Red Hat OpenShift con NetApp

NetApp container solutions

NetApp
January 21, 2026

Tabla de contenidos

Red Hat OpenShift con NetApp

NVA-1160: Red Hat OpenShift con NetApp

Alan Cowles y Nikhil M Kulkarni, NetApp

Este documento de referencia proporciona validación de la implementación de la solución Red Hat OpenShift, implementada a través de la Infraestructura aprovisionada por el instalador (IPI) en varios entornos de centros de datos diferentes, según lo validado por NetApp. También detalla la integración del almacenamiento con los sistemas de almacenamiento NetApp mediante el uso del orquestador de almacenamiento Trident para la gestión del almacenamiento persistente. Por último, se exploran y documentan una serie de validaciones de soluciones y casos de uso del mundo real.

Casos de uso

La solución Red Hat OpenShift con NetApp está diseñada para ofrecer un valor excepcional a los clientes con los siguientes casos de uso:

- Fácil de implementar y administrar Red Hat OpenShift implementado mediante IPI (Infraestructura aprovisionada por el instalador) en hardware, Red Hat OpenStack Platform, Red Hat Virtualization y VMware vSphere.
- Potencia combinada de contenedores empresariales y cargas de trabajo virtualizadas con Red Hat OpenShift implementado virtualmente en OSP, RHV o vSphere, o en hardware con OpenShift Virtualization.
- Configuración del mundo real y casos de uso que resaltan las características de Red Hat OpenShift cuando se utiliza con almacenamiento de NetApp y Trident, el orquestador de almacenamiento de código abierto para Kubernetes.

Valor empresarial

Las empresas están adoptando cada vez más prácticas de DevOps para crear nuevos productos, acortar los ciclos de lanzamiento y agregar rápidamente nuevas funciones. Debido a su naturaleza ágil innata, los contenedores y los microservicios juegan un papel crucial en el apoyo a las prácticas de DevOps. Sin embargo, practicar DevOps a escala de producción en un entorno empresarial presenta sus propios desafíos e impone ciertos requisitos en la infraestructura subyacente, como los siguientes:

- Alta disponibilidad en todas las capas de la pila
- Facilidad de procedimientos de implementación
- Operaciones y actualizaciones sin interrupciones
- Infraestructura programable e impulsada por API para mantenerse al día con la agilidad de los microservicios
- Multitenencia con garantías de rendimiento
- Capacidad de ejecutar cargas de trabajo virtualizadas y en contenedores simultáneamente
- Capacidad de escalar la infraestructura de forma independiente en función de las demandas de carga de trabajo

Red Hat OpenShift con NetApp reconoce estos desafíos y presenta una solución que ayuda a abordar cada inquietud al implementar la implementación totalmente automatizada de RedHat OpenShift IPI en el entorno de centro de datos elegido por el cliente.

Descripción general de la tecnología

La solución Red Hat OpenShift con NetApp se compone de los siguientes componentes principales:

Plataforma de contenedores Red Hat OpenShift

Red Hat OpenShift Container Platform es una plataforma Kubernetes empresarial totalmente compatible. Red Hat realiza varias mejoras en Kubernetes de código abierto para ofrecer una plataforma de aplicaciones con todos los componentes totalmente integrados para crear, implementar y administrar aplicaciones en contenedores.

Para obtener más información, visite el sitio web de OpenShift "aquí" .

Sistemas de almacenamiento NetApp

NetApp tiene varios sistemas de almacenamiento perfectos para centros de datos empresariales e implementaciones de nube híbrida. La cartera de NetApp incluye los sistemas de almacenamiento NetApp ONTAP, NetApp Element y NetApp e-Series, todos los cuales pueden proporcionar almacenamiento persistente para aplicaciones en contenedores.

Para obtener más información, visite el sitio web de NetApp "aquí" .

Integraciones de almacenamiento de NetApp

Trident es un orquestador de almacenamiento de código abierto y totalmente compatible con contenedores y distribuciones de Kubernetes, incluido Red Hat OpenShift.

Para obtener más información, visite el sitio web de Trident "aquí" .

Opciones de configuración avanzadas

Esta sección está dedicada a las personalizaciones que los usuarios del mundo real probablemente necesitarían realizar al implementar esta solución en producción, como crear un registro de imágenes privado dedicado o implementar instancias de equilibrador de carga personalizadas.

Matriz de soporte actual para versiones validadas

Tecnología	Objetivo	Versión del software
ONTAP de NetApp	Almacenamiento	9.8, 9.9.1, 9.12.1
NetApp Element	Almacenamiento	12,3
Trident de NetApp	Orquestación de almacenamiento	22.01.0, 23.04, 23.07, 23.10, 24.02
Red Hat OpenShift	Orquestación de contenedores	4.6 EUS, 4.7, 4.8, 4.10, 4.11, 4.12, 4.13, 4.14
VMware vSphere	Virtualización de centros de datos	7.0, 8.0.2

Red Hat Openshift

Descripción general de OpenShift

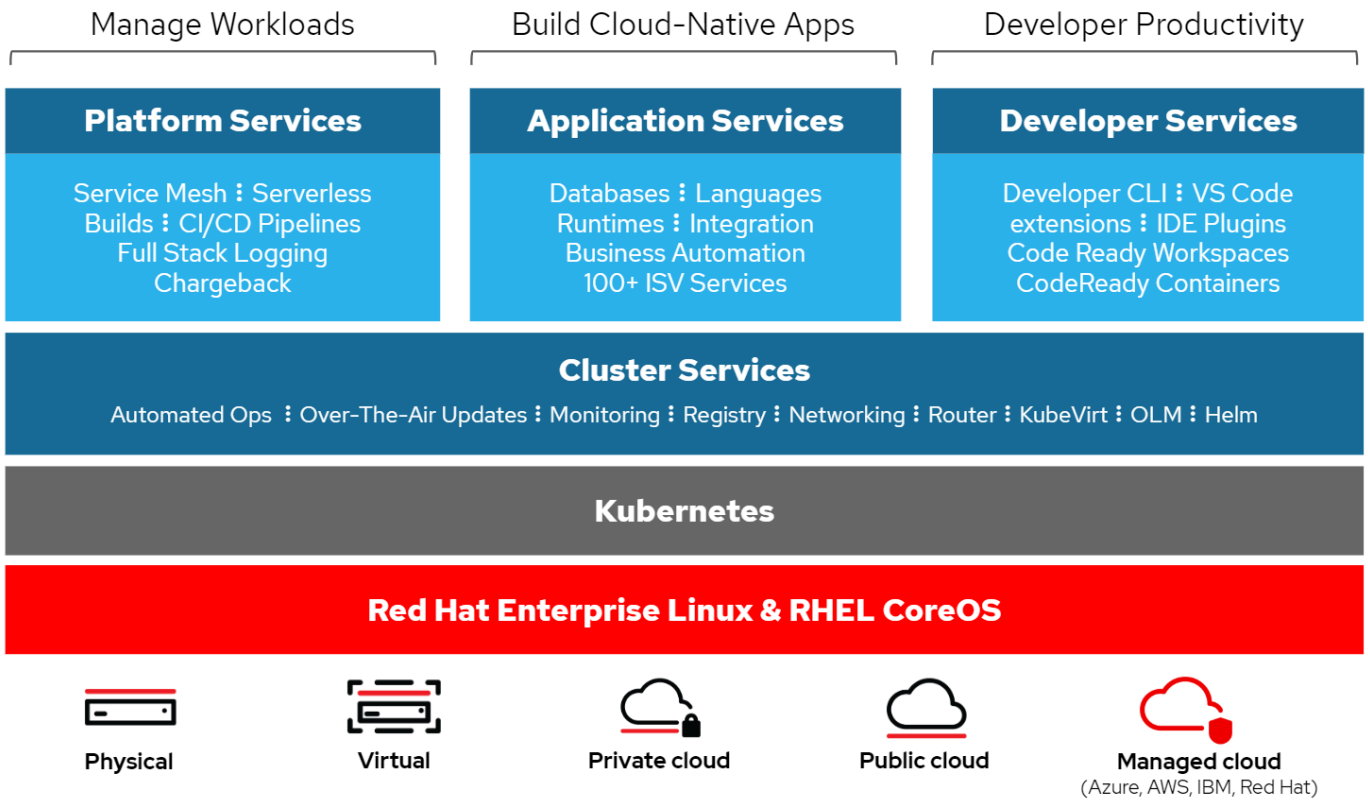
Red Hat OpenShift Container Platform une las operaciones de desarrollo y TI en una única plataforma para crear, implementar y gestionar aplicaciones de forma consistente en infraestructuras locales y de nube híbrida. Red Hat OpenShift se basa en la innovación de código abierto y en los estándares de la industria, incluidos Kubernetes y Red Hat Enterprise Linux CoreOS, la distribución de Linux empresarial líder en el mundo diseñada para cargas de trabajo basadas en contenedores. OpenShift es parte del programa Kubernetes certificado de la Cloud Native Computing Foundation (CNCF), que proporciona portabilidad e interoperabilidad de cargas de trabajo de contenedores.

Red Hat OpenShift ofrece las siguientes capacidades:

- **Aprovisionamiento de autoservicio** Los desarrolladores pueden crear aplicaciones a pedido de manera rápida y sencilla desde las herramientas que más usan, mientras que las operaciones mantienen el control total sobre todo el entorno.
- **Almacenamiento persistente** Al brindar soporte para almacenamiento persistente, OpenShift Container Platform le permite ejecutar aplicaciones con estado y aplicaciones sin estado nativas de la nube.
- **Integración continua y desarrollo continuo (CI/CD)** Esta plataforma de código fuente administra imágenes de compilación e implementación a escala.
- **Estándares de código abierto** Estos estándares incorporan la Open Container Initiative (OCI) y Kubernetes para la orquestación de contenedores, además de otras tecnologías de código abierto. No está restringido a la tecnología ni a la hoja de ruta comercial de un proveedor específico.
- **Canalizaciones de CI/CD** OpenShift ofrece soporte listo para usar para canalizaciones de CI/CD para que los equipos de desarrollo puedan automatizar cada paso del proceso de entrega de aplicaciones y asegurarse de que se ejecute en cada cambio que se realice en el código o la configuración de la aplicación.
- **Control de acceso basado en roles (RBAC)** Esta función proporciona seguimiento de equipos y usuarios para ayudar a organizar un gran grupo de desarrolladores.
- **Compilación e implementación automatizadas** OpenShift ofrece a los desarrolladores la opción de crear sus propias aplicaciones en contenedores o hacer que la plataforma cree los contenedores a partir del código fuente de la aplicación o incluso de los binarios. Luego, la plataforma automatiza la implementación de estas aplicaciones en toda la infraestructura en función de las características definidas para las aplicaciones. Por ejemplo, qué cantidad de recursos se deben asignar y en qué parte de la infraestructura se deben implementar para que cumplan con las licencias de terceros.
- **Entornos consistentes** OpenShift se asegura de que el entorno provisto para los desarrolladores y durante todo el ciclo de vida de la aplicación sea consistente desde el sistema operativo hasta las bibliotecas, la versión del tiempo de ejecución (por ejemplo, Java Runtime) e incluso el tiempo de ejecución de la aplicación en uso (por ejemplo, Tomcat) con el fin de eliminar los riesgos originados por entornos inconsistentes.
- **Gestión de la configuración** La gestión de la configuración y de los datos confidenciales está integrada en la plataforma para garantizar que se proporcione una configuración de aplicación coherente e independiente del entorno a la aplicación, sin importar qué tecnologías se utilicen para crear la aplicación o en qué entorno se implemente.
- **Registros y métricas de la aplicación.** La retroalimentación rápida es un aspecto importante del desarrollo de aplicaciones. La monitorización integrada y la gestión de registros de OpenShift

proporcionan métricas inmediatas a los desarrolladores para que puedan estudiar cómo se comporta la aplicación ante los cambios y solucionar los problemas lo antes posible en el ciclo de vida de la aplicación.

- **Catálogo de contenedores y seguridad** OpenShift ofrece multitenencia y protege al usuario de la ejecución de código dañino mediante el uso de seguridad establecida con Security-Enhanced Linux (SELinux), CGroups y Secure Computing Mode (seccomp) para aislar y proteger los contenedores. También proporciona cifrado a través de certificados TLS para los distintos subsistemas y acceso a contenedores certificados de Red Hat (access.redhat.com/containers) que se escanean y califican con un énfasis específico en la seguridad para proporcionar contenedores de aplicaciones certificados, confiables y seguros a los usuarios finales.



Métodos de implementación para Red Hat OpenShift

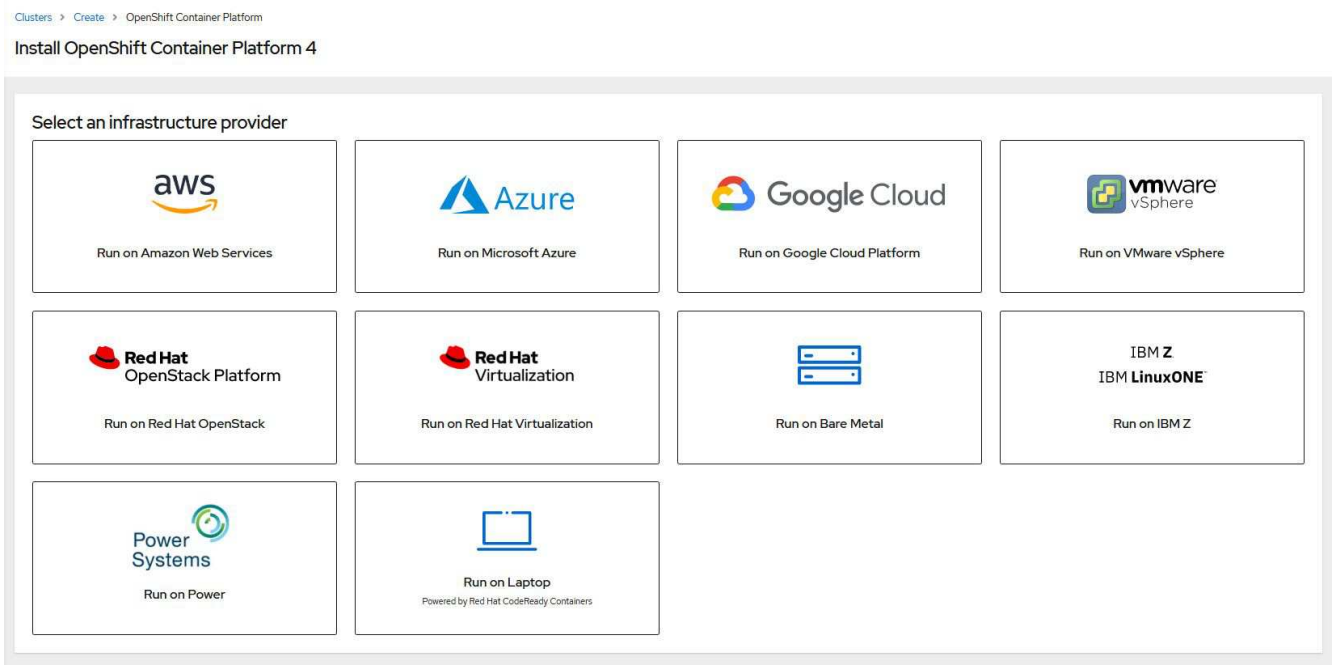
A partir de Red Hat OpenShift 4, los métodos de implementación para OpenShift incluyen implementaciones manuales utilizando Infraestructura aprovisionada por el usuario (UPI) para implementaciones altamente personalizadas o implementaciones totalmente automatizadas utilizando Infraestructura aprovisionada por el instalador (IPI).

El método de instalación IPI es el método preferido en la mayoría de los casos porque permite la implementación rápida de clústeres OpenShift para entornos de desarrollo, prueba y producción.

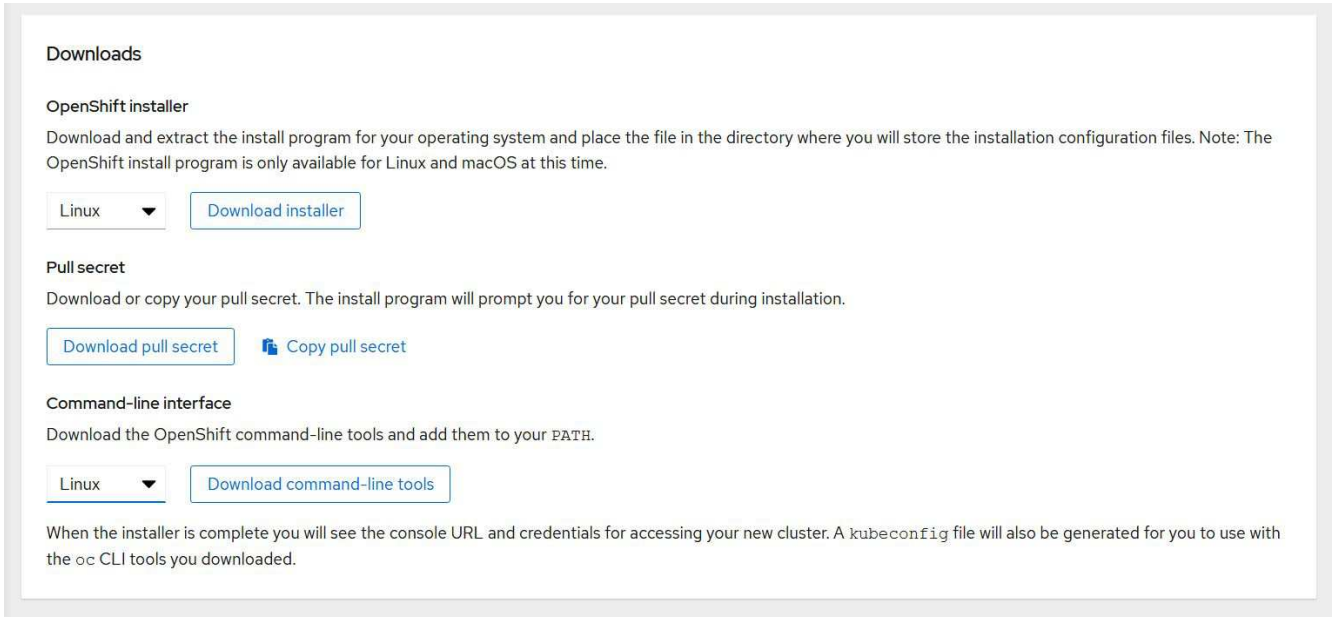
Instalación IPI de Red Hat OpenShift

La implementación de la infraestructura aprovisionada por el instalador (IPI) de OpenShift implica estos pasos de alto nivel:

1. Visita Red Hat OpenShift"sitio web" e inicie sesión con sus credenciales SSO.
2. Seleccione el entorno en el que desea implementar Red Hat OpenShift.



3. En la siguiente pantalla, descargue el instalador, el secreto de extracción único y las herramientas CLI para la administración.



4. Sigue el"instrucciones de instalación" proporcionado por Red Hat para implementar en el entorno de su elección.

Implementaciones de OpenShift validadas por NetApp

NetApp ha probado y validado la implementación de Red Hat OpenShift en sus laboratorios utilizando el método de implementación de Infraestructura aprovisionada por el instalador (IPI) en cada uno de los siguientes entornos de centros de datos:

- "OpenShift en Bare Metal"
- "OpenShift en la plataforma Red Hat OpenStack"

- "OpenShift en Red Hat Virtualization"
- "OpenShift en VMware vSphere"

OpenShift en Bare Metal

OpenShift on Bare Metal proporciona una implementación automatizada de OpenShift Container Platform en servidores básicos.

OpenShift on Bare Metal es similar a las implementaciones virtuales de OpenShift, que brindan facilidad de implementación, aprovisionamiento rápido y escalamiento de clústeres OpenShift, al tiempo que admiten cargas de trabajo virtualizadas para aplicaciones que no están listas para ser contenerizadas. Al realizar la implementación en hardware, no necesita la sobrecarga adicional necesaria para administrar el entorno del hipervisor del host además del entorno de OpenShift. Al implementar directamente en servidores físicos, también puede reducir las limitaciones de sobrecarga física de tener que compartir recursos entre el host y el entorno OpenShift.

OpenShift on Bare Metal ofrece las siguientes características:

- **Implementación de IPI o instalador asistido** Con un clúster OpenShift implementado mediante Infraestructura aprovisionada por el instalador (IPI) en servidores físicos, los clientes pueden implementar un entorno OpenShift altamente versátil y fácilmente escalable directamente en servidores básicos, sin la necesidad de administrar una capa de hipervisor.
- **Diseño de clúster compacto** Para minimizar los requisitos de hardware, OpenShift en hardware permite a los usuarios implementar clústeres de solo 3 nodos, al habilitar que los nodos del plano de control de OpenShift también actúen como nodos de trabajo y contenedores de host.
- **Virtualización de OpenShift** OpenShift puede ejecutar máquinas virtuales dentro de contenedores mediante el uso de virtualización de OpenShift. Esta virtualización nativa de contenedores ejecuta el hipervisor KVM dentro de un contenedor y adjunta volúmenes persistentes para el almacenamiento de VM.
- **Infraestructura optimizada para IA/ML** Implemente aplicaciones como Kubeflow para aplicaciones de aprendizaje automático incorporando nodos de trabajo basados en GPU a su entorno OpenShift y aprovechando la programación avanzada de OpenShift.

Diseño de red

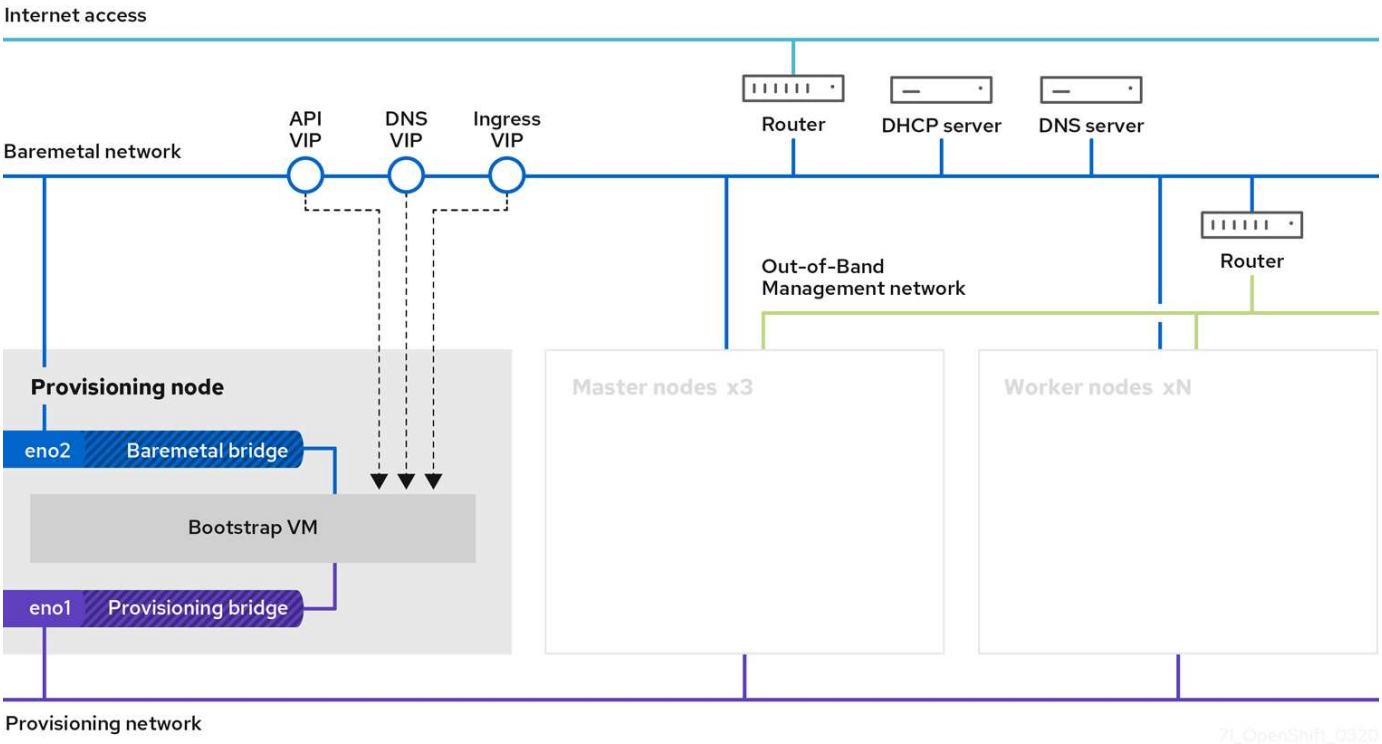
La solución Red Hat OpenShift en NetApp utiliza dos conmutadores de datos para proporcionar conectividad de datos primaria a 25 Gbps. También utiliza dos conmutadores de gestión que proporcionan conectividad a 1 Gbps para la gestión en banda de los nodos de almacenamiento y la gestión fuera de banda para la funcionalidad IPMI.

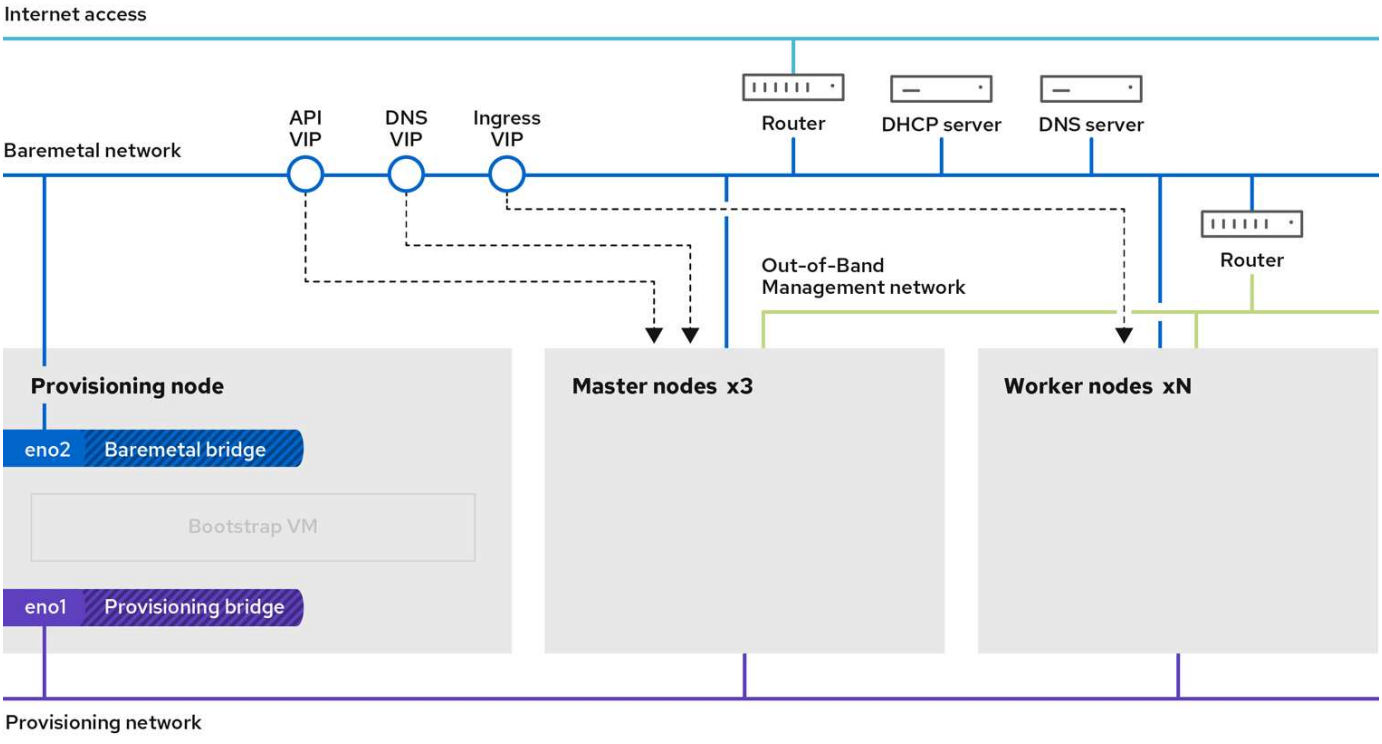
Para la implementación de IPI desde hardware real de OpenShift, debe crear un nodo de aprovisionamiento, una máquina Red Hat Enterprise Linux 8 que debe tener interfaces de red conectadas a redes separadas.

- **Red de aprovisionamiento** Esta red se utiliza para arrancar los nodos físicos e instalar las imágenes y los paquetes necesarios para implementar el clúster OpenShift.
- **Red de hardware** Esta red se utiliza para la comunicación pública del clúster después de su implementación.

Para configurar el nodo aprovisionador, el cliente crea interfaces de puente que permiten que el tráfico se enrute correctamente en el propio nodo y en la máquina virtual Bootstrap aprovisionada para fines de implementación. Una vez implementado el clúster, las direcciones API y VIP de ingreso se migran desde el nodo de arranque al clúster recién implementado.

Las siguientes imágenes representan el entorno tanto durante la implementación de IPI como después de completarse la implementación.





Requisitos de VLAN

La solución Red Hat OpenShift con NetApp está diseñada para separar lógicamente el tráfico de red para diferentes propósitos mediante el uso de redes de área local virtuales (VLAN).

VLAN	Objetivo	ID de VLAN
Red de gestión fuera de banda	Gestión de nodos de hardware y de IPMI	16
Red de metal desnudo	Red para servicios OpenShift una vez que el clúster esté disponible	181
Red de aprovisionamiento	Red para arranque PXE e instalación de nodos bare metal a través de IPI	3485

Aunque cada una de estas redes está virtualmente separada por VLAN, cada puerto físico debe configurarse en modo de acceso con la VLAN principal asignada, porque no hay forma de pasar una etiqueta VLAN durante una secuencia de arranque PXE.

Recursos de soporte de infraestructura de red

La siguiente infraestructura debe estar disponible antes de la implementación de la plataforma de contenedores OpenShift:

- Al menos un servidor DNS que proporcione una resolución de nombre de host completa accesible desde la red de administración en banda y la red de VM.
- Al menos un servidor NTP al que se pueda acceder desde la red de administración en banda y la red de VM.
- (Opcional) Conectividad a Internet saliente tanto para la red de administración en banda como para la red de VM.

OpenShift en la plataforma Red Hat OpenStack

La plataforma Red Hat OpenStack ofrece una base integrada para crear, implementar y escalar una nube OpenStack privada segura y confiable.

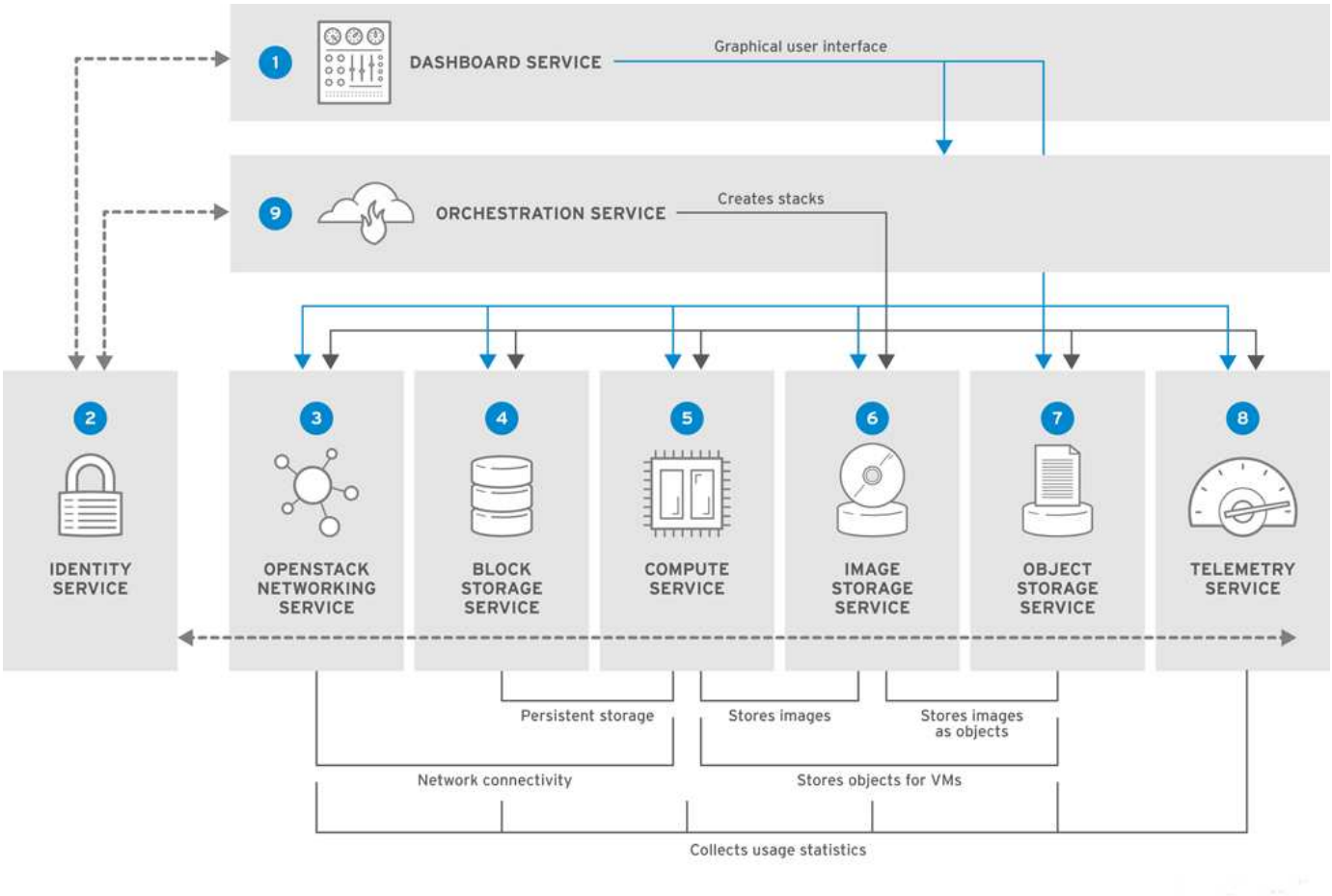
OSP es una nube de infraestructura como servicio (IaaS) implementada por una colección de servicios de control que administran recursos informáticos, de almacenamiento y de red. El entorno se administra mediante una interfaz basada en web que permite a los administradores y usuarios controlar, aprovisionar y automatizar los recursos de OpenStack. Además, la infraestructura de OpenStack se facilita a través de una extensa interfaz de línea de comandos y API que permite capacidades de automatización completas para administradores y usuarios finales.

El proyecto OpenStack es un proyecto comunitario de rápido desarrollo que ofrece versiones actualizadas cada seis meses. Inicialmente, Red Hat OpenStack Platform mantuvo el ritmo de este ciclo de lanzamiento publicando una nueva versión junto con cada versión original y brindando soporte a largo plazo para cada tercer lanzamiento. Recientemente, con el lanzamiento de OSP 16.0 (basado en OpenStack Train), Red Hat decidió no seguir el ritmo de los números de lanzamiento, sino que incorporó nuevas características en sub-lanzamientos. La versión más reciente es Red Hat OpenStack Platform 16.1, que incluye características avanzadas incorporadas desde las versiones Ussuri y Victoria.

Para obtener más información sobre OSP, consulte el"Sitio web de Red Hat OpenStack Platform" .

Servicios de OpenStack

Los servicios de la plataforma OpenStack se implementan como contenedores, lo que aísla los servicios entre sí y permite actualizaciones fáciles. La plataforma OpenStack utiliza un conjunto de contenedores creados y administrados con Kolla. La implementación de servicios se realiza extrayendo imágenes de contenedores del Portal personalizado de Red Hat. Estos contenedores de servicio se administran mediante el comando Podman y se implementan, configuran y mantienen con Red Hat OpenStack Director.



Servicio	Nombre del proyecto	Descripción
Consola	Horizonte	Panel de control basado en navegador web que se utiliza para administrar los servicios de OpenStack.
Identidad	Keystone	Servicio centralizado para la autenticación y autorización de servicios OpenStack y para la gestión de usuarios, proyectos y roles.
Redes OpenStack	Neutrón	Proporciona conectividad entre las interfaces de los servicios de OpenStack.
Almacenamiento en bloque	Ceniza	Administra volúmenes de almacenamiento en bloque persistentes para máquinas virtuales (VM).
Calcular	Estrella nueva	Administra y aprovisiona máquinas virtuales que se ejecutan en nodos de cómputo.
Imagen	Mirada	Servicio de registro utilizado para almacenar recursos como imágenes de máquinas virtuales e instantáneas de volumen.

Servicio	Nombre del proyecto	Descripción
Almacenamiento de objetos	Rápido	Permite a los usuarios almacenar y recuperar archivos y datos arbitrarios.
Telemetría	Ceilómetro	Proporciona mediciones del uso de los recursos de la nube.
Orquestación	Calor	Motor de orquestación basado en plantillas que admite la creación automática de pilas de recursos.

Diseño de red

La solución Red Hat OpenShift con NetApp utiliza dos conmutadores de datos para proporcionar conectividad de datos primaria a 25 Gbps. También utiliza dos conmutadores de gestión adicionales que proporcionan conectividad a 1 Gbps para la gestión en banda de los nodos de almacenamiento y la gestión fuera de banda para la funcionalidad IPMI.

Red Hat OpenStack Director requiere la funcionalidad IPMI para implementar Red Hat OpenStack Platform utilizando el servicio de aprovisionamiento bare-metal de Ironic.

Requisitos de VLAN

Red Hat OpenShift con NetApp está diseñado para separar lógicamente el tráfico de red para diferentes propósitos mediante el uso de redes de área local virtuales (VLAN). Esta configuración se puede escalar para satisfacer las demandas de los clientes o para proporcionar mayor aislamiento para servicios de red específicos. En la siguiente tabla se enumeran las VLAN necesarias para implementar la solución durante la validación de la solución en NetApp.

VLAN	Objetivo	ID de VLAN
Red de gestión fuera de banda	Red utilizada para la gestión de nodos físicos y servicio IPMI para Ironic.	16
Infraestructura de almacenamiento	Red utilizada para que los nodos controladores asignen volúmenes directamente para respaldar servicios de infraestructura como Swift.	201
Cinder de almacenamiento	Red utilizada para mapear y adjuntar volúmenes de bloques directamente a instancias virtuales implementadas en el entorno.	202
API interna	Red utilizada para la comunicación entre los servicios OpenStack mediante comunicación API, mensajes RPC y comunicación de base de datos.	301
Arrendatario	Neutron proporciona a cada inquilino sus propias redes a través de tunelización mediante VXLAN. El tráfico de red está aislado dentro de cada red de inquilino. Cada red de inquilinos tiene una subred IP asociada y los espacios de nombres de red significan que varias redes de inquilinos pueden usar el mismo rango de direcciones sin causar conflictos.	302
Gestión de almacenamiento	OpenStack Object Storage (Swift) utiliza esta red para sincronizar objetos de datos entre los nodos de réplica participantes. El servicio proxy actúa como interfaz intermediaria entre las solicitudes del usuario y la capa de almacenamiento subyacente. El proxy recibe solicitudes entrantes y localiza la réplica necesaria para recuperar los datos solicitados.	303

VLAN	Objetivo	ID de VLAN
PXE	El Director de OpenStack proporciona arranque PXE como parte del servicio de aprovisionamiento de hardware de Ironic para orquestar la instalación de OSP Overcloud.	3484
Externo	Red disponible públicamente que aloja el panel de control de OpenStack (Horizon) para la gestión gráfica y permite llamadas API públicas para administrar los servicios de OpenStack.	3485
Red de gestión en banda	Proporciona acceso a funciones de administración del sistema, como acceso SSH, tráfico DNS y tráfico de Protocolo de tiempo de red (NTP). Esta red también actúa como puerta de enlace para nodos no controladores.	3486

Recursos de soporte de infraestructura de red

La siguiente infraestructura debe estar disponible antes de la implementación de OpenShift Container Platform:

- Al menos un servidor DNS que proporcione una resolución completa del nombre de host.
- Al menos tres servidores NTP que puedan mantener la hora sincronizada de los servidores de la solución.
- (Opcional) Conectividad a Internet saliente para el entorno OpenShift.

