Red Hat OpenShift mit NetApp

NetApp container solutions

NetApp
January 21, 2026

Inhalt

Red Hat OpenShift mit NetApp

NVA-1160: Red Hat OpenShift mit NetApp

Alan Cowles und Nikhil M Kulkarni, NetApp

Dieses Referenzdokument bietet eine Bereitstellungsvalidierung der Red Hat OpenShift-Lösung, die über Installer Provisioned Infrastructure (IPI) in mehreren verschiedenen Rechenzentrumsumgebungen bereitgestellt wird, wie von NetApp validiert. Darüber hinaus wird die Speicherintegration mit NetApp -Speichersystemen detailliert beschrieben, indem der Trident -Speicherorchestrator zur Verwaltung des persistenten Speichers verwendet wird. Abschließend werden eine Reihe von Lösungsvalidierungen und Anwendungsfällen aus der Praxis untersucht und dokumentiert.

Anwendungsfälle

Die Red Hat OpenShift-Lösung mit NetApp ist so konzipiert, dass sie Kunden mit den folgenden Anwendungsfällen einen außergewöhnlichen Mehrwert bietet:

- Einfache Bereitstellung und Verwaltung von Red Hat OpenShift mit IPI (Installer Provisioned Infrastructure) auf Bare Metal, Red Hat OpenStack Platform, Red Hat Virtualization und VMware vSphere.
- Kombinierte Leistung von Enterprise-Containern und virtualisierten Workloads mit Red Hat OpenShift, virtuell bereitgestellt auf OSP, RHV oder vSphere oder auf Bare Metal mit OpenShift Virtualization.
- Reale Konfigurationen und Anwendungsfälle, die die Funktionen von Red Hat OpenShift bei Verwendung mit NetApp -Speicher und Trident, dem Open-Source-Speicherorchestrator für Kubernetes, hervorheben.

Geschäftswert

Unternehmen wenden zunehmend DevOps-Praktiken an, um neue Produkte zu entwickeln, Release-Zyklen zu verkürzen und schnell neue Funktionen hinzuzufügen. Aufgrund ihrer von Natur aus agilen Natur spielen Container und Microservices eine entscheidende Rolle bei der Unterstützung von DevOps-Praktiken. Die Umsetzung von DevOps im Produktionsmaßstab in einer Unternehmensumgebung bringt jedoch eigene Herausforderungen mit sich und stellt bestimmte Anforderungen an die zugrunde liegende Infrastruktur, beispielsweise die folgenden:

- Hohe Verfügbarkeit auf allen Ebenen des Stacks
- Einfache Bereitstellungsverfahren
- Unterbrechungsfreier Betrieb und Upgrades
- API-gesteuerte und programmierbare Infrastruktur, um mit der Agilität von Microservices Schritt zu halten
- Mandantenfähigkeit mit Leistungsgarantien
- Möglichkeit, virtualisierte und containerisierte Workloads gleichzeitig auszuführen
- Möglichkeit, die Infrastruktur unabhängig von den Arbeitslastanforderungen zu skalieren

Red Hat OpenShift mit NetApp berücksichtigt diese Herausforderungen und bietet eine Lösung, die bei der Bewältigung aller Probleme hilft, indem die vollständig automatisierte Bereitstellung von RedHat OpenShift IPI in der vom Kunden gewählten Rechenzentrumsumgebung implementiert wird.

Technologieübersicht

Die Red Hat OpenShift-Lösung mit NetApp besteht aus den folgenden Hauptkomponenten:

Red Hat OpenShift Container Platform

Red Hat OpenShift Container Platform ist eine vollständig unterstützte Kubernetes-Plattform für Unternehmen. Red Hat nimmt mehrere Verbesserungen am Open-Source-Kubernetes vor, um eine Anwendungsplattform mit allen vollständig integrierten Komponenten zum Erstellen, Bereitstellen und Verwalten von containerisierten Anwendungen bereitzustellen.

Weitere Informationen finden Sie auf der OpenShift-Website "hier," .

NetApp Speichersysteme

NetApp verfügt über mehrere Speichersysteme, die sich perfekt für Unternehmensrechenzentren und Hybrid-Cloud-Bereitstellungen eignen. Das NetApp Portfolio umfasst die Speichersysteme NetApp ONTAP, NetApp Element und NetApp e-Series, die alle persistenten Speicher für containerisierte Anwendungen bereitstellen können.

Weitere Informationen finden Sie auf der NetApp -Website "hier," .

NetApp Storage-Integrationen

Trident ist ein Open-Source- und vollständig unterstützter Speicherorchestrator für Container und Kubernetes-Distributionen, einschließlich Red Hat OpenShift.

Weitere Informationen finden Sie auf der Trident -Website "hier," .

Erweiterte Konfigurationsoptionen

Dieser Abschnitt befasst sich mit Anpassungen, die reale Benutzer wahrscheinlich vornehmen müssen, wenn sie diese Lösung in der Produktion einsetzen, wie etwa das Erstellen eines dedizierten privaten Image-Registers oder das Bereitstellen benutzerdefinierter Load Balancer-Instanzen.

Aktuelle Support-Matrix für validierte Releases

Technologie	Zweck	Softwareversion
NetApp ONTAP	Storage	9.8, 9.9.1, 9.12.1
NetApp Element	Storage	12,3
NetApp Trident	Speicherorchestrierung	22.01.0, 23.04, 23.07, 23.10, 24.02
Red Hat OpenShift	Container-Orchestrierung	4.6 EUS, 4.7, 4.8, 4.10, 4.11, 4.12, 4.13, 4.14
VMware vSphere	Rechenzentrumsvirtualisierung	7.0, 8.0.2

Red Hat Openshift

OpenShift-Übersicht

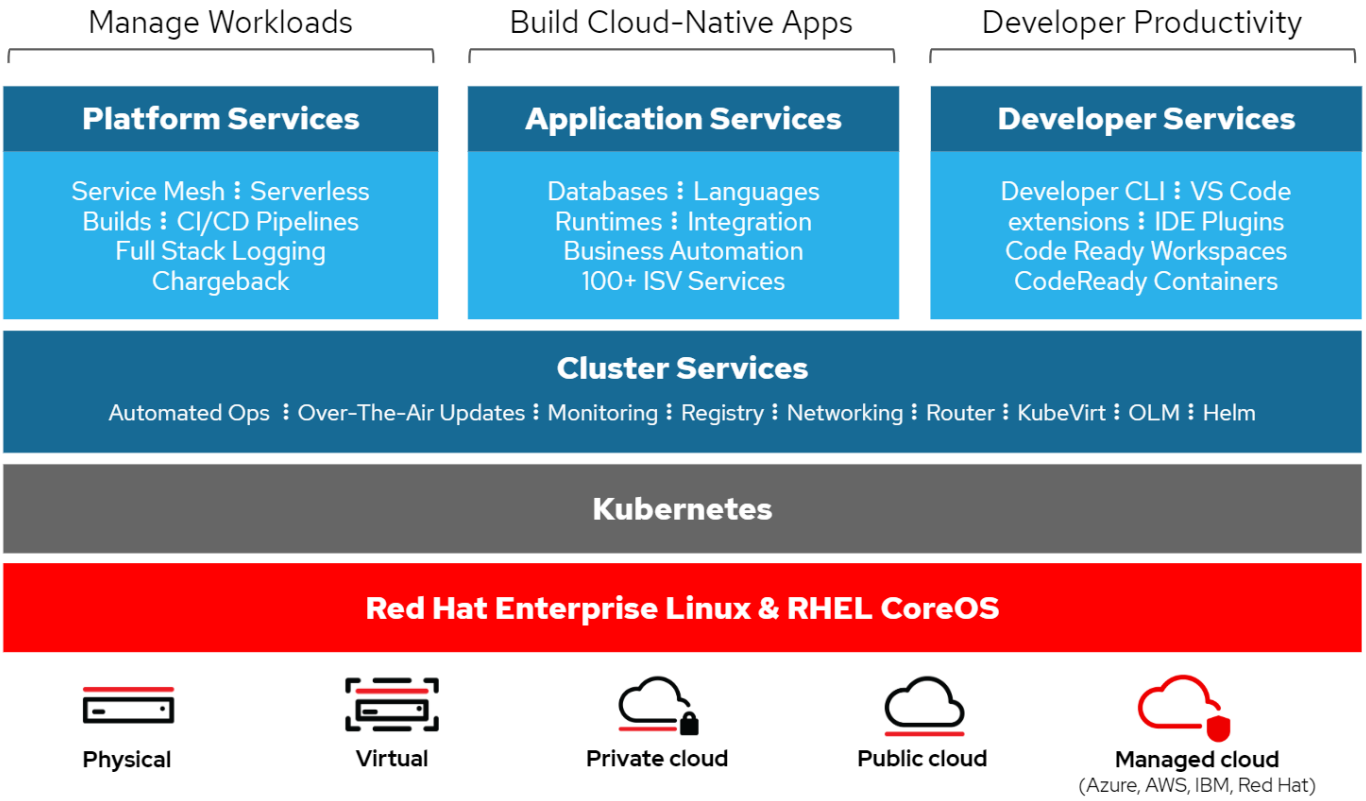
Die Red Hat OpenShift Container Platform vereint Entwicklung und IT-Betrieb auf einer einzigen Plattform, um Anwendungen konsistent über lokale und hybride Cloud-Infrastrukturen hinweg zu erstellen, bereitzustellen und zu verwalten. Red Hat OpenShift basiert auf Open-Source-Innovationen und Industriestandards, darunter Kubernetes und Red Hat Enterprise Linux CoreOS, die weltweit führende Enterprise-Linux-Distribution für containerbasierte Workloads. OpenShift ist Teil des Certified Kubernetes-Programms der Cloud Native Computing Foundation (CNCF) und bietet Portabilität und Interoperabilität von Container-Workloads.

Red Hat OpenShift bietet die folgenden Funktionen:

- **Self-Service-Bereitstellung** Entwickler können schnell und einfach Anwendungen nach Bedarf mit den Tools erstellen, die sie am häufigsten verwenden, während der Betrieb die volle Kontrolle über die gesamte Umgebung behält.
- **Persistenter Speicher** Durch die Unterstützung von persistentem Speicher ermöglicht Ihnen die OpenShift Container Platform, sowohl zustandsbehaftete als auch Cloud-native zustandslose Anwendungen auszuführen.
- **Kontinuierliche Integration und kontinuierliche Entwicklung (CI/CD)** Diese Quellcode-Plattform verwaltet Build- und Bereitstellungsimages im großen Maßstab.
- **Open-Source-Standards** Diese Standards umfassen neben anderen Open-Source-Technologien die Open Container Initiative (OCI) und Kubernetes für die Container-Orchestrierung. Sie sind nicht auf die Technologie oder die Geschäfts-Roadmap eines bestimmten Anbieters beschränkt.
- **CI/CD-Pipelines** OpenShift bietet sofort einsatzbereite Unterstützung für CI/CD-Pipelines, sodass Entwicklungsteams jeden Schritt des Anwendungsbereitstellungsprozesses automatisieren und sicherstellen können, dass er bei jeder Änderung am Code oder an der Konfiguration der Anwendung ausgeführt wird.
- **Rollenbasierte Zugriffskontrolle (RBAC)** Diese Funktion bietet Team- und Benutzerverfolgung, um die Organisation einer großen Entwicklergruppe zu erleichtern.
- **Automatisiertes Erstellen und Bereitstellen** OpenShift bietet Entwicklern die Möglichkeit, ihre containerisierten Anwendungen zu erstellen oder die Container von der Plattform aus dem Anwendungsquellcode oder sogar den Binärdateien erstellen zu lassen. Die Plattform automatisiert dann die Bereitstellung dieser Anwendungen in der gesamten Infrastruktur basierend auf den für die Anwendungen definierten Merkmalen. Beispielsweise, welche Menge an Ressourcen zugewiesen werden sollte und wo auf der Infrastruktur sie bereitgestellt werden sollten, damit sie mit Lizenzen von Drittanbietern kompatibel sind.
- **Konsistente Umgebungen** OpenShift stellt sicher, dass die für Entwickler und über den gesamten Lebenszyklus der Anwendung bereitgestellte Umgebung vom Betriebssystem über Bibliotheken und Laufzeitversionen (z. B. Java-Laufzeit) bis hin zur verwendeten Anwendungslaufzeit (z. B. Tomcat) konsistent ist, um die durch inkonsistente Umgebungen entstehenden Risiken zu beseitigen.
- **Konfigurationsverwaltung** Die Konfiguration und Verwaltung sensibler Daten ist in die Plattform integriert, um sicherzustellen, dass der Anwendung eine konsistente und umgebungsunabhängige Anwendungskonfiguration zur Verfügung gestellt wird, unabhängig davon, welche Technologien zum Erstellen der Anwendung verwendet werden oder in welcher Umgebung sie bereitgestellt wird.
- **Anwendungsprotokolle und Metriken.** Schnelles Feedback ist ein wichtiger Aspekt der Anwendungsentwicklung. Die integrierte Überwachung und Protokollverwaltung von OpenShift liefert den Entwicklern sofortige Messdaten, damit sie das Verhalten der Anwendung bei Änderungen untersuchen

und Probleme so früh wie möglich im Anwendungslebenszyklus beheben können.

- **Sicherheit und Containerkatalog** OpenShift bietet Mandantenfähigkeit und schützt den Benutzer vor der Ausführung schädlichen Codes, indem es etablierte Sicherheit mit Security-Enhanced Linux (SELinux), CGroups und Secure Computing Mode (seccomp) verwendet, um Container zu isolieren und zu schützen. Es bietet außerdem Verschlüsselung durch TLS-Zertifikate für die verschiedenen Subsysteme und Zugriff auf Red Hat-zertifizierte Container (access.redhat.com/containers), die mit besonderem Augenmerk auf Sicherheit gescannt und bewertet werden, um Endbenutzern zertifizierte, vertrauenswürdige und sichere Anwendungscontainer bereitzustellen.



Bereitstellungsmethoden für Red Hat OpenShift

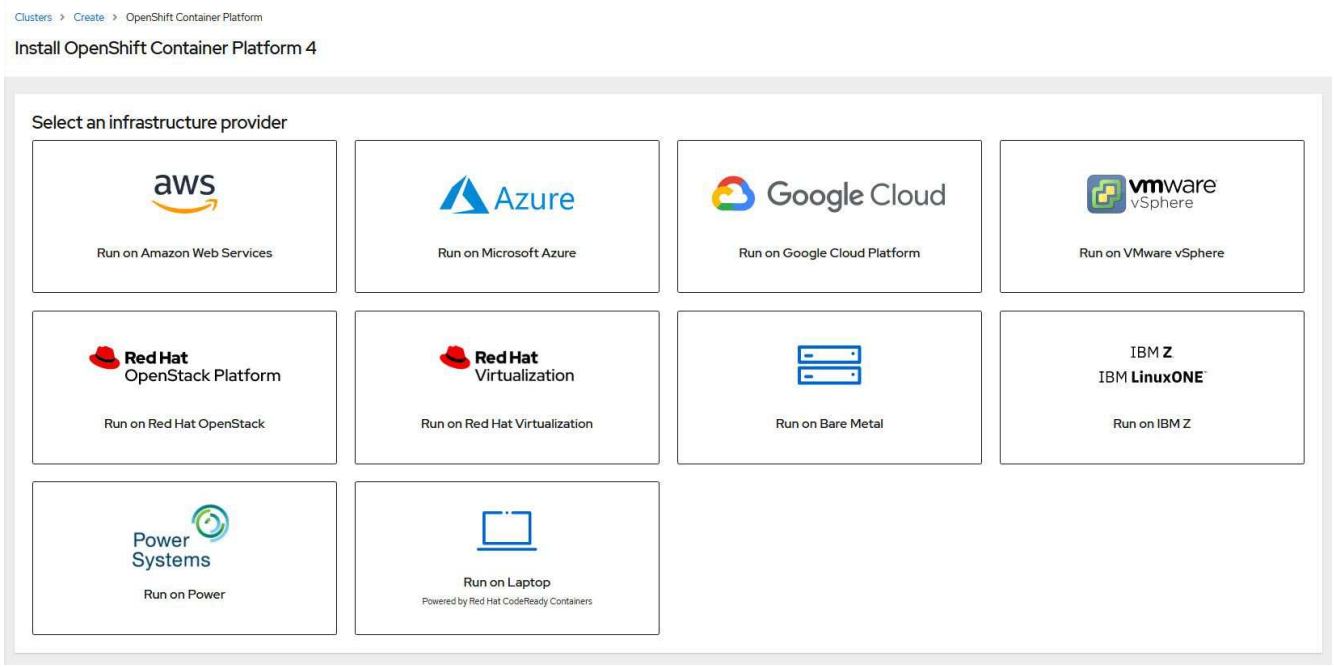
Ab Red Hat OpenShift 4 umfassen die Bereitstellungsmethoden für OpenShift manuelle Bereitstellungen mithilfe von User Provisioned Infrastructure (UPI) für hochgradig angepasste Bereitstellungen oder vollständig automatisierte Bereitstellungen mithilfe von Installer Provisioned Infrastructure (IPI).

Die IPI-Installationsmethode ist in den meisten Fällen die bevorzugte Methode, da sie die schnelle Bereitstellung von OpenShift-Clustern für Entwicklungs-, Test- und Produktionsumgebungen ermöglicht.

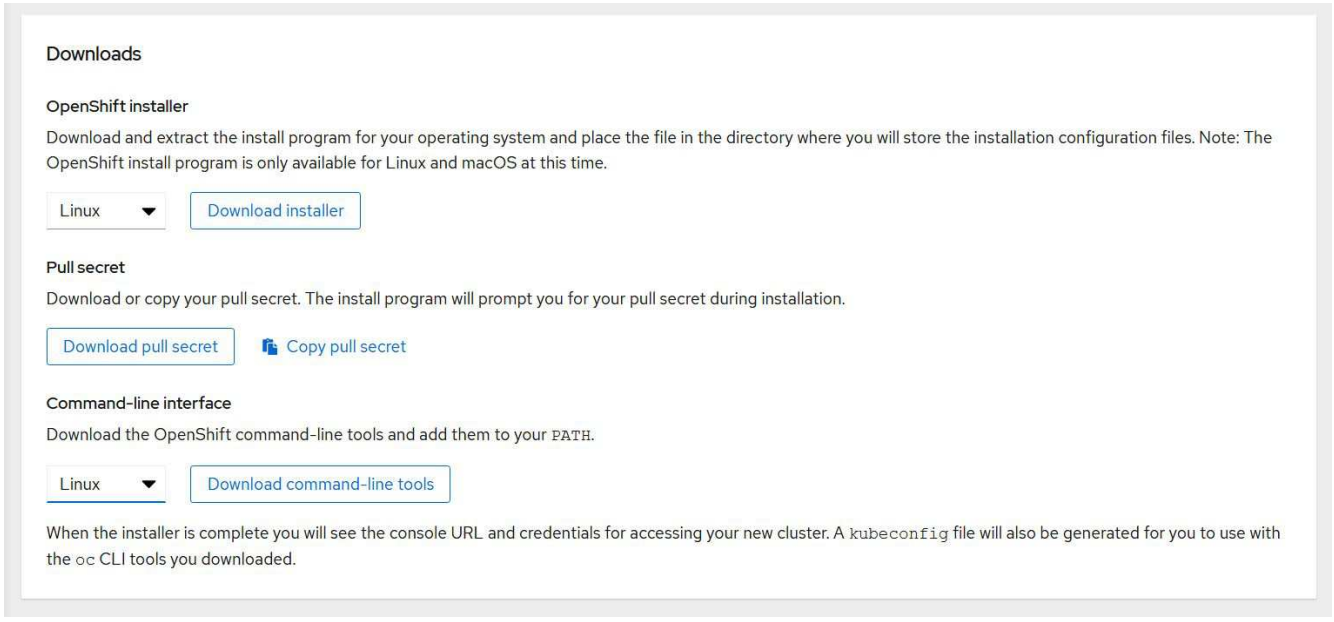
IPI-Installation von Red Hat OpenShift

Die Bereitstellung der Installer Provisioned Infrastructure (IPI) von OpenShift umfasst die folgenden allgemeinen Schritte:

1. Besuchen Sie die Red Hat OpenShift"Webseite" und melden Sie sich mit Ihren SSO-Anmeldeinformationen an.
2. Wählen Sie die Umgebung aus, in der Sie Red Hat OpenShift bereitstellen möchten.



3. Laden Sie auf dem nächsten Bildschirm das Installationsprogramm, das eindeutige Pull-Geheimnis und die CLI-Tools für die Verwaltung herunter.



4. Folgen Sie den"Installationsanweisungen" von Red Hat zur Bereitstellung in der Umgebung Ihrer Wahl.

NetApp validierte OpenShift-Bereitstellungen

NetApp hat die Bereitstellung von Red Hat OpenShift in seinen Laboren mithilfe der IPI-Bereitstellungsmethode (Installer Provisioned Infrastructure) in jeder der folgenden Rechenzentrumsumgebungen getestet und validiert:

- "OpenShift auf Bare Metal"
- "OpenShift auf der Red Hat OpenStack-Plattform"
- "OpenShift auf Red Hat Virtualization"

- "OpenShift auf VMware vSphere"

OpenShift auf Bare Metal

OpenShift on Bare Metal bietet eine automatisierte Bereitstellung der OpenShift Container Platform auf Standardservern.

OpenShift auf Bare Metal ähnelt virtuellen Bereitstellungen von OpenShift, die eine einfache Bereitstellung, schnelle Bereitstellung und Skalierung von OpenShift-Clustern ermöglichen und gleichzeitig virtualisierte Workloads für Anwendungen unterstützen, die noch nicht für die Containerisierung bereit sind. Durch die Bereitstellung auf Bare Metal entfällt der zusätzliche Aufwand, der für die Verwaltung der Host-Hypervisor-Umgebung zusätzlich zur OpenShift-Umgebung erforderlich ist. Durch die direkte Bereitstellung auf Bare-Metal-Servern können Sie auch die physischen Overhead-Einschränkungen reduzieren, die durch die gemeinsame Nutzung von Ressourcen zwischen dem Host und der OpenShift-Umgebung entstehen.

OpenShift auf Bare Metal bietet die folgenden Funktionen:

- **IPI oder unterstützte Installationsbereitstellung** Mit einem OpenShift-Cluster, der von Installer Provisioned Infrastructure (IPI) auf Bare-Metal-Servern bereitgestellt wird, können Kunden eine äußerst vielseitige, leicht skalierbare OpenShift-Umgebung direkt auf Standardservern bereitstellen, ohne eine Hypervisor-Schicht verwalten zu müssen.
- **Kompaktes Clusterdesign** Um die Hardwareanforderungen zu minimieren, ermöglicht OpenShift auf Bare Metal den Benutzern die Bereitstellung von Clustern mit nur 3 Knoten, indem die OpenShift-Steuerebenenknoten auch als Worker-Knoten und Host-Container fungieren.
- **OpenShift-Virtualisierung** OpenShift kann mithilfe der OpenShift-Virtualisierung virtuelle Maschinen in Containern ausführen. Diese containernative Virtualisierung führt den KVM-Hypervisor innerhalb eines Containers aus und fügt persistente Volumes für die VM-Speicherung hinzu.
- **Für KI/ML optimierte Infrastruktur** Stellen Sie Anwendungen wie Kubeflow für maschinelles Lernen bereit, indem Sie GPU-basierte Worker-Knoten in Ihre OpenShift-Umgebung integrieren und OpenShift Advanced Scheduling nutzen.

Netzwerkdesign

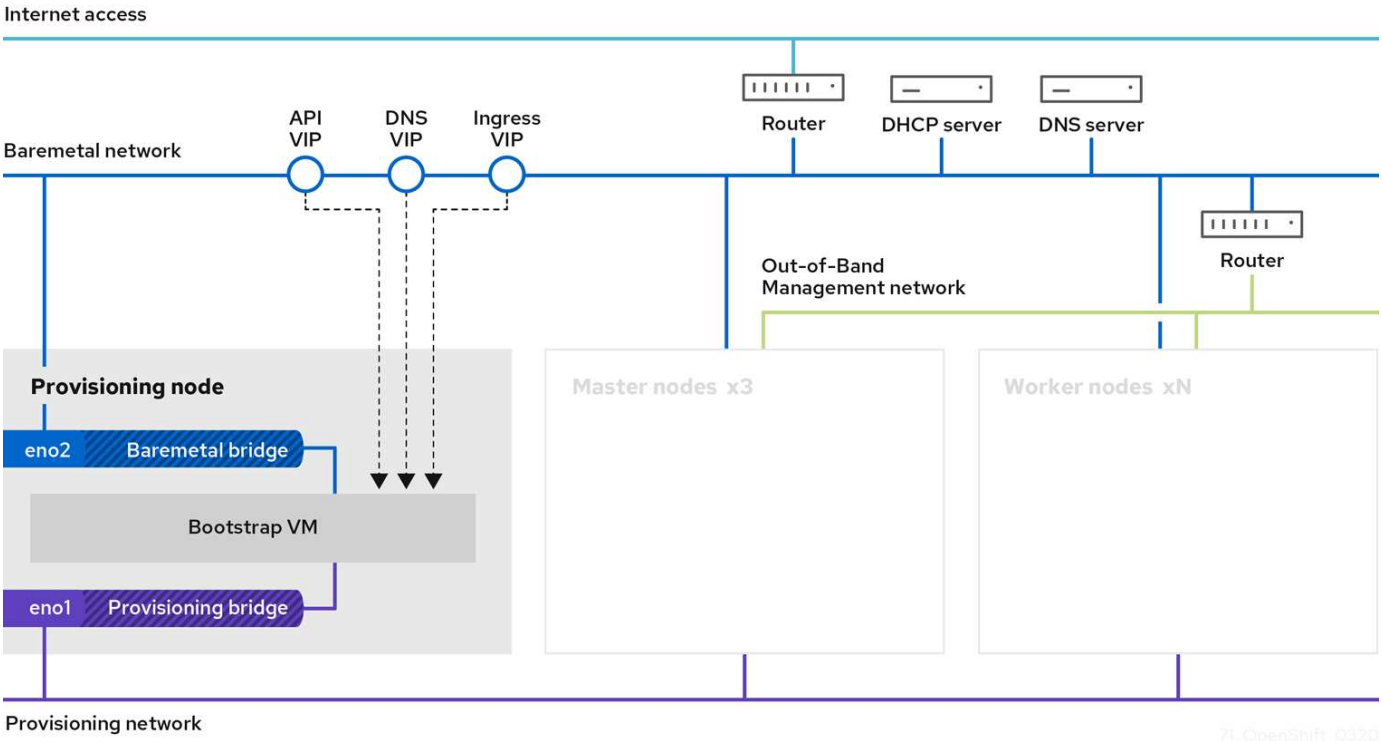
Die Red Hat OpenShift-Lösung auf NetApp verwendet zwei Daten-Switches, um eine primäre Datenkonnektivität mit 25 Gbit/s bereitzustellen. Außerdem werden zwei Verwaltungsswitches verwendet, die eine Konnektivität mit 1 Gbit/s für die In-Band-Verwaltung der Speicherknoten und eine Out-of-Band-Verwaltung für die IPMI-Funktionalität bieten.

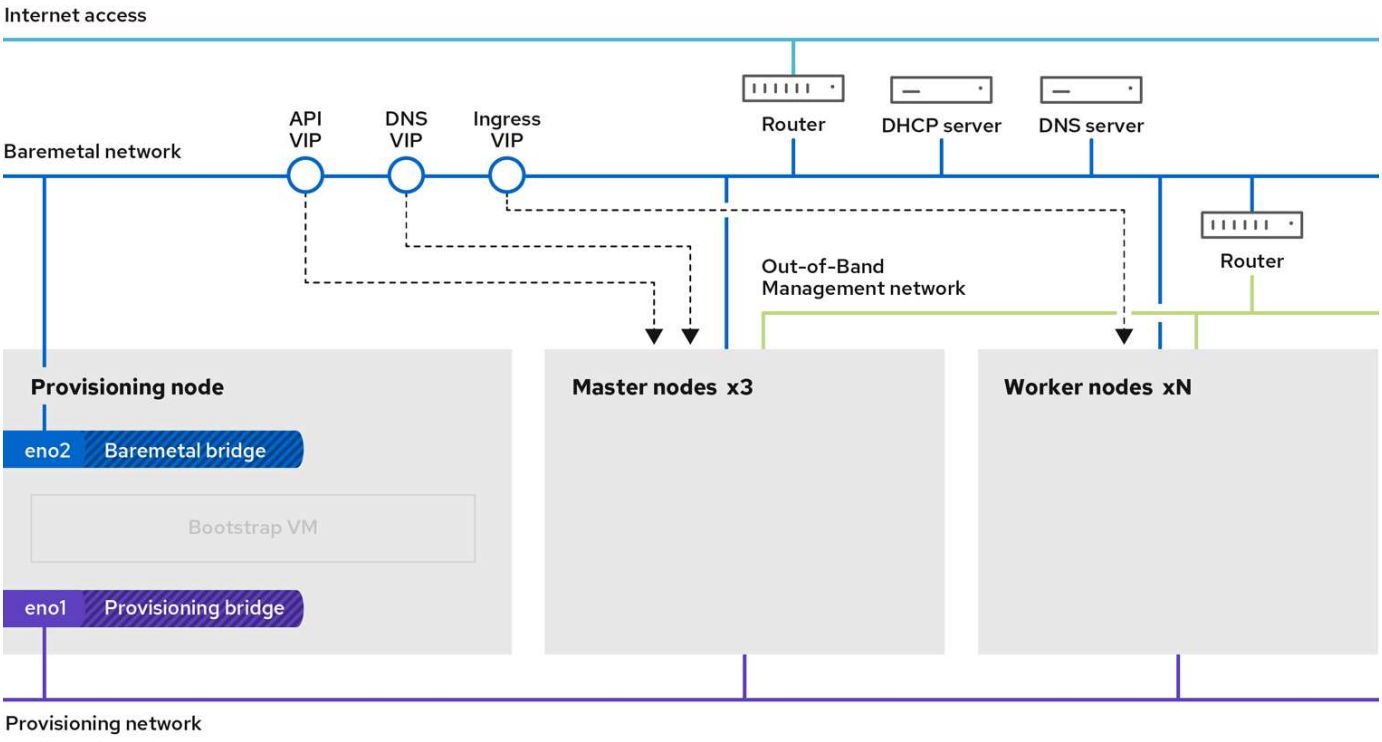
Für die Bare-Metal-IPI-Bereitstellung von OpenShift müssen Sie einen Provisioner-Knoten erstellen, eine Red Hat Enterprise Linux 8-Maschine, die über Netzwerkschnittstellen verfügen muss, die an separate Netzwerke angeschlossen sind.

- **Bereitstellungsnetzwerk** Dieses Netzwerk wird zum Booten der Bare-Metal-Knoten und zum Installieren der erforderlichen Images und Pakete zum Bereitstellen des OpenShift-Clusters verwendet.
- **Bare-Metal-Netzwerk** Dieses Netzwerk wird für die öffentliche Kommunikation des Clusters nach seiner Bereitstellung verwendet.

Für die Einrichtung des Provisioner-Knotens erstellt der Kunde Bridge-Schnittstellen, die eine ordnungsgemäße Weiterleitung des Datenverkehrs auf dem Knoten selbst und auf der für Bereitstellungszwecke bereitgestellten Bootstrap-VM ermöglichen. Nachdem der Cluster bereitgestellt wurde, werden die API- und Eingangs-VIP-Adressen vom Bootstrap-Knoten zum neu bereitgestellten Cluster migriert.

Die folgenden Bilder zeigen die Umgebung sowohl während der IPI-Bereitstellung als auch nach Abschluss der Bereitstellung.





VLAN Anforderungen

Die Lösung Red Hat OpenShift mit NetApp ist darauf ausgelegt, den Netzwerkverkehr für verschiedene Zwecke durch die Verwendung virtueller lokaler Netzwerke (VLANs) logisch zu trennen.

VLANs	Zweck	VLAN-ID
Out-of-Band-Verwaltungsnetzwerk	Verwaltung für Bare-Metal-Knoten und IPMI	16
Bare-Metal-Netzwerk	Netzwerk für OpenShift-Dienste, sobald der Cluster verfügbar ist	181
Bereitstellungsnetzwerk	Netzwerk für PXE-Boot und Installation von Bare-Metal-Knoten über IPI	3485

Obwohl jedes dieser Netzwerke virtuell durch VLANs getrennt ist, muss jeder physische Port im Zugriffsmodus mit zugewiesenem primären VLAN eingerichtet werden, da es keine Möglichkeit gibt, während einer PXE-Boot-Sequenz ein VLAN-Tag zu übergeben.

Ressourcen zur Unterstützung der Netzwerkinfrastruktur

Vor der Bereitstellung der OpenShift-Containerplattform sollte die folgende Infrastruktur vorhanden sein:

- Mindestens ein DNS-Server, der eine vollständige Hostnamenauflösung bereitstellt, auf die vom In-Band-Verwaltungsnetzwerk und dem VM-Netzwerk aus zugegriffen werden kann.
- Mindestens ein NTP-Server, der vom In-Band-Verwaltungsnetzwerk und dem VM-Netzwerk aus zugänglich ist.
- (Optional) Ausgehende Internetkonnektivität sowohl für das In-Band-Verwaltungsnetzwerk als auch für das VM-Netzwerk.

OpenShift auf der Red Hat OpenStack-Plattform

Die Red Hat OpenStack Platform bietet eine integrierte Grundlage zum Erstellen, Bereitstellen und Skalieren einer sicheren und zuverlässigen privaten OpenStack-Cloud.

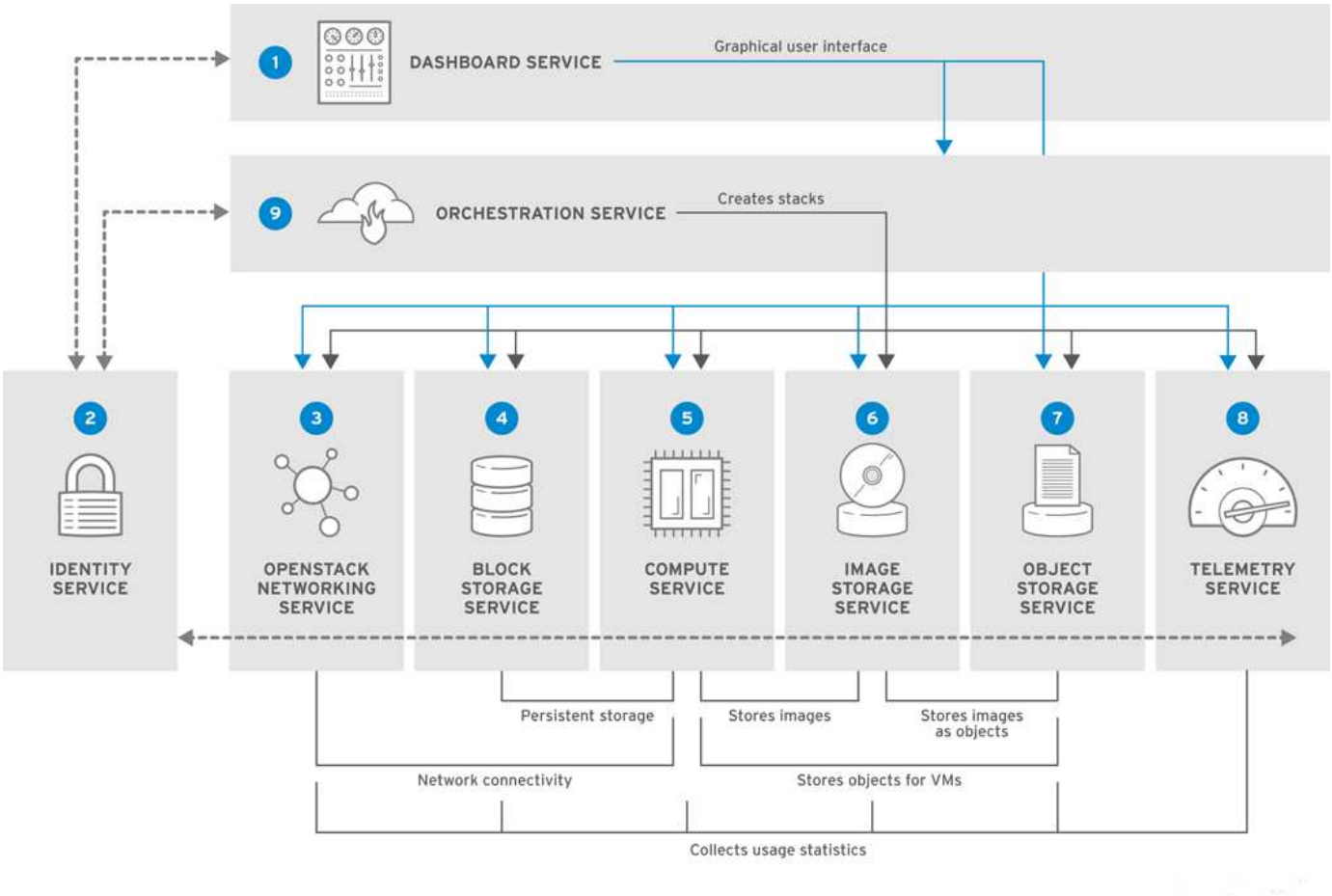
OSP ist eine Infrastructure-as-a-Service (IaaS)-Cloud, die durch eine Sammlung von Steuerungsdiensten implementiert wird, die Rechen-, Speicher- und Netzwerkressourcen verwalten. Die Umgebung wird über eine webbasierte Schnittstelle verwaltet, die es Administratoren und Benutzern ermöglicht, OpenStack-Ressourcen zu steuern, bereitzustellen und zu automatisieren. Darüber hinaus wird die OpenStack-Infrastruktur durch eine umfangreiche Befehlszeilenschnittstelle und API unterstützt, die Administratoren und Endbenutzern vollständige Automatisierungsfunktionen ermöglichen.

Das OpenStack-Projekt ist ein schnell entwickeltes Community-Projekt, das alle sechs Monate aktualisierte Versionen bereitstellt. Anfangs hielt Red Hat OpenStack Platform mit diesem Veröffentlichungszyklus Schritt, indem es zusammen mit jeder Upstream-Version eine neue Version veröffentlichte und für jede dritte Version langfristigen Support bereitstellte. Mit der kürzlich erfolgten Veröffentlichung von OSP 16.0 (basierend auf OpenStack Train) hat sich Red Hat dazu entschieden, mit den Veröffentlichungsnummern nicht Schritt zu halten, sondern stattdessen neue Funktionen in Unterversionen zurückzuportieren. Die neueste Version ist Red Hat OpenStack Platform 16.1, die zurückportierte erweiterte Funktionen aus den Upstream-Versionen Ussuri und Victoria enthält.

Weitere Informationen zu OSP finden Sie im"Red Hat OpenStack Platform-Website" .

OpenStack-Dienste

OpenStack Platform-Dienste werden als Container bereitgestellt, wodurch die Dienste voneinander isoliert werden und einfache Upgrades ermöglicht werden. Die OpenStack-Plattform verwendet eine Reihe von Containern, die mit Kolla erstellt und verwaltet werden. Die Bereitstellung der Dienste erfolgt durch Abrufen von Container-Images aus dem Red Hat Custom Portal. Diese Service-Container werden mit dem Podman-Befehl verwaltet und mit Red Hat OpenStack Director bereitgestellt, konfiguriert und gewartet.



