Red Hat OpenShift com NetApp

NetApp container solutions

NetApp
January 21, 2026

Índice

Red Hat OpenShift com NetApp

NVA-1160: Red Hat OpenShift com NetApp

Alan Cowles e Nikhil M Kulkarni, NetApp

Este documento de referência fornece validação de implantação da solução Red Hat OpenShift, implantada por meio da Infraestrutura Provisionada pelo Instalador (IPI) em vários ambientes de data center diferentes, conforme validado pela NetApp. Ele também detalha a integração de armazenamento com os sistemas de armazenamento NetApp , fazendo uso do orquestrador de armazenamento Trident para o gerenciamento de armazenamento persistente. Por fim, uma série de validações de soluções e casos de uso do mundo real são explorados e documentados.

Casos de uso

A solução Red Hat OpenShift com NetApp foi arquitetada para oferecer valor excepcional para clientes com os seguintes casos de uso:

- Fácil de implantar e gerenciar, o Red Hat OpenShift é implantado usando IPI (Installer Provisioned Infrastructure) em bare metal, Red Hat OpenStack Platform, Red Hat Virtualization e VMware vSphere.
- Poder combinado de contêineres empresariais e cargas de trabalho virtualizadas com o Red Hat OpenShift implantado virtualmente em OSP, RHV ou vSphere, ou em bare metal com o OpenShift Virtualization.
- Configuração do mundo real e casos de uso destacando os recursos do Red Hat OpenShift quando usado com armazenamento NetApp e Trident, o orquestrador de armazenamento de código aberto para Kubernetes.

Valor comercial

As empresas estão adotando cada vez mais práticas de DevOps para criar novos produtos, encurtar ciclos de lançamento e adicionar novos recursos rapidamente. Devido à sua natureza ágil inata, contêineres e microsserviços desempenham um papel crucial no suporte às práticas de DevOps. No entanto, praticar DevOps em escala de produção em um ambiente corporativo apresenta seus próprios desafios e impõe certos requisitos à infraestrutura subjacente, como os seguintes:

- Alta disponibilidade em todas as camadas da pilha
- Facilidade de procedimentos de implantação
- Operações e atualizações não disruptivas
- Infraestrutura programável e orientada por API para acompanhar a agilidade dos microsserviços
- Multilocação com garantias de desempenho
- Capacidade de executar cargas de trabalho virtualizadas e em contêineres simultaneamente
- Capacidade de dimensionar a infraestrutura de forma independente com base nas demandas da carga de trabalho

O Red Hat OpenShift com NetApp reconhece esses desafios e apresenta uma solução que ajuda a resolver cada preocupação implementando a implantação totalmente automatizada do RedHat OpenShift IPI no

ambiente de data center escolhido pelo cliente.

Visão geral da tecnologia

A solução Red Hat OpenShift com NetApp é composta pelos seguintes componentes principais:

Plataforma de contêineres Red Hat OpenShift

O Red Hat OpenShift Container Platform é uma plataforma Kubernetes empresarial totalmente suportada. A Red Hat faz vários aprimoramentos no Kubernetes de código aberto para fornecer uma plataforma de aplicativos com todos os componentes totalmente integrados para criar, implantar e gerenciar aplicativos em contêineres.

Para mais informações visite o site do OpenShift "aqui" .

Sistemas de armazenamento NetApp

A NetApp tem vários sistemas de armazenamento perfeitos para data centers empresariais e implantações de nuvem híbrida. O portfólio da NetApp inclui os sistemas de armazenamento NetApp ONTAP, NetApp Element e NetApp e-Series, todos os quais podem fornecer armazenamento persistente para aplicativos em contêineres.

Para mais informações, visite o site da NetApp "aqui" .

Integrações de armazenamento NetApp

O Trident é um orquestrador de armazenamento de código aberto e totalmente suportado para contêineres e distribuições Kubernetes, incluindo o Red Hat OpenShift.

Para mais informações, visite o site da Trident "aqui" .

Opções de configuração avançadas

Esta seção é dedicada às personalizações que usuários do mundo real provavelmente precisariam realizar ao implantar esta solução em produção, como criar um registro de imagem privada dedicado ou implantar instâncias personalizadas do balanceador de carga.

Matriz de suporte atual para versões validadas

Tecnologia	Propósito	Versão do software
NetApp ONTAP	Armazenar	9.8, 9.9.1, 9.12.1
NetApp Element	Armazenar	12,3
NetApp Trident	Orquestração de Armazenamento	22.01.0, 23.04, 23.07, 23.10, 24.02
Red Hat OpenShift	Orquestração de contêineres	4,6 EUS, 4,7, 4,8, 4,10, 4,11, 4,12, 4,13, 4,14
VMware vSphere	Virtualização de data center	7.0, 8.0.2

Red Hat Openshift

Visão geral do OpenShift

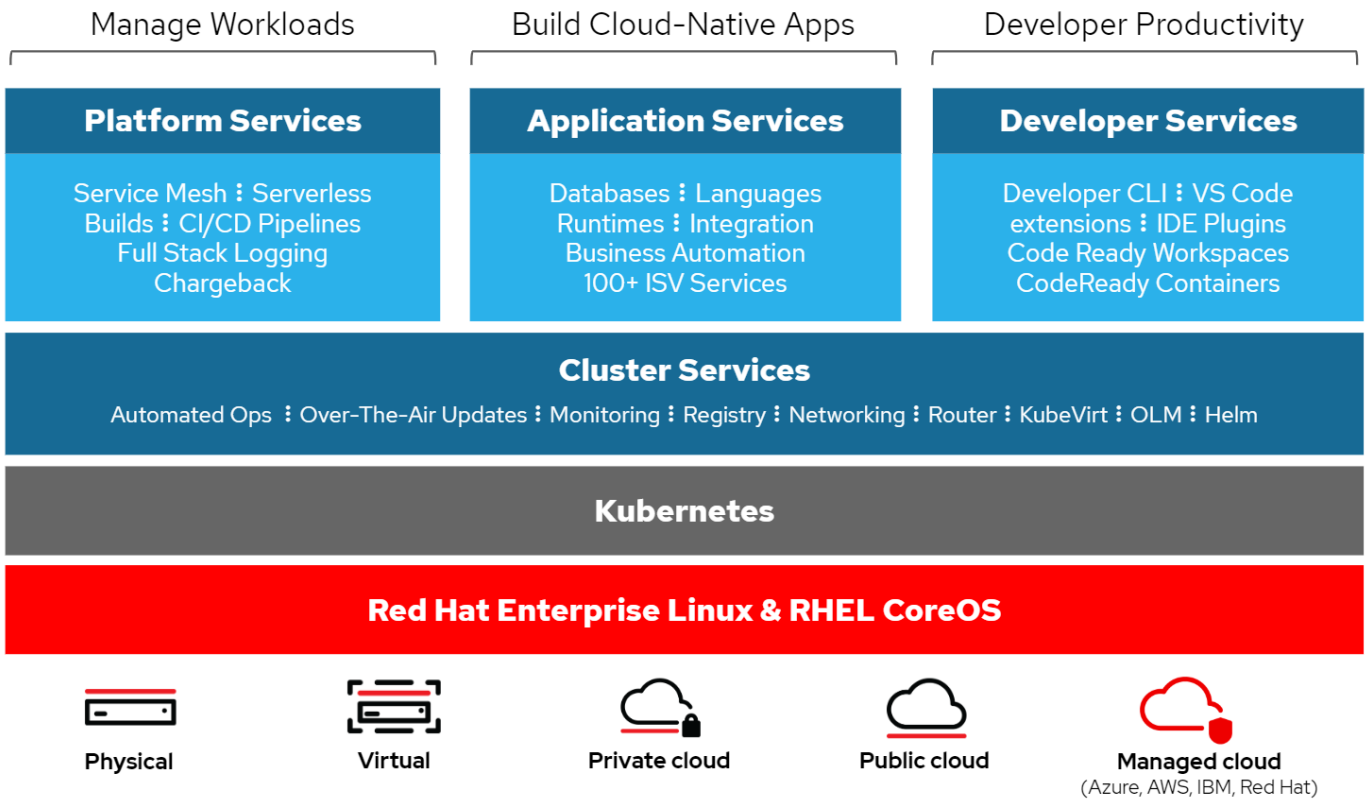
O Red Hat OpenShift Container Platform une o desenvolvimento e as operações de TI em uma única plataforma para criar, implantar e gerenciar aplicativos de forma consistente em infraestruturas de nuvem híbrida e local. O Red Hat OpenShift é construído com base em inovação de código aberto e padrões do setor, incluindo Kubernetes e Red Hat Enterprise Linux CoreOS, a principal distribuição Linux empresarial do mundo, projetada para cargas de trabalho baseadas em contêineres. O OpenShift faz parte do programa Kubernetes certificado pela Cloud Native Computing Foundation (CNCF), fornecendo portabilidade e interoperabilidade de cargas de trabalho de contêiner.

O Red Hat OpenShift fornece os seguintes recursos:

- **Provisionamento de autoatendimento** Os desenvolvedores podem criar aplicativos sob demanda de forma rápida e fácil a partir das ferramentas que mais usam, enquanto as operações mantêm controle total sobre todo o ambiente.
- **Armazenamento persistente** Ao fornecer suporte para armazenamento persistente, o OpenShift Container Platform permite que você execute aplicativos com estado e aplicativos sem estado nativos da nuvem.
- **Integração e desenvolvimento contínuos (CI/CD)** Esta plataforma de código-fonte gerencia imagens de construção e implantação em escala.
- **Padrões de código aberto** Esses padrões incorporam a Open Container Initiative (OCI) e o Kubernetes para orquestração de contêineres, além de outras tecnologias de código aberto. Você não está restrito à tecnologia ou ao roteiro de negócios de um fornecedor específico.
- **Pipelines de CI/CD** O OpenShift fornece suporte pronto para uso para pipelines de CI/CD para que as equipes de desenvolvimento possam automatizar cada etapa do processo de entrega do aplicativo e garantir que ele seja executado em cada alteração feita no código ou na configuração do aplicativo.
- **Controle de acesso baseado em função (RBAC)** Este recurso fornece rastreamento de equipe e usuário para ajudar a organizar um grande grupo de desenvolvedores.
- **Construção e implantação automatizadas** O OpenShift dá aos desenvolvedores a opção de construir seus aplicativos em contêineres ou deixar que a plataforma construa os contêineres a partir do código-fonte do aplicativo ou até mesmo dos binários. A plataforma então automatiza a implantação desses aplicativos na infraestrutura com base nas características definidas para os aplicativos. Por exemplo, qual a quantidade de recursos que devem ser alocados e onde na infraestrutura eles devem ser implantados para que estejam em conformidade com licenças de terceiros.
- **Ambientes consistentes** O OpenShift garante que o ambiente provisionado para desenvolvedores e ao longo do ciclo de vida do aplicativo seja consistente, desde o sistema operacional, até as bibliotecas, a versão do tempo de execução (por exemplo, Java Runtime) e até mesmo o tempo de execução do aplicativo em uso (por exemplo, Tomcat) para remover os riscos originados de ambientes inconsistentes.
- **Gerenciamento de configuração** A configuração e o gerenciamento de dados confidenciais são incorporados à plataforma para garantir que uma configuração de aplicativo consistente e independente do ambiente seja fornecida ao aplicativo, independentemente de quais tecnologias são usadas para construir o aplicativo ou em qual ambiente ele é implantado.
- **Registros e métricas do aplicativo.** O feedback rápido é um aspecto importante do desenvolvimento de

aplicativos. O monitoramento integrado e o gerenciamento de logs do OpenShift fornecem métricas imediatas aos desenvolvedores para que eles estudem como o aplicativo está se comportando diante das mudanças e consigam corrigir problemas o mais cedo possível no ciclo de vida do aplicativo.

- **Segurança e catálogo de contêineres** O OpenShift oferece multilocação e protege o usuário contra execução de código prejudicial usando segurança estabelecida com Security-Enhanced Linux (SELinux), CGroups e Secure Computing Mode (seccomp) para isolar e proteger contêineres. Ele também fornece criptografia por meio de certificados TLS para vários subsistemas e acesso a contêineres certificados pela Red Hat (access.redhat.com/containers) que são escaneados e classificados com ênfase específica em segurança para fornecer contêineres de aplicativos certificados, confiáveis e seguros aos usuários finais.



Métodos de implantação para Red Hat OpenShift

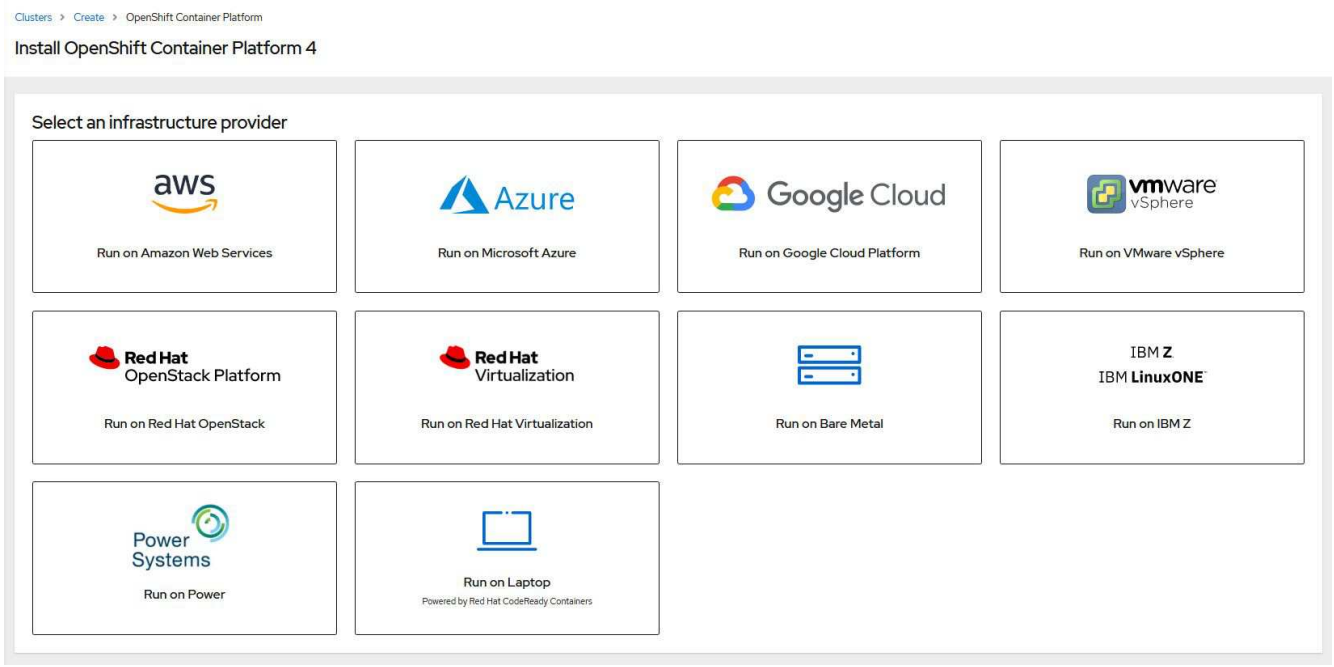
A partir do Red Hat OpenShift 4, os métodos de implantação do OpenShift incluem implantações manuais usando a Infraestrutura Provisionada pelo Usuário (UPI) para implantações altamente personalizadas ou implantações totalmente automatizadas usando a Infraestrutura Provisionada pelo Instalador (IPI).

O método de instalação IPI é o método preferido na maioria dos casos porque permite a implantação rápida de clusters OpenShift para ambientes de desenvolvimento, teste e produção.

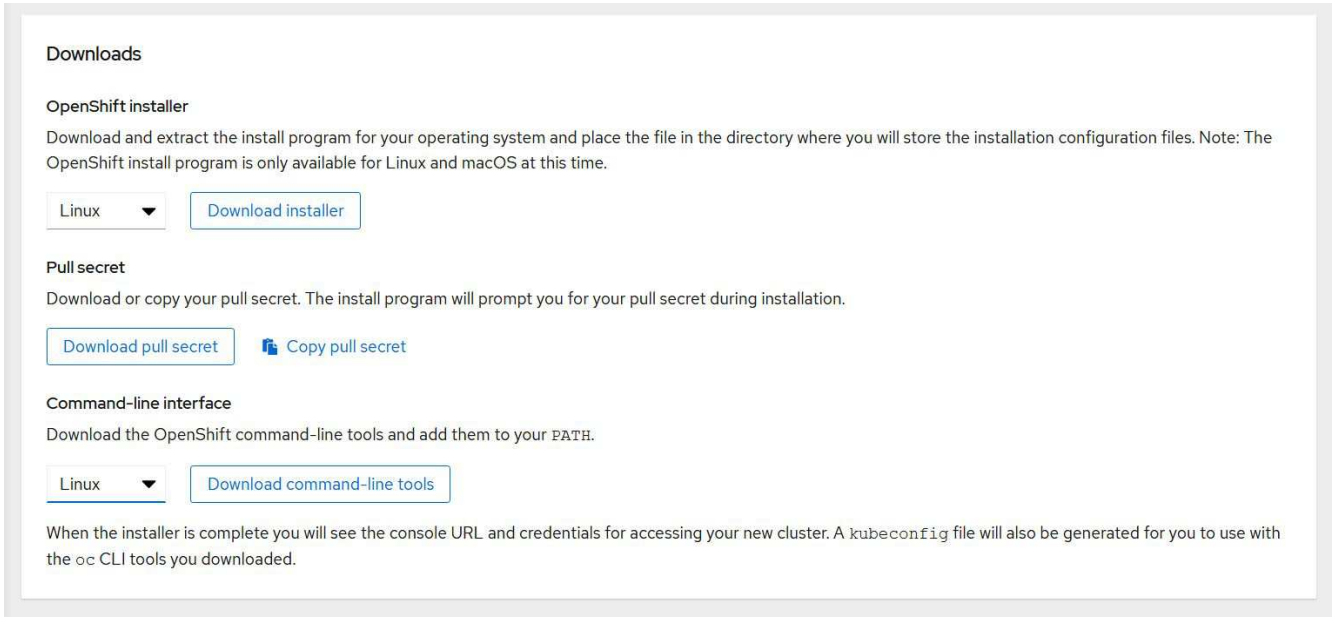
Instalação IPI do Red Hat OpenShift

A implantação da infraestrutura provisionada pelo instalador (IPI) do OpenShift envolve estas etapas de alto nível:

1. Visite o Red Hat OpenShift"site" e faça login com suas credenciais SSO.
2. Selecione o ambiente no qual você gostaria de implantar o Red Hat OpenShift.



3. Na próxima tela, baixe o instalador, o pull secret exclusivo e as ferramentas CLI para gerenciamento.



4. Siga o"instruções de instalação" fornecido pela Red Hat para implantação no ambiente de sua escolha.

Implantações OpenShift validadas pela NetApp

A NetApp testou e validou a implantação do Red Hat OpenShift em seus laboratórios usando o método de implantação de infraestrutura provisionada pelo instalador (IPI) em cada um dos seguintes ambientes de data center:

- "OpenShift em Bare Metal"
- "OpenShift na plataforma Red Hat OpenStack"
- "OpenShift na virtualização Red Hat"
- "OpenShift no VMware vSphere"

OpenShift em Bare Metal

O OpenShift on Bare Metal fornece uma implantação automatizada da OpenShift Container Platform em servidores comuns.

O OpenShift on Bare Metal é semelhante às implantações virtuais do OpenShift, que oferecem facilidade de implantação, provisionamento rápido e dimensionamento de clusters OpenShift, ao mesmo tempo em que oferece suporte a cargas de trabalho virtualizadas para aplicativos que não estão prontos para serem conteinerizados. Ao implantar em bare metal, você não precisa da sobrecarga extra necessária para gerenciar o ambiente do hipervisor do host, além do ambiente OpenShift. Ao implantar diretamente em servidores bare metal, você também pode reduzir as limitações de sobrecarga física de ter que compartilhar recursos entre o host e o ambiente OpenShift.

O OpenShift no Bare Metal oferece os seguintes recursos:

- **Implantação de IPI ou instalador assistido** Com um cluster OpenShift implantado pela Installer Provisioned Infrastructure (IPI) em servidores bare metal, os clientes podem implantar um ambiente OpenShift altamente versátil e facilmente escalável diretamente em servidores comuns, sem a necessidade de gerenciar uma camada de hipervisor.
- **Design de cluster compacto** Para minimizar os requisitos de hardware, o OpenShift em bare metal permite que os usuários implantem clusters de apenas 3 nós, permitindo que os nós do plano de controle do OpenShift também atuem como nós de trabalho e contêineres de host.
- **Virtualização OpenShift** O OpenShift pode executar máquinas virtuais dentro de contêineres usando a Virtualização OpenShift. Essa virtualização nativa de contêiner executa o hipervisor KVM dentro de um contêiner e anexa volumes persistentes para armazenamento de VM.
- **Infraestrutura otimizada para IA/ML** Implante aplicativos como o Kubeflow para aplicativos de aprendizado de máquina incorporando nós de trabalho baseados em GPU ao seu ambiente OpenShift e aproveitando o OpenShift Advanced Scheduling.

Projeto de rede

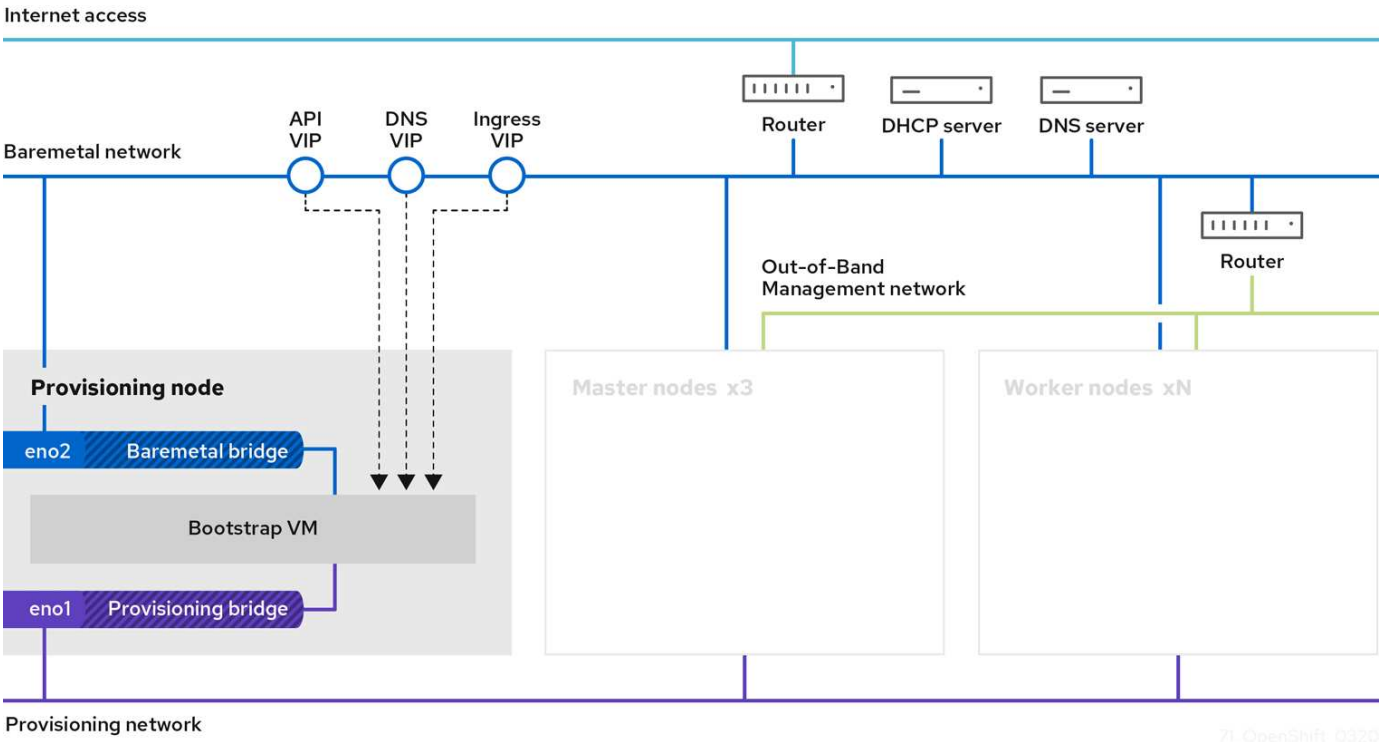
A solução Red Hat OpenShift on NetApp usa dois switches de dados para fornecer conectividade de dados primária a 25 Gbps. Ele também usa dois switches de gerenciamento que fornecem conectividade a 1 Gbps para gerenciamento em banda para os nós de armazenamento e gerenciamento fora de banda para a funcionalidade IPMI.

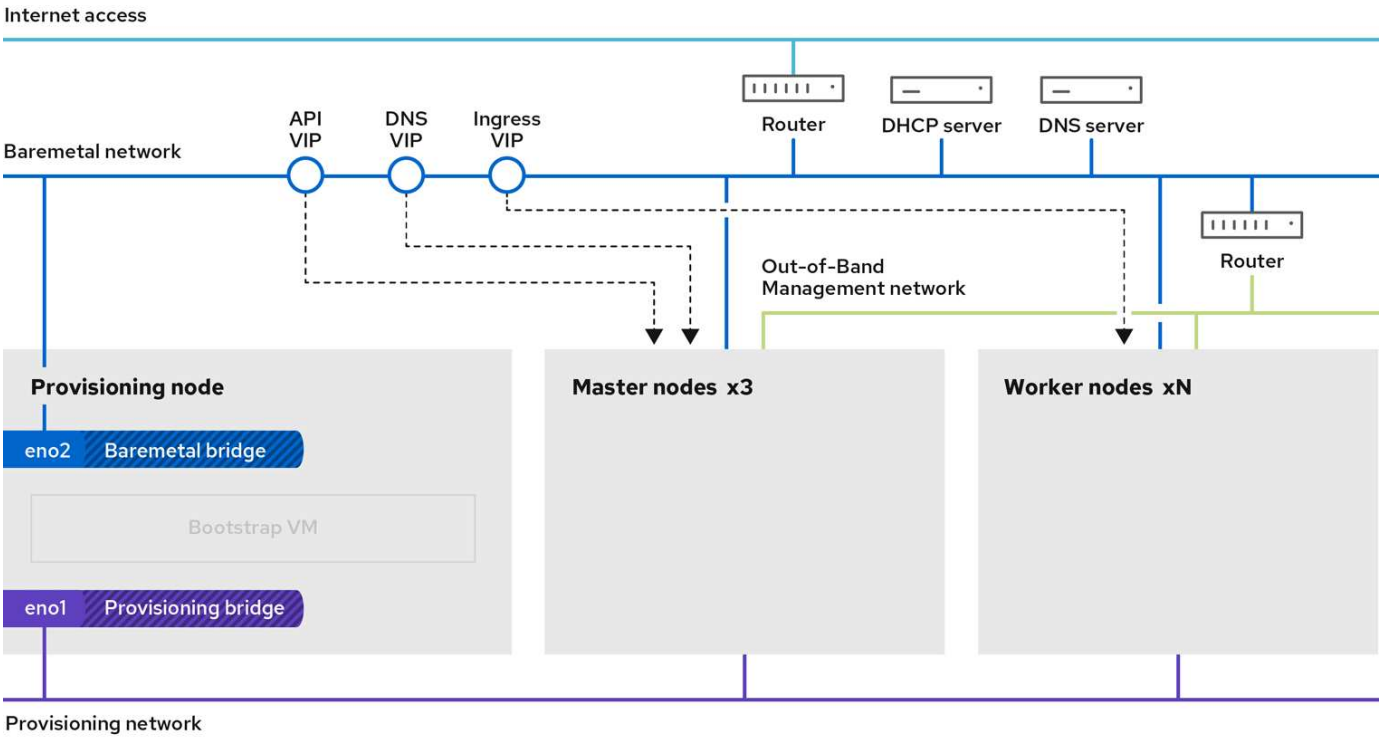
Para implantação bare-metal do IPI do OpenShift, você deve criar um nó de provisionamento, uma máquina Red Hat Enterprise Linux 8 que deve ter interfaces de rede conectadas a redes separadas.

- **Rede de provisionamento** Esta rede é usada para inicializar os nós bare-metal e instalar as imagens e pacotes necessários para implantar o cluster OpenShift.
- **Rede bare-metal** Esta rede é usada para comunicação pública do cluster após sua implantação.

Para a configuração do nó do provisionador, o cliente cria interfaces de ponte que permitem que o tráfego seja roteado corretamente no próprio nó e na VM Bootstrap provisionada para fins de implantação. Após a implantação do cluster, a API e os endereços VIP de entrada são migrados do nó de bootstrap para o cluster recém-implantado.

As imagens a seguir retratam o ambiente durante a implantação do IPI e após a conclusão da implantação.





Requisitos do VLAN

A solução Red Hat OpenShift com NetApp foi projetada para separar logicamente o tráfego de rede para diferentes propósitos usando redes locais virtuais (VLANs).

VLANs	Propósito	ID da VLAN
Rede de gerenciamento fora de banda	Gerenciamento para nós bare metal e IPMI	16
Rede bare-metal	Rede para serviços OpenShift assim que o cluster estiver disponível	181
Rede de provisionamento	Rede para inicialização PXE e instalação de nós bare metal via IPI	3485

Embora cada uma dessas redes seja virtualmente separada por VLANs, cada porta física deve ser configurada no Modo de Acesso com a VLAN primária atribuída, porque não há como passar uma tag VLAN durante uma sequência de inicialização PXE.

Recursos de suporte de infraestrutura de rede

A seguinte infraestrutura deve estar em vigor antes da implantação da plataforma de contêiner OpenShift:

- Pelo menos um servidor DNS que forneça uma resolução completa de nome de host acessível a partir da rede de gerenciamento em banda e da rede de VM.
- Pelo menos um servidor NTP acessível pela rede de gerenciamento em banda e pela rede de VM.
- (Opcional) Conectividade de saída de internet para a rede de gerenciamento em banda e a rede de VM.

OpenShift na plataforma Red Hat OpenStack

A Red Hat OpenStack Platform oferece uma base integrada para criar, implantar e dimensionar uma nuvem OpenStack privada segura e confiável.

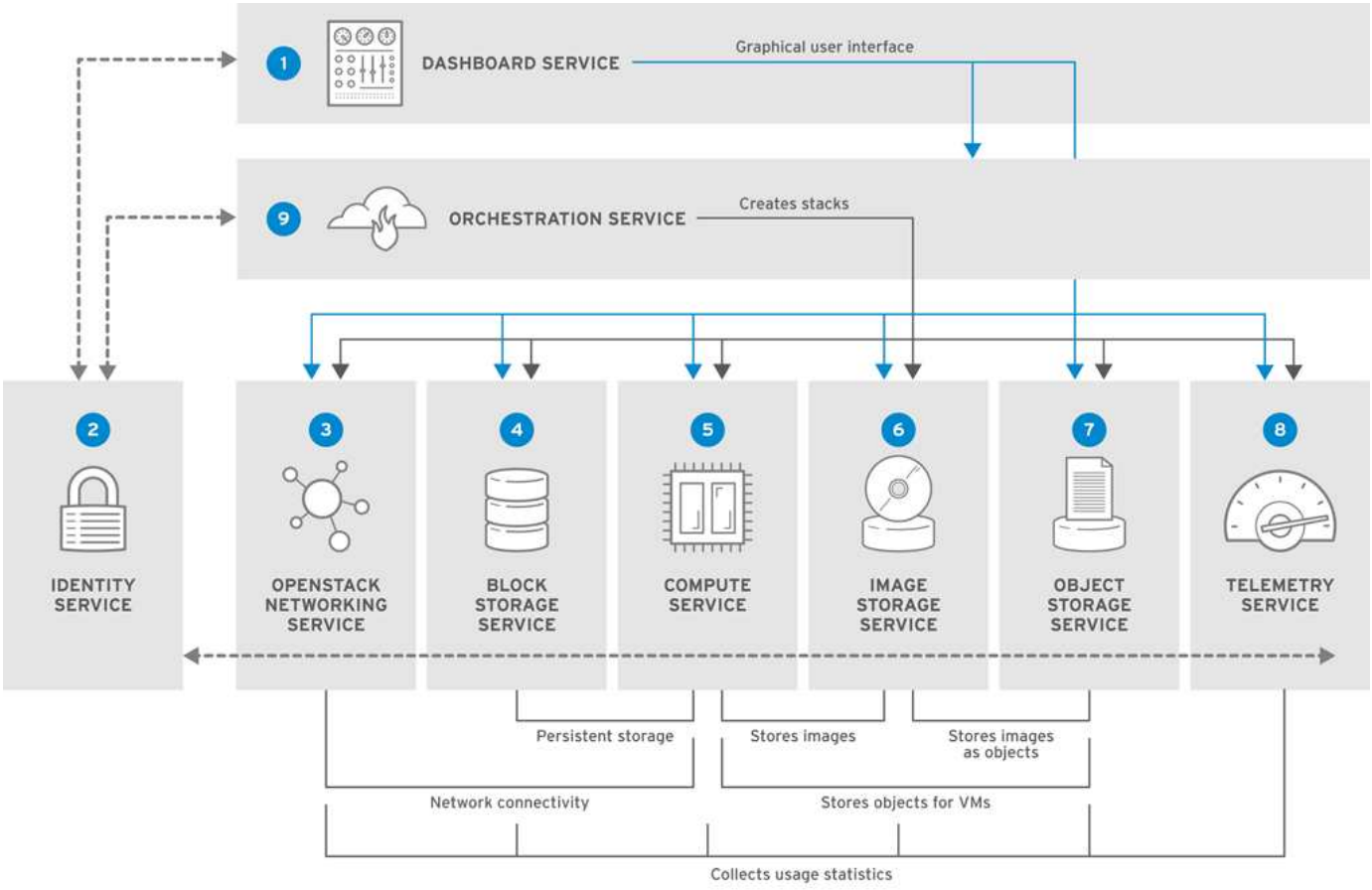
OSP é uma nuvem de infraestrutura como serviço (IaaS) implementada por uma coleção de serviços de controle que gerenciam recursos de computação, armazenamento e rede. O ambiente é gerenciado usando uma interface baseada na web que permite que administradores e usuários controlem, provisionem e automatizem recursos do OpenStack. Além disso, a infraestrutura do OpenStack é facilitada por uma extensa interface de linha de comando e API, permitindo recursos completos de automação para administradores e usuários finais.

O projeto OpenStack é um projeto comunitário de rápido desenvolvimento que fornece versões atualizadas a cada seis meses. Inicialmente, o Red Hat OpenStack Platform acompanhou esse ciclo de lançamento publicando uma nova versão junto com cada versão upstream e fornecendo suporte de longo prazo para cada terceira versão. Recentemente, com o lançamento do OSP 16.0 (baseado no OpenStack Train), a Red Hat optou por não acompanhar os números de lançamentos, mas, em vez disso, retroportou novos recursos em sub-lançamentos. O lançamento mais recente é o Red Hat OpenStack Platform 16.1, que inclui recursos avançados retroportados das versões Ussuri e Victoria.

Para mais informações sobre o OSP, consulte o"Site da plataforma Red Hat OpenStack" .

Serviços OpenStack

Os serviços da plataforma OpenStack são implantados como contêineres, o que isola os serviços uns dos outros e permite atualizações fáceis. A plataforma OpenStack usa um conjunto de contêineres criados e gerenciados com Kolla. A implantação de serviços é realizada extraindo imagens de contêiner do Portal Personalizado do Red Hat. Esses contêineres de serviço são gerenciados usando o comando Podman e são implantados, configurados e mantidos com o Red Hat OpenStack Director.