Mejores prácticas para implementaciones de producción

En esta sección se enumeran varias prácticas recomendadas que una organización debe tener en cuenta antes de implementar esta solución en producción.

Implementar OpenShift en una nube privada de OSP con al menos tres nodos de cómputo

La arquitectura verificada descrita en este documento presenta la implementación de hardware mínima adecuada para operaciones de alta disponibilidad mediante la implementación de tres nodos de controlador OSP y dos nodos de cómputo OSP. Esta arquitectura garantiza una configuración tolerante a fallas en la que ambos nodos de cómputo pueden lanzar instancias virtuales y las máquinas virtuales implementadas pueden migrar entre los dos hipervisores.

Debido a que Red Hat OpenShift se implementa inicialmente con tres nodos maestros, una configuración de dos nodos podría provocar que al menos dos maestros ocupen el mismo nodo, lo que puede generar una posible interrupción de OpenShift si ese nodo específico deja de estar disponible. Por lo tanto, una buena práctica de Red Hat es implementar al menos tres nodos de cómputo OSP para que los maestros OpenShift puedan distribuirse de manera uniforme y la solución reciba un grado adicional de tolerancia a fallas.

Configurar la afinidad entre la máquina virtual y el host

La distribución de los maestros OpenShift entre múltiples nodos de hipervisor se puede lograr habilitando la afinidad VM/host.

La afinidad es una forma de definir reglas para un conjunto de máquinas virtuales o hosts que determinan si las máquinas virtuales se ejecutan juntas en el mismo host o hosts del grupo o en diferentes hosts. Se aplica a las máquinas virtuales mediante la creación de grupos de afinidad que consisten en máquinas virtuales y/o hosts con un conjunto de parámetros y condiciones idénticos. Dependiendo de si las máquinas virtuales en un grupo de afinidad se ejecutan en el mismo host o hosts del grupo o por separado en diferentes hosts, los parámetros del grupo de afinidad pueden definir afinidad positiva o afinidad negativa. En la plataforma Red Hat OpenStack, se pueden crear y aplicar reglas de afinidad y antiafinidad de host mediante la creación de

grupos de servidores y la configuración de filtros para que las instancias implementadas por Nova en un grupo de servidores se implementen en diferentes nodos de cómputo.

Un grupo de servidores tiene un máximo predeterminado de 10 instancias virtuales cuya ubicación puede administrar. Esto se puede modificar actualizando las cuotas predeterminadas para Nova.

> Hay un límite específico de afinidad/antiafinidad para los grupos de servidores OSP; si no hay suficientes recursos para implementar en nodos separados o no hay suficientes recursos para permitir compartir nodos, la VM no puede iniciarse.

Para configurar grupos de afinidad, consulte"¿Cómo configuro Affinity y Anti-Affinity para instancias de OpenStack?" .

Utilice un archivo de instalación personalizado para la implementación de OpenShift

IPI facilita la implementación de clústeres OpenShift a través del asistente interactivo analizado anteriormente en este documento. Sin embargo, es posible que necesites cambiar algunos valores predeterminados como parte de una implementación de clúster.

En estos casos, puede ejecutar y asignar la tarea al asistente sin implementar inmediatamente un clúster; en lugar de eso, crea un archivo de configuración desde el cual se puede implementar el clúster más tarde. Esto es muy útil si necesita cambiar algún valor predeterminado de IPI o si desea implementar varios clústeres idénticos en su entorno para otros usos, como multitenencia. Para obtener más información sobre cómo crear una configuración de instalación personalizada para OpenShift, consulte "Red Hat OpenShift: Instalación de un clúster en OpenStack con personalizaciones".

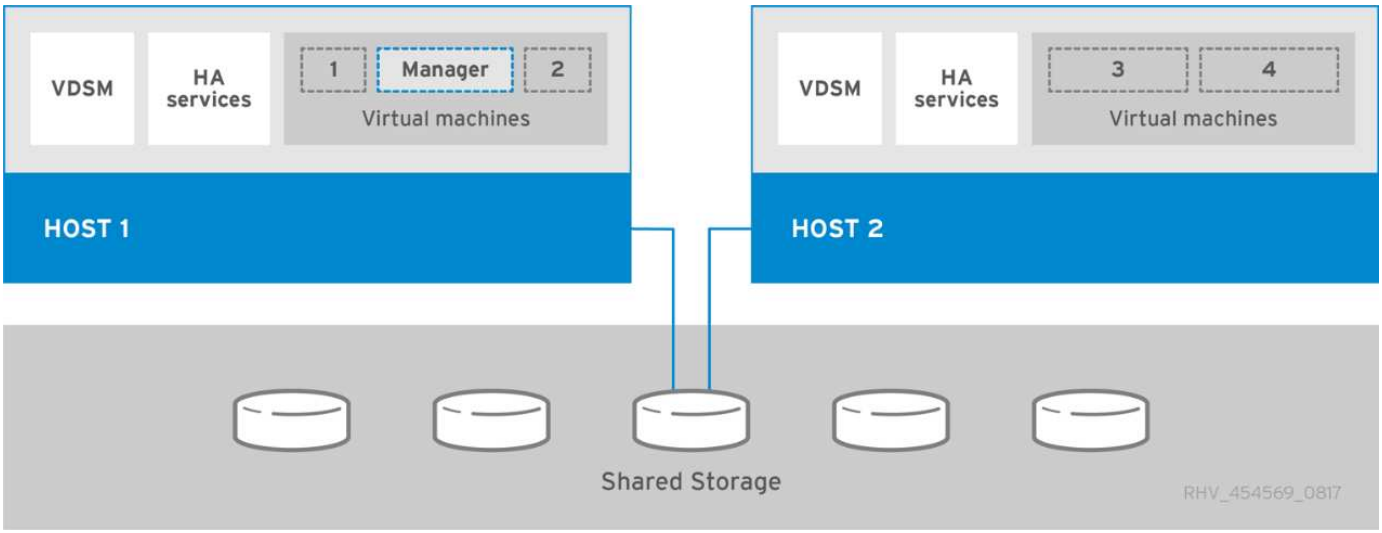
OpenShift en Red Hat Virtualization

Red Hat Virtualization (RHV) es una plataforma de centro de datos virtual empresarial que se ejecuta en Red Hat Enterprise Linux (RHEL) y utiliza el hipervisor KVM.

Para obtener más información sobre RHV, consulte la"Sitio web de Red Hat Virtualization" .

RHV ofrece las siguientes características:

- **Administración centralizada de máquinas virtuales y hosts** El administrador de RHV se ejecuta como una máquina física o virtual (VM) en la implementación y proporciona una GUI basada en web para la administración de la solución desde una interfaz central.
- **Motor autohospedado** Para minimizar los requisitos de hardware, RHV permite que RHV Manager (RHV-M) se implemente como una VM en los mismos hosts que ejecutan VM invitadas.
- **Alta disponibilidad** Para evitar interrupciones en caso de fallas del host, RHV permite configurar las máquinas virtuales para alta disponibilidad. Las máquinas virtuales de alta disponibilidad se controlan a nivel de clúster mediante políticas de resiliencia.
- **Alta escalabilidad** Un solo clúster RHV puede tener hasta 200 hosts de hipervisor, lo que le permite soportar requisitos de máquinas virtuales masivas para alojar cargas de trabajo de clase empresarial que consumen muchos recursos.
- **Seguridad mejorada** Las tecnologías Secure Virtualization (sVirt) y Security Enhanced Linux (SELinux), heredadas de RHV, son empleadas por RHV con el propósito de brindar mayor seguridad y fortalecimiento a los hosts y las máquinas virtuales. La principal ventaja de estas características es el aislamiento lógico de una máquina virtual y sus recursos asociados.



Diseño de red

La solución Red Hat OpenShift en NetApp utiliza dos conmutadores de datos para proporcionar conectividad de datos primaria a 25 Gbps. También utiliza dos conmutadores de gestión adicionales que proporcionan conectividad a 1 Gbps para la gestión en banda de los nodos de almacenamiento y la gestión fuera de banda para la funcionalidad IPMI. OCP utiliza la red lógica de la máquina virtual en RHV para la administración del clúster. Esta sección describe la disposición y el propósito de cada segmento de red virtual utilizado en la solución y describe los requisitos previos para implementar la solución.

Requisitos de VLAN

Red Hat OpenShift en RHV está diseñado para separar lógicamente el tráfico de red para diferentes propósitos mediante el uso de redes de área local virtuales (VLAN). Esta configuración se puede escalar para satisfacer las demandas de los clientes o para proporcionar mayor aislamiento para servicios de red específicos. En la siguiente tabla se enumeran las VLAN necesarias para implementar la solución durante la validación de la solución en NetApp.

VLAN	Objetivo	ID de VLAN
Red de gestión fuera de banda	Gestión de nodos físicos e IPMI	16
Red de máquinas virtuales	Acceso a la red de invitados virtual	1172
Red de gestión en banda	Administración de nodos RHV-H, RHV-Manager y red ovirtmgmt	3343
Red de almacenamiento	Red de almacenamiento para NetApp Element iSCSI	3344
Red de migración	Red para migración de invitados virtuales	3345

Recursos de soporte de infraestructura de red

La siguiente infraestructura debe estar disponible antes de la implementación de OpenShift Container Platform:

- Al menos un servidor DNS que proporcione resolución de nombre de host completa y al que se pueda acceder desde la red de administración en banda y la red de VM.
- Al menos un servidor NTP al que se pueda acceder desde la red de administración en banda y la red de

VM.

- (Opcional) Conectividad a Internet saliente tanto para la red de administración en banda como para la red de VM.

Mejores prácticas para implementaciones de producción

En esta sección se enumeran varias prácticas recomendadas que una organización debe tener en cuenta antes de implementar esta solución en producción.

Implementar OpenShift en un clúster RHV de al menos tres nodos

La arquitectura verificada que se describe en este documento presenta la implementación de hardware mínima adecuada para operaciones de alta disponibilidad mediante la implementación de dos nodos de hipervisor RHV-H y garantizando una configuración tolerante a fallas donde ambos hosts pueden administrar el motor alojado y las máquinas virtuales implementadas pueden migrar entre los dos hipervisores.

Debido a que Red Hat OpenShift se implementa inicialmente con tres nodos maestros, en una configuración de dos nodos se garantiza que al menos dos maestros ocuparán el mismo nodo, lo que puede generar una posible interrupción de OpenShift si ese nodo específico deja de estar disponible. Por lo tanto, una buena práctica de Red Hat es que se implementen al menos tres nodos de hipervisor RHV-H como parte de la solución para que los maestros OpenShift puedan distribuirse de manera uniforme y la solución reciba un grado adicional de tolerancia a fallas.

Configurar la afinidad entre la máquina virtual y el host

Puede distribuir los maestros OpenShift entre múltiples nodos de hipervisor habilitando la afinidad VM/host.

La afinidad es una forma de definir reglas para un conjunto de máquinas virtuales o hosts que determinan si las máquinas virtuales se ejecutan juntas en el mismo host o hosts del grupo o en diferentes hosts. Se aplica a las máquinas virtuales mediante la creación de grupos de afinidad que consisten en máquinas virtuales y/o hosts con un conjunto de parámetros y condiciones idénticos. Dependiendo de si las máquinas virtuales en un grupo de afinidad se ejecutan en el mismo host o hosts del grupo o por separado en diferentes hosts, los parámetros del grupo de afinidad pueden definir afinidad positiva o afinidad negativa.

Las condiciones definidas para los parámetros pueden ser de aplicación estricta o de aplicación flexible. La aplicación estricta garantiza que las máquinas virtuales de un grupo de afinidad siempre sigan estrictamente la afinidad positiva o negativa sin tener en cuenta las condiciones externas. La aplicación suave garantiza que se establezca una preferencia más alta para las máquinas virtuales de un grupo de afinidad para seguir la afinidad positiva o negativa siempre que sea posible. En la configuración de dos o tres hipervisores descrita en este documento, la afinidad suave es la configuración recomendada. En clústeres más grandes, la afinidad dura puede distribuir correctamente los nodos OpenShift.

Para configurar grupos de afinidad, consulte la"Sombrero rojo 6.11. Documentación de grupos de afinidad" .

Utilice un archivo de instalación personalizado para la implementación de OpenShift

IPI facilita la implementación de clústeres OpenShift a través del asistente interactivo analizado anteriormente en este documento. Sin embargo, es posible que existan algunos valores predeterminados que deban modificarse como parte de la implementación del clúster.

En estos casos, puede ejecutar y asignar tareas al asistente sin implementar inmediatamente un clúster. En lugar de ello, se crea un archivo de configuración desde el cual se puede implementar el clúster más adelante. Esto es muy útil si desea cambiar algún valor predeterminado de IPI o si desea implementar varios clústeres idénticos en su entorno para otros usos, como multitenencia. Para obtener más información sobre cómo crear una configuración de instalación personalizada para OpenShift, consulte"Red Hat OpenShift: Instalación de un

clúster en RHV con personalizaciones" .

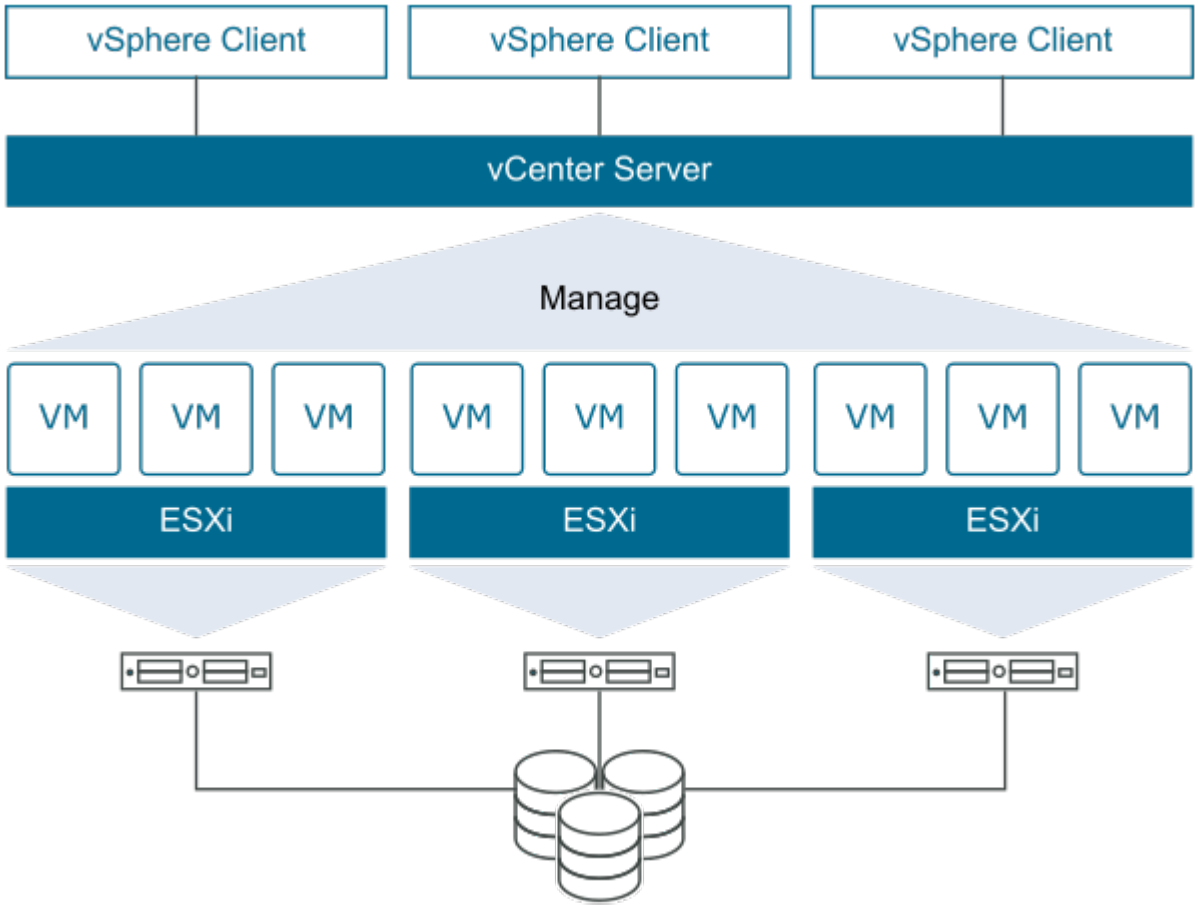
OpenShift en VMware vSphere

VMware vSphere es una plataforma de virtualización para administrar de forma centralizada una gran cantidad de servidores y redes virtualizados que se ejecutan en el hipervisor ESXi.

Para obtener más información sobre VMware vSphere, consulte la"Sitio web de VMware vSphere" .

VMware vSphere ofrece las siguientes características:

- **VMware vCenter Server** VMware vCenter Server proporciona una administración unificada de todos los hosts y máquinas virtuales desde una única consola y agrega la supervisión del rendimiento de clústeres, hosts y máquinas virtuales.
- **VMware vSphere vMotion** VMware vCenter le permite migrar máquinas virtuales en caliente entre nodos del clúster a pedido y de manera no disruptiva.
- **vSphere High Availability** Para evitar interrupciones en caso de fallas del host, VMware vSphere permite agrupar los hosts y configurarlos para alta disponibilidad. Las máquinas virtuales que se interrumpen debido a una falla del host se reinician brevemente en otros hosts del clúster, lo que restaura los servicios.
- **Programador de recursos distribuidos (DRS)** Se puede configurar un clúster de VMware vSphere para equilibrar la carga de las necesidades de recursos de las máquinas virtuales que aloja. Las máquinas virtuales con contenciones de recursos se pueden migrar en caliente a otros nodos del clúster para garantizar que haya suficientes recursos disponibles.



Diseño de red

La solución Red Hat OpenShift en NetApp utiliza dos conmutadores de datos para proporcionar conectividad de datos primaria a 25 Gbps. También utiliza dos conmutadores de gestión adicionales que proporcionan conectividad a 1 Gbps para la gestión en banda de los nodos de almacenamiento y la gestión fuera de banda para la funcionalidad IPMI. OCP utiliza la red lógica de VM en VMware vSphere para la gestión de su clúster. Esta sección describe la disposición y el propósito de cada segmento de red virtual utilizado en la solución y describe los requisitos previos para la implementación de la solución.

Requisitos de VLAN

Red Hat OpenShift en VMware vSphere está diseñado para separar lógicamente el tráfico de red para diferentes propósitos mediante el uso de redes de área local virtuales (VLAN). Esta configuración se puede escalar para satisfacer las demandas de los clientes o para proporcionar mayor aislamiento para servicios de red específicos. En la siguiente tabla se enumeran las VLAN necesarias para implementar la solución durante la validación de la solución en NetApp.

VLAN	Objetivo	ID de VLAN
Red de gestión fuera de banda	Gestión de nodos físicos e IPMI	16
Red de máquinas virtuales	Acceso a la red de invitados virtual	181
Red de almacenamiento	Red de almacenamiento para ONTAP NFS	184
Red de almacenamiento	Red de almacenamiento para ONTAP iSCSI	185
Red de gestión en banda	Administración de nodos ESXi, VCenter Server, ONTAP Select	3480
Red de almacenamiento	Red de almacenamiento para NetApp Element iSCSI	3481
Red de migración	Red para migración de invitados virtuales	3482

Recursos de soporte de infraestructura de red

La siguiente infraestructura debe estar disponible antes de la implementación de OpenShift Container Platform:

- Al menos un servidor DNS que proporcione resolución de nombre de host completa y al que se pueda acceder desde la red de administración en banda y la red de VM.
- Al menos un servidor NTP al que se pueda acceder desde la red de administración en banda y la red de VM.
- (Opcional) Conectividad a Internet saliente tanto para la red de administración en banda como para la red de VM.

Mejores prácticas para implementaciones de producción

En esta sección se enumeran varias prácticas recomendadas que una organización debe tener en cuenta antes de implementar esta solución en producción.

Implementar OpenShift en un clúster ESXi de al menos tres nodos

La arquitectura verificada que se describe en este documento presenta la implementación de hardware

mínima adecuada para operaciones de alta disponibilidad mediante la implementación de dos nodos de hipervisor ESXi y garantizando una configuración tolerante a fallas al habilitar VMware vSphere HA y VMware vMotion. Esta configuración permite que las máquinas virtuales implementadas migren entre los dos hipervisores y se reinicien si un host deja de estar disponible.

Debido a que Red Hat OpenShift se implementa inicialmente con tres nodos maestros, al menos dos maestros en una configuración de dos nodos pueden ocupar el mismo nodo en algunas circunstancias, lo que puede generar una posible interrupción de OpenShift si ese nodo específico deja de estar disponible. Por lo tanto, una buena práctica de Red Hat es que se implementen al menos tres nodos de hipervisor ESXi para que los maestros OpenShift se puedan distribuir de manera uniforme, lo que proporciona un grado adicional de tolerancia a fallas.

Configurar la afinidad de la máquina virtual y el host

Para garantizar la distribución de los maestros OpenShift entre múltiples nodos de hipervisor se puede habilitar la afinidad de máquinas virtuales y hosts.

La afinidad o antiafinidad es una forma de definir reglas para un conjunto de máquinas virtuales o hosts que determinan si las máquinas virtuales se ejecutan juntas en el mismo host o hosts del grupo o en diferentes hosts. Se aplica a las máquinas virtuales mediante la creación de grupos de afinidad que consisten en máquinas virtuales y/o hosts con un conjunto de parámetros y condiciones idénticos. Dependiendo de si las máquinas virtuales en un grupo de afinidad se ejecutan en el mismo host o hosts del grupo o por separado en diferentes hosts, los parámetros del grupo de afinidad pueden definir afinidad positiva o afinidad negativa.

Para configurar grupos de afinidad, consulte la "Documentación de vSphere 9.0: Uso de reglas de afinidad de DRS".

Utilice un archivo de instalación personalizado para la implementación de OpenShift

IPI facilita la implementación de clústeres OpenShift a través del asistente interactivo analizado anteriormente en este documento. Sin embargo, es posible que necesites cambiar algunos valores predeterminados como parte de una implementación de clúster.

En estos casos, puede ejecutar y asignar tareas al asistente sin implementar inmediatamente un clúster, sino que el asistente crea un archivo de configuración desde el cual se puede implementar el clúster más adelante. Esto es muy útil si necesita cambiar algún valor predeterminado de IPI o si desea implementar varios clústeres idénticos en su entorno para otros usos, como multitenencia. Para obtener más información sobre cómo crear una configuración de instalación personalizada para OpenShift, consulte"Red Hat OpenShift: Instalación de un clúster en vSphere con personalizaciones" .

Servicio Red Hat OpenShift en AWS

Red Hat OpenShift Service on AWS (ROSA) es un servicio administrado que puede utilizar para crear, escalar e implementar aplicaciones en contenedores con la plataforma empresarial Kubernetes Red Hat OpenShift en AWS. ROSA agiliza el traslado de cargas de trabajo locales de Red Hat OpenShift a AWS y ofrece una integración estrecha con otros servicios de AWS.

Para obtener más información sobre ROSA, consulte la documentación aquí:"Servicio Red Hat OpenShift en AWS (documentación de AWS)" . "Servicio Red Hat OpenShift en AWS (documentación de Red Hat)" .

Sistemas de almacenamiento NetApp

ONTAP de NetApp

NetApp ONTAP es una poderosa herramienta de software de almacenamiento con capacidades como una GUI intuitiva, API REST con integración de automatización, análisis predictivo basado en IA y acciones correctivas, actualizaciones de hardware sin interrupciones e importación entre almacenamientos.

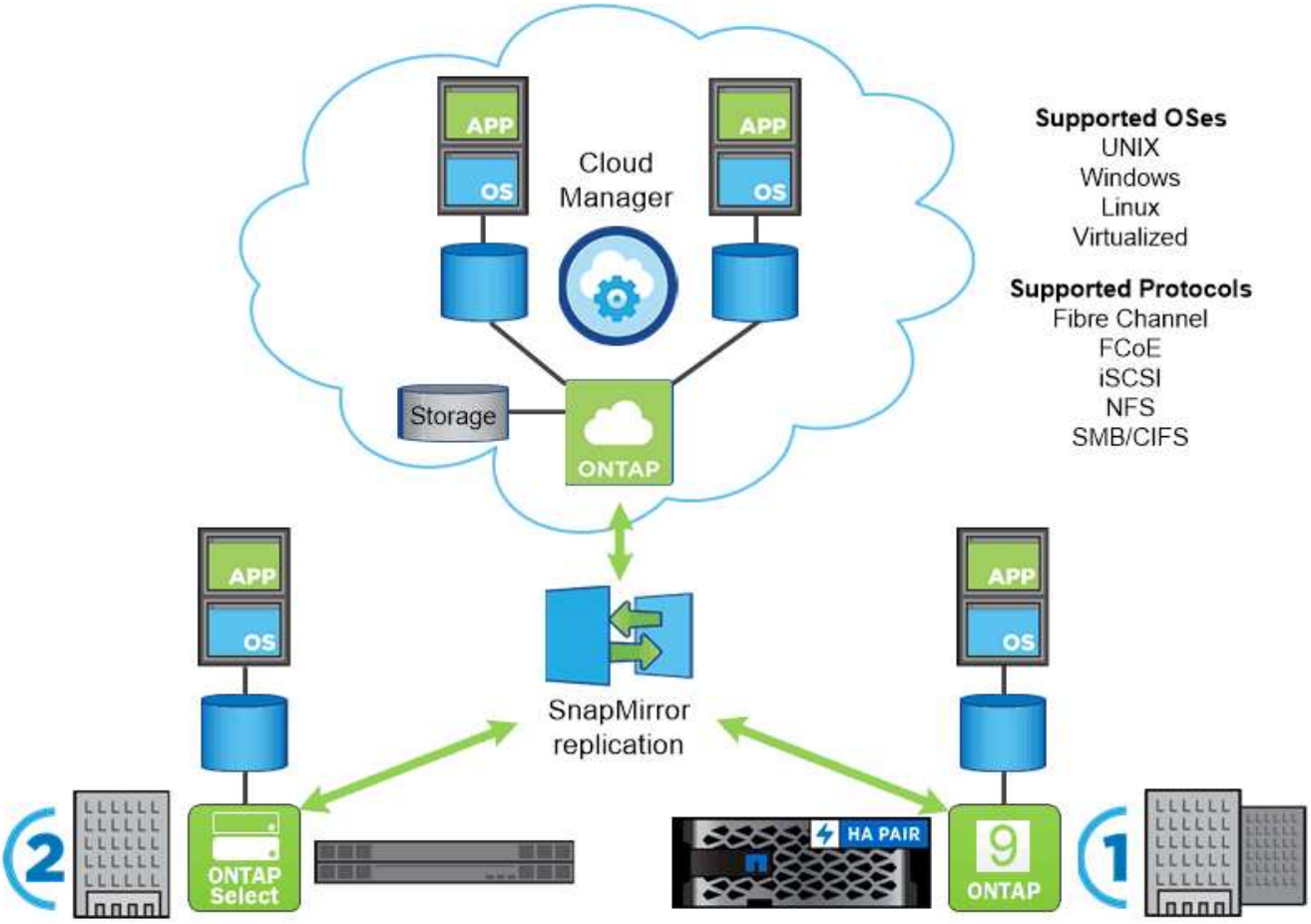
Para obtener más información sobre el sistema de almacenamiento NetApp ONTAP , visite el sitio web "Sitio web de NetApp ONTAP" .

ONTAP ofrece las siguientes características:

- Un sistema de almacenamiento unificado con acceso simultáneo a datos y gestión de protocolos NFS, CIFS, iSCSI, FC, FCoE y FC-NVMe.
- Los diferentes modelos de implementación incluyen configuraciones de hardware locales totalmente flash, híbridas y totalmente HDD; plataformas de almacenamiento basadas en máquinas virtuales en un hipervisor compatible como ONTAP Select; y en la nube como Cloud Volumes ONTAP.
- Mayor eficiencia en el almacenamiento de datos en sistemas ONTAP con soporte para niveles automáticos de datos, compresión de datos en línea, deduplicación y compactación.
- Almacenamiento controlado por QoS y basado en carga de trabajo.
- Integración perfecta con una nube pública para la clasificación y protección de datos. ONTAP también ofrece sólidas capacidades de protección de datos que lo distinguen en cualquier entorno:
  - **Copias instantáneas de NetApp .** Una copia de seguridad de datos rápida y en un momento determinado utilizando una cantidad mínima de espacio en disco y sin sobrecarga de rendimiento adicional.
  - * NetApp SnapMirror.* Duplica las copias instantáneas de datos de un sistema de almacenamiento a otro. ONTAP también permite reflejar datos en otras plataformas físicas y servicios nativos de la nube.
  - * NetApp SnapLock.* Administración eficiente de datos no regrabables escribiéndolos en volúmenes especiales que no se pueden sobrescribir ni borrar durante un período designado.
  - * NetApp SnapVault.* Realiza copias de seguridad de datos de varios sistemas de almacenamiento en una copia instantánea central que sirve como copia de seguridad para todos los sistemas designados.
  - * NetApp SyncMirror.* Proporciona duplicación de datos a nivel RAID en tiempo real en dos complejos diferentes de discos que están conectados físicamente al mismo controlador.
  - * NetApp SnapRestore.* Proporciona una restauración rápida de datos respaldados a pedido a partir de copias instantáneas.
  - * NetApp FlexClone.* Proporciona aprovisionamiento instantáneo de una copia totalmente legible y escribible de un volumen NetApp basado en una copia Snapshot.

Para obtener más información sobre ONTAP, consulte el "Centro de documentación de ONTAP 9" .

NetApp ONTAP está disponible en instalaciones locales, virtualizadas o en la nube.



Plataformas NetApp

NetApp AFF/ FAS

NetApp ofrece plataformas de almacenamiento robustas, all-flash (AFF) e híbridas de escalamiento horizontal (FAS), diseñadas a medida con rendimiento de baja latencia, protección de datos integrada y compatibilidad con múltiples protocolos.

Ambos sistemas funcionan con el software de gestión de datos NetApp ONTAP , el software de gestión de datos más avanzado de la industria para una gestión de almacenamiento simplificada, altamente disponible e integrada en la nube para brindar la velocidad, la eficiencia y la seguridad de clase empresarial que su estructura de datos necesita.

Para obtener más información sobre las plataformas NETAPP AFF/ FAS , haga clic en "aquí" .

ONTAP Select

ONTAP Select es una implementación definida por software de NetApp ONTAP que se puede implementar en un hipervisor en su entorno. Se puede instalar en VMware vSphere o en KVM y proporciona toda la funcionalidad y experiencia de un sistema ONTAP basado en hardware.

Para obtener más información sobre ONTAP Select, haga clic en "aquí" .

Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP es una versión implementada en la nube de NetApp ONTAP disponible para implementarse en varias nubes públicas, incluidas: Amazon AWS, Microsoft Azure y Google Cloud.

Para obtener más información sobre Cloud Volumes ONTAP, haga clic en "aquí" .

Amazon FSx ONTAP

Amazon FSx ONTAP proporciona almacenamiento compartido totalmente administrado en la nube de AWS con las populares capacidades de acceso y administración de datos de ONTAP. Para obtener más información sobre Amazon FSx ONTAP, haga clic en "aquí" .

Azure NetApp Files

Azure NetApp Files es un servicio de almacenamiento de archivos nativo de Azure, de primera mano, de clase empresarial y de alto rendimiento. Proporciona volúmenes como un servicio para el cual puede crear cuentas de NetApp , grupos de capacidad y volúmenes. También puede seleccionar niveles de servicio y rendimiento y administrar la protección de datos. Puede crear y administrar recursos compartidos de archivos escalables, de alto rendimiento y altamente disponibles mediante el uso de los mismos protocolos y herramientas con los que está familiarizado y en los que confía localmente. Para obtener más información sobre Azure NetApp Files, haga clic en "aquí" .

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes es un servicio de almacenamiento de datos basado en la nube totalmente administrado que brinda capacidades avanzadas de administración de datos y un rendimiento altamente escalable. Le permite trasladar aplicaciones basadas en archivos a Google Cloud. Tiene soporte integrado para los protocolos Network File System (NFSv3 y NFSv4.1) y Server Message Block (SMB), por lo que no necesita rediseñar sus aplicaciones y puede continuar obteniendo almacenamiento persistente para sus aplicaciones. Para obtener más información sobre Google Cloud NetApp VolumesP, haga clic en "aquí" .

NetApp Element: Red Hat OpenShift con NetApp

El software NetApp Element ofrece un rendimiento modular y escalable, y cada nodo de almacenamiento proporciona capacidad y rendimiento garantizados al entorno. Los sistemas NetApp Element pueden escalar de 4 a 100 nodos en un solo clúster y ofrecen una serie de funciones avanzadas de administración de almacenamiento.

Para obtener más información sobre los sistemas de almacenamiento NetApp Element , visite el sitio web "Sitio web de NetApp Solidfire" .

Redirección de inicio de sesión iSCSI y capacidades de autocuración

El software NetApp Element aprovecha el protocolo de almacenamiento iSCSI, una forma estándar de encapsular comandos SCSI en una red TCP/IP tradicional. Cuando los estándares SCSI cambian o cuando mejora el rendimiento de las redes Ethernet, el protocolo de almacenamiento iSCSI se beneficia sin necesidad de realizar ningún cambio.

Aunque todos los nodos de almacenamiento tienen una IP de administración y una IP de almacenamiento, el software NetApp Element anuncia una única dirección IP virtual de almacenamiento (dirección SVIP) para todo el tráfico de almacenamiento en el clúster. Como parte del proceso de inicio de sesión de iSCSI, el almacenamiento puede responder que el volumen de destino se ha movido a una dirección diferente y, por lo tanto, no puede continuar con el proceso de negociación. Luego, el host vuelve a emitir la solicitud de inicio de sesión a la nueva dirección en un proceso que no requiere reconfiguración del lado del host. Este proceso se conoce como redirección de inicio de sesión iSCSI.

La redirección de inicio de sesión iSCSI es una parte clave del clúster de software NetApp Element . Cuando se recibe una solicitud de inicio de sesión de host, el nodo decide qué miembro del clúster debe manejar el tráfico en función de las IOPS y los requisitos de capacidad del volumen. Los volúmenes se distribuyen en el clúster de software NetApp Element y se redistribuyen si un solo nodo maneja demasiado tráfico para sus volúmenes o si se agrega un nuevo nodo. Se asignan varias copias de un volumen determinado en toda la matriz.

De esta manera, si una falla de nodo es seguida por una redistribución de volumen, no hay ningún efecto en la conectividad del host más allá de un cierre de sesión e inicio de sesión con redirección a la nueva ubicación. Con la redirección de inicio de sesión iSCSI, un clúster de software NetApp Element es una arquitectura de escalamiento horizontal y autorreparación capaz de realizar actualizaciones y operaciones sin interrupciones.

Calidad de servicio del clúster de software NetApp Element

Un clúster de software NetApp Element permite configurar QoS de forma dinámica por volumen. Puede utilizar configuraciones de QoS por volumen para controlar el rendimiento del almacenamiento en función de los SLA que defina. Los siguientes tres parámetros configurables definen la QoS:

- **IOPS mínimos.** La cantidad mínima de IOPS sostenidas que el clúster de software NetApp Element proporciona a un volumen. La IOPS mínima configurada para un volumen es el nivel de rendimiento garantizado para un volumen. El rendimiento por volumen no cae por debajo de este nivel.
- **IOPS máximos.** La cantidad máxima de IOPS sostenidas que el clúster de software NetApp Element proporciona a un volumen en particular.
- **IOPS ráfaga.** La cantidad máxima de IOPS permitida en un escenario de ráfaga corta. La duración de la ráfaga se puede configurar, con un valor predeterminado de 1 minuto. Si un volumen ha estado funcionando por debajo del nivel máximo de IOPS, se acumulan créditos de ráfaga. Cuando los niveles de rendimiento se vuelven muy altos y se fuerzan, se permiten ráfagas cortas de IOPS más allá del IOPS máximo en el volumen.

Multitenencia

La multitenencia segura se consigue con las siguientes características:

- **Autenticación segura.** El protocolo de autenticación por desafío mutuo (CHAP) se utiliza para el acceso seguro al volumen. El Protocolo ligero de acceso a directorios (LDAP) se utiliza para el acceso seguro al clúster para administración y generación de informes.
- **Grupos de acceso a volumen (VAG).** De manera opcional, se pueden utilizar VAG en lugar de la autenticación, asignando cualquier número de nombres calificados iSCSI (IQN) específicos del iniciador iSCSI a uno o más volúmenes. Para acceder a un volumen en un VAG, el IQN del iniciador debe estar en

la lista de IQN permitidos para el grupo de volúmenes.

- **LAN virtuales de inquilino (VLAN).** A nivel de red, la seguridad de red de extremo a extremo entre los iniciadores iSCSI y el clúster de software NetApp Element se facilita mediante el uso de VLAN. Para cualquier VLAN que se crea para aislar una carga de trabajo o un inquilino, NetApp Element Software crea una dirección SVIP de destino iSCSI independiente a la que solo se puede acceder a través de la VLAN específica.
- **VLAN habilitadas para VRF.** Para respaldar aún más la seguridad y la escalabilidad en el centro de datos, el software NetApp Element le permite habilitar cualquier VLAN de inquilino para una funcionalidad similar a VRF. Esta característica agrega estas dos capacidades clave:
  - **Enrutamiento L3 a una dirección SVIP de inquilino.** Esta función le permite ubicar los iniciadores iSCSI en una red o VLAN separada de la del clúster de software NetApp Element .
  - **Subredes IP superpuestas o duplicadas.** Esta función le permite agregar una plantilla a los entornos de inquilinos, lo que permite que a cada VLAN de inquilino respectivo se le asignen direcciones IP de la misma subred IP. Esta capacidad puede ser útil para entornos de proveedores de servicios donde la escala y la preservación del espacio IP son importantes.

