Red Hat OpenShift con NetApp

NetApp container solutions

NetApp
January 21, 2026

Sommario

Red Hat OpenShift con NetApp

NVA-1160: Red Hat OpenShift con NetApp

Alan Cowles e Nikhil M Kulkarni, NetApp

Questo documento di riferimento fornisce la convalida della distribuzione della soluzione Red Hat OpenShift, distribuita tramite Installer Provisioned Infrastructure (IPI) in diversi ambienti di data center, come convalidato da NetApp. Descrive inoltre in dettaglio l'integrazione dello storage con i sistemi di storage NetApp mediante l'utilizzo dell'orchestratore di storage Trident per la gestione dello storage persistente. Infine, vengono esaminate e documentate una serie di convalide di soluzioni e casi d'uso reali.

Casi d'uso

La soluzione Red Hat OpenShift con NetApp è progettata per offrire un valore eccezionale ai clienti con i seguenti casi d'uso:

- Red Hat OpenShift è facile da implementare e gestire, distribuito tramite IPI (Installer Provisioned Infrastructure) su bare metal, Red Hat OpenStack Platform, Red Hat Virtualization e VMware vSphere.
- Potenza combinata di container aziendali e carichi di lavoro virtualizzati con Red Hat OpenShift distribuiti virtualmente su OSP, RHV o vSphere, oppure su bare metal con OpenShift Virtualization.
- Configurazione e casi d'uso reali che evidenziano le funzionalità di Red Hat OpenShift quando utilizzato con lo storage NetApp e Trident, l'orchestratore di storage open source per Kubernetes.

Valore aziendale

Le aziende stanno adottando sempre più pratiche DevOps per creare nuovi prodotti, abbreviare i cicli di rilascio e aggiungere rapidamente nuove funzionalità. Grazie alla loro innata natura agile, i container e i microservizi svolgono un ruolo cruciale nel supportare le pratiche DevOps. Tuttavia, mettere in pratica DevOps su scala produttiva in un ambiente aziendale presenta le sue sfide e impone determinati requisiti all'infrastruttura sottostante, come i seguenti:

- Elevata disponibilità a tutti i livelli dello stack
- Facilità delle procedure di distribuzione
- Operazioni e aggiornamenti non dirompenti
- Infrastruttura programmabile e basata su API per tenere il passo con l'agilità dei microservizi
- Multitenancy con garanzie di prestazione
- Capacità di eseguire carichi di lavoro virtualizzati e containerizzati simultaneamente
- Capacità di scalare l'infrastruttura in modo indipendente in base alle richieste del carico di lavoro

Red Hat OpenShift con NetApp riconosce queste sfide e presenta una soluzione che aiuta a risolvere ogni problema implementando la distribuzione completamente automatizzata di RedHat OpenShift IPI nell'ambiente del data center scelto dal cliente.

Panoramica della tecnologia

La soluzione Red Hat OpenShift con NetApp è composta dai seguenti componenti principali:

Piattaforma container Red Hat OpenShift

Red Hat OpenShift Container Platform è una piattaforma Kubernetes aziendale completamente supportata. Red Hat apporta diversi miglioramenti a Kubernetes open source per fornire una piattaforma applicativa con tutti i componenti completamente integrati per creare, distribuire e gestire applicazioni containerizzate.

Per maggiori informazioni visita il sito web di OpenShift "Qui" .

Sistemi di archiviazione NetApp

NetApp dispone di diversi sistemi di storage ideali per data center aziendali e implementazioni cloud ibride. Il portfolio NetApp include i sistemi di storage NetApp ONTAP, NetApp Element e NetApp e-Series, tutti in grado di fornire storage persistente per applicazioni containerizzate.

Per maggiori informazioni visita il sito web NetApp "Qui" .

Integrazioni di storage NetApp

Trident è un orchestratore di storage open source e completamente supportato per container e distribuzioni Kubernetes, tra cui Red Hat OpenShift.

Per maggiori informazioni, visita il sito web Trident "Qui" .

Opzioni di configurazione avanzate

Questa sezione è dedicata alle personalizzazioni che gli utenti reali potrebbero dover eseguire durante la distribuzione di questa soluzione in produzione, ad esempio la creazione di un registro di immagini privato dedicato o la distribuzione di istanze di bilanciamento del carico personalizzate.

Matrice di supporto attuale per le versioni convalidate

Tecnologia	Scopo	Versione del software
NetApp ONTAP	Magazzinaggio	9.8, 9.9.1, 9.12.1
NetApp Element	Magazzinaggio	12,3
NetApp Trident	Orchestrazione dell’archiviazione	22.01.0, 23.04, 23.07, 23.10, 24.02
Red Hat OpenShift	Orchestrazione dei contenitori	4.6 UE, 4.7, 4.8, 4.10, 4.11, 4.12, 4.13, 4.14
VMware vSphere	Virtualizzazione del data center	7.0, 8.0.2

Red Hat Openshift

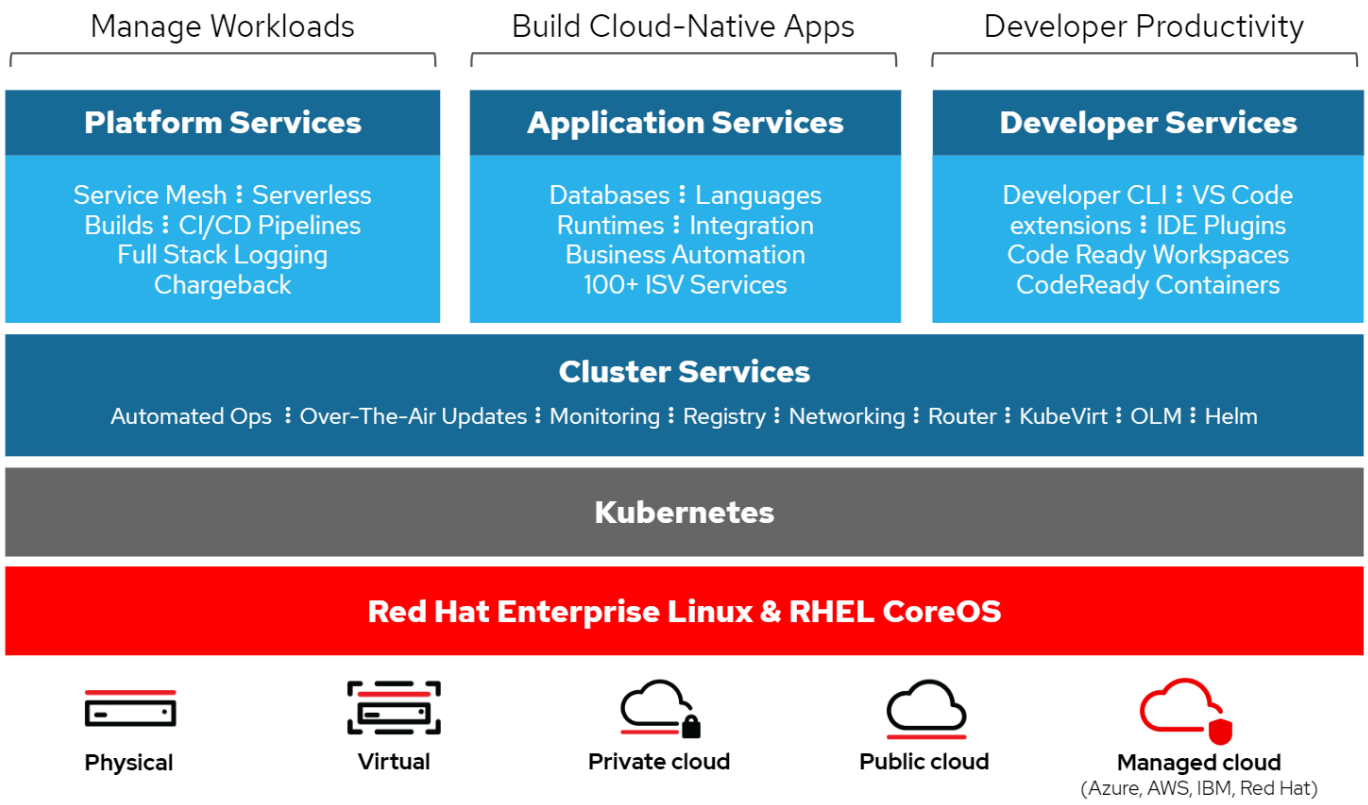
Panoramica di OpenShift

Red Hat OpenShift Container Platform unisce sviluppo e operazioni IT in un’unica piattaforma per creare, distribuire e gestire le applicazioni in modo coerente su

infrastrutture cloud ibride e on-premise. Red Hat OpenShift si basa sull'innovazione open source e sugli standard di settore, tra cui Kubernetes e Red Hat Enterprise Linux CoreOS, la distribuzione Linux aziendale leader al mondo progettata per carichi di lavoro basati su container. OpenShift fa parte del programma Certified Kubernetes della Cloud Native Computing Foundation (CNCF), che garantisce portabilità e interoperabilità dei carichi di lavoro dei container.

Red Hat OpenShift offre le seguenti funzionalità:

- **Provisioning self-service** Gli sviluppatori possono creare applicazioni su richiesta in modo rapido e semplice utilizzando gli strumenti che utilizzano più spesso, mentre gli operatori mantengono il controllo completo sull'intero ambiente.
- **Archiviazione persistente** Grazie al supporto per l'archiviazione persistente, OpenShift Container Platform consente di eseguire sia applicazioni stateful che applicazioni stateless native del cloud.
- **Integrazione continua e sviluppo continuo (CI/CD)** Questa piattaforma di codice sorgente gestisce le immagini di build e distribuzione su larga scala.
- **Standard open source** Questi standard incorporano l'Open Container Initiative (OCI) e Kubernetes per l'orchestrazione dei container, oltre ad altre tecnologie open source. Non sei vincolato alla tecnologia o alla roadmap aziendale di un fornitore specifico.
- **Pipeline CI/CD** OpenShift fornisce supporto immediato per le pipeline CI/CD, in modo che i team di sviluppo possano automatizzare ogni fase del processo di distribuzione delle applicazioni e assicurarsi che venga eseguita a ogni modifica apportata al codice o alla configurazione dell'applicazione.
- **Controllo degli accessi basato sui ruoli (RBAC)** Questa funzionalità consente di monitorare team e utenti per aiutare a organizzare un ampio gruppo di sviluppatori.
- **Creazione e distribuzione automatizzate** OpenShift offre agli sviluppatori la possibilità di creare le proprie applicazioni containerizzate o di far sì che la piattaforma crei i container a partire dal codice sorgente dell'applicazione o persino dai file binari. La piattaforma automatizza quindi la distribuzione di queste applicazioni nell'infrastruttura in base alle caratteristiche definite per le applicazioni. Ad esempio, la quantità di risorse da allocare e la posizione sull'infrastruttura in cui distribuirle affinché siano conformi alle licenze di terze parti.
- **Ambienti coerenti** OpenShift garantisce che l'ambiente fornito agli sviluppatori e per l'intero ciclo di vita dell'applicazione sia coerente dal sistema operativo, alle librerie, alla versione runtime (ad esempio, Java runtime) e persino al runtime dell'applicazione in uso (ad esempio, Tomcat), per eliminare i rischi derivanti da ambienti incoerenti.
- **Gestione della configurazione** La gestione della configurazione e dei dati sensibili è integrata nella piattaforma per garantire che all'applicazione venga fornita una configurazione dell'applicazione coerente e indipendente dall'ambiente, indipendentemente dalle tecnologie utilizzate per creare l'applicazione o dall'ambiente in cui viene distribuita.
- **Registri e metriche delle applicazioni.** Il feedback rapido è un aspetto importante nello sviluppo delle applicazioni. Il monitoraggio integrato e la gestione dei log di OpenShift forniscono metriche immediate agli sviluppatori, consentendo loro di analizzare il comportamento dell'applicazione in seguito alle modifiche e di risolvere i problemi il prima possibile nel ciclo di vita dell'applicazione.
- **Sicurezza e catalogo dei container** OpenShift offre multi-tenancy e protegge l'utente dall'esecuzione di codice dannoso utilizzando la sicurezza consolidata con Security-Enhanced Linux (SELinux), CGroups e Secure Computing Mode (seccomp) per isolare e proteggere i container. Fornisce inoltre la crittografia tramite certificati TLS per i vari sottosistemi e l'accesso ai container certificati Red Hat (access.redhat.com/containers) che vengono scansionati e classificati con particolare attenzione alla sicurezza per fornire agli utenti finali container applicativi certificati, affidabili e sicuri.



Metodi di distribuzione per Red Hat OpenShift

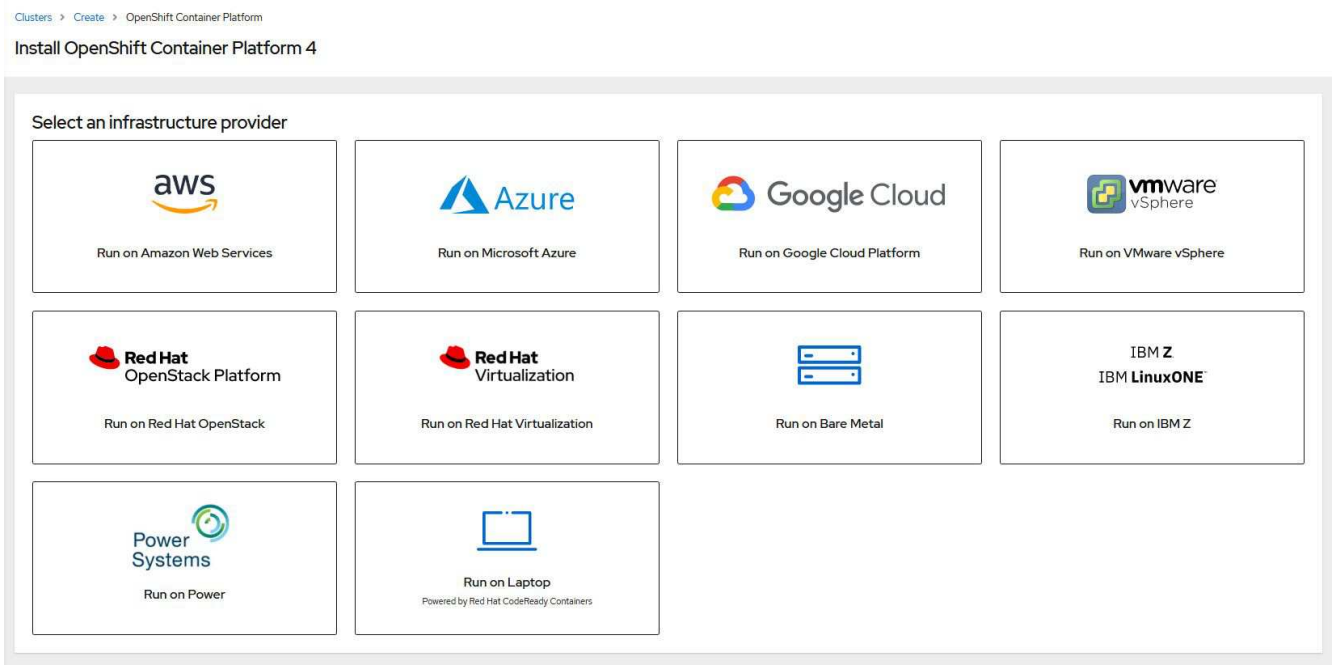
A partire da Red Hat OpenShift 4, i metodi di distribuzione per OpenShift includono distribuzioni manuali mediante User Provisioned Infrastructure (UPI) per distribuzioni altamente personalizzate o distribuzioni completamente automatizzate mediante Installer Provisioned Infrastructure (IPI).

Il metodo di installazione IPI è quello preferito nella maggior parte dei casi perché consente la rapida distribuzione dei cluster OpenShift per ambienti di sviluppo, test e produzione.

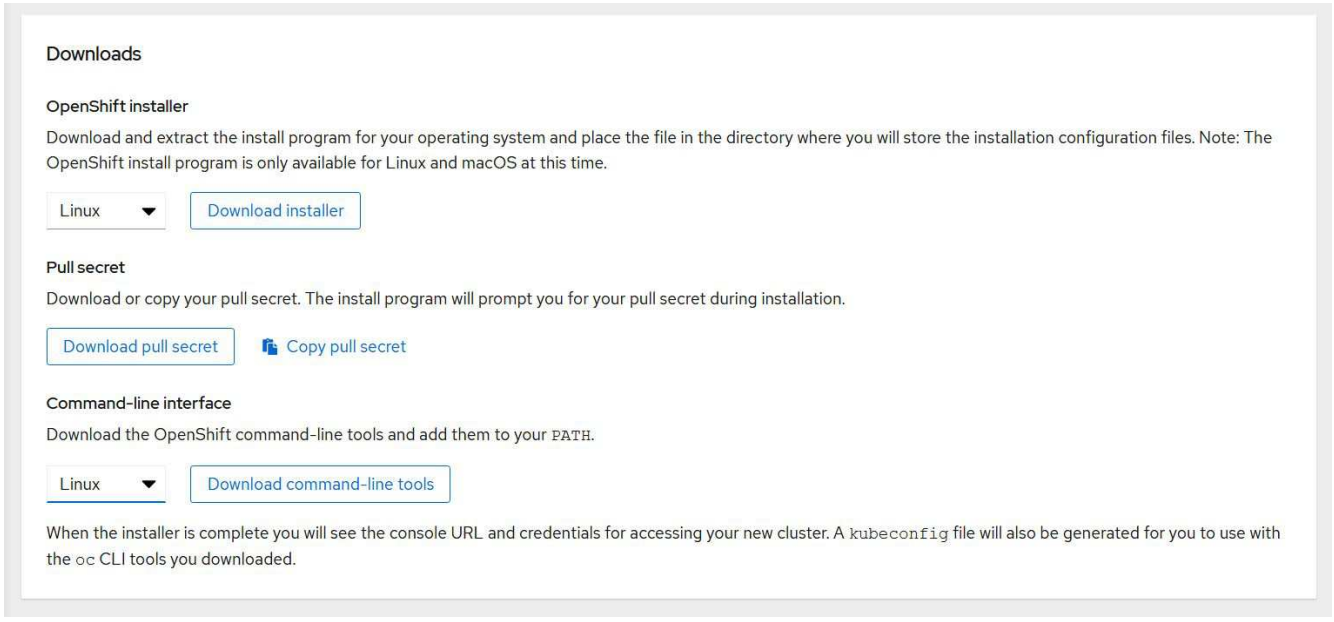
Installazione IPI di Red Hat OpenShift

L'implementazione dell'infrastruttura di provisioning dell'installatore (IPI) di OpenShift prevede i seguenti passaggi di alto livello:

1. Visita Red Hat OpenShift"sito web" ed effettua l'accesso con le tue credenziali SSO.
2. Seleziona l'ambiente in cui desideri distribuire Red Hat OpenShift.



3. Nella schermata successiva scarica il programma di installazione, il segreto pull univoco e gli strumenti CLI per la gestione.



4. Segui il"istruzioni di installazione" fornito da Red Hat per l'implementazione nell'ambiente di tua scelta.

Distribuzioni OpenShift convalidate da NetApp

NetApp ha testato e convalidato la distribuzione di Red Hat OpenShift nei suoi laboratori utilizzando il metodo di distribuzione Installer Provisioned Infrastructure (IPI) in ciascuno dei seguenti ambienti di data center:

- "OpenShift su Bare Metal"
- "OpenShift sulla piattaforma Red Hat OpenStack"
- "OpenShift su Red Hat Virtualization"
- "OpenShift su VMware vSphere"

OpenShift su Bare Metal

OpenShift su Bare Metal fornisce una distribuzione automatizzata di OpenShift Container Platform su server commerciali.

OpenShift su Bare Metal è simile alle distribuzioni virtuali di OpenShift, che garantiscono semplicità di distribuzione, provisioning rapido e scalabilità dei cluster OpenShift, supportando al contempo carichi di lavoro virtualizzati per applicazioni che non sono pronte per essere containerizzate. Eseguendo l'implementazione su bare metal, non è necessario il sovraccarico aggiuntivo necessario per gestire l'ambiente hypervisor host oltre all'ambiente OpenShift. Eseguendo l'implementazione direttamente su server bare metal, è inoltre possibile ridurre le limitazioni di overhead fisico derivanti dalla condivisione delle risorse tra l'host e l'ambiente OpenShift.

OpenShift su Bare Metal offre le seguenti funzionalità:

- **Distribuzione IPI o con installazione assistita** Con un cluster OpenShift distribuito tramite Installer Provisioned Infrastructure (IPI) su server bare metal, i clienti possono distribuire un ambiente OpenShift altamente versatile e facilmente scalabile direttamente su server commodity, senza la necessità di gestire un livello hypervisor.
- **Progettazione di cluster compatta** Per ridurre al minimo i requisiti hardware, OpenShift su bare metal consente agli utenti di distribuire cluster di soli 3 nodi, consentendo ai nodi del piano di controllo OpenShift di fungere anche da nodi worker e container host.
- **Virtualizzazione OpenShift** OpenShift può eseguire macchine virtuali all'interno di container utilizzando la virtualizzazione OpenShift. Questa virtualizzazione nativa del contenitore esegue l'hypervisor KVM all'interno di un contenitore e collega volumi persistenti per l'archiviazione della VM.
- **Infrastruttura ottimizzata per AI/ML** Distribuisci applicazioni come Kubeflow per applicazioni di apprendimento automatico incorporando nodi worker basati su GPU nel tuo ambiente OpenShift e sfruttando OpenShift Advanced Scheduling.

Progettazione di rete

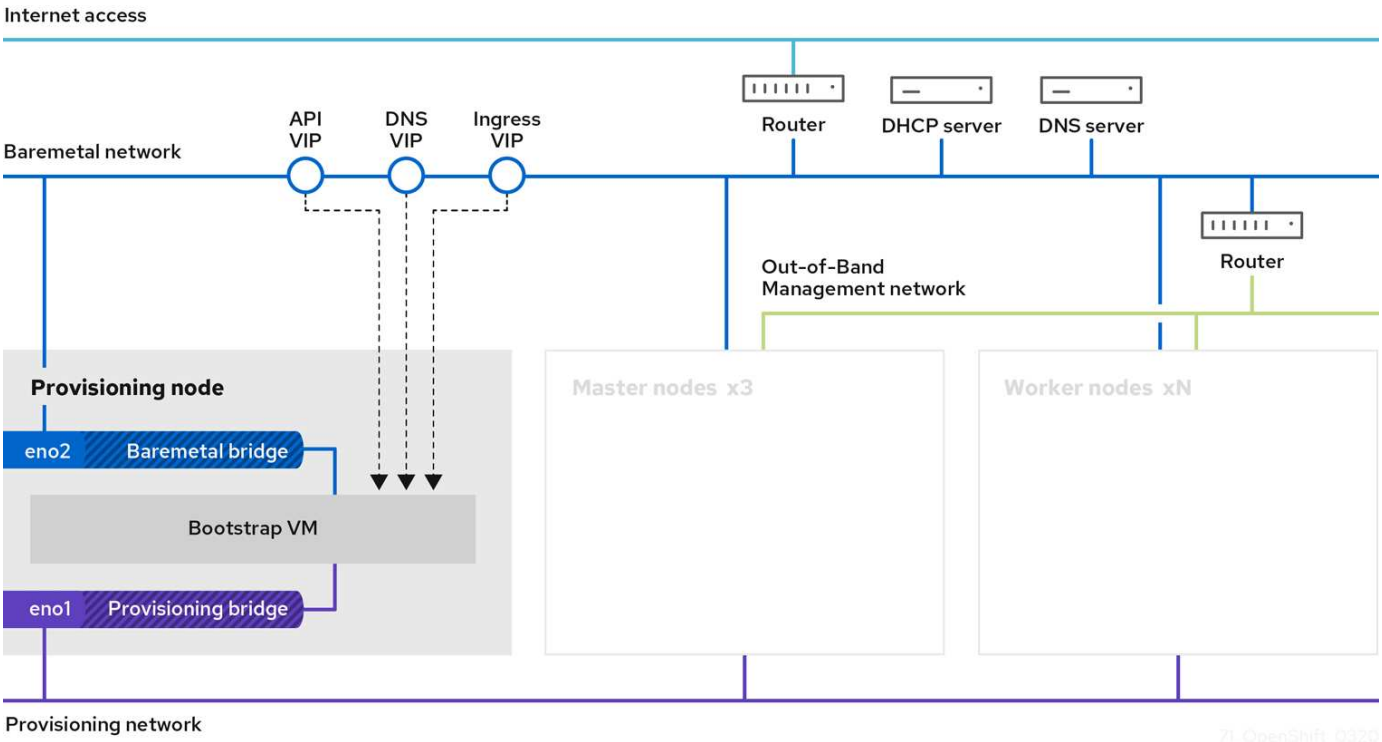
La soluzione Red Hat OpenShift su NetApp utilizza due switch dati per fornire connettività dati primaria a 25 Gbps. Utilizza inoltre due switch di gestione che forniscono connettività a 1 Gbps per la gestione in banda per i nodi di archiviazione e la gestione fuori banda per la funzionalità IPMI.

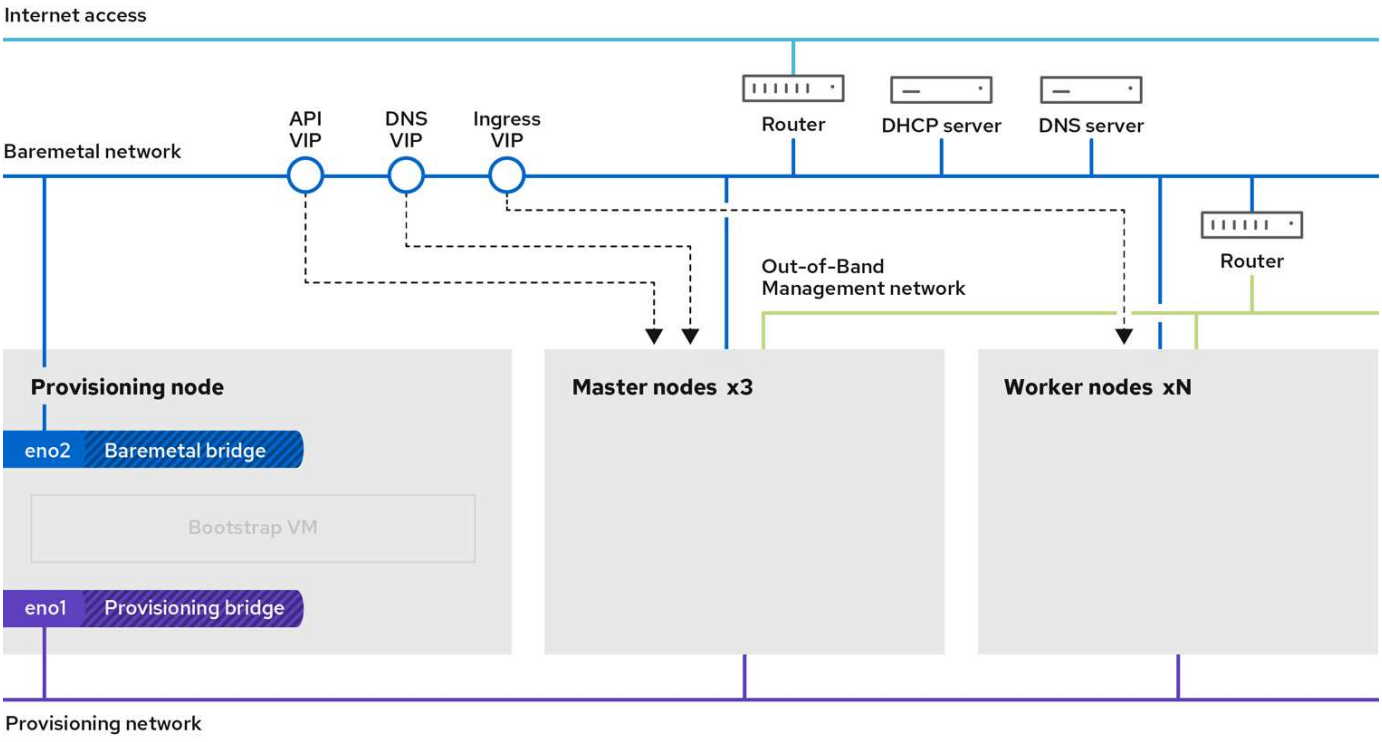
Per la distribuzione IPI bare-metal di OpenShift, è necessario creare un nodo di provisioning, una macchina Red Hat Enterprise Linux 8 che deve avere interfacce di rete collegate a reti separate.

- **Rete di provisioning** Questa rete viene utilizzata per avviare i nodi bare-metal e installare le immagini e i pacchetti necessari per distribuire il cluster OpenShift.
- **Rete bare-metal** Questa rete viene utilizzata per la comunicazione rivolta al pubblico del cluster dopo la sua distribuzione.

Per la configurazione del nodo di provisioning, il cliente crea interfacce bridge che consentono al traffico di essere instradato correttamente sul nodo stesso e sulla VM Bootstrap predisposta per scopi di distribuzione. Dopo l'implementazione del cluster, l'API e gli indirizzi VIP di ingresso vengono migrati dal nodo bootstrap al cluster appena implementato.

Le immagini seguenti illustrano l'ambiente sia durante la distribuzione IPI sia dopo il completamento della distribuzione.





Requisiti VLAN

La soluzione Red Hat OpenShift con NetApp è progettata per separare logicamente il traffico di rete per scopi diversi utilizzando reti locali virtuali (VLAN).

VLAN	Scopo	ID VLAN
Rete di gestione fuori banda	Gestione per nodi bare metal e IPMI	16
Rete bare-metal	Rete per i servizi OpenShift una volta che il cluster è disponibile	181
Rete di provisioning	Rete per l'avvio PXE e l'installazione di nodi bare metal tramite IPI	3485

Sebbene ciascuna di queste reti sia virtualmente separata da VLAN, ogni porta fisica deve essere configurata in modalità di accesso con la VLAN primaria assegnata, perché non è possibile passare un tag VLAN durante una sequenza di avvio PXE.

Risorse di supporto all'infrastruttura di rete

Prima di implementare la piattaforma container OpenShift, è necessario predisporre la seguente infrastruttura:

- Almeno un server DNS che fornisca una risoluzione completa del nome host accessibile dalla rete di gestione in banda e dalla rete VM.
- Almeno un server NTP accessibile dalla rete di gestione in banda e dalla rete VM.
- (Facoltativo) Connettività Internet in uscita sia per la rete di gestione in banda che per la rete VM.

OpenShift sulla piattaforma Red Hat OpenStack

La piattaforma Red Hat OpenStack fornisce una base integrata per creare, distribuire e scalare un cloud OpenStack privato sicuro e affidabile.

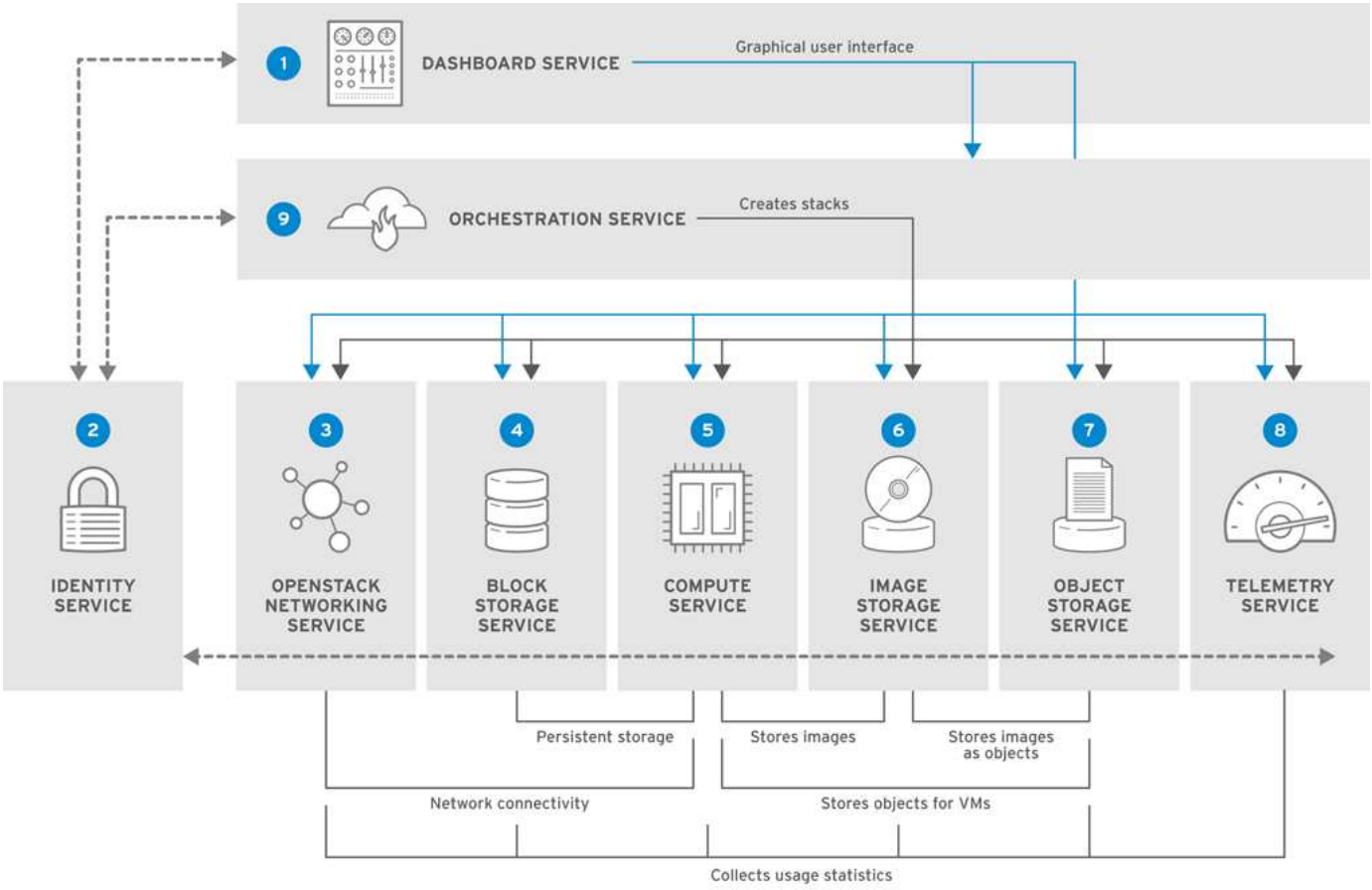
OSP è un cloud IaaS (Infrastructure-as-a-Service) implementato da una serie di servizi di controllo che gestiscono risorse di elaborazione, archiviazione e rete. L'ambiente è gestito tramite un'interfaccia basata sul Web che consente agli amministratori e agli utenti di controllare, fornire e automatizzare le risorse OpenStack. Inoltre, l'infrastruttura OpenStack è facilitata da un'ampia interfaccia a riga di comando e da un'API che consente funzionalità di automazione complete per amministratori e utenti finali.

Il progetto OpenStack è un progetto comunitario in rapido sviluppo che fornisce versioni aggiornate ogni sei mesi. Inizialmente Red Hat OpenStack Platform ha tenuto il passo con questo ciclo di rilascio pubblicando una nuova versione insieme a ogni versione upstream e fornendo supporto a lungo termine per ogni terza versione. Di recente, con la versione OSP 16.0 (basata su OpenStack Train), Red Hat ha scelto di non tenere il passo con i numeri delle versioni, ma ha invece retroportato le nuove funzionalità in versioni secondarie. La versione più recente è Red Hat OpenStack Platform 16.1, che include funzionalità avanzate backported dalle versioni upstream Ussuri e Victoria.

Per maggiori informazioni su OSP vedere"Sito web della piattaforma Red Hat OpenStack" .

