 | unused124 | unused125 | unused126 |
unused127 | unused128 | unused129 | unused13 | unused130 |
unused131 | unused132 | unused133 | unused134 | unused135 |
unused136 | unused137 | unused138 | unused139 | unused14 |
unused140 | unused141 | unused142 | unused143 | unused144 |
unused145 | unused146 | unused147 | unused148 | unused149 |
unused15 | unused150 | unused151 | unused152 | unused153 |
unused154 | unused155 | unused156 | unused157 | unused158 |
unused159 | unused16 | unused160 | unused161 | unused162 |
unused163 | unused164 | unused165 | unused166 | unused167 |
unused168 | unused169 | unused17 | unused170 | unused171 |
```

```
unused172 | unused173 | unused174 | unused175 | unused176 |
unused177 | unused178 | unused179 | unused18 | unused180 |
unused181 | unused182 | unused183 | unused184 | unused185 |
unused186 | unused187 | unused188 | unused189 | unused19 |
unused190 | unused191 | unused192 | unused193 | unused194 |
unused195 | unused196 | unused197 | unused198 | unused199 | unused2
| unused20 | unused200 | unused201 | unused202 | unused203 |
unused204 | unused205 | unused206 | unused207 | unused208 |
unused209 | unused21 | unused210 | unused211 | unused212 |
unused213 | unused214 | unused215 | unused216 | unused217 |
unused218 | unused219 | unused22 | unused220 | unused221 |
unused222 | unused223 | unused224 | unused225 | unused226 |
unused227 | unused228 | unused229 | unused23 | unused230 |
unused231 | unused232 | unused233 | unused234 | unused235 |
unused236 | unused237 | unused238 | unused239 | unused24 |
unused240 | unused241 | unused242 | unused243 | unused244 |
unused245 | unused246 | unused247 | unused248 | unused249 |
unused25 | unused250 | unused251 | unused252 | unused253 |
unused254 | unused255 | unused26 | unused27 | unused28 | unused29 |
unused3 | unused30 | unused31 | unused32 | unused33 | unused34 |
unused35 | unused36 | unused37 | unused38 | unused39 | unused4 |
unused40 | unused41 | unused42 | unused43 | unused44 | unused45 |
unused46 | unused47 | unused48 | unused49 | unused5 | unused50 |
unused51 | unused52 | unused53 | unused54 | unused55 | unused56 |
unused57 | unused58 | unused59 | unused6 | unused60 | unused61 |
unused62 | unused63 | unused64 | unused65 | unused66 | unused67 |
unused68 | unused69 | unused7 | unused70 | unused71 | unused72 |
unused73 | unused74 | unused75 | unused76 | unused77 | unused78 |
unused79 | unused8 | unused80 | unused81 | unused82 | unused83 |
unused84 | unused85 | unused86 | unused87 | unused88 | unused89 |
unused9 | unused90 | unused91 | unused92 | unused93 | unused94 |
unused95 | unused96 | unused97 | unused98 | unused99 | virtio0 |
virtio1 | virtio10 | virtio11 | virtio12 | virtio13 | virtio14 |
virtio15 | virtio2 | virtio3 | virtio4 | virtio5 | virtio6 |
virtio7 | virtio8 | virtio9>
```

Nombre del disco de configuración al que se importará el volumen, por ejemplo scsi1.

qm disk move <vmid> <disk> [<storage>] [OPTIONS]

Mueve un volumen a otro almacenamiento o incluso a otra VM.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<disk> <efidisk0 | ide0 | ide1 | ide2 | ide3 | sata0 | sata1
| sata2 | sata3 | sata4 | sata5 | scsi0 | scsi1 | scsi10 | scsi11 |
scsi12 | scsi13 | scsi14 | scsi15 | scsi16 | scsi17 | scsi18 |
scsi19 | scsi2 | scsi20 | scsi21 | scsi22 | scsi23 | scsi24 |
scsi25 | scsi26 | scsi27 | scsi28 | scsi29 | scsi3 | scsi30 | scsi4
| scsi5 | scsi6 | scsi7 | scsi8 | scsi9 | tpmstate0 | unused0 |
unused1 | unused10 | unused100 | unused101 | unused102 | unused103 |
unused104 | unused105 | unused106 | unused107 | unused108 |
unused109 | unused11 | unused110 | unused111 | unused112 |
unused113 | unused114 | unused115 | unused116 | unused117 |
unused118 | unused119 | unused12 | unused120 | unused121 |
unused122 | unused123 | unused124 | unused125 | unused126 |
unused127 | unused128 | unused129 | unused13 | unused130 |
unused131 | unused132 | unused133 | unused134 | unused135 |
unused136 | unused137 | unused138 | unused139 | unused14 |
unused140 | unused141 | unused142 | unused143 | unused144 |
unused145 | unused146 | unused147 | unused148 | unused149 |
unused15 | unused150 | unused151 | unused152 | unused153 |
unused154 | unused155 | unused156 | unused157 | unused158 |
unused159 | unused16 | unused160 | unused161 | unused162 |
unused163 | unused164 | unused165 | unused166 | unused167 |
unused168 | unused169 | unused17 | unused170 | unused171 |
unused172 | unused173 | unused174 | unused175 | unused176 |
unused177 | unused178 | unused179 | unused18 | unused180 |
unused181 | unused182 | unused183 | unused184 | unused185 |
unused186 | unused187 | unused188 | unused189 | unused19 |
unused190 | unused191 | unused192 | unused193 | unused194 |
unused195 | unused196 | unused197 | unused198 | unused199 | unused2
| unused20 | unused200 | unused201 | unused202 | unused203 |
unused204 | unused205 | unused206 | unused207 | unused208 |
unused209 | unused21 | unused210 | unused211 | unused212 |
unused213 | unused214 | unused215 | unused216 | unused217 |
unused218 | unused219 | unused22 | unused220 | unused221 |
unused222 | unused223 | unused224 | unused225 | unused226 |
unused227 | unused228 | unused229 | unused23 | unused230 |
unused231 | unused232 | unused233 | unused234 | unused235 |
unused236 | unused237 | unused238 | unused239 | unused24 |
unused240 | unused241 | unused242 | unused243 | unused244 |
unused245 | unused246 | unused247 | unused248 | unused249 |
unused25 | unused250 | unused251 | unused252 | unused253 |
unused254 | unused255 | unused26 | unused27 | unused28 | unused29 |
unused3 | unused30 | unused31 | unused32 | unused33 | unused34 |
unused35 | unused36 | unused37 | unused38 | unused39 | unused4 |
unused40 | unused41 | unused42 | unused43 | unused44 | unused45 |
unused46 | unused47 | unused48 | unused49 | unused5 | unused50 |
unused51 | unused52 | unused53 | unused54 | unused55 | unused56 |
```

```
unused57 | unused58 | unused59 | unused6 | unused60 | unused61 |
unused62 | unused63 | unused64 | unused65 | unused66 | unused67 |
unused68 | unused69 | unused7 | unused70 | unused71 | unused72 |
unused73 | unused74 | unused75 | unused76 | unused77 | unused78 |
unused79 | unused8 | unused80 | unused81 | unused82 | unused83 |
unused84 | unused85 | unused86 | unused87 | unused88 | unused89 |
unused9 | unused90 | unused91 | unused92 | unused93 | unused94 |
unused95 | unused96 | unused97 | unused98 | unused99 | virtio0 |
virtio1 | virtio10 | virtio11 | virtio12 | virtio13 | virtio14 |
virtio15 | virtio2 | virtio3 | virtio4 | virtio5 | virtio6 |
virtio7 | virtio8 | virtio9>
```

Disco que se desea mover.

```
<storage>: <storage ID>
```

Almacenamiento de destino.

```
--bwlimit <integer> (0 - N) (default = move limit from datacenter or storage config)
```

Sobrescribe el límite de ancho de banda de E/S, en KiB/s.

```
--delete <boolean> (default = 0)
```

Elimina el disco original después de una copia correcta. De forma predeterminada, el disco original se conserva como unused.

```
--digest <string>
```

Impide cambios si el archivo de configuración actual tiene un digest SHA1 diferente. Sirve para detectar modificaciones concurrentes.

```
--format <qcow2 | raw | vmdk>
```

Formato del disco de destino. Solo se utiliza al mover a otro almacenamiento. Si el almacenamiento no admite ese formato, se utiliza su formato predeterminado.

```
--target-digest <string>
```

Impide cambios si la configuración actual de la VM de destino tiene un digest SHA1 diferente. Permite detectar modificaciones concurrentes.

```
--target-disk <efidisk0 | ide0 | ide1 | ide2 | ide3 | sata0 | sata1
| sata2 | sata3 | sata4 | sata5 | scsi0 | scsi1 | scsi10 | scsi11 |
scsi12 | scsi13 | scsi14 | scsi15 | scsi16 | scsi17 | scsi18 |
scsi19 | scsi2 | scsi20 | scsi21 | scsi22 | scsi23 | scsi24 |
scsi25 | scsi26 | scsi27 | scsi28 | scsi29 | scsi3 | scsi30 | scsi4
| scsi5 | scsi6 | scsi7 | scsi8 | scsi9 | tpmstate0 | unused0 |
unused1 | unused10 | unused100 | unused101 | unused102 | unused103 |
unused104 | unused105 | unused106 | unused107 | unused108 |
unused109 | unused11 | unused110 | unused111 | unused112 |
unused113 | unused114 | unused115 | unused116 | unused117 |
unused118 | unused119 | unused12 | unused120 | unused121 |
unused122 | unused123 | unused124 | unused125 | unused126 |
unused127 | unused128 | unused129 | unused13 | unused130 |
unused131 | unused132 | unused133 | unused134 | unused135 |
unused136 | unused137 | unused138 | unused139 | unused14 |
unused140 | unused141 | unused142 | unused143 | unused144 |
unused145 | unused146 | unused147 | unused148 | unused149 |
unused15 | unused150 | unused151 | unused152 | unused153 |
unused154 | unused155 | unused156 | unused157 | unused158 |
unused159 | unused16 | unused160 | unused161 | unused162 |
unused163 | unused164 | unused165 | unused166 | unused167 |
unused168 | unused169 | unused17 | unused170 | unused171 |
unused172 | unused173 | unused174 | unused175 | unused176 |
unused177 | unused178 | unused179 | unused18 | unused180 |
unused181 | unused182 | unused183 | unused184 | unused185 |
unused186 | unused187 | unused188 | unused189 | unused19 |
unused190 | unused191 | unused192 | unused193 | unused194 |
unused195 | unused196 | unused197 | unused198 | unused199 | unused2
| unused20 | unused200 | unused201 | unused202 | unused203 |
unused204 | unused205 | unused206 | unused207 | unused208 |
unused209 | unused21 | unused210 | unused211 | unused212 |
unused213 | unused214 | unused215 | unused216 | unused217 |
unused218 | unused219 | unused22 | unused220 | unused221 |
unused222 | unused223 | unused224 | unused225 | unused226 |
unused227 | unused228 | unused229 | unused23 | unused230 |
unused231 | unused232 | unused233 | unused234 | unused235 |
unused236 | unused237 | unused238 | unused239 | unused24 |
unused240 | unused241 | unused242 | unused243 | unused244 |
unused245 | unused246 | unused247 | unused248 | unused249 |
unused25 | unused250 | unused251 | unused252 | unused253 |
unused254 | unused255 | unused26 | unused27 | unused28 | unused29 |
unused3 | unused30 | unused31 | unused32 | unused33 | unused34 |
unused35 | unused36 | unused37 | unused38 | unused39 | unused4 |
unused40 | unused41 | unused42 | unused43 | unused44 | unused45 |
unused46 | unused47 | unused48 | unused49 | unused5 | unused50 |
unused51 | unused52 | unused53 | unused54 | unused55 | unused56 |
unused57 | unused58 | unused59 | unused6 | unused60 | unused61 |
unused62 | unused63 | unused64 | unused65 | unused66 | unused67 |
unused68 | unused69 | unused7 | unused70 | unused71 | unused72 |
unused73 | unused74 | unused75 | unused76 | unused77 | unused78 |
```

```
unused79 | unused8 | unused80 | unused81 | unused82 | unused83 |
unused84 | unused85 | unused86 | unused87 | unused88 | unused89 |
unused9 | unused90 | unused91 | unused92 | unused93 | unused94 |
unused95 | unused96 | unused97 | unused98 | unused99 | virtio0 |
virtio1 | virtio10 | virtio11 | virtio12 | virtio13 | virtio14 |
virtio15 | virtio2 | virtio3 | virtio4 | virtio5 | virtio6 |
virtio7 | virtio8 | virtio9>
```

Clave de configuración a la que se moverá el disco en la VM de destino, por ejemplo ide0 o scsi1. De forma predeterminada se conserva la clave del disco de origen.

```
--target-vmid <integer> (100 - 999999999)
```

ID único de la VM de destino.

qm disk rescan [OPTIONS]

Vuelve a explorar todos los almacenamientos y actualiza los tamaños de disco y las imágenes de disco no utilizadas.

```
--dryrun <boolean> (default = 0)
```

No escribe realmente cambios en las configuraciones de las VMs.

```
--vmid <integer> (100 - 999999999)
```

Limita la operación al VMID indicado.

qm disk resize <vmid> <disk> <size> [OPTIONS]

Amplía el tamaño de un volumen.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<disk>: <efidisk0 | ide0 | ide1 | ide2 | ide3 | sata0 | sata1 |
sata2 | sata3 | sata4 | sata5 | scsi0 | scsi1 | scsi10 | scsi11 |
scsi12 | scsi13 | scsi14 | scsi15 | scsi16 | scsi17 | scsi18 |
scsi19 | scsi2 | scsi20 | scsi21 | scsi22 | scsi23 | scsi24 |
scsi25 | scsi26 | scsi27 | scsi28 | scsi29 | scsi3 | scsi30 | scsi4 |
scsi5 | scsi6 | scsi7 | scsi8 | scsi9 | tpmstate0 | virtio0 |
virtio1 | virtio10 | virtio11 | virtio12 | virtio13 | virtio14 |
virtio15 | virtio2 | virtio3 | virtio4 | virtio5 | virtio6 |
virtio7 | virtio8 | virtio9>
```

Disco cuyo tamaño se desea ampliar.

```
<size>: \+?\d+(\.\d+)?[KMGT]?
```

Nuevo tamaño. Con el signo +, el valor se suma al tamaño actual; sin él, se interpreta como tamaño absoluto. No se admite reducir el tamaño del disco.

```
--digest <string>
```

Impide cambios si la configuración actual tiene un digest SHA1 diferente. Puede utilizarse para evitar modificaciones concurrentes.

```
--skiplock <boolean>
```

Ignora los locks. Solo root puede utilizar esta opción.

qm disk unlink <vmid> --idlist <string> [OPTIONS]

Desvincula o elimina imágenes de disco.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
--force <boolean>
```

Fuerza la eliminación física. Sin esta opción, el disco se elimina de la configuración y se crea una entrada unused[n] con el ID del volumen. Desvincular un unused[n] provoca siempre la eliminación física.

```
--idlist <string>
```

Lista de IDs de disco que se desean eliminar.

qm enroll-efi-keys <vmid>

Incorpora certificados importantes actualizados al disco EFI configurado con pre-enrolled-keys. Actualmente se trata de los certificados UEFI 2023 de Microsoft. La VM debe estar apagada.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

qm guest cmd <vmid> <command>

Ejecuta comandos del QEMU Guest Agent.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<command>: <fsfreeze-freeze | fsfreeze-status | fsfreeze-thaw |
fstrim | get-fsinfo | get-host-name | get-memory-block-info |
get-memory-blocks | get-osinfo | get-time | get-timezone |
get-users | get-vcpus | info | network-get-interfaces | ping |
shutdown | suspend-disk | suspend-hybrid | suspend-ram>
```

Comando QGA que se ejecutará.

qm guest exec <vmid> [<extra-args>] [OPTIONS]

Ejecuta el comando indicado dentro del invitado mediante el Guest Agent.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<extra-args>: <array>
```

Argumentos adicionales en forma de array.

```
--pass-stdin <boolean> (default = 0)
```

Si se activa, lee STDIN hasta EOF y lo reenvía al Guest Agent mediante input-data, normalmente como STDIN del proceso lanzado. Admite como máximo 1 MiB.

```
--synchronous <boolean> (default = 1)
```

Si se establece en off, devuelve inmediatamente el PID en lugar de esperar a que termine el comando o se alcance el timeout.

```
--timeout <integer> (0 - N) (default = 30)
```

Tiempo máximo de espera síncrona, en segundos. Si se alcanza, se devuelve el PID. El valor 0 desactiva el límite.

qm guest exec-status <vmid> <pid>

Obtiene el estado del PID indicado, iniciado previamente mediante el Guest Agent.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<pid>: <integer>
```

PID que se desea consultar.

qm guest passwd <vmid> <username> [OPTIONS]

Establece la contraseña del usuario indicado dentro del invitado.

```
<vmid>: <integer> (100 - 999999999)
```

ID único de la VM.

```
<username>: <string>
```

Usuario cuya contraseña se desea establecer.

```
--crypted <boolean> (default = 0)
```

Establezca 1 si la contraseña ya se ha procesado mediante crypt().

qm help [OPTIONS]

Muestra ayuda sobre el comando especificado.

```
--extra-args <array>
```

Muestra ayuda para un comando específico.

```
--verbose <boolean>
```

Utiliza un formato de salida detallado.

CONTINUACIÓN: La siguiente parte comienza en qm import <vmid> <source> --storage <string> [OPTIONS], que importa una máquina virtual extranjera desde una fuente admitida como un almacenamiento ESXi.

Fin del Apéndice A5

Esta entrega continúa A.9 qm desde qm delsnapshot hasta qm help. El siguiente bloque comenzará con qm import y continuará con las operaciones restantes de gestión de máquinas virtuales.

CRITERIO DE TRADUCCIÓN: Se han conservado literalmente comandos, parámetros, identificadores, expresiones regulares, nombres de opciones y valores válidos. Únicamente se han traducido las descripciones del manual.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A · PARTE A6

Interfaz de línea de comandos · qm (continuación)

qm import · importovf · migración · monitor · pending · reboot · remote-migrate · rollback

Importación	Discos/plantillas	Migración	Control de VM
qm import	importdisk / importovf	migrate / remote-migrate	reboot / reset / resume / rollback

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, parámetros, identificadores, valores aceptados y sintaxis.

A.9 qm - QEMU/KVM Virtual Machine Manager (continuación)

qm import

```
qm import <vmid> <source> --storage <string> [OPTIONS]
```

Importa una máquina virtual externa desde una fuente de importación compatible, por ejemplo un almacenamiento ESXi.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

<source>: <string>
Identificador de volumen de la fuente de importación.

--acpi <boolean> (default = 1)
Activa o desactiva ACPI.

--affinity <string>
Lista de núcleos del host usados para ejecutar los procesos del invitado, por ejemplo 0,5,8-11.

--agent [enabled=]<1|0> [,freeze-fs=<1|0>] [,fstrim_cloned_disks=<1|0>] [,type=<virtio|isa>]
Activa o desactiva la comunicación con QEMU Guest Agent y sus propiedades.

--allow-ksm <boolean> (default = 1)
Permite combinar páginas de memoria de este invitado mediante KSM (Kernel Samepage Merging).

--amd-sev [type=]<sev-type> [,allow-smt=<1|0>] [,kernel-hashes=<1|0>] [,no-debug=<1|0>] [,no-key-sharing=<1|0>]
Funciones Secure Encrypted Virtualization (SEV) de CPU AMD.

--arch <aarch64 | x86_64>
Arquitectura del procesador virtual. De forma predeterminada usa la arquitectura del host.

--args <string>
Argumentos arbitrarios pasados a kvm.

--audio0 device=<ich9-intel-hda|intel-hda|AC97> [,driver=<spice|none>]
Configura un dispositivo de audio, útil junto con QXL/SPICE.

--autostart <boolean> (default = 0)
Reinicio automático tras un fallo; actualmente se ignora.

--balloon <integer> (0 - N)
Cantidad de RAM objetivo en MiB. El driver balloon está habilitado por defecto salvo que se establezca a cero.

--bios <ovmf | seabios> (default = seabios)
Selecciona la implementación BIOS.

--boot [[legacy=]<[acdn]{1,4}>] [,order=<device[;device...]>]
Define el orden de arranque. Se recomienda order=; el uso sin clave o legacy= está obsoleto.

--bootdisk (ide|sata|scsi|virtio)\d+
Permite arrancar desde el disco indicado. Obsoleto: use boot: order=foo;bar.

--cdrom <volume>
Alias de la opción -ide2.

--cicustom [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>]
cloud-init: archivos personalizados para sustituir los generados automáticamente al iniciar.

--cipassword <string>
cloud-init: contraseña del usuario. El manual desaconseja su uso y recomienda claves SSH. Versiones antiguas de cloud-init no admiten contraseñas con hash.

--citype <configdrive2 | nocloud | opennebula>
Formato de configuración cloud-init. El valor por defecto depende de ostype: nocloud para Linux y configdrive2 para Windows.

--ciupgrade <boolean> (default = 1)
cloud-init: realiza una actualización automática de paquetes después del primer arranque.

--ciuser <string>
cloud-init: usuario al que se aplicarán claves SSH y contraseña en lugar del usuario predeterminado de la imagen.

--cores <integer> (1 - N) (default = 1)
Número de núcleos por socket.

--cpu [[cputype=]<string>] [,flags=<+FLAG[;-FLAG...]>] [,guest-phys-bits=<integer>] [,hidden=<1|0>] [,hv-vendor-id=<vendor-id>] [,level=<integer>] [,phys-bits=<8-64|host>] [,reported-model=<enum>]
Tipo de CPU emulada.

--cpulimit <number> (0 - 128) (default = 0)
Límite de uso de CPU.

--cpuunits <integer> (1 - 262144) (default = cgroup v1: 1024, cgroup v2: 100)
Peso de CPU para la VM; en cgroup v2 se limita al intervalo [1,10000].

--delete <string>
Lista de ajustes que se desea eliminar.

--description <string>
Descripción de la VM mostrada en el resumen de la interfaz web; se guarda como comentario en el archivo de configuración.

--dryrun <boolean> (default = 0)
Muestra el comando de creación y sale sin realizar cambios.

--efidisk0 [file=]<volume> [,efitype=<2m|4m>] [,format=<enum>] [,import-from=<source volume>] [,ms-cert=<enum>] [,pre-enrolled-keys=<1|0>] [,size=<DiskSize>]
Configura el disco de variables EFI. STORAGE_ID:SIZE_IN_GiB asigna un volumen nuevo; aquí el tamaño se ignora y se copian las variables EFI predeterminadas. STORAGE_ID:0 junto con import-from importa un volumen existente.

--format <qcow2 | raw | vmdk>
Formato de destino.

--freeze <boolean>
Congela la CPU al iniciar; use el comando c del monitor para comenzar la ejecución.

--hookscript <string>
Script ejecutado durante distintas fases del ciclo de vida de la VM.

--hostpci[n] [[host=]<HOSTPCIID[;HOSTPCIID2...]>] [,device-id=<hex id>] [,driver=<vfio|keep>] [,legacy-igd=<1|0>] [,mapping=<mapping-id>] [,mdev=<string>] [,pcie=<1|0>] [,rombar=<1|0>] [,romfile=<string>] [,sub-device-id=<hex id>] [,sub-vendor-id=<hex id>] [,vendor-id=<hex id>] [,x-vga=<1|0>]
Asigna dispositivos PCI del host al invitado.

--hotplug <string> (default = network,disk,usb)
Activa selectivamente hotplug de network, disk, cpu, memory, usb y cloudinit. Use 0 para desactivarlo; 1 equivale a network,disk,usb.

--hugepages <1024 | 2 | any>
Activa memoria hugepages. El valor es el tamaño en MiB; any intenta 1 GiB y, si no es posible, usa 2 MiB.

--ide[n] [file=]<volume> [,...]
Usa un volumen como disco IDE o CD-ROM (n de 0 a 3). STORAGE_ID:SIZE_IN_GiB asigna un volumen nuevo; STORAGE_ID:0 con import-from importa uno existente.

--intel-tdx [type=]<tdx-type> ,attestation=<1|0> [,vsock-cid=<integer>] [,vsock-port=<integer>]
Funciones Trusted Domain Extension (TDX) de CPU Intel.

--ipconfig[n] [gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>]
cloud-init: direcciones IP y gateways. Admite dhcp para IPv4 y auto para autoconfiguración IPv6 sin estado. Si no se especifica IP y cloud-init está habilitado, usa DHCP IPv4 por defecto.

--ivshmem size=<integer> [,name=<string>]
Memoria compartida entre VMs, útil para comunicación directa entre VMs o con el host.

--keephugepages <boolean> (default = 0)
Con hugepages, evita eliminar las páginas tras apagar la VM para reutilizarlas en arranques posteriores.

--keyboard <da | de | de-ch | en-gb | en-us | es | fi | fr | fr-be | fr-ca | fr-ch | hu | is | it | ja | lt | mk | nl | no | pl | pt | pt-br | sl | sv | tr>
Distribución de teclado del servidor VNC. Normalmente es preferible gestionarla dentro del sistema operativo invitado.

--kvm <boolean> (default = 1)
Activa o desactiva virtualización KVM por hardware.

--live-import <boolean> (default = 0)
Inicia inmediatamente la VM y copia los datos en segundo plano.

--localtime <boolean>
Configura el RTC con hora local. Se habilita por defecto si ostype indica Microsoft Windows.

--lock <backup | clone | create | migrate | rollback | snapshot | snapshot-delete | suspended | suspending>
Bloquea o desbloquea la VM.

--machine [[type=]<machine type>] [,aw-bits=<number>] [,enable-s3=<1|0>] [,enable-s4=<1|0>] [,viommu=<intel|virtio>]
Especifica la máquina QEMU.

--memory [current=]<integer>
Propiedades de memoria.

--migrate_downtime <number> (0 - N) (default = 0.1)
Tiempo máximo de parada tolerado durante migraciones. Si la migración no converge, el límite aumenta paso a paso hasta un máximo de 2000 segundos.

--migrate_speed <integer> (0 - N) (default = 0)
Velocidad máxima de migración en MB/s. 0 significa sin límite.

--name <string>
Nombre de la VM, usado únicamente en la interfaz web.

--nameserver <string>
cloud-init: IP del servidor DNS. Si no se establecen nameserver ni searchdomain, create puede usar automáticamente los ajustes del host.

--net[n] [model=]<enum> [,bridge=<bridge>] [,firewall=<1|0>] [,host-tunnel=<1|0>] [,link_down=<1|0>] [,macaddr=<XX:XX:XX:XX:XX:XX>] [,mtu=<integer>] [,queues=<integer>] [,rate=<number>] [,tag=<integer>] [,trunks=<vlanid[;vlanid...]>] [,<model>=<macaddr>]
Configura dispositivos de red.

--numa <boolean> (default = 0)
Activa o desactiva NUMA.

--numa[n] cpus=<id[-id];...> [,hostnodes=<id[-id];...>] [,memory=<number>] [,policy=<preferred|bind|interleave>]
Configura la topología NUMA.

--onboot <boolean> (default = 0)
Indica si la VM se inicia durante el arranque del sistema.

--ostype <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> (default = other)
Sistema operativo invitado.

--parallel[n] /dev/parport\d+|/dev/usb/lp\d+
Asigna dispositivos paralelos del host (n de 0 a 2).

--protection <boolean> (default = 0)
Activa la protección de la VM, deshabilitando las operaciones de eliminar VM y eliminar disco.

--reboot <boolean> (default = 1)
Permite reinicio. Si se establece a 0, la VM sale al intentar reiniciar.

--rng0 [source=]</dev/urandom|/dev/random|/dev/hwrng> [,max_bytes=<integer>] [,period=<integer>]
Configura un generador de números aleatorios basado en VirtIO.

--sata[n] [file=]<volume> [,...]
Usa un volumen como disco SATA o CD-ROM (n de 0 a 5). STORAGE_ID:SIZE_IN_GiB asigna un volumen nuevo; STORAGE_ID:0 con import-from importa uno existente.

--scsi[n] [file=]<volume> [,...]
Usa un volumen como disco SCSI o CD-ROM (n de 0 a 30). STORAGE_ID:SIZE_IN_GiB asigna un volumen nuevo; STORAGE_ID:0 con import-from importa uno existente.

--scsihw <lsi | lsi53c810 | megasas | pvscsi | virtio-scsi-pci | virtio-scsi-single> (default = lsi)
Modelo de controlador SCSI.

--searchdomain <string>
cloud-init: dominios de búsqueda DNS.

--serial[n] (/dev/[^,]+|socket)
Crea un dispositivo serie dentro de la VM (n de 0 a 3).

--shares <integer> (0 - 50000) (default = 1000)
Peso de memoria para auto-ballooning. Un valor mayor otorga proporcionalmente más memoria. 0 desactiva auto-ballooning; pvestatd realiza el ajuste.

--smbios1 [base64=<1|0>] [,family=<Base64 encoded string>] [,manufacturer=<Base64 encoded string>] [,product=<Base64 encoded string>] [,serial=<Base64 encoded string>] [,sku=<Base64 encoded string>] [,uuid=<UUID>] [,version=<Base64 encoded string>]
Define campos SMBIOS tipo 1.

--smp <integer> (1 - N) (default = 1)
Número de CPUs. El manual indica que debe usarse la opción -sockets.

--sockets <integer> (1 - N) (default = 1)
Número de sockets de CPU.

--spice_enhancements [foldersharing=<1|0>] [,videostreaming=<off|all|filter>]
Configura mejoras adicionales para SPICE.

--sshkeys <string>
cloud-init: configura claves SSH públicas, una por línea en formato OpenSSH.

--startdate (now | YYYY-MM-DD | YYYY-MM-DDTHH:MM:SS) (default = now)
Fecha inicial del RTC. Admite now, una fecha o una fecha/hora.

--startup [[order=]\d+] [,up=\d+] [,down=\d+]
Comportamiento de arranque y apagado. order define el orden general; el apagado usa orden inverso. up y down establecen retardos en segundos.

--storage <storage ID>
Almacenamiento predeterminado.

--tablet <boolean> (default = 1)
Activa o desactiva el dispositivo USB tablet.

--tags <string>
Etiquetas de la VM; son metadatos.

--tdf <boolean> (default = 0)
Activa o desactiva la corrección de deriva temporal.

--template <boolean> (default = 0)
Activa o desactiva el estado de plantilla.

--tpmstate0 [file=]<volume> [,format=<raw|qcow2|vmdk>] [,import-from=<source volume>] [,size=<DiskSize>] [,version=<v1.2|v2.0>]
Configura un disco para el estado TPM. El formato se fija a raw. STORAGE_ID:SIZE_IN_GiB asigna un volumen nuevo y usa 4 MiB; STORAGE_ID:0 con import-from importa uno existente.

--unused[n] [file=]<volume>
Referencia a volúmenes no utilizados. Es de uso interno y no debe modificarse manualmente.

--usb[n] [[host=]<HOSTUSBDEVICE|spice>] [,mapping=<mapping-id>] [,usb3=<1|0>]
Configura un dispositivo USB. n es 0 a 4; con machine >= 7.1 y ostype l26 o Windows > 7 puede llegar hasta 14.

--vcpus <integer> (1 - N) (default = 0)
Número de vCPU conectadas mediante hotplug.

--vga [[type=]<enum>] [,clipboard=<vnc>] [,memory=<integer>]
Configura el hardware VGA.

--virtio[n] [file=]<volume> [,...]
Usa un volumen como disco VIRTIO (n de 0 a 15). STORAGE_ID:SIZE_IN_GiB asigna un volumen nuevo; STORAGE_ID:0 con import-from importa uno existente.

--virtiofs[n] [dirid=]<mapping-id> [,cache=<enum>] [,direct-io=<1|0>] [,expose-acl=<1|0>] [,expose-xattr=<1|0>]
Configura el uso compartido de directorios entre host e invitado mediante Virtio-fs.

--vmgenid <UUID> (default = 1 (autogenerated))
Configura el VM Generation ID. Use 1 para autogenerarlo al crear/actualizar y 0 para desactivarlo.

--vmstatestorage <storage ID>
Almacenamiento predeterminado para volúmenes o archivos de estado de la VM.

--watchdog [[model=]<i6300esb|ib700>] [,action=<enum>]
Crea un watchdog de hardware virtual.

Comandos posteriores de qm