Eficiencias del almacenamiento empresarial

El clúster de software NetApp Element aumenta la eficiencia y el rendimiento general del almacenamiento. Las siguientes funciones se realizan en línea, están siempre activas y no requieren configuración manual por parte del usuario:

- **Desduplicación.** El sistema solo almacena bloques 4K únicos. Cualquier bloque 4K duplicado se asocia automáticamente a una versión ya almacenada de los datos. Los datos se encuentran en unidades de bloque y se reflejan mediante el software de protección de datos Helix de NetApp Element . Este sistema reduce significativamente el consumo de capacidad y las operaciones de escritura dentro del sistema.
- **Compresión.** La compresión se realiza en línea antes de que los datos se escriban en la NVRAM. Los datos se comprimen, se almacenan en bloques de 4K y permanecen comprimidos en el sistema. Esta compresión reduce significativamente el consumo de capacidad, las operaciones de escritura y el consumo de ancho de banda en todo el clúster.
- **Aprovisionamiento fino.** Esta capacidad proporciona la cantidad adecuada de almacenamiento en el momento que lo necesita, eliminando el consumo de capacidad causado por volúmenes sobreaprovisionados o subutilizados.
- **Hélice.** Los metadatos de un volumen individual se almacenan en una unidad de metadatos y se replican en una unidad de metadatos secundaria para redundancia.

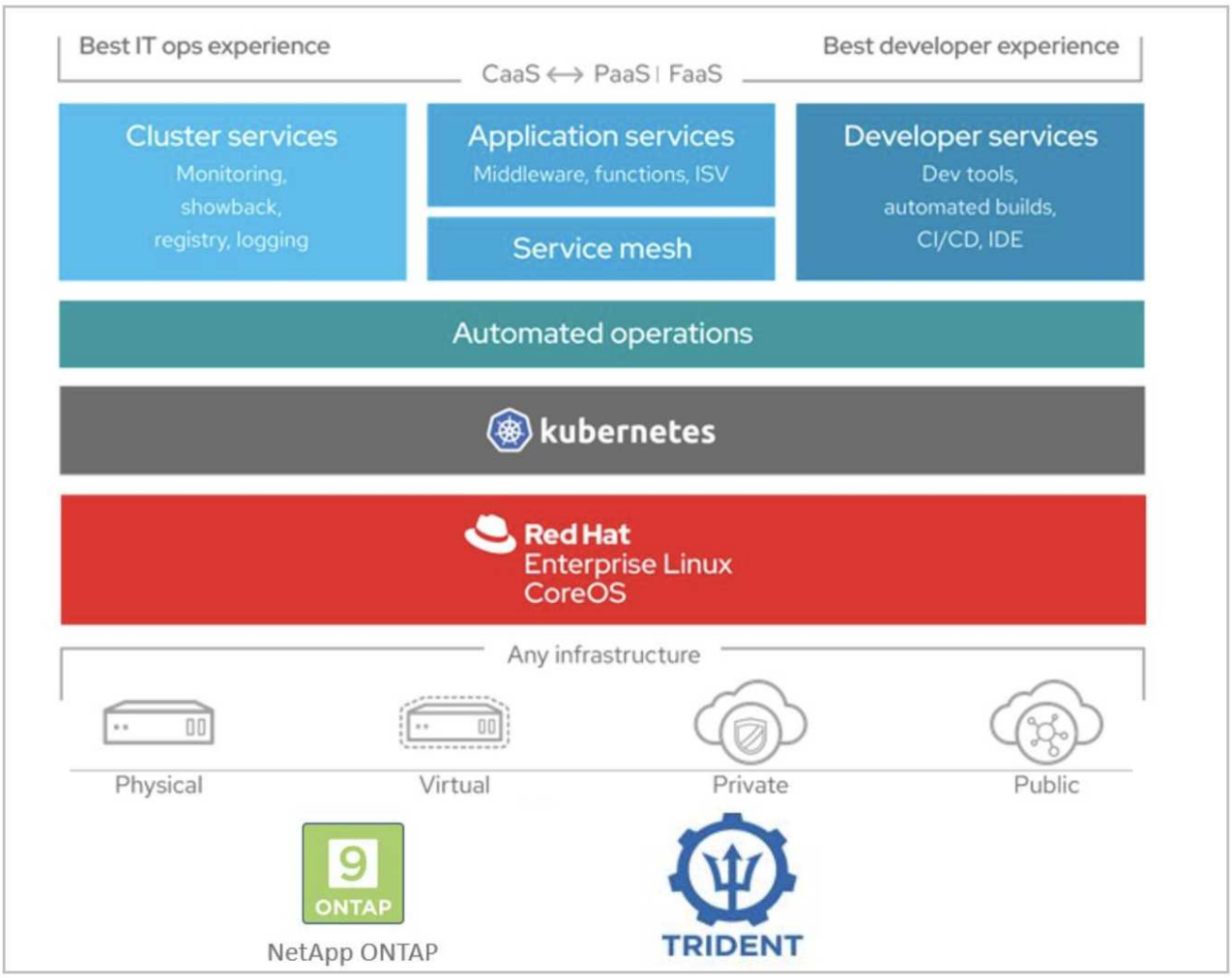
> El elemento fue diseñado para la automatización. Todas las funciones de almacenamiento están disponibles a través de API. Estas API son el único método que utiliza la interfaz de usuario para controlar el sistema.

Integraciones de almacenamiento de NetApp

Obtenga más información sobre la integración de NetApp Trident con Red Hat OpenShift

Obtenga información sobre NetApp Trident Protect, que ha sido validado para la administración de aplicaciones y almacenamiento persistente para la solución de virtualización OpenShift.

Trident, un orquestador y aprovisionador de almacenamiento de código abierto mantenido por NetApp y NetApp Trident Protect lo ayuda a orquestar y administrar datos persistentes en entornos basados en contenedores, como Red Hat OpenShift.



Las siguientes páginas contienen información adicional sobre los productos NetApp que han sido validados para la gestión de aplicaciones y almacenamiento persistente en la solución Red Hat OpenShift con NetApp :

- "Documentación de Trident"
- "Documentación de protección de Trident"

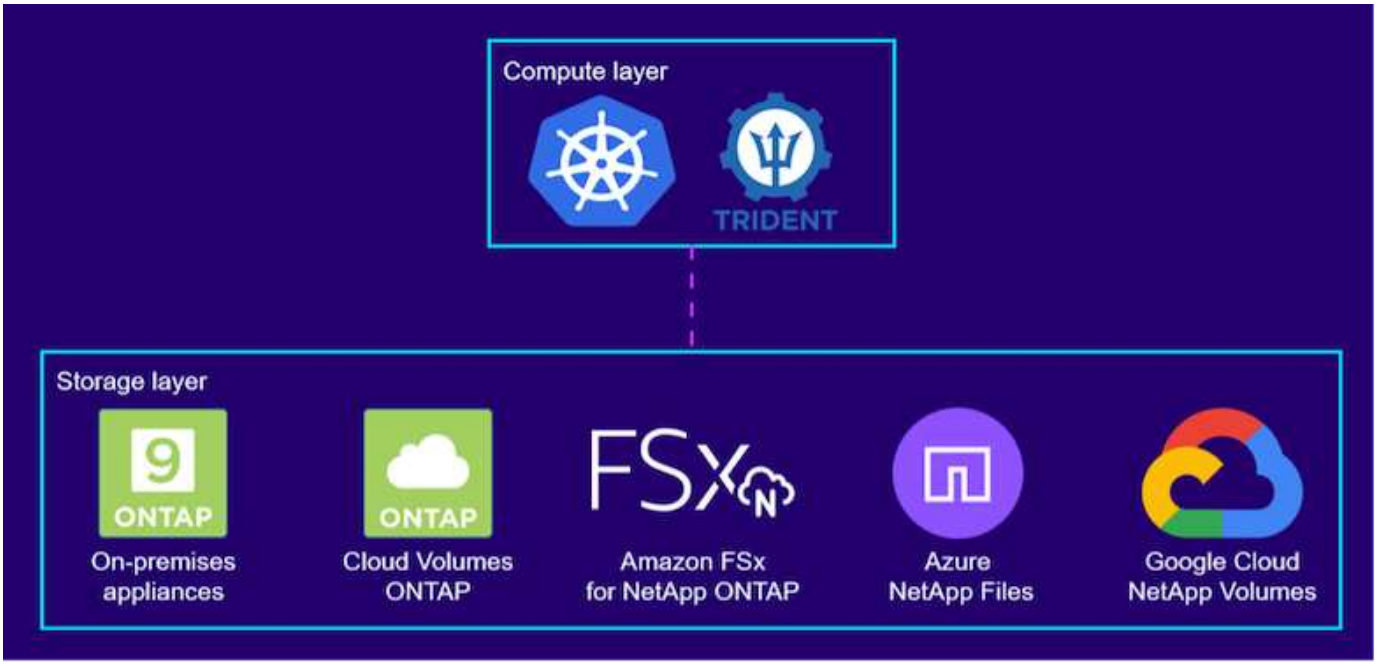
Trident de NetApp

Descripción general de Trident

Trident es un orquestador de almacenamiento de código abierto y totalmente compatible con contenedores y distribuciones de Kubernetes, incluido Red Hat OpenShift. Trident funciona con todo el portafolio de almacenamiento de NetApp , incluidos los sistemas de almacenamiento NetApp ONTAP y Element, y también admite conexiones NFS e iSCSI. Trident acelera el flujo de trabajo de DevOps al permitir que los usuarios finales aprovisionen y administren almacenamiento desde sus sistemas de almacenamiento

NetApp sin necesidad de la intervención de un administrador de almacenamiento.

Un administrador puede configurar una serie de backends de almacenamiento según las necesidades del proyecto y los modelos del sistema de almacenamiento que habilitan funciones de almacenamiento avanzadas, incluida la compresión, tipos de discos específicos o niveles de QoS que garantizan un cierto nivel de rendimiento. Una vez definidos, estos backends pueden ser utilizados por los desarrolladores en sus proyectos para crear reclamos de volumen persistentes (PVC) y para adjuntar almacenamiento persistente a sus contenedores a pedido.



Trident tiene un ciclo de desarrollo rápido y, al igual que Kubernetes, se lanza cuatro veces al año.

Se puede encontrar una matriz de soporte para qué versión de Trident se ha probado con qué distribución de Kubernetes "aquí" .

Por favor consulte la"Documentación del producto Trident" Para detalles de instalación y configuración.

Descargar Trident

Para instalar Trident en el clúster de usuarios implementado y aprovisionar un volumen persistente, complete los siguientes pasos:

1. Descargue el archivo de instalación en la estación de trabajo del administrador y extraiga el contenido. La versión actual de Trident se puede descargar "aquí" .
2. Extraiga la instalación de Trident del paquete descargado.

```
[netapp-user@rhel7 ~]$ tar -xzf trident-installer-22.01.0.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$
```

Instalar el operador Trident con Helm

1. Primero, establezca la ubicación del clúster de usuarios. `kubeconfig` archivo como una variable de entorno para que no tenga que hacer referencia a él, porque Trident no tiene ninguna opción para pasar este archivo.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-
install/auth/kubeconfig
```

2. Ejecute el comando Helm para instalar el operador Trident desde el archivo tar en el directorio helm mientras crea el espacio de nombres trident en su clúster de usuarios.

```
[netapp-user@rhel7 trident-installer]$ helm install trident
helm/trident-operator-22.01.0.tgz --create-namespace --namespace trident
NAME: trident
LAST DEPLOYED: Fri May  7 12:54:25 2021
NAMESPACE: trident
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing trident-operator, which will deploy and manage
NetApp's Trident CSI
storage provisioner for Kubernetes.

Your release is named 'trident' and is installed into the 'trident'
namespace.
Please note that there must be only one instance of Trident (and
trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy
of tridentctl, which is
available in pre-packaged Trident releases.  You may find all Trident
releases and source code
online at https://github.com/NetApp/trident.

To learn more about the release, try:

  $ helm status trident
  $ helm get all trident
```

3. Puede verificar que Trident se haya instalado correctamente verificando los pods que se ejecutan en el espacio de nombres o utilizando el binario tridentctl para verificar la versión instalada.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-5z45l                   1/2     Running   2          30s
trident-csi-696b685cf8-htdb2        6/6     Running   0          30s
trident-csi-b74p2                   2/2     Running   0          30s
trident-csi-lrw4n                   2/2     Running   0          30s
trident-operator-7c748d957-gr2gw    1/1     Running   0          36s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0        | 22.01.0        |
+----------------+----------------+
```

En algunos casos, los entornos de los clientes pueden requerir la personalización de la implementación de Trident . En estos casos, también es posible instalar manualmente el operador Trident y actualizar los manifiestos incluidos para personalizar la implementación.

Instalar manualmente el operador Trident

1. Primero, establezca la ubicación del clúster de usuarios. `kubeconfig` archivo como una variable de entorno para que no tenga que hacer referencia a él, porque Trident no tiene ninguna opción para pasar este archivo.

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/ocp-
install/auth/kubeconfig
```

2. El `trident-installer` El directorio contiene manifiestos para definir todos los recursos necesarios. Utilizando los manifiestos apropiados, cree el `TridentOrchestrator` definición de recurso personalizado.

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.tride
nt.netapp.io created
```

3. Si no existe ninguno, cree un espacio de nombres Trident en su clúster utilizando el manifiesto proporcionado.

```
[netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
namespace/trident created
```

4. Cree los recursos necesarios para la implementación del operador Trident , como un `ServiceAccount` Para el operador, un `ClusterRole` y `ClusterRoleBinding` hacia `ServiceAccount` , un dedicado `PodSecurityPolicy` , o el propio operador.

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

5. Puede comprobar el estado del operador después de su implementación con los siguientes comandos:

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
trident-operator   1/1     1            1           23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-operator-66f48895cc-lzczk   1/1     Running   0          41s
```

6. Con el operador desplegado, ahora podemos usarlo para instalar Trident. Esto requiere crear una `TridentOrchestrator` .

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:         trident
Namespace:
Labels:       <none>
Annotations:  <none>
API Version:  trident.netapp.io/v1
Kind:         TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-05-07T17:00:28Z
  Generation:          1
  Managed Fields:
    API Version:  trident.netapp.io/v1
    Fields Type:  FieldsV1
    fieldsV1:
      f:spec:
        .:
        f:debug:
        f:namespace:
```

```
    Manager:      kubectl-create
    Operation:    Update
    Time:         2021-05-07T17:00:28Z
    API Version:  trident.netapp.io/v1
    Fields Type:  FieldsV1
    fieldsV1:
      f:status:
        .:
        f:currentInstallationParams:
          .:
          f:IPv6:
          f:autosupportHostname:
          f:autosupportimage:
          f:autosupportProxy:
          f:autosupportSerialNumber:
          f:debug:
          f:enableNodePrep:
          f:imagePullSecrets:
          f:imageRegistry:
          f:k8sTimeout:
          f:kubeletDir:
          f:logFormat:
          f:silenceAutosupport:
          f:tridentimage:
        f:message:
        f:namespace:
        f:status:
        f:version:
    Manager:         trident-operator
    Operation:       Update
    Time:            2021-05-07T17:00:28Z
  Resource Version:  931421
  Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
  UID:               8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:      true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:                       false
    Autosupport Hostname:
    Autosupport image:          netapp/trident-autosupport:21.01
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:                      true
```

```
    Enable Node Prep:           false
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:           30
    Kubelet Dir:          /var/lib/kubelet
    Log Format:           text
    Silence Autosupport:  false
    Trident image:        netapp/trident:22.01.0
  Message:                Trident installed
  Namespace:              trident
  Status:                 Installed
  Version:                v22.01.0
Events:
  Type    Reason      Age   From                           Message
  ----    ------      ----  ----                           -------
  Normal  Installing  80s   trident-operator.netapp.io  Installing
Trident
  Normal  Installed   68s   trident-operator.netapp.io  Trident
installed
```

7. Puede verificar que Trident se haya instalado correctamente verificando los pods que se ejecutan en el espacio de nombres o utilizando el binario tridentctl para verificar la versión instalada.

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h         6/6     Running   0          82s
trident-csi-gn59q                   2/2     Running   0          82s
trident-csi-m4szj                   2/2     Running   0          82s
trident-csi-sb9k9                   2/2     Running   0          82s
trident-operator-66f48895cc-lzczk   1/1     Running   0          2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+----------------+----------------+
| SERVER VERSION | CLIENT VERSION |
+----------------+----------------+
| 22.01.0        | 22.01.0        |
+----------------+----------------+
```

Preparar los nodos de trabajo para el almacenamiento

Sistema Nacional de Archivos

La mayoría de las distribuciones de Kubernetes vienen con los paquetes y utilidades para montar backends NFS instalados de forma predeterminada, incluido Red Hat OpenShift.

Sin embargo, para NFSv3, no existe ningún mecanismo para negociar la concurrencia entre el cliente y el

servidor. Por lo tanto, el número máximo de entradas de la tabla de ranuras sunrpc del lado del cliente se debe sincronizar manualmente con el valor admitido en el servidor para garantizar el mejor rendimiento de la conexión NFS sin que el servidor tenga que disminuir el tamaño de la ventana de la conexión.

Para ONTAP, la cantidad máxima admitida de entradas en la tabla de ranuras sunrpc es 128, es decir, ONTAP puede atender 128 solicitudes NFS simultáneas a la vez. Sin embargo, de forma predeterminada, Red Hat CoreOS/Red Hat Enterprise Linux tiene un máximo de 65 536 entradas en la tabla de ranuras sunrpc por conexión. Necesitamos establecer este valor en 128 y esto se puede hacer usando el Operador de configuración de máquina (MCO) en OpenShift.

Para modificar las entradas de la tabla de ranuras sunrpc máximas en los nodos de trabajo de OpenShift, complete los siguientes pasos:

1. Inicie sesión en la consola web de OCP y navegue a Computación > Configuraciones de la máquina. Haga clic en Crear configuración de máquina. Copie y pegue el archivo YAML y haga clic en Crear.

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 98-worker-nfs-rpc-slot-tables
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-
8;base64,b3B0aW9ucyBzdW5ycGMgdGNwX21heF9zbG90X3RhYmxlX2VudHJpZXM9MTI4Cg=
=
          filesystem: root
          mode: 420
          path: /etc/modprobe.d/sunrpc.conf
```

2. Una vez creado el MCO, la configuración debe aplicarse en todos los nodos de trabajo y reiniciarse uno por uno. Todo el proceso tarda aproximadamente entre 20 y 30 minutos. Verifique si la configuración de la máquina se aplica mediante `oc get mcp` y asegúrese de que el grupo de configuración de la máquina para los trabajadores esté actualizado.

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                             UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False
```

iSCSI

Para preparar los nodos de trabajo para permitir el mapeo de volúmenes de almacenamiento en bloque a través del protocolo iSCSI, debe instalar los paquetes necesarios para soportar esa funcionalidad.

En Red Hat OpenShift, esto se gestiona aplicando un MCO (Operador de configuración de máquina) a su clúster después de su implementación.

Para configurar los nodos de trabajo para ejecutar servicios iSCSI, complete los siguientes pasos:

1. Inicie sesión en la consola web de OCP y navegue a Computación > Configuraciones de la máquina. Haga clic en Crear configuración de máquina. Copie y pegue el archivo YAML y haga clic en Crear.

   Cuando no se utiliza multitrayecto:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-element-iscsi
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
  osImageURL: ""
```

   Al utilizar rutas múltiples:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-
8;base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgbm8KICAgICA
gICBmaW5kX211bHRpcGF0aHMgbm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICA
gIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSIKfQoKYmxhY2tsaXN0IHsKfQoK
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
        - name: multipathd.service
          enabled: true
          state: started
  osImageURL: ""
```

2. Una vez creada la configuración, toma aproximadamente entre 20 y 30 minutos aplicarla a los nodos de trabajo y recargarlos. Verifique si la configuración de la máquina se aplica mediante `oc get mcp` y asegúrese de que el grupo de configuración de la máquina para los trabajadores esté actualizado. También puede iniciar sesión en los nodos de trabajo para confirmar que el servicio iscsid se está ejecutando (y el servicio multipathd se está ejecutando si se utilizan rutas múltiples).

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME     CONFIG                                             UPDATED   UPDATING
DEGRADED
master   rendered-master-a520ae930e1d135e0dee7168   True      False
False
worker   rendered-worker-de321b36eeba62df41feb7bc   True      False
False

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
vendor preset: disabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
 Main PID: 1242 (iscsid)
   Status: "Ready to process requests"
    Tasks: 1
   Memory: 4.9M
      CPU: 9ms
   CGroup: /system.slice/iscsid.service
           └─1242 /usr/sbin/iscsid -f

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status multipathd
 ● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled;
vendor preset: enabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
  Main PID: 918 (multipathd)
    Status: "up"
    Tasks: 7
    Memory: 13.7M
    CPU: 57ms
    CGroup: /system.slice/multipathd.service
            └─918 /sbin/multipathd -d -s
```

También es posible confirmar que MachineConfig se ha aplicado correctamente y que los servicios se han iniciado como se esperaba ejecutando el `oc debug` comando con las banderas apropiadas.

Crear backends del sistema de almacenamiento

Después de completar la instalación de Trident Operator, debe configurar el backend para la plataforma de

almacenamiento NetApp específica que esté utilizando. Siga los enlaces a continuación para continuar con la instalación y configuración de Trident.

- "NFS de NetApp ONTAP"
- "iSCSI de NetApp ONTAP"
- "NetApp Element"

Configuración de NFS de NetApp ONTAP

Para habilitar la integración de Trident con el sistema de almacenamiento NetApp ONTAP , debe crear un backend que permita la comunicación con el sistema de almacenamiento.

1. Hay archivos de backend de muestra disponibles en el archivo de instalación descargado en `sample-input` jerarquía de carpetas. Para los sistemas NetApp ONTAP que prestan servicio a NFS, copie el archivo `backend-ontap-nas.json` archivo a su directorio de trabajo y edite el archivo.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-nas/backend-ontap-nas.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
```

2. Edite los valores backendName, managementLIF, dataLIF, svm, username y password en este archivo.

```
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "ontap-nas+10.61.181.221",
    "managementLIF": "172.21.224.201",
    "dataLIF": "10.61.181.221",
    "svm": "trident_svm",
    "username": "cluster-admin",
    "password": "password"
}
```

> Se recomienda definir el valor backendName personalizado como una combinación de storageDriverName y dataLIF que sirve NFS para una fácil identificación.

3. Con este archivo backend en su lugar, ejecute el siguiente comando para crear su primer backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| ontap-nas+10.61.181.221 | ontap-nas      | be7a619d-c81d-445c-b80c-
5c87a73c5b1e | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. Con el backend creado, a continuación debes crear una clase de almacenamiento. Al igual que con el backend, hay un archivo de clase de almacenamiento de muestra que se puede editar para el entorno disponible en la carpeta sample-inputs. Cópielo en el directorio de trabajo y realice las modificaciones necesarias para reflejar el backend creado.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. La única edición que se debe realizar en este archivo es definir el `backendType` valor al nombre del controlador de almacenamiento desde el backend recién creado. Tenga en cuenta también el valor del campo de nombre, al que se debe hacer referencia en un paso posterior.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
```

> Hay un campo opcional llamado `fsType` que se define en este archivo. Esta línea se puede eliminar en los backends de NFS.

6. Ejecutar el `oc` Comando para crear la clase de almacenamiento.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. Una vez creada la clase de almacenamiento, deberá crear la primera reclamación de volumen persistente (PVC). Hay una muestra `pvc-basic.yaml` archivo que se puede utilizar para realizar esta acción y que también se encuentra en sample-inputs.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. La única edición que se debe realizar en este archivo es asegurarse de que `storageClassName` El campo coincide con el recién creado. La definición de PVC se puede personalizar aún más según lo requiera la carga de trabajo que se aprovisionará.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. Cree el PVC emitiendo el `oc` dominio. La creación puede tardar algún tiempo dependiendo del tamaño del volumen de respaldo que se esté creando, por lo que puedes observar el proceso a medida que se completa.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d   1Gi
RWO            basic-csi      7s
```

Configuración de iSCSI de NetApp ONTAP

Para habilitar la integración de Trident con el sistema de almacenamiento NetApp ONTAP , debe crear un backend que permita la comunicación con el sistema de almacenamiento.

1. Hay archivos de backend de muestra disponibles en el archivo de instalación descargado en `sample-`

`input` jerarquía de carpetas. Para los sistemas NetApp ONTAP que prestan servicio a iSCSI, copie el archivo `backend-ontap-san.json` archivo a su directorio de trabajo y edite el archivo.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-san/backend-ontap-san.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-san.json
```

2. Edite los valores managementLIF, dataLIF, svm, nombre de usuario y contraseña en este archivo.

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.240",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password"
}
```

3. Con este archivo backend en su lugar, ejecute el siguiente comando para crear su primer backend.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797-
fb9bb3322b91 | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

4. Con el backend creado, a continuación debes crear una clase de almacenamiento. Al igual que con el backend, hay un archivo de clase de almacenamiento de muestra que se puede editar para el entorno disponible en la carpeta sample-inputs. Cópielo en el directorio de trabajo y realice las modificaciones necesarias para reflejar el backend creado.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. La única edición que se debe realizar en este archivo es definir el `backendType` valor al nombre del controlador de almacenamiento desde el backend recién creado. Tenga en cuenta también el valor del campo de nombre, al que se debe hacer referencia en un paso posterior.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"
```

> Hay un campo opcional llamado `fsType` que se define en este archivo. En los backends iSCSI, este valor se puede configurar para un tipo de sistema de archivos Linux específico (XFS, ext4, etc.) o se puede eliminar para permitir que OpenShift decida qué sistema de archivos utilizar.

6. Ejecutar el `oc` Comando para crear la clase de almacenamiento.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. Una vez creada la clase de almacenamiento, deberá crear la primera reclamación de volumen persistente (PVC). Hay una muestra `pvc-basic.yaml` archivo que se puede utilizar para realizar esta acción y que también se encuentra en sample-inputs.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. La única edición que se debe realizar en este archivo es asegurarse de que `storageClassName` El campo coincide con el recién creado. La definición de PVC se puede personalizar aún más según lo requiera la carga de trabajo que se aprovisionará.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. Cree el PVC emitiendo el `oc` dominio. La creación puede tardar algún tiempo dependiendo del tamaño del volumen de respaldo que se esté creando, por lo que puedes observar el proceso a medida que se completa.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-7ceac1ba-0189-43c7-8f98-094719f7956c   1Gi
RWO            basic-csi      3s
```

Configuración de iSCSI de NetApp Element

Para habilitar la integración de Trident con el sistema de almacenamiento NetApp Element , debe crear un backend que permita la comunicación con el sistema de almacenamiento mediante el protocolo iSCSI.

1. Hay archivos de backend de muestra disponibles en el archivo de instalación descargado en `sample-input` jerarquía de carpetas. Para los sistemas NetApp Element que prestan servicio a iSCSI, copie el archivo `backend-solidfire.json` archivo a su directorio de trabajo y edite el archivo.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/solidfire/backend-solidfire.json ./
[netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
```

   a. Edite el usuario, la contraseña y el valor MVIP en el `EndPoint` línea.

   b. Editar el `SVIP` valor.

```
 {
    "version": 1,
    "storageDriverName": "solidfire-san",
    "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
    "SVIP": "10.61.180.200:3260",
    "TenantName": "trident",
    "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS":
2000, "burstIOPS": 4000}},
              {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS":
6000, "burstIOPS": 8000}},
              {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS":
8000, "burstIOPS": 10000}}]
}
```

2. Con este archivo back-end en su lugar, ejecute el siguiente comando para crear su primer back-end.

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-solidfire.json
+------------------------+----------------
+--------------------------------------+--------+---------+
|          NAME          | STORAGE DRIVER |                 UUID
| STATE  | VOLUMES |
+------------------------+----------------
+--------------------------------------+--------+---------+
| solidfire_10.61.180.200 | solidfire-san  | b90783ee-e0c9-49af-8d26-
3ea87ce2efdf | online |       0 |
+------------------------+----------------
+--------------------------------------+--------+---------+
```

3. Con el backend creado, a continuación debes crear una clase de almacenamiento. Al igual que con el backend, hay un archivo de clase de almacenamiento de muestra que se puede editar para el entorno disponible en la carpeta sample-inputs. Cópielo en el directorio de trabajo y realice las modificaciones necesarias para reflejar el backend creado.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.templ ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

4. La única edición que se debe realizar en este archivo es definir el `backendType` valor al nombre del controlador de almacenamiento desde el backend recién creado. Tenga en cuenta también el valor del campo de nombre, al que se debe hacer referencia en un paso posterior.

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
```

Hay un campo opcional llamado `fsType` que se define en este archivo. En los backends iSCSI, este valor se puede configurar para un tipo de sistema de archivos Linux específico (XFS, ext4, etc.) o se puede eliminar para permitir que OpenShift decida qué sistema de archivos utilizar.

5. Ejecutar el `oc` Comando para crear la clase de almacenamiento.

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

6. Una vez creada la clase de almacenamiento, deberá crear la primera reclamación de volumen persistente (PVC). Hay una muestra `pvc-basic.yaml` archivo que se puede utilizar para realizar esta acción y que también se encuentra en sample-inputs.

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

7. La única edición que se debe realizar en este archivo es asegurarse de que `storageClassName` El campo coincide con el recién creado. La definición de PVC se puede personalizar aún más según lo requiera la carga de trabajo que se aprovisionará.

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

8. Cree el PVC emitiendo el `oc` dominio. La creación puede tardar algún tiempo dependiendo del tamaño del volumen de respaldo que se esté creando, por lo que puedes observar el proceso a medida que se completa.

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME    STATUS   VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS   AGE
basic   Bound    pvc-3445b5cc-df24-453d-a1e6-b484e874349d   1Gi
RWO            basic-csi      5s
```

Opciones de configuración avanzadas

Explorar las opciones del balanceador de carga

Explorando las opciones del balanceador de carga: Red Hat OpenShift con NetApp

En la mayoría de los casos, Red Hat OpenShift pone las aplicaciones a disposición del mundo exterior a través de rutas. Un servicio se expone asignándole un nombre de host accesible externamente. La ruta definida y los puntos finales identificados por su servicio pueden ser consumidos por un enrutador OpenShift para proporcionar esta conectividad denominada a clientes externos.

Sin embargo, en algunos casos, las aplicaciones requieren la implementación y configuración de balanceadores de carga personalizados para exponer los servicios adecuados. Un ejemplo de esto es NetApp Trident Protect. Para satisfacer esta necesidad, hemos evaluado una serie de opciones de equilibrador de carga personalizados. En esta sección se describe su instalación y configuración.

Las siguientes páginas contienen información adicional sobre las opciones de equilibrador de carga validadas en la solución Red Hat OpenShift con NetApp :

- "MetalLB"
- "F5 BIG-IP"

Instalación de balanceadores de carga MetalLB: Red Hat OpenShift con NetApp

Esta página enumera las instrucciones de instalación y configuración del balanceador de carga MetalLB.

MetalLB es un balanceador de carga de red autohospedado instalado en su clúster OpenShift que permite la creación de servicios OpenShift de tipo balanceador de carga en clústeres que no se ejecutan en un proveedor de nube. Las dos características principales de MetalLB que trabajan juntas para soportar los servicios LoadBalancer son la asignación de direcciones y el anuncio externo.

Opciones de configuración de MetalLB

Según cómo MetalLB anuncia la dirección IP asignada a los servicios LoadBalancer fuera del clúster OpenShift, opera en dos modos:

- **Modo de capa 2.** En este modo, un nodo del clúster OpenShift toma posesión del servicio y responde a las solicitudes ARP para esa IP para que sea accesible fuera del clúster OpenShift. Debido a que solo el nodo anuncia la IP, tiene un cuello de botella de ancho de banda y limitaciones de conmutación por error lenta. Para obtener más información, consulte la documentación."aquí" .
- **Modo BGP.** En este modo, todos los nodos del clúster OpenShift establecen sesiones de intercambio de tráfico BGP con un enrutador y anuncian las rutas para reenviar tráfico a las IP del servicio. El requisito previo para esto es integrar MetalLB con un enrutador en esa red. Debido al mecanismo de hashing en BGP, existen ciertas limitaciones cuando cambia el mapeo de IP a nodo para un servicio. Para obtener más información, consulte la documentación."aquí" .

Para los fines de este documento, configuramos MetalLB en modo de capa 2.

Instalación del balanceador de carga MetalLB

1. Descargue los recursos de MetalLB.

```
[netapp-user@rhel7 ~]$ wget
https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/name
space.yaml
[netapp-user@rhel7 ~]$ wget
https://raw.githubusercontent.com/metallb/metallb/v0.10.2/manifests/meta
llb.yaml
```

2. Editar archivo `metallb.yaml` y eliminar `spec.template.spec.securityContext` desde el controlador Deployment y el altavoz DaemonSet.

   Líneas a eliminar:

```
securityContext:
  runAsNonRoot: true
  runAsUser: 65534
```

3. Crea el `metallb-system` espacio de nombres.

```
[netapp-user@rhel7 ~]$ oc create -f namespace.yaml
namespace/metallb-system created
```

4. Crear el CR de MetalLB.

```
[netapp-user@rhel7 ~]$ oc create -f metallb.yaml
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
role.rbac.authorization.k8s.io/controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller
created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker
created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
rolebinding.rbac.authorization.k8s.io/controller created
daemonset.apps/speaker created
deployment.apps/controller created
```

5. Antes de configurar el altavoz MetalLB, otorgue al altavoz DaemonSet privilegios elevados para que pueda realizar la configuración de red necesaria para que los balanceadores de carga funcionen.

```
[netapp-user@rhel7 ~]$ oc adm policy add-scc-to-user privileged -n
metallb-system -z speaker
clusterrole.rbac.authorization.k8s.io/system:openshift:scc:privileged
added: "speaker"
```

6. Configure MetalLB creando un `ConfigMap` en el `metallb-system` espacio de nombres.

```
[netapp-user@rhel7 ~]$ vim metallb-config.yaml

apiVersion: v1
kind: ConfigMap
metadata:
  namespace: metallb-system
  name: config
data:
  config: |
    address-pools:
    - name: default
      protocol: layer2
      addresses:
      - 10.63.17.10-10.63.17.200

[netapp-user@rhel7 ~]$ oc create -f metallb-config.yaml
configmap/config created
```

7. Ahora, cuando se crean servicios de balanceo de carga, MetalLB asigna una IP externa a los servicios y anuncia la dirección IP respondiendo a las solicitudes ARP.

> Si desea configurar MetalLB en modo BGP, omita el paso 6 anterior y siga el procedimiento de la documentación de MetalLB."aquí" .

Instalación de balanceadores de carga F5 BIG-IP

F5 BIG-IP es un controlador de entrega de aplicaciones (ADC) que ofrece un amplio conjunto de servicios avanzados de seguridad y gestión de tráfico de nivel de producción, como equilibrio de carga L4-L7, descarga SSL/TLS, DNS, firewall y muchos más. Estos servicios aumentan drásticamente la disponibilidad, la seguridad y el rendimiento de sus aplicaciones.

F5 BIG-IP se puede implementar y consumir de diversas maneras: en hardware dedicado, en la nube o como un dispositivo virtual en las instalaciones. Consulte la documentación aquí para explorar e implementar F5 BIG-IP según los requisitos.

Para una integración eficiente de los servicios F5 BIG-IP con Red Hat OpenShift, F5 ofrece el Servicio de ingreso de contenedores BIG-IP (CIS). CIS se instala como un módulo de controlador que supervisa la API de OpenShift en busca de determinadas definiciones de recursos personalizados (CRD) y administra la configuración del sistema F5 BIG-IP. F5 BIG-IP CIS se puede configurar para controlar los tipos de servicios LoadBalancers y Rutas en OpenShift.

Además, para la asignación automática de direcciones IP para dar servicio al tipo LoadBalancer, puede utilizar el controlador IPAM F5. El controlador F5 IPAM se instala como un pod de controlador que supervisa la API de OpenShift en busca de servicios LoadBalancer con una anotación ipamLabel para asignar la dirección IP desde un grupo preconfigurado.

Esta página enumera las instrucciones de instalación y configuración del controlador F5 BIG-IP CIS e IPAM.

Como requisito previo, debe tener un sistema F5 BIG-IP implementado y licenciado. También debe tener licencia para servicios SDN, que se incluyen de forma predeterminada con la licencia base de BIG-IP VE.

> F5 BIG-IP se puede implementar en modo independiente o en clúster. Para el propósito de esta validación, F5 BIG-IP se implementó en modo independiente, pero, para fines de producción, se prefiere tener un clúster de BIG-IP para evitar un único punto de falla.

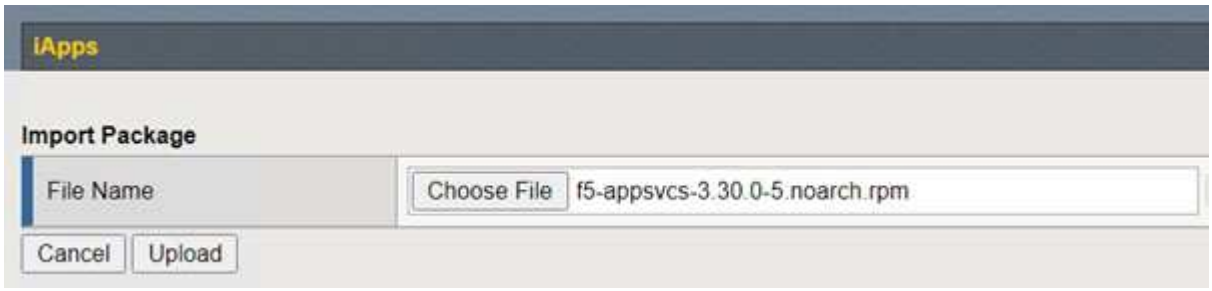
> Un sistema F5 BIG-IP se puede implementar en hardware dedicado, en la nube o como un dispositivo virtual local con versiones superiores a 12.x para que se integre con F5 CIS. Para los fines de este documento, el sistema F5 BIG-IP fue validado como un dispositivo virtual, por ejemplo utilizando la edición BIG-IP VE.

Versiones validadas

Tecnología	Versión del software
Red Hat OpenShift	4.6 EUS, 4.7
Edición F5 BIG-IP VE	16.1.0
Servicio de ingreso de contenedores F5	2.5.1
Controlador F5 IPAM	0.1.4
F5 AS3	3.30.0

Instalación

1. Instale la extensión F5 Application Services 3 para permitir que los sistemas BIG-IP acepten configuraciones en JSON en lugar de comandos imperativos. Ir a "Repositorio de GitHub de F5 AS3" y descargue el último archivo RPM.
2. Inicie sesión en el sistema F5 BIG-IP, navegue a iApps > Administración de paquetes LX y haga clic en Importar.
3. Haga clic en Elegir archivo y seleccione el archivo RPM AS3 descargado, haga clic en Aceptar y luego haga clic en Cargar.

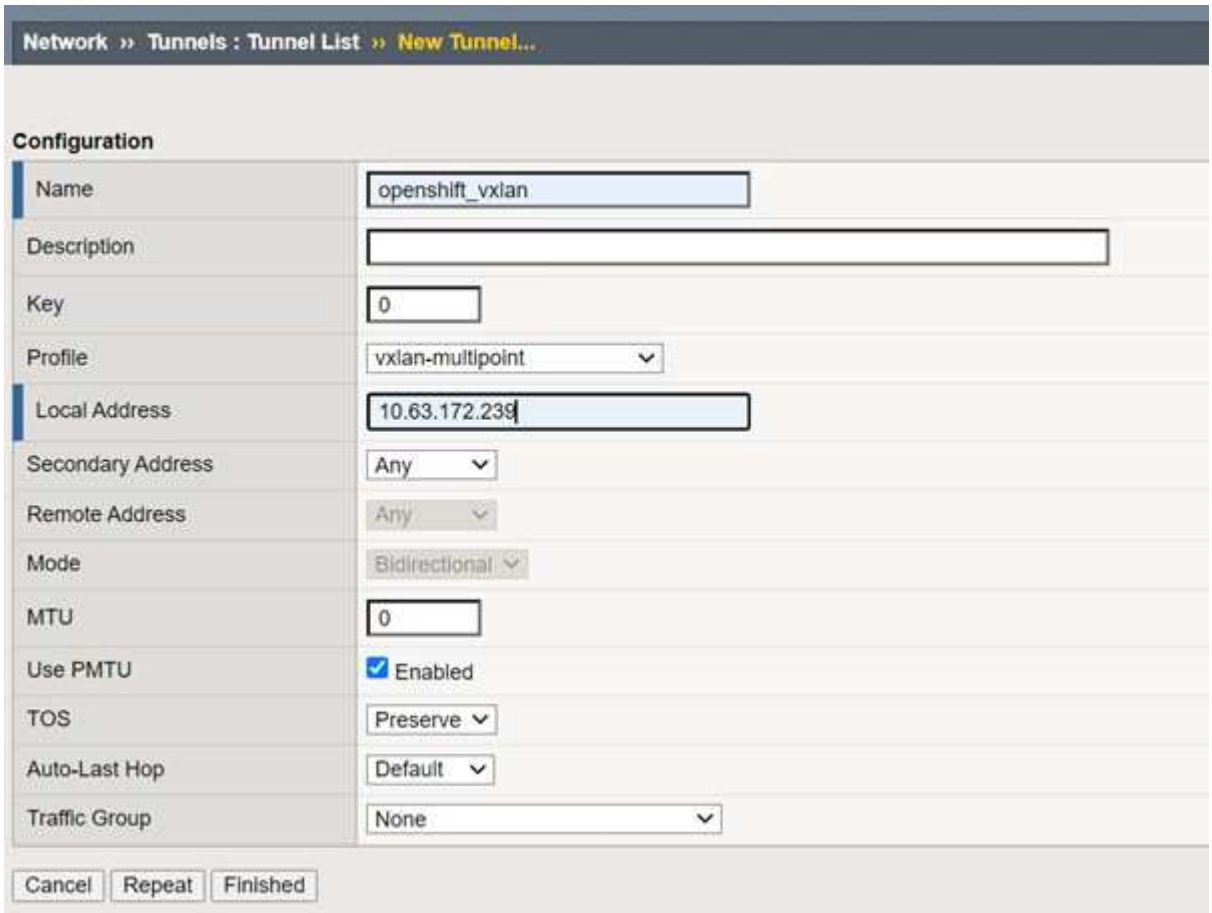


4. Confirme que la extensión AS3 se haya instalado correctamente.

5. A continuación, configure los recursos necesarios para la comunicación entre los sistemas OpenShift y

BIG-IP. Primero, cree un túnel entre OpenShift y el servidor BIG-IP creando una interfaz de túnel VXLAN en el sistema BIG-IP para OpenShift SDN. Vaya a Red > Túneles > Perfiles, haga clic en Crear y configure el Perfil principal en vxlan y el Tipo de inundación en Multidifusión. Ingrese un nombre para el perfil y haga clic en Finalizar.

6. Vaya a Red > Túneles > Lista de túneles, haga clic en Crear e ingrese el nombre y la dirección IP local para el túnel. Seleccione el perfil de túnel que se creó en el paso anterior y haga clic en Finalizado.



7. Inicie sesión en el clúster Red Hat OpenShift con privilegios de administrador del clúster.
8. Cree una subred de host en OpenShift para el servidor F5 BIG-IP, que extienda la subred del clúster OpenShift al servidor F5 BIG-IP. Descargue la definición YAML de la subred del host.

