

# Análisis Comparativo de Plataformas de Nube: Openstack vs. Proxmox VE

La computación en la nube se ha establecido como un paradigma fundamental en la infraestructura tecnológica moderna, ofreciendo diversos modelos de implementación adaptados a las necesidades específicas de las organizaciones. Comprender las diferencias entre las nubes privadas, públicas, comunitarias e híbridas es esencial para seleccionar la estrategia de nube más adecuada. Este informe profundiza en estos modelos, examina la arquitectura y los módulos de Openstack, y presenta una comparación detallada con Proxmox Virtual Environment (VE), una alternativa de virtualización de código abierto.

## Tipos de Nubes: Definiciones y Ejemplos

La elección de un modelo de nube depende de varios factores, incluyendo los requisitos de seguridad, las necesidades de cumplimiento normativo, el presupuesto y la escalabilidad deseada.

- **Nube Privada:** Una nube privada es un entorno de computación dedicado a una única organización, gestionado internamente o por un tercero, y puede estar alojado on-premise o en un centro de datos externo <sup>1</sup>. Inicialmente, el concepto de nube privada surgió para distinguir las infraestructuras internas que adoptaban los principios de la computación en la nube pública, de los servicios ofrecidos por proveedores externos <sup>2</sup>. Antes de la popularización de servicios como Amazon Web Services (AWS), las empresas tradicionalmente adquirían y mantenían su propio hardware. La introducción de la nube pública llevó a las organizaciones a intentar replicar este modelo de agilidad y escalabilidad dentro de sus propias infraestructuras, dando origen al término "nube privada" <sup>2</sup>. Por lo tanto, una nube privada no se limita a un centro de datos dedicado, sino que también implica la implementación de tecnologías como la virtualización, el autoservicio y la medición del uso, características distintivas de la computación en la nube <sup>3</sup>.

Las características principales de una nube privada incluyen el aislamiento de recursos para un solo inquilino, lo que proporciona un mayor control sobre la infraestructura y los datos. Esta dedicación permite a las organizaciones personalizar el entorno para cumplir con sus necesidades específicas de seguridad, cumplimiento y rendimiento <sup>2</sup>. El control y la capacidad de personalización son, de hecho, los principales impulsores para la adopción de nubes privadas, permitiendo a las empresas adaptar la infraestructura a sus requisitos únicos <sup>3</sup>. Por ejemplo, las industrias altamente reguladas, como el

sector salud o el financiero, pueden tener restricciones legales sobre dónde pueden almacenar datos sensibles <sup>4</sup>. Una nube privada les ofrece la flexibilidad y escalabilidad de la computación en la nube, manteniendo al mismo tiempo el control total sobre la ubicación y el acceso a sus datos.

Existen varios tipos de soluciones de nube privada. Una **nube privada on-premise** se implementa en la propia infraestructura de una organización, en su centro de datos interno. En este modelo, la empresa es responsable de la compra, propiedad, operación y mantenimiento de todos los recursos <sup>2</sup>. Si bien ofrece el máximo control sobre la infraestructura y los datos, este modelo también implica una inversión inicial considerable y la asunción de responsabilidades continuas de mantenimiento, actualización y seguridad <sup>2</sup>. Los costos asociados con la adquisición de hardware, software de virtualización y gestión, así como los gastos operativos de energía, refrigeración y personal de TI, pueden ser significativos.

Otro tipo es la **nube privada gestionada**, donde un proveedor externo se encarga de la gestión completa de la infraestructura dedicada a una sola organización. Esta infraestructura puede estar ubicada en el centro de datos del proveedor y es mantenida, actualizada y gestionada remotamente por este <sup>2</sup>. Si bien este modelo puede resultar más costoso que una solución on-premise, ofrece una mayor comodidad al liberar a la organización de las cargas operativas directas. Es una opción atractiva para aquellas empresas que carecen de la experiencia interna o el deseo de gestionar una infraestructura compleja.

La **nube privada virtual (VPC)** representa un entorno privado y aislado que se implementa dentro de la infraestructura compartida de una nube pública <sup>2</sup>. Una VPC permite a las organizaciones beneficiarse de la escalabilidad y los bajos costos iniciales de la nube pública, al tiempo que mantienen un alto grado de control y seguridad dentro de su propio espacio virtual aislado. Los usuarios tienen acceso a un grupo configurable de recursos que son compartidos solo dentro de su organización, mientras que el proveedor de la nube pública se encarga del mantenimiento del hardware subyacente. La VPC se ha convertido en una opción popular para las organizaciones que desean migrar cargas de trabajo a la nube con preocupaciones sobre seguridad y cumplimiento, buscando un equilibrio entre costo, flexibilidad y control.

Los casos de uso típicos de las nubes privadas incluyen organizaciones en industrias reguladas con estrictos requisitos de cumplimiento, como el sector salud, financiero o gubernamental <sup>4</sup>. También son adecuadas para cargas de trabajo que manejan datos sensibles, empresas que requieren un alto grado de personalización y control sobre su infraestructura, y aquellas que desean modernizar su TI manteniendo ciertos sistemas on-premise <sup>4</sup>. La capacidad de

cumplir con las regulaciones y mantener la soberanía de los datos es un factor clave para la adopción de nubes privadas en estos sectores <sup>3</sup>.

- **Nube Pública:** Una nube pública se define como servicios de TI, incluyendo almacenamiento, servidores, bases de datos, redes y software, que son ofrecidos por proveedores externos a través de la internet y están disponibles para el uso del público en general <sup>8</sup>. El modelo multi-inquilino es fundamental para la eficiencia y la rentabilidad de la nube pública, ya que permite a los proveedores compartir recursos de infraestructura entre múltiples usuarios <sup>4</sup>. Al agrupar recursos en centros de datos distribuidos globalmente, los proveedores pueden lograr economías de escala significativas, lo que se traduce en costos más bajos para los clientes, quienes solo pagan por los recursos que realmente consumen <sup>9</sup>. Las características distintivas de la nube pública incluyen una escalabilidad elástica y prácticamente ilimitada, que permite a las organizaciones aumentar o disminuir el uso de recursos según la demanda sin preocuparse por la capacidad <sup>9</sup>. El modelo de pago por uso asegura que los clientes solo incurran en costos por los servicios que realmente utilizan <sup>9</sup>. Además, los usuarios no tienen que preocuparse por el mantenimiento del hardware subyacente, ya que esta responsabilidad recae en el proveedor de la nube <sup>11</sup>. La nube pública también ofrece alta disponibilidad y fiabilidad, con acceso a los recursos desde cualquier lugar con conexión a Internet <sup>11</sup>.

Entre los proveedores más destacados de nube pública se encuentran Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP) <sup>4</sup>. Estos proveedores dominan el mercado, ofreciendo una amplia gama de servicios y una infraestructura global masiva <sup>4</sup>. Cada uno tiene sus propias fortalezas y especializaciones, pero todos proporcionan una cartera extensa de servicios que abarcan desde la infraestructura básica (IaaS) hasta plataformas de desarrollo (PaaS) y aplicaciones listas para usar (SaaS) <sup>10</sup>.

Los modelos de servicio comunes en la nube pública son Infraestructura como Servicio (IaaS), que proporciona acceso a recursos computacionales, de almacenamiento y de red virtualizados; Plataforma como Servicio (PaaS), que ofrece un entorno para desarrollar, ejecutar y gestionar aplicaciones sin la complejidad de gestionar la infraestructura; y Software como Servicio (SaaS), que proporciona aplicaciones de software completas accesibles a través de Internet, generalmente mediante un modelo de suscripción <sup>10</sup>. La elección del modelo de servicio depende de las necesidades específicas del usuario y del nivel de control que desee mantener sobre la infraestructura.

- **Nube Comunitaria:** Una nube comunitaria es una infraestructura compartida por un grupo específico de organizaciones que comparten objetivos comunes, requisitos de seguridad o necesidades regulatorias <sup>7</sup>. Este modelo surgió como

una forma de equilibrar la seguridad y la privacidad de una nube privada con los beneficios de costos de una infraestructura compartida, especialmente para organizaciones en industrias reguladas <sup>7</sup>. A diferencia de la nube pública, donde los recursos son compartidos por numerosos usuarios no relacionados, el acceso a una nube comunitaria está restringido a un grupo definido de organizaciones que tienen requisitos de cumplimiento o normativos similares <sup>7</sup>. Esto permite la implementación de protocolos de seguridad personalizados y asegura el cumplimiento de regulaciones específicas de la industria.

Las nubes comunitarias se caracterizan por una infraestructura compartida diseñada para el uso colaborativo, con aislamiento de datos y controles de acceso para cada miembro. Aunque es un entorno multi-inquilino, las configuraciones pueden personalizarse para satisfacer las necesidades del grupo. Los modelos de seguridad y gobernanza son a menudo colaborativos, con los miembros trabajando juntos para definir y aplicar políticas. Además, se benefician de un pooling de recursos centralizado y del reparto de costos entre las organizaciones participantes <sup>7</sup>. La colaboración en prácticas de seguridad y estrategias de respuesta ante amenazas es una característica distintiva, permitiendo a los miembros trabajar conjuntamente para mejorar su postura general de seguridad <sup>7</sup>.

Ejemplos de nubes comunitarias incluyen un grupo de hospitales que comparten una infraestructura para intercambiar datos de investigación y aplicaciones de salud, agencias gubernamentales que colaboran en el uso de recursos informáticos seguros, o instituciones financieras que comparten plataformas para facilitar las transacciones interbancarias <sup>14</sup>. Estos ejemplos ilustran cómo este modelo se adapta a sectores específicos con necesidades de colaboración y cumplimiento únicas <sup>14</sup>.

Los beneficios de utilizar una nube comunitaria incluyen una mayor seguridad y cumplimiento en comparación con la nube pública, un control más estricto sobre la soberanía y privacidad de los datos, un cumplimiento normativo facilitado por las políticas compartidas, una colaboración mejorada entre los miembros y una posible reducción del riesgo de dependencia de un proveedor específico, especialmente en aquellas nubes gestionadas por las propias organizaciones miembro <sup>7</sup>. Sin embargo, también existen desafíos, como posibles problemas en la distribución de costos entre los miembros, una personalización que puede ser más limitada en comparación con las nubes privadas y la necesidad de establecer acuerdos de gobernanza entre las organizaciones participantes para la toma de decisiones y la gestión de la infraestructura <sup>7</sup>.

- **Nube Híbrida:** Una nube híbrida es un entorno de computación mixto que combina la infraestructura de una o más nubes públicas, como AWS, Microsoft

Azure o Google Cloud Platform, con la infraestructura privada de una organización, que puede ser su propio centro de datos on-premise o una nube privada dedicada <sup>19</sup>. Este modelo permite a las organizaciones optimizar sus cargas de trabajo al seleccionar el entorno más adecuado para cada una, aprovechando las ventajas tanto de la nube pública como de la privada <sup>21</sup>. Las características clave de una nube híbrida incluyen la portabilidad de las cargas de trabajo entre los diferentes entornos, lo que permite mover aplicaciones y datos según las necesidades <sup>20</sup>. También requiere la integración de datos de múltiples fuentes, tanto de la nube pública como de la infraestructura privada, para obtener una visión unificada. La seguridad es un aspecto crítico, ya que la nube híbrida necesita proteger los datos y las aplicaciones tanto en los entornos on-premise como en la nube pública, así como durante su movimiento entre ellos <sup>20</sup>. Finalmente, la nube híbrida implica la capacidad de orquestar y coordinar los recursos a través de estos diferentes entornos para asegurar un funcionamiento coherente <sup>20</sup>.

Existen varios escenarios de uso comunes para la nube híbrida. El **cloud bursting** es una configuración donde los recursos de la nube pública se utilizan únicamente cuando la infraestructura on-premise alcanza su capacidad máxima, permitiendo manejar picos de demanda de forma rentable <sup>20</sup>. La **replicación de datos** desde el almacenamiento on-premise a la nube pública se utiliza para backup y recuperación ante desastres, o para aumentar la disponibilidad de los datos para equipos distribuidos <sup>20</sup>. Otra estrategia es el **cloud tiering**, donde los datos se mueven entre diferentes niveles de almacenamiento según su importancia y frecuencia de acceso, almacenando los datos más críticos en la infraestructura privada para un acceso rápido y los datos menos críticos en la nube pública para ahorrar costos <sup>20</sup>.