Service	Projektname	Beschreibung
Dashboard	Horizont	Webbrowserbasiertes Dashboard, das Sie zum Verwalten von OpenStack-Diensten verwenden.
Identität	Keystone	Zentralisierter Dienst zur Authentifizierung und Autorisierung von OpenStack-Diensten und zur Verwaltung von Benutzern, Projekten und Rollen.
OpenStack-Netzwerk	Neutron	Bietet Konnektivität zwischen den Schnittstellen der OpenStack-Dienste.
Blockspeicher	Asche	Verwaltet persistente Blockspeichervolumes für virtuelle Maschinen (VMs).
Berechnen	Nova	Verwaltet und stellt VMs bereit, die auf Compute-Knoten ausgeführt werden.
Bild	Blick	Registrierungsdienst zum Speichern von Ressourcen wie VM-Images und Volume-Snapshots.
Objektspeicher	Schnell	Ermöglicht Benutzern das Speichern und Abrufen von Dateien und beliebigen Daten.

Service	Projektname	Beschreibung
Telemetrie	Wolkenhöhenmesser	Bietet Messungen zur Nutzung von Cloud-Ressourcen.
Orchestrierung	Hitze	Vorlagenbasierte Orchestrierungs-Engine, die die automatische Erstellung von Ressourcenstapeln unterstützt.

Netzwerkdesign

Die Red Hat OpenShift-Lösung mit NetApp verwendet zwei Daten-Switches, um eine primäre Datenkonnektivität mit 25 Gbit/s bereitzustellen. Außerdem werden zwei zusätzliche Verwaltungsswitches verwendet, die eine Konnektivität mit 1 Gbit/s für die In-Band-Verwaltung der Speicherknoten und eine Out-of-Band-Verwaltung für die IPMI-Funktionalität bieten.

Red Hat OpenStack Director benötigt die IPMI-Funktionalität, um Red Hat OpenStack Platform mithilfe des Ironic Bare-Metal-Bereitstellungsdienstes bereitzustellen.

VLAN Anforderungen

Red Hat OpenShift mit NetApp ist darauf ausgelegt, den Netzwerkverkehr für verschiedene Zwecke durch die Verwendung virtueller lokaler Netzwerke (VLANs) logisch zu trennen. Diese Konfiguration kann skaliert werden, um den Kundenanforderungen gerecht zu werden oder um eine weitere Isolierung für bestimmte Netzwerkdienste bereitzustellen. In der folgenden Tabelle sind die VLANs aufgeführt, die zur Implementierung der Lösung während der Validierung der Lösung bei NetApp erforderlich sind.

VLANs	Zweck	VLAN-ID
Out-of-Band-Verwaltungsnetzwerk	Netzwerk, das für die Verwaltung physischer Knoten und des IPMI-Dienstes für Ironic verwendet wird.	16
Speicherinfrastruktur	Netzwerk, das für Controller-Knoten verwendet wird, um Volumes direkt zuzuordnen und so Infrastrukturdienste wie Swift zu unterstützen.	201
Lagerasche	Netzwerk, das zum Zuordnen und Anhängen von Block-Volumes direkt an in der Umgebung bereitgestellte virtuelle Instanzen verwendet wird.	202
Interne API	Netzwerk, das für die Kommunikation zwischen den OpenStack-Diensten mithilfe von API-Kommunikation, RPC-Nachrichten und Datenbankkommunikation verwendet wird.	301
Mieter	Neutron stellt jedem Mieter per Tunneling über VXLAN eigene Netzwerke zur Verfügung. Der Netzwerkverkehr ist innerhalb jedes Mandantennetzwerks isoliert. Jedem Mandantennetzwerk ist ein IP-Subnetz zugeordnet und Netzwerk-Namespaces bedeuten, dass mehrere Mandantennetzwerke denselben Adressbereich verwenden können, ohne dass es zu Konflikten kommt.	302
Speicherverwaltung	OpenStack Object Storage (Swift) verwendet dieses Netzwerk, um Datenobjekte zwischen teilnehmenden Replikationsknoten zu synchronisieren. Der Proxy-Dienst fungiert als Zwischenschnittstelle zwischen Benutzeranforderungen und der zugrunde liegenden Speicherschicht. Der Proxy empfängt eingehende Anfragen und sucht die erforderliche Replik, um die angeforderten Daten abzurufen.	303

VLANs	Zweck	VLAN-ID
PXE	Der OpenStack Director bietet PXE-Boot als Teil des Ironic Bare Metal Provisioning Service, um die Installation der OSP Overcloud zu orchestrieren.	3484
Extern	Öffentlich verfügbares Netzwerk, das das OpenStack-Dashboard (Horizon) für die grafische Verwaltung hostet und öffentliche API-Aufrufe zur Verwaltung von OpenStack-Diensten ermöglicht.	3485
In-Band-Management-Netzwerk	Bietet Zugriff auf Systemadministrationsfunktionen wie SSH-Zugriff, DNS-Verkehr und Network Time Protocol (NTP)-Verkehr. Dieses Netzwerk fungiert auch als Gateway für Nicht-Controller-Knoten.	3486

Ressourcen zur Unterstützung der Netzwerkinfrastruktur

Vor der Bereitstellung der OpenShift Container Platform sollte die folgende Infrastruktur vorhanden sein:

- Mindestens ein DNS-Server, der eine vollständige Hostnamenauflösung bereitstellt.
- Mindestens drei NTP-Server, die die Zeit für die Server in der Lösung synchronisieren können.
- (Optional) Ausgehende Internetkonnektivität für die OpenShift-Umgebung.

Best Practices für Produktionsbereitstellungen

In diesem Abschnitt werden mehrere bewährte Methoden aufgeführt, die ein Unternehmen berücksichtigen sollte, bevor es diese Lösung in der Produktion einsetzt.

Stellen Sie OpenShift in einer privaten OSP-Cloud mit mindestens drei Rechenknoten bereit

Die in diesem Dokument beschriebene verifizierte Architektur stellt die für HA-Vorgänge geeignete Mindesthardwarebereitstellung dar, indem drei OSP-Controllerknoten und zwei OSP-Rechenknoten bereitgestellt werden. Diese Architektur gewährleistet eine fehlertolerante Konfiguration, in der beide Rechenknoten virtuelle Instanzen starten und bereitgestellte VMs zwischen den beiden Hypervisoren migrieren können.

Da Red Hat OpenShift zunächst mit drei Masterknoten bereitgestellt wird, kann eine Konfiguration mit zwei Knoten dazu führen, dass mindestens zwei Master denselben Knoten belegen. Dies kann zu einem möglichen Ausfall von OpenShift führen, wenn dieser bestimmte Knoten nicht mehr verfügbar ist. Daher ist es eine bewährte Methode von Red Hat, mindestens drei OSP-Rechenknoten bereitzustellen, damit die OpenShift-Master gleichmäßig verteilt werden können und die Lösung ein zusätzliches Maß an Fehlertoleranz erhält.

Konfigurieren der Affinität zwischen virtueller Maschine und Host

Die Verteilung der OpenShift-Master auf mehrere Hypervisor-Knoten kann durch die Aktivierung der VM/Host-Affinität erreicht werden.

Affinität ist eine Möglichkeit, Regeln für eine Reihe von VMs und/oder Hosts zu definieren, die bestimmen, ob die VMs zusammen auf demselben Host oder denselben Hosts in der Gruppe oder auf verschiedenen Hosts ausgeführt werden. Es wird auf VMs angewendet, indem Affinitätsgruppen erstellt werden, die aus VMs und/oder Hosts mit einem Satz identischer Parameter und Bedingungen bestehen. Abhängig davon, ob die VMs in einer Affinitätsgruppe auf demselben Host oder denselben Hosts in der Gruppe oder separat auf verschiedenen Hosts ausgeführt werden, können die Parameter der Affinitätsgruppe entweder eine positive oder eine negative Affinität definieren. In der Red Hat OpenStack-Plattform können Host-Affinitäts- und Anti-Affinitätsregeln erstellt und durchgesetzt werden, indem Servergruppen erstellt und Filter konfiguriert werden, sodass von Nova in einer Servergruppe bereitgestellte Instanzen auf verschiedenen Rechenknoten

bereitgestellt werden.

Eine Servergruppe verfügt standardmäßig über maximal 10 virtuelle Instanzen, für die sie die Platzierung verwalten kann. Dies kann durch Aktualisieren der Standardkontingente für Nova geändert werden.

> Für OSP-Servergruppen gibt es eine bestimmte harte Affinitäts-/Anti-Affinitätsgrenze. Wenn nicht genügend Ressourcen für die Bereitstellung auf separaten Knoten oder nicht genügend Ressourcen für die gemeinsame Nutzung von Knoten vorhanden sind, kann die VM nicht gestartet werden.

Informationen zum Konfigurieren von Affinitätsgruppen finden Sie unter"Wie konfiguriere ich Affinität und Anti-Affinität für OpenStack-Instanzen?" .

Verwenden Sie eine benutzerdefinierte Installationsdatei für die OpenShift-Bereitstellung

IPI vereinfacht die Bereitstellung von OpenShift-Clustern durch den interaktiven Assistenten, der weiter oben in diesem Dokument beschrieben wurde. Es ist jedoch möglich, dass Sie im Rahmen einer Clusterbereitstellung einige Standardwerte ändern müssen.

In diesen Fällen können Sie den Assistenten ausführen und Aufgaben bearbeiten, ohne sofort einen Cluster bereitzustellen; stattdessen wird eine Konfigurationsdatei erstellt, aus der der Cluster später bereitgestellt werden kann. Dies ist sehr nützlich, wenn Sie IPI-Standardeinstellungen ändern müssen oder wenn Sie mehrere identische Cluster in Ihrer Umgebung für andere Zwecke wie z. B. Mandantenfähigkeit bereitstellen möchten. Weitere Informationen zum Erstellen einer benutzerdefinierten Installationskonfiguration für OpenShift finden Sie unter "Red Hat OpenShift: Installieren eines Clusters auf OpenStack mit Anpassungen"Die

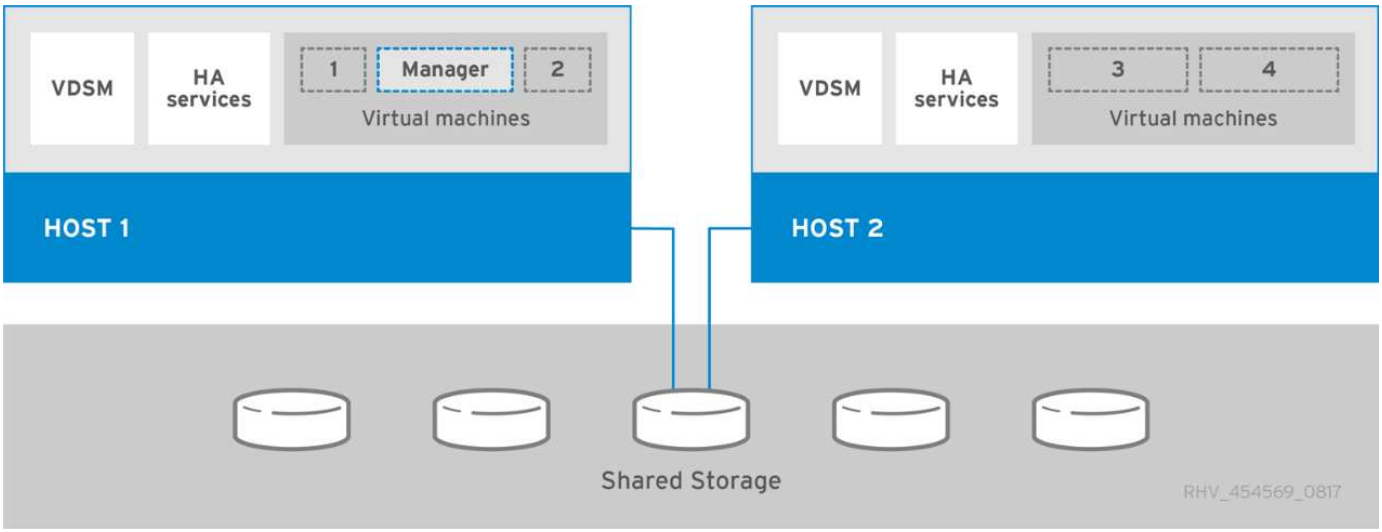
OpenShift auf Red Hat Virtualization

Red Hat Virtualization (RHV) ist eine virtuelle Unternehmens-Rechenzentrumsplattform, die auf Red Hat Enterprise Linux (RHEL) läuft und den KVM-Hypervisor verwendet.

Weitere Informationen zu RHV finden Sie im"Red Hat Virtualization-Website" .

RHV bietet die folgenden Funktionen:

- **Zentralisierte Verwaltung von VMs und Hosts** Der RHV-Manager wird als physische oder virtuelle Maschine (VM) in der Bereitstellung ausgeführt und bietet eine webbasierte GUI für die Verwaltung der Lösung über eine zentrale Schnittstelle.
- **Selbstgehostete Engine** Um die Hardwareanforderungen zu minimieren, ermöglicht RHV die Bereitstellung von RHV Manager (RHV-M) als VM auf denselben Hosts, auf denen Gast-VMs ausgeführt werden.
- **Hohe Verfügbarkeit** Um Störungen im Falle von Hostausfällen zu vermeiden, ermöglicht RHV die Konfiguration von VMs für hohe Verfügbarkeit. Die hochverfügbaren VMs werden auf Clusterebene mithilfe von Resilienzrichtlinien gesteuert.
- **Hohe Skalierbarkeit** Ein einzelner RHV-Cluster kann bis zu 200 Hypervisor-Hosts haben, wodurch er die Anforderungen massiver VMs zum Hosten ressourcenintensiver Workloads der Enterprise-Klasse unterstützen kann.
- **Verbesserte Sicherheit** Die von RHV übernommenen Technologien Secure Virtualization (sVirt) und Security Enhanced Linux (SELinux) werden von RHV zum Zwecke erhöhter Sicherheit und Härtung der Hosts und VMs eingesetzt. Der Hauptvorteil dieser Funktionen ist die logische Isolierung einer VM und der zugehörigen Ressourcen.



Netzwerkdesign

Die Red Hat OpenShift-Lösung auf NetApp verwendet zwei Daten-Switches, um eine primäre Datenkonnektivität mit 25 Gbit/s bereitzustellen. Außerdem werden zwei zusätzliche Verwaltungsswitches verwendet, die eine Konnektivität mit 1 Gbit/s für die In-Band-Verwaltung der Speicherknoten und eine Out-of-Band-Verwaltung für die IPMI-Funktionalität bieten. OCP verwendet das logische Netzwerk der virtuellen Maschine auf RHV für die Clusterverwaltung. In diesem Abschnitt werden die Anordnung und der Zweck jedes in der Lösung verwendeten virtuellen Netzwerksegments beschrieben und die Voraussetzungen für die Bereitstellung der Lösung dargelegt.

VLAN Anforderungen

Red Hat OpenShift auf RHV ist darauf ausgelegt, den Netzwerkverkehr für verschiedene Zwecke durch die Verwendung virtueller lokaler Netzwerke (VLANs) logisch zu trennen. Diese Konfiguration kann skaliert werden, um den Kundenanforderungen gerecht zu werden oder um eine weitere Isolierung für bestimmte Netzwerkdienste bereitzustellen. In der folgenden Tabelle sind die VLANs aufgeführt, die zur Implementierung der Lösung während der Validierung der Lösung bei NetApp erforderlich sind.

VLANs	Zweck	VLAN-ID
Out-of-Band-Verwaltungsnetzwerk	Verwaltung für physische Knoten und IPMI	16
VM-Netzwerk	Virtueller Gastnetzwerkzugriff	1172
In-Band-Management-Netzwerk	Verwaltung für RHV-H-Knoten, RHV-Manager und ovirtmgmt-Netzwerk	3343
Speichernetzwerk	Speichernetzwerk für NetApp Element iSCSI	3344
Migrationsnetzwerk	Netzwerk für die Migration virtueller Gäste	3345

Ressourcen zur Unterstützung der Netzwerkinfrastruktur

Vor der Bereitstellung der OpenShift Container Platform sollte die folgende Infrastruktur vorhanden sein:

- Mindestens ein DNS-Server, der eine vollständige Hostnamenauflösung bereitstellt und vom In-Band-Verwaltungsnetzwerk und dem VM-Netzwerk aus zugänglich ist.
- Mindestens ein NTP-Server, der vom In-Band-Verwaltungsnetzwerk und dem VM-Netzwerk aus zugänglich

ist.

- (Optional) Ausgehende Internetkonnektivität sowohl für das In-Band-Verwaltungsnetzwerk als auch für das VM-Netzwerk.

Best Practices für Produktionsbereitstellungen

In diesem Abschnitt werden mehrere bewährte Methoden aufgeführt, die ein Unternehmen berücksichtigen sollte, bevor es diese Lösung in der Produktion einsetzt.

Stellen Sie OpenShift in einem RHV-Cluster mit mindestens drei Knoten bereit

Die in diesem Dokument beschriebene verifizierte Architektur stellt die für HA-Vorgänge geeignete Mindesthardwarebereitstellung dar, indem zwei RHV-H-Hypervisorknoten bereitgestellt und eine fehlertolerante Konfiguration sichergestellt werden, bei der beide Hosts die gehostete Engine verwalten und bereitgestellte VMs zwischen den beiden Hypervisoren migrieren können.

Da Red Hat OpenShift zunächst mit drei Masterknoten bereitgestellt wird, ist in einer Zwei-Knoten-Konfiguration sichergestellt, dass mindestens zwei Master denselben Knoten belegen. Dies kann zu einem möglichen Ausfall von OpenShift führen, wenn dieser bestimmte Knoten nicht verfügbar ist. Daher ist es eine bewährte Methode von Red Hat, mindestens drei RHV-H-Hypervisorknoten als Teil der Lösung bereitzustellen, sodass die OpenShift-Master gleichmäßig verteilt werden können und die Lösung ein zusätzliches Maß an Fehlertoleranz erhält.

Konfigurieren der Affinität zwischen virtueller Maschine und Host

Sie können die OpenShift-Master auf mehrere Hypervisor-Knoten verteilen, indem Sie die VM/Host-Affinität aktivieren.

Affinität ist eine Möglichkeit, Regeln für eine Reihe von VMs und/oder Hosts zu definieren, die bestimmen, ob die VMs zusammen auf demselben Host oder denselben Hosts in der Gruppe oder auf verschiedenen Hosts ausgeführt werden. Es wird auf VMs angewendet, indem Affinitätsgruppen erstellt werden, die aus VMs und/oder Hosts mit einem Satz identischer Parameter und Bedingungen bestehen. Abhängig davon, ob die VMs in einer Affinitätsgruppe auf demselben Host oder denselben Hosts in der Gruppe oder separat auf verschiedenen Hosts ausgeführt werden, können die Parameter der Affinitätsgruppe entweder eine positive oder eine negative Affinität definieren.

Die für die Parameter definierten Bedingungen können entweder harte oder weiche Durchsetzung sein. Durch harte Durchsetzung wird sichergestellt, dass die VMs in einer Affinitätsgruppe immer strikt der positiven oder negativen Affinität folgen, ohne Rücksicht auf externe Bedingungen. Durch sanfte Durchsetzung wird sichergestellt, dass für die VMs in einer Affinitätsgruppe eine höhere Präferenz festgelegt wird, um der positiven oder negativen Affinität zu folgen, wann immer dies möglich ist. In der in diesem Dokument beschriebenen Konfiguration mit zwei oder drei Hypervisoren ist die weiche Affinität die empfohlene Einstellung. In größeren Clustern können OpenShift-Knoten mithilfe der harten Affinität korrekt verteilt werden.

Informationen zum Konfigurieren von Affinitätsgruppen finden Sie im"Red Hat 6.11. Affinity Groups-Dokumentation" .

Verwenden Sie eine benutzerdefinierte Installationsdatei für die OpenShift-Bereitstellung

IPI vereinfacht die Bereitstellung von OpenShift-Clustern durch den interaktiven Assistenten, der weiter oben in diesem Dokument beschrieben wurde. Es ist jedoch möglich, dass einige Standardwerte im Rahmen der Clusterbereitstellung geändert werden müssen.

In diesen Fällen können Sie den Assistenten ausführen und Aufgaben ausführen, ohne sofort einen Cluster bereitzustellen. Vielmehr wird eine Konfigurationsdatei erstellt, aus der später der Cluster bereitgestellt werden

kann. Dies ist sehr nützlich, wenn Sie IPI-Standardeinstellungen ändern möchten oder wenn Sie mehrere identische Cluster in Ihrer Umgebung für andere Zwecke wie Multitenancy bereitstellen möchten. Weitere Informationen zum Erstellen einer benutzerdefinierten Installationskonfiguration für OpenShift finden Sie unter"Red Hat OpenShift: Installieren eines Clusters auf RHV mit Anpassungen" .

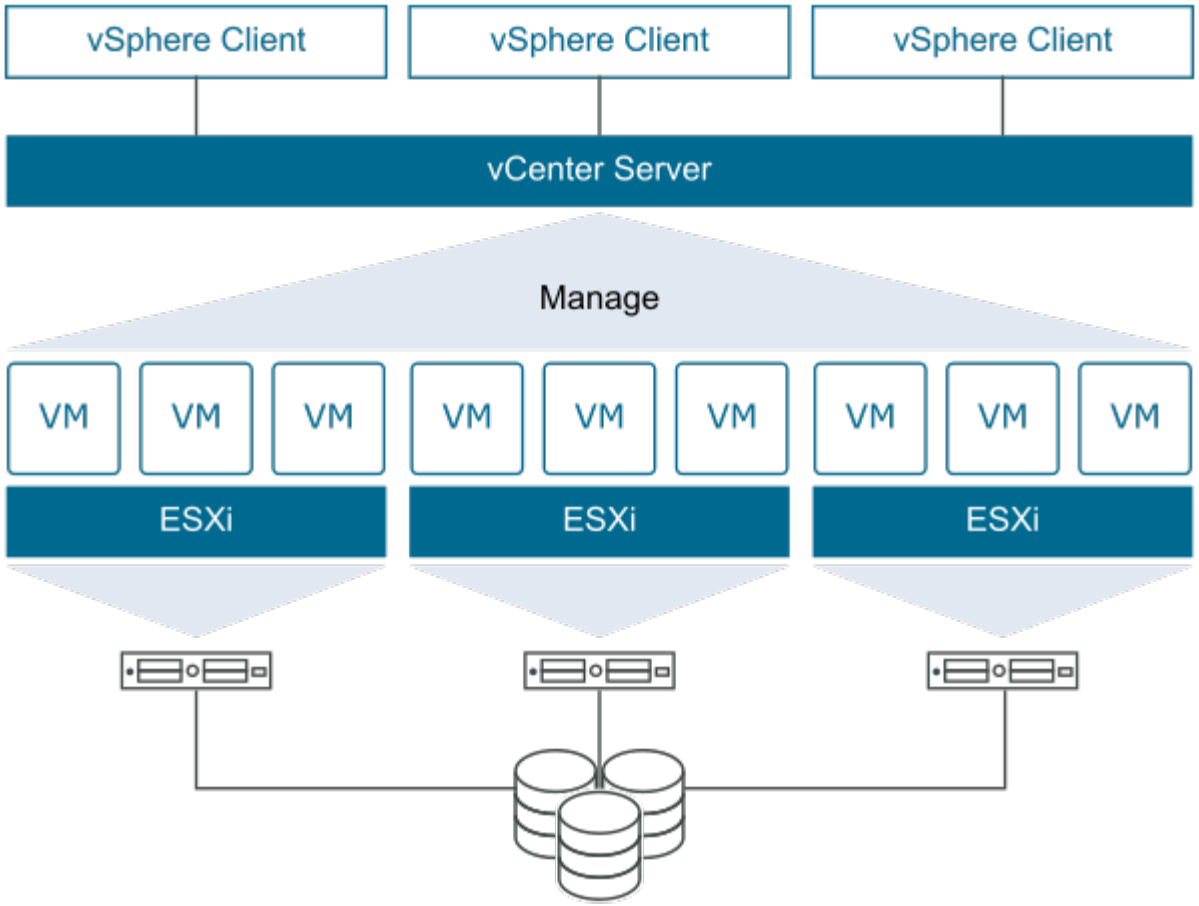
OpenShift auf VMware vSphere

VMware vSphere ist eine Virtualisierungsplattform zur zentralen Verwaltung einer großen Anzahl virtualisierter Server und Netzwerke, die auf dem ESXi-Hypervisor laufen.

Weitere Informationen zu VMware vSphere finden Sie im"VMware vSphere-Website" .

VMware vSphere bietet die folgenden Funktionen:

- **VMware vCenter Server** VMware vCenter Server bietet eine einheitliche Verwaltung aller Hosts und VMs von einer einzigen Konsole aus und aggregiert die Leistungsüberwachung von Clustern, Hosts und VMs.
- **VMware vSphere vMotion** VMware vCenter ermöglicht Ihnen auf Anfrage die unterbrechungsfreie Hot-Migration von VMs zwischen Knoten im Cluster.
- **Hochverfügbarkeit von vSphere** Um Störungen im Falle von Hostausfällen zu vermeiden, ermöglicht VMware vSphere die Clusterung und Konfiguration von Hosts für Hochverfügbarkeit. VMs, die durch einen Hostausfall gestört werden, werden kurz darauf auf anderen Hosts im Cluster neu gestartet, wodurch die Dienste wiederhergestellt werden.
- **Distributed Resource Scheduler (DRS)** Ein VMware vSphere-Cluster kann so konfiguriert werden, dass der Ressourcenbedarf der von ihm gehosteten VMs ausgeglichen wird. VMs mit Ressourcenkonflikten können im laufenden Betrieb auf andere Knoten im Cluster migriert werden, um sicherzustellen, dass genügend Ressourcen verfügbar sind.



Netzwerkdesign

Die Red Hat OpenShift-Lösung auf NetApp verwendet zwei Daten-Switches, um eine primäre Datenkonnektivität mit 25 Gbit/s bereitzustellen. Außerdem werden zwei zusätzliche Verwaltungsswitches verwendet, die eine Konnektivität mit 1 Gbit/s für die In-Band-Verwaltung der Speicherknoten und eine Out-of-Band-Verwaltung für die IPMI-Funktionalität bieten. OCP verwendet das logische VM-Netzwerk auf VMware vSphere für seine Clusterverwaltung. In diesem Abschnitt werden die Anordnung und der Zweck jedes in der Lösung verwendeten virtuellen Netzwerksegments beschrieben und die Voraussetzungen für die Bereitstellung der Lösung dargelegt.

VLAN Anforderungen

Red Hat OpenShift auf VMware vSphere ist darauf ausgelegt, den Netzwerkverkehr für verschiedene Zwecke durch die Verwendung virtueller lokaler Netzwerke (VLANs) logisch zu trennen. Diese Konfiguration kann skaliert werden, um den Kundenanforderungen gerecht zu werden oder um eine weitere Isolierung für bestimmte Netzwerkdienste bereitzustellen. In der folgenden Tabelle sind die VLANs aufgeführt, die zur Implementierung der Lösung während der Validierung der Lösung bei NetApp erforderlich sind.

VLANs	Zweck	VLAN-ID
Out-of-Band-Verwaltungsnetzwerk	Verwaltung für physische Knoten und IPMI	16
VM-Netzwerk	Virtueller Gastnetzwerkzugriff	181
Speichernetzwerk	Speichernetzwerk für ONTAP NFS	184
Speichernetzwerk	Speichernetzwerk für ONTAP iSCSI	185

VLANs	Zweck	VLAN-ID
In-Band-Management-Netzwerk	Verwaltung für ESXi-Knoten, VCenter Server, ONTAP Select	3480
Speichernetzwerk	Speichernetzwerk für NetApp Element iSCSI	3481
Migrationsnetzwerk	Netzwerk für die Migration virtueller Gäste	3482

Ressourcen zur Unterstützung der Netzwerkinfrastruktur

Vor der Bereitstellung der OpenShift Container Platform sollte die folgende Infrastruktur vorhanden sein:

- Mindestens ein DNS-Server, der eine vollständige Hostnamenauflösung bereitstellt und vom In-Band-Verwaltungsnetzwerk und dem VM-Netzwerk aus zugänglich ist.
- Mindestens ein NTP-Server, der vom In-Band-Verwaltungsnetzwerk und dem VM-Netzwerk aus zugänglich ist.
- (Optional) Ausgehende Internetkonnektivität sowohl für das In-Band-Verwaltungsnetzwerk als auch für das VM-Netzwerk.

Best Practices für Produktionsbereitstellungen

In diesem Abschnitt werden mehrere bewährte Methoden aufgeführt, die ein Unternehmen berücksichtigen sollte, bevor es diese Lösung in der Produktion einsetzt.

Stellen Sie OpenShift auf einem ESXi-Cluster mit mindestens drei Knoten bereit

Die in diesem Dokument beschriebene verifizierte Architektur stellt die für HA-Vorgänge geeignete Mindesthardwarebereitstellung dar, indem zwei ESXi-Hypervisorknoten bereitgestellt und durch die Aktivierung von VMware vSphere HA und VMware vMotion eine fehlertolerante Konfiguration sichergestellt wird. Diese Konfiguration ermöglicht die Migration bereitgestellter VMs zwischen den beiden Hypervisoren und einen Neustart, falls ein Host nicht verfügbar ist.

Da Red Hat OpenShift zunächst mit drei Masterknoten bereitgestellt wird, können unter bestimmten Umständen mindestens zwei Master in einer Zwei-Knoten-Konfiguration denselben Knoten belegen, was zu einem möglichen Ausfall von OpenShift führen kann, wenn dieser bestimmte Knoten nicht verfügbar ist. Daher ist es eine bewährte Methode von Red Hat, mindestens drei ESXi-Hypervisorknoten bereitzustellen, damit die OpenShift-Master gleichmäßig verteilt werden können, was ein zusätzliches Maß an Fehlertoleranz bietet.

Konfigurieren der virtuellen Maschine und der Hostaffinität

Die Verteilung der OpenShift-Master auf mehrere Hypervisor-Knoten kann durch die Aktivierung der VM- und Host-Affinität sichergestellt werden.

Affinität oder Anti-Affinität ist eine Möglichkeit, Regeln für eine Reihe von VMs und/oder Hosts zu definieren, die bestimmen, ob die VMs zusammen auf demselben Host oder denselben Hosts in der Gruppe oder auf verschiedenen Hosts ausgeführt werden. Es wird auf VMs angewendet, indem Affinitätsgruppen erstellt werden, die aus VMs und/oder Hosts mit einem Satz identischer Parameter und Bedingungen bestehen. Abhängig davon, ob die VMs in einer Affinitätsgruppe auf demselben Host oder denselben Hosts in der Gruppe oder separat auf verschiedenen Hosts ausgeführt werden, können die Parameter der Affinitätsgruppe entweder eine positive oder eine negative Affinität definieren.

Informationen zum Konfigurieren von Affinitätsgruppen finden Sie unter "vSphere 9.0-Dokumentation: Verwenden von DRS-Affinitätsregeln"Die

Verwenden Sie eine benutzerdefinierte Installationsdatei für die OpenShift-Bereitstellung

IPI vereinfacht die Bereitstellung von OpenShift-Clustern durch den interaktiven Assistenten, der weiter oben in diesem Dokument beschrieben wurde. Es ist jedoch möglich, dass Sie im Rahmen einer Clusterbereitstellung einige Standardwerte ändern müssen.

In diesen Fällen können Sie den Assistenten ausführen und Aufgaben ausführen, ohne sofort einen Cluster bereitzustellen. Stattdessen erstellt der Assistent eine Konfigurationsdatei, aus der der Cluster später bereitgestellt werden kann. Dies ist sehr nützlich, wenn Sie IPI-Standardeinstellungen ändern müssen oder wenn Sie mehrere identische Cluster in Ihrer Umgebung für andere Zwecke wie Multitenancy bereitstellen möchten. Weitere Informationen zum Erstellen einer benutzerdefinierten Installationskonfiguration für OpenShift finden Sie unter"Red Hat OpenShift: Installieren eines Clusters auf vSphere mit Anpassungen" .

Red Hat OpenShift Service auf AWS

Red Hat OpenShift Service auf AWS (ROSA) ist ein verwalteter Dienst, mit dem Sie containerisierte Anwendungen mit der Red Hat OpenShift Enterprise Kubernetes-Plattform auf AWS erstellen, skalieren und bereitstellen können. ROSA optimiert die Verlagerung lokaler Red Hat OpenShift-Workloads zu AWS und bietet eine enge Integration mit anderen AWS-Diensten.

Weitere Informationen zu ROSA finden Sie in der Dokumentation hier:"Red Hat OpenShift Service auf AWS (AWS-Dokumentation)" . "Red Hat OpenShift Service auf AWS (Red Hat-Dokumentation)" .

NetApp Speichersysteme

NetApp ONTAP

NetApp ONTAP ist ein leistungsstarkes Storage-Softwaretool mit Funktionen wie einer intuitiven GUI, REST-APIs mit Automatisierungsintegration, KI-gestützter prädiktiver Analyse und Korrekturmaßnahmen, unterbrechungsfreien Hardware-Upgrades und speicherübergreifendem Import.

Weitere Informationen zum NetApp ONTAP Speichersystem finden Sie auf der "NetApp ONTAP -Website" .

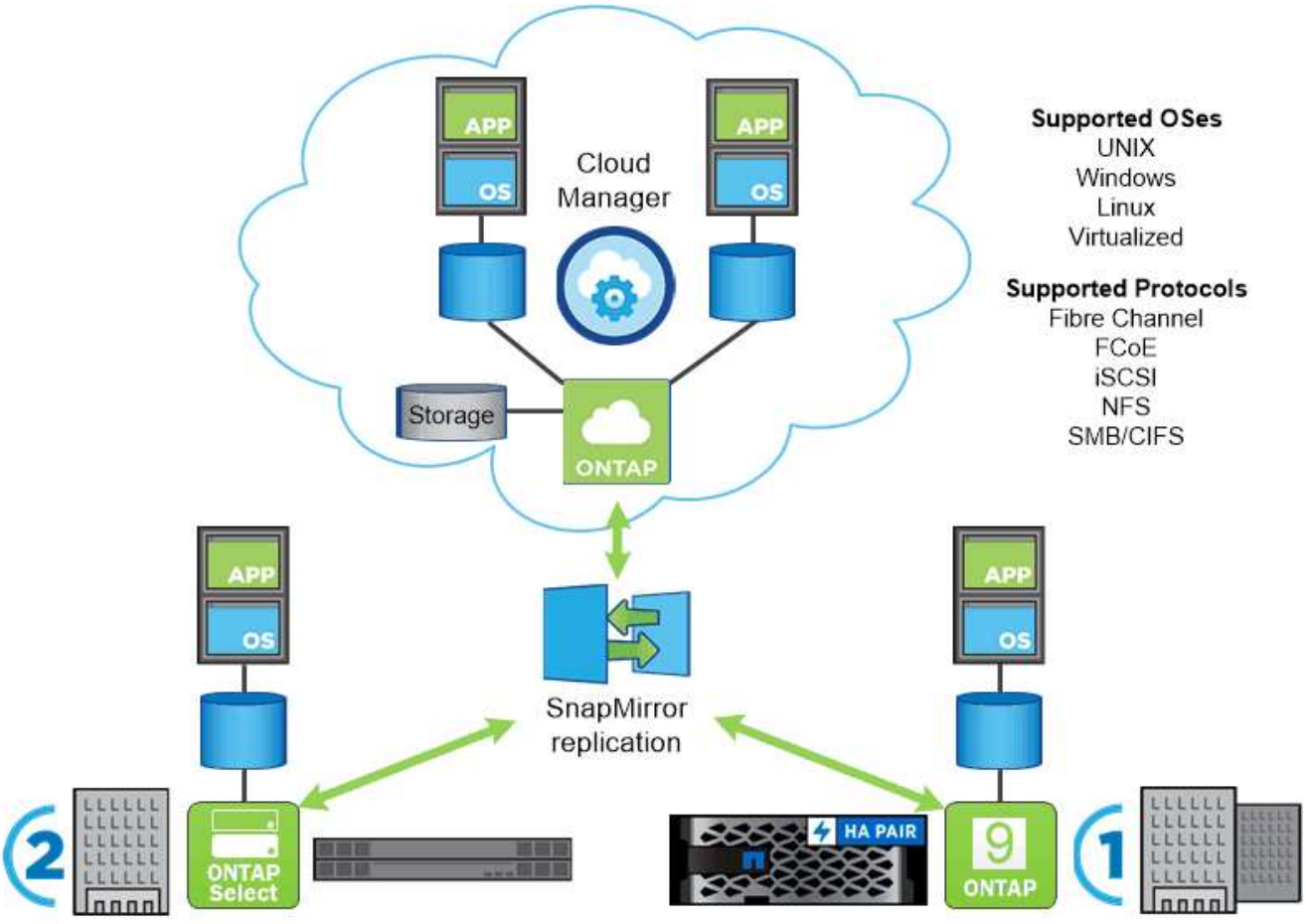
ONTAP bietet die folgenden Funktionen:

- Ein einheitliches Speichersystem mit gleichzeitigem Datenzugriff und Verwaltung der Protokolle NFS, CIFS, iSCSI, FC, FCoE und FC-NVMe.
- Zu den verschiedenen Bereitstellungsmodellen gehören On-Premises auf All-Flash-, Hybrid- und All-HDD-Hardwarekonfigurationen, VM-basierte Speicherplattformen auf einem unterstützten Hypervisor wie ONTAP Select und in der Cloud als Cloud Volumes ONTAP.
- Erhöhte Datenspeichereffizienz auf ONTAP -Systemen mit Unterstützung für automatisches Daten-Tiering, Inline-Datenkomprimierung, Deduplizierung und Komprimierung.
- Arbeitslastbasierter, QoS-gesteuerter Speicher.
- Nahtlose Integration mit einer öffentlichen Cloud zur Einstufung und zum Schutz von Daten. ONTAP bietet außerdem robuste Datenschutzfunktionen, die es in jeder Umgebung auszeichnen:
    - * NetApp Snapshot-Kopien.* Eine schnelle, zeitpunktbezogene Datensicherung mit minimalem Speicherplatzbedarf und ohne zusätzliche Leistungseinbußen.

- * NetApp SnapMirror.* Spiegelt die Snapshot-Kopien der Daten von einem Speichersystem auf ein anderes. ONTAP unterstützt auch die Spiegelung von Daten auf andere physische Plattformen und Cloud-native Dienste.
- * NetApp SnapLock.* Effiziente Verwaltung nicht wiederbeschreibbarer Daten durch Schreiben auf spezielle Datenträger, die für einen bestimmten Zeitraum nicht überschrieben oder gelöscht werden können.
- * NetApp SnapVault.* Sichert Daten von mehreren Speichersystemen auf einer zentralen Snapshot-Kopie, die als Backup für alle vorgesehenen Systeme dient.
- * NetApp SyncMirror.* Bietet Echtzeit-Spiegelung von Daten auf RAID-Ebene auf zwei verschiedene Festplattenplexe, die physisch mit demselben Controller verbunden sind.
- * NetApp SnapRestore.* Ermöglicht die schnelle Wiederherstellung gesicherter Daten auf Anfrage aus Snapshot-Kopien.
- * NetApp FlexClone.* Bietet die sofortige Bereitstellung einer vollständig lesbaren und beschreibbaren Kopie eines NetApp -Volumes basierend auf einer Snapshot-Kopie.