```
wget https://github.com/F5Networks/k8s-bigip-
ctlr/blob/master/docs/config_examples/openshift/f5-kctlr-openshift-
hostsubnet.yaml
```

9. Edite el archivo de subred del host y agregue la IP VTEP (túnel VXLAN) de BIG-IP para OpenShift SDN.

```
apiVersion: v1
kind: HostSubnet
metadata:
  name: f5-server
  annotations:
    pod.network.openshift.io/fixed-vnid-host: "0"
    pod.network.openshift.io/assign-subnet: "true"
# provide a name for the node that will serve as BIG-IP's entry into the
cluster
host: f5-server
# The hostIP address will be the BIG-IP interface address routable to
the
# OpenShift Origin nodes.
# This address is the BIG-IP VTEP in the SDN's VXLAN.
hostIP: 10.63.172.239
```

Cambie la IP del host y otros detalles según corresponda a su entorno.

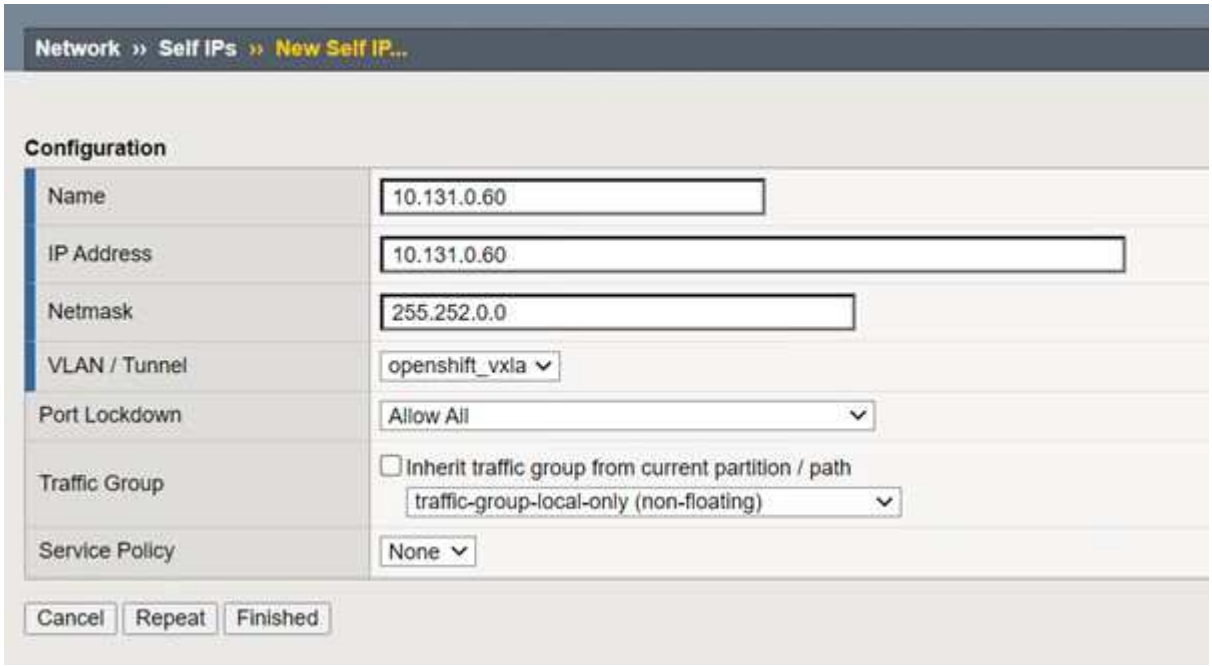
10. Cree el recurso HostSubnet.

```
[admin@rhel-7 ~]$ oc create -f f5-kctlr-openshift-hostsubnet.yaml

hostsubnet.network.openshift.io/f5-server created
```

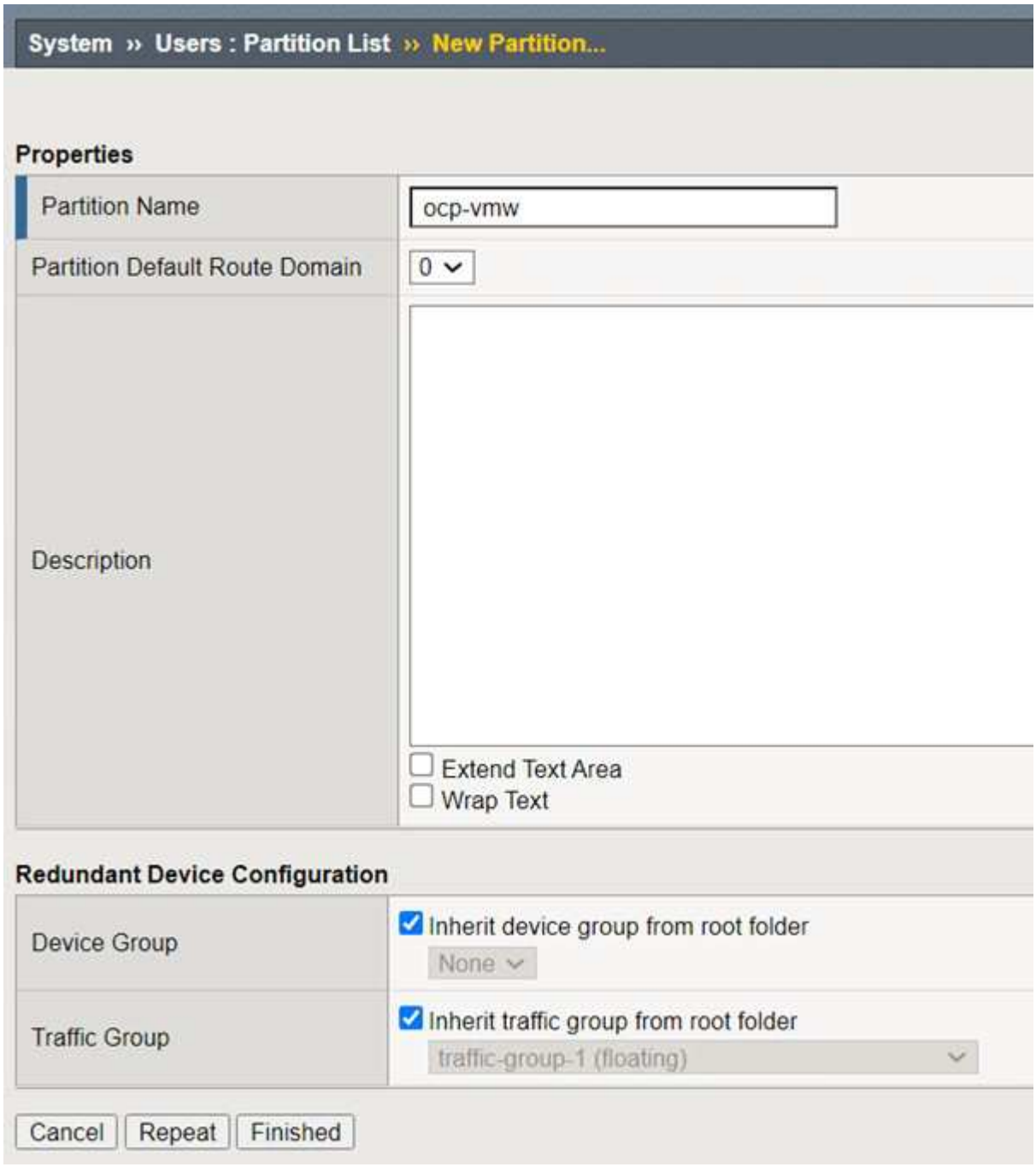
11. Obtenga el rango de subred IP del clúster para la subred del host creada para el servidor F5 BIG-IP.

```
[admin@rhel-7 ~]$ oc get hostsubnet

NAME                           HOST                           HOST IP
SUBNET          EGRESS CIDRS   EGRESS IPS
f5-server                      f5-server                      10.63.172.239
10.131.0.0/23
ocp-vmw-nszws-master-0         ocp-vmw-nszws-master-0         10.63.172.44
10.128.0.0/23
ocp-vmw-nszws-master-1         ocp-vmw-nszws-master-1         10.63.172.47
10.130.0.0/23
ocp-vmw-nszws-master-2         ocp-vmw-nszws-master-2         10.63.172.48
10.129.0.0/23
ocp-vmw-nszws-worker-r8fh4     ocp-vmw-nszws-worker-r8fh4     10.63.172.7
10.130.2.0/23
ocp-vmw-nszws-worker-tvr46     ocp-vmw-nszws-worker-tvr46     10.63.172.11
10.129.2.0/23
ocp-vmw-nszws-worker-wdxhg     ocp-vmw-nszws-worker-wdxhg     10.63.172.24
10.128.2.0/23
ocp-vmw-nszws-worker-wg8r4     ocp-vmw-nszws-worker-wg8r4     10.63.172.15
10.131.2.0/23
ocp-vmw-nszws-worker-wtgfw     ocp-vmw-nszws-worker-wtgfw     10.63.172.17
10.128.4.0/23
```

12. Cree una IP propia en OpenShift VXLAN con una IP en el rango de subred de host de OpenShift correspondiente al servidor F5 BIG-IP. Inicie sesión en el sistema F5 BIG-IP, navegue a Red > IP propias y haga clic en Crear. Ingrese una IP de la subred IP del clúster creada para la subred del host F5 BIG-IP, seleccione el túnel VXLAN e ingrese los demás detalles. Luego haga clic en Finalizar.



13. Cree una partición en el sistema F5 BIG-IP para configurarla y utilizarla con CIS. Vaya a Sistema > Usuarios > Lista de particiones, haga clic en Crear e ingrese los detalles. Luego haga clic en Finalizar.



F5 recomienda que no se realice ninguna configuración manual en la partición administrada por CIS.

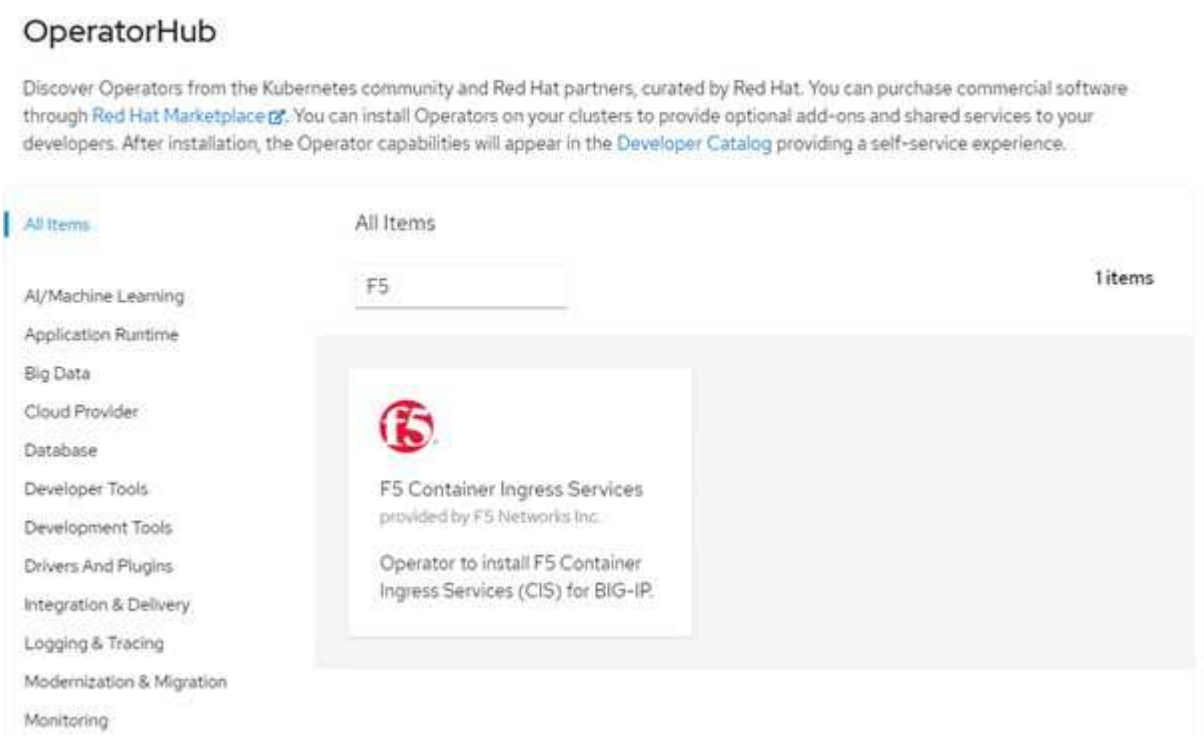
14. Instale F5 BIG-IP CIS utilizando el operador de OperatorHub. Inicie sesión en el clúster Red Hat OpenShift con privilegios de administrador del clúster y cree un secreto con las credenciales de inicio de sesión del sistema F5 BIG-IP, lo cual es un requisito previo para el operador.

```
[admin@rhel-7 ~]$ oc create secret generic bigip-login -n kube-system
--from-literal=username=admin --from-literal=password=admin

secret/bigip-login created
```

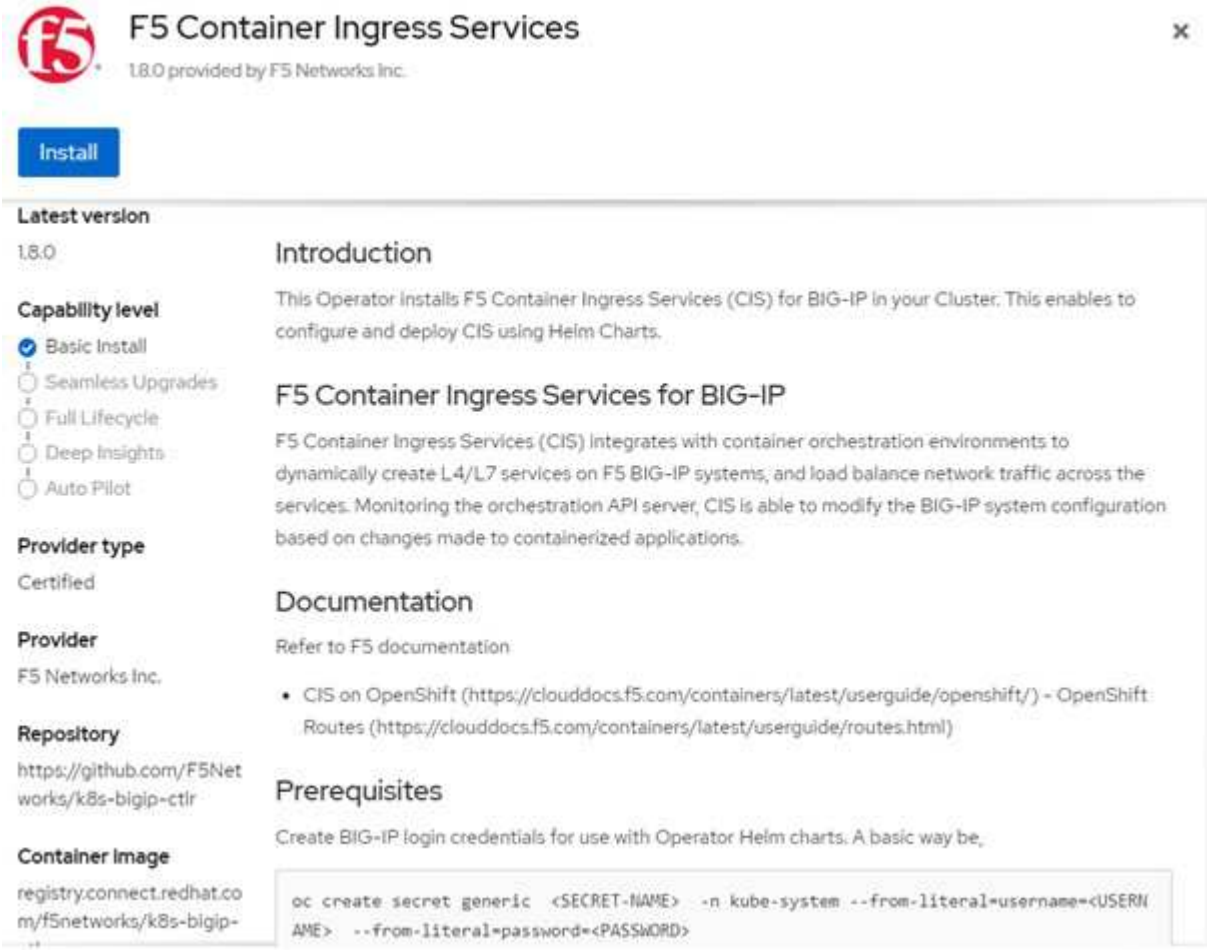
15. Instalar los CRD CIS F5.

```
[admin@rhel-7 ~]$ oc apply -f
https://raw.githubusercontent.com/F5Networks/k8s-bigip-
ctlr/master/docs/config_examples/crd/Install/customresourcedefinitions.y
ml

customresourcedefinition.apiextensions.k8s.io/virtualservers.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/tlsprofiles.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/transportservers.cis.f5.co
m created
customresourcedefinition.apiextensions.k8s.io/externaldnss.cis.f5.com
created
customresourcedefinition.apiextensions.k8s.io/ingresslinks.cis.f5.com
created
```

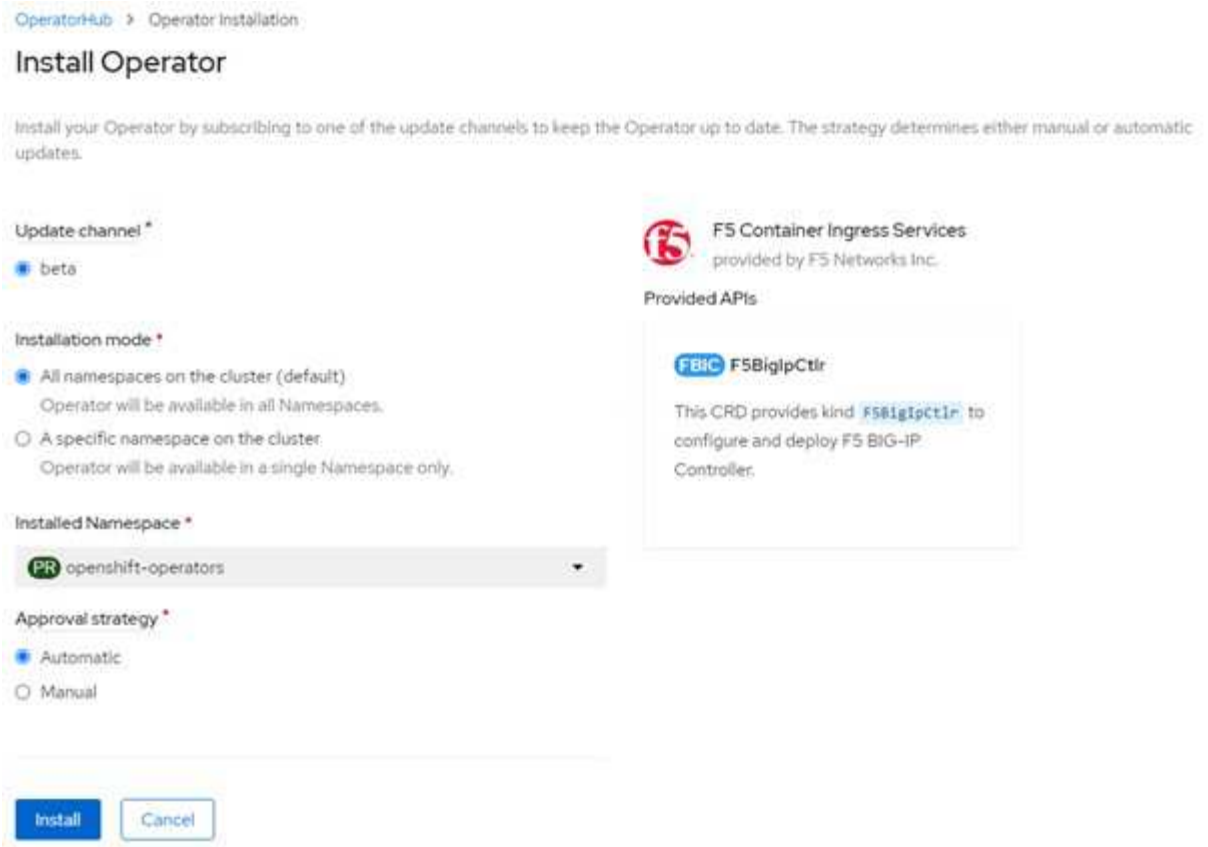
16. Vaya a Operadores > OperatorHub, busque la palabra clave F5 y haga clic en el mosaico Servicio de ingreso de contenedor F5.



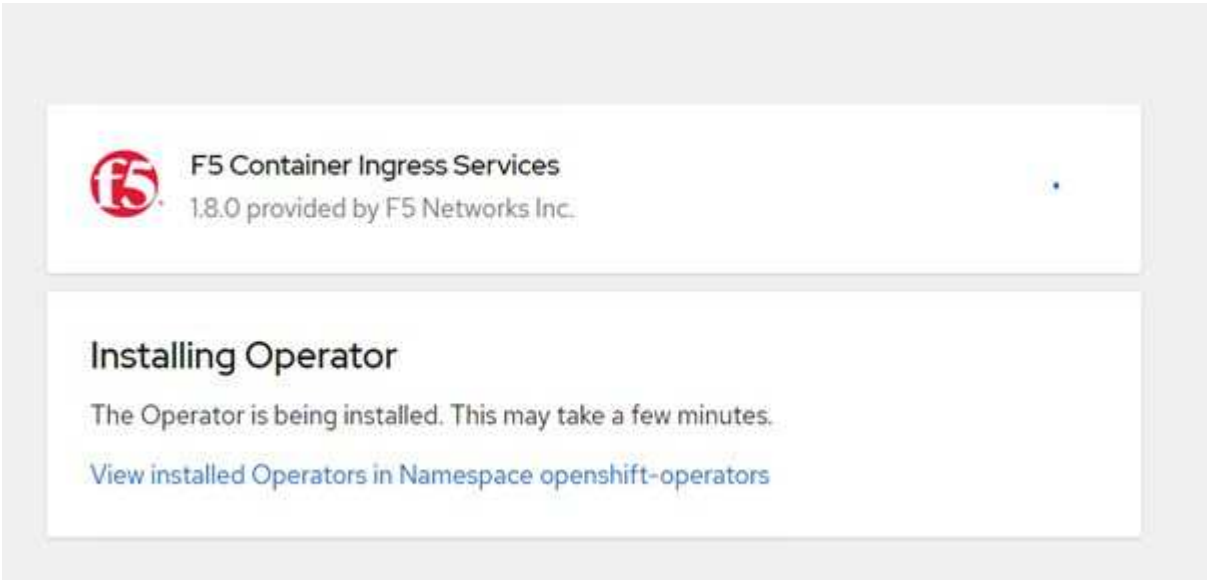
17. Lea la información del operador y haga clic en Instalar.



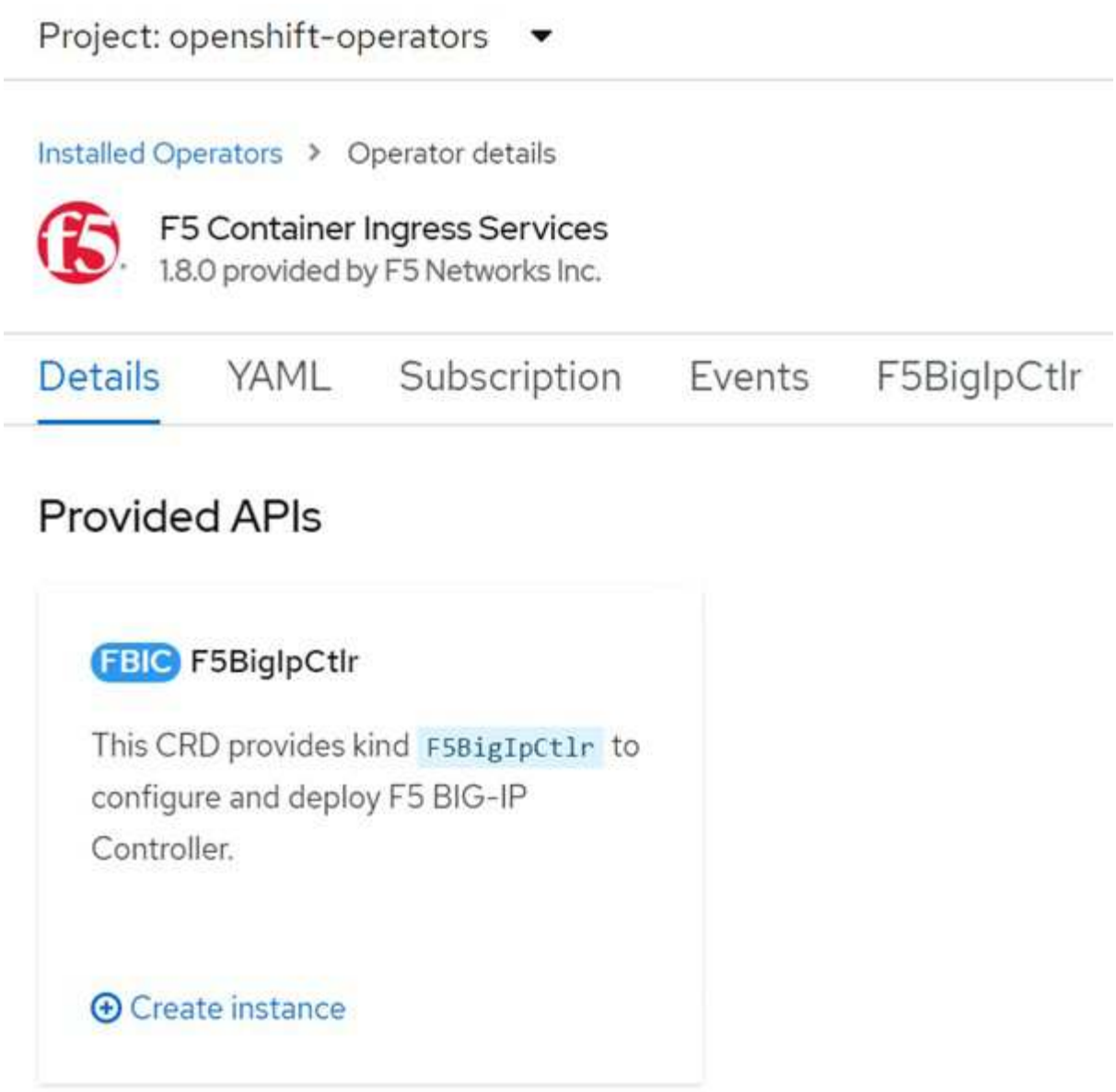
18. En la pantalla Instalar operador, deje todos los parámetros predeterminados y haga clic en Instalar.



19. La instalación del operador demora un tiempo.



20. Una vez instalado el operador, se muestra el mensaje Instalación exitosa.
21. Vaya a Operadores > Operadores instalados, haga clic en Servicio de ingreso de contenedor F5 y, a continuación, haga clic en Crear instancia debajo del mosaico F5BigIpCtlr.



22. Haga clic en Vista YAML y pegue el siguiente contenido después de actualizar los parámetros necesarios.

Actualizar los parámetros `bigip_partition` , `openshift_sdn_name`, `bigip_url` y `bigip_login_secret` a continuación para reflejar los valores de su configuración antes de copiar el contenido.

```
apiVersion: cis.f5.com/v1
kind: F5BigIpCtlr
metadata:
  name: f5-server
  namespace: openshift-operators
spec:
  args:
    log_as3_response: true
    agent: as3
    log_level: DEBUG
    bigip_partition: ocp-vmw
    openshift_sdn_name: /Common/openshift_vxlan
    bigip_url: 10.61.181.19
    insecure: true
    pool-member-type: cluster
    custom_resource_mode: true
    as3_validation: true
    ipam: true
    manage_configmaps: true
  bigip_login_secret: bigip-login
  image:
    pullPolicy: Always
    repo: f5networks/cntr-ingress-svcs
    user: registry.connect.redhat.com
  namespace: kube-system
  rbac:
    create: true
  resources: {}
  serviceAccount:
    create: true
  version: latest
```

23. Después de pegar este contenido, haga clic en Crear. Esto instala los pods CIS en el espacio de nombres kube-system.

Red Hat OpenShift, de forma predeterminada, proporciona una forma de exponer los servicios a través de rutas para el equilibrio de carga L7. Un enrutador OpenShift incorporado es responsable de publicitar y gestionar el tráfico de estas rutas. Sin embargo, también puede configurar el F5 CIS para admitir las rutas a través de un sistema F5 BIG-IP externo, que puede ejecutarse como un enrutador auxiliar o un reemplazo del enrutador OpenShift autohospedado. CIS crea un servidor virtual en el sistema BIG-IP que actúa como enrutador para las rutas OpenShift, y BIG-IP maneja la publicidad y el enrutamiento del tráfico. Consulte la documentación aquí para obtener información sobre los parámetros para habilitar esta función. Tenga en cuenta que estos parámetros están definidos para el recurso de implementación de OpenShift en la API apps/v1. Por lo tanto, al utilizarlos con el recurso F5BigIpCtlr API cis.f5.com/v1, reemplace los guiones (-) con guiones bajos (_) para los nombres de los parámetros.

24. Los argumentos que se pasan a la creación de recursos CIS incluyen `ipam: true` y `custom_resource_mode: true`. Estos parámetros son necesarios para habilitar la integración de CIS con un controlador IPAM. Verifique que el CIS haya habilitado la integración de IPAM creando el recurso IPAM de F5.

```
[admin@rhel-7 ~]$ oc get f5ipam -n kube-system

NAMESPACE   NAME                              AGE
kube-system   ipam.10.61.181.19.ocp-vmw       43s
```

25. Cree la cuenta de servicio, el rol y la vinculación de roles necesarios para el controlador F5 IPAM. Crea un archivo YAML y pega el siguiente contenido.

```
[admin@rhel-7 ~]$ vi f5-ipam-rbac.yaml

kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole
rules:
  - apiGroups: ["fic.f5.com"]
    resources: ["ipams","ipams/status"]
    verbs: ["get", "list", "watch", "update", "patch"]
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ipam-ctlr-clusterrole-binding
  namespace: kube-system
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: ipam-ctlr-clusterrole
subjects:
  - apiGroup: ""
    kind: ServiceAccount
    name: ipam-ctlr
    namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ipam-ctlr
  namespace: kube-system
```

26. Crear los recursos.

```
[admin@rhel-7 ~]$ oc create -f f5-ipam-rbac.yaml

clusterrole.rbac.authorization.k8s.io/ipam-ctlr-clusterrole created
clusterrolebinding.rbac.authorization.k8s.io/ipam-ctlr-clusterrole-
binding created
serviceaccount/ipam-ctlr created
```

27. Cree un archivo YAML y pegue la definición de implementación de F5 IPAM que se proporciona a continuación.

Actualice el parámetro ip-range en spec.template.spec.containers[0].args a continuación para reflejar las ipamLabels y los rangos de direcciones IP correspondientes a su configuración.

Etiquetas ipam[`range1` y `range2` [en el siguiente ejemplo] se requiere anotar los valores para los servicios de tipo LoadBalancer para que el controlador IPAM detecte y asigne una dirección IP del rango definido.

```
[admin@rhel-7 ~]$ vi f5-ipam-deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    name: f5-ipam-controller
  name: f5-ipam-controller
  namespace: kube-system
spec:
  replicas: 1
  selector:
    matchLabels:
      app: f5-ipam-controller
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: f5-ipam-controller
    spec:
      containers:
      - args:
        - --orchestration=openshift
        - --ip-range='{"range1":"10.63.172.242-10.63.172.249",
"range2":"10.63.170.111-10.63.170.129"}'
        - --log-level=DEBUG
        command:
        - /app/bin/f5-ipam-controller
        image: registry.connect.redhat.com/f5networks/f5-ipam-
controller:latest
        imagePullPolicy: IfNotPresent
        name: f5-ipam-controller
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      serviceAccount: ipam-ctlr
      serviceAccountName: ipam-ctlr
```

28. Cree la implementación del controlador F5 IPAM.

```
[admin@rhel-7 ~]$ oc create -f f5-ipam-deployment.yaml

deployment/f5-ipam-controller created
```

29. Verifique que los pods del controlador IPAM F5 estén funcionando.

```
[admin@rhel-7 ~]$ oc get pods -n kube-system

NAME                                      READY   STATUS    RESTARTS
AGE
f5-ipam-controller-5986cff5bd-2bvn6       1/1     Running   0
30s
f5-server-f5-bigip-ctlr-5d7578667d-qxdgj  1/1     Running   0
14m
```

30. Cree el esquema IPAM de F5.

```
[admin@rhel-7 ~]$ oc create -f
https://raw.githubusercontent.com/F5Networks/f5-ipam-
controller/main/docs/_static/schemas/ipam_schema.yaml

customresourcedefinition.apiextensions.k8s.io/ipams.fic.f5.com
```

Verificación

1. Crear un servicio de tipo LoadBalancer

```
[admin@rhel-7 ~]$ vi example_svc.yaml

apiVersion: v1
kind: Service
metadata:
  annotations:
    cis.f5.com/ipamLabel: range1
  labels:
    app: f5-demo-test
  name: f5-demo-test
  namespace: default
spec:
  ports:
  - name: f5-demo-test
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: f5-demo-test
  sessionAffinity: None
  type: LoadBalancer
```

```
[admin@rhel-7 ~]$ oc create -f example_svc.yaml

service/f5-demo-test created
```

2. Compruebe si el controlador IPAM le asigna una IP externa.

```
[admin@rhel-7 ~]$ oc get svc

NAME           TYPE           CLUSTER-IP       EXTERNAL-IP
PORT(S)        AGE
f5-demo-test   LoadBalancer   172.30.210.108   10.63.172.242
80:32605/TCP   27s
```

3. Cree una implementación y utilice el servicio LoadBalancer que se creó.

```
[admin@rhel-7 ~]$ vi example_deployment.yaml

apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
    app: f5-demo-test
  name: f5-demo-test
spec:
  replicas: 2
  selector:
    matchLabels:
      app: f5-demo-test
  template:
    metadata:
      labels:
        app: f5-demo-test
    spec:
      containers:
      - env:
        - name: service_name
          value: f5-demo-test
        image: nginx
        imagePullPolicy: Always
        name: f5-demo-test
        ports:
        - containerPort: 80
          protocol: TCP
```

```
[admin@rhel-7 ~]$ oc create -f example_deployment.yaml

deployment/f5-demo-test created
```

4. Comprueba si los pods están funcionando.