Servizi OpenStack

I servizi della piattaforma OpenStack vengono distribuiti come contenitori, il che isola i servizi l'uno dall'altro e consente facili aggiornamenti. La piattaforma OpenStack utilizza un set di container creati e gestiti con Kolla. La distribuzione dei servizi viene eseguita estraendo le immagini dei container dal Red Hat Custom Portal. Questi contenitori di servizi vengono gestiti tramite il comando Podman e vengono distribuiti, configurati e mantenuti con Red Hat OpenStack Director.



Servizio	Nome del progetto	Descrizione
Pannello di controllo	Orizzonte	Dashboard basata su browser Web utilizzata per gestire i servizi OpenStack.
Identità	Keystone	Servizio centralizzato per l'autenticazione e l'autorizzazione dei servizi OpenStack e per la gestione di utenti, progetti e ruoli.
Rete OpenStack	Neutrone	Fornisce connettività tra le interfacce dei servizi OpenStack.
Archiviazione a blocchi	Cenere	Gestisce volumi di archiviazione a blocchi persistenti per macchine virtuali (VM).
Calcolare	Nuova	Gestisce e fornisce VM in esecuzione sui nodi di elaborazione.
Immagine	Occhiata	Servizio di registro utilizzato per archiviare risorse quali immagini di macchine virtuali e snapshot di volumi.
Archiviazione di oggetti	Veloce	Consente agli utenti di archiviare e recuperare file e dati arbitrari.
Telemetria	Ceilometro	Fornisce misurazioni dell'utilizzo delle risorse cloud.
Orchestrazione	Calore	Motore di orchestrazione basato su modelli che supporta la creazione automatica di stack di risorse.

Progettazione di rete

La soluzione Red Hat OpenShift con NetApp utilizza due switch dati per fornire connettività dati primaria a 25 Gbps. Utilizza inoltre due switch di gestione aggiuntivi che forniscono connettività a 1 Gbps per la gestione in banda per i nodi di archiviazione e la gestione fuori banda per la funzionalità IPMI.

La funzionalità IPMI è richiesta da Red Hat OpenStack Director per distribuire Red Hat OpenStack Platform utilizzando il servizio di provisioning bare-metal Ironic.

Requisiti VLAN

Red Hat OpenShift con NetApp è progettato per separare logicamente il traffico di rete per scopi diversi utilizzando reti locali virtuali (VLAN). Questa configurazione può essere adattata alle esigenze dei clienti o per garantire un ulteriore isolamento per specifici servizi di rete. Nella tabella seguente sono elencate le VLAN necessarie per implementare la soluzione durante la convalida della soluzione in NetApp.

VLAN	Scopo	ID VLAN
Rete di gestione fuori banda	Rete utilizzata per la gestione dei nodi fisici e del servizio IPMI per Ironic.	16
Infrastruttura di archiviazione	Rete utilizzata dai nodi controller per mappare direttamente i volumi per supportare servizi infrastrutturali come Swift.	201
Cenere di stoccaggio	Rete utilizzata per mappare e collegare volumi di blocchi direttamente alle istanze virtuali distribuite nell'ambiente.	202
API interna	Rete utilizzata per la comunicazione tra i servizi OpenStack mediante comunicazione API, messaggi RPC e comunicazione database.	301
Inquilino	Neutron fornisce a ciascun tenant le proprie reti tramite tunneling tramite VXLAN. Il traffico di rete è isolato all'interno di ogni rete tenant. A ciascuna rete tenant è associata una subnet IP e gli spazi dei nomi di rete consentono a più reti tenant di utilizzare lo stesso intervallo di indirizzi senza causare conflitti.	302
Gestione dello storage	OpenStack Object Storage (Swift) utilizza questa rete per sincronizzare gli oggetti dati tra i nodi replica partecipanti. Il servizio proxy funge da interfaccia intermedia tra le richieste degli utenti e il livello di archiviazione sottostante. Il proxy riceve le richieste in arrivo e individua la replica necessaria per recuperare i dati richiesti.	303
PXE	OpenStack Director fornisce l'avvio PXE come parte del servizio di provisioning bare metal di Ironic per orchestrare l'installazione di OSP Overcloud.	3484
Esterno	Rete disponibile al pubblico che ospita l'OpenStack Dashboard (Horizon) per la gestione grafica e consente chiamate API pubbliche per gestire i servizi OpenStack.	3485
Rete di gestione in banda	Fornisce l'accesso alle funzioni di amministrazione del sistema, quali l'accesso SSH, il traffico DNS e il traffico Network Time Protocol (NTP). Questa rete funge anche da gateway per i nodi non controller.	3486

Risorse di supporto all'infrastruttura di rete

Prima di implementare OpenShift Container Platform, è necessario predisporre la seguente infrastruttura:

- Almeno un server DNS che fornisca una risoluzione completa dei nomi host.
- Almeno tre server NTP in grado di mantenere sincronizzato l'orario per i server nella soluzione.
- (Facoltativo) Connettività Internet in uscita per l'ambiente OpenShift.

Best practice per le distribuzioni di produzione

In questa sezione sono elencate alcune best practice che un'organizzazione dovrebbe prendere in considerazione prima di implementare questa soluzione in produzione.

Distribuisci OpenShift su un cloud privato OSP con almeno tre nodi di elaborazione

L'architettura verificata descritta in questo documento presenta la distribuzione hardware minima adatta per le operazioni HA mediante la distribuzione di tre nodi controller OSP e due nodi di elaborazione OSP. Questa architettura garantisce una configurazione fault tolerant in cui entrambi i nodi di elaborazione possono avviare istanze virtuali e le VM distribuite possono migrare tra i due hypervisor.

Poiché Red Hat OpenShift inizialmente viene distribuito con tre nodi master, una configurazione a due nodi potrebbe far sì che almeno due master occupino lo stesso nodo, il che può comportare una possibile interruzione di OpenShift se quel nodo specifico diventa non disponibile. Pertanto, una delle best practice di Red Hat è quella di distribuire almeno tre nodi di elaborazione OSP in modo che i master OpenShift possano essere distribuiti uniformemente e la soluzione riceva un ulteriore grado di tolleranza agli errori.

Configurare l'affinità macchina virtuale/host

È possibile distribuire i master OpenShift su più nodi hypervisor abilitando l'affinità VM/host.

L'affinità è un modo per definire regole per un set di VM e/o host che determinano se le VM vengono eseguite insieme sullo stesso host o sugli stessi host del gruppo oppure su host diversi. Viene applicato alle VM creando gruppi di affinità costituiti da VM e/o host con un set di parametri e condizioni identici. A seconda che le VM in un gruppo di affinità vengano eseguite sullo stesso host o sugli stessi host del gruppo oppure separatamente su host diversi, i parametri del gruppo di affinità possono definire un'affinità positiva o negativa. Nella piattaforma Red Hat OpenStack, le regole di affinità e anti-affinità host possono essere create e applicate creando gruppi di server e configurando filtri in modo che le istanze distribuite da Nova in un gruppo di server vengano distribuite su nodi di elaborazione diversi.

Un gruppo di server ha un massimo predefinito di 10 istanze virtuali per le quali può gestire il posizionamento. È possibile modificare questa impostazione aggiornando le quote predefinite per Nova.

> Esiste un limite specifico di affinità/anti-affinità per i gruppi di server OSP; se non ci sono risorse sufficienti per la distribuzione su nodi separati o per consentire la condivisione dei nodi, la VM non riesce ad avviarsi.

Per configurare i gruppi di affinità, vedere"Come posso configurare Affinity e Anti-Affinity per le istanze OpenStack?" .

Utilizzare un file di installazione personalizzato per la distribuzione di OpenShift

IPI semplifica l'implementazione dei cluster OpenShift tramite la procedura guidata interattiva descritta in precedenza in questo documento. Tuttavia, è possibile che sia necessario modificare alcuni valori predefiniti come parte di una distribuzione del cluster.

In questi casi, è possibile eseguire e assegnare attività alla procedura guidata senza distribuire immediatamente un cluster; al contrario, viene creato un file di configurazione da cui è possibile distribuire il cluster in un secondo momento. Questa funzionalità è molto utile se è necessario modificare i valori predefiniti

dell'IPI o se si desidera distribuire più cluster identici nel proprio ambiente per altri usi, ad esempio il multi-tenancy. Per ulteriori informazioni sulla creazione di una configurazione di installazione personalizzata per OpenShift, vedere "Red Hat OpenShift Installazione di un cluster su OpenStack con personalizzazioni".

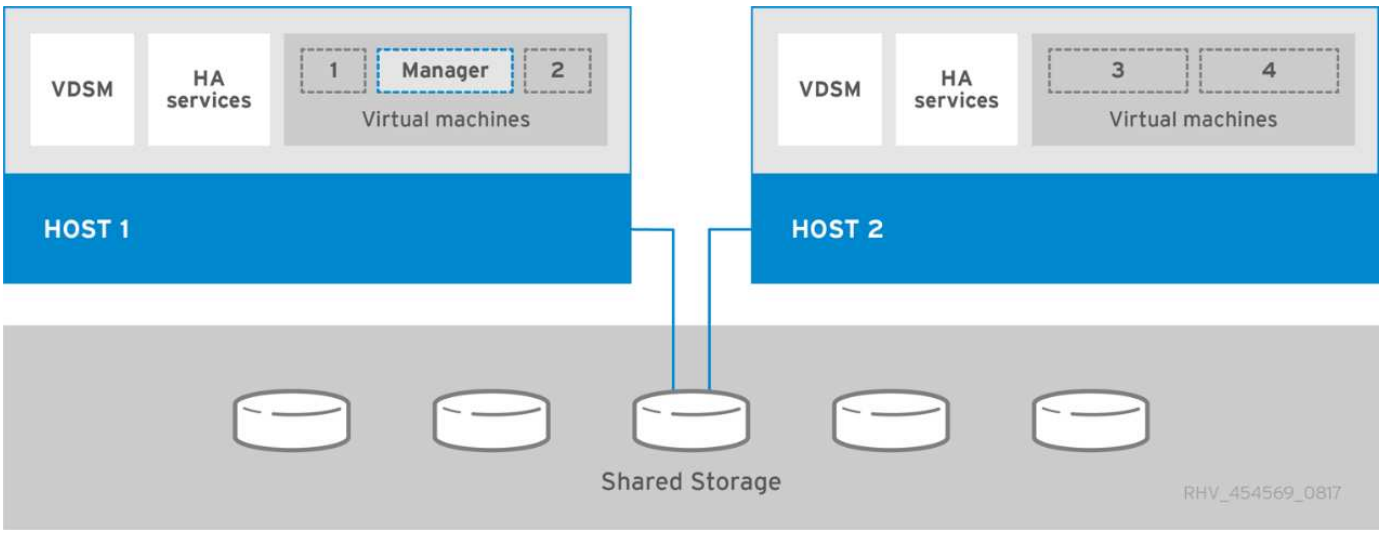
OpenShift su Red Hat Virtualization

Red Hat Virtualization (RHV) è una piattaforma di data center virtuale aziendale che funziona su Red Hat Enterprise Linux (RHEL) e utilizza l'hypervisor KVM.

Per maggiori informazioni su RHV, vedere"Sito web di Red Hat Virtualization" .

RHV offre le seguenti funzionalità:

- **Gestione centralizzata di VM e host** Il gestore RHV viene eseguito come macchina fisica o virtuale (VM) nella distribuzione e fornisce un'interfaccia utente grafica basata sul Web per la gestione della soluzione da un'interfaccia centrale.
- **Motore self-hosted** Per ridurre al minimo i requisiti hardware, RHV consente di distribuire RHV Manager (RHV-M) come VM sugli stessi host che eseguono le VM guest.
- **Alta disponibilità** Per evitare interruzioni in caso di guasti dell'host, RHV consente di configurare le VM per un'elevata disponibilità. Le VM ad alta disponibilità sono controllate a livello di cluster mediante criteri di resilienza.
- **Elevata scalabilità** Un singolo cluster RHV può avere fino a 200 host hypervisor, consentendogli di supportare i requisiti di VM di grandi dimensioni per ospitare carichi di lavoro di classe enterprise ad alto consumo di risorse.
- **Sicurezza avanzata** Ereditate da RHV, le tecnologie Secure Virtualization (sVirt) e Security Enhanced Linux (SELinux) vengono impiegate da RHV per garantire una maggiore sicurezza e un rafforzamento degli host e delle VM. Il vantaggio principale di queste funzionalità è l'isolamento logico di una VM e delle risorse ad essa associate.



Progettazione di rete

La soluzione Red Hat OpenShift su NetApp utilizza due switch dati per fornire connettività dati primaria a 25 Gbps. Utilizza inoltre due switch di gestione aggiuntivi che forniscono connettività a 1 Gbps per la gestione in banda dei nodi di archiviazione e la gestione fuori banda per la funzionalità IPMI. OCP utilizza la rete logica della macchina virtuale su RHV per la gestione del cluster. Questa sezione descrive la disposizione e lo scopo di ciascun segmento di rete virtuale utilizzato nella soluzione e delinea i prerequisiti per l'implementazione

della soluzione.

Requisiti VLAN

Red Hat OpenShift su RHV è progettato per separare logicamente il traffico di rete per scopi diversi utilizzando reti locali virtuali (VLAN). Questa configurazione può essere adattata alle esigenze dei clienti o per garantire un ulteriore isolamento per specifici servizi di rete. Nella tabella seguente sono elencate le VLAN necessarie per implementare la soluzione durante la convalida della soluzione in NetApp.

VLAN	Scopo	ID VLAN
Rete di gestione fuori banda	Gestione per nodi fisici e IPMI	16
Rete VM	Accesso alla rete virtuale per gli ospiti	1172
Rete di gestione in banda	Gestione per nodi RHV-H, RHV-Manager e rete ovirtmgmt	3343
Rete di archiviazione	Rete di archiviazione per NetApp Element iSCSI	3344
Rete di migrazione	Rete per la migrazione degli ospiti virtuali	3345

Risorse di supporto all'infrastruttura di rete

Prima di implementare OpenShift Container Platform, è necessario predisporre la seguente infrastruttura:

- Almeno un server DNS che fornisca la risoluzione completa dei nomi host, accessibile dalla rete di gestione in banda e dalla rete VM.
- Almeno un server NTP accessibile dalla rete di gestione in banda e dalla rete VM.
- (Facoltativo) Connettività Internet in uscita sia per la rete di gestione in banda che per la rete VM.

Best practice per le distribuzioni di produzione

In questa sezione sono elencate alcune best practice che un'organizzazione dovrebbe prendere in considerazione prima di implementare questa soluzione in produzione.

Distribuisci OpenShift su un cluster RHV di almeno tre nodi

L'architettura verificata descritta in questo documento presenta la distribuzione hardware minima adatta per le operazioni HA distribuendo due nodi hypervisor RHV-H e garantendo una configurazione fault tolerant in cui entrambi gli host possono gestire il motore ospitato e le VM distribuite possono migrare tra i due hypervisor.

Poiché Red Hat OpenShift inizialmente viene distribuito con tre nodi master, in una configurazione a due nodi è garantito che almeno due master occuperanno lo stesso nodo, il che può comportare una possibile interruzione di OpenShift se quel nodo specifico diventa non disponibile. Pertanto, una delle best practice di Red Hat è quella di implementare almeno tre nodi hypervisor RHV-H come parte della soluzione, in modo che i master OpenShift possano essere distribuiti uniformemente e la soluzione riceva un ulteriore grado di tolleranza agli errori.

Configurare l'affinità macchina virtuale/host

È possibile distribuire i master OpenShift su più nodi hypervisor abilitando l'affinità VM/host.

L'affinità è un modo per definire regole per un set di VM e/o host che determinano se le VM vengono eseguite insieme sullo stesso host o sugli stessi host del gruppo oppure su host diversi. Viene applicato alle VM

creando gruppi di affinità costituiti da VM e/o host con un set di parametri e condizioni identici. A seconda che le VM in un gruppo di affinità vengano eseguite sullo stesso host o sugli stessi host del gruppo oppure separatamente su host diversi, i parametri del gruppo di affinità possono definire un'affinità positiva o negativa.

Le condizioni definite per i parametri possono essere di tipo hard enforcement o soft enforcement. L'applicazione rigida garantisce che le VM in un gruppo di affinità seguano sempre rigorosamente l'affinità positiva o negativa, senza tenere conto di condizioni esterne. L'applicazione soft garantisce che venga impostata una preferenza più elevata per le VM in un gruppo di affinità, in modo che seguano l'affinità positiva o negativa, ove possibile. Nella configurazione a due o tre hypervisor descritta in questo documento, l'impostazione consigliata è l'affinità soft. Nei cluster più grandi, l'affinità rigida può distribuire correttamente i nodi OpenShift.

Per configurare i gruppi di affinità, vedere"Red Hat 6.11. Documentazione dei gruppi di affinità" .

Utilizzare un file di installazione personalizzato per la distribuzione di OpenShift

IPI semplifica l'implementazione dei cluster OpenShift tramite la procedura guidata interattiva descritta in precedenza in questo documento. Tuttavia, è possibile che alcuni valori predefiniti debbano essere modificati durante la distribuzione del cluster.

In questi casi, è possibile eseguire e assegnare attività alla procedura guidata senza dover distribuire immediatamente un cluster. Piuttosto, viene creato un file di configurazione dal quale il cluster può essere distribuito in un secondo momento. Questa funzionalità è molto utile se si desidera modificare i valori predefiniti IPI o se si desidera distribuire più cluster identici nel proprio ambiente per altri usi, ad esempio il multi-tenancy. Per ulteriori informazioni sulla creazione di una configurazione di installazione personalizzata per OpenShift, vedere"Red Hat OpenShift Installazione di un cluster su RHV con personalizzazioni" .

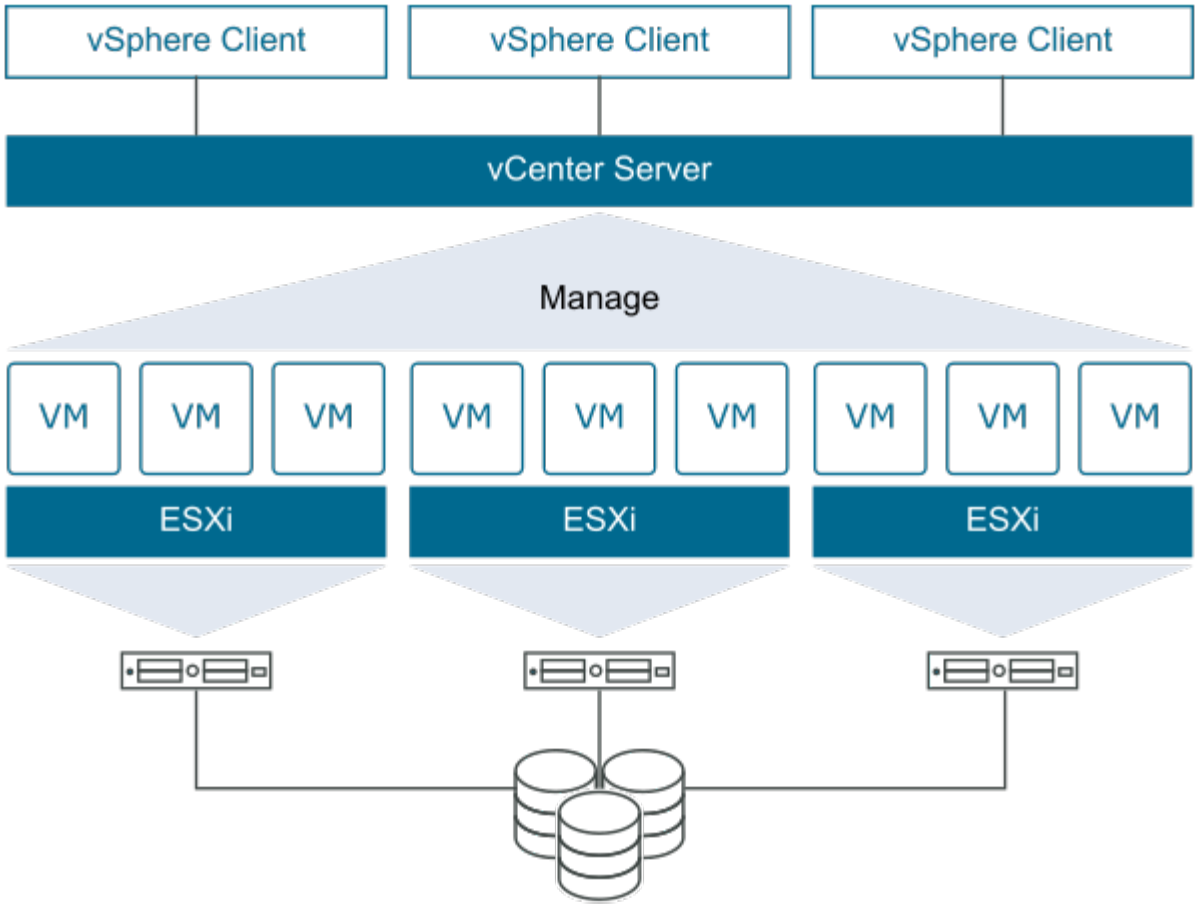
OpenShift su VMware vSphere

VMware vSphere è una piattaforma di virtualizzazione per la gestione centralizzata di un gran numero di server e reti virtualizzati in esecuzione sull'hypervisor ESXi.

Per ulteriori informazioni su VMware vSphere, vedere"Sito web VMware vSphere" .

VMware vSphere offre le seguenti funzionalità:

- **VMware vCenter Server** VMware vCenter Server fornisce una gestione unificata di tutti gli host e le VM da un'unica console e aggrega il monitoraggio delle prestazioni di cluster, host e VM.
- **VMware vSphere vMotion** VMware vCenter consente di migrare a caldo le VM tra i nodi del cluster su richiesta, senza interruzioni.
- **vSphere High Availability** Per evitare interruzioni in caso di guasti dell'host, VMware vSphere consente di raggruppare gli host e configurarli per l'alta disponibilità. Le VM interrotte da un errore dell'host vengono riavviate a breve sugli altri host del cluster, ripristinando i servizi.
- **Distributed Resource Scheduler (DRS)** Un cluster VMware vSphere può essere configurato per bilanciare il carico delle risorse necessarie alle VM che ospita. Le VM con conflitti di risorse possono essere migrate a caldo su altri nodi del cluster per garantire che siano disponibili risorse sufficienti.



Progettazione di rete

La soluzione Red Hat OpenShift su NetApp utilizza due switch dati per fornire connettività dati primaria a 25 Gbps. Utilizza inoltre due switch di gestione aggiuntivi che forniscono connettività a 1 Gbps per la gestione in banda per i nodi di archiviazione e la gestione fuori banda per la funzionalità IPMI. OCP utilizza la rete logica VM su VMware vSphere per la gestione del cluster. Questa sezione descrive la disposizione e lo scopo di ciascun segmento di rete virtuale utilizzato nella soluzione e delinea i prerequisiti per l'implementazione della soluzione.

Requisiti VLAN

Red Hat OpenShift su VMware vSphere è progettato per separare logicamente il traffico di rete per scopi diversi utilizzando reti locali virtuali (VLAN). Questa configurazione può essere adattata alle esigenze dei clienti o per garantire un ulteriore isolamento per specifici servizi di rete. Nella tabella seguente sono elencate le VLAN necessarie per implementare la soluzione durante la convalida della soluzione in NetApp.

VLAN	Scopo	ID VLAN
Rete di gestione fuori banda	Gestione per nodi fisici e IPMI	16
Rete VM	Accesso alla rete virtuale per gli ospiti	181
Rete di archiviazione	Rete di archiviazione per ONTAP NFS	184
Rete di archiviazione	Rete di archiviazione per ONTAP iSCSI	185
Rete di gestione in banda	Gestione per nodi ESXi, VCenter Server, ONTAP Select	3480

VLAN	Scopo	ID VLAN
Rete di archiviazione	Rete di archiviazione per NetApp Element iSCSI	3481
Rete di migrazione	Rete per la migrazione degli ospiti virtuali	3482

Risorse di supporto all'infrastruttura di rete

Prima di implementare OpenShift Container Platform, è necessario predisporre la seguente infrastruttura:

- Almeno un server DNS che fornisca la risoluzione completa dei nomi host, accessibile dalla rete di gestione in banda e dalla rete VM.
- Almeno un server NTP accessibile dalla rete di gestione in banda e dalla rete VM.
- (Facoltativo) Connettività Internet in uscita sia per la rete di gestione in banda che per la rete VM.

Best practice per le distribuzioni di produzione

In questa sezione sono elencate alcune best practice che un'organizzazione dovrebbe prendere in considerazione prima di implementare questa soluzione in produzione.

Distribuisci OpenShift su un cluster ESXi di almeno tre nodi

L'architettura verificata descritta in questo documento presenta la distribuzione hardware minima adatta per le operazioni HA distribuendo due nodi hypervisor ESXi e garantendo una configurazione a tolleranza di errore tramite l'abilitazione di VMware vSphere HA e VMware vMotion. Questa configurazione consente alle VM distribuite di migrare tra i due hypervisor e di riavviarsi nel caso in cui un host non sia disponibile.

Poiché Red Hat OpenShift inizialmente viene distribuito con tre nodi master, in alcune circostanze almeno due master in una configurazione a due nodi possono occupare lo stesso nodo, il che può comportare una possibile interruzione di OpenShift se quel nodo specifico diventa non disponibile. Pertanto, una delle best practice di Red Hat è quella di implementare almeno tre nodi hypervisor ESXi in modo che i master OpenShift possano essere distribuiti in modo uniforme, il che garantisce un ulteriore grado di tolleranza agli errori.

Configurare l'affinità tra macchina virtuale e host

È possibile garantire la distribuzione dei master OpenShift su più nodi hypervisor abilitando l'affinità tra VM e host.

L'affinità o l'anti-affinità è un modo per definire regole per un set di VM e/o host che determinano se le VM vengono eseguite insieme sullo stesso host o sugli stessi host del gruppo oppure su host diversi. Viene applicato alle VM creando gruppi di affinità costituiti da VM e/o host con un set di parametri e condizioni identici. A seconda che le VM in un gruppo di affinità vengano eseguite sullo stesso host o sugli stessi host del gruppo oppure separatamente su host diversi, i parametri del gruppo di affinità possono definire un'affinità positiva o negativa.

Per configurare i gruppi di affinità, vedere "Documentazione di vSphere 9.0: utilizzo delle regole di affinità DRS".

Utilizzare un file di installazione personalizzato per la distribuzione di OpenShift

IPI semplifica l'implementazione dei cluster OpenShift tramite la procedura guidata interattiva descritta in precedenza in questo documento. Tuttavia, è possibile che sia necessario modificare alcuni valori predefiniti come parte di una distribuzione del cluster.

In questi casi, è possibile eseguire e assegnare attività alla procedura guidata senza distribuire immediatamente un cluster; la procedura guidata crea invece un file di configurazione da cui è possibile distribuire il cluster in un secondo momento. Questa funzionalità è molto utile se è necessario modificare i valori predefiniti IPI o se si desidera distribuire più cluster identici nel proprio ambiente per altri usi, ad esempio il multi-tenancy. Per ulteriori informazioni sulla creazione di una configurazione di installazione personalizzata per OpenShift, vedere"Red Hat OpenShift Installazione di un cluster su vSphere con personalizzazioni" .

Servizio Red Hat OpenShift su AWS

Red Hat OpenShift Service on AWS (ROSA) è un servizio gestito che puoi utilizzare per creare, scalare e distribuire applicazioni containerizzate con la piattaforma Kubernetes aziendale Red Hat OpenShift su AWS. ROSA semplifica lo spostamento dei carichi di lavoro Red Hat OpenShift on-premise su AWS e offre una stretta integrazione con altri servizi AWS.

Per maggiori informazioni su ROSA, consultare la documentazione qui:"Red Hat OpenShift Service su AWS (documentazione AWS)" . "Red Hat OpenShift Service su AWS (documentazione Red Hat)" .

Sistemi di archiviazione NetApp

NetApp ONTAP

NetApp ONTAP è un potente strumento software di storage con funzionalità quali un'interfaccia utente grafica intuitiva, API REST con integrazione dell'automazione, analisi predittiva basata sull'intelligenza artificiale e azioni correttive, aggiornamenti hardware senza interruzioni e importazione tra storage.

Per ulteriori informazioni sul sistema di storage NetApp ONTAP , visitare il sito "Sito web NetApp ONTAP" .

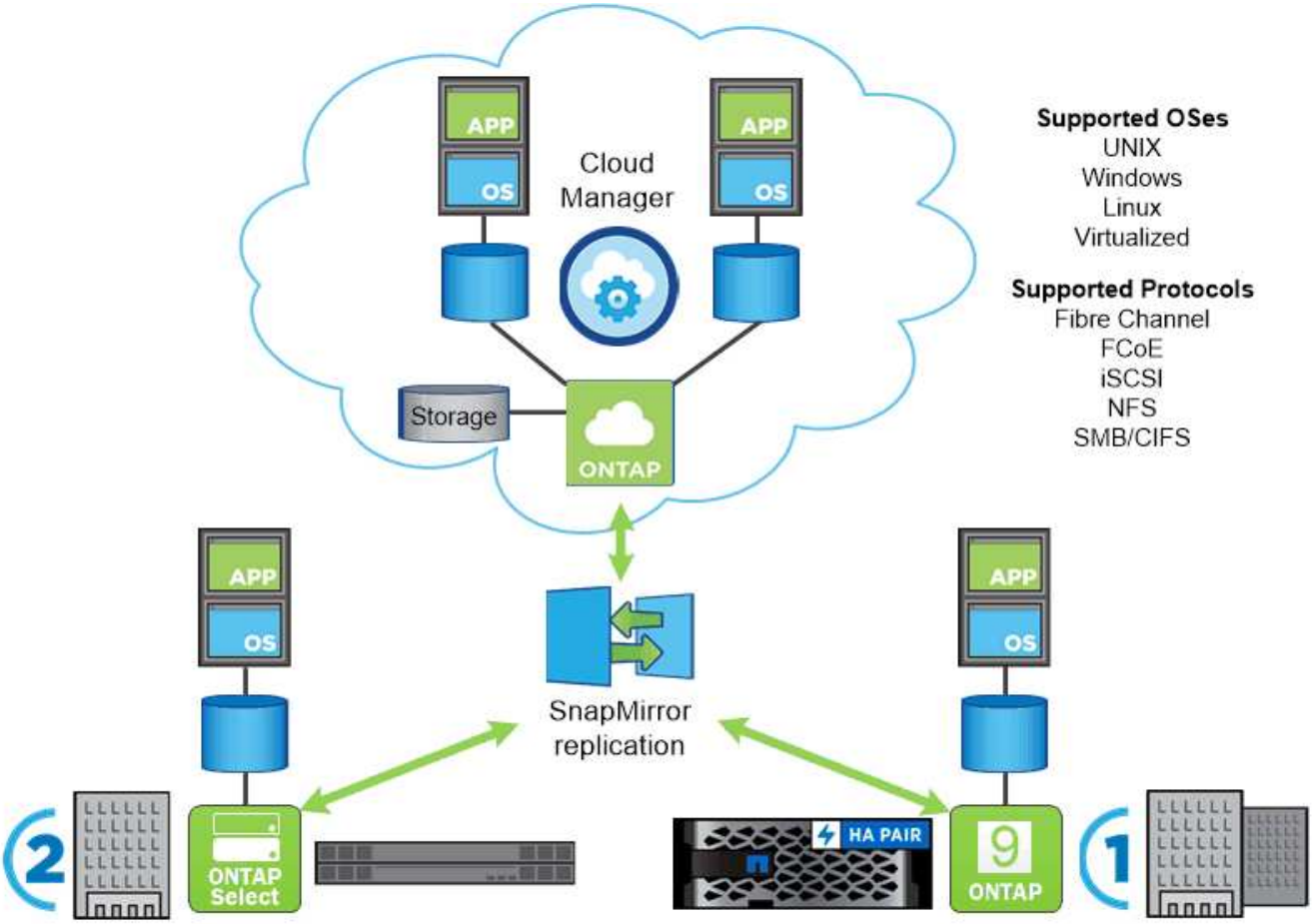
ONTAP offre le seguenti funzionalità:

- Un sistema di archiviazione unificato con accesso simultaneo ai dati e gestione dei protocolli NFS, CIFS, iSCSI, FC, FCoE e FC-NVMe.
- I diversi modelli di distribuzione includono configurazioni hardware on-premise all-flash, ibride e all-HDD; piattaforme di storage basate su VM su un hypervisor supportato come ONTAP Select; e nel cloud come Cloud Volumes ONTAP.
- Maggiore efficienza di archiviazione dei dati sui sistemi ONTAP con supporto per la suddivisione automatica dei dati in livelli, la compressione dei dati in linea, la deduplicazione e la compattazione.
- Archiviazione basata sul carico di lavoro e controllata dalla qualità del servizio.
- Integrazione perfetta con un cloud pubblico per la suddivisione in livelli e la protezione dei dati. ONTAP offre inoltre solide funzionalità di protezione dei dati che lo distinguono in qualsiasi ambiente:
  - * Copie Snapshot NetApp .* Un backup rapido e puntuale dei dati utilizzando una quantità minima di spazio su disco, senza alcun sovraccarico aggiuntivo delle prestazioni.
  - * NetApp SnapMirror.* Esegue il mirroring delle copie Snapshot dei dati da un sistema di archiviazione a un altro. ONTAP supporta anche il mirroring dei dati su altre piattaforme fisiche e servizi cloud-native.
  - * NetApp SnapLock.* Gestione efficiente dei dati non riscrivibili mediante scrittura su volumi speciali che non possono essere sovrascritti o cancellati per un periodo di tempo designato.

- * NetApp SnapVault.* Esegue il backup dei dati da più sistemi di archiviazione su una copia Snapshot centrale che funge da backup per tutti i sistemi designati.
- * NetApp SyncMirror.* Fornisce il mirroring dei dati in tempo reale a livello RAID su due diversi plex di dischi collegati fisicamente allo stesso controller.
- * NetApp SnapRestore.* Fornisce un rapido ripristino dei dati sottoposti a backup su richiesta dalle copie Snapshot.
- * NetApp FlexClone.* Fornisce il provisioning istantaneo di una copia completamente leggibile e scrivibile di un volume NetApp basata su una copia Snapshot.

Per maggiori informazioni su ONTAP, vedere "Centro di documentazione ONTAP 9" .

NetApp ONTAP è disponibile in locale, virtualizzato o nel cloud.



Piattaforme NetApp

NetApp AFF/ FAS

NetApp fornisce piattaforme di storage all-flash (AFF) e ibride scale-out (FAS) robuste, realizzate su misura con prestazioni a bassa latenza, protezione dei dati integrata e supporto multiprotocollo.

Entrambi i sistemi sono basati sul software di gestione dati NetApp ONTAP , il software di gestione dati più avanzato del settore per una gestione dello storage semplificata, integrata nel cloud e ad alta disponibilità, per garantire la velocità, l'efficienza e la sicurezza di livello aziendale di cui il tuo data fabric ha bisogno.

Per ulteriori informazioni sulle piattaforme NETAPP AFF/ FAS , fare clic su "Qui" .

ONTAP Select

ONTAP Select è una distribuzione software-defined di NetApp ONTAP che può essere distribuita su un hypervisor nel tuo ambiente. Può essere installato su VMware vSphere o su KVM e fornisce tutte le funzionalità e l'esperienza di un sistema ONTAP basato su hardware.

Per maggiori informazioni su ONTAP Select, clicca "Qui" .

Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP è una versione distribuita nel cloud di NetApp ONTAP , disponibile per l'implementazione in numerosi cloud pubblici, tra cui: Amazon AWS, Microsoft Azure e Google Cloud.

Per ulteriori informazioni su Cloud Volumes ONTAP, fare clic su "Qui" .

Amazon FSx ONTAP

Amazon FSx ONTAP fornisce storage condiviso completamente gestito nel cloud AWS con le diffuse funzionalità di gestione e accesso ai dati di ONTAP. Per ulteriori informazioni su Amazon FSx ONTAP, fare clic su "Qui" .