Weitere Informationen zu ONTAP finden Sie im "ONTAP 9 Dokumentationscenter" .

NetApp ONTAP ist vor Ort, virtualisiert oder in der Cloud verfügbar.



NetApp -Plattformen

NetApp AFF/ FAS

NetApp bietet robuste All-Flash- (AFF) und Scale-Out-Hybrid- (FAS) Speicherplattformen, die maßgeschneidert sind und eine Leistung mit geringer Latenz, integrierten Datenschutz und Multiprotokoll-Unterstützung bieten.

Beide Systeme basieren auf der Datenverwaltungssoftware NetApp ONTAP , der branchenweit fortschrittlichsten Datenverwaltungssoftware für hochverfügbares, Cloud-integriertes und vereinfachtes Speichermanagement, um die Geschwindigkeit, Effizienz und Sicherheit der Enterprise-Klasse zu bieten, die Ihr Data Fabric benötigt.

Weitere Informationen zu NETAPP AFF/ FAS -Plattformen finden Sie unter "hier," .

ONTAP Select

ONTAP Select ist eine softwaredefinierte Bereitstellung von NetApp ONTAP , die auf einem Hypervisor in Ihrer Umgebung bereitgestellt werden kann. Es kann auf VMware vSphere oder KVM installiert werden und bietet die volle Funktionalität und Erfahrung eines hardwarebasierten ONTAP Systems.

Weitere Informationen zu ONTAP Select erhalten Sie, indem Sie auf "hier," .

Cloud Volumes ONTAP

NetApp Cloud Volumes ONTAP ist eine in der Cloud bereitgestellte Version von NetApp ONTAP , die in einer Reihe öffentlicher Clouds bereitgestellt werden kann, darunter: Amazon AWS, Microsoft Azure und Google Cloud.

Weitere Informationen zu Cloud Volumes ONTAP erhalten Sie, indem Sie auf "hier," .

Amazon FSx ONTAP

Amazon FSx ONTAP bietet vollständig verwalteten gemeinsam genutzten Speicher in der AWS Cloud mit den beliebten Datenzugriffs- und Verwaltungsfunktionen von ONTAP. Weitere Informationen zu Amazon FSx ONTAP erhalten Sie, indem Sie auf "hier," .

Azure NetApp Files

Azure NetApp Files ist ein nativer, leistungsstarker Dateispeicherdienst der Enterprise-Klasse von Azure. Es stellt Volumes als Dienst bereit, für den Sie NetApp Konten, Kapazitätspools und Volumes erstellen können. Sie können außerdem Service- und Leistungsstufen auswählen und den Datenschutz verwalten. Sie können leistungsstarke, hochverfügbare und skalierbare Dateifreigaben erstellen und verwalten, indem Sie dieselben Protokolle und Tools verwenden, mit denen Sie vertraut sind und auf die Sie sich vor Ort verlassen. Weitere Informationen zu Azure NetApp Files erhalten Sie, indem Sie auf "hier," .

Google Cloud NetApp Volumes

Google Cloud NetApp Volumes ist ein vollständig verwalteter, Cloud-basierter Datenspeicherdienst, der erweiterte Datenverwaltungsfunktionen und hochgradig skalierbare Leistung bietet. Damit können Sie dateibasierte Anwendungen in die Google Cloud verschieben. Es bietet integrierte Unterstützung für die Protokolle Network File System (NFSv3 und NFSv4.1) und Server Message Block (SMB), sodass Sie Ihre Anwendungen nicht neu strukturieren müssen und weiterhin dauerhaften Speicher für Ihre Anwendungen erhalten können. Weitere Informationen zu Google Cloud NetApp VolumesP finden Sie unter "hier," .

NetApp Element: Red Hat OpenShift mit NetApp

Die NetApp Element -Software bietet modulare, skalierbare Leistung, wobei jeder Speicherknoten garantierte Kapazität und Durchsatz für die Umgebung bereitstellt. NetApp Element -Systeme können von 4 auf 100 Knoten in einem einzigen Cluster skaliert werden und bieten eine Reihe erweiterter Speicherverwaltungsfunktionen.

Weitere Informationen zu NetApp Element -Speichersystemen finden Sie auf der "NetApp Solidfire-Website" .

iSCSI-Anmeldeumleitung und Selbstheilungsfunktionen

Die NetApp Element -Software nutzt das iSCSI-Speicherprotokoll, eine Standardmethode zum Kapseln von SCSI-Befehlen in einem herkömmlichen TCP/IP-Netzwerk. Wenn sich die SCSI-Standards ändern oder die Leistung von Ethernet-Netzwerken verbessert wird, profitiert das iSCSI-Speicherprotokoll, ohne dass Änderungen erforderlich sind.

Obwohl alle Speicherknoten über eine Verwaltungs-IP und eine Speicher-IP verfügen, gibt die NetApp Element -Software eine einzige virtuelle Speicher-IP-Adresse (SVIP-Adresse) für den gesamten Speicherverkehr im Cluster bekannt. Als Teil des iSCSI-Anmeldevorgangs kann der Speicher antworten, dass das Zielvolume an eine andere Adresse verschoben wurde und daher der Verhandlungsprozess nicht fortgesetzt werden kann. Der Host sendet dann die Anmeldeanforderung erneut an die neue Adresse in einem Prozess, der keine Neukonfiguration auf Hostseite erfordert. Dieser Vorgang wird als iSCSI-Anmeldeumleitung bezeichnet.

Die iSCSI-Anmeldeumleitung ist ein wichtiger Bestandteil des NetApp Element -Softwareclusters. Wenn eine Host-Anmeldeanforderung eingeht, entscheidet der Knoten basierend auf den IOPS und den Kapazitätsanforderungen für das Volume, welches Mitglied des Clusters den Datenverkehr verarbeiten soll. Volumes werden über den NetApp Element -Softwarecluster verteilt und neu verteilt, wenn ein einzelner Knoten zu viel Datenverkehr für seine Volumes verarbeitet oder wenn ein neuer Knoten hinzugefügt wird. Mehrere Kopien eines bestimmten Volumes werden über das Array verteilt.

Auf diese Weise hat eine auf einen Knotenausfall folgende Neuverteilung des Volumes keine Auswirkungen auf die Hostkonnektivität, abgesehen von einer Abmeldung und Anmeldung mit Umleitung zum neuen Standort. Mit der iSCSI-Anmeldeumleitung ist ein NetApp Element -Softwarecluster eine selbstheilende, skalierbare Architektur, die unterbrechungsfreie Upgrades und Vorgänge ermöglicht.

NetApp Element Softwarecluster QoS

Ein NetApp Element -Softwarecluster ermöglicht die dynamische Konfiguration von QoS auf Volume-Basis. Sie können QoS-Einstellungen pro Volume verwenden, um die Speicherleistung basierend auf den von Ihnen definierten SLAs zu steuern. Die folgenden drei konfigurierbaren Parameter definieren die QoS:

- **Mindest-IOPS.** Die Mindestanzahl dauerhafter IOPS, die der NetApp Element -Softwarecluster einem Volume bereitstellt. Der für ein Volume konfigurierte Mindest-IOPS-Wert ist das garantierte Leistungsniveau für ein Volume. Die Leistung pro Volumen fällt nicht unter dieses Niveau.

- **Maximale IOPS.** Die maximale Anzahl dauerhafter IOPS, die der NetApp Element -Softwarecluster einem bestimmten Volume bereitstellt.
- **Burst-IOPS.** Die maximale Anzahl an IOPS, die in einem Short-Burst-Szenario zulässig ist. Die Einstellung der Burst-Dauer ist konfigurierbar, der Standardwert beträgt 1 Minute. Wenn ein Volume unterhalb des maximalen IOPS-Levels ausgeführt wurde, werden Burst-Guthaben angesammelt. Wenn die Leistungsstufen sehr hoch werden und ausgereizt werden, sind auf dem Volume kurze IOPS-Spitzen über den maximalen IOPS-Wert hinaus zulässig.

Mandantenfähigkeit

Sichere Mandantenfähigkeit wird durch die folgenden Funktionen erreicht:

- **Sichere Authentifizierung.** Für den sicheren Volume-Zugriff wird das Challenge-Handshake Authentication Protocol (CHAP) verwendet. Für den sicheren Zugriff auf den Cluster zur Verwaltung und Berichterstellung wird das Lightweight Directory Access Protocol (LDAP) verwendet.
- **Volume Access Groups (VAGs).** Optional können VAGs anstelle der Authentifizierung verwendet werden, indem eine beliebige Anzahl iSCSI-initiatorspezifischer iSCSI Qualified Names (IQNs) einem oder mehreren Volumes zugeordnet wird. Um auf ein Volume in einer VAG zuzugreifen, muss sich der IQN des Initiators in der Liste der zulässigen IQNs für die Volumegruppe befinden.
- **Virtuelle LANs (VLANs) des Mandanten.** Auf Netzwerkebene wird die End-to-End-Netzwerksicherheit zwischen iSCSI-Initiatoren und dem NetApp Element -Softwarecluster durch die Verwendung von VLANs erleichtert. Für jedes VLAN, das zur Isolierung einer Arbeitslast oder eines Mandanten erstellt wird, erstellt die NetApp Element Software eine separate iSCSI-Ziel-SVIP-Adresse, auf die nur über das spezifische VLAN zugegriffen werden kann.
- **VRF-fähige VLANs.** Um die Sicherheit und Skalierbarkeit im Rechenzentrum weiter zu unterstützen, können Sie mit der NetApp Element -Software jedes beliebige Tenant-VLAN für VRF-ähnliche Funktionen aktivieren. Diese Funktion fügt diese beiden wichtigen Funktionen hinzu:
  - **L3-Routing zu einer SVIP-Adresse des Mandanten.** Mit dieser Funktion können Sie iSCSI-Initiatoren in einem anderen Netzwerk oder VLAN als dem des NetApp Element -Softwareclusters platzieren.
  - **Überlappende oder doppelte IP-Subnetze.** Mit dieser Funktion können Sie Mandantenumgebungen eine Vorlage hinzufügen, sodass jedem jeweiligen Mandanten-VLAN IP-Adressen aus demselben IP-Subnetz zugewiesen werden können. Diese Funktion kann für In-Service-Provider-Umgebungen nützlich sein, in denen Skalierung und Erhaltung des IP-Speicherplatzes wichtig sind.

Speichereffizienz für Unternehmen

Der NetApp Element Softwarecluster steigert die Gesamteffizienz und Leistung des Speichers. Die folgenden Funktionen werden inline ausgeführt, sind immer aktiviert und erfordern keine manuelle Konfiguration durch den Benutzer:

- **Deduplizierung.** Das System speichert nur eindeutige 4K-Blöcke. Alle doppelten 4K-Blöcke werden automatisch einer bereits gespeicherten Version der Daten zugeordnet. Die Daten befinden sich auf Blocklaufwerken und werden mithilfe der Helix-Datensicherungssoftware von NetApp Element gespiegelt. Dieses System reduziert den Kapazitätsverbrauch und die Schreibvorgänge innerhalb des Systems erheblich.
- **Kompression.** Die Komprimierung wird inline durchgeführt, bevor die Daten in den NVRAM geschrieben werden. Daten werden komprimiert, in 4K-Blöcken gespeichert und bleiben im System komprimiert. Diese Komprimierung reduziert den Kapazitätsverbrauch, die Schreibvorgänge und den Bandbreitenverbrauch im gesamten Cluster erheblich.
- **Thin Provisioning.** Diese Funktion stellt die richtige Speichermenge zum benötigten Zeitpunkt bereit und verhindert so den Kapazitätsverbrauch, der durch überdimensionierte oder nicht ausgelastete Volumes

verursacht wird.

- **Wendel.** Die Metadaten für ein einzelnes Volume werden auf einem Metadatenlaufwerk gespeichert und zur Redundanz auf ein sekundäres Metadatenlaufwerk repliziert.

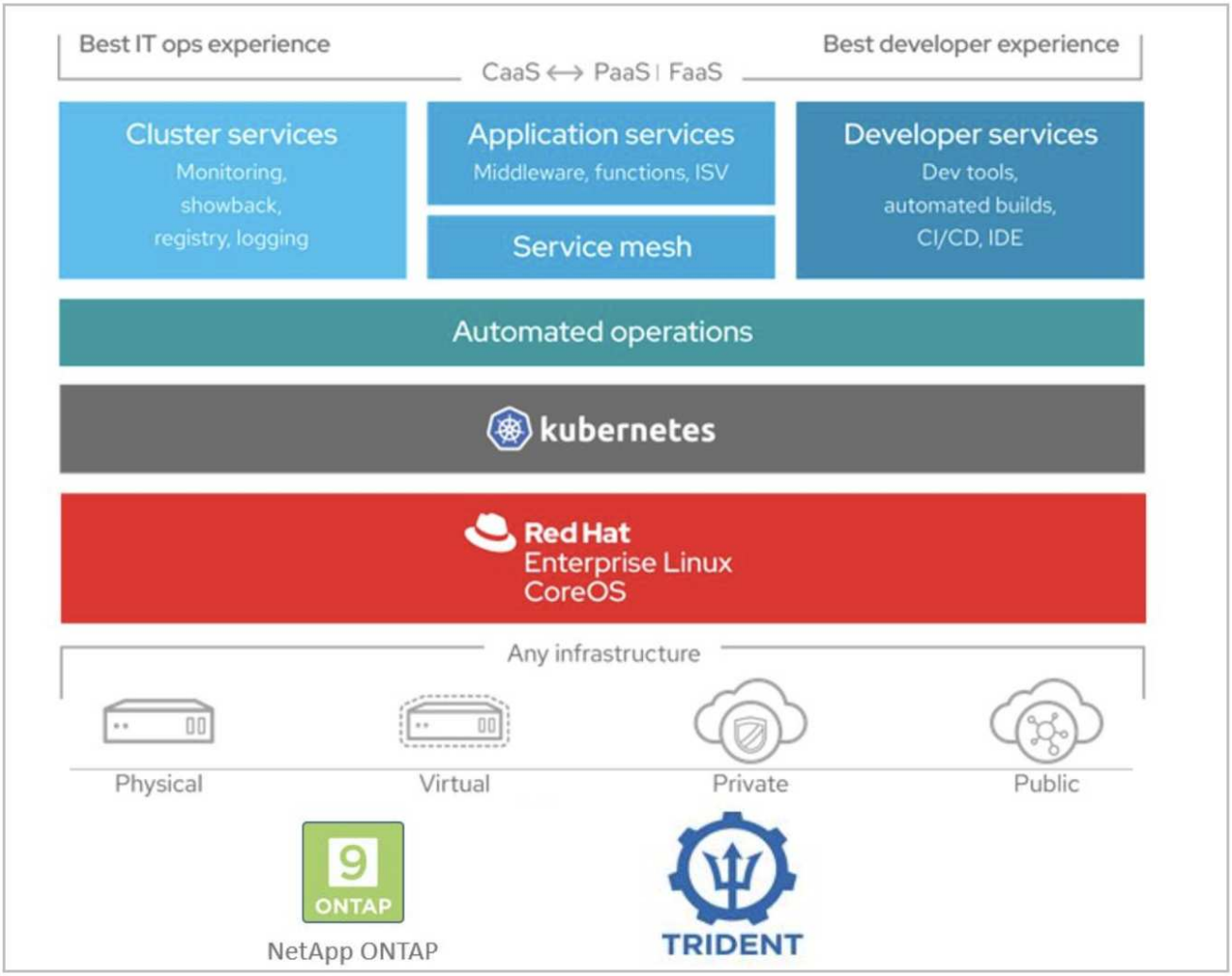
Element wurde für die Automatisierung entwickelt. Alle Speicherfunktionen sind über APIs verfügbar. Diese APIs sind die einzige Methode, die die Benutzeroberfläche zur Steuerung des Systems verwendet.

NetApp Storage-Integrationen

Erfahren Sie mehr über die Integration von NetApp Trident mit Red Hat OpenShift

Informieren Sie sich über NetApp Trident Protect, das für die Anwendungs- und persistente Speicherverwaltung für die OpenShift-Virtualisierungslösung validiert wurde.

Trident, ein von NetApp und NetApp Trident Protect gepflegter Open-Source-Speicherbereitsteller und -Orchestrator, hilft Ihnen bei der Orchestrierung und Verwaltung persistenter Daten in containerbasierten Umgebungen wie Red Hat OpenShift.



Auf den folgenden Seiten finden Sie zusätzliche Informationen zu den NetApp -Produkten, die für die

Anwendungs- und persistente Speicherverwaltung in der Red Hat OpenShift mit NetApp -Lösung validiert wurden:

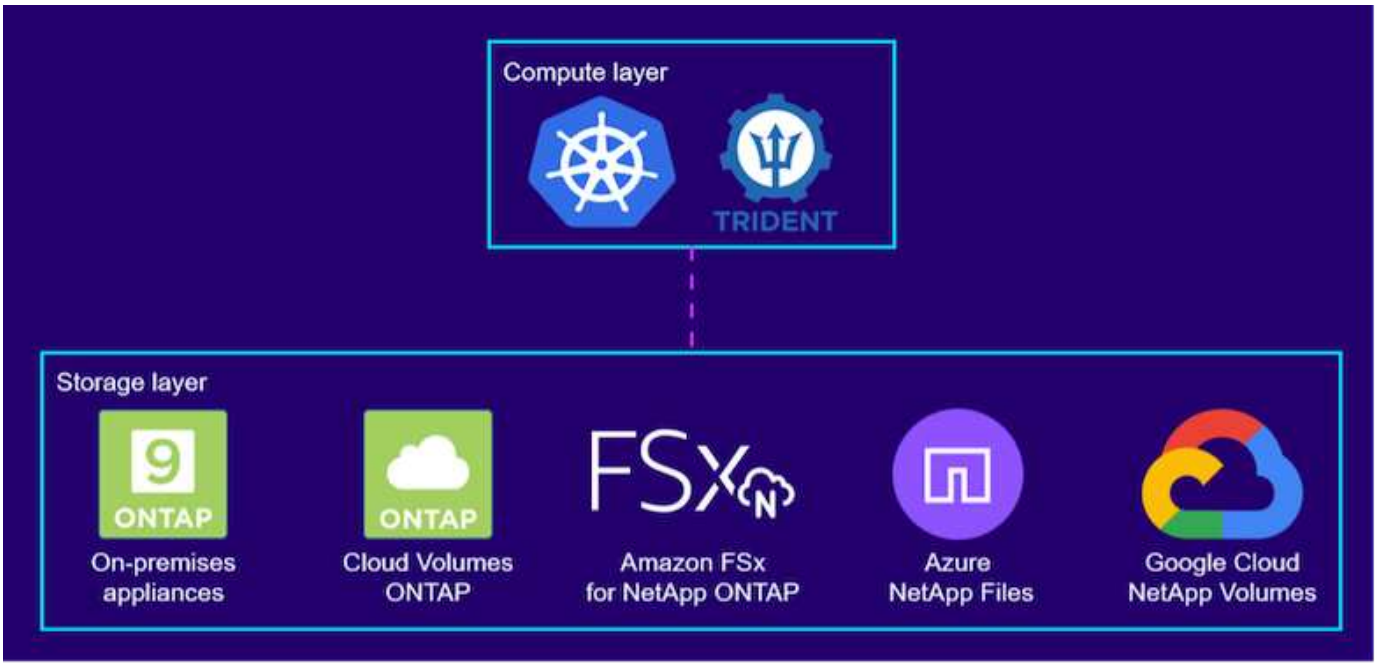
- "Trident -Dokumentation"
- "Trident Protect-Dokumentation"

NetApp Trident

Trident Übersicht

Trident ist ein Open-Source- und vollständig unterstützter Speicherorchestrator für Container und Kubernetes-Distributionen, einschließlich Red Hat OpenShift. Trident funktioniert mit dem gesamten NetApp -Speicherportfolio, einschließlich der NetApp ONTAP und Element-Speichersysteme, und unterstützt auch NFS- und iSCSI-Verbindungen. Trident beschleunigt den DevOps-Workflow, indem es Endbenutzern ermöglicht, Speicher von ihren NetApp -Speichersystemen bereitzustellen und zu verwalten, ohne dass ein Speicheradministrator eingreifen muss.

Ein Administrator kann basierend auf den Projektanforderungen und Speichersystemmodellen eine Reihe von Speicher-Backends konfigurieren, die erweiterte Speicherfunktionen ermöglichen, darunter Komprimierung, bestimmte Datenträgertypen oder QoS-Stufen, die ein bestimmtes Leistungsniveau garantieren. Nachdem sie definiert wurden, können diese Backends von Entwicklern in ihren Projekten verwendet werden, um Persistent Volume Claims (PVCs) zu erstellen und bei Bedarf persistenten Speicher an ihre Container anzuhängen.



Trident hat einen schnellen Entwicklungszyklus und wird genau wie Kubernetes viermal im Jahr veröffentlicht.

Eine Support-Matrix, welche Version von Trident mit welcher Kubernetes-Distribution getestet wurde, finden Sie "hier," .

Bitte beachten Sie die"Trident -Produktdokumentation" für Installations- und Konfigurationsdetails.

Trident herunterladen

Führen Sie die folgenden Schritte aus, um Trident auf dem bereitgestellten Benutzercluster zu installieren und ein persistentes Volume bereitzustellen:

1. Laden Sie das Installationsarchiv auf die Administrator-Workstation herunter und extrahieren Sie den Inhalt. Die aktuelle Version von Trident kann heruntergeladen werden "hier," .
2. Extrahieren Sie die Trident -Installation aus dem heruntergeladenen Paket.

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

Installieren Sie den Trident Operator mit Helm

1. Legen Sie zunächst den Standort des Benutzerclusters fest `kubeconfig` Datei als Umgebungsvariable, sodass Sie nicht darauf verweisen müssen, da Trident keine Möglichkeit hat, diese Datei zu übergeben.

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

2. Führen Sie den Helm-Befehl aus, um den Trident Operator aus dem Tarball im Helm-Verzeichnis zu installieren, während Sie den Trident-Namespace in Ihrem Benutzercluster erstellen.

```
[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. Sie können überprüfen, ob Trident erfolgreich installiert wurde, indem Sie die im Namespace ausgeführten Pods prüfen oder die installierte Version mithilfe der Binärdatei „tridentctl“ überprüfen.

```
[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 einigen Fällen erfordern Kundenumgebungen möglicherweise eine Anpassung der Trident Bereitstellung. In diesen Fällen ist es auch möglich, den Trident Operator manuell zu installieren und die enthaltenen Manifeste zu aktualisieren, um die Bereitstellung anzupassen.

Installieren Sie den Trident Operator manuell

1. Legen Sie zunächst den Standort des Benutzerclusters fest `kubeconfig` Datei als Umgebungsvariable, sodass Sie nicht darauf verweisen müssen, da Trident keine Möglichkeit hat, diese Datei zu übergeben.

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

2. Der `trident-installer` Das Verzeichnis enthält Manifeste zum Definieren aller erforderlichen Ressourcen. Erstellen Sie mit den entsprechenden Manifesten die `TridentOrchestrator` benutzerdefinierte Ressourcendefinition.

```
[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. Wenn keiner vorhanden ist, erstellen Sie mithilfe des bereitgestellten Manifests einen Trident -Namespace in Ihrem Cluster.

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

4. Erstellen Sie die für die Bereitstellung des Trident -Operators erforderlichen Ressourcen, z. B.

`ServiceAccount` für den Betreiber eine `ClusterRole` Und `ClusterRoleBinding` zum `ServiceAccount` , ein engagierter `PodSecurityPolicy` oder der Operator selbst.

```
[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. Sie können den Status des Operators nach seiner Bereitstellung mit den folgenden Befehlen überprüfen:

```
[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. Nachdem der Operator bereitgestellt wurde, können wir ihn jetzt zum Installieren von Trident verwenden. Dies erfordert die Erstellung eines `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. Sie können überprüfen, ob Trident erfolgreich installiert wurde, indem Sie die im Namespace ausgeführten Pods prüfen oder die installierte Version mithilfe der Binärdatei „tridentctl“ überprüfen.

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

Vorbereiten von Worker-Knoten für die Speicherung

NFS

Die meisten Kubernetes-Distributionen werden mit den Paketen und Dienstprogrammen zum Mounten von NFS-Backends geliefert, die standardmäßig installiert sind, einschließlich Red Hat OpenShift.

Für NFSv3 gibt es jedoch keinen Mechanismus zum Aushandeln der Parallelität zwischen Client und Server. Daher muss die maximale Anzahl clientseitiger SunRPC-Slot-Tabelleneinträge manuell mit dem unterstützten

Wert auf dem Server synchronisiert werden, um die beste Leistung für die NFS-Verbindung sicherzustellen, ohne dass der Server die Fenstergröße der Verbindung verringern muss.

Für ONTAP beträgt die unterstützte maximale Anzahl von SunRPC-Slot-Tabelleneinträgen 128, d. h. ONTAP kann 128 gleichzeitige NFS-Anfragen gleichzeitig bedienen. Standardmäßig verfügt Red Hat CoreOS/Red Hat Enterprise Linux jedoch über maximal 65.536 SunRPC-Slot-Tabelleneinträge pro Verbindung. Wir müssen diesen Wert auf 128 setzen. Dies kann mit dem Machine Config Operator (MCO) in OpenShift erfolgen.

Führen Sie die folgenden Schritte aus, um die maximalen SunRPC-Slot-Tabelleneinträge in OpenShift-Workerknoten zu ändern:

1. Melden Sie sich bei der OCP-Webkonsole an und navigieren Sie zu Compute > Machine Configs. Klicken Sie auf „Maschinenkonfiguration erstellen“. Kopieren Sie die YAML-Datei, fügen Sie sie ein und klicken Sie auf „Erstellen“.

```
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. Nachdem das MCO erstellt wurde, muss die Konfiguration auf allen Worker-Knoten angewendet und einzeln neu gestartet werden. Der gesamte Vorgang dauert etwa 20 bis 30 Minuten. Überprüfen Sie, ob die Maschinenkonfiguration angewendet wird, indem Sie `oc get mcp` und stellen Sie sicher, dass der Maschinenkonfigurationspool für Worker aktualisiert wird.

```
[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

Um Worker-Knoten für die Zuordnung von Blockspeicher-Volumes über das iSCSI-Protokoll vorzubereiten, müssen Sie die erforderlichen Pakete zur Unterstützung dieser Funktionalität installieren.

In Red Hat OpenShift wird dies durch Anwenden eines MCO (Machine Config Operator) auf Ihren Cluster nach der Bereitstellung erledigt.

Führen Sie die folgenden Schritte aus, um die Worker-Knoten für die Ausführung von iSCSI-Diensten zu konfigurieren:

1. Melden Sie sich bei der OCP-Webkonsole an und navigieren Sie zu Compute > Machine Configs. Klicken Sie auf „Maschinenkonfiguration erstellen“. Kopieren Sie die YAML-Datei, fügen Sie sie ein und klicken Sie auf „Erstellen“.

   Wenn Multipathing nicht verwendet wird:

   ```
   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: ""
   ```

   Bei Verwendung von 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. Nachdem die Konfiguration erstellt wurde, dauert es ungefähr 20 bis 30 Minuten, um die Konfiguration auf die Worker-Knoten anzuwenden und sie neu zu laden. Überprüfen Sie, ob die Maschinenkonfiguration angewendet wird, indem Sie `oc get mcp` und stellen Sie sicher, dass der Maschinenkonfigurationspool für Worker aktualisiert wird. Sie können sich auch bei den Worker-Knoten anmelden, um zu bestätigen, dass der iscsid-Dienst ausgeführt wird (und der multipathd-Dienst, wenn Sie Multipathing verwenden).

```
[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
```

Sie können auch bestätigen, dass die MachineConfig erfolgreich angewendet wurde und die Dienste wie erwartet gestartet wurden, indem Sie den `oc debug` Befehl mit den entsprechenden Flags.

Erstellen Sie Speichersystem-Backends

Nach Abschluss der Trident Operator-Installation müssen Sie das Backend für die von Ihnen verwendete

NetApp -Speicherplattform konfigurieren. Folgen Sie den unten stehenden Links, um mit der Einrichtung und Konfiguration von Trident fortzufahren.

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

NetApp ONTAP NFS-Konfiguration

Um die Trident -Integration mit dem NetApp ONTAP Speichersystem zu ermöglichen, müssen Sie ein Backend erstellen, das die Kommunikation mit dem Speichersystem ermöglicht.

1. Im heruntergeladenen Installationsarchiv im `sample-input` Ordnerhierarchie. Für NetApp ONTAP -Systeme, die NFS bedienen, kopieren Sie die `backend-ontap-nas.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[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. Bearbeiten Sie die Werte „backendName“, „managementLIF“, „dataLIF“, „svm“, „username“ und „password“ in dieser Datei.

```
{
    "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"
}
```

Es empfiehlt sich, den benutzerdefinierten BackendName-Wert zur einfachen Identifizierung als Kombination aus StorageDriverName und DataLIF zu definieren, das NFS bereitstellt.

3. Führen Sie mit dieser Backend-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[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. Nachdem das Backend erstellt wurde, müssen Sie als Nächstes eine Speicherklasse erstellen. Genau wie beim Backend gibt es im Ordner „sample-inputs“ eine Beispiel-Speicherklassendatei, die für die Umgebung bearbeitet werden kann. Kopieren Sie es in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen vor, um das erstellte Backend widerzuspiegeln.

```
[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. Die einzige Änderung, die an dieser Datei vorgenommen werden muss, ist die Definition der `backendType` Wert auf den Namen des Speichertreibers aus dem neu erstellten Backend. Beachten Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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

Es gibt ein optionales Feld namens `fsType` das in dieser Datei definiert ist. Diese Zeile kann in NFS-Backends gelöscht werden.

6. Führen Sie den `oc` Befehl zum Erstellen der Speicherklasse.

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

7. Nachdem Sie die Speicherklasse erstellt haben, müssen Sie den ersten Persistent Volume Claim (PVC) erstellen. Es gibt eine Probe `pvc-basic.yaml` Datei, die zum Ausführen dieser Aktion verwendet werden kann und sich ebenfalls in den Beispieleingaben befindet.

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

8. Die einzige Änderung, die an dieser Datei vorgenommen werden muss, besteht darin, sicherzustellen, dass die `storageClassName` Das Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf der bereitzustellenden Arbeitslast weiter angepasst werden.

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

9. Erstellen Sie den PVC, indem Sie die `oc` Befehl. Die Erstellung kann je nach Größe des zu erstellenden Sicherungsvolumes einige Zeit in Anspruch nehmen. Sie können den Vorgang daher während des Abschlusses verfolgen.

```
[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
```

NetApp ONTAP iSCSI-Konfiguration

Um die Trident -Integration mit dem NetApp ONTAP Speichersystem zu ermöglichen, müssen Sie ein Backend erstellen, das die Kommunikation mit dem Speichersystem ermöglicht.

1. Im heruntergeladenen Installationsarchiv im `sample-input` Ordnerhierarchie. Für NetApp ONTAP

-Systeme, die iSCSI bedienen, kopieren Sie die `backend-ontap-san.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

```
[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. Bearbeiten Sie die Werte „managementLIF“, „dataLIF“, „svm“, „username“ und „password“ in dieser Datei.

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

3. Führen Sie mit dieser Backend-Datei den folgenden Befehl aus, um Ihr erstes Backend zu erstellen.

```
[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. Nachdem das Backend erstellt wurde, müssen Sie als Nächstes eine Speicherklasse erstellen. Genau wie beim Backend gibt es im Ordner „sample-inputs“ eine Beispiel-Speicherklassendatei, die für die Umgebung bearbeitet werden kann. Kopieren Sie es in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen vor, um das erstellte Backend widerzuspiegeln.

```
[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. Die einzige Änderung, die an dieser Datei vorgenommen werden muss, ist die Definition der `backendType` Wert auf den Namen des Speichertreibers aus dem neu erstellten Backend. Beachten Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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

> Es gibt ein optionales Feld namens `fsType` das in dieser Datei definiert ist. In iSCSI-Backends kann dieser Wert auf einen bestimmten Linux-Dateisystemtyp (XFS, ext4 usw.) festgelegt oder gelöscht werden, damit OpenShift entscheiden kann, welches Dateisystem verwendet werden soll.

6. Führen Sie den `oc` Befehl zum Erstellen der Speicherklasse.

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

7. Nachdem Sie die Speicherklasse erstellt haben, müssen Sie den ersten Persistent Volume Claim (PVC) erstellen. Es gibt eine Probe `pvc-basic.yaml` Datei, die zum Ausführen dieser Aktion verwendet werden kann und sich ebenfalls in den Beispieleingaben befindet.

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

8. Die einzige Änderung, die an dieser Datei vorgenommen werden muss, besteht darin, sicherzustellen, dass die `storageClassName` Das Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf der bereitzustellenden Arbeitslast weiter angepasst werden.

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

9. Erstellen Sie den PVC, indem Sie die `oc` Befehl. Die Erstellung kann je nach Größe des zu erstellenden Sicherungsvolumes einige Zeit in Anspruch nehmen. Sie können den Vorgang daher während des Abschlusses verfolgen.

```
[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
```

NetApp Element iSCSI-Konfiguration

Um die Trident Integration mit dem NetApp Element -Speichersystem zu ermöglichen, müssen Sie ein Backend erstellen, das die Kommunikation mit dem Speichersystem über das iSCSI-Protokoll ermöglicht.

1. Im heruntergeladenen Installationsarchiv im `sample-input` Ordnerhierarchie. Für NetApp Element -Systeme, die iSCSI bereitstellen, kopieren Sie die `backend-solidfire.json` Datei in Ihr Arbeitsverzeichnis und bearbeiten Sie die Datei.

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

   a. Bearbeiten Sie den Benutzer, das Passwort und den MVIP-Wert auf der `EndPoint` Linie.

   b. Bearbeiten Sie die `SVIP` Wert.

```
 {
    "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. Führen Sie mit dieser Back-End-Datei den folgenden Befehl aus, um Ihr erstes Back-End zu erstellen.

```
[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. Nachdem das Backend erstellt wurde, müssen Sie als Nächstes eine Speicherklasse erstellen. Genau wie beim Backend gibt es im Ordner „sample-inputs“ eine Beispiel-Speicherklassendatei, die für die Umgebung bearbeitet werden kann. Kopieren Sie es in das Arbeitsverzeichnis und nehmen Sie die erforderlichen Änderungen vor, um das erstellte Backend widerzuspiegeln.

```
[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. Die einzige Änderung, die an dieser Datei vorgenommen werden muss, ist die Definition der `backendType` Wert auf den Namen des Speichertreibers aus dem neu erstellten Backend. Beachten Sie auch den Wert des Namensfelds, auf den in einem späteren Schritt verwiesen werden muss.

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

Es gibt ein optionales Feld namens `fsType` das in dieser Datei definiert ist. In iSCSI-Backends kann dieser Wert auf einen bestimmten Linux-Dateisystemtyp (XFS, ext4 usw.) festgelegt oder gelöscht werden, damit OpenShift entscheiden kann, welches Dateisystem verwendet werden soll.

5. Führen Sie den `oc` Befehl zum Erstellen der Speicherklasse.

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

6. Nachdem Sie die Speicherklasse erstellt haben, müssen Sie den ersten Persistent Volume Claim (PVC) erstellen. Es gibt eine Probe `pvc-basic.yaml` Datei, die zum Ausführen dieser Aktion verwendet werden kann und sich ebenfalls in den Beispieleingaben befindet.

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

7. Die einzige Änderung, die an dieser Datei vorgenommen werden muss, besteht darin, sicherzustellen, dass die `storageClassName` Das Feld entspricht dem gerade erstellten. Die PVC-Definition kann je nach Bedarf der bereitzustellenden Arbeitslast weiter angepasst werden.

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

8. Erstellen Sie den PVC, indem Sie die `oc` Befehl. Die Erstellung kann je nach Größe des zu erstellenden Sicherungsvolumes einige Zeit in Anspruch nehmen. Sie können den Vorgang daher während des Abschlusses verfolgen.

```
[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
```

Erweiterte Konfigurationsoptionen

Entdecken Sie die Load Balancer-Optionen

Load Balancer-Optionen erkunden: Red Hat OpenShift mit NetApp

In den meisten Fällen stellt Red Hat OpenShift Anwendungen über Routen der Außenwelt zur Verfügung. Ein Dienst wird verfügbar gemacht, indem ihm ein extern erreichbarer Hostname zugewiesen wird. Die definierte Route und die von ihrem Dienst identifizierten Endpunkte können von einem OpenShift-Router genutzt werden, um diese benannte Konnektivität für externe Clients bereitzustellen.

In einigen Fällen erfordern Anwendungen jedoch die Bereitstellung und Konfiguration angepasster Lastenausgleichsmodule, um die entsprechenden Dienste bereitzustellen. Ein Beispiel hierfür ist NetApp Trident Protect. Um diesem Bedarf gerecht zu werden, haben wir eine Reihe von benutzerdefinierten Load Balancer-Optionen evaluiert. Ihre Installation und Konfiguration werden in diesem Abschnitt beschrieben.

Auf den folgenden Seiten finden Sie zusätzliche Informationen zu den Load Balancer-Optionen, die in der Red Hat OpenShift-Lösung mit NetApp validiert wurden:

- "MetalLB"
- "F5 BIG-IP"

Installieren von MetalLB-Load Balancern: Red Hat OpenShift mit NetApp

Auf dieser Seite finden Sie die Installations- und Konfigurationsanweisungen für den MetalLB-Load Balancer.

MetalLB ist ein selbstgehosteter Netzwerk-Load Balancer, der auf Ihrem OpenShift-Cluster installiert wird und die Erstellung von OpenShift-Diensten vom Typ Load Balancer in Clustern ermöglicht, die nicht auf einem Cloud-Anbieter ausgeführt werden. Die beiden Hauptfunktionen von MetalLB, die zusammenarbeiten, um LoadBalancer-Dienste zu unterstützen, sind Adresszuweisung und externe Ankündigung.

MetalLB-Konfigurationsoptionen

Basierend darauf, wie MetalLB die den LoadBalancer-Diensten außerhalb des OpenShift-Clusters zugewiesene IP-Adresse ankündigt, arbeitet es in zwei Modi:

- **Layer 2-Modus.** In diesem Modus übernimmt ein Knoten im OpenShift-Cluster den Besitz des Dienstes und antwortet auf ARP-Anfragen für diese IP, um sie außerhalb des OpenShift-Clusters erreichbar zu machen. Da nur der Knoten die IP ankündigt, kommt es zu einem Bandbreitenengpass und langsamen Failover-Einschränkungen. Weitere Informationen finden Sie in der Dokumentation"hier," .
- **BGP-Modus.** In diesem Modus stellen alle Knoten im OpenShift-Cluster BGP-Peering-Sitzungen mit einem Router her und geben die Routen bekannt, um den Datenverkehr an die Service-IPs weiterzuleiten. Voraussetzung hierfür ist die Integration von MetalLB mit einem Router in das Netzwerk. Aufgrund des Hashing-Mechanismus in BGP gibt es gewisse Einschränkungen, wenn sich die IP-zu-Knoten-Zuordnung für einen Dienst ändert. Weitere Informationen finden Sie in der Dokumentation"hier," .