Los escenarios de uso para la nube híbrida son diversos. Es ideal para cargas de trabajo dinámicas o que cambian con frecuencia, donde la escalabilidad de la nube pública es beneficiosa, mientras que las cargas de trabajo críticas o sensibles pueden permanecer en la nube privada para mayor control y seguridad <sup>21</sup>. También es útil para el procesamiento de big data, donde se pueden utilizar los recursos altamente escalables de la nube pública para el análisis, manteniendo los datos sensibles detrás del firewall en una nube privada <sup>21</sup>. La nube híbrida también facilita la migración gradual a la nube, permitiendo a las organizaciones mover cargas de trabajo a su propio ritmo <sup>21</sup>. Además, puede satisfacer necesidades temporales de capacidad de procesamiento para proyectos a corto plazo <sup>21</sup>. Finalmente, es una solución común para organizaciones en industrias reguladas que necesitan mantener ciertos datos en sus instalaciones para cumplir con las normativas, aprovechando al mismo tiempo la flexibilidad de la

nube pública para otras necesidades <sup>21</sup>.

## Openstack: Arquitectura y Funcionalidad de sus Módulos

Openstack es una plataforma de computación en la nube de código abierto que permite a las organizaciones construir y gestionar infraestructuras de nube privadas y públicas. Su diseño modular y su amplia gama de componentes lo convierten en una solución versátil para diversos casos de uso.

- **Arquitectura General de Openstack:** Openstack se basa en una arquitectura modular, donde cada servicio es responsable de una funcionalidad específica de la nube. Esto incluye la gestión de identidad, un catálogo de imágenes, así como todas las funciones de computación, red y almacenamiento necesarias para aprovisionar y desaprovisionar máquinas virtuales <sup>24</sup>. Esta arquitectura modular permite a los usuarios desplegar solo los componentes que necesitan, facilitando la personalización y la escalabilidad de la plataforma para cumplir con requisitos empresariales específicos <sup>26</sup>.

La arquitectura de Openstack se divide generalmente en dos planos principales: el Control Plane y el Data Plane <sup>25</sup>. El **Control Plane** actúa como el cerebro de la infraestructura de Openstack, gestionando los recursos y la orquestación. Incluye servicios core como Nova (computación), Neutron (redes), Keystone (identidad), Glance (imágenes) y Horizon (dashboard) <sup>25</sup>. Estos servicios trabajan en conjunto para autenticar y autorizar a los usuarios, aprovisionar y gestionar máquinas virtuales, configurar redes virtuales, almacenar y distribuir imágenes de disco, y proporcionar una interfaz de usuario para la gestión. La estabilidad y el rendimiento del control plane son críticos para la disponibilidad y el funcionamiento general de la nube de Openstack. El **Data Plane** comprende la infraestructura física subyacente, incluyendo los nodos de computación, almacenamiento y red, que son los que realmente ejecutan las cargas de trabajo de los usuarios <sup>25</sup>. Los nodos de computación ejecutan los hipervisores que alojan las máquinas virtuales, los nodos de almacenamiento proporcionan el espacio para los datos, y los nodos de red facilitan la conectividad. La capacidad y el rendimiento del data plane determinan la escalabilidad y la capacidad de respuesta de la nube de Openstack.

El flujo de trabajo típico en Openstack comienza cuando un usuario solicita recursos a través de la API, el dashboard (Horizon) o un cliente de línea de comandos. Dependiendo del tipo de solicitud, uno u otro servicio se involucra en su procesamiento. Cada servicio gestiona su propio dominio y proporciona todas las funciones necesarias para manejar la tarea dada. Esto incluye la autenticación y autorización del usuario, la programación de recursos, la recopilación de



métricas, etc<sup>24</sup>.. Para aprovisionar el recurso real (una máquina virtual, un volumen de almacenamiento en bloque, una instantánea o una dirección IP externa), Openstack se basa en otras tecnologías como el hipervisor KVM, OVN para redes y Ceph para almacenamiento <sup>24</sup>. Este flujo de trabajo automatizado permite el aprovisionamiento de recursos bajo demanda sin necesidad de intervención administrativa, de manera similar al comportamiento de las principales nubes públicas <sup>24</sup>.

La arquitectura de Openstack se puede visualizar a través de varios diagramas que muestran la interconexión de sus componentes <sup>27</sup>. Estos diagramas ilustran cómo los diferentes servicios interactúan entre sí para proporcionar una plataforma de Infraestructura como Servicio (IaaS) completa. Un diagrama típico mostraría el panel de Horizon interactuando con Keystone para la autenticación, Keystone proporcionando tokens a otros servicios como Nova, Glance, Cinder y Neutron, y estos servicios interactuando entre sí para aprovisionar los recursos solicitados por el usuario. Comprender el diagrama de arquitectura es fundamental para los administradores y desarrolladores que trabajan con Openstack, ya que les ayuda a diagnosticar problemas, planificar despliegues y desarrollar nuevas funcionalidades.

- **Descripción Detallada de los Módulos de Openstack:** Openstack se compone de numerosos módulos, también conocidos como proyectos, cada uno con una función específica dentro de la plataforma de nube. A continuación, se describen algunos de los módulos core, de orquestación y gestión, de interfaz y otros proyectos relevantes.

- **Módulos Core:**

- **Nova (Compute):** Nova es el servicio de computación de Openstack. Gestiona el ciclo de vida de las instancias de computación (máquinas virtuales), incluyendo su creación, eliminación y programación en los nodos de cómputo. Actúa como el motor principal de computación de Openstack <sup>25</sup>. Nova es el corazón de Openstack, responsable de la gestión de los recursos de cómputo y la orquestación de las máquinas virtuales a través de los hipervisores (generalmente KVM) en los nodos de cómputo <sup>33</sup>. Cuando un usuario solicita una nueva máquina virtual, Nova interactúa con Glance para obtener la imagen del disco, con Neutron para configurar la red, con Keystone para la autenticación y autorización, y con Placement para determinar el nodo de cómputo adecuado con suficientes recursos disponibles. El rendimiento y la escalabilidad de Nova son cruciales para la capacidad de Openstack de manejar grandes cargas de trabajo y proporcionar una experiencia de usuario fluida.
- **Glance (Image Service):** Glance es el servicio de imágenes de

Openstack. Proporciona un catálogo de imágenes de máquinas virtuales que se utilizan como plantillas para el aprovisionamiento de instancias. Permite a los usuarios descubrir, registrar y recuperar estas imágenes <sup>25</sup>. Glance actúa como un repositorio centralizado para las imágenes de disco virtual, soportando múltiples formatos de imagen y sistemas de almacenamiento back-end como Swift o Ceph <sup>35</sup>. Cuando Nova necesita lanzar una nueva máquina virtual, consulta a Glance para obtener la imagen base especificada por el usuario. Glance gestiona el almacenamiento y la entrega de estas imágenes a los nodos de cómputo. Un Glance eficiente y escalable es fundamental para el rápido aprovisionamiento de nuevas instancias en Openstack.

- **Cinder (Block Storage):** Cinder es el servicio de almacenamiento en bloque de Openstack. Ofrece almacenamiento persistente a nivel de bloques para las instancias de computación. Permite a los usuarios crear, adjuntar y gestionar volúmenes de almacenamiento que pueden utilizarse como discos duros virtuales <sup>25</sup>. Cinder proporciona almacenamiento persistente que sobrevive al ciclo de vida de una máquina virtual, permitiendo a los usuarios almacenar datos de forma segura y acceder a ellos incluso después de que una instancia se elimine o se reconstruya <sup>37</sup>. Cinder interactúa con Nova para adjuntar y desadjuntar volúmenes a las instancias. Soporta una amplia gama de plataformas de almacenamiento back-end, incluyendo discos locales, NFS, Ceph y NetApp. La fiabilidad y el rendimiento de Cinder son esenciales para las aplicaciones que requieren almacenamiento persistente y de alto rendimiento en la nube de Openstack.
- **Neutron (Networking Service):** Neutron es el servicio de redes de Openstack. Gestiona todos los aspectos de la red dentro de un despliegue de Openstack, proporcionando conectividad entre las interfaces de otros servicios y las máquinas virtuales. Permite la creación de redes, subredes, routers y firewalls virtuales <sup>25</sup>. Neutron ofrece un modelo de "red como servicio" (Networking-as-a-Service), permitiendo a los usuarios construir topologías de red complejas y definir políticas de red avanzadas en la nube <sup>38</sup>. Neutron se integra con Nova para proporcionar interfaces de red a las máquinas virtuales y gestionar la conectividad entre ellas y con redes externas a través de IPs flotantes y NAT. La flexibilidad y la capacidad de Neutron para soportar redes definidas por software (SDN) son fundamentales para la creación de entornos de nube complejos y escalables con Openstack.
- **Keystone (Identity Service):** Keystone es el servicio de identidad de



Openstack. Proporciona servicios de autenticación y autorización para todos los componentes de Openstack, así como un catálogo de servicios. Gestiona usuarios, proyectos (tenants) y roles <sup>25</sup>. Keystone es el servicio fundamental para la seguridad en Openstack, actuando como un punto centralizado para la gestión de identidades y el control de acceso a los recursos de la nube <sup>39</sup>. Otros servicios de Openstack confían en Keystone para verificar la identidad de los usuarios y asegurar que tienen los permisos necesarios para realizar acciones. Keystone también mantiene un catálogo de todos los servicios disponibles en el despliegue de Openstack y sus puntos finales. Un Keystone robusto y bien configurado es esencial para la seguridad y la gestión de usuarios en un entorno de nube de Openstack.

- **Horizon (Dashboard):** Horizon es el panel de control de Openstack. Ofrece una interfaz de usuario basada en web que permite a los usuarios y administradores interactuar con los servicios de Openstack. Proporciona una forma gráfica de gestionar recursos como instancias, redes y almacenamiento <sup>25</sup>. Horizon simplifica la gestión de Openstack al proporcionar una interfaz intuitiva que abstrae la complejidad de la línea de comandos y las APIs <sup>42</sup>. Los usuarios pueden utilizar Horizon para realizar tareas comunes como lanzar nuevas máquinas virtuales, configurar redes, gestionar volúmenes de almacenamiento y monitorizar el uso de recursos. Horizon facilita la adopción de Openstack por parte de usuarios con diferentes niveles de experiencia técnica.
- **Swift (Object Storage):** Swift es el servicio de almacenamiento de objetos de Openstack. Proporciona un sistema de almacenamiento de objetos altamente disponible, distribuido y eventualmente consistente. Es ideal para almacenar grandes cantidades de datos no estructurados como copias de seguridad, archivos y contenido multimedia <sup>25</sup>. Swift está diseñado para la escalabilidad horizontal y la tolerancia a fallos, replicando los datos a través de múltiples nodos para garantizar la disponibilidad incluso en caso de fallos de hardware <sup>43</sup>. Los usuarios pueden acceder a los objetos almacenados en Swift a través de una API RESTful basada en HTTP. Swift es adecuado para datos estáticos que no requieren un esquema fijo. Swift ofrece una solución robusta y rentable para el almacenamiento de grandes volúmenes de datos en la nube de Openstack.
- **Módulos de Orquestación y Gestión:**
  - **Heat (Orchestration):** Heat es el servicio de orquestación de Openstack. Permite a los usuarios describir la infraestructura de sus aplicaciones en

la nube utilizando un formato de plantilla declarativo. Heat se encarga de realizar las llamadas a las APIs de Openstack necesarias para crear las aplicaciones en ejecución <sup>25</sup>. Heat facilita la gestión de la infraestructura como código, permitiendo a los usuarios definir y desplegar entornos complejos de forma automatizada y consistente utilizando plantillas <sup>45</sup>. Las plantillas de Heat describen los recursos necesarios (instancias, redes, almacenamiento) y sus relaciones. Heat se encarga de la creación, actualización y eliminación de estos recursos. Heat simplifica la implementación y gestión de aplicaciones en la nube, permitiendo la automatización y la reproducibilidad de los despliegues.