Azure NetApp Files

Azure NetApp Files è un servizio di archiviazione file nativo di Azure, di prima parte, di classe enterprise e ad alte prestazioni. Fornisce volumi come servizio per il quale è possibile creare account NetApp , pool di capacità e volumi. È inoltre possibile selezionare i livelli di servizio e di prestazione e gestire la protezione dei dati. Puoi creare e gestire condivisioni di file ad alte prestazioni, altamente disponibili e scalabili utilizzando gli stessi protocolli e strumenti che conosci e su cui fai affidamento in locale. Per ulteriori informazioni su Azure NetApp Files, fare clic su "Qui" .

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes è un servizio di archiviazione dati basato su cloud completamente gestito che offre funzionalità avanzate di gestione dei dati e prestazioni altamente scalabili. Consente di spostare le applicazioni basate su file su Google Cloud. Supporta i protocolli Network File System (NFSv3 e NFSv4.1) e Server Message Block (SMB) integrati, quindi non è necessario riprogettare le applicazioni e si può continuare a ottenere uno storage persistente per le applicazioni. Per ulteriori informazioni su Google Cloud NetApp VolumesP, fare clic su "Qui" .

NetApp Element: Red Hat OpenShift con NetApp

Il software NetApp Element offre prestazioni modulari e scalabili, con ogni nodo di storage che fornisce capacità e throughput garantiti all'ambiente. I sistemi NetApp Element possono essere scalati da 4 a 100 nodi in un singolo cluster e offrono una serie di funzionalità avanzate di gestione dello storage.

Per ulteriori informazioni sui sistemi di storage NetApp Element , visitare il sito "Sito web NetApp Solidfire" .

Reindirizzamento dell'accesso iSCSI e capacità di auto-riparazione

Il software NetApp Element sfrutta il protocollo di archiviazione iSCSI, un metodo standard per incapsulare i comandi SCSI su una rete TCP/IP tradizionale. Quando cambiano gli standard SCSI o quando migliorano le prestazioni delle reti Ethernet, il protocollo di archiviazione iSCSI ne trae vantaggio senza dover apportare alcuna modifica.

Sebbene tutti i nodi di storage abbiano un IP di gestione e un IP di storage, il software NetApp Element pubblicizza un singolo indirizzo IP virtuale di storage (indirizzo SVIP) per tutto il traffico di storage nel cluster. Come parte del processo di accesso iSCSI, l'archiviazione può rispondere che il volume di destinazione è stato spostato a un indirizzo diverso e pertanto non può procedere con il processo di negoziazione. L'host quindi invia nuovamente la richiesta di accesso al nuovo indirizzo in un processo che non richiede alcuna riconfigurazione lato host. Questo processo è noto come reindirizzamento dell'accesso iSCSI.

Il reindirizzamento dell'accesso iSCSI è una parte fondamentale del cluster software NetApp Element . Quando viene ricevuta una richiesta di accesso all'host, il nodo decide quale membro del cluster deve gestire il traffico in base agli IOPS e ai requisiti di capacità per il volume. I volumi vengono distribuiti nel cluster software NetApp Element e vengono ridistribuiti se un singolo nodo gestisce troppo traffico per i suoi volumi o se viene aggiunto un nuovo nodo. Vengono allocate più copie di un dato volume nell'array.

In questo modo, se un errore del nodo è seguito da una ridistribuzione del volume, non vi è alcun effetto sulla connettività dell'host oltre a un logout e un login con reindirizzamento alla nuova posizione. Grazie al reindirizzamento degli accessi iSCSI, un cluster software NetApp Element è un'architettura scalabile e auto-riparante, in grado di eseguire aggiornamenti e operazioni senza interruzioni.

Cluster software NetApp Element QoS

Un cluster software NetApp Element consente di configurare dinamicamente la qualità del servizio in base al volume. È possibile utilizzare le impostazioni QoS per volume per controllare le prestazioni di archiviazione in base agli SLA definiti. I seguenti tre parametri configurabili definiscono la QoS:

- **IOPS minimi.** Numero minimo di IOPS sostenuti che il cluster software NetApp Element fornisce a un volume. Il numero minimo di IOPS configurato per un volume rappresenta il livello di prestazioni garantito per un volume. Le prestazioni per volume non scendono al di sotto di questo livello.
- **IOPS massimi.** Numero massimo di IOPS sostenuti che il cluster software NetApp Element fornisce a un volume specifico.
- **IOPS a raffica.** Numero massimo di IOPS consentito in uno scenario di burst breve. La durata della raffica è configurabile, con un valore predefinito di 1 minuto. Se un volume è stato eseguito al di sotto del livello massimo di IOPS, vengono accumulati crediti burst. Quando i livelli di prestazioni diventano molto elevati e vengono spinti al limite, sul volume sono consentiti brevi raffiche di IOPS oltre il massimo IOPS.

Multi-tenancy

La multitenancy sicura è garantita dalle seguenti caratteristiche:

- **Autenticazione sicura.** Per l'accesso sicuro al volume viene utilizzato il protocollo CHAP (Challenge-Handshake Authentication Protocol). Il protocollo LDAP (Lightweight Directory Access Protocol) viene utilizzato per l'accesso sicuro al cluster a fini di gestione e reporting.
- **Gruppi di accesso al volume (VAG).** Facoltativamente, i VAG possono essere utilizzati al posto dell'autenticazione, mappando un numero qualsiasi di nomi qualificati iSCSI (IQN) specifici dell'iniziatore iSCSI a uno o più volumi. Per accedere a un volume in un VAG, l'IQN dell'iniziatore deve essere presente nell'elenco degli IQN consentiti per il gruppo di volumi.
- **LAN virtuali tenant (VLAN).** A livello di rete, la sicurezza di rete end-to-end tra gli iniziatori iSCSI e il cluster software NetApp Element è facilitata dall'utilizzo di VLAN. Per ogni VLAN creata per isolare un carico di lavoro o un tenant, NetApp Element Software crea un indirizzo SVIP di destinazione iSCSI separato, accessibile solo tramite la VLAN specifica.
- **VLAN abilitate per VRF.** Per supportare ulteriormente la sicurezza e la scalabilità nel data center, il software NetApp Element consente di abilitare qualsiasi VLAN tenant per funzionalità di tipo VRF. Questa funzionalità aggiunge due funzionalità chiave:
  - **Routing L3 verso un indirizzo SVIP del tenant.** Questa funzionalità consente di posizionare gli iniziatori iSCSI su una rete o VLAN separata da quella del cluster software NetApp Element .
  - **Sottoreti IP sovrapposte o duplicate.** Questa funzionalità consente di aggiungere un modello agli ambienti tenant, consentendo a ciascuna rispettiva VLAN tenant di ricevere indirizzi IP dalla stessa subnet IP. Questa funzionalità può essere utile per gli ambienti dei provider di servizi in cui la scalabilità e la conservazione dello spazio IP sono importanti.

Efficienza di archiviazione aziendale

Il cluster software NetApp Element aumenta l'efficienza e le prestazioni complessive dello storage. Le seguenti funzionalità vengono eseguite in linea, sono sempre attive e non richiedono alcuna configurazione manuale da parte dell'utente:

- **Deduplicazione.** Il sistema memorizza solo blocchi univoci da 4K. Tutti i blocchi 4K duplicati vengono automaticamente associati a una versione dei dati già memorizzata. I dati sono archiviati su unità a blocchi e vengono replicati tramite la protezione dati Helix del software NetApp Element . Questo sistema riduce significativamente il consumo di capacità e le operazioni di scrittura all'interno del sistema.
- **Compressione.** La compressione viene eseguita in linea prima che i dati vengano scritti nella NVRAM. I dati vengono compressi, memorizzati in blocchi da 4K e rimangono compressi nel sistema. Questa compressione riduce significativamente il consumo di capacità, le operazioni di scrittura e il consumo di larghezza di banda nel cluster.
- **Provisioning sottile.** Questa funzionalità fornisce la giusta quantità di storage nel momento in cui ne hai bisogno, eliminando il consumo di capacità causato da volumi sovradimensionati o sottoutilizzati.
- **Elica.** I metadati di un singolo volume vengono archiviati su un'unità metadati e replicati su un'unità metadati secondaria per ridondanza.

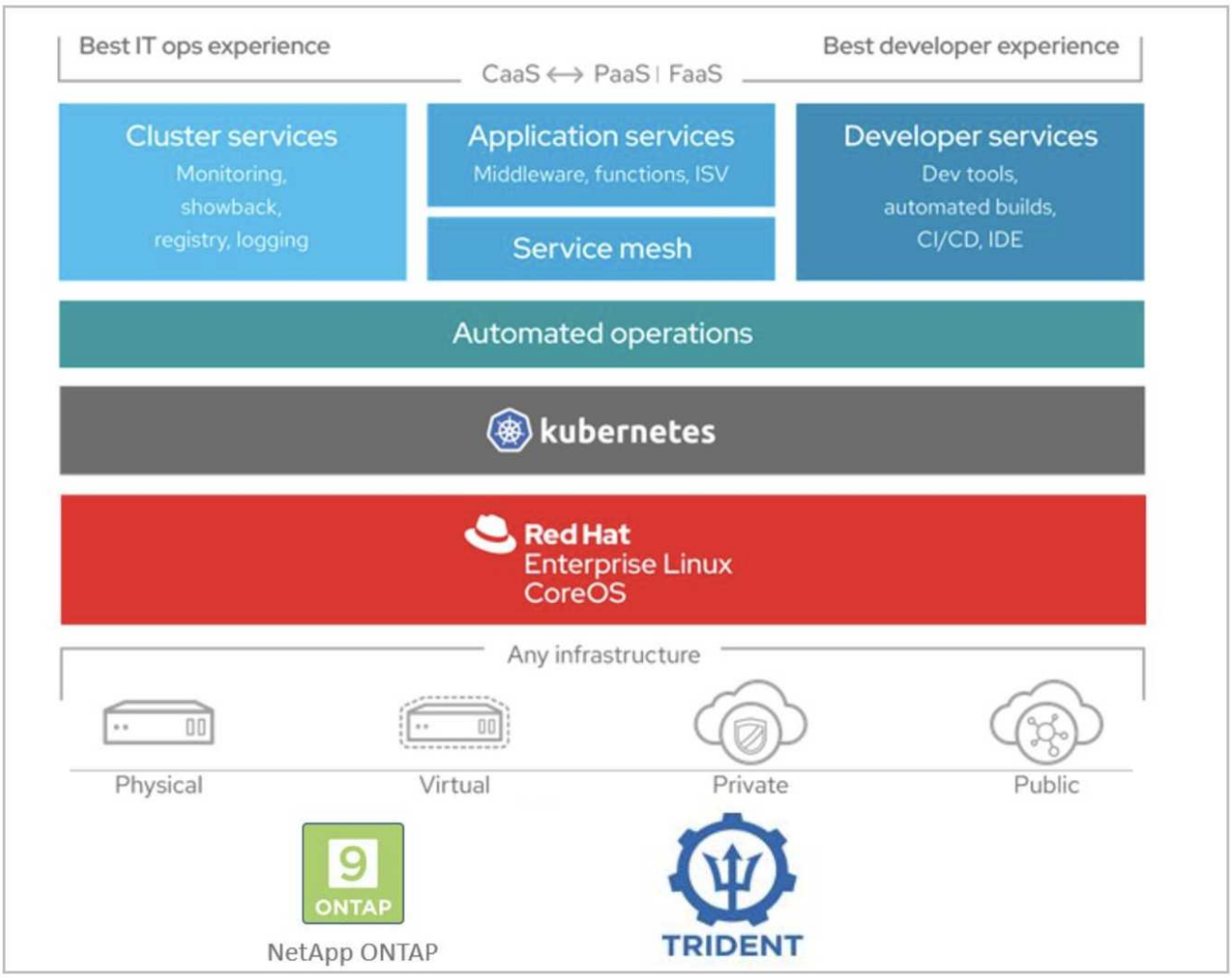
> Element è stato progettato per l'automazione. Tutte le funzionalità di archiviazione sono disponibili tramite API. Queste API sono l'unico metodo utilizzato dall'interfaccia utente per controllare il sistema.

Integrazioni di storage NetApp

Scopri di più sull'integrazione di NetApp Trident con Red Hat OpenShift

Scopri di più su NetApp Trident Protect, convalidato per la gestione delle applicazioni e dello storage persistente per la soluzione OpenShift Virtualization.

Trident, un provider e orchestratore di storage open source gestito da NetApp , e NetApp Trident Protect ti aiutano a orchestrare e gestire dati persistenti in ambienti basati su container, come Red Hat OpenShift.



Le pagine seguenti contengono informazioni aggiuntive sui prodotti NetApp convalidati per la gestione delle applicazioni e dello storage persistente nella soluzione Red Hat OpenShift con NetApp :

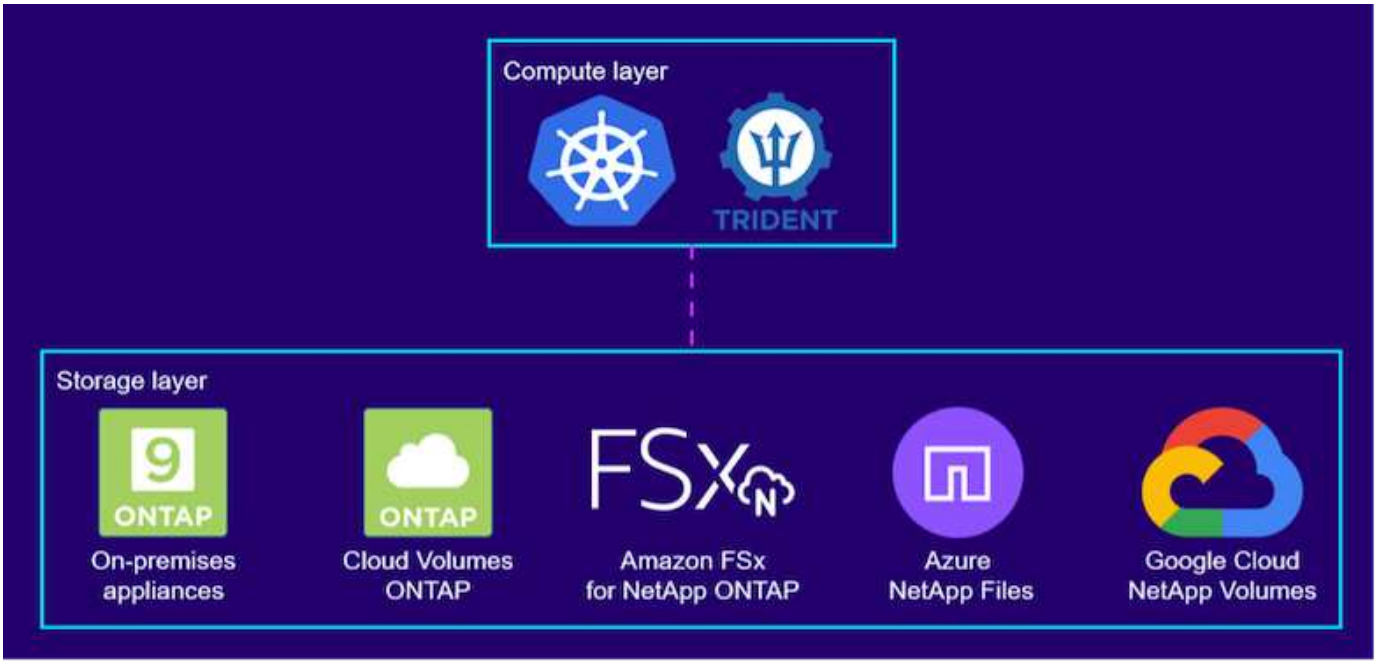
- "Documentazione Trident"
- "Documentazione di protezione Trident"

NetApp Trident

Panoramica Trident

Trident è un orchestratore di storage open source e completamente supportato per container e distribuzioni Kubernetes, tra cui Red Hat OpenShift. Trident funziona con l'intero portfolio di storage NetApp , inclusi i sistemi di storage NetApp ONTAP ed Element, e supporta anche connessioni NFS e iSCSI. Trident accelera il flusso di lavoro DevOps consentendo agli utenti finali di effettuare il provisioning e gestire lo storage dai propri sistemi di storage NetApp senza richiedere l'intervento di un amministratore dello storage.

Un amministratore può configurare una serie di backend di archiviazione in base alle esigenze del progetto e ai modelli di sistema di archiviazione che abilitano funzionalità di archiviazione avanzate, tra cui compressione, tipi di dischi specifici o livelli QoS che garantiscono un certo livello di prestazioni. Una volta definiti, questi backend possono essere utilizzati dagli sviluppatori nei loro progetti per creare richieste di volume persistenti (PVC) e per collegare storage persistente ai loro contenitori su richiesta.



Trident ha un ciclo di sviluppo rapido e, proprio come Kubernetes, viene rilasciato quattro volte all'anno.

È possibile trovare una matrice di supporto per la versione di Trident testata con quale distribuzione Kubernetes "Qui" .

Si prega di fare riferimento al"Documentazione del prodotto Trident" per i dettagli di installazione e configurazione.

Scarica Trident

Per installare Trident sul cluster utente distribuito e predisporre un volume persistente, completare i seguenti passaggi:

1. Scaricare l'archivio di installazione sulla postazione di amministrazione ed estrarne il contenuto. La versione attuale di Trident può essere scaricata "Qui" .
2. Estrarre l'installazione Trident dal pacchetto scaricato.

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

Installare l'operatore Trident con timone

1. Per prima cosa imposta la posizione del cluster utente `kubeconfig` file come variabile di ambiente in modo da non doverlo più referenziare, perché Trident non ha alcuna opzione per passare questo file.

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

2. Eseguire il comando Helm per installare l'operatore Trident dal tarball nella directory helm durante la creazione dello spazio dei nomi trident nel cluster utente.

```
[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. È possibile verificare che Trident sia stato installato correttamente controllando i pod in esecuzione nello spazio dei nomi oppure utilizzando il binario tridentctl per controllare la versione installata.

```
[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        |
+----------------+----------------+
```

In alcuni casi, gli ambienti dei clienti potrebbero richiedere la personalizzazione della distribuzione Trident . In questi casi, è anche possibile installare manualmente l'operatore Trident e aggiornare i manifest inclusi per personalizzare la distribuzione.

Installare manualmente l'operatore Trident

1. Per prima cosa, imposta la posizione del cluster utente `kubeconfig` file come variabile di ambiente in modo da non doverlo più referenziare, perché Trident non ha alcuna opzione per passare questo file.

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

2. IL `trident-installer` La directory contiene i manifesti per definire tutte le risorse richieste. Utilizzando i manifesti appropriati, creare il `TridentOrchestrator` definizione di risorsa personalizzata.

```
[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 non ne esiste uno, crea uno spazio dei nomi Trident nel tuo cluster utilizzando il manifesto fornito.

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

4. Creare le risorse necessarie per la distribuzione dell'operatore Trident , come ad esempio un `ServiceAccount` per l'operatore, un `ClusterRole` E `ClusterRoleBinding` al `ServiceAccount` , un dedicato `PodSecurityPolicy` , o l'operatore stesso.

```
[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. È possibile verificare lo stato dell'operatore dopo la sua distribuzione con i seguenti comandi:

```
[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. Una volta implementato l'operatore, possiamo utilizzarlo per installare Trident. Ciò richiede la creazione di un `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. È possibile verificare che Trident sia stato installato correttamente controllando i pod in esecuzione nello spazio dei nomi oppure utilizzando il binario tridentctl per controllare la versione installata.

```
[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        |
+----------------+----------------+
```

Preparare i nodi worker per l'archiviazione

NFS

La maggior parte delle distribuzioni di Kubernetes sono dotate di pacchetti e utilità per montare i backend NFS installati di default, tra cui Red Hat OpenShift.

Tuttavia, per NFSv3, non esiste alcun meccanismo per negoziare la concorrenza tra il client e il server.

Pertanto, il numero massimo di voci della tabella degli slot sunrpc lato client deve essere sincronizzato manualmente con il valore supportato sul server per garantire le migliori prestazioni per la connessione NFS senza che il server debba ridurre le dimensioni della finestra della connessione.

Per ONTAP, il numero massimo supportato di voci nella tabella degli slot sunrpc è 128, ovvero ONTAP può gestire 128 richieste NFS simultanee alla volta. Tuttavia, per impostazione predefinita, Red Hat CoreOS/Red Hat Enterprise Linux ha un massimo di 65.536 voci nella tabella degli slot sunrpc per connessione. Dobbiamo impostare questo valore su 128 e questo può essere fatto utilizzando Machine Config Operator (MCO) in OpenShift.

Per modificare il numero massimo di voci della tabella degli slot sunrpc nei nodi worker di OpenShift, completare i seguenti passaggi:

1. Accedi alla console web OCP e vai su Calcolo > Configurazioni macchina. Fare clic su Crea configurazione macchina. Copia e incolla il file YAML e fai clic su Crea.

```
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. Dopo aver creato l'MCO, la configurazione deve essere applicata a tutti i nodi worker e riavviata uno alla volta. L'intero processo dura circa 20-30 minuti. Verificare se la configurazione della macchina è applicata utilizzando `oc get mcp` e assicurarsi che il pool di configurazione delle macchine per i worker sia aggiornato.

```
[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

Per preparare i nodi worker a consentire la mappatura dei volumi di archiviazione a blocchi tramite il protocollo iSCSI, è necessario installare i pacchetti necessari per supportare tale funzionalità.

In Red Hat OpenShift, questa operazione viene gestita applicando un MCO (Machine Config Operator) al cluster dopo la sua distribuzione.

Per configurare i nodi worker per l'esecuzione dei servizi iSCSI, completare i seguenti passaggi:

1. Accedi alla console web OCP e vai su Calcolo > Configurazioni macchina. Fare clic su Crea configurazione macchina. Copia e incolla il file YAML e fai clic su Crea.

   Quando non si utilizza il 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: ""
```

   Quando si utilizza il 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. Dopo aver creato la configurazione, occorrono circa 20-30 minuti per applicarla ai nodi worker e ricaricarli. Verificare se la configurazione della macchina è applicata utilizzando `oc get mcp` e assicurarsi che il pool di configurazione delle macchine per i worker sia aggiornato. È anche possibile accedere ai nodi worker per confermare che il servizio iscsid sia in esecuzione (e che il servizio multipathd sia in esecuzione se si utilizza il 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
```

È anche possibile confermare che MachineConfig è stato applicato correttamente e che i servizi sono stati avviati come previsto eseguendo il comando `oc debug` comando con i flag appropriati.

Creare backend del sistema di archiviazione

Dopo aver completato l'installazione di Trident Operator, è necessario configurare il backend per la specifica

piattaforma di storage NetApp utilizzata. Seguire i link sottostanti per continuare l'installazione e la configurazione di Trident.

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

Configurazione NFS NetApp ONTAP

Per abilitare l'integrazione Trident con il sistema di storage NetApp ONTAP , è necessario creare un backend che consenta la comunicazione con il sistema di storage.

1. Sono disponibili file di backend di esempio nell'archivio di installazione scaricato in `sample-input` gerarchia delle cartelle. Per i sistemi NetApp ONTAP che servono NFS, copiare il `backend-ontap-nas.json` file nella directory di lavoro e modificarlo.

```
[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. Modificare i valori backendName, managementLIF, dataLIF, svm, username e password in questo file.

```
{
    "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"
}
```

> Per una facile identificazione, è consigliabile definire il valore backendName personalizzato come una combinazione di storageDriverName e dataLIF che gestisce NFS.

3. Con questo file backend al suo posto, esegui il seguente comando per creare il tuo primo 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. Dopo aver creato il backend, è necessario creare una classe di archiviazione. Proprio come per il backend, nella cartella sample-inputs è disponibile un file di classe di archiviazione di esempio che può essere modificato per l'ambiente. Copialo nella directory di lavoro e apporta le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire il `backendType` valore al nome del driver di archiviazione dal backend appena creato. Si noti anche il valore del campo nome, a cui si dovrà fare riferimento in un passaggio successivo.

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

> C'è un campo facoltativo chiamato `fsType` che è definito in questo file. Questa riga può essere eliminata nei backend NFS.

6. Esegui il `oc` comando per creare la classe di archiviazione.

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

7. Dopo aver creato la classe di archiviazione, è necessario creare la prima richiesta di volume persistente (PVC). C'è un campione `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, anch'esso presente in 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. L'unica modifica che deve essere apportata a questo file è assicurarsi che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

9. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di supporto creato, quindi è possibile osservare il processo mentre viene completato.

```
[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
```

Configurazione iSCSI NetApp ONTAP

Per abilitare l'integrazione Trident con il sistema di storage NetApp ONTAP , è necessario creare un backend che consenta la comunicazione con il sistema di storage.

1. Sono disponibili file di backend di esempio nell'archivio di installazione scaricato in `sample-input` gerarchia delle cartelle. Per i sistemi NetApp ONTAP che servono iSCSI, copiare il `backend-ontap-`

`san.json` file nella directory di lavoro e modificarlo.

```
[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. Modificare i valori managementLIF, dataLIF, svm, username e password in questo file.

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

3. Con questo file backend al suo posto, esegui il seguente comando per creare il tuo primo 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. Dopo aver creato il backend, è necessario creare una classe di archiviazione. Proprio come per il backend, nella cartella sample-inputs è disponibile un file di classe di archiviazione di esempio che può essere modificato per l'ambiente. Copialo nella directory di lavoro e apporta le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire il `backendType` valore al nome del

driver di archiviazione dal backend appena creato. Si noti anche il valore del campo nome, a cui si dovrà fare riferimento in un passaggio successivo.

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

C'è un campo facoltativo chiamato `fsType` che è definito in questo file. Nei backend iSCSI, questo valore può essere impostato su un tipo specifico di file system Linux (XFS, ext4, ecc.) oppure può essere eliminato per consentire a OpenShift di decidere quale file system utilizzare.

6. Esegui il `oc` comando per creare la classe di archiviazione.

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

7. Dopo aver creato la classe di archiviazione, è necessario creare la prima richiesta di volume persistente (PVC). C'è un campione `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, anch'esso presente in 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. L'unica modifica che deve essere apportata a questo file è assicurarsi che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

9. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di supporto creato, quindi è possibile osservare il processo mentre viene completato.

```
[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
```

Configurazione iSCSI NetApp Element

Per abilitare l'integrazione Trident con il sistema di storage NetApp Element , è necessario creare un backend che consenta la comunicazione con il sistema di storage tramite il protocollo iSCSI.

1. Sono disponibili file di backend di esempio nell'archivio di installazione scaricato in `sample-input` gerarchia delle cartelle. Per i sistemi NetApp Element che servono iSCSI, copiare il `backend-solidfire.json` file nella directory di lavoro e modificarlo.

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

   a. Modificare l'utente, la password e il valore MVIP su `EndPoint` linea.

   b. Modifica il `SVIP` valore.

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

2. Con questo file back-end al suo posto, esegui il seguente comando per creare il tuo primo 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. Dopo aver creato il backend, è necessario creare una classe di archiviazione. Proprio come per il backend, nella cartella sample-inputs è disponibile un file di classe di archiviazione di esempio che può essere modificato per l'ambiente. Copialo nella directory di lavoro e apporta le modifiche necessarie per riflettere il backend creato.

```
[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. L'unica modifica che deve essere apportata a questo file è definire il `backendType` valore al nome del driver di archiviazione dal backend appena creato. Si noti anche il valore del campo nome, a cui si dovrà fare riferimento in un passaggio successivo.

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

C'è un campo facoltativo chiamato `fsType` che è definito in questo file. Nei backend iSCSI, questo valore può essere impostato su un tipo di file system Linux specifico (XFS, ext4 e così via) oppure può essere eliminato per consentire a OpenShift di decidere quale file system utilizzare.

5. Esegui il `oc` comando per creare la classe di archiviazione.

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

6. Dopo aver creato la classe di archiviazione, è necessario creare la prima richiesta di volume persistente (PVC). C'è un campione `pvc-basic.yaml` file che può essere utilizzato per eseguire questa azione, anch'esso presente in 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. L'unica modifica che deve essere apportata a questo file è assicurarsi che il `storageClassName` il campo corrisponde a quello appena creato. La definizione PVC può essere ulteriormente personalizzata in base alle esigenze del carico di lavoro da fornire.

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

8. Creare il PVC emettendo il `oc` comando. La creazione può richiedere del tempo a seconda delle dimensioni del volume di supporto creato, quindi è possibile osservare il processo mentre viene completato.

```
[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
```

Opzioni di configurazione avanzate

Esplora le opzioni del bilanciatore del carico

Esplorazione delle opzioni di bilanciamento del carico: Red Hat OpenShift con NetApp

Nella maggior parte dei casi, Red Hat OpenShift rende le applicazioni disponibili al mondo esterno tramite percorsi. Un servizio viene esposto assegnandogli un nome host raggiungibile esternamente. Il percorso definito e gli endpoint identificati dal suo servizio possono essere utilizzati da un router OpenShift per fornire questa connettività denominata ai client esterni.

Tuttavia, in alcuni casi, le applicazioni richiedono l'implementazione e la configurazione di bilanciatori di carico personalizzati per esporre i servizi appropriati. Un esempio è NetApp Trident Protect. Per soddisfare questa esigenza, abbiamo valutato diverse opzioni di bilanciamento del carico personalizzate. La loro installazione e configurazione sono descritte in questa sezione.

Le pagine seguenti contengono informazioni aggiuntive sulle opzioni di bilanciamento del carico convalidate nella soluzione Red Hat OpenShift con NetApp :

- "MetalloLB"
- "F5 BIG-IP"

Installazione dei bilanciatori di carico MetalLB: Red Hat OpenShift con NetApp

In questa pagina sono elencate le istruzioni di installazione e configurazione per il bilanciatore del carico MetalLB.

MetalLB è un bilanciatore del carico di rete auto-ospitato installato sul tuo cluster OpenShift che consente la creazione di servizi OpenShift di tipo bilanciatore del carico in cluster che non vengono eseguiti su un provider cloud. Le due principali funzionalità di MetalLB che interagiscono per supportare i servizi LoadBalancer sono l'allocazione degli indirizzi e l'annuncio esterno.

Opzioni di configurazione MetalLB

In base al modo in cui MetalLB annuncia l'indirizzo IP assegnato ai servizi LoadBalancer al di fuori del cluster OpenShift, funziona in due modalità:

- **Modalità Livello 2.** In questa modalità, un nodo nel cluster OpenShift assume la proprietà del servizio e risponde alle richieste ARP per quell'IP per renderlo raggiungibile al di fuori del cluster OpenShift. Poiché solo il nodo pubblicizza l'IP, si verifica un collo di bottiglia nella larghezza di banda e si verificano limitazioni nel failover lento. Per maggiori informazioni, consultare la documentazione"Qui" .
- **Modalità BGP.** In questa modalità, tutti i nodi del cluster OpenShift stabiliscono sessioni di peering BGP con un router e pubblicizzano i percorsi per inoltrare il traffico agli IP del servizio. Il prerequisito per questo è integrare MetalLB con un router in quella rete. A causa del meccanismo di hashing in BGP, si verificano alcune limitazioni quando cambia la mappatura IP-nodo per un servizio. Per maggiori informazioni, fare riferimento alla documentazione"Qui" .

Ai fini del presente documento, stiamo configurando MetalLB in modalità layer-2.

Installazione del bilanciatore di carico MetalLB

1. Scarica le risorse 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. Modifica file `metallb.yaml` e rimuovere `spec.template.spec.securityContext` dal controller Deployment e dall'altoparlante DaemonSet.

   Righe da eliminare:

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

3. Crea il `metallb-system` spazio dei nomi.

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

4. Crea il CR MetalLB.

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

5. Prima di configurare l'altoparlante MetalLB, concedere all'altoparlante privilegi elevati DaemonSet in modo che possa eseguire la configurazione di rete necessaria per far funzionare i bilanciatori del carico.