Für die Zwecke dieses Dokuments konfigurieren wir MetalLB im Layer-2-Modus.

Installieren des MetalLB Load Balancers

1. Laden Sie die MetalLB-Ressourcen herunter.

```
[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. Datei bearbeiten `metallb.yaml` und entfernen `spec.template.spec.securityContext` vom Controller-Deployment und dem Speaker-DaemonSet.

   Zu löschende Zeilen:

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

3. Erstellen Sie die `metallb-system` Namespace.

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

4. Erstellen Sie den MetalLB CR.

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

5. Bevor Sie den MetalLB-Lautsprecher konfigurieren, erteilen Sie dem Lautsprecher-DaemonSet erhöhte Berechtigungen, damit er die Netzwerkkonfiguration durchführen kann, die für die Funktion der Lastenausgleichsmodule erforderlich ist.

```
[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. Konfigurieren Sie MetalLB, indem Sie eine `ConfigMap` im `metallb-system` Namespace.

```
[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. Wenn jetzt Loadbalancer-Dienste erstellt werden, weist MetalLB den Diensten eine externe IP zu und gibt die IP-Adresse bekannt, indem es auf ARP-Anfragen antwortet.

> Wenn Sie MetalLB im BGP-Modus konfigurieren möchten, überspringen Sie Schritt 6 oben und folgen Sie dem Verfahren in der MetalLB-Dokumentation"hier," .

Installieren von F5 BIG-IP Load Balancern

F5 BIG-IP ist ein Application Delivery Controller (ADC), der eine breite Palette fortschrittlicher Verkehrsmanagement- und Sicherheitsdienste in Produktionsqualität bietet, wie z. B. L4-L7-Lastausgleich, SSL/TLS-Offload, DNS, Firewall und vieles mehr. Diese Dienste erhöhen die Verfügbarkeit, Sicherheit und Leistung Ihrer Anwendungen drastisch.

F5 BIG-IP kann auf verschiedene Weise bereitgestellt und genutzt werden: auf dedizierter Hardware, in der Cloud oder als virtuelle Appliance vor Ort. Lesen Sie die Dokumentation hier, um F5 BIG-IP je nach Bedarf zu erkunden und bereitzustellen.

Für eine effiziente Integration von F5 BIG-IP-Diensten mit Red Hat OpenShift bietet F5 den BIG-IP Container Ingress Service (CIS) an. CIS wird als Controller-Pod installiert, der die OpenShift-API auf bestimmte benutzerdefinierte Ressourcendefinitionen (CRDs) überwacht und die F5 BIG-IP-Systemkonfiguration verwaltet. F5 BIG-IP CIS kann so konfiguriert werden, dass es die Servicetypen LoadBalancer und Routen in OpenShift steuert.

Darüber hinaus können Sie für die automatische IP-Adresszuweisung zur Bedienung des Typs LoadBalancer den F5 IPAM-Controller verwenden. Der F5 IPAM-Controller wird als Controller-Pod installiert, der die OpenShift-API auf LoadBalancer-Dienste mit einer ipamLabel-Annotation überwacht, um die IP-Adresse aus einem vorkonfigurierten Pool zuzuweisen.

Auf dieser Seite sind die Installations- und Konfigurationsanweisungen für den F5 BIG-IP CIS- und IPAM-

Controller aufgeführt. Als Voraussetzung müssen Sie über ein bereitgestelltes und lizenziertes F5 BIG-IP-System verfügen. Es muss auch für SDN-Dienste lizenziert werden, die standardmäßig in der BIG-IP VE-Basislizenz enthalten sind.

F5 BIG-IP kann im Standalone- oder Clustermodus bereitgestellt werden. Für die Zwecke dieser Validierung wurde F5 BIG-IP im Standalone-Modus bereitgestellt. Für Produktionszwecke ist jedoch ein Cluster von BIG-IPs vorzuziehen, um einen einzelnen Fehlerpunkt zu vermeiden.

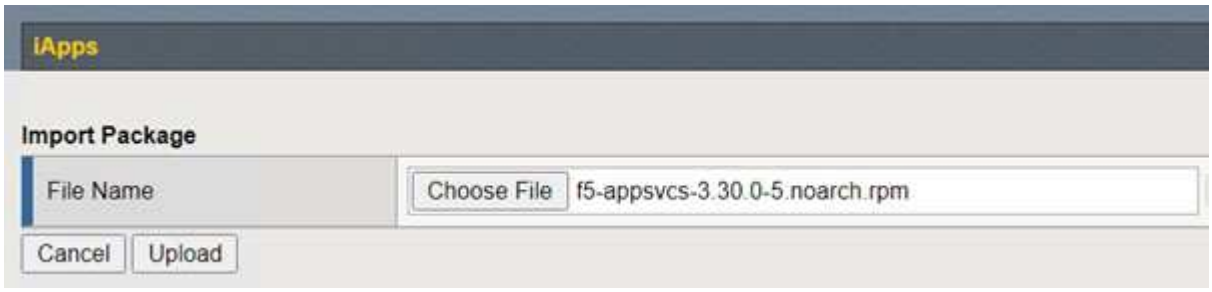
Ein F5 BIG-IP-System kann auf dedizierter Hardware, in der Cloud oder als virtuelle Appliance vor Ort mit Versionen über 12.x bereitgestellt werden, um es in F5 CIS zu integrieren. Für die Zwecke dieses Dokuments wurde das F5 BIG-IP-System als virtuelle Appliance validiert, beispielsweise mithilfe der BIG-IP VE-Edition.

Validierte Releases

Technologie	Softwareversion
Red Hat OpenShift	4,6 EUS, 4,7
F5 BIG-IP VE-Edition	16.1.0
F5 Container Ingress-Dienst	2.5.1
F5 IPAM-Controller	0,1,4
F5 AS3	3.30.0

Installation

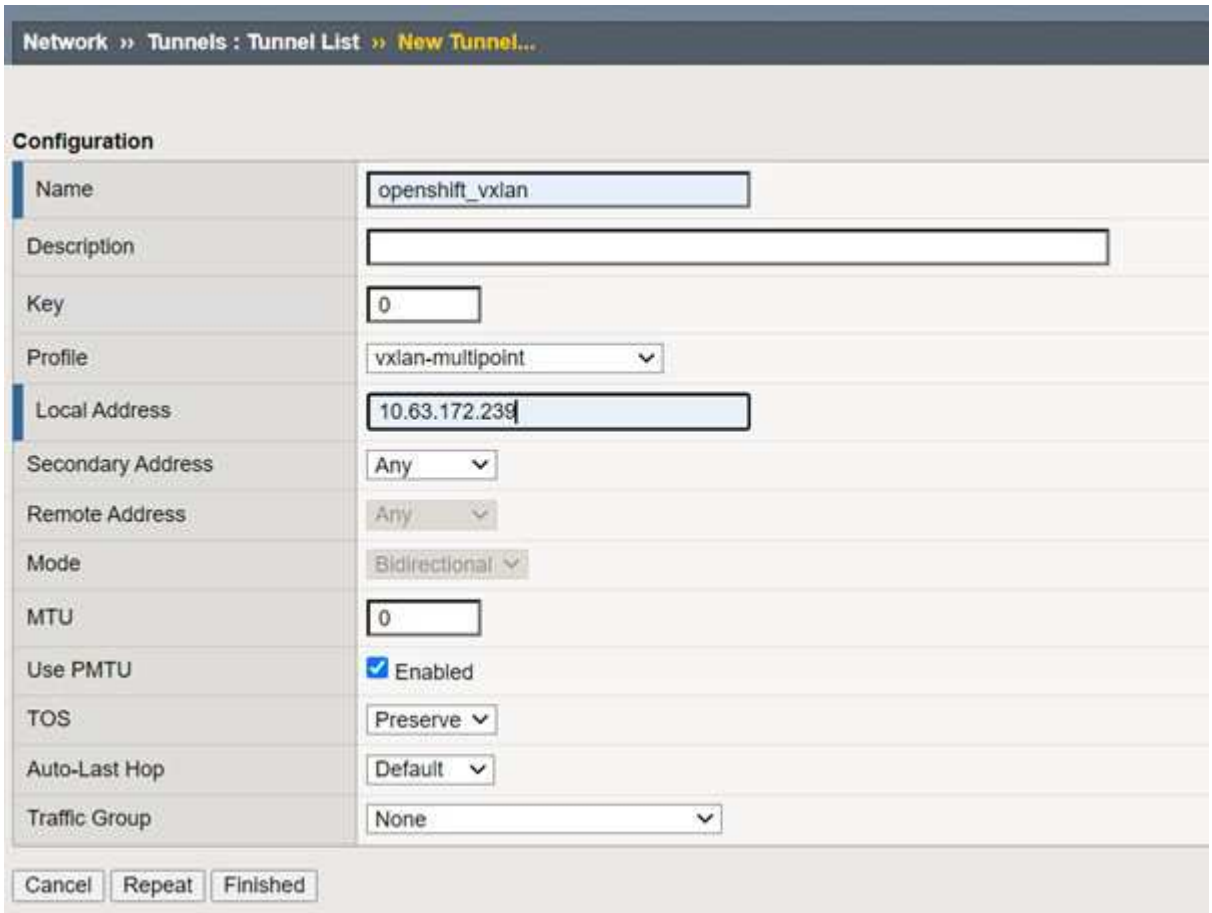
1. Installieren Sie die Erweiterung F5 Application Services 3, damit BIG-IP-Systeme Konfigurationen in JSON anstelle von imperativen Befehlen akzeptieren können. Gehe zu "F5 AS3 GitHub-Repository" , und laden Sie die neueste RPM-Datei herunter.
2. Melden Sie sich beim F5 BIG-IP-System an, navigieren Sie zu iApps > Package Management LX und klicken Sie auf Importieren.
3. Klicken Sie auf „Datei auswählen“ und wählen Sie die heruntergeladene AS3-RPM-Datei aus. Klicken Sie auf „OK“ und dann auf „Hochladen“.



4. Bestätigen Sie, dass die AS3-Erweiterung erfolgreich installiert wurde.

5. Konfigurieren Sie als Nächstes die für die Kommunikation zwischen OpenShift- und BIG-IP-Systemen erforderlichen Ressourcen. Erstellen Sie zunächst einen Tunnel zwischen OpenShift und dem BIG-IP-Server, indem Sie auf dem BIG-IP-System eine VXLAN-Tunnelschnittstelle für OpenShift SDN erstellen. Navigieren Sie zu Netzwerk > Tunnel > Profile, klicken Sie auf Erstellen und legen Sie das übergeordnete Profil auf vxlan und den Flooding-Typ auf Multicast fest. Geben Sie einen Namen für das Profil ein und klicken Sie auf „Fertig“.

6. Navigieren Sie zu Netzwerk > Tunnel > Tunnelliste, klicken Sie auf Erstellen und geben Sie den Namen und die lokale IP-Adresse für den Tunnel ein. Wählen Sie das im vorherigen Schritt erstellte Tunnelprofil aus und klicken Sie auf „Fertig“.



7. Melden Sie sich mit Cluster-Administratorberechtigungen beim Red Hat OpenShift-Cluster an.
8. Erstellen Sie auf OpenShift ein Hostsubnetz für den F5 BIG-IP-Server, das das Subnetz vom OpenShift-Cluster auf den F5 BIG-IP-Server erweitert. Laden Sie die YAML-Definition des Host-Subnetzes herunter.

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

9. Bearbeiten Sie die Host-Subnetzdatei und fügen Sie die BIG-IP VTEP (VXLAN-Tunnel)-IP für das OpenShift SDN hinzu.

```
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
```

Ändern Sie die Host-IP und andere Details entsprechend Ihrer Umgebung.

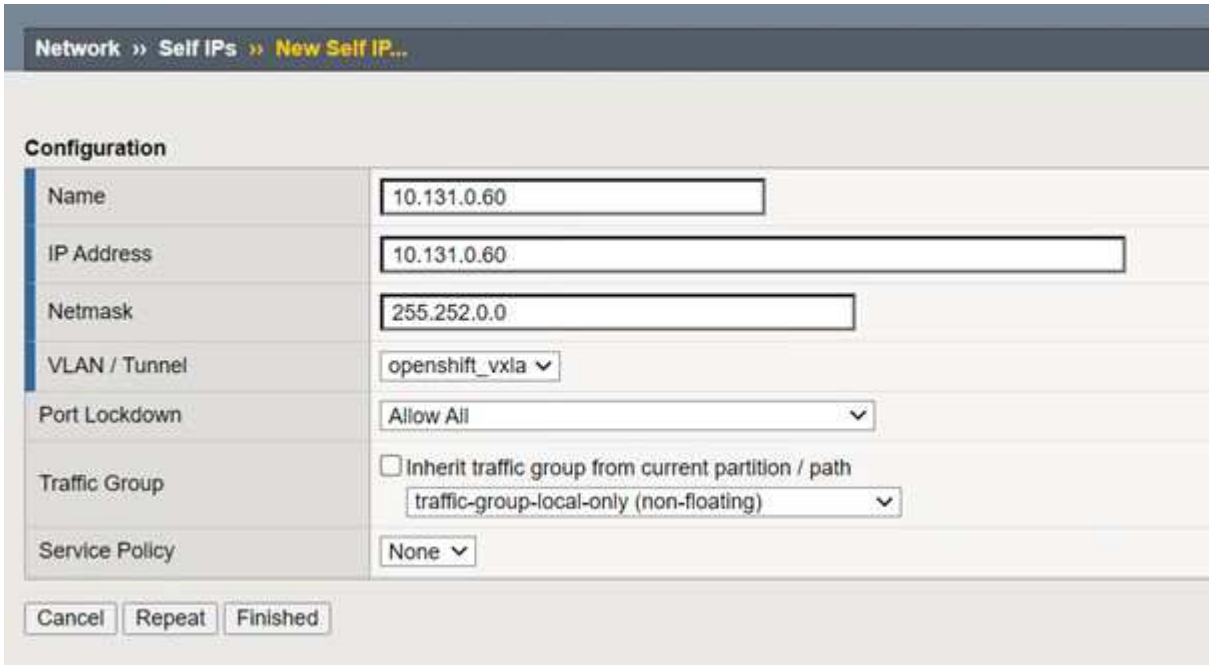
10. Erstellen Sie die HostSubnet-Ressource.

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

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

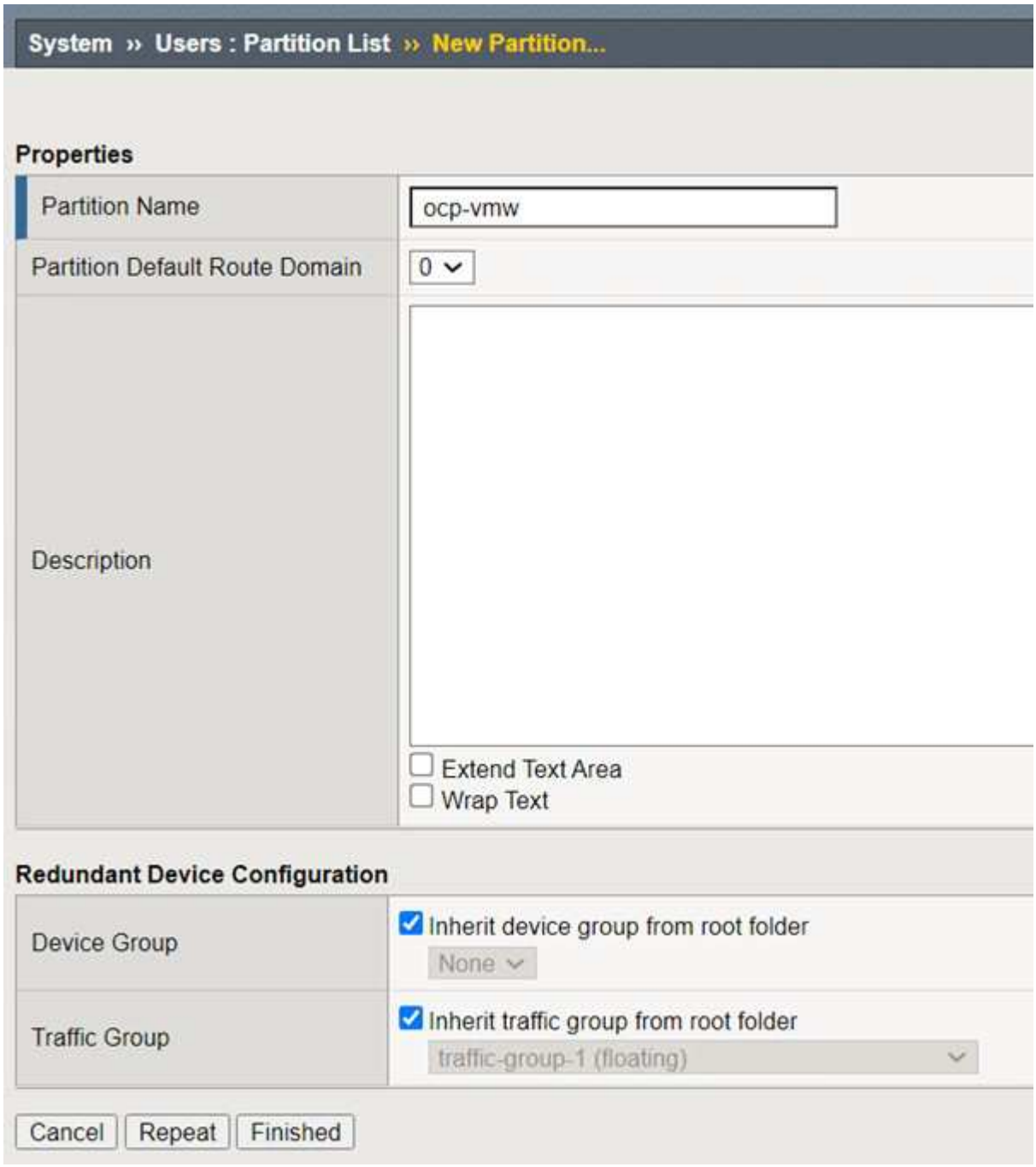
11. Rufen Sie den Cluster-IP-Subnetzbereich für das für den F5 BIG-IP-Server erstellte Host-Subnetz ab.

```
[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. Erstellen Sie eine eigene IP auf OpenShift VXLAN mit einer IP im Host-Subnetzbereich von OpenShift, die dem F5 BIG-IP-Server entspricht. Melden Sie sich beim F5 BIG-IP-System an, navigieren Sie zu Netzwerk > Eigene IPs und klicken Sie auf Erstellen. Geben Sie eine IP aus dem für das F5 BIG-IP-Host-Subnetz erstellten Cluster-IP-Subnetz ein, wählen Sie den VXLAN-Tunnel aus und geben Sie die anderen Details ein. Klicken Sie dann auf „Fertig“.



13. Erstellen Sie eine Partition im F5 BIG-IP-System, die mit CIS konfiguriert und verwendet werden soll. Navigieren Sie zu System > Benutzer > Partitionsliste, klicken Sie auf Erstellen und geben Sie die Details ein. Klicken Sie dann auf „Fertig“.



F5 empfiehlt, auf der von CIS verwalteten Partition keine manuelle Konfiguration vorzunehmen.

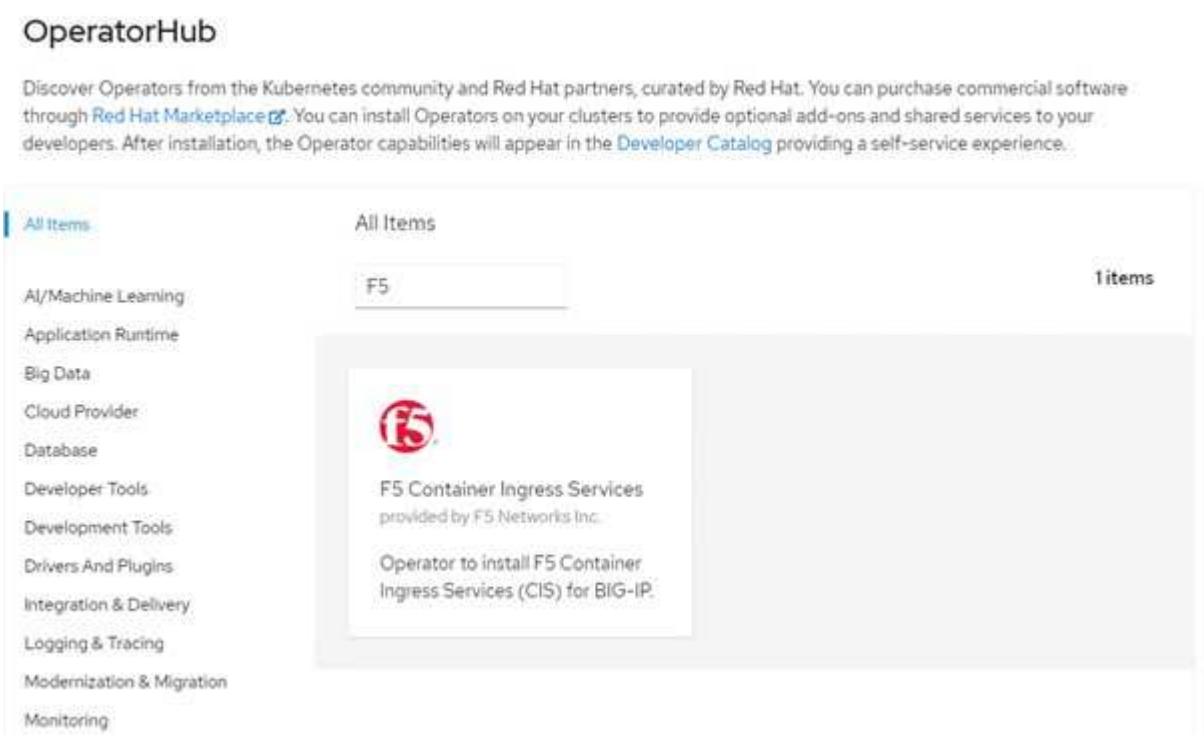
14. Installieren Sie das F5 BIG-IP CIS mit dem Operator von OperatorHub. Melden Sie sich mit Cluster-Admin-Berechtigungen beim Red Hat OpenShift-Cluster an und erstellen Sie ein Geheimnis mit den Anmeldeinformationen des F5 BIG-IP-Systems. Dies ist eine Voraussetzung für den Operator.

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

secret/bigip-login created
```

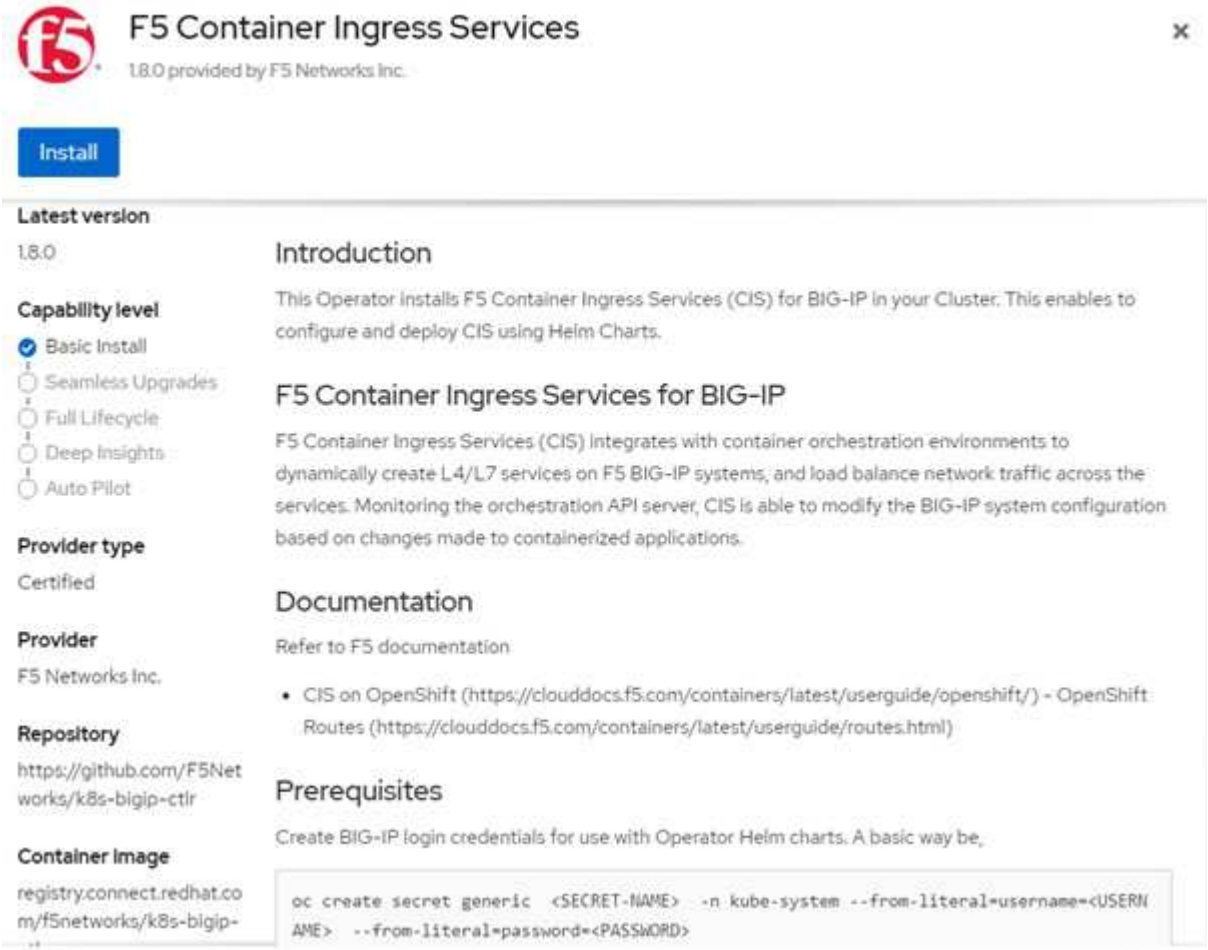
15. Installieren Sie die F5 CIS CRDs.

```
[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
```

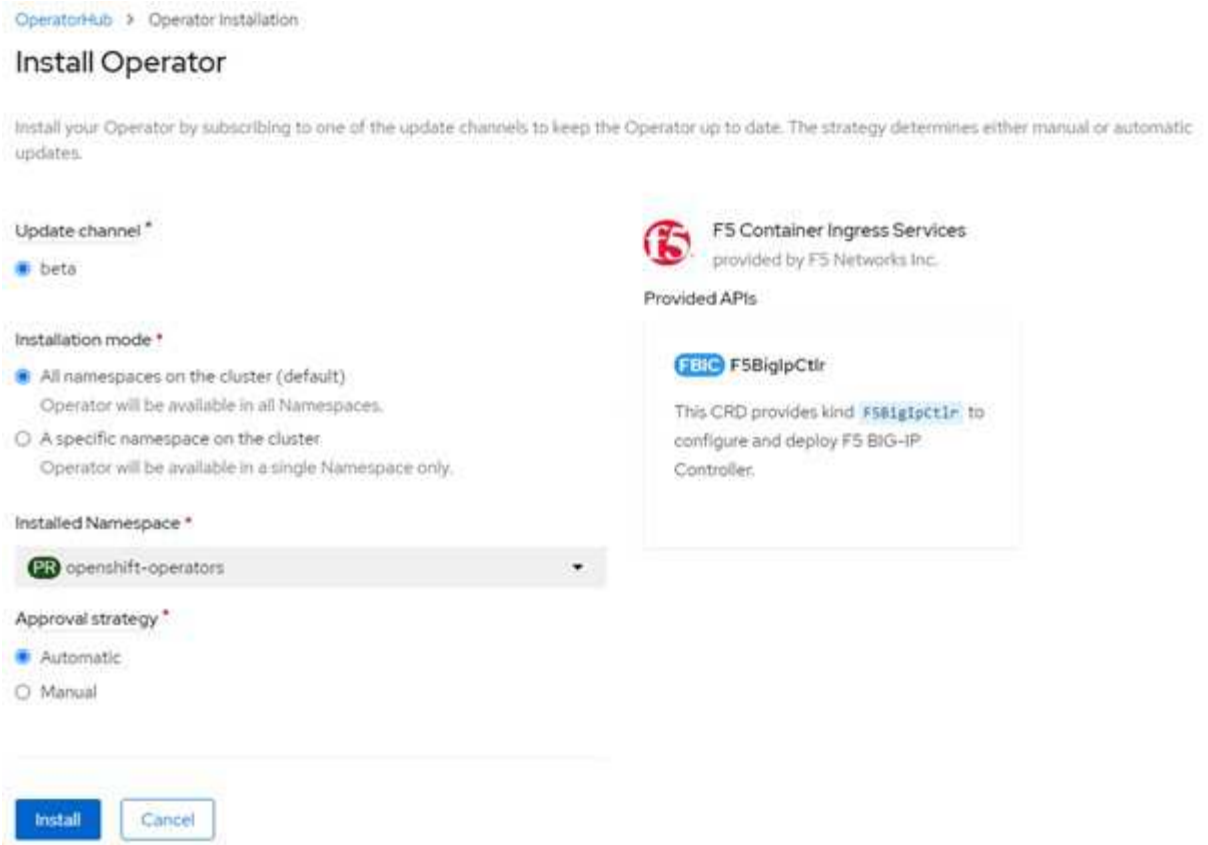
16. Navigieren Sie zu Operatoren > OperatorHub, suchen Sie nach dem Schlüsselwort F5 und klicken Sie auf die Kachel F5 Container Ingress Service.



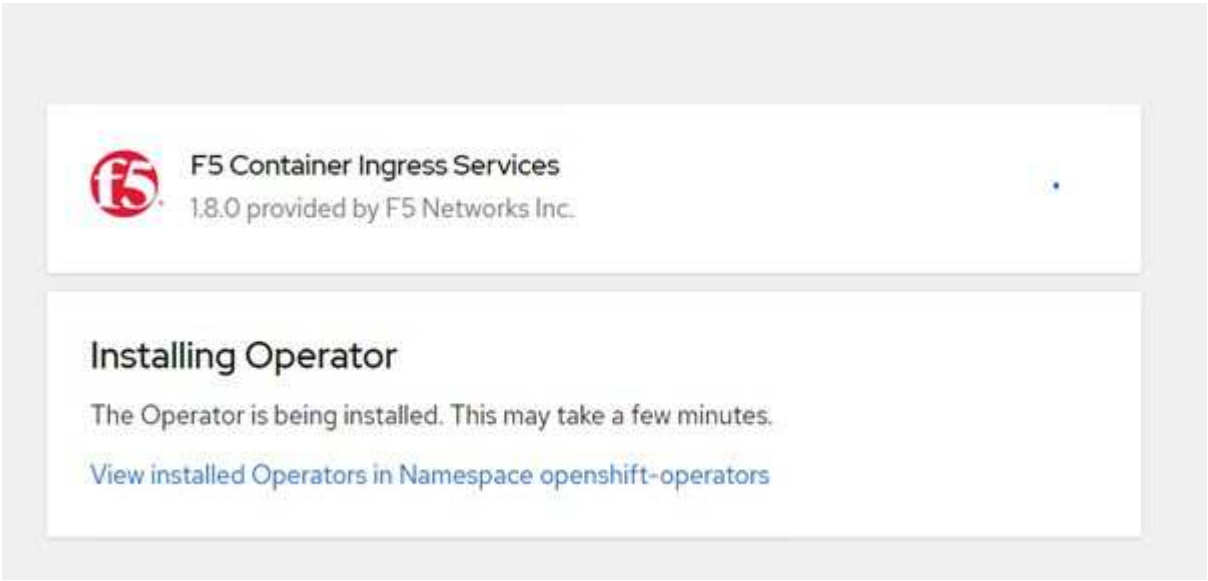
17. Lesen Sie die Betreiberinformationen und klicken Sie auf Installieren.



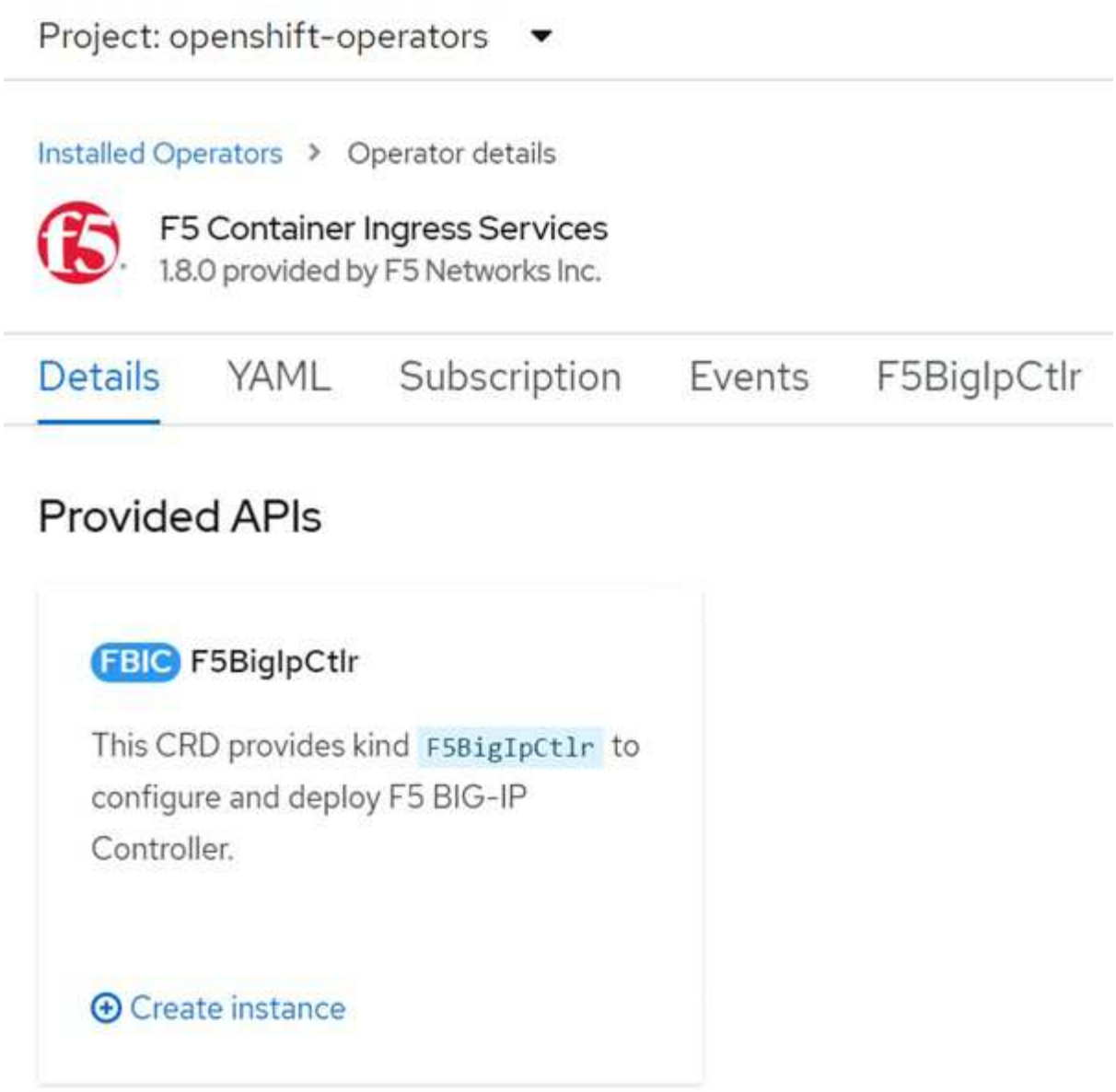
18. Behalten Sie auf dem Bildschirm „Operator installieren“ alle Standardparameter bei und klicken Sie auf „Installieren“.



19. Die Installation des Operators dauert eine Weile.



20. Nachdem der Operator installiert wurde, wird die Meldung „Installation erfolgreich“ angezeigt.
21. Navigieren Sie zu „Operatoren > Installierte Operatoren“, klicken Sie auf „F5 Container Ingress Service“ und dann unter der Kachel „F5BigIpCtlr“ auf „Instanz erstellen“.



22. Klicken Sie auf „YAML-Ansicht“ und fügen Sie den folgenden Inhalt ein, nachdem Sie die erforderlichen Parameter aktualisiert haben.

> Aktualisieren der Parameter `bigip_partition` , ` openshift_sdn_name`, `bigip_url` Und `bigip_login_secret` unten, um die Werte für Ihr Setup widerzuspiegeln, bevor Sie den Inhalt kopieren.

```
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. Klicken Sie nach dem Einfügen dieses Inhalts auf „Erstellen“. Dadurch werden die CIS-Pods im Kube-System-Namespace installiert.

Red Hat OpenShift bietet standardmäßig eine Möglichkeit, die Dienste über Routen für den L7-Lastausgleich verfügbar zu machen. Ein integrierter OpenShift-Router ist für die Werbung und die Verkehrsabwicklung dieser Routen verantwortlich. Sie können das F5 CIS jedoch auch so konfigurieren, dass die Routen über ein externes F5 BIG-IP-System unterstützt werden, das entweder als Hilfsrouter oder als Ersatz für den selbst gehosteten OpenShift-Router ausgeführt werden kann. CIS erstellt einen virtuellen Server im BIG-IP-System, der als Router für die OpenShift-Routen fungiert, und BIG-IP übernimmt die Werbung und das Verkehrsrouting. Informationen zu den Parametern zum Aktivieren dieser Funktion finden Sie in der Dokumentation hier. Beachten Sie, dass diese Parameter für die OpenShift-Bereitstellungsressource in der Apps/v1-API definiert sind. Ersetzen Sie daher bei der Verwendung mit der F5BigIpCtlr-Ressource cis.f5.com/v1-API die Bindestriche (-) durch Unterstriche (_) für die Parameternamen.

24. Zu den Argumenten, die an die Erstellung von CIS-Ressourcen übergeben werden, gehören `ipam: true` Und `custom_resource_mode: true` . Diese Parameter sind erforderlich, um die CIS-Integration mit einem IPAM-Controller zu aktivieren. Überprüfen Sie, ob das CIS die IPAM-Integration aktiviert hat, indem Sie die F5 IPAM-Ressource erstellen.

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

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

25. Erstellen Sie das für den F5 IPAM-Controller erforderliche Dienstkonto, die Rolle und die Rollenbindung. Erstellen Sie eine YAML-Datei und fügen Sie den folgenden Inhalt ein.

```
[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. Erstellen Sie die Ressourcen.

```
[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. Erstellen Sie eine YAML-Datei und fügen Sie die unten angegebene F5 IPAM-Bereitstellungsdefinition ein.

Aktualisieren Sie den IP-Bereichsparameter in spec.template.spec.containers[0].args unten, um die IPamLabels und IP-Adressbereiche entsprechend Ihrem Setup widerzuspiegeln.

ipamLabels[`range1` Und `range2` im folgenden Beispiel] müssen für die Dienste vom Typ LoadBalancer annotiert werden, damit der IPAM-Controller eine IP-Adresse aus dem definierten Bereich erkennen und zuweisen kann.

```
[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. Erstellen Sie die F5 IPAM-Controller-Bereitstellung.

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

deployment/f5-ipam-controller created
```

29. Überprüfen Sie, ob die F5 IPAM-Controller-Pods ausgeführt werden.

```
[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. Erstellen Sie das F5 IPAM-Schema.