- **Ceilometer (Telemetry):** Ceilometer es el servicio de telemetría de Openstack. Recopila y normaliza datos de telemetría de varios componentes de Openstack, que pueden utilizarse para la facturación de clientes, el seguimiento del uso de recursos y la generación de alarmas <sup>25</sup>. Ceilometer proporciona una visión unificada del uso de recursos en toda la infraestructura de Openstack, lo que es esencial para la monitorización operativa y la gestión de costos <sup>46</sup>. Ceilometer recopila datos de las notificaciones enviadas por otros componentes de Openstack y los publica en varios destinos, incluyendo bases de datos y colas de mensajes. La información proporcionada por Ceilometer es fundamental para la gestión eficiente de una nube de Openstack.
- **Trove (Database Service):** Trove es el servicio de base de datos de Openstack. Proporciona una funcionalidad de "base de datos como servicio" (DBaaS) en Openstack. Permite a los usuarios aprovisionar y gestionar instancias de bases de datos relacionales de forma rápida y sencilla <sup>25</sup>. Trove automatiza tareas administrativas complejas asociadas con la gestión de bases de datos, como la implementación, la configuración, el parcheado, las copias de seguridad y la restauración <sup>48</sup>. Trove se centra inicialmente en proporcionar aislamiento de recursos y alto rendimiento para bases de datos relacionales. Trove simplifica el uso de bases de datos en la nube de Openstack, permitiendo a los usuarios centrarse en sus aplicaciones en lugar de en la administración de la infraestructura de la base de datos.
- **Sahara (Data Processing):** Sahara es un proyecto de Openstack que tiene como objetivo proporcionar un medio sencillo para aprovisionar y gestionar clusters de procesamiento de big data como Hadoop y Spark <sup>25</sup>. Sahara permite a los usuarios escalar recursos de procesamiento de big data bajo demanda, integrándose con otros servicios de Openstack como Nova, Neutron y Glance <sup>49</sup>. Sahara facilita la implementación y gestión del

ciclo de vida de los clusters de big data, permitiendo a los usuarios centrarse en el análisis de datos. Aunque el snippet indica que el proyecto ya no se mantiene, históricamente fue importante para habilitar el procesamiento de big data en Openstack.

- **IroniC (Bare Metal Provisioning):** IroniC es el servicio de aprovisionamiento bare metal de Openstack. Permite el aprovisionamiento de servidores físicos (bare metal) a través de Openstack. Esto permite a los usuarios gestionar servidores físicos de la misma manera que gestionan las máquinas virtuales <sup>25</sup>. IroniC abstrae la complejidad de la gestión del hardware bare metal, proporcionando una interfaz unificada para el aprovisionamiento, despliegue y gestión de servidores físicos <sup>51</sup>. IroniC se integra con otros servicios de Openstack como Keystone para la gestión de identidades, Nova para la computación, Neutron para la red y Glance para las imágenes. IroniC extiende las capacidades de Openstack para incluir la gestión de infraestructura física, lo que es útil para cargas de trabajo que requieren acceso directo al hardware.
- **Designate (DNS Service):** Designate es el servicio DNS de Openstack. Proporciona un servicio de DNS como servicio (DNSaaS) multi-inquilino para Openstack. Permite a los usuarios gestionar registros, nombres y zonas DNS a través de una API RESTful <sup>25</sup>. Designate puede configurarse para generar automáticamente registros DNS basados en acciones en Nova y Neutron, simplificando la gestión del DNS para los recursos de la nube <sup>55</sup>. Designate soporta varios backends de servidores DNS como Bind9 y PowerDNS, ofreciendo flexibilidad en la implementación. Designate facilita la gestión del DNS en entornos de nube de Openstack, mejorando la disponibilidad y la resolución de nombres para las aplicaciones y servicios.
- **Manila (Shared File Systems):** Manila es el servicio de sistemas de archivos compartidos de Openstack. Ofrece un conjunto de servicios para gestionar sistemas de archivos compartidos en un entorno de nube multi-inquilino, de forma similar a como Cinder gestiona el almacenamiento basado en bloques <sup>25</sup>. Manila permite a los usuarios crear sistemas de archivos remotos, montarlos en sus instancias y leer y escribir datos desde sus instancias hacia y desde su sistema de archivos <sup>56</sup>. Manila proporciona extensibilidad para múltiples backends, soportando las particularidades y capacidades específicas de diferentes sistemas de archivos compartidos. Manila facilita el uso de almacenamiento compartido en la nube de Openstack, lo que es esencial para ciertas aplicaciones y cargas de trabajo.

- **Barbican (Key Manager):** Barbican es el servicio de gestión de claves de Openstack. Es el servicio de gestión de claves de Openstack. Proporciona almacenamiento, aprovisionamiento y gestión seguros de datos secretos, incluyendo claves simétricas y asimétricas, certificados y datos binarios sin formato <sup>32</sup>. Barbican ayuda a los usuarios a asegurar sus datos de forma transparente y a mantener su accesibilidad sin tener que gestionar personalmente sus claves <sup>57</sup>. Barbican tiene una arquitectura basada en plugins que permite a los usuarios almacenar sus datos en múltiples almacenes secretos, incluyendo opciones basadas en software y hardware (HSM). Barbican mejora la seguridad en Openstack al proporcionar una forma centralizada y segura de gestionar secretos.
- **Zaqar (Message Service):** Zaqar es el servicio de mensajería de Openstack. Es un servicio de mensajería y notificación en la nube multi-inquilino diseñado para desarrolladores web y móviles. Proporciona una API REST que permite a los desarrolladores enviar mensajes entre diferentes componentes de sus aplicaciones <sup>32</sup>. Zaqar soporta varios patrones de comunicación, incluyendo la difusión de eventos, la distribución de tareas y la mensajería punto a punto, utilizando tanto APIs HTTP como Websocket <sup>59</sup>. Zaqar está diseñado para ser altamente disponible y escalable, con un enfoque en la baja latencia y el alto rendimiento. Zaqar facilita la construcción de aplicaciones en la nube distribuidas y escalables en Openstack.
- **Mistral (Workflow Service):** Mistral es el servicio de flujo de trabajo de Openstack. Es un servicio de flujo de trabajo en Openstack. Permite a los usuarios automatizar procesos complejos describiéndolos como un conjunto de tareas interconectadas que deben ejecutarse en un orden particular <sup>32</sup>. Mistral gestiona el estado, el orden de ejecución, el paralelismo, la sincronización y la alta disponibilidad de estos flujos de trabajo <sup>60</sup>. Los usuarios pueden describir un proceso como un conjunto de tareas y transiciones entre ellas, y cargar esta definición en Mistral, que se encarga de su ejecución. Mistral permite la automatización de tareas administrativas, la ejecución de tareas de larga duración y el despliegue de software distribuido a gran escala en Openstack.
- **Tacker (NFV Orchestration):** Tacker es el servicio de orquestación NFV de Openstack. Es un servicio de Openstack diseñado para la Orquestación de Funciones de Red Virtualizadas (NFV). Su función principal es desplegar y operar Funciones de Red Virtuales (VNFs) y Servicios de Red en una Plataforma NFV <sup>32</sup>. Tacker se basa en el Marco Arquitectónico ETSI MANO y proporciona la API de Orquestación NFV de

Openstack, que se alinea con los estándares ETSI NFV-SOL <sup>61</sup>. Tacker incluye un Gestor de VNF de propósito general y se integra con otros componentes de Openstack para gestionar los recursos necesarios para las VNFs. Tacker facilita el despliegue y la gestión de funciones de red virtualizadas en entornos de nube de Openstack, lo que es crucial para proveedores de telecomunicaciones y otras organizaciones que adoptan NFV.

- **Senlin (Clustering Service):** Senlin es el servicio de clustering de Openstack. Es un servicio de clustering para nubes de Openstack. Crea y opera clusters de objetos homogéneos expuestos por otros servicios de Openstack. El objetivo es facilitar la orquestación de colecciones de objetos similares <sup>32</sup>. Senlin interactúa con otros servicios de Openstack para crear y operar clusters de recursos, utilizando plugins de tipo de perfil para gestionar el ciclo de vida de los objetos <sup>62</sup>. Los usuarios pueden asociar diferentes políticas a un cluster a través de APIs de servicio, como políticas de creación, eliminación, balanceo de carga, escalado y salud. Senlin simplifica la gestión de grupos de recursos homogéneos en Openstack, facilitando la escalabilidad y la alta disponibilidad.
- **Congress (Policy Service):** Congress es el servicio de políticas de Openstack. Es un proyecto de Openstack para proporcionar "política como servicio" a través de cualquier colección de servicios en la nube con el fin de ofrecer gobernanza y cumplimiento para infraestructuras dinámicas <sup>32</sup>. Congress permite a los administradores y tenants de la nube utilizar un lenguaje declarativo de alto nivel para describir la lógica empresarial y garantizar el cumplimiento de las políticas <sup>66</sup>. Congress ofrece una arquitectura basada en plugins que se conecta a cualquier colección de servicios en la nube y puede aplicar políticas de forma proactiva, reactiva o interactiva. Congress facilita la gobernanza y el cumplimiento en entornos de nube de Openstack.
- **Masakari (Instances High Availability):** Masakari es el servicio de alta disponibilidad de instancias de Openstack. Es un servicio de Openstack que proporciona alta disponibilidad para las instancias recuperando automáticamente las instancias fallidas. Actualmente, puede recuperar máquinas virtuales basadas en KVM de eventos de falla como la caída del proceso de la VM o la falla del host nova-compute <sup>32</sup>. Masakari proporciona un servicio de API para gestionar y controlar el mecanismo de rescate automatizado para las instancias <sup>71</sup>. Masakari-monitors proporciona herramientas de monitorización para Masakari que monitorizan el nodo de cómputo de Openstack nova y envían

notificaciones a Masakari utilizando las APIs de Masakari. Masakari mejora la resiliencia de las aplicaciones en Openstack al garantizar la recuperación automática de las instancias fallidas.

- **Kolla (Containerised deployment):** Kolla es el proyecto de despliegue contenerizado de Openstack. El proyecto Kolla proporciona contenedores listos para producción y herramientas de despliegue para operar nubes de Openstack. Proporciona contenedores Docker y playbooks de Ansible para desplegar Openstack en bare metal o máquinas virtuales <sup>32</sup>. Kolla permite a los operadores con poca experiencia desplegar Openstack rápidamente y, a medida que aumenta su experiencia, modificar la configuración de Openstack para adaptarla a sus requisitos exactos <sup>72</sup>. Kolla proporciona imágenes para desplegar los siguientes proyectos de Openstack: Aodh, Barbican, Cinder, Glance, Heat, Horizon, Keystone, Neutron, Nova, Swift, Trove, entre otros. Kolla simplifica el despliegue y la gestión de Openstack mediante el uso de contenedores.
- **Kuryr (Containers Networking):** Kuryr es el proyecto de redes para contenedores de Openstack. Kuryr tiene como objetivo ser el "puente de integración" entre las comunidades de Docker y Neutron, proponiendo e impulsando los cambios necesarios en Neutron (o en Docker) para poder satisfacer los casos de uso necesarios específicamente para la creación de redes de contenedores <sup>32</sup>. Kuryr no es una solución de red por sí misma ni intenta convertirse en una. El esfuerzo de Kuryr se centra en ser el mensajero que entrega las redes y los servicios de Neutron a Docker <sup>74</sup>. Kuryr pretende aprovechar la abstracción y todo el trabajo realizado en Neutron y sus plugins y servicios para proporcionar redes de grado de producción para casos de uso de contenedores. Kuryr facilita la integración de contenedores con las capacidades de red de Openstack.
- **Octavia (Load Balancer):** Octavia es el servicio de balanceo de carga de Openstack. El módulo octavia forma parte de Openstack y se utiliza para configurar y gestionar de forma flexible el servicio de balanceador de carga escalable para Openstack <sup>32</sup>. Octavia proporciona balanceo de carga de red para Openstack, permitiendo la distribución del tráfico entre múltiples instancias para mejorar la disponibilidad y el rendimiento. El módulo octavia intenta hacer que Puppet sea capaz de gestionar la totalidad de octavia, incluyendo manifiestos para aprovisionar puntos finales y conexiones de bases de datos específicos de la región. Octavia mejora la escalabilidad y la fiabilidad de las aplicaciones en Openstack.
- **Placement (Placement Service):** Placement es el servicio de placement de Openstack. El módulo placement forma parte de Openstack y se utiliza



para configurar y gestionar de forma flexible el servicio de placement para Openstack. El servicio de placement es responsable de rastrear los inventarios y usos de los proveedores de recursos, junto con diferentes clases de recursos<sup>32</sup>. Placement mantiene un modelo de datos que rastrea los proveedores de recursos, el inventario, el uso y las clases de recursos, ayudando a Nova a tomar decisiones informadas sobre dónde aprovisionar los recursos. El módulo placement intenta hacer que Puppet sea capaz de gestionar la totalidad del placement, incluyendo manifiestos para aprovisionar puntos finales y conexiones de bases de datos específicos de la región. Placement optimiza la utilización de los recursos en Openstack.