```
qm importdisk
```

Alias de qm disk import.

```
qm importovf <vmid> <manifest> <storage> [OPTIONS]
```

Crea una nueva VM utilizando parámetros leídos de un manifiesto OVF.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

<manifest>: <string>
Ruta al archivo OVF.

<storage>: <storage ID>
ID del almacenamiento de destino.

--dryrun <boolean>
Muestra una representación analizada de los parámetros extraídos del OVF, pero no crea la VM.

--format <qcow2 | raw | vmdk>
Formato de destino.

```
qm list [OPTIONS]
```

Índice de máquinas virtuales del nodo.

--full <boolean>
Determina el estado completo de las VMs activas.

```
qm listsnapshot <vmid>
```

Lista todos los snapshots de la VM.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

```
qm migrate <vmid> <target> [OPTIONS]
```

Migra una máquina virtual y crea una nueva tarea de migración.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

<target>: <string>
Nodo de destino.

--bwlimit <integer> (0 - N) (default = migrate limit from datacenter or storage config)
Sobrescribe el límite de ancho de banda de E/S en KiB/s.

--force <boolean>
Permite migrar VMs que utilizan dispositivos locales. Solo root puede usar esta opción.
--migration_network <string>
CIDR de la red o subred utilizada para la migración.
--migration_type <insecure | secure>
El tráfico de migración se cifra mediante túnel SSH de forma predeterminada. En redes seguras y totalmente privadas puede desactivarse para aumentar rendimiento.
--online <boolean>
Usa migración online/live si la VM está en ejecución. Se ignora si está detenida.
--targetstorage <string>
Mapeo de almacenamientos origen-destino. Un único storage ID asigna todos los orígenes a ese destino; el valor especial 1 asigna cada almacenamiento a sí mismo.
--with-conntrack-state <boolean> (default = 0)
Indica si deben migrarse las entradas conntrack de VMs en ejecución.
--with-local-disks <boolean>
Activa migración de almacenamiento en vivo para discos locales.

```
qm monitor <vmid>
```

Entra en la interfaz QEMU Monitor.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

```
qm move-disk
```

Alias de qm disk move.

```
qm move_disk
```

Alias de qm disk move.

```
qm mtunnel
```

Usado por qmigrate; no debe utilizarse manualmente.

```
qm nbdstop <vmid>
```

Detiene el servidor NBD embebido.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

```
qm pending <vmid>
```

Obtiene la configuración de la VM con los valores actuales y pendientes.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

```
qm reboot <vmid> [OPTIONS]
```

Reinicia la VM apagándola y arrancándola de nuevo. Aplica los cambios pendientes.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.
--timeout <integer> (0 - N)
Tiempo máximo, en segundos, que se espera al apagado.

```
qm remote-migrate <vmid> [<target-vmid>] <target-endpoint> --target-bridge <string> --target-storage
<string> [OPTIONS]
```

Migra una VM a un clúster remoto. Crea una nueva tarea de migración. Función EXPERIMENTAL.
<vmid>: <integer> (100 - 999999999)
ID único de la VM de origen.
<target-vmid>: <integer> (100 - 999999999)
ID único de la VM de destino.
<target-endpoint>: apitoken=<PVEAPIToken=user@realm!token=SECRET>,host=<ADDRESS> [,fingerprint=<FINGERPRINT>] [,port=<PORT>]
Endpoint remoto de destino.
--bwlimit <integer> (0 - N) (default = migrate limit from datacenter or storage config)
Sobrescribe el límite de ancho de banda de E/S en KiB/s.
--delete <boolean> (default = 0)
Elimina la VM original y sus datos tras una migración correcta. Por defecto, la VM original queda detenida en el clúster de origen.
--online <boolean>
Usa migración online/live si la VM está en ejecución; se ignora si está detenida.

--target-bridge <string>
Mapeo de bridges origen-destino. Un único bridge ID mapea todos los bridges; el valor 1 mapea cada bridge a sí mismo.
--target-storage <string>
Mapeo de almacenamientos origen-destino. Un único storage ID mapea todos; el valor 1 mapea cada almacenamiento a sí mismo.

```
qm rescan
```

Alias de qm disk rescan.

```
qm reset <vmid> [OPTIONS]
```

Reinicia de forma forzada la máquina virtual.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.
--skiplock <boolean>
Ignora locks. Solo root puede usar esta opción.

```
qm resize
```

Alias de qm disk resize.

```
qm resume <vmid> [OPTIONS]
```

Reanuda una máquina virtual.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.
--nocheck <boolean>
No comprueba si la VM está en ejecución; se usa internamente durante migraciones. Solo root puede usarlo.
--skiplock <boolean>
Ignora locks. Solo root puede usar esta opción.

```
qm rollback <vmid> <snapname> [OPTIONS]
```

Revierte el estado de la VM al snapshot indicado.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.
<snapname>: <string>
Nombre del snapshot.
--start <boolean> (default = 0)
Indica si la VM debe iniciarse tras completar el rollback. Si el snapshot incluye RAM, la VM se iniciará automáticamente.

```
qm sendkey <vmid> <key> [OPTIONS]
```

Envía un evento de teclado a la máquina virtual.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.
<key>: <string>
Tecla en codificación del monitor QEMU.
--skiplock <boolean>
Ignora locks. Solo root puede usar esta opción.

CONTINUACIÓN: La siguiente parte comienza con qm set <vmid> [OPTIONS], que abre un bloque extenso de opciones de configuración de VM.

Fin de la Parte A6

Esta entrega continúa A.9 qm desde qm import y termina inmediatamente antes de qm set.

CRITERIO DE TRADUCCIÓN: Los comandos, nombres de parámetros, valores aceptados y expresiones técnicas se mantienen en su forma original. Solo se traducen las descripciones explicativas del manual.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A · PARTE A7

Interfaz de línea de comandos

A.9 qm (continuación) · A.10 qmrestore

Configuración	Ciclo de vida	Snapshots / consola	Restauración
qm set	start · shutdown · stop	snapshot · terminal · vncproxy	qmrestore

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, opciones, valores admitidos, expresiones y rutas. Se traducen únicamente las descripciones.

A.9 qm - QEMU/KVM Virtual Machine Manager (continuación)

qm set <vmid> [OPTIONS]

Establece opciones de una máquina virtual mediante la API síncrona. Para acciones que impliquen hotplug o asignación de almacenamiento, el manual recomienda considerar el método POST.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

Opciones generales, CPU y memoria

--acpi <boolean> (default = 1)
Activa o desactiva ACPI.

--affinity <string>
Lista de cores del host usados para ejecutar procesos del invitado, por ejemplo: 0,5,8-11.

--agent [enabled=]<1|0> [,freeze-fs=<1|0>] [,fstrim_cloned_disks=<1|0>] [,type=<virtio|isa>]
Activa o desactiva la comunicación con QEMU Guest Agent y sus propiedades.

--allow-ksm <boolean> (default = 1)
Permite que KSM (Kernel Samepage Merging) combine páginas de memoria de este invitado.

--amd-sev [type=]<sev-type> [,allow-smt=<1|0>] [,kernel-hashes=<1|0>] [,no-debug=<1|0>] [,no-key-sharing=<1|0>]
Configura funciones AMD Secure Encrypted Virtualization (SEV).

--arch <aarch64 | x86_64>
Arquitectura del procesador virtual. Por defecto se usa la arquitectura del host.

--args <string>
Argumentos arbitrarios pasados a KVM.

--audio0 device=<ich9-intel-hda|intel-hda|AC97> [,driver=<spice|none>]
Configura un dispositivo de audio; útil junto con QXL/SPICE.

--autostart <boolean> (default = 0)
Reinicio automático tras un crash (actualmente ignorado).

--balloon <integer> (0 - N)
Cantidad objetivo de RAM para la VM en MiB. Ballooning está habilitado por defecto salvo que se configure a 0.

--bios <ovmf | seabios> (default = seabios)
Selecciona la implementación de BIOS.

--cores <integer> (1 - N) (default = 1)
Número de cores por socket.

--cpu [[cputype=]<string>] [,flags=<+FLAG[;-FLAG...]>] [,guest-phys-bits=<integer>] [,hidden=<1|0>] [,hv-vendor-id=<vendor-id>] [,level=<integer>] [,phys-bits=<8-64|host>] [,reported-model=<enum>]
Tipo de CPU emulada.

--cpulimit <number> (0 - 128) (default = 0)
Límite de uso de CPU.

--cpuunits <integer> (1 - 262144) (default = cgroup v1: 1024, cgroup v2: 100)
Peso de CPU para la VM; con cgroup v2 se limita al rango [1,10000].

--memory [current=]<integer>
Propiedades de memoria.

--numa <boolean> (default = 0)
Activa o desactiva NUMA.

--numa[n] cpus=<id[-id];...> [,hostnodes=<id[-id];...>] [,memory=<number>] [,policy=<preferred|bind|interleave>]
Topología NUMA.

--shares <integer> (0 - 50000) (default = 1000)
Peso relativo de memoria para auto-ballooning. Usar 0 desactiva auto-ballooning. pvestatd realiza el ajuste.

--smp <integer> (1 - N) (default = 1)
Número de CPUs. El manual recomienda usar sockets en su lugar.

--sockets <integer> (1 - N) (default = 1)
Número de sockets de CPU.

--vcpus <integer> (1 - N) (default = 0)
Número de vCPU conectadas mediante hotplug.

Arranque, Cloud-Init y configuración de la VM

--boot [[legacy=]<[acdn]{1,4}>] [,order=<device[;device...]>]
Orden de arranque del invitado. Se recomienda usar la subpropiedad order=; legacy= está obsoleta.

--bootdisk (ide|sata|scsi|virtio)\d+
Disco de arranque. Opción obsoleta; use boot: order=foo;bar.

--cdrom <volume>
Alias de -ide2.

--cicustom [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>]
Cloud-Init: usa archivos personalizados en lugar de los generados automáticamente al inicio.

--cipassword <password>
Cloud-Init: contraseña para el usuario. En general no se recomienda; es preferible usar claves SSH. Versiones antiguas de Cloud-Init no admiten contraseñas hasheadas.

--citype <configdrive2 | nocloud | opennebula>
Formato de configuración Cloud-Init. El valor predeterminado depende de ostype: nocloud para Linux y configdrive2 para Windows.

--ciupgrade <boolean> (default = 1)
Cloud-Init: actualiza paquetes automáticamente tras el primer arranque.

--ciuser <string>
Cloud-Init: usuario al que se aplican claves SSH y contraseña en lugar del usuario por defecto de la imagen.

--description <string>
Descripción de la VM. Se muestra en el resumen web y se guarda como comentario en la configuración.

--delete <string>
Lista de ajustes que se desean eliminar.

--digest <string>
Impide cambios si el archivo de configuración actual tiene un SHA1 distinto. Permite detectar modificaciones concurrentes.

--force <boolean>
Fuerza la eliminación física. Sin esta opción, el disco se retira de la configuración y se crea una entrada unused[n]. Desvincular unused[n] siempre provoca eliminación física.

--freeze <boolean>
Congela la CPU al inicio; use el comando c del monitor para comenzar la ejecución.

--hookscript <string>
Script ejecutado en distintas fases del ciclo de vida de la VM.

--hotplug <string> (default = network,disk,usb)
Activa selectivamente hotplug: network, disk, cpu, memory, usb y cloudinit. 0 lo desactiva por completo; 1 equivale a network,disk,usb.

--kvm <boolean> (default = 1)
Activa o desactiva la virtualización hardware KVM.

--localtime <boolean>
Configura el RTC en hora local. Se activa por defecto si ostype indica Microsoft Windows.

--lock <backup | clone | create | migrate | rollback | snapshot | snapshot-delete | suspended | suspending>
Bloquea o desbloquea la VM.

--machine [[type=]<machine type>] [,aw-bits=<number>] [,enable-s3=<1|0>] [,enable-s4=<1|0>] [,viommu=<intel|virtio>]
Especifica la máquina QEMU.

--name <string>
Nombre de la VM, usado en la interfaz web.

--onboot <boolean> (default = 0)
Indica si la VM se inicia durante el arranque del sistema.

--ostype <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> (default = other)
Sistema operativo invitado.

--protection <boolean> (default = 0)
Activa el flag de protección. Impide eliminar la VM y sus discos.

--reboot <boolean> (default = 1)
Permite reboot. Si vale 0, la VM termina al reiniciar.

--revert <string>
Revierte un cambio pendiente.

--skiplock <boolean>
Ignora locks; solo root puede usar esta opción.

--startdate (now | YYYY-MM-DD | YYYY-MM-DDTHH:MM:SS) (default = now)
Fecha inicial del reloj en tiempo real.

--startup [[order=]\d+] [,up=\d+] [,down=\d+]
Comportamiento de arranque/apagado. order define el orden; el apagado se realiza en orden inverso. up y down añaden retardos en segundos.

--tags <string>
Etiquetas de la VM. Solo son metadatos.

--tdf <boolean> (default = 0)
Activa o desactiva la corrección de deriva temporal.

--template <boolean> (default = 0)
Activa o desactiva el modo Template.

--vmgenid <UUID> (default = 1 (autogenerated))
Establece VM Generation ID. Use 1 para autogenerarlo o 0 para deshabilitarlo explícitamente.

--vmstatestorage <storage ID>
Almacenamiento predeterminado para volúmenes/archivos de estado de la VM.

NOTA: --force requiere la opción delete.

Dispositivos, PCI, red y almacenamiento

--efidisk0 [file=]<volume> [,efitype=<2m|4m>] [,format=<enum>] [,import-from=<source volume>] [,ms-cert=<enum>] [,pre-enrolled-keys=<1|0>] [,size=<DiskSize>]

Configura el disco para variables EFI. STORAGE_ID:SIZE_IN_GiB crea un volumen nuevo. El tamaño indicado se ignora aquí y se copian las variables EFI predeterminadas. Use STORAGE_ID:0 junto con import-from para importar un volumen existente.

--hostpci[n] [[host=]<HOSTPCIID[;HOSTPCIID2...]>] [,device-id=<hex id>] [,driver=<vfio|keep>] [,legacy-igd=<1|0>] [,mapping=<mapping-id>] [,mdev=<string>] [,pcie=<1|0>] [,rombar=<1|0>] [,romfile=<string>] [,sub-device-id=<hex id>] [,sub-vendor-id=<hex id>] [,vendor-id=<hex id>] [,x-vga=<1|0>]

Mapea dispositivos PCI del host dentro del invitado.

--hugepages <1024 | 2 | any>

Activa memoria hugepages. El valor es el tamaño en MiB; any intenta usar páginas de 1 GiB y, si no es posible, cae a 2 MiB.

--ide[n] [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] [,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>] [,bps_rd_max_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_max_length=<seconds>] [,cache=<enum>] [,detect_zeroes=<1|0>] [,discard=<ignore|on>] [,format=<enum>] [,import-from=<source volume>] [,iops=<iops>] [,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>] [,iops_rd_max=<iops>] [,iops_rd_max_length=<seconds>] [,iops_wr=<iops>] [,iops_wr_max=<iops>] [,iops_wr_max_length=<seconds>] [,mbps=<mbps>] [,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>] [,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>] [,model=<model>] [,replicate=<1|0>] [,rerror=<ignore|report|stop>] [,serial=<serial>] [,shared=<1|0>] [,size=<DiskSize>] [,snapshot=<1|0>] [,ssd=<1|0>] [,werror=<enum>] [,wwn=<wwn>]

Usa un volumen como disco IDE o CD-ROM (n de 0 a 3). STORAGE_ID:SIZE_IN_GiB crea un volumen y STORAGE_ID:0 + import-from importa uno existente.

--intel-tdx [type=]<tdx-type> ,attestation=<1|0> [,vsock-cid=<integer>] [,vsock-port=<integer>]

Funciones Intel Trusted Domain Extensions (TDX).

--ipconfig[n] [gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>]

Cloud-Init: IPs y gateways de la interfaz. Las IP usan CIDR. dhcp activa DHCP en IPv4; auto permite autoconfiguración IPv6 sin estado (Cloud-Init 19.4 o posterior). Si no se indica IPv4 ni IPv6, se usa DHCP en IPv4.

--ivshmem size=<integer> [,name=<string>]

Memoria compartida entre VMs; útil para comunicación directa entre VMs o con el host.

--keephugepages <boolean> (default = 0)

Usado con hugepages. Si se activa, las hugepages no se eliminan al apagar la VM y pueden reutilizarse.

--keyboard <da | de | de-ch | en-gb | en-us | es | fi | fr | fr-be | fr-ca | fr-ch | hu | is | it | ja | lt | mk | nl | no | pl | pt | pt-br | sl | sv | tr>

Distribución de teclado para VNC. Normalmente es mejor gestionarla dentro del SO invitado.

--migrate_downtime <number> (0 - N) (default = 0.1)

Downtime máximo tolerado, en segundos, durante migraciones. QEMU puede incrementarlo gradualmente hasta un máximo de 2000 s para lograr convergencia.

--migrate_speed <integer> (0 - N) (default = 0)

Velocidad máxima de migración en MB/s. 0 significa sin límite.

--nameserver <string>

Cloud-Init: establece la IP del servidor DNS. Si no se especifica ni nameserver ni searchdomain, se utiliza automáticamente la configuración del host.

--net[n] [model=]<enum> [,bridge=<bridge>] [,firewall=<1|0>] [,host-tunnel=<1|0>] [,link_down=<1|0>] [,macaddr=<XX:XX:XX:XX:XX:XX>] [,mtu=<integer>] [,queues=<integer>] [,rate=<number>] [,tag=<integer>] [,trunks=<vlanid[;vlanid...]>] [,<model>=<macaddr>]

Configura dispositivos de red.

--parallel[n] /dev/parport\d+|/dev/usb/lp\d+

Mapea dispositivos paralelos del host (n de 0 a 2).

--rng0 [source=]</dev/urandom|/dev/random|/dev/hwrng> [,max_bytes=<integer>] [,period=<integer>]

Configura un generador de números aleatorios basado en VirtIO.

--sata[n] [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] [,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>] [,bps_rd_max_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_max_length=<seconds>] [,cache=<enum>] [,detect_zeroes=<1|0>] [,discard=<ignore|on>] [,format=<enum>] [,import-from=<source volume>] [,iops=<iops>] [,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>] [,iops_rd_max=<iops>] [,iops_rd_max_length=<seconds>] [,iops_wr=<iops>] [,iops_wr_max=<iops>] [,iops_wr_max_length=<seconds>] [,mbps=<mbps>] [,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>] [,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>] [,replicate=<1|0>] [,rerror=<ignore|report|stop>] [,serial=<serial>] [,shared=<1|0>] [,size=<DiskSize>] [,snapshot=<1|0>] [,ssd=<1|0>] [,werror=<enum>] [,wwn=<wwn>]

Usa un volumen como disco SATA o CD-ROM (n de 0 a 5). Admite creación mediante STORAGE_ID:SIZE_IN_GiB o importación mediante STORAGE_ID:0 + import-from.

--scsi[n] [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] [,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>] [,bps_rd_max_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_max_length=<seconds>] [,cache=<enum>] [,detect_zeroes=<1|0>] [,discard=<ignore|on>] [,format=<enum>] [,import-from=<source volume>] [,iops=<iops>] [,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>] [,iops_rd_max=<iops>] [,iops_rd_max_length=<seconds>] [,iops_wr=<iops>] [,iops_wr_max=<iops>] [,iops_wr_max_length=<seconds>] [,iothread=<1|0>] [,mbps=<mbps>] [,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>] [,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>] [,product=<product>] [,queues=<integer>] [,replicate=<1|0>] [,rerror=<ignore|report|stop>] [,ro=<1|0>] [,scsiblock=<1|0>] [,serial=<serial>] [,shared=<1|0>] [,size=<DiskSize>] [,snapshot=<1|0>] [,ssd=<1|0>] [,vendor=<vendor>] [,werror=<enum>] [,wwn=<wwn>]

Usa un volumen como disco SCSI o CD-ROM (n de 0 a 30). Admite creación o importación con la sintaxis estándar de almacenamiento.

--scsihw <lsi | lsi53c810 | megasas | pvscsi | virtio-scsi-pci | virtio-scsi-single> (default = lsi)

Modelo de controlador SCSI.

--searchdomain <string>

Cloud-Init: dominios de búsqueda DNS. Si no se especifica ni searchdomain ni nameserver, se toma la configuración del host.

--serial[n] (/dev/[^,]+|socket)

Crea un dispositivo serie dentro de la VM (n de 0 a 3).

--smbios1 [base64=<1|0>] [,family=<Base64 encoded string>] [,manufacturer=<Base64 encoded string>] [,product=<Base64 encoded string>] [,serial=<Base64 encoded string>] [,sku=<Base64 encoded string>] [,uuid=<UUID>] [,version=<Base64 encoded string>]

Especifica campos SMBIOS tipo 1.

--spice_enhancements [foldersharing=<1|0>] [,videostreaming=<off|all|filter>]

Mejoras adicionales para SPICE.

--sshkeys <filepath>

Cloud-Init: configura claves SSH públicas, una por línea, en formato OpenSSH.

--tablet <boolean> (default = 1)

Activa o desactiva el dispositivo USB tablet.

--tpmstate0 [file=]<volume> [,format=<raw|qcow2|vmdk>] [,import-from=<source volume>] [,size=<DiskSize>] [,version=<v1.2|v2.0>]

Configura el disco de estado TPM. El formato se fija a raw. Para un volumen nuevo se utilizan 4 MiB; también admite import-from.

--unused[n] [file=]<volume>

Referencia a volúmenes sin usar. Se utiliza internamente y no debe modificarse manualmente.

--usb[n] [[host=]<HOSTUSBDEVICE|spice>] [,mapping=<mapping-id>] [,usb3=<1|0>]

Configura un dispositivo USB. n es 0..4; con machine >= 7.1 y ostype l26 o Windows > 7 puede llegar hasta 14.

--vga [[type=]<enum>] [,clipboard=<vnc>] [,memory=<integer>]

Configura el hardware VGA.

--virtio[n] [file=]<volume> [,aio=<native|threads|io_uring>] [,backup=<1|0>] [,bps=<bps>] [,bps_max_length=<seconds>] [,bps_rd=<bps>] [,bps_rd_max_length=<seconds>] [,bps_wr=<bps>] [,bps_wr_max_length=<seconds>] [,cache=<enum>] [,detect_zeroes=<1|0>] [,discard=<ignore|on>] [,format=<enum>] [,import-from=<source volume>] [,iops=<iops>] [,iops_max=<iops>] [,iops_max_length=<seconds>] [,iops_rd=<iops>] [,iops_rd_max=<iops>] [,iops_rd_max_length=<seconds>] [,iops_wr=<iops>] [,iops_wr_max=<iops>] [,iops_wr_max_length=<seconds>] [,iothread=<1|0>] [,mbps=<mbps>] [,mbps_max=<mbps>] [,mbps_rd=<mbps>] [,mbps_rd_max=<mbps>] [,mbps_wr=<mbps>] [,mbps_wr_max=<mbps>] [,media=<cdrom|disk>] [,replicate=<1|0>] [,rerror=<ignore|report|stop>] [,ro=<1|0>] [,serial=<serial>] [,shared=<1|0>] [,size=<DiskSize>] [,snapshot=<1|0>] [,werror=<enum>]

Usa un volumen como disco VirtIO (n de 0 a 15). Admite creación e importación mediante la sintaxis estándar.

--virtiofs[n] [dirid=]<mapping-id> [,cache=<enum>] [,direct-io=<1|0>] [,expose-acl=<1|0>] [,expose-xattr=<1|0>]

Comparte un directorio entre host e invitado mediante Virtio-fs.

--watchdog [[model=]<i6300esb|ib700>] [,action=<enum>]

Crea un dispositivo watchdog virtual.

Resto de comandos qm

qm showcmd <vmid> [OPTIONS]

Muestra la línea de comandos usada para iniciar la VM (información de depuración).

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--pretty <boolean> (default = 0)
Coloca cada opción en una línea distinta para mejorar la legibilidad.

--snapshot <string>
Obtiene los valores de configuración desde el snapshot indicado.

qm shutdown <vmid> [OPTIONS]

Apaga la máquina virtual enviando un evento ACPI al sistema operativo invitado, de forma análoga a pulsar el botón de encendi do de un equipo físico.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--forceStop <boolean> (default = 0)
Asegura que la VM se detenga.

--keepActive <boolean> (default = 0)
No desactiva los volúmenes de almacenamiento.

--skiplock <boolean>
Ignora locks; solo root puede usar esta opción.

--timeout <integer> (0 - N)
Tiempo máximo de espera, en segundos.

qm snapshot <vmid> <snapname> [OPTIONS]

Crea un snapshot de una VM.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

<snapname>: <string>
Nombre del snapshot.

--description <string>
Descripción o comentario textual.

--vmstate <boolean>
Guarda también el estado de la VM.

qm start <vmid> [OPTIONS]

Inicia una máquina virtual.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--force-cpu <string>
Sobrescribe el argumento -cpu de QEMU con la cadena indicada.

--machine [[type=]<machine type>] [,aw-bits=<number>] [,enable-s3=<1|0>] [,enable-s4=<1|0>] [,viommu=<intel|virtio>]
Especifica la máquina QEMU.

--migratedfrom <string>
Nombre del nodo de clúster de origen.

--migration_network <string>
CIDR de la red/subred utilizada para la migración.

--migration_type <insecure | secure>
Por defecto el tráfico de migración se cifra mediante un túnel SSH. En redes seguras y totalmente privadas puede desactivarse para mejorar rendimiento.

--nets-host-mtu net\d+=\d+(,net\d+=\d+)*
Opción de compatibilidad de migración. Lista dispositivos VirtIO y su host_mtu efectivo en el lado origen. Un valor 0 indica que debe evitarse host_mtu para ese dispositivo.

--skiplock <boolean>
Ignora locks; solo root puede usar esta opción.

--stateuri <string>
Ubicación desde la que algunos comandos guardan/restauran estado.

--targetstorage <string>
Mapeo de almacenamientos origen a destino. Un único ID mapea todos los orígenes; el valor especial 1 mantiene cada almacenamiento en sí mismo.

--timeout <integer> (0 - N) (default = max(30, vm memory in GiB))
Tiempo máximo de espera en segundos.

--with-conntrack-state <boolean> (default = 0)
Indica si deben migrarse entradas conntrack de VMs en ejecución.

qm status <vmid> [OPTIONS]

Muestra el estado de la VM.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--verbose <boolean>
Formato de salida detallado.

qm stop <vmid> [OPTIONS]

Detiene inmediatamente la VM. Es equivalente a desconectar la alimentación de un equipo en funcionamiento y puede dañar los datos del invitado.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--keepActive <boolean> (default = 0)
No desactiva los volúmenes de almacenamiento.

--migratedfrom <string>
Nombre del nodo de clúster de origen.

--overrule-shutdown <boolean> (default = 0)
Intenta abortar tareas qmshutdown activas antes de detener.

--skiplock <boolean>
Ignora locks; solo root puede usar esta opción.

--timeout <integer> (0 - N)
Tiempo máximo de espera en segundos.

qm suspend <vmid> [OPTIONS]

Suspende una máquina virtual.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--skiplock <boolean>
Ignora locks; solo root puede usar esta opción.

--statestorage <storage ID>
Almacenamiento para el estado de la VM. Requiere todisk.

--todisk <boolean> (default = 0)
Si se activa, suspende la VM a disco; se reanudará en el siguiente arranque.

qm template <vmid> [OPTIONS]

Convierte una VM en plantilla.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--disk <efidisk0 | ide0 | ide1 | ide2 | ide3 | sata0 | sata1 | sata2 | sata3 | sata4 | sata5 | scsi0..scsi30 | tpmstate0 | virtio0..virtio15>
Permite convertir únicamente un disco en imagen base. El manual enumera de forma explícita todos los IDs válidos dentro de es os rangos.

qm terminal <vmid> [OPTIONS]

Abre un terminal mediante un dispositivo serie. La VM debe tener un dispositivo serie configurado, por ejemplo serial0: socket.
<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--escape <string> (default = ^O)
Carácter de escape.

--iface <serial0 | serial1 | serial2 | serial3>
Selecciona el dispositivo serie. Si se omite, se utiliza el primero adecuado.

Aliases y utilidades finales

qm unlink
Alias de qm disk unlink.

qm unlock <vmid>
Desbloquea la VM. <vmid> es su ID único.

qm vncproxy <vmid>
Proxy del tráfico VNC de la VM hacia stdin/stdout.

qm wait <vmid> [OPTIONS]
Espera hasta que la VM se detenga.

--timeout <integer> (1 - N)
Timeout en segundos. Por defecto espera indefinidamente.

A.10 qmrestore - Restaurar backups vzdump de QemuServer