```
[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
```

Überprüfung

1. Erstellen Sie einen Dienst vom Typ 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. Überprüfen Sie, ob der IPAM-Controller ihm eine externe IP zuweist.

```
[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. Erstellen Sie eine Bereitstellung und verwenden Sie den erstellten LoadBalancer-Dienst.

```
[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. Überprüfen Sie, ob die Pods ausgeführt werden.

```
[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. Prüfen Sie, ob für den Dienst vom Typ LoadBalancer in OpenShift der entsprechende virtuelle Server im BIG-IP-System angelegt ist. Navigieren Sie zu Lokaler Datenverkehr > Virtuelle Server > Liste der

virtuellen Server.

Erstellen privater Image-Registrierungen

Für die meisten Bereitstellungen von Red Hat OpenShift wird ein öffentliches Register wie "Quay.io" oder "DockerHub" erfüllt die Bedürfnisse der meisten Kunden. Es gibt jedoch Fälle, in denen ein Kunde seine eigenen privaten oder benutzerdefinierten Bilder hosten möchte.

Dieses Verfahren dokumentiert die Erstellung einer privaten Image-Registrierung, die durch ein persistentes Volume von Trident und NetApp ONTAP unterstützt wird.

> Trident Protect erfordert ein Register zum Hosten der von den Astra Containern benötigten Bilder. Der folgende Abschnitt beschreibt die Schritte zum Einrichten eines privaten Registers auf einem Red Hat OpenShift-Cluster und zum Übertragen der Images, die zur Unterstützung der Installation von Trident Protect erforderlich sind.

Erstellen einer privaten Bildregistrierung

1. Entfernen Sie die Standardannotation aus der aktuellen Standardspeicherklasse und kommentieren Sie die von Trident unterstützte Speicherklasse als Standard für den OpenShift-Cluster.

```
[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. Bearbeiten Sie den Imageregistry-Operator, indem Sie die folgenden Speicherparameter in das Feld `spec` Abschnitt.

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

storage:
  pvc:
    claim:
```

3. Geben Sie die folgenden Parameter in das `spec` Abschnitt zum Erstellen einer OpenShift-Route mit einem benutzerdefinierten Hostnamen. Speichern und beenden.

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

Die obige Routenkonfiguration wird verwendet, wenn Sie einen benutzerdefinierten Hostnamen für Ihre Route wünschen. Wenn Sie möchten, dass OpenShift eine Route mit einem Standardhostnamen erstellt, können Sie die folgenden Parameter zum `spec` Abschnitt: `defaultRoute: true` .

Benutzerdefinierte TLS-Zertifikate

Wenn Sie einen benutzerdefinierten Hostnamen für die Route verwenden, wird standardmäßig die Standard-TLS-Konfiguration des OpenShift Ingress-Operators verwendet. Sie können der Route jedoch eine benutzerdefinierte TLS-Konfiguration hinzufügen. Führen Sie dazu die folgenden Schritte aus.

a. Erstellen Sie ein Geheimnis mit den TLS-Zertifikaten und dem Schlüssel der Route.

```
[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. Bearbeiten Sie den Imageregistry-Operator und fügen Sie die folgenden Parameter hinzu `spec` Abschnitt.

```
[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. Bearbeiten Sie den Imageregistry-Operator erneut und ändern Sie den Verwaltungsstatus des Operators in `Managed` Zustand. Speichern und beenden.

```
oc edit configs.imageregistry/cluster

managementState: Managed
```

5. Wenn alle Voraussetzungen erfüllt sind, werden PVCs, Pods und Dienste für die private Image-Registrierung erstellt. In wenigen Minuten sollte die Registrierung verfügbar sein.

```
[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. Wenn Sie die Standard-TLS-Zertifikate für die OpenShift-Registrierungsroute des Ingress-Operators verwenden, können Sie die TLS-Zertifikate mit dem folgenden Befehl abrufen.

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

7. Um OpenShift-Knoten den Zugriff auf die Images und deren Abruf aus der Registrierung zu ermöglichen, fügen Sie die Zertifikate dem Docker-Client auf den OpenShift-Knoten hinzu. Erstellen Sie eine Konfigurationskarte im `openshift-config` Namespace mithilfe der TLS-Zertifikate und patchen Sie es in die Cluster-Image-Konfiguration, um das Zertifikat vertrauenswürdig zu machen.

```
[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. Die interne OpenShift-Registrierung wird durch Authentifizierung gesteuert. Alle OpenShift-Benutzer können auf die OpenShift-Registrierung zugreifen, aber die Vorgänge, die der angemeldete Benutzer ausführen kann, hängen von den Benutzerberechtigungen ab.
   a. Um einem Benutzer oder einer Gruppe von Benutzern das Abrufen von Bildern aus der Registrierung zu ermöglichen, muss dem/den Benutzer(n) die Rolle „Registrierungsbetrachter“ zugewiesen sein.

```
[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. Um einem Benutzer oder einer Benutzergruppe das Schreiben oder Pushen von Bildern zu ermöglichen, muss dem/den Benutzer(n) die Rolle des Registrierungseditors zugewiesen sein.

```
[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. Damit OpenShift-Knoten auf die Registrierung zugreifen und die Bilder pushen oder pullen können, müssen Sie ein Pull-Geheimnis konfigurieren.

```
[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. Dieses Pull-Geheimnis kann dann in Servicekonten gepatcht oder in der entsprechenden Pod-Definition referenziert werden.
    a. Um es auf Dienstkonten zu patchen, führen Sie den folgenden Befehl aus.

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

    b. Um das Pull-Geheimnis in der Pod-Definition zu referenzieren, fügen Sie den folgenden Parameter zum `spec` Abschnitt.

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

11. Führen Sie die folgenden Schritte aus, um ein Image von anderen Workstations als dem OpenShift-Knoten zu pushen oder zu pullen.
    a. Fügen Sie die TLS-Zertifikate zum Docker-Client hinzu.

```
[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. Melden Sie sich mit dem Befehl „oc login“ bei OpenShift an.

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

    c. Melden Sie sich mit den OpenShift-Benutzeranmeldeinformationen und dem Befehl „podman/docker“ beim Register an.

Podman

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

+ HINWEIS: Wenn Sie `kubeadmin` Benutzer, um sich beim privaten Register anzumelden, verwenden Sie dann ein Token anstelle eines Kennworts.

Docker

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

+ HINWEIS: Wenn Sie `kubeadmin` Benutzer, um sich beim privaten Register anzumelden, verwenden Sie dann ein Token anstelle eines Kennworts.

d. Schieben oder ziehen Sie die Bilder.

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
```

Docker

```
[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
```

Lösungsvalidierung und Anwendungsfälle

Lösungsvalidierung und Anwendungsfälle: Red Hat OpenShift mit NetApp

Bei den auf dieser Seite bereitgestellten Beispielen handelt es sich um Lösungsvalidierungen und Anwendungsfälle für Red Hat OpenShift mit NetApp.

- "Stellen Sie eine Jenkins CI/CD-Pipeline mit persistentem Speicher bereit"
- "Konfigurieren Sie Multitenancy auf Red Hat OpenShift mit NetApp"

- "Red Hat OpenShift-Virtualisierung mit NetApp ONTAP"
- "Erweitertes Cluster-Management für Kubernetes auf Red Hat OpenShift mit NetApp"

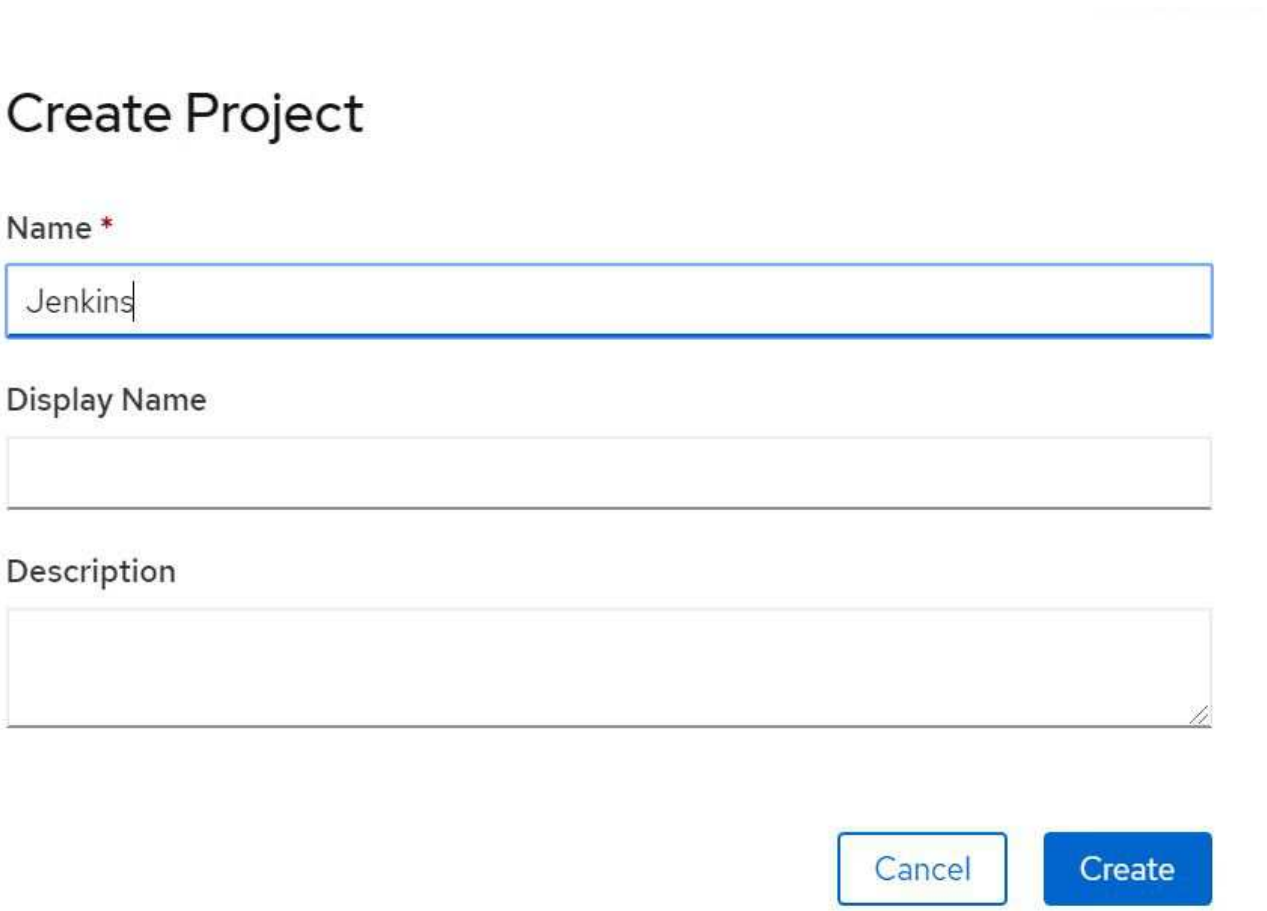
Bereitstellen einer Jenkins CI/CD-Pipeline mit persistentem Speicher: Red Hat OpenShift mit NetApp

In diesem Abschnitt werden die Schritte zum Bereitstellen einer CI/CD-Pipeline (Continuous Integration/Continuous Delivery oder Deployment) mit Jenkins beschrieben, um den Betrieb der Lösung zu validieren.

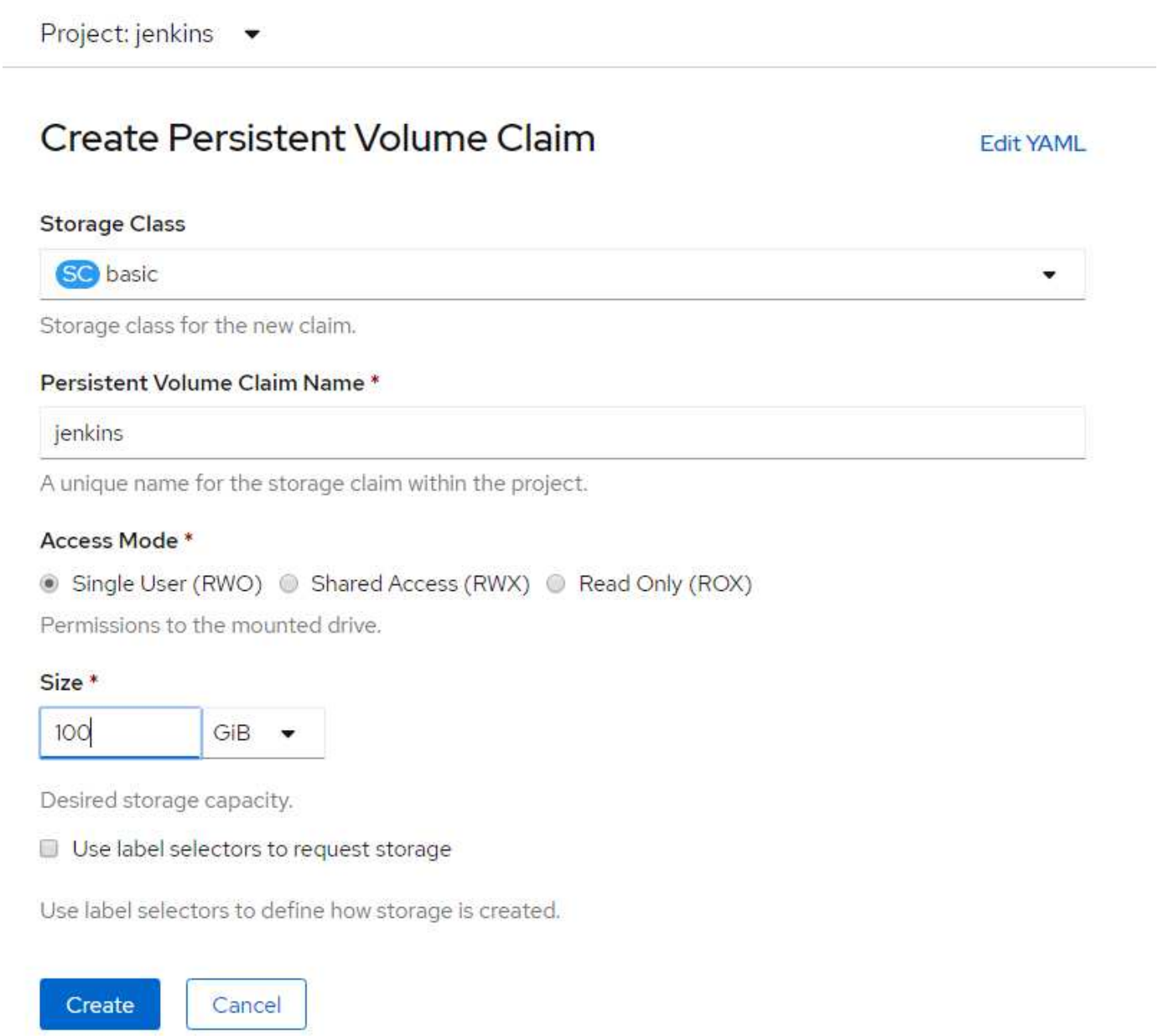
Erstellen Sie die für die Jenkins-Bereitstellung erforderlichen Ressourcen

Führen Sie die folgenden Schritte aus, um die für die Bereitstellung der Jenkins-Anwendung erforderlichen Ressourcen zu erstellen:

1. Erstellen Sie ein neues Projekt mit dem Namen Jenkins.



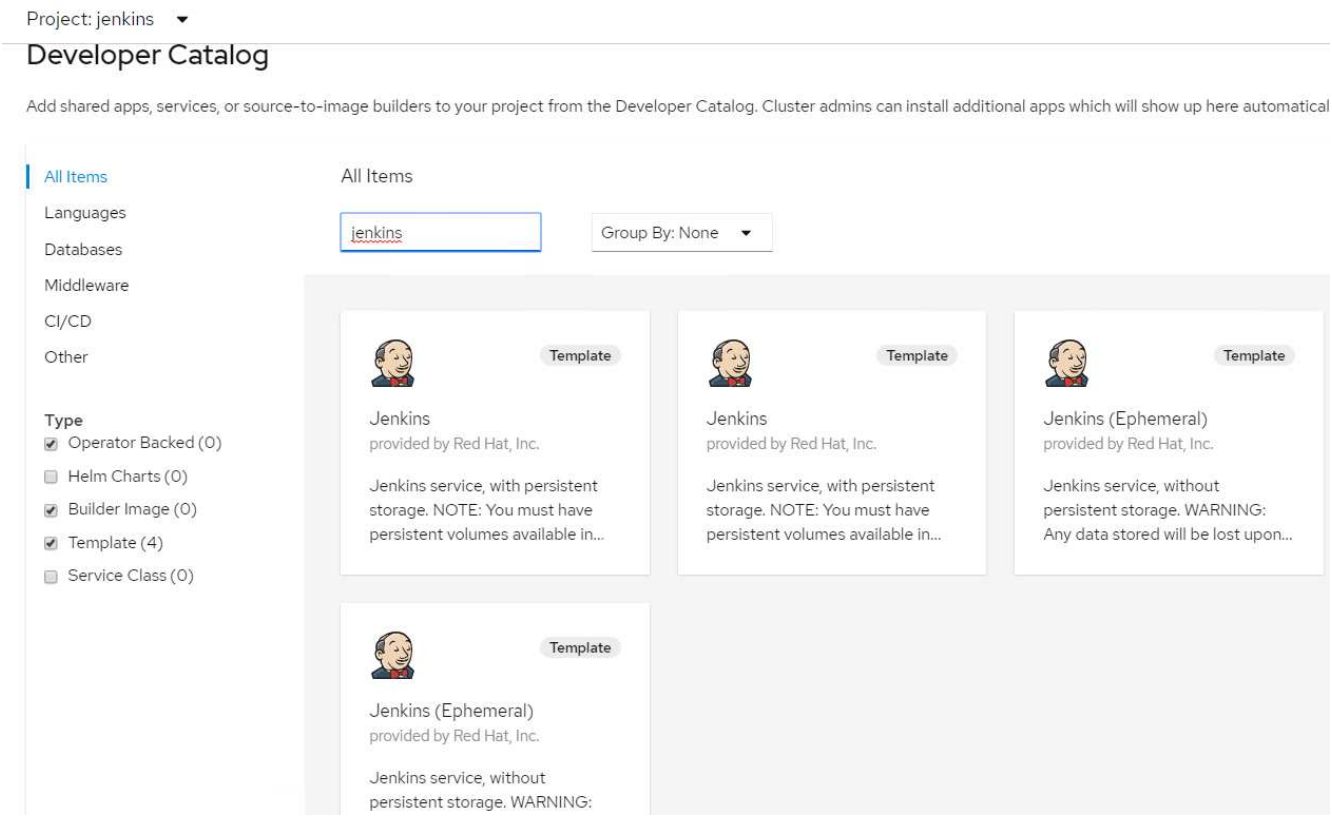
2. In diesem Beispiel haben wir Jenkins mit persistentem Speicher bereitgestellt. Erstellen Sie das PVC, um den Jenkins-Build zu unterstützen. Navigieren Sie zu Speicher > Persistent Volume Claims und klicken Sie auf Persistent Volume Claim erstellen. Wählen Sie die erstellte Speicherklasse aus, stellen Sie sicher, dass der Anspruchsname des persistenten Volumes „Jenkins“ lautet, wählen Sie die entsprechende Größe und den entsprechenden Zugriffsmodus aus und klicken Sie dann auf „Erstellen“.



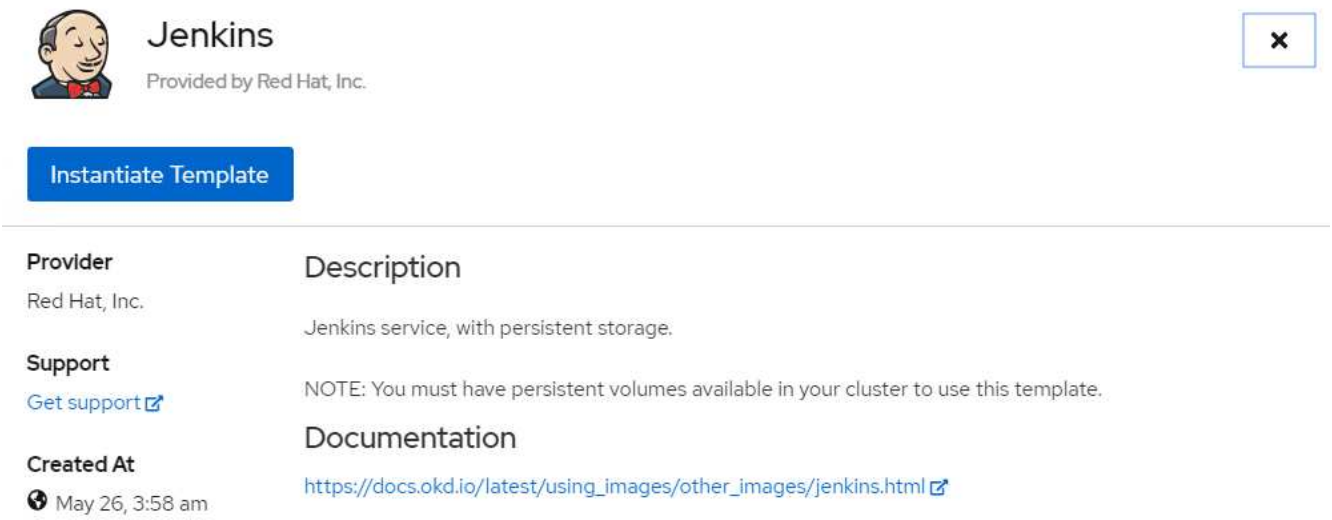
Jenkins mit persistentem Speicher bereitstellen

Führen Sie die folgenden Schritte aus, um Jenkins mit persistentem Speicher bereitzustellen:

1. Ändern Sie in der oberen linken Ecke die Rolle von Administrator zu Entwickler. Klicken Sie auf +Hinzufügen und wählen Sie „Aus Katalog“ aus. Suchen Sie in der Leiste „Nach Stichwort filtern“ nach Jenkins. Wählen Sie den Jenkins-Dienst mit persistentem Speicher.

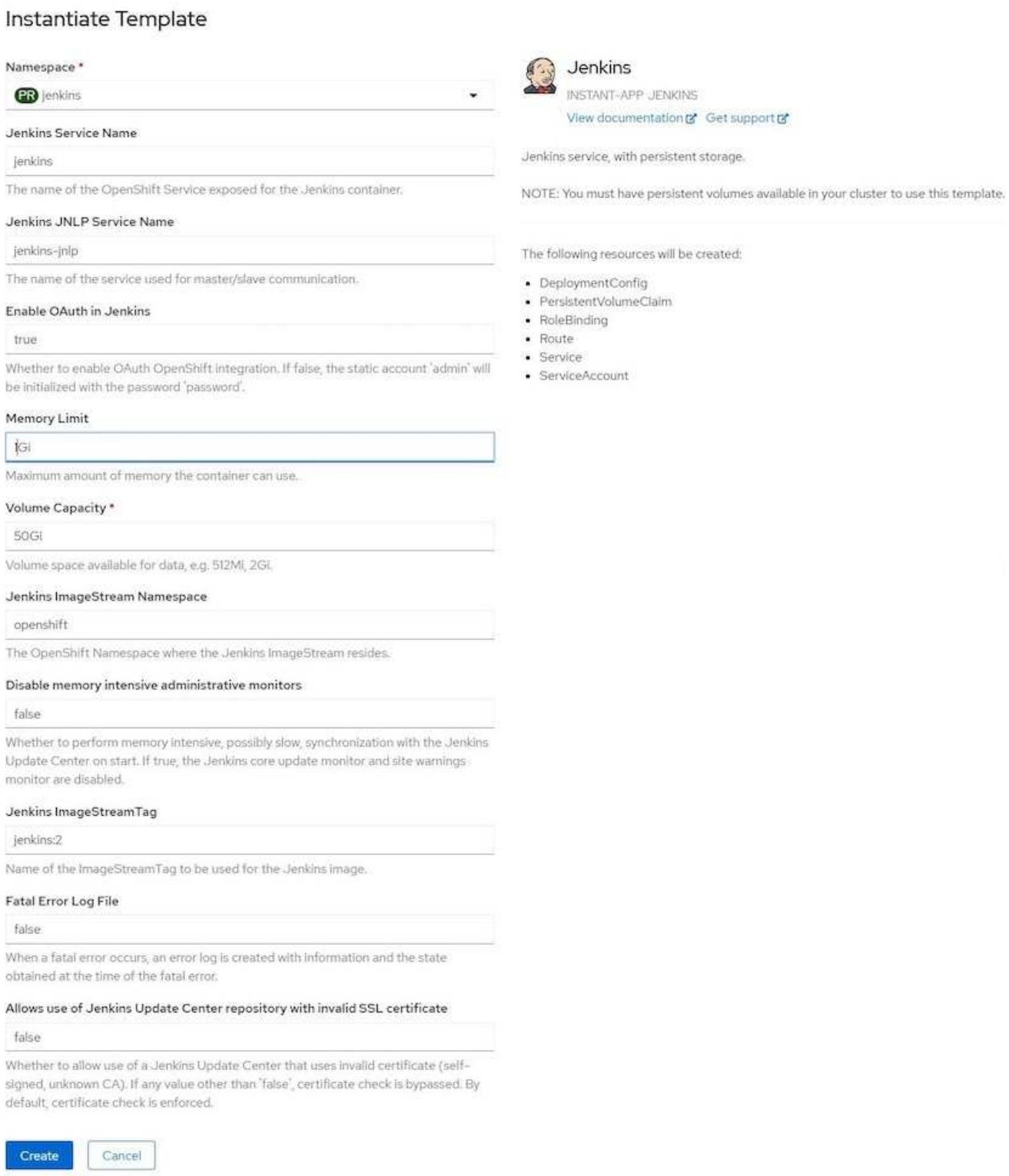


2. Klicken `Instantiate Template` .

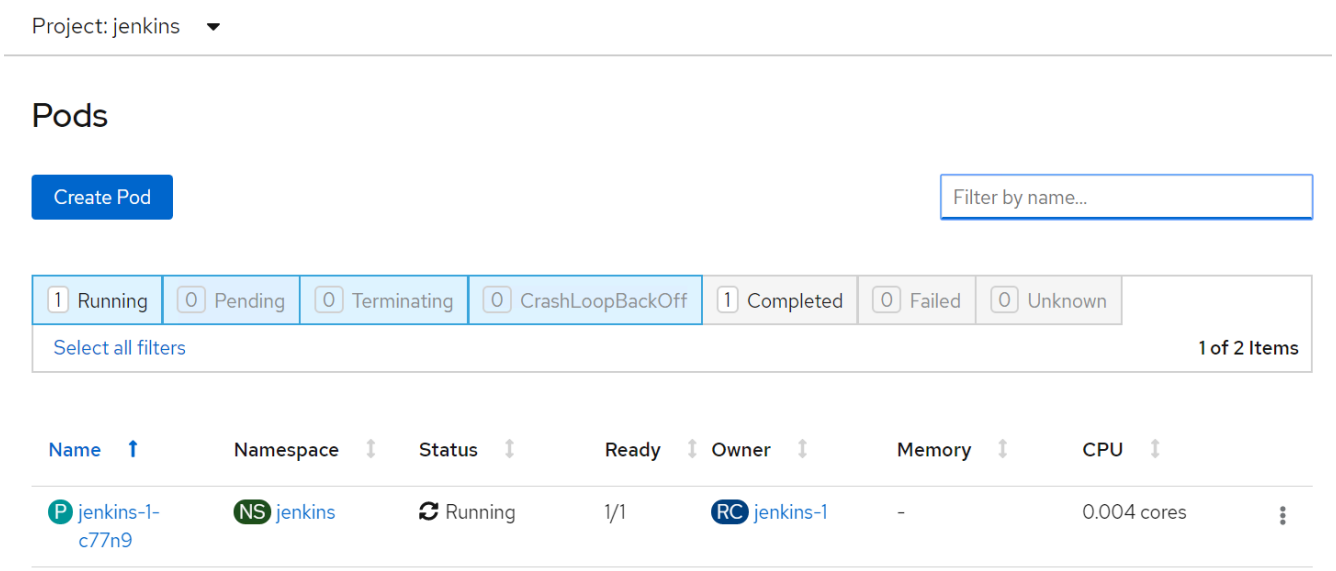


3. Standardmäßig werden die Details für die Jenkins-Anwendung ausgefüllt. Ändern Sie die Parameter entsprechend Ihren Anforderungen und klicken Sie auf „Erstellen“. Dieser Prozess erstellt alle

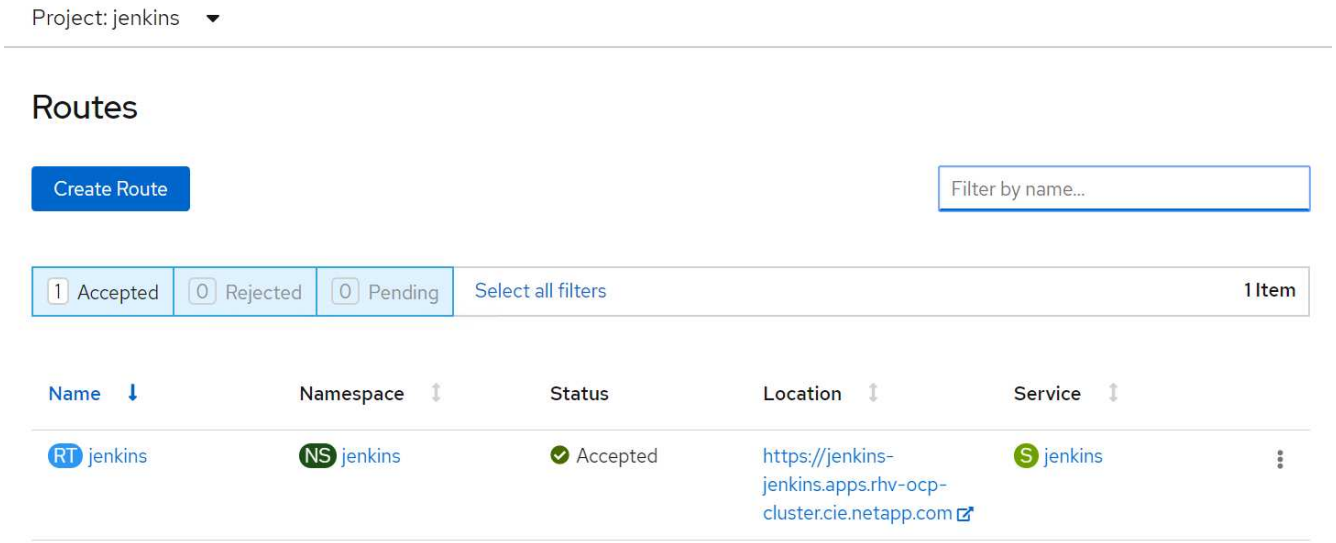
erforderlichen Ressourcen zur Unterstützung von Jenkins auf OpenShift.



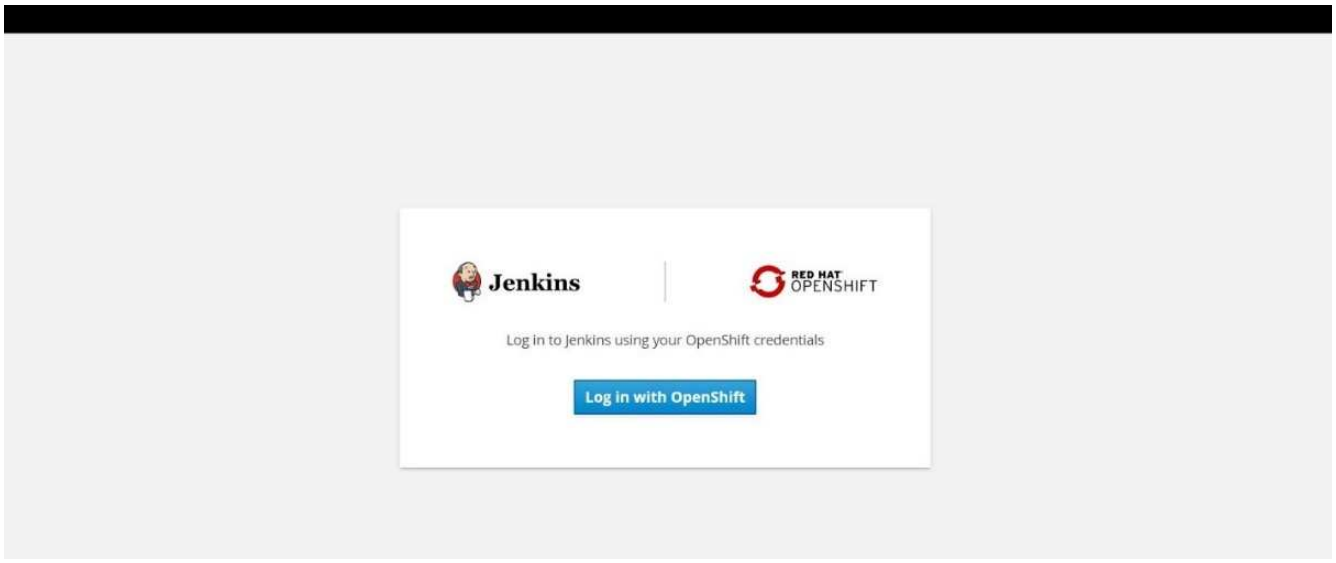
4. Es dauert ungefähr 10 bis 12 Minuten, bis die Jenkins-Pods in den Bereitschaftszustand wechseln.



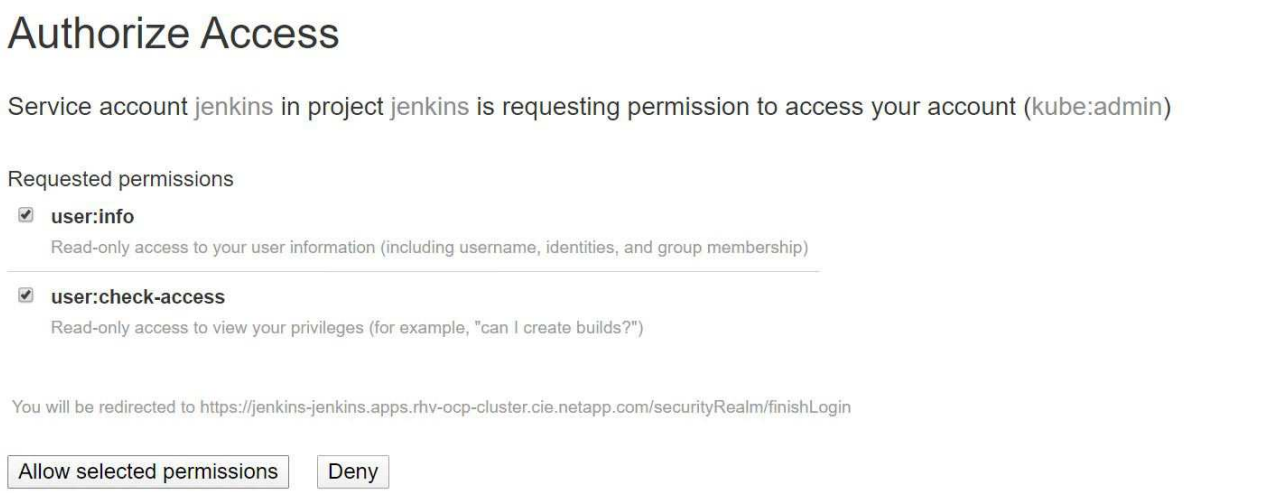
5. Navigieren Sie nach der Instanziierung der Pods zu „Netzwerk > Routen“. Um die Jenkins-Webseite zu öffnen, klicken Sie auf die für die Jenkins-Route angegebene URL.



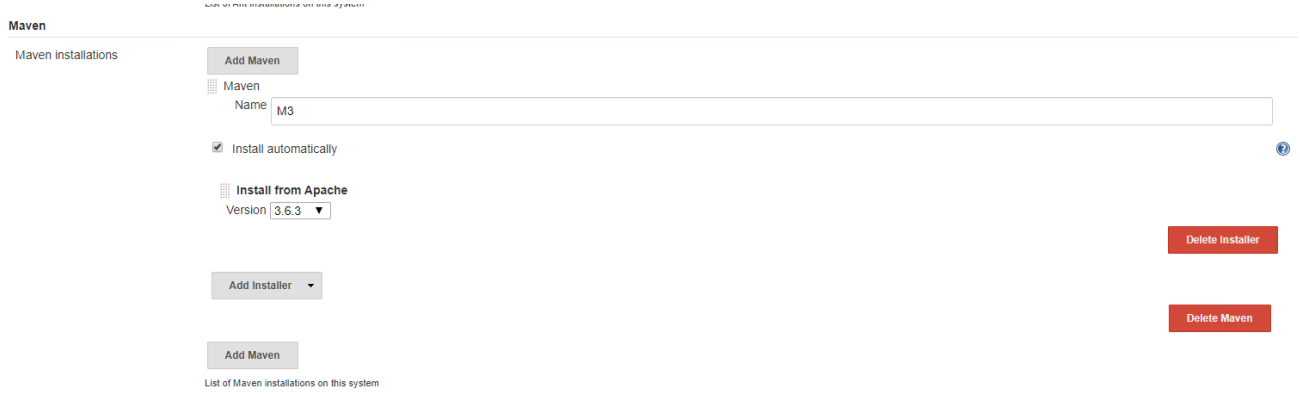
6. Da beim Erstellen der Jenkins-App OpenShift OAuth verwendet wurde, klicken Sie auf „Mit OpenShift anmelden“.



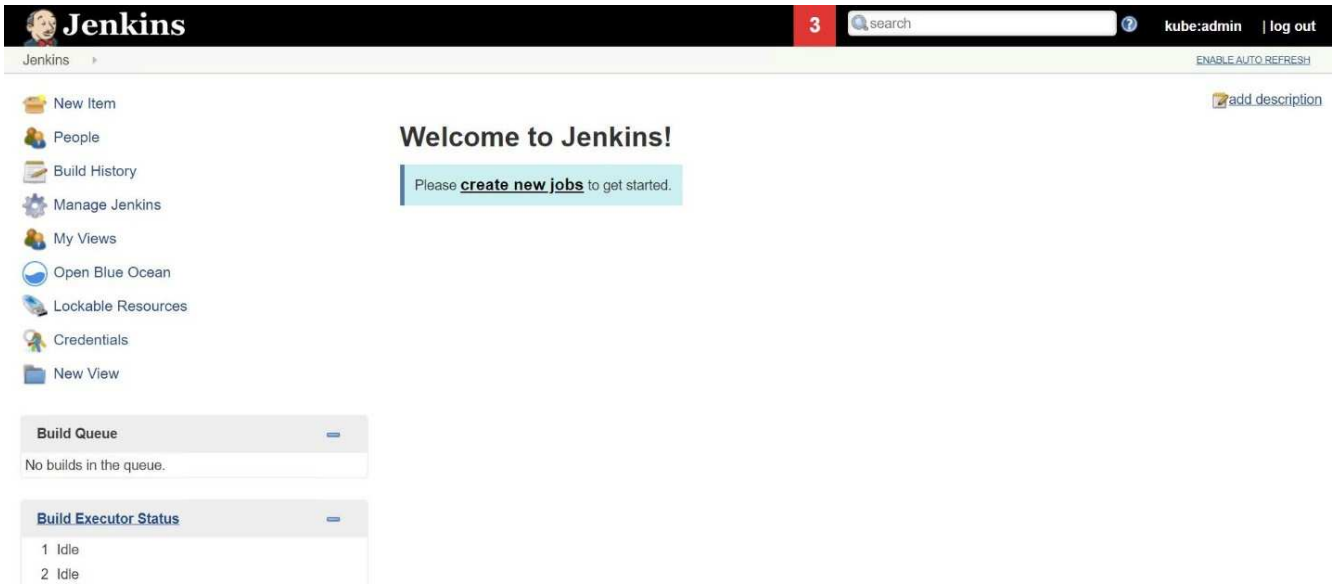
7. Autorisieren Sie das Jenkins-Dienstkonto für den Zugriff auf die OpenShift-Benutzer.



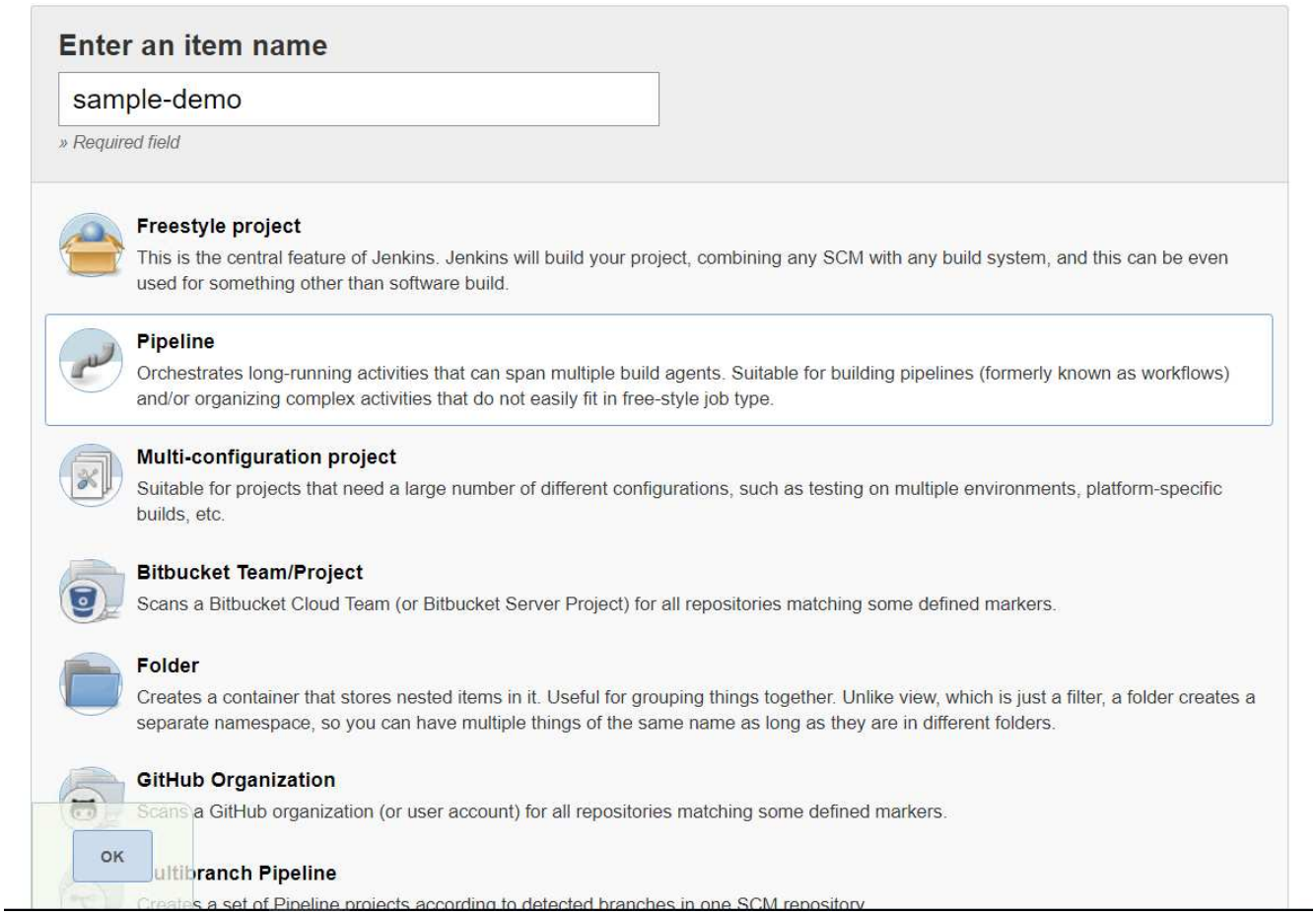
8. Die Jenkins-Willkommensseite wird angezeigt. Da wir einen Maven-Build verwenden, schließen Sie zuerst die Maven-Installation ab. Navigieren Sie zu „Jenkins verwalten“ > „Globale Toolkonfiguration“ und klicken Sie dann in der Maven-Unterüberschrift auf „Maven hinzufügen“. Geben Sie den gewünschten Namen ein und stellen Sie sicher, dass die Option „Automatisch installieren“ ausgewählt ist. Klicken Sie auf Speichern.



9. Sie können jetzt eine Pipeline erstellen, um den CI/CD-Workflow zu demonstrieren. Klicken Sie auf der Startseite im linken Menü auf „Neue Jobs erstellen“ oder „Neues Element“.

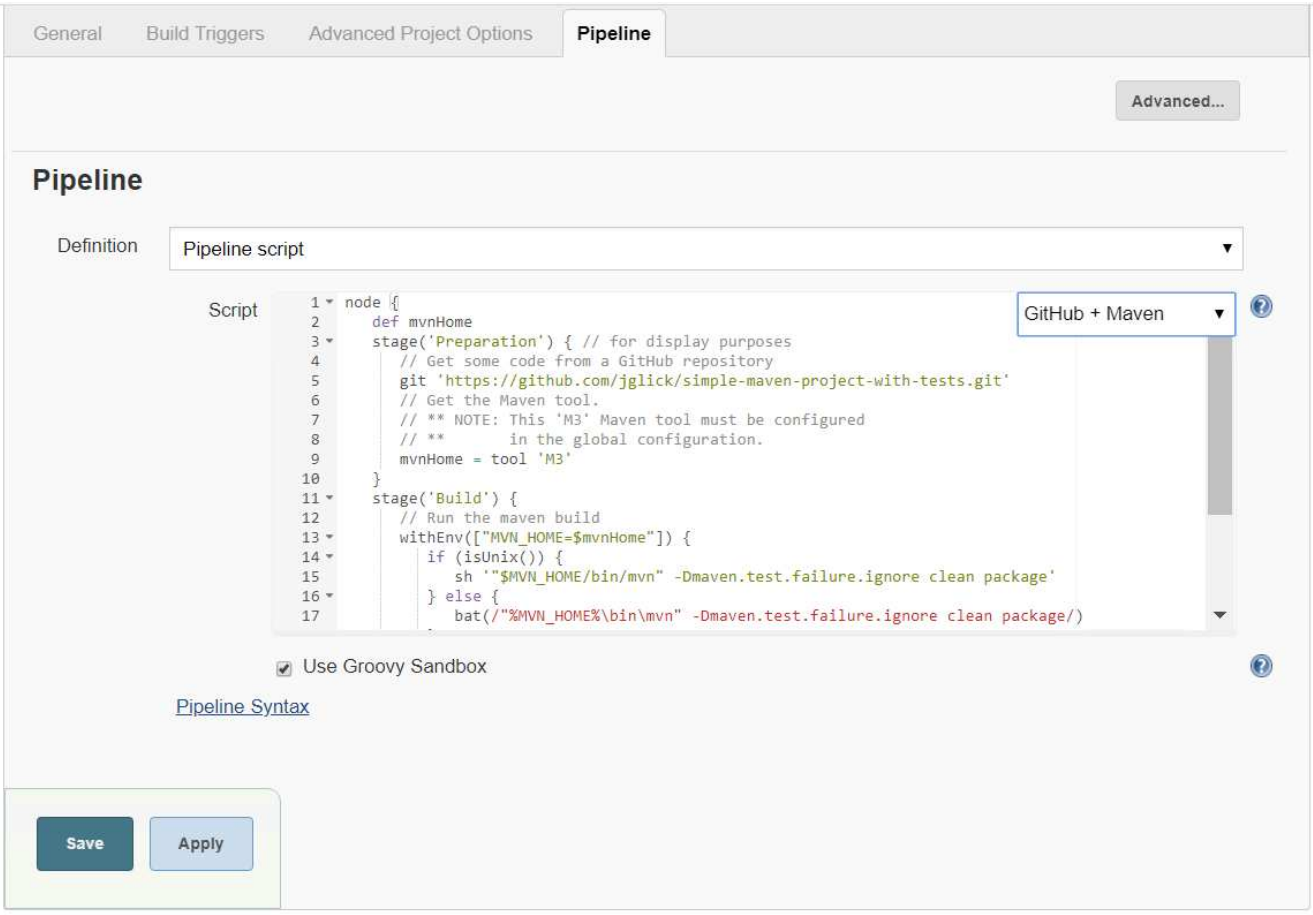


10. Geben Sie auf der Seite „Element erstellen“ den gewünschten Namen ein, wählen Sie „Pipeline“ aus und klicken Sie auf „OK“.



11. Wählen Sie die Registerkarte „Pipeline“ aus. Wählen Sie im Dropdown-Menü „Beispielpipeline ausprobieren“ die Option „Github + Maven“ aus. Der Code wird automatisch ausgefüllt. Klicken Sie auf

Speichern.



12. Klicken Sie auf „Jetzt erstellen“, um die Entwicklung durch die Vorbereitungs-, Erstellungs- und Testphase zu starten. Es kann mehrere Minuten dauern, bis der gesamte Build-Prozess abgeschlossen ist und die Build-Ergebnisse angezeigt werden.

13. Bei Codeänderungen kann die Pipeline neu erstellt werden, um die neue Softwareversion zu patchen und so eine kontinuierliche Integration und Bereitstellung zu ermöglichen. Klicken Sie auf „Letzte Änderungen“, um die Änderungen seit der vorherigen Version zu verfolgen.

Konfigurieren der Mandantenfähigkeit

Konfigurieren von Multitenancy auf Red Hat OpenShift mit NetApp

Viele Organisationen, die mehrere Anwendungen oder Workloads auf Containern ausführen, neigen dazu, einen Red Hat OpenShift-Cluster pro Anwendung oder Workload bereitzustellen. Dadurch können sie eine strikte Isolierung der Anwendung oder Arbeitslast implementieren, die Leistung optimieren und Sicherheitslücken reduzieren. Allerdings bringt die Bereitstellung eines separaten Red Hat OpenShift-Clusters für jede Anwendung eigene Probleme mit sich. Es erhöht den Betriebsaufwand, da jeder Cluster einzeln überwacht und verwaltet werden muss, erhöht die Kosten aufgrund dedizierter Ressourcen für verschiedene Anwendungen und behindert eine effiziente Skalierbarkeit.

Um diese Probleme zu überwinden, kann man in Erwägung ziehen, alle Anwendungen oder Workloads in einem einzigen Red Hat OpenShift-Cluster auszuführen. Doch in einer solchen Architektur stellen die Isolierung von Ressourcen und Sicherheitslücken in der Anwendung eine der größten Herausforderungen dar. Jede Sicherheitslücke in einer Arbeitslast kann natürlich auf eine andere Arbeitslast übergreifen und so den Auswirkungsbereich vergrößern. Darüber hinaus kann jede abrupte, unkontrollierte Ressourcennutzung durch eine Anwendung die Leistung einer anderen Anwendung beeinträchtigen, da standardmäßig keine Richtlinie zur Ressourcenzuweisung vorhanden ist.

Daher suchen Unternehmen nach Lösungen, die das Beste aus beiden Welten vereinen, indem sie ihnen beispielsweise ermöglichen, alle ihre Workloads in einem einzigen Cluster auszuführen und ihnen dennoch die Vorteile eines dedizierten Clusters für jede Workload bieten.

Eine solche effektive Lösung besteht darin, Multitenancy auf Red Hat OpenShift zu konfigurieren. Multitenancy ist eine Architektur, die die Koexistenz mehrerer Mandanten auf demselben Cluster mit angemessener Isolierung von Ressourcen, Sicherheit usw. ermöglicht. In diesem Kontext kann ein Mandant als eine Teilmenge der Clusterressourcen betrachtet werden, die für die Verwendung durch eine bestimmte Benutzergruppe zu einem exklusiven Zweck konfiguriert sind. Das Konfigurieren von Multitenancy auf einem Red Hat OpenShift-Cluster bietet die folgenden Vorteile:

- Eine Reduzierung der Investitions- und Betriebskosten durch die gemeinsame Nutzung von Clusterressourcen
- Geringerer Betriebs- und Verwaltungsaufwand
- Schutz der Workloads vor Kreuzkontamination durch Sicherheitsverletzungen
- Schutz von Workloads vor unerwarteten Leistungseinbußen aufgrund von Ressourcenkonflikten

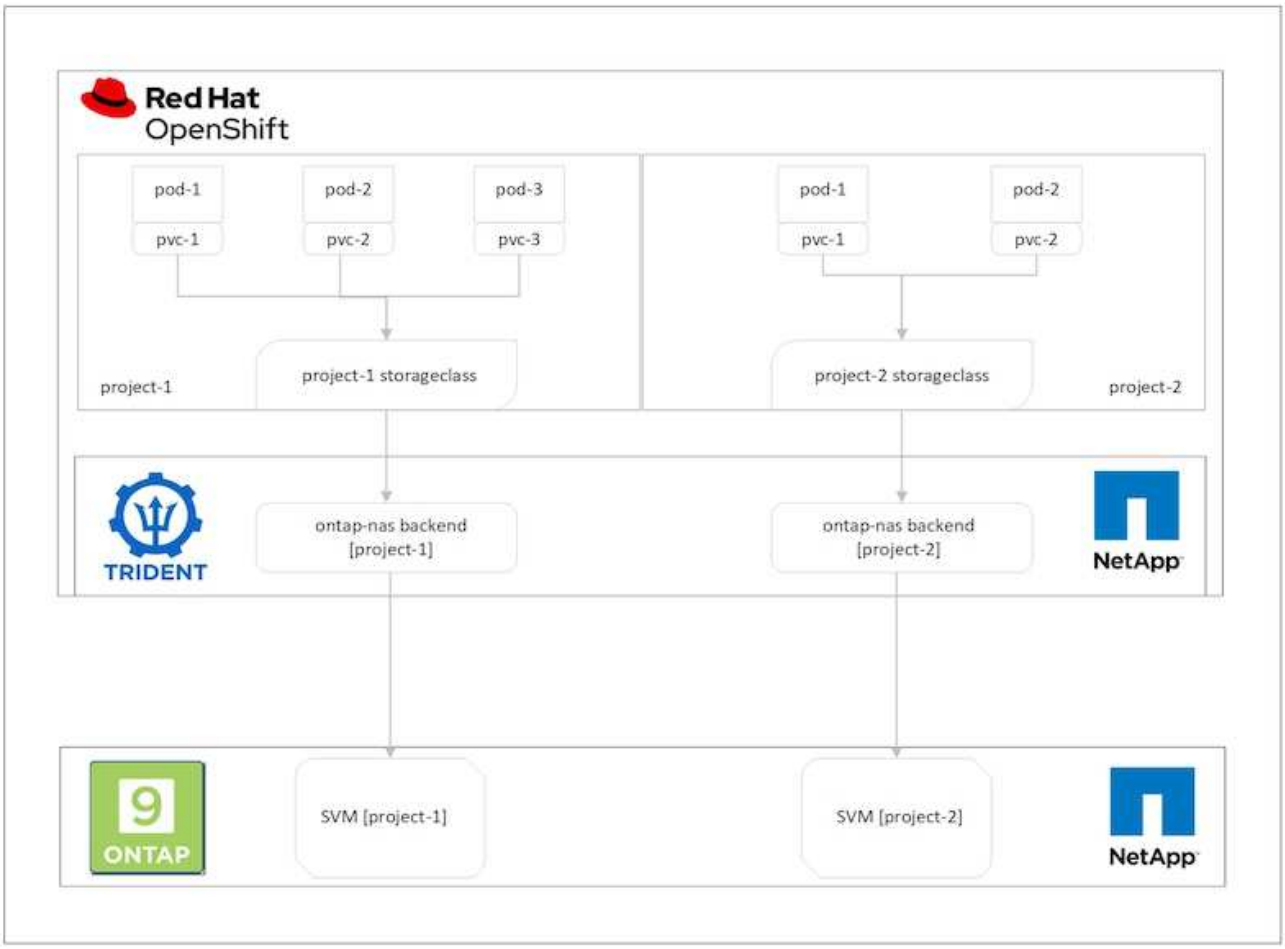
Für einen vollständig realisierten Multitenant-OpenShift-Cluster müssen Kontingente und Einschränkungen für Clusterressourcen konfiguriert werden, die zu verschiedenen Ressourcenbereichen gehören: Compute, Speicher, Netzwerk, Sicherheit usw. Obwohl wir bestimmte Aspekte aller Ressourcenbereiche in dieser Lösung abdecken, konzentrieren wir uns auf bewährte Methoden zum Isolieren und Sichern der Daten, die von mehreren Workloads auf demselben Red Hat OpenShift-Cluster bereitgestellt oder genutzt werden, indem wir Multitenancy auf Speicherressourcen konfigurieren, die dynamisch von Trident zugewiesen werden, unterstützt von NetApp ONTAP.

Architektur

Obwohl Red Hat OpenShift und Trident , unterstützt von NetApp ONTAP , standardmäßig keine Isolierung zwischen Workloads bieten, verfügen sie über eine breite Palette an Funktionen, die zur Konfiguration von Multitenancy verwendet werden können. Um die Entwicklung einer mandantenfähigen Lösung auf einem Red Hat OpenShift-Cluster mit Trident und NetApp ONTAP Unterstützung besser zu verstehen, betrachten wir ein Beispiel mit einer Reihe von Anforderungen und skizzieren die Konfiguration darum herum.

Nehmen wir an, dass eine Organisation zwei ihrer Workloads auf einem Red Hat OpenShift-Cluster als Teil von zwei Projekten ausführt, an denen zwei verschiedene Teams arbeiten. Die Daten für diese Workloads befinden sich auf PVCs, die von Trident dynamisch auf einem NetApp ONTAP NAS-Backend bereitgestellt werden. Die Organisation muss für diese beiden Workloads eine mandantenfähige Lösung entwickeln und die für diese Projekte verwendeten Ressourcen isolieren, um die Aufrechterhaltung von Sicherheit und Leistung sicherzustellen, wobei der Schwerpunkt in erster Linie auf den Daten liegt, die diesen Anwendungen dienen.

Die folgende Abbildung zeigt die Multitenant-Lösung auf einem Red Hat OpenShift-Cluster mit Trident , unterstützt von NetApp ONTAP.



Technologieanforderungen

1. NetApp ONTAP Speichercluster
2. Red Hat OpenShift-Cluster
3. Trident

Red Hat OpenShift – Cluster-Ressourcen

Aus Sicht des Red Hat OpenShift-Clusters ist das Projekt die Ressource der obersten Ebene, mit der man beginnen sollte. Ein OpenShift-Projekt kann als Clusterressource betrachtet werden, die den gesamten OpenShift-Cluster in mehrere virtuelle Cluster aufteilt. Daher bietet die Isolation auf Projektebene eine Grundlage für die Konfiguration der Mandantenfähigkeit.

Als Nächstes muss RBAC im Cluster konfiguriert werden. Die beste Vorgehensweise besteht darin, alle Entwickler, die an einem einzelnen Projekt oder einer einzelnen Arbeitslast arbeiten, in einer einzelnen Benutzergruppe im Identitätsanbieter (IdP) zu konfigurieren. Red Hat OpenShift ermöglicht die IdP-Integration und Benutzergruppensynchronisierung, sodass die Benutzer und Gruppen vom IdP in den Cluster importiert werden können. Dies hilft den Cluster-Administratoren, den Zugriff auf die einem Projekt zugewiesenen Cluster-Ressourcen auf eine oder mehrere Benutzergruppen zu beschränken, die an diesem Projekt arbeiten, und so den unbefugten Zugriff auf Cluster-Ressourcen zu unterbinden. Weitere Informationen zur IdP-Integration mit Red Hat OpenShift finden Sie in der Dokumentation "hier," .

NetApp ONTAP

Es ist wichtig, den gemeinsam genutzten Speicher zu isolieren, der als persistenter Speicheranbieter für einen Red Hat OpenShift-Cluster dient, um sicherzustellen, dass die auf dem Speicher für jedes Projekt erstellten Volumes den Hosts so erscheinen, als wären sie auf einem separaten Speicher erstellt worden. Erstellen Sie dazu auf NetApp ONTAP so viele SVMs (Storage Virtual Machines), wie Projekte oder Workloads vorhanden sind, und weisen Sie jede SVM einem Workload zu.

Trident

Nachdem Sie unterschiedliche SVMs für unterschiedliche Projekte auf NetApp ONTAP erstellt haben, müssen Sie jedes SVM einem anderen Trident Backend zuordnen. Die Backend-Konfiguration auf Trident steuert die Zuweisung von persistentem Speicher zu OpenShift-Clusterressourcen und erfordert die Details der SVM, der eine Zuordnung erfolgen soll. Dies sollte mindestens der Protokolltreiber für das Backend sein. Optional können Sie definieren, wie die Volumes auf dem Speicher bereitgestellt werden, und Grenzwerte für die Größe der Volumes oder die Verwendung von Aggregaten usw. festlegen. Details zur Definition der Trident -Backends finden Sie "hier," .

Red Hat OpenShift – Speicherressourcen

Nach der Konfiguration der Trident -Backends besteht der nächste Schritt darin, StorageClasses zu konfigurieren. Konfigurieren Sie so viele Speicherklassen wie Backends vorhanden sind und gewähren Sie jeder Speicherklasse Zugriff auf das Hochfahren von Volumes nur auf einem Backend. Wir können die StorageClass einem bestimmten Trident Backend zuordnen, indem wir beim Definieren der Speicherklasse den Parameter storagePools verwenden. Die Details zur Definition einer Speicherklasse finden Sie "hier," . Daher gibt es eine Eins-zu-eins-Zuordnung von StorageClass zum Trident Backend, die auf eine SVM zurückverweist. Dadurch wird sichergestellt, dass alle Speicheransprüche über die diesem Projekt zugewiesene StorageClass nur von der SVM bedient werden, die diesem Projekt gewidmet ist.

Da Speicherklassen keine Namespace-Ressourcen sind, wie stellen wir sicher, dass Speicheransprüche auf die Speicherklasse eines Projekts durch Pods in einem anderen Namespace oder Projekt abgelehnt werden? Die Antwort ist die Verwendung von ResourceQuotas. ResourceQuotas sind Objekte, die die Gesamtnutzung von Ressourcen pro Projekt steuern. Es kann sowohl die Anzahl als auch die Gesamtmenge der Ressourcen begrenzen, die von Objekten im Projekt verbraucht werden können. Fast alle Ressourcen eines Projekts können mithilfe von ResourceQuotas begrenzt werden. Durch die effiziente Nutzung dieser Ressourcen können Unternehmen Kosten senken und Ausfälle aufgrund von Überbereitstellung oder übermäßigem Ressourcenverbrauch vermeiden. Weitere Informationen finden Sie in der Dokumentation "hier," für weitere Informationen.

Für diesen Anwendungsfall müssen wir die Pods in einem bestimmten Projekt daran hindern, Speicher von Speicherklassen zu beanspruchen, die nicht für ihr Projekt bestimmt sind. Dazu müssen wir die persistenten Volume-Ansprüche für andere Speicherklassen begrenzen, indem wir `<storage-class-name>.storageclass.storage.k8s.io/persistentvolumeclaims` auf 0. Darüber hinaus muss ein Clusteradministrator sicherstellen, dass die Entwickler in einem Projekt keinen Zugriff auf die Änderung der ResourceQuotas haben.

Konfiguration

Bei jeder Multitenant-Lösung kann kein Benutzer auf mehr Clusterressourcen zugreifen als erforderlich. Daher wird der gesamte Satz an Ressourcen, der im Rahmen der Multitenancy-Konfiguration konfiguriert werden soll, zwischen Cluster-Administrator, Speicheradministrator und Entwicklern aufgeteilt, die an den einzelnen Projekten arbeiten.

In der folgenden Tabelle sind die verschiedenen Aufgaben aufgeführt, die von verschiedenen Benutzern ausgeführt werden müssen:

Rolle	Aufgaben
Cluster-Administrator	Erstellen Sie Projekte für verschiedene Anwendungen oder Workloads
	Erstellen Sie ClusterRoles und RoleBindings für Storage-Admin
	Erstellen Sie Rollen und RoleBindings für Entwickler, die Zugriff auf bestimmte Projekte zuweisen
	[Optional] Konfigurieren Sie Projekte, um Pods auf bestimmten Knoten zu planen
Speicheradministrator	Erstellen Sie SVMs auf NetApp ONTAP
	Erstellen Sie Trident -Backends
	Erstellen von StorageClasses
	Erstellen von Speicherressourcenkontingenten
Entwickler	Validieren Sie den Zugriff zum Erstellen oder Patchen von PVCs oder Pods im zugewiesenen Projekt
	Validieren Sie den Zugriff zum Erstellen oder Patchen von PVCs oder Pods in einem anderen Projekt
	Überprüfen Sie den Zugriff zum Anzeigen oder Bearbeiten von Projekten, Ressourcenkontingenten und Speicherklassen.

Konfiguration

Im Folgenden sind die Voraussetzungen für die Konfiguration von Multitenancy auf Red Hat OpenShift mit NetApp aufgeführt.

Voraussetzungen

- NetApp ONTAP Cluster
- Red Hat OpenShift-Cluster
- Trident auf dem Cluster installiert
- Admin-Workstation mit installierten und zu $PATH hinzugefügten Tridentctl- und OC-Tools
- Administratorzugriff auf ONTAP
- Cluster-Admin-Zugriff auf OpenShift-Cluster
- Der Cluster ist in den Identitätsanbieter integriert
- Der Identitätsanbieter ist so konfiguriert, dass er effizient zwischen Benutzern in verschiedenen Teams unterscheiden kann

Konfiguration: Cluster-Admin-Aufgaben

Die folgenden Aufgaben werden vom Red Hat OpenShift-Clusteradministrator ausgeführt:

1. Melden Sie sich als Cluster-Administrator beim Red Hat OpenShift-Cluster an.
2. Erstellen Sie zwei Projekte, die unterschiedlichen Projekten entsprechen.

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

3. Erstellen Sie die Entwicklerrolle für Projekt 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
```