```
[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. Configura MetalLB creando un `ConfigMap` nel `metallb-system` spazio dei nomi.

```
[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. Ora, quando vengono creati i servizi di bilanciamento del carico, MetalLB assegna un IP esterno ai servizi e pubblicizza l'indirizzo IP rispondendo alle richieste ARP.

> Se desideri configurare MetalLB in modalità BGP, salta il passaggio 6 sopra e segui la procedura nella documentazione di MetalLB"Qui" .

Installazione dei bilanciatori di carico F5 BIG-IP

F5 BIG-IP è un Application Delivery Controller (ADC) che offre un'ampia gamma di servizi avanzati di gestione del traffico e di sicurezza di livello produttivo, come bilanciamento del carico L4-L7, offload SSL/TLS, DNS, firewall e molto altro. Questi servizi aumentano notevolmente la disponibilità, la sicurezza e le prestazioni delle tue applicazioni.

F5 BIG-IP può essere distribuito e utilizzato in vari modi: su hardware dedicato, nel cloud o come appliance virtuale in sede. Per esplorare e distribuire F5 BIG-IP in base alle esigenze, fare riferimento alla documentazione qui.

Per un'integrazione efficiente dei servizi F5 BIG-IP con Red Hat OpenShift, F5 offre il servizio BIG-IP Container Ingress Service (CIS). CIS è installato come un pod controller che controlla l'API OpenShift per determinate definizioni di risorse personalizzate (CRD) e gestisce la configurazione del sistema F5 BIG-IP. F5 BIG-IP CIS può essere configurato per controllare i tipi di servizio LoadBalancers e Routes in OpenShift.

Inoltre, per l'assegnazione automatica degli indirizzi IP al servizio del tipo LoadBalancer, è possibile utilizzare il controller F5 IPAM. Il controller F5 IPAM è installato come un pod controller che controlla l'API OpenShift per i servizi LoadBalancer con un'annotazione ipamLabel per allocare l'indirizzo IP da un pool preconfigurato.

In questa pagina sono elencate le istruzioni di installazione e configurazione per il controller F5 BIG-IP CIS e IPAM. Come prerequisito, è necessario disporre di un sistema F5 BIG-IP distribuito e dotato di licenza. Deve inoltre disporre di una licenza per i servizi SDN, inclusi di default nella licenza base BIG-IP VE.

F5 BIG-IP può essere distribuito in modalità standalone o cluster. Ai fini di questa convalida, F5 BIG-IP è stato distribuito in modalità autonoma, ma, per scopi di produzione, è preferibile disporre di un cluster di BIG-IP per evitare un singolo punto di errore.

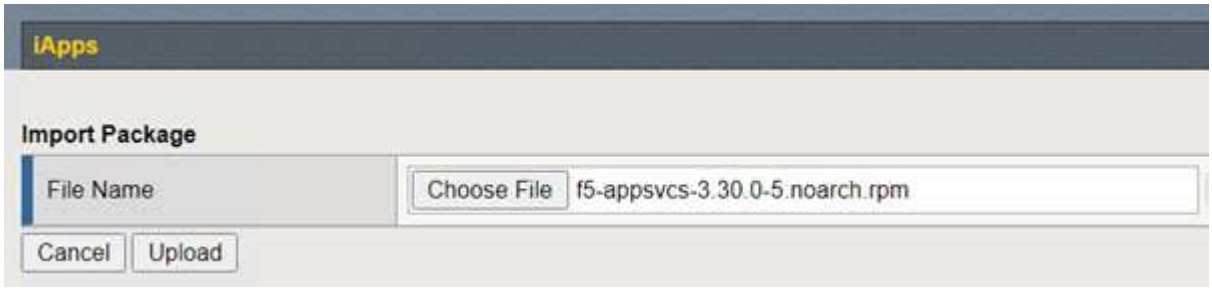
Un sistema F5 BIG-IP può essere distribuito su hardware dedicato, nel cloud o come appliance virtuale in sede con versioni successive alla 12.x per essere integrato con F5 CIS. Ai fini del presente documento, il sistema F5 BIG-IP è stato convalidato come appliance virtuale, ad esempio utilizzando l'edizione BIG-IP VE.

Versioni convalidate

Tecnologia	Versione del software
Red Hat OpenShift	4,6 UE, 4,7
Edizione F5 BIG-IP VE	16.1.0
Servizio di ingresso del contenitore F5	2.5.1
Controllore IPAM F5	0.1.4
F5 AS3	3.30.0

Installazione

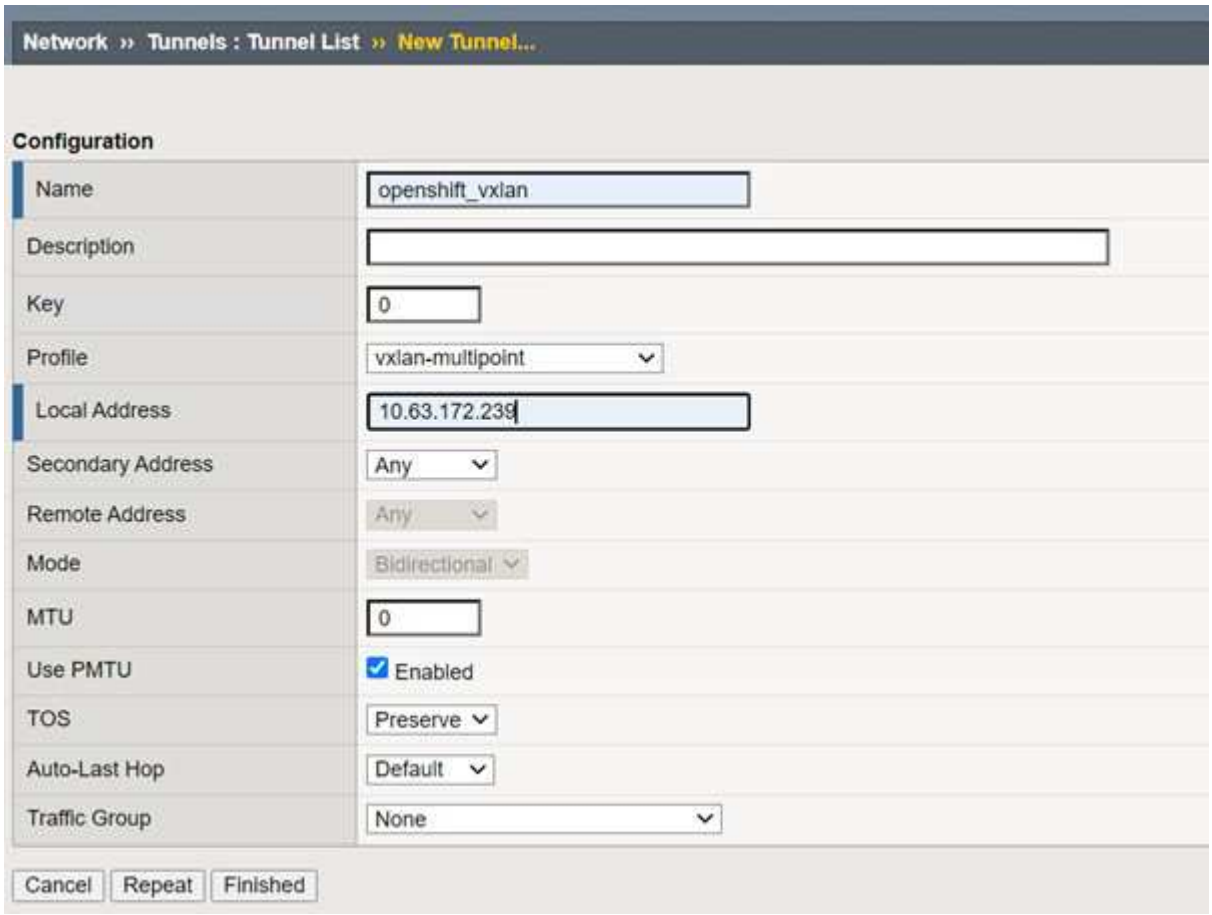
1. Installare l'estensione F5 Application Services 3 per consentire ai sistemi BIG-IP di accettare configurazioni in JSON anziché comandi imperativi. Vai a "Repository GitHub F5 AS3" e scarica l'ultimo file RPM.
2. Accedi al sistema F5 BIG-IP, vai su iApp > Gestione pacchetti LX e fai clic su Importa.
3. Fare clic su Scegli file e selezionare il file AS3 RPM scaricato, fare clic su OK e quindi su Carica.



4. Verificare che l'estensione AS3 sia stata installata correttamente.

5. Successivamente, configurare le risorse necessarie per la comunicazione tra i sistemi OpenShift e BIG-IP. Per prima cosa, creare un tunnel tra OpenShift e il server BIG-IP creando un'interfaccia tunnel VXLAN sul sistema BIG-IP per OpenShift SDN. Vai su Rete > Tunnel > Profili, fai clic su Crea e imposta il Profilo padre su vxlan e il Tipo di flooding su Multicast. Inserisci un nome per il profilo e fai clic su Fine.

6. Vai su Rete > Tunnel > Elenco tunnel, fai clic su Crea e inserisci il nome e l'indirizzo IP locale per il tunnel. Selezionare il profilo del tunnel creato nel passaggio precedente e fare clic su Fine.



7. Accedi al cluster Red Hat OpenShift con privilegi di amministratore del cluster.
8. Creare una hostsubnet su OpenShift per il server F5 BIG-IP, che estende la subnet dal cluster OpenShift al server F5 BIG-IP. Scarica la definizione YAML della sottorete host.

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

9. Modificare il file della subnet host e aggiungere l'IP BIG-IP VTEP (tunnel VXLAN) per 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
```

Modifica l'hostIP e altri dettagli in base al tuo ambiente.

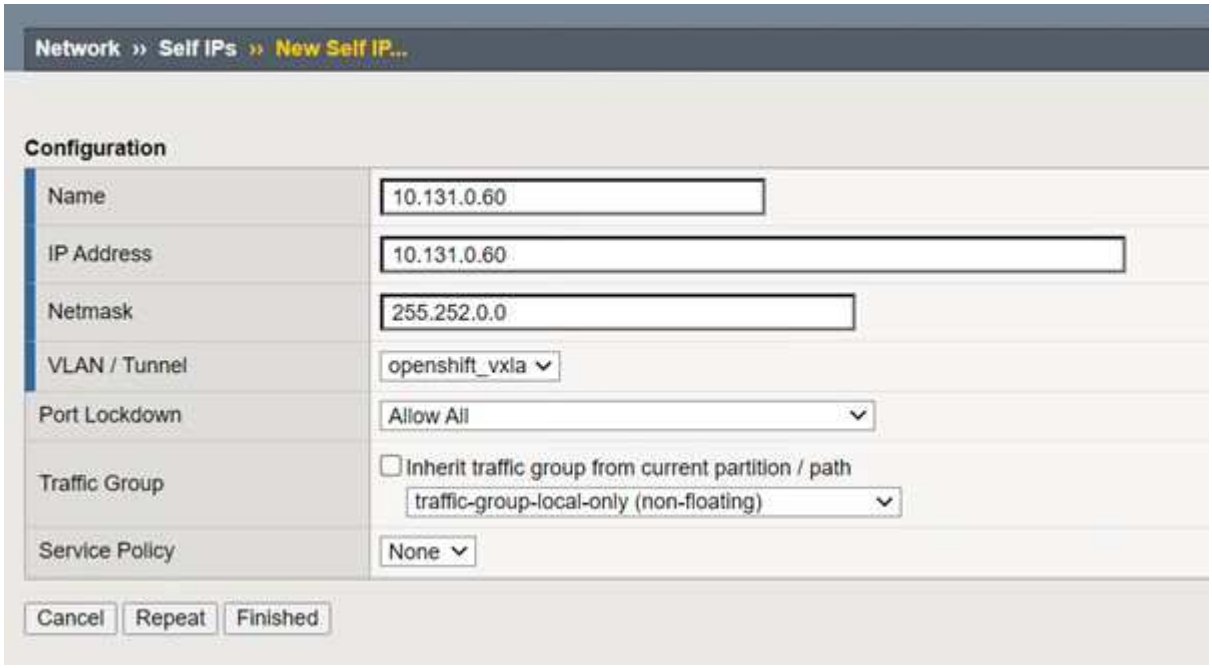
10. Creare la risorsa HostSubnet.

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

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

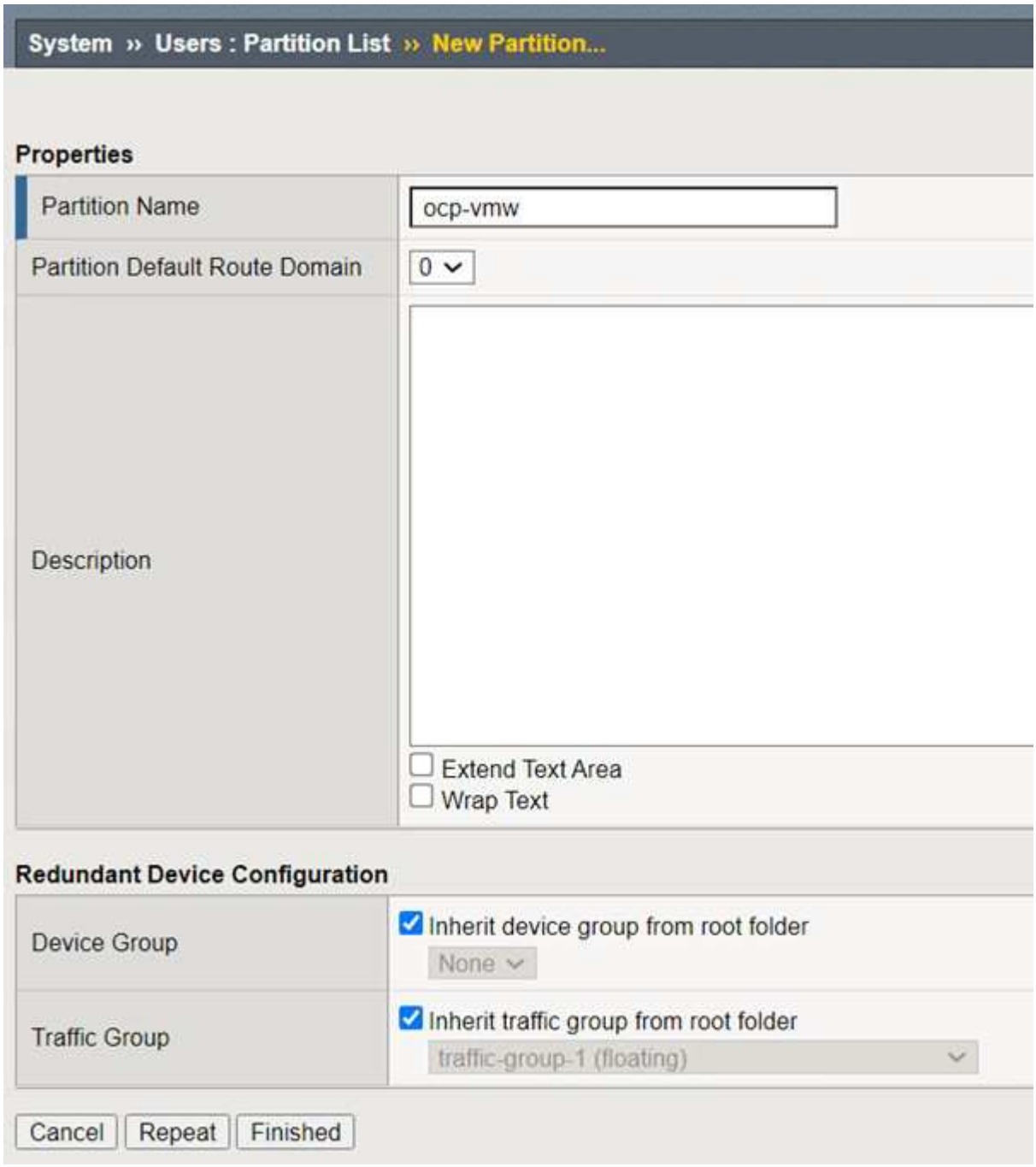
11. Ottieni l'intervallo di subnet IP del cluster per la subnet host creata per il server F5 BIG-IP.

```
[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. Creare un IP autonomo su OpenShift VXLAN con un IP nell'intervallo di subnet host di OpenShift corrispondente al server F5 BIG-IP. Accedi al sistema F5 BIG-IP, vai su Rete > IP personali e fai clic su Crea. Immettere un IP dalla subnet IP del cluster creata per la subnet host F5 BIG-IP, selezionare il tunnel VXLAN e immettere gli altri dettagli. Quindi fare clic su Fine.



13. Creare una partizione nel sistema F5 BIG-IP da configurare e utilizzare con CIS. Vai su Sistema > Utenti > Elenco partizioni, fai clic su Crea e inserisci i dettagli. Quindi fare clic su Fine.



F5 consiglia di non effettuare alcuna configurazione manuale sulla partizione gestita da CIS.

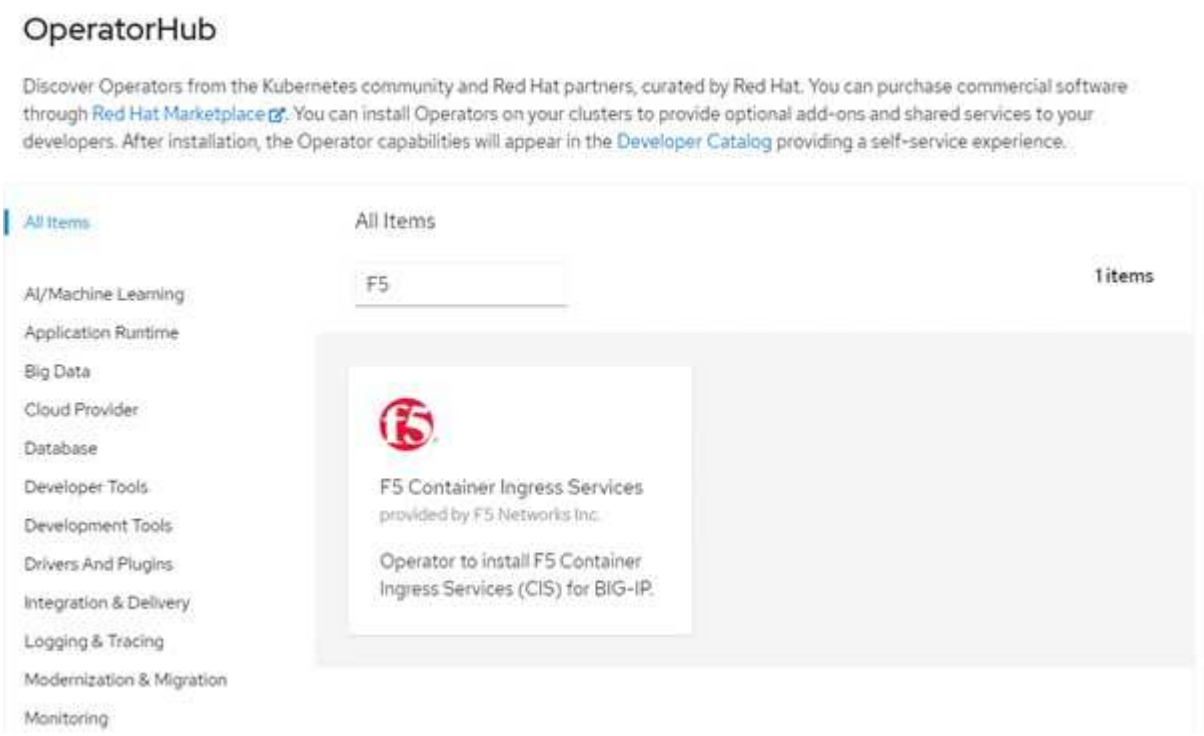
14. Installare F5 BIG-IP CIS utilizzando l'operatore di OperatorHub. Accedere al cluster Red Hat OpenShift con privilegi di amministratore del cluster e creare un segreto con le credenziali di accesso al sistema F5 BIG-IP, che è un prerequisito per l'operatore.

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

secret/bigip-login created
```

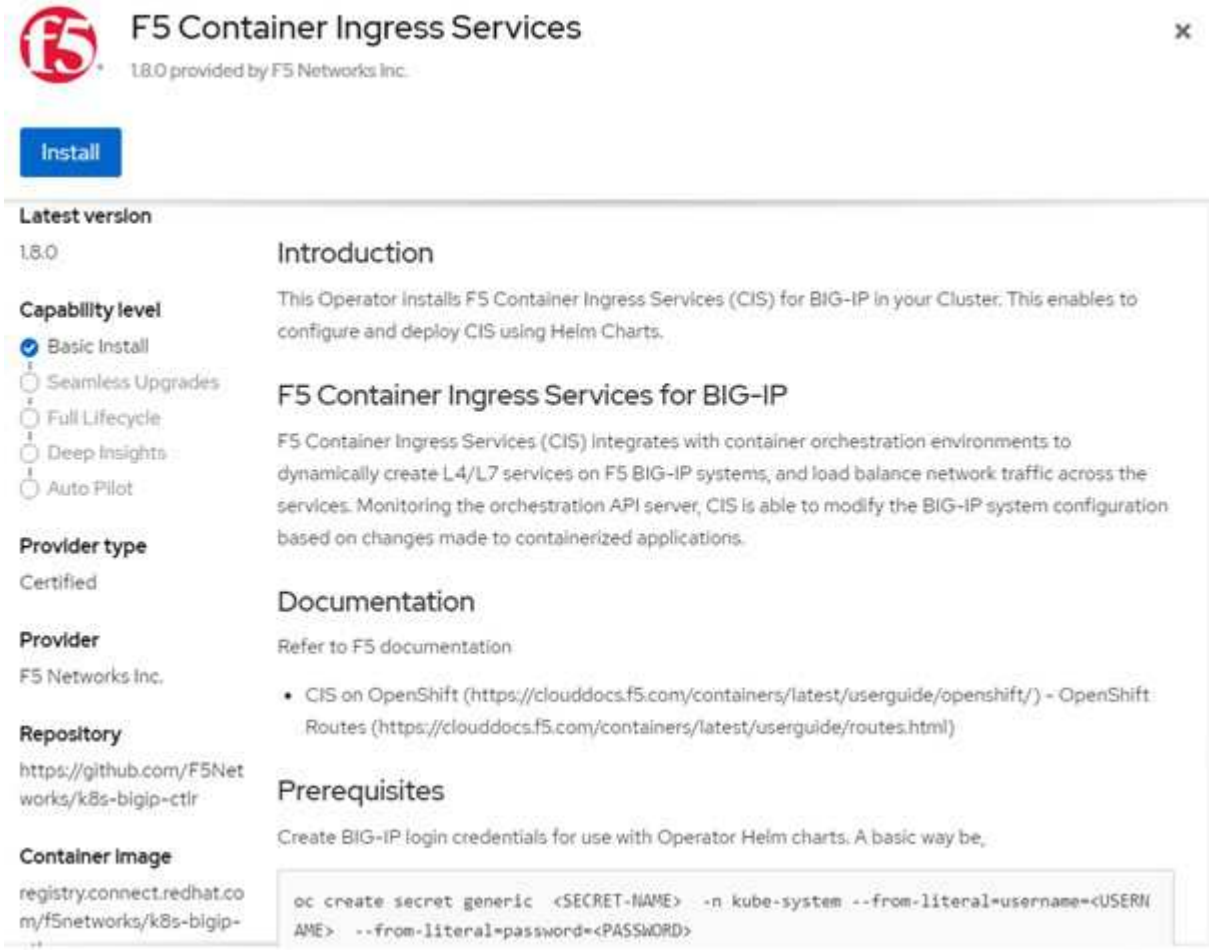
15. Installare i CRD F5 CIS.

```
[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
```

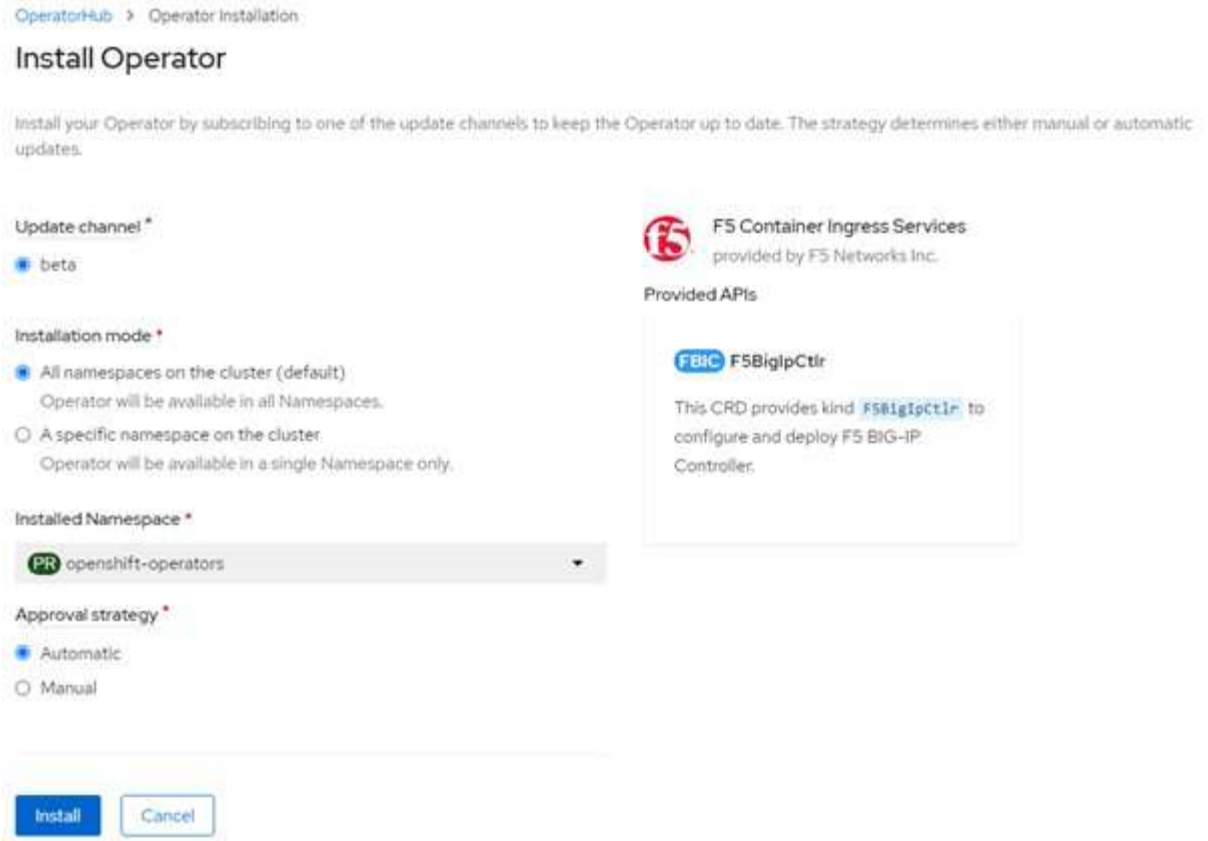
16. Vai su Operatori > OperatorHub, cerca la parola chiave F5 e fai clic sul riquadro F5 Container Ingress Service.



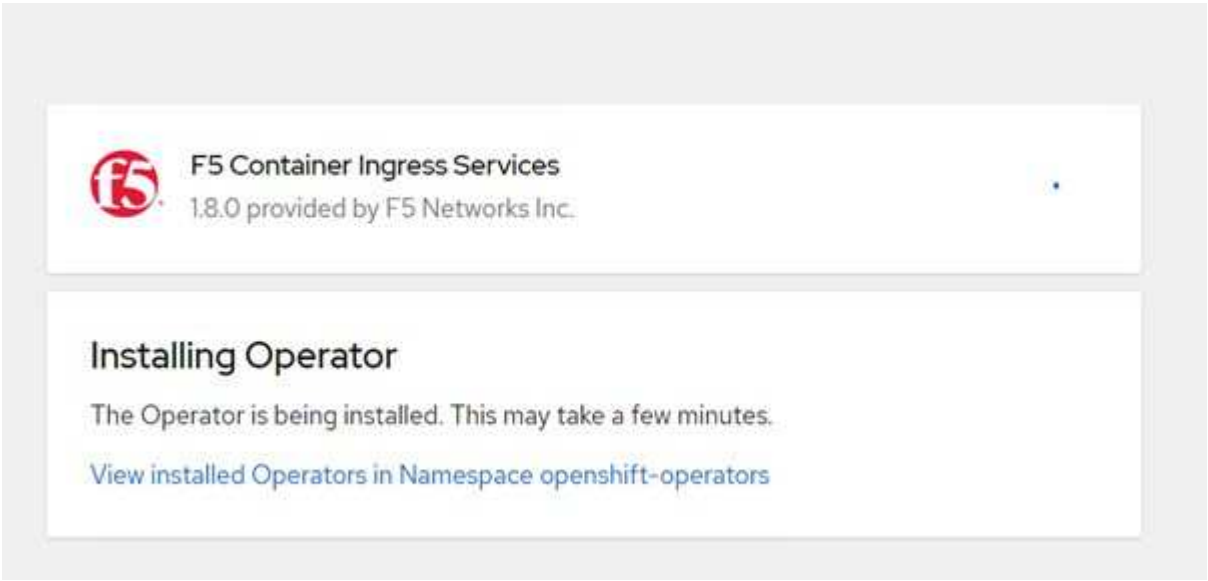
17. Leggere le informazioni dell'operatore e fare clic su Installa.



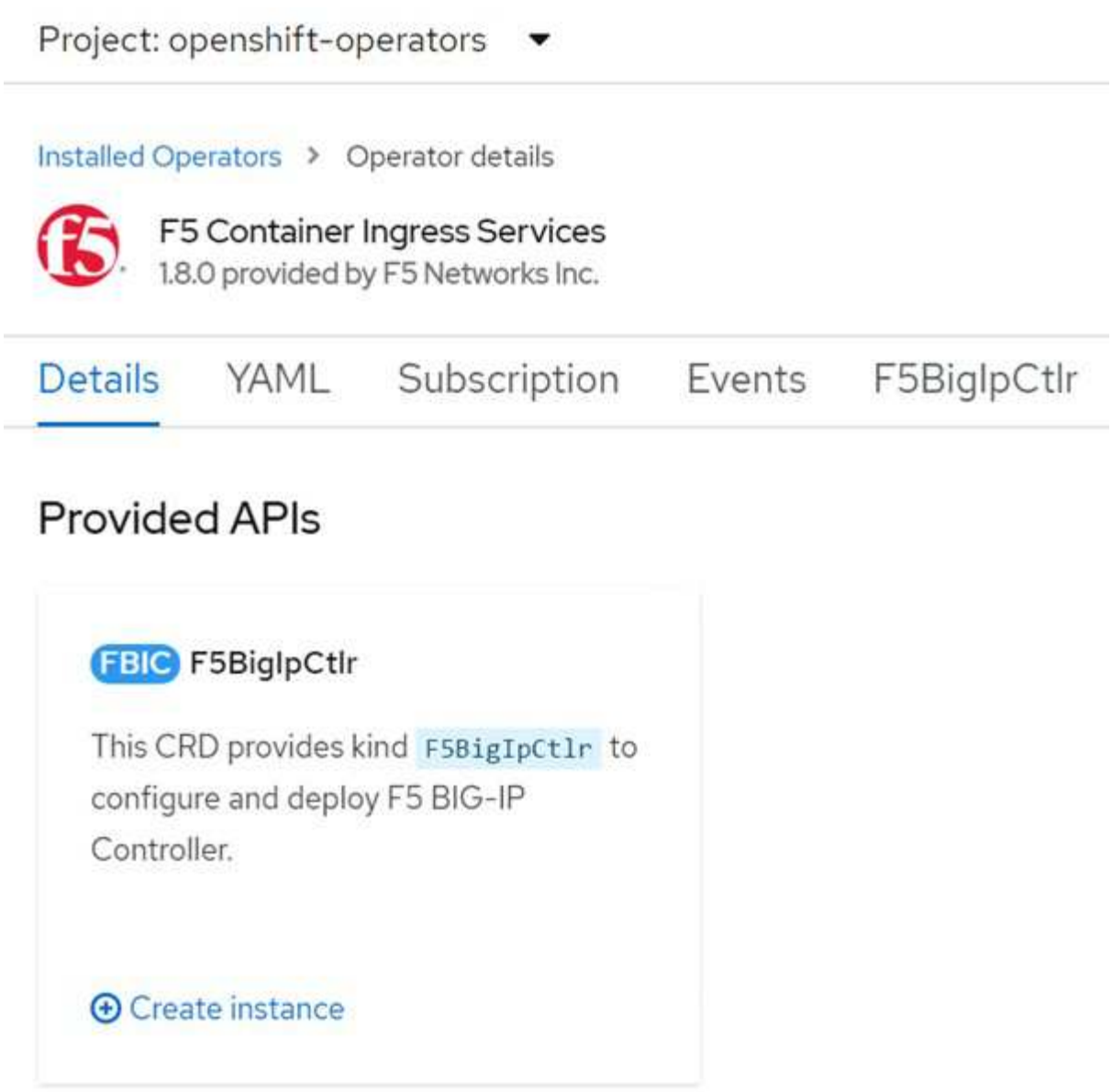
18. Nella schermata Installa operatore, lasciare tutti i parametri predefiniti e fare clic su Installa.



19. L'installazione dell'operatore richiede un po' di tempo.



20. Dopo l'installazione dell'operatore, viene visualizzato il messaggio Installazione riuscita.
21. Passare a Operatori > Operatori installati, fare clic su F5 Container Ingress Service, quindi fare clic su Crea istanza nel riquadro F5BigIpCtlr.



22. Fare clic su Vista YAML e incollare il seguente contenuto dopo aver aggiornato i parametri necessari.

Aggiorna i parametri `bigip_partition` , ` openshift_sdn_name`, `bigip_url` E `bigip_login_secret` di seguito per riflettere i valori per la tua configurazione prima di copiare il contenuto.

```
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. Dopo aver incollato questo contenuto, fai clic su Crea. In questo modo i pod CIS vengono installati nello spazio dei nomi kube-system.

Per impostazione predefinita, Red Hat OpenShift fornisce un modo per esporre i servizi tramite Routes per il bilanciamento del carico L7. Un router OpenShift integrato è responsabile della pubblicità e della gestione del traffico per questi percorsi. Tuttavia, è anche possibile configurare F5 CIS per supportare i percorsi tramite un sistema F5 BIG-IP esterno, che può essere eseguito come router ausiliario o come sostituto del router OpenShift auto-ospitato. CIS crea un server virtuale nel sistema BIG-IP che funge da router per le rotte OpenShift, mentre BIG-IP gestisce la pubblicità e il routing del traffico. Per informazioni sui parametri per abilitare questa funzionalità, fare riferimento alla documentazione qui. Si noti che questi parametri sono definiti per la risorsa OpenShift Deployment nell'API apps/v1. Pertanto, quando si utilizzano questi con l'API cis.f5.com/v1 della risorsa F5BigIpCtlr, sostituire i trattini (-) con caratteri di sottolineatura (_) per i nomi dei parametri.

24. Gli argomenti che vengono passati alla creazione delle risorse CIS includono `ipam: true` E `custom_resource_mode: true` . Questi parametri sono necessari per abilitare l'integrazione CIS con un controller IPAM. Verificare che il CIS abbia abilitato l'integrazione IPAM creando la risorsa 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. Creare l'account di servizio, il ruolo e il rolebinding richiesti per il controller F5 IPAM. Crea un file YAML e incolla il seguente contenuto.

```
[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. Creare le risorse.

```
[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. Creare un file YAML e incollare la definizione di distribuzione F5 IPAM fornita di seguito.

Aggiornare il parametro ip-range in spec.template.spec.containers[0].args di seguito per riflettere gli intervalli di indirizzi IP e ipamLabels corrispondenti alla configurazione.

Etichette ipam[`range1` E `range2` nell'esempio seguente] devono essere annotati per i servizi di tipo LoadBalancer affinché il controller IPAM rilevi e assegni un indirizzo IP dall'intervallo definito.

```
[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. Creare la distribuzione del controller F5 IPAM.

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

deployment/f5-ipam-controller created
```

29. Verificare che i pod del controller F5 IPAM siano in esecuzione.

```
[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. Creare lo schema IPAM F5.

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

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

Verifica

1. Crea un servizio di 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. Verificare se il controller IPAM gli assegna un IP esterno.

```
[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. Creare una distribuzione e utilizzare il servizio LoadBalancer creato.

```
[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. Controllare se i pod sono in funzione.

```
[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. Verificare se il server virtuale corrispondente è stato creato nel sistema BIG-IP per il servizio di tipo LoadBalancer in OpenShift. Vai a Traffico locale > Server virtuali > Elenco server virtuali.

Creazione di registri di immagini private

Per la maggior parte delle distribuzioni di Red Hat OpenShift, utilizzando un registro pubblico come "Quay.io" O "DockerHub" soddisfa la maggior parte delle esigenze dei clienti. Tuttavia, ci sono momenti in cui un cliente potrebbe voler ospitare le proprie immagini private o personalizzate.

Questa procedura documenta la creazione di un registro di immagini privato supportato da un volume persistente fornito da Trident e NetApp ONTAP.

> Trident Protect necessita di un registro per ospitare le immagini richieste dai contenitori Astra . Nella sezione seguente vengono descritti i passaggi per configurare un registro privato sul cluster Red Hat OpenShift e per inviare le immagini necessarie a supportare l'installazione di Trident Protect.

Creazione di un registro di immagini privato

1. Rimuovere l'annotazione predefinita dalla classe di archiviazione predefinita corrente e annotare la classe di archiviazione supportata da Trident come predefinita per il 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. Modificare l'operatore imageregistry immettendo i seguenti parametri di archiviazione nel `spec` sezione.

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

storage:
  pvc:
    claim:
```

3. Inserire i seguenti parametri nel `spec` sezione per la creazione di un percorso OpenShift con un nome host personalizzato. Salva ed esci.

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

> La configurazione del percorso sopra riportata viene utilizzata quando si desidera un nome host personalizzato per il percorso. Se vuoi che OpenShift crei un percorso con un nome host predefinito, puoi aggiungere i seguenti parametri al `spec` sezione: `defaultRoute: true` .

Certificati TLS personalizzati

Quando si utilizza un nome host personalizzato per il percorso, per impostazione predefinita viene utilizzata la configurazione TLS predefinita dell'operatore OpenShift Ingress. Tuttavia, è possibile aggiungere una configurazione TLS personalizzata al percorso. Per farlo, completa i seguenti passaggi.

a. Creare un segreto con i certificati TLS e la chiave del percorso.

```
[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. Modifica l'operatore imageregistry e aggiungi i seguenti parametri al `spec` sezione.

```
[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. Modificare nuovamente l'operatore imageregistry e cambiare lo stato di gestione dell'operatore in `Managed` stato. Salva ed esci.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Se tutti i prerequisiti sono soddisfatti, vengono creati PVC, pod e servizi per il registro delle immagini private. Tra pochi minuti il registro dovrebbe essere attivo.

```
[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 si utilizzano i certificati TLS predefiniti per il percorso del registro OpenShift dell'operatore di ingresso, è possibile recuperare i certificati TLS utilizzando il seguente comando.

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

7. Per consentire ai nodi OpenShift di accedere ed estrarre le immagini dal registro, aggiungere i certificati al client Docker sui nodi OpenShift. Crea una configmap in `openshift-config` namespace utilizzando i certificati TLS e applicarne una patch alla configurazione dell'immagine del cluster per rendere il certificato attendibile.

```
[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. Il registro interno di OpenShift è controllato tramite autenticazione. Tutti gli utenti OpenShift possono accedere al registro OpenShift, ma le operazioni che l'utente registrato può eseguire dipendono dalle autorizzazioni dell'utente.
   a. Per consentire a un utente o a un gruppo di utenti di estrarre immagini dal registro, è necessario che all'utente sia assegnato il ruolo di visualizzatore del registro.