Serviço	Nome do projeto	Descrição
Painel	Horizonte	Painel baseado em navegador da Web que você usa para gerenciar serviços OpenStack.
Identidade	Keystone	Serviço centralizado para autenticação e autorização de serviços OpenStack e para gerenciamento de usuários, projetos e funções.
Rede OpenStack	Nêutron	Fornece conectividade entre as interfaces dos serviços OpenStack.
Armazenamento em bloco	Cinza	Gerencia volumes de armazenamento em bloco persistentes para máquinas virtuais (VMs).
Calcular	Nova	Gerencia e provisiona VMs em execução em nós de computação.
Imagem	Olhar	Serviço de registro usado para armazenar recursos como imagens de VM e instantâneos de volume.
Armazenamento de objetos	Rápido	Permite que os usuários armazenem e recuperem arquivos e dados arbitrários.
Telemetria	Tetometro	Fornece medições de uso de recursos de nuvem.
Orquestração	Aquecer	Mecanismo de orquestração baseado em modelo que suporta criação automática de pilhas de recursos.

Projeto de rede

A solução Red Hat OpenShift com NetApp usa dois switches de dados para fornecer conectividade de dados primária a 25 Gbps. Ele também usa dois switches de gerenciamento adicionais que fornecem conectividade a 1 Gbps para gerenciamento em banda para os nós de armazenamento e gerenciamento fora de banda para a funcionalidade IPMI.

A funcionalidade IPMI é exigida pelo Red Hat OpenStack Director para implantar o Red Hat OpenStack Platform usando o serviço de provisionamento bare-metal da Ironic.

Requisitos do VLAN

O Red Hat OpenShift com NetApp foi projetado para separar logicamente o tráfego de rede para diferentes propósitos usando redes locais virtuais (VLANs). Essa configuração pode ser dimensionada para atender às demandas dos clientes ou para fornecer maior isolamento para serviços de rede específicos. A tabela a seguir lista as VLANs necessárias para implementar a solução durante a validação da solução na NetApp.

VLANs	Propósito	ID da VLAN
Rede de gerenciamento fora de banda	Rede usada para gerenciamento de nós físicos e serviço IPMI para Ironic.	16
Infraestrutura de armazenamento	Rede usada para nós controladores mapearem volumes diretamente para dar suporte a serviços de infraestrutura como o Swift.	201
Cinzas de armazenamento	Rede usada para mapear e anexar volumes de bloco diretamente a instâncias virtuais implantadas no ambiente.	202
API interna	Rede usada para comunicação entre os serviços OpenStack usando comunicação de API, mensagens RPC e comunicação de banco de dados.	301
Inquilino	O Neutron fornece a cada locatário suas próprias redes por meio de tunelamento através do VXLAN. O tráfego de rede é isolado dentro de cada rede de locatário. Cada rede de locatários tem uma sub-rede IP associada a ela, e namespaces de rede significam que várias redes de locatários podem usar o mesmo intervalo de endereços sem causar conflitos.	302
Gerenciamento de armazenamento	O OpenStack Object Storage (Swift) usa essa rede para sincronizar objetos de dados entre nós de réplica participantes. O serviço de proxy atua como interface intermediária entre as solicitações do usuário e a camada de armazenamento subjacente. O proxy recebe solicitações de entrada e localiza a réplica necessária para recuperar os dados solicitados.	303
PXE	O OpenStack Director fornece inicialização PXE como parte do serviço de provisionamento bare metal do Ironic para orquestrar a instalação do OSP Overcloud.	3484
Externo	Rede disponível publicamente que hospeda o OpenStack Dashboard (Horizon) para gerenciamento gráfico e permite chamadas de API públicas para gerenciar serviços OpenStack.	3485
Rede de gerenciamento em banda	Fornece acesso para funções de administração do sistema, como acesso SSH, tráfego DNS e tráfego Network Time Protocol (NTP). Esta rede também atua como um gateway para nós não controladores.	3486

Recursos de suporte de infraestrutura de rede

A seguinte infraestrutura deve estar em vigor antes da implantação da OpenShift Container Platform:

- Pelo menos um servidor DNS que fornece uma resolução completa de nome de host.
- Pelo menos três servidores NTP que podem manter o tempo sincronizado para os servidores na solução.
- (Opcional) Conectividade de saída de internet para o ambiente OpenShift.

Melhores práticas para implantações de produção

Esta seção lista diversas práticas recomendadas que uma organização deve levar em consideração antes de implantar esta solução em produção.

Implante o OpenShift em uma nuvem privada OSP com pelo menos três nós de computação

A arquitetura verificada descrita neste documento apresenta a implantação mínima de hardware adequada para operações de HA, implantando três nós controladores OSP e dois nós de computação OSP. Essa arquitetura garante uma configuração tolerante a falhas na qual ambos os nós de computação podem iniciar instâncias virtuais e as VMs implantadas podem migrar entre os dois hipervisores.

Como o Red Hat OpenShift é implantado inicialmente com três nós mestres, uma configuração de dois nós pode fazer com que pelo menos dois mestres ocupem o mesmo nó, o que pode levar a uma possível interrupção do OpenShift se esse nó específico ficar indisponível. Portanto, é uma prática recomendada da Red Hat implantar pelo menos três nós de computação OSP para que os mestres do OpenShift possam ser distribuídos uniformemente e a solução receba um grau adicional de tolerância a falhas.

Configurar afinidade de máquina virtual/host

A distribuição dos mestres do OpenShift entre vários nós do hipervisor pode ser obtida habilitando a afinidade VM/host.

Afinidade é uma maneira de definir regras para um conjunto de VMs e/ou hosts que determinam se as VMs serão executadas juntas no mesmo host ou hosts no grupo ou em hosts diferentes. Ele é aplicado a VMs criando grupos de afinidade que consistem em VMs e/ou hosts com um conjunto de parâmetros e condições idênticos. Dependendo se as VMs em um grupo de afinidade são executadas no mesmo host ou hosts no grupo ou separadamente em hosts diferentes, os parâmetros do grupo de afinidade podem definir afinidade positiva ou negativa. Na Red Hat OpenStack Platform, regras de afinidade e antiafinidade de host podem ser criadas e aplicadas criando grupos de servidores e configurando filtros para que as instâncias implantadas pelo Nova em um grupo de servidores sejam implantadas em diferentes nós de computação.

Um grupo de servidores tem um máximo padrão de 10 instâncias virtuais cujo posicionamento ele pode gerenciar. Isso pode ser modificado atualizando as cotas padrões do Nova.

> Há um limite específico de afinidade/antiafinidade para grupos de servidores OSP; se não houver recursos suficientes para implantar em nós separados ou recursos suficientes para permitir o compartilhamento de nós, a VM não inicializará.

Para configurar grupos de afinidade, consulte"Como configuro Affinity e Anti-Affinity para instâncias do OpenStack?" .

Use um arquivo de instalação personalizado para implantação do OpenShift

O IPI facilita a implantação de clusters OpenShift por meio do assistente interativo discutido anteriormente neste documento. No entanto, é possível que você precise alterar alguns valores padrão como parte de uma

implantação de cluster.

Nesses casos, você pode executar o assistente sem implantar um cluster imediatamente; em vez disso, ele cria um arquivo de configuração a partir do qual o cluster pode ser implantado posteriormente. Isso é muito útil se você precisar alterar os valores padrão do IPI ou se quiser implantar vários clusters idênticos em seu ambiente para outros usos, como multilocação. Para obter mais informações sobre como criar uma configuração de instalação personalizada para o OpenShift, consulte "Red Hat OpenShift Instalando um Cluster no OpenStack com Personalizações".

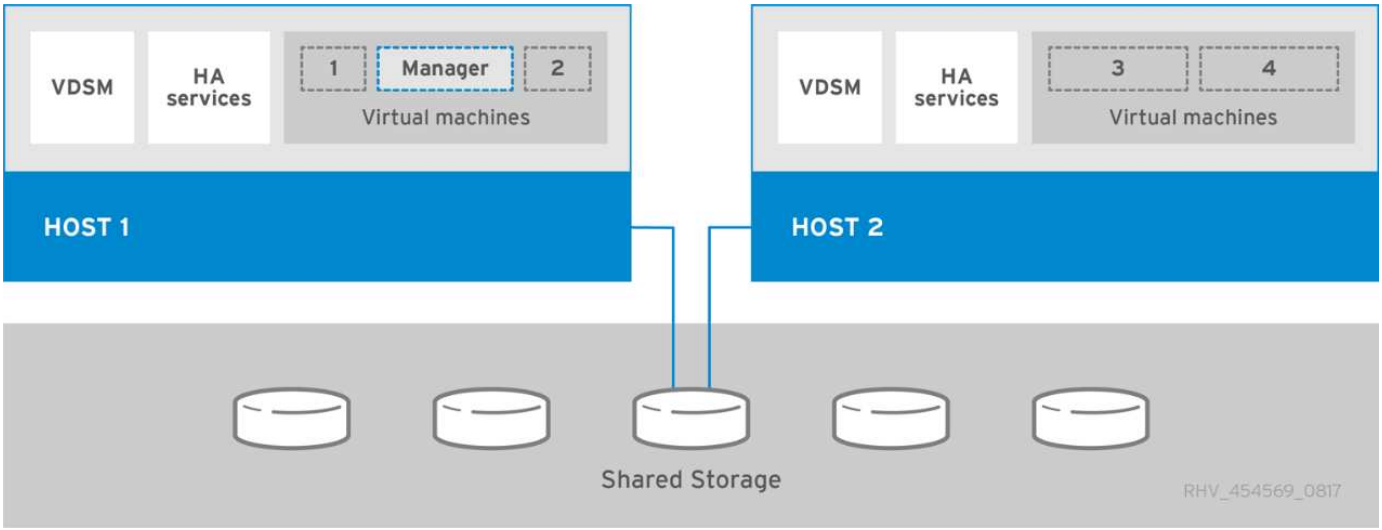
OpenShift na virtualização Red Hat

O Red Hat Virtualization (RHV) é uma plataforma de data center virtual empresarial que roda no Red Hat Enterprise Linux (RHEL) e usa o hipervisor KVM.

Para obter mais informações sobre RHV, consulte o"Site de virtualização do Red Hat" .

O RHV oferece os seguintes recursos:

- **Gerenciamento centralizado de VMs e hosts** O gerenciador RHV é executado como uma máquina física ou virtual (VM) na implantação e fornece uma GUI baseada na Web para o gerenciamento da solução a partir de uma interface central.
- **Mecanismo auto-hospedado** Para minimizar os requisitos de hardware, o RHV permite que o RHV Manager (RHV-M) seja implantado como uma VM nos mesmos hosts que executam VMs convidadas.
- **Alta disponibilidade** Para evitar interrupções em caso de falhas no host, o RHV permite que as VMs sejam configuradas para alta disponibilidade. As VMs de alta disponibilidade são controladas no nível do cluster usando políticas de resiliência.
- **Alta escalabilidade** Um único cluster RHV pode ter até 200 hosts de hipervisor, o que lhe permite dar suporte aos requisitos de VMs enormes para hospedar cargas de trabalho de nível empresarial e com alto consumo de recursos.
- **Segurança aprimorada** Herdadas do RHV, as tecnologias Secure Virtualization (sVirt) e Security Enhanced Linux (SELinux) são empregadas pelo RHV para fins de segurança elevada e reforço para hosts e VMs. A principal vantagem desses recursos é o isolamento lógico de uma VM e seus recursos associados.



Projeto de rede

A solução Red Hat OpenShift on NetApp usa dois switches de dados para fornecer conectividade de dados primária a 25 Gbps. Ele também usa dois switches de gerenciamento adicionais que fornecem conectividade a 1 Gbps para gerenciamento em banda dos nós de armazenamento e gerenciamento fora de banda para funcionalidade IPMI. O OCP usa a rede lógica da máquina virtual no RHV para gerenciamento de cluster. Esta seção descreve o arranjo e a finalidade de cada segmento de rede virtual usado na solução e descreve os pré-requisitos para implantar a solução.

Requisitos do VLAN

O Red Hat OpenShift no RHV foi projetado para separar logicamente o tráfego de rede para diferentes propósitos usando redes locais virtuais (VLANs). Essa configuração pode ser dimensionada para atender às demandas dos clientes ou para fornecer maior isolamento para serviços de rede específicos. A tabela a seguir lista as VLANs necessárias para implementar a solução durante a validação da solução na NetApp.

VLANs	Propósito	ID da VLAN
Rede de gerenciamento fora de banda	Gerenciamento para nós físicos e IPMI	16
Rede VM	Acesso à rede de convidados virtuais	1172
Rede de gerenciamento em banda	Gerenciamento para nós RHV-H, RHV-Manager e rede ovirtmgmt	3343
Rede de armazenamento	Rede de armazenamento para NetApp Element iSCSI	3344
Rede de migração	Rede para migração de convidados virtuais	3345

Recursos de suporte de infraestrutura de rede

A seguinte infraestrutura deve estar em vigor antes da implantação da OpenShift Container Platform:

- Pelo menos um servidor DNS fornecendo resolução completa de nome de host que pode ser acessado pela rede de gerenciamento em banda e pela rede de VM.
- Pelo menos um servidor NTP acessível pela rede de gerenciamento em banda e pela rede de VM.
- (Opcional) Conectividade de saída de internet para a rede de gerenciamento em banda e a rede de VM.

Melhores práticas para implantações de produção

Esta seção lista diversas práticas recomendadas que uma organização deve levar em consideração antes de implantar esta solução em produção.

Implante o OpenShift em um cluster RHV de pelo menos três nós

A arquitetura verificada descrita neste documento apresenta a implantação mínima de hardware adequada para operações de HA, implantando dois nós de hipervisor RHV-H e garantindo uma configuração tolerante a falhas em que ambos os hosts podem gerenciar o mecanismo hospedado e as VMs implantadas podem migrar entre os dois hipervisores.

Como o Red Hat OpenShift é implantado inicialmente com três nós mestres, é garantido que em uma configuração de dois nós pelo menos dois mestres ocuparão o mesmo nó, o que pode levar a uma possível interrupção do OpenShift se esse nó específico ficar indisponível. Portanto, é uma prática recomendada da Red Hat que pelo menos três nós do hipervisor RHV-H sejam implantados como parte da solução para que os

mestres do OpenShift possam ser distribuídos uniformemente e a solução receba um grau adicional de tolerância a falhas.

Configurar afinidade de máquina virtual/host

Você pode distribuir os mestres do OpenShift entre vários nós do hipervisor habilitando a afinidade VM/host.

Afinidade é uma maneira de definir regras para um conjunto de VMs e/ou hosts que determinam se as VMs serão executadas juntas no mesmo host ou hosts no grupo ou em hosts diferentes. Ele é aplicado a VMs criando grupos de afinidade que consistem em VMs e/ou hosts com um conjunto de parâmetros e condições idênticos. Dependendo se as VMs em um grupo de afinidade são executadas no mesmo host ou hosts no grupo ou separadamente em hosts diferentes, os parâmetros do grupo de afinidade podem definir afinidade positiva ou negativa.

As condições definidas para os parâmetros podem ser de aplicação rígida ou flexível. A aplicação rígida garante que as VMs em um grupo de afinidade sempre sigam estritamente a afinidade positiva ou negativa, sem levar em conta condições externas. A aplicação suave garante que uma preferência maior seja definida para que as VMs em um grupo de afinidade sigam a afinidade positiva ou negativa sempre que possível. Na configuração de dois ou três hipervisores descrita neste documento, a afinidade suave é a configuração recomendada. Em clusters maiores, a afinidade rígida pode distribuir corretamente os nós do OpenShift.

Para configurar grupos de afinidade, consulte o"Chapéu vermelho 6.11. Documentação de Grupos de Afinidade" .

Use um arquivo de instalação personalizado para implantação do OpenShift

O IPI facilita a implantação de clusters OpenShift por meio do assistente interativo discutido anteriormente neste documento. No entanto, é possível que haja alguns valores padrão que precisem ser alterados como parte da implantação do cluster.

Nesses casos, você pode executar e executar a tarefa do assistente sem implantar imediatamente um cluster. Em vez disso, um arquivo de configuração é criado a partir do qual o cluster pode ser implantado posteriormente. Isso é muito útil se você quiser alterar algum padrão de IPI ou se quiser implantar vários clusters idênticos em seu ambiente para outros usos, como multilocação. Para obter mais informações sobre como criar uma configuração de instalação personalizada para o OpenShift, consulte"Red Hat OpenShift Instalando um Cluster no RHV com Personalizações" .

OpenShift no VMware vSphere

O VMware vSphere é uma plataforma de virtualização para gerenciamento centralizado de um grande número de servidores e redes virtualizados em execução no hipervisor ESXi.

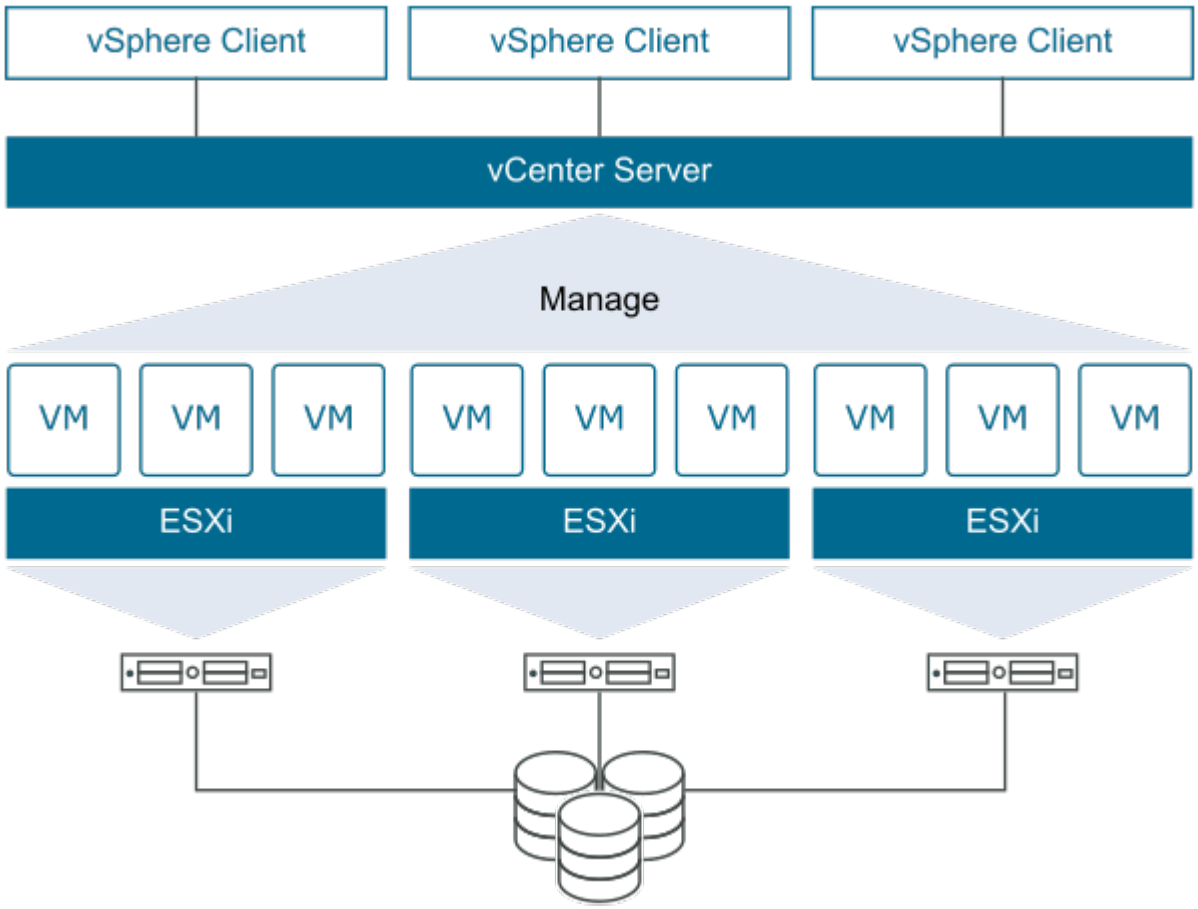
Para obter mais informações sobre o VMware vSphere, consulte o"Site do VMware vSphere" .

O VMware vSphere oferece os seguintes recursos:

- **VMware vCenter Server** O VMware vCenter Server fornece gerenciamento unificado de todos os hosts e VMs a partir de um único console e agrega o monitoramento de desempenho de clusters, hosts e VMs.
- **VMware vSphere vMotion** O VMware vCenter permite que você migre VMs a quente entre nós no cluster mediante solicitação, de maneira não disruptiva.
- **vSphere High Availability** Para evitar interrupções em caso de falhas no host, o VMware vSphere permite que os hosts sejam agrupados e configurados para alta disponibilidade. As VMs que são

interrompidas por falha de host são reinicializadas em breve em outros hosts no cluster, restaurando os serviços.

- **Distributed Resource Scheduler (DRS)** Um cluster VMware vSphere pode ser configurado para balancear a carga das necessidades de recursos das VMs que ele está hospedando. VMs com contenções de recursos podem ser migradas dinamicamente para outros nós no cluster para garantir que haja recursos suficientes disponíveis.



Projeto de rede

A solução Red Hat OpenShift on NetApp usa dois switches de dados para fornecer conectividade de dados primária a 25 Gbps. Ele também usa dois switches de gerenciamento adicionais que fornecem conectividade a 1 Gbps para gerenciamento em banda para os nós de armazenamento e gerenciamento fora de banda para a funcionalidade IPMI. O OCP usa a rede lógica da VM no VMware vSphere para seu gerenciamento de cluster. Esta seção descreve o arranjo e a finalidade de cada segmento de rede virtual usado na solução e descreve os pré-requisitos para a implantação da solução.

Requisitos do VLAN

O Red Hat OpenShift no VMware vSphere foi projetado para separar logicamente o tráfego de rede para diferentes propósitos usando redes locais virtuais (VLANs). Essa configuração pode ser dimensionada para atender às demandas dos clientes ou para fornecer maior isolamento para serviços de rede específicos. A tabela a seguir lista as VLANs necessárias para implementar a solução durante a validação da solução na NetApp.

VLANs	Propósito	ID da VLAN
Rede de gerenciamento fora de banda	Gerenciamento para nós físicos e IPMI	16
Rede VM	Acesso à rede de convidados virtuais	181
Rede de armazenamento	Rede de armazenamento para ONTAP NFS	184
Rede de armazenamento	Rede de armazenamento para ONTAP iSCSI	185
Rede de gerenciamento em banda	Gerenciamento para nós ESXi, VCenter Server, ONTAP Select	3480
Rede de armazenamento	Rede de armazenamento para NetApp Element iSCSI	3481
Rede de migração	Rede para migração de convidados virtuais	3482

Recursos de suporte de infraestrutura de rede

A seguinte infraestrutura deve estar em vigor antes da implantação da OpenShift Container Platform:

- Pelo menos um servidor DNS fornecendo resolução completa de nome de host que pode ser acessado pela rede de gerenciamento em banda e pela rede de VM.
- Pelo menos um servidor NTP acessível pela rede de gerenciamento em banda e pela rede de VM.
- (Opcional) Conectividade de saída de internet para a rede de gerenciamento em banda e a rede de VM.

Melhores práticas para implantações de produção

Esta seção lista diversas práticas recomendadas que uma organização deve levar em consideração antes de implantar esta solução em produção.

Implante o OpenShift em um cluster ESXi de pelo menos três nós

A arquitetura verificada descrita neste documento apresenta a implantação mínima de hardware adequada para operações de HA, implantando dois nós de hipervisor ESXi e garantindo uma configuração tolerante a falhas ao habilitar o VMware vSphere HA e o VMware vMotion. Essa configuração permite que as VMs implantadas migrem entre os dois hipervisores e sejam reinicializadas caso um host fique indisponível.

Como o Red Hat OpenShift é implantado inicialmente com três nós mestres, pelo menos dois mestres em uma configuração de dois nós podem ocupar o mesmo nó em algumas circunstâncias, o que pode levar a uma possível interrupção do OpenShift se esse nó específico ficar indisponível. Portanto, é uma prática recomendada da Red Hat que pelo menos três nós do hipervisor ESXi sejam implantados para que os mestres do OpenShift possam ser distribuídos uniformemente, o que fornece um grau adicional de tolerância a falhas.

Configurar afinidade de máquina virtual e host

É possível garantir a distribuição dos mestres do OpenShift entre vários nós do hipervisor habilitando a afinidade de VM e host.

Afinidade ou antiafinidade é uma maneira de definir regras para um conjunto de VMs e/ou hosts que determinam se as VMs serão executadas juntas no mesmo host ou hosts do grupo ou em hosts diferentes.

Ele é aplicado a VMs criando grupos de afinidade que consistem em VMs e/ou hosts com um conjunto de parâmetros e condições idênticos. Dependendo se as VMs em um grupo de afinidade são executadas no mesmo host ou hosts no grupo ou separadamente em hosts diferentes, os parâmetros do grupo de afinidade podem definir afinidade positiva ou negativa.

Para configurar grupos de afinidade, consulte o "Documentação do vSphere 9.0: Usando regras de afinidade DRS".

Use um arquivo de instalação personalizado para implantação do OpenShift

O IPI facilita a implantação de clusters OpenShift por meio do assistente interativo discutido anteriormente neste documento. No entanto, é possível que você precise alterar alguns valores padrão como parte de uma implantação de cluster.

Nesses casos, você pode executar e executar a tarefa do assistente sem implantar imediatamente um cluster. Em vez disso, o assistente cria um arquivo de configuração a partir do qual o cluster pode ser implantado posteriormente. Isso é muito útil se você precisar alterar algum padrão de IPI ou se quiser implantar vários clusters idênticos em seu ambiente para outros usos, como multilocação. Para obter mais informações sobre como criar uma configuração de instalação personalizada para o OpenShift, consulte"Red Hat OpenShift Instalando um Cluster no vSphere com Personalizações" .

Serviço Red Hat OpenShift na AWS

O Red Hat OpenShift Service on AWS (ROSA) é um serviço gerenciado que você pode usar para criar, dimensionar e implantar aplicativos em contêineres com a plataforma corporativa Kubernetes do Red Hat OpenShift na AWS. O ROSA simplifica a migração de cargas de trabalho locais do Red Hat OpenShift para a AWS e oferece integração completa com outros serviços da AWS.

Para mais informações sobre o ROSA, consulte a documentação aqui:"Serviço Red Hat OpenShift na AWS (documentação da AWS)" . "Serviço Red Hat OpenShift na AWS (documentação da Red Hat)" .

Sistemas de armazenamento NetApp

NetApp ONTAP

O NetApp ONTAP é uma poderosa ferramenta de software de armazenamento com recursos como uma interface gráfica de usuário intuitiva, APIs REST com integração de automação, análise preditiva e ação corretiva baseadas em IA, atualizações de hardware não disruptivas e importação entre armazenamentos.

Para obter mais informações sobre o sistema de armazenamento NetApp ONTAP , visite o "Site NetApp ONTAP" .

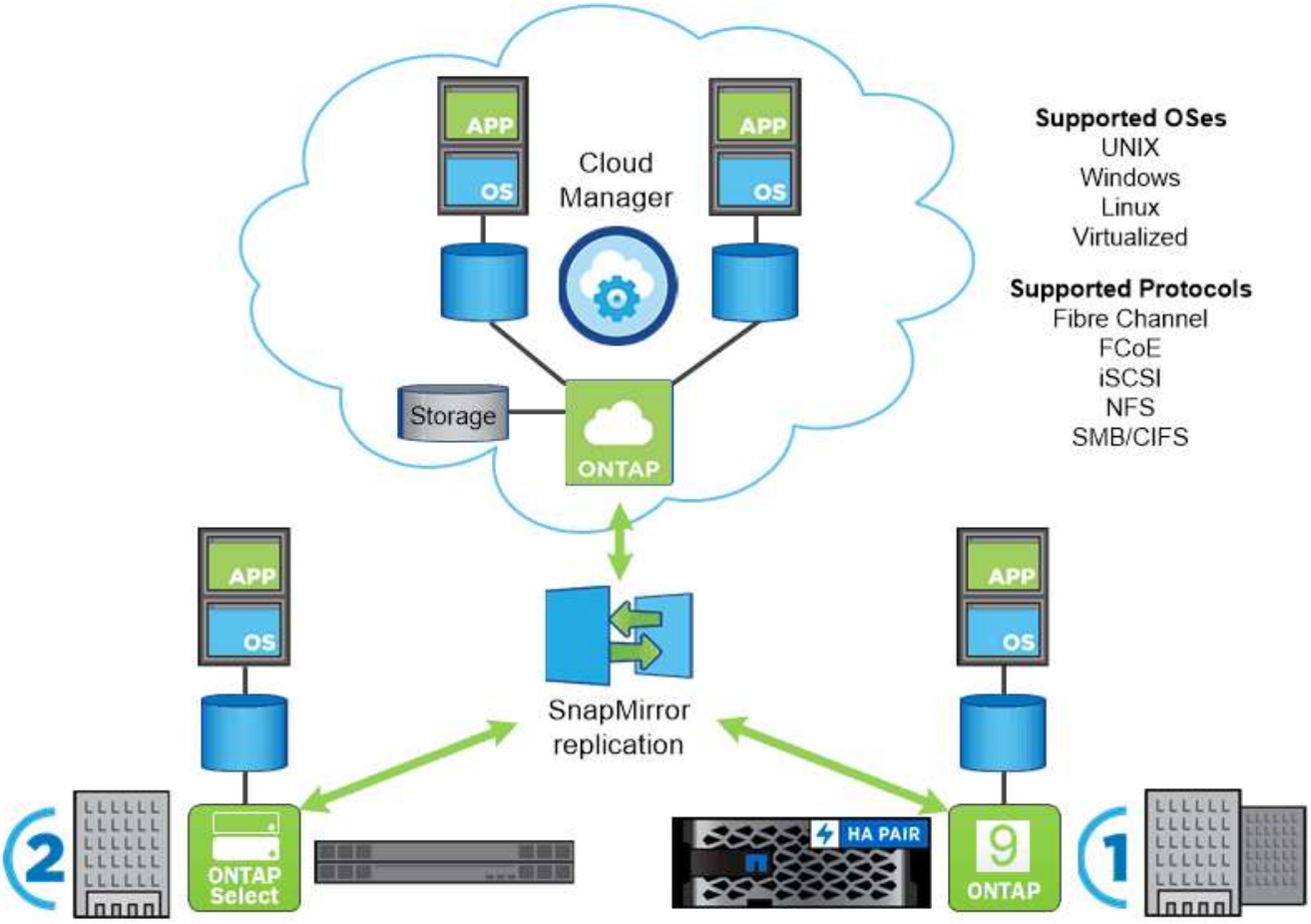
O ONTAP oferece os seguintes recursos:

- Um sistema de armazenamento unificado com acesso simultâneo a dados e gerenciamento de protocolos NFS, CIFS, iSCSI, FC, FCoE e FC-NVMe.
- Diferentes modelos de implantação incluem configurações de hardware all-flash, híbridas e all-HDD no local; plataformas de armazenamento baseadas em VM em um hipervisor compatível, como o ONTAP Select; e na nuvem como Cloud Volumes ONTAP.

- Maior eficiência de armazenamento de dados em sistemas ONTAP com suporte para hierarquização automática de dados, compactação de dados em linha, desduplicação e compactação.
- Armazenamento baseado em carga de trabalho e controlado por QoS.
- Integração perfeita com uma nuvem pública para hierarquização e proteção de dados. O ONTAP também fornece recursos robustos de proteção de dados que o diferenciam em qualquer ambiente:
  - * Cópias do NetApp Snapshot.* Um backup rápido de dados em um momento específico usando uma quantidade mínima de espaço em disco, sem sobrecarga de desempenho adicional.
  - * NetApp SnapMirror.* Espelha as cópias instantâneas de dados de um sistema de armazenamento para outro. O ONTAP também oferece suporte ao espelhamento de dados para outras plataformas físicas e serviços nativos da nuvem.
  - * NetApp SnapLock.* Administração eficiente de dados não regraváveis, gravando-os em volumes especiais que não podem ser substituídos ou apagados por um período designado.
  - * NetApp SnapVault.* Faz backup de dados de vários sistemas de armazenamento para uma cópia central do Snapshot que serve como backup para todos os sistemas designados.
  - * NetApp SyncMirror.* Fornece espelhamento de dados em nível RAID em tempo real para dois plexos diferentes de discos conectados fisicamente ao mesmo controlador.
  - * NetApp SnapRestore.* Fornece restauração rápida de dados de backup sob demanda a partir de cópias de Snapshot.
  - * NetApp FlexClone.* Fornece provisionamento instantâneo de uma cópia totalmente legível e gravável de um volume NetApp com base em uma cópia de instantâneo.

Para mais informações sobre o ONTAP, consulte o "Centro de Documentação ONTAP 9" .

O NetApp ONTAP está disponível no local, virtualizado ou na nuvem.



Plataformas NetApp

NetApp AFF/ FAS

A NetApp fornece plataformas de armazenamento robustas all-flash (AFF) e híbridas escaláveis (FAS), feitas sob medida com desempenho de baixa latência, proteção de dados integrada e suporte a vários protocolos.

Ambos os sistemas são equipados com o software de gerenciamento de dados NetApp ONTAP , o software de gerenciamento de dados mais avançado do setor para gerenciamento de armazenamento simplificado, integrado à nuvem e de alta disponibilidade, para oferecer velocidade, eficiência e segurança de nível empresarial que sua estrutura de dados precisa.

Para mais informações sobre as plataformas NETAPP AFF/ FAS , clique "aqui" .

ONTAP Select

O ONTAP Select é uma implantação definida por software do NetApp ONTAP que pode ser implantada em um hipervisor no seu ambiente. Ele pode ser instalado no VMware vSphere ou no KVM e fornece toda a funcionalidade e experiência de um sistema ONTAP baseado em hardware.

Para mais informações sobre o ONTAP Select, clique em "aqui" .

Cloud Volumes ONTAP

O NetApp Cloud Volumes ONTAP é uma versão do NetApp ONTAP implantada na nuvem, disponível para

implantação em diversas nuvens públicas, incluindo: Amazon AWS, Microsoft Azure e Google Cloud.

Para obter mais informações sobre o Cloud Volumes ONTAP, clique em "aqui" .

Amazon FSx ONTAP

O Amazon FSx ONTAP fornece armazenamento compartilhado totalmente gerenciado na Nuvem AWS com os recursos populares de acesso e gerenciamento de dados do ONTAP. Para obter mais informações sobre o Amazon FSx ONTAP, clique em "aqui" .

Azure NetApp Files

O Azure NetApp Files é um serviço de armazenamento de arquivos nativo do Azure, de primeira linha, de nível empresarial e de alto desempenho. Ele fornece Volumes como um serviço para o qual você pode criar contas NetApp , pools de capacidade e volumes. Você também pode selecionar níveis de serviço e desempenho e gerenciar a proteção de dados. Você pode criar e gerenciar compartilhamentos de arquivos de alto desempenho, altamente disponíveis e escaláveis usando os mesmos protocolos e ferramentas com os quais você está familiarizado e confia no local. Para obter mais informações sobre o Azure NetApp Files, clique em "aqui" .

Google Cloud NetApp Volumes

O Google Cloud NetApp Volumes é um serviço de armazenamento de dados totalmente gerenciado e baseado em nuvem que fornece recursos avançados de gerenciamento de dados e desempenho altamente escalável. Ele permite que você mova aplicativos baseados em arquivos para o Google Cloud. Ele tem suporte integrado para os protocolos Network File System (NFSv3 e NFSv4.1) e Server Message Block (SMB), para que você não precise reestruturar seus aplicativos e possa continuar a obter armazenamento persistente para eles. Para obter mais informações sobre o Google Cloud NetApp VolumesP, clique em "aqui" .