- **Módulos de Interfaz y Otros Proyectos Relevantes:**

- **Skyline (Dashboard):** Skyline es el panel moderno de Openstack. Skyline Console es una parte del Panel Moderno de Openstack, que proporciona una interfaz de usuario basada en web para los servicios de Openstack, incluyendo Nova, Swift, Keystone, etc<sup>32</sup>. Skyline Console permite a los usuarios interactuar con su entorno de Openstack a través de una interfaz gráfica, gestionando redes, instancias y volúmenes. Para usar Skyline Console, se requiere la instalación, configuración y consulta de la documentación del usuario y del administrador. Skyline Console ofrece una alternativa moderna al panel de Horizon.
- **Venus (Log management):** Venus es el servicio de gestión de logs de Openstack. Venus es un proyecto de Openstack que tiene como objetivo proporcionar una solución integral para la recopilación, limpieza, indexación, análisis, alarmas, visualización, generación de informes y otras necesidades de logs<sup>32</sup>. Venus ayuda a los operadores a resolver problemas rápidamente, a comprender la salud operativa de la plataforma y a mejorarla. Todas las funciones de Venus se exponen a través de una API REST. Venus centraliza la gestión de logs en Openstack.
- **Vitrage (RCA Service):** Vitrage es el servicio de análisis de causa raíz de Openstack. Vitrage es el servicio RCA (Root Cause Analysis) de Openstack para organizar, analizar y expandir las alarmas y eventos de Openstack, proporcionando información sobre la causa raíz de los problemas y deduciendo su existencia antes de que se detecten directamente<sup>32</sup>. Vitrage utiliza fuentes de datos para importar información sobre el estado del sistema, un gráfico para mantener esta información y sus interrelaciones, un evaluador para coordinar el análisis y notificadores para alertar a sistemas externos. Vitrage soporta fuentes de datos de Openstack como Nova, Cinder, Neutron, Heat y Aodh, así como monitores

externos como Nagios y Zabbix. Vitrage mejora la capacidad de diagnóstico y resolución de problemas en Openstack.

- **Watcher (Infrastructure Optimization):** Watcher es el servicio de optimización de infraestructura de Openstack. Watcher es un proyecto de Openstack que monitoriza la infraestructura y realiza optimizaciones bajo demanda. Permite a los administradores crear objetivos para la optimización de recursos y Watcher realiza las acciones necesarias o proporciona un plan de acción <sup>32</sup>. Watcher aprovecha los servicios existentes de Openstack como Keystone, Nova, Ironi y Ceilometer para encontrar formas de reducir el costo operativo total de la nube <sup>80</sup>. Hay varios objetivos predefinidos que Watcher puede ayudar a lograr, como la consolidación de servidores y el ahorro de energía. Watcher ayuda a optimizar el uso de recursos y a reducir costos en Openstack.
- **Adjudant (Operations Processes automation):** Adjudant es el framework de automatización de procesos operativos de Openstack. Adjudant es un framework de flujo de trabajo básico construido con Django y Django-Rest-Framework para ayudar a automatizar las tareas de administración dentro de un cluster de Openstack <sup>48</sup>. Adjudant tiene como objetivo proporcionar un lugar y acciones estándar para completar la funcionalidad que falta en Keystone, y permitir la fácil adición de lógica de negocio a tareas más complejas y conexiones con sistemas externos. Las tareas de Adjudant se construyen en torno a tres estados: envío inicial, aprobación del administrador y envío del token. Adjudant facilita la automatización de tareas administrativas en Openstack.
- **Blazar (Resource reservation service):** Blazar es el servicio de reserva de recursos de Openstack. Blazar es el Servicio de Reserva de Recursos para Openstack. Permite a los usuarios reservar varios recursos dentro de un entorno de nube de Openstack <sup>32</sup>. Blazar permite la reserva de hosts, instancias e IPs flotantes a través de una CLI o una API REST. Blazar proporciona guías detalladas para usuarios y administradores, así como guías de instalación y configuración. Blazar facilita la gestión de la capacidad y la planificación de recursos en Openstack.
- **Cyborg (Accelerator Life Cycle Management):** Cyborg es el framework de gestión del ciclo de vida de aceleradores de Openstack. Cyborg es un framework de gestión general para aceleradores. Los usuarios pueden usar Cyborg para crear y gestionar aceleradores con herramientas o la API directamente <sup>32</sup>. Cyborg proporciona una interfaz unificada para gestionar una flota diversa de servidores utilizando una única interfaz, permitiendo que el servicio de Computación (Nova) trate los servidores

físicos como si fueran máquinas virtuales <sup>51</sup>. Cyborg se integra con varios servicios de Openstack, como Keystone para la gestión de identidades, Nova para la computación, Neutron para la red, Glance para la gestión de imágenes y Swift para el almacenamiento de objetos. Cyborg extiende las capacidades de Openstack para gestionar hardware especializado como GPUs y FPGAs.

- **Freezer (Backup, Restore, and Disaster Recovery):** Freezer es el servicio de backup, restauración y recuperación ante desastres de Openstack. Freezer es una plataforma distribuida de backup y restauración como servicio diseñada para entornos multi-OS (Linux, Windows, OSX, BSD). Su función principal es proporcionar soluciones de backup eficientes y flexibles para diversas necesidades, incluyendo grandes nubes efímeras <sup>32</sup>. Freezer ofrece capacidades de backup a nivel de sistema de archivos, backup directo del árbol de archivos, backups específicos de aplicaciones (MongoDB, MySQL), backup basado en bloques, backups incrementales y diferenciales, y restauración automática desde una fecha específica. Freezer soporta múltiples algoritmos de compresión, múltiples medios de almacenamiento (Swift, sistemas de archivos locales, SSH) y cifrado fuerte. Freezer proporciona una solución completa para la protección de datos en entornos Openstack.
- **Gnocchi (Time Series Database):** Gnocchi es la base de datos de series temporales de Openstack. Gnocchi es el nombre del proyecto de una TDBaaS (Time Series Database as a Service) iniciado bajo el paraguas del programa Ceilometer. Proporciona un almacenamiento de series temporales y un indexador de recursos que son rápidos y escalables. También proporciona una API REST <sup>81</sup>. Gnocchi está diseñado para almacenar métricas a muy gran escala al tiempo que proporciona acceso a información de métricas y recursos a operadores y usuarios <sup>82</sup>. Gnocchi utiliza el servicio de identidad para la autenticación y Redis para el almacenamiento de medidas entrantes. Para almacenar las medidas agregadas, Gnocchi se basa en Swift o Ceph (Object Storage) y también utiliza MySQL para almacenar el índice de recursos y métricas. Gnocchi es fundamental para la monitorización y la facturación en Openstack.
- **Karbor (Application Data Protection as a Service):** Karbor es el servicio de protección de datos de aplicaciones de Openstack. Karbor es un proyecto de Openstack que proporciona un framework enchufable para proteger y restaurar los datos y metadatos que comprenden una aplicación desplegada en Openstack - Protección de Datos de Aplicaciones como Servicio <sup>32</sup>. La misión de Karbor es proteger los datos y

metadatos que comprenden una aplicación desplegada en Openstack contra la pérdida/daño (por ejemplo, backup, replicación) proporcionando un framework estándar de APIs y servicios que permite a los proveedores ofrecer plugins a través de una interfaz unificada <sup>83</sup>. Karbor proporciona APIs para gestionar recursos protegibles, planes de protección, proveedores y checkpoints. Karbor mejora la resiliencia de las aplicaciones en Openstack al proporcionar capacidades de protección de datos.

- **Magnum (Container Infrastructure Management):** Magnum es el servicio de gestión de infraestructura de contenedores de Openstack. Magnum es un proyecto de Openstack diseñado para proporcionar motores de orquestación de contenedores para desplegar y gestionar contenedores como recursos de primera clase dentro de un entorno de Openstack. Permite a los usuarios aprovechar tecnologías de contenedores populares como Kubernetes junto con su infraestructura de Openstack <sup>32</sup>. Magnum ofrece abstracciones para gestionar clusters de nodos de cómputo donde se pueden programar cargas de trabajo contenerizadas. Se integra con varios servicios core de Openstack para proporcionar una solución integral de gestión de contenedores <sup>87</sup>. Magnum utiliza Keystone para la seguridad multi-inquilino, Neutron para la seguridad de red multi-inquilino de Kubernetes y Cinder para proporcionar servicios de volumen persistente para contenedores. Magnum facilita el despliegue y la gestión de aplicaciones contenerizadas en Openstack.
- **Monasca (Monitoring):** Monasca es el servicio de monitorización de Openstack. Monasca es una solución de monitorización como servicio diseñada para entornos Openstack. Su función principal es proporcionar una plataforma multi-inquilino, altamente escalable, de alto rendimiento y tolerante a fallos para la monitorización avanzada. Monasca permite tanto a los operadores como a los tenants obtener información operativa sobre su infraestructura y aplicaciones <sup>32</sup>. Monasca utiliza APIs REST para el procesamiento y la consulta de alta velocidad de métricas y logs. También integra un motor de alarmas de streaming, un motor de notificaciones y un motor de agregación. Monasca sigue una arquitectura de microservicios, con sus funcionalidades divididas en múltiples repositorios. Monasca proporciona una solución completa para la monitorización en Openstack.
- **Murano (Application Catalog):** Murano es el catálogo de aplicaciones de Openstack. Murano es un proyecto Openstack de código abierto que combina un catálogo de aplicaciones con herramientas versátiles para

simplificar y acelerar el empaquetamiento y el despliegue. Se puede utilizar con casi cualquier aplicación y servicio en Openstack <sup>32</sup>. Murano tiene como objetivo facilitar a los desarrolladores externos la creación de aplicaciones y servicios para que los utilicen los usuarios finales de Openstack. De esta manera, estos desarrolladores de terceros pueden enriquecer el ecosistema de Openstack para hacerlo más atractivo para los usuarios, y los usuarios pueden sacar más provecho de sus clusters de Openstack más fácilmente, fomentando la adopción del propio Openstack <sup>90</sup>. Las aplicaciones en Murano se definen utilizando un lenguaje orientado a objetos llamado MuranoPL. Murano no permite ejecutar ninguna operación de E/S directamente desde MuranoPL; todas las llamadas a las APIs HTTP necesarias deben escribirse en Python y luego consumirse desde MuranoPL. Murano simplifica el despliegue de aplicaciones en Openstack a través de un catálogo de aplicaciones.

- **Rally (Benchmark Service):** Rally es el servicio de benchmarking de Openstack. Rally es una herramienta de pruebas para el ecosistema Openstack. Su función principal es responder a la pregunta: "¿Cómo funciona Openstack a escala?". Para lograr esto, Rally automatiza y unifica varios procesos clave: despliegue multi-nodo de Openstack, verificación de la nube y pruebas y perfiles <sup>32</sup>. Rally está diseñado para verificar el rendimiento y la estabilidad de Openstack bajo una carga significativa, como una instalación de 1000 servidores. Rally se puede utilizar como una herramienta fundamental para los sistemas CI/CD de Openstack, para verificar los despliegues de Openstack, para perfilar el rendimiento de Openstack y para probar la estabilidad de Openstack. Rally ayuda a garantizar la calidad y el rendimiento de los despliegues de Openstack.
- **Searchlight (Indexing and Search):** Searchlight es el servicio de indexado y búsqueda de Openstack. El proyecto Openstack Searchlight proporciona una API de búsqueda que presenta una interfaz para consultar los diversos recursos disponibles en una nube Openstack. Searchlight crea un índice (utilizando la tecnología Elasticsearch de código abierto) y lo mantiene actualizado <sup>32</sup>. La API de búsqueda permite construir consultas detalladas utilizando el Lenguaje Específico de Dominio de Consulta de Elasticsearch. Searchlight crea un documento para representar cada elemento individual (servidor, imagen, red) en una nube Openstack. Estos documentos se almacenan en un índice de Elasticsearch <sup>93</sup>. Para indexar los recursos de Openstack, Searchlight requiere un plugin específico para cada tipo de recurso. Por lo tanto, los únicos recursos que se pueden indexar en una nube Openstack particular

son aquellos para los que (a) existe un plugin y (b) el plugin está habilitado por el operador de la nube. Searchlight mejora las capacidades de búsqueda en Openstack.