```
[admin@rhel-7 ~]$ oc get pods

NAME                           READY   STATUS    RESTARTS   AGE
f5-demo-test-57c46f6f98-47wwp   1/1     Running   0          27s
f5-demo-test-57c46f6f98-cl2m8   1/1     Running   0          27s
```

5. Verifique si el servidor virtual correspondiente está creado en el sistema BIG-IP para el servicio de tipo LoadBalancer en OpenShift. Vaya a Tráfico local > Servidores virtuales > Lista de servidores virtuales.

Creación de registros de imágenes privadas

Para la mayoría de las implementaciones de Red Hat OpenShift, se utiliza un registro público como "Quay.io" o "DockerHub" Satisface la mayoría de las necesidades de los clientes. Sin embargo, hay ocasiones en las que un cliente puede querer alojar sus propias imágenes privadas o personalizadas.

Este procedimiento documenta la creación de un registro de imágenes privado respaldado por un volumen persistente proporcionado por Trident y NetApp ONTAP.

> Trident Protect requiere un registro para alojar las imágenes que requieren los contenedores Astra . La siguiente sección describe los pasos para configurar un registro privado en el clúster Red Hat OpenShift y enviar las imágenes necesarias para soportar la instalación de Trident Protect.

Creación de un registro de imágenes privado

1. Elimine la anotación predeterminada de la clase de almacenamiento predeterminada actual y anote la clase de almacenamiento respaldada por Trident como predeterminada para el clúster OpenShift.

```
[netapp-user@rhel7 ~]$ oc patch storageclass thin -p '{"metadata":
{"annotations": {"storageclass.kubernetes.io/is-default-class":
"false"}}}'
storageclass.storage.k8s.io/thin patched

[netapp-user@rhel7 ~]$ oc patch storageclass ocp-trident -p
'{"metadata": {"annotations": {"storageclass.kubernetes.io/is-default-
class": "true"}}}'
storageclass.storage.k8s.io/ocp-trident patched
```

2. Edite el operador de registro de imágenes ingresando los siguientes parámetros de almacenamiento en el `spec` sección.

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

storage:
  pvc:
    claim:
```

3. Introduzca los siguientes parámetros en el `spec` Sección para crear una ruta OpenShift con un nombre de host personalizado. Guardar y salir.

```
routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
```

La configuración de ruta anterior se utiliza cuando desea un nombre de host personalizado para su ruta. Si desea que OpenShift cree una ruta con un nombre de host predeterminado, puede agregar los siguientes parámetros a la `spec` sección: `defaultRoute: true` .

Certificados TLS personalizados

Cuando se utiliza un nombre de host personalizado para la ruta, de manera predeterminada, se utiliza la configuración TLS predeterminada del operador Ingress de OpenShift. Sin embargo, puede agregar una configuración TLS personalizada a la ruta. Para ello, complete los siguientes pasos.

a. Crea un secreto con los certificados TLS y la clave de la ruta.

```
[netapp-user@rhel7 ~]$ oc create secret tls astra-route-tls -n
openshift-image-registry –cert/home/admin/netapp-astra/tls.crt
--key=/home/admin/netapp-astra/tls.key
```

b. Edite el operador de registro de imágenes y agregue los siguientes parámetros a la `spec` sección.

```
[netapp-user@rhel7 ~]$ oc edit
configs.imageregistry.operator.openshift.io

routes:
  - hostname: astra-registry.apps.ocp-vmw.cie.netapp.com
    name: netapp-astra-route
    secretName: astra-route-tls
```

4. Edite nuevamente el operador de registro de imágenes y cambie el estado de administración del operador al `Managed` estado. Guardar y salir.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Si se cumplen todos los requisitos previos, se crean PVC, pods y servicios para el registro de imágenes

privadas. En unos minutos el registro debería estar activo.

```
[netapp-user@rhel7 ~]$oc get all -n openshift-image-registry

NAME                                                READY   STATUS
RESTARTS   AGE
pod/cluster-image-registry-operator-74f6d954b6-rb7zr   1/1     Running
3          90d
pod/image-pruner-1627257600-f5cpj                   0/1     Completed
0          2d9h
pod/image-pruner-1627344000-swqx9                   0/1     Completed
0          33h
pod/image-pruner-1627430400-rv5nt                   0/1     Completed
0          9h
pod/image-registry-6758b547f-6pnj8                  1/1     Running
0          76m
pod/node-ca-bwb5r                                   1/1     Running
0          90d
pod/node-ca-f8w54                                   1/1     Running
0          90d
pod/node-ca-gjx7h                                   1/1     Running
0          90d
pod/node-ca-lcx4k                                   1/1     Running
0          33d
pod/node-ca-v7zmx                                   1/1     Running
0          7d21h
pod/node-ca-xpppp                                   1/1     Running
0          89d

NAME                              TYPE        CLUSTER-IP       EXTERNAL-
IP   PORT(S)     AGE
service/image-registry            ClusterIP   172.30.196.167   <none>
5000/TCP    15h
service/image-registry-operator   ClusterIP   None             <none>
60000/TCP   90d

NAME                     DESIRED   CURRENT   READY   UP-TO-DATE
AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/node-ca   6         6         6       6            6
kubernetes.io/os=linux   90d

NAME                                           READY   UP-TO-DATE
AVAILABLE   AGE
deployment.apps/cluster-image-registry-operator   1/1     1            1
90d
deployment.apps/image-registry                    1/1     1            1
```

```
15h

NAME                                                                DESIRED
CURRENT   READY   AGE
replicaset.apps/cluster-image-registry-operator-74f6d954b6   1         1
1       90d
replicaset.apps/image-registry-6758b547f                     1         1
1       76m
replicaset.apps/image-registry-78bfbd7f59                    0         0
0       15h
replicaset.apps/image-registry-7fcc8d6cc8                    0         0
0       80m
replicaset.apps/image-registry-864f88f5b                     0         0
0       15h
replicaset.apps/image-registry-cb47fffb                      0         0
0       10h

NAME                                  COMPLETIONS   DURATION   AGE
job.batch/image-pruner-1627257600   1/1           10s        2d9h
job.batch/image-pruner-1627344000   1/1           6s         33h
job.batch/image-pruner-1627430400   1/1           5s         9h

NAME                          SCHEDULE    SUSPEND   ACTIVE   LAST
SCHEDULE   AGE
cronjob.batch/image-pruner   0 0 * * *   False     0        9h
90d

NAME                                       HOST/PORT
PATH   SERVICES         PORT    TERMINATION   WILDCARD
route.route.openshift.io/public-routes   astra-registry.apps.ocp-
vmw.cie.netapp.com          image-registry   <all>   reencrypt     None
```

6. Si está utilizando los certificados TLS predeterminados para la ruta de registro de OpenShift del operador de ingreso, puede obtener los certificados TLS mediante el siguiente comando.

```
[netapp-user@rhel7 ~]$ oc extract secret/router-ca --keys=tls.crt -n
openshift-ingress-operator
```

7. Para permitir que los nodos OpenShift accedan y extraigan las imágenes del registro, agregue los certificados al cliente Docker en los nodos OpenShift. Crea un mapa de configuración en el `openshift-config` espacio de nombres que utiliza los certificados TLS y lo parchea en la configuración de la imagen del clúster para que el certificado sea confiable.

```
[netapp-user@rhel7 ~]$ oc create configmap astra-ca -n openshift-config
--from-file=astra-registry.apps.ocp-vmw.cie.netapp.com=tls.crt

[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster
--patch '{"spec":{"additionalTrustedCA":{"name":"astra-ca"}}}'
--type=merge
```

8. El registro interno de OpenShift está controlado por autenticación. Todos los usuarios de OpenShift pueden acceder al registro de OpenShift, pero las operaciones que puede realizar el usuario conectado dependen de los permisos del usuario.
   a. Para permitir que un usuario o un grupo de usuarios extraigan imágenes del registro, los usuarios deben tener asignado el rol de visor de registro.

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-viewer
ocp-user

[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-viewer
ocp-user-group
```

   b. Para permitir que un usuario o grupo de usuarios escriba o envíe imágenes, los usuarios deben tener asignado el rol de editor de registro.

```
[netapp-user@rhel7 ~]$ oc policy add-role-to-user registry-editor
ocp-user

[netapp-user@rhel7 ~]$ oc policy add-role-to-group registry-editor
ocp-user-group
```

9. Para que los nodos OpenShift accedan al registro e inserten o extraigan las imágenes, debe configurar un secreto de extracción.

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-registry-
credentials --docker-server=astra-registry.apps.ocp-vmw.cie.netapp.com
--docker-username=ocp-user --docker-password=password
```

10. Luego, este secreto de extracción se puede parchear en las cuentas de servicio o hacer referencia a él en la definición de pod correspondiente.
    a. Para aplicar el parche a las cuentas de servicio, ejecute el siguiente comando.

```
[netapp-user@rhel7 ~]$ oc secrets link <service_account_name> astra-
registry-credentials --for=pull
```

b. Para hacer referencia al secreto de extracción en la definición del pod, agregue el siguiente parámetro a la `spec` sección.

```
imagePullSecrets:
  - name: astra-registry-credentials
```

11. Para enviar o recibir una imagen desde estaciones de trabajo distintas del nodo OpenShift, complete los siguientes pasos.

a. Agregue los certificados TLS al cliente de Docker.

```
[netapp-user@rhel7 ~]$ sudo mkdir /etc/docker/certs.d/astra-
registry.apps.ocp-vmw.cie.netapp.com

[netapp-user@rhel7 ~]$ sudo cp /path/to/tls.crt
/etc/docker/certs.d/astra-registry.apps.ocp-vmw.cie.netapp.com
```

b. Inicie sesión en OpenShift utilizando el comando oc login.

```
[netapp-user@rhel7 ~]$ oc login --token=sha256~D49SpB_lesSrJYwrM0LIO
-VRcjWHu0a27vKa0 --server=https://api.ocp-vmw.cie.netapp.com:6443
```

c. Inicie sesión en el registro utilizando las credenciales de usuario de OpenShift con el comando podman/docker.

hombre de pod

```
[netapp-user@rhel7 ~]$ podman login astra-registry.apps.ocp-
vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t) --tls
-verify=false
```

+ NOTA: Si está utilizando `kubeadmin` El usuario debe iniciar sesión en el registro privado y luego usar el token en lugar de la contraseña.

estibador

```
[netapp-user@rhel7 ~]$ docker login astra-registry.apps.ocp-
vmw.cie.netapp.com -u kubeadmin -p $(oc whoami -t)
```

+ NOTA: Si está utilizando `kubeadmin` El usuario debe iniciar sesión en el registro privado y luego usar el token en lugar de la contraseña.

d. Empuja o tira las imágenes.

hombre de pod

```
[netapp-user@rhel7 ~]$ podman push astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
[netapp-user@rhel7 ~]$ podman pull astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

estibador

```
[netapp-user@rhel7 ~]$ docker push astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
[netapp-user@rhel7 ~]$ docker pull astra-registry.apps.ocp-
vmw.cie.netapp.com/netapp-astra/vault-controller:latest
```

Validación de soluciones y casos de uso

Validación de soluciones y casos de uso: Red Hat OpenShift con NetApp

Los ejemplos proporcionados en esta página son validaciones de soluciones y casos de uso para Red Hat OpenShift con NetApp.

- "Implementar una canalización de CI/CD de Jenkins con almacenamiento persistente"
- "Configurar multitenencia en Red Hat OpenShift con NetApp"
- "Virtualización de Red Hat OpenShift con NetApp ONTAP"
- "Gestión avanzada de clústeres para Kubernetes en Red Hat OpenShift con NetApp"

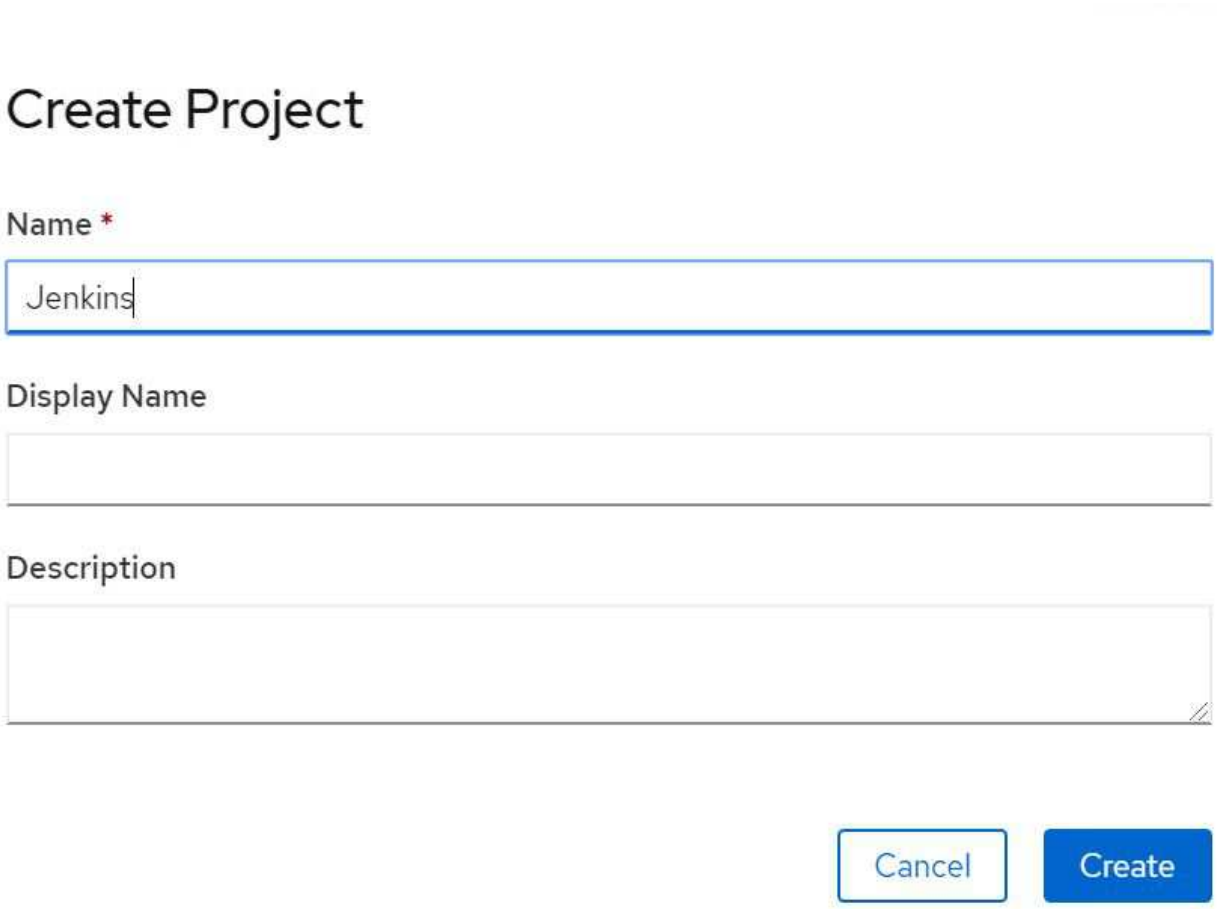
Implementar una canalización de CI/CD de Jenkins con almacenamiento persistente: Red Hat OpenShift con NetApp

Esta sección proporciona los pasos para implementar una canalización de integración continua/entrega continua o implementación (CI/CD) con Jenkins para validar el funcionamiento de la solución.

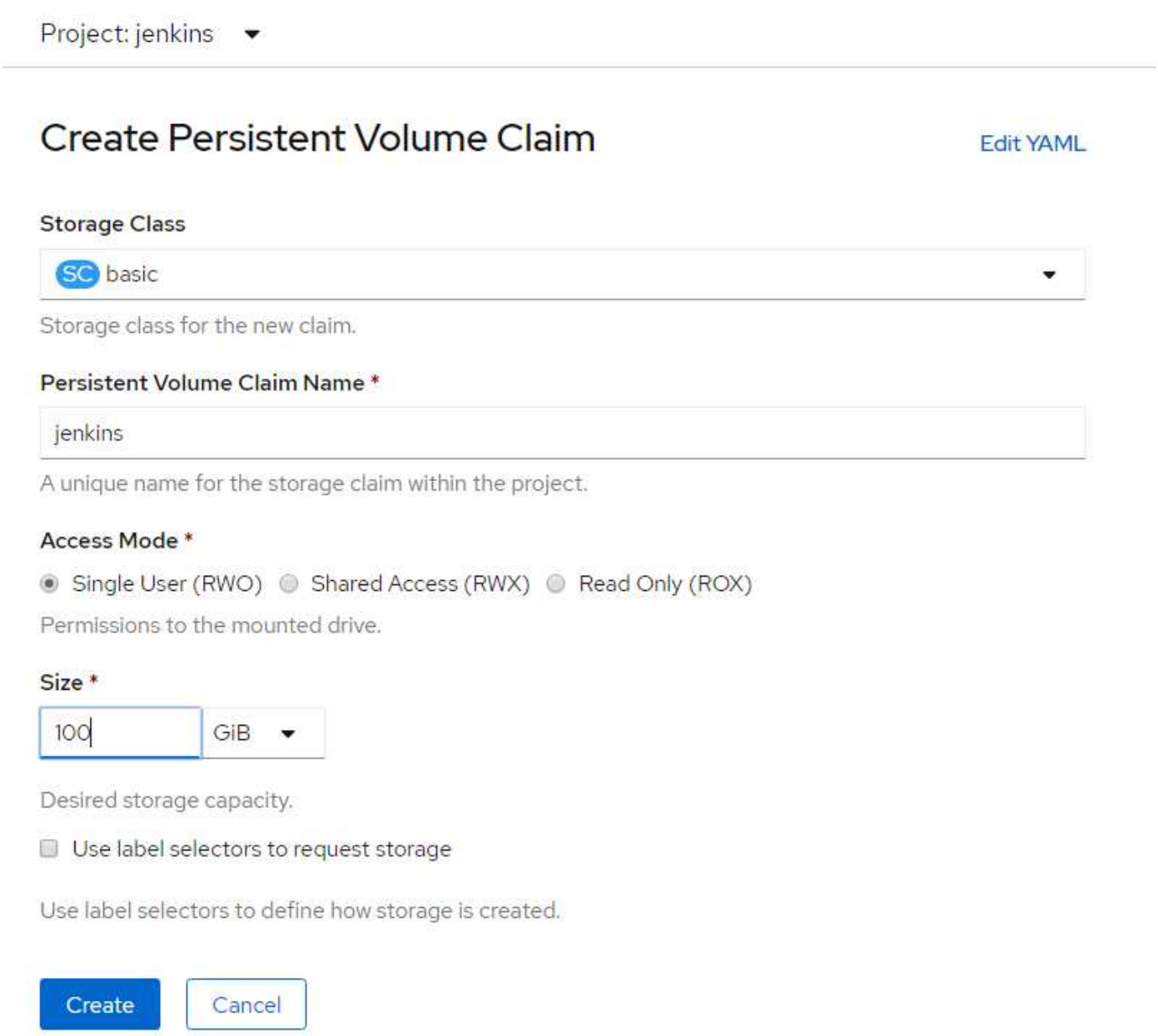
Cree los recursos necesarios para la implementación de Jenkins

Para crear los recursos necesarios para implementar la aplicación Jenkins, complete los siguientes pasos:

1. Crea un nuevo proyecto llamado Jenkins.



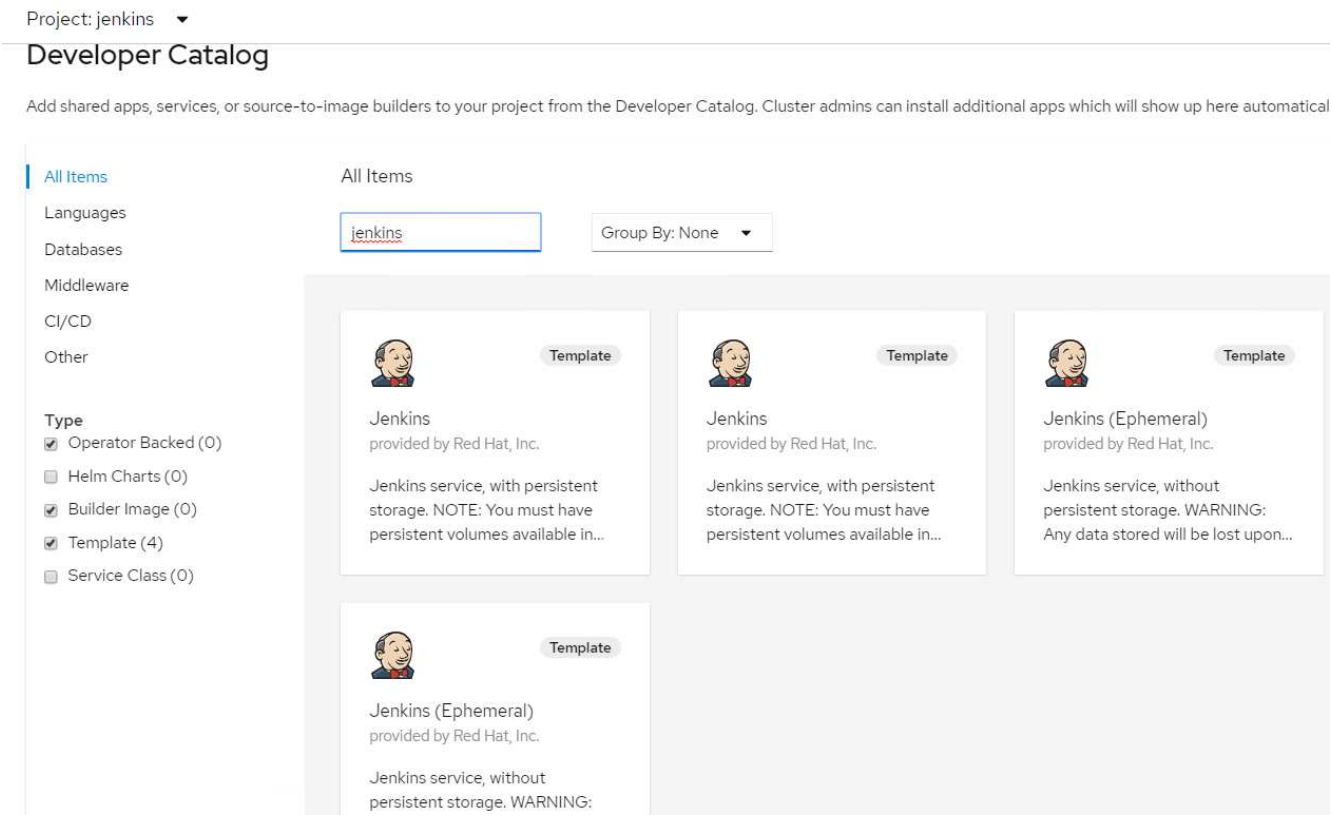
2. En este ejemplo, implementamos Jenkins con almacenamiento persistente. Para respaldar la compilación de Jenkins, cree el PVC. Vaya a Almacenamiento > Reclamaciones de volumen persistente y haga clic en Crear reclamación de volumen persistente. Seleccione la clase de almacenamiento que se creó, asegúrese de que el nombre de reclamo de volumen persistente sea jenkins, seleccione el tamaño y el modo de acceso adecuados y luego haga clic en Crear.



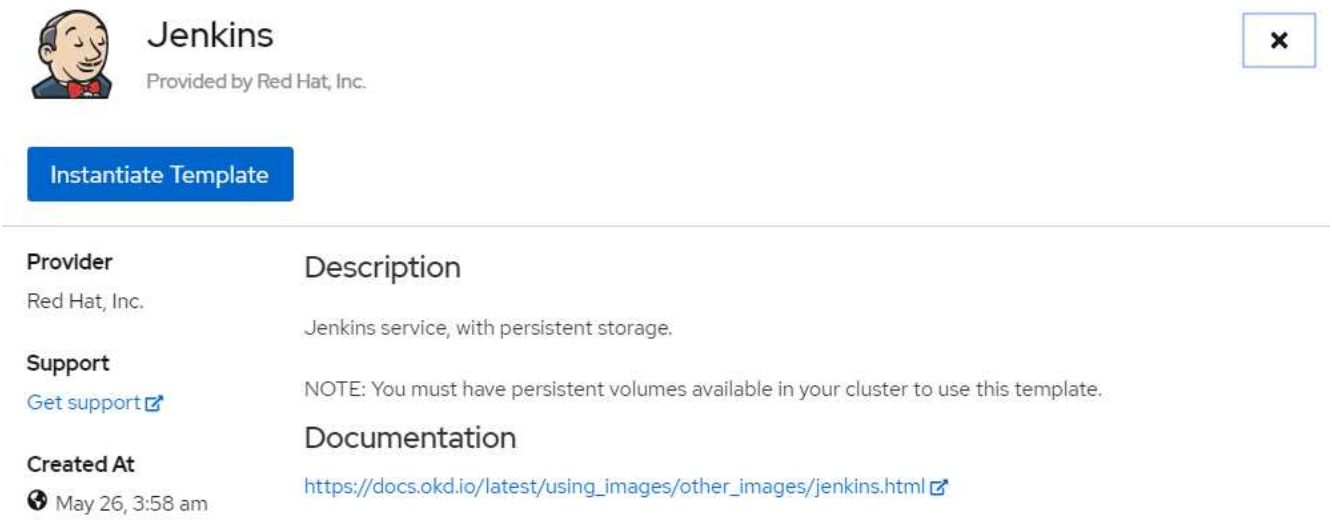
Implementar Jenkins con almacenamiento persistente

Para implementar Jenkins con almacenamiento persistente, complete los siguientes pasos:

1. En la esquina superior izquierda, cambie el rol de Administrador a Desarrollador. Haga clic en +Agregar y seleccione Desde catálogo. En la barra Filtrar por palabra clave, busque jenkins. Seleccione el servicio Jenkins con almacenamiento persistente.

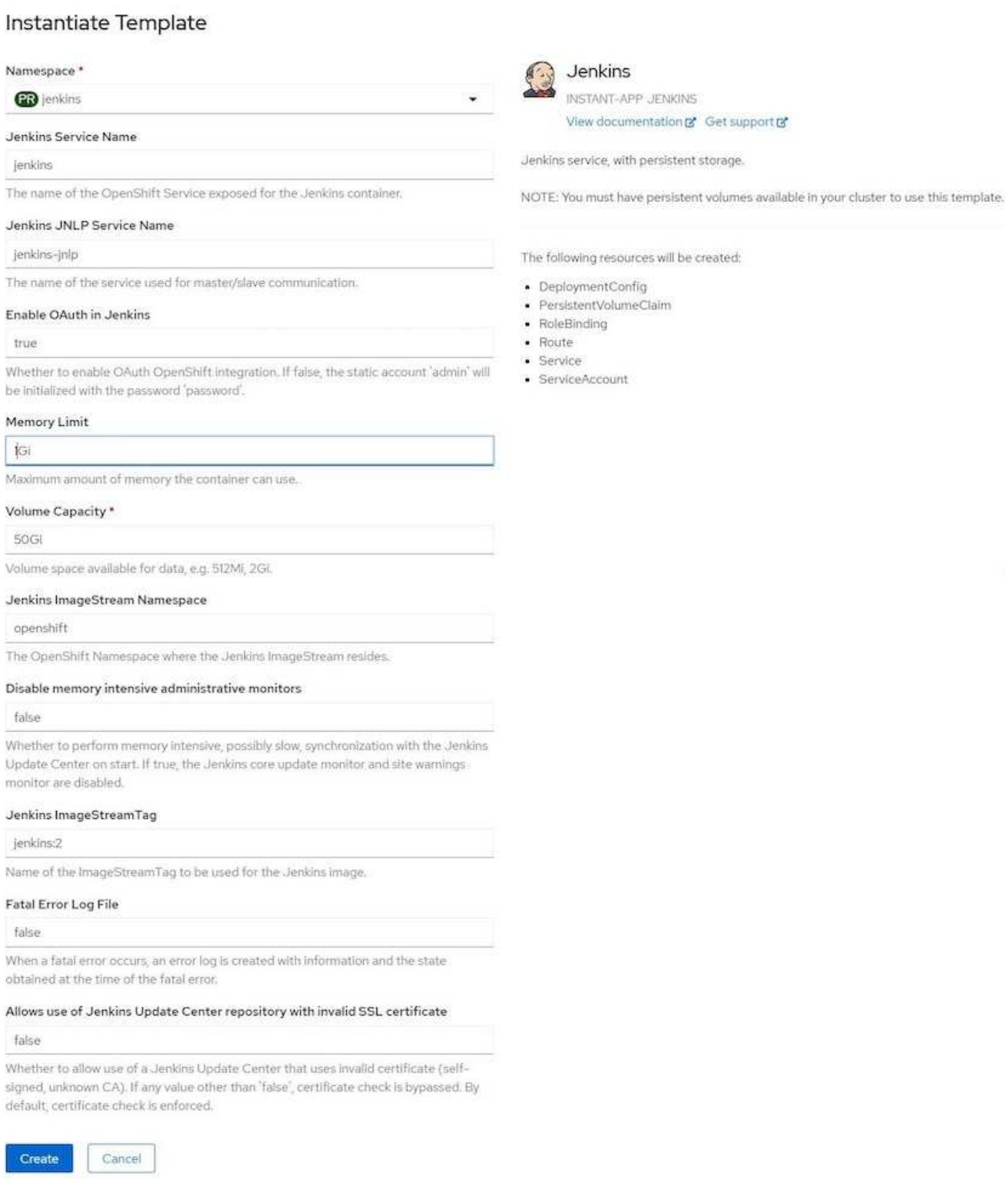


2. Hacer clic `Instantiate Template` .

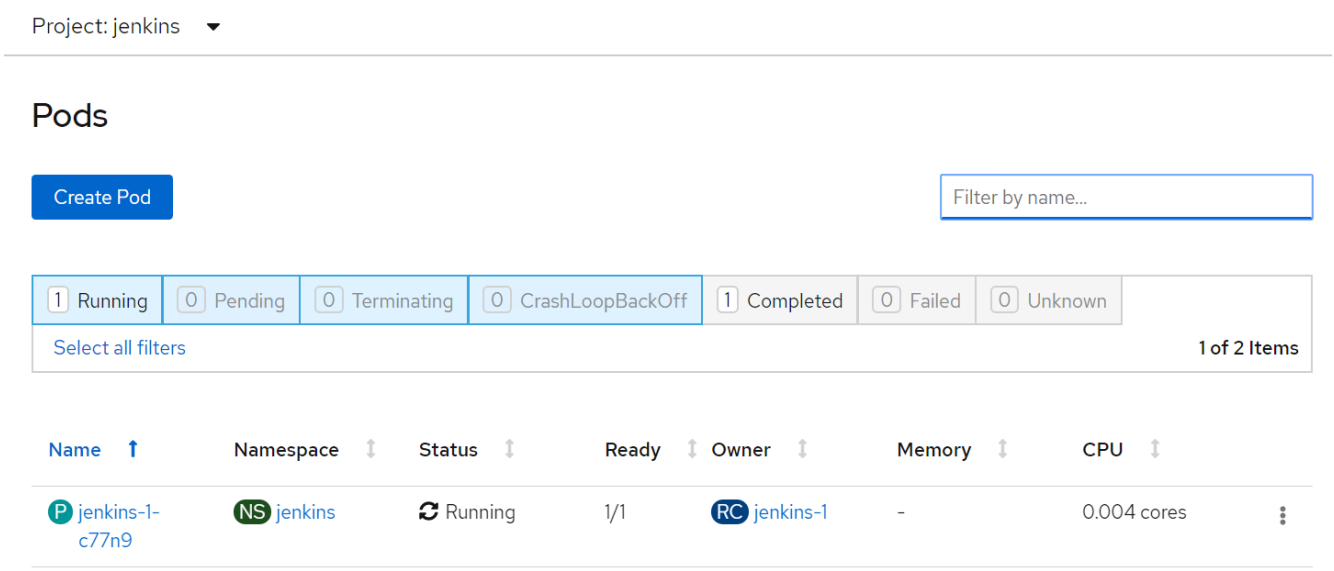


3. De forma predeterminada, se completan los detalles de la aplicación Jenkins. Según sus requisitos, modifique los parámetros y haga clic en Crear. Este proceso crea todos los recursos necesarios para

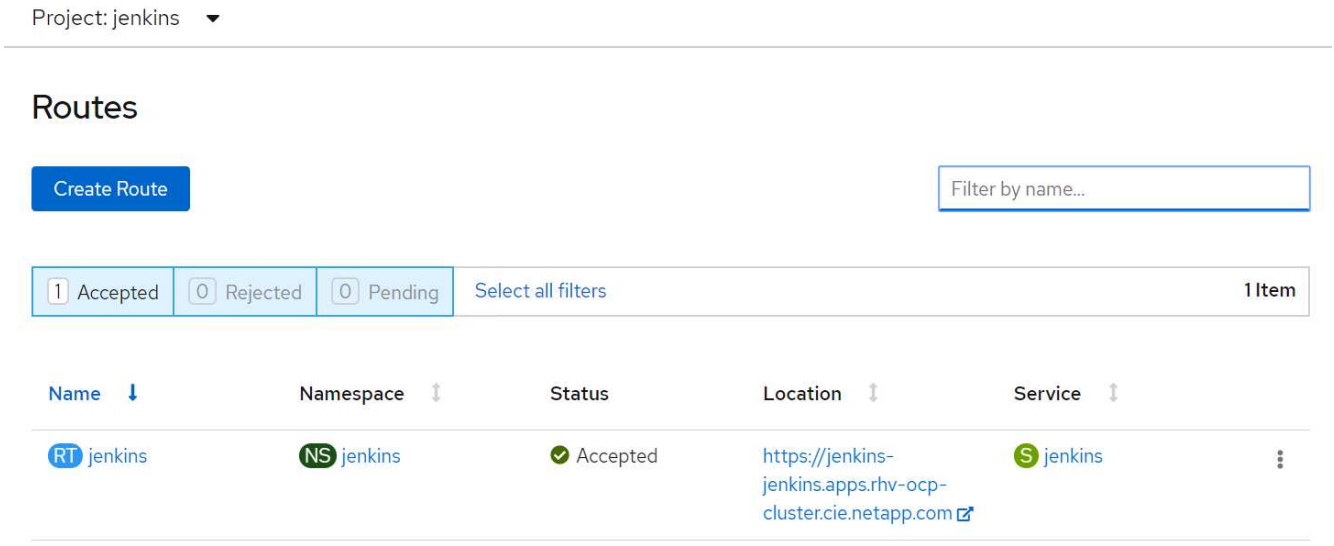
respaldar Jenkins en OpenShift.



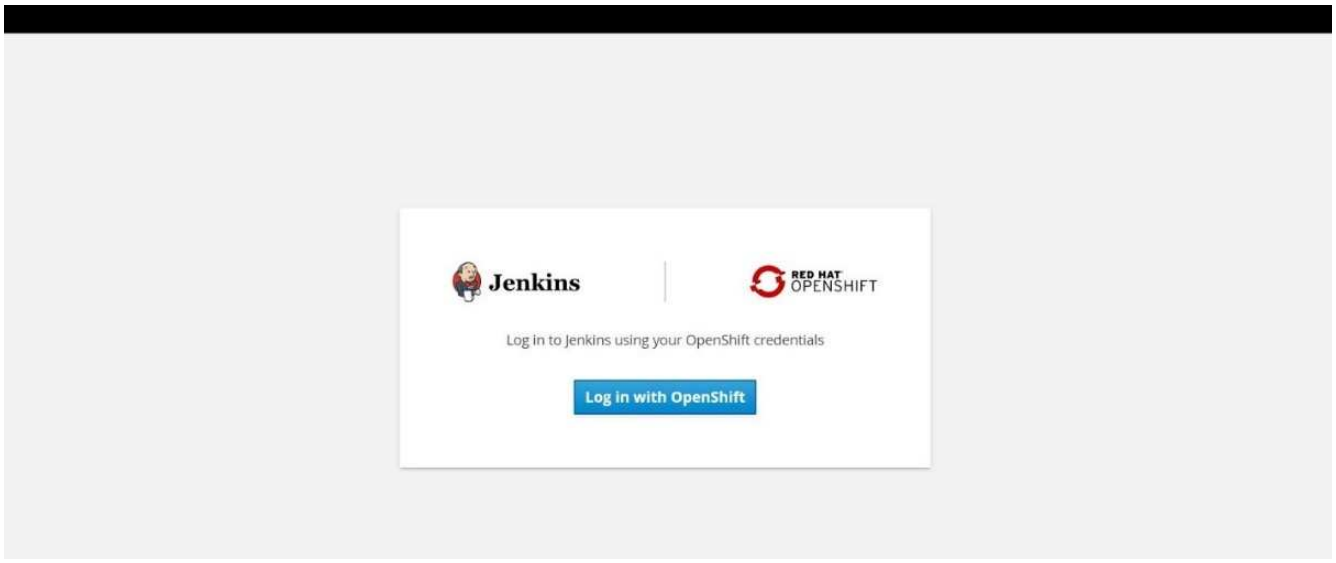
4. Los pods Jenkins tardan aproximadamente entre 10 y 12 minutos en entrar en el estado Listo.



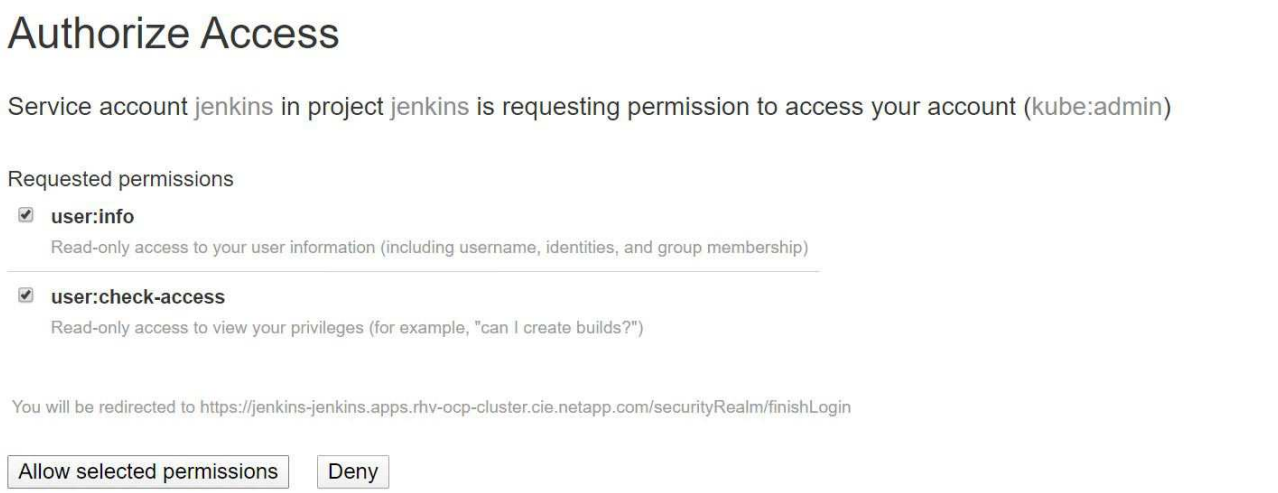
5. Una vez instanciados los pods, navegue a Redes > Rutas. Para abrir la página web de Jenkins, haga clic en la URL proporcionada para la ruta de Jenkins.



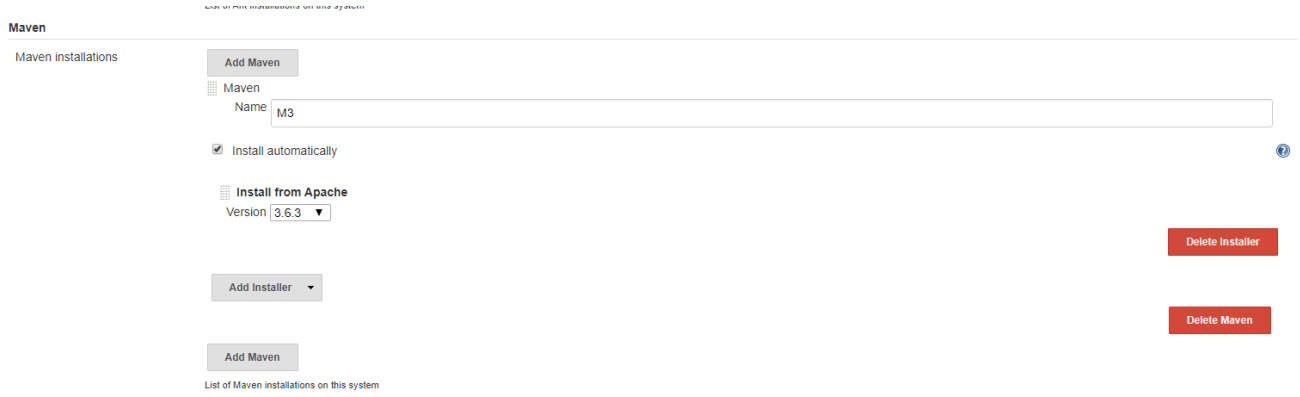
6. Dado que se utilizó OpenShift OAuth al crear la aplicación Jenkins, haga clic en Iniciar sesión con OpenShift.



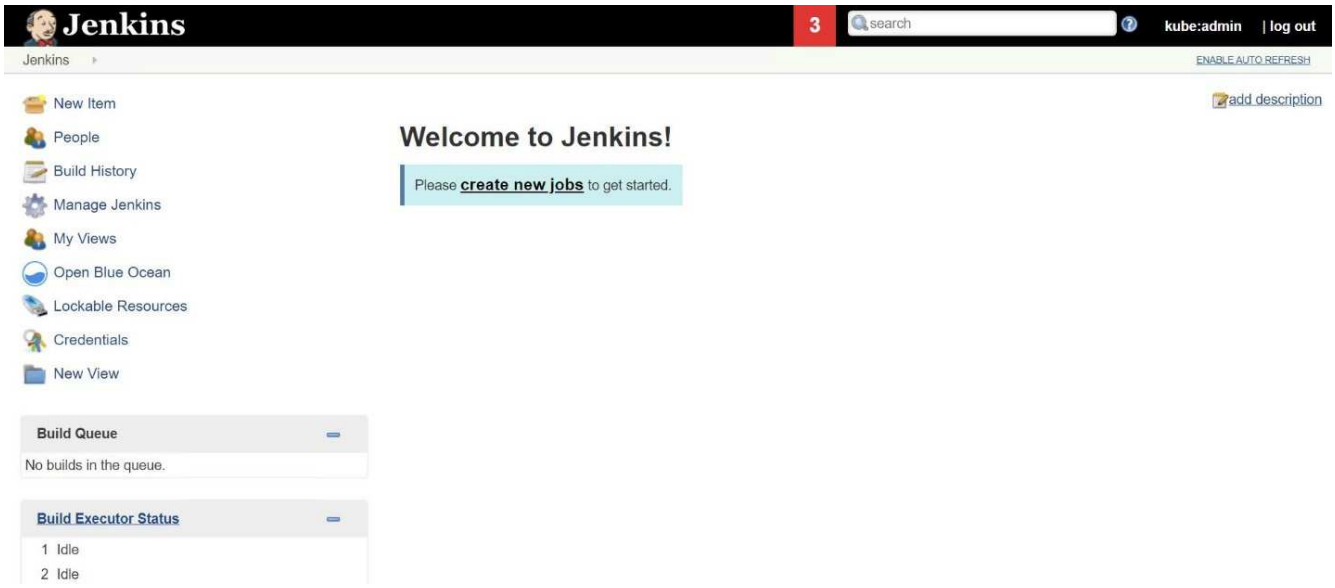
7. Autorice la cuenta de servicio de Jenkins para acceder a los usuarios de OpenShift.



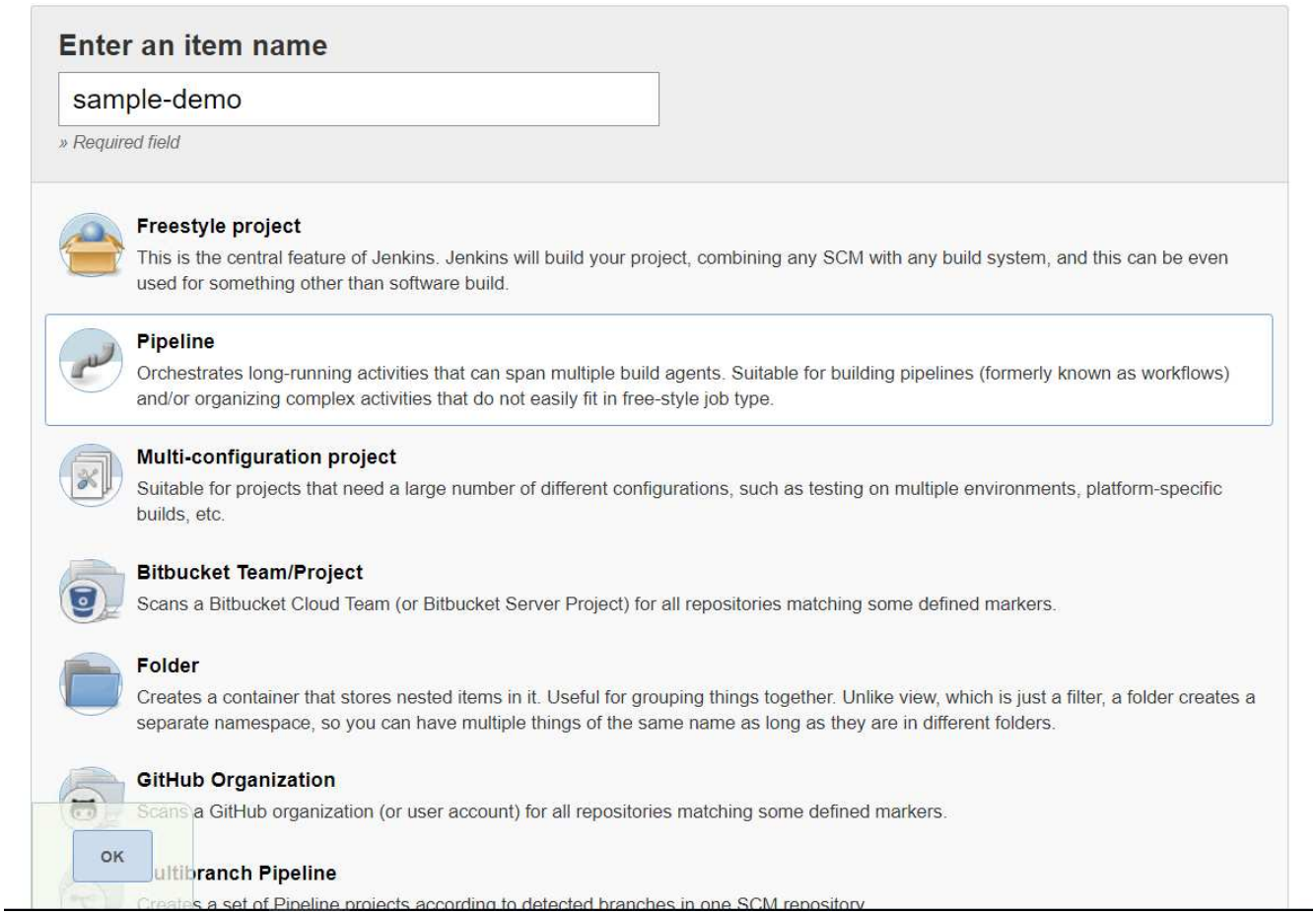
8. Se muestra la página de bienvenida de Jenkins. Dado que utilizamos una compilación de Maven, primero complete la instalación de Maven. Vaya a Administrar Jenkins > Configuración global de herramientas y, luego, en el subtítulo Maven, haga clic en Agregar Maven. Ingrese el nombre de su elección y asegúrese de que la opción Instalar automáticamente esté seleccionada. Haga clic en Guardar.



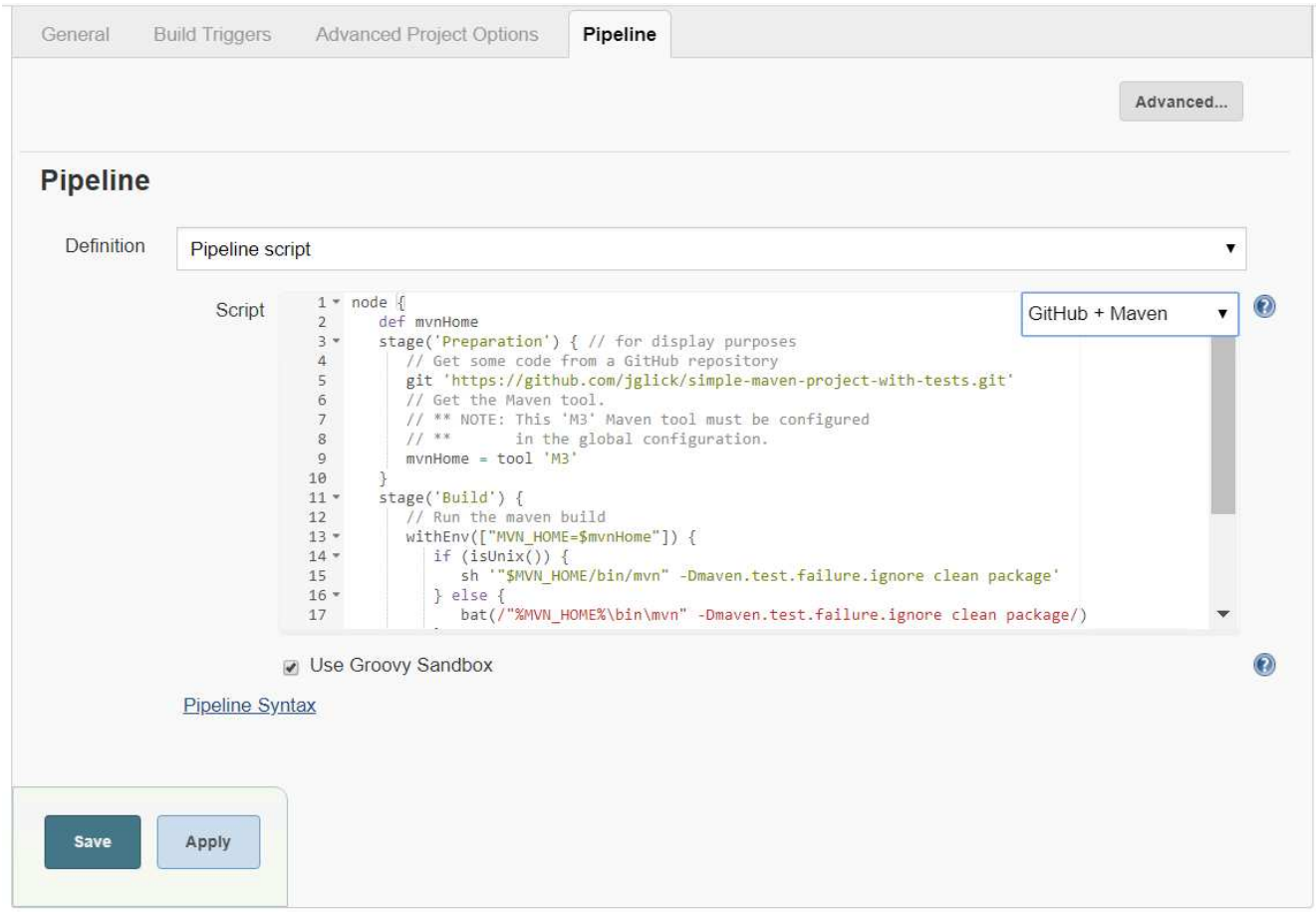
9. Ahora puede crear una canalización para demostrar el flujo de trabajo de CI/CD. En la página de inicio, haga clic en Crear nuevos trabajos o Nuevo elemento en el menú de la izquierda.



10. En la página Crear elemento, ingrese el nombre de su elección, seleccione Pipeline y haga clic en Aceptar.



11. Seleccione la pestaña Pipeline. En el menú desplegable Probar canalización de muestra, seleccione Github + Maven. El código se completa automáticamente. Haga clic en Guardar.



12. Haga clic en Construir ahora para iniciar el desarrollo a través de la fase de preparación, construcción y prueba. Puede tomar varios minutos completar todo el proceso de compilación y mostrar los resultados de la compilación.

13. Siempre que haya cambios en el código, se puede reconstruir el pipeline para parchear la nueva versión del software, permitiendo la integración continua y la entrega continua. Haga clic en Cambios recientes para realizar un seguimiento de los cambios desde la versión anterior.

Configurar multi-tenencia

Configuración de multitenencia en Red Hat OpenShift con NetApp

Muchas organizaciones que ejecutan múltiples aplicaciones o cargas de trabajo en contenedores tienden a implementar un clúster Red Hat OpenShift por aplicación o carga de trabajo. Esto les permite implementar un aislamiento estricto para la aplicación o carga de trabajo, optimizar el rendimiento y reducir las vulnerabilidades de seguridad. Sin embargo, implementar un clúster Red Hat OpenShift separado para cada aplicación plantea su propio conjunto de problemas. Aumenta la sobrecarga operativa al tener que supervisar y gestionar cada clúster por sí solo, aumenta los costos debido a los recursos dedicados para diferentes aplicaciones y dificulta la escalabilidad eficiente.

Para superar estos problemas, se puede considerar ejecutar todas las aplicaciones o cargas de trabajo en un solo clúster Red Hat OpenShift. Pero en una arquitectura de este tipo, el aislamiento de recursos y las vulnerabilidades de seguridad de las aplicaciones son uno de los principales desafíos. Cualquier vulnerabilidad de seguridad en una carga de trabajo podría naturalmente extenderse a otra carga de trabajo, aumentando así la zona de impacto. Además, cualquier utilización abrupta y descontrolada de recursos por parte de una aplicación puede afectar el rendimiento de otra aplicación, porque no existe una política de asignación de recursos predeterminada.

Por lo tanto, las organizaciones buscan soluciones que recojan lo mejor de ambos mundos, por ejemplo, permitiéndoles ejecutar todas sus cargas de trabajo en un solo clúster y, al mismo tiempo, ofrecer los beneficios de un clúster dedicado para cada carga de trabajo.

Una de esas soluciones efectivas es configurar la multitenencia en Red Hat OpenShift. La multitenencia es una arquitectura que permite que varios inquilinos coexistan en el mismo clúster con un aislamiento adecuado de recursos, seguridad, etc. En este contexto, un inquilino puede verse como un subconjunto de los recursos del clúster que están configurados para ser utilizados por un grupo particular de usuarios para un propósito exclusivo. La configuración de múltiples inquilinos en un clúster Red Hat OpenShift ofrece las siguientes ventajas:

- Una reducción en CapEx y OpEx al permitir que se compartan los recursos del clúster
- Menores gastos operativos y de gestión
- Proteger las cargas de trabajo de la contaminación cruzada por brechas de seguridad
- Protección de cargas de trabajo contra una degradación inesperada del rendimiento debido a la contención de recursos

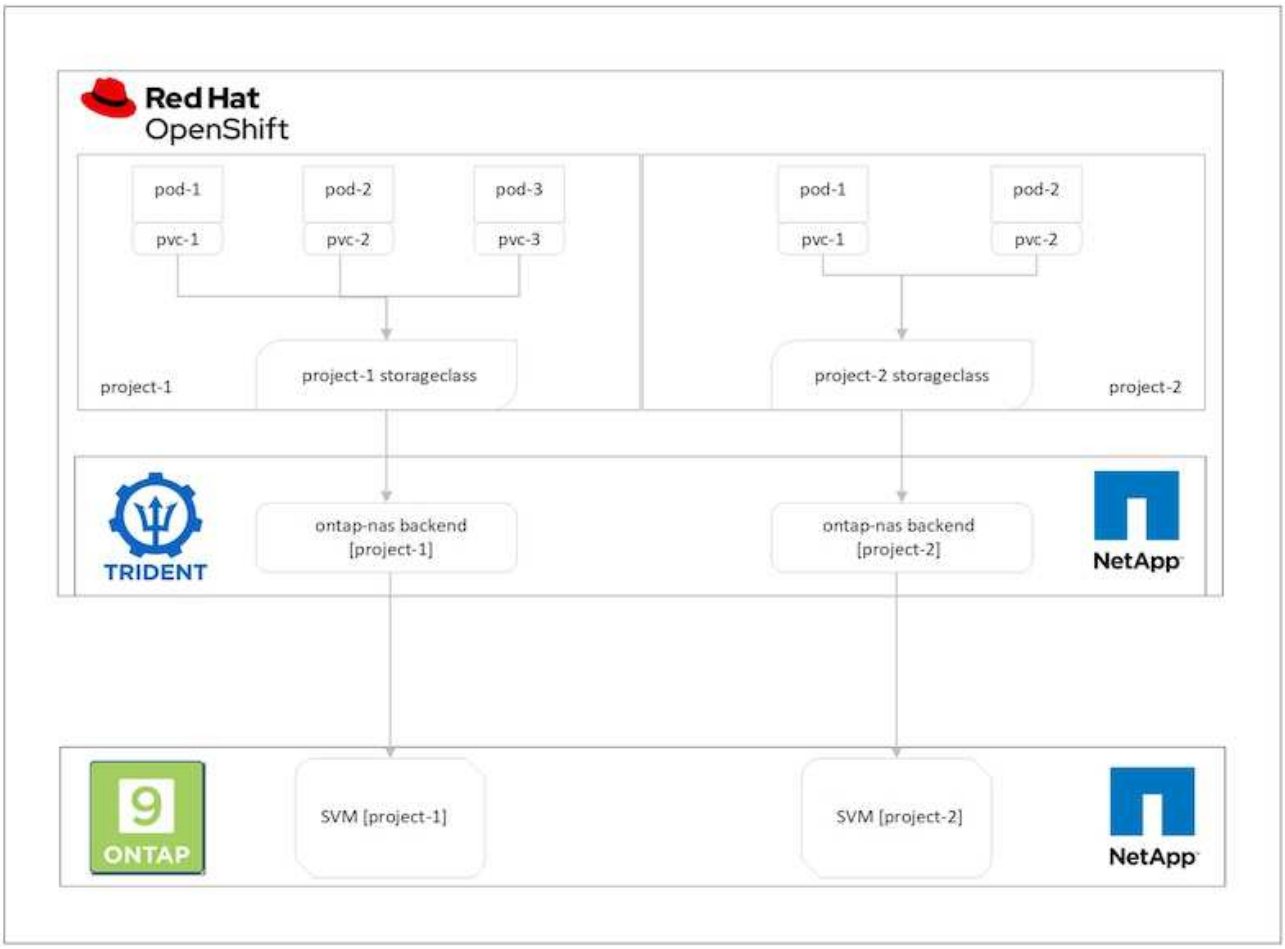
Para un clúster OpenShift multiinquilino completamente realizado, se deben configurar cuotas y restricciones para los recursos del clúster que pertenecen a diferentes grupos de recursos: cómputo, almacenamiento, redes, seguridad, etc. Si bien cubrimos ciertos aspectos de todos los recursos en esta solución, nos centramos en las mejores prácticas para aislar y proteger los datos atendidos o consumidos por múltiples cargas de trabajo en el mismo clúster Red Hat OpenShift mediante la configuración de múltiples inquilinos en recursos de almacenamiento asignados dinámicamente por Trident respaldado por NetApp ONTAP.

Arquitectura

Si bien Red Hat OpenShift y Trident respaldados por NetApp ONTAP no proporcionan aislamiento entre cargas de trabajo de manera predeterminada, ofrecen una amplia gama de características que se pueden usar para configurar la multitenencia. Para comprender mejor el diseño de una solución multiinquilino en un clúster Red Hat OpenShift con Trident respaldado por NetApp ONTAP, consideremos un ejemplo con un conjunto de requisitos y describamos la configuración en torno a él.

Supongamos que una organización ejecuta dos de sus cargas de trabajo en un clúster Red Hat OpenShift como parte de dos proyectos en los que trabajan dos equipos diferentes. Los datos de estas cargas de trabajo residen en PVC que Trident aprovisiona dinámicamente en un backend NAS de NetApp ONTAP . La organización tiene el requisito de diseñar una solución multiinquilino para estas dos cargas de trabajo y aislar los recursos utilizados para estos proyectos para asegurarse de que se mantenga la seguridad y el rendimiento, centrados principalmente en los datos que sirven a esas aplicaciones.

La siguiente figura muestra la solución multiinquilino en un clúster Red Hat OpenShift con Trident respaldado por NetApp ONTAP.



Requisitos tecnológicos

1. Clúster de almacenamiento NetApp ONTAP
2. Clúster Red Hat OpenShift
3. Trident

Red Hat OpenShift – Recursos de clúster

Desde el punto de vista del clúster Red Hat OpenShift, el recurso de nivel superior con el que comenzar es el proyecto. Un proyecto OpenShift puede verse como un recurso de clúster que divide todo el clúster OpenShift en múltiples clústeres virtuales. Por lo tanto, el aislamiento a nivel de proyecto proporciona una base para configurar la multitenencia.

El siguiente paso es configurar RBAC en el clúster. La mejor práctica es tener a todos los desarrolladores que trabajan en un solo proyecto o carga de trabajo configurados en un solo grupo de usuarios en el proveedor de identidad (IdP). Red Hat OpenShift permite la integración de IdP y la sincronización de grupos de usuarios, permitiendo así que los usuarios y grupos del IdP se importen al clúster. Esto ayuda a los administradores del clúster a segregar el acceso a los recursos del clúster dedicados a un proyecto a un grupo o grupos de usuarios que trabajan en ese proyecto, restringiendo así el acceso no autorizado a cualquier recurso del clúster. Para obtener más información sobre la integración de IdP con Red Hat OpenShift, consulte la documentación "aquí" .

ONTAP de NetApp

Es importante aislar el almacenamiento compartido que funciona como proveedor de almacenamiento persistente para un clúster de Red Hat OpenShift para asegurarse de que los volúmenes creados en el almacenamiento para cada proyecto aparezcan para los hosts como si se hubieran creado en un almacenamiento separado. Para ello, cree tantas SVM (máquinas virtuales de almacenamiento) en NetApp ONTAP como proyectos o cargas de trabajo y dedique cada SVM a una carga de trabajo.

Trident

Una vez que tenga diferentes SVM para diferentes proyectos creados en NetApp ONTAP, debe asignar cada SVM a un backend Trident diferente. La configuración del backend en Trident impulsa la asignación de almacenamiento persistente a los recursos del clúster OpenShift y requiere que se asignen los detalles de la SVM. Este debería ser el controlador de protocolo para el backend como mínimo. De manera opcional, permite definir cómo se aprovisionan los volúmenes en el almacenamiento y establecer límites para el tamaño de los volúmenes o el uso de agregados, etc. Los detalles sobre la definición de los backends de Trident se pueden encontrar "aquí" .

Red Hat OpenShift: recursos de almacenamiento

Después de configurar los backends de Trident , el siguiente paso es configurar StorageClasses. Configure tantas clases de almacenamiento como backends haya, proporcionando a cada clase de almacenamiento acceso para activar volúmenes solo en un backend. Podemos asignar StorageClass a un backend Trident particular utilizando el parámetro storagePools al definir la clase de almacenamiento. Los detalles para definir una clase de almacenamiento se pueden encontrar "aquí" . Por lo tanto, existe una asignación uno a uno desde StorageClass al backend de Trident que apunta a una SVM. Esto garantiza que todas las reclamaciones de almacenamiento a través de la StorageClass asignada a ese proyecto sean atendidas únicamente por la SVM dedicada a ese proyecto.

Debido a que las clases de almacenamiento no son recursos con espacios de nombres, ¿cómo garantizamos que las reclamaciones de almacenamiento a la clase de almacenamiento de un proyecto por parte de pods en otro espacio de nombres o proyecto sean rechazadas? La respuesta es utilizar ResourceQuotas. ResourceQuotas son objetos que controlan el uso total de recursos por proyecto. Puede limitar el número y la cantidad total de recursos que pueden consumir los objetos del proyecto. Casi todos los recursos de un proyecto se pueden limitar mediante el uso de ResourceQuotas y su uso eficiente puede ayudar a las organizaciones a reducir costos y cortes debido al exceso de aprovisionamiento o consumo excesivo de recursos. Consulte la documentación "aquí" Para más información.

Para este caso de uso, necesitamos limitar que los pods de un proyecto en particular reclamen almacenamiento de clases de almacenamiento que no están dedicadas a su proyecto. Para hacer eso, necesitamos limitar las reclamaciones de volumen persistente para otras clases de almacenamiento configurando `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` a 0. Además, un administrador de clúster debe asegurarse de que los desarrolladores de un proyecto no tengan acceso para modificar las ResourceQuotas.

Configuración

En cualquier solución multiinquilino, ningún usuario puede tener acceso a más recursos del clúster de los necesarios. Por lo tanto, todo el conjunto de recursos que se deben configurar como parte de la configuración multiinquilino se divide entre el administrador del clúster, el administrador de almacenamiento y los desarrolladores que trabajan en cada proyecto.

En la siguiente tabla se describen las diferentes tareas que deben realizar los distintos usuarios:

Role	Tareas
Administrador del clúster	Crear proyectos para diferentes aplicaciones o cargas de trabajo
	Crear roles de clúster y enlaces de roles para el administrador de almacenamiento
	Crear roles y vinculaciones de roles para desarrolladores que asignen acceso a proyectos específicos
	[Opcional] Configurar proyectos para programar pods en nodos específicos
Administrador de almacenamiento	Crear SVM en NetApp ONTAP
	Crear backends de Trident
	Crear clases de almacenamiento
	Crear cuotas de recursos de almacenamiento
Desarrolladores	Validar el acceso para crear o parchar PVC o pods en el proyecto asignado
	Validar el acceso para crear o parchar PVC o pods en otro proyecto
	Validar el acceso para ver o editar proyectos, cuotas de recursos y clases de almacenamiento

Configuración

A continuación se detallan los requisitos previos para configurar la multitenencia en Red Hat OpenShift con NetApp.

Prerrequisitos

- Clúster NetApp ONTAP
- Clúster Red Hat OpenShift
- Trident instalado en el clúster
- Estación de trabajo de administración con herramientas tridentctl y oc instaladas y agregadas a $PATH
- Acceso de administrador a ONTAP
- Acceso de administrador de clúster al clúster OpenShift
- El clúster está integrado con el proveedor de identidad
- El proveedor de identidad está configurado para distinguir de manera eficiente entre usuarios de diferentes equipos.

Configuración: tareas de administración del clúster

El administrador del clúster de Red Hat OpenShift realiza las siguientes tareas:

1. Inicie sesión en el clúster Red Hat OpenShift como administrador del clúster.

2. Crea dos proyectos correspondientes a proyectos diferentes.