NetApp Element: Red Hat OpenShift com NetApp

O software NetApp Element oferece desempenho modular e escalável, com cada nó de armazenamento fornecendo capacidade e taxa de transferência garantidas ao ambiente. Os sistemas NetApp Element podem ser dimensionados de 4 a 100 nós em um único cluster e oferecem vários recursos avançados de gerenciamento de armazenamento.

Para obter mais informações sobre os sistemas de armazenamento NetApp Element , visite o "Site da NetApp Solidfire" .

Redirecionamento de login iSCSI e recursos de autocorreção

O software NetApp Element utiliza o protocolo de armazenamento iSCSI, uma maneira padrão de encapsular comandos SCSI em uma rede TCP/IP tradicional. Quando os padrões SCSI mudam ou quando o

desempenho das redes Ethernet melhora, o protocolo de armazenamento iSCSI é beneficiado sem a necessidade de nenhuma alteração.

Embora todos os nós de armazenamento tenham um IP de gerenciamento e um IP de armazenamento, o software NetApp Element anuncia um único endereço IP virtual de armazenamento (endereço SVIP) para todo o tráfego de armazenamento no cluster. Como parte do processo de login iSCSI, o armazenamento pode responder que o volume de destino foi movido para um endereço diferente e, portanto, não pode prosseguir com o processo de negociação. O host então emite novamente a solicitação de login para o novo endereço em um processo que não requer reconfiguração do lado do host. Esse processo é conhecido como redirecionamento de login iSCSI.

O redirecionamento de login iSCSI é uma parte fundamental do cluster de software NetApp Element . Quando uma solicitação de login do host é recebida, o nó decide qual membro do cluster deve manipular o tráfego com base no IOPS e nos requisitos de capacidade do volume. Os volumes são distribuídos pelo cluster de software NetApp Element e são redistribuídos se um único nó estiver lidando com muito tráfego para seus volumes ou se um novo nó for adicionado. Várias cópias de um determinado volume são alocadas na matriz.

Dessa forma, se uma falha de nó for seguida por redistribuição de volume, não haverá efeito na conectividade do host além de um logout e login com redirecionamento para o novo local. Com o redirecionamento de login iSCSI, um cluster de software NetApp Element é uma arquitetura auto-reparável e escalável, capaz de atualizações e operações sem interrupções.

QoS do cluster de software NetApp Element

Um cluster de software NetApp Element permite que o QoS seja configurado dinamicamente por volume. Você pode usar configurações de QoS por volume para controlar o desempenho do armazenamento com base nos SLAs que você definir. Os três parâmetros configuráveis a seguir definem o QoS:

- **IOPS mínimo.** O número mínimo de IOPS sustentados que o cluster de software NetApp Element fornece a um volume. O IOPS mínimo configurado para um volume é o nível de desempenho garantido para um volume. O desempenho por volume não cai abaixo desse nível.
- **IOPS máximo.** O número máximo de IOPS sustentados que o cluster de software NetApp Element fornece a um volume específico.
- **IOPS de estouro.** O número máximo de IOPS permitido em um cenário de burst curto. A duração do burst é configurável, com um padrão de 1 minuto. Se um volume estiver abaixo do nível máximo de IOPS, créditos de burst serão acumulados. Quando os níveis de desempenho se tornam muito altos e são exigidos, pequenos picos de IOPS além do IOPS máximo são permitidos no volume.

Multilocação

A multilocação segura é alcançada com os seguintes recursos:

- **Autenticação segura.** O Protocolo de Autenticação Challenge-Handshake (CHAP) é usado para acesso seguro ao volume. O Lightweight Directory Access Protocol (LDAP) é usado para acesso seguro ao cluster para gerenciamento e geração de relatórios.
- **Grupos de acesso a volume (VAGs).** Opcionalmente, os VAGs podem ser usados no lugar da autenticação, mapeando qualquer número de Nomes Qualificados iSCSI (IQNs) específicos do iniciador iSCSI para um ou mais volumes. Para acessar um volume em um VAG, o IQN do iniciador deve estar na lista de IQN permitidos para o grupo de volumes.
- **LANs virtuais de locatário (VLANs).** No nível de rede, a segurança de rede de ponta a ponta entre iniciadores iSCSI e o cluster de software NetApp Element é facilitada pelo uso de VLANs. Para qualquer VLAN criada para isolar uma carga de trabalho ou um locatário, o NetApp Element Software cria um endereço SVIP de destino iSCSI separado que pode ser acessado somente por meio da VLAN específica.

- **VLANs habilitadas para VRF.** Para dar ainda mais suporte à segurança e à escalabilidade no data center, o software NetApp Element permite que você habilite qualquer VLAN de locatário para funcionalidade semelhante a VRF. Este recurso adiciona estas duas capacidades principais:
  - **Roteamento L3 para um endereço SVIP de locatário.** Este recurso permite que você situe iniciadores iSCSI em uma rede ou VLAN separada daquela do cluster de software NetApp Element .
  - **Sub-redes IP sobrepostas ou duplicadas.** Este recurso permite que você adicione um modelo aos ambientes de locatário, permitindo que cada VLAN de locatário receba endereços IP da mesma sub-rede IP. Esse recurso pode ser útil para ambientes de provedores de serviço onde a escala e a preservação do espaço IP são importantes.

Eficiências de armazenamento empresarial

O cluster de software NetApp Element aumenta a eficiência e o desempenho geral do armazenamento. Os seguintes recursos são executados em linha, estão sempre ativos e não exigem configuração manual pelo usuário:

- **Desduplicação.** O sistema armazena apenas blocos exclusivos de 4K. Quaisquer blocos duplicados de 4K são automaticamente associados a uma versão já armazenada dos dados. Os dados estão em unidades de bloco e são espelhados usando a proteção de dados Helix do software NetApp Element . Este sistema reduz significativamente o consumo de capacidade e as operações de gravação dentro do sistema.
- **Compressão.** A compactação é realizada em linha antes dos dados serem gravados na NVRAM. Os dados são compactados, armazenados em blocos de 4K e permanecem compactados no sistema. Essa compactação reduz significativamente o consumo de capacidade, as operações de gravação e o consumo de largura de banda no cluster.
- **Provisionamento fino.** Esse recurso fornece a quantidade certa de armazenamento no momento em que você precisa, eliminando o consumo de capacidade causado por volumes superprovisionados ou subutilizados.
- **Hélice.** Os metadados de um volume individual são armazenados em uma unidade de metadados e replicados para uma unidade de metadados secundária para redundância.

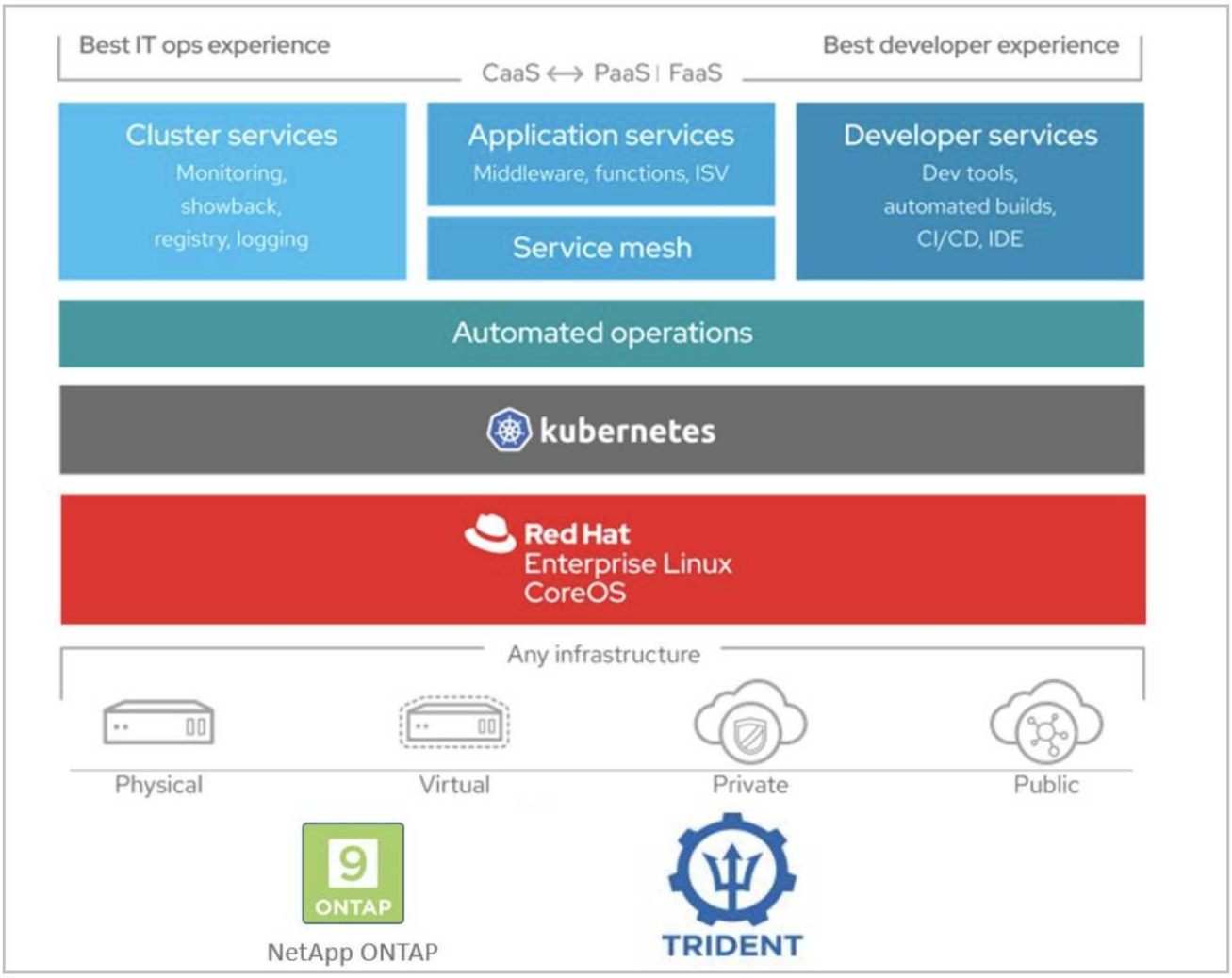
> O Element foi projetado para automação. Todos os recursos de armazenamento estão disponíveis por meio de APIs. Essas APIs são o único método que a interface do usuário usa para controlar o sistema.

Integrações de armazenamento NetApp

Saiba mais sobre a integração do NetApp Trident com o Red Hat OpenShift

Saiba mais sobre o NetApp Trident Protect, que foi validado para gerenciamento de aplicativos e armazenamento persistente para a solução OpenShift Virtualization.

O Trident, um orquestrador e provedor de armazenamento de código aberto mantido pela NetApp , ajuda você a orquestrar e gerenciar dados persistentes em ambientes baseados em contêineres, como o Trident Hat OpenShift.



As páginas a seguir contêm informações adicionais sobre os produtos NetApp que foram validados para gerenciamento de aplicativos e armazenamento persistente na solução Red Hat OpenShift com NetApp :

- "Documentação Trident"
- "Documentação de proteção do Trident"

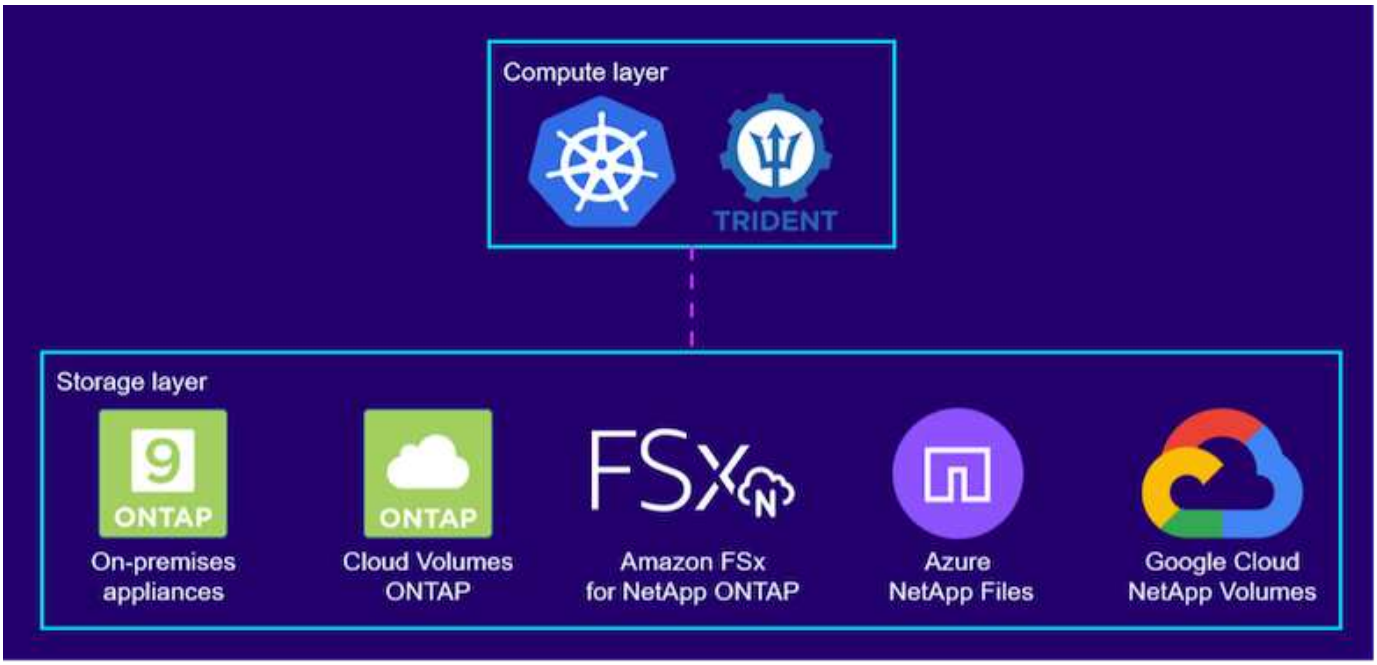
NetApp Trident

Visão geral do Trident

O Trident é um orquestrador de armazenamento de código aberto e totalmente suportado para contêineres e distribuições Kubernetes, incluindo o Red Hat OpenShift. O Trident funciona com todo o portfólio de armazenamento da NetApp , incluindo os sistemas de armazenamento NetApp ONTAP e Element, e também oferece suporte a conexões NFS e iSCSI. O Trident acelera o fluxo de trabalho do DevOps permitindo que os usuários finais provisionem e gerenciem o armazenamento de seus sistemas de armazenamento NetApp sem exigir a intervenção de um administrador de armazenamento.

Um administrador pode configurar vários backends de armazenamento com base nas necessidades do

projeto e nos modelos de sistema de armazenamento que permitem recursos avançados de armazenamento, incluindo compactação, tipos de disco específicos ou níveis de QoS que garantem um determinado nível de desempenho. Depois de definidos, esses backends podem ser usados pelos desenvolvedores em seus projetos para criar declarações de volume persistentes (PVCs) e anexar armazenamento persistente aos seus contêineres sob demanda.



O Trident tem um ciclo de desenvolvimento rápido e, assim como o Kubernetes, é lançado quatro vezes por ano.

Uma matriz de suporte para qual versão do Trident foi testada com qual distribuição do Kubernetes pode ser encontrada "aqui" .

Por favor, consulte o"Documentação do produto Trident" para detalhes de instalação e configuração.

Baixar Trident

Para instalar o Trident no cluster de usuários implantado e provisionar um volume persistente, conclua as seguintes etapas:

1. Baixe o arquivo de instalação para a estação de trabalho do administrador e extraia o conteúdo. A versão atual do Trident pode ser baixada "aqui" .
2. Extraia a instalação do Trident do pacote baixado.

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

Instalar o Trident Operator com Helm

1. Primeiro defina a localização do cluster do usuário `kubeconfig` arquivo como uma variável de ambiente para que você não precise referenciá-lo, porque o Trident não tem opção para passar esse arquivo.

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

2. Execute o comando Helm para instalar o operador Trident do tarball no diretório helm enquanto cria o namespace trident no seu cluster de usuários.

```
[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. Você pode verificar se o Trident foi instalado com sucesso verificando os pods que estão sendo executados no namespace ou usando o binário tridentctl para verificar a versão 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          |
+----------------+----------------+
```

Em alguns casos, os ambientes do cliente podem exigir a personalização da implantação do Trident . Nesses casos, também é possível instalar manualmente o operador Trident e atualizar os manifestos incluídos para personalizar a implantação.

Instalar manualmente o Operador Trident

1. Primeiro, defina a localização do cluster do usuário `kubeconfig` arquivo como uma variável de ambiente para que você não precise referenciá-lo, porque o Trident não tem opção para passar esse arquivo.

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

2. O `trident-installer` O diretório contém manifestos para definir todos os recursos necessários. Usando os manifestos apropriados, crie o `TridentOrchestrator` definição 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. Se não houver um, crie um namespace Trident no seu cluster usando o manifesto fornecido.

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

4. Crie os recursos necessários para a implantação do operador Trident , como um `ServiceAccount` para o operador, um `ClusterRole` e `ClusterRoleBinding` para o `ServiceAccount` , um dedicado `PodSecurityPolicy` , ou o próprio 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. Você pode verificar o status do operador após sua implantação com os seguintes 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. Com o operador implantado, agora podemos usá-lo para instalar o Trident. Isso requer a criação de um `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. Você pode verificar se o Trident foi instalado com sucesso verificando os pods que estão sendo executados no namespace ou usando o binário tridentctl para verificar a versão 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 nós de trabalho para armazenamento

NFS

A maioria das distribuições do Kubernetes vem com os pacotes e utilitários para montar backends NFS instalados por padrão, incluindo o Red Hat OpenShift.

Entretanto, para o NFSv3, não há mecanismo para negociar a simultaneidade entre o cliente e o servidor. Portanto, o número máximo de entradas da tabela de slots sunrpc do lado do cliente deve ser sincronizado manualmente com o valor suportado no servidor para garantir o melhor desempenho para a conexão NFS sem que o servidor tenha que diminuir o tamanho da janela da conexão.

Para o ONTAP, o número máximo suportado de entradas na tabela de slots sunrpc é 128, ou seja, o ONTAP

pode atender a 128 solicitações NFS simultâneas por vez. Entretanto, por padrão, o Red Hat CoreOS/Red Hat Enterprise Linux tem no máximo 65.536 entradas na tabela de slots sunrpc por conexão. Precisamos definir esse valor como 128 e isso pode ser feito usando o Machine Config Operator (MCO) no OpenShift.

Para modificar as entradas máximas da tabela de slots sunrpc nos nós de trabalho do OpenShift, conclua as seguintes etapas:

1. Efetue login no console da web do OCP e navegue até Computação > Configurações da máquina. Clique em Criar configuração de máquina. Copie e cole o arquivo YAML e clique em Criar.

```
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. Após a criação do MCO, a configuração precisa ser aplicada em todos os nós de trabalho e reinicializada um por um. Todo o processo leva aproximadamente de 20 a 30 minutos. Verifique se a configuração da máquina foi aplicada usando `oc get mcp` e certifique-se de que o pool de configuração da máquina para trabalhadores esteja atualizado.

```
[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 nós de trabalho para permitir o mapeamento de volumes de armazenamento em bloco por meio

do protocolo iSCSI, você deve instalar os pacotes necessários para dar suporte a essa funcionalidade.

No Red Hat OpenShift, isso é feito aplicando um MCO (Operador de Configuração de Máquina) ao seu cluster após sua implantação.

Para configurar os nós de trabalho para executar serviços iSCSI, conclua as seguintes etapas:

1. Efetue login no console da web do OCP e navegue até Computação > Configurações da máquina. Clique em Criar configuração de máquina. Copie e cole o arquivo YAML e clique em Criar.

   Quando não estiver usando multipathing:

```
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: ""
```

   Ao usar multipathing:

```
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. Após a criação da configuração, leva aproximadamente 20 a 30 minutos para aplicá-la aos nós de trabalho e recarregá-los. Verifique se a configuração da máquina foi aplicada usando `oc get mcp` e certifique-se de que o pool de configuração da máquina para trabalhadores esteja atualizado. Você também pode efetuar login nos nós de trabalho para confirmar se o serviço iscsid está em execução (e se o serviço multipathd está em execução se estiver usando multipathing).

```
[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
```

Também é possível confirmar se o MachineConfig foi aplicado com sucesso e os serviços foram iniciados conforme o esperado executando o comando `oc debug` comando com os sinalizadores apropriados.

Criar backends de sistema de armazenamento

Após concluir a instalação do Trident Operator, você deve configurar o backend para a plataforma de

armazenamento NetApp específica que está usando. Siga os links abaixo para continuar a instalação e configuração do Trident.

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

Configuração do NetApp ONTAP NFS

Para habilitar a integração do Trident com o sistema de armazenamento NetApp ONTAP , você deve criar um backend que permita a comunicação com o sistema de armazenamento.

1. Existem arquivos de backend de amostra disponíveis no arquivo de instalação baixado no `sample-input` hierarquia de pastas. Para sistemas NetApp ONTAP que atendem NFS, copie o `backend-ontap-nas.json` arquivo para seu diretório de trabalho e edite o arquivo.

```
[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 os valores backendName, managementLIF, dataLIF, svm, username e password neste arquivo.

```
{
    "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"
}
```

É uma prática recomendada definir o valor backendName personalizado como uma combinação do storageDriverName e do dataLIF que atende ao NFS para facilitar a identificação.

3. Com esse arquivo de backend em vigor, execute o seguinte comando para criar seu primeiro 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. Com o backend criado, você deve criar uma classe de armazenamento. Assim como no backend, há um arquivo de classe de armazenamento de exemplo que pode ser editado para o ambiente disponível na pasta sample-inputs. Copie-o para o diretório de trabalho e faça as edições necessárias para refletir o backend criado.

```
[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. A única edição que deve ser feita neste arquivo é definir o `backendType` valor para o nome do driver de armazenamento do backend recém-criado. Observe também o valor do campo nome, que deve ser referenciado em uma etapa posterior.

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

> Existe um campo opcional chamado `fsType` que é definido neste arquivo. Esta linha pode ser excluída em backends NFS.

6. Execute o `oc` comando para criar a classe de armazenamento.

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

7. Com a classe de armazenamento criada, você deve criar a primeira reivindicação de volume persistente (PVC). Há uma amostra `pvc-basic.yaml` arquivo que pode ser usado para executar esta ação localizado também em 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. A única edição que deve ser feita neste arquivo é garantir que o `storageClassName` campo corresponde ao que acabou de ser criado. A definição de PVC pode ser ainda mais personalizada conforme exigido pela carga de trabalho a ser provisionada.

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

9. Crie o PVC emitindo o `oc` comando. A criação pode levar algum tempo dependendo do tamanho do volume de apoio que está sendo criado, então você pode acompanhar o processo conforme ele é concluído.

```
[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
```

Configuração do NetApp ONTAP iSCSI

Para habilitar a integração do Trident com o sistema de armazenamento NetApp ONTAP , você deve criar um backend que permita a comunicação com o sistema de armazenamento.

1. Existem arquivos de backend de amostra disponíveis no arquivo de instalação baixado no `sample-input`

hierarquia de pastas. Para sistemas NetApp ONTAP que atendem iSCSI, copie o `backend-ontap-san.json` arquivo para seu diretório de trabalho e edite o arquivo.

```
[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 os valores managementLIF, dataLIF, svm, username e password neste arquivo.

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

3. Com esse arquivo de backend em vigor, execute o seguinte comando para criar seu primeiro 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. Com o backend criado, você deve criar uma classe de armazenamento. Assim como no backend, há um arquivo de classe de armazenamento de exemplo que pode ser editado para o ambiente disponível na pasta sample-inputs. Copie-o para o diretório de trabalho e faça as edições necessárias para refletir o backend criado.

```
[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. A única edição que deve ser feita neste arquivo é definir o `backendType` valor para o nome do driver de armazenamento do backend recém-criado. Observe também o valor do campo nome, que deve ser referenciado em uma etapa posterior.

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

> Existe um campo opcional chamado `fsType` que é definido neste arquivo. Em backends iSCSI, esse valor pode ser definido para um tipo específico de sistema de arquivos Linux (XFS, ext4, etc.) ou pode ser excluído para permitir que o OpenShift decida qual sistema de arquivos usar.

6. Execute o `oc` comando para criar a classe de armazenamento.

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

7. Com a classe de armazenamento criada, você deve criar a primeira reivindicação de volume persistente (PVC). Há uma amostra `pvc-basic.yaml` arquivo que pode ser usado para executar esta ação localizado também em 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. A única edição que deve ser feita neste arquivo é garantir que o `storageClassName` campo corresponde ao que acabou de ser criado. A definição de PVC pode ser ainda mais personalizada conforme exigido pela carga de trabalho a ser provisionada.

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

9. Crie o PVC emitindo o `oc` comando. A criação pode levar algum tempo dependendo do tamanho do volume de apoio que está sendo criado, então você pode acompanhar o processo conforme ele é concluído.

```
[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
```

Configuração iSCSI do NetApp Element

Para habilitar a integração do Trident com o sistema de armazenamento NetApp Element , você deve criar um backend que permita a comunicação com o sistema de armazenamento usando o protocolo iSCSI.

1. Existem arquivos de backend de amostra disponíveis no arquivo de instalação baixado no `sample-input` hierarquia de pastas. Para sistemas NetApp Element que atendem iSCSI, copie o `backend-solidfire.json` arquivo para seu diretório de trabalho e edite o arquivo.

```
[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 o usuário, a senha e o valor MVIP no `EndPoint` linha.

   b. Editar o `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. Com esse arquivo de back-end instalado, execute o seguinte comando para criar seu primeiro 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. Com o backend criado, você deve criar uma classe de armazenamento. Assim como no backend, há um arquivo de classe de armazenamento de exemplo que pode ser editado para o ambiente disponível na pasta sample-inputs. Copie-o para o diretório de trabalho e faça as edições necessárias para refletir o backend criado.

```
[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. A única edição que deve ser feita neste arquivo é definir o `backendType` valor para o nome do driver de armazenamento do backend recém-criado. Observe também o valor do campo nome, que deve ser referenciado em uma etapa posterior.

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

Existe um campo opcional chamado `fsType` que é definido neste arquivo. Em backends iSCSI, esse valor pode ser definido para um tipo específico de sistema de arquivos Linux (XFS, ext4 e assim por diante) ou pode ser excluído para permitir que o OpenShift decida qual sistema de arquivos usar.

5. Execute o `oc` comando para criar a classe de armazenamento.

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

6. Com a classe de armazenamento criada, você deve criar a primeira reivindicação de volume persistente (PVC). Há uma amostra `pvc-basic.yaml` arquivo que pode ser usado para executar esta ação localizado também em 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. A única edição que deve ser feita neste arquivo é garantir que o `storageClassName` campo corresponde ao que acabou de ser criado. A definição de PVC pode ser ainda mais personalizada conforme exigido pela carga de trabalho a ser provisionada.

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

8. Crie o PVC emitindo o `oc` comando. A criação pode levar algum tempo dependendo do tamanho do volume de apoio que está sendo criado, então você pode acompanhar o processo conforme ele é concluído.

```
[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
```

Opções de configuração avançadas

Explorar opções de balanceador de carga

Explorando opções de balanceador de carga: Red Hat OpenShift com NetApp

Na maioria dos casos, o Red Hat OpenShift disponibiliza aplicativos para o mundo externo por meio de rotas. Um serviço é exposto ao fornecer a ele um nome de host acessível externamente. A rota definida e os pontos de extremidade identificados por seu serviço podem ser consumidos por um roteador OpenShift para fornecer essa conectividade nomeada a clientes externos.

Entretanto, em alguns casos, os aplicativos exigem a implantação e a configuração de balanceadores de carga personalizados para expor os serviços apropriados. Um exemplo disso é o NetApp Trident Protect. Para atender a essa necessidade, avaliamos diversas opções de balanceadores de carga personalizados. Sua instalação e configuração são descritas nesta seção.

As páginas a seguir contêm informações adicionais sobre as opções do balanceador de carga validadas na solução Red Hat OpenShift com NetApp :

- "MetalLB"
- "F5 BIG-IP"

Instalando balanceadores de carga MetalLB: Red Hat OpenShift com NetApp

Esta página lista as instruções de instalação e configuração do balanceador de carga MetalLB.

O MetalLB é um balanceador de carga de rede auto-hospedado instalado no seu cluster OpenShift que permite a criação de serviços OpenShift do tipo balanceador de carga em clusters que não são executados em um provedor de nuvem. Os dois principais recursos do MetalLB que funcionam juntos para dar suporte aos serviços do LoadBalancer são a alocação de endereços e o anúncio externo.

Opções de configuração do MetalLB

Com base em como o MetalLB anuncia o endereço IP atribuído aos serviços do LoadBalancer fora do cluster OpenShift, ele opera em dois modos:

- **Modo de camada 2.** Nesse modo, um nó no cluster OpenShift assume a propriedade do serviço e responde às solicitações ARP para esse IP para torná-lo acessível fora do cluster OpenShift. Como somente o nó anuncia o IP, ele tem um gargalo de largura de banda e limitações de failover lento. Para mais informações, consulte a documentação"aqui" .
- **Modo BGP.** Neste modo, todos os nós no cluster OpenShift estabelecem sessões de peering BGP com um roteador e anunciam as rotas para encaminhar o tráfego para os IPs de serviço. O pré-requisito para isso é integrar o MetalLB com um roteador nessa rede. Devido ao mecanismo de hash no BGP, há certas limitações quando o mapeamento de IP para nó de um serviço muda. Para mais informações, consulte a documentação"aqui" .

Para os fins deste documento, estamos configurando o MetalLB no modo de camada 2.

Instalando o balanceador de carga MetalLB

1. Baixe os recursos do 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 arquivo `metallb.yaml` e remover `spec.template.spec.securityContext` do controlador Deployment e do speaker DaemonSet.

   Linhas a serem deletadas:

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

3. Crie o `metallb-system` espaço para nome.

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

4. Crie o MetalLB CR.

```
[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 o alto-falante MetalLB, conceda ao alto-falante privilégios elevados do DaemonSet para que ele possa executar a configuração de rede necessária para fazer os balanceadores de carga funcionarem.

```
[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 o MetalLB criando um `ConfigMap` no `metallb-system` espaço para nome.

```
[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. Agora, quando os serviços do balanceador de carga são criados, o MetalLB atribui um IP externo aos serviços e anuncia o endereço IP respondendo às solicitações ARP.

> Se você deseja configurar o MetalLB no modo BGP, pule a etapa 6 acima e siga o procedimento na documentação do MetalLB"aqui" .

Instalando balanceadores de carga F5 BIG-IP

O F5 BIG-IP é um Application Delivery Controller (ADC) que oferece um amplo conjunto de serviços avançados de gerenciamento de tráfego e segurança de nível de produção, como balanceamento de carga L4-L7, descarregamento SSL/TLS, DNS, firewall e muito mais. Esses serviços aumentam drasticamente a disponibilidade, a segurança e o desempenho de seus aplicativos.

O F5 BIG-IP pode ser implantado e consumido de várias maneiras: em hardware dedicado, na nuvem ou como um dispositivo virtual local. Consulte a documentação aqui para explorar e implantar o F5 BIG-IP conforme a necessidade.

Para integração eficiente dos serviços F5 BIG-IP com o Red Hat OpenShift, a F5 oferece o BIG-IP Container Ingress Service (CIS). O CIS é instalado como um pod controlador que monitora a API OpenShift para determinadas Definições de Recursos Personalizados (CRDs) e gerencia a configuração do sistema F5 BIG-IP. O F5 BIG-IP CIS pode ser configurado para controlar os tipos de serviço LoadBalancers e Routes no OpenShift.

Além disso, para alocação automática de endereços IP para atender ao tipo LoadBalancer, você pode utilizar o controlador F5 IPAM. O controlador F5 IPAM é instalado como um pod de controlador que monitora a API OpenShift para serviços do LoadBalancer com uma anotação ipamLabel para alocar o endereço IP de um pool pré-configurado.

Esta página lista as instruções de instalação e configuração do controlador F5 BIG-IP CIS e IPAM. Como pré-

requisito, você deve ter um sistema F5 BIG-IP implantado e licenciado. Ele também deve ser licenciado para serviços SDN, que são incluídos por padrão na licença base do BIG-IP VE.

O F5 BIG-IP pode ser implantado no modo autônomo ou em cluster. Para fins desta validação, o F5 BIG-IP foi implantado no modo autônomo, mas, para fins de produção, é preferível ter um cluster de BIG-IPs para evitar um único ponto de falha.

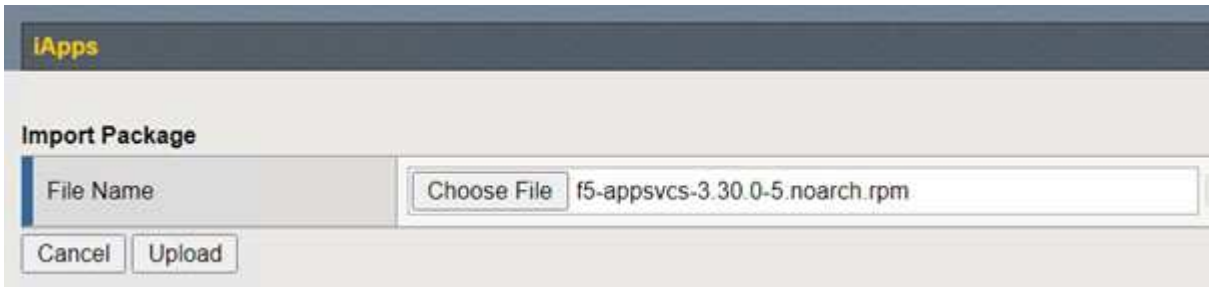
Um sistema F5 BIG-IP pode ser implantado em hardware dedicado, na nuvem ou como um dispositivo virtual local com versões superiores a 12.x para ser integrado ao F5 CIS. Para os fins deste documento, o sistema F5 BIG-IP foi validado como um dispositivo virtual, por exemplo, usando a edição BIG-IP VE.

Lançamentos validados

Tecnologia	Versão do software
Red Hat OpenShift	4,6 EUS, 4,7
Edição F5 BIG-IP VE	16.1.0
Serviço de entrada de contêiner F5	2.5.1
Controlador IPAM F5	0.1.4
F5 AS3	3.30.0

Instalação

1. Instale a extensão F5 Application Services 3 para permitir que sistemas BIG-IP aceitem configurações em JSON em vez de comandos imperativos. Vá para "Repositório F5 AS3 GitHub" e baixe o arquivo RPM mais recente.
2. Efetue login no sistema F5 BIG-IP, navegue até iApps > Package Management LX e clique em Importar.
3. Clique em Escolher arquivo e selecione o arquivo AS3 RPM baixado, clique em OK e depois clique em Carregar.