- **Solum (Development Lifecycle Automation):** Solum fue un proyecto de automatización del ciclo de vida del desarrollo de Openstack que ya no se mantiene.
- **Storlets (Compute inside Object Storage):** Storlets permite la computación dentro del almacenamiento de objetos de Openstack. Storlets extiende Swift permitiendo a los usuarios ejecutar sus propias computaciones, conocidas como storlets, cerca de los datos de forma segura y aislada. Un storlet es esencialmente código compilado y empaquetado, como un archivo .jar, que los usuarios pueden cargar en Swift al igual que cualquier otro objeto. Una vez cargado, este storlet se puede invocar para procesar objetos de datos dentro de Swift <sup>32</sup>. La función principal de los storlets es acercar la computación a los datos almacenados en Swift. Esto reduce la latencia de la red y mejora el rendimiento para las tareas con gran cantidad de datos. Los usuarios necesitan escribir su lógica de computación en un lenguaje de programación compatible, como Java o Python, y empaquetarla en un formato desplegable, como un archivo .jar para Java. El storlet empaquetado se carga en Swift como un objeto regular utilizando la API o las herramientas de cliente de Swift. Una vez cargado, se puede invocar sobre objetos de datos específicos almacenados en Swift. Storlets permite la computación cerca de los datos en Openstack Swift.
- **Sunbeam (Deployment and operational tooling):** Sunbeam es un conjunto de herramientas de despliegue y operación de Openstack. Sunbeam es un proyecto upstream bajo la gobernanza de la Fundación OpenInfra (OIF) creado para reducir la barrera de entrada a Openstack, simplificar su proceso de adopción y sentar las bases para una nube privada autónoma. Sunbeam utiliza una arquitectura nativa de la nube y una automatización total de abajo hacia arriba para hacer que Openstack sea más accesible para los recién llegados y para ayudar a los usuarios a familiarizarse con la plataforma de inmediato <sup>53</sup>. Sunbeam permite el despliegue y la operación de Openstack a cualquier escala, desde un solo nodo hasta despliegues de borde a pequeña escala y nubes a gran escala que contienen muchos miles de hipervisores, aprovechando un modelo de despliegue híbrido utilizando Juju para gestionar tanto los componentes de Kubernetes como los componentes basados en máquinas mediante el uso de charms <sup>99</sup>. MicroStack (basado en Sunbeam) es una distribución de



Openstack diseñada para entornos de nube a pequeña escala. Si bien está disponible con soporte comercial completo de Canonical, también se puede auto-desplegar sin problemas, eliminando efectivamente la necesidad de una contratación de consultoría de pago. Sunbeam simplifica el despliegue de Openstack para entornos más pequeños.

- **Zun (Containers service):** Zun es el servicio de contenedores de Openstack. Zun es un servicio de contenedores de Openstack que proporciona una API para ejecutar contenedores de aplicaciones sin necesidad de gestionar servidores o clusters. Requiere los siguientes servicios de Openstack para la funcionalidad básica: Keystone, Neutron y Kuryr-libnetwork. Zun también puede integrarse con otros servicios como Cinder, Heat y Glance <sup>32</sup>. Como usuario final, puede usar Zun para crear y gestionar cargas de trabajo contenerizadas utilizando varias herramientas o directamente a través de la API. Todas las funciones para el usuario final son accesibles a través de una API REST. Los operadores pueden instalar y configurar Zun para habilitar los servicios de contenedores dentro de su entorno Openstack. Una instalación funcional de Zun requiere tener instalados previamente Keystone, Neutron y Kuryr-libnetwork. Zun permite la gestión de contenedores en Openstack.
- **Otros Proyectos Relevantes:**
  - **Aodh (Alarming service):** Aodh es el servicio de alarmas de Openstack. El servicio de Alarma (aodh) en Openstack proporciona la funcionalidad para activar acciones basadas en reglas definidas. Estas reglas se evalúan con datos de métricas o eventos recopilados por otros servicios de Openstack como Ceilometer o Gnocchi <sup>32</sup>. Aodh permite a los usuarios configurar alarmas que monitorizan condiciones específicas dentro de su entorno Openstack. Cuando se cumplen estas condiciones, Aodh puede activar acciones predefinidas. Aodh esencialmente permite a los usuarios configurar alarmas que monitorizan condiciones específicas dentro de su entorno Openstack. Cuando estas condiciones se cumplen, Aodh puede activar acciones predefinidas. Esto permite respuestas automatizadas a varios eventos y estados dentro de la infraestructura de la nube. Aodh proporciona capacidades de alarma en Openstack.
  - **Cloudkitty (Rating service):** Cloudkitty es el servicio de tarificación de Openstack. CloudKitty es un proyecto de Rating-as-a-Service para Openstack. Está diseñado para ser una solución genérica para la facturación y la tarificación de una nube. Si bien históricamente funcionó solo dentro de un contexto Openstack, ahora puede

ejecutarse en modo autónomo<sup>32</sup>. CloudKitty realiza la tarificación basada en métricas. Esto implica los siguientes pasos: Recopilación de métricas, Aplicación de reglas de tarificación, Almacenamiento de datos tarificados y Recuperación de información tarificada. Con CloudKitty, puede recopilar métricas de varias fuentes, incluyendo Openstack y Prometheus, definir y aplicar reglas de tarificación personalizadas a estas métricas y recuperar los datos tarificados para fines de facturación. Cloudkitty proporciona servicios de tarificación en Openstack.

- **Openstack Charms:** El sitio web de Charmhub.io/openstack no está accesible.
- **Openstack-Helm:** Openstack-Helm facilita el despliegue de Openstack en Kubernetes. La función de Openstack-Helm es proporcionar una colección de charts de Helm para desplegar Openstack y servicios relacionados en Kubernetes de forma sencilla, resiliente y flexible<sup>53</sup>. Openstack-Helm permite el despliegue de componentes de Openstack en un cluster de Kubernetes utilizando Helm, que es un gestor de paquetes para Kubernetes. Esto simplifica el proceso de despliegue y gestión. Proporciona una colección de charts de Helm para varios servicios de Openstack como Aodh, Barbican, Cinder, Glance, Heat, Horizon, Keystone, Nova, Neutron y muchos otros. Cada chart contiene las configuraciones y los manifiestos necesarios para desplegar un servicio Openstack específico en Kubernetes.
- **Openstackansible:** Openstack-Ansible facilita el despliegue de Openstack utilizando Ansible. Openstack-Ansible proporciona playbooks y roles de Ansible para el despliegue y la configuración de un entorno Openstack. Es una herramienta de despliegue para Openstack<sup>53</sup>. La documentación cubre varias versiones de Openstack-Ansible, incluyendo 2025.1 (Epoxy), 2024.2 (Dalmatian), 2024.1 (Caracal), 2023.2 (Bobcat), 2023.1 (Antelope) y versiones más antiguas. Openstack-Ansible simplifica el proceso de despliegue y gestión de una nube Openstack utilizando las capacidades de automatización de Ansible. Proporciona una forma estructurada de instalar y configurar varios componentes de Openstack.
- **Openstacksdk:** Openstacksdk facilita la interacción programática con Openstack. El openstacksdk es un SDK (Software Development Kit) de Python diseñado para permitir a los usuarios interactuar con las nubes de Openstack. Esta documentación proporciona información para tres

grupos principales: aquellos que buscan instalar el SDK, los usuarios que desean construir aplicaciones utilizando Openstack y los contribuyentes al proyecto openstacksdk<sup>53</sup>. La función principal de openstacksdk es simplificar el proceso de interacción con varios servicios de Openstack a través de una interfaz unificada de Python. Abstrae las complejidades de las llamadas individuales a las APIs de Openstack, permitiendo a los desarrolladores gestionar recursos como máquinas virtuales (Nova), almacenamiento (Cinder, Swift), redes (Neutron), identidad (Keystone) y más, utilizando código Python. La documentación se divide en secciones que atienden a diferentes aspectos del uso de openstacksdk: Instalación, Para Usuarios (Usando el SDK de Openstack, Guías de Usuario, Pruebas, Documentación de la API, Presentaciones) y Para Contribuyentes (Acerca del Proyecto, Mecánica de Contribución, Contacto con los Desarrolladores, Estándares de Codificación, Entorno de Desarrollo, Pruebas, Diseño del Proyecto, Adición de Funciones).

- **Oslo:** El sitio web de Oslo no está accesible.
- **Puppet Openstack:** Puppet Openstack facilita el despliegue de Openstack utilizando Puppet. Los módulos de Puppet Openstack proporcionan una automatización de TI escalable y fiable para los despliegues de nubes Openstack. Estos módulos están diseñados para ayudar a los usuarios a desplegar y gestionar los componentes de Openstack<sup>53</sup>. La guía ofrece dos secciones principales: Guía del Contribuyente, que proporciona información para aquellos que desean contribuir al proyecto Puppet Openstack, y Guía de Instalación, que se centra en cómo instalar y desplegar los módulos de Puppet Openstack. Puppet Openstack funciona como una herramienta de despliegue para Openstack, aprovechando las capacidades de automatización de Puppet para simplificar el proceso de configuración y mantenimiento de un entorno de nube Openstack. Su objetivo es hacer que los despliegues de Openstack sean más escalables y fiables a través de la automatización.
- **Quality Assurance:** El sitio web de la wiki de Openstack sobre Quality Assurance no está accesible.
- **Release Management:** El equipo de Gestión de Lanzamientos asegura la entrega regular y estable de nuevas versiones de Openstack. El equipo de Gestión de Lanzamientos de Openstack es responsable del proceso general de lanzamiento del software Openstack. Sus funciones incluyen la definición del calendario de

lanzamientos, la coordinación del trabajo de los diferentes equipos de proyecto y la garantía de la calidad y la estabilidad de cada lanzamiento <sup>53</sup>. El ciclo de lanzamiento de Openstack opera con una cadencia de aproximadamente seis meses. Después del lanzamiento principal inicial, se ponen a disposición versiones de punto estables posteriores dentro de la misma serie de lanzamiento. Los lanzamientos de Openstack se clasifican en diferentes fases: Desarrollo, Mantenimiento, No mantenido y Fin de vida (EOL). Algunos lanzamientos de Openstack están designados como lanzamientos del Proceso de Lanzamiento de Actualización de Nivel de Salto (SLURP). Esto significa que las actualizaciones son compatibles entre estos lanzamientos SLURP específicos, además de las actualizaciones estándar entre lanzamientos principales adyacentes.

- **Requirements:** El sitio web de las especificaciones de Openstack sobre Requirements no está accesible.

## **Proxmox VE: Arquitectura y Características Fundamentales**

Proxmox Virtual Environment (VE) es una plataforma de gestión de virtualización de servidores de código abierto completa y diseñada para la virtualización empresarial. Integra estrechamente el hipervisor KVM y los Contenedores Linux (LXC), la funcionalidad de redes y almacenamiento definidos por software, todo en una única plataforma <sup>102</sup>.

- **Arquitectura General de Proxmox VE:** Proxmox VE se basa en una distribución de servidor Debian Linux implementada en un kernel Linux especialmente optimizado <sup>103</sup>. Esta base proporciona una plataforma estable y bien conocida, facilitando la administración y la compatibilidad con una amplia gama de software y hardware <sup>106</sup>. Al estar construido sobre Debian, Proxmox VE se beneficia de la gran comunidad de desarrolladores de Debian y de su robusto sistema de gestión de paquetes (APT), lo que simplifica la instalación, las actualizaciones y la gestión de software. La familiaridad con Debian para muchos administradores de sistemas Linux reduce la curva de aprendizaje para Proxmox VE. Proxmox VE integra estrechamente el hipervisor KVM para la virtualización completa y los contenedores Linux (LXC) para la virtualización a nivel de sistema operativo <sup>102</sup>. Esta combinación ofrece flexibilidad para ejecutar diferentes tipos de cargas de trabajo, desde sistemas operativos completos en VMs hasta aplicaciones aisladas y ligeras en contenedores <sup>102</sup>. KVM proporciona un rendimiento cercano al nativo para las VMs y es adecuado para ejecutar sistemas operativos como Windows y Linux con requisitos de recursos dedicados. LXC, al

compartir el kernel del host, es más eficiente en términos de uso de recursos y es ideal para ejecutar aplicaciones Linux aisladas. La capacidad de elegir entre KVM y LXC permite a los usuarios optimizar el uso de recursos y la densidad de las cargas de trabajo según sus necesidades.

Los componentes clave de la arquitectura de Proxmox VE incluyen QEMU (Quick EMUlator) y KVM (Kernel-based Virtual Machine) en un enfoque híbrido <sup>104</sup>. QEMU es un emulador de máquina genérico de código abierto y virtualizador que Proxmox VE utiliza en combinación con KVM para la emulación de dispositivos virtuales <sup>104</sup>. QEMU se ejecuta en el espacio de usuario y se encarga de la emulación de dispositivos como discos virtuales e interfaces de red, mientras que KVM gestiona la virtualización de la CPU y la memoria con aceleración por hardware <sup>104</sup>. Esta combinación permite que Proxmox VE funcione de manera similar a un hipervisor de tipo 1, a pesar de que QEMU técnicamente se ejecuta en el espacio de usuario. La sinergia entre QEMU y KVM resulta en un rendimiento eficiente para las máquinas virtuales. KVM, por su parte, es un hipervisor de tipo 1 que es un módulo del kernel de Linux y proporciona la infraestructura para la virtualización completa de máquinas virtuales <sup>104</sup>. KVM aprovecha las extensiones de virtualización por hardware (Intel VT-x o AMD-V) para ofrecer un rendimiento casi nativo en hardware compatible.