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

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

   b. Per consentire a un utente o a un gruppo di utenti di scrivere o inviare immagini, è necessario che all'utente sia assegnato il ruolo di editor del registro.

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

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

9. Per consentire ai nodi OpenShift di accedere al registro ed eseguire il push o il pull delle immagini, è necessario configurare un segreto pull.

```
[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. Questo segreto pull può quindi essere applicato agli account di servizio o essere referenziato nella definizione del pod corrispondente.
    a. Per applicare la patch agli account di servizio, eseguire il seguente comando.

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

b. Per fare riferimento al segreto pull nella definizione del pod, aggiungere il seguente parametro a `spec` sezione.

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

11. Per eseguire il push o il pull di un'immagine da workstation diverse dal nodo OpenShift, completare i seguenti passaggi.

a. Aggiungere i certificati TLS al client 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. Accedi a OpenShift utilizzando il comando oc login.

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

c. Accedi al registro utilizzando le credenziali utente OpenShift con il comando podman/docker.

podman

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

+ NOTA: Se stai utilizzando `kubeadmin` utente per accedere al registro privato, quindi utilizzare il token anziché la password.

scaricatore mobile

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

+ NOTA: Se stai utilizzando `kubeadmin` utente per accedere al registro privato, quindi utilizzare il token anziché la password.

d. Spingere o tirare le immagini.

podman

```
[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
```

scaricatore mobile

```
[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
```

Validazione della soluzione e casi d'uso

Validazione della soluzione e casi d'uso: Red Hat OpenShift con NetApp

Gli esempi forniti in questa pagina sono convalide di soluzioni e casi d'uso per Red Hat OpenShift con NetApp.

- "Distribuisci una pipeline CI/CD Jenkins con storage persistente"
- "Configurare Multitenancy su Red Hat OpenShift con NetApp"
- "Virtualizzazione Red Hat OpenShift con NetApp ONTAP"
- "Gestione avanzata dei cluster per Kubernetes su Red Hat OpenShift con NetApp"

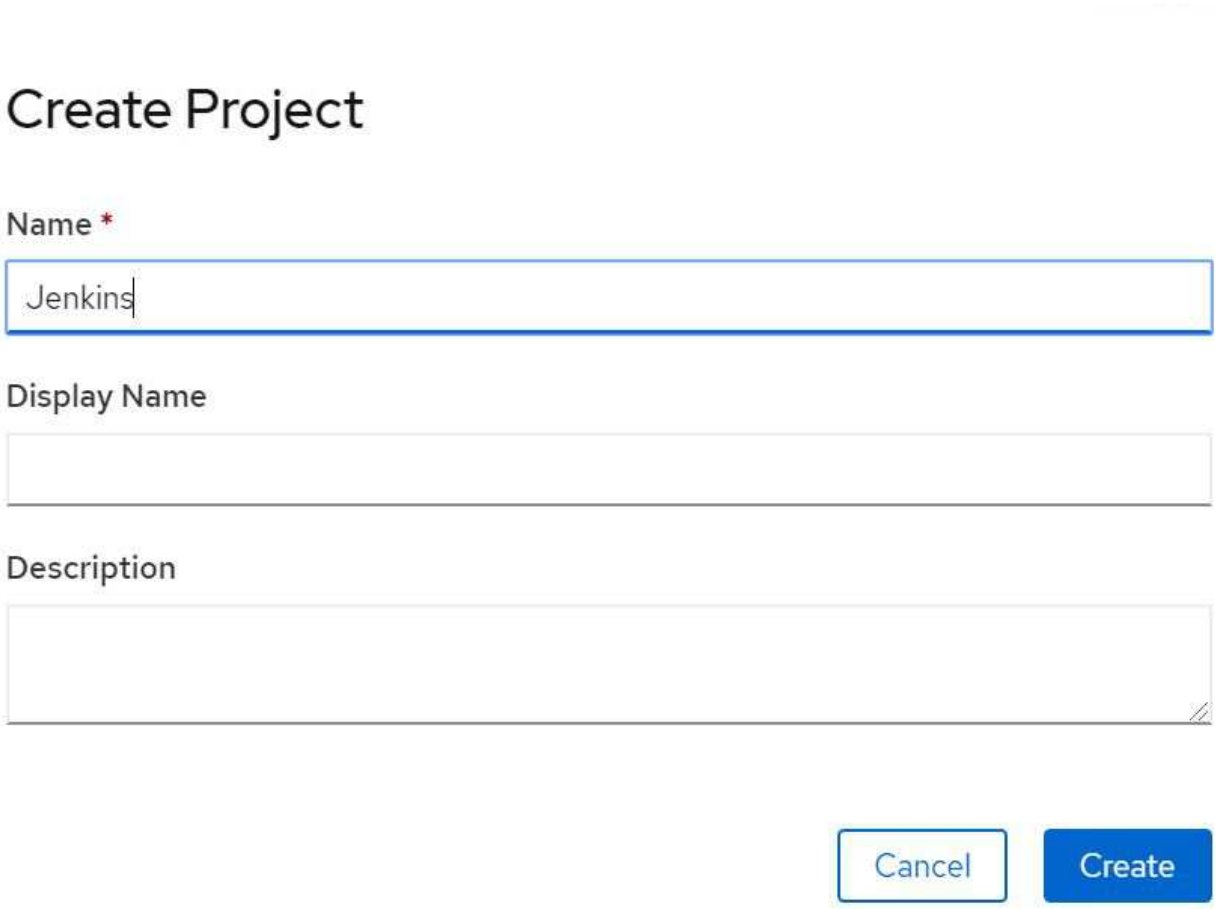
Distribuisci una pipeline CI/CD Jenkins con storage persistente: Red Hat OpenShift con NetApp

Questa sezione illustra i passaggi per distribuire una pipeline di integrazione continua/distribuzione o distribuzione continua (CI/CD) con Jenkins per convalidare il funzionamento della soluzione.

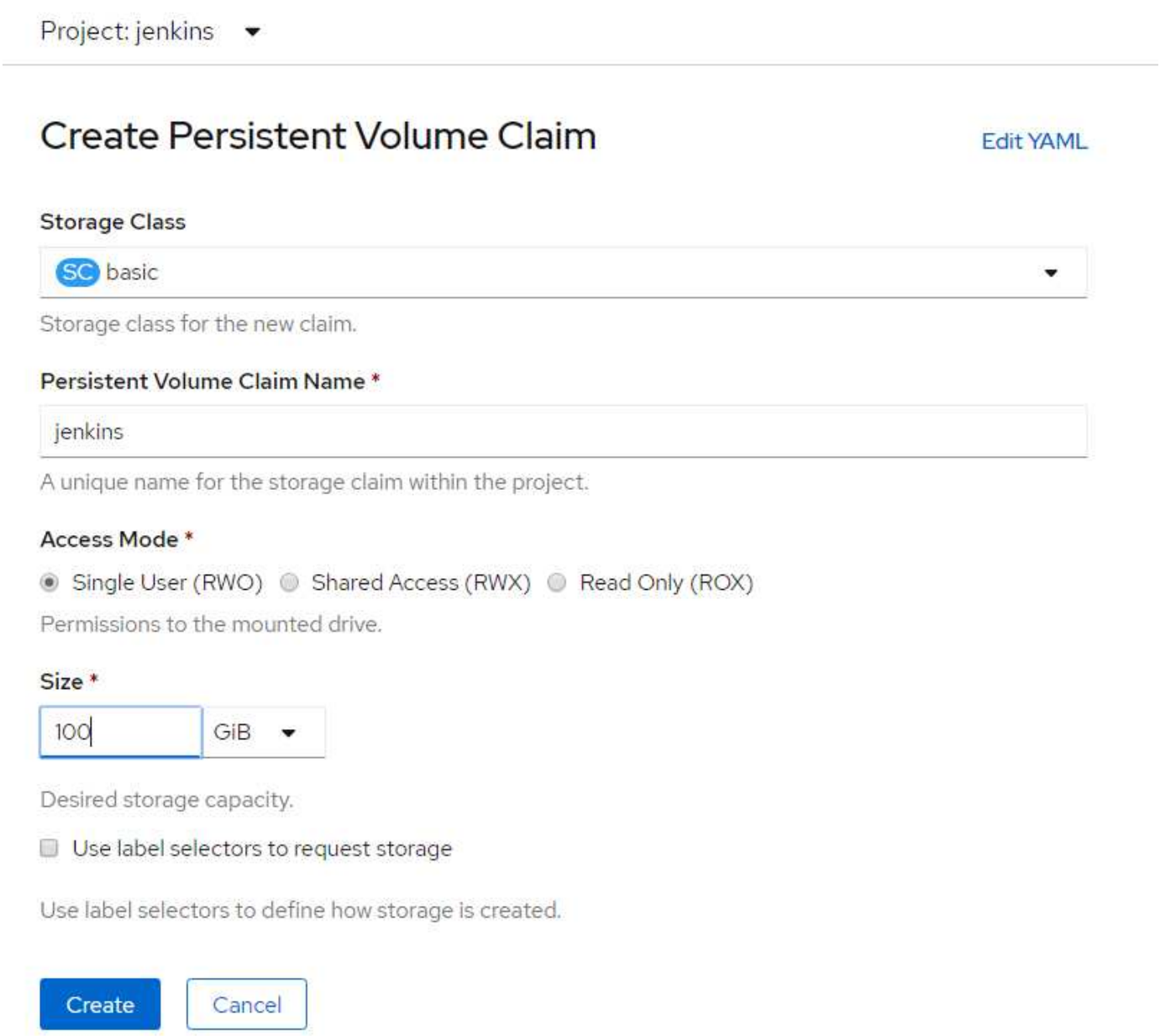
Creare le risorse necessarie per la distribuzione di Jenkins

Per creare le risorse necessarie per la distribuzione dell'applicazione Jenkins, completare i seguenti passaggi:

1. Crea un nuovo progetto denominato Jenkins.



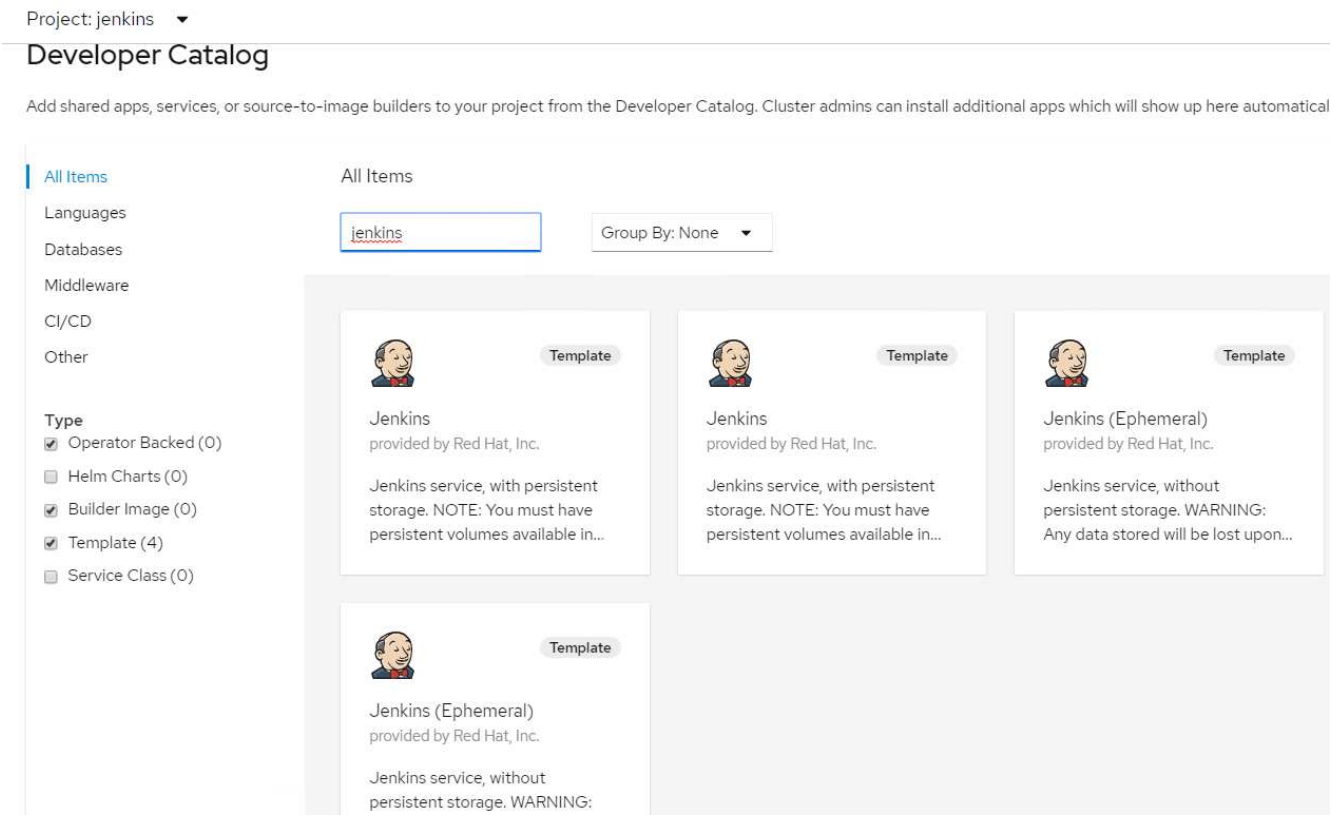
2. In questo esempio abbiamo distribuito Jenkins con storage persistente. Per supportare la build di Jenkins, creare il PVC. Passare a Archiviazione > Richieste di volume persistente e fare clic su Crea richiesta di volume persistente. Selezionare la classe di archiviazione creata, assicurarsi che il nome della richiesta del volume persistente sia jenkins, selezionare la dimensione e la modalità di accesso appropriate, quindi fare clic su Crea.



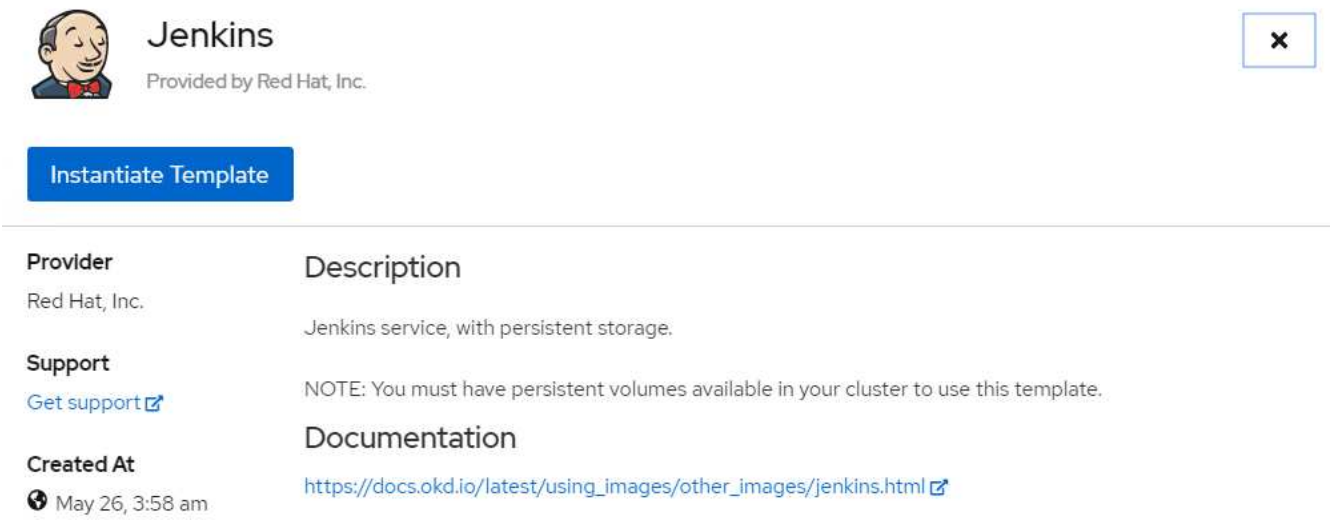
Distribuisci Jenkins con storage persistente

Per distribuire Jenkins con storage persistente, completare i seguenti passaggi:

1. Nell'angolo in alto a sinistra, cambia il ruolo da Amministratore a Sviluppatore. Fare clic su +Aggiungi e selezionare Dal catalogo. Nella barra Filtra per parola chiave, cerca jenkins. Selezionare il servizio Jenkins con archiviazione persistente.

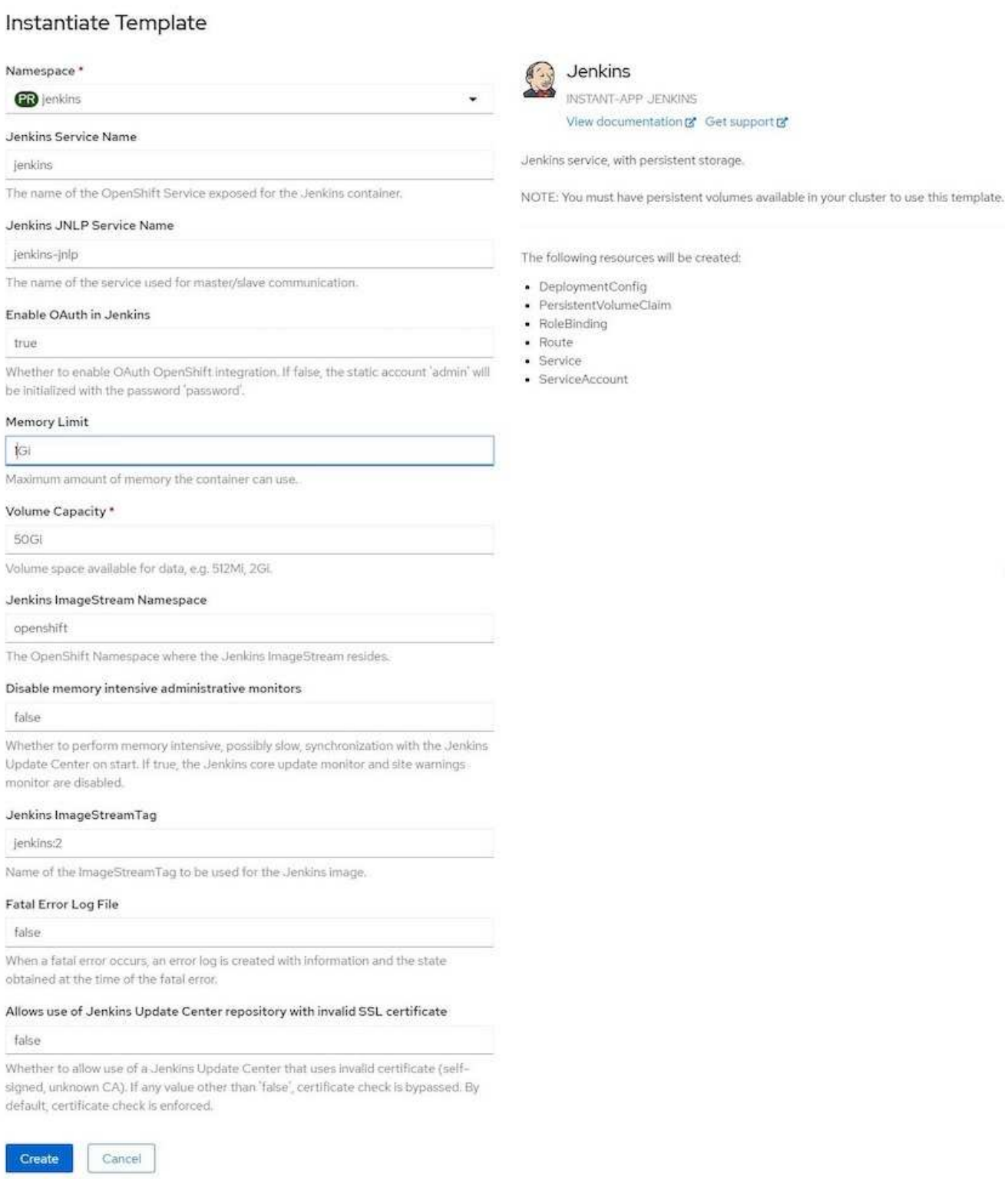


2. Clic `Instantiate Template` .

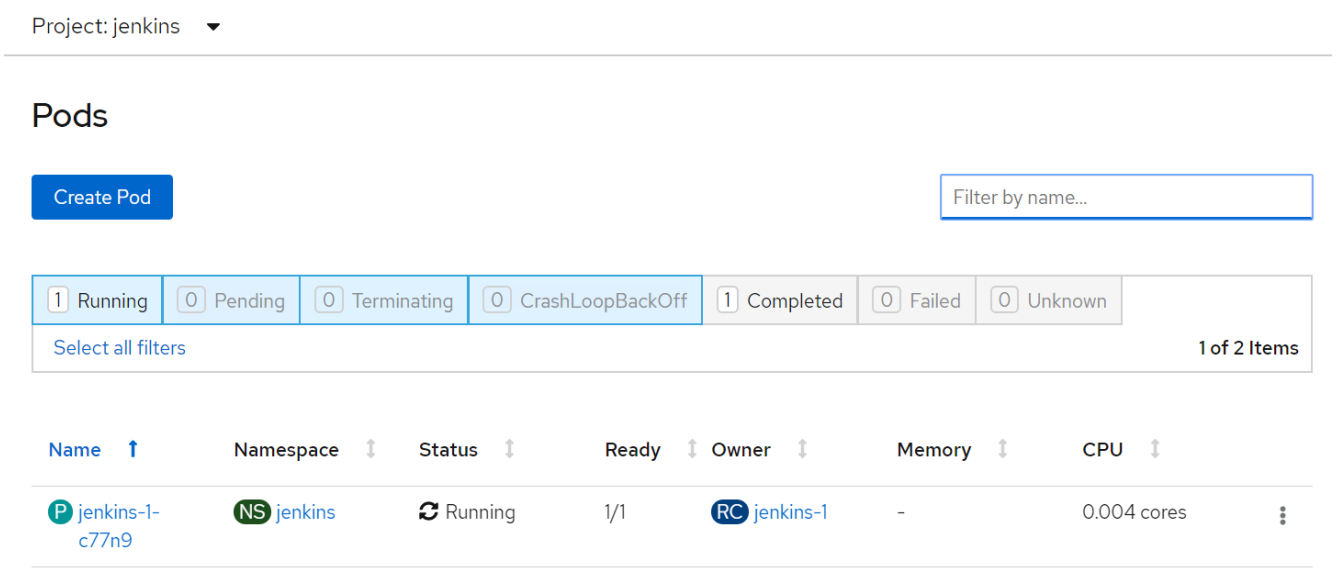


3. Per impostazione predefinita, vengono compilati i dettagli per l'applicazione Jenkins. In base alle tue esigenze, modifica i parametri e fai clic su Crea. Questo processo crea tutte le risorse necessarie per

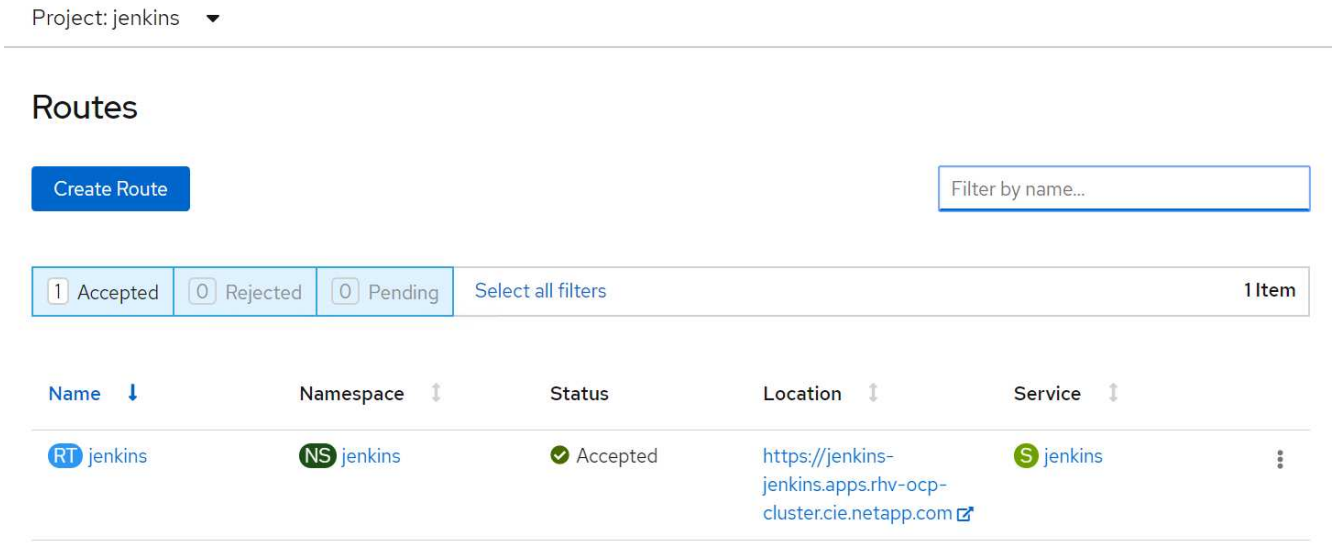
supportare Jenkins su OpenShift.



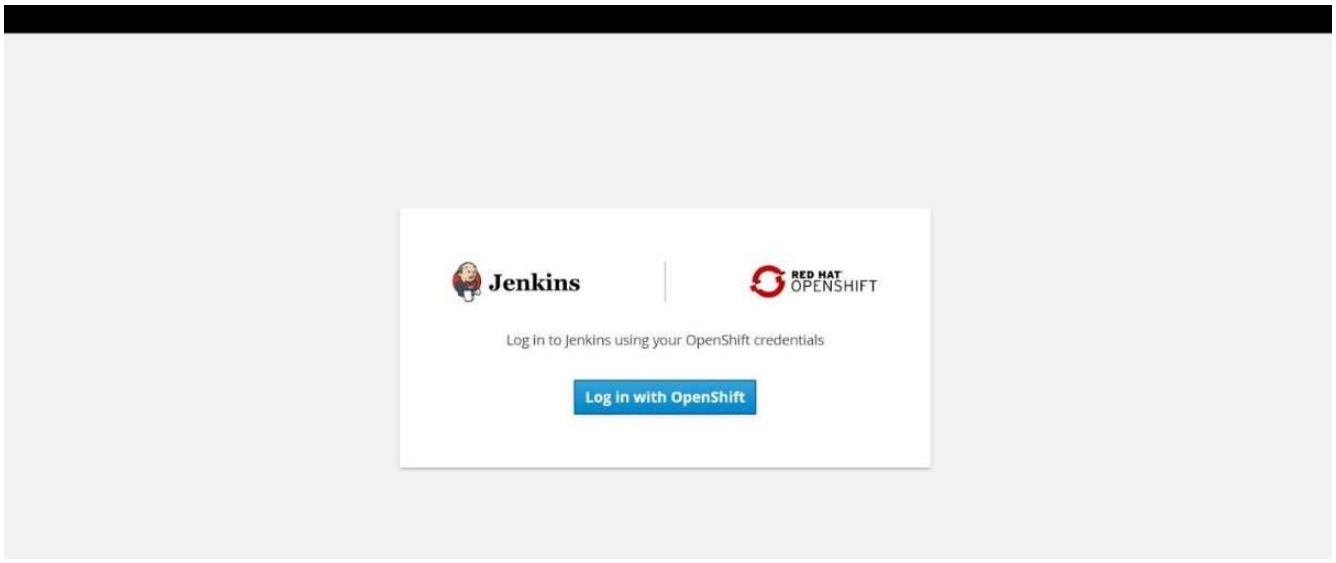
4. I pod Jenkins impiegano circa 10-12 minuti per entrare nello stato Pronto.



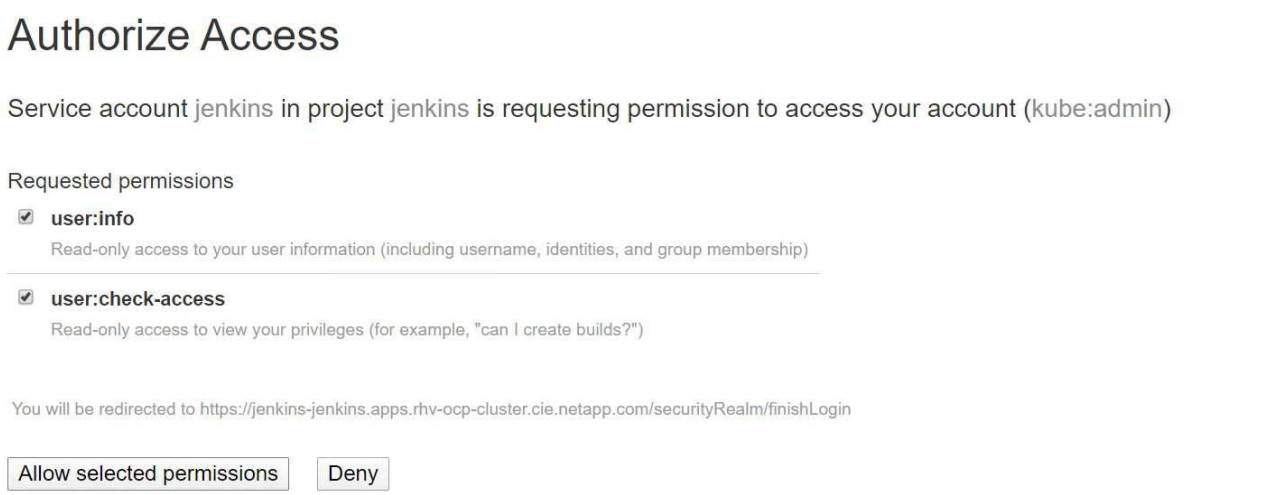
5. Dopo aver creato le istanze dei pod, vai su Networking > Routes. Per aprire la pagina web di Jenkins, fare clic sull'URL fornito per il percorso Jenkins.



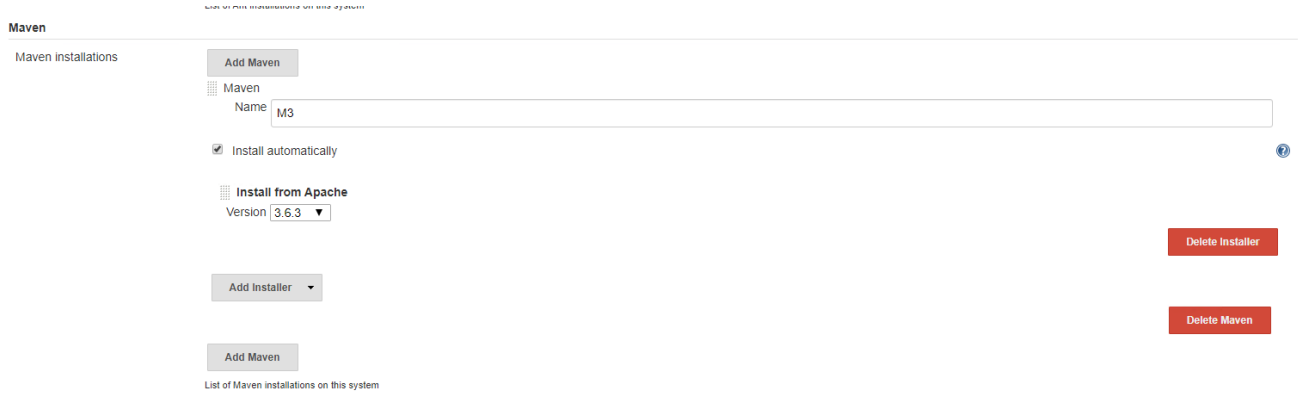
6. Poiché durante la creazione dell'app Jenkins è stato utilizzato OpenShift OAuth, fare clic su Accedi con OpenShift.



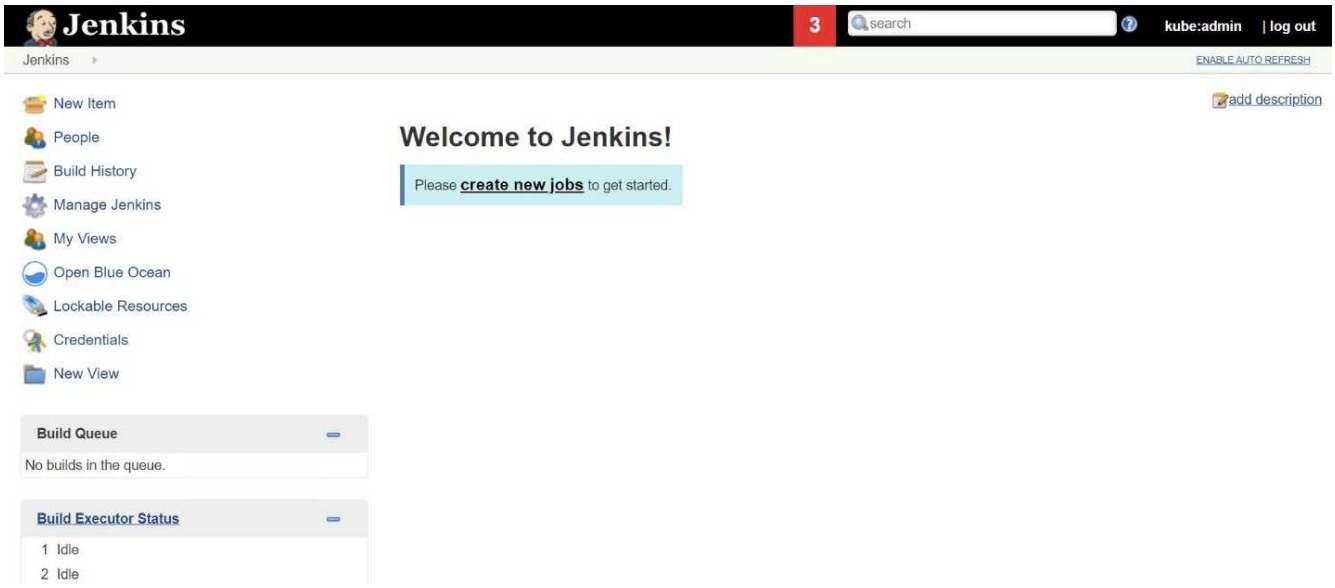
7. Autorizza l'account di servizio Jenkins ad accedere agli utenti OpenShift.



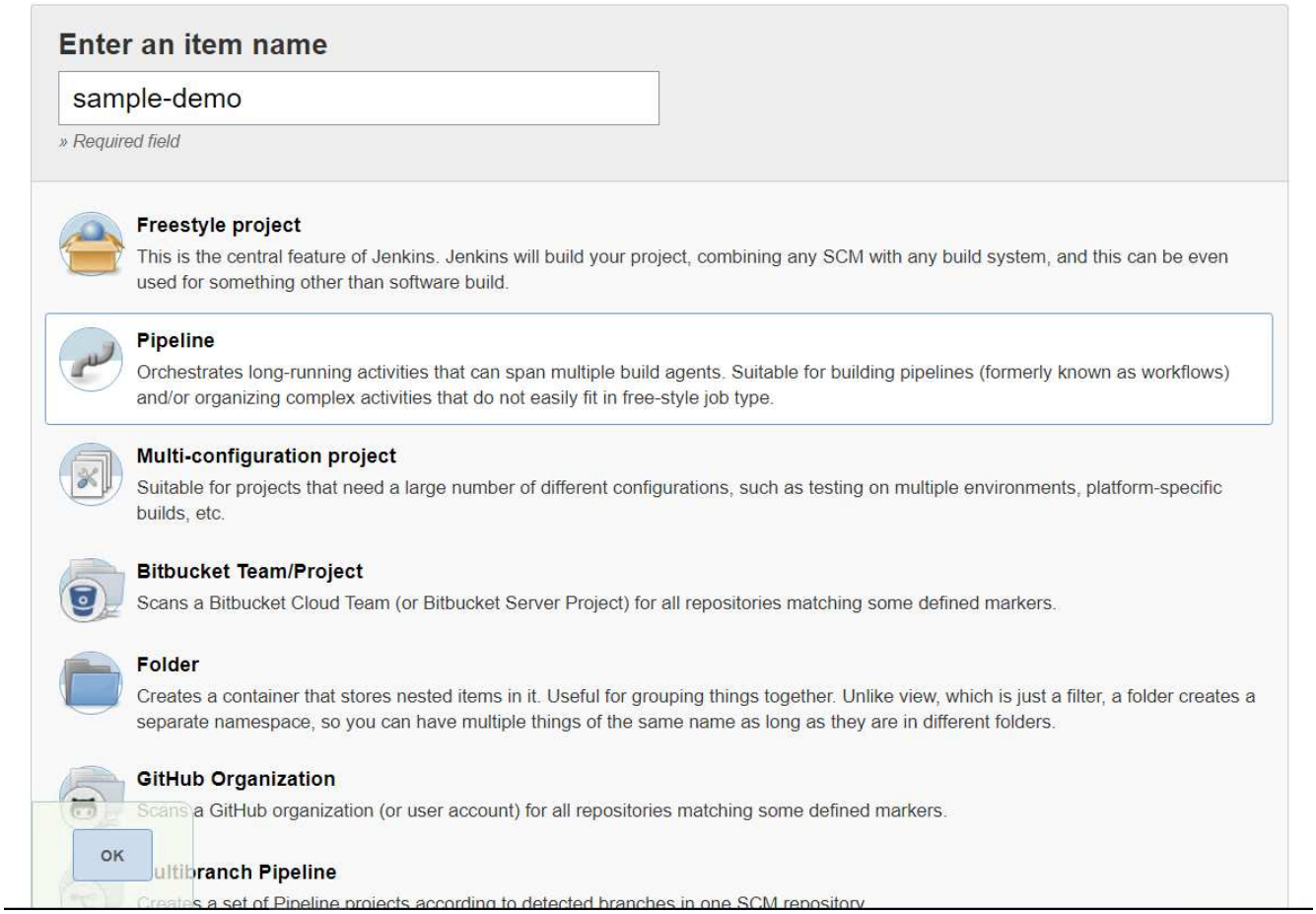
8. Viene visualizzata la pagina di benvenuto di Jenkins. Poiché stiamo utilizzando una build Maven, completiamo prima l'installazione di Maven. Passare a Gestisci Jenkins > Configurazione strumento globale, quindi, nella sottointestazione Maven, fare clic su Aggiungi Maven. Inserisci il nome che preferisci e assicurati che l'opzione Installa automaticamente sia selezionata. Fare clic su Salva.



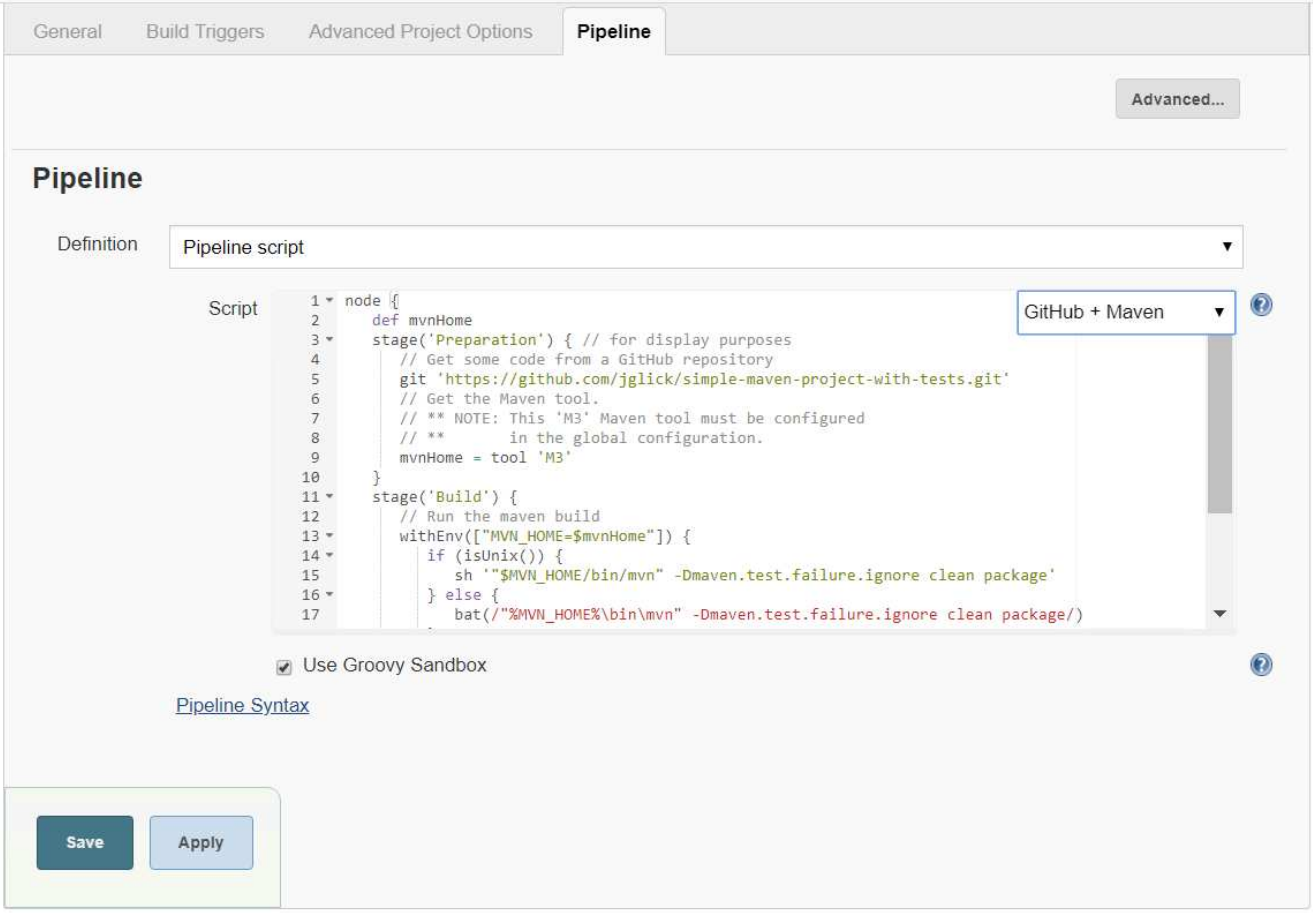
9. Ora puoi creare una pipeline per dimostrare il flusso di lavoro CI/CD. Nella home page, clicca su Crea nuovi lavori o Nuovo elemento dal menu a sinistra.



10. Nella pagina Crea elemento, inserisci il nome che preferisci, seleziona Pipeline e fai clic su OK.



11. Selezionare la scheda Pipeline. Dal menu a discesa Prova pipeline di esempio, seleziona Github + Maven. Il codice viene compilato automaticamente. Fare clic su Salva.



12. Fare clic su Crea ora per avviare lo sviluppo attraverso le fasi di preparazione, creazione e test. Potrebbero essere necessari diversi minuti per completare l'intero processo di compilazione e visualizzarne i risultati.

13. Ogni volta che si verificano modifiche al codice, la pipeline può essere ricostruita per applicare patch alla nuova versione del software, consentendo l'integrazione e la distribuzione continue. Fare clic su Modifiche recenti per tenere traccia delle modifiche rispetto alla versione precedente.

Configurare il multi-tenancy

Configurazione multitenancy su Red Hat OpenShift con NetApp

Molte organizzazioni che eseguono più applicazioni o carichi di lavoro su container tendono a distribuire un cluster Red Hat OpenShift per applicazione o carico di lavoro. Ciò consente loro di implementare un isolamento rigoroso per l'applicazione o il carico di lavoro, ottimizzare le prestazioni e ridurre le vulnerabilità di sicurezza. Tuttavia, l'implementazione di un cluster Red Hat OpenShift separato per ogni applicazione pone una serie di problemi. Aumenta i costi operativi dovuti al monitoraggio e alla gestione di ogni cluster singolarmente, aumenta i costi dovuti alle risorse dedicate per diverse applicazioni e ostacola una scalabilità efficiente.

Per superare questi problemi, si può prendere in considerazione l'esecuzione di tutte le applicazioni o dei carichi di lavoro in un singolo cluster Red Hat OpenShift. Ma in un'architettura di questo tipo, l'isolamento delle risorse e le vulnerabilità della sicurezza delle applicazioni rappresentano una delle sfide più importanti. Qualsiasi vulnerabilità di sicurezza in un carico di lavoro potrebbe naturalmente estendersi a un altro carico di lavoro, aumentando così la zona di impatto. Inoltre, qualsiasi utilizzo improvviso e incontrollato delle risorse da parte di un'applicazione può influire sulle prestazioni di un'altra applicazione, poiché non esiste una politica di allocazione delle risorse predefinita.

Per questo motivo, le organizzazioni cercano soluzioni che raccolgano il meglio da entrambi i mondi, ad esempio consentendo loro di eseguire tutti i carichi di lavoro in un unico cluster e offrendo al contempo i vantaggi di un cluster dedicato per ciascun carico di lavoro.

Una di queste soluzioni efficaci è configurare il multitenancy su Red Hat OpenShift. Il multitenancy è un'architettura che consente a più tenant di coesistere sullo stesso cluster con un adeguato isolamento delle risorse, della sicurezza e così via. In questo contesto, un tenant può essere visto come un sottoinsieme delle risorse del cluster configurate per essere utilizzate da un particolare gruppo di utenti per uno scopo esclusivo. La configurazione multi-tenancy su un cluster Red Hat OpenShift offre i seguenti vantaggi:

- Una riduzione di CapEx e OpEx consentendo la condivisione delle risorse del cluster
- Minori spese generali operative e di gestione
- Proteggere i carichi di lavoro dalla contaminazione incrociata delle violazioni della sicurezza
- Protezione dei carichi di lavoro da un degrado imprevisto delle prestazioni dovuto alla contesa delle risorse

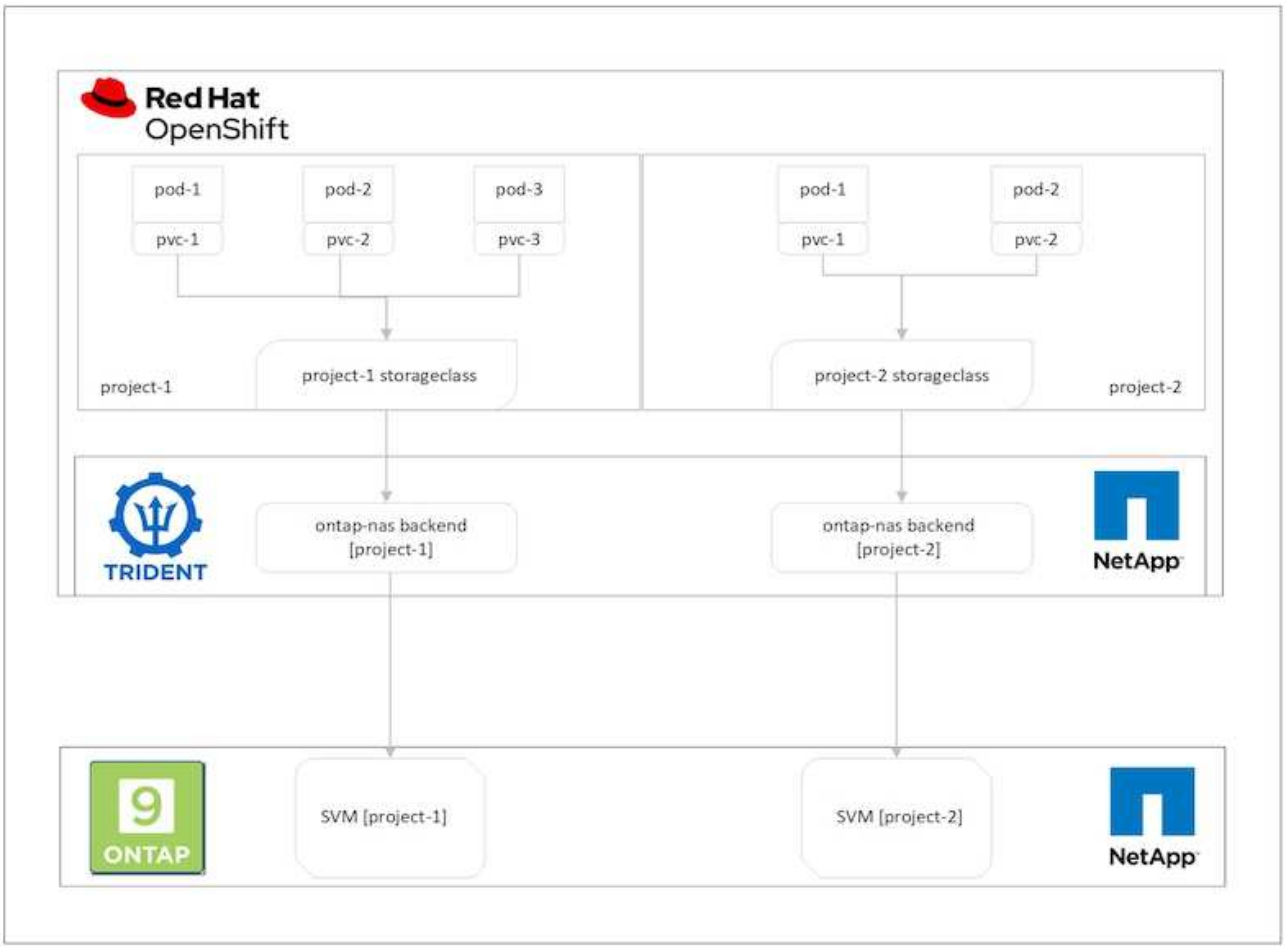
Per un cluster OpenShift multitenant completamente realizzato, è necessario configurare quote e restrizioni per le risorse del cluster appartenenti a diversi bucket di risorse: elaborazione, archiviazione, rete, sicurezza e così via. Sebbene in questa soluzione trattiamo determinati aspetti di tutti i bucket di risorse, ci concentriamo sulle best practice per isolare e proteggere i dati serviti o consumati da più carichi di lavoro sullo stesso cluster Red Hat OpenShift configurando la multitenancy sulle risorse di storage allocate dinamicamente da Trident supportato da NetApp ONTAP.

Architettura

Sebbene Red Hat OpenShift e Trident supportati da NetApp ONTAP non forniscano l'isolamento tra carichi di lavoro per impostazione predefinita, offrono un'ampia gamma di funzionalità che possono essere utilizzate per configurare il multi-tenancy. Per comprendere meglio la progettazione di una soluzione multitenant su un cluster Red Hat OpenShift con Trident supportato da NetApp ONTAP, prendiamo in considerazione un esempio con una serie di requisiti e descriviamo la configurazione relativa.

Supponiamo che un'organizzazione esegua due dei suoi carichi di lavoro su un cluster Red Hat OpenShift come parte di due progetti su cui stanno lavorando due team diversi. I dati per questi carichi di lavoro risiedono su PVC forniti dinamicamente da Trident su un backend NAS NetApp ONTAP . L'organizzazione ha l'esigenza di progettare una soluzione multitenant per questi due carichi di lavoro e di isolare le risorse utilizzate per questi progetti per garantire che la sicurezza e le prestazioni siano mantenute, concentrandosi principalmente sui dati che servono a tali applicazioni.

La figura seguente illustra la soluzione multitenant su un cluster Red Hat OpenShift con Trident supportato da NetApp ONTAP.



Requisiti tecnologici

1. Cluster di archiviazione NetApp ONTAP
2. Cluster Red Hat OpenShift
3. Trident

Red Hat OpenShift – Risorse del cluster

Dal punto di vista del cluster Red Hat OpenShift, la risorsa di primo livello da cui partire è il progetto. Un progetto OpenShift può essere visto come una risorsa cluster che divide l'intero cluster OpenShift in più cluster virtuali. Pertanto, l'isolamento a livello di progetto fornisce una base per la configurazione del multi-tenancy.

Il passo successivo è configurare RBAC nel cluster. La procedura migliore è quella di configurare tutti gli sviluppatori che lavorano su un singolo progetto o carico di lavoro in un singolo gruppo di utenti nell'Identity Provider (IdP). Red Hat OpenShift consente l'integrazione IdP e la sincronizzazione dei gruppi di utenti, consentendo così di importare gli utenti e i gruppi dall'IdP nel cluster. Ciò aiuta gli amministratori del cluster a separare l'accesso alle risorse del cluster dedicate a un progetto a un gruppo di utenti o a gruppi che lavorano su quel progetto, limitando così l'accesso non autorizzato a qualsiasi risorsa del cluster. Per saperne di più sull'integrazione IdP con Red Hat OpenShift, consulta la documentazione "Qui" .

NetApp ONTAP

È importante isolare lo storage condiviso che funge da provider di storage persistente per un cluster Red Hat OpenShift per garantire che i volumi creati sullo storage per ciascun progetto appaiano agli host come se

fossero stati creati su uno storage separato. Per fare ciò, crea tante SVM (macchine virtuali di storage) su NetApp ONTAP quanti sono i progetti o i carichi di lavoro e dedica ogni SVM a un carico di lavoro.

Trident

Dopo aver creato diverse SVM per progetti diversi su NetApp ONTAP, è necessario mappare ciascuna SVM a un diverso backend Trident . La configurazione del backend su Trident gestisce l'allocazione dello storage persistente alle risorse del cluster OpenShift e richiede la mappatura dei dettagli dell'SVM. Questo dovrebbe essere almeno il driver del protocollo per il backend. Facoltativamente, consente di definire come i volumi vengono forniti nello storage e di impostare limiti per le dimensioni dei volumi o l'utilizzo degli aggregati e così via. I dettagli riguardanti la definizione dei backend Trident possono essere trovati "Qui" .

Red Hat OpenShift – risorse di storage

Dopo aver configurato i backend Trident , il passaggio successivo consiste nel configurare StorageClasses. Configurare tante classi di archiviazione quanti sono i backend, consentendo a ciascuna classe di archiviazione di accedere ai volumi solo su un backend. Possiamo mappare StorageClass su un particolare backend Trident utilizzando il parametro storagePools durante la definizione della classe di archiviazione. I dettagli per definire una classe di archiviazione possono essere trovati "Qui" . Pertanto, esiste una mappatura uno a uno da StorageClass al backend Trident che punta a una SVM. Ciò garantisce che tutte le richieste di archiviazione tramite StorageClass assegnate a quel progetto vengano gestite solo dall'SVM dedicato a quel progetto.

Poiché le classi di archiviazione non sono risorse con namespace, come possiamo garantire che le richieste di archiviazione per la classe di archiviazione di un progetto da parte di pod in un altro namespace o progetto vengano rifiutate? La risposta è usare ResourceQuotas. Le ResourceQuota sono oggetti che controllano l'utilizzo totale delle risorse per progetto. Può limitare il numero e la quantità totale di risorse che possono essere consumate dagli oggetti nel progetto. Quasi tutte le risorse di un progetto possono essere limitate utilizzando ResourceQuotas; un utilizzo efficiente di questa funzionalità può aiutare le organizzazioni a ridurre i costi e le interruzioni dovute all'eccessivo approvvigionamento o al consumo eccessivo di risorse. Fare riferimento alla documentazione "Qui" per maggiori informazioni.

Per questo caso d'uso, dobbiamo impedire ai pod di un particolare progetto di richiedere spazio di archiviazione da classi di archiviazione non dedicate al loro progetto. Per fare ciò, dobbiamo limitare le richieste di volume persistente per altre classi di archiviazione impostando `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` a 0. Inoltre, un amministratore di cluster deve garantire che gli sviluppatori di un progetto non abbiano accesso per modificare ResourceQuotas.

Configurazione

Per qualsiasi soluzione multitenant, nessun utente può avere accesso a più risorse del cluster di quelle necessarie. Pertanto, l'intero set di risorse da configurare come parte della configurazione multi-tenancy è suddiviso tra l'amministratore del cluster, l'amministratore dello storage e gli sviluppatori che lavorano su ciascun progetto.

Nella tabella seguente sono illustrate le diverse attività che devono essere svolte dai diversi utenti:

Ruolo	Compiti
Amministratore del cluster	Crea progetti per diverse applicazioni o carichi di lavoro
	Crea ClusterRoles e RoleBindings per storage-admin
	Creare ruoli e RoleBinding per gli sviluppatori che assegnano l'accesso a progetti specifici
	[Facoltativo] Configurare i progetti per pianificare i pod su nodi specifici
Amministratore di archiviazione	Creare SVM su NetApp ONTAP
	Crea backend Trident
	Crea classi di archiviazione
	Crea ResourceQuotas di archiviazione
Sviluppatori	Convalida l'accesso per creare o applicare patch a PVC o pod nel progetto assegnato
	Convalida l'accesso per creare o applicare patch a PVC o pod in un altro progetto
	Convalida l'accesso per visualizzare o modificare Progetti, ResourceQuota e StorageClass

Configurazione

Di seguito sono riportati i prerequisiti per la configurazione di Multitenancy su Red Hat OpenShift con NetApp.

Prerequisiti

- Cluster NetApp ONTAP
- Cluster Red Hat OpenShift
- Trident installato sul cluster
- Postazione di lavoro amministrativa con strumenti tridentctl e oc installati e aggiunti a $PATH
- Accesso amministrativo a ONTAP
- Accesso amministratore del cluster al cluster OpenShift
- Cluster è integrato con Identity Provider
- Il provider di identità è configurato per distinguere in modo efficiente gli utenti nei diversi team

Configurazione: attività di amministrazione del cluster

Le seguenti attività vengono eseguite dall'amministratore del cluster Red Hat OpenShift:

1. Accedi al cluster Red Hat OpenShift come cluster-admin.
2. Crea due progetti corrispondenti a progetti diversi.

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

3. Creare il ruolo di sviluppatore per il progetto-1.

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

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

La definizione del ruolo fornita in questa sezione è solo un esempio. I ruoli degli sviluppatori devono essere definiti in base ai requisiti dell'utente finale.

1. Allo stesso modo, creare ruoli di sviluppatore per il progetto 2.
2. Tutte le risorse di storage OpenShift e NetApp sono solitamente gestite da un amministratore di storage. L'accesso per gli amministratori di storage è controllato dal ruolo di operatore Trident creato al momento dell'installazione Trident . Oltre a ciò, l'amministratore dello storage necessita anche dell'accesso a ResourceQuotas per controllare come viene utilizzato lo storage.
3. Creare un ruolo per la gestione di ResourceQuotas in tutti i progetti del cluster per associarlo all'amministratore di storage.