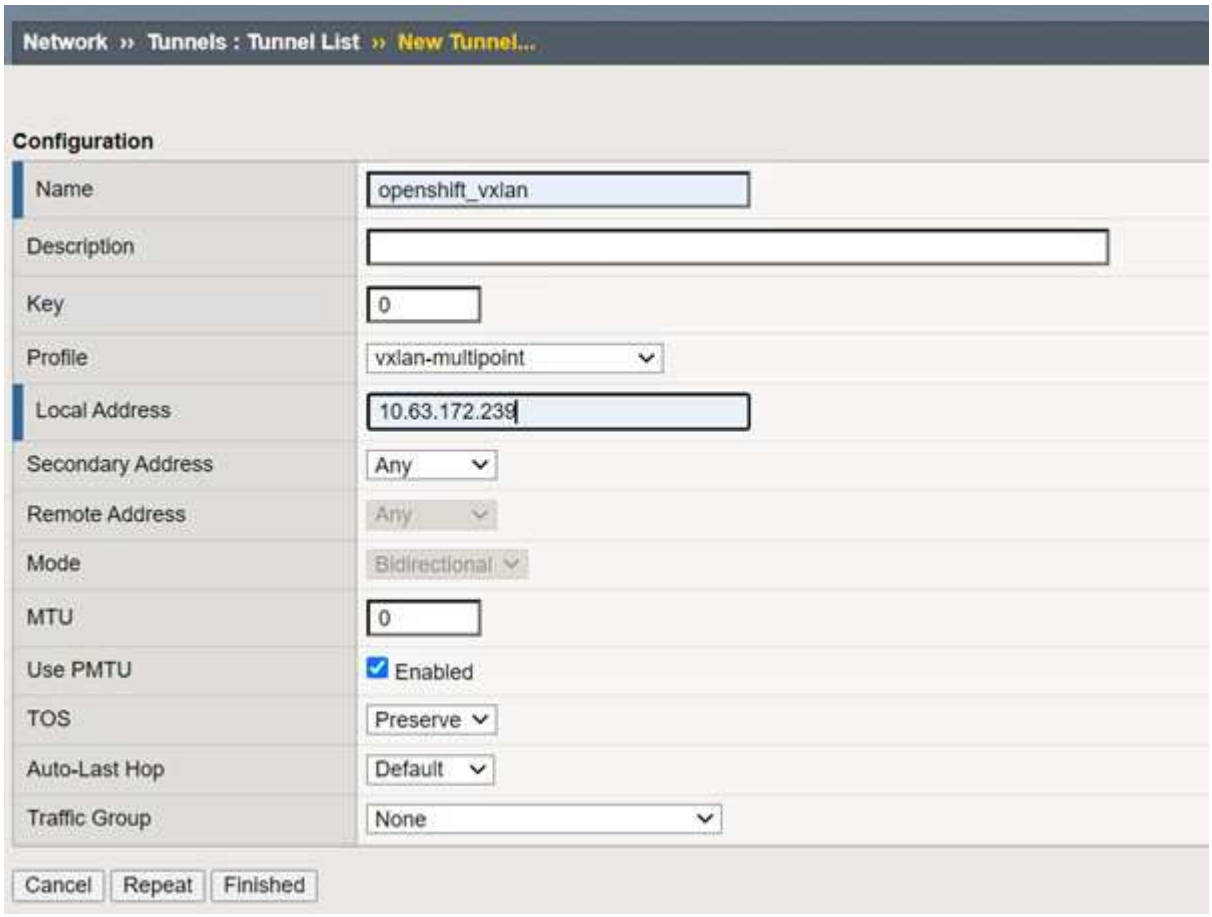


4. Confirme se a extensão AS3 foi instalada com sucesso.

5. Em seguida, configure os recursos necessários para a comunicação entre os sistemas OpenShift e BIG-IP. Primeiro, crie um túnel entre o OpenShift e o servidor BIG-IP criando uma interface de túnel VXLAN no

sistema BIG-IP para o OpenShift SDN. Navegue até Rede > Túneis > Perfis, clique em Criar e defina o Perfil Pai como vxlan e o Tipo de Inundação como Multicast. Digite um nome para o perfil e clique em Concluído.

6. Navegue até Rede > Túneis > Lista de Túneis, clique em Criar e insira o nome e o endereço IP local para o túnel. Selecione o perfil de túnel que foi criado na etapa anterior e clique em Concluído.



7. Efetue login no cluster do Red Hat OpenShift com privilégios de administrador de cluster.
8. Crie uma hostsubnet no OpenShift para o servidor F5 BIG-IP, que estende a sub-rede do cluster OpenShift para o servidor F5 BIG-IP. Baixe a definição YAML da sub-rede do host.

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

9. Edite o arquivo de sub-rede do host e adicione o IP BIG-IP VTEP (túnel VXLAN) para o 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
```

Altere o hostIP e outros detalhes aplicáveis ao seu ambiente.

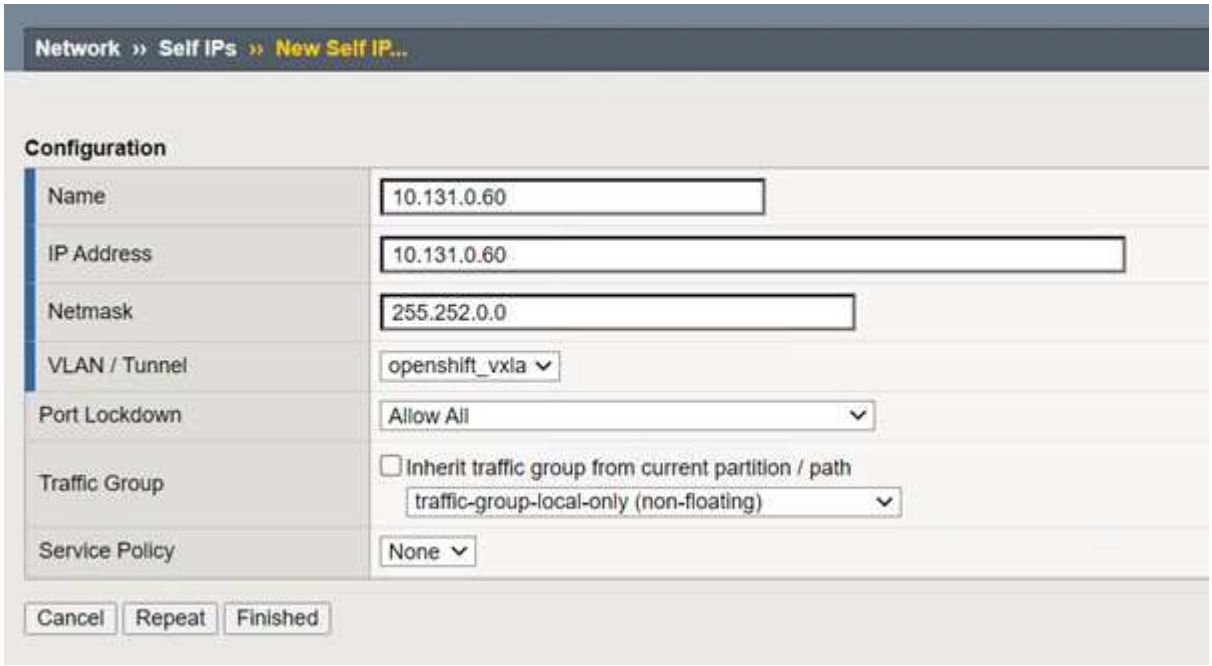
10. Crie o recurso HostSubnet.

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

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

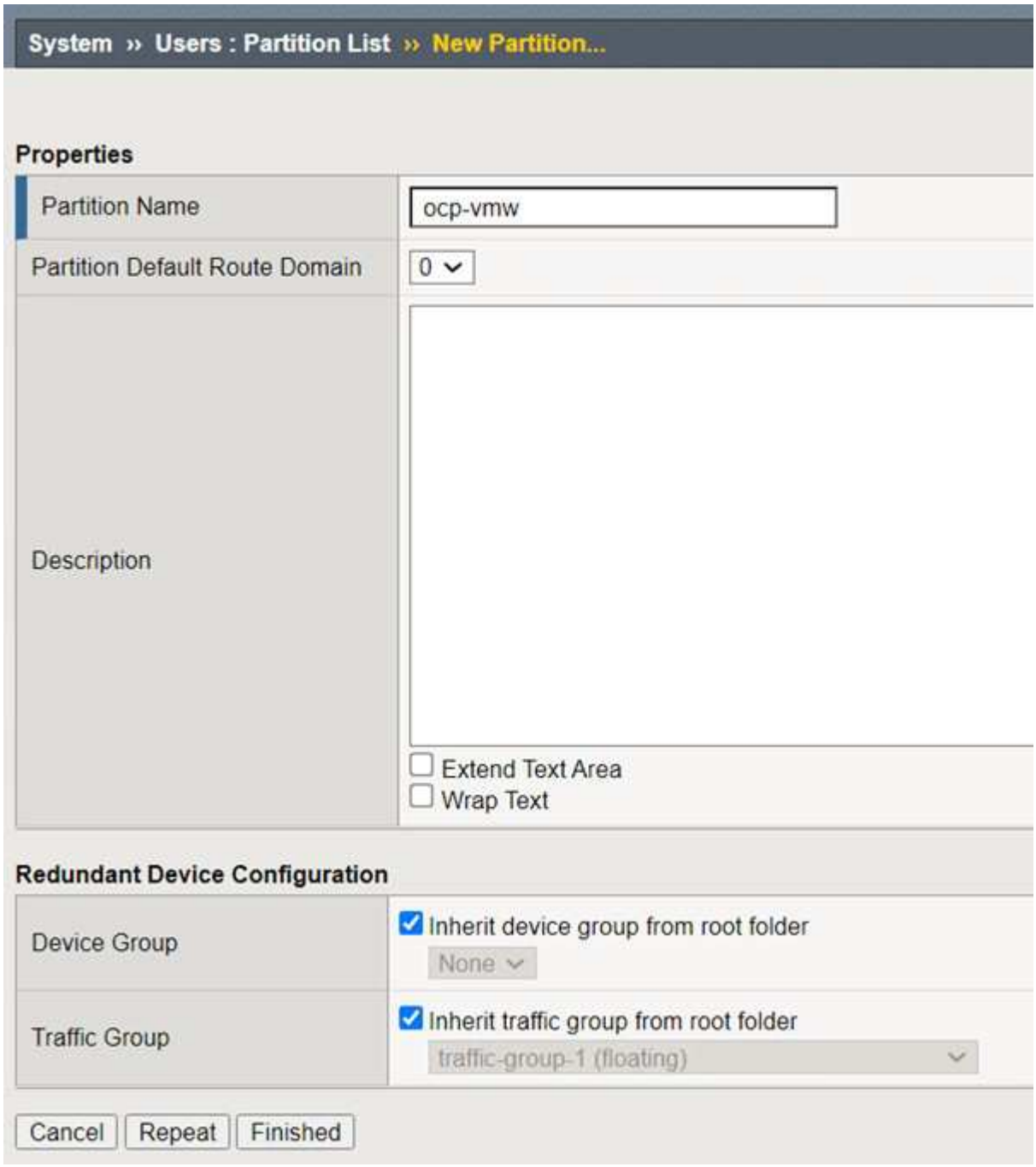
11. Obtenha o intervalo de sub-rede IP do cluster para a sub-rede do host criada para o servidor F5 BIG-IP.

```
[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. Crie um IP próprio no OpenShift VXLAN com um IP no intervalo de sub-rede do host do OpenShift correspondente ao servidor F5 BIG-IP. Efetue login no sistema F5 BIG-IP, navegue até Rede > IPs próprios e clique em Criar. Insira um IP da sub-rede IP do cluster criada para a sub-rede do host F5 BIG-IP, selecione o túnel VXLAN e insira os outros detalhes. Em seguida, clique em Concluído.



13. Crie uma partição no sistema F5 BIG-IP para ser configurada e usada com o CIS. Navegue até Sistema > Usuários > Lista de partições, clique em Criar e insira os detalhes. Em seguida, clique em Concluído.



O F5 recomenda que nenhuma configuração manual seja feita na partição gerenciada pelo CIS.

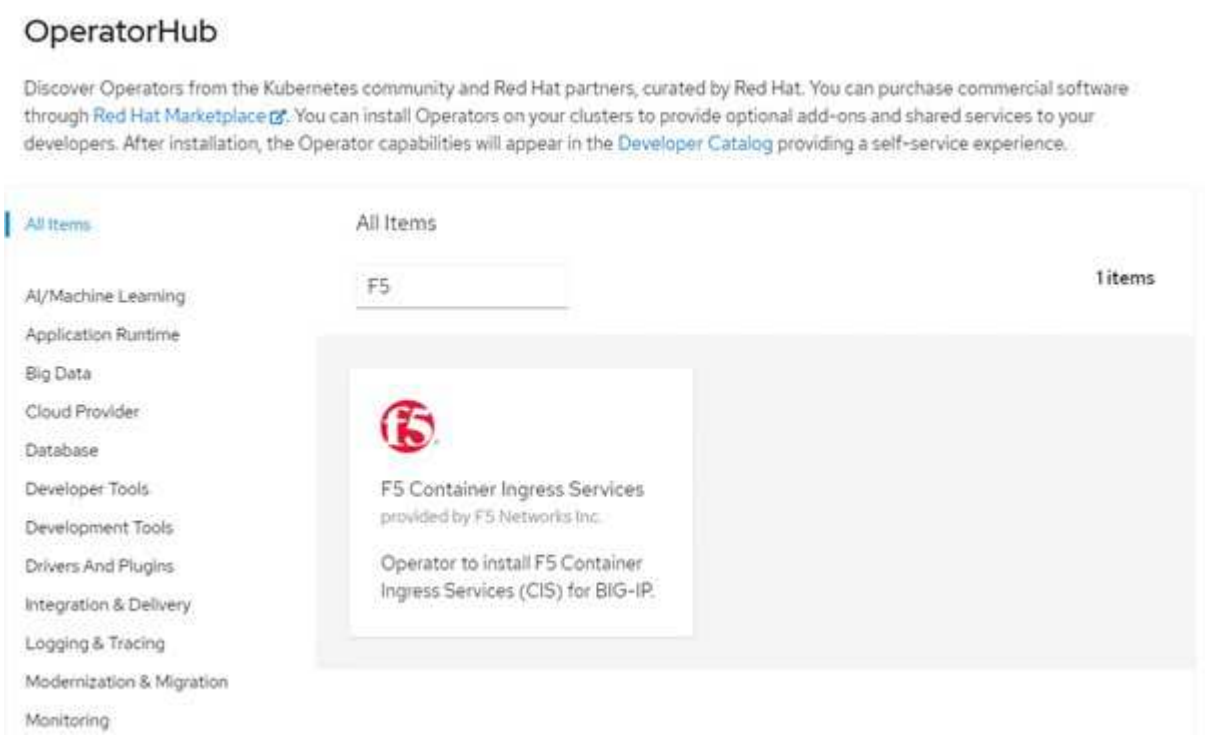
14. Instale o F5 BIG-IP CIS usando o operador do OperatorHub. Efetue login no cluster Red Hat OpenShift com privilégios de administrador de cluster e crie um segredo com as credenciais de login do sistema F5 BIG-IP, que é um pré-requisito para o operador.

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

secret/bigip-login created
```

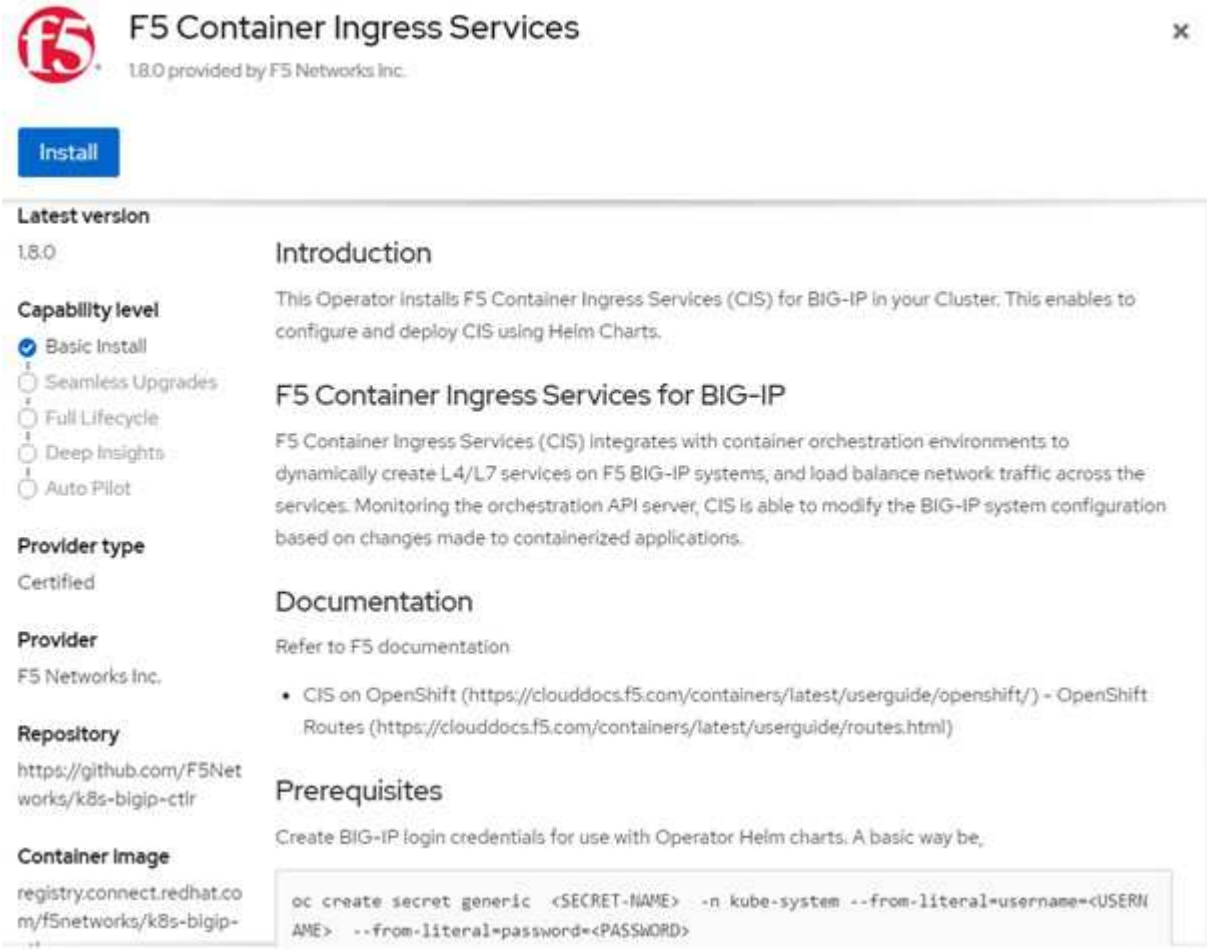
15. Instale os CRDs F5 CIS.

```
[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
```

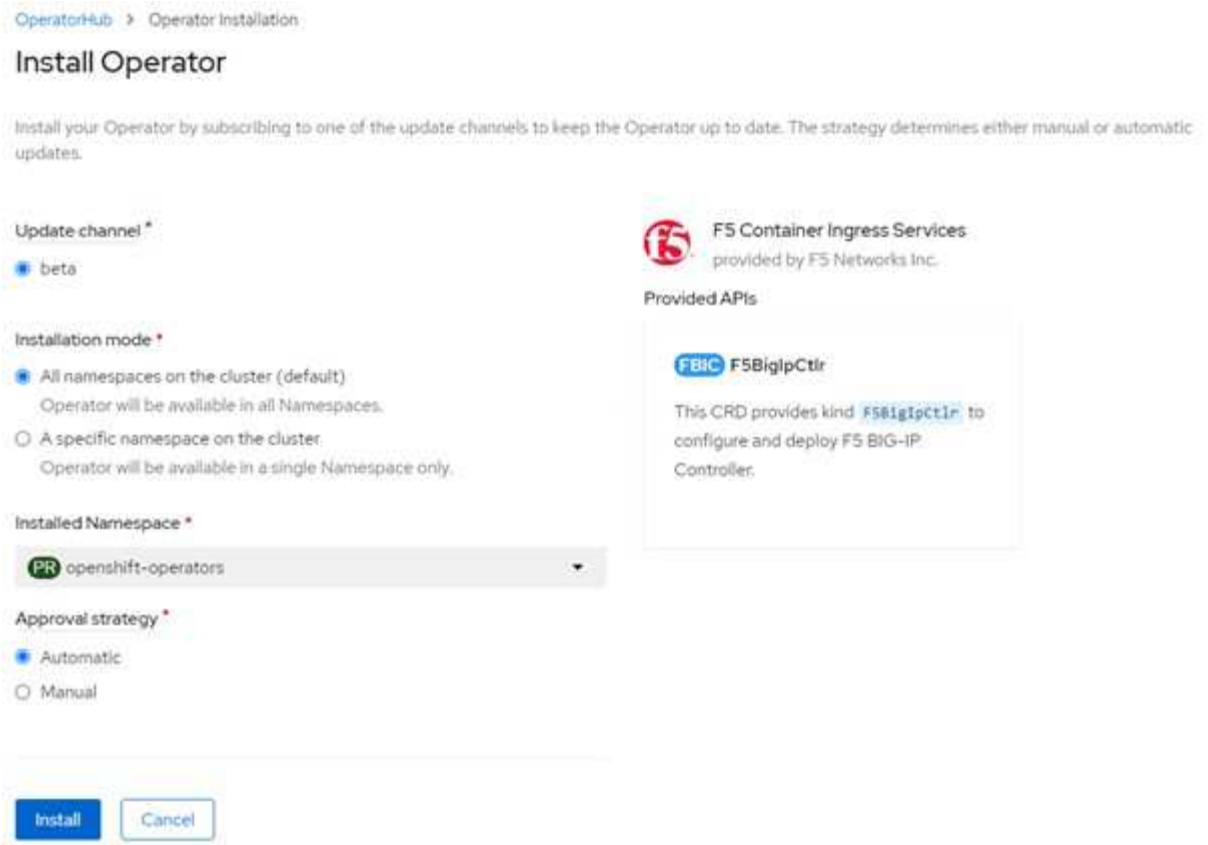
16. Navegue até Operadores > OperatorHub, pesquise a palavra-chave F5 e clique no bloco Serviço de entrada de contêiner F5.



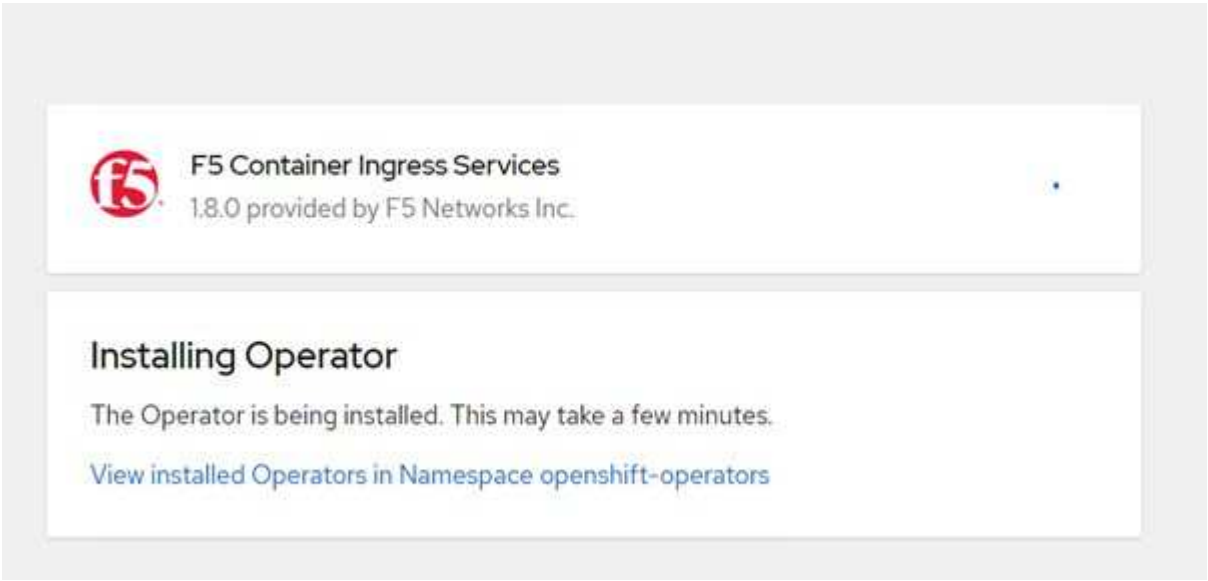
17. Leia as informações do operador e clique em Instalar.



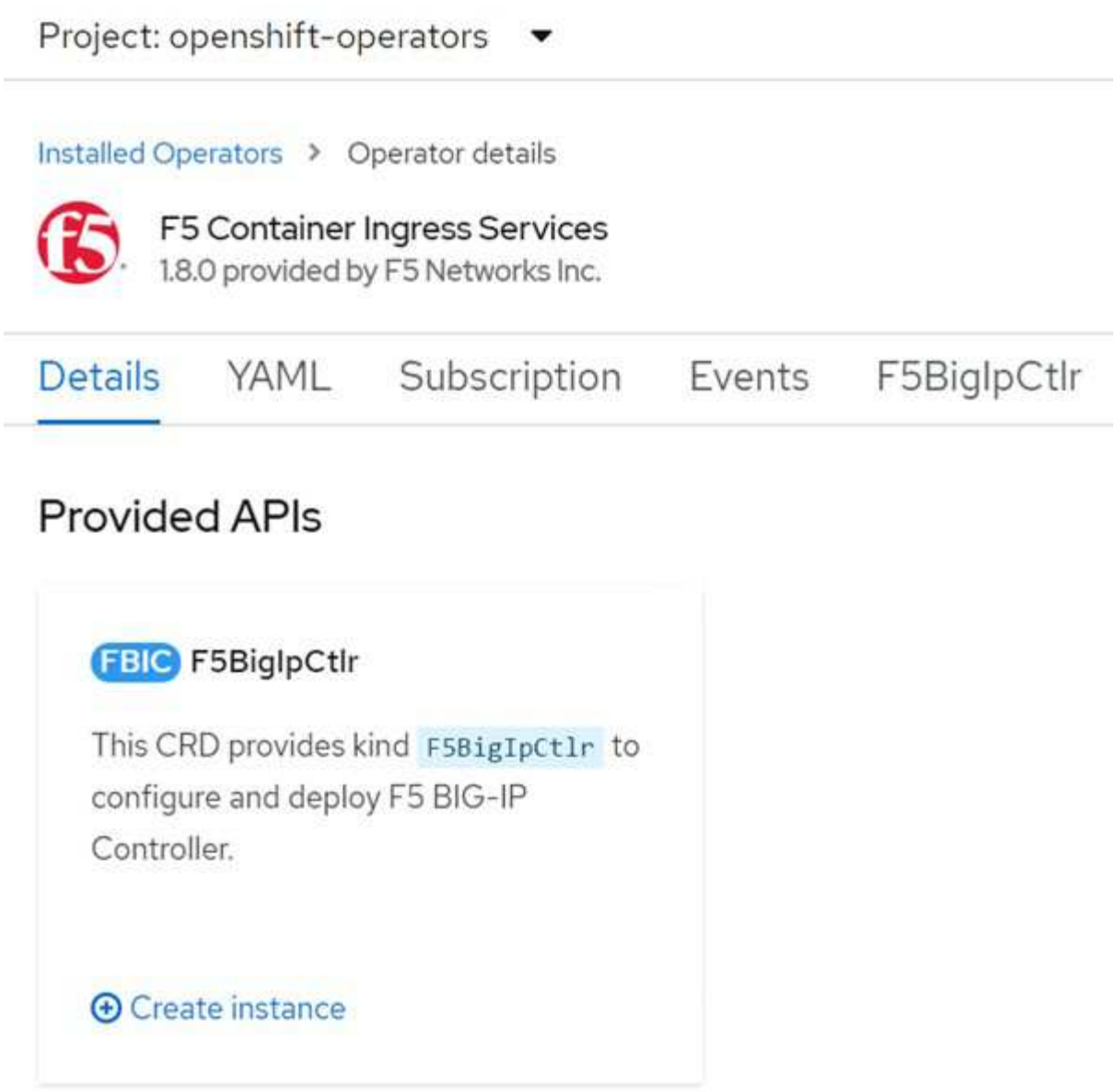
18. Na tela Instalar operador, deixe todos os parâmetros padrão e clique em Instalar.



19. Demora um pouco para instalar o operador.



20. Após a instalação do operador, a mensagem Instalação bem-sucedida será exibida.
21. Navegue até Operadores > Operadores instalados, clique em F5 Container Ingress Service e, em seguida, clique em Criar instância no bloco F5BigIpCtlr.



22. Clique em Visualização YAML e cole o seguinte conteúdo depois de atualizar os parâmetros necessários.

> Atualizar os parâmetros `bigip_partition` , ` openshift_sdn_name`, `bigip_url` e `bigip_login_secret` abaixo para refletir os valores da sua configuração antes de copiar o conteúdo.

```
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. Depois de colar este conteúdo, clique em Criar. Isso instala os pods do CIS no namespace kube-system.

O Red Hat OpenShift, por padrão, fornece uma maneira de expor os serviços por meio de rotas para balanceamento de carga L7. Um roteador OpenShift integrado é responsável por anunciar e manipular o tráfego para essas rotas. No entanto, você também pode configurar o F5 CIS para oferecer suporte às rotas por meio de um sistema F5 BIG-IP externo, que pode ser executado como um roteador auxiliar ou um substituto para o roteador OpenShift auto-hospedado. O CIS cria um servidor virtual no sistema BIG-IP que atua como um roteador para as rotas do OpenShift, e o BIG-IP lida com o roteamento de anúncios e tráfego. Consulte a documentação aqui para obter informações sobre os parâmetros para habilitar esse recurso. Observe que esses parâmetros são definidos para o recurso OpenShift Deployment na API apps/v1. Portanto, ao usá-los com o recurso F5BigIpCtlr cis.f5.com/v1 API, substitua os hifens (-) por sublinhados (_) para os nomes dos parâmetros.

24. Os argumentos que são passados para a criação de recursos do CIS incluem `ipam: true` e `custom_resource_mode: true` . Esses parâmetros são necessários para habilitar a integração do CIS com um controlador IPAM. Verifique se o CIS habilitou a integração do IPAM criando o recurso F5 IPAM.

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

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

25. Crie a conta de serviço, a função e a associação de função necessárias para o controlador F5 IPAM. Crie um arquivo YAML e cole o seguinte conteúdo.

```
[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. Crie os 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. Crie um arquivo YAML e cole a definição de implantação do F5 IPAM fornecida abaixo.

Atualize o parâmetro ip-range em spec.template.spec.containers[0].args abaixo para refletir os ipamLabels e intervalos de endereços IP correspondentes à sua configuração.

ipamLabels[`range1` e `range2` [no exemplo abaixo] precisam ser anotados para os serviços do tipo LoadBalancer para que o controlador IPAM detecte e atribua um endereço IP do intervalo 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. Crie a implantação do controlador F5 IPAM.

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

deployment/f5-ipam-controller created
```

29. Verifique se os pods do controlador F5 IPAM estão em execução.

```
[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. Crie o esquema F5 IPAM.

```
[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
```

Verificação

1. Crie um serviço do 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. Verifique se o controlador IPAM atribui um IP externo a ele.

```
[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. Crie uma implantação e use o serviço LoadBalancer que foi criado.

```
[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. Verifique se os pods estão 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 se o servidor virtual correspondente foi criado no sistema BIG-IP para o serviço do tipo LoadBalancer no OpenShift. Navegue até Tráfego local > Servidores virtuais > Lista de servidores virtuais.

Criação de registros de imagens privadas

Para a maioria das implantações do Red Hat OpenShift, usar um registro público como "Quay.io" ou "DockerHub" atende à maioria das necessidades dos clientes. No entanto, há momentos em que um cliente pode querer hospedar suas próprias imagens privadas ou personalizadas.

Este procedimento documenta a criação de um registro de imagem privado que é apoiado por um volume persistente fornecido pelo Trident e NetApp ONTAP.

> ⓘ O Trident Protect requer um registro para hospedar as imagens exigidas pelos contêineres Astra . A seção a seguir descreve as etapas para configurar um registro privado no cluster Red Hat OpenShift e enviar as imagens necessárias para dar suporte à instalação do Trident Protect.

Criando um registro de imagem privado

1. Remova a anotação padrão da classe de armazenamento padrão atual e anote a classe de armazenamento apoiada pelo Trident como padrão para o cluster 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 o operador imageregistry inserindo os seguintes parâmetros de armazenamento no `spec` seção.

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

storage:
  pvc:
    claim:
```

3. Insira os seguintes parâmetros no `spec` seção para criar uma rota OpenShift com um nome de host personalizado. Salvar e sair.

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

A configuração de rota acima é usada quando você deseja um nome de host personalizado para sua rota. Se você deseja que o OpenShift crie uma rota com um nome de host padrão, você pode adicionar os seguintes parâmetros ao `spec` seção: `defaultRoute: true` .

Certificados TLS personalizados

Quando você usa um nome de host personalizado para a rota, por padrão, ele usa a configuração TLS padrão do operador OpenShift Ingress. No entanto, você pode adicionar uma configuração TLS personalizada à rota. Para fazer isso, siga os seguintes passos.

a. Crie um segredo com os certificados TLS e a chave da rota.

```
[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 o operador imageregistry e adicione os seguintes parâmetros ao `spec` seção.

```
[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 o operador imageregistry novamente e altere o estado de gerenciamento do operador para `Managed` estado. Salvar e sair.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Se todos os pré-requisitos forem atendidos, PVCs, pods e serviços serão criados para o registro de imagem privada. Em poucos minutos, o registro deverá estar ativo.

```
[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. Se estiver usando os certificados TLS padrão para a rota de registro do operador de entrada OpenShift, você poderá buscar os certificados TLS usando o seguinte comando.

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

7. Para permitir que os nós do OpenShift acessem e extraiam as imagens do registro, adicione os certificados ao cliente do Docker nos nós do OpenShift. Crie um configmap no `openshift-config` namespace usando os certificados TLS e aplique o patch na configuração da imagem do cluster para tornar o certificado confiável.

```
[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. O registro interno do OpenShift é controlado por autenticação. Todos os usuários do OpenShift podem acessar o registro do OpenShift, mas as operações que o usuário conectado pode executar dependem das permissões do usuário.
   a. Para permitir que um usuário ou um grupo de usuários extraia imagens do registro, o(s) usuário(s) deve(m) ter a função de visualizador de registro atribuída.

```
[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 um usuário ou grupo de usuários grave ou envie imagens, o(s) usuário(s) deve(m) ter a função de editor de registro atribuída.

```
[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 os nós do OpenShift acessem o registro e enviem ou recebam as imagens, você precisa configurar um segredo de recebimento.

```
[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. Esse segredo de pull pode então ser corrigido para contas de serviço ou ser referenciado na definição de pod correspondente.
    a. Para aplicar o patch às contas de serviço, execute o seguinte comando.

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

b. Para fazer referência ao segredo de pull na definição do pod, adicione o seguinte parâmetro ao `spec` seção.

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

11. Para enviar ou receber uma imagem de estações de trabalho diferentes do nó OpenShift, conclua as seguintes etapas.

a. Adicione os certificados TLS ao cliente do 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. Efetue login no OpenShift usando o comando oc login.

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

c. Efetue login no registro usando as credenciais de usuário do OpenShift com o comando podman/docker.

homem-pod

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

+ NOTA: Se você estiver usando `kubeadmin` usuário faça login no registro privado e use o token em vez da senha.

estivador

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

+ NOTA: Se você estiver usando `kubeadmin` usuário faça login no registro privado e use o token em vez da senha.

d. Empurre ou puxe as imagens.

homem-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
```

estivador

```
[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
```

Validação de soluções e casos de uso

Validação de soluções e casos de uso: Red Hat OpenShift com NetApp

Os exemplos fornecidos nesta página são validações de soluções e casos de uso para o Red Hat OpenShift com NetApp.