```
qmrestore help
qmrestore <archive> <vmid> [OPTIONS]
```

Restaura backups vzdump de QemuServer.

<archive>: <string>
Archivo de backup. Puede usarse - para leer desde la entrada estándar.

<vmid>: <integer> (100 - 999999999)
ID único de la VM.

--bwlimit <number> (0 - N)
Sobrescribe el límite de ancho de banda de E/S en KiB/s.

--force <boolean>
Permite sobrescribir una VM existente.

--ha-managed <boolean> (default = 0)
Añade la VM como recurso HA después de restaurarla.

--live-restore <boolean>
Inicia la VM inmediatamente desde el backup y continúa la restauración en segundo plano. Solo PBS.

--pool <string>
Añade la VM al pool indicado.

--start <boolean> (default = 0)
Inicia la VM cuando la restauración termina correctamente.

--storage <storage ID>
Almacenamiento predeterminado.

--unique <boolean>
Asigna una dirección Ethernet aleatoria y única.

FIN DE LA PARTE A7: Con esta entrega queda completado A.9 qm y también A.10 qmrestore. La siguiente parte comienza en A.11 pct - Proxmox Container Toolkit.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

APÉNDICE A8

A.11 pct - Proxmox Container Toolkit

Clone · Create · Console · Discos · Migración · Ficheros · Resize

Contenedores	Almacenamiento	Operación	Migración
clone · create · config	rootfs · mp[n] · fsck	enter · exec · push/pull	migrate · remote-migrate

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, parámetros, identificadores, valores y expresiones de la CLI.

A.11 pct - Proxmox Container Toolkit

```
pct <COMMAND> [ARGS] [OPTIONS]
```

pct clone

```
pct clone <vmid> <newid> [OPTIONS]
```

Crea un clon o una copia de un contenedor.
<vmid>: <integer> (100 - 999999999) - ID único del contenedor de origen.

<newid>: <integer> (100 - 999999999) - VMID que se asignará al clon.

--bwlimit <number> (0 - N) - Sobrescribe el límite de ancho de banda de E/S en KiB/s; por defecto usa el límite de clone del Datacenter o del almacenamiento.

--description <string> - Descripción del nuevo CT.

--full <boolean> - Crea una copia completa de todos los discos. En un CT normal se hace siempre; en plantillas se intenta crear un clon enlazado por defecto.

--hostname <string> - Establece el nombre de host del nuevo CT.

--pool <string> - Añade el nuevo CT al pool indicado.

--snapname <string> - Nombre del snapshot.

--storage <storage ID> - Almacenamiento de destino para un clon completo.

--target <string> - Nodo de destino. Solo se permite si el CT original utiliza almacenamiento compartido.

pct config

```
pct config <vmid> [OPTIONS]
```

Obtiene la configuración del contenedor.
--current <boolean> (default = 0) - Obtiene los valores actuales en lugar de los valores pendientes.

--snapshot <string> - Obtiene la configuración del snapshot indicado.

pct console

```
pct console <vmid> [OPTIONS]
```

Abre una consola para el contenedor indicado.
--escape ^?[a-z] (default = ^a) - Prefijo de la secuencia de escape. Por ejemplo, para usar <Ctrl+b q> como secuencia de escape, indique ^b.

pct cpusets

```
pct cpusets
```

Muestra la lista de conjuntos de CPU asignados.

pct create

```
pct create <vmid> <ostemplate> [OPTIONS]
```

Crea o restaura un contenedor.
<vmid>: <integer> (100 - 999999999) - ID único del contenedor.

<ostemplate>: <string> - Plantilla del sistema operativo o fichero de backup.

--arch <amd64 | arm64 | armhf | i386 | riscv32 | riscv64> (default = amd64) - Arquitectura del sistema operativo.

--bwlimit <number> (0 - N) - Sobrescribe el límite de ancho de banda de E/S en KiB/s; por defecto usa el límite de restore del Datacenter o almacenamiento.

--cmode <console | shell | tty> (default = tty) - Modo de consola. tty intenta abrir uno de los TTY disponibles; console se conecta a /dev/console; shell ejecuta directamente una shell dentro del contenedor, sin login.

--console <boolean> (default = 1) - Conecta un dispositivo de consola (/dev/console) al contenedor.

--cores <integer> (1 - 8192) - Número de cores asignados. Por defecto el contenedor puede utilizar todos los cores disponibles.

--cpulimit <number> (0 - 8192) (default = 0) - Límite de uso de CPU.

NOTA: Si el equipo tiene 2 CPUs, dispone de un total de 2 unidades de tiempo de CPU. El valor 0 significa que no existe límite de CPU.

--cpuunits <integer> (0 - 500000) - Peso de CPU del contenedor. En cgroup v2 se limita al rango [1, 10000]. Valores predeterminados: cgroup v1 = 1024; cgroup v2 = 100.

--debug <boolean> (default = 0) - Intenta generar información más detallada. Actualmente activa el nivel de log debug durante el arranque.

--description <string> - Descripción del contenedor, mostrada en el resumen de la interfaz web y guardada como comentario en la configuración.

--dev[n] [[path=]<Path>] [,deny-write=<1|0>] [,gid=<integer>] [,mode=<Octal access mode>] [,uid=<integer>] - Dispositivo que se pasa al contenedor.

--entrypoint <...> (default = /sbin/init) - Comando que se ejecutará como init, opcionalmente con argumentos; puede comenzar con una ruta absoluta, relativa o con un binario presente en $PATH.

--env <...> - Entorno de ejecución del contenedor como lista separada por NUL. Sustituye las entradas lxc.environment.runtime de la configuración.

--features [force_rw_sys=<1|0>] [,fuse=<1|0>] [,keyctl=<1|0>] [,mknod=<1|0>] [,mount=<fstype;fstype;...>] [,nesting=<1|0>] - Permite al contenedor acceder a características avanzadas.

--force <boolean> - Permite sobrescribir un contenedor existente.

--ha-managed <boolean> (default = 0) - Añade el CT como recurso HA después de crearlo.

--hookscript <string> - Script que se ejecutará en distintos momentos del ciclo de vida del contenedor.

--hostname <string> - Establece el hostname del contenedor.

--ignore-unpack-errors <boolean> - Ignora errores al extraer la plantilla.

--lock <backup | create | destroyed | disk | fstrim | migrate | mounted | rollback | snapshot | snapshot-delete> - Bloquea o desbloquea el contenedor.

--memory <integer> (16 - N) (default = 512) - Cantidad de RAM del contenedor en MB.

--mp[n] [volume=]<volume>,mp=<Path> [...] - Usa un volumen como punto de montaje del contenedor. Puede emplearse STORAGE_ID:SIZE_IN_GiB para asignar un nuevo volumen.

```
--mp[n] [volume=]<volume> ,mp=<Path> [,acl=<1|0>] [,backup=<1|0>]
[,idmap=<type:container:disk:range-size[;type:container:disk:range-size;...]>]
[,keepattrs=<1|0>] [,mountoptions=<opt[;opt...]>] [,quota=<1|0>]
[,replicate=<1|0>] [,ro=<1|0>] [,shared=<1|0>] [,size=<DiskSize>]
```

--nameserver <string> - Establece la dirección IP del servidor DNS. Si no se define ni searchdomain ni nameserver, create utiliza automáticamente la configuración del host.

```
--net[n] name=<string> [,bridge=<bridge>] [,firewall=<1|0>]
[,gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,host-managed=<1|0>]
[,hwaddr=<XX:XX:XX:XX:XX:XX>] [,ip=<(IPv4/CIDR|dhcp|manual)>]
[,ip6=<(IPv6/CIDR|auto|dhcp|manual)>] [,link_down=<1|0>]
[,mtu=<integer>] [,rate=<mbps>] [,tag=<integer>]
[,trunks=<vlanid[;vlanid...]>] [,type=<veth>]
```

Especifica las interfaces de red del contenedor.

--onboot <boolean> (default = 0) - Indica si el contenedor se iniciará durante el arranque del sistema.

--ostype <alpine | archlinux | centos | debian | devuan | fedora | gentoo | nixos | opensuse | ubuntu | unmanaged> - Tipo de sistema operativo. Se utiliza para preparar la configuración dentro del contenedor y se corresponde con los scripts LXC de /usr/share/lxc/config/<ostype>.common.conf. unmanaged omite la configuración específica del SO.

--password <password> - Establece la contraseña de root dentro del contenedor.

--pool <string> - Añade el CT al pool indicado.

--protection <boolean> (default = 0) - Activa la protección del contenedor e impide eliminar el CT o eliminar/modificar sus discos.

--restore <boolean> - Marca la tarea como una restauración.

```
--rootfs [volume=]<volume> [,acl=<1|0>]
[,idmap=<type:container:disk:range-size[;type:container:disk:range-size;...]>]
[,mountoptions=<opt[;opt...]>] [,quota=<1|0>] [,replicate=<1|0>]
[,ro=<1|0>] [,shared=<1|0>] [,size=<DiskSize>]
```

Usa el volumen como sistema de archivos raíz del contenedor.

--searchdomain <string> - Establece los dominios de búsqueda DNS. Si no se define ni searchdomain ni nameserver, create usa automáticamente la configuración del host.

--ssh-public-keys <filepath> - Configura claves SSH públicas, una por línea en formato OpenSSH.

--start <boolean> (default = 0) - Inicia el CT después de que la creación finalice correctamente.

--startup [[order=]\d+] [,up=\d+] [,down=\d+] - Comportamiento de arranque y parada. order define el orden general; el apagado se realiza en orden inverso. up y down añaden retardos en segundos entre invitados.

--storage <storage ID> (default = local) - Almacenamiento predeterminado.

--swap <integer> (0 - N) (default = 512) - Cantidad de SWAP en MB.

--tags <string> - Etiquetas del contenedor; son únicamente metadatos.

--template <boolean> (default = 0) - Activa o desactiva el estado de plantilla.

--timezone <string> - Zona horaria del contenedor. Puede ser host para igualar la del host o cualquier valor válido de /usr/share/zoneinfo/zone.tab.

--tty <integer> (0 - 6) (default = 2) - Número de TTY disponibles para el contenedor.

--unique <boolean> - Asigna una dirección Ethernet aleatoria única.

NOTA: La opción --unique requiere restore.

--unprivileged <boolean> (default = 0) - Ejecuta el contenedor como usuario no privilegiado. En creación, el valor predeterminado efectivo es 1; en restauración, se toma del backup. No debería modificarse manualmente.

--unused[n] [volume=]<volume> - Referencia a volúmenes no utilizados. Se usa internamente y no debe modificarse manualmente.

Snapshots, destrucción y acceso

pct delsnapshot

```
pct delsnapshot <vmid> <snapname> [OPTIONS]
```

Elimina un snapshot LXC.

--force <boolean> - Elimina la referencia de la configuración incluso si falla el borrado de los snapshots de disco.

pct destroy

```
pct destroy <vmid> [OPTIONS]
```

Destruye el contenedor y elimina todos sus ficheros asociados.

--destroy-unreferenced-disks <boolean> - Elimina además los discos con ese VMID presentes en todos los almacenamientos habilitados que no estén referenciados por la configuración.

--force <boolean> (default = 0) - Fuerza la destrucción incluso si el contenedor está en ejecución.

--purge <boolean> (default = 0) - Elimina el contenedor de configuraciones relacionadas, como trabajos de backup, replicación o HA. Las ACL y reglas de firewall asociadas se eliminan siempre.

pct df

```
pct df <vmid>
```

Muestra el uso de disco actual del contenedor.

pct enter

```
pct enter <vmid> [OPTIONS]
```

Abre una shell dentro del contenedor indicado.

--keep-env <boolean> (default = 1) - Conserva el entorno actual. El manual indica que esta opción se desactivará por defecto en PVE 9; si depende de preservar el entorno, debe especificarla explícitamente para mantener compatibilidad futura.

pct exec

```
pct exec <vmid> [<extra-args>] [OPTIONS]
```

Ejecuta un comando dentro del contenedor.

<extra-args>: <array> - Argumentos adicionales como array.

--keep-env <boolean> (default = 1) - Conserva el entorno actual. El manual indica que se desactivará por defecto en PVE 9; especifíquela si necesita ese comportamiento.

Comprobación y mantenimiento del sistema de archivos

pct fsck

```
pct fsck <vmid> [OPTIONS]
```

Ejecuta una comprobación del sistema de archivos (fsck) sobre un volumen del contenedor.

```
--device <mp0 | mp1 | ... | mp255 | rootfs>
```

Volumen sobre el que se ejecutará la comprobación. El CLI generado enumera mp0 a mp255 y rootfs como valores válidos.

--force <boolean> (default = 0) - Fuerza la comprobación incluso si el sistema de archivos parece limpio.

pct fstrim

```
pct fstrim <vmid> [OPTIONS]
```

Ejecuta fstrim sobre el CT elegido y sus puntos de montaje, excepto los bind mounts y los puntos de montaje de solo lectura.

--ignore-mountpoints <boolean> - Omite todos los puntos de montaje y ejecuta fstrim únicamente sobre el rootfs del contenedor.

pct help / list / listsnapshot

```
pct help [OPTIONS]
pct list
pct listsnapshot <vmid>
```

--extra-args <array> - Muestra ayuda para un comando concreto.

--verbose <boolean> - Formato de salida detallado.

pct list muestra el índice de contenedores LXC del nodo. pct listsnapshot lista todos los snapshots del contenedor indicado.

Migración y montaje

pct migrate

```
pct migrate <vmid> <target> [OPTIONS]
```

Migra el contenedor a otro nodo y crea una nueva tarea de migración.

<target>: <string> - Nodo de destino.

--bwlimit <number> (0 - N) - Sobrescribe el límite de ancho de banda de E/S de la migración en KiB/s.

--online <boolean> - Utiliza migración online/live.

--restart <boolean> - Utiliza migración con reinicio.

--target-storage <string> - Mapeo de almacenamientos origen-destino. Un único storage ID mapea todos los almacenamientos al mismo destino; el valor especial 1 conserva cada almacenamiento en sí mismo.

--timeout <integer> (default = 180) - Tiempo de espera en segundos para el apagado durante una migración con reinicio.

pct mount

```
pct mount <vmid>
```

Monta el sistema de archivos del contenedor en el host. Mantiene un bloqueo sobre el contenedor y está pensado únicamente para mantenimiento de emergencia, ya que impide otras operaciones salvo start y stop.

pct move-volume

```
pct move-volume <vmid> <volume> [<storage>] [<target-vmid>] [<target-volume>] [OPTIONS]
```

Mueve un volumen rootfs/mp a otro almacenamiento o a otro contenedor.

<volume>: <mp0 | mp1 | ... | mp255 | rootfs | unused0 | ... | unused255> - Volumen que se moverá. La documentación generada enumera explícitamente mp0 a mp255, rootfs y unused0 a unused255.

<storage>: <storage ID> - Almacenamiento de destino.

<target-vmid>: <integer> (100 - 999999999) - ID único del contenedor de destino.

<target-volume>: <mp0 | mp1 | ... | mp255 | rootfs | unused0 | ... | unused255> - Clave de configuración a la que se moverá el volumen. De forma predeterminada se conserva la clave de volumen de origen.

--bwlimit <number> (0 - N) - Sobrescribe el límite de ancho de banda de E/S en KiB/s; por defecto utiliza el límite de clone del Datacenter o del almacenamiento.

--delete <boolean> (default = 0) - Elimina el volumen original después de una copia correcta. Por defecto se conserva como entrada unused.

--digest <string> - Impide cambios si la configuración actual tiene un digest SHA1 diferente; permite detectar modificaciones concurrentes.

--target-digest <string> - Impide cambios si la configuración del contenedor de destino tiene un digest SHA1 diferente; permite detectar modificaciones concurrentes.

pct move_volume

```
pct move_volume
```

Alias de pct move-volume.

Configuración pendiente y transferencia de ficheros

pct pending

```
pct pending <vmid>
```

Obtiene la configuración del contenedor, incluidos los cambios pendientes.

pct pull

```
pct pull <vmid> <path> <destination> [OPTIONS]
```

Copia un fichero desde el contenedor al sistema local.

<path>: <string> - Ruta del fichero dentro del contenedor.

<destination>: <string> - Destino local.

--group <string> - Nombre o ID del grupo propietario.

--perms <string> - Permisos del fichero; octal por defecto o hexadecimal con prefijo 0x.

--user <string> - Nombre o ID del usuario propietario.

pct push

```
pct push <vmid> <file> <destination> [OPTIONS]
```

Copia un fichero local al contenedor.

<file>: <string> - Ruta al fichero local.

<destination>: <string> - Destino dentro del contenedor.

--group <string> - Nombre o ID del grupo propietario. Si se usa un nombre, debe existir dentro del contenedor.

--perms <string> - Permisos del fichero; octal por defecto o hexadecimal con prefijo 0x.

--user <string> - Nombre o ID del usuario propietario. Si se usa un nombre, debe existir dentro del contenedor.

pct reboot

```
pct reboot <vmid> [OPTIONS]
```

Reinicia el contenedor apagándolo y arrancándolo de nuevo. Aplica los cambios pendientes.

--timeout <integer> (0 - N) - Tiempo máximo de espera, en segundos, para el apagado.

pct remote-migrate