```
oc create namespace project-1
oc create namespace project-2
```

3. Cree el rol de desarrollador para el proyecto 1.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-1
  name: developer-project-1
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
```

```
      - podtemplates
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

La definición de rol proporcionada en esta sección es sólo un ejemplo. Los roles de desarrollador deben definirse en función de los requisitos del usuario final.

1. De manera similar, cree roles de desarrollador para el proyecto 2.
2. Todos los recursos de almacenamiento de OpenShift y NetApp generalmente son administrados por un administrador de almacenamiento. El acceso de los administradores de almacenamiento está controlado por el rol de operador de Trident que se crea cuando se instala Trident . Además de esto, el administrador de almacenamiento también requiere acceso a ResourceQuotas para controlar cómo se consume el almacenamiento.
3. Cree un rol para administrar ResourceQuotas en todos los proyectos del clúster para adjuntarlo al administrador de almacenamiento.

```
cat << EOF | oc create -f -
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: resource-quotas-role
rules:
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - resourcequotas
  - verbs:
      - '*'
    apiGroups:
      - quota.openshift.io
    resources:
      - '*'
EOF
```

4. Asegúrese de que el clúster esté integrado con el proveedor de identidad de la organización y que los grupos de usuarios estén sincronizados con los grupos del clúster. El siguiente ejemplo muestra que el proveedor de identidad se ha integrado con el clúster y se ha sincronizado con los grupos de usuarios.

```
$ oc get groups
NAME                          USERS
ocp-netapp-storage-admins     ocp-netapp-storage-admin
ocp-project-1                 ocp-project-1-user
ocp-project-2                 ocp-project-2-user
```

1. Configurar ClusterRoleBindings para administradores de almacenamiento.

```
cat << EOF | oc create -f -
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-trident-operator
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: trident-operator
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: netapp-storage-admin-resource-quotas-cr
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-netapp-storage-admins
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: resource-quotas-role
EOF
```

Para los administradores de almacenamiento, se deben vincular dos roles: trident-operator y resource-quotas.

1. Cree RoleBindings para desarrolladores que vinculen el rol desarrollador-proyecto-1 al grupo correspondiente (ocp-proyecto-1) en proyecto-1.

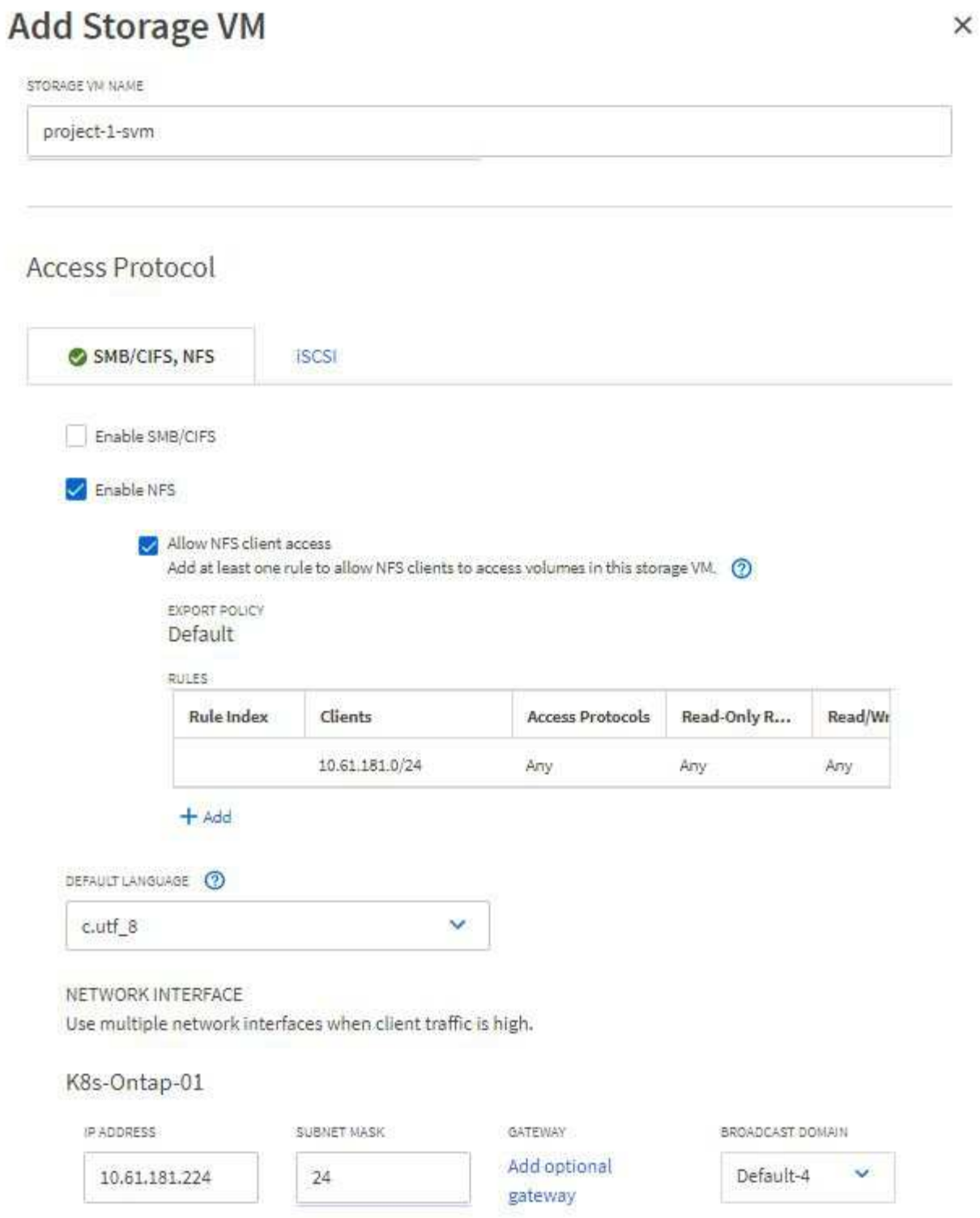
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-1
EOF
```

2. De manera similar, cree RoleBindings para los desarrolladores que vinculen los roles de desarrollador al grupo de usuarios correspondiente en el proyecto 2.

Configuración: Tareas de administración de almacenamiento

Los siguientes recursos deben ser configurados por un administrador de almacenamiento:

1. Inicie sesión en el clúster NetApp ONTAP como administrador.
2. Vaya a Almacenamiento > Máquinas virtuales de almacenamiento y haga clic en Agregar. Cree dos SVM, uno para el proyecto 1 y otro para el proyecto 2, proporcionando los detalles requeridos. También cree una cuenta vsadmin para administrar el SVM y sus recursos.



1. Inicie sesión en el clúster Red Hat OpenShift como administrador de almacenamiento.
2. Cree el backend para el proyecto 1 y asígnelo al SVM dedicado al proyecto. NetApp recomienda utilizar la cuenta vsadmin de SVM para conectar el backend a SVM en lugar de utilizar el administrador de clúster de ONTAP .

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_1",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.224",
    "svm": "project-1-svm",
    "username": "vsadmin",
    "password": "NetApp123"
}
EOF
```

Estamos utilizando el controlador ontap-nas para este ejemplo. Utilice el controlador apropiado al crear el backend según el caso de uso.

Suponemos que Trident está instalado en el proyecto trident.

1. De manera similar, cree el backend Trident para el proyecto 2 y asígnelo al SVM dedicado al proyecto 2.
2. A continuación, cree las clases de almacenamiento. Cree la clase de almacenamiento para el proyecto-1 y configúrela para utilizar los grupos de almacenamiento del backend dedicados al proyecto-1 configurando el parámetro storagePools.

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-1-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_1:.*"
EOF
```

3. Del mismo modo, cree una clase de almacenamiento para el proyecto 2 y configúrela para utilizar los grupos de almacenamiento del backend dedicados al proyecto 2.
4. Cree una ResourceQuota para restringir los recursos en el proyecto 1 que solicitan almacenamiento de clases de almacenamiento dedicadas a otros proyectos.

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-1-sc-rq
  namespace: project-1
spec:
  hard:
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

5. De manera similar, cree un ResourceQuota para restringir los recursos en el proyecto 2 que solicitan almacenamiento de clases de almacenamiento dedicadas a otros proyectos.

Validación

Para validar la arquitectura multiinquilino que se configuró en los pasos anteriores, complete los siguientes pasos:

Validar el acceso para crear PVC o pods en el proyecto asignado

1. Inicie sesión como ocp-project-1-user, desarrollador en proyecto-1.
2. Verifique el acceso para crear un nuevo proyecto.

```
oc create ns sub-project-1
```

3. Cree una PVC en el proyecto-1 utilizando la clase de almacenamiento asignada al proyecto-1.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. Verifique el PV asociado con el PVC.

```
oc get pv
```

5. Valide que el PV y su volumen se creen en una SVM dedicada al proyecto 1 en NetApp ONTAP.

```
volume show -vserver project-1-svm
```

6. Cree un pod en el proyecto-1 y monte el PVC creado en el paso anterior.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  volumes:
    - name: test-pvc-project-1
      persistentVolumeClaim:
       claimName: test-pvc-project-1
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: test-pvc-project-1
EOF
```

7. Verifique si el pod está ejecutándose y si montó el volumen.