Die in diesem Abschnitt bereitgestellte Rollendefinition ist nur ein Beispiel. Entwicklerrollen müssen basierend auf den Anforderungen des Endbenutzers definiert werden.

1. Erstellen Sie auf ähnliche Weise Entwicklerrollen für Projekt 2.
2. Alle OpenShift- und NetApp -Speicherressourcen werden normalerweise von einem Speicheradministrator verwaltet. Der Zugriff für Speicheradministratoren wird durch die Trident-Operatorrolle gesteuert, die bei der Installation von Trident erstellt wird. Darüber hinaus benötigt der Speicheradministrator auch Zugriff auf ResourceQuotas, um die Speichernutzung zu steuern.
3. Erstellen Sie eine Rolle zum Verwalten von ResourceQuotas in allen Projekten im Cluster, um sie dem Speicheradministrator zuzuordnen.

```
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. Stellen Sie sicher, dass der Cluster in den Identitätsanbieter der Organisation integriert ist und dass Benutzergruppen mit Clustergruppen synchronisiert werden. Das folgende Beispiel zeigt, dass der Identitätsanbieter in den Cluster integriert und mit den Benutzergruppen synchronisiert wurde.

```
$ 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. Konfigurieren Sie ClusterRoleBindings für Speicheradministratoren.

```
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
```

Für Speicheradministratoren müssen zwei Rollen gebunden werden: Trident-Operator und Resource-Quotas.

1. Erstellen Sie RoleBindings für Entwickler, die die Rolle „developer-project-1“ an die entsprechende Gruppe (ocp-project-1) in project-1 binden.

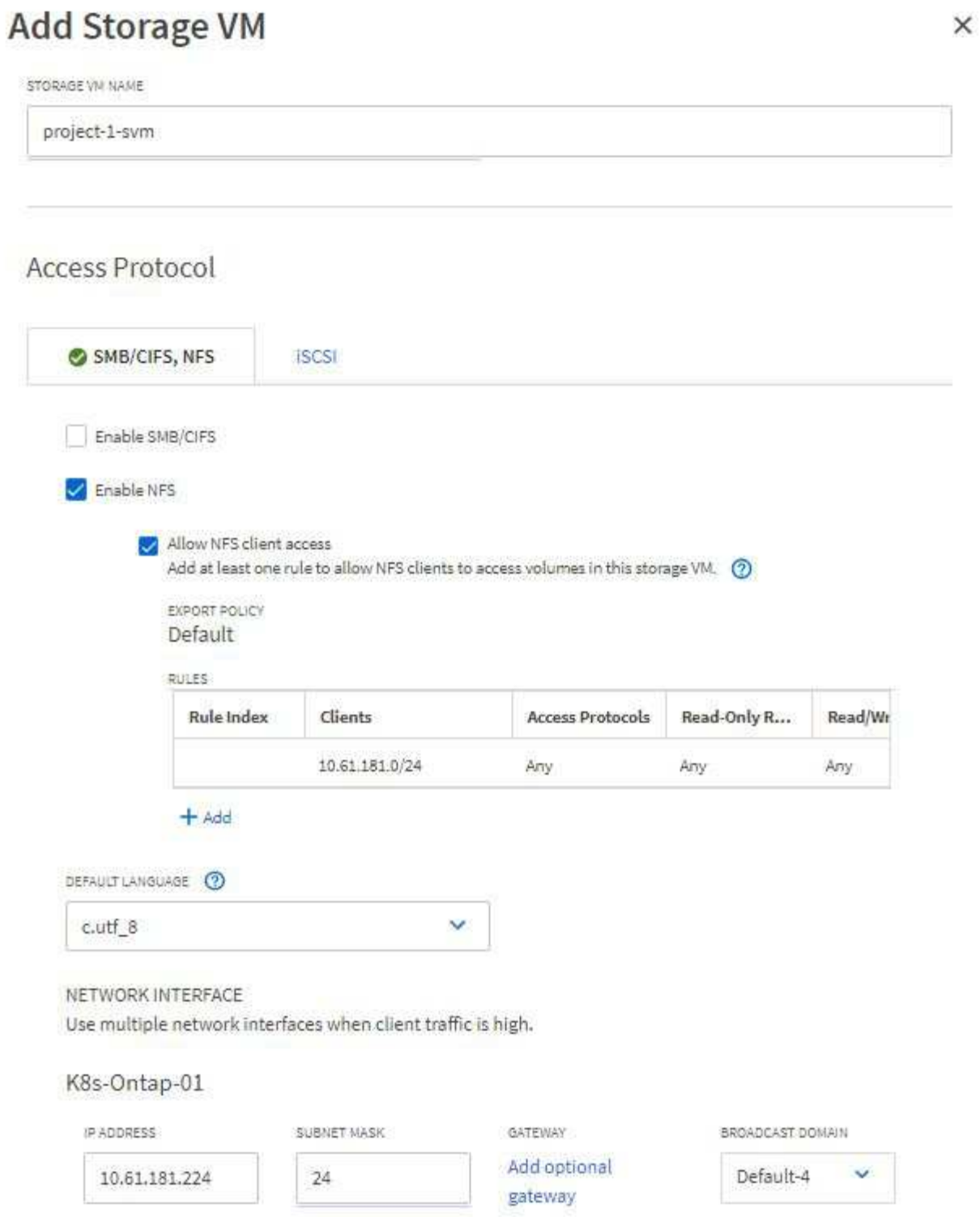
```
cat << EOF | oc create -f -
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: project-1-developer
  namespace: project-1
subjects:
  - kind: Group
    apiGroup: rbac.authorization.k8s.io
    name: ocp-project-1
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role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

2. Erstellen Sie auf ähnliche Weise RoleBindings für Entwickler, die die Entwicklerrollen an die entsprechende Benutzergruppe in Projekt 2 binden.

Konfiguration: Speicheradministratoraufgaben

Die folgenden Ressourcen müssen von einem Speicheradministrator konfiguriert werden:

1. Melden Sie sich als Administrator beim NetApp ONTAP -Cluster an.
2. Navigieren Sie zu Speicher > Speicher-VMs und klicken Sie auf Hinzufügen. Erstellen Sie zwei SVMs, eine für Projekt 1 und die andere für Projekt 2, indem Sie die erforderlichen Details angeben. Erstellen Sie außerdem ein vsadmin-Konto, um die SVM und ihre Ressourcen zu verwalten.



1. Melden Sie sich als Speicheradministrator beim Red Hat OpenShift-Cluster an.
2. Erstellen Sie das Backend für Projekt 1 und ordnen Sie es der für das Projekt vorgesehenen SVM zu. NetApp empfiehlt, das vsadmin-Konto des SVM zu verwenden, um das Backend mit dem SVM zu verbinden, anstatt den ONTAP Clusteradministrator zu verwenden.

```
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
```

Für dieses Beispiel verwenden wir den ontap-nas-Treiber. Verwenden Sie beim Erstellen des Backends je nach Anwendungsfall den entsprechenden Treiber.

Wir gehen davon aus, dass Trident im Trident-Projekt installiert ist.

1. Erstellen Sie auf ähnliche Weise das Trident Backend für Projekt 2 und ordnen Sie es dem für Projekt 2 vorgesehenen SVM zu.
2. Erstellen Sie als Nächstes die Speicherklassen. Erstellen Sie die Speicherklasse für Projekt 1 und konfigurieren Sie sie so, dass sie die Speicherpools aus dem für Projekt 1 reservierten Backend verwendet, indem Sie den Parameter „storagePools“ festlegen.

```
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. Erstellen Sie auf ähnliche Weise eine Speicherklasse für Projekt 2 und konfigurieren Sie sie so, dass die Speicherpools vom Backend verwendet werden, die für Projekt 2 reserviert sind.
4. Erstellen Sie ein ResourceQuota, um Ressourcen in Projekt 1 einzuschränken, die Speicher von Speicherklassen anfordern, die anderen Projekten gewidmet sind.

```
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. Erstellen Sie auf ähnliche Weise ein ResourceQuota, um Ressourcen in Projekt 2 einzuschränken, die Speicher von Speicherklassen anfordern, die anderen Projekten gewidmet sind.

Validierung

Führen Sie die folgenden Schritte aus, um die in den vorherigen Schritten konfigurierte Multitenant-Architektur zu validieren:

Validieren Sie den Zugriff zum Erstellen von PVCs oder Pods im zugewiesenen Projekt

1. Melden Sie sich als ocp-project-1-user, Entwickler in project-1, an.
2. Überprüfen Sie den Zugriff, um ein neues Projekt zu erstellen.

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

3. Erstellen Sie einen PVC in Projekt 1 unter Verwendung der Speicherklasse, die Projekt 1 zugewiesen ist.

```
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. Überprüfen Sie das mit dem PVC verknüpfte PV.

```
oc get pv
```

5. Überprüfen Sie, ob das PV und sein Volume in einer SVM erstellt werden, die für Projekt 1 auf NetApp ONTAP vorgesehen ist.

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

6. Erstellen Sie einen Pod in Projekt 1 und mounten Sie das im vorherigen Schritt erstellte PVC.

```
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. Überprüfen Sie, ob der Pod ausgeführt wird und ob er das Volume gemountet hat.

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

Validieren Sie den Zugriff, um PVCs oder Pods in einem anderen Projekt zu erstellen oder Ressourcen zu verwenden, die einem anderen Projekt zugewiesen sind.

1. Melden Sie sich als ocp-project-1-user, Entwickler in project-1, an.
2. Erstellen Sie einen PVC in Projekt 1 unter Verwendung der Speicherklasse, die Projekt 2 zugewiesen ist.

```
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. Erstellen Sie ein PVC in Projekt 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. Stellen Sie sicher, dass PVCs `test-pvc-project-1-sc-2` Und `test-pvc-project-2-sc-1` wurden nicht erstellt.

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

5. Erstellen Sie einen Pod in Projekt 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
```

Überprüfen Sie den Zugriff zum Anzeigen und Bearbeiten von Projekten, Ressourcenkontingenten und Speicherklassen.

1. Melden Sie sich als ocp-project-1-user, Entwickler in project-1, an.
2. Überprüfen Sie den Zugriff, um neue Projekte zu erstellen.

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

3. Bestätigen Sie den Zugriff, um Projekte anzuzeigen.

```
oc get ns
```

4. Überprüfen Sie, ob der Benutzer ResourceQuotas in Projekt 1 anzeigen oder bearbeiten kann.

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

5. Überprüfen Sie, ob der Benutzer Zugriff auf die Anzeige der Speicherklassen hat.