```
pct remote-migrate <vmid> [<target-vmid>] <target-endpoint> --target-bridge <string>
--target-storage <string> [OPTIONS]
```

Migra el contenedor a un clúster remoto. Crea una nueva tarea de migración. Función EXPERIMENTAL.

<target-endpoint> - Endpoint remoto con el formato apitoken=<PVEAPIToken=user@realm!token=SECRET>,host=<ADDRESS>[,fingerprint=<FINGERPRINT>][,port=<PORT>].

--bwlimit <integer> (0 - N) - Sobrescribe el límite de ancho de banda de E/S de migración en KiB/s.

--delete <boolean> (default = 0) - Elimina el CT original y los datos relacionados después de una migración correcta. Por defecto se conserva detenido en el clúster de origen.

--online <boolean> - Utiliza migración online/live.

--restart <boolean> - Utiliza migración con reinicio.

--target-bridge <string> - Mapeo de bridges origen-destino. Un único bridge ID mapea todos al mismo destino; el valor especial 1 conserva cada bridge.

--target-storage <string> - Mapeo de almacenamientos origen-destino. Un único ID mapea todos al mismo destino; el valor especial 1 conserva cada almacenamiento.

--timeout <integer> (default = 180) - Tiempo de espera en segundos para el apagado durante la migración con reinicio.

pct rescan

```
pct rescan [OPTIONS]
```

Vuelve a explorar todos los almacenamientos y actualiza los tamaños de disco y las imágenes de disco no utilizadas.

--dryrun <boolean> (default = 0) - No escribe realmente cambios en la configuración.

--vmid <integer> (100 - 999999999) - Limita la operación al contenedor indicado.

pct resize

```
pct resize <vmid> <disk> <size> [OPTIONS]
```

Redimensiona un punto de montaje del contenedor.

<disk>: <mp0 | mp1 | ... | mp255 | rootfs> - Disco que se desea redimensionar. La documentación generada enumera mp0 a mp255 y rootfs.

<size>: +?\d+(\.\d+)?[KMGT]? - Nuevo tamaño. Con el signo + el valor se añade al tamaño actual; sin + se interpreta como tamaño absoluto. No se admite reducir el tamaño del disco.

--digest <string> - Impide cambios si la configuración actual tiene un digest SHA1 diferente, para detectar modificaciones concurrentes.

Fin de la Parte A8

Esta entrega cubre A.11 pct desde el inicio de la herramienta hasta pct resize. La siguiente parte continuará con pct restore y el resto de comandos de administración de contenedores LXC.

CRITERIO DE TRADUCCIÓN: Se han conservado literalmente los nombres de comandos, opciones y formatos que deben introducirse en la CLI. En las listas generadas de índices mp0..mp255 y unused0..unused255 se presenta el rango completo de valores válidos de forma compacta para mejorar la legibilidad sin cambiar su alcance.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE A9

Apéndice A · Interfaz de línea de comandos

A.11 pct - Proxmox Container Toolkit · continuación y final

Restauración	Configuración	Ciclo de vida	Mantenimiento
pct restore	pct set	start · shutdown · stop	snapshot · rollback · unlock

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, parámetros, valores y rutas técnicas.

A.11 pct - Proxmox Container Toolkit (continuación)

pct restore <vmid> <ostemplate> [OPTIONS]

Crea o restaura un contenedor a partir de una plantilla de sistema operativo o de un archivo de backup.

<vmid>: <integer> (100 - 999999999) - ID único del contenedor.

<ostemplate>: <string> - Plantilla del sistema operativo o archivo de backup.

--arch <amd64 | arm64 | armhf | i386 | riscv32 | riscv64> (default = amd64) - Arquitectura del sistema operativo.

--bwlimit <number> (0 - N) - Sobrescribe el límite de ancho de banda de E/S para la restauración, en KiB/s.

--cmode <console | shell | tty> (default = tty) - Modo de consola. console se conecta a /dev/console; shell abre directamente un shell; tty intenta usar uno de los TTY disponibles.

--console <boolean> (default = 1) - Adjunta un dispositivo de consola /dev/console al contenedor.

--cores <integer> (1 - 8192) - Número de núcleos asignados. Por defecto el contenedor puede utilizar todos los disponibles.

--cpulimit <number> (0 - 8192) (default = 0) - Límite de uso de CPU. 0 significa sin límite.

--cpuunits <integer> (0 - 500000) - Peso relativo de CPU. En cgroup v2 se limita al intervalo [1, 10000].

--debug <boolean> (default = 0) - Aumenta la verbosidad; actualmente activa el nivel debug durante el arranque.

--description <string> - Descripción del contenedor, visible en el resumen de la interfaz web.

--dev[n] [[path=]<Path>] [,deny-write=<1|0>] [,gid=<integer>] [,mode=<Octal access mode>] [,uid=<integer>] - Dispositivo del host que se pasa al contenedor.

--entrypoint (...) (default = /sbin/init) - Comando que se ejecutará como init. Puede incluir argumentos y ser ruta absoluta, relativa o un binario disponible en $PATH.

--env <NUL-separated list> - Entorno de ejecución del contenedor como lista separada por NUL. Sustituye entradas lxc.environment.runtime de la configuración.

--features [force_rw_sys=<1|0>] [,fuse=<1|0>] [,keyctl=<1|0>] [,mknod=<1|0>] [,mount=<fstype;...>] [,nesting=<1|0>] - Permite acceso a funciones avanzadas del contenedor.

--force <boolean> - Permite sobrescribir un contenedor existente.

--ha-managed <boolean> (default = 0) - Añade el CT como recurso HA después de crearlo.

--hookscript <string> - Script que se ejecutará en distintas fases del ciclo de vida del contenedor.

--hostname <string> - Nombre de host del contenedor.

--ignore-unpack-errors <boolean> - Ignora errores al extraer la plantilla.

--lock <backup | create | destroyed | disk | fstrim | migrate | mounted | rollback | snapshot | snapshot-delete> - Bloquea o desbloquea el contenedor.

--memory <integer> (16 - N) (default = 512) - RAM asignada al contenedor, en MB.

--mp[n] [volume=]<volume> ,mp=<Path> [...] - Configura un punto de montaje del contenedor. Puede reservar un nuevo volumen con STORAGE_ID:SIZE_IN_GiB y admite ACL, backup, idmap, mountoptions, quota, replicate, ro, shared y size.

--nameserver <string> - Servidor DNS del contenedor. Si tampoco se define searchdomain, create/restore puede heredar la configuración del host.

--net[n] name=<string> [...] - Configura una interfaz de red: bridge, firewall, gateways, modo host-managed, MAC, IPv4/IPv6, estado del enlace, MTU, rate, VLAN tag/trunks y tipo veth.

--onboot <boolean> (default = 0) - Indica si el contenedor debe arrancar con el sistema.

--ostype <alpine | archlinux | centos | debian | devuan | fedora | gentoo | nixos | opensuse | ubuntu | unmanaged> - Tipo de sistema operativo. Determina los scripts LXC de preparación. unmanaged omite la configuración específica del SO.

--password <password> - Establece la contraseña de root dentro del contenedor.

--pool <string> - Añade el contenedor al Resource Pool indicado.

--protection <boolean> (default = 0) - Activa protección para impedir el borrado del CT o de sus discos.

--rootfs [volume=]<volume> [...] - Configura el sistema de archivos raíz. Admite ACL, idmap, mountoptions, quota, replicate, ro, shared y size.

--searchdomain <string> - Dominios de búsqueda DNS.

--ssh-public-keys <filepath> - Instala claves SSH públicas, una por línea, en formato OpenSSH.

--start <boolean> (default = 0) - Arranca el CT cuando finaliza correctamente la creación/restauración.

--startup [[order=]N] [,up=N] [,down=N] - Orden y retardos de arranque/apagado.

--storage <storage ID> (default = local) - Almacenamiento predeterminado.

--swap <integer> (0 - N) (default = 512) - SWAP del contenedor, en MB.

--tags <string> - Etiquetas del contenedor; son metadatos.

--template <boolean> (default = 0) - Activa o desactiva el modo plantilla.

--timezone <string> - Zona horaria. Puede usarse host o cualquier zona válida de /usr/share/zoneinfo/zone.tab.

--tty <integer> (0 - 6) (default = 2) - Número de TTY disponibles.

--unique <boolean> - Asigna una dirección Ethernet aleatoria única. Requiere restore.

--unprivileged <boolean> (default = 0) - Ejecuta el contenedor como usuario no privilegiado. Para creación el valor predeterminado es 1; para restore se conserva el valor del backup.

--unused[n] [volume=]<volume> - Referencia interna a volúmenes no utilizados; no debe modificarse manualmente.

NOTA: El manual recalca que --unique requiere la opción restore y que --unprivileged no debería modificarse manualmente al restaurar, salvo que se comprenda exactamente el efecto.

pct resume <vmid>

Reanuda un contenedor suspendido.

<vmid> - ID único del contenedor.

pct rollback <vmid> <snapname> [OPTIONS]

Revierte el estado LXC al snapshot indicado.

<snapname> - Nombre del snapshot.

--start <boolean> (default = 0) - Arranca el contenedor después de una reversión correcta.

pct set <vmid> [OPTIONS]

Modifica las opciones de configuración de un contenedor.

--arch <amd64 | arm64 | armhf | i386 | riscv32 | riscv64> - Arquitectura del sistema operativo.

--cmode <console | shell | tty> (default = tty) - Modo de consola.

--console <boolean> (default = 1) - Adjunta /dev/console.

--cores <integer> (1 - 8192) - Núcleos asignados.

--cpulimit <number> (0 - 8192) (default = 0) - Límite total de CPU; 0 significa sin límite.

--cpuunits <integer> (0 - 500000) - Peso relativo de CPU; en cgroup v2 se limita a [1, 10000].

--debug <boolean> (default = 0) - Activa nivel de log debug en el arranque.

--delete <string> - Lista de parámetros que se desean eliminar.

--description <string> - Descripción del contenedor.

--dev[n] [[path=]<Path>] ... - Dispositivo que se pasa al contenedor.

--digest <string> - Evita cambios si la configuración tiene un SHA1 distinto; sirve para detectar modificaciones concurrentes.

--entrypoint (...) (default = /sbin/init) - Comando de init del contenedor.

--env <NUL-separated list> - Entorno de ejecución del contenedor.

--features [...] - Funciones avanzadas: force_rw_sys, fuse, keyctl, mknod, mount y nesting.

--hookscript <string> - Hookscript del ciclo de vida.

--hostname <string> - Nombre de host.

--lock <...> - Bloquea o desbloquea el CT.

--memory <integer> (16 - N) (default = 512) - RAM en MB.

--mp[n] [volume=]<volume> ,mp=<Path> [...] - Punto de montaje adicional con ACL, backup, idmap, keepattrs, mountoptions, quota, replicate, ro, shared y size.

--nameserver <string> - Servidor DNS.

--net[n] name=<string> [...] - Interfaz de red del CT.

--onboot <boolean> (default = 0) - Arranque automático con el nodo.

--ostype <...> - Tipo de sistema operativo para los scripts de configuración LXC.

--protection <boolean> (default = 0) - Protege el contenedor y sus discos frente a borrado/modificación destructiva.

--revert <string> - Revierte un cambio pendiente.

--rootfs [volume=]<volume> [...] - Sistema de archivos raíz.

--searchdomain <string> - Dominios de búsqueda DNS.

--startup [[order=]N] [,up=N] [,down=N] - Orden y retardos de arranque/apagado.

--swap <integer> (0 - N) (default = 512) - SWAP en MB.

--tags <string> - Etiquetas del contenedor.

--template <boolean> (default = 0) - Convierte o desconvierte en plantilla.
--timezone <string> - Zona horaria del contenedor.
--tty <integer> (0 - 6) (default = 2) - Cantidad de TTY.
--unprivileged <boolean> (default = 0) - Ejecuta el CT como no privilegiado.
--unused[n] [volume=]<volume> - Volumen no usado mantenido en la configuración.

pct shutdown <vmid> [OPTIONS]

Apaga limpiamente el contenedor, siguiendo el comportamiento de lxc-stop(1).
--forceStop <boolean> (default = 0) - Asegura que el contenedor termine deteniéndose.
--timeout <integer> (0 - N) (default = 60) - Tiempo máximo de espera, en segundos.

pct snapshot <vmid> <snapname> [OPTIONS]

Crea un snapshot del contenedor.
<snapname> - Nombre del snapshot.
--description <string> - Descripción o comentario textual.

pct start <vmid> [OPTIONS]

Arranca el contenedor.
--debug <boolean> (default = 0) - Activa un nivel de log muy detallado durante el arranque.
--skiplock <boolean> - Ignora locks; solo root puede utilizar esta opción.

pct status <vmid> [OPTIONS]

Muestra el estado del contenedor.
--verbose <boolean> - Formato de salida detallado.

pct stop <vmid> [OPTIONS]

Detiene abruptamente todos los procesos del contenedor.

ADVERTENCIA: A diferencia de pct shutdown, pct stop es una parada brusca y no realiza un apagado limpio del sistema invitado.

--overrule-shutdown <boolean> (default = 0) - Intenta abortar tareas vzshutdown activas antes de detener.
--skiplock <boolean> - Ignora locks; solo root puede usarlo.

pct suspend <vmid>

Suspende el contenedor. Esta función se considera experimental en el manual.

pct template <vmid>

Convierte el contenedor en una plantilla.

pct unlock <vmid>

Desbloquea el contenedor.

pct unmount <vmid>

Desmonta el sistema de archivos del contenedor.

Fin de A.11 · pct - Proxmox Container Toolkit

Con esta Parte A9 queda completada la sección A.11 del Apéndice A. La siguiente sección del manual es A.12 pveam - Proxmox VE Appliance Manager.

CRITERIO DE TRADUCCIÓN: Se han mantenido los comandos, identificadores, sintaxis y valores válidos tal como aparecen en la documentación. Las explicaciones se han traducido al castellano sin añadir comportamiento que no figure en el manual fuente.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE A10

Apéndice A · Interfaz de línea de comandos

A.12 pveam · A.13 pvecm

Plantillas	Clúster	Corosync	QDevice
pveam	pvecm	Links, votos y claves	Quorum externo

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan comandos, nombres de opciones, valores y sintaxis técnica.

A.12 pveam - Proxmox VE Appliance Manager

Gestor de appliances y plantillas de contenedores de Proxmox VE.

```
pveam <COMMAND> [ARGS] [OPTIONS]
```

pveam available [OPTIONS]

Lista las plantillas disponibles.

--section <mail | system | turnkeylinux> - Restringe la lista a la sección especificada.

pveam download <storage> <template>

Descarga una plantilla de appliance.

<storage>: <storage ID> - Almacenamiento en el que se guardará la plantilla.

<template>: <string> - Plantilla que se descargará.

pveam help [OPTIONS]

Muestra ayuda sobre el comando indicado.

--extra-args <array> - Muestra ayuda para un comando concreto.

--verbose <boolean> - Formato de salida detallado.

pveam list <storage>

Obtiene la lista de todas las plantillas presentes en un almacenamiento.

<storage>: <storage ID> - Muestra únicamente las plantillas del almacenamiento indicado.

pveam remove <template_path>

Elimina una plantilla.

<template_path>: <string> - Ruta de la plantilla que se desea eliminar.

pveam update

Actualiza la base de datos de plantillas de contenedores.

A.13 pvecm - Proxmox VE Cluster Manager

Herramienta de administración del clúster de Proxmox VE.

```
pvecm <COMMAND> [ARGS] [OPTIONS]
```

pvecm add <hostname> [OPTIONS]

Añade el nodo actual a un clúster existente.

<hostname>: <string> - Nombre de host o dirección IP de un miembro existente del clúster.

--fingerprint ([A-Fa-f0-9]{2}:){31}[A-Fa-f0-9]{2} - Huella SHA-256 del certificado.

--force <boolean> - No devuelve error si el nodo ya existe.

--link[n] [address=]<IP> [,priority=<integer>] - Dirección y prioridad de un enlace Corosync. Se admiten hasta 8 enlaces: link0..link7.

--nodeid <integer> (1 - N) - ID de nodo asignado a este nodo.

--use_ssh <boolean> - Fuerza el uso de SSH para la unión, incluso si el peer puede realizarla mediante API.

--votes <integer> (0 - N) - Número de votos asignados al nodo.

pvecm addnode <node> [OPTIONS]

Añade un nodo a la configuración del clúster. Esta llamada es para uso interno.

<node>: <string> - Nombre del nodo del clúster.

--apiversion <integer> - Versión JOIN_API_VERSION del nuevo nodo.

--force <boolean> - No devuelve error si el nodo ya existe.

--link[n] [address=]<IP> [,priority=<integer>] - Información de dirección y prioridad de un enlace Corosync; hasta 8 enlaces.

--new_node_ip <string> - Dirección IP del nodo que se añadirá. Se utiliza como alternativa si no se proporcionan enlaces.

--nodeid <integer> (1 - N) - ID de nodo para el nuevo miembro.

--votes <integer> (0 - N) - Número de votos del nodo.

pvecm apiver

Devuelve la versión de la API de unión al clúster disponible en este nodo.

pvecm create <clustername> [OPTIONS]

Genera una nueva configuración de clúster. Si no se proporcionan enlaces, utiliza por defecto la dirección IP local como link 0.

<clustername>: <string> - Nombre del clúster.

--link[n] [address=]<IP> [,priority=<integer>] - Dirección y prioridad de un enlace Corosync. Se admiten link0..link7.

--nodeid <integer> (1 - N) - ID del nodo actual.

--token-coefficient <integer> (0 - N) (default = 125) - Coeficiente utilizado para determinar el timeout del token de Corosync. Consulte corosync.conf(5) para más detalles.

--votes <integer> (1 - N) - Número de votos para este nodo.

pvecm delnode <node>

Elimina un nodo de la configuración del clúster.

<node>: <string> - Nombre del nodo del clúster.

pvecm expected <expected>

Indica a Corosync un nuevo valor de votos esperados.

<expected>: <integer> (1 - N) - Número esperado de votos.

pvecm help [OPTIONS]

Muestra ayuda sobre el comando indicado.

--extra-args <array> - Muestra ayuda para un comando concreto.

--verbose <boolean> - Formato de salida detallado.

pvecm keygen <filename>

Genera una nueva clave criptográfica para Corosync.

<filename>: <string> - Nombre del archivo de salida.

pvecm mtunnel [<extra-args>] [OPTIONS]

Utilizado por la migración de VMs/CT. No debe utilizarse manualmente.

<extra-args>: <array> - Argumentos adicionales como array.

--get_migration_ip <boolean> (default = 0) - Devuelve la IP de migración, si está configurada.

--migration_network <string> - Red de migración utilizada para detectar la IP local de migración.

--run-command <boolean> - Ejecuta un comando utilizando un socket TCP como entrada estándar. Primero se imprimen la dirección IP y el puerto, cada uno en una línea separada, por la salida estándar del comando.

pvecm nodes

Muestra la vista local de los nodos del clúster.

pvecm qdevice remove

Elimina un QDevice configurado.

pvecm qdevice setup <address> [OPTIONS]

Configura el uso de un QDevice.

<address>: <string> - Dirección de red del QDevice externo de Corosync.

--force <boolean> - No devuelve error ante determinadas operaciones potencialmente peligrosas.

--network <string> - Red que debe utilizarse para conectar con el QDevice externo.

pvecm status

Muestra la vista local del estado del clúster.

pvecm updatecerts [OPTIONS]

Actualiza los certificados del nodo y genera todos los archivos y directorios necesarios.

--force <boolean> - Fuerza la generación de un nuevo certificado SSL.

--silent <boolean> - Ignora errores, por ejemplo cuando el clúster no dispone de quorum.

--unmerge-known-hosts <boolean> (default = 0) - Deshace la unificación heredada de SSH known_hosts.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4 · Traducción al castellano

PARTE A11

Apéndice A · Interfaz de línea de comandos

A.14 pvesr · A.15 pveum (ACL, grupos y pools)

Replicación	ACL	Grupos	Pools
pvesr	pveum acl	pveum group	pveum pool

Basado en Proxmox VE Administration Guide, Release 9.2.4

Se conservan literalmente comandos, opciones, identificadores y sintaxis técnica.

A.14 pvesr - Proxmox VE Storage Replication

Herramienta de administración de la replicación de almacenamiento de Proxmox VE.

```
pvesr <COMMAND> [ARGS] [OPTIONS]
```

pvesr create-local-job <id> <target> [OPTIONS]

Crea un nuevo trabajo de replicación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación. Se compone de un Guest ID y un número de trabajo separados por un guion: <GUEST>-<JOBNUM>.

<target>: <string> - Nodo de destino.

--comment <string> - Descripción.

--disable <boolean> - Marca para deshabilitar o desactivar la entrada.

--rate <number> (1 - N) - Límite de velocidad en MB/s (megabytes por segundo), expresado como número de coma flotante.

--remove_job <full | local> - Marca el trabajo para su eliminación. El trabajo elimina todas las instantáneas locales de replicación. Con full también intenta eliminar los volúmenes replicados en el destino y después se elimina a sí mismo del archivo de configuración.

--schedule <string> (default = */15) - Planificación de la replicación de almacenamiento. El formato es un subconjunto de los eventos de calendario de systemd.

--source <string> - Para uso interno, permite detectar si el guest fue apropiado por otro nodo.

pvesr delete <id> [OPTIONS]

Marca un trabajo de replicación para su eliminación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación, con formato <GUEST>-<JOBNUM>.

--force <boolean> (default = 0) - Elimina la entrada jobconfig, pero no realiza la limpieza.

--keep <boolean> (default = 0) - Conserva los datos replicados en el destino; no los elimina.

pvesr disable <id>

Deshabilita un trabajo de replicación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

pvesr enable <id>

Habilita un trabajo de replicación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

pvesr finalize-local-job <id> [<extra-args>] [OPTIONS]

Finaliza un trabajo de replicación. Elimina todas las instantáneas de replicación cuyo timestamp sea distinto de <last_sync>.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

<extra-args>: <array> - Lista de IDs de volumen que deben considerarse.

--last_sync <integer> (0 - N) - Hora de la última sincronización correcta en Unix epoch. Si no se especifica, se eliminan todas las instantáneas de replicación.

pvesr help [OPTIONS]

Muestra ayuda sobre el comando indicado.

--extra-args <array> - Muestra ayuda para un comando concreto.

--verbose <boolean> - Formato de salida detallado.

pvesr list

Lista los trabajos de replicación.

pvesr prepare-local-job <id> [<extra-args>] [OPTIONS]

Prepara el inicio de un trabajo de replicación. Se ejecuta en el nodo de destino antes de comenzar la replicación. Esta llamada es para uso interno y devuelve un objeto JSON por stdout. Primero comprueba si la VM <vmid> reside en el nodo local; si es así, finaliza inmediatamente. Después examina todos los IDs de volumen en busca de instantáneas, elimina las instantáneas de replicación cuyo timestamp sea distinto de <last_sync> y elimina también los volúmenes no utilizados. Devuelve un hash con marcadores booleanos para los volúmenes que conservan instantáneas de replicación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

<extra-args>: <array> - Lista de IDs de volumen que deben considerarse.

--force <boolean> (default = 0) - Permite eliminar todos los volúmenes existentes cuando la lista de volúmenes está vacía.

--last_sync <integer> (0 - N) - Hora de la última sincronización correcta en Unix epoch. Si no se especifica, se eliminan todas las instantáneas de replicación.

--parent_snapname <string> - Nombre de la instantánea.

--scan <string> - Lista de IDs de almacenamiento que deben explorarse en busca de volúmenes obsoletos.

pvesr read <id>

Lee la configuración de un trabajo de replicación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

pvesr run [OPTIONS]

Este método es llamado por el temporizador de systemd y ejecuta todos los trabajos de sincronización, o uno concreto.

--id [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

--mail <boolean> (default = 0) - Envía una notificación por correo electrónico en caso de fallo.

--verbose <boolean> (default = 0) - Muestra registros más detallados por stdout.

pvesr schedule-now <id>

Programa un trabajo de replicación para iniciarse lo antes posible.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

pvesr set-state <vmid> <state>

Establece el estado de replicación del trabajo durante una migración. Esta llamada es para uso interno y acepta el estado del trabajo como objeto JSON.

<vmid>: <integer> (100 - 999999999) - ID único de la VM.

<state>: <string> - Estado del trabajo como cadena JSON decodificada.

pvesr status [OPTIONS]

Lista el estado de todos los trabajos de replicación de este nodo.

--guest <integer> (100 - 999999999) - Muestra únicamente los trabajos de replicación de este guest.

pvesr update <id> [OPTIONS]

Actualiza la configuración de un trabajo de replicación.

<id>: [1-9][0-9]{2,8}-\d{1,9} - ID del trabajo de replicación.

--comment <string> - Descripción.

--delete <string> - Lista de ajustes que se desean eliminar.

--digest <string> - Impide cambios si el archivo de configuración actual tiene un digest diferente. Puede utilizarse para evitar modificaciones concurrentes.

--disable <boolean> - Marca para deshabilitar o desactivar la entrada.

--rate <number> (1 - N) - Límite de velocidad en MB/s como número de coma flotante.

--remove_job <full | local> - Marca el trabajo para su eliminación; full también intenta eliminar los volúmenes replicados del destino.

--schedule <string> (default = */15) - Planificación de la replicación de almacenamiento usando un subconjunto de los eventos de calendario de systemd.

--source <string> - Para uso interno, permite detectar si el guest fue apropiado por otro nodo.

A.15 pveum - Proxmox VE User Manager

Gestor de usuarios y permisos de Proxmox VE. En esta parte se cubren ACL, grupos, contraseña y pools. La gestión de realms continúa en la siguiente entrega.