Proxmox VE también incorpora funcionalidades de redes y almacenamiento definidos por software directamente en la plataforma <sup>102</sup>. Para las redes, Proxmox VE utiliza la pila de red de Linux, ofreciendo flexibilidad en la configuración a través de interfaces de puente (bridges), enlaces (bonds) y VLANs. En cuanto al almacenamiento, Proxmox VE soporta una amplia gama de opciones, incluyendo almacenamiento local (LVM, ZFS), almacenamiento de red (NFS, iSCSI, SMB/CIFS) y almacenamiento distribuido (Ceph, GlusterFS) <sup>106</sup>. El modelo de almacenamiento es muy flexible, permitiendo almacenar imágenes de VMs en uno o varios dispositivos locales o en almacenamiento compartido.

La gestión de Proxmox VE se realiza principalmente a través de una interfaz gráfica de usuario (GUI) basada en web, intuitiva y accesible desde cualquier navegador moderno <sup>106</sup>. Esta interfaz centralizada permite realizar todas las tareas de administración, incluyendo la gestión de VMs y contenedores, la configuración de alta disponibilidad para clusters y la gestión de las herramientas de recuperación ante desastres integradas <sup>102</sup>. Además de la GUI, Proxmox VE también ofrece una interfaz de línea de comandos (CLI) para usuarios avanzados y una API RESTful para la integración con herramientas de gestión de terceros <sup>106</sup>.

- **Características Fundamentales de Proxmox VE:** Proxmox VE ofrece una amplia gama de características diseñadas para la virtualización empresarial. Una de las más destacadas es su soporte para virtualización completa con KVM y

virtualización basada en contenedores con LXC, proporcionando máxima flexibilidad para diferentes cargas de trabajo <sup>102</sup>.

La gestión de Proxmox VE es sencilla gracias a su interfaz gráfica de usuario (GUI) integrada, que permite realizar todas las tareas de administración sin necesidad de herramientas separadas <sup>106</sup>. La GUI proporciona una visión general del historial de tareas y los logs del sistema de cada nodo, incluyendo tareas de backup, migración en vivo y actividades de alta disponibilidad <sup>106</sup>. Para usuarios avanzados, Proxmox VE también ofrece una interfaz de línea de comandos (CLI) con autocompletado inteligente y documentación completa en forma de páginas man de UNIX <sup>106</sup>. Además, Proxmox VE utiliza una API RESTful con formato JSON, lo que facilita la integración con herramientas de gestión de terceros <sup>106</sup>.

Proxmox VE permite la creación de clusters de hasta 32 hosts sin necesidad de hosts de gestión dedicados, utilizando tecnologías como el overcommitment de CPU y el ballooning de RAM para optimizar la utilización de recursos <sup>108</sup>. La función de migración en vivo/online permite mover máquinas virtuales en ejecución de un nodo a otro dentro del cluster sin tiempo de inactividad <sup>106</sup>.

Proxmox VE soporta múltiples fuentes de autenticación, incluyendo Linux PAM, un servidor de autenticación integrado de Proxmox VE, LDAP, Microsoft Active Directory y OpenID Connect <sup>106</sup>. Para la alta disponibilidad (HA), Proxmox VE permite crear clusters multi-nodo que garantizan la continuidad de los servidores virtuales en caso de fallo de un nodo. El HA Cluster se basa en tecnologías probadas de Linux HA, proporcionando un servicio estable y fiable <sup>106</sup>. El gestor de HA de Proxmox VE monitoriza todas las VMs y contenedores del cluster y actúa automáticamente si alguno falla <sup>106</sup>.

En cuanto a las redes, Proxmox VE utiliza un modelo de red en puente (bridged) para proporcionar conectividad entre los entornos virtualizados y la red externa <sup>106</sup>. Ofrece opciones de configuración flexibles para nodos locales.

Proxmox VE también destaca por sus opciones de almacenamiento flexibles, soportando LVM Group, iSCSI target, NFS Share, SMB/CIFS, Ceph RBD, acceso directo a iSCSI LUN y GlusterFS <sup>106</sup>. Las imágenes de las VMs pueden almacenarse tanto en dispositivos locales como en almacenamiento compartido como NFS y SAN. Se pueden configurar tantos almacenamientos como se necesiten. El almacenamiento compartido permite la migración en vivo de máquinas en ejecución sin tiempo de inactividad. Proxmox VE también ofrece almacenamiento definido por software con Ceph, que facilita la configuración y gestión a través de la GUI y la CLI, es auto-reparable y escalable <sup>106</sup>.

Finalmente, Proxmox VE incluye un firewall que permite configurar reglas para todos los hosts dentro de un cluster o definir reglas específicas para VMs y contenedores. Soporta IPv4 e IPv6 y ofrece características como macros de



firewall, grupos de seguridad, conjuntos de IPs y alias para facilitar la tarea <sup>106</sup>. El firewall distribuido proporciona aislamiento entre las VMs y mayor ancho de banda en comparación con soluciones centralizadas.

## Comparación entre Openstack y Proxmox VE

Openstack y Proxmox VE son dos plataformas de código abierto que ofrecen capacidades de virtualización y gestión de la nube, pero están diseñadas para diferentes escalas y casos de uso.

- **Funcionalidades:** Ambas plataformas soportan la virtualización basada en KVM, pero Proxmox VE también integra contenedores LXC, ofreciendo mayor flexibilidad en la elección de la tecnología de virtualización <sup>109</sup>. Openstack, con su arquitectura modular, ofrece una gama mucho más amplia de servicios, incluyendo computación, almacenamiento de objetos y bloques, redes avanzadas (SDN), gestión de identidad, orquestación, telemetría y muchos otros, lo que lo convierte en una plataforma de nube integral <sup>109</sup>. Proxmox VE, por otro lado, se centra más en la gestión de la virtualización y ofrece características como alta disponibilidad, migración en vivo y herramientas de backup integradas <sup>109</sup>. Si bien ambos soportan multi-tenancy, Openstack está especialmente diseñado para entornos multi-inquilino a gran escala con controles de acceso y separación de recursos robustos <sup>108</sup>. Proxmox VE ofrece capacidades de multi-tenancy a través de controles de usuarios y roles, pero no a la misma escala que Openstack <sup>108</sup>.
- **Complejidad:** Proxmox VE se considera generalmente más sencillo y menos complejo de implementar y gestionar que Openstack <sup>108</sup>. Ofrece una interfaz web intuitiva y un proceso de despliegue más directo, lo que lo hace ideal para organizaciones con recursos limitados o menos experiencia técnica <sup>108</sup>. Openstack, con su arquitectura modular y numerosos componentes, puede ser complejo de configurar y mantener, requiriendo un mayor nivel de experiencia técnica y un equipo dedicado <sup>108</sup>. La curva de aprendizaje para Openstack es significativamente más pronunciada que para Proxmox VE <sup>113</sup>.
- **Casos de Uso:** Openstack es más adecuado para despliegues de nube a gran escala, infraestructuras complejas con requisitos de multi-tenancy avanzados y organizaciones que buscan una plataforma altamente personalizable y extensible para construir nubes públicas o privadas <sup>108</sup>. Es ideal para empresas que necesitan una amplia gama de servicios de infraestructura como servicio (IaaS) y plataforma como servicio (PaaS) <sup>115</sup>. Proxmox VE, por otro lado, es una excelente opción para entornos de virtualización de pequeña a mediana escala, centros de datos más pequeños, laboratorios domésticos y entornos de prueba y desarrollo donde la multi-tenancy no es un requisito primordial y se prefiere la facilidad de

uso <sup>108</sup>. También es una alternativa rentable a soluciones comerciales como VMware ESXi <sup>108</sup>.

- **Requisitos de Hardware:** Proxmox VE generalmente tiene requisitos de hardware más bajos y es más eficiente en el uso de recursos en comparación con Openstack <sup>108</sup>. Puede funcionar bien incluso en hardware más antiguo y requiere menos infraestructura para empezar <sup>108</sup>. Openstack, especialmente para despliegues a gran escala, requiere una infraestructura de hardware más robusta y una cantidad considerable de recursos de cómputo, almacenamiento y red para funcionar de manera eficiente <sup>108</sup>. Los requisitos de hardware para Openstack pueden aumentar significativamente con el tamaño y la complejidad del despliegue.

La siguiente tabla resume las principales diferencias entre Openstack y Proxmox VE:

Característica	Openstack	Proxmox VE
Arquitectura	Modular (microservicios)	Integrada (hiperconvergente)
Virtualización	KVM (principal), soporta otros	KVM y LXC
Interfaz de Usuario	Horizon/Skyline (web), CLI	GUI web intuitiva, CLI
API	REST API extensa	REST API
Complejidad	Alta	Moderada
Escalabilidad	Muy alta (miles de nodos)	Moderada (hasta 32 nodos recomendado)
Multi-Tenancy	Robusta, diseñada para gran escala	Limitada, requiere soluciones de terceros para gran escala
Alta Disponibilidad	Requiere configuración adicional (HAproxy, Pacemaker)	Integrada

Casos de Uso	Nubes públicas/privadas grandes, IaaS/PaaS	Virtualización PYMES, laboratorios, entornos de prueba/desarrollo
Requisitos de Hardware	Altos	Bajos a moderados
Licencia	Apache 2.0	GNU AGPLv3
Soporte	Comunidad grande, soporte comercial opcional	Comunidad, suscripciones de soporte comercial disponibles
Redes	SDN avanzado (Neutron)	SDN básico (puentes, VLANs, VxLAN)
Almacenamiento	Soporta múltiples backends (Ceph, Swift, etc.)	Soporta varios backends (ZFS, Ceph, NFS, etc.)
Backup	Requiere configuración adicional (Freezer, Karbor)	Integrado

### Ventajas y Desventajas de Usar Openstack versus Proxmox

La elección entre Openstack y Proxmox VE depende en gran medida de las necesidades y recursos específicos de cada organización.

- Ventajas de Usar Openstack sobre Proxmox VE:** Openstack ofrece una escalabilidad superior, diseñada para infraestructuras a gran escala que pueden crecer a miles de nodos en múltiples centros de datos <sup>112</sup>. Su arquitectura modular proporciona una gran flexibilidad y capacidad de personalización, permitiendo a las organizaciones seleccionar e integrar solo los componentes que necesitan, lo que lo hace adaptable a entornos complejos y multi-inquilino <sup>109</sup>. Openstack también ofrece un soporte más robusto para la multi-tenancy, con funciones sofisticadas para gestionar múltiples usuarios o tenants con aislamiento estricto y asignación granular de recursos <sup>109</sup>. Además, Openstack proporciona opciones de redes avanzadas a través de su servicio Neutron, incluyendo redes definidas por software (SDN), balanceo de carga, VPN como servicio y capacidades de seguridad complejas, lo que lo hace ideal para entornos de nube con requisitos de red exigentes <sup>108</sup>. Su enfoque en la interoperabilidad y la compatibilidad con una amplia gama de hardware y software también es una ventaja significativa <sup>110</sup>.
- Desventajas de Usar Openstack sobre Proxmox VE:** La principal desventaja de Openstack es su complejidad. Su configuración, despliegue y gestión pueden ser

significativamente más difíciles que Proxmox VE, requiriendo un alto nivel de experiencia técnica y un equipo dedicado <sup>109</sup>. Esto puede representar una barrera de entrada para organizaciones con recursos limitados o sin personal especializado en la nube <sup>113</sup>. Además, Openstack puede ser más intensivo en recursos de hardware en comparación con Proxmox VE, especialmente para despliegues a gran escala, lo que puede resultar en mayores costos operativos <sup>108</sup>. La falta de estandarización en las implementaciones de Openstack también puede ser un desafío, ya que al ser un proyecto de código abierto, no existe un único proveedor que garantice la compatibilidad y el soporte integral <sup>116</sup>.