```
oc get sc
```

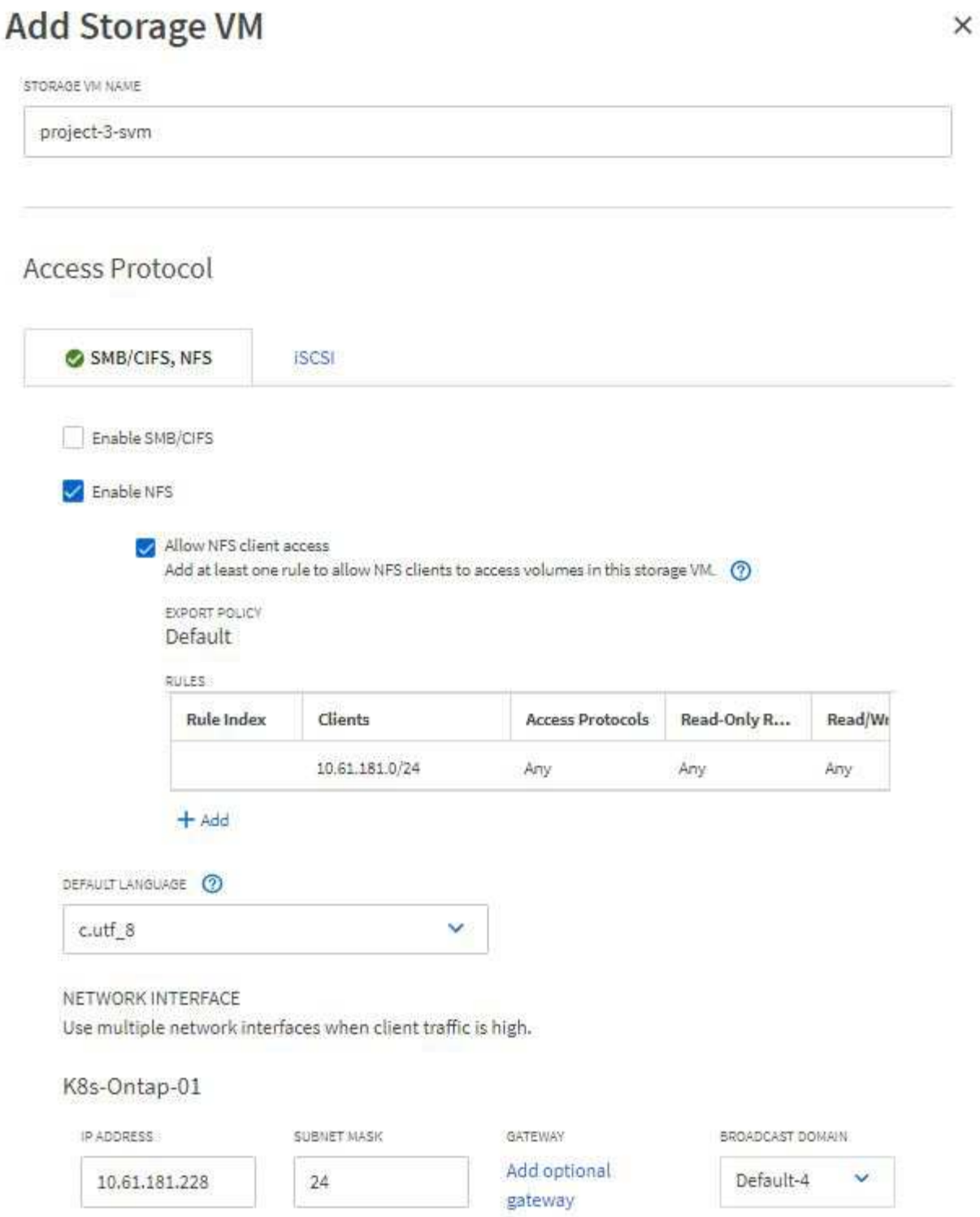
6. Überprüfen Sie den Zugriff, um die Speicherklassen zu beschreiben.
7. Überprüfen Sie den Zugriff des Benutzers zum Bearbeiten der Speicherklassen.

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

Skalierung: Hinzufügen weiterer Projekte

In einer Multitenant-Konfiguration erfordert das Hinzufügen neuer Projekte mit Speicherressourcen eine zusätzliche Konfiguration, um sicherzustellen, dass die Multitenant-Funktionalität nicht verletzt wird. Führen Sie die folgenden Schritte aus, um einem Multitenant-Cluster weitere Projekte hinzuzufügen:

1. Melden Sie sich als Speicheradministrator beim NetApp ONTAP -Cluster an.
2. Navigieren Sie zu `Storage → Storage VMs` und klicken Sie auf `Add` . Erstellen Sie eine neue SVM, die für Projekt 3 bestimmt ist. Erstellen Sie außerdem ein vsadmin-Konto, um die SVM und ihre Ressourcen zu verwalten.



1. Melden Sie sich als Clusteradministrator beim Red Hat OpenShift-Cluster an.
2. Erstellen Sie ein neues Projekt.

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

3. Stellen Sie sicher, dass die Benutzergruppe für Projekt 3 auf IdP erstellt und mit dem OpenShift-Cluster synchronisiert wird.

```
oc get groups
```

4. Erstellen Sie die Entwicklerrolle für Projekt 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
```

Die in diesem Abschnitt bereitgestellte Rollendefinition ist nur ein Beispiel. Die Entwicklerrolle muss basierend auf den Anforderungen des Endbenutzers definiert werden.

1. Erstellen Sie RoleBinding für Entwickler in Projekt 3, indem Sie die Rolle „Entwicklerprojekt 3“ an die entsprechende Gruppe (ocp-projekt 3) in Projekt 3 binden.

```
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. Melden Sie sich als Speicheradministrator beim Red Hat OpenShift-Cluster an
3. Erstellen Sie ein Trident -Backend und ordnen Sie es dem für Projekt 3 vorgesehenen SVM zu. NetApp empfiehlt, das vsadmin-Konto des SVM zu verwenden, um das Backend mit dem SVM zu verbinden, anstatt den ONTAP Clusteradministrator zu verwenden.

```
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
```

Für dieses Beispiel verwenden wir den ontap-nas-Treiber. Verwenden Sie den entsprechenden Treiber zum Erstellen des Backends basierend auf dem Anwendungsfall.

Wir gehen davon aus, dass Trident im Trident-Projekt installiert ist.

1. Erstellen Sie die Speicherklasse für Projekt 3 und konfigurieren Sie sie so, dass die für Projekt 3 vorgesehenen Speicherpools vom Backend verwendet werden.

```
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. Erstellen Sie ein ResourceQuota, um Ressourcen in Projekt 3 einzuschränken, die Speicher von Speicherklassen anfordern, die anderen Projekten gewidmet sind.

```
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. Patchen Sie die ResourceQuotas in anderen Projekten, um den Ressourcen in diesen Projekten den Zugriff auf Speicher aus der für Projekt 3 vorgesehenen Speicherklasse zu untersagen.