```
pveum <COMMAND> [ARGS] [OPTIONS]
```

pveum acl delete <path> --roles <string> [OPTIONS]

Actualiza la lista de control de acceso (ACL), añadiendo o eliminando permisos.

<path>: <string> - Ruta de control de acceso.

--groups <string> - Lista de grupos.

--propagate <boolean> (default = 1) - Permite propagar o heredar los permisos.

--roles <string> - Lista de roles.

--tokens <string> - Lista de tokens de API.

--users <string> - Lista de usuarios.

pveum acl list [FORMAT_OPTIONS]

Obtiene las listas de control de acceso (ACL).

pveum acl modify <path> --roles <string> [OPTIONS]

Actualiza la lista de control de acceso (ACL), añadiendo o eliminando permisos.

<path>: <string> - Ruta de control de acceso.

--groups <string> - Lista de grupos.

--propagate <boolean> (default = 1) - Permite propagar o heredar los permisos.

--roles <string> - Lista de roles.

--tokens <string> - Lista de tokens de API.

--users <string> - Lista de usuarios.

Alias de ACL

```
pveum acldel    # alias de pveum acl delete
pveum aclmod    # alias de pveum acl modify
```

pveum group add <groupid> [OPTIONS]

Crea un nuevo grupo.

<groupid>: <string> - No hay descripción disponible en la guía original.

--comment <string> - No hay descripción disponible en la guía original.

pveum group delete <groupid>

Elimina un grupo.

<groupid>: <string> - No hay descripción disponible en la guía original.

pveum group list [FORMAT_OPTIONS]

Índice de grupos.

pveum group modify <groupid> [OPTIONS]

Actualiza los datos de un grupo.

<groupid>: <string> - No hay descripción disponible en la guía original.

--comment <string> - No hay descripción disponible en la guía original.

Alias de grupos

```
pveum groupadd    # alias de pveum group add
pveum groupdel    # alias de pveum group delete
pveum groupmod    # alias de pveum group modify
```

pveum help [OPTIONS]

Muestra ayuda sobre el comando indicado.

--extra-args <array> - Muestra ayuda para un comando concreto.

--verbose <boolean> - Formato de salida detallado.

pveum passwd <userid> [OPTIONS]

Cambia la contraseña de un usuario.

<userid>: <string> - ID completo del usuario, con formato name@realm.

--confirmation-password <string> - Contraseña actual del usuario que está realizando el cambio.

pveum pool add <poolid> [OPTIONS]

Crea un nuevo pool.

<poolid>: <string> - No hay descripción disponible en la guía original.

--comment <string> - No hay descripción disponible en la guía original.

pveum pool delete <poolid>

Elimina un pool.

<poolid>: <string> - No hay descripción disponible en la guía original.

pveum pool list [OPTIONS] [FORMAT_OPTIONS]

Lista los pools o devuelve la configuración de un pool.

--poolid <string> - No hay descripción disponible en la guía original.

--type <lxc | qemu | storage> - No hay descripción disponible en la guía original.

NOTA: La opción --type requiere que se especifique --poolid.

pveum pool modify <poolid> [OPTIONS]

Actualiza un pool.

<poolid>: <string> - No hay descripción disponible en la guía original.

--allow-move <boolean> (default = 0) - Permite añadir un guest aunque ya pertenezca a otro pool. El guest se elimina de su pool actual y se añade a este.

--comment <string> - No hay descripción disponible en la guía original.

--delete <boolean> (default = 0) - Elimina los VMID y/o IDs de almacenamiento proporcionados en lugar de añadirlos.

--storage <string> - Lista de IDs de almacenamiento que se añadirán o eliminarán de este pool.

--vms <string> - Lista de VMID de guests que se añadirán o eliminarán de este pool.

CONTINUACIÓN: La siguiente parte comienza con pveum realm add <realm> --type <string> [OPTIONS] y continúa con la gestión de realms de autenticación.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE A12

Apéndice A · Interfaz de línea de comandos

A.15 pveum - Proxmox VE User Manager (continuación y final)

Traducción al castellano basada en el manual oficial.
Los comandos, nombres de opciones, rutas, identificadores y expresiones de sintaxis se mantienen sin traducir.

A.15 pveum - Gestor de usuarios de Proxmox VE

Continuación de la referencia de pveum. Esta parte cubre realms de autenticación, sincronización, roles, tickets, usuarios, TFA y tokens de API.

Realms de autenticación

`pveum realm add <realm> --type <string> [OPTIONS]`

Añade un servidor o dominio de autenticación.

`<realm>: <string>`

ID del dominio de autenticación.

`--acr-values ^[^\x00-\x1F\x7F <>#"]*$`

Valores Authentication Context Class Reference que se solicita al servidor de autorización para la petición de autenticación.

`--audiences ^[^\x00-\x1F\x7F <>#"]*$`

Lista de audiencias que el emisor OpenID puede incluir y que se aceptan además de client-id.

`--autocreate <boolean> (default = 0)`

Crea automáticamente los usuarios si no existen.

`--base_dn <string>`

Nombre de dominio base LDAP.

`--bind_dn <string>`

Nombre de dominio de enlace (bind DN) LDAP.

`--capath <string> (default = /etc/ssl/certs)`

Ruta al almacén de certificados de la CA.

`--case-sensitive <boolean> (default = 1)`

El nombre de usuario distingue entre mayúsculas y minúsculas.

`--cert <string>`

Ruta al certificado de cliente.

`--certkey <string>`

Ruta a la clave del certificado de cliente.

`--check-connection <boolean> (default = 0)`

Comprueba la conexión bind con el servidor.

`--client-id <string>`

ID de cliente OpenID.

`--client-key <string>`

Clave de cliente OpenID.

`--comment <string>`

Descripción.

`--default <boolean>`

Usa este realm como predeterminado.

`--domain \S+`

Nombre del dominio AD.

`--filter <string>`

Filtro LDAP para la sincronización de usuarios.

`--group_classes <string> (default = groupOfNames, group, univentionGroup, ipausergroup)`

Clases de objeto empleadas para los grupos.

`--group_dn <string>`

Nombre de dominio base LDAP para sincronizar grupos. Si no se establece, se utiliza base_dn.

`--group_filter <string>`

Filtro LDAP para la sincronización de grupos.

`--group_name_attr <string>`

Atributo LDAP que representa el nombre del grupo. Si no se establece o no se encuentra, se usa como nombre el primer valor del DN.

`--groups-autocreate <boolean> (default = 0)`

Crea automáticamente los grupos si no existen.

`--groups-claim (?:[A-Za-z0-9\.\-_]+)`

Claim de OpenID utilizado para obtener los grupos.

`--groups-overwrite <boolean> (default = 0)`

Sobrescribe todos los grupos del usuario al iniciar sesión.

`--issuer-url <string>`

URL del emisor OpenID.

`--mode <ldap | ldap+starttls | ldaps> (default = ldap)`

Modo de protocolo LDAP.

`--password <string>`

Contraseña LDAP para el bind. Se guarda en /etc/pve/priv/realm/<REALM>.pw.

`--port <integer> (1 - 65535)`

Puerto del servidor.

`--prompt (?:none|login|consent|select_account|\S+)`

Indica si el servidor de autorización solicita al usuario final reautenticación o consentimiento.

`--query-userinfo <boolean> (default = 1)`

Habilita la consulta del endpoint userinfo para obtener valores de claims.

`--scopes <string> (default = email profile)`

Scopes o datos de usuario que deben autorizarse y devolverse, por ejemplo email o profile.

`--secure <boolean>`

Usa el protocolo seguro LDAPS. OBSOLETO: utilice mode en su lugar.

`--server1 <string>`

Dirección IP o nombre DNS del servidor.

`--server2 <string>`

Dirección IP o nombre DNS del servidor alternativo.

`--sslversion <tlsv1 | tlsv1_1 | tlsv1_2 | tlsv1_3>`

Versión TLS/SSL de LDAPS. No se recomienda utilizar versiones anteriores a 1.2.

`--sync-defaults-options [enable-new=<1|0>] [,full=<1|0>] [,purge=<1|0>] [,remove-vanished=([acl];[properties];[entry])|none] [,scope=<users|groups|both>]`

Opciones predeterminadas que controlan el comportamiento de las sincronizaciones.

`--sync_attributes \w+=[^,]+(,\s*\w+=[^,]+)*`

Lista separada por comas de pares key=value que indica qué atributos LDAP se asignan a cada campo de usuario de PVE. Por ejemplo, email=mail. Por defecto, cada campo de usuario de PVE se corresponde con un atributo LDAP del mismo nombre.

`--tfa type=<TFATYPE> [,digits=<COUNT>] [,id=<ID>] [,key=<KEY>] [,step=<SECONDS>] [,url=<URL>]`

Usa autenticación de dos factores.

`--type <ad | ldap | openid | pam | pve>`

Tipo de realm.

`--user_attr \S{2,}`

Nombre del atributo LDAP de usuario.

`--user_classes <string> (default = inetorgperson, posixaccount, person, user)`

Clases de objeto para usuarios.

`--username-claim <string>`

Claim de OpenID utilizado para generar el nombre de usuario único.

`--verify <boolean> (default = 0)`

Verifica el certificado SSL del servidor.

`pveum realm delete <realm>`

Elimina un servidor de autenticación.

`<realm>: <string>`

ID del dominio de autenticación.

`pveum realm list [FORMAT_OPTIONS]`

Muestra el índice de dominios de autenticación.

`pveum realm modify <realm> [OPTIONS]`

Actualiza la configuración de un servidor de autenticación.

`<realm>: <string>`

ID del dominio de autenticación.

`--acr-values ^[^\x00-\x1F\x7F <>#"]*$`

Valores Authentication Context Class Reference solicitados al servidor de autorización.

`--audiences ^[^\x00-\x1F\x7F <>#"]*$`

Audiencias aceptadas además de client-id.

`--autocreate <boolean> (default = 0)`

Crea automáticamente los usuarios si no existen.

`--base_dn <string>`

Nombre de dominio base LDAP.

`--bind_dn <string>`

Nombre de dominio de enlace LDAP.

`--capath <string> (default = /etc/ssl/certs)`

Ruta al almacén de certificados de la CA.

`--case-sensitive <boolean> (default = 1)`

El nombre de usuario distingue entre mayúsculas y minúsculas.

`--cert <string>`

Ruta al certificado de cliente.

`--certkey <string>`

Ruta a la clave del certificado de cliente.

`--check-connection <boolean> (default = 0)`

Comprueba la conexión bind al servidor.

`--client-id <string>`

ID de cliente OpenID.

`--client-key <string>`

Clave de cliente OpenID.

`--comment <string>`

Descripción.

`--default <boolean>`

Usa este realm como predeterminado.

`--delete <string>`

Lista de ajustes que desea eliminar.

`--digest <string>`

Evita cambios si el fichero de configuración actual tiene un digest diferente. Puede utilizarse para impedir modificaciones concurrentes.

`--domain \S+`

Nombre del dominio AD.

`--filter <string>`

Filtro LDAP para sincronizar usuarios.

`--group_classes <string> (default = groupOfNames, group, univentionGroup, ipausergroup)`

Clases de objeto de los grupos.

`--group_dn <string>`

Nombre de dominio base LDAP para sincronizar grupos. Si no se establece, se utiliza base_dn.

`--group_filter <string>`

Filtro LDAP para sincronizar grupos.

`--group_name_attr <string>`

Atributo LDAP que representa el nombre del grupo. Si no se establece o no se encuentra, se usa el primer valor del DN.

`--groups-autocreate <boolean> (default = 0)`

Crea automáticamente los grupos si no existen.

`--groups-claim (?:[A-Za-z0-9\.\-_]+)`

Claim de OpenID utilizado para recuperar grupos.

`--groups-overwrite <boolean> (default = 0)`

Sobrescribe todos los grupos del usuario al iniciar sesión.

`--issuer-url <string>`

URL del emisor OpenID.

`--mode <ldap | ldap+starttls | ldaps> (default = ldap)`

Modo de protocolo LDAP.

`--password <string>`

Contraseña LDAP del bind. Se almacena en /etc/pve/priv/realm/<REALM>.pw.

`--port <integer> (1 - 65535)`

Puerto del servidor.

`--prompt (?:none|login|consent|select_account|\S+)`

Controla si el servidor de autorización pide reautenticación o consentimiento al usuario final.

`--query-userinfo <boolean> (default = 1)`

Habilita la consulta del endpoint userinfo para obtener claims.

`--scopes <string> (default = email profile)`

Scopes que se autorizan y devuelven, por ejemplo email o profile.

`--secure <boolean>`

Usa LDAPS. OBSOLETO: utilice mode en su lugar.

`--server1 <string>`

Dirección IP o nombre DNS del servidor.

`--server2 <string>`

Dirección IP o nombre DNS del servidor alternativo.

`--sslversion <tlsv1 | tlsv1_1 | tlsv1_2 | tlsv1_3>`

Versión TLS/SSL de LDAPS. No se recomienda una versión anterior a 1.2.

`--sync-defaults-options [enable-new=<1|0>] [,full=<1|0>] [,purge=<1|0>] [,remove-vanished=([acl];[properties];[entry])|none] [,scope=<users|groups|both>]`

Opciones predeterminadas para las sincronizaciones.

`--sync_attributes \w+=[^,]+(,\s*\w+=[^,]+)*`

Pares key=value separados por comas para asignar atributos LDAP a campos de usuario PVE. Ejemplo: email=mail.

`--tfa type=<TFATYPE> [,digits=<COUNT>] [,id=<ID>] [,key=<KEY>] [,step=<SECONDS>] [,url=<URL>]`

Usa autenticación de dos factores.

`--user_attr \S{2,}`

Nombre del atributo LDAP de usuario.

`--user_classes <string> (default = inetorgperson, posixaccount, person, user)`

Clases de objeto de los usuarios.

`--verify <boolean> (default = 0)`

Verifica el certificado SSL del servidor.

`pveum realm sync <realm> [OPTIONS]`

Sincroniza usuarios y/o grupos desde el LDAP configurado hacia user.cfg. Los grupos sincronizados tendrán el nombre name-$realm; asegúrese de que esos grupos no existan para evitar sobrescrituras.

`<realm>: <string>`

ID del dominio de autenticación.

`--dry-run <boolean> (default = 0)`

Si se establece, no escribe ningún cambio.

`--enable-new <boolean> (default = 1)`

Habilita inmediatamente los usuarios recién sincronizados.

`--full <boolean>`

OBSOLETO: utilice remove-vanished. Si se establece, LDAP se toma como fuente de verdad: elimina usuarios o grupos no devueltos por la sincronización y retira propiedades modificadas localmente de usuarios sincronizados.

`--purge <boolean>`

OBSOLETO: utilice remove-vanished. Elimina ACL de usuarios o grupos retirados de la configuración durante una sincronización.

`--remove-vanished ([acl];[properties];[entry])|none (default = none)`

Lista separada por punto y coma de elementos que deben eliminarse cuando desaparecen de la sincronización: entry elimina el usuario/grupo; properties elimina propiedades no presentes en el origen; acl elimina las ACL. También puede usarse none.

`--scope <both | groups | users>`

Selecciona qué debe sincronizarse.

Roles y tickets

`pveum role add <roleid> [OPTIONS]`

Crea un nuevo rol.

`<roleid>: <string>`

Sin descripción disponible.

`--privs <string>`

Sin descripción disponible.

`pveum role delete <roleid>`

Elimina un rol.

`<roleid>: <string>`

Sin descripción disponible.

`pveum role list [FORMAT_OPTIONS]`

Muestra el índice de roles.

`pveum role modify <roleid> [OPTIONS]`

Actualiza un rol existente.

`<roleid>: <string>`

Sin descripción disponible.

`--append <boolean>`

Sin descripción disponible.

Requiere la opción: privs

`--privs <string>`

Sin descripción disponible.

`pveum roleadd`

Alias de pveum role add.

`pveum roledel`

Alias de pveum role delete.

`pveum rolemod`

Alias de pveum role modify.

`pveum ticket <username> [OPTIONS]`

Crea o verifica un ticket de autenticación.

`<username>: <string>`

Nombre de usuario.

`--new-format <boolean> (default = 1)`

Este parámetro se ignora actualmente y se asume el valor 1.

`--otp <string>`

Contraseña de un solo uso para autenticación de dos factores.

`--path <string>`

Verifica el ticket y comprueba si el usuario tiene privilegios de acceso sobre la ruta.

Requiere la opción: privs

`--privs <string>`

Verifica el ticket y comprueba si el usuario dispone de privilegios de acceso sobre la ruta.

Requiere la opción: path

`--realm <string>`

Permite indicar opcionalmente el realm. Normalmente se añade al usuario como <username>@<realm>.

`--tfa-challenge <string>`

Cadena firmada del desafío TFA a la que desea responder el usuario.

Usuarios

`pveum user add <userid> [OPTIONS]`

Crea un nuevo usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`--comment <string>`

Sin descripción disponible.

`--email <string>`

Sin descripción disponible.

`--enable <boolean> (default = 1)`

Habilita la cuenta. Puede establecerse a 0 para deshabilitarla.

`--expire <integer> (0 - N)`

Fecha de expiración de la cuenta en segundos desde epoch. 0 significa sin fecha de expiración.

`--firstname <string>`

Sin descripción disponible.

`--groups <string>`

Sin descripción disponible.

`--keys [0-9a-zA-Z!=]{0,4096}`

Claves para autenticación de dos factores (yubico).

`--lastname <string>`

Sin descripción disponible.

`--password <string>`

Contraseña inicial.

`pveum user delete <userid>`

Elimina un usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`pveum user list [OPTIONS] [FORMAT_OPTIONS]`

Índice de usuarios.

`--enabled <boolean>`

Filtro opcional para la propiedad enable.

`--full <boolean> (default = 0)`

Incluye información de grupos y tokens.

`pveum user modify <userid> [OPTIONS]`

Actualiza la configuración de un usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`--append <boolean>`

Sin descripción disponible.

Requiere la opción: groups

`--comment <string>`

Sin descripción disponible.

`--email <string>`

Sin descripción disponible.

`--enable <boolean> (default = 1)`

Habilita la cuenta. Puede establecerse a 0 para deshabilitarla.

`--expire <integer> (0 - N)`

Fecha de expiración en segundos desde epoch. 0 significa sin expiración.

`--firstname <string>`

Sin descripción disponible.

`--groups <string>`

Sin descripción disponible.

`--keys [0-9a-zA-Z!=]{0,4096}`

Claves para autenticación de dos factores (yubico).

`--lastname <string>`

Sin descripción disponible.

`pveum user permissions [<userid>] [OPTIONS] [FORMAT_OPTIONS]`

Obtiene los permisos efectivos del usuario o token indicado.

`<userid>: (?:^(?:[^\s:/]+)\@(?:[A-Za-z][A-Za-z0-9\.\-_]+)(?:!(?:[A-Za-z][A-Za-z0-9\.\-_]+))?$)`

ID de usuario o ID completo de token de API.

`--path <string>`

Muestra únicamente esta ruta concreta, no todo el árbol.

TFA de usuario

`pveum user tfa delete <userid> [OPTIONS]`

Elimina entradas TFA de un usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`--id <string>`

ID de TFA. Si no se indica, se eliminan todas las entradas TFA.

`pveum user tfa list [<userid>]`

Lista las entradas TFA.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`pveum user tfa unlock <userid>`

Desbloquea la autenticación TFA de un usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

Tokens de API

`pveum user token add <userid> <tokenid> [OPTIONS] [FORMAT_OPTIONS]`

Genera un nuevo token de API para un usuario. IMPORTANTE: devuelve el valor del token; debe guardarse, ya que no puede recuperarse posteriormente.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`<tokenid>: (?:[A-Za-z][A-Za-z0-9\.\-_]+)`

Identificador del token específico del usuario.

`--comment <string>`

Sin descripción disponible.

`--expire <integer> (0 - N) (default = same as user)`

Fecha de expiración del token de API en segundos desde epoch. 0 significa sin expiración.

`--privsep <boolean> (default = 1)`

Restringe los privilegios del token de API mediante ACL separadas (predeterminado), o le concede todos los privilegios del usuario correspondiente.

`pveum user token delete <userid> <tokenid> [FORMAT_OPTIONS]`

Elimina un token de API de un usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`<tokenid>: (?:[A-Za-z][A-Za-z0-9\.\-_]+)`

Identificador del token específico del usuario.

`pveum user token list <userid> [FORMAT_OPTIONS]`

Obtiene los tokens de API del usuario.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`pveum user token modify <userid> <tokenid> [OPTIONS] [FORMAT_OPTIONS]`

Actualiza un token de API de un usuario. Si se establece regenerate, el valor de token devuelto debe guardarse, ya que no pod rá recuperarse posteriormente.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`<tokenid>: (?:[A-Za-z][A-Za-z0-9\.\-_]+)`

Identificador del token específico del usuario.

`--comment <string>`

Sin descripción disponible.

`--delete <string>`

Lista de ajustes que desea eliminar.

`--expire <integer> (0 - N) (default = same as user)`

Fecha de expiración del token en segundos desde epoch. 0 significa sin expiración.

`--privsep <boolean> (default = 1)`

Restringe los privilegios del token mediante ACL separadas o le concede todos los privilegios del usuario correspondiente.

`--regenerate <boolean> (default = 0)`

Regenera el valor secreto del token. Todos los usuarios del secreto anterior perderán el acceso después de esta operación.

`pveum user token permissions <userid> <tokenid> [OPTIONS] [FORMAT_OPTIONS]`

Obtiene los permisos efectivos del token indicado.

`<userid>: <string>`

ID completo del usuario, con formato name@realm.

`<tokenid>: (?:[A-Za-z][A-Za-z0-9\.\-_]+)`

Identificador del token específico del usuario.

`--path <string>`

Muestra únicamente esta ruta concreta, no todo el árbol.

`pveum user token remove`

Alias de pveum user token delete.

`pveum useradd`

Alias de pveum user add.

`pveum userdel`

Alias de pveum user delete.

`pveum usermod`

Alias de pveum user modify.

Fin de A.15 pveum. La siguiente sección del manual es A.16 vzdump - Backup Utility for VMs and Containers.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE A13

Apéndice A · Interfaz de línea de comandos

A.16 vzdump y A.17 ha-manager

Referencia completa de la utilidad de copias de seguridad vzdump y del gestor de alta disponibilidad ha-manager. Los comandos, parámetros, nombres de opciones, identificadores y valores se conservan literalmente; las explicaciones se han traducido al castellano.

A.16 vzdump - Utilidad de backup para VMs y contenedores

```
vzdump help
vzdump {<vmid>} [OPTIONS]
```

Crea una copia de seguridad.

```
<vmid>: <string>
```

ID del sistema invitado del que se desea realizar la copia de seguridad.

```
--all <boolean> (default = 0)
```

Realiza copia de seguridad de todos los sistemas invitados conocidos en este host.

```
--bwlimit <integer> (0 - N) (default = 0)
```

Limita el ancho de banda de E/S (en KiB/s).

```
--compress <0 | 1 | gzip | lzo | zstd> (default = 0)
```

Comprime el archivo de volcado.

```
--dumpdir <string>
```

Guarda los archivos resultantes en el directorio especificado.

```
--exclude <string>
```

Excluye los sistemas invitados indicados (presupone --all).

```
--exclude-path <array>
```

Excluye determinados archivos/directorios mediante patrones glob del shell. Las rutas que comienzan por / se anclan a la raíz del contenedor; las demás coinciden de forma relativa en cada subdirectorio.

```
--fleecing [[enabled=]<1|0>] [,storage=<storage ID>]
```

Opciones para backup fleecing (solo VMs).

```
--ionice <integer> (0 - 8) (default = 7)
```

Establece la prioridad de E/S al usar el planificador BFQ. En backups de VMs en modo snapshot y suspend, solo afecta al compresor. El valor 8 usa prioridad idle; los demás usan best-effort con el valor especificado.

```
--job-id \S+
```

ID del trabajo de backup. Si se establece, el campo de metadatos backup-job de la notificación se ajusta a este valor. Solo root@pam puede definir este parámetro.

```
--lockwait <integer> (0 - N) (default = 180)
```

Tiempo máximo de espera para el bloqueo global, en minutos.

```
--mailnotification <always | failure> (default = always)
```

Obsoleto: use targets/matchers de notificaciones. Indica cuándo enviar una notificación por correo.

```
--mailto <string>
```

Obsoleto: use targets/matchers de notificaciones. Lista separada por comas de direcciones de correo o usuarios que deben recibir las notificaciones.

```
--mode <snapshot | stop | suspend> (default = snapshot)
```

Modo de copia de seguridad.

```
--node <string>
```

Solo se ejecuta si el comando se lanza en este nodo.

```
--notes-template <string>
```

Cadena de plantilla para generar notas del backup. Puede contener variables sustituidas por sus valores. Actualmente se admiten {{cluster}}, {{guestname}}, {{node}} y {{vmid}}. Debe ser una sola línea; los saltos de línea y las barras invertidas deben escaparse como \n y \\, respectivamente.

Nota: Requiere la opción storage.

```
--notification-mode <auto | legacy-sendmail | notification-system> (default = auto)
```

Determina qué sistema de notificaciones se utiliza. legacy-sendmail toma en cuenta mailto/mailnotification y envía correo mediante sendmail. notification-system usa el sistema de notificaciones de PVE e ignora mailto/mailnotification. auto envía correo si mailto está definido; en caso contrario usa el sistema de notificaciones.

```
--pbs-change-detection-mode <data | legacy | metadata>
```

Modo PBS utilizado para detectar cambios de archivos y cambiar el formato de codificación en backups de contenedores.

```
--performance [max-workers=<integer>] [,pbs-entries-max=<integer>]
```

Otros ajustes relacionados con el rendimiento.

```
--pigz <integer> (default = 0)
```

Usa pigz en lugar de gzip cuando N>0. N=1 usa la mitad de los núcleos; N>1 usa N como número de hilos.

```
--pool <string>
```

Realiza backup de todos los sistemas invitados conocidos incluidos en el pool especificado.

```
--protected <boolean>
```

Si es true, marca la(s) copia(s) de seguridad como protegida(s).

Nota: Requiere la opción storage.

```
--prune-backups [keep-all=<1|0>] [,keep-daily=<N>] [,keep-hourly=<N>] [,keep-last=<N>] [,keep-monthly=<N>] [,keep-weekly=<N>] [,keep-yearly=<N>] (default = keep-all=1)
```

Usa estas opciones de retención en lugar de las definidas en la configuración del almacenamiento.

```
--quiet <boolean> (default = 0)
```

Reduce la salida.

```
--remove <boolean> (default = 1)
```

Elimina backups antiguos según prune-backups.