- "Implantar um pipeline de CI/CD do Jenkins com armazenamento persistente"
- "Configurar multilocação no Red Hat OpenShift com NetApp"
- "Virtualização Red Hat OpenShift com NetApp ONTAP"
- "Gerenciamento avançado de cluster para Kubernetes no Red Hat OpenShift com NetApp"

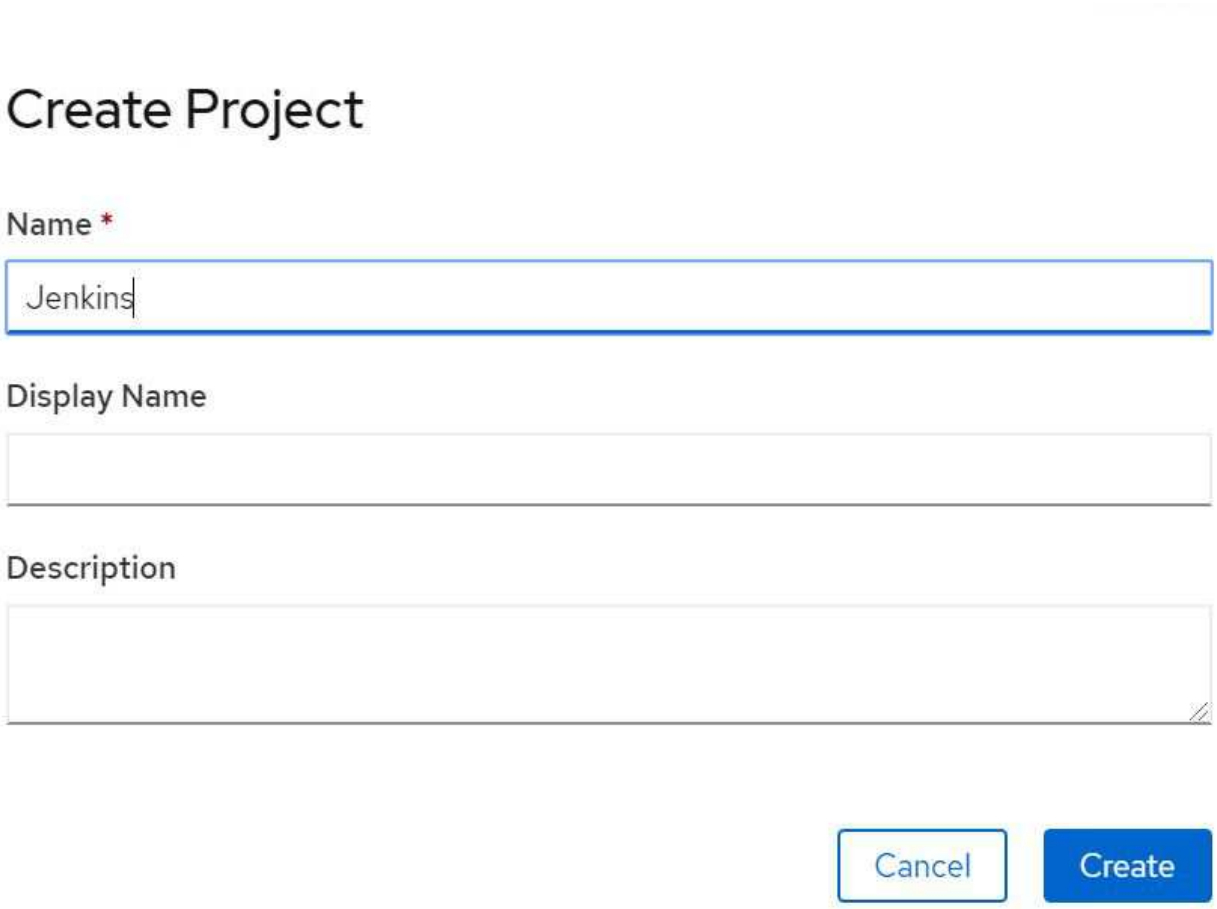
Implantar um pipeline de CI/CD do Jenkins com armazenamento persistente: Red Hat OpenShift com NetApp

Esta seção fornece as etapas para implantar um pipeline de integração contínua/entrega ou implantação contínua (CI/CD) com o Jenkins para validar a operação da solução.

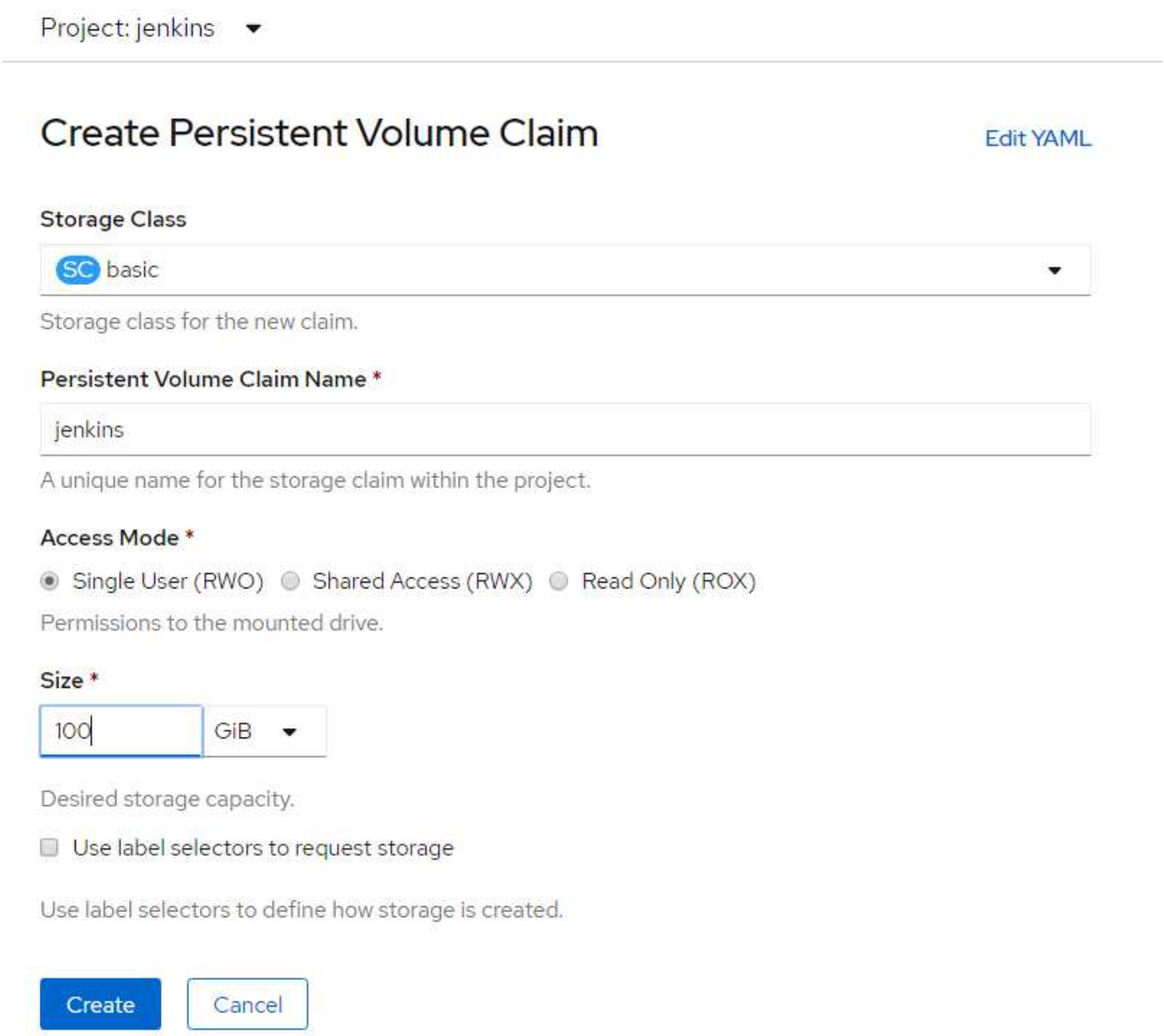
Crie os recursos necessários para a implantação do Jenkins

Para criar os recursos necessários para implantar o aplicativo Jenkins, conclua as seguintes etapas:

1. Crie um novo projeto chamado Jenkins.



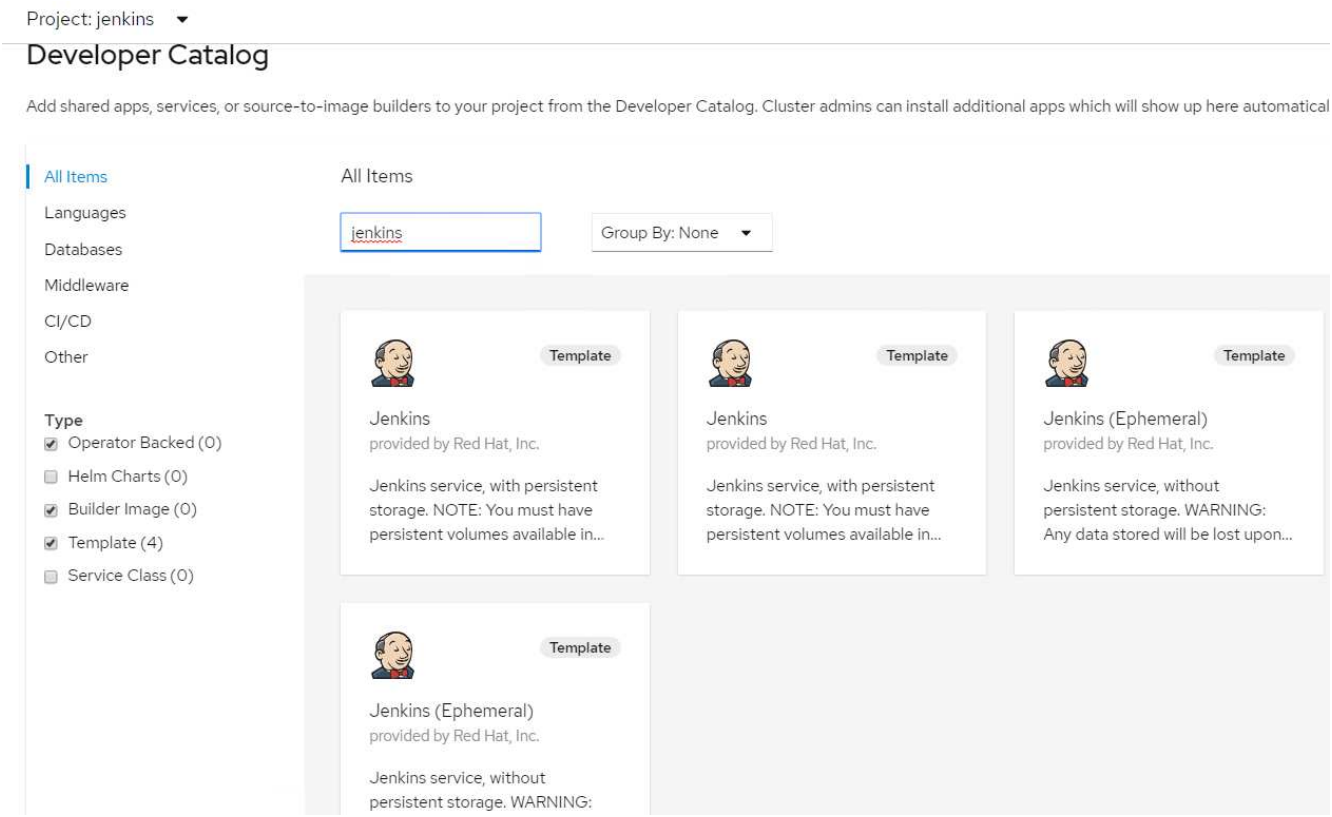
2. Neste exemplo, implantamos o Jenkins com armazenamento persistente. Para dar suporte à construção do Jenkins, crie o PVC. Navegue até Armazenamento > Declarações de volume persistente e clique em Criar reivindicação de volume persistente. Selecione a classe de armazenamento que foi criada, certifique-se de que o Nome da Reivindicação de Volume Persistente seja jenkins, selecione o tamanho e o modo de acesso apropriados e clique em Criar.



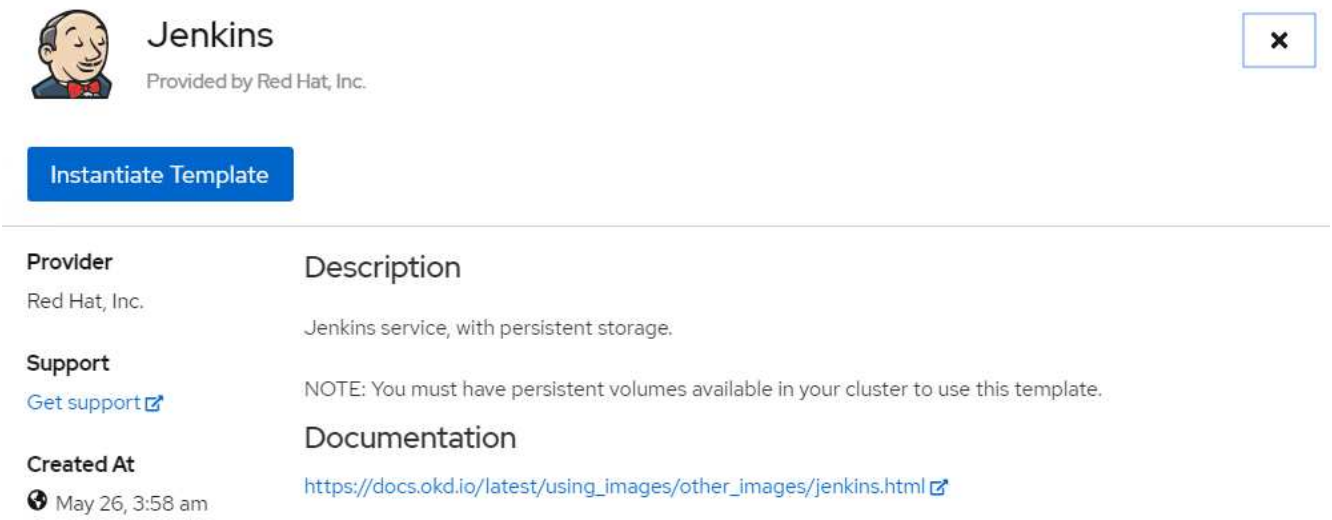
Implantar Jenkins com armazenamento persistente

Para implantar o Jenkins com armazenamento persistente, conclua as seguintes etapas:

1. No canto superior esquerdo, altere a função de Administrador para Desenvolvedor. Clique em +Adicionar e selecione Do catálogo. Na barra Filtrar por palavra-chave, procure por jenkins. Selecione o serviço Jenkins com armazenamento persistente.

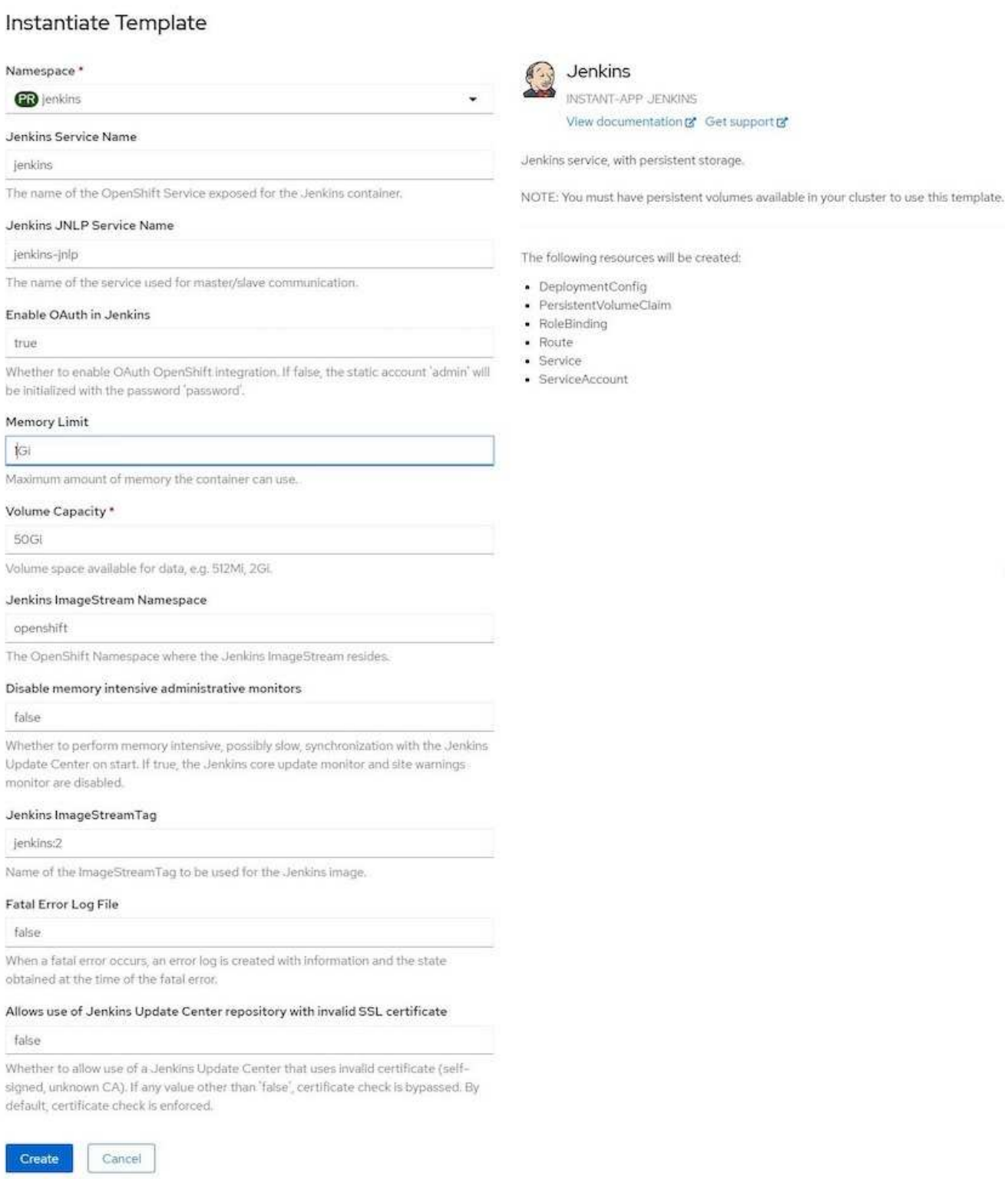


2. Clique `Instantiate Template`.

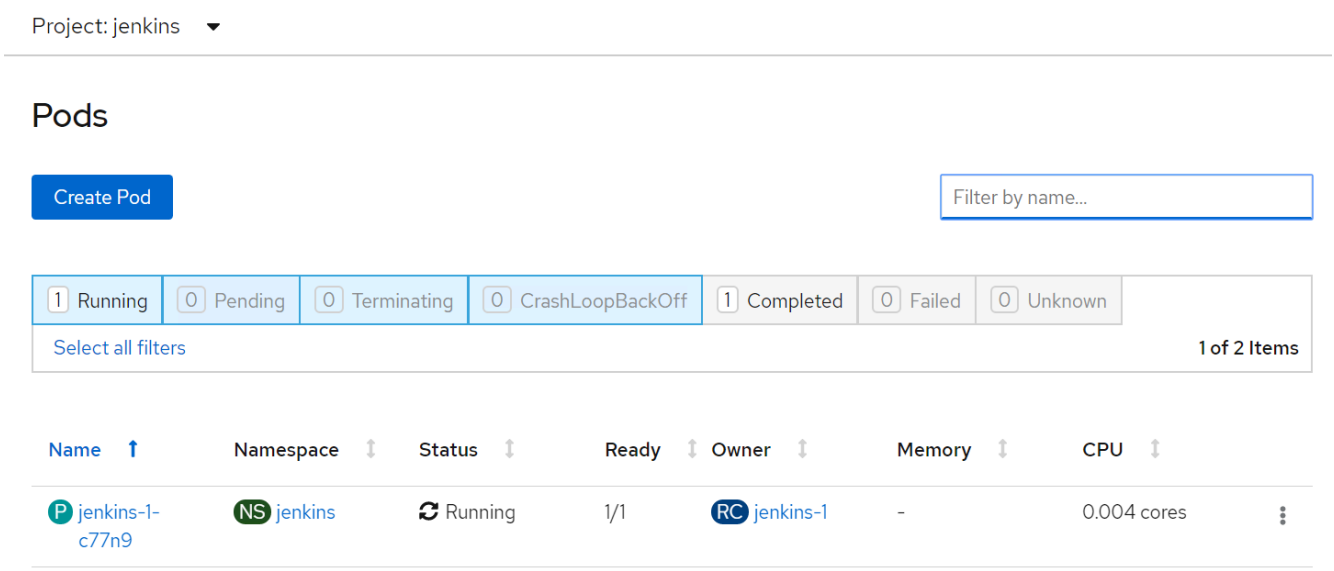


3. Por padrão, os detalhes do aplicativo Jenkins são preenchidos. Com base em suas necessidades, modifique os parâmetros e clique em Criar. Este processo cria todos os recursos necessários para dar

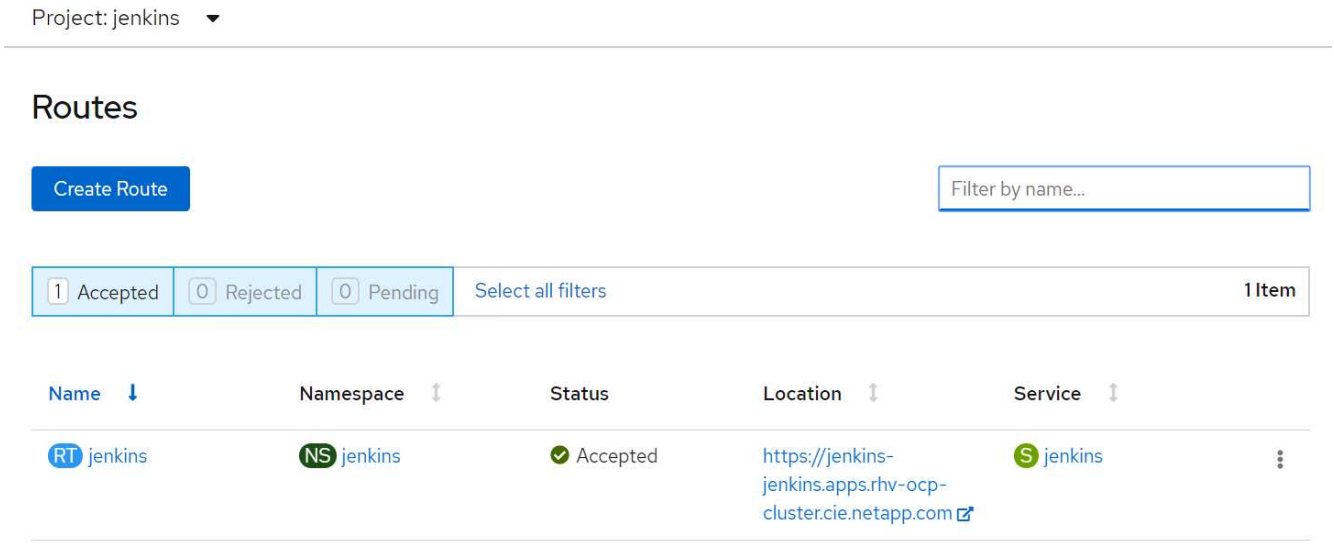
suporte ao Jenkins no OpenShift.



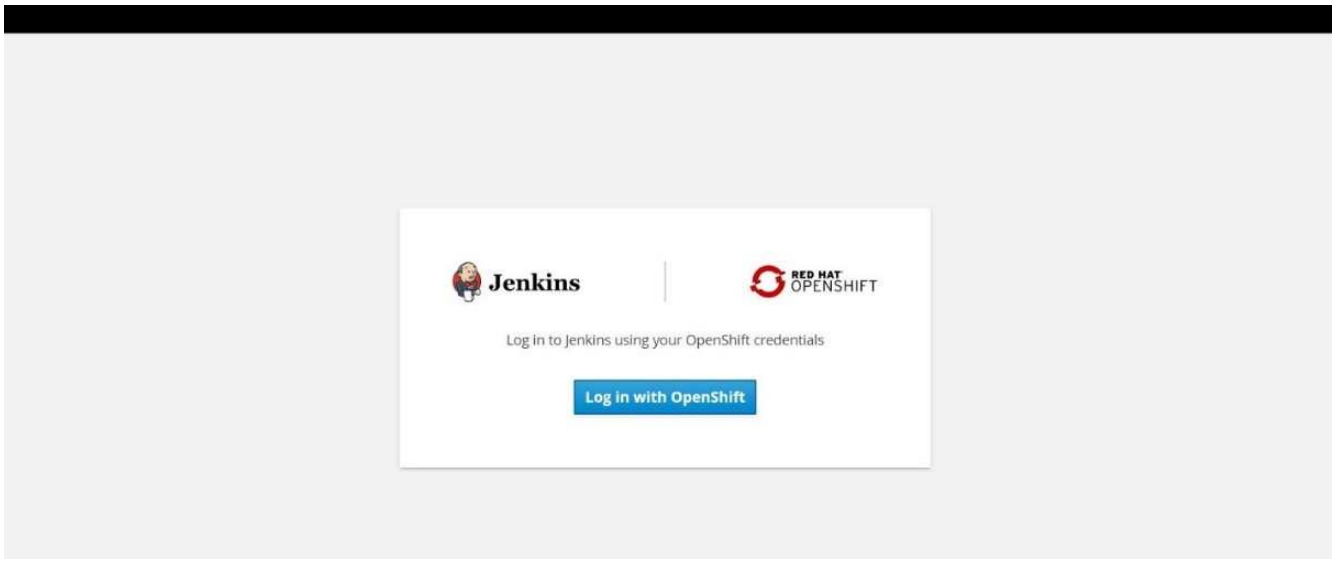
4. Os pods do Jenkins levam aproximadamente de 10 a 12 minutos para entrar no estado Pronto.



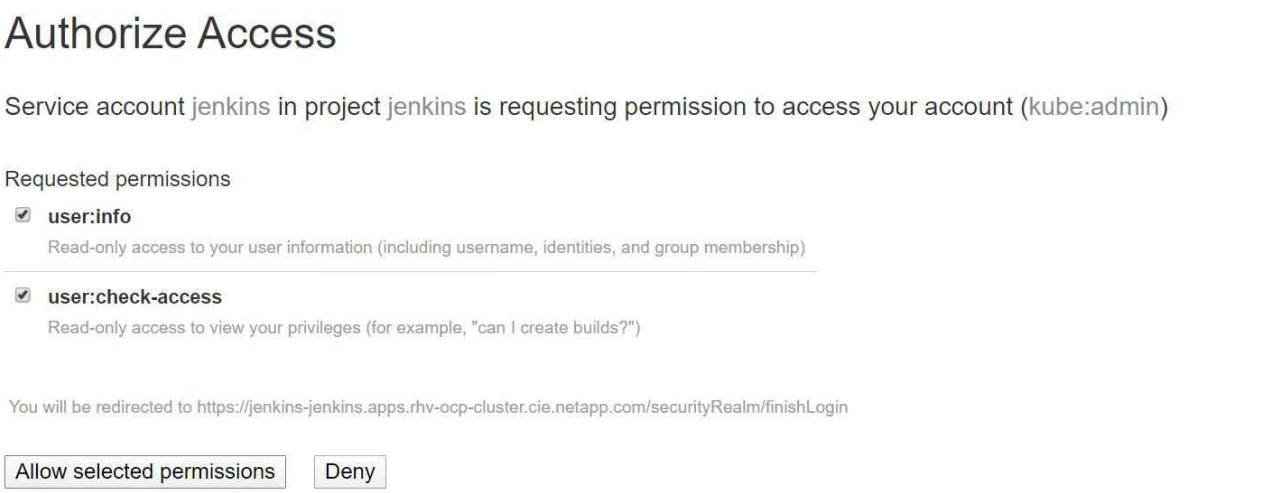
5. Depois que os pods forem instanciados, navegue até Rede > Rotas. Para abrir a página do Jenkins, clique no URL fornecido para a rota do Jenkins.



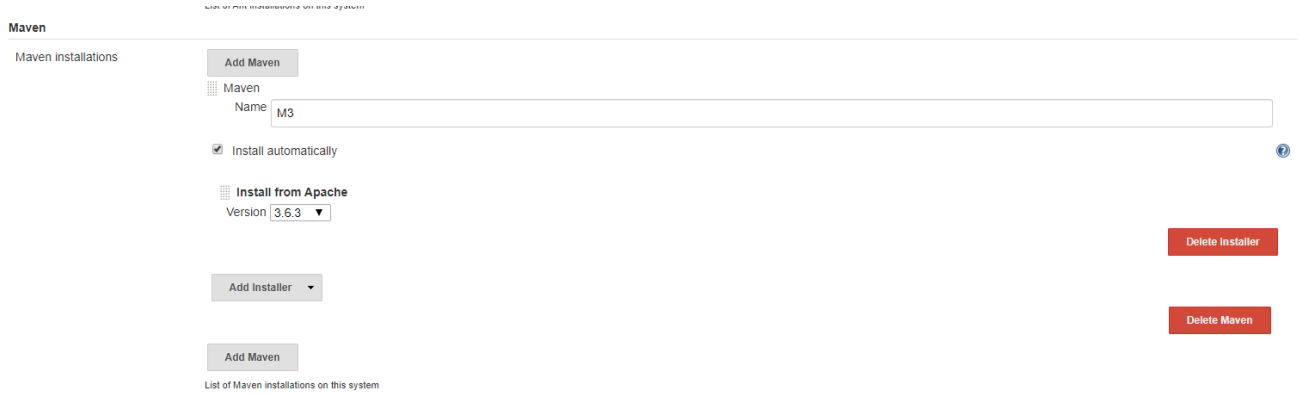
6. Como o OpenShift OAuth foi usado durante a criação do aplicativo Jenkins, clique em Fazer login com OpenShift.



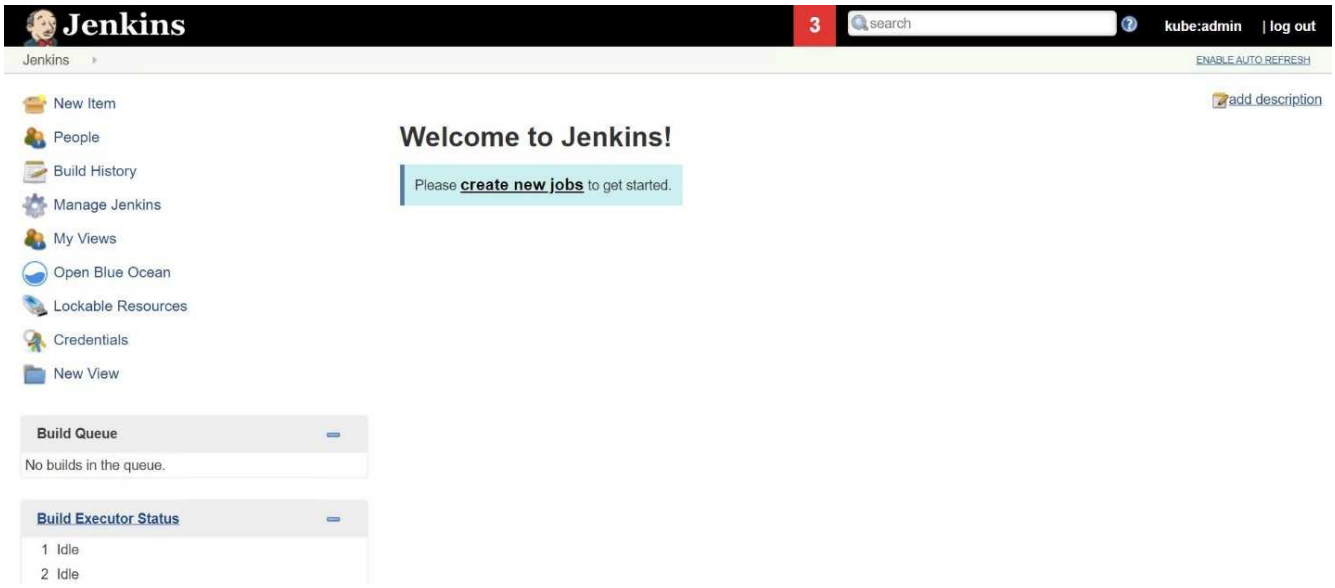
7. Autorize a conta de serviço do Jenkins a acessar os usuários do OpenShift.



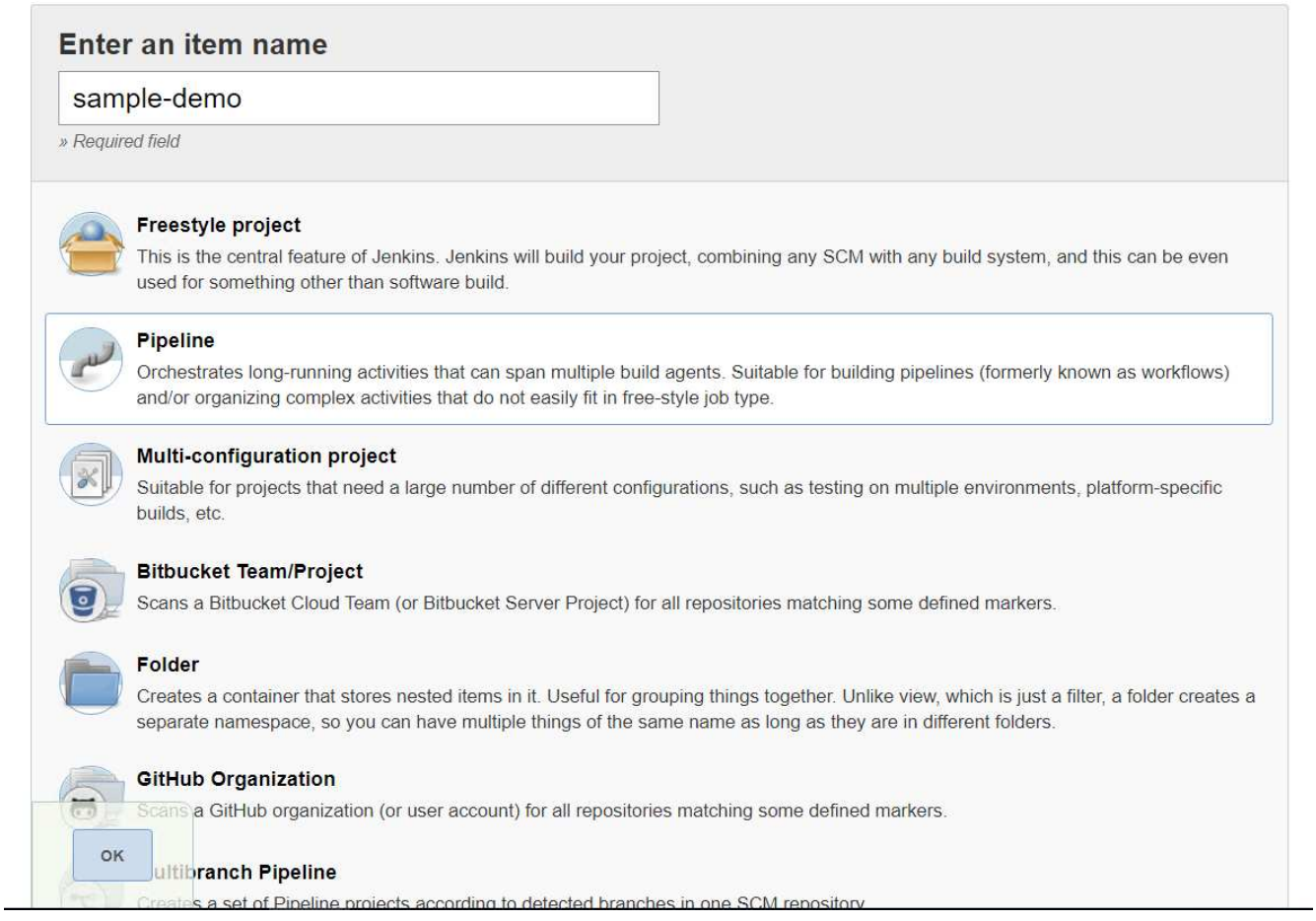
8. A página de boas-vindas do Jenkins é exibida. Como estamos usando uma compilação Maven, conclua a instalação do Maven primeiro. Navegue até Gerenciar Jenkins > Configuração global de ferramentas e, no subtítulo Maven, clique em Adicionar Maven. Digite o nome de sua escolha e certifique-se de que a opção Instalar automaticamente esteja selecionada. Clique em Salvar.