```
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. Assicurarsi che il cluster sia integrato con il provider di identità dell'organizzazione e che i gruppi di utenti siano sincronizzati con i gruppi del cluster. L'esempio seguente mostra che il provider di identità è stato integrato con il cluster e sincronizzato con i gruppi di utenti.

```
$ 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. Configurare ClusterRoleBindings per gli amministratori di storage.

```
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
```

Per gli amministratori di storage è necessario associare due ruoli: trident-operator e resource-quotas.

1. Creare RoleBindings per gli sviluppatori che associano il ruolo developer-project-1 al gruppo corrispondente (ocp-project-1) in project-1.

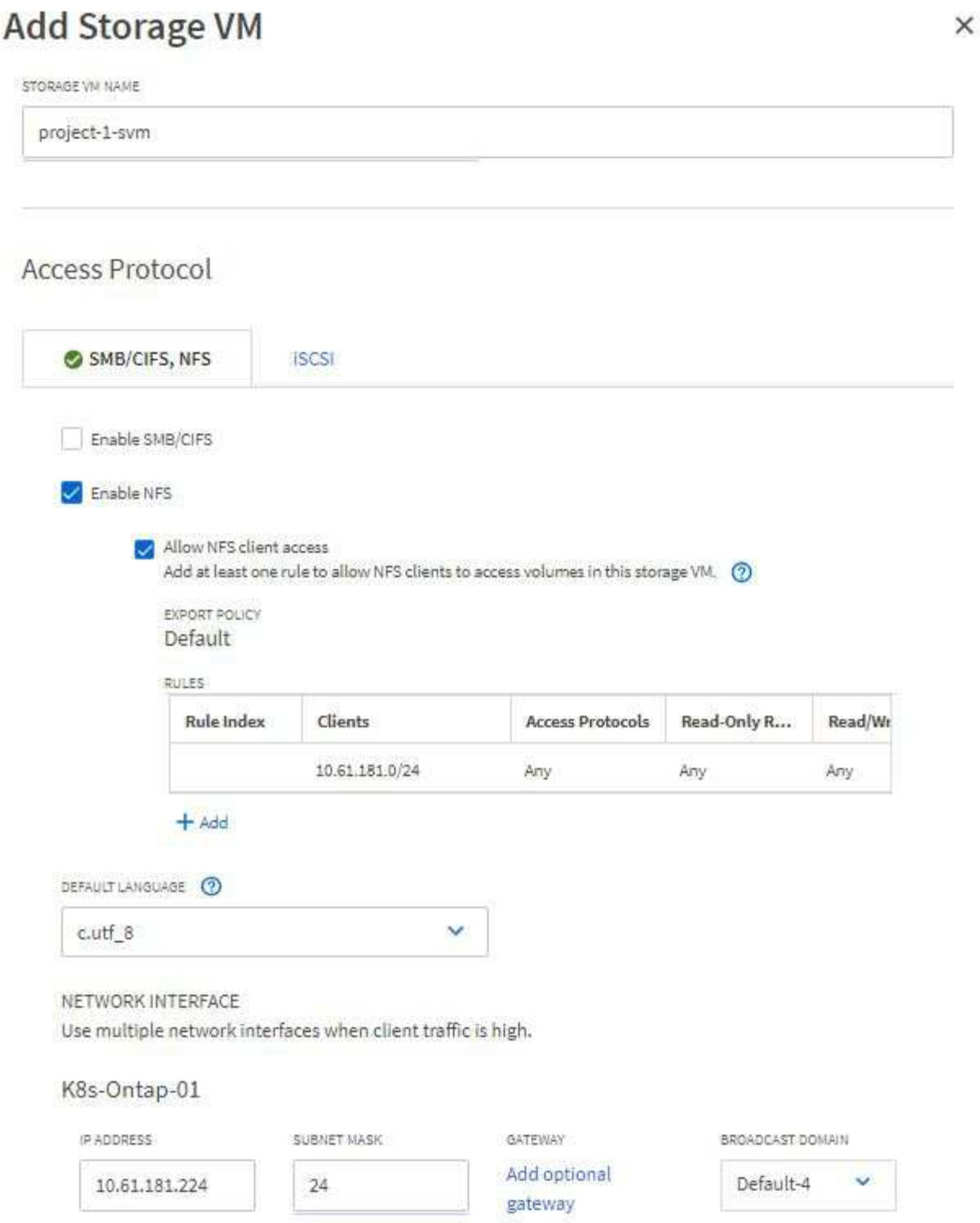
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role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

2. Allo stesso modo, creare RoleBindings per gli sviluppatori che associano i ruoli degli sviluppatori al gruppo di utenti corrispondente nel progetto 2.

Configurazione: Attività di amministrazione dell'archiviazione

Le seguenti risorse devono essere configurate da un amministratore di storage:

1. Accedi al cluster NetApp ONTAP come amministratore.
2. Vai su Archiviazione > VM di archiviazione e fai clic su Aggiungi. Creare due SVM, una per il progetto 1 e l'altra per il progetto 2, fornendo i dettagli richiesti. Creare anche un account vsadmin per gestire l'SVM e le sue risorse.



1. Accedi al cluster Red Hat OpenShift come amministratore dello storage.
2. Creare il backend per il progetto-1 e mapparlo all'SVM dedicato al progetto. NetApp consiglia di utilizzare l'account vsadmin dell'SVM per connettere il backend all'SVM anziché utilizzare l'amministratore del 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
```

Per questo esempio utilizziamo il driver ontap-nas. Utilizzare il driver appropriato durante la creazione del backend in base al caso d'uso.

Supponiamo che Trident sia installato nel progetto Trident.

1. Allo stesso modo, creare il backend Trident per il progetto 2 e mapparlo all'SVM dedicato al progetto 2.
2. Successivamente, creare le classi di archiviazione. Creare la classe di archiviazione per il progetto 1 e configurarla per utilizzare i pool di archiviazione dal backend dedicato al progetto 1 impostando il parametro 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. Allo stesso modo, creare una classe di archiviazione per il progetto 2 e configurarla per utilizzare i pool di archiviazione dal backend dedicato al progetto 2.
4. Creare una ResourceQuota per limitare le risorse nel progetto 1 che richiedono spazio di archiviazione da storageclass dedicate ad altri progetti.

```
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. Allo stesso modo, creare una ResourceQuota per limitare le risorse nel progetto 2 che richiedono spazio di archiviazione da storageclass dedicate ad altri progetti.

Validazione

Per convalidare l'architettura multitenant configurata nei passaggi precedenti, completare i seguenti passaggi:

Convalida l'accesso per creare PVC o pod nel progetto assegnato

1. Accedi come ocp-project-1-user, sviluppatore nel progetto-1.
2. Controlla l'accesso per creare un nuovo progetto.

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

3. Creare un PVC nel progetto-1 utilizzando la classe di archiviazione assegnata al progetto-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. Controllare il PV associato al PVC.

```
oc get pv
```

5. Verificare che il PV e il suo volume siano creati in una SVM dedicata al progetto 1 su NetApp ONTAP.

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

6. Crea un pod nel progetto 1 e monta il PVC creato nel passaggio precedente.

```
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. Controllare se il pod è in esecuzione e se ha montato il volume.

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

Convalida l'accesso per creare PVC o pod in un altro progetto o utilizzare risorse dedicate a un altro progetto

1. Accedi come ocp-project-1-user, sviluppatore nel progetto-1.
2. Creare un PVC nel progetto-1 utilizzando la classe di archiviazione assegnata al progetto-2.

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

3. Crea un PVC nel progetto-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. Assicurarsi che i PVC `test-pvc-project-1-sc-2` E `test-pvc-project-2-sc-1` non sono stati creati.

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

5. Crea un pod nel progetto-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
```

Convalida l'accesso per visualizzare e modificare Progetti, ResourceQuota e StorageClass

1. Accedi come ocp-project-1-user, sviluppatore nel progetto-1.
2. Controlla l'accesso per creare nuovi progetti.

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

3. Convalida l'accesso per visualizzare i progetti.

```
oc get ns
```

4. Verificare se l'utente può visualizzare o modificare ResourceQuotas nel progetto-1.

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

5. Verificare che l'utente abbia accesso per visualizzare le classi di archiviazione.

```
oc get sc
```

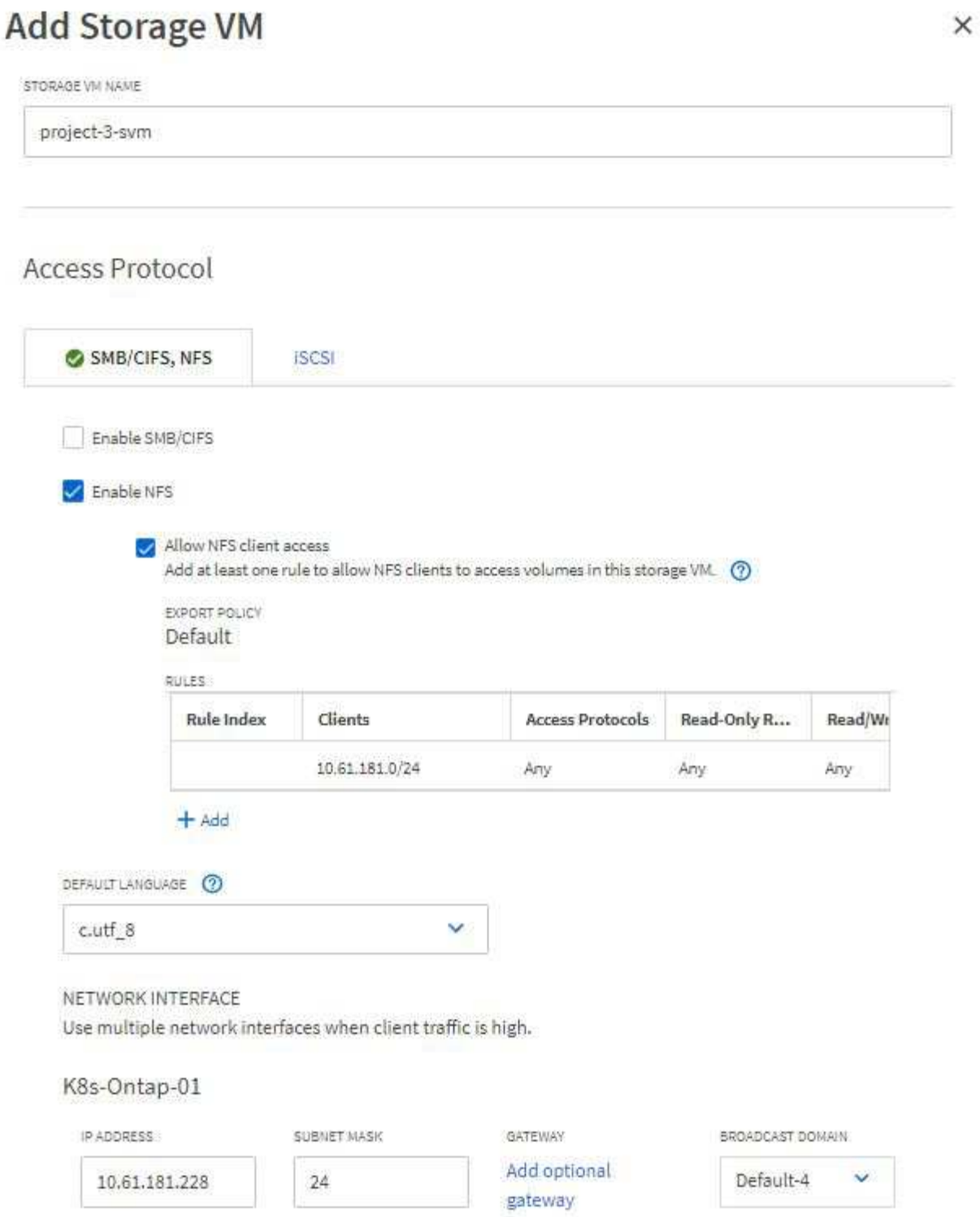
6. Controllare l'accesso per descrivere le classi di archiviazione.
7. Convalida l'accesso dell'utente per modificare le classi di archiviazione.

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

Scalabilità: aggiunta di più progetti

In una configurazione multi-tenant, l'aggiunta di nuovi progetti con risorse di storage richiede una configurazione aggiuntiva per garantire che il multi-tenant non venga violato. Per aggiungere altri progetti in un cluster multitenant, completare i seguenti passaggi:

1. Accedi al cluster NetApp ONTAP come amministratore dello storage.
2. Vai a `Storage → Storage VMs` e clicca `Add` . Creare un nuovo SVM dedicato al progetto 3. Creare anche un account vsadmin per gestire l'SVM e le sue risorse.



1. Accedi al cluster Red Hat OpenShift come amministratore del cluster.
2. Crea un nuovo progetto.

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

3. Assicurarsi che il gruppo utenti per il progetto 3 sia creato su IdP e sincronizzato con il cluster OpenShift.

```
oc get groups
```

4. Creare il ruolo di sviluppatore per il progetto 3.

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

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

La definizione del ruolo fornita in questa sezione è solo un esempio. Il ruolo dello sviluppatore deve essere definito in base ai requisiti dell'utente finale.

1. Crea RoleBinding per gli sviluppatori nel progetto 3, associando il ruolo developer-project-3 al gruppo corrispondente (ocp-project-3) nel progetto 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. Accedi al cluster Red Hat OpenShift come amministratore dello storage
3. Creare un backend Trident e mapparlo sull'SVM dedicato al progetto 3. NetApp consiglia di utilizzare l'account vsadmin dell'SVM per connettere il backend all'SVM anziché utilizzare l'amministratore del 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
```

Per questo esempio utilizziamo il driver ontap-nas. Utilizzare il driver appropriato per creare il backend in base al caso d'uso.