```
--script <string>
```

Usa el script hook especificado.

```
--stdexcludes <boolean> (default = 1)
```

Excluye archivos temporales y logs.

```
--stdout <boolean>
```

Escribe el tar en stdout en lugar de hacerlo en un archivo.

```
--stop <boolean> (default = 0)
```

Detiene los trabajos de backup en ejecución en este host.

```
--stopwait <integer> (0 - N) (default = 10)
```

Tiempo máximo de espera hasta que se detenga un sistema invitado, en minutos.

```
--storage <storage ID>
```

Guarda el archivo resultante en este almacenamiento.

```
--tmpdir <string>
```

Guarda los archivos temporales en el directorio especificado.

```
--zstd <integer> (default = 1)
```

Hilos de Zstd. N=0 usa la mitad de los núcleos disponibles; si N es mayor que 0, se usa N como número de hilos.

A.17 ha-manager - Gestor HA de Proxmox VE

```
ha-manager <COMMAND> [ARGS] [OPTIONS]
```

ha-manager add

```
ha-manager add <sid> [OPTIONS]
```

Crea un nuevo recurso HA.

```
<sid>: <type>:<name>
```

ID del recurso HA. Consta de un tipo de recurso seguido de un nombre específico, separados por dos puntos (por ejemplo, vm:100 / ct:100). Para máquinas virtuales y contenedores se puede usar directamente el ID de la VM o CT como atajo (por ejemplo, 100).

```
--auto-rebalance <boolean> (default = 1)
```

El recurso HA puede migrarse durante el reequilibrado automático.

```
--comment <string>
```

Descripción.

```
--failback <boolean> (default = 1)
```

Migra automáticamente el recurso HA al nodo con mayor prioridad según sus reglas de afinidad de nodos si vuelve a estar disponible un nodo con prioridad superior a la del nodo actual.

```
--group <string>
```

Identificador del grupo HA.

```
--max_relocate <integer> (0 - N) (default = 1)
```

Número máximo de intentos de reubicación del recurso cuando falla su inicio.

```
--max_restart <integer> (0 - N) (default = 1)
```

Número máximo de intentos de reinicio del recurso en un nodo tras fallar el arranque. Al alcanzarlo, HA Manager intentará reubicar el recurso en un nodo elegible.

```
--state <disabled | enabled | ignored | started | stopped> (default = started)
```

Estado solicitado del recurso.

```
--type <ct | vm>
```

Tipo de recurso.

ha-manager config

```
ha-manager config [OPTIONS]
```

Lista los recursos HA.

```
--type <ct | vm>
```

Muestra solo recursos del tipo indicado.

Comandos CRM

```
ha-manager crm-command arm-ha
```

Solicita rearmar la pila HA después de haber sido desarmada.

```
ha-manager crm-command disarm-ha <resource-mode>
```

Solicita desarmar la pila HA, liberando todos los watchdogs en todo el clúster.

```
<resource-mode>: <freeze | ignore>
```

Controla cómo se tratan los recursos gestionados por HA mientras la pila está desarmada. El estado actual de los recursos no cambia. freeze: no se aplican nuevos comandos ni cambios de estado. ignore: los recursos se retiran del seguimiento de HA y pueden administrarse como si no estuvieran gestionados por HA.

```
ha-manager crm-command migrate <sid> <node>
```

Solicita la migración online de un recurso a otro nodo.

```
<sid>: <type>:<name>
```

ID del recurso HA; para VMs y CTs puede usarse simplemente el ID numérico.

```
<node>: <string>
```

Nodo de destino.

```
ha-manager crm-command node-maintenance disable <node>
```

Cambia el estado de la solicitud de mantenimiento del nodo.

```
<node>: <string>
```

Nombre del nodo del clúster.

```
ha-manager crm-command node-maintenance enable <node>
```

Cambia el estado de la solicitud de mantenimiento del nodo.

```
<node>: <string>
```

Nombre del nodo del clúster.

```
ha-manager crm-command relocate <sid> <node>
```

Solicita reubicar el recurso en otro nodo. Detiene el servicio en el nodo antiguo y lo reinicia en el nodo de destino.

```
<sid>: <type>:<name>
```

ID del recurso HA; para VMs y CTs puede usarse simplemente el ID numérico.

```
<node>: <string>
```

Nodo de destino.

```
ha-manager crm-command stop <sid> <timeout>
```

Solicita detener el servicio.

```
<sid>: <type>:<name>
```

ID del recurso HA; para VMs y CTs puede usarse simplemente el ID numérico.

```
<timeout>: <integer> (0 - N)
```

Tiempo de espera en segundos. Si se establece en 0 se realiza una parada forzada.

Grupos HA (obsoleto en favor de las reglas HA)

```
ha-manager groupadd <group> --nodes <string> [OPTIONS]
```

Crea un nuevo grupo HA. Esta función está obsoleta en favor de las reglas HA.

```
<group>: <string>
```

Identificador del grupo HA.

```
--comment <string>
```

Descripción.

```
--nodes <node>[:<pri>]{,<node>[:<pri>]}*
```

Lista de nombres de nodos del clúster con prioridad opcional.

```
--nofailback <boolean> (default = 0)
```

El CRM intenta ejecutar los servicios en el nodo de mayor prioridad. Si aparece un nodo con prioridad superior, el CRM migra el servicio a ese nodo. Activar nofailback evita este comportamiento.

```
--restricted <boolean> (default = 0)
```

Los recursos vinculados a grupos restringidos solo pueden ejecutarse en los nodos definidos por el grupo.

```
--type <group>
```

Tipo de grupo.

```
ha-manager groupconfig
```

Obtiene los grupos HA. Obsoleto en favor de las reglas HA.

```
ha-manager groupremove <group>
```

Elimina la configuración de un grupo HA. Obsoleto en favor de las reglas HA.

```
<group>: <string>
```

Identificador del grupo HA.

```
ha-manager groupset <group> [OPTIONS]
```

Actualiza la configuración de un grupo HA. Obsoleto en favor de las reglas HA.

```
<group>: <string>
```

Identificador del grupo HA.

```
--comment <string>
```

Descripción.

```
--delete <string>
```

Lista de ajustes que se desea eliminar.

```
--digest <string>
```

Impide cambios si el archivo de configuración actual tiene un digest diferente. Puede usarse para evitar modificaciones concurrentes.

```
--nodes <node>[:<pri>]{,<node>[:<pri>]}*
```

Lista de nombres de nodos del clúster con prioridad opcional.

```
--nofailback <boolean> (default = 0)
```

Evita que el CRM migre de vuelta el servicio cuando aparece un nodo con prioridad superior.

```
--restricted <boolean> (default = 0)
```

Los recursos vinculados a grupos restringidos solo pueden ejecutarse en los nodos definidos por el grupo.

Ayuda, alias y eliminación de recursos

```
ha-manager help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
ha-manager migrate
```

Alias de ha-manager crm-command migrate.

```
ha-manager relocate
```

Alias de ha-manager crm-command relocate.

```
ha-manager remove <sid> [OPTIONS]
```

Elimina la configuración de un recurso.

```
<sid>: <type>:<name>
```

ID del recurso HA.

```
--purge <boolean> (default = 1)
```

Elimina este recurso de las reglas que lo referencian; si es el único recurso de la regla, elimina también la regla.

Reglas HA

```
ha-manager rules add <type> <rule> [OPTIONS]
```

Crea una regla HA.

```
<type>: <node-affinity | resource-affinity>
```

Tipo de regla HA.

```
<rule>: <string>
```

Identificador de la regla HA.

Opciones condicionales: type=node-affinity

```
--affinity <negative | positive> (default = positive)
```

Indica si los recursos HA deben colocarse en los nodos indicados (positive) o en cualquier nodo excepto los indicados (negative).

```
--comment <string>
```

Descripción de la regla HA.

```
--disable <boolean>
```

Indica si la regla HA está deshabilitada.

```
--nodes <node>[:<pri>]{,<node>[:<pri>]}*
```

Lista de nombres de nodos del clúster con prioridad opcional.

```
--resources <type>:<name>{,<type>:<name>}*
```

Lista de IDs de recursos HA, formada por tipos de recurso seguidos de nombres específicos separados por dos puntos (por ejemplo, vm:100,ct:101).

```
--strict <boolean> (default = 0)
```

Indica si la regla de afinidad de nodos es estricta o no estricta.

Opciones condicionales: type=resource-affinity

```
--affinity <negative | positive>
```

Indica si los recursos HA deben mantenerse en el mismo nodo (positive) o en nodos separados (negative).

```
--comment <string>
```

Descripción de la regla HA.

```
--disable <boolean>
```

Indica si la regla HA está deshabilitada.

```
--resources <type>:<name>{,<type>:<name>}*
```

Lista de IDs de recursos HA.

```
ha-manager rules config [OPTIONS] [FORMAT_OPTIONS]
```

Obtiene las reglas HA.

```
--resource <string>
```

Limita la lista devuelta a reglas que afectan al recurso especificado.

```
--type <node-affinity | resource-affinity>
```

Limita la lista al tipo de regla especificado.

```
ha-manager rules list [OPTIONS] [FORMAT_OPTIONS]
```

Obtiene las reglas HA.

```
--resource <string>
```

Limita la lista devuelta a reglas que afectan al recurso especificado.

```
--type <node-affinity | resource-affinity>
```

Limita la lista al tipo de regla especificado.

```
ha-manager rules remove <rule>
```

Elimina una regla HA.

```
<rule>: <string>
```

Identificador de la regla HA.

```
ha-manager rules set <type> <rule> [OPTIONS]
```

Actualiza una regla HA.

```
<type>: <node-affinity | resource-affinity>
```

Tipo de regla HA.

```
<rule>: <string>
```

Identificador de la regla HA.

```
--delete <string>
```

Lista de ajustes que se desea eliminar.

```
--digest <string>
```

Impide cambios si el archivo de configuración actual tiene un digest diferente. Puede usarse para evitar modificaciones concurrentes.

Opciones condicionales de rules set: type=node-affinity

```
--affinity <negative | positive> (default = positive)
```

Indica si los recursos HA deben colocarse en los nodos indicados (positive) o en cualquier otro nodo (negative).

```
--comment <string>
```

Descripción de la regla HA.

```
--disable <boolean>
```

Indica si la regla HA está deshabilitada.

```
--nodes <node>[:<pri>]{,<node>[:<pri>]}*
```

Lista de nodos del clúster con prioridad opcional.

```
--resources <type>:<name>{,<type>:<name>}*
```

Lista de IDs de recursos HA.

```
--strict <boolean> (default = 0)
```

Indica si la regla de afinidad de nodos es estricta o no estricta.

Opciones condicionales de rules set: type=resource-affinity

```
--affinity <negative | positive>
```

Indica si los recursos HA deben mantenerse en el mismo nodo (positive) o en nodos separados (negative).

```
--comment <string>
```

Descripción de la regla HA.

```
--disable <boolean>
```

Indica si la regla HA está deshabilitada.

```
--resources <type>:<name>{,<type>:<name>}*
```

Lista de IDs de recursos HA.

ha-manager set

```
ha-manager set <sid> [OPTIONS]
```

Actualiza la configuración del recurso.

```
<sid>: <type>:<name>
```

ID del recurso HA.

```
--auto-rebalance <boolean> (default = 1)
```

El recurso HA puede migrarse durante el reequilibrado automático.

```
--comment <string>
```

Descripción.

```
--delete <string>
```

Lista de ajustes que se desea eliminar.

```
--digest <string>
```

Impide cambios si el archivo de configuración actual tiene un digest diferente. Puede usarse para evitar modificaciones concurrentes.

```
--failback <boolean> (default = 1)
```

Migra automáticamente el recurso HA al nodo con mayor prioridad según sus reglas de afinidad cuando dicho nodo vuelve a estar disponible.

```
--group <string>
```

Identificador del grupo HA.

```
--max_relocate <integer> (0 - N) (default = 1)
```

Número máximo de intentos de reubicación del recurso cuando falla su inicio.

```
--max_restart <integer> (0 - N) (default = 1)
```

Número máximo de intentos de reinicio del recurso en un nodo después de fallar el arranque. Al alcanzarlo, HA Manager intentará reubicarlo en un nodo elegible.

```
--state <disabled | enabled | ignored | started | stopped> (default = started)
```

Estado solicitado del recurso.

ha-manager status

```
ha-manager status [OPTIONS]
```

Muestra el estado del gestor HA.

```
--verbose <boolean> (default = 0)
```

Salida detallada. Incluye el estado completo de CRM y LRM en JSON.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE B1

Apéndice B · Daemons de servicio

B.1 a B.9 · Referencia completa

Referencia en castellano de los principales daemons y servicios de Proxmox VE incluidos en el Apéndice B. Los nombres de comandos, opciones y servicios se conservan literalmente.

Apéndice B · Daemons de servicio

B.1 pve-firewall - Daemon del firewall de Proxmox VE

```
pve-firewall <COMMAND> [ARGS] [OPTIONS]
pve-firewall compile
```

Compila e imprime las reglas del firewall. Resulta útil para realizar pruebas.

```
pve-firewall help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pve-firewall localnet
```

Muestra información sobre la red local.

```
pve-firewall restart
```

Reinicia el servicio de firewall de Proxmox VE.

```
pve-firewall simulate [OPTIONS]
```

Simula las reglas del firewall. No simula la tabla de enrutamiento del kernel; simplemente supone que es posible enrutar desd e la zona de origen a la zona de destino.

```
--dest <string>
```

Dirección IP de destino.

```
--dport <integer>
```

Puerto de destino.

```
--from (host|outside|vm\d+|ct\d+|([a-zA-Z][a-zA-Z0-9]{0,9})/(\S+)) (default = outside)
```

Zona de origen.

```
--protocol (tcp|udp) (default = tcp)
```

Protocolo.

```
--source <string>
```

Dirección IP de origen.

```
--sport <integer>
```

Puerto de origen.

```
--to (host|outside|vm\d+|ct\d+|([a-zA-Z][a-zA-Z0-9]{0,9})/(\S+)) (default = host)
```

Zona de destino.

```
--verbose <boolean> (default = 0)
```

Salida detallada.

```
pve-firewall start [OPTIONS]
```

Inicia el servicio de firewall de Proxmox VE.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pve-firewall status
```

Obtiene el estado del firewall.

```
pve-firewall stop
```

Detiene el servicio de firewall de Proxmox VE. Tenga en cuenta que al detenerlo se eliminan activamente todas las reglas iptables relacionadas con Proxmox VE, lo que puede dejar el host sin protección.

B.2 pvedaemon - Daemon de la API de Proxmox VE

```
pvedaemon <COMMAND> [ARGS] [OPTIONS]
pvedaemon help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pvedaemon restart
```

Reinicia el daemon, o lo inicia si no está en ejecución.

```
pvedaemon start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pvedaemon status
```

Obtiene el estado del daemon.

```
pvedaemon stop
```

Detiene el daemon.

B.3 pveproxy - Daemon proxy de la API de Proxmox VE

```
pveproxy <COMMAND> [ARGS] [OPTIONS]
pveproxy help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pveproxy restart
```

Reinicia el daemon, o lo inicia si no está en ejecución.

```
pveproxy start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pveproxy status
```

Obtiene el estado del daemon.

```
pveproxy stop
```

Detiene el daemon.

B.4 pvestatd - Daemon de estado de Proxmox VE

```
pvestatd <COMMAND> [ARGS] [OPTIONS]
pvestatd help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pvestatd restart
```

Reinicia el daemon, o lo inicia si no está en ejecución.

```
pvestatd start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pvestatd status
```

Obtiene el estado del daemon.

```
pvestatd stop
```

Detiene el daemon.

B.5 spiceproxy - Servicio proxy SPICE

```
spiceproxy <COMMAND> [ARGS] [OPTIONS]
spiceproxy help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
spiceproxy restart
```

Reinicia el daemon, o lo inicia si no está en ejecución.

```
spiceproxy start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
spiceproxy status
```

Obtiene el estado del daemon.

```
spiceproxy stop
```

Detiene el daemon.

B.6 pmxcfs - Sistema de archivos de clúster de Proxmox

```
pmxcfs [OPTIONS]
```

Opciones de ayuda:

```
-h, --help
```

Muestra las opciones de ayuda.

Opciones de la aplicación:

```
-d, --debug
```

Activa los mensajes de depuración.

```
-f, --foreground
```

No convierte el servidor en daemon; permanece en primer plano.

```
-l, --local
```

Fuerza el modo local: ignora corosync.conf y fuerza quorum.

Este servicio normalmente se inicia y administra mediante las herramientas de systemd. El servicio se denomina pve-cluster.

```
systemctl start pve-cluster
systemctl stop pve-cluster
systemctl status pve-cluster
```

B.7 pve-ha-crm - Daemon Cluster Resource Manager

```
pve-ha-crm <COMMAND> [ARGS] [OPTIONS]
pve-ha-crm help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pve-ha-crm start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pve-ha-crm status
```

Obtiene el estado del daemon.

```
pve-ha-crm stop
```

Detiene el daemon.

B.8 pve-ha-lrm - Daemon Local Resource Manager

```
pve-ha-lrm <COMMAND> [ARGS] [OPTIONS]
pve-ha-lrm help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pve-ha-lrm start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pve-ha-lrm status
```

Obtiene el estado del daemon.

```
pve-ha-lrm stop
```

Detiene el daemon.

B.9 pvescheduler - Daemon planificador de Proxmox VE

```
pvescheduler <COMMAND> [ARGS] [OPTIONS]
pvescheduler help [OPTIONS]
--extra-args <array>
```

Muestra ayuda para un comando concreto.

```
--verbose <boolean>
```

Formato de salida detallado.

```
pvescheduler restart
```

Reinicia el daemon, o lo inicia si no está en ejecución.

```
pvescheduler start [OPTIONS]
```

Inicia el daemon.

```
--debug <boolean> (default = 0)
```

Modo depuración: permanece en primer plano.

```
pvescheduler status
```

Obtiene el estado del daemon.

```
pvescheduler stop
```

Detiene el daemon.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE C1

Apéndice C · Archivos de configuración

C.1 General y C.2 Configuración del Datacenter

Traducción íntegra del Apéndice C. Se conservan literalmente las rutas, nombres de propiedades, valores, expresiones y sintaxis de configuración; las explicaciones se presentan en castellano.

Apéndice C · Archivos de configuración

C.1 General

La mayoría de los archivos de configuración de Proxmox VE se encuentran en el sistema de archivos compartido del clúster montado en /etc/pve.

Hay excepciones, como el archivo de configuración específico de nodo para copias de seguridad situado en /etc/vzdump.conf.

Normalmente, las propiedades de un archivo de configuración se derivan del JSON Schema que también se utiliza para los endpoints de API asociados.

C.1.1 Uso de mayúsculas y formato en los nombres de propiedades

Históricamente, las propiedades más largas (y las subpropiedades) utilizaban con frecuencia snake_case o se escribían como una sola palabra.

Esto probablemente se debe a que la pila de Proxmox VE se desarrolló principalmente en Perl, donde acceder a propiedades con kebab-case requiere comillas adicionales, y también a que en las primeras etapas del desarrollo había menos imposición de estilo, por lo que distintos desarrolladores empleaban convenciones diferentes.

Para las propiedades nuevas, kebab-case es la forma preferida. Está previsto introducir alias para las propiedades snake_case existentes y, a largo plazo, migrar a kebab-case en la API, CLI y archivos de configuración en uso, manteniendo compatibilidad hacia atrás al restaurar una configuración.

C.2 Configuración del Datacenter

El archivo /etc/pve/datacenter.cfg es un archivo de configuración de Proxmox VE. Contiene valores predeterminados a nivel de clúster utilizados por todos los nodos.

C.2.1 Formato del archivo

El archivo utiliza un formato sencillo clave/valor separado por dos puntos. Cada línea tiene el siguiente formato:

```
OPTION: value
```

Las líneas en blanco se ignoran, y las líneas que empiezan por el carácter # se tratan como comentarios y también se ignoran.

C.2.2 Opciones

```
bwlimit: [clone=<LIMIT>] [,default=<LIMIT>] [,migration=<LIMIT>] [,move=<LIMIT>] [,restore=<LIMIT>]
```

Establece límites de ancho de banda de E/S para distintas operaciones, en KiB/s.

```
clone=<LIMIT>
```

Límite de ancho de banda en KiB/s para clonar discos.

```
default=<LIMIT>
```

Límite de ancho de banda predeterminado en KiB/s.

```
migration=<LIMIT>
```

Límite de ancho de banda en KiB/s para migrar invitados, incluido el movimiento de discos locales.

```
move=<LIMIT>
```

Límite de ancho de banda en KiB/s para mover discos.

```
restore=<LIMIT>
```

Límite de ancho de banda en KiB/s para restaurar invitados desde backups.

```
consent-text: <string>
```

Texto de consentimiento que se muestra antes de iniciar sesión.

```
console: <applet | html5 | vv | xtermjs>
```

Selecciona el visor de consola predeterminado. Puede usarse el applet Java integrado (VNC; obsoleto y mapeado a html5), una aplicación externa compatible con virt-viewer (SPICE), un visor VNC basado en HTML5 (noVNC) o un cliente de consola HTML5 (xtermjs). Si el visor seleccionado no está disponible, por ejemplo si SPICE no está activado para la VM, se utiliza noVNC como alternativa.

```
crs: [ha=<basic|static|dynamic>] [,ha-auto-rebalance=<1|0>] [,ha-auto-rebalance-hold-duration=<number>] [,ha-auto-rebalance-margin=<number>] [,ha-auto-rebalance-method=<bruteforce|topsis>] [,ha-auto-rebalance-threshold=<number>] [,ha-rebalance-on-start=<1|0>]
```

Configuración de planificación de recursos del clúster.

```
ha=<basic | dynamic | static> (default = basic)
```

Configura cómo debe seleccionar nodos HA Manager para iniciar o recuperar servicios: basic solo utiliza el número de servicios; static considera la configuración estática de CPU y memoria de los servicios; dynamic considera tanto la configuración estática como el uso dinámico de CPU y memoria.

```
ha-auto-rebalance=<boolean> (default = 0)
```

Indica si debe usarse CRS para equilibrar automáticamente los recursos HA según el desequilibrio actual entre nodos.

```
ha-auto-rebalance-hold-duration=<number> (0 - N) (default = 3)
```

Número de rondas HA durante las cuales debe superarse el umbral de desequilibrio de nodos del clúster antes de activar una migración automática de equilibrado de recursos.

Nota: Requiere la opción ha-auto-rebalance.

```
ha-auto-rebalance-margin=<number> (0 - 100) (default = 10)
```

Mejora relativa mínima, en porcentaje, del desequilibrio entre nodos del clúster necesaria para ejecutar una migración de equilibrado.

Nota: Requiere la opción ha-auto-rebalance.

```
ha-auto-rebalance-method=<bruteforce | topsis> (default = bruteforce)
```

Método utilizado para puntuar las migraciones de equilibrado.

Nota: Requiere la opción ha-auto-rebalance.

```
ha-auto-rebalance-threshold=<number> (0 - 100) (default = 30)
```

Desequilibrio entre nodos del clúster, en porcentaje, que al superarse activa el sistema automático de equilibrado de recursos.

Nota: Requiere la opción ha-auto-rebalance.

```
ha-rebalance-on-start=<boolean> (default = 0)
```

Permite usar CRS para seleccionar un nodo adecuado cuando el estado solicitado de un servicio HA cambia de stop a start.

```
description: <string>
```

Descripción del Datacenter. Se muestra en el panel de notas del Datacenter de la interfaz web y se guarda como comentario dentro del archivo de configuración.

```
email_from: <string>
```

Especifica la dirección de correo desde la que se envían las notificaciones. El valor predeterminado es root@$hostname.

```
fencing: <both | hardware | watchdog> (default = watchdog)
```

Establece el modo de fencing del clúster HA. El modo hardware requiere una configuración válida de dispositivos de fencing en /etc/pve/ha/fence.cfg. Con both se utilizan ambos modos.

Advertencia: hardware y both son EXPERIMENTALES y están en desarrollo (WIP).

```
ha: shutdown_policy=<enum>
```

Configuración HA a nivel de clúster.

```
shutdown_policy=<conditional | failover | freeze | migrate> (default = conditional)
```

Define la política para gestionar servicios HA al apagar o reiniciar un nodo. freeze congela siempre los servicios que perman ecen en el nodo durante el apagado y HA Manager no los recuperará. failover no los marca como congelados, por lo que se recuperarán en otros nodos si el nodo apagado no vuelve rápidamente (< 1 min). conditional elige automáticamente según el tipo de apagado : en un reinicio congela el servicio, mientras que en un apagado lo deja como está, por lo que se recuperará tras unos 2 minuto s. migrate intenta mover todos los servicios en ejecución a otro nodo cuando se solicita un reinicio o apagado; el apagado solo continúa cuando ya no quedan servicios en ejecución en el nodo. Si el nodo vuelve, el servicio se moverá de nuevo al nodo previamente apagado, siempre que no haya ocurrido otra migración, reubicación o recuperación.

```
http_proxy: http://.*
```

Especifica un proxy HTTP externo utilizado para descargas. Ejemplo: http://username:password@host:port/

```
keyboard: <da | de | de-ch | en-gb | en-us | es | fi | fr | fr-be | fr-ca | fr-ch | hu | is | it | ja | lt | mk | nl | no | pl | pt | pt-br | sl | sv | tr>
```

Distribución de teclado predeterminada para el servidor VNC.

```
language: <ar | ca | da | de | en | es | eu | fa | fr | he | hr | it | ja | ka | kr | nb | nl | nn | pl | pt_BR | ru | sl | sv | tr | ukr | zh_CN | zh_TW>
```

Idioma predeterminado de la interfaz gráfica.

```
location: latitude=<number> ,longitude=<number> [,name=<name>]
```

Ubicación del clúster.

```
latitude=<number> (-90 - 90)
```

Latitud de la ubicación de los nodos en grados.

```
longitude=<number> (-180 - 180)
```

Longitud de la ubicación de los nodos en grados.

```
name=<name>
```

Nombre de la ubicación de este nodo.

```
mac_prefix: <string> (default = BC:24:11)
```

Prefijo utilizado para las direcciones MAC generadas automáticamente de los invitados virtuales. El valor predeterminado BC:24:11 es el OUI asignado por IEEE a Proxmox Server Solutions GmbH para un MAC Address Block Large (MA-L). Se permite su uso en redes locales, es decir, redes no accesibles directamente desde la red pública, como una LAN o una red con NAT/Masquerading.