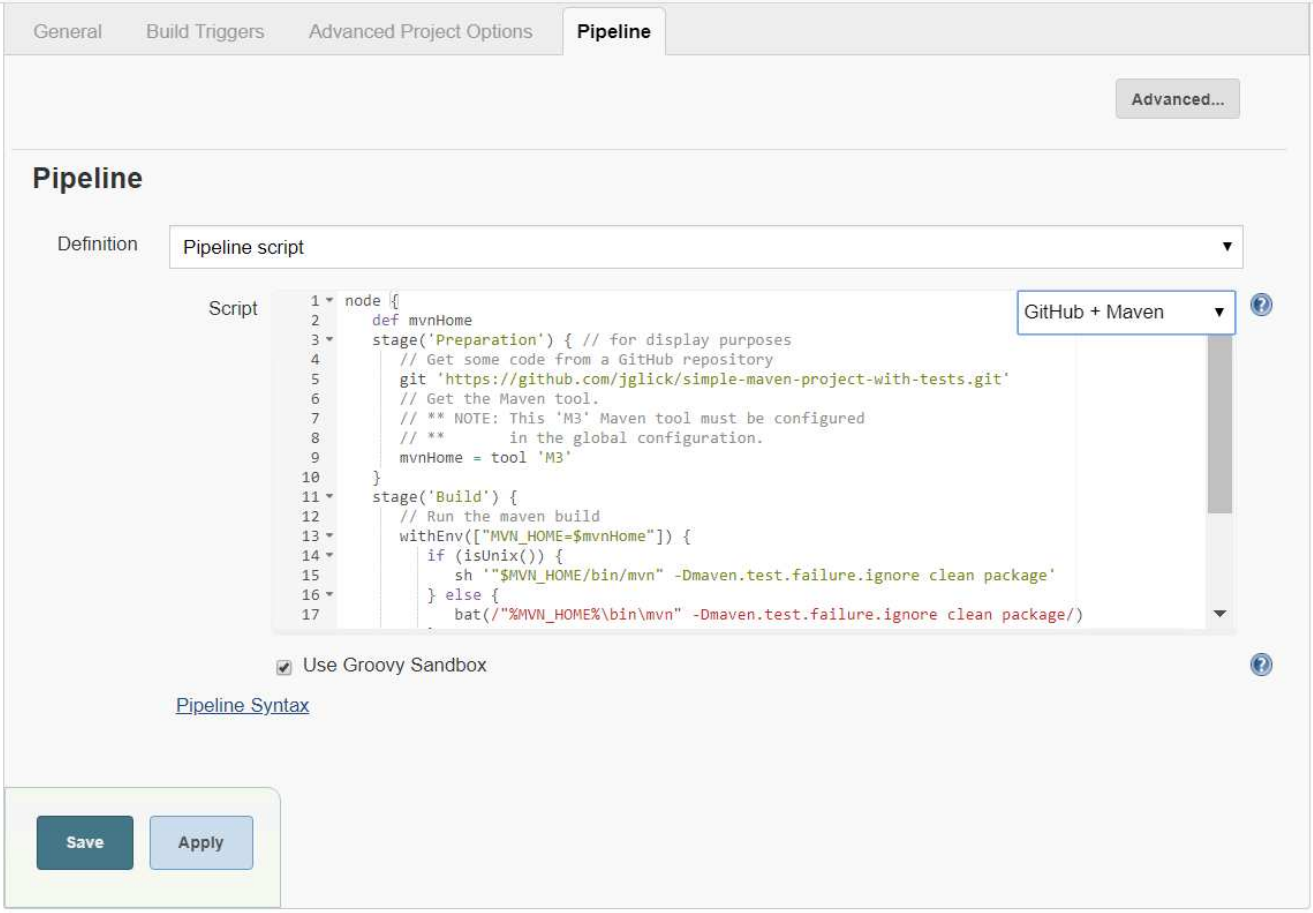
9. Agora você pode criar um pipeline para demonstrar o fluxo de trabalho de CI/CD. Na página inicial, clique em Criar novos trabalhos ou Novo item no menu à esquerda.



10. Na página Criar item, insira o nome de sua escolha, selecione Pipeline e clique em Ok.



11. Selecione a aba Pipeline. No menu suspenso Try Sample Pipeline, selecione Github + Maven. O código é preenchido automaticamente. Clique em Salvar.



12. Clique em Construir agora para iniciar o desenvolvimento durante as fases de preparação, construção e teste. Pode levar vários minutos para concluir todo o processo de compilação e exibir os resultados da compilação.

13. Sempre que houver alguma alteração no código, o pipeline pode ser reconstruído para corrigir a nova versão do software, permitindo integração e entrega contínuas. Clique em Alterações recentes para acompanhar as alterações da versão anterior.

Configurar multilocação

Configurando multilocação no Red Hat OpenShift com NetApp

Muitas organizações que executam vários aplicativos ou cargas de trabalho em contêineres tendem a implantar um cluster Red Hat OpenShift por aplicativo ou carga de trabalho. Isso permite que eles implementem isolamento rigoroso para o aplicativo ou carga de trabalho, otimizem o desempenho e reduzam vulnerabilidades de segurança. No entanto, implantar um cluster Red Hat OpenShift separado para cada aplicativo apresenta seu próprio conjunto de problemas. Isso aumenta a sobrecarga operacional, tendo que monitorar e gerenciar cada cluster individualmente, aumenta os custos devido a recursos dedicados para diferentes aplicativos e prejudica a escalabilidade eficiente.

Para superar esses problemas, pode-se considerar executar todos os aplicativos ou cargas de trabalho em um único cluster do Red Hat OpenShift. Mas, em tal arquitetura, o isolamento de recursos e as vulnerabilidades de segurança dos aplicativos são um dos maiores desafios. Qualquer vulnerabilidade de segurança em uma carga de trabalho pode naturalmente se espalhar para outra carga de trabalho, aumentando assim a zona de impacto. Além disso, qualquer utilização abrupta e descontrolada de recursos por um aplicativo pode afetar o desempenho de outro aplicativo, porque não há uma política de alocação de recursos por padrão.

Portanto, as organizações buscam soluções que reúnam o melhor dos dois mundos, por exemplo, permitindo que elas executem todas as suas cargas de trabalho em um único cluster e ainda oferecendo os benefícios de um cluster dedicado para cada carga de trabalho.

Uma solução eficaz é configurar a multilocação no Red Hat OpenShift. Multilocação é uma arquitetura que permite que vários locatários coexistam no mesmo cluster com isolamento adequado de recursos, segurança e assim por diante. Nesse contexto, um locatário pode ser visto como um subconjunto dos recursos do cluster que são configurados para serem usados por um grupo específico de usuários para uma finalidade exclusiva. Configurar multilocação em um cluster Red Hat OpenShift oferece as seguintes vantagens:

- Uma redução no CapEx e OpEx ao permitir que os recursos do cluster sejam compartilhados
- Menor sobrecarga operacional e de gestão
- Protegendo as cargas de trabalho contra contaminação cruzada de violações de segurança
- Proteção de cargas de trabalho contra degradação inesperada de desempenho devido à contenção de recursos

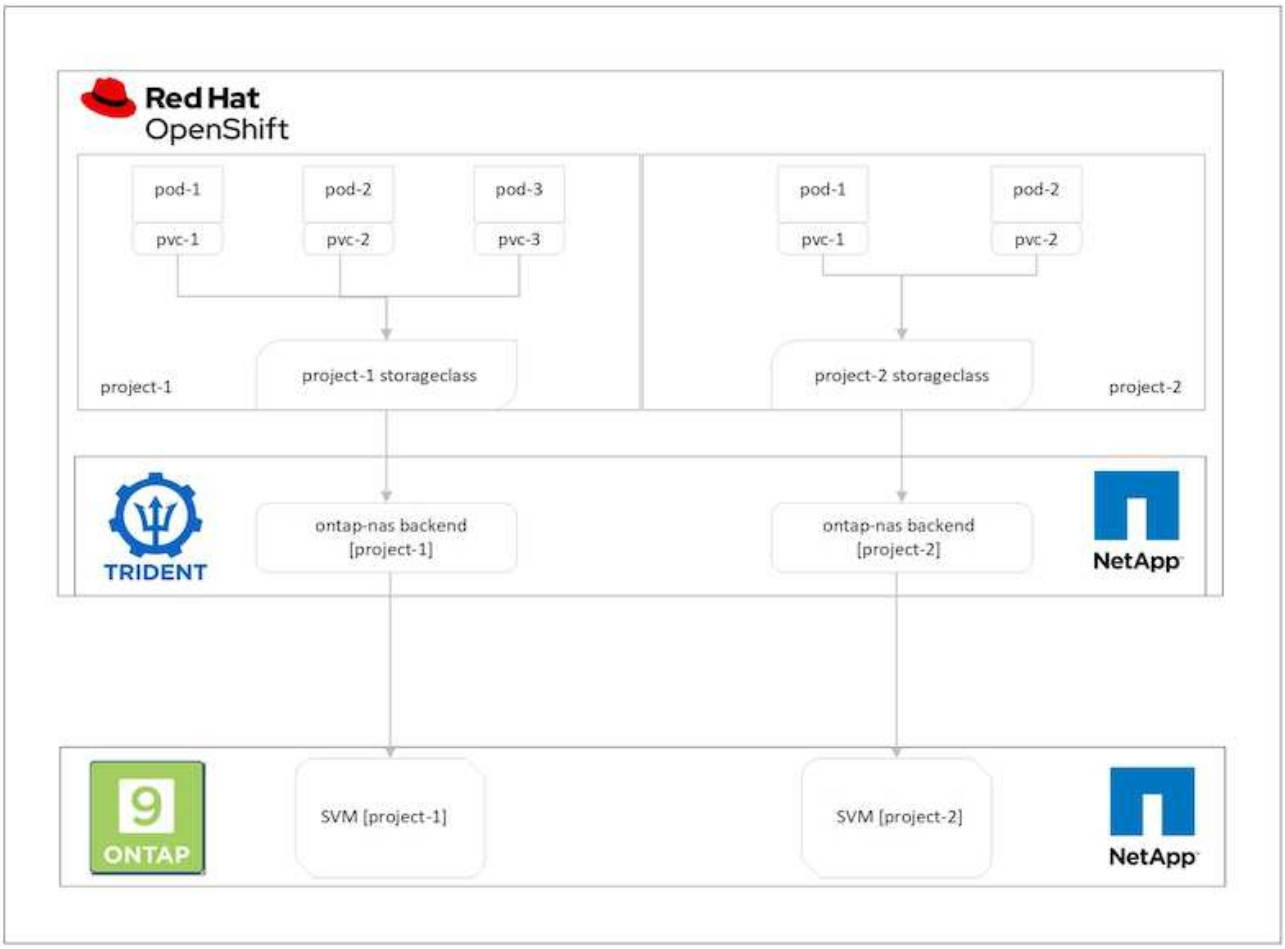
Para um cluster OpenShift multilocatário totalmente realizado, cotas e restrições devem ser configuradas para recursos de cluster pertencentes a diferentes grupos de recursos: computação, armazenamento, rede, segurança e assim por diante. Embora abordemos certos aspectos de todos os buckets de recursos nesta solução, nos concentramos nas melhores práticas para isolar e proteger os dados fornecidos ou consumidos por várias cargas de trabalho no mesmo cluster Red Hat OpenShift, configurando a multilocação em recursos de armazenamento que são alocados dinamicamente pelo Trident com suporte do NetApp ONTAP.

Arquitetura

Embora o Red Hat OpenShift e o Trident apoiados pelo NetApp ONTAP não forneçam isolamento entre cargas de trabalho por padrão, eles oferecem uma ampla gama de recursos que podem ser usados para configurar a multilocação. Para entender melhor o design de uma solução multilocatário em um cluster Red Hat OpenShift com Trident apoiado pelo NetApp ONTAP, vamos considerar um exemplo com um conjunto de requisitos e descrever a configuração em torno dele.

Vamos supor que uma organização execute duas de suas cargas de trabalho em um cluster Red Hat OpenShift como parte de dois projetos nos quais duas equipes diferentes estão trabalhando. Os dados dessas cargas de trabalho residem em PVCs que são provisionados dinamicamente pelo Trident em um backend NetApp ONTAP NAS. A organização tem a necessidade de projetar uma solução multilocatário para essas duas cargas de trabalho e isolar os recursos usados nesses projetos para garantir que a segurança e o desempenho sejam mantidos, com foco principalmente nos dados que atendem a esses aplicativos.

A figura a seguir descreve a solução multilocatário em um cluster Red Hat OpenShift com Trident apoiado pelo NetApp ONTAP.



Requisitos de tecnologia

1. Cluster de armazenamento NetApp ONTAP
2. Cluster Red Hat OpenShift
3. Trident

Red Hat OpenShift – Recursos de cluster

Do ponto de vista do cluster Red Hat OpenShift, o recurso de nível superior para começar é o projeto. Um projeto OpenShift pode ser visto como um recurso de cluster que divide todo o cluster OpenShift em vários clusters virtuais. Portanto, o isolamento no nível do projeto fornece uma base para configurar a multilocação.

O próximo passo é configurar o RBAC no cluster. A melhor prática é ter todos os desenvolvedores trabalhando em um único projeto ou carga de trabalho configurados em um único grupo de usuários no Provedor de Identidade (IdP). O Red Hat OpenShift permite a integração de IdP e a sincronização de grupos de usuários, permitindo que usuários e grupos do IdP sejam importados para o cluster. Isso ajuda os administradores do cluster a segregar o acesso dos recursos do cluster dedicados a um projeto para um ou mais grupos de usuários que trabalham naquele projeto, restringindo assim o acesso não autorizado a quaisquer recursos do cluster. Para saber mais sobre a integração do IdP com o Red Hat OpenShift, consulte a documentação "aqui" .

NetApp ONTAP

É importante isolar o armazenamento compartilhado que serve como um provedor de armazenamento persistente para um cluster Red Hat OpenShift para garantir que os volumes criados no armazenamento para cada projeto apareçam para os hosts como se tivessem sido criados em um armazenamento separado. Para fazer isso, crie tantas SVMs (máquinas virtuais de armazenamento) no NetApp ONTAP quantos projetos ou cargas de trabalho houver e dedique cada SVM a uma carga de trabalho.

Trident

Depois de ter diferentes SVMs para diferentes projetos criados no NetApp ONTAP, você deve mapear cada SVM para um backend Trident diferente. A configuração de backend no Trident direciona a alocação de armazenamento persistente para recursos de cluster OpenShift e requer que os detalhes do SVM sejam mapeados. Este deve ser o driver de protocolo para o backend, no mínimo. Opcionalmente, ele permite que você defina como os volumes são provisionados no armazenamento e defina limites para o tamanho dos volumes ou uso de agregados e assim por diante. Detalhes sobre a definição dos backends do Trident podem ser encontrados "aqui" .

Red Hat OpenShift – recursos de armazenamento

Depois de configurar os backends do Trident , o próximo passo é configurar o StorageClasses. Configure tantas classes de armazenamento quantos forem os backends, fornecendo a cada classe de armazenamento acesso para iniciar volumes somente em um backend. Podemos mapear o StorageClass para um backend Trident específico usando o parâmetro storagePools ao definir a classe de armazenamento. Os detalhes para definir uma classe de armazenamento podem ser encontrados "aqui" . Portanto, há um mapeamento um-para-um do StorageClass para o backend Trident que aponta de volta para uma SVM. Isso garante que todas as reivindicações de armazenamento por meio do StorageClass atribuído a esse projeto sejam atendidas somente pelo SVM dedicado a esse projeto.

Como as classes de armazenamento não são recursos com namespace, como podemos garantir que as declarações de armazenamento para a classe de armazenamento de um projeto por pods em outro namespace ou projeto sejam rejeitadas? A resposta é usar ResourceQuotas. ResourceQuotas são objetos que controlam o uso total de recursos por projeto. Ele pode limitar o número e a quantidade total de recursos que podem ser consumidos por objetos no projeto. Quase todos os recursos de um projeto podem ser limitados usando ResourceQuotas e usar isso de forma eficiente pode ajudar as organizações a reduzir custos e interrupções devido ao excesso de provisionamento ou consumo excessivo de recursos. Consulte a documentação "aqui" para maiores informações.

Para este caso de uso, precisamos limitar os pods em um projeto específico de reivindicar armazenamento de classes de armazenamento que não são dedicadas ao projeto. Para fazer isso, precisamos limitar as reivindicações de volume persistentes para outras classes de armazenamento, definindo `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` para 0. Além disso, um administrador de cluster deve garantir que os desenvolvedores em um projeto não tenham acesso para modificar as ResourceQuotas.

Configuração

Para qualquer solução multilocatário, nenhum usuário pode ter acesso a mais recursos de cluster do que o necessário. Portanto, todo o conjunto de recursos que devem ser configurados como parte da configuração de multilocação é dividido entre o administrador do cluster, o administrador do armazenamento e os desenvolvedores que trabalham em cada projeto.

A tabela a seguir descreve as diferentes tarefas a serem executadas por diferentes usuários:

Papel	Tarefas
Administrador do cluster	Crie projetos para diferentes aplicações ou cargas de trabalho
	Crie ClusterRoles e RoleBindings para storage-admin
	Crie funções e RoleBindings para desenvolvedores que atribuem acesso a projetos específicos
	[Opcional] Configurar projetos para agendar pods em nós específicos
Administrador de armazenamento	Criar SVMs no NetApp ONTAP
	Criar backends Trident
	Criar StorageClasses
	Criar ResourceQuotas de armazenamento
Desenvolvedores	Validar acesso para criar ou corrigir PVCs ou pods no projeto atribuído
	Validar acesso para criar ou corrigir PVCs ou pods em outro projeto
	Validar acesso para visualizar ou editar Projetos, ResourceQuotas e StorageClasses

Configuração

A seguir estão os pré-requisitos para configurar a multilocação no Red Hat OpenShift com o NetApp.

Pré-requisitos

- Cluster NetApp ONTAP
- Cluster Red Hat OpenShift
- Trident instalado no cluster
- Estação de trabalho de administração com ferramentas tridentctl e oc instaladas e adicionadas ao $PATH
- Acesso de administrador ao ONTAP
- Acesso de administrador de cluster ao cluster OpenShift
- O cluster é integrado ao provedor de identidade
- O provedor de identidade é configurado para distinguir com eficiência entre usuários em equipes diferentes

Configuração: tarefas de administração do cluster

As seguintes tarefas são executadas pelo administrador de cluster do Red Hat OpenShift:

1. Efetue login no cluster Red Hat OpenShift como administrador do cluster.
2. Crie dois projetos correspondentes a projetos diferentes.

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

3. Crie a função de desenvolvedor para o projeto-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
```

A definição de função fornecida nesta seção é apenas um exemplo. As funções do desenvolvedor devem ser definidas com base nos requisitos do usuário final.

1. Da mesma forma, crie funções de desenvolvedor para o projeto-2.
2. Todos os recursos de armazenamento do OpenShift e NetApp geralmente são gerenciados por um administrador de armazenamento. O acesso para administradores de armazenamento é controlado pela função de operador do Trident, criada quando o Trident é instalado. Além disso, o administrador de armazenamento também precisa de acesso ao ResourceQuotas para controlar como o armazenamento é consumido.
3. Crie uma função para gerenciar ResourceQuotas em todos os projetos no cluster para anexá-la ao administrador de armazenamento.

```
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. Certifique-se de que o cluster esteja integrado ao provedor de identidade da organização e que os grupos de usuários estejam sincronizados com os grupos do cluster. O exemplo a seguir mostra que o provedor de identidade foi integrado ao cluster e sincronizado com os grupos de usuários.

```
$ 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 armazenamento.

```
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 administradores de armazenamento, duas funções devem ser vinculadas: trident-operator e resource-quotas.

1. Crie RoleBindings para desenvolvedores vincularem a função developer-project-1 ao grupo correspondente (ocp-project-1) no project-1.

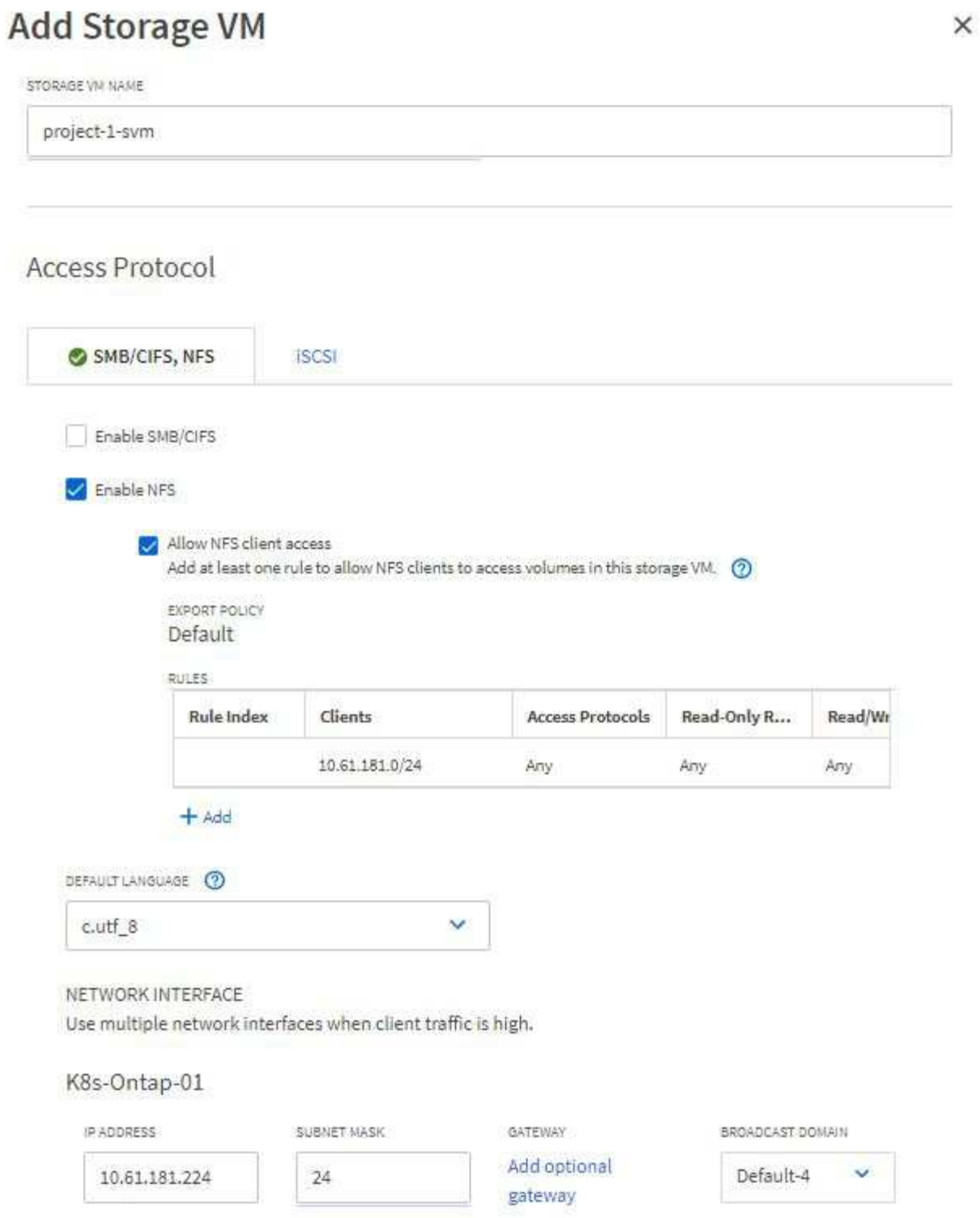
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role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

2. Da mesma forma, crie RoleBindings para desenvolvedores vinculando as funções de desenvolvedor ao grupo de usuários correspondente no projeto-2.

Configuração: tarefas de administração de armazenamento

Os seguintes recursos devem ser configurados por um administrador de armazenamento:

1. Efetue login no cluster NetApp ONTAP como administrador.
2. Navegue até Armazenamento > VMs de armazenamento e clique em Adicionar. Crie dois SVMs, um para o projeto 1 e outro para o projeto 2, fornecendo os detalhes necessários. Crie também uma conta vsadmin para gerenciar o SVM e seus recursos.



1. Efetue login no cluster do Red Hat OpenShift como administrador de armazenamento.
2. Crie o backend para o projeto-1 e mapeie-o para o SVM dedicado ao projeto. A NetApp recomenda usar a conta vsadmin do SVM para conectar o backend ao SVM em vez de usar o administrador do cluster 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 usando o driver ontap-nas para este exemplo. Use o driver apropriado ao criar o backend com base no caso de uso.

Assumimos que o Trident está instalado no projeto Trident.

1. Da mesma forma, crie o backend Trident para o projeto-2 e mapeie-o para o SVM dedicado ao projeto-2.
2. Em seguida, crie as classes de armazenamento. Crie a classe de armazenamento para o projeto-1 e configure-a para usar os pools de armazenamento do backend dedicados ao projeto-1 definindo o parâmetro 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. Da mesma forma, crie uma classe de armazenamento para o projeto-2 e configure-a para usar os pools de armazenamento do backend dedicados ao projeto-2.
4. Crie um ResourceQuota para restringir recursos no projeto-1 solicitando armazenamento de storageclasses dedicadas a outros projetos.

```
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. Da mesma forma, crie um ResourceQuota para restringir recursos no projeto-2 solicitando armazenamento de storageclasses dedicadas a outros projetos.

Validação

Para validar a arquitetura multilocatário configurada nas etapas anteriores, conclua as seguintes etapas:

Validar acesso para criar PVCs ou pods no projeto atribuído

1. Efetue login como ocp-project-1-user, desenvolvedor no projeto-1.
2. Verifique o acesso para criar um novo projeto.

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

3. Crie um PVC no projeto-1 usando a classe de armazenamento atribuída ao projeto-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 o PV associado ao PVC.

```
oc get pv
```

5. Valide se o PV e seu volume foram criados em um SVM dedicado ao projeto-1 no NetApp ONTAP.

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

6. Crie um pod no projeto-1 e monte o PVC criado na etapa 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 se o pod está em execução e se o volume foi montado.

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

Validar acesso para criar PVCs ou pods em outro projeto ou usar recursos dedicados a outro projeto

1. Efetue login como ocp-project-1-user, desenvolvedor no projeto-1.
2. Crie um PVC no projeto-1 usando a classe de armazenamento atribuída ao projeto-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. Crie um PVC no projeto-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. Certifique-se de que os PVCs `test-pvc-project-1-sc-2` e `test-pvc-project-2-sc-1` não foram criados.

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

5. Crie um pod no projeto-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 acesso para visualizar e editar Projetos, ResourceQuotas e StorageClasses

1. Efetue login como ocp-project-1-user, desenvolvedor no projeto-1.
2. Verifique o acesso para criar novos projetos.

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

3. Valide o acesso para visualizar projetos.

```
oc get ns
```

4. Verifique se o usuário pode visualizar ou editar ResourceQuotas no projeto-1.

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

5. Valide se o usuário tem acesso para visualizar as classes de armazenamento.

```
oc get sc
```

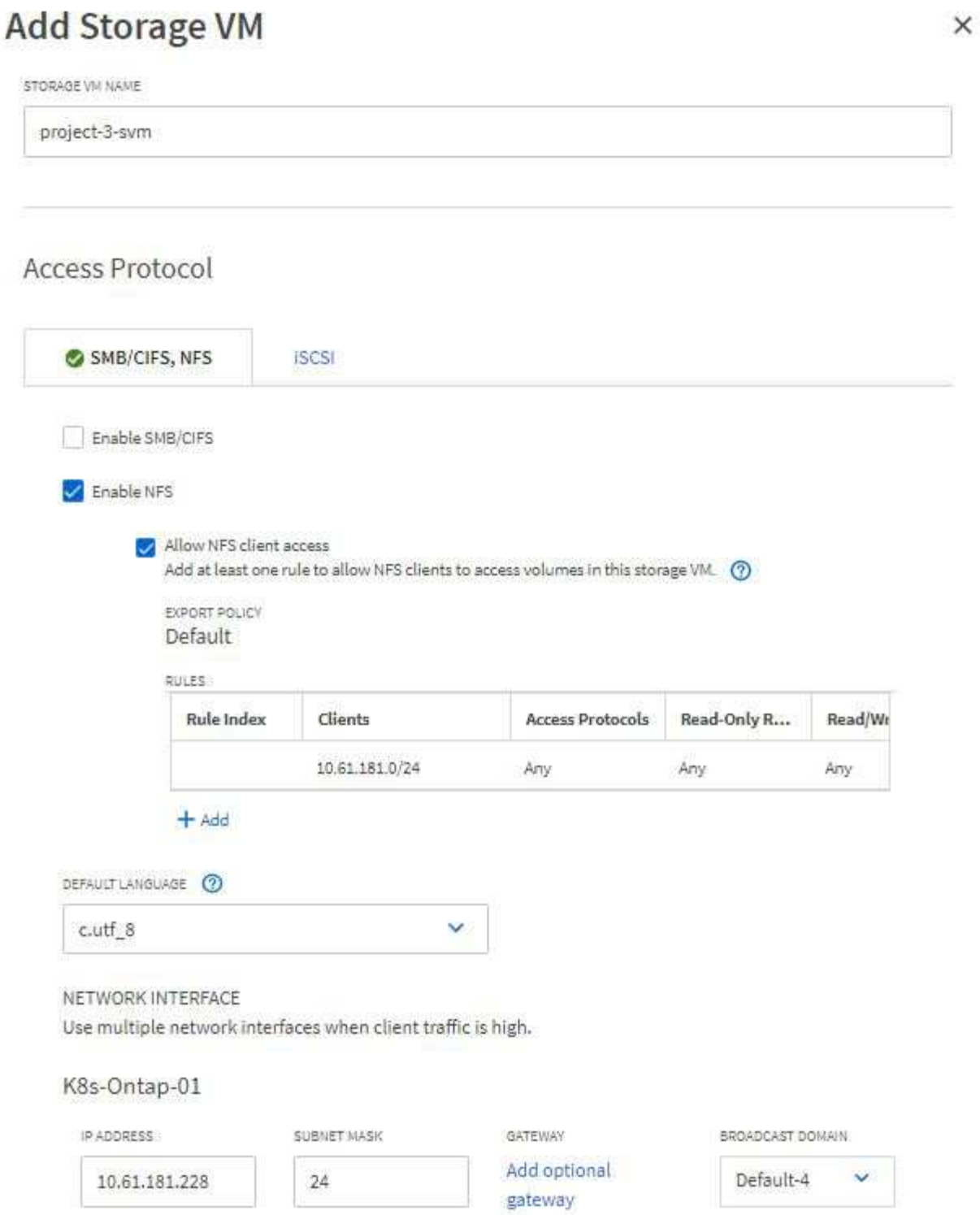
6. Verifique o acesso para descrever as classes de armazenamento.
7. Valide o acesso do usuário para editar as classes de armazenamento.

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

Escalonamento: Adicionando mais projetos

Em uma configuração multilocatário, adicionar novos projetos com recursos de armazenamento requer configuração adicional para garantir que a multilocação não seja violada. Para adicionar mais projetos em um cluster multilocatário, conclua as seguintes etapas:

1. Efetue login no cluster NetApp ONTAP como administrador de armazenamento.
2. Navegar para `Storage → Storage VMs` e clique `Add` . Crie um novo SVM dedicado ao projeto-3. Crie também uma conta vsadmin para gerenciar o SVM e seus recursos.



1. Efetue login no cluster do Red Hat OpenShift como administrador do cluster.
2. Crie um novo projeto.

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

3. Certifique-se de que o grupo de usuários do projeto-3 seja criado no IdP e sincronizado com o cluster OpenShift.

```
oc get groups
```

4. Crie a função de desenvolvedor para o projeto-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
```

A definição de função fornecida nesta seção é apenas um exemplo. A função do desenvolvedor deve ser definida com base nos requisitos do usuário final.

1. Crie RoleBinding para desenvolvedores no projeto-3 vinculando a função developer-project-3 ao grupo correspondente (ocp-project-3) no projeto-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. Efetue login no cluster Red Hat OpenShift como administrador de armazenamento
3. Crie um backend Trident e mapeie-o para o SVM dedicado ao projeto-3. A NetApp recomenda usar a conta vsadmin do SVM para conectar o backend ao SVM em vez de usar o administrador do cluster 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 usando o driver ontap-nas para este exemplo. Use o driver apropriado para criar o backend com base no caso de uso.

Assumimos que o Trident está instalado no projeto Trident.

1. Crie a classe de armazenamento para o projeto-3 e configure-a para usar os pools de armazenamento do backend dedicados ao projeto-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. Crie um ResourceQuota para restringir recursos no projeto-3 solicitando armazenamento de classes de armazenamento dedicadas a outros projetos.

```
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. Aplique patch no ResourceQuotas em outros projetos para restringir o acesso de recursos nesses projetos ao armazenamento da classe de armazenamento dedicada ao projeto-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}}}'
```

Gerenciamento avançado de cluster para Kubernetes

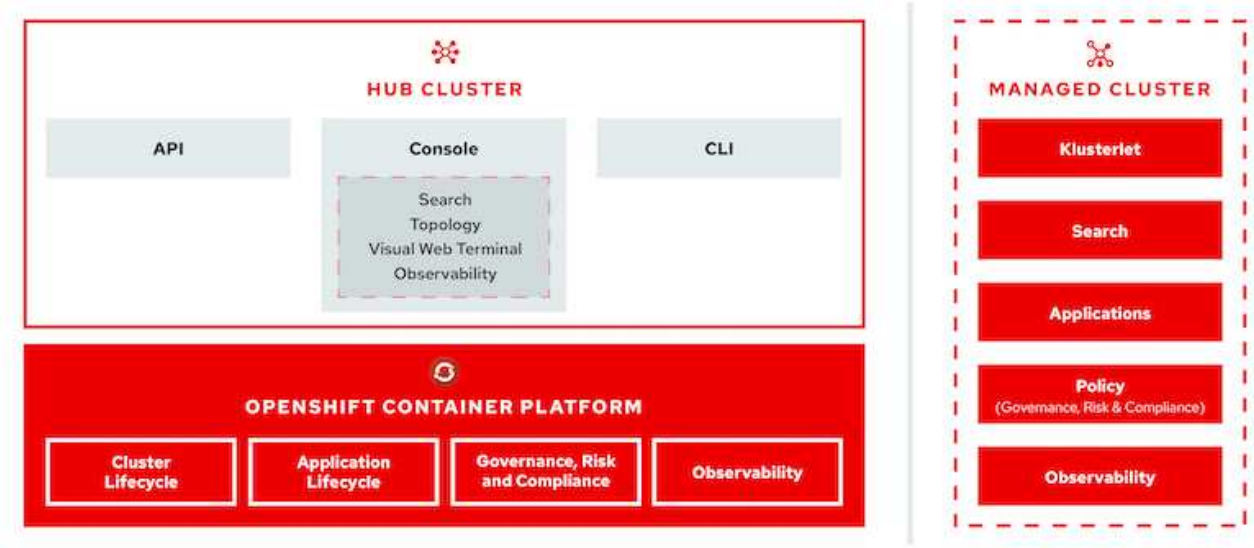
Gerenciamento avançado de cluster para Kubernetes: Red Hat OpenShift com NetApp - Visão geral

À medida que um aplicativo em contêiner passa do desenvolvimento para a produção, muitas organizações exigem vários clusters do Red Hat OpenShift para dar suporte aos testes e à implantação desse aplicativo. Junto com isso, as organizações geralmente hospedam vários aplicativos ou cargas de trabalho em clusters OpenShift. Portanto, cada organização acaba gerenciando um conjunto de clusters, e os administradores do OpenShift precisam enfrentar o desafio adicional de gerenciar e manter vários clusters em uma variedade de ambientes que abrangem vários data centers locais e nuvens públicas. Para enfrentar esses desafios, a Red Hat introduziu o Advanced Cluster Management para Kubernetes.