```
oc describe pods test-pvc-pod -n project-1
```

Validar el acceso para crear PVC o pods en otro proyecto o utilizar recursos dedicados a otro proyecto

1. Inicie sesión como ocp-project-1-user, desarrollador en proyecto-1.
2. Cree una PVC en el proyecto-1 utilizando la clase de almacenamiento asignada al proyecto-2.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-1-sc-2
  namespace: project-1
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-2-sc
EOF
```

3. Crea un PVC en el proyecto-2.

```
cat << EOF | oc create -f -
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc-project-2-sc-1
  namespace: project-2
  annotations:
    trident.netapp.io/reclaimPolicy: Retain
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: project-1-sc
EOF
```

4. Asegúrese de que los PVC `test-pvc-project-1-sc-2` y `test-pvc-project-2-sc-1` no fueron creados

```
oc get pvc -n project-1
oc get pvc -n project-2
```

5. Crear un pod en el proyecto-2.

```
cat << EOF | oc create -f -
kind: Pod
apiVersion: v1
metadata:
  name: test-pvc-pod
  namespace: project-1
spec:
  containers:
    - name: test-container
      image: nginx
      ports:
        - containerPort: 80
          name: "http-server"
EOF
```

Validar el acceso para ver y editar proyectos, cuotas de recursos y clases de almacenamiento

1. Inicie sesión como ocp-project-1-user, desarrollador en proyecto-1.
2. Verificar el acceso para crear nuevos proyectos.

```
oc create ns sub-project-1
```

3. Validar el acceso para ver proyectos.

```
oc get ns
```

4. Verifique si el usuario puede ver o editar ResourceQuotas en el proyecto-1.

```
oc get resourcequotas -n project-1
oc edit resourcequotas project-1-sc-rq -n project-1
```

5. Validar que el usuario tenga acceso para ver las clases de almacenamiento.

```
oc get sc
```

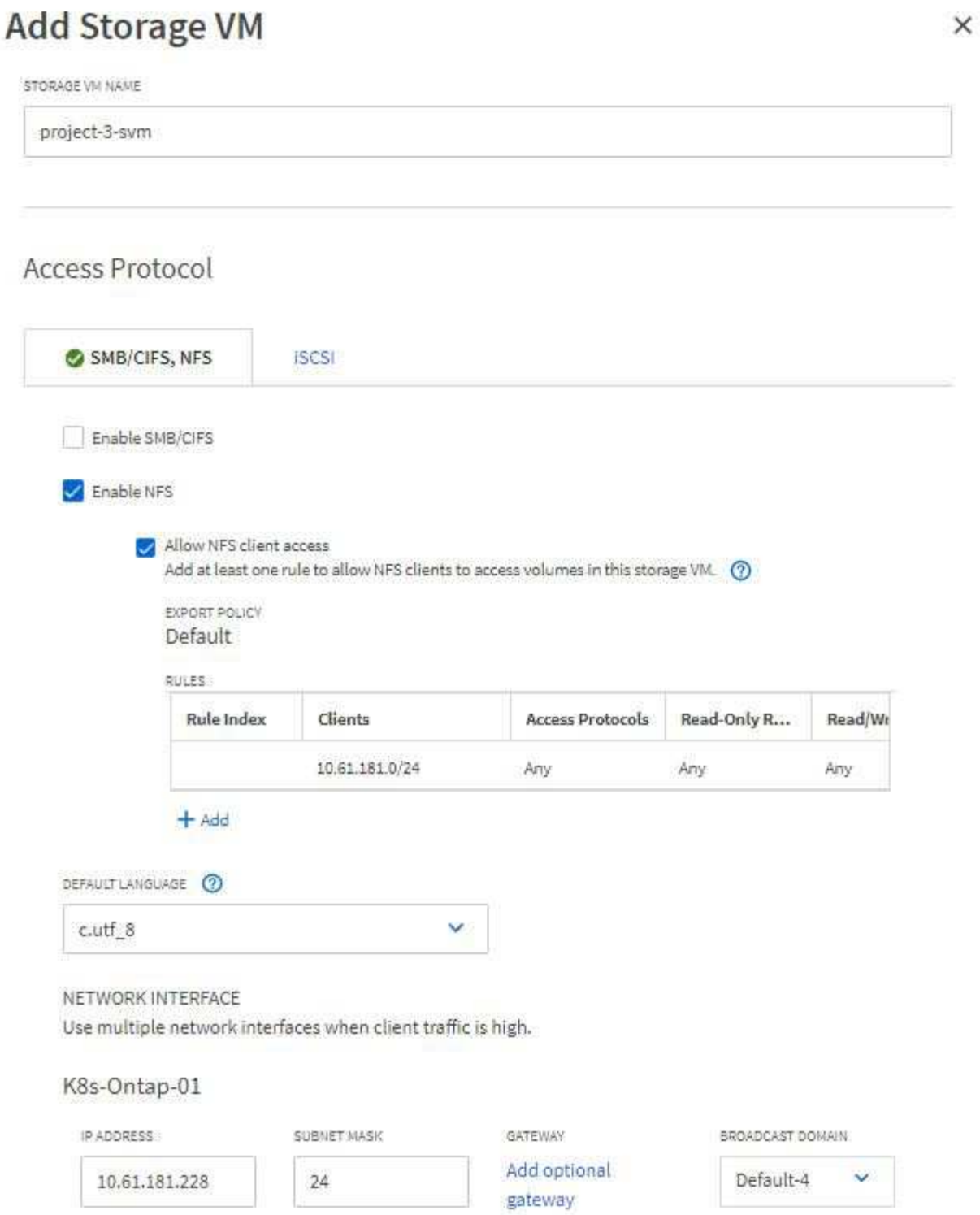
6. Compruebe el acceso para describir las clases de almacenamiento.
7. Validar el acceso del usuario para editar las clases de almacenamiento.

```
oc edit sc project-1-sc
```

Escalado: agregar más proyectos

En una configuración multiinquilino, agregar nuevos proyectos con recursos de almacenamiento requiere una configuración adicional para garantizar que no se viole la multiinquilino. Para agregar más proyectos en un clúster multiinquilino, complete los siguientes pasos:

1. Inicie sesión en el clúster NetApp ONTAP como administrador de almacenamiento.
2. Navegar a `Storage → Storage VMs` y haga clic `Add` . Cree una nueva SVM dedicada al proyecto 3. También cree una cuenta vsadmin para administrar el SVM y sus recursos.



1. Inicie sesión en el clúster Red Hat OpenShift como administrador del clúster.
2. Crear un nuevo proyecto.

```
oc create ns project-3
```

3. Asegúrese de que el grupo de usuarios del proyecto 3 esté creado en IdP y sincronizado con el clúster OpenShift.

```
oc get groups
```

4. Crear el rol de desarrollador para el proyecto 3.

```
cat << EOF | oc create -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  namespace: project-3
  name: developer-project-3
rules:
  - verbs:
      - '*'
    apiGroups:
      - apps
      - batch
      - autoscaling
      - extensions
      - networking.k8s.io
      - policy
      - apps.openshift.io
      - build.openshift.io
      - image.openshift.io
      - ingress.operator.openshift.io
      - route.openshift.io
      - snapshot.storage.k8s.io
      - template.openshift.io
    resources:
      - '*'
  - verbs:
      - '*'
    apiGroups:
      - ''
    resources:
      - bindings
      - configmaps
      - endpoints
      - events
      - persistentvolumeclaims
      - pods
      - pods/log
      - pods/attach
```

```
      - podtemplates
      - replicationcontrollers
      - services
      - limitranges
      - namespaces
      - componentstatuses
      - nodes
  - verbs:
      - '*'
    apiGroups:
      - trident.netapp.io
    resources:
      - tridentsnapshots
EOF
```

La definición de rol proporcionada en esta sección es sólo un ejemplo. El rol de desarrollador debe definirse en función de los requisitos del usuario final.

1. Cree un RoleBinding para los desarrolladores en el proyecto 3 que vincule el rol de desarrollador-proyecto-3 al grupo correspondiente (ocp-proyecto-3) en el proyecto-3.

```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-3-developer
  namespace: project-3
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-3
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: developer-project-3
EOF
```

2. Inicie sesión en el clúster Red Hat OpenShift como administrador de almacenamiento
3. Cree un backend Trident y asígnelo al SVM dedicado al proyecto 3. NetApp recomienda utilizar la cuenta vsadmin de SVM para conectar el backend a SVM en lugar de utilizar el administrador de clúster de ONTAP .

```
cat << EOF | tridentctl -n trident create backend -f
{
    "version": 1,
    "storageDriverName": "ontap-nas",
    "backendName": "nfs_project_3",
    "managementLIF": "172.21.224.210",
    "dataLIF": "10.61.181.228",
    "svm": "project-3-svm",
    "username": "vsadmin",
    "password": "NetApp!23"
}
EOF
```

Estamos utilizando el controlador ontap-nas para este ejemplo. Utilice el controlador apropiado para crear el backend según el caso de uso.

Suponemos que Trident está instalado en el proyecto trident.

1. Cree la clase de almacenamiento para el proyecto 3 y configúrela para utilizar los grupos de almacenamiento del backend dedicados al proyecto 3.

```
cat << EOF | oc create -f -
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: project-3-sc
provisioner: csi.trident.netapp.io
parameters:
  backendType: ontap-nas
  storagePools: "nfs_project_3:.*"
EOF
```

2. Cree una ResourceQuota para restringir los recursos en el proyecto 3 que solicitan almacenamiento de clases de almacenamiento dedicadas a otros proyectos.

```
cat << EOF | oc create -f -
kind: ResourceQuota
apiVersion: v1
metadata:
  name: project-3-sc-rq
  namespace: project-3
spec:
  hard:
    project-1-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
    project-2-sc.storageclass.storage.k8s.io/persistentvolumeclaims: 0
EOF
```

3. Parchee ResourceQuotas en otros proyectos para restringir que los recursos en esos proyectos accedan al almacenamiento desde la clase de almacenamiento dedicada al proyecto 3.

```
oc patch resourcequotas project-1-sc-rq -n project-1 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
oc patch resourcequotas project-2-sc-rq -n project-2 --patch
'{"spec":{"hard":{ "project-3-
sc.storageclass.storage.k8s.io/persistentvolumeclaims": 0}}}'
```

Gestión avanzada de clústeres para Kubernetes

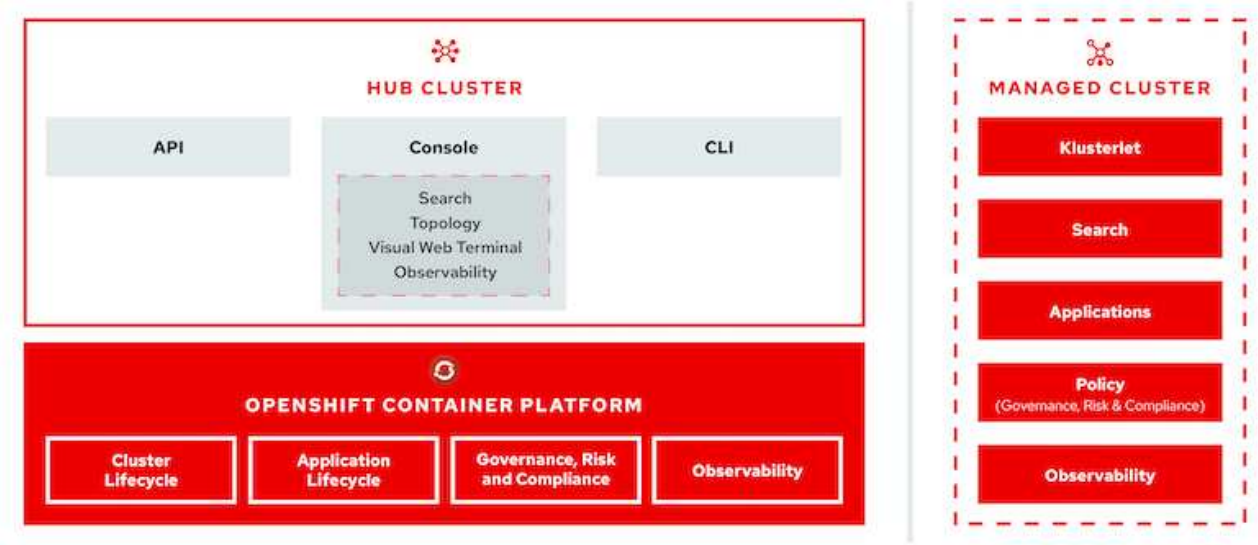
Gestión avanzada de clústeres para Kubernetes: Red Hat OpenShift con NetApp - Descripción general

A medida que una aplicación en contenedores pasa del desarrollo a la producción, muchas organizaciones requieren múltiples clústeres Red Hat OpenShift para respaldar las pruebas y la implementación de esa aplicación. Junto con esto, las organizaciones generalmente alojan múltiples aplicaciones o cargas de trabajo en clústeres de OpenShift. Por lo tanto, cada organización termina administrando un conjunto de clústeres, y los administradores de OpenShift deben enfrentar el desafío adicional de administrar y mantener múltiples clústeres en una variedad de entornos que abarcan múltiples centros de datos locales y nubes públicas. Para abordar estos desafíos, Red Hat presentó la gestión avanzada de clústeres para Kubernetes.

Red Hat Advanced Cluster Management for Kubernetes le permite realizar las siguientes tareas:

1. Cree, importe y administre múltiples clústeres en centros de datos y nubes públicas
2. Implemente y administre aplicaciones o cargas de trabajo en múltiples clústeres desde una única consola
3. Supervisar y analizar la salud y el estado de los diferentes recursos del clúster
4. Supervisar y hacer cumplir el cumplimiento de la seguridad en múltiples clústeres

Red Hat Advanced Cluster Management for Kubernetes se instala como un complemento a un clúster Red Hat OpenShift y utiliza este clúster como controlador central para todas sus operaciones. Este clúster se conoce como clúster concentrador y expone un plano de administración para que los usuarios se conecten a la Administración avanzada de clúster. Todos los demás clústeres OpenShift que se importan o crean a través de la consola de administración avanzada de clústeres son administrados por el clúster central y se denominan clústeres administrados. Instala un agente llamado Klusterlet en los clústeres administrados para conectarlos al clúster central y atender las solicitudes para diferentes actividades relacionadas con la administración del ciclo de vida del clúster, la administración del ciclo de vida de las aplicaciones, la observabilidad y el cumplimiento de la seguridad.



Para obtener más información, consulte la documentación. "aquí" .

Implementar ACM para Kubernetes

Implementar la gestión avanzada de clústeres para Kubernetes

Esta sección cubre la gestión avanzada de clústeres para Kubernetes en Red Hat OpenShift con NetApp.

Prerrequisitos

1. Un clúster Red Hat OpenShift (superior a la versión 4.5) para el clúster central
2. Clústeres Red Hat OpenShift (superiores a la versión 4.4.3) para clústeres administrados
3. Acceso de administrador de clúster al clúster Red Hat OpenShift
4. Una suscripción a Red Hat para la gestión avanzada de clústeres para Kubernetes

La gestión avanzada de clústeres es un complemento para el clúster OpenShift, por lo que existen ciertos requisitos y restricciones en los recursos de hardware según las funciones utilizadas en el concentrador y los clústeres administrados. Es necesario tener en cuenta estas cuestiones a la hora de dimensionar los clústeres. Ver la documentación "aquí" Para más detalles.

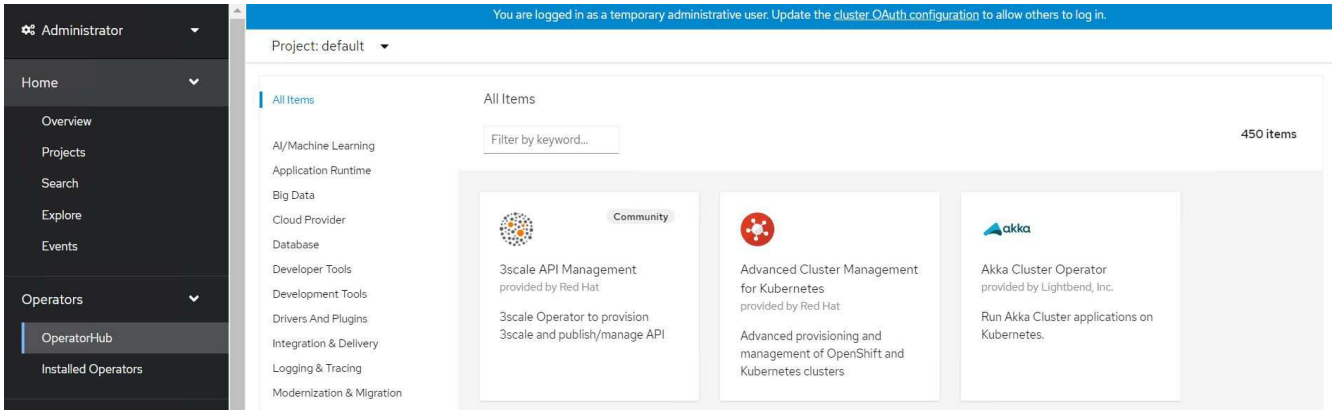
De manera opcional, si el clúster central tiene nodos dedicados para alojar componentes de infraestructura y

desea instalar recursos de Administración avanzada de clúster solo en esos nodos, debe agregar tolerancias y selectores a esos nodos según corresponda. Para más detalles, consulte la documentación. "aquí" .

Implementar la gestión avanzada de clústeres para Kubernetes

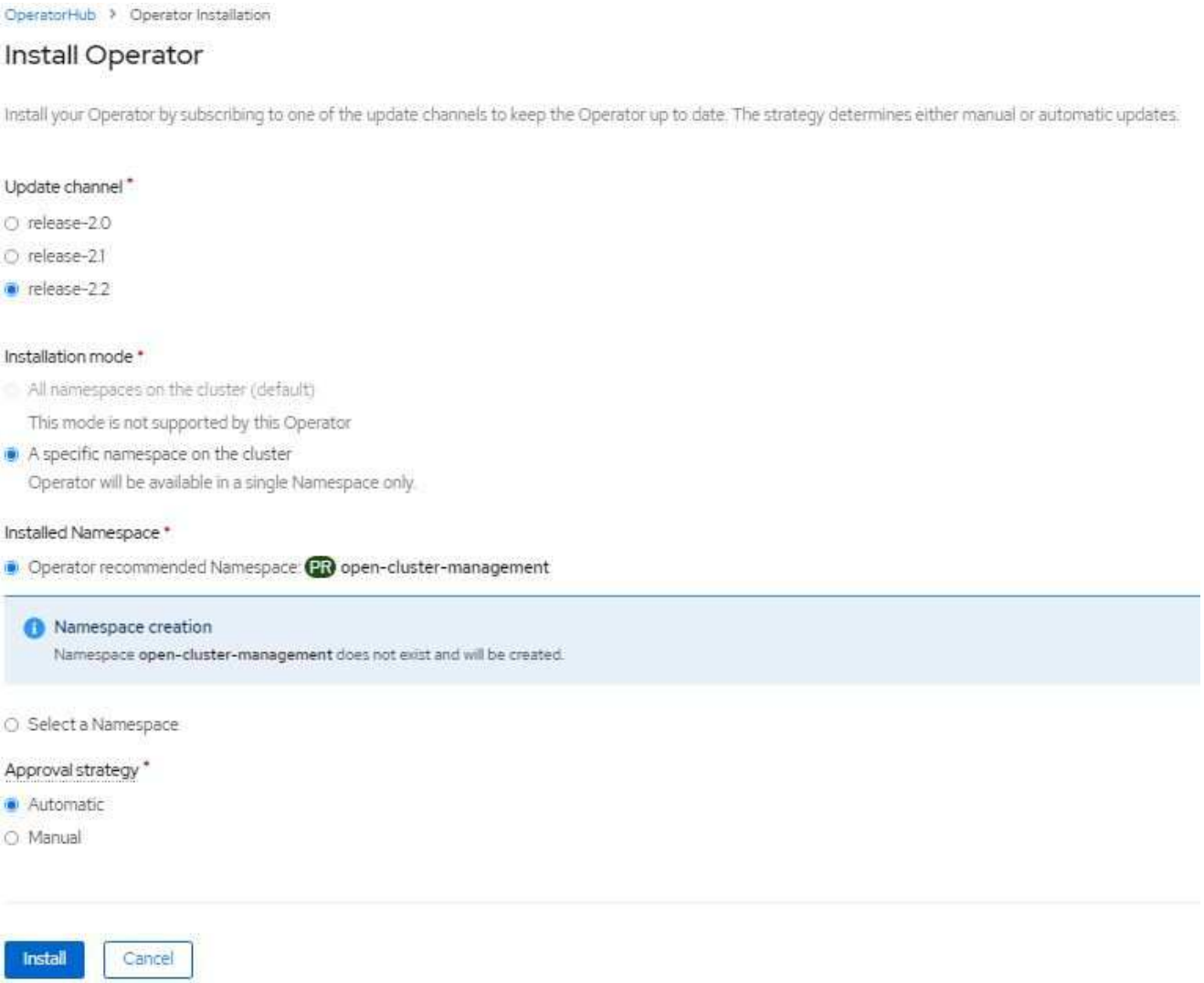
Para instalar Advanced Cluster Management for Kubernetes en un clúster de OpenShift, complete los siguientes pasos:

1. Seleccione un clúster OpenShift como clúster central e inicie sesión en él con privilegios de administrador del clúster.
2. Vaya a Operadores > Centro de operadores y busque Administración avanzada de clústeres para Kubernetes.

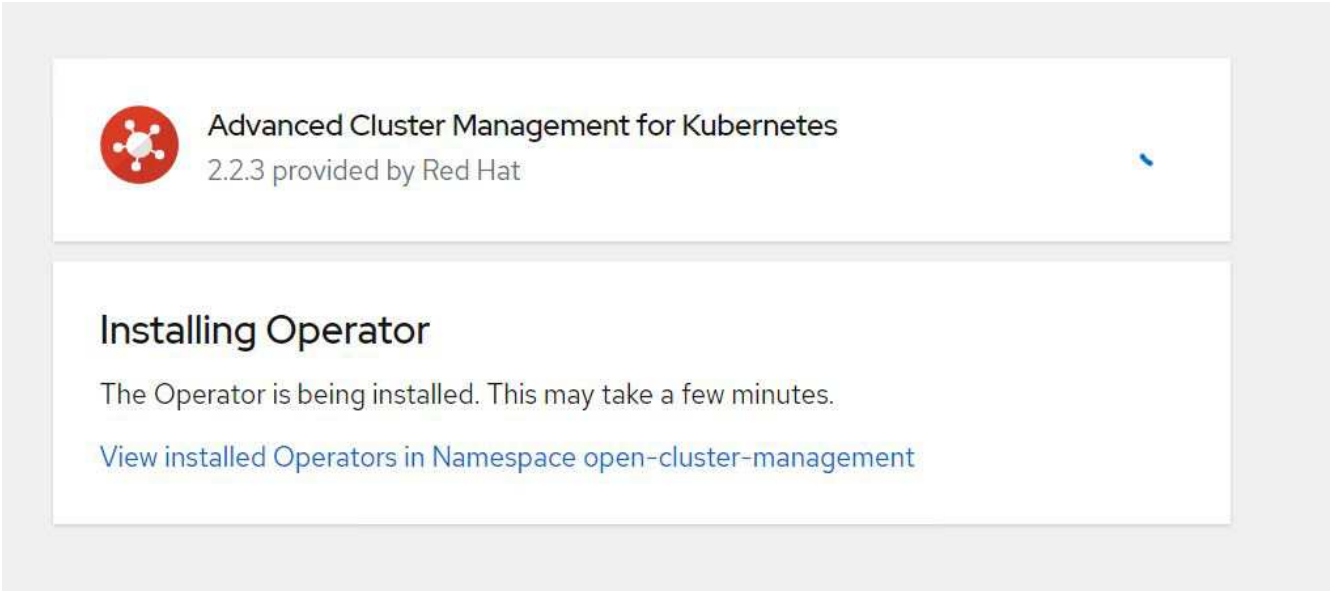


3. Seleccione Administración avanzada de clústeres para Kubernetes y haga clic en Instalar.

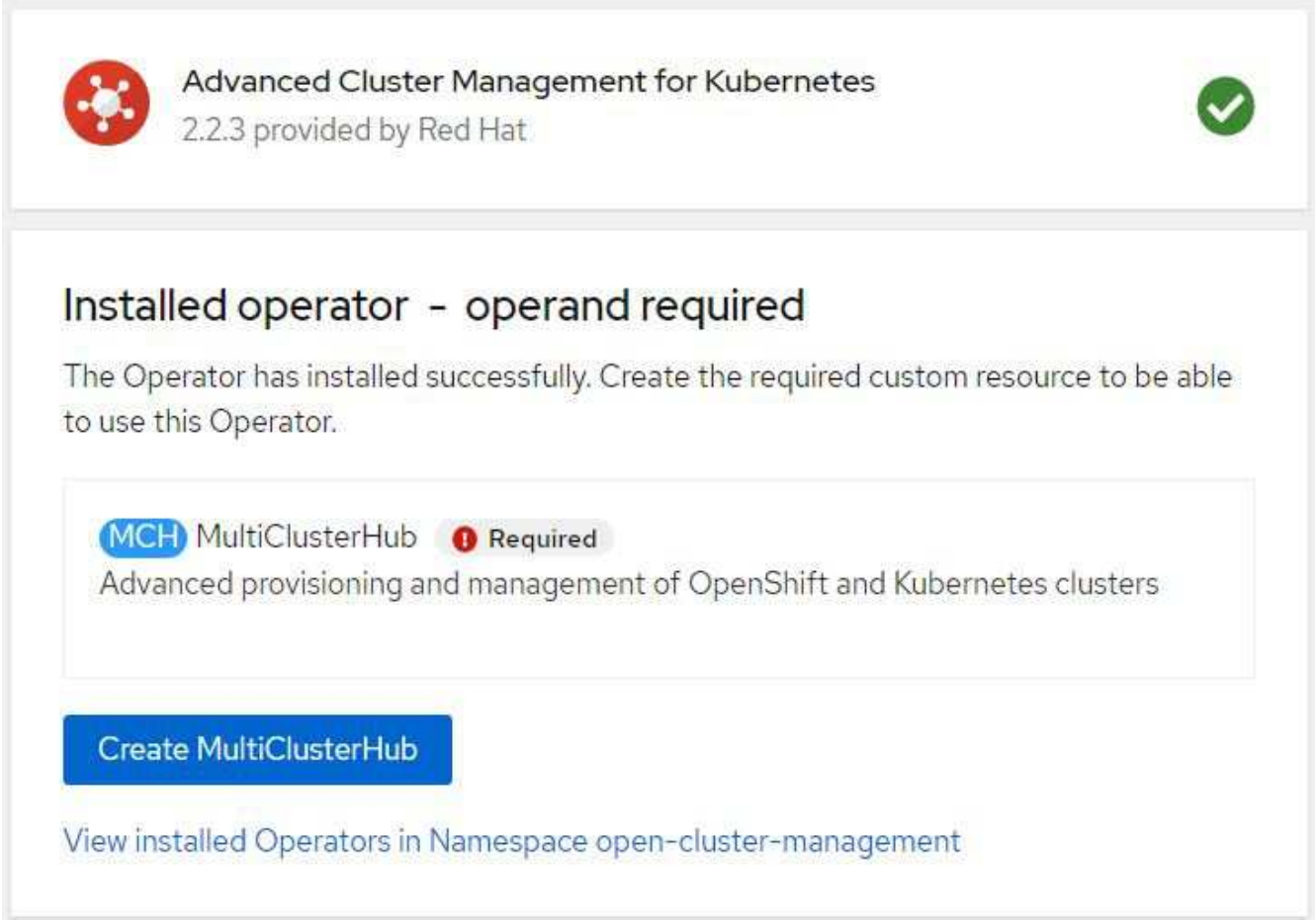
4. En la pantalla Instalar operador, proporcione los detalles necesarios (NetApp recomienda conservar los parámetros predeterminados) y haga clic en Instalar.



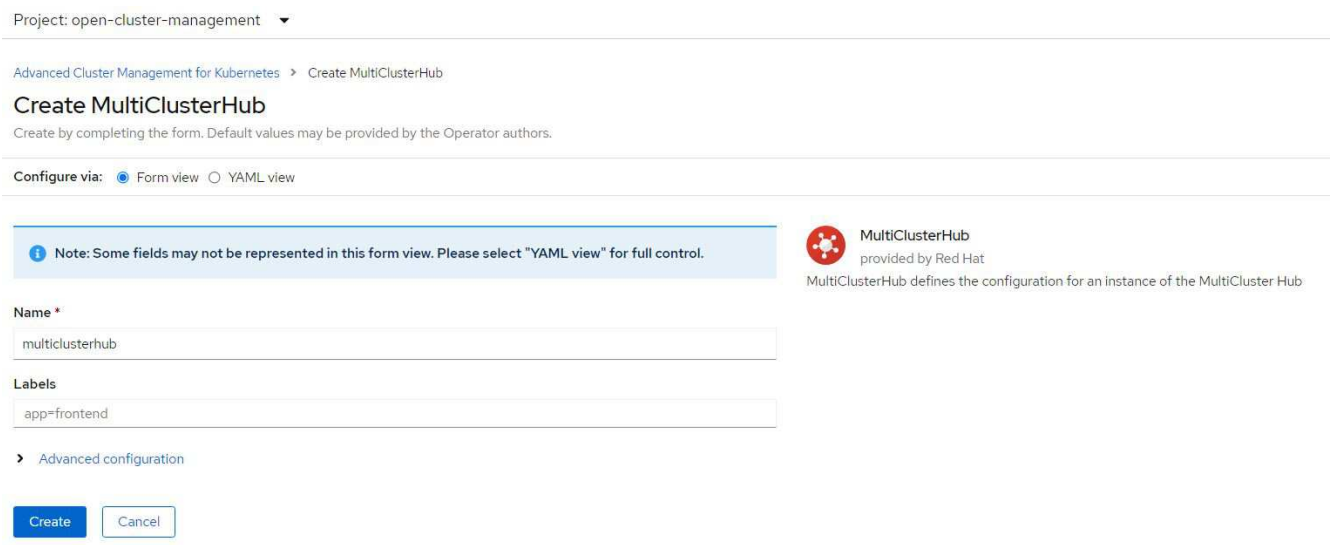
5. Espere a que se complete la instalación del operador.



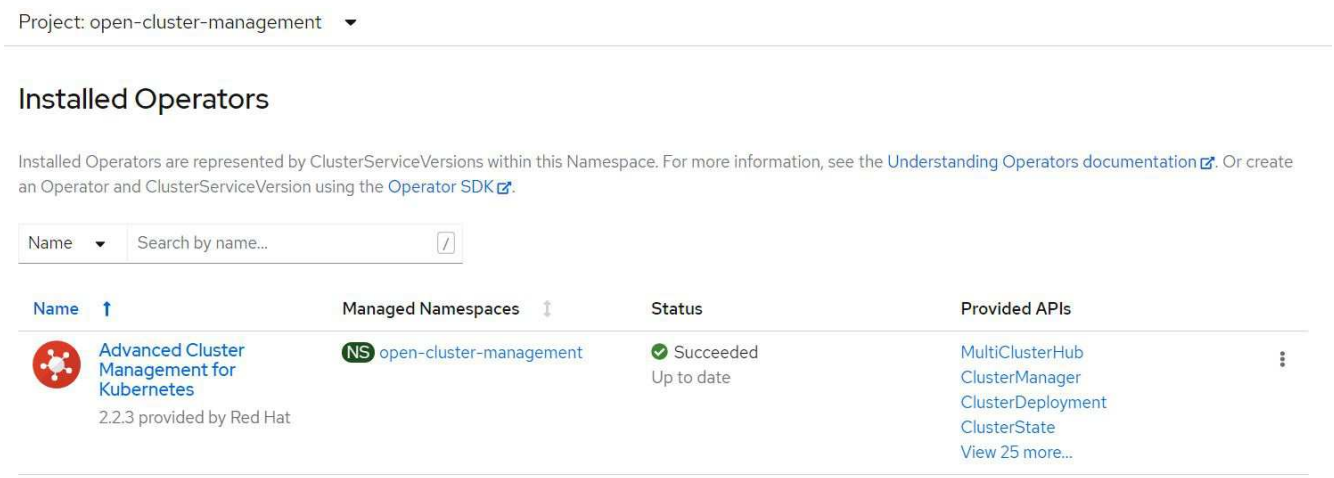
6. Una vez instalado el operador, haga clic en Crear MultiClusterHub.



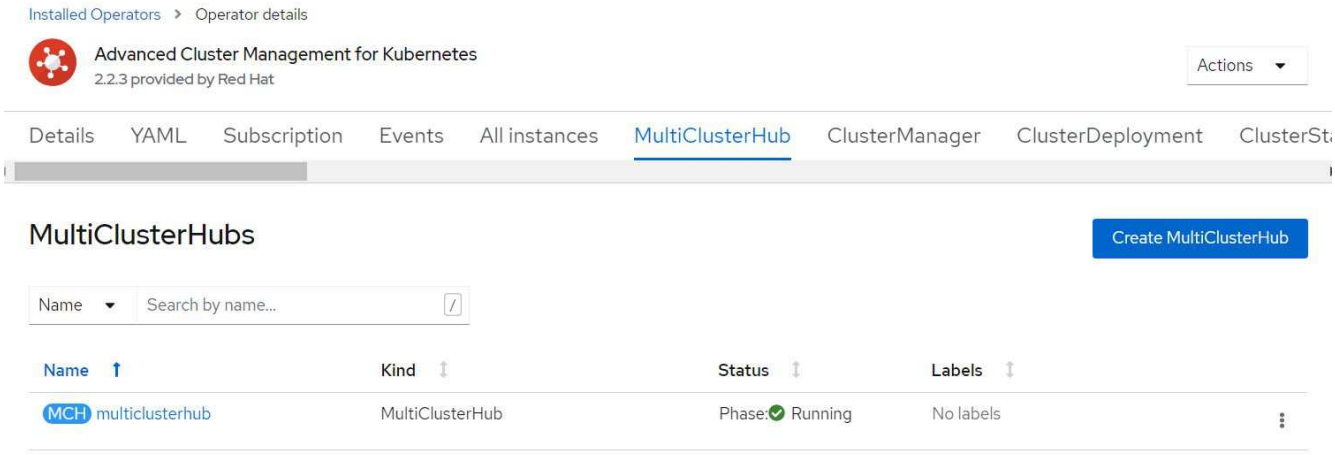
7. En la pantalla Crear MultiClusterHub, haga clic en Crear después de proporcionar los detalles. Esto inicia la instalación de un concentrador multiclúster.



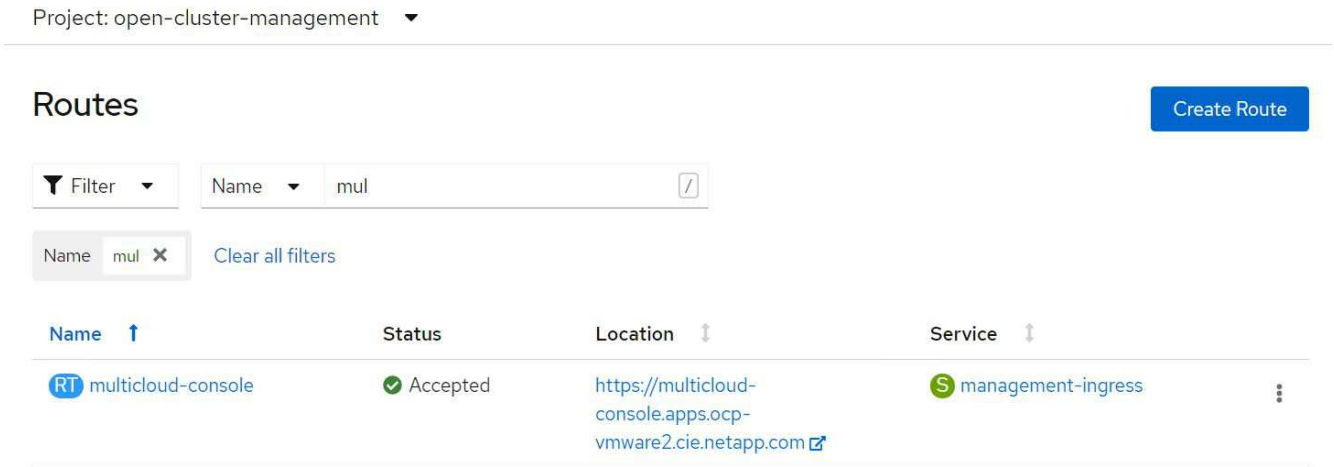
8. Una vez que todos los pods pasan al estado En ejecución en el espacio de nombres open-cluster-management y el operador pasa al estado Correcto, se instala Advanced Cluster Management para Kubernetes.



9. La instalación del concentrador tarda un tiempo en completarse y, una vez realizada, el concentrador MultiCluster pasa al estado en ejecución.



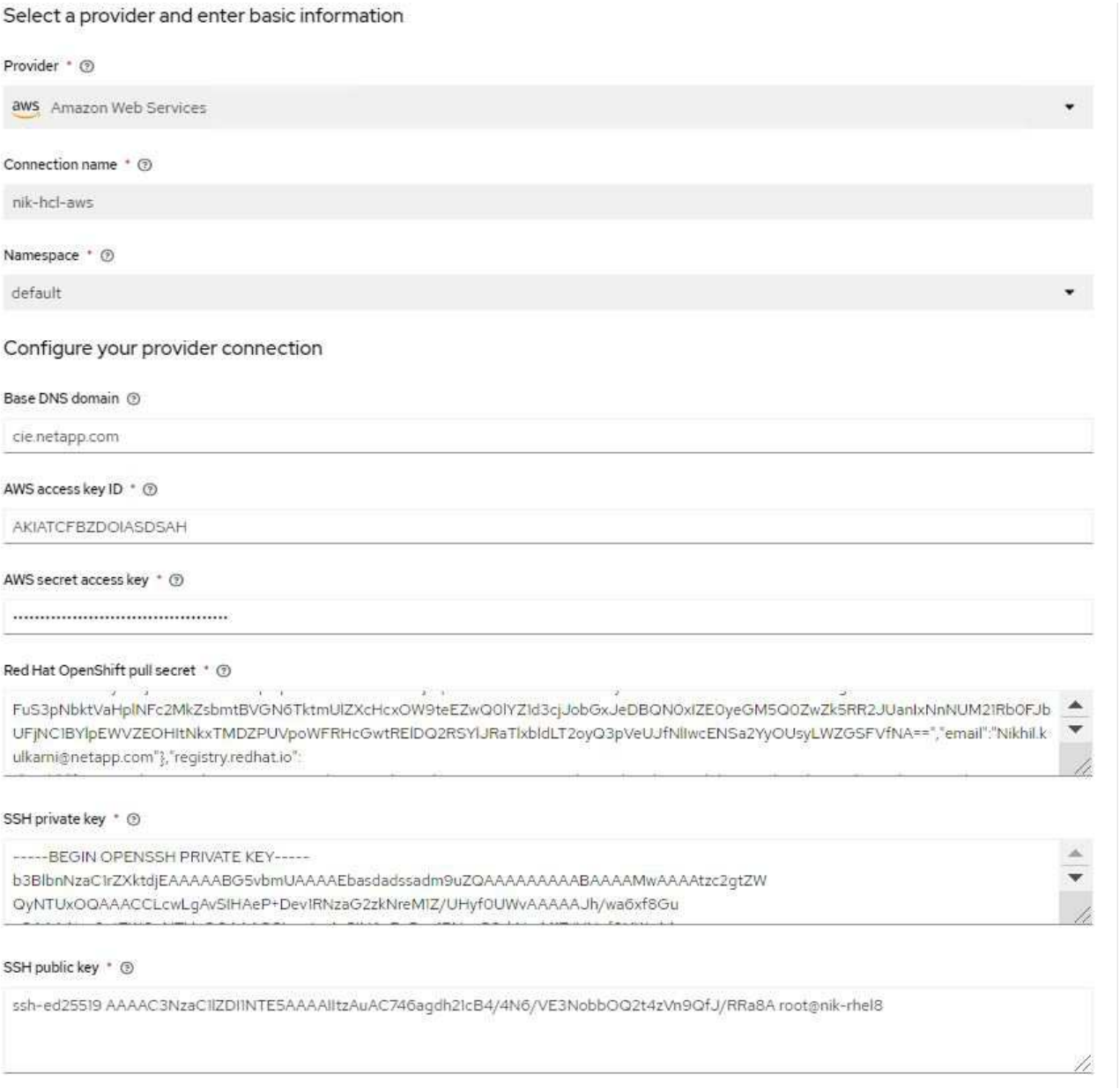
10. Crea una ruta en el espacio de nombres open-cluster-management. Conéctese a la URL en la ruta para acceder a la consola de administración avanzada de clúster.