Si se ejecutan varios clústeres que comparten total o parcialmente las redes de sus invitados virtuales, se recomienda encarecidamente ampliar el prefijo MAC predeterminado o generar uno personalizado y válido para reducir la posibilidad de colisiones MAC. Por ejemplo, puede añadirse un hexadecimal adicional diferente al OUI de Proxmox para cada clúster:

BC:24:11:0 para el primero, BC:24:11:1 para el segundo, etc. Como alternativa, también pueden separarse lógicamente las redes de los invitados, por ejemplo mediante VLANs.

Para invitados accesibles públicamente, se recomienda obtener un OUI propio registrado por IEEE o coordinarlo con los administradores de red propios o del proveedor de hosting.

```
max_workers: <integer> (1 - N)
```

Define el número máximo de workers por nodo que pueden iniciarse para acciones como detener todas las VMs o tareas de ha-manager.

```
migration: [type=]<secure|insecure> [,network=<CIDR>]
```

Configuración de migración a nivel de clúster.

```
network=<CIDR>
```

CIDR de la subred utilizada para la migración. Se usa como alternativa para los trabajos de replicación si no se ha definido la red de replicación.

```
type=<insecure | secure> (default = secure)
```

El tráfico de migración se cifra mediante un túnel SSH de forma predeterminada. En redes seguras y completamente privadas puede deshabilitarse para aumentar el rendimiento.

```
migration_unsecure: <boolean>
```

La migración es segura mediante túnel SSH de forma predeterminada. En redes privadas seguras puede deshabilitarse para acelerar la migración. Obsoleto: use la propiedad migration.

```
next-id: [lower=<integer>] [,upper=<integer>]
```

Controla el rango utilizado por el pool de selección automática de VMID libres.

```
lower=<integer> (default = 100)
```

Límite inferior inclusivo del rango next-id libre de la API.

```
upper=<integer> (default = 1000000)
```

Límite superior exclusivo del rango next-id libre de la API.

```
notify: [fencing=<always|never>] [,package-updates=<auto|always|never>] [,replication=<always|never>] [,target-fencing=<TARGET>] [,target-package-updates=<TARGET>] [,target-replication=<TARGET>]
```

Configuración de notificaciones a nivel de clúster.

```
fencing=<always | never>
```

SIN USO: utilice la configuración de notificaciones del Datacenter.

```
package-updates=<always | auto | never> (default = auto)
```

OBSOLETO: utilice la configuración de notificaciones del Datacenter. Controla con qué frecuencia el trabajo diario de actualización envía notificaciones: auto las envía diariamente en sistemas con una suscripción válida; always en cada actualización si hay actualizaciones pendientes nuevas; never nunca envía notificaciones por nuevas actualizaciones pendientes.

```
replication=<always | never>
```

SIN USO: utilice la configuración de notificaciones del Datacenter.

```
target-fencing=<TARGET>
```

SIN USO: utilice la configuración de notificaciones del Datacenter.

```
target-package-updates=<TARGET>
```

SIN USO: utilice la configuración de notificaciones del Datacenter.

```
target-replication=<TARGET>
```

SIN USO: utilice la configuración de notificaciones del Datacenter.

```
registered-tags: <tag>[;<tag>...]
```

Lista de etiquetas que requieren Sys.Modify sobre / para poder asignarlas o eliminarlas. Las etiquetas incluidas aquí que también estén en user-tag-access también requieren Sys.Modify.

```
replication: [type=]<secure|insecure> [,network=<CIDR>]
```

Configuración de replicación a nivel de clúster.

```
network=<CIDR>
```

CIDR de la subred utilizada para los trabajos de replicación.

```
type=<insecure | secure> (default = secure)
```

El tráfico de replicación se cifra mediante un túnel SSH de forma predeterminada. En redes seguras y completamente privadas puede deshabilitarse para aumentar el rendimiento.

```
tag-style: [case-sensitive=<1|0>] [,color-map=<tag>:<hex-color>[:<hex-color-for-text>][;<tag>=...]] [,ordering=<config|alphabetical>] [,shape=<enum>]
```

Opciones de estilo para las etiquetas.

```
case-sensitive=<boolean> (default = 0)
```

Controla si el filtrado de etiquetas únicas durante una actualización distingue entre mayúsculas y minúsculas.

```
color-map=<tag>:<hex-color>[:<hex-color-for-text>][;<tag>=...]
```

Mapeo manual de colores para etiquetas, separado por punto y coma.

`ordering=<alphabetical | config> (default = alphabetical)`

Controla la ordenación de las etiquetas en la interfaz web y en la actualización de la API.

`shape=<circle | dense | full | none> (default = circle)`

Forma de las etiquetas en el árbol de la interfaz web: full dibuja la etiqueta completa; circle muestra solo un círculo con el color de fondo; dense muestra un pequeño rectángulo, útil cuando cada invitado tiene muchas etiquetas; none deshabilita su visualización.

`u2f: [appid=<APPID>] [,origin=<URL>]`

Configuración U2F.

`appid=<APPID>`

Sustituye la URL AppId de U2F. El valor predeterminado es el origen.

`origin=<URL>`

Sustituye el origen U2F. Resulta especialmente útil para nodos individuales con una única URL.

`user-tag-access: [user-allow=<enum>] [,user-allow-list=<tag>[;<tag>...]]`

Opciones de privilegios para las etiquetas que pueden establecer los usuarios.

`user-allow=<existing | free | list | none> (default = free)`

Controla qué etiquetas pueden establecerse o eliminarse en recursos controlados por un usuario, como invitados. Los usuarios con el privilegio Sys.Modify sobre / no tienen restricciones: none no permite etiquetas; list permite las de user-allow-list; existing permite las de la lista y las ya existentes en los recursos; free no aplica restricciones.

`user-allow-list=<tag>[;<tag>...]`

Lista, separada por punto y coma, de etiquetas que los usuarios pueden establecer o eliminar cuando user-allow tiene los valores list o existing.

`webauthn: [allow-subdomains=<1|0>] [,id=<DOMAINNAME>] [,origin=<URL>] [,rp=<RELYING_PARTY>]`

Configuración de WebAuthn.

`allow-subdomains=<boolean> (default = 1)`

Indica si se permite que el origen sea un subdominio en lugar de exigir la URL exacta.

`id=<DOMAINNAME>`

ID del relying party. Debe ser el nombre de dominio sin protocolo, puerto ni ruta. Cambiarlo invalidará las credenciales existentes.

`origin=<URL>`

Origen del sitio. Debe ser una URL https://, o http://localhost. Debe contener la dirección que los usuarios escriben en el navegador para acceder a la interfaz web. Cambiarlo puede invalidar credenciales existentes.

`rp=<RELYING_PARTY>`

Nombre del relying party. Puede ser cualquier identificador de texto. Cambiarlo puede invalidar credenciales existentes.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE D1

Apéndice D · Eventos de calendario

Formato de planificación, especificación y ejemplos

Traducción íntegra del Apéndice D. Se conserva literalmente la sintaxis de expresiones de calendario y los ejemplos de programación.

Apéndice D · Eventos de calendario

D.1 Formato de planificación

Proxmox VE dispone de una configuración de planificación muy flexible. Está basada en el formato de eventos de calendario de tiempo de systemd. Los eventos de calendario pueden utilizarse para hacer referencia a uno o varios instantes temporales mediante una sola expresión.

Un evento de calendario utiliza el siguiente formato:

```
[WEEKDAY] [[YEARS-]MONTHS-DAYS] [HOURS:MINUTES[:SECONDS]]
```

Este formato permite configurar un conjunto de días en los que debe ejecutarse el trabajo. También se pueden establecer una o varias horas de inicio. De este modo, el planificador de replicación conoce los instantes en los que debe comenzar un trabajo .

Por ejemplo, para crear un trabajo que se ejecute todos los días laborables a las 22:00 puede utilizarse:

```
mon,tue,wed,thu,fri 22
```

La misma expresión puede abreviarse como:

```
mon..fri 22
```

De esta forma pueden expresarse de manera bastante intuitiva la mayoría de las planificaciones habituales.

Nota: Las horas se expresan en formato de 24 horas.

Para disponer de una configuración más cómoda y breve, pueden definirse uno o varios intervalos de repetición por invitado. Indican que las replicaciones se ejecutan en la hora o las horas de inicio y, posteriormente, en todos los múltiplos del valor de repetición.

Por ejemplo, para comenzar la replicación a las 08:00 y repetirla cada 15 minutos hasta las 09:00 se utilizaría:

```
8:00/15
```

Si no se utiliza el separador de horas (:), el valor se interpreta como minutos. Si se utiliza, el valor situado a la izquierda representa la hora o las horas y el de la derecha los minutos. También puede utilizarse * para hacer coincidir todos los valores posibles.

En los ejemplos posteriores se muestran más formas de construir planificaciones.

D.2 Especificación detallada

weekdays

Los días se especifican mediante sus abreviaturas en inglés: sun, mon, tue, wed, thu, fri y sat.

Pueden indicarse varios días como una lista separada por comas. También puede definirse un intervalo de días indicando el día inicial y el final separados por '..', por ejemplo mon..fri. Ambos formatos pueden combinarse. Si se omite este campo, se presupone '*'.

time-format

El formato de hora está formado por listas o intervalos de horas y minutos. Las horas y los minutos se separan mediante ':'.

Tanto las horas como los minutos pueden ser listas e intervalos de valores utilizando el mismo formato que para los días. Primero se indican las horas y después los minutos.

Las horas pueden omitirse si no son necesarias; en ese caso se presupone '*'. El intervalo válido es 0-23 para las horas y 0-59 para los minutos.

Nota: Puede usar systemd-analyze calendar para comprobar si una especificación de evento de calendario es válida y cuándo se activará la próxima vez. Con --iterations=<N> puede mostrar también las próximas <N> activaciones, sustituyendo <N> por un entero válido.

D.2.1 Ejemplos

Existen algunos valores especiales con un significado concreto:

Valor	Sintaxis
minutely	*-*-* *:*:00
hourly	*-*-* *:00:00
daily	*-*-* 00:00:00
weekly	mon *-*-* 00:00:00
monthly	*-*-01 00:00:00
yearly o annually	*-01-01 00:00:00
quarterly	*-01,04,07,10-01 00:00:00
semiannually o semi-annually	*-01,07-01 00:00:00

Cadena de planificación	Alternativa	Significado
mon,tue,wed,thu,fri	mon..fri	Todos los días laborables a las 00:00
sat,sun	sat..sun	Solo fines de semana a las 00:00
mon,wed,fri	—	Solo lunes, miércoles y viernes a las 00:00
12:05	12:05	Todos los días a las 12:05
*/5	0/5	Cada cinco minutos
mon..wed 30/10	mon,tue,wed 30/10	Lunes, martes y miércoles, a los minutos 30, 40 y 50 de cada hora
mon..fri 8..17,22:0/15	—	Cada día laborable, cada 15 minutos entre las 08:00 y las 18:00 y entre las 22:00 y las 23:00
fri 12..13:5/20	fri 12,13:5/20	Viernes a las 12:05, 12:25, 12:45, 13:05, 13:25 y 13:45
12,14,16,18,20,22:5	12/2:5	Todos los días desde las 12:05 hasta las 22:05, cada 2 horas
*	*/1	Cada minuto (intervalo mínimo)
*-05	—	El día 5 de cada mes
Sat *-1..7 15:00	—	El primer sábado de cada mes a las 15:00
2015-10-21	—	21 de octubre de 2015 a las 00:00

Referencia del manual: para más información, consulte man 7 systemd.time.

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE E1

Apéndice E · Lista de vCPU de QEMU

Tipos de CPU Intel y AMD

Traducción íntegra del Apéndice E. Se conservan literalmente los nombres de CPU, versiones y flags de QEMU.

Apéndice E · Lista de vCPU de QEMU

E.1 Introducción

Esta es una lista de tipos de CPU AMD e Intel x86-64/amd64 definidos en QEMU, que se remonta a 2007.

E.2 Tipos de CPU Intel

Procesadores Intel

- **Nahelem :** 1.ª generación del procesador Intel Core
- **Nahelem-IBRS (v2) :** añade protección Spectre v1 (+spec-ctrl)
- **Westmere :** 1.ª generación del procesador Intel Core (Xeon E7-)
- **Westmere-IBRS (v2) :** añade protección Spectre v1 (+spec-ctrl)
- **SandyBridge :** 2.ª generación del procesador Intel Core
- **SandyBridge-IBRS (v2) :** añade protección Spectre v1 (+spec-ctrl)
- **IvyBridge :** 3.ª generación del procesador Intel Core
- **IvyBridge-IBRS (v2) :** añade protección Spectre v1 (+spec-ctrl)
- **Haswell :** 4.ª generación del procesador Intel Core
- **Haswell-noTSX (v2) :** deshabilita TSX (-hle, -rtm)
- **Haswell-IBRS (v3) :** vuelve a habilitar TSX y añade protección Spectre v1 (+hle, +rtm, +spec-ctrl)
- **Haswell-noTSX-IBRS (v4) :** deshabilita TSX (-hle, -rtm)
- **Broadwell :** 5.ª generación del procesador Intel Core
- **Skylake :** 1.ª generación de procesadores de servidor Xeon Scalable
- **Skylake-IBRS (v2) :** añade protección Spectre v1 y deshabilita CLFLUSHOPT (+spec-ctrl, -clflushopt)
- **Skylake-noTSX-IBRS (v3) :** deshabilita TSX (-hle, -rtm)
- **Skylake-v4 :** añade conmutación EPT (+vmx-eptp-switching)
- **Cascadelake :** 2.ª generación de procesadores Xeon Scalable
- **Cascadelake-v2 :** añade el MSR arch_capabilities (+arch-capabilities, +rdctl-no, +ibrs-all, +skip-l1dfl-vmentry, +mds-no)
- **Cascadelake-v3 :** deshabilita TSX (-hle, -rtm)
- **Cascadelake-v4 :** añade conmutación EPT (+vmx-eptp-switching)
- **Cascadelake-v5 :** añade XSAVES (+xsaves, +vmx-xsaves)
- **Cooperlake :** 3.ª generación de procesadores Xeon Scalable para servidores de 4 y 8 sockets
- **Cooperlake-v2 :** añade XSAVES (+xsaves, +vmx-xsaves)
- **Icelake :** 3.ª generación de procesadores de servidor Xeon Scalable
- **Icelake-v2 :** deshabilita TSX (-hle, -rtm)
- **Icelake-v3 :** añade el MSR arch_capabilities (+arch-capabilities, +rdctl-no, +ibrs-all, +skip-l1dfl-vmentry, +mds-no, +pschange-mc-no, +taa-no)
- **Icelake-v4 :** añade flags que faltaban (+sha-ni, +avx512ifma, +rdpid, +fsrm, +vmx-rdseed-exit, +vmx-pml, +vmx-eptp-switching)
- **Icelake-v5 :** añade XSAVES (+xsaves, +vmx-xsaves)
- **Icelake-v6 :** añade "EPT de 5 niveles" (+vmx-page-walk-5)
- **SapphireRapids :** 4.ª generación de procesadores de servidor Xeon Scalable

E.3 Tipos de CPU AMD

Procesadores AMD

- **Opteron_G3 :** K10
- **Opteron_G4 :** Bulldozer
- **Opteron_G5 :** Piledriver
- **EPYC :** 1.ª generación de procesadores Zen
- **EPYC-IBPB (v2) :** añade protección Spectre v1 (+ibpb)
- **EPYC-v3 :** añade flags que faltaban (+perfctr-core, +clzero, +xsaveerptr, +xsaves)
- **EPYC-Rome :** 2.ª generación de procesadores Zen
- **EPYC-Rome-v2 :** añade protección frente a Spectre v2 y v4 (+ibrs, +amd-ssbd)
- **EPYC-Milan :** 3.ª generación de procesadores Zen
- **EPYC-Milan-v2 :** añade flags que faltaban (+vaes, +vpclmulqdq, +stibp-always-on, +amd-psfd, +no-nested-data-bp, +lfence-always-serializing, +null-sel-clr-base)

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE F1

Apéndice F · Definiciones de macros de firewall

Amanda → OpenVPN

Primera parte del catálogo de macros predefinidas del firewall. Se conservan literalmente los nombres de las macros, protocolos, puertos, rangos y parámetros; solo se traducen las descripciones.

Apéndice F · Definiciones de macros de firewall

Las siguientes macros agrupan reglas de protocolo y puerto que pueden utilizarse en el firewall de Proxmox VE. Esta parte cub re las entradas desde Amanda hasta OpenVPN.

Amanda Backup de Amanda

```
Action proto dport sport
PARAM udp 10080
PARAM tcp 10080
```

Auth Tráfico Auth (identd)

```
Action proto dport sport
PARAM tcp 113
```

BGP Tráfico Border Gateway Protocol

```
Action proto dport sport
PARAM tcp 179
```

BitTorrent Tráfico BitTorrent para BitTorrent 3.1 y anteriores

```
Action proto dport sport
PARAM tcp 6881:6889
PARAM udp 6881
```

BitTorrent32 Tráfico BitTorrent para BitTorrent 3.2 y posteriores

```
Action proto dport sport
PARAM tcp 6881:6999
PARAM udp 6881
```

CVS Tráfico pserver de Concurrent Versions System

```
Action proto dport sport
PARAM tcp 2401
```

Ceph Tráfico del clúster de almacenamiento Ceph (Ceph Monitors, daemons OSD y MDS)

```
Action proto dport sport
PARAM tcp 6789
PARAM tcp 3300
PARAM tcp 6800:7300
```

Citrix Tráfico Citrix/ICA (ICA, ICA Browser, CGP)

```
Action proto dport sport
PARAM tcp 1494
PARAM udp 1604
PARAM tcp 2598
```

DAAP Tráfico Digital Audio Access Protocol (daemons iTunes, Rythmbox)

```
Action proto dport sport
PARAM tcp 3689
PARAM udp 3689
```

DCC Mecanismo de filtrado de spam Distributed Checksum Clearinghouse

```
Action proto dport sport
PARAM tcp 6277
```

DHCPfwd Tráfico DHCP reenviado

```
Action proto dport sport
PARAM udp 67:68 67:68
```

DHCPv6 Tráfico DHCPv6

```
Action proto dport sport
PARAM udp 546:547 546:547
```

DNS Tráfico Domain Name System (upd y tcp)

```
Action proto dport sport
PARAM udp 53
PARAM tcp 53
```

Distcc Servicio Distributed Compiler

```
Action proto dport sport
PARAM tcp 3632
```

FTP File Transfer Protocol

```
Action proto dport sport
PARAM tcp 21
```

Finger Protocolo Finger (RFC 742)

```
Action proto dport sport
PARAM tcp 79
```

GNUnet Tráfico de red peer-to-peer segura GNUnet

```
Action proto dport sport
PARAM tcp 2086
PARAM udp 2086
PARAM tcp 1080
PARAM udp 1080
```

GRE Protocolo de túnel Generic Routing Encapsulation

```
Action proto dport sport
PARAM 47
```

Git Tráfico del sistema distribuido de control de revisiones Git

```
Action proto dport sport
PARAM tcp 9418
```

HKP Tráfico del protocolo de servidor de claves OpenPGP HTTP

```
Action proto dport sport
PARAM tcp 11371
```

HTTP Hypertext Transfer Protocol (WWW)

```
Action proto dport sport
PARAM tcp 80
```

HTTPS Hypertext Transfer Protocol (WWW) sobre SSL

```
Action proto dport sport
PARAM tcp 443
```

ICPV2 Tráfico Internet Cache Protocol V2 (Squid)

```
Action proto dport sport
PARAM udp 3130
```

ICQ Tráfico AOL Instant Messenger

```
Action proto dport sport
PARAM tcp 5190
```

IMAP Internet Message Access Protocol

```
Action proto dport sport
PARAM tcp 143
```

IMAPS Internet Message Access Protocol sobre SSL

```
Action proto dport sport
PARAM tcp 993
```

IPIP Tráfico de encapsulación IPIP

```
Action proto dport sport
PARAM 94
```

IPsec Tráfico IPsec

```
Action proto dport sport
PARAM udp 500 500
PARAM 50
```

IPsecah Tráfico de autenticación IPsec (AH)

```
Action proto dport sport
PARAM udp 500 500
PARAM 51
```

IPsecnat Tráfico IPsec y Nat-Traversal

```
Action proto dport sport
PARAM udp 500
PARAM udp 4500
PARAM 50
```

IRC Tráfico Internet Relay Chat

```
Action proto dport sport
PARAM tcp 6667
```

Jetdirect Impresión HP Jetdirect

```
Action proto dport sport
PARAM tcp 9100
```

L2TP Tráfico Layer 2 Tunneling Protocol

```
Action proto dport sport
PARAM udp 1701
```

LDAP Tráfico Lightweight Directory Access Protocol

```
Action proto dport sport
PARAM tcp 389
```

LDAPS Tráfico seguro Lightweight Directory Access Protocol

```
Action proto dport sport
PARAM tcp 636
```

MDNS Multicast DNS

```
Action proto dport sport
PARAM udp 5353
```

MSNP Microsoft Notification Protocol

```
Action proto dport sport
PARAM tcp 1863
```

MSSQL Microsoft SQL Server

```
Action proto dport sport
PARAM tcp 1433
```

Mail Tráfico de correo (SMTP, SMTPS, Submission)

```
Action proto dport sport
PARAM tcp 25
PARAM tcp 465
PARAM tcp 587
```

Munin Tráfico de monitorización de recursos en red Munin

```
Action proto dport sport
PARAM tcp 4949
```

MySQL Servidor MySQL

```
Action proto dport sport
PARAM tcp 3306
```

NNTP Tráfico NNTP (Usenet).

```
Action proto dport sport
PARAM tcp 119
```

NNTPS Tráfico NNTP cifrado (Usenet)

```
Action proto dport sport
PARAM tcp 563
```

NTP Network Time Protocol (ntpd)

```
Action proto dport sport
PARAM udp 123
```

NeighborDiscovery solicitud de vecino IPv6 y anuncios de vecino y de router

```
Action proto dport sport
PARAM icmpv6 router-solicitation
PARAM icmpv6 router-advertisement
PARAM icmpv6 neighbor-solicitation
PARAM icmpv6 neighbor-advertisement
```

OSPF Tráfico multicast OSPF

```
Action proto dport sport
PARAM 89
```

OpenVPN Tráfico OpenVPN

```
Action proto dport sport
PARAM udp 1194
```

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE F2

Apéndice F · Definiciones de macros de firewall

De PCA a Whois · Final del Apéndice F

Segunda y última parte de las macros de firewall. Se conservan literalmente nombres, protocolos, puertos, rangos y parámetros; las descripciones se traducen al castellano.

Apéndice F · Definiciones de macros de firewall (continuación)

PCA

Symantec PCAnywere (tm)

```
Action proto dport sport
PARAM udp 5632
PARAM tcp 5631
```

PMG

Interfaz web de Proxmox Mail Gateway

```
Action proto dport sport
PARAM tcp 8006
```

POP3

Tráfico POP3

```
Action proto dport sport
PARAM tcp 110
```

POP3S

Tráfico POP3 cifrado

```
Action proto dport sport
PARAM tcp 995
```

PPtP

Point-to-Point Tunneling Protocol

```
Action proto dport sport
PARAM 47
PARAM tcp 1723
```

Ping

Solicitud echo de ICMP

```
Action proto dport sport
PARAM icmp echo-request
```

PostgreSQL

Servidor PostgreSQL

```
Action proto dport sport
PARAM tcp 5432
```

Printer

Impresión mediante Line Printer protocol

```
Action proto dport sport
PARAM tcp 515
```

RDP

Tráfico de Microsoft Remote Desktop Protocol

```
Action proto dport sport
PARAM tcp 3389
```

RIP

Routing Information Protocol, bidireccional

```
Action proto dport sport
PARAM udp 520
```

RNDC

Protocolo de administración remota de BIND

```
Action proto dport sport
PARAM tcp 953
```

Razor

Sistema antispam Razor

```
Action proto dport sport
PARAM tcp 2703
```

Rdate

Obtención remota de hora (rdate)

```
Action proto dport sport
PARAM tcp 37
```

Rsync

Servidor Rsync

```
Action proto dport sport
PARAM tcp 873
```

SANE

Escaneo de red SANE

```
Action proto dport sport
PARAM tcp 6566
```

SMB

Tráfico Microsoft SMB

```
Action proto dport sport
PARAM udp 135,445
PARAM udp 137:139
PARAM udp 1024:65535 137
PARAM tcp 135,139,445
```

SMBswat

Samba Web Administration Tool

```
Action proto dport sport
PARAM tcp 901
```

SMTP

Simple Mail Transfer Protocol

```
Action proto dport sport
PARAM tcp 25
```

SMTPS

Simple Mail Transfer Protocol cifrado

```
Action proto dport sport
PARAM tcp 465
```

SNMP

Simple Network Management Protocol

```
Action proto dport sport
PARAM udp 161:162
PARAM tcp 161
```

SPAMD

Tráfico SPAMD de SpamAssassin

```
Action proto dport sport
PARAM tcp 783
```

SPICEproxy

Tráfico del proxy de pantalla SPICE de Proxmox VE

```
Action proto dport sport
PARAM tcp 3128
```

SSH

Tráfico Secure Shell

```
Action proto dport sport
PARAM tcp 22
```

SVN

Servidor Subversion (svnserve)

```
Action proto dport sport
PARAM tcp 3690
```

SixXS

SixXS IPv6 Deployment and Tunnel Broker

```
Action proto dport sport
PARAM tcp 3874
PARAM udp 3740
PARAM 41
PARAM udp 5072,8374
```

Squid

Tráfico del proxy web Squid

```
Action proto dport sport
PARAM tcp 3128
```

Submission

Tráfico de envío de mensajes de correo

```
Action proto dport sport
PARAM tcp 587
```

Syslog

Tráfico del protocolo Syslog (RFC 5424)

```
Action proto dport sport
PARAM udp 514
PARAM tcp 514
```

TFTP

Tráfico Trivial File Transfer Protocol

```
Action proto dport sport
PARAM udp 69
```

Telnet

Tráfico Telnet

```
Action proto dport sport
PARAM tcp 23
```

Telnets

Telnet sobre SSL

```
Action proto dport sport
PARAM tcp 992
```

Time

Protocolo de tiempo RFC 868

```
Action proto dport sport
PARAM tcp 37
```

Trcrt

Tráfico Traceroute, hasta 30 saltos

```
Action proto dport sport
PARAM udp 33434:33524
PARAM icmp echo-request
```

VNC

Tráfico VNC para pantallas VNC 0 - 99

```
Action proto dport sport
PARAM tcp 5900:5999
```

VNCL

Tráfico VNC desde Vncservers hacia Vncviewers en modo listen

```
Action proto dport sport
PARAM tcp 5500
```

Web

Tráfico WWW (HTTP y HTTPS)

```
Action proto dport sport
PARAM tcp 80
PARAM tcp 443
```

Webcache

Tráfico Web Cache/Proxy (puerto 8080)

```
Action proto dport sport
PARAM tcp 8080
```

Webmin

Tráfico Webmin

```
Action proto dport sport
PARAM tcp 10000
```

Whois

Tráfico Whois (nicname, RFC 3912)

```
Action proto dport sport
PARAM tcp 43
```

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE G1

Apéndice G · Introducción a Markdown

Sintaxis básica, listas, tablas, citas y código

Traducción íntegra del Apéndice G. Los ejemplos Markdown, URLs, símbolos y bloques de código se conservan literalmente; las explicaciones se presentan en castellano.