O Red Hat Advanced Cluster Management for Kubernetes permite que você execute as seguintes tarefas:

1. Crie, importe e gerencie vários clusters em data centers e nuvens públicas
2. Implante e gerencie aplicativos ou cargas de trabalho em vários clusters a partir de um único console
3. Monitorar e analisar a saúde e o status de diferentes recursos do cluster
4. Monitore e aplique a conformidade de segurança em vários clusters

O Red Hat Advanced Cluster Management for Kubernetes é instalado como um complemento para um cluster Red Hat OpenShift e usa esse cluster como um controlador central para todas as suas operações. Este cluster é conhecido como cluster de hub e expõe um plano de gerenciamento para os usuários se conectarem ao Advanced Cluster Management. Todos os outros clusters OpenShift que são importados ou criados por meio do console do Advanced Cluster Management são gerenciados pelo cluster hub e são chamados de clusters gerenciados. Ele instala um agente chamado Klusterlet nos clusters gerenciados para conectá-los ao cluster hub e atender às solicitações de diferentes atividades relacionadas ao gerenciamento do ciclo de vida do cluster, gerenciamento do ciclo de vida do aplicativo, observabilidade e conformidade de segurança.



Para mais informações, consulte a documentação "aqui" .

Implantar o ACM para Kubernetes

Implantar o Advanced Cluster Management para Kubernetes

Esta seção aborda o gerenciamento avançado de cluster para Kubernetes no Red Hat OpenShift com NetApp.

Pré-requisitos

1. Um cluster Red Hat OpenShift (maior que a versão 4.5) para o cluster do hub
2. Clusters Red Hat OpenShift (superiores à versão 4.4.3) para clusters gerenciados
3. Acesso de administrador de cluster ao cluster Red Hat OpenShift
4. Uma assinatura do Red Hat para Advanced Cluster Management para Kubernetes

O Advanced Cluster Management é um complemento para o cluster OpenShift, portanto, há certos requisitos e restrições nos recursos de hardware com base nos recursos usados no hub e nos clusters gerenciados. Você precisa levar essas questões em consideração ao dimensionar os clusters. Veja a documentação "aqui" para mais detalhes.

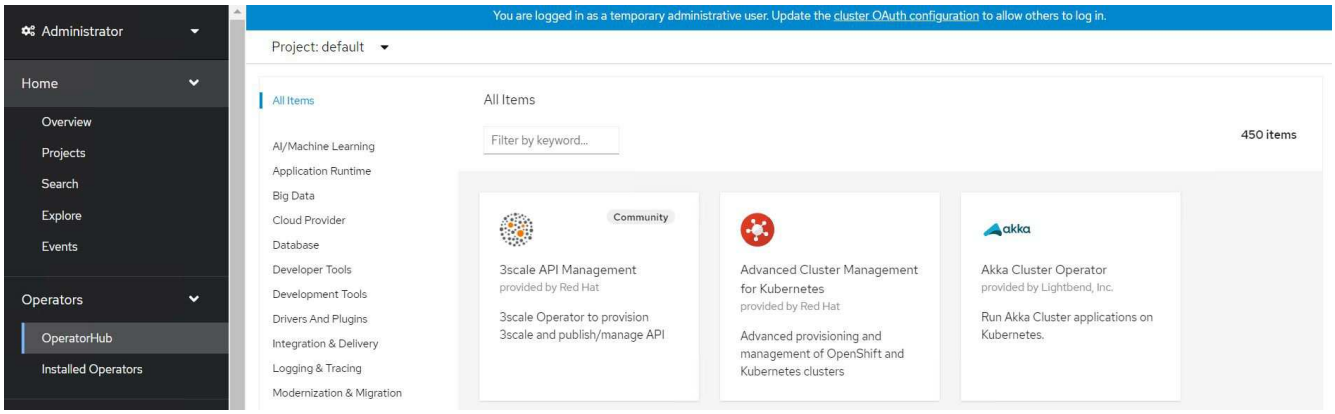
Opcionalmente, se o cluster do hub tiver nós dedicados para hospedar componentes de infraestrutura e você quiser instalar recursos do Advanced Cluster Management somente nesses nós, será necessário adicionar

tolerâncias e seletores a esses nós adequadamente. Para mais detalhes, consulte a documentação "aqui" .

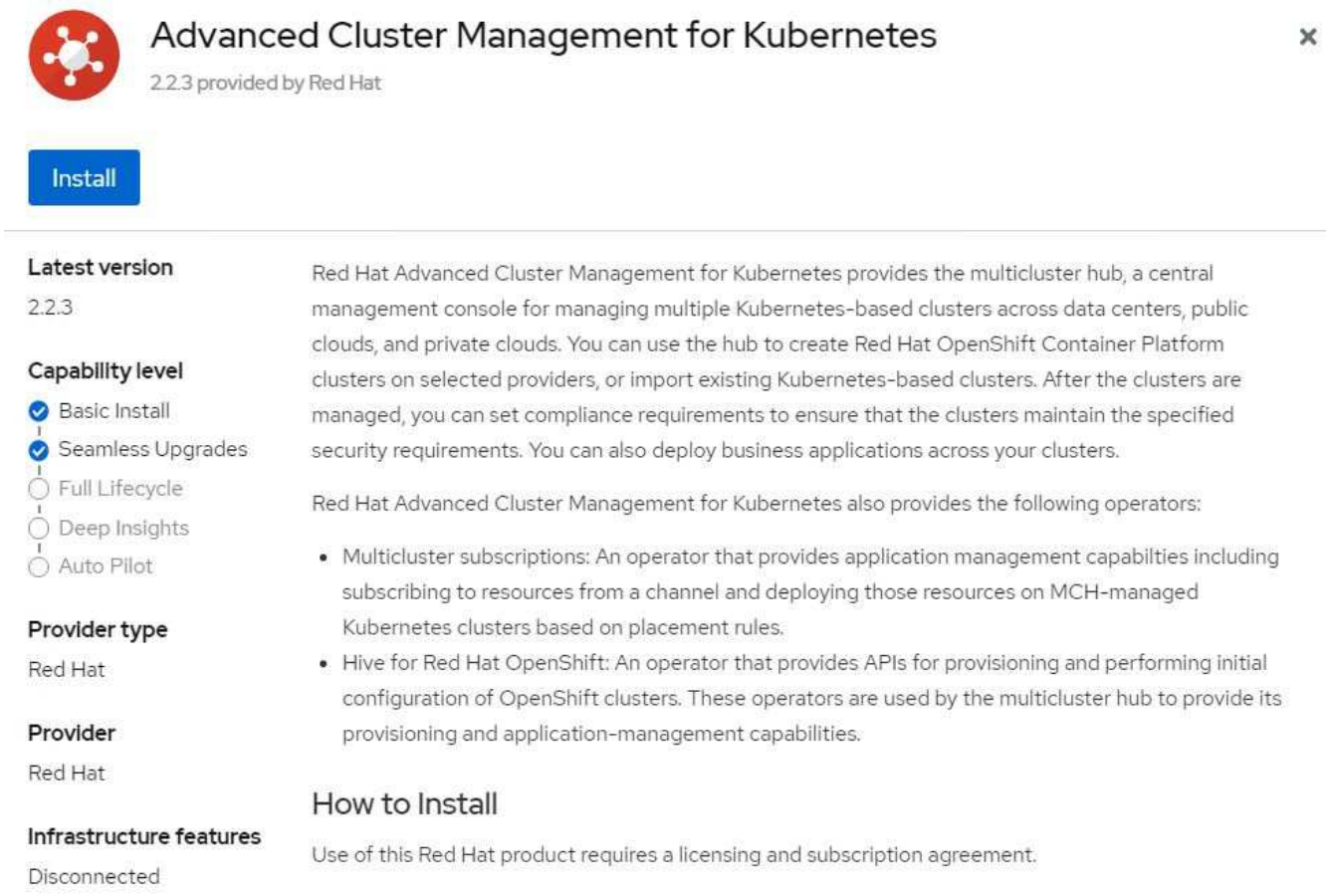
Implantar o Advanced Cluster Management para Kubernetes

Para instalar o Advanced Cluster Management for Kubernetes em um cluster OpenShift, conclua as seguintes etapas:

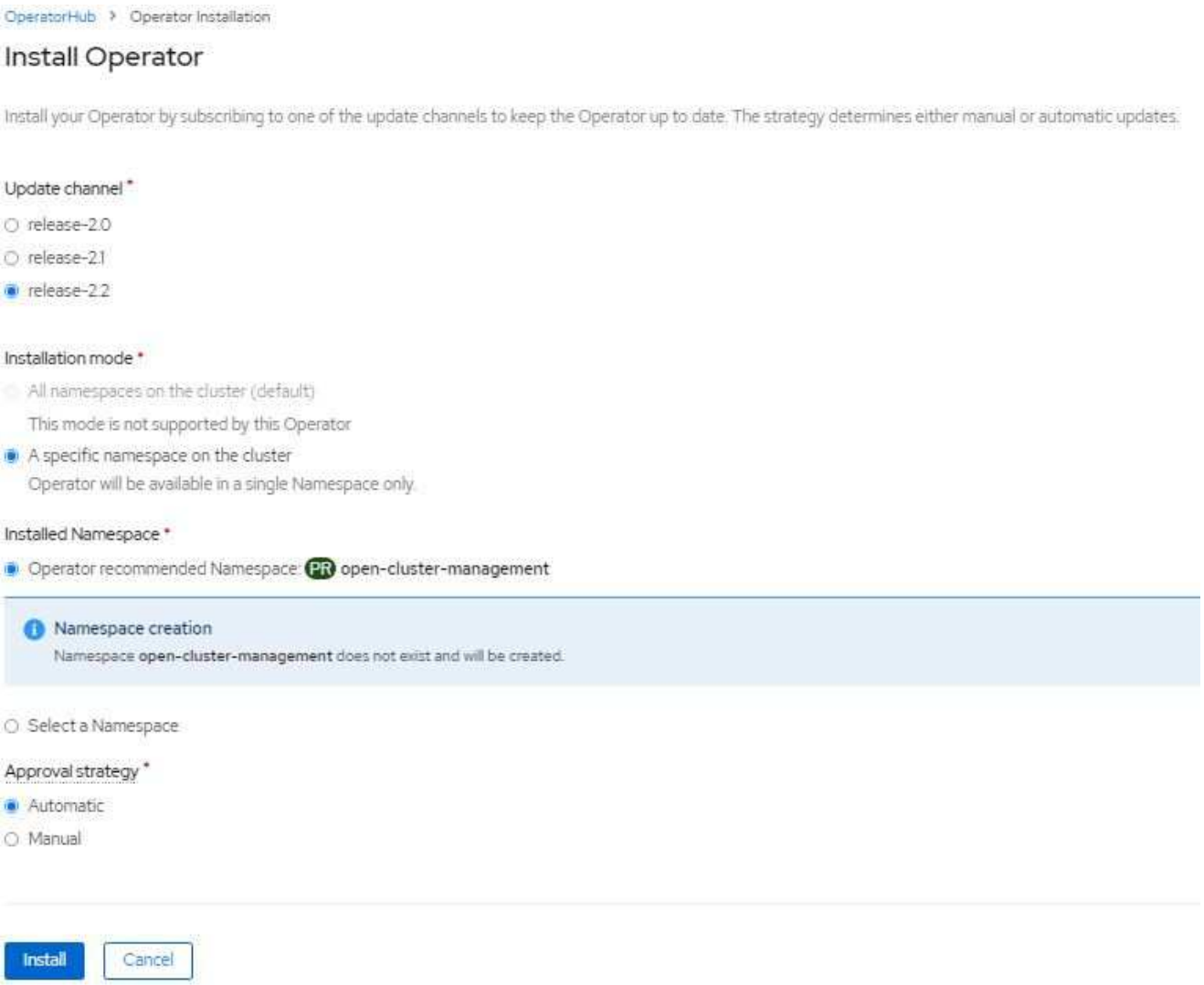
1. Escolha um cluster OpenShift como o cluster central e faça login nele com privilégios de administrador de cluster.
2. Navegue até Operadores > Hub de Operadores e pesquise por Gerenciamento Avançado de Cluster para Kubernetes.



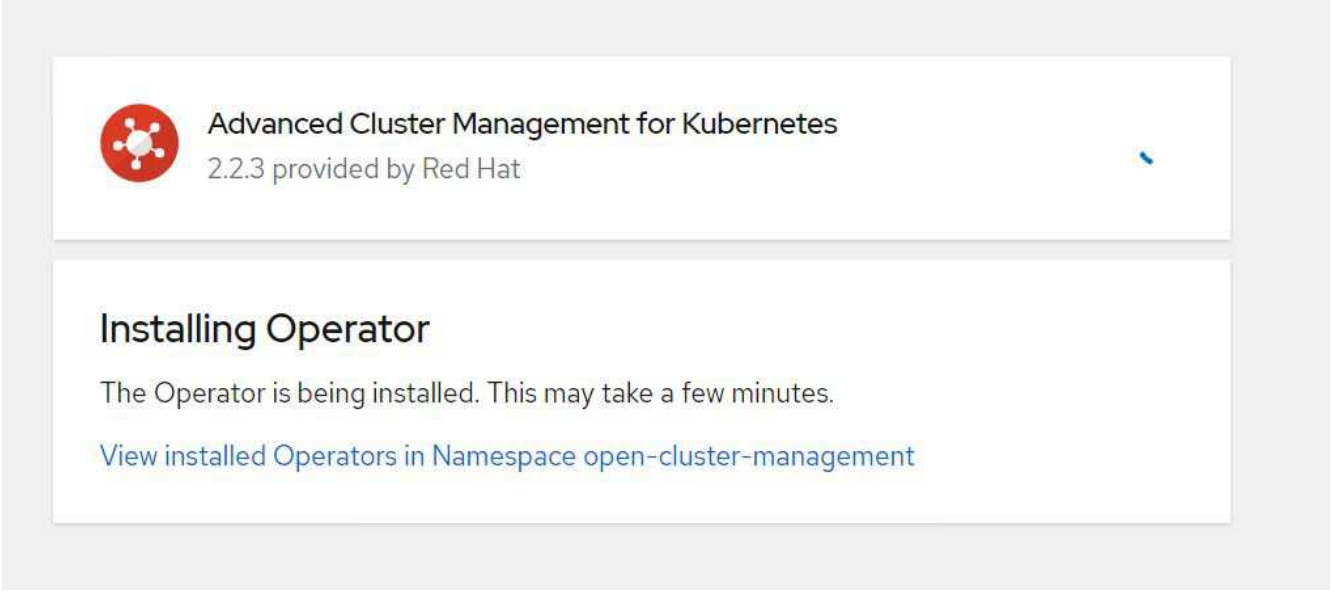
3. Selecione Gerenciamento avançado de cluster para Kubernetes e clique em Instalar.



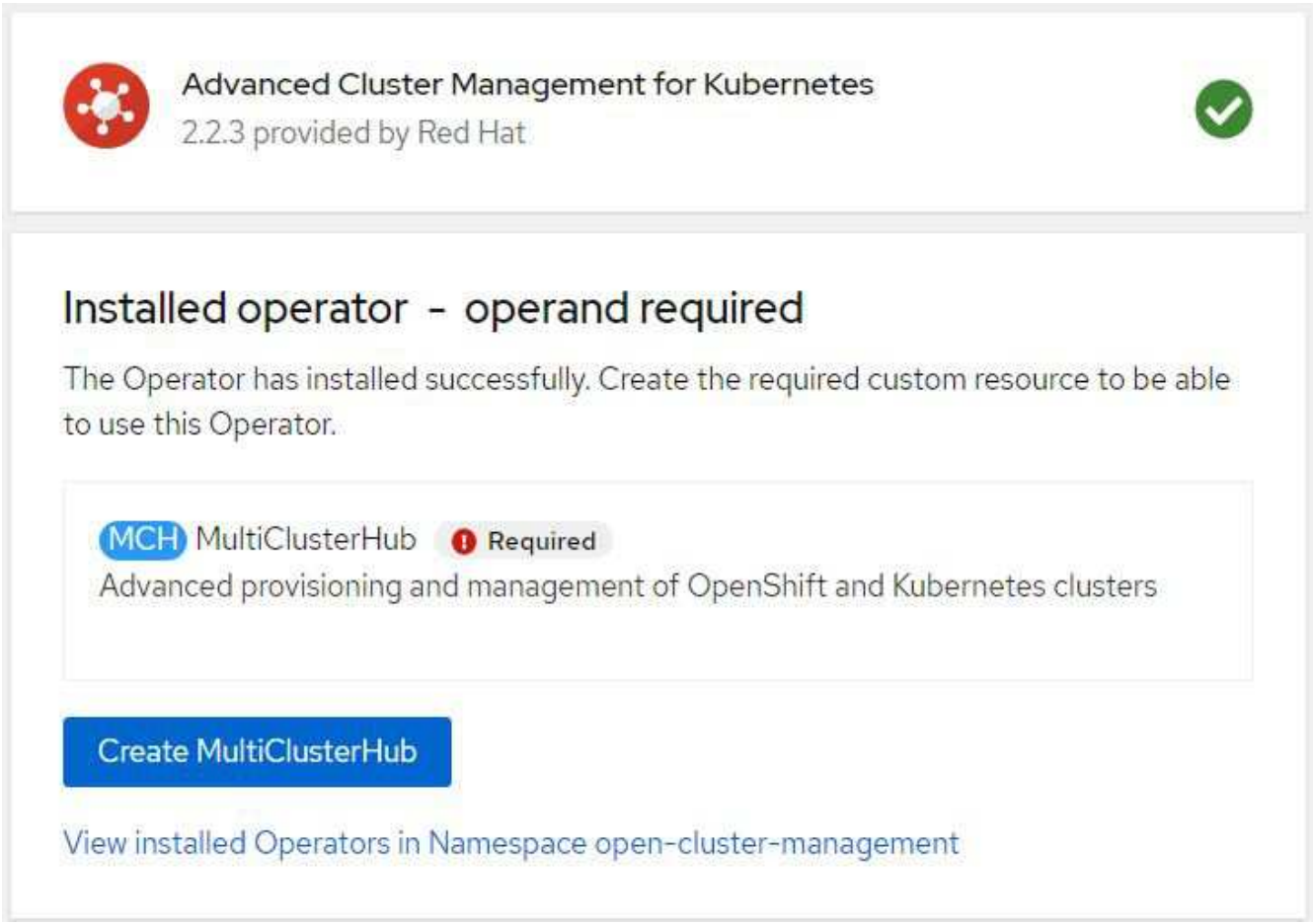
4. Na tela Instalar Operador, forneça os detalhes necessários (a NetApp recomenda manter os parâmetros padrão) e clique em Instalar.



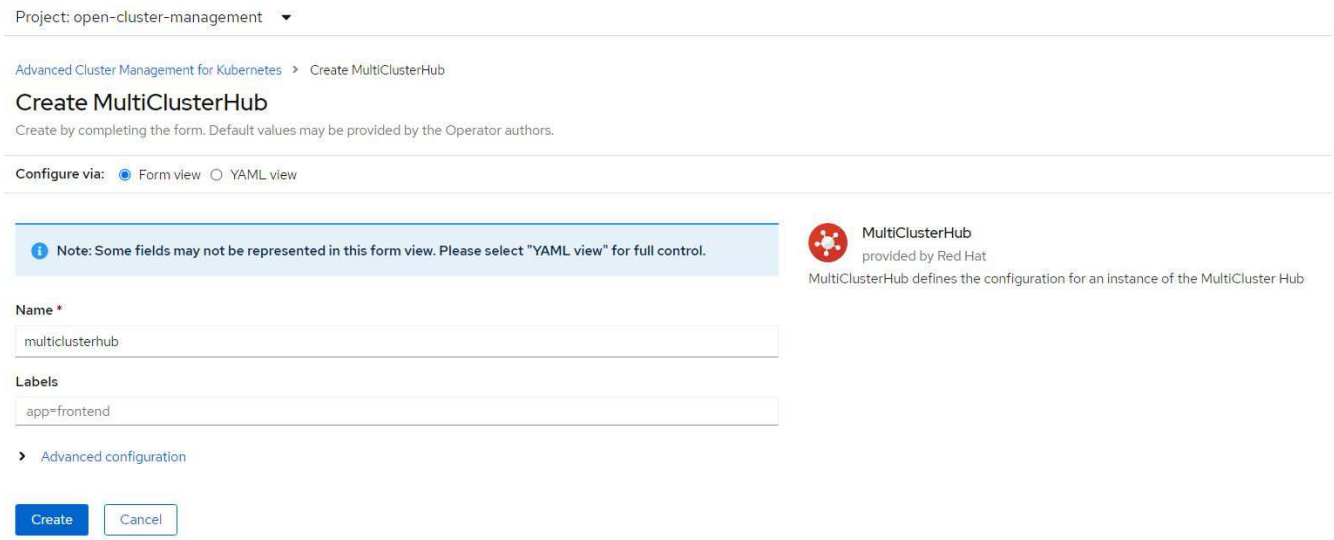
5. Aguarde a conclusão da instalação do operador.



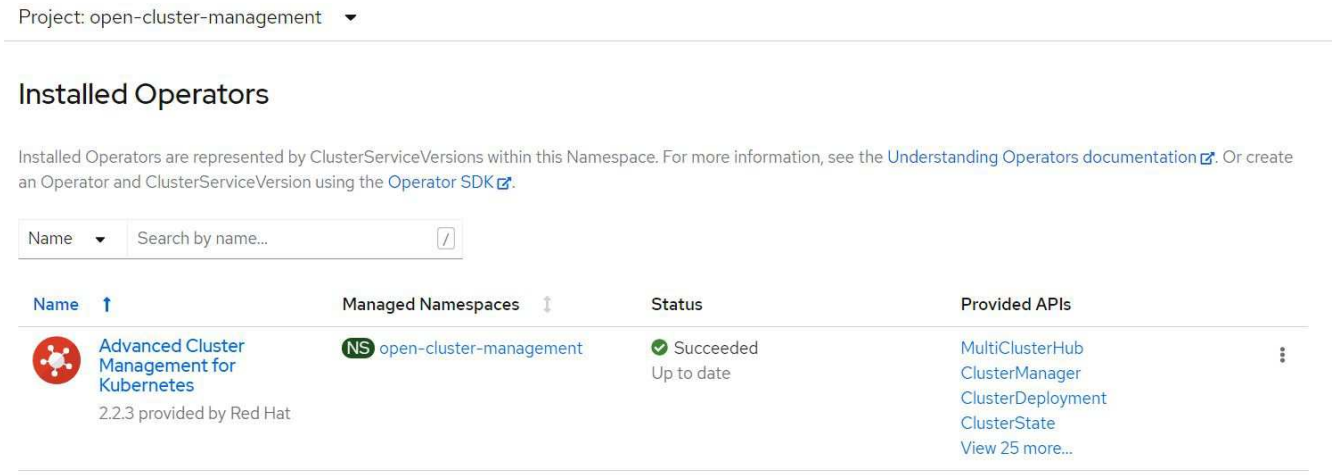
6. Após a instalação do operador, clique em Criar MultiClusterHub.



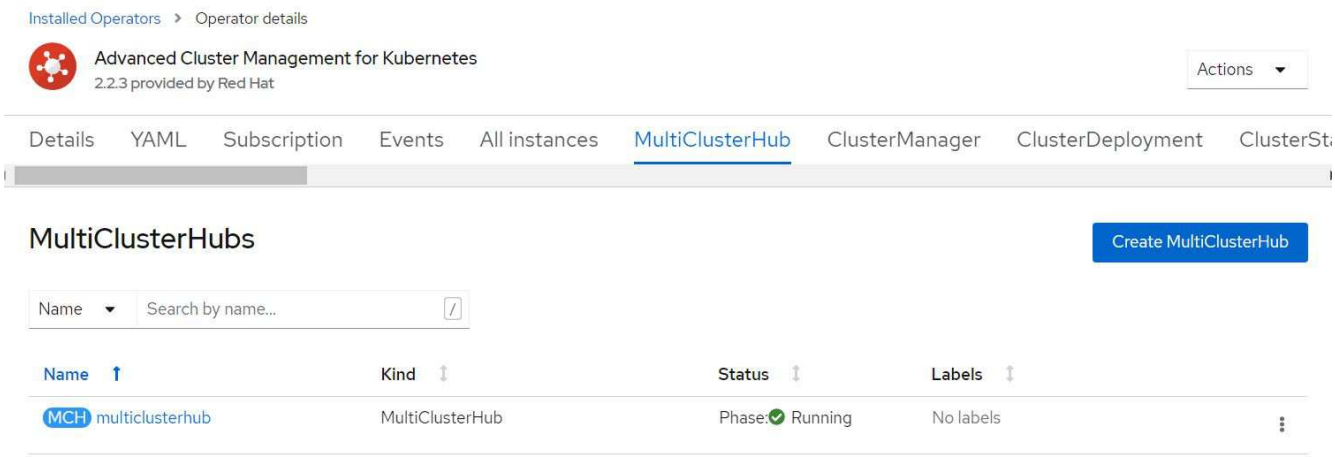
7. Na tela Criar MultiClusterHub, clique em Criar depois de fornecer os detalhes. Isso inicia a instalação de um hub multicluster.



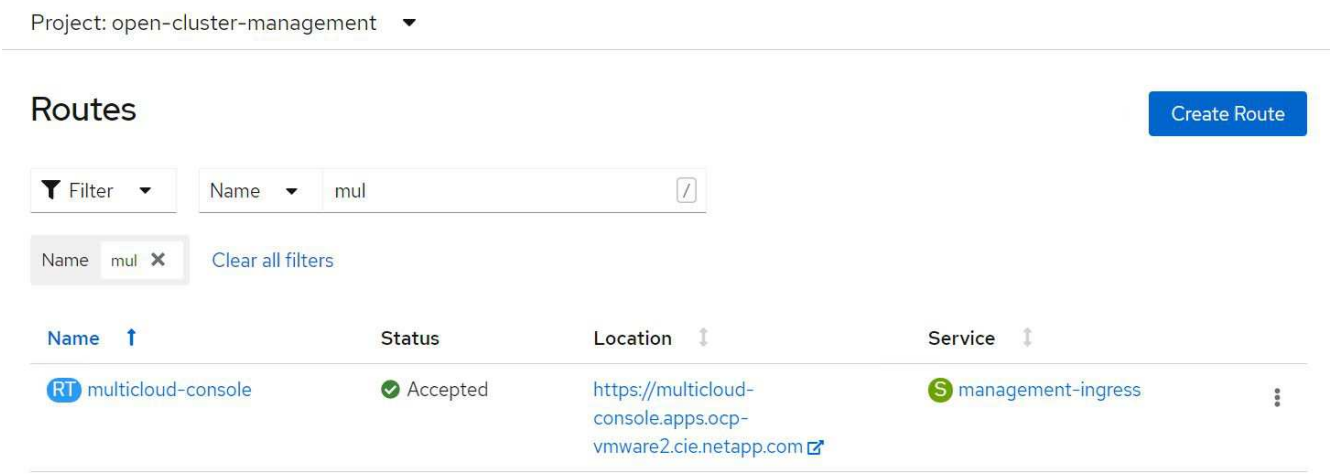
8. Depois que todos os pods passam para o estado Em execução no namespace open-cluster-management e o operador passa para o estado Bem-sucedido, o Advanced Cluster Management for Kubernetes é instalado.



9. Leva algum tempo para concluir a instalação do hub e, depois que isso for concluído, o hub MultiCluster passará para o estado Em execução.



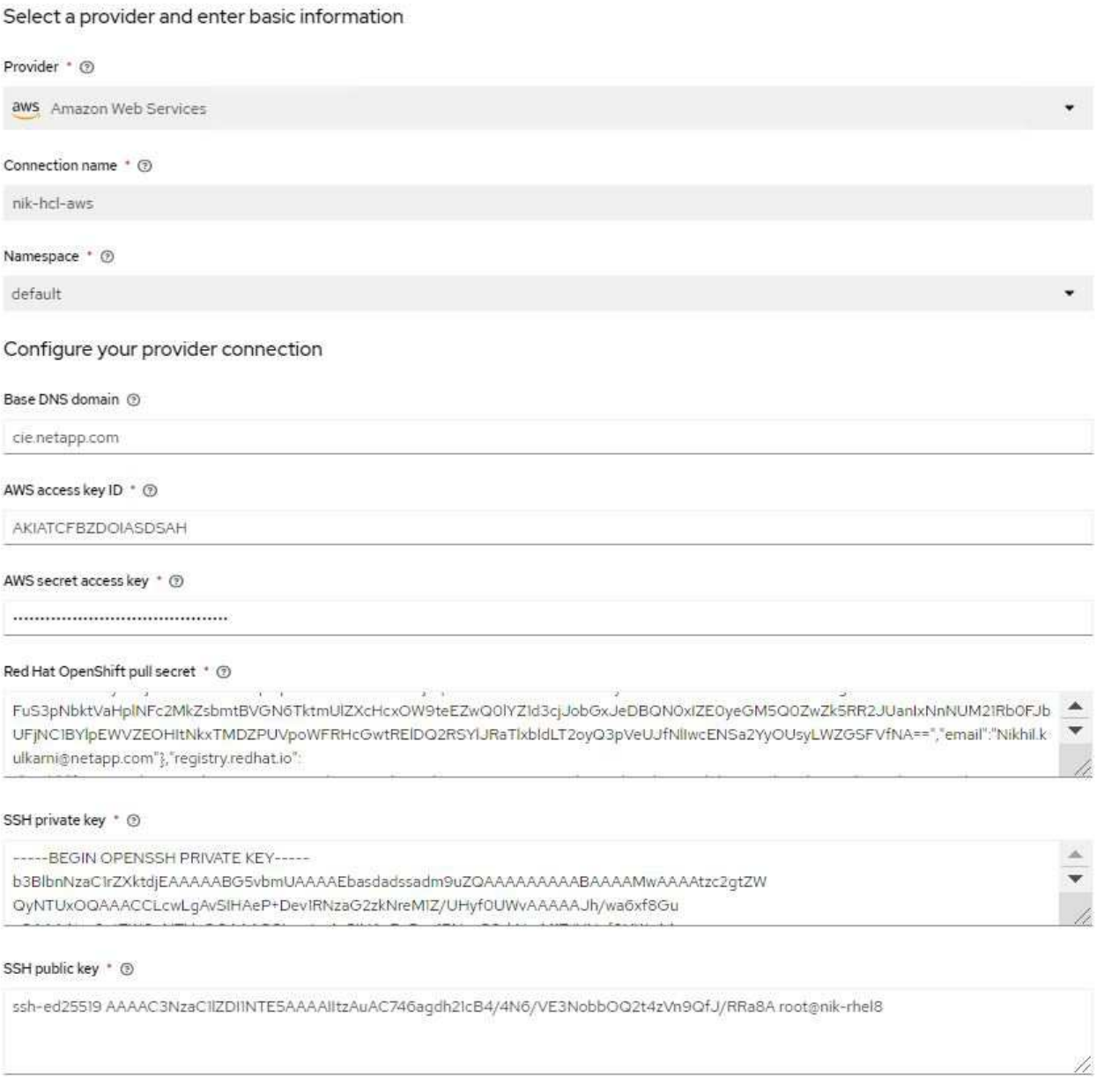
10. Ele cria uma rota no namespace open-cluster-management. Conecte-se à URL na rota para acessar o console do Advanced Cluster Management.



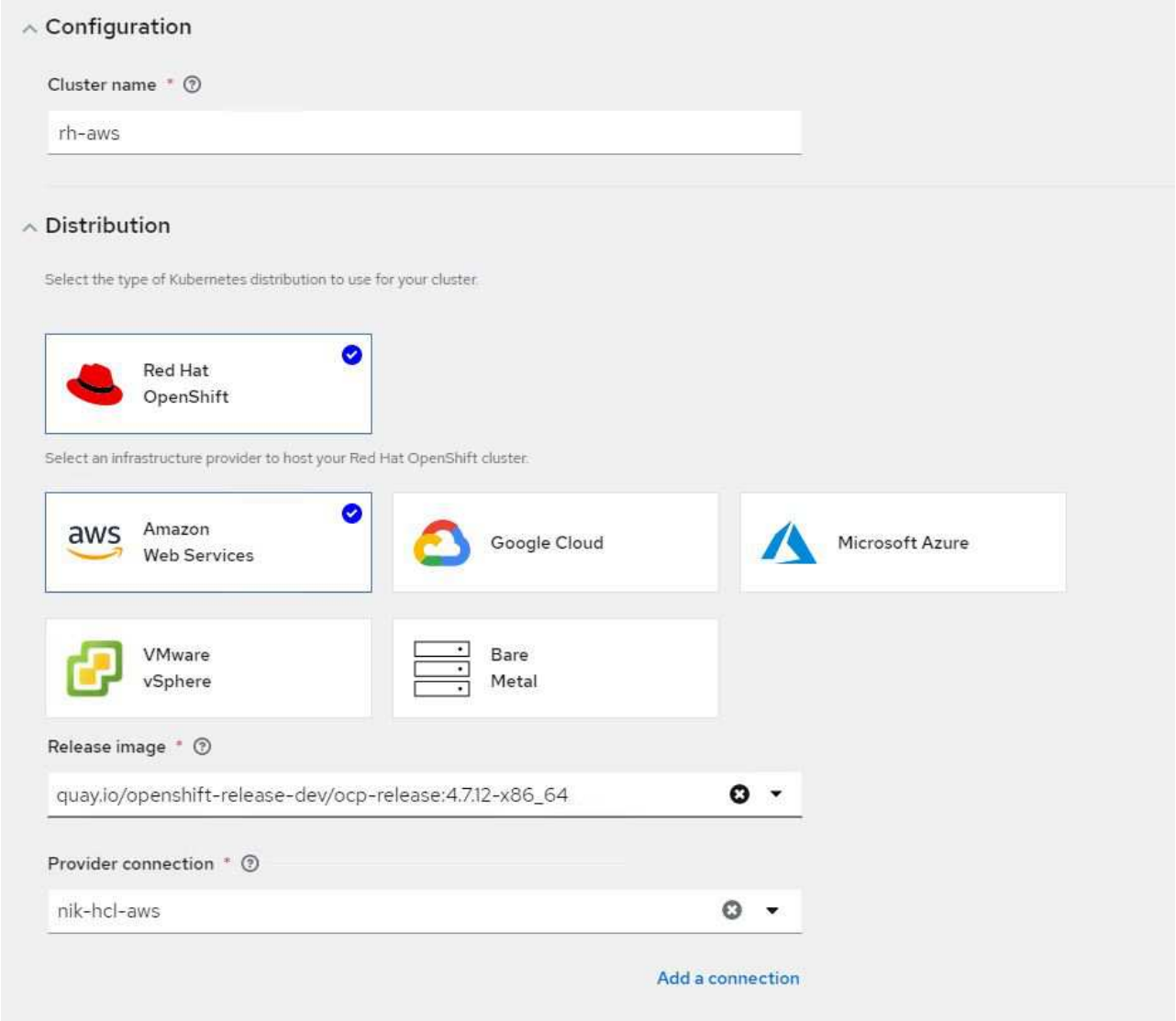
Gerenciamento do ciclo de vida do cluster

Para gerenciar diferentes clusters OpenShift, você pode criá-los ou importá-los para o Advanced Cluster Management.

1. Primeiro navegue até Automatizar infraestruturas > Clusters.
2. Para criar um novo cluster OpenShift, conclua as seguintes etapas:
   a. Criar uma conexão com provedor: navegue até Conexões de provedor e clique em Adicionar uma conexão, forneça todos os detalhes correspondentes ao tipo de provedor selecionado e clique em Adicionar.