- **Ventajas de Usar Proxmox VE sobre Openstack:** Proxmox VE destaca por su facilidad de uso y configuración, ofreciendo una interfaz web intuitiva que simplifica la gestión de entornos virtualizados <sup>111</sup>. Su proceso de despliegue es más sencillo y rápido en comparación con Openstack, lo que permite una rápida puesta en marcha incluso en hardware más antiguo <sup>108</sup>. Proxmox VE integra en una única plataforma tanto la virtualización completa (KVM) como la virtualización basada en contenedores (LXC), proporcionando flexibilidad para diferentes tipos de cargas de trabajo <sup>109</sup>. También incluye características importantes como alta disponibilidad, migración en vivo y herramientas de backup y restauración integradas, lo que simplifica la gestión de la continuidad del negocio <sup>109</sup>. En términos de requisitos de hardware, Proxmox VE es generalmente menos exigente y más eficiente en el uso de recursos, lo que puede resultar en menores costos de implementación y operación, especialmente para entornos de pequeña a mediana escala <sup>108</sup>.
- **Desventajas de Usar Proxmox VE sobre Openstack:** La principal limitación de Proxmox VE en comparación con Openstack es su escalabilidad. Si bien es adecuado para clusters de tamaño pequeño a mediano (generalmente recomendado hasta 32 nodos), puede no escalar tan bien como Openstack en organizaciones más grandes o en despliegues a gran escala <sup>108</sup>. Proxmox VE también ofrece menos funcionalidades avanzadas en áreas como redes definidas por software (SDN) y balanceo de carga en comparación con Openstack <sup>115</sup>. Su soporte para multi-tenancy es más limitado y puede requerir soluciones de terceros para implementaciones a gran escala <sup>115</sup>. Además, la compatibilidad de Proxmox VE con una amplia gama de hardware y software de terceros puede ser más limitada en comparación con Openstack <sup>110</sup>. Finalmente, aunque Proxmox VE tiene una comunidad activa, es más pequeña en comparación con la vasta comunidad de Openstack, lo que podría afectar la disponibilidad de recursos de terceros y soporte impulsado por la comunidad <sup>114</sup>.

## Conclusiones

Openstack y Proxmox VE representan dos enfoques distintos para la virtualización y la gestión de la nube. Openstack se presenta como una plataforma altamente escalable, flexible y rica en funcionalidades, ideal para organizaciones con necesidades complejas y despliegues a gran escala, incluyendo la construcción de nubes públicas y privadas. Su arquitectura modular y su amplio ecosistema de proyectos ofrecen una gran capacidad de personalización e integración, aunque a costa de una mayor complejidad en su implementación y gestión.

Por otro lado, Proxmox VE se destaca por su facilidad de uso, su integración de virtualización completa y basada en contenedores, y sus características listas para usar como alta disponibilidad y migración en vivo. Es una solución más adecuada para entornos de virtualización de pequeña a mediana escala, centros de datos con recursos limitados o para aquellos que buscan una alternativa sencilla y rentable a soluciones comerciales.

La elección entre Openstack y Proxmox VE debe basarse en una evaluación cuidadosa de las necesidades específicas de cada organización, incluyendo la escala de la infraestructura requerida, la complejidad de las cargas de trabajo, la experiencia técnica del personal de TI y los requisitos de presupuesto. Para organizaciones que priorizan la escalabilidad masiva, la multi-tenancy avanzada y una amplia gama de servicios en la nube, Openstack probablemente sea la mejor opción. Para aquellas que buscan una solución de virtualización potente pero más sencilla de gestionar para entornos más pequeños, Proxmox VE ofrece una alternativa atractiva y eficiente.

## Obras citadas

1. learn.microsoft.com, fecha de acceso: marzo 29, 2025, <https://learn.microsoft.com/en-us/answers/questions/1195636/private-cloud-definition#:~:text=The%20answer%20to%20this%20question,a%20third%2Dparty%20data%20center.>
2. What is a Private Cloud? - AWS, fecha de acceso: marzo 29, 2025, <https://aws.amazon.com/what-is/private-cloud/>
3. What is private cloud? | VMware, fecha de acceso: marzo 29, 2025, <https://www.vmware.com/topics/private-cloud>
4. Private Cloud - Definition, Benefits & FAQs - Nutanix, fecha de acceso: marzo 29, 2025, <https://www.nutanix.com/info/private-cloud>
5. What is private cloud | Google Cloud, fecha de acceso: marzo 29, 2025, <https://cloud.google.com/discover/what-is-a-private-cloud>
6. What is a Private Cloud - Definition - Microsoft Azure, fecha de acceso: marzo 29, 2025,

<https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-a-private-cloud>

7. What is Community Cloud? Definition, Features, and Use Cases - Zesty, fecha de acceso: marzo 29, 2025, <https://zesty.co/finops-glossary/community-cloud/>
8. azure.microsoft.com, fecha de acceso: marzo 29, 2025, [https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-a-re-private-public-hybrid-clouds#:~:text=The%20cloud%20resources%20\(like%20servers,example%20of%20a%20public%20cloud.](https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-a-re-private-public-hybrid-clouds#:~:text=The%20cloud%20resources%20(like%20servers,example%20of%20a%20public%20cloud.)
9. What Is a Public Cloud?, fecha de acceso: marzo 29, 2025, <https://cloud.google.com/learn/what-is-public-cloud>
10. What Is Public Cloud Computing? Examples, Definition, Benefits! - SaM Solutions, fecha de acceso: marzo 29, 2025, <https://sam-solutions.us/insights/what-is-public-cloud-computing/>
11. What Is a Public Cloud? - Definition, Examples & More | Proofpoint US, fecha de acceso: marzo 29, 2025, <https://www.proofpoint.com/us/threat-reference/public-cloud>
12. What is a Public Cloud? - AWS, fecha de acceso: marzo 29, 2025, <https://aws.amazon.com/what-is/public-cloud/>
13. What is Public Cloud? | Glossary | HPE, fecha de acceso: marzo 29, 2025, <https://www.hpe.com/us/en/what-is/public-cloud.html>
14. zesty.co, fecha de acceso: marzo 29, 2025, <https://zesty.co/finops-glossary/community-cloud/#:~:text=Shared%20Infrastructure%20for%20Collaborative%20Use,%2C%20research%2C%20and%20healthcare%20applications.>
15. How does a community cloud work? - IONOS, fecha de acceso: marzo 29, 2025, <https://www.ionos.com/digitalguide/server/know-how/community-cloud/>
16. What is Community Cloud? Benefits & Examples - ESDS, fecha de acceso: marzo 29, 2025, <https://www.esds.co.in/blog/what-is-community-cloud-make-the-most-out-of-esds-community-cloud/>
17. What is Community Cloud? Advantages & Examples - phoenixNAP, fecha de acceso: marzo 29, 2025, <https://phoenixnap.com/blog/community-cloud>
18. Community cloud - Wikipedia, fecha de acceso: marzo 29, 2025, [https://en.wikipedia.org/wiki/Community\\_cloud](https://en.wikipedia.org/wiki/Community_cloud)
19. cloud.google.com, fecha de acceso: marzo 29, 2025, <https://cloud.google.com/learn/what-is-hybrid-cloud#:~:text=A%20hybrid%20cloud%20is%20a.centers%20or%20%E2%80%9Cedge%E2%80%9D%20locations.>
20. What is Hybrid Cloud? - F5, fecha de acceso: marzo 29, 2025, <https://www.f5.com/glossary/hybrid-cloud>
21. What is hybrid cloud? | NetApp, fecha de acceso: marzo 29, 2025, <https://www.netapp.com/hybrid-cloud/what-is-hybrid-cloud/>
22. What is a Hybrid Cloud?, fecha de acceso: marzo 29, 2025, <https://cloud.google.com/learn/what-is-hybrid-cloud>
23. What is a Hybrid Cloud? | Microsoft Azure, fecha de acceso: marzo 29, 2025, <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is>



[-hybrid-cloud-computing](#)

24. OpenStack architecture | Canonical, fecha de acceso: marzo 29, 2025, <https://canonical.com/openstack/architecture>
25. Mastering the Core Concepts of OpenStack Cloud Architecture - Forgeahead, fecha de acceso: marzo 29, 2025, <https://forgeahead.io/2024/07/02/master-openstack-cloud-architecture/>
26. Introduction to OpenStack | GeeksforGeeks, fecha de acceso: marzo 29, 2025, <https://www.geeksforgeeks.org/introduction-to-openstack/>
27. Diagram of openstack components - ResearchGate, fecha de acceso: marzo 29, 2025, [https://www.researchgate.net/figure/Diagram-of-openstack-components\\_fig1\\_347804817](https://www.researchgate.net/figure/Diagram-of-openstack-components_fig1_347804817)
28. OpenStack: Open Source Cloud Computing Infrastructure, fecha de acceso: marzo 29, 2025, <https://www.openstack.org/>
29. OpenStack architecture [10] | Download Scientific Diagram - ResearchGate, fecha de acceso: marzo 29, 2025, [https://www.researchgate.net/figure/OpenStack-architecture-10\\_fig3\\_349248333](https://www.researchgate.net/figure/OpenStack-architecture-10_fig3_349248333)
30. Large Scale SUSE OpenStack Clouds - An Architecture Guide, fecha de acceso: marzo 29, 2025, [https://documentation.suse.com/sbp/cloud/pdf/SBP-CloudLS-master\\_en.pdf](https://documentation.suse.com/sbp/cloud/pdf/SBP-CloudLS-master_en.pdf)
31. Architecture Guide | Red Hat Product Documentation, fecha de acceso: marzo 29, 2025, [https://docs.redhat.com/en/documentation/red\\_hat\\_openstack\\_platform/8/html-single/architecture\\_guide/index](https://docs.redhat.com/en/documentation/red_hat_openstack_platform/8/html-single/architecture_guide/index)
32. What are OpenStack Services? - OpenMetal, fecha de acceso: marzo 29, 2025, <https://openmetal.io/docs/glossary/what-are-openstack-services/>
33. 3. Computing service: OpenStack Nova - University of Cambridge, fecha de acceso: marzo 29, 2025, <https://docs.hpc.cam.ac.uk/cloud/userguide/03-nova.html>
34. OpenStack Glance Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025, <https://github.com/openstack/puppet-glance>
35. What is OpenStack Glance? - OpenMetal, fecha de acceso: marzo 29, 2025, <https://openmetal.io/docs/glossary/what-is-openstack-glance/>
36. Understanding Cinder in the OpenStack Ecosystem - Alibaba Cloud, fecha de acceso: marzo 29, 2025, [https://www.alibabacloud.com/tech-news/a/cinder\\_/4opvkbznzi2a-understanding-cinder-in-the-openstack-ecosystem](https://www.alibabacloud.com/tech-news/a/cinder_/4opvkbznzi2a-understanding-cinder-in-the-openstack-ecosystem)
37. 4. Storage services: OpenStack Cinder and Swift - University of Cambridge, fecha de acceso: marzo 29, 2025, [https://docs.hpc.cam.ac.uk/cloud/userguide/06-cinder\\_swift.html](https://docs.hpc.cam.ac.uk/cloud/userguide/06-cinder_swift.html)
38. 2. Networking service: OpenStack Neutron - University of Cambridge, fecha de acceso: marzo 29, 2025, <https://docs.hpc.cam.ac.uk/cloud/userguide/02-neutron.html>
39. OpenStack Keystone for Cloud Security - Alibaba Cloud, fecha de acceso: marzo

29, 2025,

<https://www.alibabacloud.com/tech-news/a/keystone/gw17802iod-openstack-keystone-for-cloud-security>

40. What is OpenStack Keystone? - OpenMetal, fecha de acceso: marzo 29, 2025, <https://openmetal.io/docs/glossary/what-is-openstack-keystone/>
41. OpenStack Horizon Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025, <https://github.com/openstack/puppet-horizon>
42. Puppet module for OpenStack Horizon, fecha de acceso: marzo 29, 2025, <https://forge.puppet.com/modules/openstack/horizon/readme>
43. OpenStack Swift Puppet Module - OpenDev, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/puppet-swift>
44. forge.puppet.com, fecha de acceso: marzo 29, 2025, <https://forge.puppet.com/modules/openstack/heat/readme#:~:text=The%20heat%20module%20is%20part,the%20orchestration%20service%20for%20OpenStack.ck.>
45. Puppet module for OpenStack Heat, fecha de acceso: marzo 29, 2025, <https://forge.puppet.com/modules/openstack/heat/readme>
46. Configuring the Telemetry (ceilometer) service (optional) — openstack-ansible 13.3.18 documentation - Index of /, fecha de acceso: marzo 29, 2025, <https://static.opendev.org/docs/openstack-ansible/13.3.18/install-guide/configure-ceilometer.html>
47. openstack/puppet-ceilometer - OpenDev, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/puppet-ceilometer>
48. openstack/puppet-trove - OpenDev, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/puppet-trove>
49. Introduction to OpenStack — Security Guide documentation, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/security-guide/introduction/introduction-to-openstack.html>
50. OpenStack Ironic Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025, <https://github.com/openstack/puppet-ironic>
51. Using OpenStack Ironic to Streamline Bare Metal Management - OpenMetal, fecha de acceso: marzo 29, 2025, <https://openmetal.io/resources/blog/openstack-ironic-bare-metal-management/>
52. Modules List — puppet-openstack-guide 0.0.1.dev251 documentation, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/puppet-openstack-guide/latest/contributor/module-list.html>
53. OpenStack Project Teams — OpenStack Technical Committee ..., fecha de acceso: marzo 29, 2025, <https://governance.openstack.org/reference/projects/>
54. OpenStack Designate Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025, <https://github.com/openstack/puppet-designate>