Supponiamo che Trident sia installato nel progetto Trident.

1. Creare la classe di archiviazione per il progetto 3 e configurarla per utilizzare i pool di archiviazione dal backend dedicato al progetto 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. Creare una ResourceQuota per limitare le risorse nel progetto 3 che richiedono spazio di archiviazione da storageclass dedicate ad altri progetti.

```
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. Applicare una patch a ResourceQuotas in altri progetti per impedire alle risorse di tali progetti di accedere allo storage dalla classe storageclass dedicata al progetto 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}}}'
```

Gestione avanzata dei cluster per Kubernetes

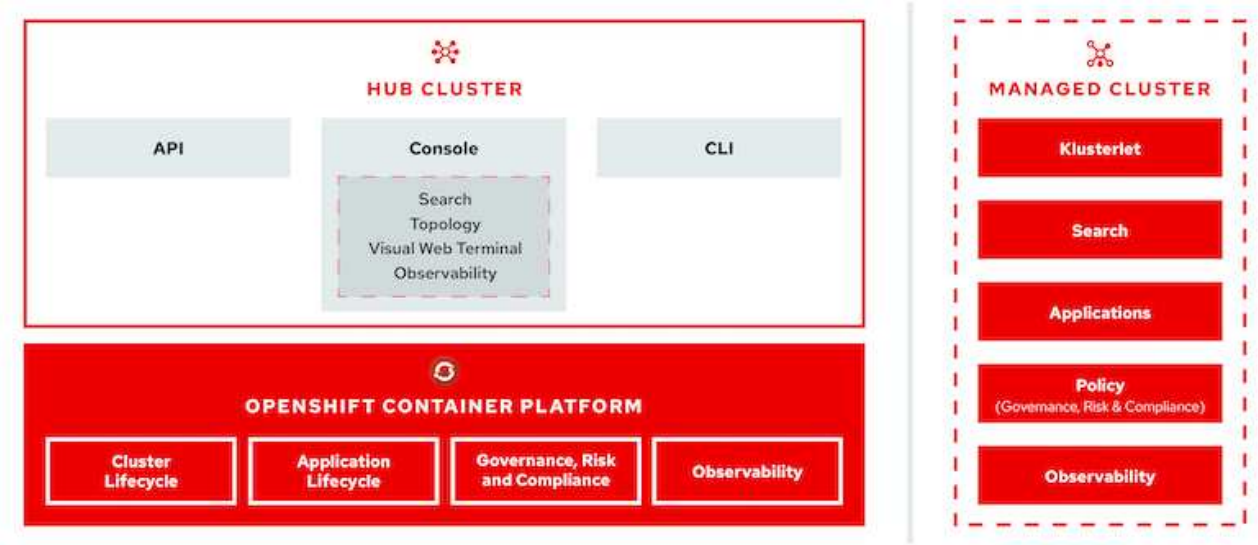
Gestione avanzata dei cluster per Kubernetes: Red Hat OpenShift con NetApp - Panoramica

Quando un'applicazione containerizzata passa dallo sviluppo alla produzione, molte organizzazioni necessitano di più cluster Red Hat OpenShift per supportare il test e la distribuzione di tale applicazione. Parallelamente, le organizzazioni solitamente ospitano più applicazioni o carichi di lavoro su cluster OpenShift. Pertanto, ogni organizzazione finisce per gestire un set di cluster e gli amministratori di OpenShift devono affrontare la sfida aggiuntiva di gestire e mantenere più cluster in una gamma di ambienti che si estendono su più data center locali e cloud pubblici. Per affrontare queste sfide, Red Hat ha introdotto Advanced Cluster Management per Kubernetes.

Red Hat Advanced Cluster Management per Kubernetes consente di eseguire le seguenti attività:

1. Crea, importa e gestisci più cluster su data center e cloud pubblici
2. Distribuisci e gestisci applicazioni o carichi di lavoro su più cluster da un'unica console
3. Monitorare e analizzare lo stato e l'integrità delle diverse risorse del cluster
4. Monitorare e applicare la conformità alla sicurezza su più cluster

Red Hat Advanced Cluster Management for Kubernetes viene installato come componente aggiuntivo su un cluster Red Hat OpenShift e utilizza questo cluster come controller centrale per tutte le sue operazioni. Questo cluster è noto come cluster hub ed espone un piano di gestione affinché gli utenti possano connettersi ad Advanced Cluster Management. Tutti gli altri cluster OpenShift importati o creati tramite la console Advanced Cluster Management vengono gestiti dal cluster hub e sono denominati cluster gestiti. Installa un agente denominato Klusterlet sui cluster gestiti per connetterli al cluster hub e soddisfare le richieste per diverse attività correlate alla gestione del ciclo di vita del cluster, alla gestione del ciclo di vita delle applicazioni, all'osservabilità e alla conformità alla sicurezza.



Per maggiori informazioni, consultare la documentazione "Qui" .

Distribuisci ACM per Kubernetes

Distribuisci Advanced Cluster Management per Kubernetes

Questa sezione riguarda la gestione avanzata dei cluster per Kubernetes su Red Hat OpenShift con NetApp.

Prerequisiti

1. Un cluster Red Hat OpenShift (versione superiore alla 4.5) per il cluster hub
2. Cluster Red Hat OpenShift (superiori alla versione 4.4.3) per cluster gestiti
3. Accesso Cluster-admin al cluster Red Hat OpenShift
4. Un abbonamento Red Hat per Advanced Cluster Management per Kubernetes

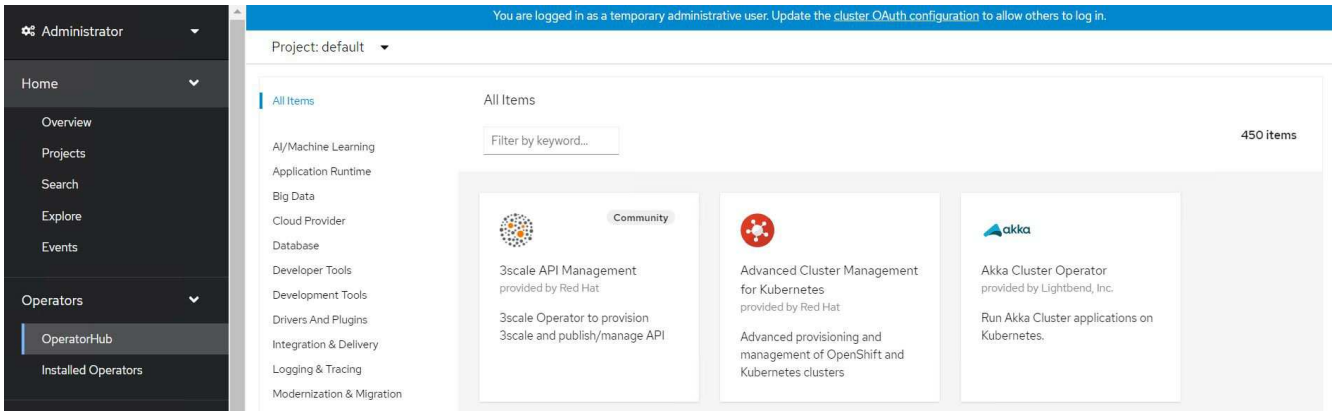
Advanced Cluster Management è un componente aggiuntivo per il cluster OpenShift, pertanto sono previsti determinati requisiti e restrizioni sulle risorse hardware in base alle funzionalità utilizzate nell'hub e nei cluster gestiti. Quando si dimensionano i cluster, è necessario tenere conto di questi aspetti. Vedi la documentazione "Qui" per maggiori dettagli.

Facoltativamente, se il cluster hub ha nodi dedicati per ospitare i componenti dell'infrastruttura e si desidera installare le risorse di Advanced Cluster Management solo su tali nodi, è necessario aggiungere tolleranze e selettori a tali nodi di conseguenza. Per maggiori dettagli, vedere la documentazione "Qui" .

Distribuisci Advanced Cluster Management per Kubernetes

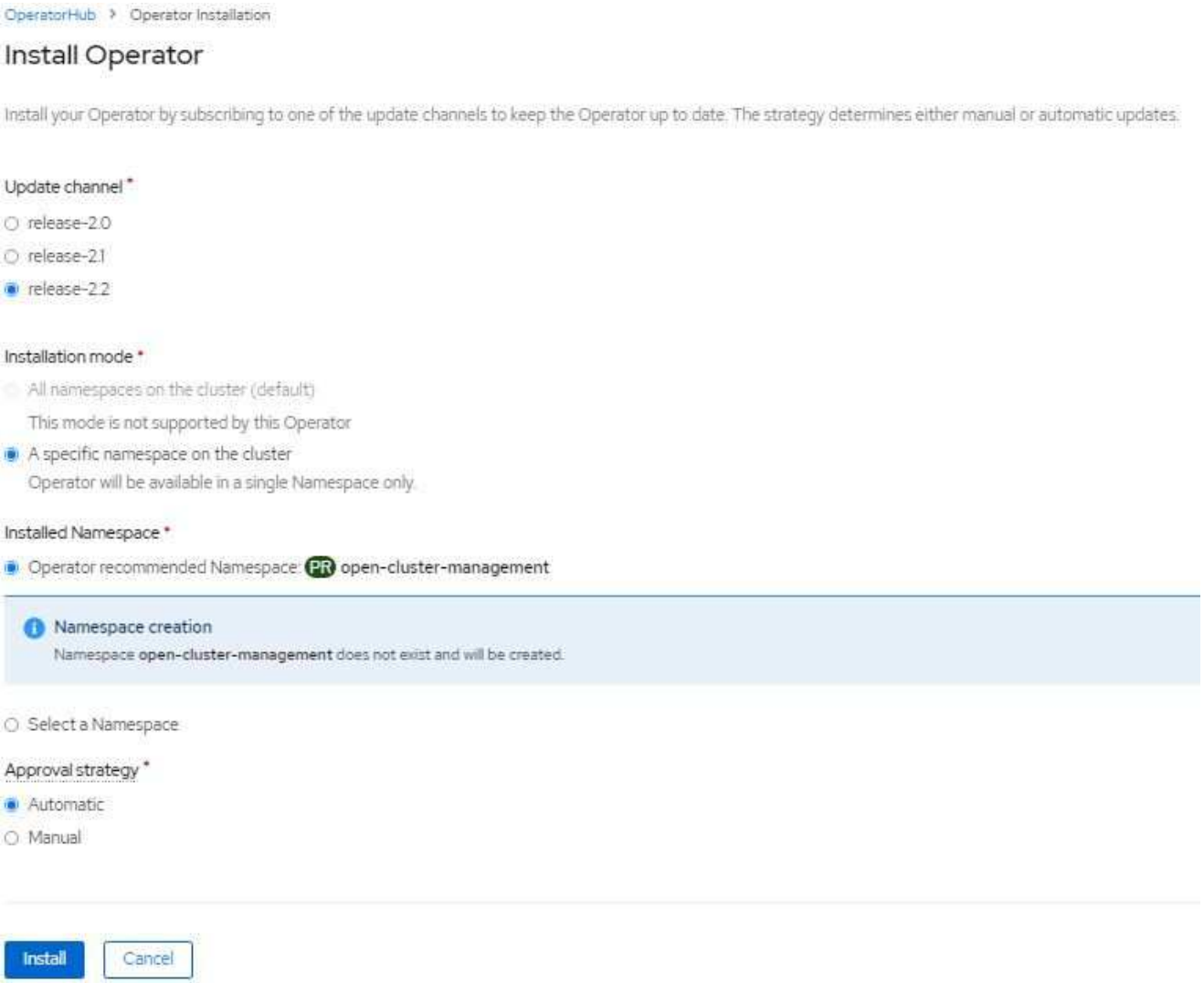
Per installare Advanced Cluster Management per Kubernetes su un cluster OpenShift, completare i seguenti passaggi:

1. Scegli un cluster OpenShift come cluster hub e accedi con privilegi di amministratore del cluster.
2. Vai su Operatori > Hub operatori e cerca Advanced Cluster Management per Kubernetes.

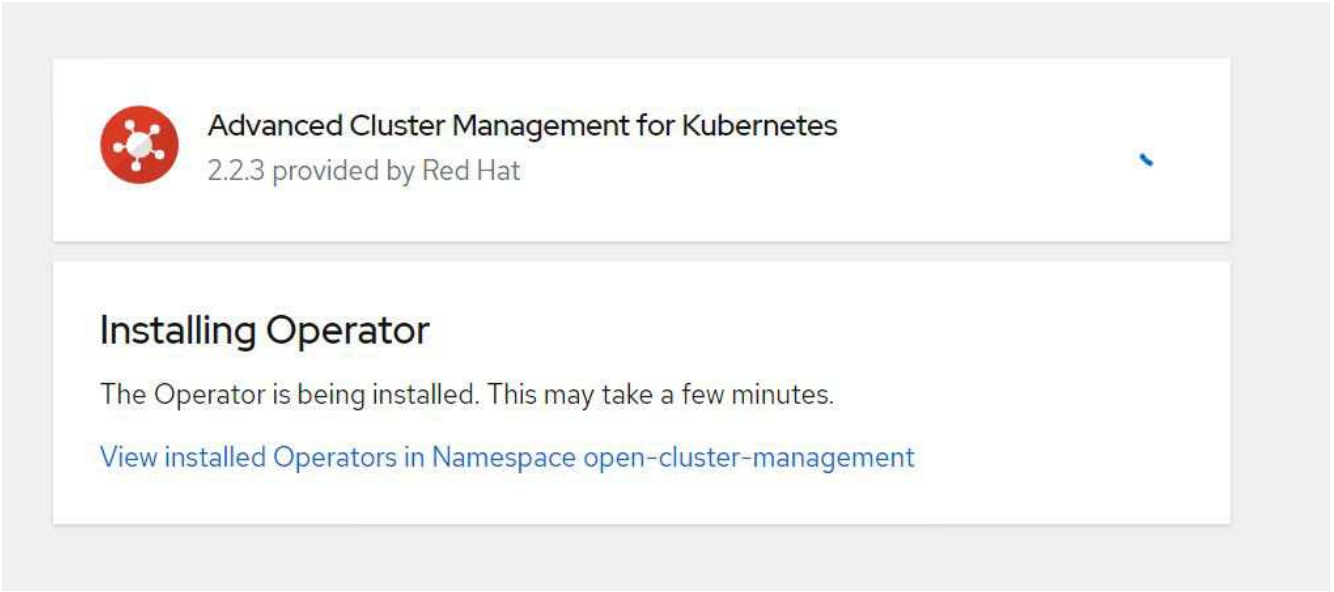


3. Selezionare Advanced Cluster Management per Kubernetes e fare clic su Installa.

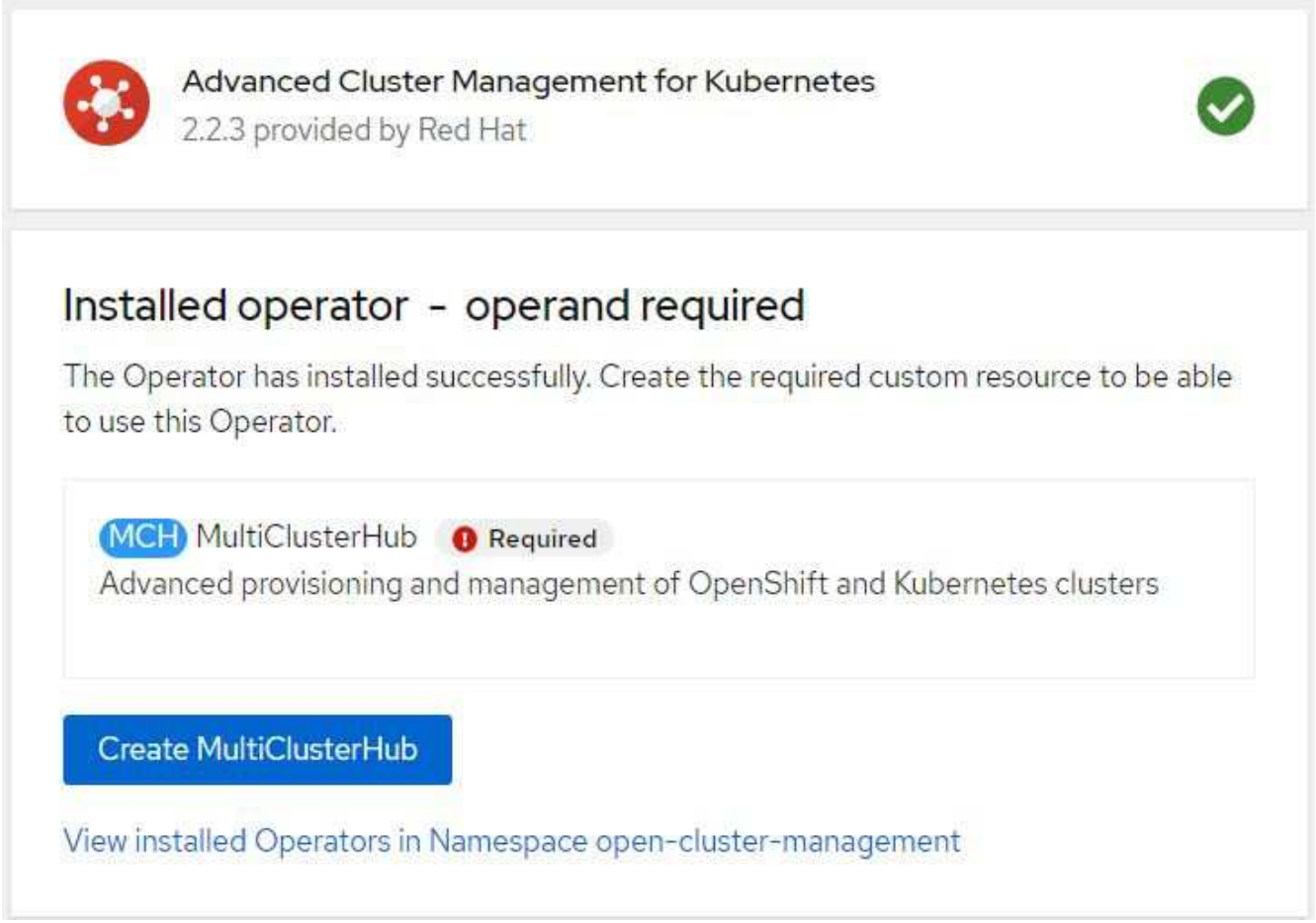
4. Nella schermata Installa operatore, fornire i dettagli necessari (NetApp consiglia di mantenere i parametri predefiniti) e fare clic su Installa.



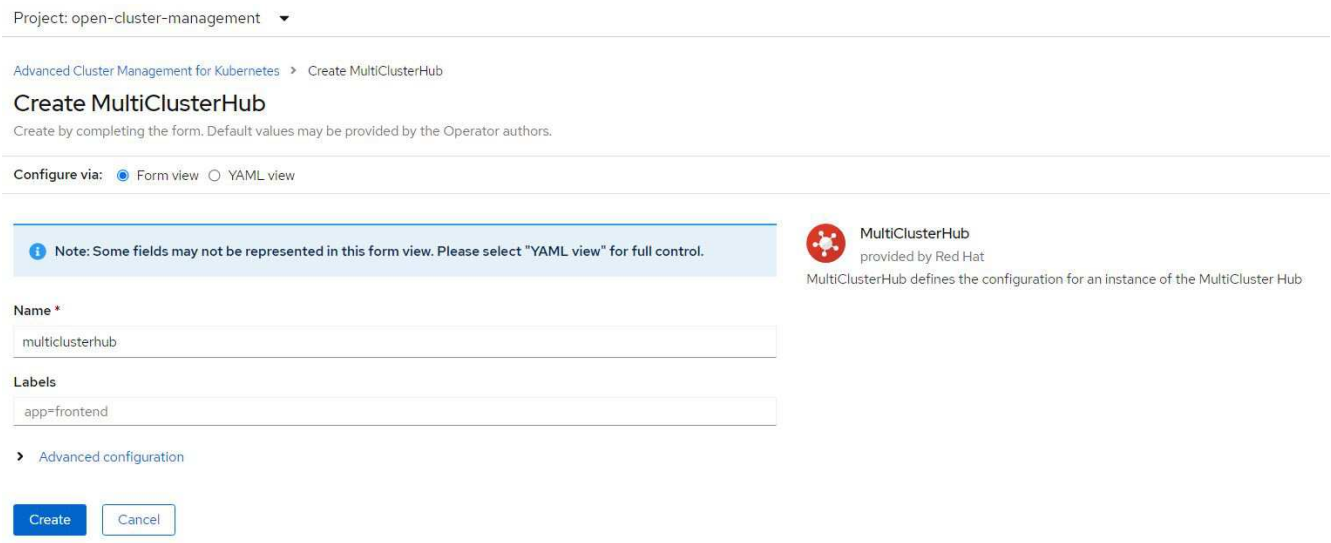
5. Attendere il completamento dell'installazione dell'operatore.



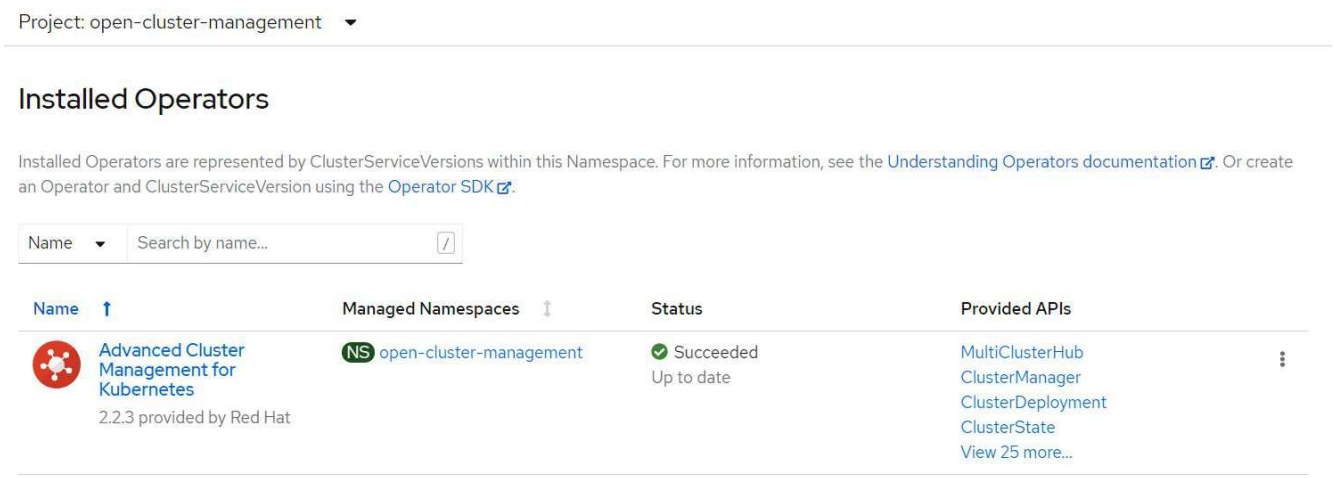
6. Dopo aver installato l'operatore, fare clic su Crea MultiClusterHub.



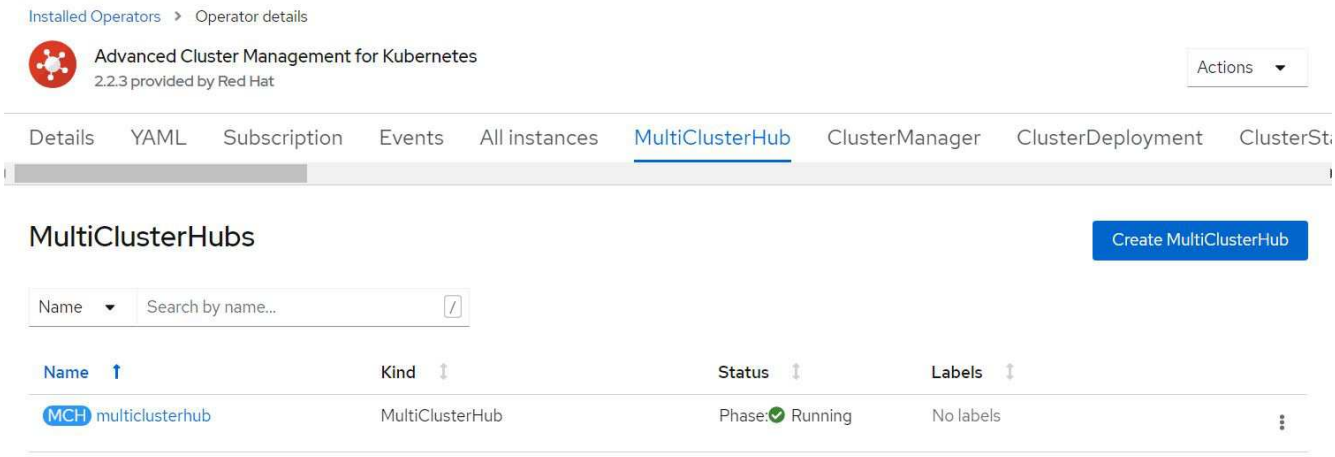
7. Nella schermata Crea MultiClusterHub, fare clic su Crea dopo aver fornito i dettagli. Ciò avvia l'installazione di un hub multi-cluster.



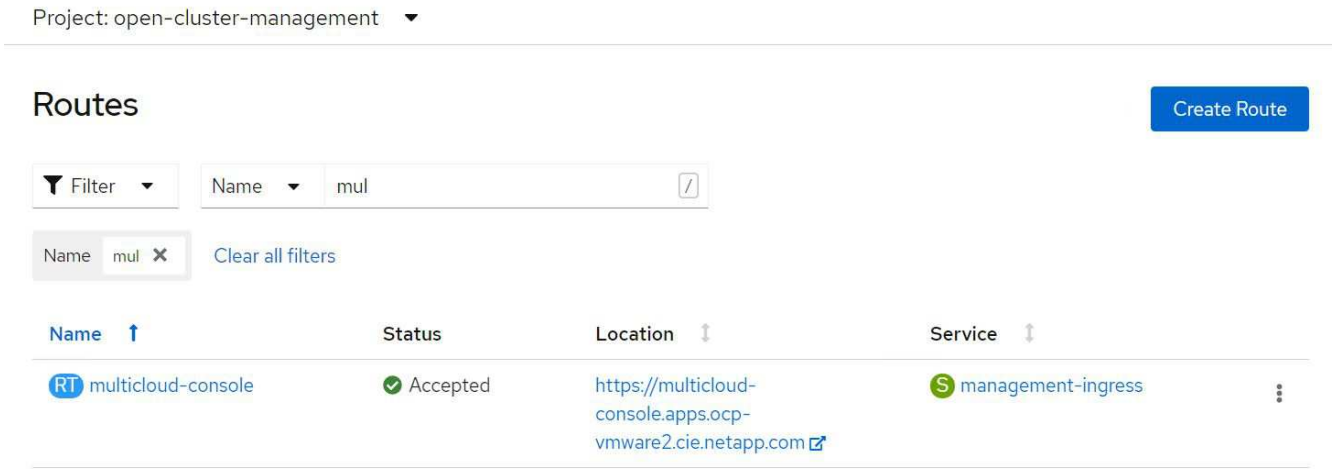
8. Dopo che tutti i pod passano allo stato In esecuzione nello spazio dei nomi open-cluster-management e l'operatore passa allo stato Riuscito, Advanced Cluster Management per Kubernetes viene installato.



9. Ci vuole un po' di tempo per completare l'installazione dell'hub e, una volta completata, l'hub MultiCluster passa allo stato In esecuzione.



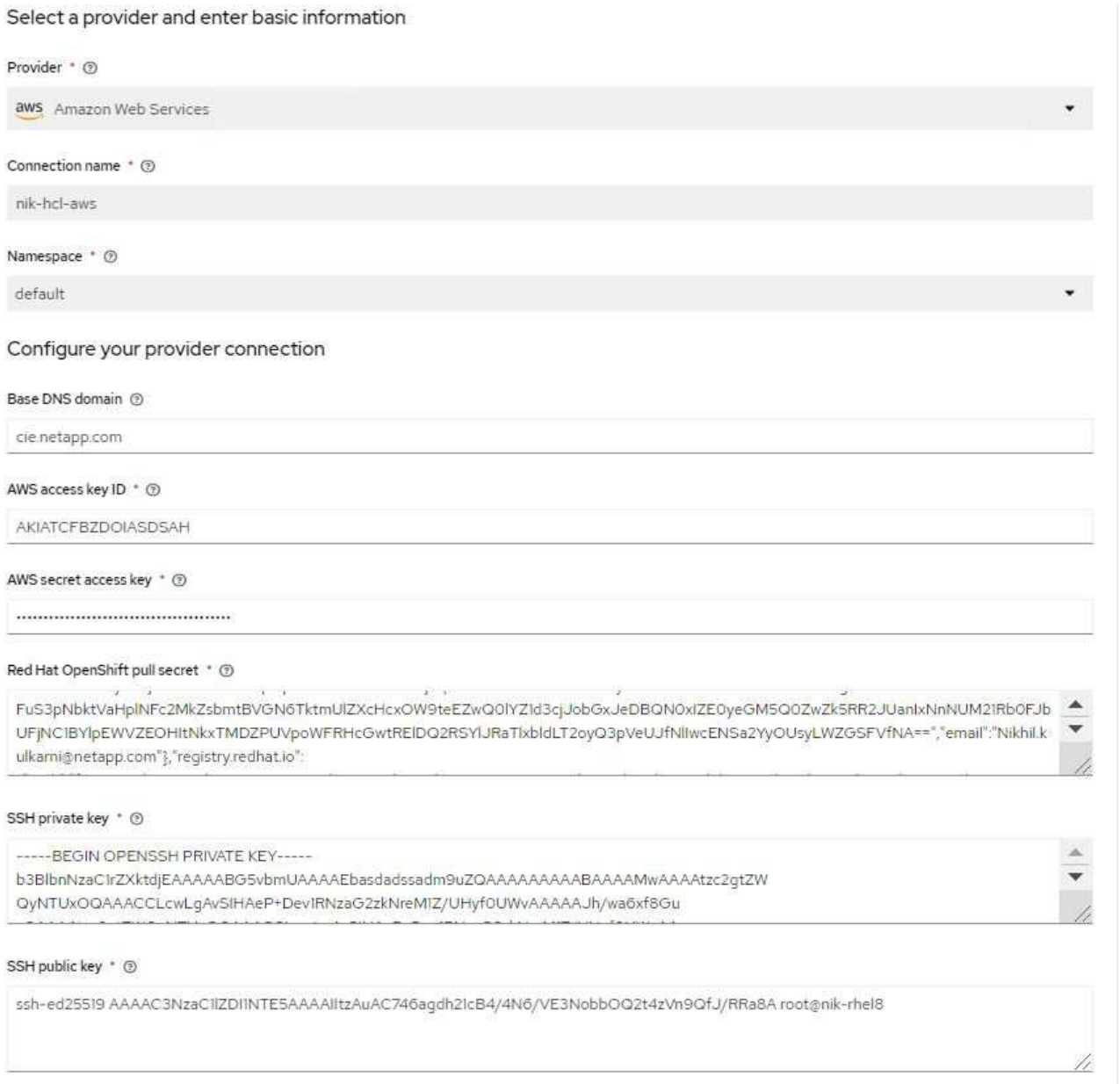
10. Crea un percorso nello spazio dei nomi open-cluster-management. Connettersi all'URL nel percorso per accedere alla console Advanced Cluster Management.



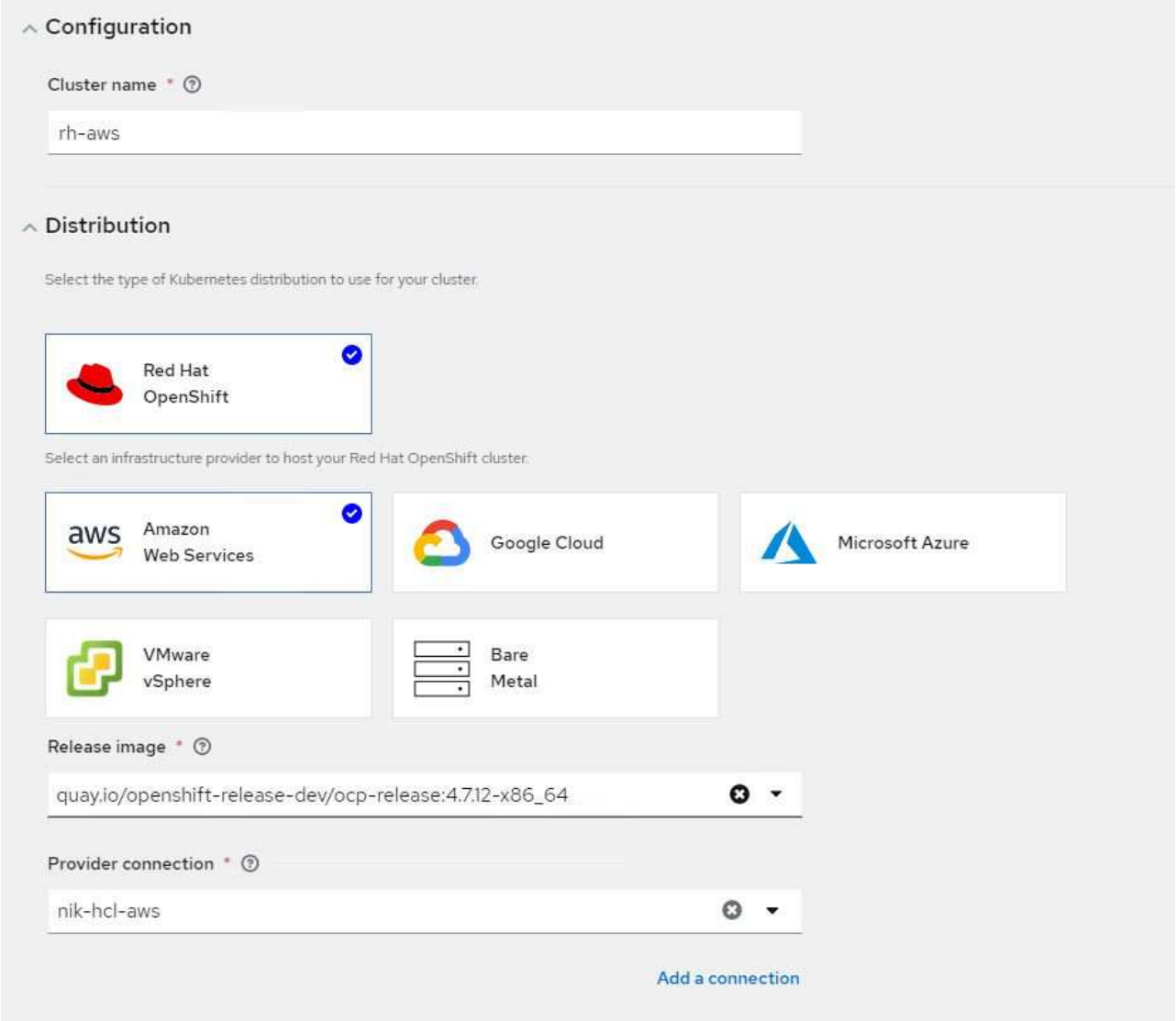
Gestione del ciclo di vita del cluster

Per gestire diversi cluster OpenShift, è possibile crearli o importarli in Advanced Cluster Management.

1. Per prima cosa vai su Automatizza infrastrutture > Cluster.
2. Per creare un nuovo cluster OpenShift, completare i seguenti passaggi:
   a. Creare una connessione al provider: andare su Connessioni provider e fare clic su Aggiungi una connessione, fornire tutti i dettagli corrispondenti al tipo di provider selezionato e fare clic su Aggiungi.