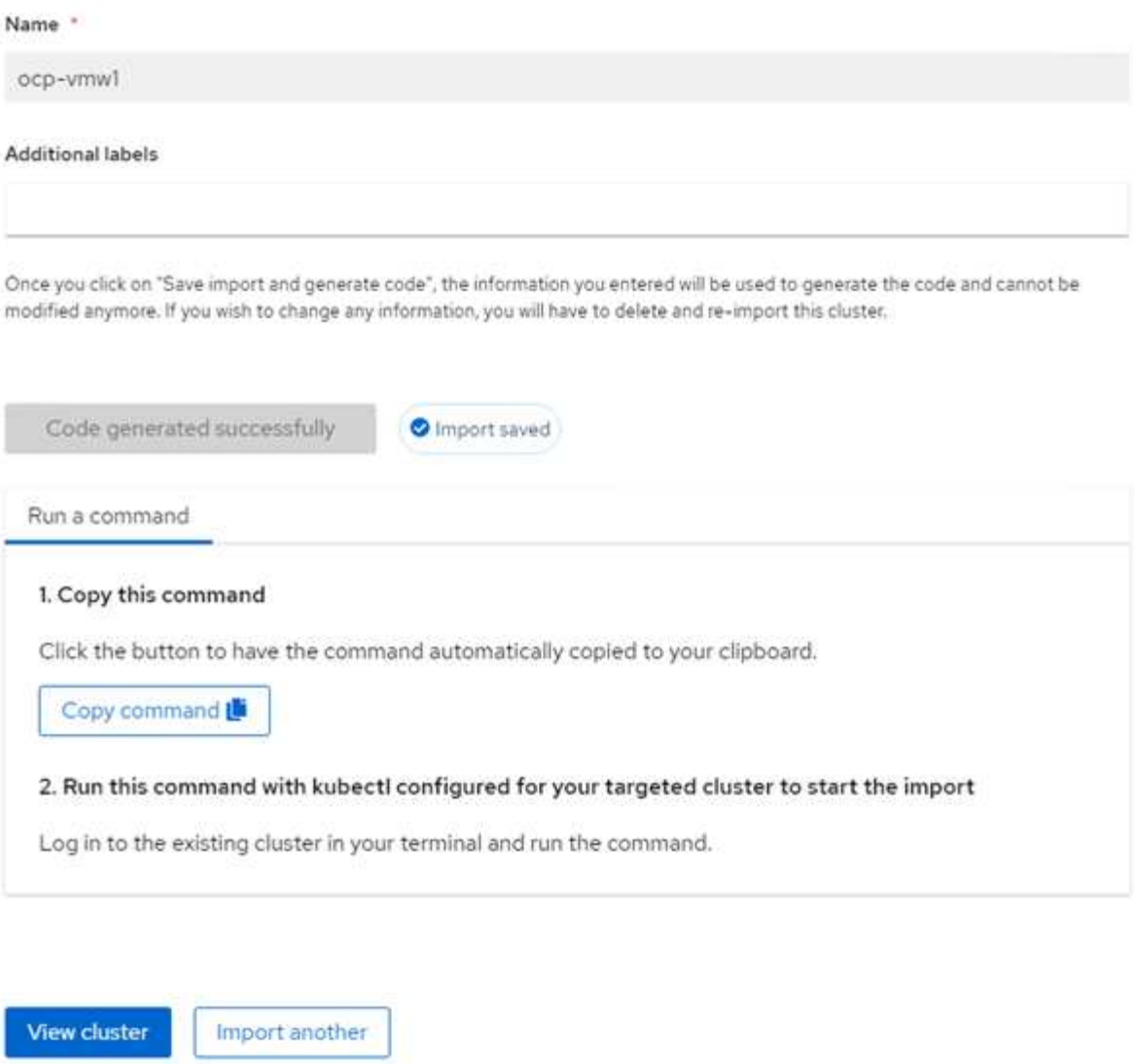


b. Para criar um novo cluster, navegue até Clusters e clique em Adicionar um cluster > Criar um cluster. Forneça os detalhes do cluster e do provedor correspondente e clique em Criar.



c. Após a criação do cluster, ele aparece na lista de clusters com o status Pronto.

3. Para importar um cluster existente, conclua as seguintes etapas:
   a. Navegue até Clusters e clique em Adicionar um cluster > Importar um cluster existente.
   b. Insira o nome do cluster e clique em Salvar importação e gerar código. Um comando para adicionar o cluster existente é exibido.
   c. Clique em Copiar comando e execute o comando no cluster a ser adicionado ao cluster do hub. Isso inicia a instalação dos agentes necessários no cluster e, após a conclusão desse processo, o cluster aparece na lista de clusters com o status Pronto.

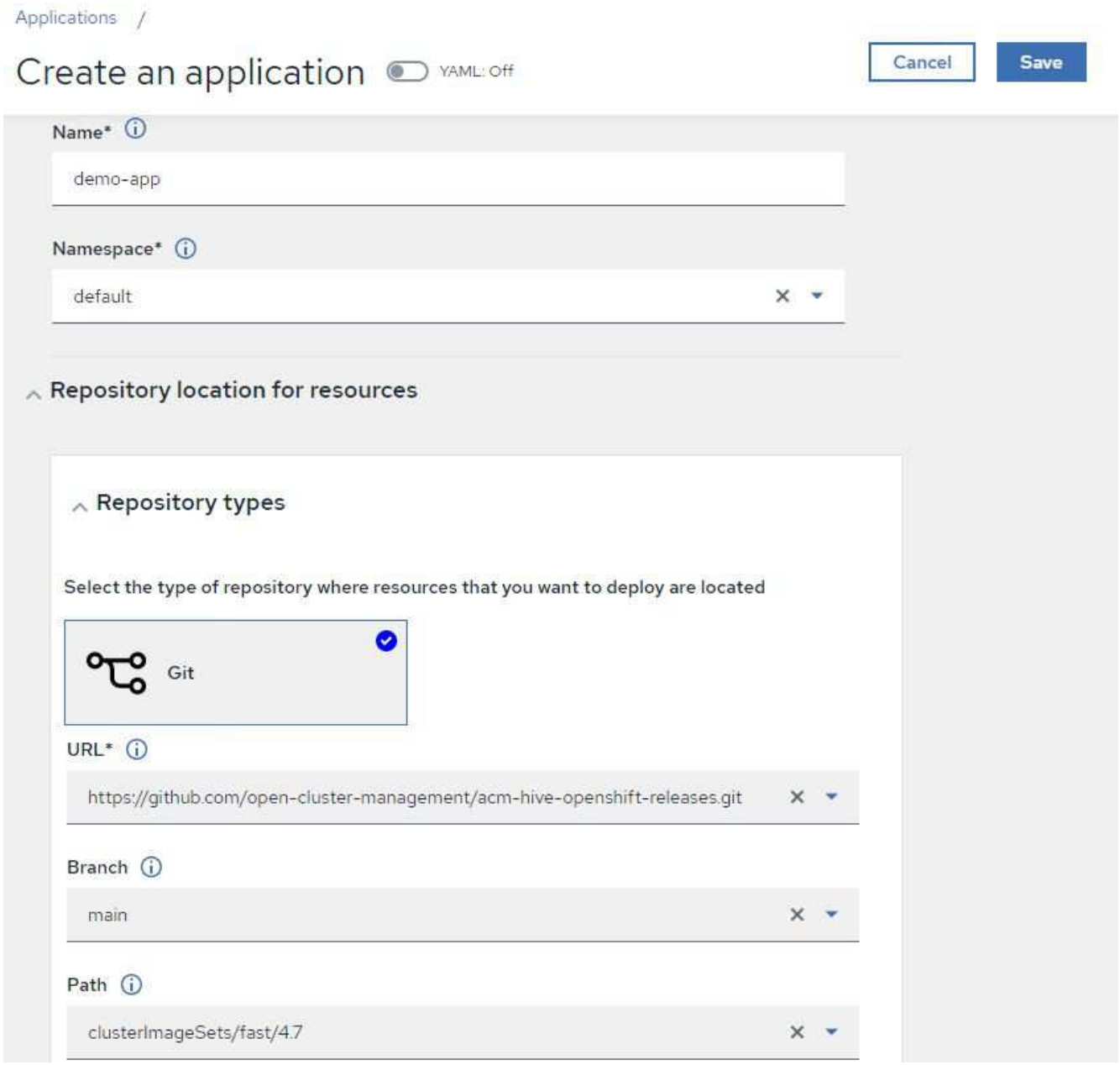


4. Depois de criar e importar vários clusters, você pode monitorá-los e gerenciá-los a partir de um único console.

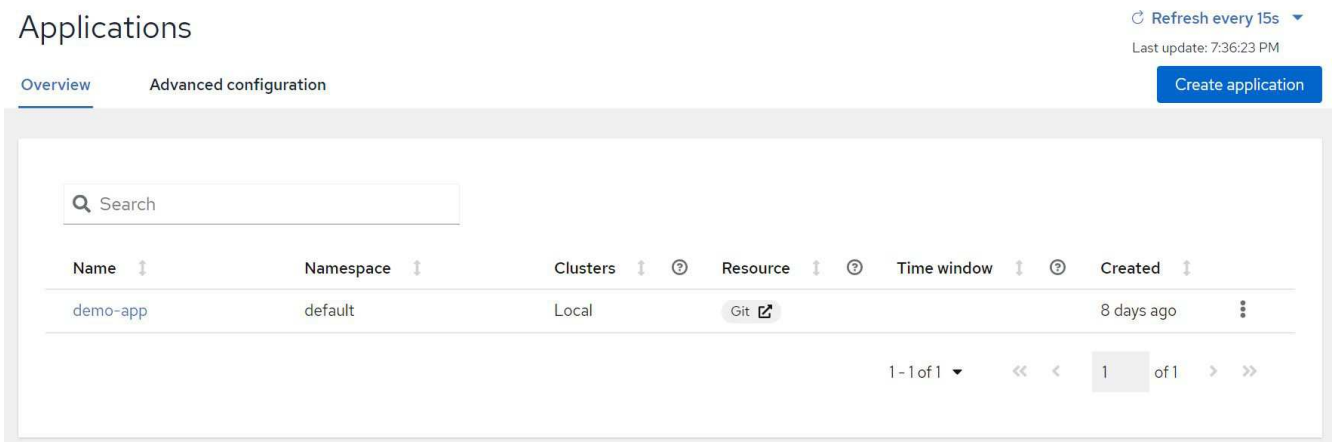
Gerenciamento do ciclo de vida do aplicativo

Para criar um aplicativo e gerenciá-lo em um conjunto de clusters,

1. Navegue até Gerenciar aplicativos na barra lateral e clique em Criar aplicativo. Forneça os detalhes do aplicativo que você gostaria de criar e clique em Salvar.



2. Após a instalação dos componentes do aplicativo, o aplicativo aparece na lista.

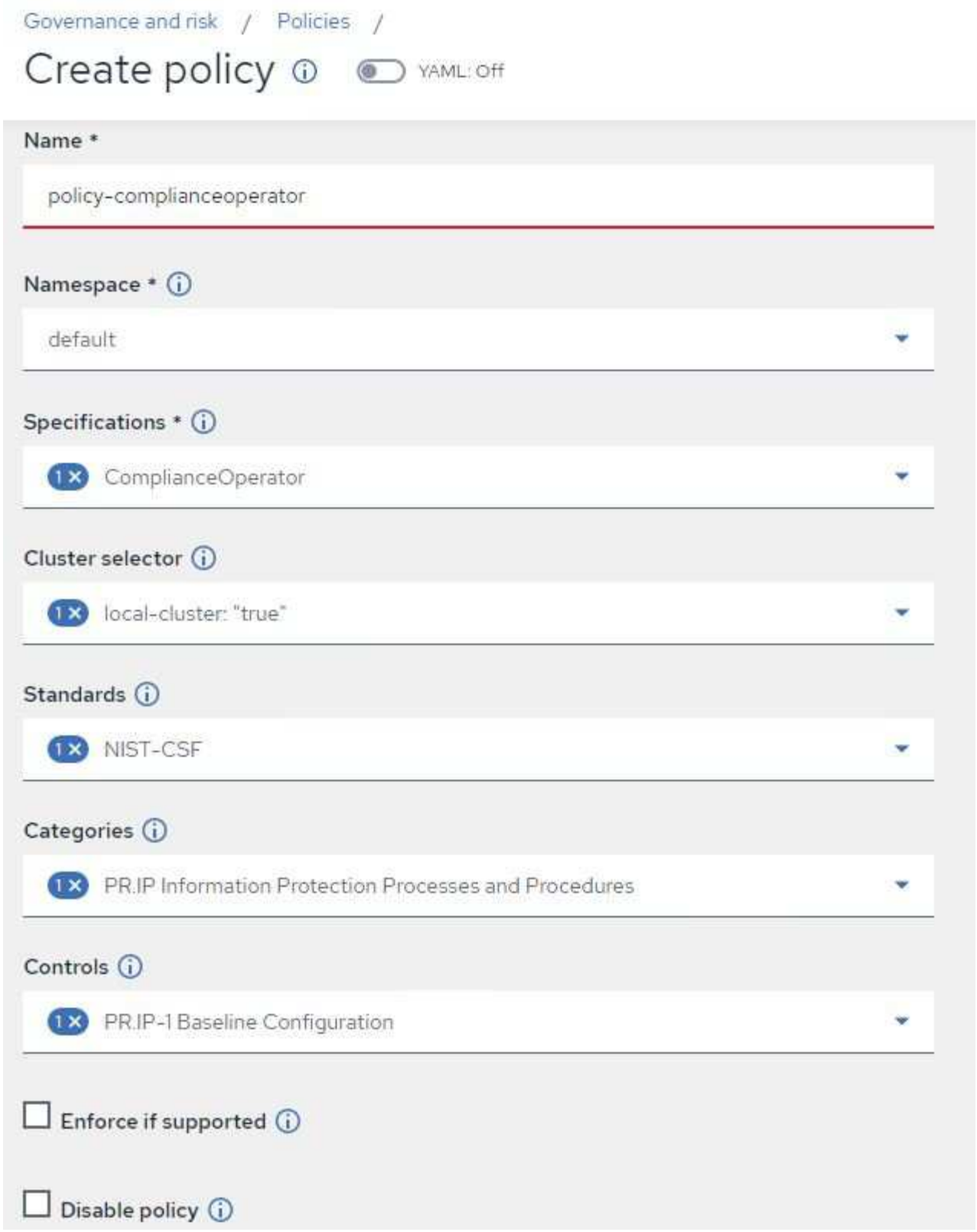


3. O aplicativo agora pode ser monitorado e gerenciado pelo console.

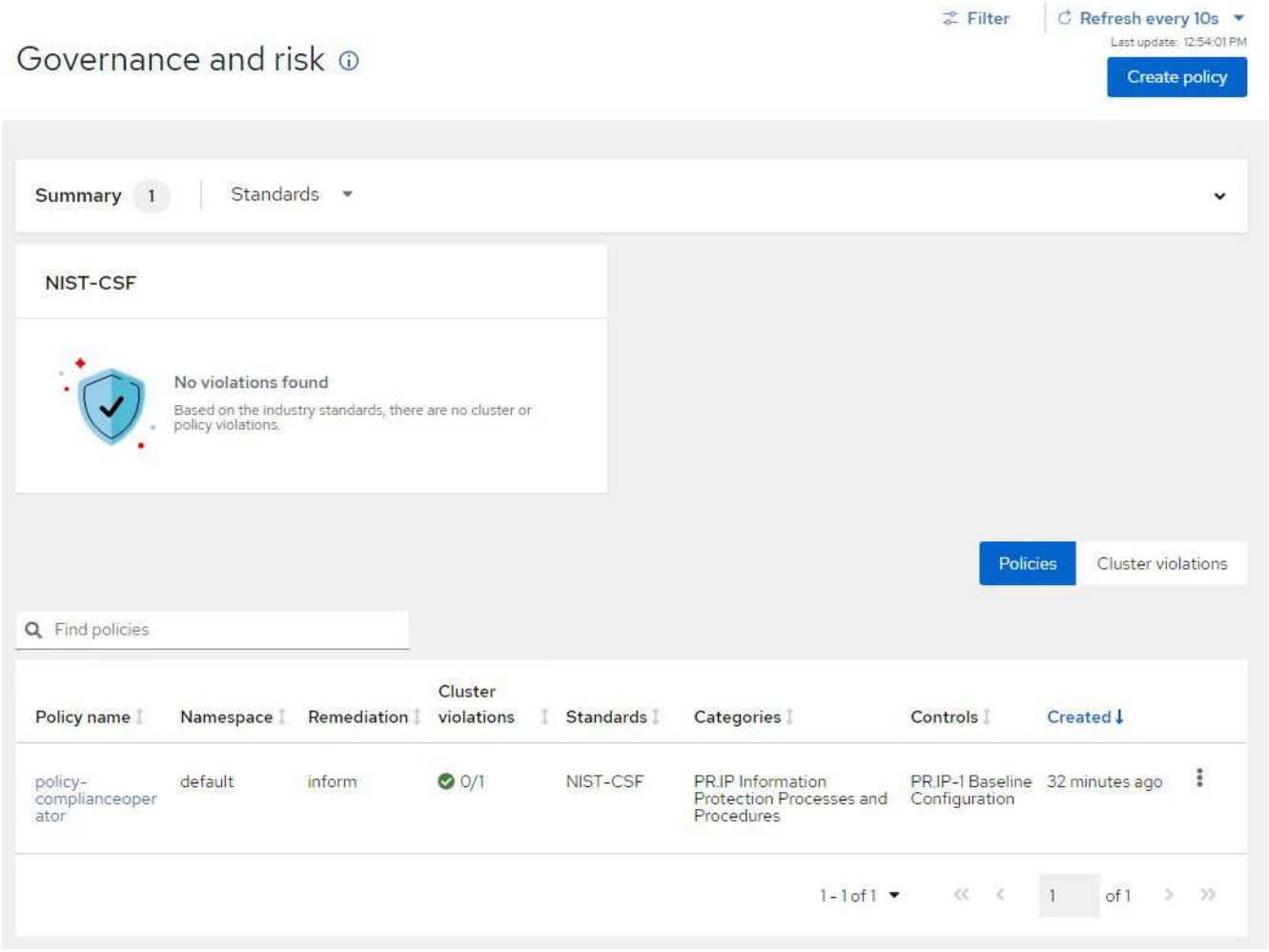
Governança e risco

Esse recurso permite que você defina as políticas de conformidade para diferentes clusters e garanta que os clusters as cumpram. Você pode configurar as políticas para informar ou remediar quaisquer desvios ou violações das regras.

1. Navegue até Governança e Risco na barra lateral.
2. Para criar políticas de conformidade, clique em Criar política, insira os detalhes dos padrões da política e selecione os clusters que devem aderir a essa política. Se você quiser corrigir automaticamente as violações desta política, marque a caixa de seleção Aplicar se suportado e clique em Criar.



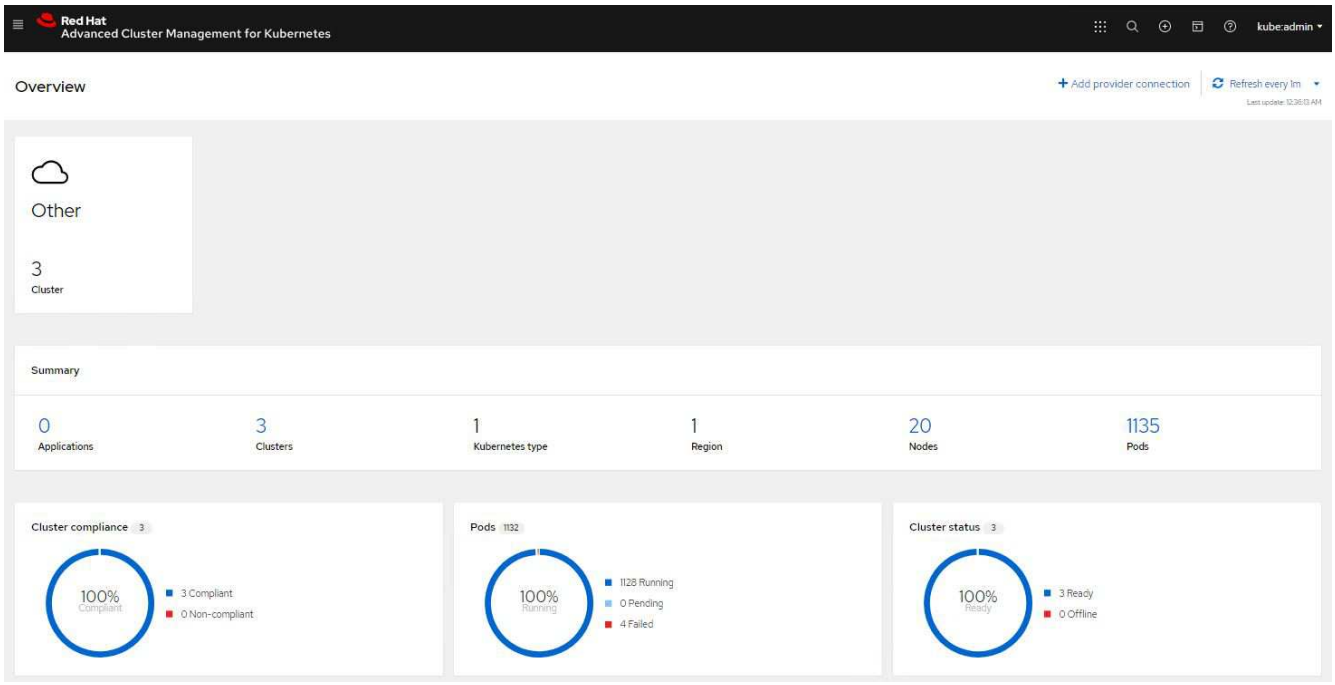
3. Depois que todas as políticas necessárias forem configuradas, qualquer violação de política ou cluster poderá ser monitorada e corrigida pelo Advanced Cluster Management.



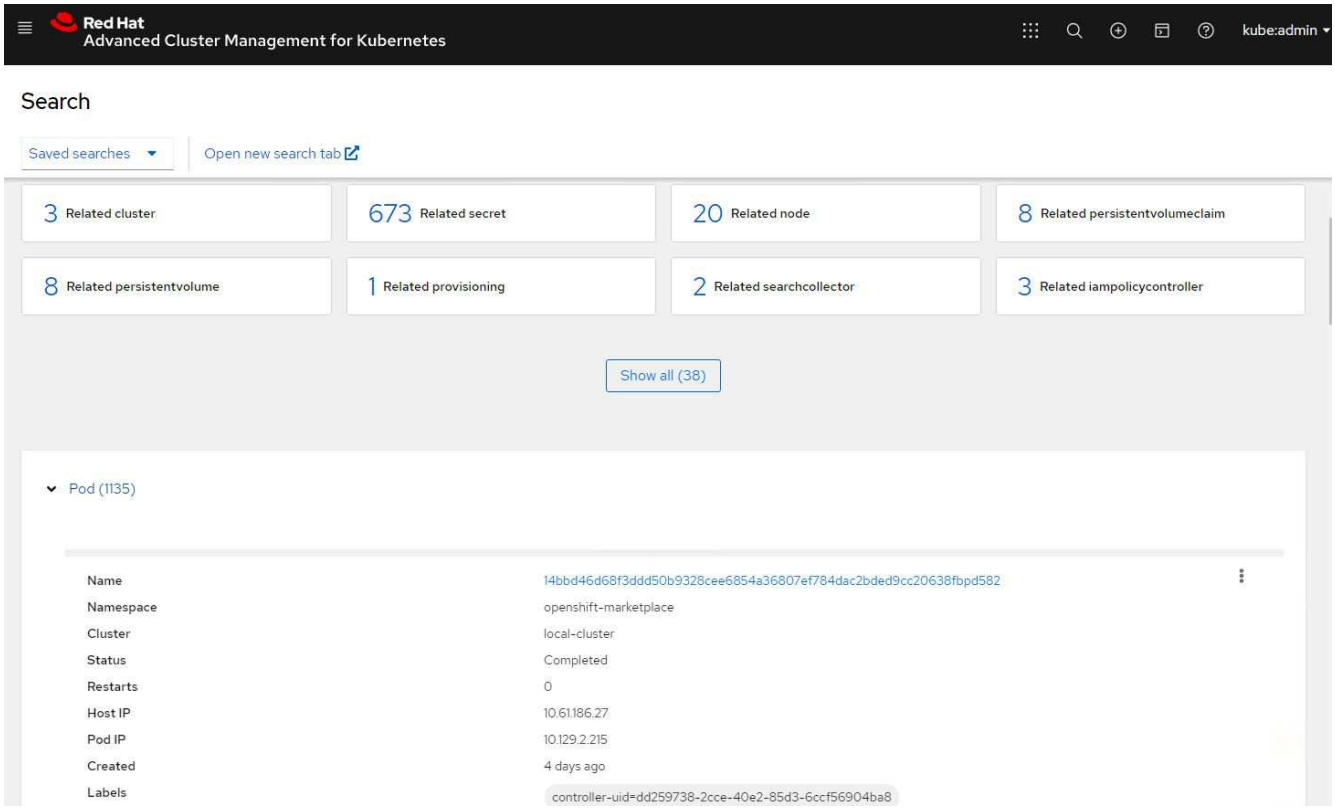
Observabilidade

O Advanced Cluster Management for Kubernetes fornece uma maneira de monitorar nós, pods, aplicativos e cargas de trabalho em todos os clusters.

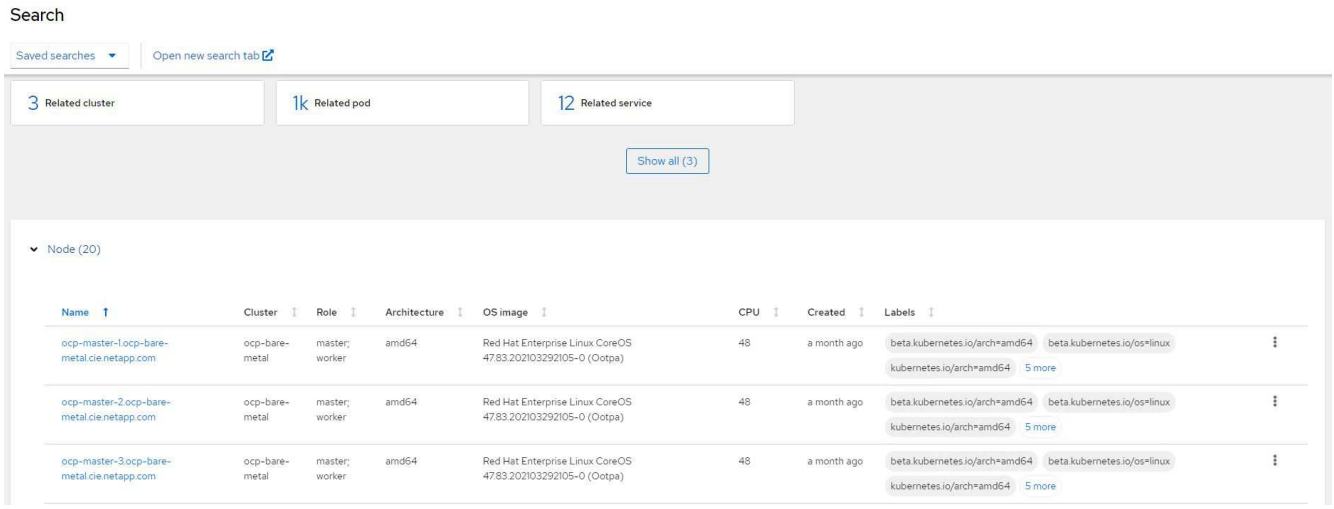
1. Navegue até Observar ambientes > Visão geral.



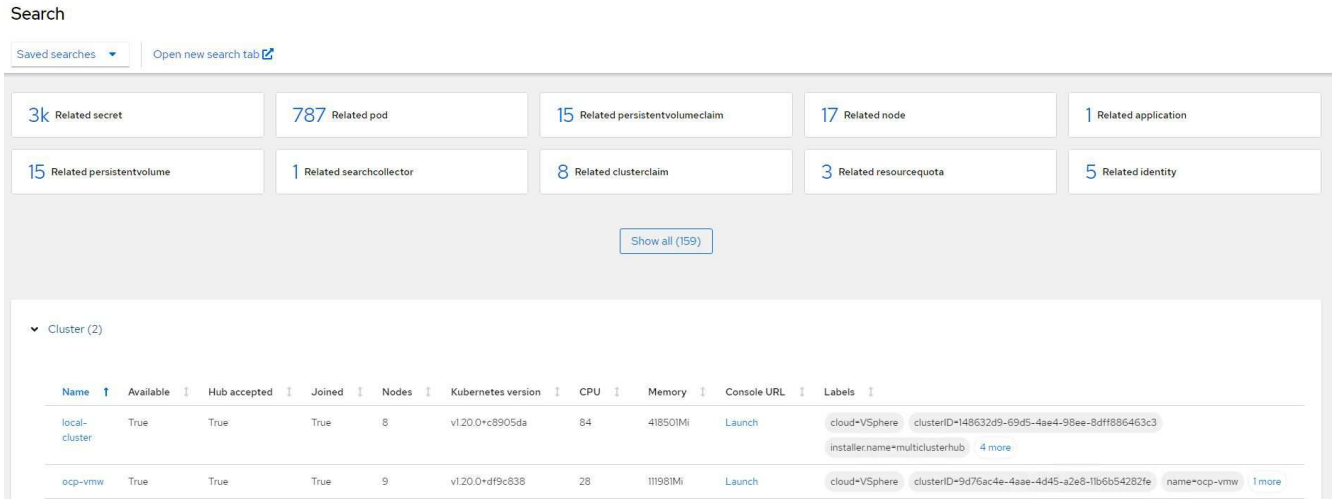
2. Todos os pods e cargas de trabalho em todos os clusters são monitorados e classificados com base em uma variedade de filtros. Clique em Pods para visualizar os dados correspondentes.



3. Todos os nós nos clusters são monitorados e analisados com base em uma variedade de pontos de dados. Clique em Nós para obter mais informações sobre os detalhes correspondentes.

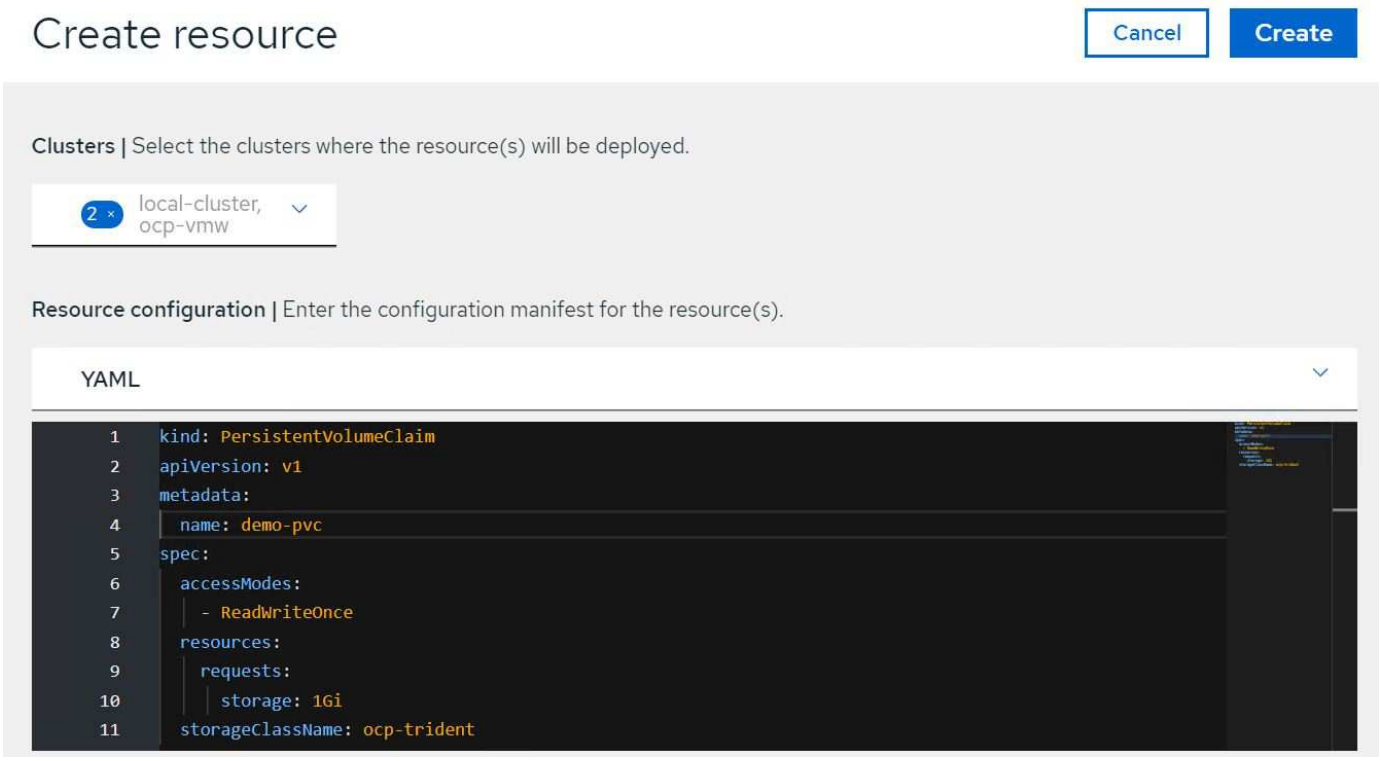


4. Todos os clusters são monitorados e organizados com base em diferentes recursos e parâmetros de cluster. Clique em Clusters para visualizar detalhes do cluster.



Crie recursos em vários clusters

O Advanced Cluster Management for Kubernetes permite que os usuários criem recursos em um ou mais clusters gerenciados simultaneamente no console. Por exemplo, se você tiver clusters OpenShift em diferentes sites com suporte a diferentes clusters NetApp ONTAP e quiser provisionar PVCs em ambos os sites, clique no sinal (+) na barra superior. Em seguida, selecione os clusters nos quais você deseja criar o PVC, cole o YAML do recurso e clique em Criar.



Proteção de dados para aplicativos de contêiner e VMs usando Trident Protect

Esta solução mostra como usar o Trident Protect para executar operações de proteção de dados para contêineres e VMs.

1. Para obter detalhes sobre como criar snapshots e backups e restaurá-los para aplicativos de contêiner na plataforma OpenShift Container, consulte"aqui" .
2. Para obter detalhes sobre como criar e restaurar a partir de um backup para VMs no OpenShift Virtualization implantado na plataforma OpenShift Container, consulte"aqui" .

Proteção de dados para aplicativos de contêiner e VMs usando ferramentas de terceiros

Esta solução mostra como usar o Velero integrado ao operador OADP na plataforma Red Hat OpenShift Container para executar operações de proteção de dados para contêineres e VMs.

1. Para obter detalhes sobre como criar e restaurar um backup para aplicativos de contêiner na plataforma OpenShift Container, consulte"aqui" .
2. Para obter detalhes sobre como criar e restaurar a partir de um backup para VMs no OpenShift Virtualization implantado na plataforma OpenShift Container, consulte"aqui" .

Recursos adicionais para aprender sobre a integração do Red Hat OpenShift Virtualization com o armazenamento NetApp

Acesse recursos adicionais que oferecem mais informações sobre suporte à implantação, gerenciamento e otimização do Red Hat OpenShift Virtualization com ONTAP em várias plataformas e tecnologias.

- Documentação da NetApp

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

- Documentação Trident

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

- Documentação do Red Hat OpenShift

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

- Documentação da plataforma Red Hat OpenStack

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

- Documentação de virtualização do Red Hat

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

- Documentação do VMware vSphere

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

Informações sobre direitos autorais

Informações sobre marcas comerciais