Apéndice G · Introducción a Markdown

> *Markdown es una herramienta de conversión de texto a HTML para autores web. Permite escribir utilizando un formato de texto plano fácil de leer y de escribir, y después convertirlo a XHTML (o HTML) estructuralmente válido.*
>
> *— John Gruber · https://daringfireball.net/projects/markdown/*

La interfaz web de Proxmox VE permite utilizar Markdown para representar formato de texto enriquecido en las notas de nodos e invitados virtuales.

Proxmox VE admite CommonMark con la mayoría de las extensiones de GFM (GitHub Flavoured Markdown), como tablas o listas de tareas.

G.1 Conceptos básicos de Markdown

Aquí solo se describen los conceptos básicos. Para información más extensa, el manual recomienda consultar recursos en la web, por ejemplo https://www.markdownguide.org/.

G.1.1 Encabezados

```
# This is a Heading h1
## This is a Heading h2
##### This is a Heading h5
```

G.1.2 Énfasis

Use *text* o _text_ para dar énfasis.

Use **text** o __text__ para texto en negrita o de mayor peso.

También es posible combinar ambos formatos, por ejemplo:

```
_You **can** combine them_
```

G.1.3 Enlaces

Puede utilizar la detección automática de enlaces. Por ejemplo, https://forum.proxmox.com/ se transformará en un enlace en el que se puede hacer clic.

También puede controlar el texto visible del enlace, por ejemplo:

```
Now, [the part in brackets will be the link text](https://forum.proxmox.com/).
```

G.1.4 Listas

Listas sin ordenar

Use * o - para crear listas sin ordenar, por ejemplo:

```
* Item 1
* Item 2
  * Item 2a
  * Item 2b
```

Añadiendo sangría pueden crearse listas anidadas.

Listas ordenadas

```
1. Item 1
1. Item 2
1. Item 3
   1. Item 3a
   1. Item 3b
```

Nota: El número entero utilizado en las listas ordenadas no necesita ser correcto; la numeración se genera automáticamente.

Listas de tareas

Las listas de tareas utilizan una casilla vacía [ ] para las tareas no terminadas y una casilla con una X para las finalizadas. Por ejemplo:

```
- [X] First task already done!
- [X] Second one too
- [ ] This one is still to-do
- [ ] So is this one
```

G.1.5 Tablas

Las tablas utilizan el símbolo de barra vertical | para separar columnas y - para separar el encabezado del cuerpo. En esa línea de separación también puede configurarse la alineación del texto de cada columna a la izquierda, al centro o a la derecha.

```
| 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 | &#x2601;&#xfe0f;&#x2601;&#xfe0f;&#x2601;&#xfe0f; |
| left rab | right rab | And | Last | Here | The End |
```

No es necesario alinear visualmente las columnas mediante espacios en blanco, aunque hacerlo facilita la edición de las tablas.

G.1.6 Citas en bloque

Puede introducir citas en bloque anteponiendo > a una línea, de forma similar a los correos electrónicos en texto plano.

```
> 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.
```

G.1.7 Código y fragmentos

Puede utilizar comillas invertidas (`) para evitar el procesamiento Markdown de unas pocas palabras o párrafos. Esto resulta útil para impedir que un fragmento de código o configuración se interprete por error como Markdown.

Código en línea

Rodear una parte de una línea con una única comilla invertida permite escribir código en línea, por ejemplo:

```
This hosts IP address is `10.0.0.1`.
```

Bloques completos de código

Para bloques de código de varias líneas pueden utilizarse tres comillas invertidas para abrir y cerrar el bloque, por ejemplo :

````
```
# 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
```
````

PROXMOX VE

GUÍA DE ADMINISTRACIÓN

Versión 9.2.4

PARTE H1 · ENTREGA FINAL

Apéndice H · GNU Free Documentation License

Versión 1.3 · 3 de noviembre de 2008

Traducción al castellano seguida de la versión inglesa original incluida en el manual. La propia sección 8 de la licencia establece que, cuando se incluye una traducción de la licencia, debe acompañarse de la versión original inglesa, que prevalece en caso de discrepancia.

Apéndice H · Licencia de Documentación Libre GNU

Versión 1.3, 3 de noviembre de 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://fsf.org/>

Se permite a todo el mundo copiar y distribuir copias literales de este documento de licencia, pero no está permitido modificarlo.

0. PREÁMBULO

El propósito de esta Licencia es hacer que un manual, libro de texto u otro documento funcional y útil sea "libre" en el sentido de libertad: garantizar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, tanto comercial como no comercialmente. De forma secundaria, esta Licencia conserva para el autor y el editor una forma de recibir reconocimiento por su trabajo, sin que se les considere responsables de las modificaciones realizadas por otros.

Esta Licencia es una forma de "copyleft", lo que significa que las obras derivadas del documento deben ser asimismo libres en el mismo sentido. Complementa la GNU General Public License, que es una licencia copyleft diseñada para software libre.

Hemos diseñado esta Licencia para utilizarla en manuales de software libre, porque el software libre necesita documentación libre: un programa libre debería ir acompañado de manuales que proporcionen las mismas libertades que el software. Sin embargo, esta Licencia no se limita a manuales de software; puede utilizarse para cualquier obra textual, con independencia de su temática o de si se publica como libro impreso. Recomendamos esta Licencia principalmente para obras cuyo propósito sea la enseñanza o la referencia.

1. APLICABILIDAD Y DEFINICIONES

Esta Licencia se aplica a cualquier manual u otra obra, en cualquier medio, que contenga un aviso colocado por el titular de los derechos de autor indicando que puede distribuirse bajo los términos de esta Licencia. Dicho aviso concede una licencia mundial, libre de regalías y de duración ilimitada, para utilizar esa obra conforme a las condiciones aquí establecidas. El "Documento", en lo sucesivo, se refiere a cualquier manual u obra de este tipo. Cualquier miembro del público es un licenciatario y se le denomina "usted". Usted acepta la licencia si copia, modifica o distribuye la obra de una manera que requiera autorización conforme a la legislación sobre derechos de autor.

Una "Versión Modificada" del Documento significa cualquier obra que contenga el Documento o una parte de él, ya sea copiado literalmente o con modificaciones y/o traducido a otro idioma.

Una "Sección Secundaria" es un apéndice con nombre o una sección preliminar del Documento que trata exclusivamente de la relación de los editores o autores del Documento con el tema general del Documento (o con asuntos relacionados) y que no contiene nada que pueda encajar directamente en ese tema general. Así, si el Documento es en parte un libro de texto de matemáticas, una Sección Secundaria no puede explicar matemáticas. La relación puede referirse a una conexión histórica con el tema o con asuntos relacionados, o a una posición jurídica, comercial, filosófica, ética o política respecto a ellos.

Las "Secciones Invariantes" son determinadas Secciones Secundarias cuyos títulos se designan como títulos de Secciones Invariantes en el aviso que indica que el Documento se publica bajo esta Licencia. Si una sección no se ajusta a la definición anterior de Sección Secundaria, no está permitido designarla como Invariante. El Documento puede contener cero Secciones Invariantes. Si el Documento no identifica ninguna Sección Invariante, entonces no existe ninguna.

Los "Textos de Cubierta" son determinados pasajes breves de texto que aparecen listados como Textos de Portada o Textos de Contraportada en el aviso que indica que el Documento se publica bajo esta Licencia. Un Texto de Portada puede tener como máximo 5 palabras y un Texto de Contraportada como máximo 25 palabras.

Una copia "Transparente" del Documento significa una copia legible por máquina, representada en un formato cuya especificación esté disponible para el público general, que sea adecuada para revisar el documento de forma sencilla con editores de texto genéricos o, para imágenes compuestas por píxeles, programas de dibujo genéricos, o, para dibujos, algún editor de dibujo ampliamente disponible, y que sea adecuada como entrada para formateadores de texto o para traducción automática a diversos formatos aptos como entrada de formateadores de texto. Una copia realizada en un formato de archivo por lo demás Transparente cuyo marcado, o ausencia de marcado, se haya dispuesto para impedir o desalentar modificaciones posteriores por los lectores no es Transparente. Un formato de imagen no es Transparente si se utiliza para una cantidad sustancial de texto. Una copia que no sea "Transparente" se denomina "Opaca".

Entre los formatos adecuados para copias Transparentes se incluyen ASCII simple sin marcado, formato de entrada Texinfo, formato de entrada LaTeX, SGML o XML que utilicen una DTD disponible públicamente, y HTML simple conforme a estándares, PostScript o PDF diseñados para modificación humana. Entre los formatos de imagen transparentes se incluyen PNG, XCF y JPG. Los formatos Opacos incluyen formatos propietarios que solo puedan leerse y editarse con procesadores de texto propietarios, SGML o XML para los que la DTD y/o las herramientas de procesamiento no estén disponibles de forma general, y el HTML, PostScript o PDF generado automáticamente por algunos procesadores de texto únicamente con fines de salida.

La "Página de Título" significa, para un libro impreso, la propia página de título más las páginas siguientes que sean necesarias para contener de forma legible el material que esta Licencia exige que aparezca en la página de título. Para obras en formatos que no tengan una página de título como tal, "Página de Título" significa el texto situado cerca de la aparición más destacada del título de la obra, antes del comienzo del cuerpo del texto.

El "editor" significa cualquier persona o entidad que distribuya copias del Documento al público.

Una sección "Titulada XYZ" significa una subunidad con nombre del Documento cuyo título sea exactamente XYZ o contenga XYZ entre paréntesis tras un texto que traduzca XYZ a otro idioma. En este caso XYZ representa un nombre concreto de sección mencionado más adelante, como "Acknowledgements", "Dedications", "Endorsements" o "History". "Conservar el Título" de una sección de este tipo al modificar el Documento significa que continúa siendo una sección "Titulada XYZ" conforme a esta definición.

El Documento puede incluir Exenciones de Garantía junto al aviso que declara que esta Licencia se aplica al Documento. Estas Exenciones de Garantía se consideran incorporadas por referencia a esta Licencia, pero únicamente en lo relativo a la exclusión de garantías; cualquier otra implicación que dichas Exenciones de Garantía pudieran tener es nula y no afecta al significado de esta Licencia.

2. COPIA LITERAL

Puede copiar y distribuir el Documento en cualquier medio, comercial o no comercialmente, siempre que esta Licencia, los avisos de copyright y el aviso de licencia que indique que esta Licencia se aplica al Documento se reproduzcan en todas las copias, y que no añada ninguna condición adicional a las de esta Licencia. No puede utilizar medidas técnicas para obstruir o controlar la lectura o la copia posterior de las copias que realice o distribuya. No obstante, puede aceptar una compensación a cambio de las copias. Si distribuye una cantidad suficientemente grande de copias, también deberá cumplir las condiciones de la sección 3.

También puede prestar copias, bajo las mismas condiciones indicadas anteriormente, y puede exhibir copias públicamente.

3. COPIA EN CANTIDAD

Si publica copias impresas (o copias en medios que normalmente tengan cubiertas impresas) del Documento en una cantidad superior a 100, y el aviso de licencia del Documento exige Textos de Cubierta, deberá incluir las copias en cubiertas que lleven, de forma clara y legible, todos esos Textos de Cubierta: los Textos de Portada en la portada y los Textos de Contraportada en la contraportada. Ambas cubiertas deberán identificarle también, de forma clara y legible, como editor de esas copias. La portada deberá presentar el título completo, con todas las palabras del título igualmente destacadas y visibles. Puede añadir otro material a las cubiertas. Las copias con cambios limitados a las cubiertas, siempre que conserven el título del Documento y satisfagan estas condiciones, pueden considerarse copias literales en los demás aspectos.

Si los textos exigidos para cualquiera de las cubiertas son demasiado extensos para caber de forma legible, deberá colocar los primeros de la lista, tantos como quepan razonablemente, en la cubierta propiamente dicha y continuar el resto en páginas adyacentes.

Si publica o distribuye más de 100 copias Opacas del Documento, deberá incluir una copia Transparente legible por máquina junto con cada copia Opaca, o bien indicar en o junto con cada copia Opaca una ubicación en una red informática desde la que el público general usuario de la red pueda descargar, mediante protocolos de red estándar públicos, una copia Transparente completa del Documento, libre de material añadido. Si utiliza esta última opción, deberá adoptar medidas razonablemente prudentes, al comenzar la distribución en cantidad de copias Opacas, para garantizar que esa copia Transparente permanezca accesible en la ubicación indicada hasta al menos un año después de la última vez que distribuya al público una copia Opaca de esa edición, directamente o por medio de sus agentes o minoristas.

Se solicita, aunque no es obligatorio, que contacte con los autores del Documento con suficiente antelación antes de redistribuir una gran cantidad de copias, para darles la oportunidad de proporcionarle una versión actualizada del Documento.

4. MODIFICACIONES

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que publique la Versión Modificada precisamente bajo esta Licencia, ocupando la Versión Modificada el papel del Documento y autorizando así la distribución y modificación de la Versión Modificada a quien posea una copia de ella. Además, deberá realizar lo siguiente en la Versión Modificada:

A. Utilizar en la Página de Título (y en las cubiertas, si las hubiera) un título distinto del del Documento y de los de versiones anteriores, que deberían figurar, si existieran, en la sección History del Documento. Puede utilizar el mismo título que una versión anterior si el editor original de esa versión lo autoriza.

B. Indicar en la Página de Título, como autores, una o más personas o entidades responsables de la autoría de las modificaciones de la Versión Modificada, junto con al menos cinco de los autores principales del Documento (todos sus autores principales si tiene menos de cinco), salvo que le eximan de este requisito.

C. Indicar en la Página de Título el nombre del editor de la Versión Modificada, como editor.

D. Conservar todos los avisos de copyright del Documento.

E. Añadir un aviso de copyright apropiado para sus modificaciones junto a los demás avisos de copyright.

F. Incluir, inmediatamente después de los avisos de copyright, un aviso de licencia que otorgue al público permiso para utilizar la Versión Modificada bajo los términos de esta Licencia, en la forma mostrada en el Addendum que figura más adelante.

G. Conservar en dicho aviso de licencia las listas completas de Secciones Invariantes y Textos de Cubierta exigidos que figuren en el aviso de licencia del Documento.

H. Incluir una copia sin alterar de esta Licencia.

I. Conservar la sección Titulada "History", Conservar su Título y añadir un elemento que indique al menos el título, el año, los nuevos autores y el editor de la Versión Modificada tal como aparecen en la Página de Título. Si no existe una sección Titulada "History" en el Documento, crear una que indique el título, el año, los autores y el editor del Documento tal como aparecen en su Página de Título, y después añadir un elemento que describa la Versión Modificada conforme a la frase anterior.

J. Conservar la ubicación de red, si la hubiera, indicada en el Documento para el acceso público a una copia Transparente del Documento, así como las ubicaciones de red indicadas en el Documento para versiones anteriores en las que se base. Estas pueden situarse en la sección "History". Puede omitir una ubicación de red correspondiente a una obra publicada al menos cuatro años antes que el propio Documento, o si el editor original de la versión a la que se refiere lo autoriza.

K. Para cualquier sección Titulada "Acknowledgements" o "Dedications", Conservar el Título de la sección y conservar en ella toda la esencia y el tono de cada uno de los agradecimientos y/o dedicatorias de colaboradores que contenga.

L. Conservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Los números de sección o equivalentes no se consideran parte de los títulos de las secciones.

M. Eliminar cualquier sección Titulada "Endorsements". Esa sección no puede incluirse en la Versión Modificada.

N. No cambiar el título de ninguna sección existente para que pase a estar Titulada "Endorsements" ni para que entre en conflicto de título con ninguna Sección Invariante.

O. Conservar cualquier Exención de Garantía.

Si la Versión Modificada incluye nuevas secciones preliminares o apéndices que cumplan los requisitos para ser Secciones Secundarias y no contengan material copiado del Documento, puede, a su elección, designar algunas o todas estas secciones como invariantes. Para hacerlo, añada sus títulos a la lista de Secciones Invariantes del aviso de licencia de la Versión Modificada. Estos títulos deberán ser distintos de cualquier otro título de sección.

Puede añadir una sección Titulada "Endorsements", siempre que no contenga nada más que adhesiones o avales de diversas partes a su Versión Modificada, por ejemplo declaraciones de revisión por pares o de que el texto ha sido aprobado por una organización como definición autorizada de una norma.

Puede añadir un pasaje de hasta cinco palabras como Texto de Portada y un pasaje de hasta 25 palabras como Texto de Contraportada al final de la lista de Textos de Cubierta de la Versión Modificada. Cada entidad solo puede añadir, por sí misma o mediante acuerdos realizados por ella, un único pasaje de Texto de Portada y uno de Texto de Contraportada. Si el Documento ya incluye un texto de cubierta para la misma cubierta, añadido anteriormente por usted o mediante un acuerdo realizado por la misma entidad en cuyo nombre actúa, no podrá añadir otro; sin embargo, podrá sustituir el anterior con autorización explícita del editor anterior que lo añadió.

Los autores y editores del Documento no conceden mediante esta Licencia permiso para utilizar sus nombres con fines publicitarios ni para afirmar o implicar su respaldo a ninguna Versión Modificada.

5. COMBINACIÓN DE DOCUMENTOS

Puede combinar el Documento con otros documentos publicados bajo esta Licencia, conforme a los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificarlas, y las enumere todas como Secciones Invariantes de la obra combinada en su aviso de licencia, y siempre que conserve todas sus Exenciones de Garantía.

La obra combinada solo necesita contener una copia de esta Licencia, y varias Secciones Invariantes idénticas pueden sustituirse por una sola copia. Si existen varias Secciones Invariantes con el mismo nombre pero contenidos diferentes, haga único el título de cada una añadiendo al final, entre paréntesis, el nombre del autor o editor original de esa sección, si se conoce, o en su defecto un número único. Realice el mismo ajuste en los títulos de sección de la lista de Secciones Invariantes del aviso de licencia de la obra combinada.

En la combinación, deberá combinar cualesquiera secciones Tituladas "History" de los distintos documentos originales, formando una sola sección Titulada "History"; del mismo modo, combinará cualesquiera secciones Tituladas "Acknowledgements" y cualesquiera secciones Tituladas "Dedications". Deberá eliminar todas las secciones Tituladas "Endorsements".

6. COLECCIONES DE DOCUMENTOS

Puede crear una colección formada por el Documento y otros documentos publicados bajo esta Licencia, y sustituir las copias individuales de esta Licencia en los distintos documentos por una sola copia incluida en la colección, siempre que cumpla las reglas de esta Licencia relativas a la copia literal de cada documento en todos los demás aspectos.

Puede extraer un documento individual de dicha colección y distribuirlo de forma independiente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído y cumpla esta Licencia en todos los demás aspectos relativos a la copia literal de ese documento.

7. AGREGACIÓN CON OBRAS INDEPENDIENTES

Una recopilación del Documento o de sus derivados junto con otros documentos u obras separados e independientes, dentro de un volumen de un medio de almacenamiento o distribución o sobre él, se denomina "agregado" si los derechos de autor resultantes de la recopilación no se utilizan para limitar los derechos legales de los usuarios de la recopilación más allá de lo que permiten las obras individuales. Cuando el Documento se incluye en un agregado, esta Licencia no se aplica a las demás obras del agregado que no sean por sí mismas obras derivadas del Documento.

Si el requisito de Textos de Cubierta de la sección 3 es aplicable a estas copias del Documento y el Documento representa menos de la mitad del agregado completo, los Textos de Cubierta del Documento pueden colocarse en cubiertas que enmarquen el Documento dentro del agregado, o en el equivalente electrónico de cubiertas si el Documento está en formato electrónico. En caso contrario, deberán aparecer en cubiertas impresas que enmarquen todo el agregado.

8. TRADUCCIÓN

La traducción se considera una forma de modificación, por lo que puede distribuir traducciones del Documento bajo los términos de la sección 4. La sustitución de Secciones Invariantes por traducciones requiere un permiso especial de sus titulares de derechos de autor, aunque puede incluir traducciones de algunas o todas las Secciones Invariantes además de las versiones originales de dichas Secciones Invariantes. Puede incluir una traducción de esta Licencia, así como de todos los avisos de licencia del Documento y de cualquier Exención de Garantía, siempre que incluya también la versión original inglesa de esta Licencia y las versiones originales de esos avisos y exenciones. En caso de discrepancia entre la traducción y la versión original de esta Licencia, o de un aviso o exención, prevalecerá la versión original.

Si una sección del Documento está Titulada "Acknowledgements", "Dedications" o "History", el requisito de la sección 4 de Conservar su Título, según la sección 1, normalmente exigirá cambiar el título real.

9. TERMINACIÓN

No puede copiar, modificar, sublicenciar ni distribuir el Documento salvo en la forma expresamente prevista por esta Licencia. Cualquier intento de copiarlo, modificarlo, sublicenciarlo o distribuirlo de otro modo será nulo y terminará automáticamente sus derechos bajo esta Licencia.

No obstante, si cesa toda infracción de esta Licencia, su licencia otorgada por un determinado titular de derechos de autor se restablece: (a) provisionalmente, salvo y hasta que dicho titular termine explícita y definitivamente su licencia; y (b) permanentemente, si el titular no le notifica la infracción por algún medio razonable antes de que transcurran 60 días desde el cese.

Además, su licencia otorgada por un determinado titular se restablece permanentemente si dicho titular le notifica la infracción por algún medio razonable, es la primera vez que recibe de ese titular un aviso de infracción de esta Licencia respecto de cualquier obra y subsana la infracción antes de que transcurran 30 días desde la recepción del aviso.

La terminación de sus derechos conforme a esta sección no termina las licencias de quienes hayan recibido de usted copias o derechos bajo esta Licencia. Si sus derechos han terminado y no han sido restablecidos permanentemente, la recepción de una copia de parte o de todo el mismo material no le concede ningún derecho para utilizarlo.

10. FUTURAS REVISIONES DE ESTA LICENCIA

La Free Software Foundation puede publicar periódicamente versiones nuevas y revisadas de la GNU Free Documentation License. Esas nuevas versiones serán similares en espíritu a la presente, aunque pueden diferir en los detalles para abordar nuevos problemas o inquietudes. Consulte http://www.gnu.org/copyleft/.

A cada versión de la Licencia se le asigna un número de versión distintivo. Si el Documento especifica que se aplica una versión numerada determinada de esta Licencia "o cualquier versión posterior", tiene la opción de seguir los términos y condiciones de esa versión especificada o de cualquier versión posterior publicada, no en borrador, por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede elegir cualquier versión publicada, no en borrador, por la Free Software Foundation. Si el Documento especifica que un representante puede decidir qué versiones futuras de esta Licencia pueden utilizarse, la declaración pública de aceptación de una versión realizada por ese representante le autoriza permanentemente a elegir esa versión para el Documento.

11. RELICENCIAMIENTO

Un "Sitio de Colaboración Masiva de Múltiples Autores" (o "Sitio MMC") significa cualquier servidor de la World Wide Web que publique obras protegibles por derechos de autor y proporcione además mecanismos destacados para que cualquier persona pueda editar esas obras. Una wiki pública que cualquiera pueda editar es un ejemplo de este tipo de servidor. Una "Colaboración Masiva de Múltiples Autores" (o "MMC") contenida en el sitio significa cualquier conjunto de obras protegibles por derechos d e autor publicadas de este modo en el Sitio MMC.

"CC-BY-SA" significa la licencia Creative Commons Attribution-Share Alike 3.0 publicada por Creative Commons Corporation, una corporación sin ánimo de lucro con sede principal en San Francisco, California, así como las futuras versiones copyleft de es a licencia publicadas por la misma organización.

"Incorporar" significa publicar o volver a publicar un Documento, total o parcialmente, como parte de otro Documento.

Una MMC es "elegible para relicenciamiento" si está licenciada bajo esta Licencia y si todas las obras que fueron publicadas por primera vez bajo esta Licencia en algún lugar distinto de esa MMC y posteriormente incorporadas total o parcialmente a la MMC: (1) no tenían Textos de Cubierta ni Secciones Invariantes, y (2) fueron incorporadas antes del 1 de noviembre de 2008.

El operador de un Sitio MMC puede volver a publicar una MMC contenida en el sitio bajo CC-BY-SA en el mismo sitio en cualquier momento anterior al 1 de agosto de 2009, siempre que la MMC sea elegible para relicenciamiento.

Original English version · GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<http://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 worldwide, 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 http://www.gnu.org/copyleft/.

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.