```
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}}}'
```

Erweitertes Cluster-Management für Kubernetes

Erweitertes Cluster-Management für Kubernetes: Red Hat OpenShift mit NetApp – Übersicht

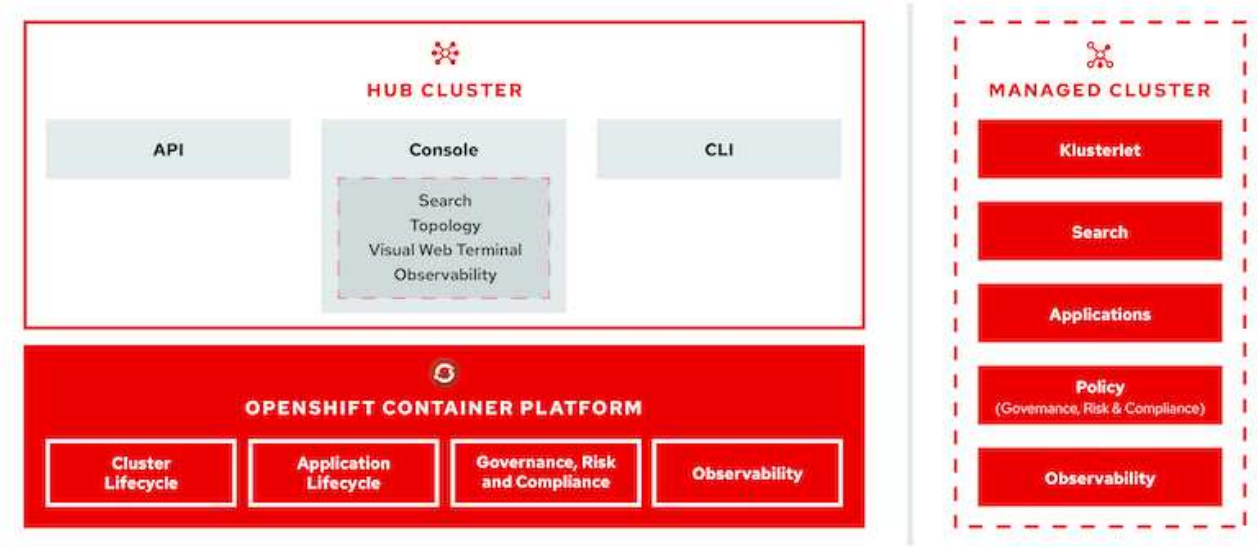
Beim Übergang einer containerisierten Anwendung von der Entwicklung zur Produktion benötigen viele Organisationen mehrere Red Hat OpenShift-Cluster, um das Testen und Bereitstellen dieser Anwendung zu unterstützen. In Verbindung damit hosten Organisationen normalerweise mehrere Anwendungen oder Workloads auf OpenShift-Clustern. Daher muss jede Organisation letztendlich eine Reihe von Clustern verwalten, und OpenShift-Administratoren stehen daher vor der zusätzlichen Herausforderung, mehrere Cluster in einer Reihe von Umgebungen zu verwalten und zu warten, die sich über mehrere lokale Rechenzentren und öffentliche Clouds erstrecken. Um diese Herausforderungen zu bewältigen, hat Red Hat Advanced Cluster Management für Kubernetes eingeführt.

Mit Red Hat Advanced Cluster Management für Kubernetes können Sie die folgenden Aufgaben ausführen:

1. Erstellen, importieren und verwalten Sie mehrere Cluster über Rechenzentren und öffentliche Clouds hinweg
2. Bereitstellen und Verwalten von Anwendungen oder Workloads auf mehreren Clustern über eine einzige Konsole

3. Überwachen und analysieren Sie den Zustand und Status verschiedener Clusterressourcen
4. Überwachen und erzwingen Sie die Einhaltung der Sicherheitsvorschriften über mehrere Cluster hinweg

Red Hat Advanced Cluster Management für Kubernetes wird als Add-on zu einem Red Hat OpenShift-Cluster installiert und verwendet diesen Cluster als zentralen Controller für alle seine Vorgänge. Dieser Cluster wird als Hub-Cluster bezeichnet und stellt eine Verwaltungsebene bereit, über die die Benutzer eine Verbindung zum erweiterten Cluster-Management herstellen können. Alle anderen OpenShift-Cluster, die entweder importiert oder über die Advanced Cluster Management-Konsole erstellt werden, werden vom Hub-Cluster verwaltet und als verwaltete Cluster bezeichnet. Es installiert einen Agenten namens Klusterlet auf den verwalteten Clustern, um sie mit dem Hub-Cluster zu verbinden und die Anforderungen für verschiedene Aktivitäten im Zusammenhang mit Cluster-Lebenszyklusmanagement, Anwendungs-Lebenszyklusmanagement, Beobachtbarkeit und Sicherheitskonformität zu erfüllen.



Weitere Informationen finden Sie in der Dokumentation "hier," .

ACM für Kubernetes bereitstellen

Bereitstellen der erweiterten Clusterverwaltung für Kubernetes

Dieser Abschnitt behandelt die erweiterte Clusterverwaltung für Kubernetes auf Red Hat OpenShift mit NetApp.

Voraussetzungen

1. Ein Red Hat OpenShift-Cluster (höher als Version 4.5) für den Hub-Cluster
2. Red Hat OpenShift-Cluster (höher als Version 4.4.3) für verwaltete Cluster
3. Cluster-Admin-Zugriff auf den Red Hat OpenShift-Cluster
4. Ein Red Hat-Abonnement für Advanced Cluster Management für Kubernetes

Advanced Cluster Management ist ein Add-on für den OpenShift-Cluster. Daher gelten bestimmte Anforderungen und Einschränkungen hinsichtlich der Hardwareressourcen, die auf den im Hub und in den verwalteten Clustern verwendeten Funktionen basieren. Sie müssen diese Probleme bei der Dimensionierung

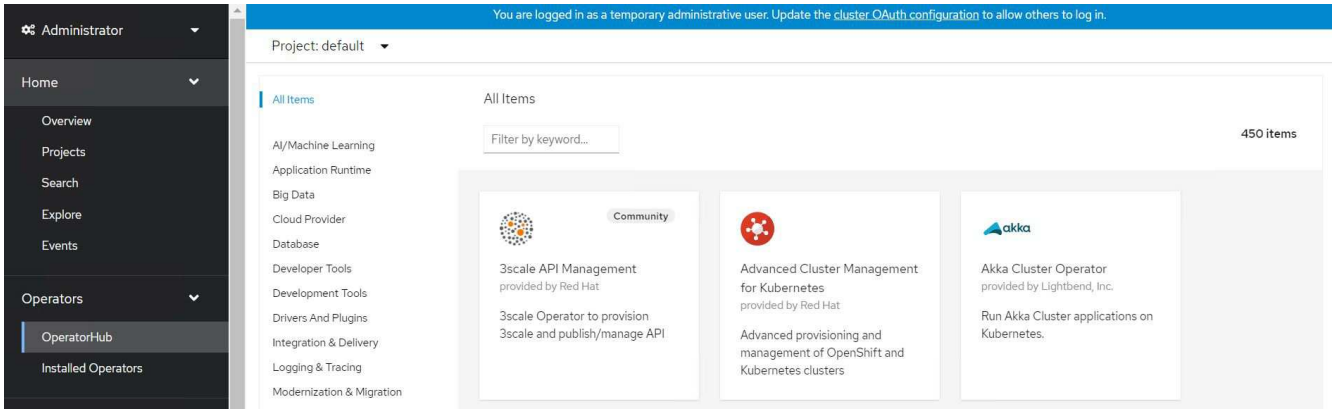
der Cluster berücksichtigen. Siehe die Dokumentation "hier," für weitere Details.

Wenn der Hub-Cluster über dedizierte Knoten zum Hosten von Infrastrukturkomponenten verfügt und Sie die Ressourcen der erweiterten Clusterverwaltung nur auf diesen Knoten installieren möchten, müssen Sie diesen Knoten optional entsprechende Toleranzen und Selektoren hinzufügen. Weitere Einzelheiten finden Sie in der Dokumentation "hier," .

Bereitstellen der erweiterten Clusterverwaltung für Kubernetes

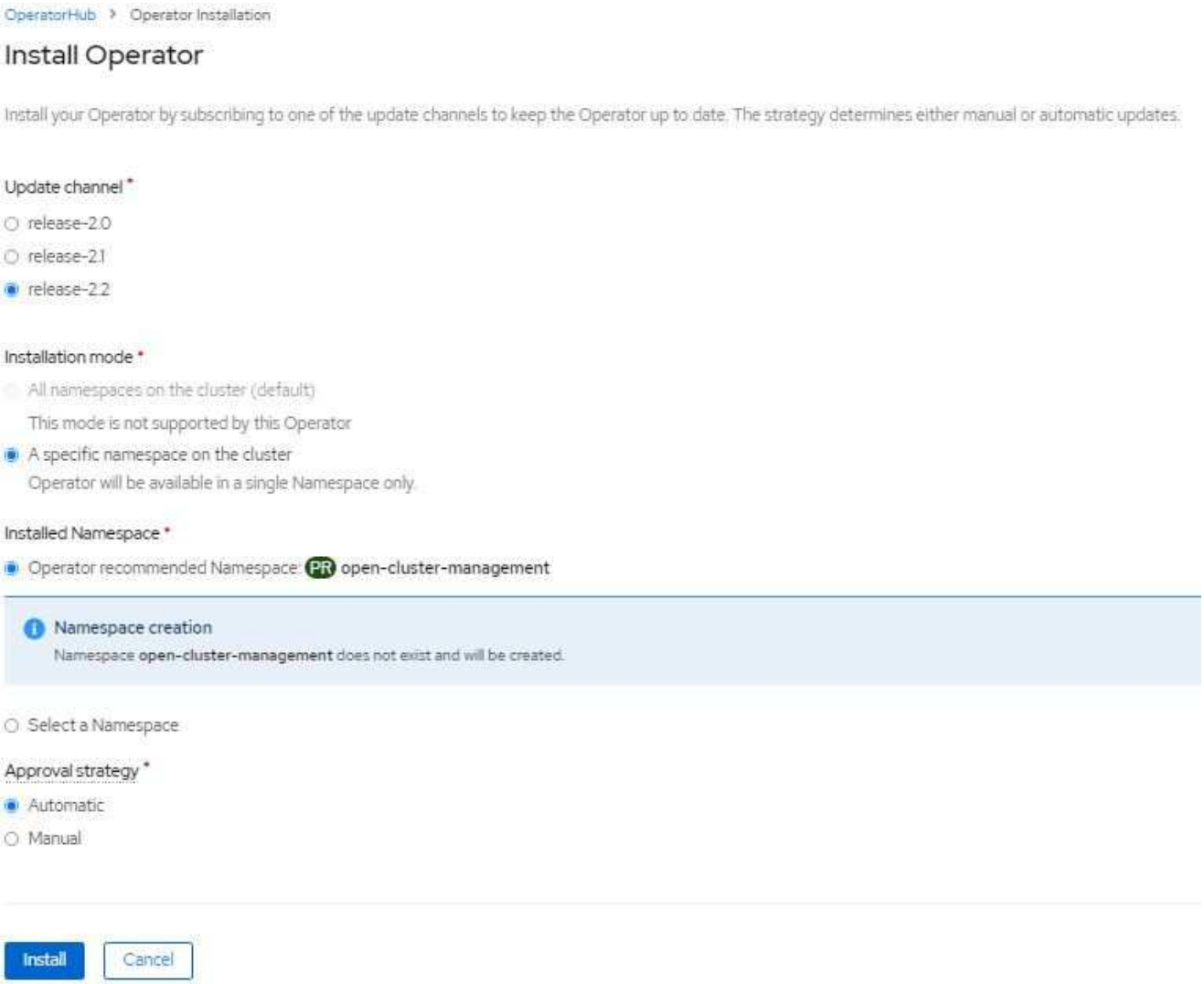
Führen Sie die folgenden Schritte aus, um Advanced Cluster Management für Kubernetes auf einem OpenShift-Cluster zu installieren:

1. Wählen Sie einen OpenShift-Cluster als Hub-Cluster und melden Sie sich mit Cluster-Admin-Berechtigungen an.
2. Navigieren Sie zu Operatoren > Operatoren-Hub und suchen Sie nach „Erweiterte Clusterverwaltung für Kubernetes“.

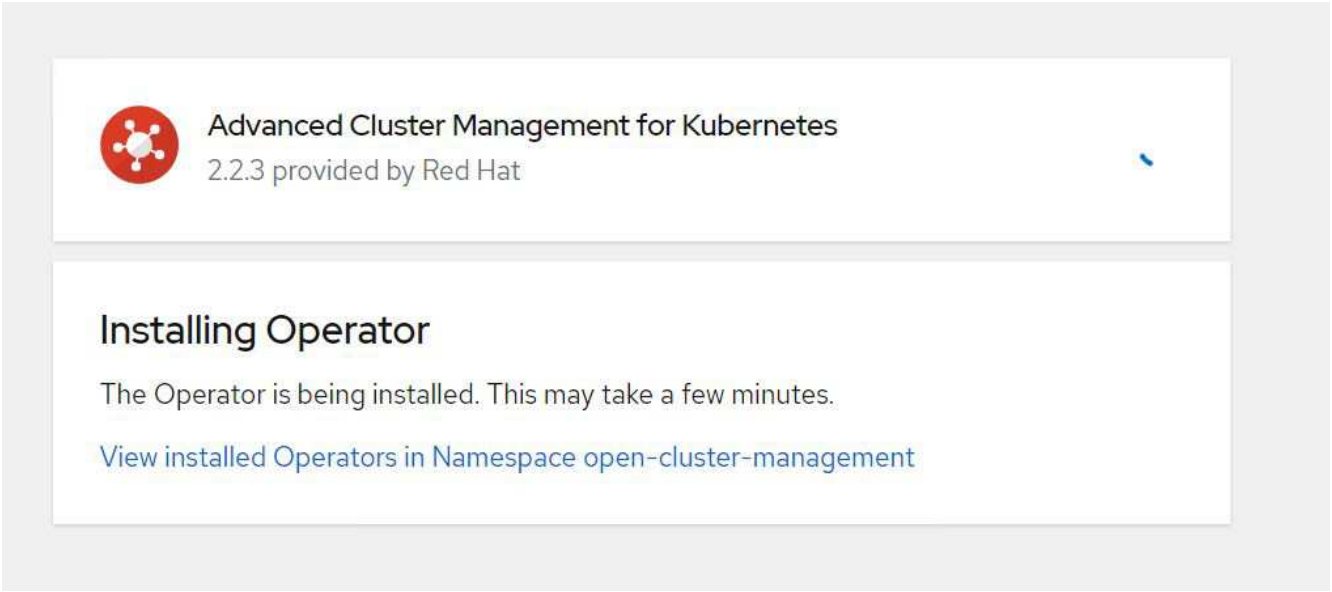


3. Wählen Sie „Erweiterte Clusterverwaltung für Kubernetes“ und klicken Sie auf „Installieren“.

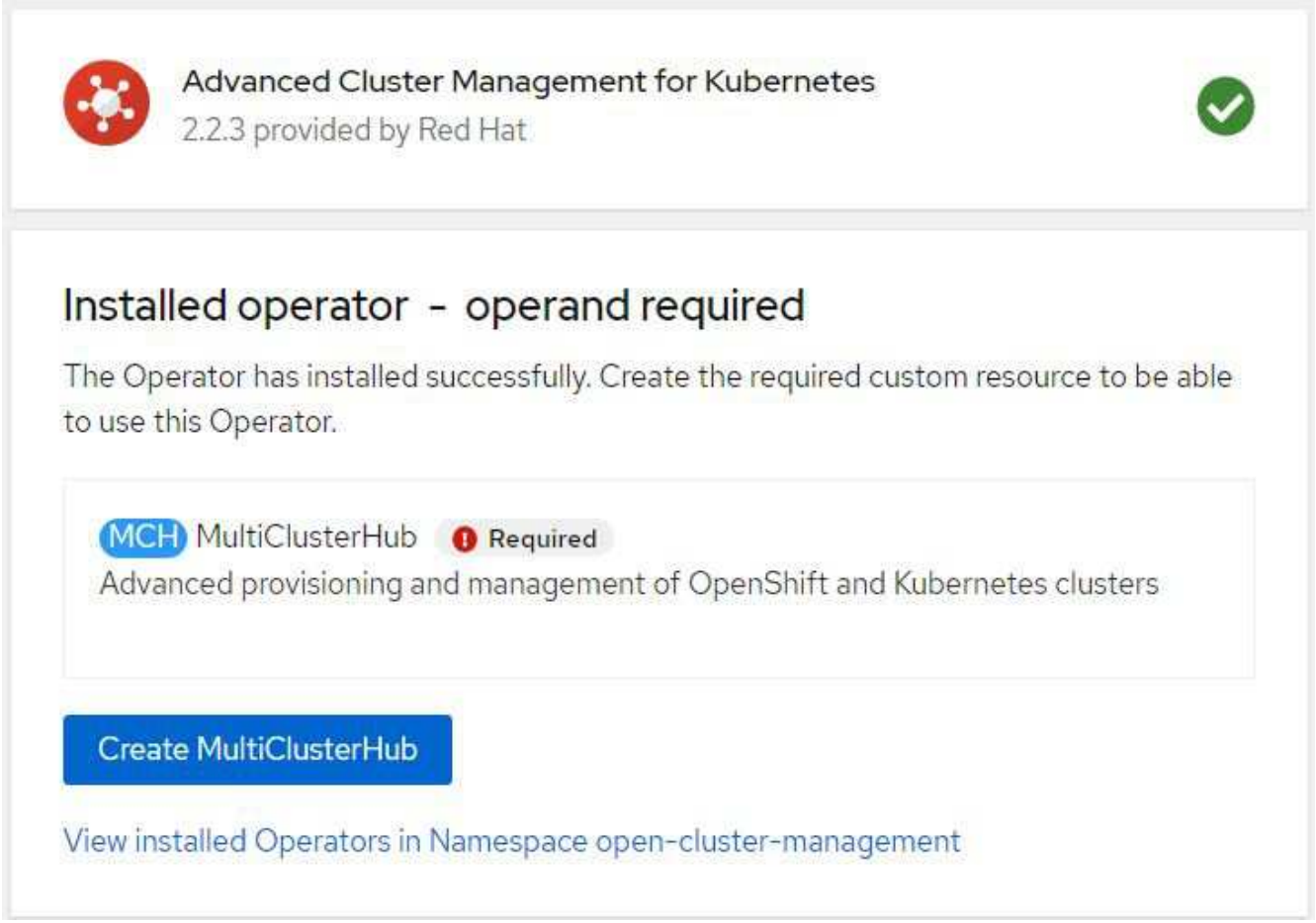
4. Geben Sie auf dem Bildschirm „Operator installieren“ die erforderlichen Details ein (NetApp empfiehlt, die Standardparameter beizubehalten) und klicken Sie auf „Installieren“.



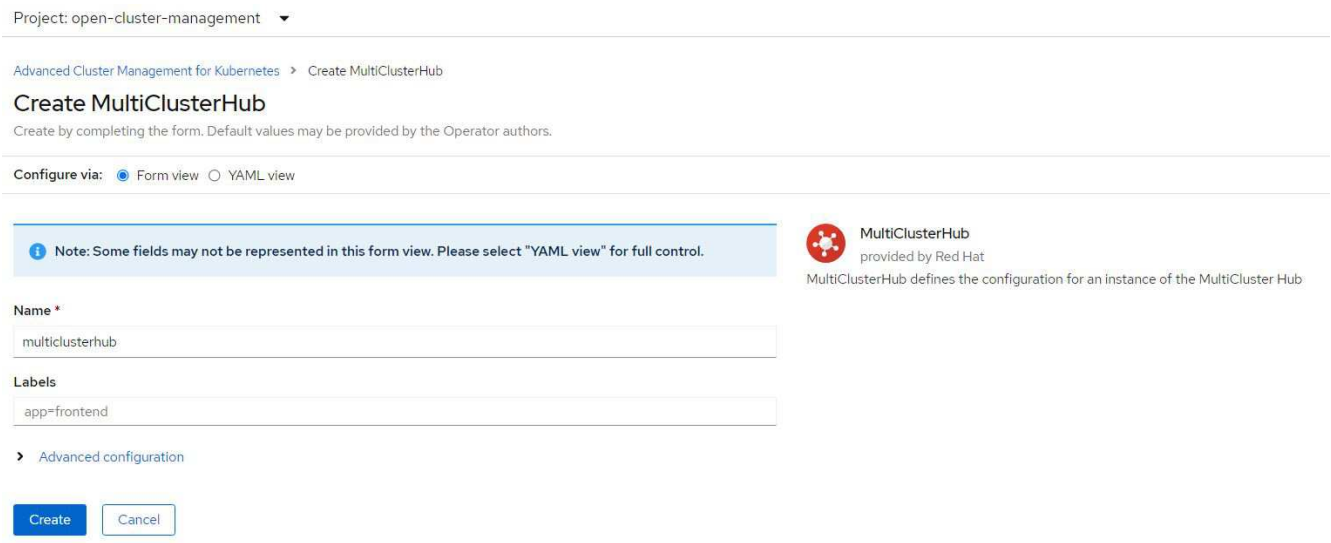
5. Warten Sie, bis die Operatorinstallation abgeschlossen ist.



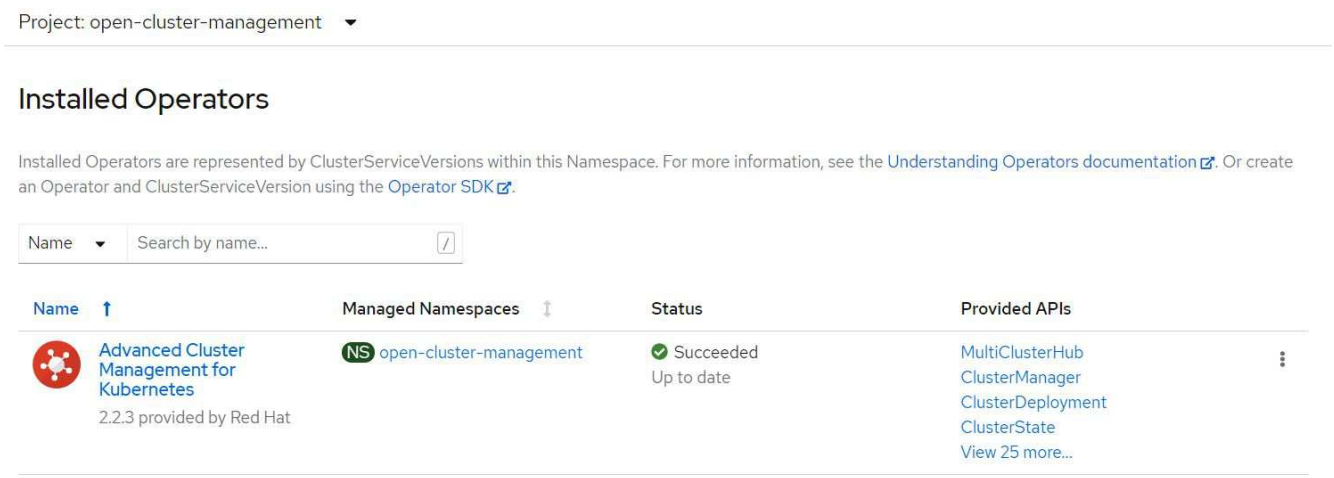
6. Klicken Sie nach der Installation des Operators auf „MultiClusterHub erstellen“.



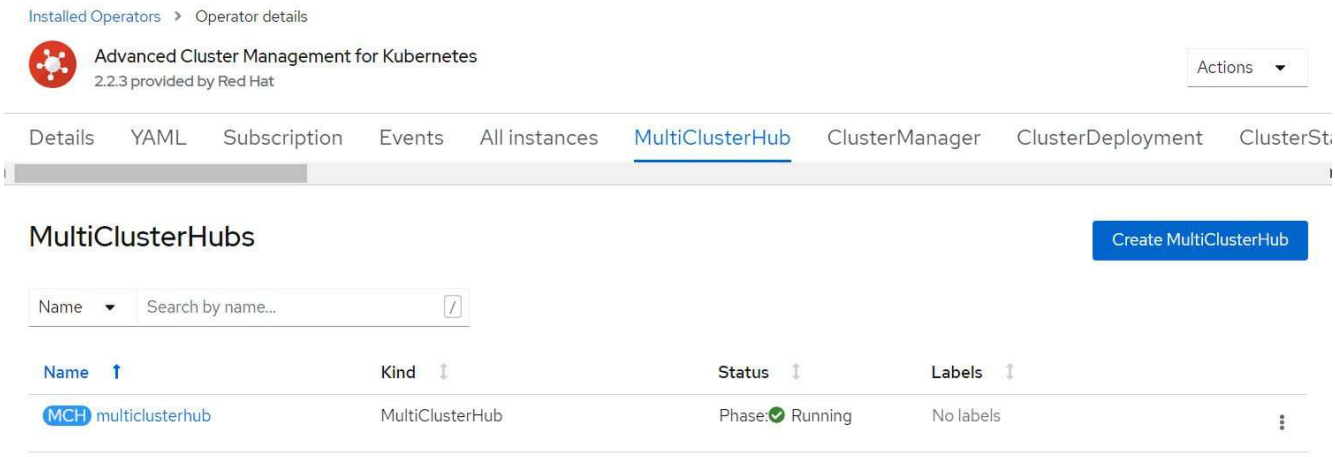
7. Klicken Sie auf dem Bildschirm „MultiClusterHub erstellen“ auf „Erstellen“, nachdem Sie die Details angegeben haben. Dadurch wird die Installation eines Multi-Cluster-Hubs eingeleitet.



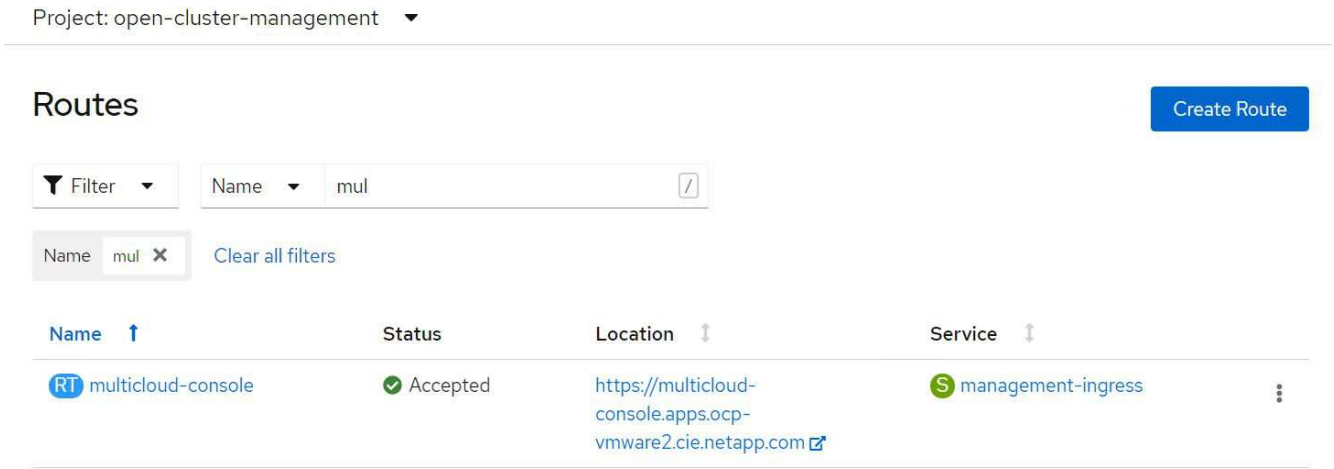
8. Nachdem alle Pods im Open-Cluster-Management-Namespace in den Status „Ausgeführt“ und der Operator in den Status „Erfolgreich“ gewechselt sind, wird Advanced Cluster Management für Kubernetes installiert.



9. Es dauert einige Zeit, bis die Hub-Installation abgeschlossen ist. Danach wechselt der MultiCluster-Hub in den Status „Wird ausgeführt“.



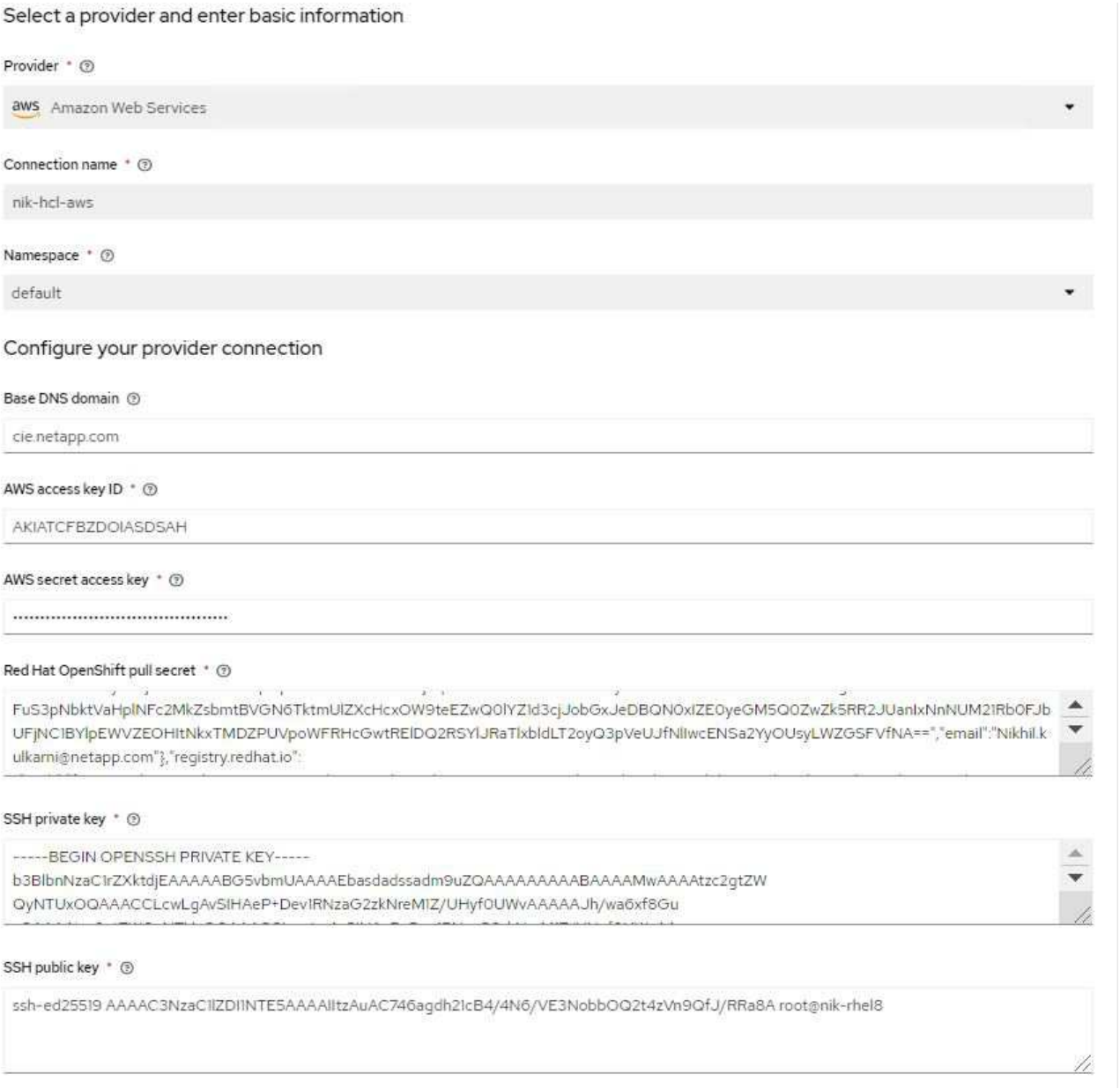
10. Es erstellt eine Route im Open-Cluster-Management-Namespace. Stellen Sie eine Verbindung mit der URL in der Route her, um auf die Advanced Cluster Management-Konsole zuzugreifen.



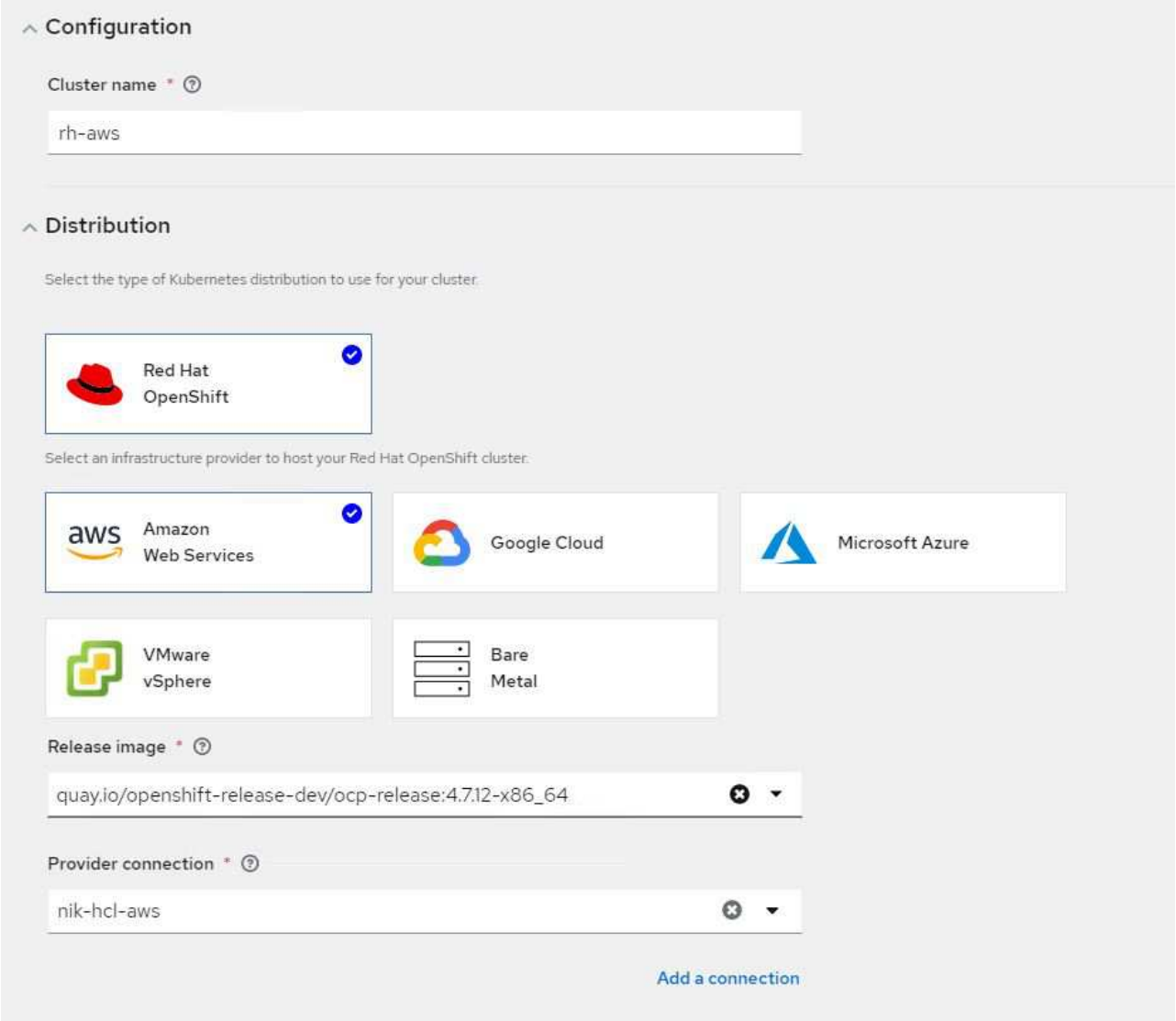
Cluster-Lebenszyklusverwaltung

Um verschiedene OpenShift-Cluster zu verwalten, können Sie diese entweder erstellen oder in Advanced Cluster Management importieren.

1. Navigieren Sie zunächst zu „Infrastrukturen automatisieren“ > „Cluster“.
2. Führen Sie die folgenden Schritte aus, um einen neuen OpenShift-Cluster zu erstellen:
   a. Erstellen Sie eine Providerverbindung: Navigieren Sie zu „Providerverbindungen“, klicken Sie auf „Verbindung hinzufügen“, geben Sie alle Details zum ausgewählten Providertyp ein und klicken Sie auf „Hinzufügen“.



   b. Um einen neuen Cluster zu erstellen, navigieren Sie zu Cluster und klicken Sie auf Cluster hinzufügen > Cluster erstellen. Geben Sie die Details für den Cluster und den entsprechenden Anbieter ein und klicken Sie auf „Erstellen“.



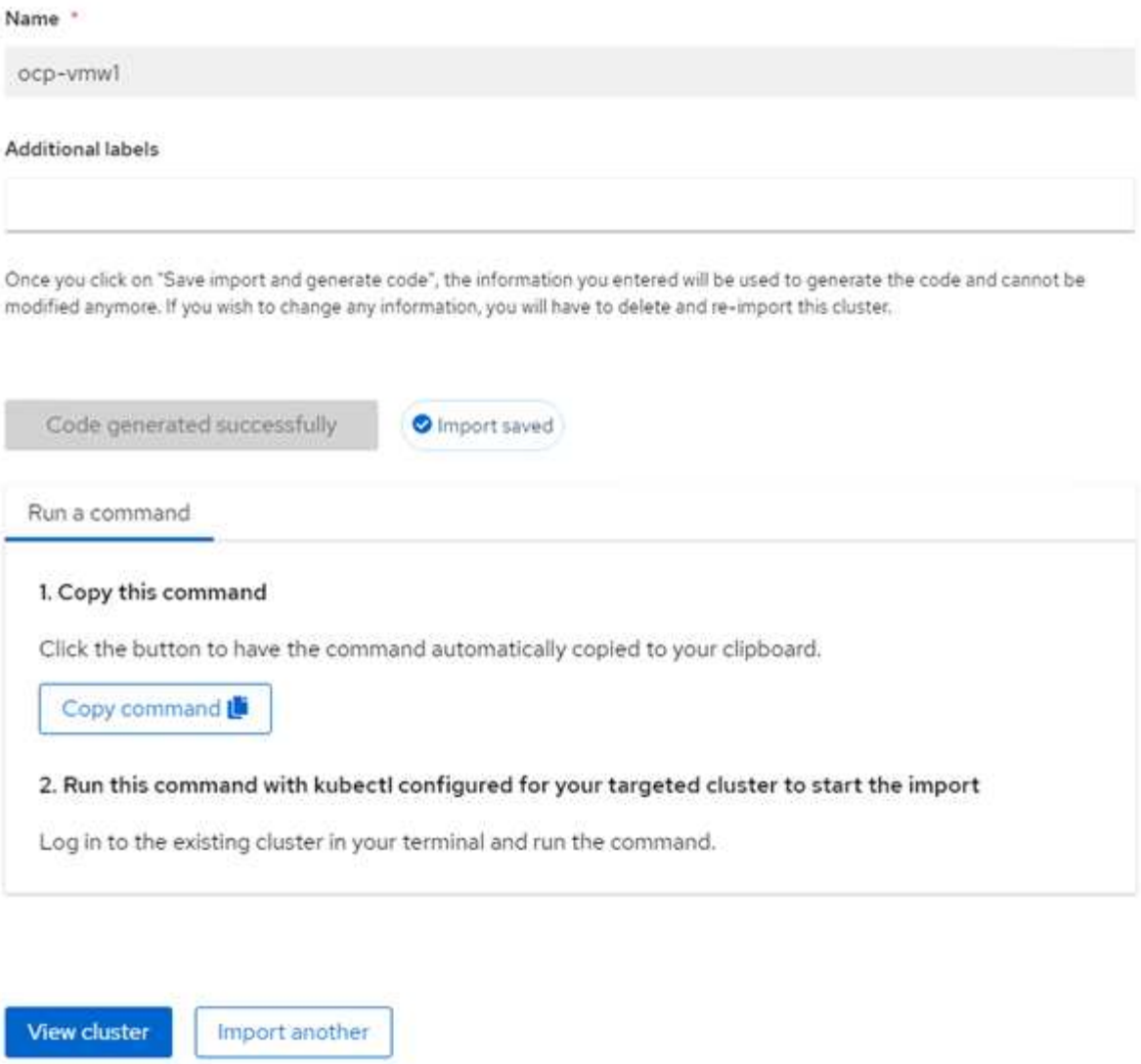
c. Nachdem der Cluster erstellt wurde, wird er in der Clusterliste mit dem Status „Bereit“ angezeigt.

3. Führen Sie die folgenden Schritte aus, um einen vorhandenen Cluster zu importieren:

   a. Navigieren Sie zu „Cluster“ und klicken Sie auf „Cluster hinzufügen“ > „Vorhandenen Cluster importieren“.

   b. Geben Sie den Namen des Clusters ein und klicken Sie auf „Import speichern und Code generieren“. Es wird ein Befehl zum Hinzufügen des vorhandenen Clusters angezeigt.

   c. Klicken Sie auf „Befehl kopieren“ und führen Sie den Befehl auf dem Cluster aus, der dem Hub-Cluster hinzugefügt werden soll. Dadurch wird die Installation der erforderlichen Agenten auf dem Cluster eingeleitet. Nach Abschluss dieses Vorgangs wird der Cluster in der Clusterliste mit dem Status „Bereit“ angezeigt.

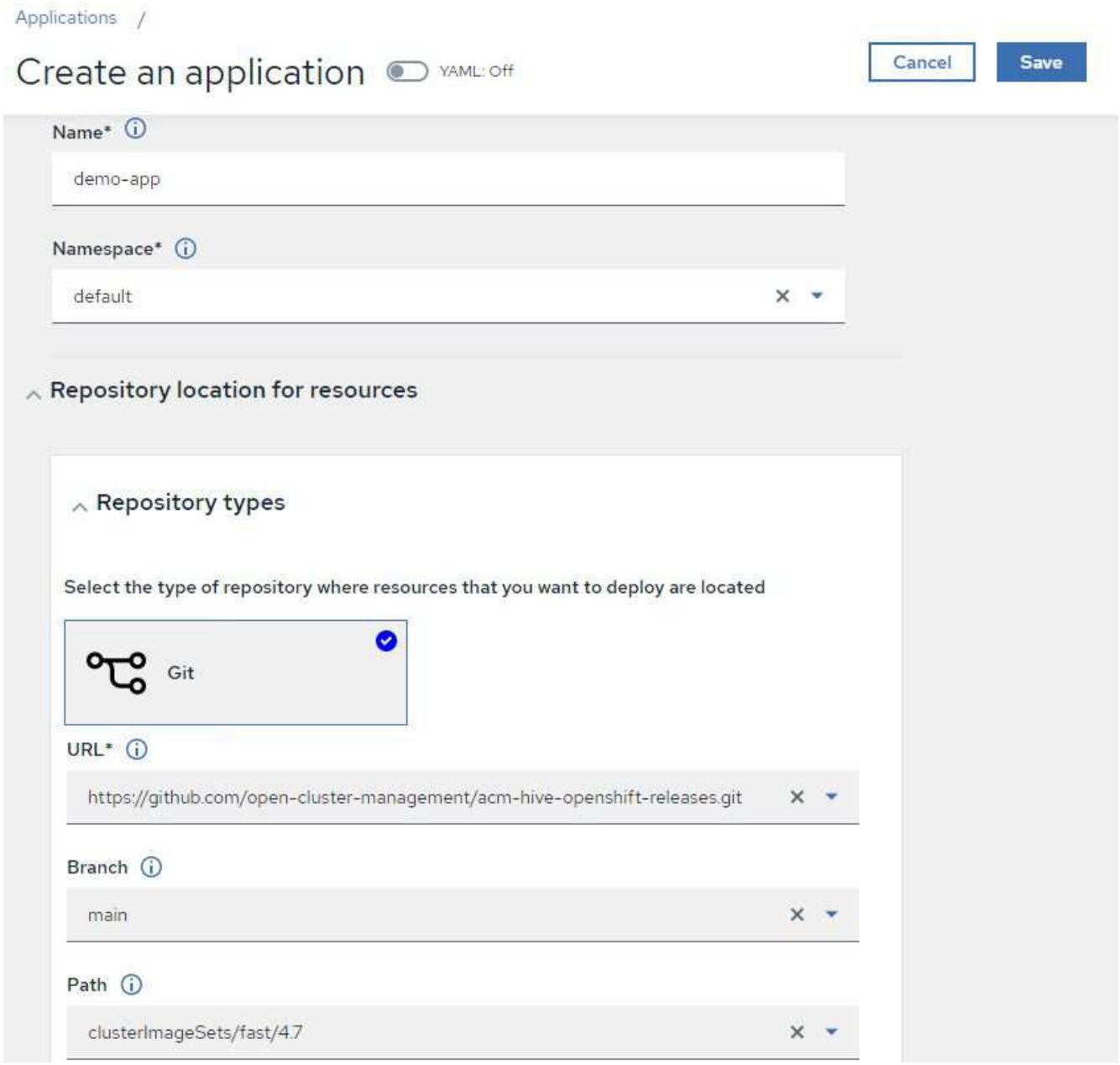


4. Nachdem Sie mehrere Cluster erstellt und importiert haben, können Sie sie von einer einzigen Konsole aus überwachen und verwalten.

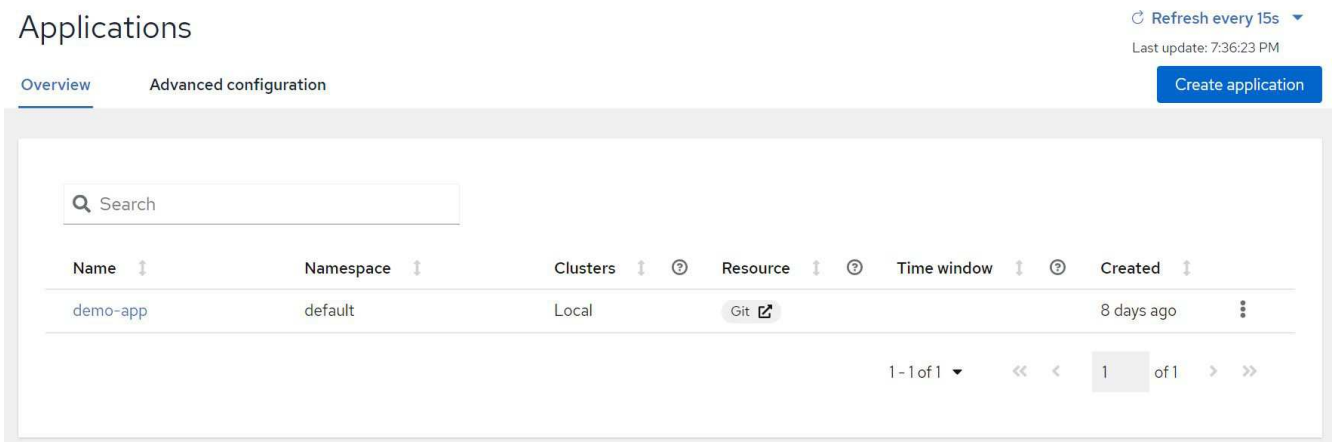
Anwendungslebenszyklusverwaltung

So erstellen Sie eine Anwendung und verwalten sie über mehrere Cluster hinweg:

1. Navigieren Sie in der Seitenleiste zu „Anwendungen verwalten“ und klicken Sie auf „Anwendung erstellen“. Geben Sie die Details der Anwendung ein, die Sie erstellen möchten, und klicken Sie auf „Speichern“.



2. Nachdem die Anwendungskomponenten installiert wurden, wird die Anwendung in der Liste angezeigt.

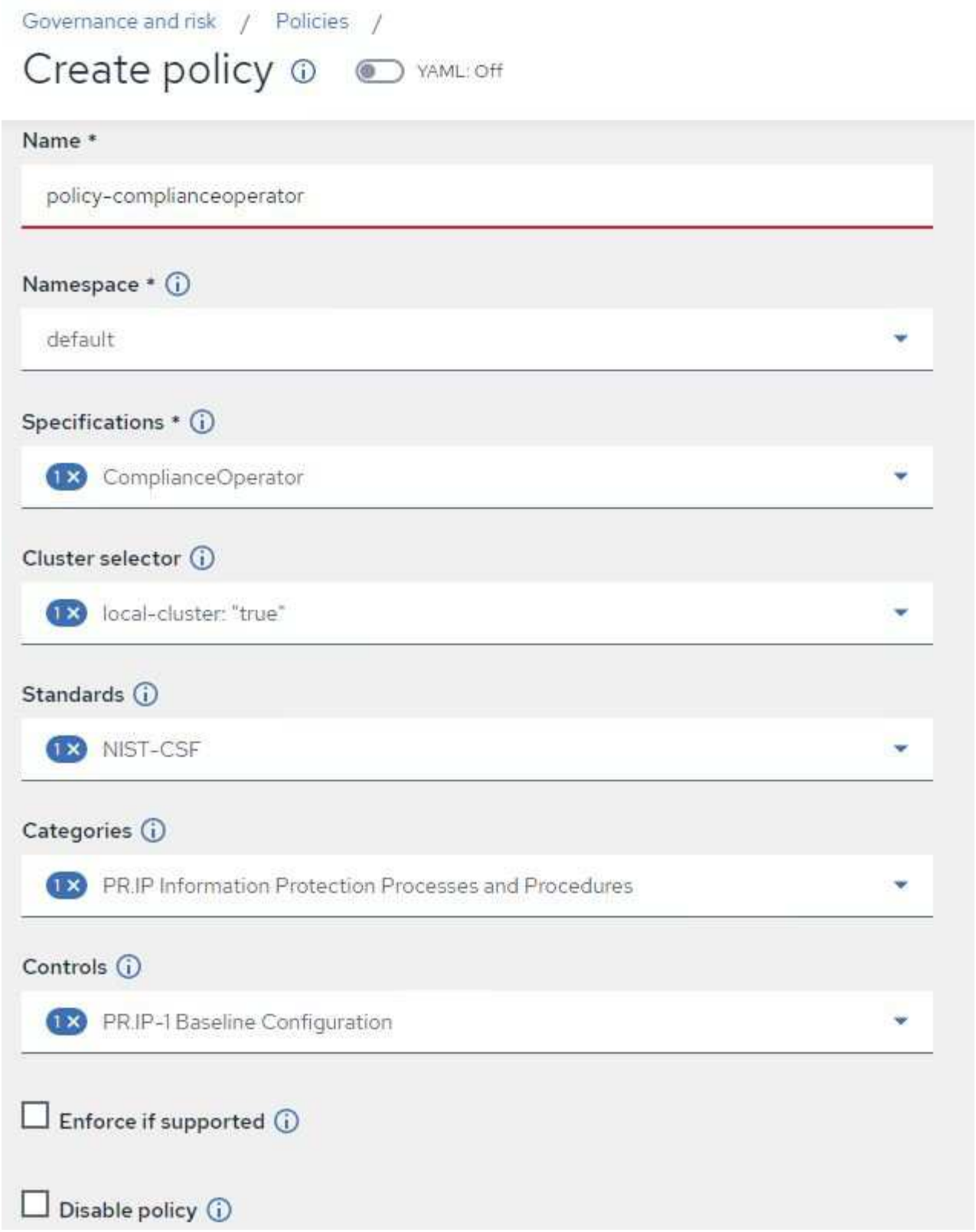


3. Die Anwendung kann jetzt von der Konsole aus überwacht und verwaltet werden.

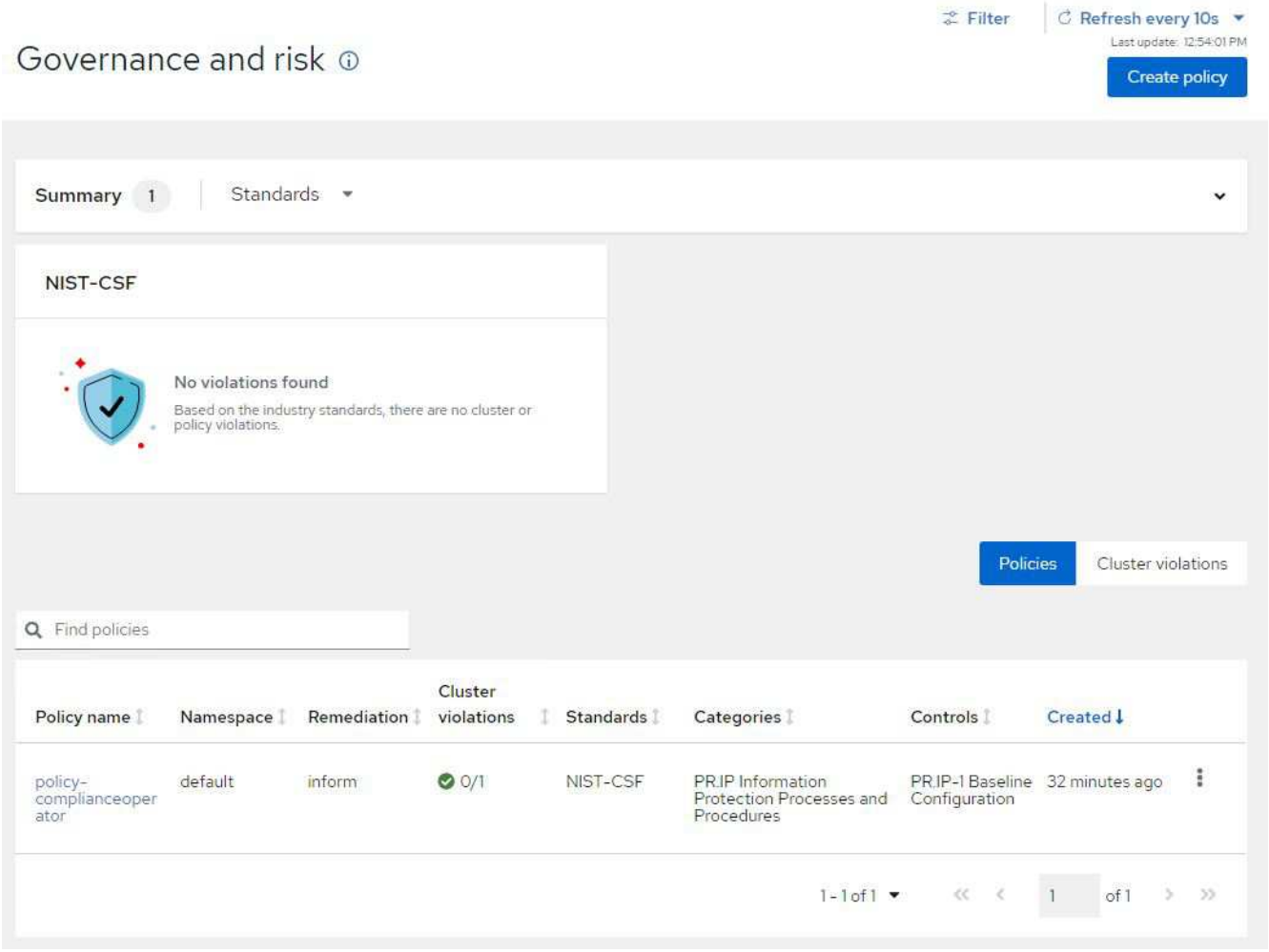
Governance und Risiko

Mit dieser Funktion können Sie die Compliance-Richtlinien für verschiedene Cluster definieren und sicherstellen, dass die Cluster diese einhalten. Sie können die Richtlinien so konfigurieren, dass Abweichungen oder Verstöße gegen die Regeln entweder gemeldet oder behoben werden.

1. Navigieren Sie in der Seitenleiste zu „Governance und Risiko“.
2. Um Konformitätsrichtlinien zu erstellen, klicken Sie auf „Richtlinie erstellen“, geben Sie die Details der Richtlinienstandards ein und wählen Sie die Cluster aus, die dieser Richtlinie entsprechen sollen. Wenn Sie Verstöße gegen diese Richtlinie automatisch beheben möchten, aktivieren Sie das Kontrollkästchen „Erzwingen, falls unterstützt“ und klicken Sie auf „Erstellen“.



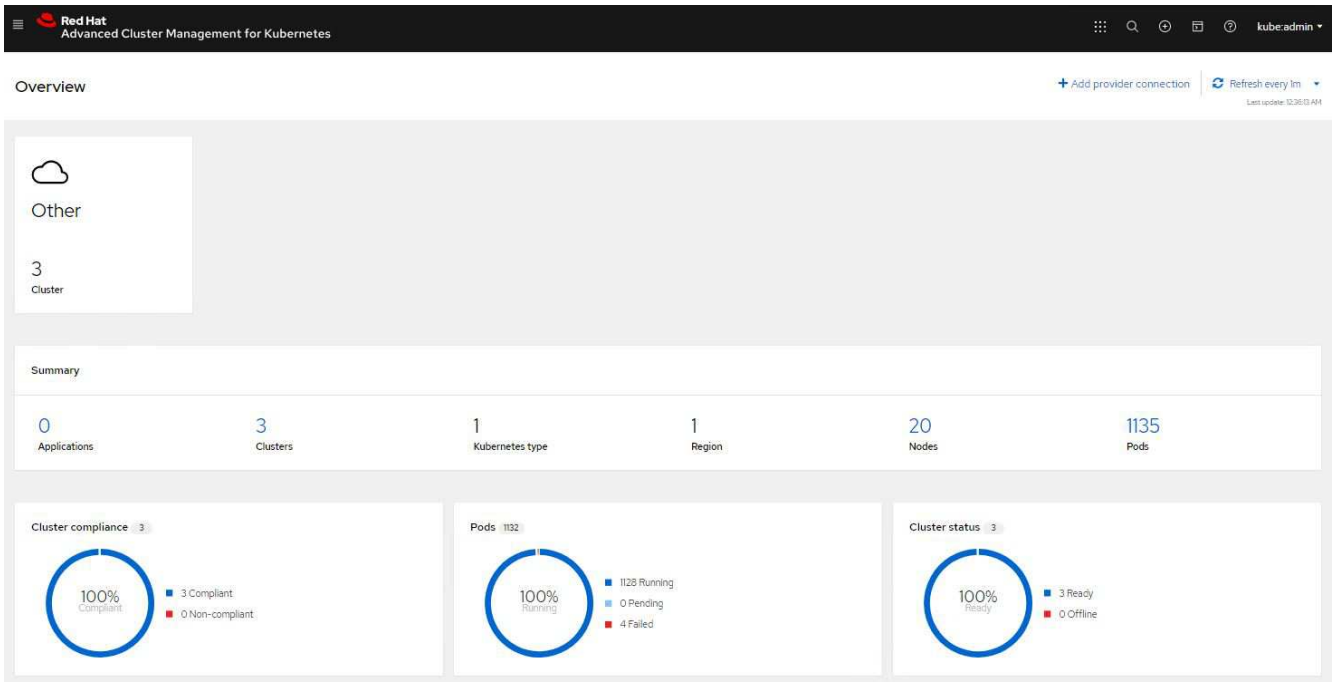
3. Nachdem alle erforderlichen Richtlinien konfiguriert wurden, können alle Richtlinien- oder Clusterverletzungen über die erweiterte Clusterverwaltung überwacht und behoben werden.



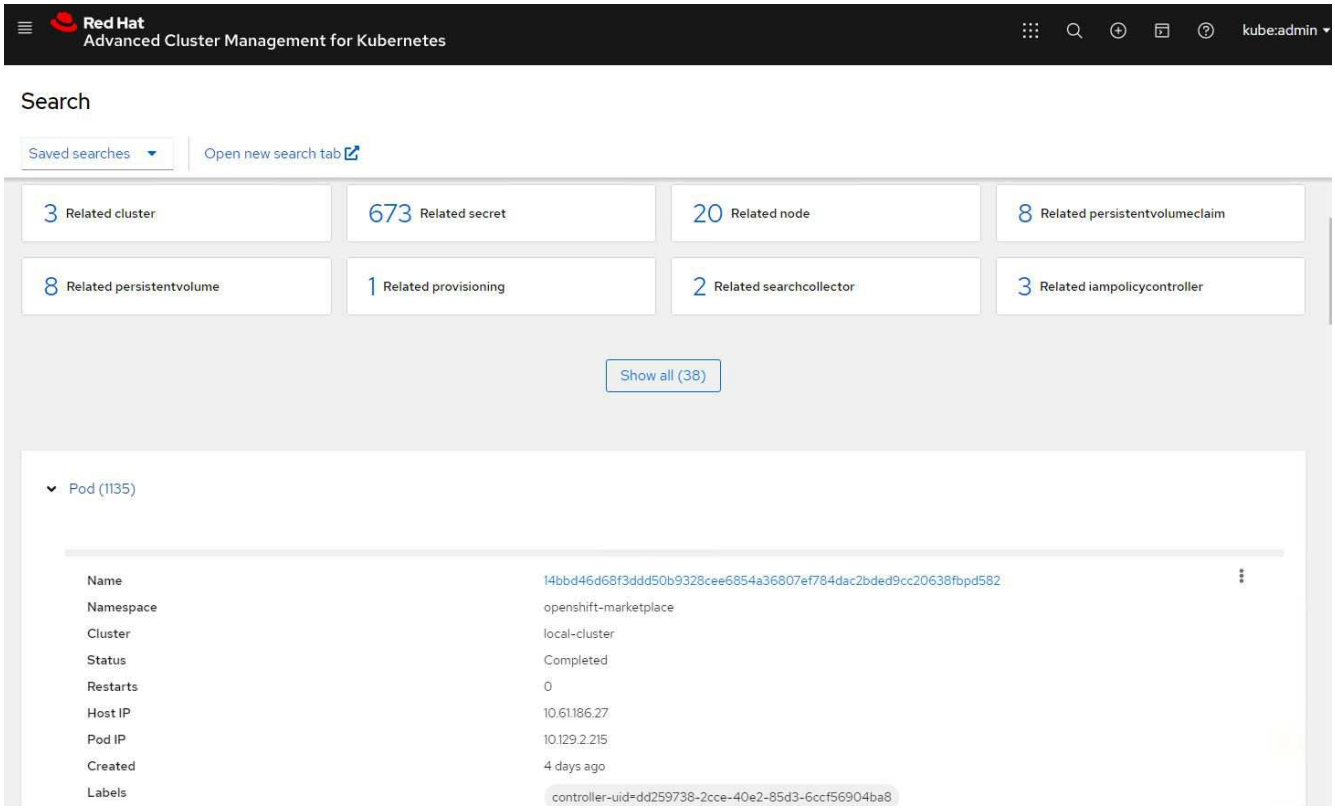
Beobachtbarkeit

Advanced Cluster Management für Kubernetes bietet eine Möglichkeit, die Knoten, Pods und Anwendungen sowie die Workloads aller Cluster zu überwachen.

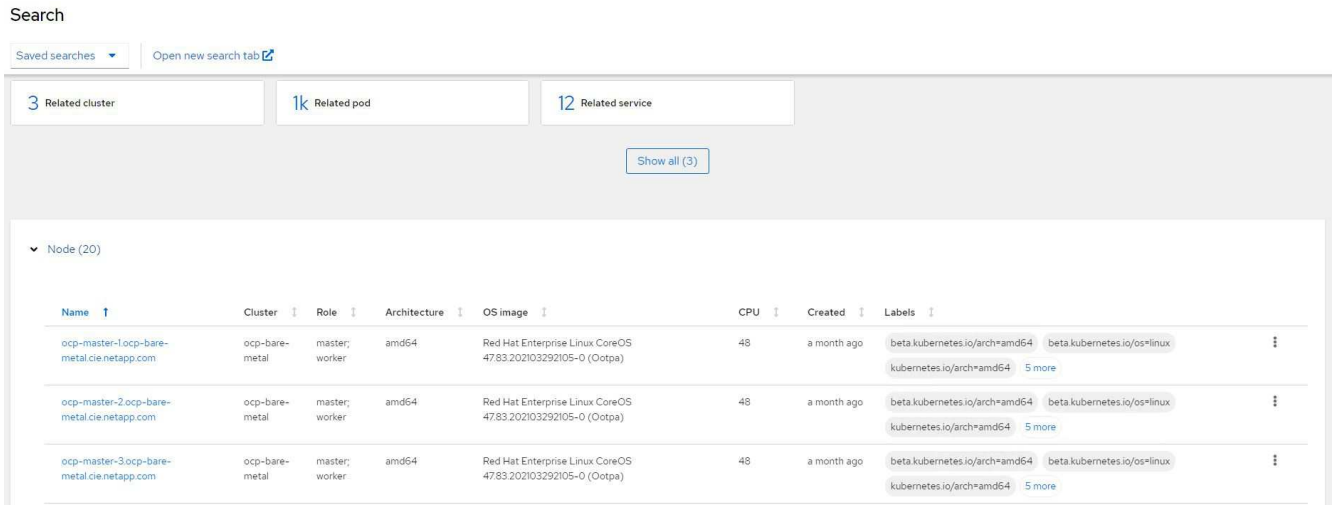
1. Navigieren Sie zu „Umgebungen beobachten“ > „Übersicht“.



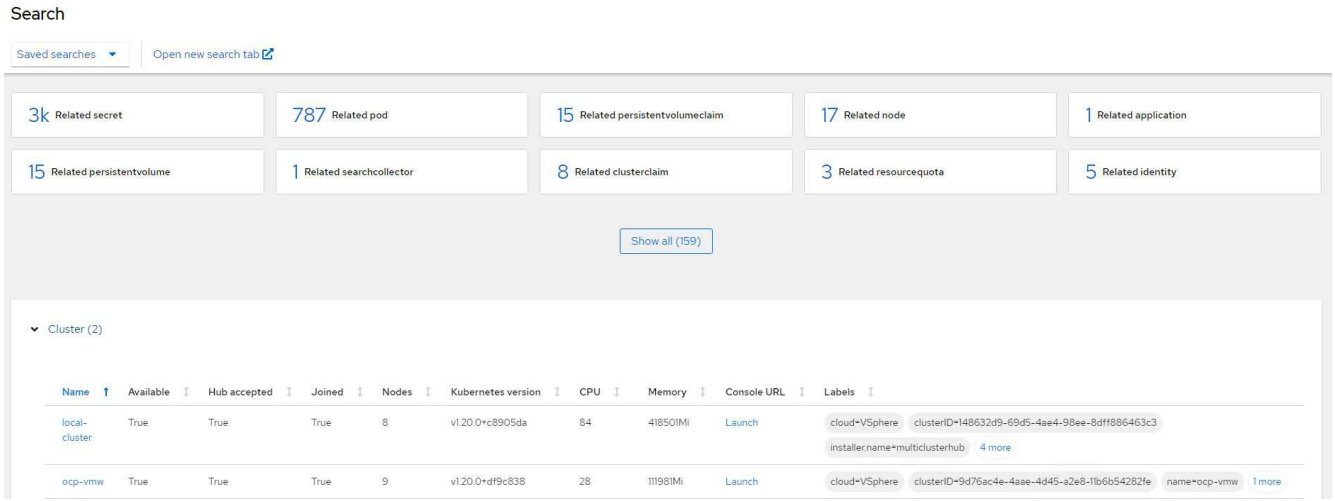
2. Alle Pods und Workloads in allen Clustern werden überwacht und anhand verschiedener Filter sortiert. Klicken Sie auf Pods, um die entsprechenden Daten anzuzeigen.



3. Alle Knoten in den Clustern werden anhand einer Vielzahl von Datenpunkten überwacht und analysiert. Klicken Sie auf Knoten, um mehr Einblick in die entsprechenden Details zu erhalten.

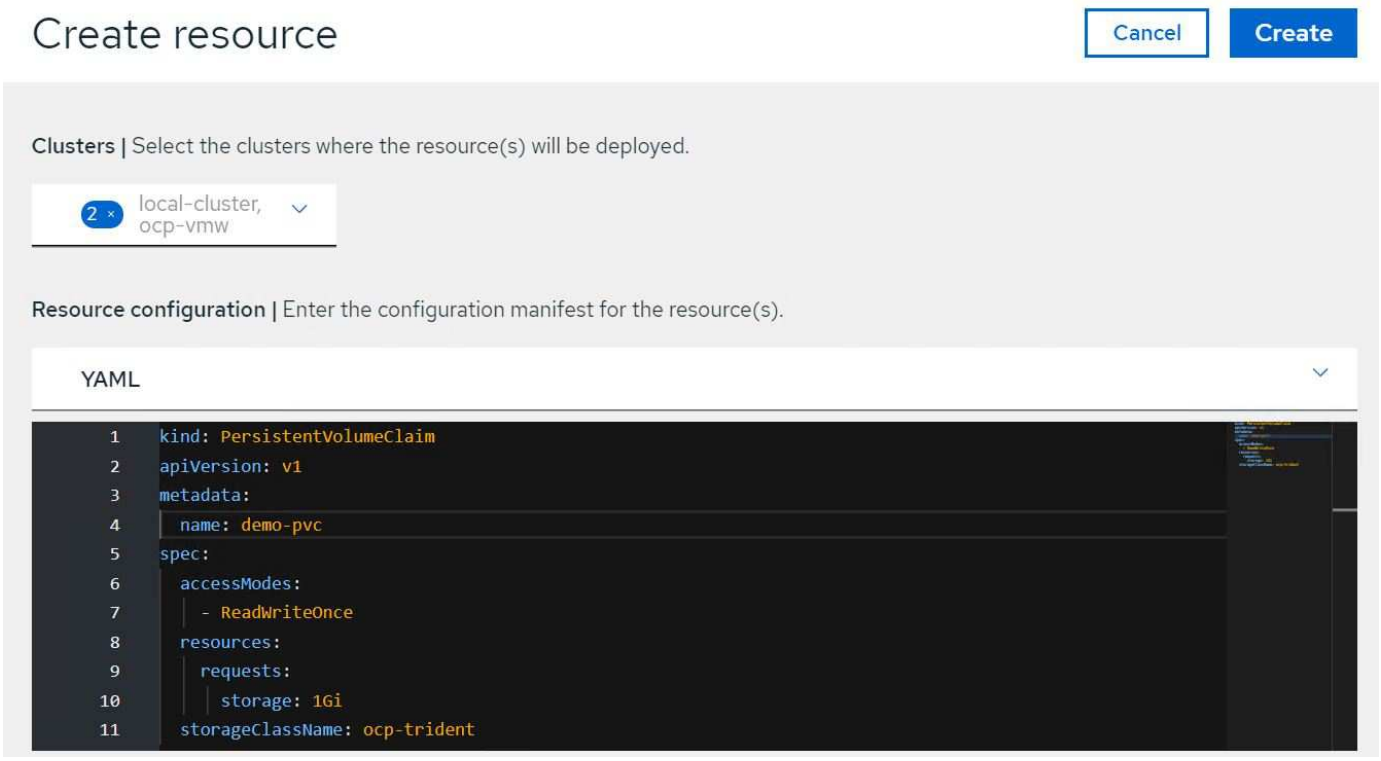


4. Alle Cluster werden basierend auf verschiedenen Clusterressourcen und -parametern überwacht und organisiert. Klicken Sie auf „Cluster“, um Clusterdetails anzuzeigen.



Erstellen von Ressourcen auf mehreren Clustern

Mit Advanced Cluster Management für Kubernetes können Benutzer über die Konsole Ressourcen auf einem oder mehreren verwalteten Clustern gleichzeitig erstellen. Wenn Sie beispielsweise OpenShift-Cluster an verschiedenen Standorten haben, die von verschiedenen NetApp ONTAP -Clustern unterstützt werden, und PVCs an beiden Standorten bereitstellen möchten, können Sie auf das (+)-Zeichen in der oberen Leiste klicken. Wählen Sie dann die Cluster aus, auf denen Sie den PVC erstellen möchten, fügen Sie die YAML-Ressource ein und klicken Sie auf „Erstellen“.



Datenschutz für Container-Apps und VMs mit Trident Protect

Diese Lösung zeigt, wie Sie mit Trident Protect Datenschutzvorgänge für Container und VMs durchführen.

1. Weitere Informationen zum Erstellen von Snapshots und Backups sowie zur Wiederherstellung daraus für Containeranwendungen in der OpenShift Container-Plattform finden Sie unter"hier," .
2. Weitere Informationen zum Erstellen und Wiederherstellen aus einem Backup für VMs in OpenShift Virtualization, die auf der OpenShift Container-Plattform bereitgestellt werden, finden Sie unter"hier," .

Datenschutz für Container-Apps und VMs mit Tools von Drittanbietern

Diese Lösung zeigt, wie Velero, das in den OADP-Operator in der Red Hat OpenShift Container-Plattform integriert ist, verwendet wird, um Datenschutzvorgänge für Container und VMs durchzuführen.

1. Weitere Informationen zum Erstellen und Wiederherstellen einer Sicherung für Containeranwendungen in der OpenShift Container-Plattform finden Sie unter"hier," .
2. Weitere Informationen zum Erstellen und Wiederherstellen aus einem Backup für VMs in OpenShift Virtualization, die auf der OpenShift Container-Plattform bereitgestellt werden, finden Sie unter"hier," .

Weitere Ressourcen zum Thema Integration von Red Hat OpenShift Virtualization mit NetApp Storage

Greifen Sie auf zusätzliche Ressourcen zu, die weitere Informationen zur Unterstützung der Bereitstellung, Verwaltung und Optimierung von Red Hat OpenShift Virtualization mit ONTAP auf verschiedenen Plattformen und Technologien bieten.

- NetApp Dokumentation

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

- Trident -Dokumentation

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

- Red Hat OpenShift-Dokumentation

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

- Red Hat OpenStack Platform-Dokumentation

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

- Red Hat Virtualisierungsdokumentation

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

- VMware vSphere-Dokumentation

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