55. Introduction to Designate - OpenStack Documentation, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/designate/2023.2/intro/index.html>
56. Manila - OpenStack Wiki, fecha de acceso: marzo 29, 2025, <https://wiki.openstack.org/wiki/Manila>
57. An Overview of OpenStack Barbican for Secure Key Management | VEXXHOST, fecha de acceso: marzo 29, 2025, <https://vexxhost.com/blog/openstack-barbican-for-key-management/>
58. Welcome to Barbican's developer documentation!, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/barbican/mitaka/>
59. OpenStack Zaqr Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025, <https://github.com/openstack/puppet-zaqr>
60. Mistral Workflow Service — Mistral - OpenStack Documentation, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/mistral/latest/>
61. Puppet module for OpenStack Tacker, fecha de acceso: marzo 29, 2025, <https://forge.puppet.com/modules/openstack/tacker/readme>
62. Senlin - OpenStack Wiki, fecha de acceso: marzo 29, 2025, <https://wiki.openstack.org/wiki/Senlin>
63. RETIRED, Clustering service for managing homogeneous objects in OpenStack - senlin - OpenDev: Free Software Needs Free Tools, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/senlin/src/commit/e060eee36da59774e2c8e835e6880dda45ef8b14>
64. Senlin Authorization - OpenStack Docs, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/senlin/ocata/developer/authorization.html>
65. Senlin Basics - OpenStack Docs, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/senlin/ocata/tutorial/basics.html>
66. Congress - OpenStack Wiki, fecha de acceso: marzo 29, 2025, <https://wiki.openstack.org/wiki/Congress>
67. openstack/congress - OpenDev, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/congress/src/commit/0f94a04d891e112561b0512627e93031ef3aca7f>
68. openstack/congress - OpenDev, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/congress/src/commit/d7f122db21f0cf903ef8d04442cbecdd9b033f17>
69. Congress Introduction and Installation - Read the Docs, fecha de acceso: marzo 29, 2025, <https://congress.readthedocs.io/en/latest/readme.html>
70. Team and repository tags — congress 6.0.0.0b3.dev14 documentation, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/developer/congress/README.html>
71. Masakari - OpenStack Wiki, fecha de acceso: marzo 29, 2025, <https://wiki.openstack.org/wiki/Masakari>
72. Kolla provides production-ready containers and deployment tools for operating OpenStack clouds - OpenDev, fecha de acceso: marzo 29, 2025, <https://opendev.org/openstack/kolla>

73. openstack/kolla-ansible - OpenDev, fecha de acceso: marzo 29, 2025,  
<https://opendev.org/openstack/kolla-ansible/src/commit/438f98ac3fc8b511e1fba981c52c0619963e3891>
74. Kuryr - OpenStack Wiki, fecha de acceso: marzo 29, 2025,  
<https://wiki.openstack.org/wiki/Kuryr>
75. OpenStack Octavia Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025,  
<https://github.com/openstack/puppet-octavia>
76. OpenStack Placement Puppet Module. Mirror of code maintained at opendev.org. - GitHub, fecha de acceso: marzo 29, 2025,  
<https://github.com/openstack/puppet-placement>
77. Skyline Console (UI of OpenStack Modern Dashboard) — skyline ..., fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/skyline-console/latest/>
78. Log management service (Venus) — Venus, fecha de acceso: marzo 29, 2025,  
<https://docs.openstack.org/venus/latest/>
79. Welcome to Vitrage documentation!, fecha de acceso: marzo 29, 2025,  
<https://docs.openstack.org/vitrage/ocata/>
80. OpenStack Watcher Demonstration | OpenMetal Docs, fecha de acceso: marzo 29, 2025,  
<https://openmetal.io/docs/manuals/operators-manual/day-4/watcher/watcher-demo>
81. Gnocchi - OpenStack Wiki, fecha de acceso: marzo 29, 2025,  
<https://wiki.openstack.org/wiki/Gnocchi>
82. Chapter 3. Configuring the Time Series Database (Gnocchi) for Telemetry | Red Hat Product Documentation, fecha de acceso: marzo 29, 2025,  
[https://docs.redhat.com/en/documentation/red\\_hat\\_openstack\\_platform/16.0/html/logging\\_monitoring\\_and\\_troubleshooting\\_guide/configuring\\_the\\_time\\_series\\_database\\_gnocchi\\_for\\_telemetry](https://docs.redhat.com/en/documentation/red_hat_openstack_platform/16.0/html/logging_monitoring_and_troubleshooting_guide/configuring_the_time_series_database_gnocchi_for_telemetry)
83. Introduction — karbor 0.1.1.dev2 documentation, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/karbor/newton/readme.html>
84. Karbor - OpenStack wiki, fecha de acceso: marzo 29, 2025,  
<https://wiki.openstack.org/wiki/Karbor>
85. Karbor: Application Data Protection - OpenStack Docs, fecha de acceso: marzo 29, 2025, <https://docs.openstack.org/karbor/rocky/>
86. OpenStack Docs: Introduction, fecha de acceso: marzo 29, 2025,  
<https://docs.openstack.org/karbor/rocky/readme.html>
87. [openstack-dev] User mailing lists for OpenStack projects, fecha de acceso: marzo 29, 2025,  
<http://openstack-dev.openstack.narkive.com/Cy5vKkHQ/user-mailing-lists-for-openstack-projects>
88. Puppet - OpenStack Wiki, fecha de acceso: marzo 29, 2025,  
<https://wiki.openstack.org/wiki/Puppet>
89. Welcome to Murano Documentation, fecha de acceso: marzo 29, 2025,  
<https://docs.openstack.org/murano/newton/>
90. Murano/ApplicationCatalog - OpenStack Wiki, fecha de acceso: marzo 29, 2025,

- <https://wiki.openstack.org/wiki/Murano/ApplicationCatalog>
91. openstack/murano-dashboard - OpenDev, fecha de acceso: marzo 29, 2025,  
<https://opendev.org/openstack/murano-dashboard/src/commit/1fc39ac63f7355a96e595111d2f7c1355fb7cf30>
  92. Murano/MultiCloudSupport - OpenStack Wiki, fecha de acceso: marzo 29, 2025,  
<https://wiki.openstack.org/wiki/Murano/MultiCloudSupport>
  93. Search Service API - OpenStack Docs, fecha de acceso: marzo 29, 2025,  
<https://docs.openstack.org/api-ref/search/>
  94. Searchlight - OpenStack Wiki, fecha de acceso: marzo 29, 2025,  
<https://wiki.openstack.org/wiki/Searchlight>
  95. openstack/searchlight - OpenDev, fecha de acceso: marzo 29, 2025,  
<https://opendev.org/openstack/searchlight/src/commit/2cd03192f8afc79c25cad16f9ad33222a73dc347>
  96. Searchlight Architecture - OpenStack Docs, fecha de acceso: marzo 29, 2025,  
<https://docs.openstack.org/searchlight/pike/admin/architecture.html>
  97. Learn about OpenStack services and their functions - Ubuntu, fecha de acceso: marzo 29, 2025,  
<https://ubuntu.com/tutorials/learn-about-openstack-services-and-their-functions>
  98. OpenStack with Sunbeam for small-scale private cloud infrastructure - Ubuntu, fecha de acceso: marzo 29, 2025,  
<https://ubuntu.com/blog/openstack-with-sunbeam-for-small-scale-private-cloud-infrastructure>
  99. Sunbeam (Deployment and operational tooling for OpenStack), fecha de acceso: marzo 29, 2025,  
<https://governance.openstack.org/tc/reference/projects/sunbeam.html>
  100. Building the Open Infrastructure Blueprint: How Linux, OpenStack, and Kubernetes Combine to Deliver an Open Source Powered Stack - OpenInfra Foundation, fecha de acceso: marzo 29, 2025,  
<https://openinfra.org/open-infrastructure-blueprint-white-paper/>
  101. OpenStack with Sunbeam as an on-prem extension of the OpenStack public cloud - Ubuntu, fecha de acceso: marzo 29, 2025,  
<https://ubuntu.com/blog/openstack-with-sunbeam-as-an-on-prem-extension-of-the-openstack-public-cloud>
  102. Proxmox Virtual Environment - Open-Source Server Virtualization ..., fecha de acceso: marzo 29, 2025,  
<https://www.proxmox.com/en/products/proxmox-virtual-environment/overview>
  103. community.veeam.com, fecha de acceso: marzo 29, 2025,  
<https://community.veeam.com/blogs-and-podcasts-57/proxmox-virtual-environment-architecture-services-and-user-tools-8067#:~:text=Proxmox%20VE%20is%20based%20on.virtual%20machines%20and%20containers%20simultaneously.&text=The%20hypervisor%20layer%20uses%20an,not%20yet%20be%20accustom%20to.>
  104. Proxmox Virtual Environment: Architecture, Services, and User Tools, fecha de acceso: marzo 29, 2025,

<https://community.veeam.com/blogs-and-podcasts-57/proxmox-virtual-environment-architecture-services-and-user-tools-8067>

105. Proxmox Virtual Environment - Wikipedia, fecha de acceso: marzo 29, 2025, [https://en.wikipedia.org/wiki/Proxmox\\_Virtual\\_Environment](https://en.wikipedia.org/wiki/Proxmox_Virtual_Environment)
106. Features - Proxmox Virtual Environment, fecha de acceso: marzo 29, 2025, <https://www.proxmox.com/en/products/proxmox-virtual-environment/features>
107. What is Proxmox Virtual Environment (VE)? - Unitrends, fecha de acceso: marzo 29, 2025, <https://www.unitrends.com/blog/what-is-proxmox-virtual-environment/>
108. Proxmox VE vs. OpenStack: Open-Source Virtualisierung im Vergleich - Cloud & Heat, fecha de acceso: marzo 29, 2025, <https://www.cloudandheat.com/en/proxmox-ve-vs-openstack-vergleich-der-open-source-virtualisierungslosungen-fur-cloud-infrastrukturen/>
109. OpenStack vs Proxmox: A Comprehensive Comparison Guide - ReadySpace, fecha de acceso: marzo 29, 2025, <https://readyspace.com/openstack-vs-proxmox/>
110. Proxmox vs OpenStack: Comparing Open Source Cloud Software, fecha de acceso: marzo 29, 2025, <https://openmetal.io/resources/blog/proxmox-vs-openstack-comparing-open-source-cloud-software/>
111. Proxmox vs Openstack: A Friendly Guide for Tech Enthusiasts - ReadySpace, fecha de acceso: marzo 29, 2025, <https://readyspace.com/proxmox-vs-openstack/>
112. Proxmox vs OpenStack: Choosing the Right Private Cloud Platform for Your Business, fecha de acceso: marzo 29, 2025, <https://www.horizoniq.com/blog/proxmox-vs-openstack/>
113. Cloud Computing Platforms: Alternative To OpenStack and Proxmox - Warren.io, fecha de acceso: marzo 29, 2025, <https://warren.io/cloud-computing-platforms-warren-io-as-an-alternative-to-openstack-and-proxmox/>
114. Openstack or Proxmox: Which Virtualization Software is Better? - Atomic Networks, fecha de acceso: marzo 29, 2025, <https://blog.atomicnetworks.co/cloud/comparisons/openstack-vs-proxmox>
115. Openstack vs. Proxmox: Which Virtualization Solution is Right for Me? - Cloudification, fecha de acceso: marzo 29, 2025, <https://cloudification.io/cloud-blog/openstack-vs-proxmox-which-virtualization-solution-is-right-for-me/>
116. The 3 Most Common OpenStack Implementation Challenges (and How to Overcome Them!), fecha de acceso: marzo 29, 2025, <https://openmetal.io/resources/blog/the-3-most-common-openstack-implementation-challenges-and-how-to-overcome-them/>