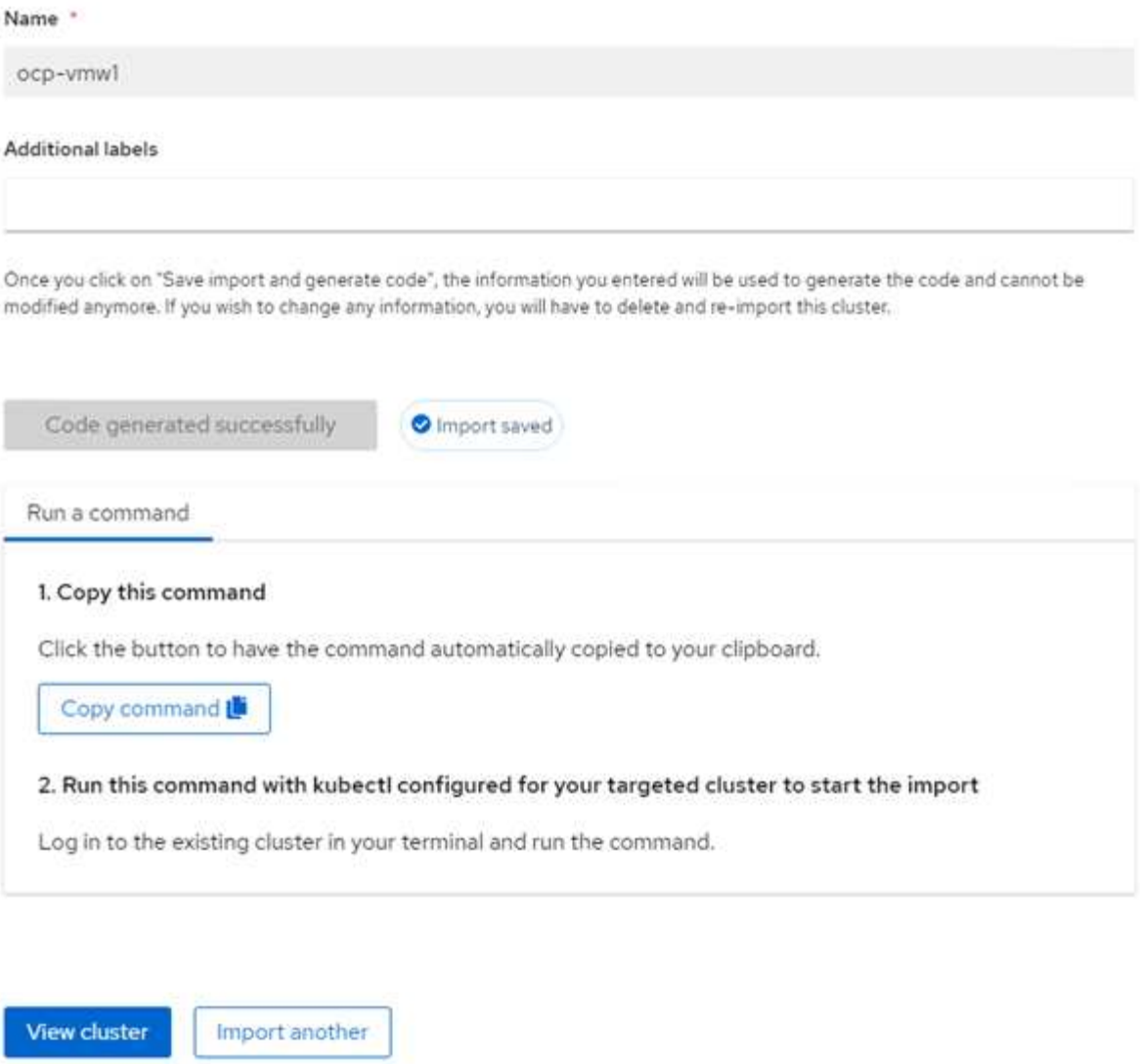


   b. Per creare un nuovo cluster, vai su Cluster e fai clic su Aggiungi un cluster > Crea un cluster. Fornire i dettagli per il cluster e il provider corrispondente e fare clic su Crea.



c. Dopo la creazione, il cluster viene visualizzato nell'elenco dei cluster con lo stato Pronto.

3. Per importare un cluster esistente, completare i seguenti passaggi:
   a. Vai a Cluster e fai clic su Aggiungi un cluster > Importa un cluster esistente.
   b. Inserisci il nome del cluster e fai clic su Salva importazione e genera codice. Viene visualizzato un comando per aggiungere il cluster esistente.
   c. Fare clic su Copia comando ed eseguire il comando sul cluster da aggiungere al cluster hub. In questo modo viene avviata l'installazione degli agenti necessari sul cluster e, una volta completato il processo, il cluster viene visualizzato nell'elenco dei cluster con lo stato Pronto.

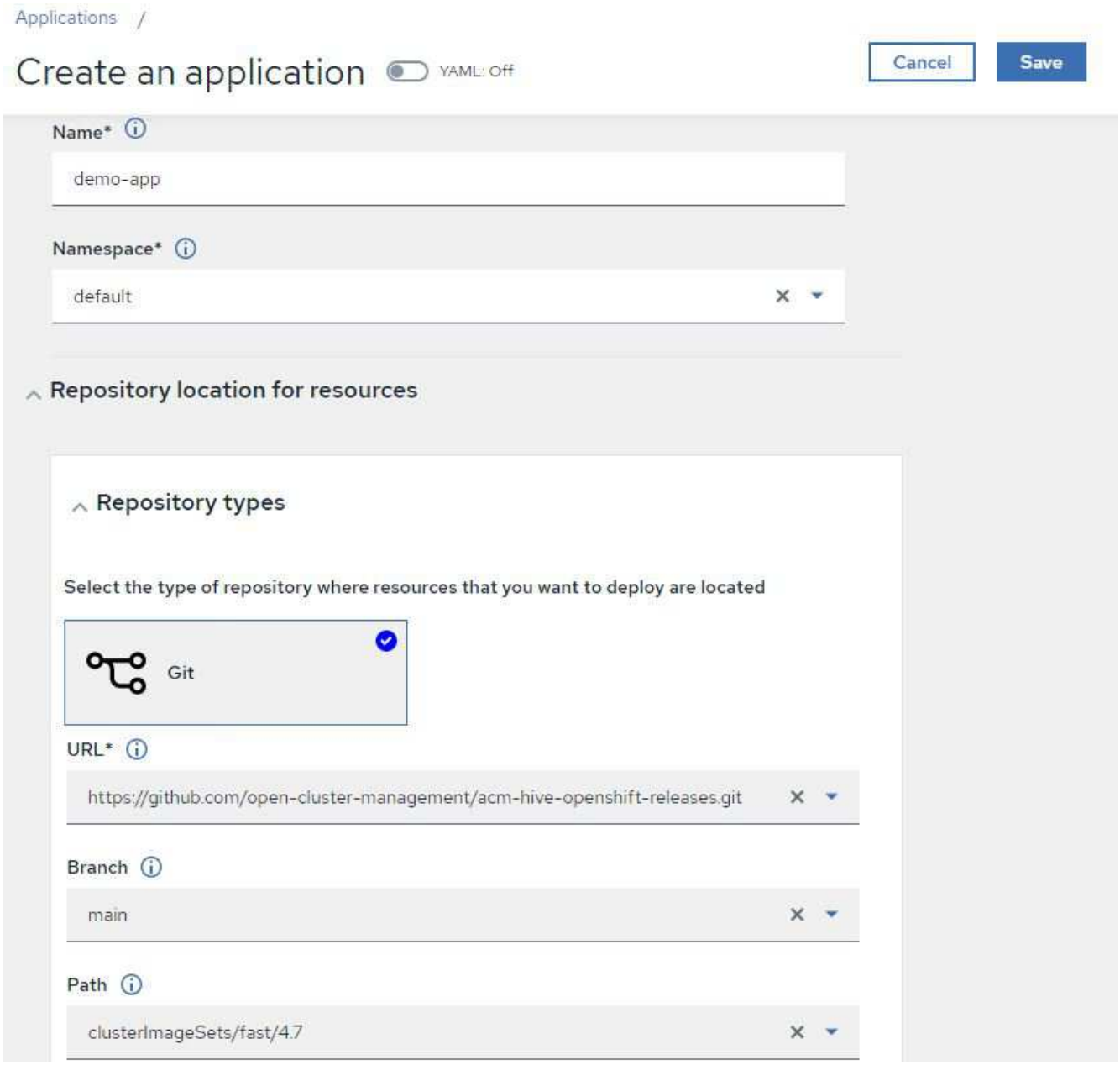


4. Dopo aver creato e importato più cluster, puoi monitorarli e gestirli da un'unica console.

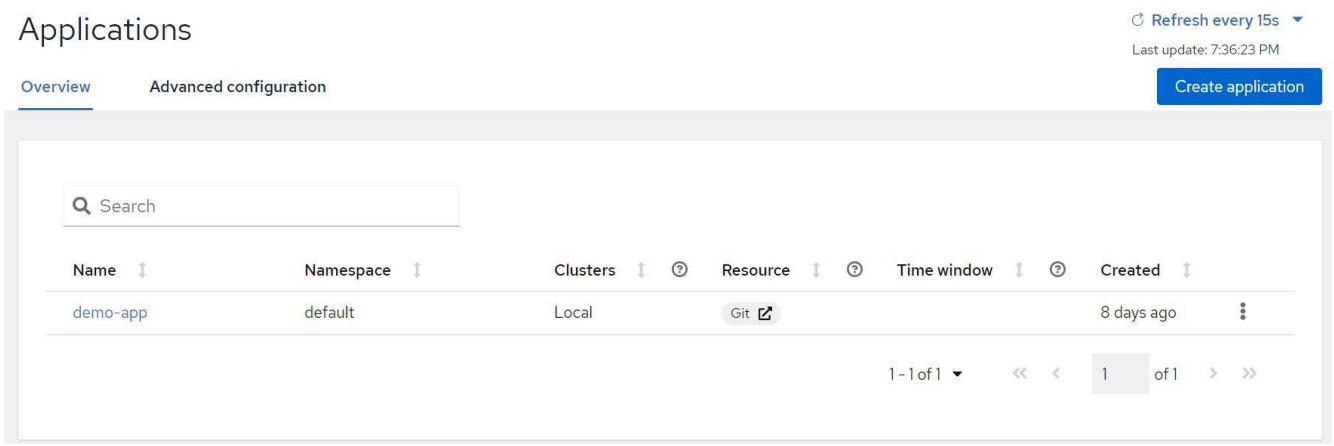
Gestione del ciclo di vita delle applicazioni

Per creare un'applicazione e gestirla su un insieme di cluster,

1. Vai su Gestisci applicazioni dalla barra laterale e fai clic su Crea applicazione. Fornisci i dettagli dell'applicazione che desideri creare e fai clic su Salva.



2. Dopo aver installato i componenti dell'applicazione, l'applicazione viene visualizzata nell'elenco.

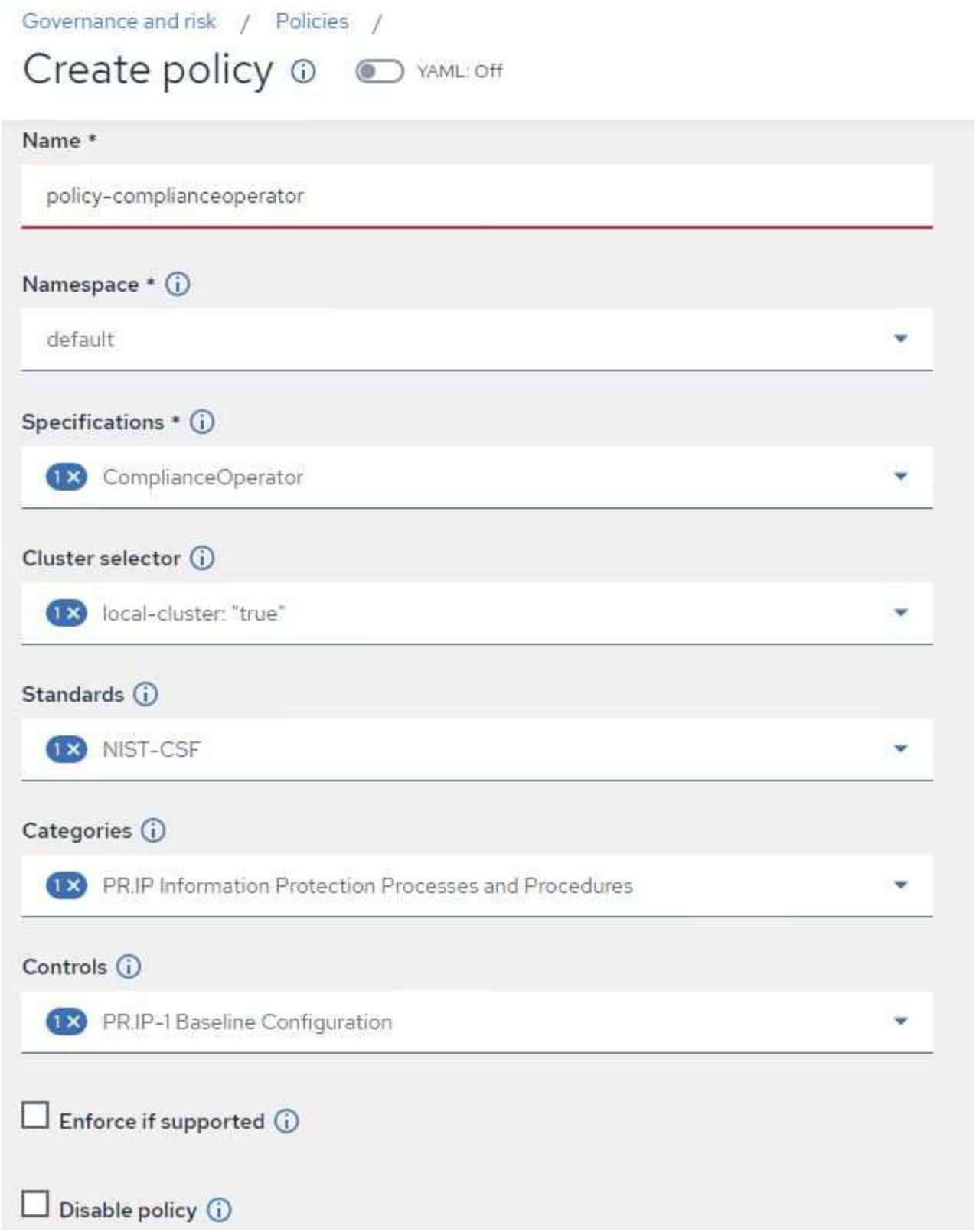


3. Ora l'applicazione può essere monitorata e gestita dalla console.

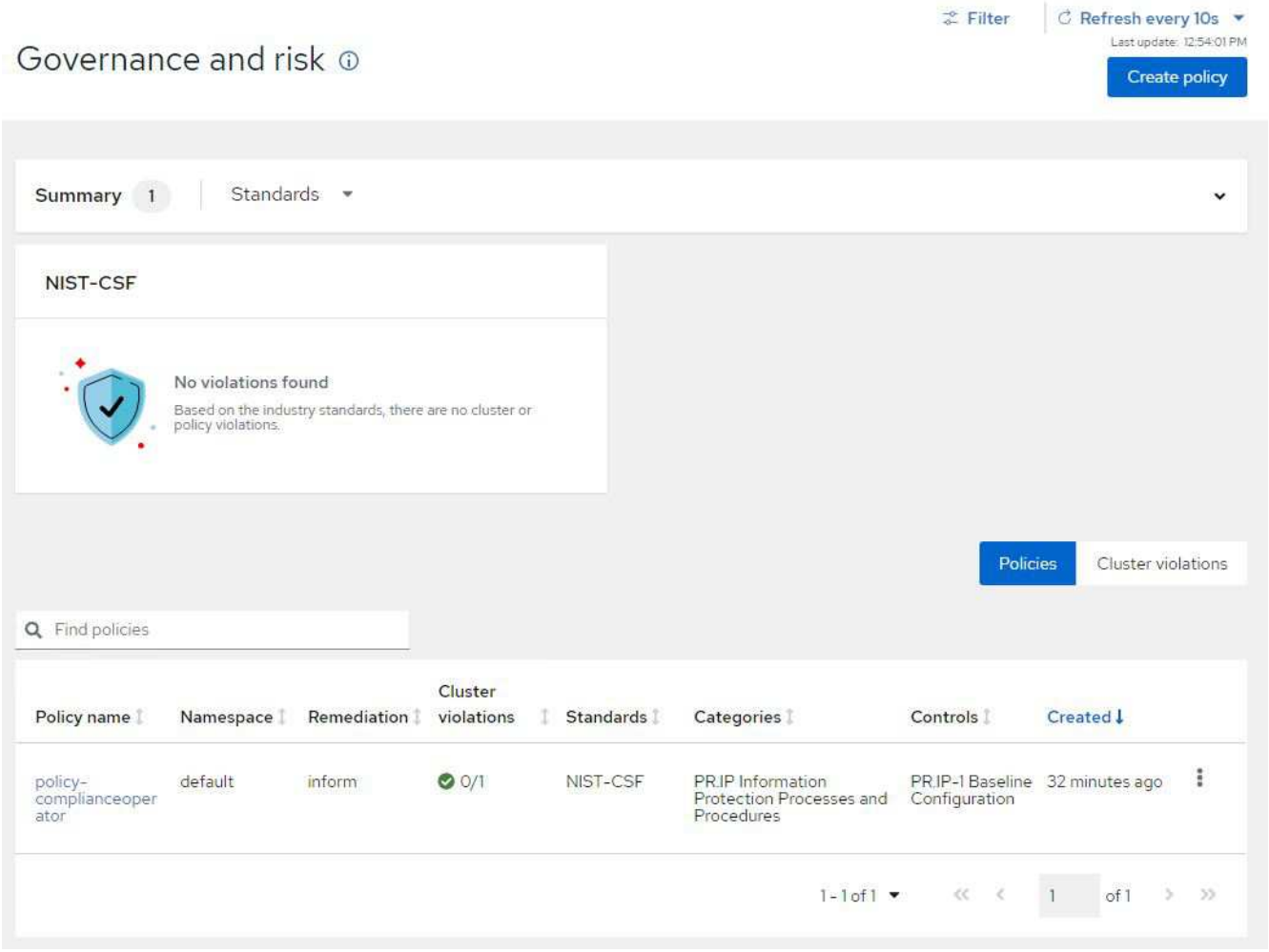
Governance e rischio

Questa funzionalità consente di definire i criteri di conformità per diversi cluster e di assicurarsi che i cluster vi aderiscano. È possibile configurare le policy per informare o correggere eventuali deviazioni o violazioni delle regole.

1. Vai a Governance e Rischio dalla barra laterale.
2. Per creare policy di conformità, fare clic su Crea policy, immettere i dettagli degli standard della policy e selezionare i cluster che devono aderire a questa policy. Se si desidera correggere automaticamente le violazioni di questa policy, selezionare la casella di controllo Applica se supportato e fare clic su Crea.



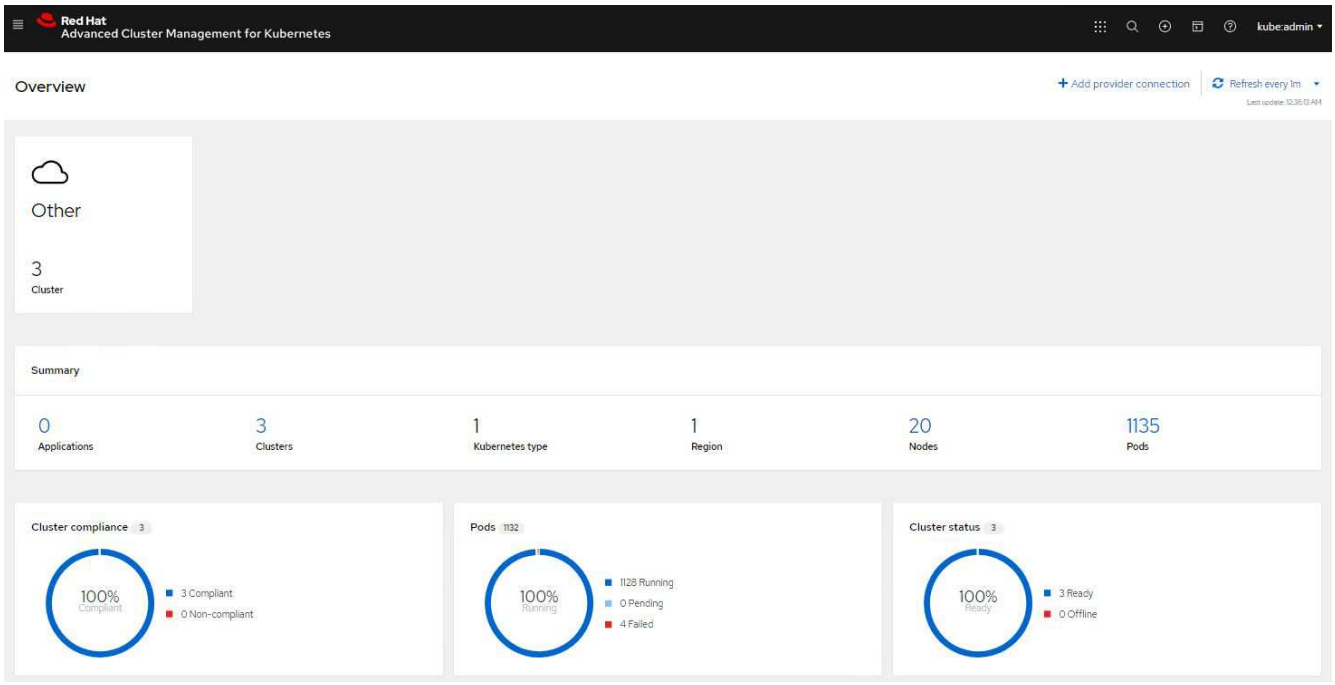
3. Dopo aver configurato tutte le policy richieste, qualsiasi violazione delle policy o del cluster può essere monitorata e risolta da Advanced Cluster Management.



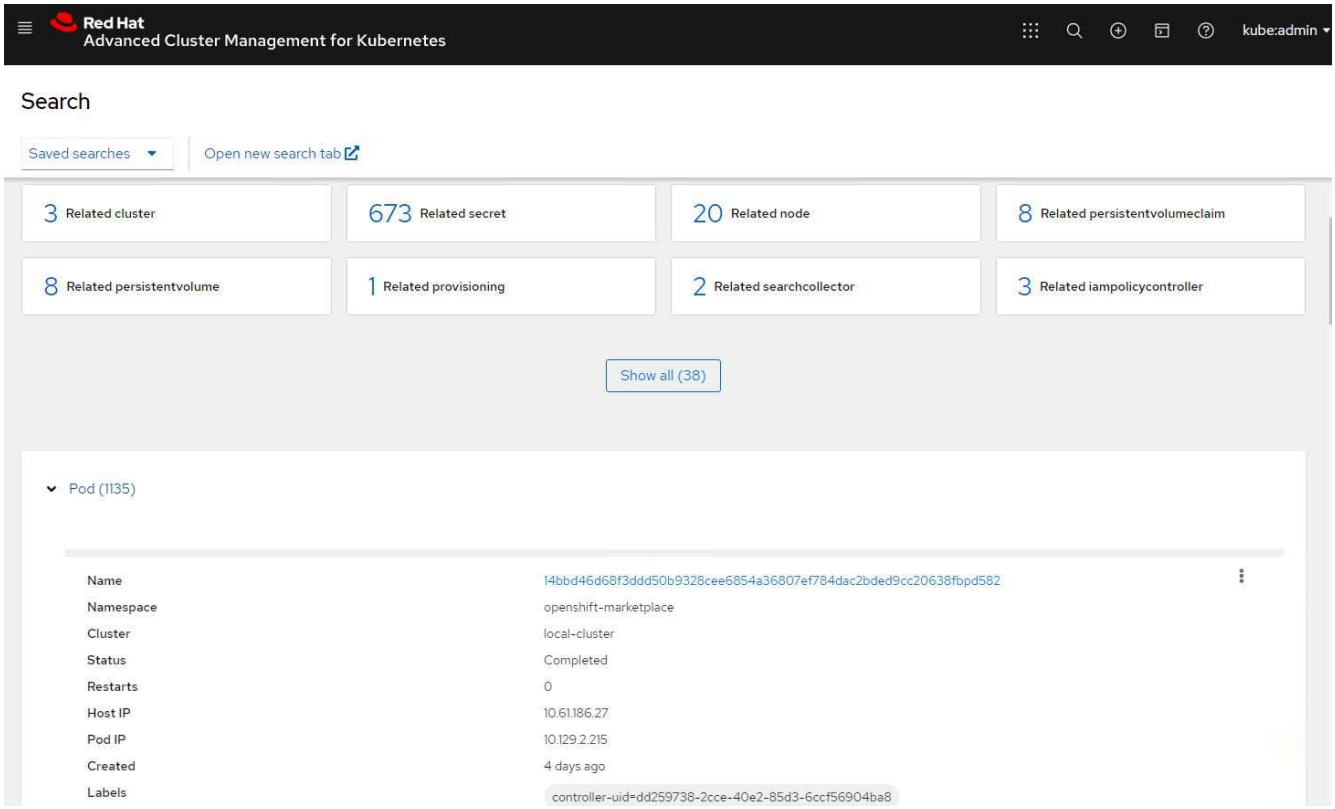
Osservabilità

Advanced Cluster Management per Kubernetes offre un modo per monitorare i nodi, i pod, le applicazioni e i carichi di lavoro in tutti i cluster.

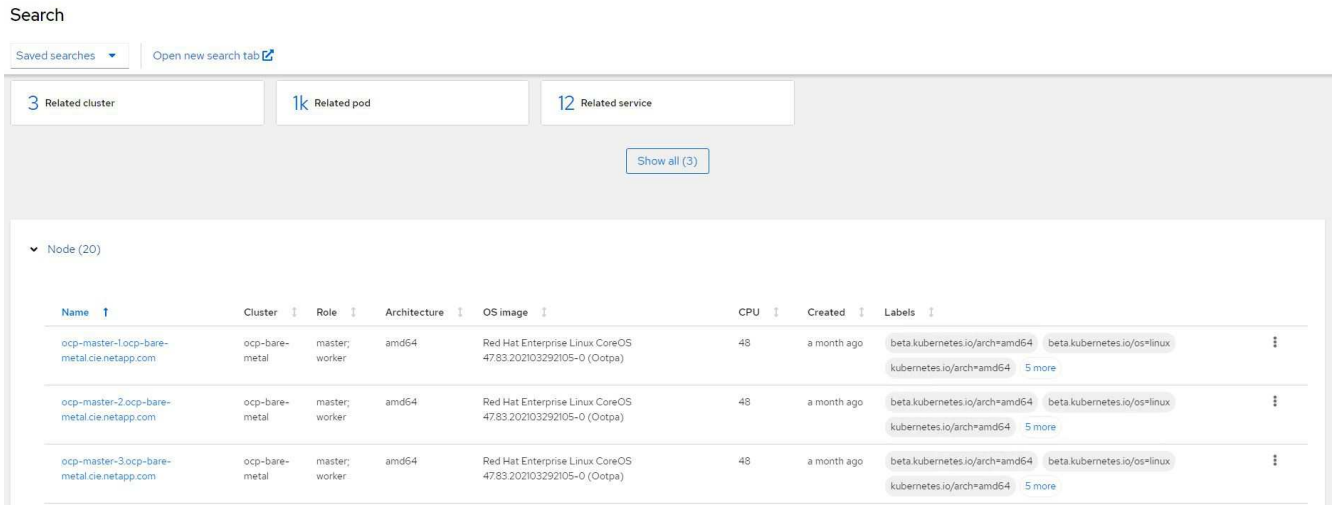
1. Vai su Osserva ambienti > Panoramica.



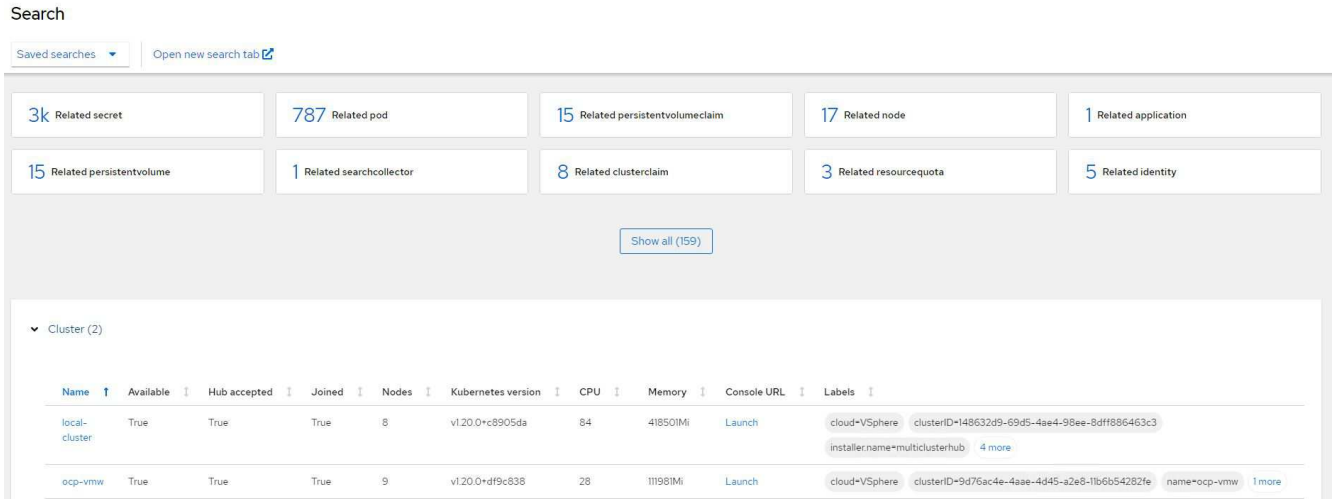
2. Tutti i pod e i carichi di lavoro in tutti i cluster vengono monitorati e ordinati in base a una serie di filtri. Fare clic su Pod per visualizzare i dati corrispondenti.



3. Tutti i nodi dei cluster vengono monitorati e analizzati in base a una serie di punti dati. Fare clic su Nodi per ottenere maggiori informazioni sui dettagli corrispondenti.

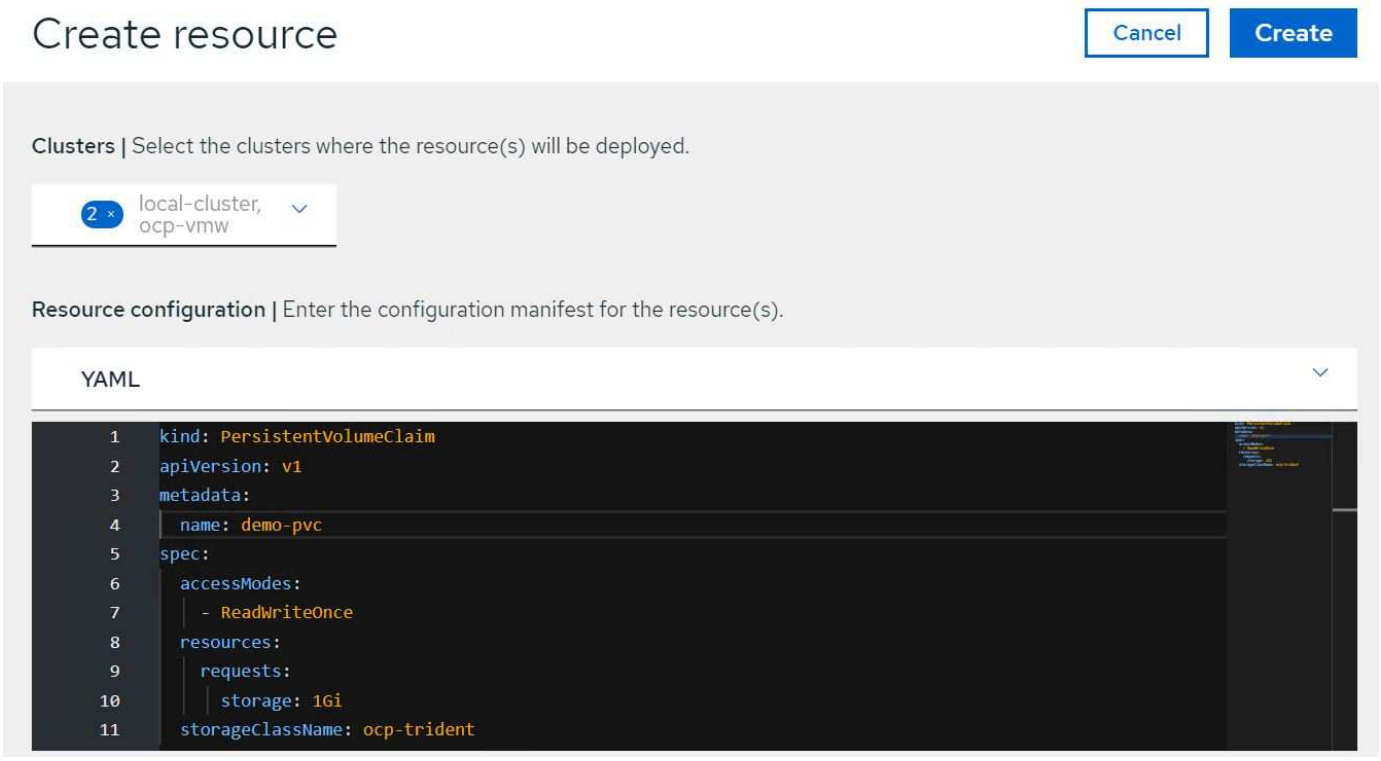


4. Tutti i cluster vengono monitorati e organizzati in base a diverse risorse e parametri del cluster. Fare clic su Cluster per visualizzare i dettagli del cluster.



Crea risorse su più cluster

Advanced Cluster Management per Kubernetes consente agli utenti di creare risorse su uno o più cluster gestiti contemporaneamente dalla console. Ad esempio, se si dispone di cluster OpenShift in siti diversi supportati da cluster NetApp ONTAP diversi e si desidera eseguire il provisioning di PVC in entrambi i siti, è possibile fare clic sul segno (+) nella barra superiore. Quindi seleziona i cluster su cui vuoi creare il PVC, incolla la risorsa YAML e fai clic su Crea.



Protezione dei dati per app container e VM tramite Trident Protect

Questa soluzione mostra come utilizzare Trident Protect per eseguire operazioni di protezione dei dati per container e VM.

1. Per i dettagli sulla creazione di snapshot e backup e sul ripristino da essi per le applicazioni container nella piattaforma OpenShift Container, fare riferimento"Qui" .
2. Per i dettagli sulla creazione e il ripristino da un backup per le VM in OpenShift Virtualization distribuite sulla piattaforma OpenShift Container, fare riferimento"Qui" .

Protezione dei dati per app container e VM tramite strumenti di terze parti

Questa soluzione mostra come utilizzare Velero, integrato con l'operatore OADP nella piattaforma Red Hat OpenShift Container, per eseguire operazioni di protezione dei dati per container e VM.

1. Per i dettagli sulla creazione e il ripristino da un backup per le applicazioni container nella piattaforma OpenShift Container, fare riferimento"Qui" .
2. Per i dettagli sulla creazione e il ripristino da un backup per le VM in OpenShift Virtualization distribuite sulla piattaforma OpenShift Container, fare riferimento"Qui" .

Risorse aggiuntive per saperne di più sull'integrazione di Red Hat OpenShift Virtualization con lo storage NetApp

Accedi a risorse aggiuntive che offrono maggiori informazioni sul supporto all'implementazione, alla gestione e all'ottimizzazione di Red Hat OpenShift Virtualization con ONTAP su diverse piattaforme e tecnologie.

- Documentazione NetApp

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

- Documentazione Trident

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

- Documentazione di Red Hat OpenShift

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

- Documentazione della piattaforma Red Hat OpenStack

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

- Documentazione sulla virtualizzazione di Red Hat

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

- Documentazione VMware vSphere

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

Informazioni sul copyright

Informazioni sul marchio commerciale