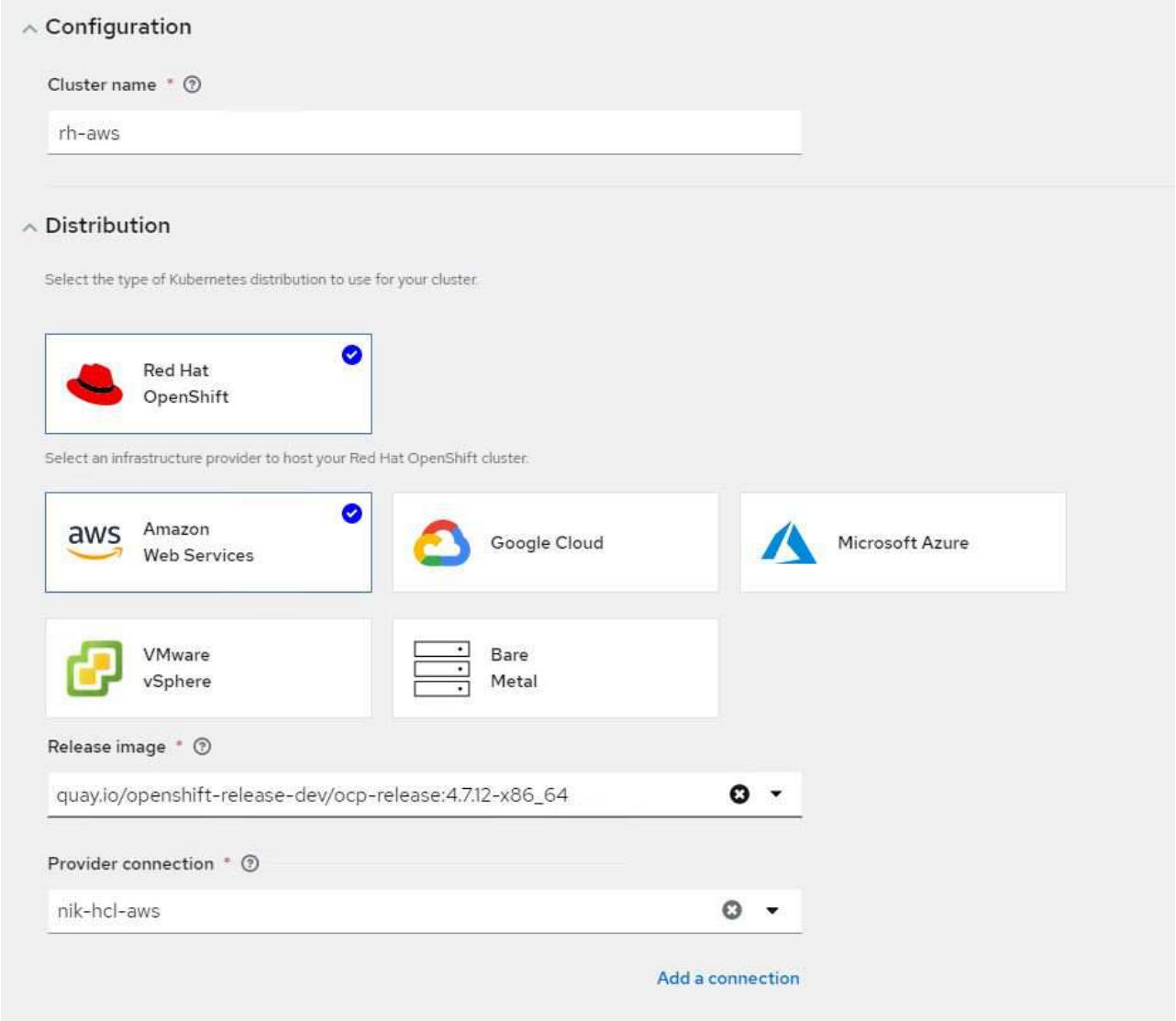
Gestión del ciclo de vida del clúster

Para administrar diferentes clústeres de OpenShift, puede crearlos o importarlos en Administración avanzada de clústeres.

1. Primero navegue a Automatizar Infraestructuras > Clústeres.
2. Para crear un nuevo clúster de OpenShift, complete los siguientes pasos:
   a. Crear una conexión de proveedor: navegue a Conexiones de proveedor y haga clic en Agregar una conexión, proporcione todos los detalles correspondientes al tipo de proveedor seleccionado y haga clic en Agregar.

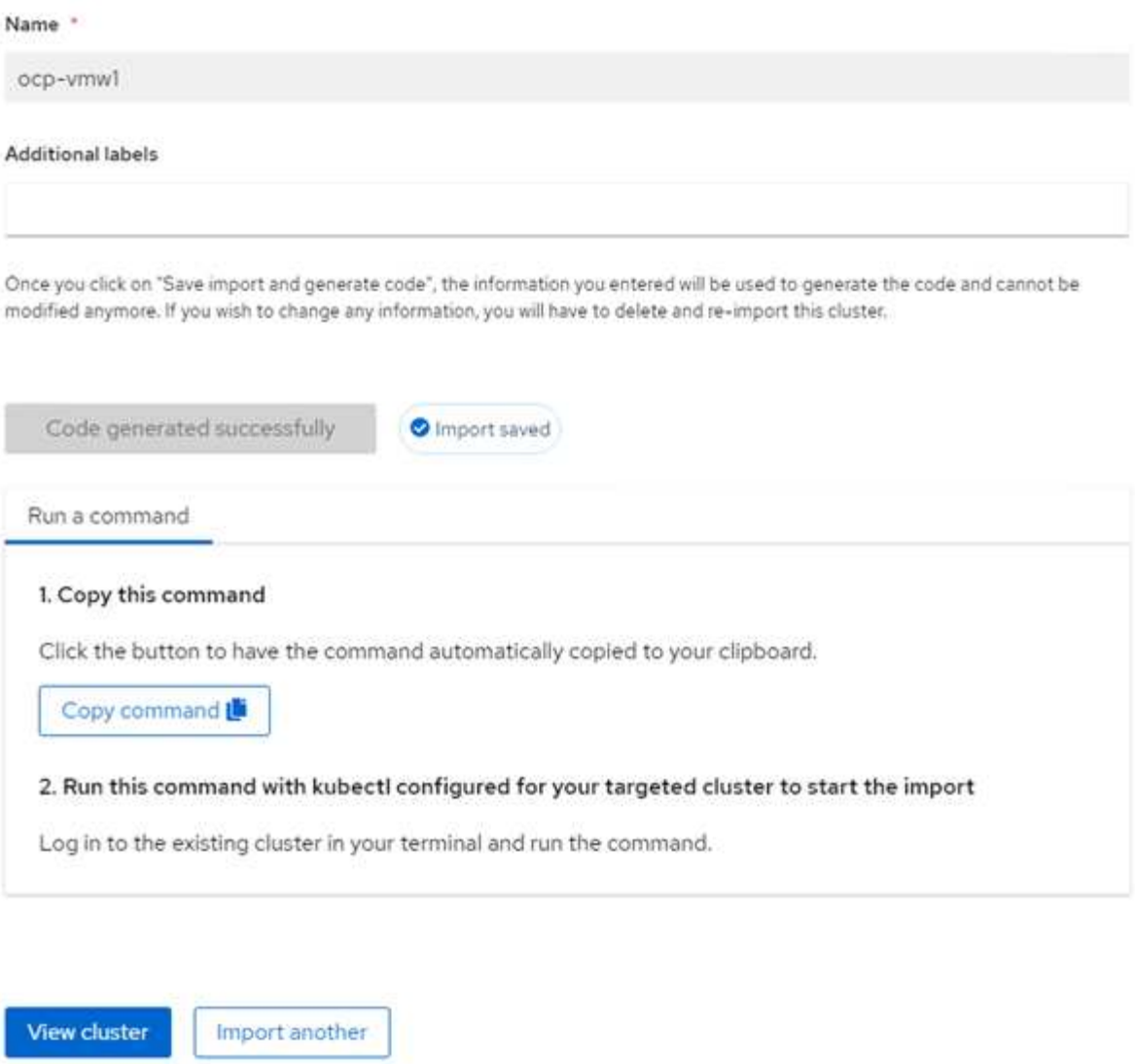


   b. Para crear un nuevo clúster, navegue a Clústeres y haga clic en Agregar un clúster > Crear un clúster. Proporcione los detalles del clúster y el proveedor correspondiente y haga clic en Crear.



c. Una vez creado el clúster, aparece en la lista de clústeres con el estado Listo.

3. Para importar un clúster existente, complete los siguientes pasos:
   a. Vaya a Clústeres y haga clic en Agregar un clúster > Importar un clúster existente.
   b. Ingrese el nombre del clúster y haga clic en Guardar importación y generar código. Se muestra un comando para agregar el clúster existente.
   c. Haga clic en Copiar comando y ejecute el comando en el clúster que se agregará al clúster central. Esto inicia la instalación de los agentes necesarios en el clúster y, una vez completado este proceso, el clúster aparece en la lista de clústeres con el estado Listo.

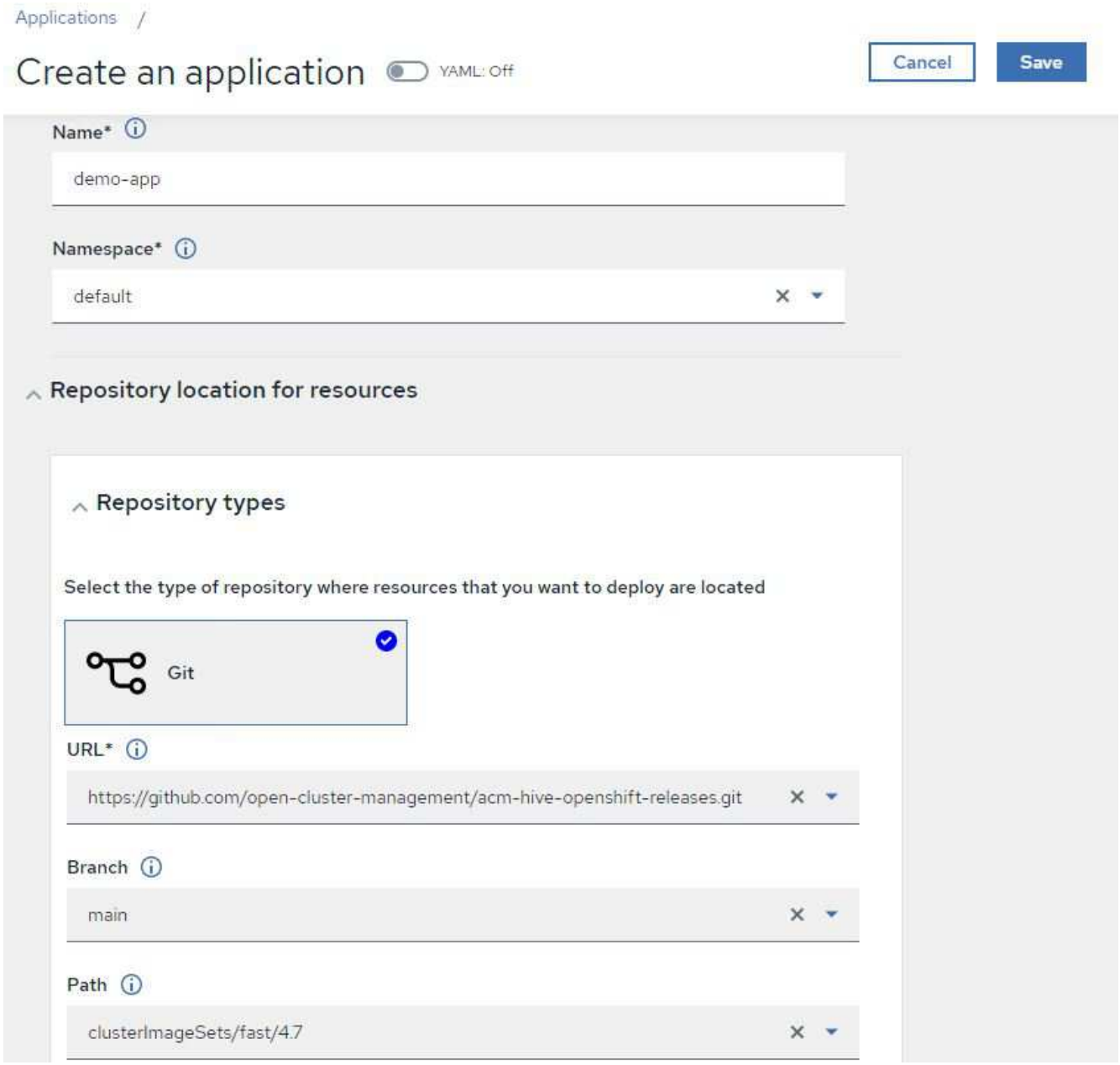


4. Después de crear e importar varios clústeres, puede supervisarlos y administrarlos desde una única consola.

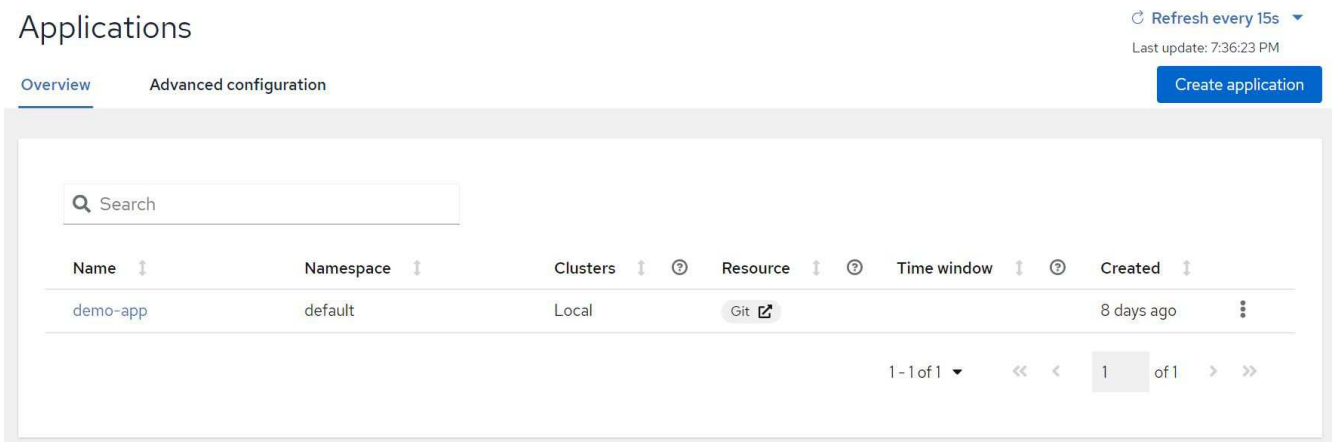
Gestión del ciclo de vida de las aplicaciones

Para crear una aplicación y administrarla en un conjunto de clústeres,

1. Vaya a Administrar aplicaciones desde la barra lateral y haga clic en Crear aplicación. Proporcione los detalles de la aplicación que desea crear y haga clic en Guardar.



2. Una vez instalados los componentes de la aplicación, ésta aparece en la lista.

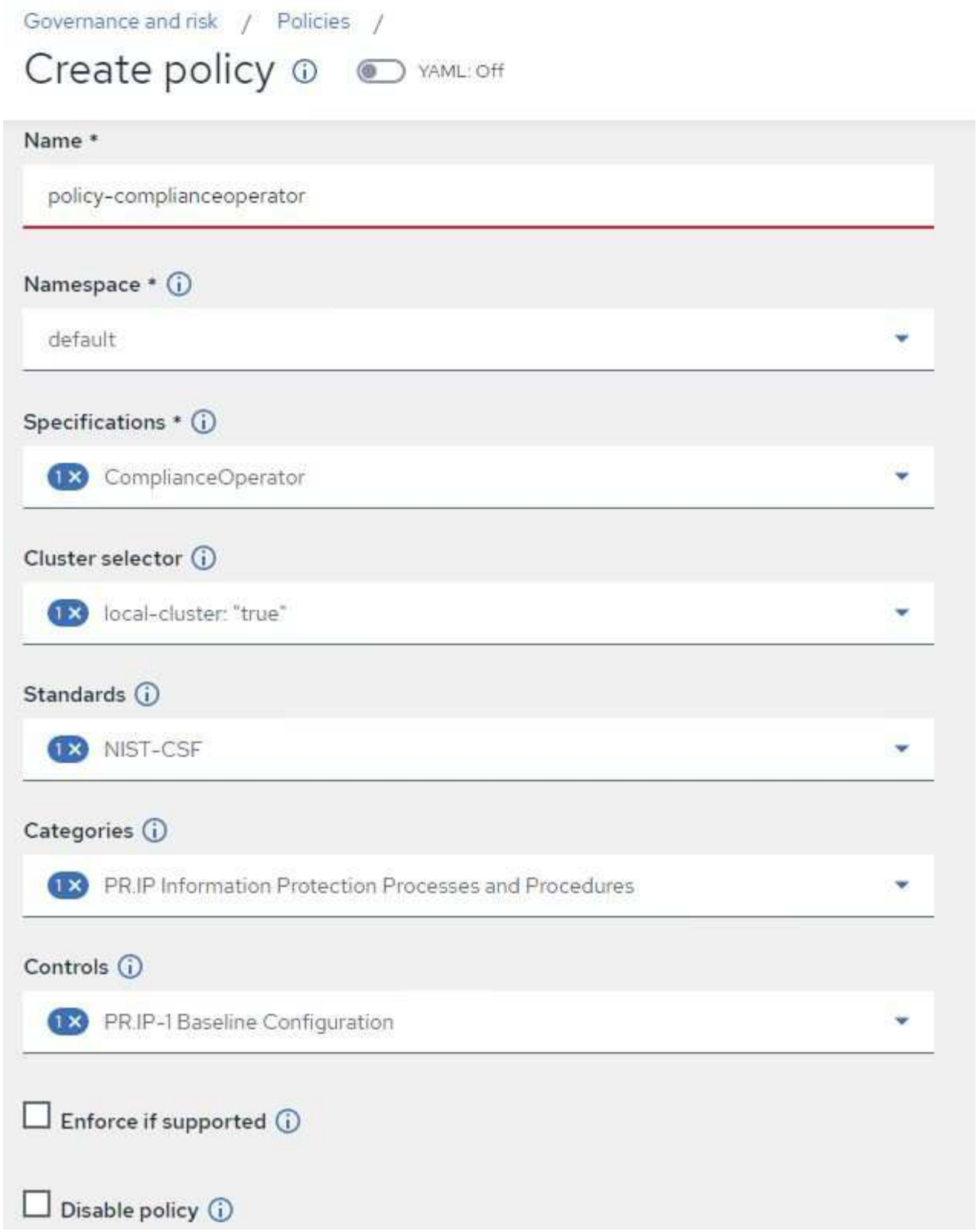


3. Ahora la aplicación se puede monitorizar y gestionar desde la consola.

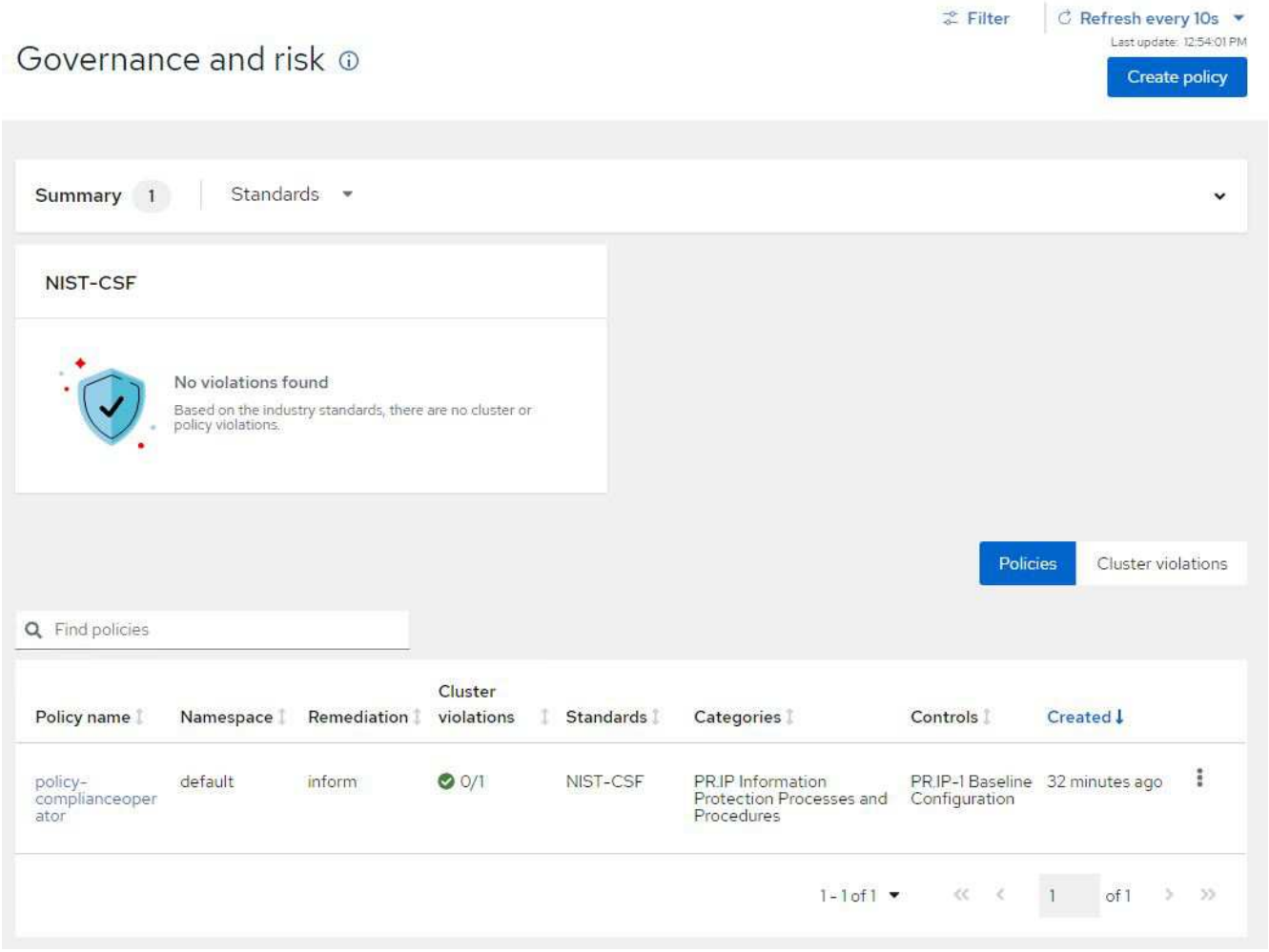
Gobernanza y riesgo

Esta función le permite definir las políticas de cumplimiento para diferentes clústeres y asegurarse de que los clústeres las cumplan. Puede configurar las políticas para informar o remediar cualquier desviación o violación de las reglas.

1. Navegue a Gobernanza y Riesgo desde la barra lateral.
2. Para crear políticas de cumplimiento, haga clic en Crear política, ingrese los detalles de los estándares de la política y seleccione los clústeres que deben cumplir con esta política. Si desea remediar automáticamente las violaciones de esta política, seleccione la casilla de verificación Aplicar si se admite y haga clic en Crear.



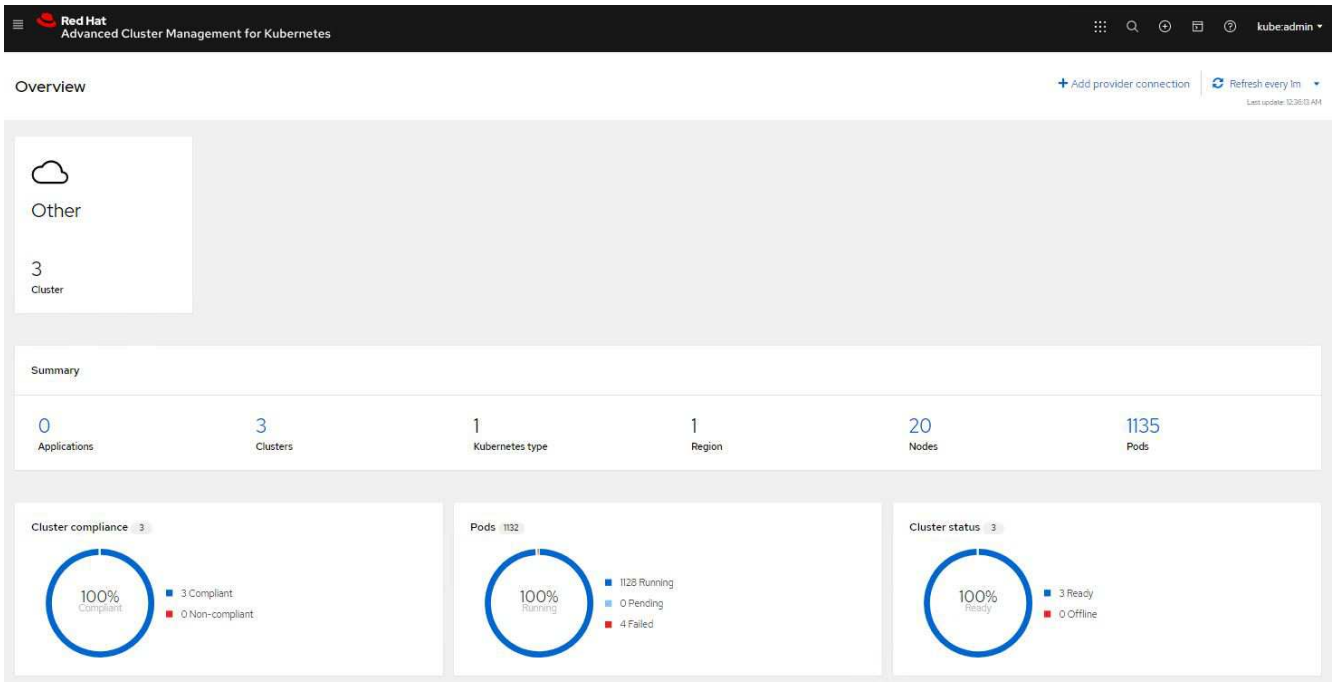
3. Una vez configuradas todas las políticas necesarias, cualquier violación de políticas o clúster se puede monitorear y remediar desde la Administración avanzada de clústeres.



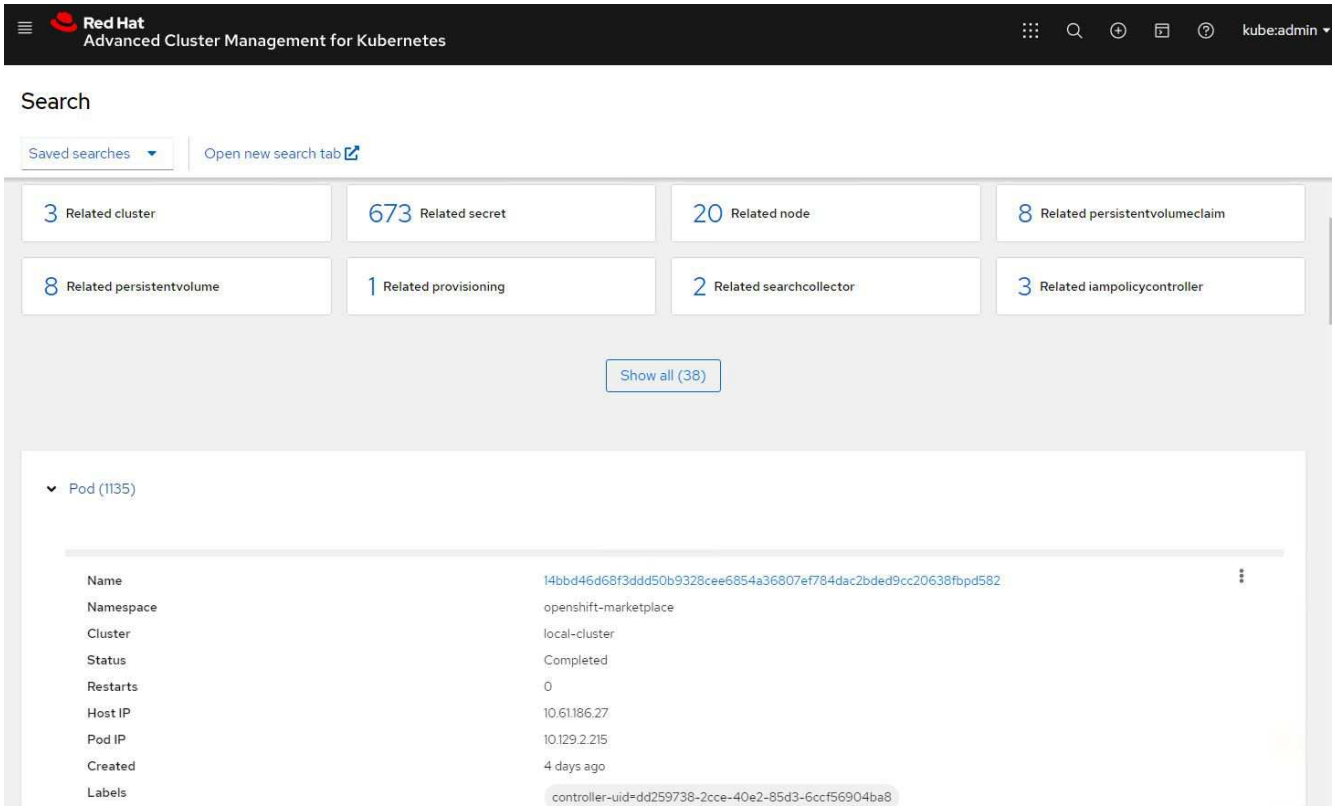
Observabilidad

La gestión avanzada de clústeres para Kubernetes proporciona una manera de supervisar los nodos, pods y aplicaciones, y las cargas de trabajo en todos los clústeres.

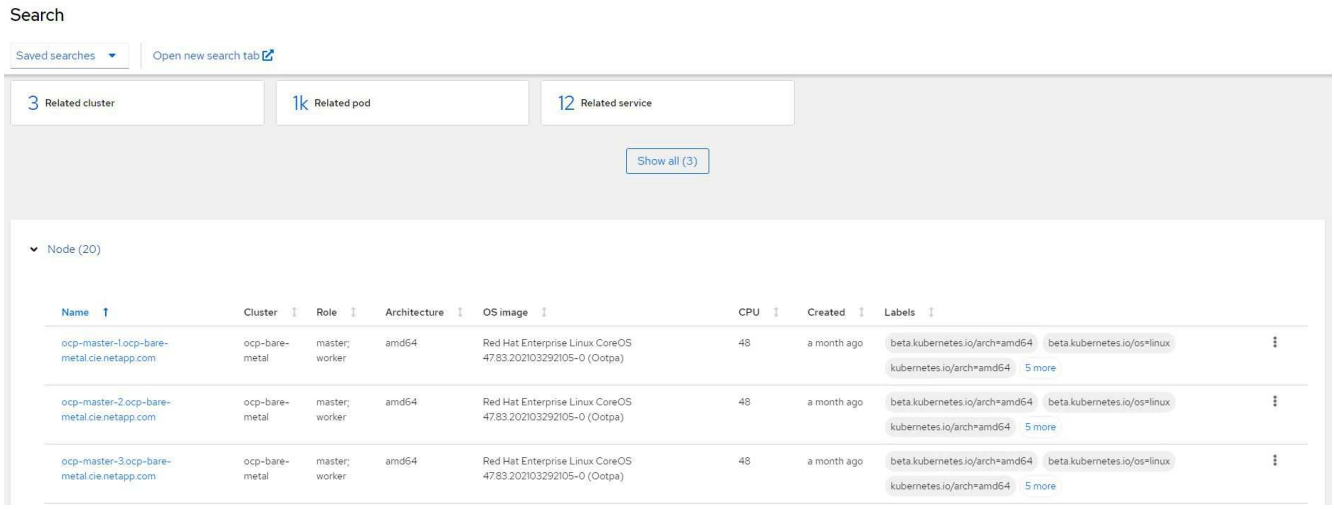
1. Vaya a Observar entornos > Descripción general.



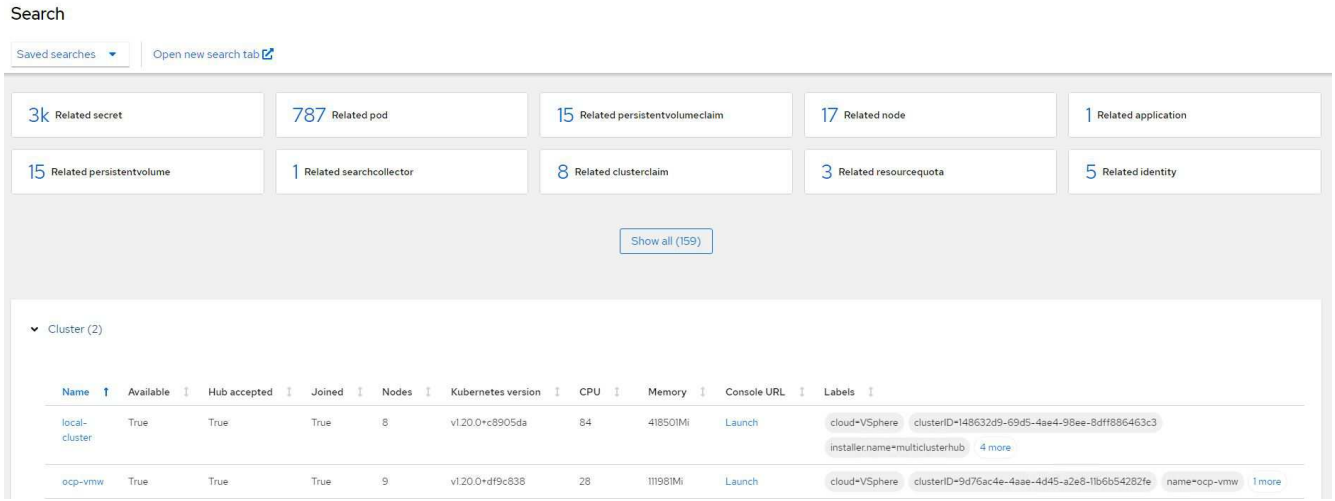
2. Todos los pods y cargas de trabajo en todos los clústeres se monitorean y clasifican según una variedad de filtros. Haga clic en Pods para ver los datos correspondientes.



3. Se monitorean y analizan todos los nodos de los clústeres en función de una variedad de puntos de datos. Haga clic en Nodos para obtener más información sobre los detalles correspondientes.

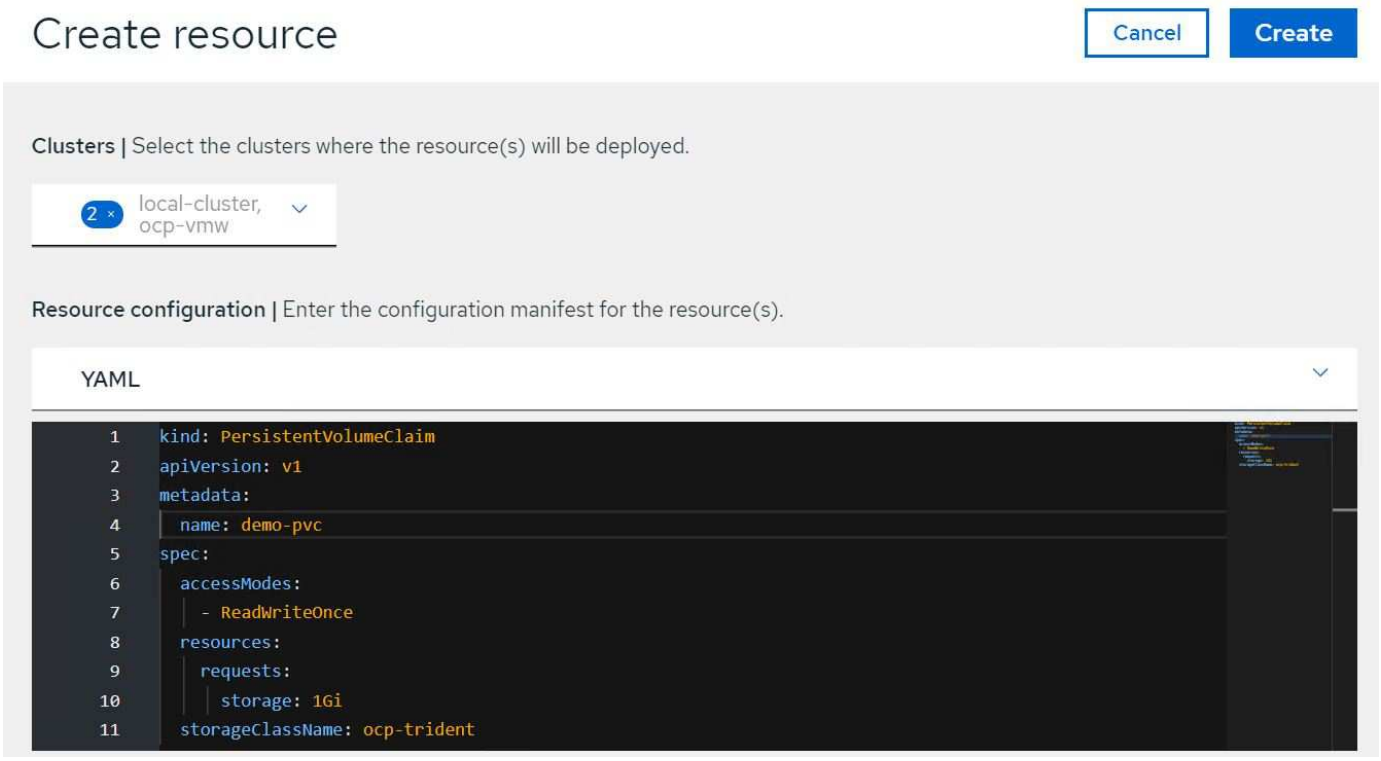


4. Todos los clústeres se monitorean y organizan en función de diferentes recursos y parámetros del clúster. Haga clic en Clústeres para ver los detalles del clúster.



Crear recursos en varios clústeres

La administración avanzada de clústeres para Kubernetes permite a los usuarios crear recursos en uno o más clústeres administrados simultáneamente desde la consola. A modo de ejemplo, si tiene clústeres OpenShift en diferentes sitios respaldados por diferentes clústeres NetApp ONTAP y desea aprovisionar PVC en ambos sitios, puede hacer clic en el signo (+) en la barra superior. Luego, seleccione los clústeres en los que desea crear el PVC, pegue el recurso YAML y haga clic en Crear.



Protección de datos para aplicaciones de contenedores y máquinas virtuales mediante Trident Protect

Esta solución muestra cómo utilizar Trident Protect para realizar operaciones de protección de datos para contenedores y máquinas virtuales.

1. Para obtener detalles sobre cómo crear instantáneas y copias de seguridad y restaurarlas para aplicaciones de contenedores en la plataforma OpenShift Container, consulte"aquí" .
2. Para obtener detalles sobre cómo crear y restaurar desde una copia de seguridad para máquinas virtuales en OpenShift Virtualization implementadas en la plataforma OpenShift Container, consulte"aquí" .

Protección de datos para aplicaciones de contenedores y máquinas virtuales mediante herramientas de terceros

Esta solución muestra cómo utilizar Velero que está integrado con el operador OADP en la plataforma Red Hat OpenShift Container para realizar operaciones de protección de datos para contenedores y máquinas virtuales.

1. Para obtener detalles sobre cómo crear y restaurar desde una copia de seguridad para aplicaciones de contenedor en la plataforma OpenShift Container, consulte"aquí" .
2. Para obtener detalles sobre cómo crear y restaurar desde una copia de seguridad para máquinas virtuales en OpenShift Virtualization implementadas en la plataforma OpenShift Container, consulte"aquí" .

Recursos adicionales para aprender sobre la integración de Red Hat OpenShift Virtualization con el almacenamiento de NetApp

Acceda a recursos adicionales que ofrecen más información sobre cómo respaldar la implementación, la administración y la optimización de Red Hat OpenShift Virtualization con ONTAP en diversas plataformas y tecnologías.

- Documentación de NetApp

  "https://docs.netapp.com/"

- Documentación de Trident

  "https://docs.netapp.com/us-en/trident/index.html"

- Documentación de Red Hat OpenShift

  "https://access.redhat.com/documentation/en-us/openshift_container_platform/4.7/"

- Documentación de la plataforma Red Hat OpenStack

  "https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/16.1/"

- Documentación de virtualización de Red Hat

  "https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.4/"

- Documentación de VMware vSphere

  "https://docs.vmware.com/"